**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**PHẦM MỀM QUẢN LÍ KHÁCH SẠN**

**GVHD: TS.NGUYỄN THÀNH SƠN**

**SVTH :**

**NGUYỄN HOÀNG PHÚC 19110052**

**TẠ BẢO MINH 19110399**

**DƯƠNG VĂN NGỌC TÍN 19110472**

**NGUYỄN ĐĂNG THẮNG 18110369**

TP.Hồ Chí Minh, ngày 20 tháng 11 năm 2021

# CHƯƠNG 1: ĐẶC TẢ HỆ THỐNG

**I. Mô tả phần mềm:**

## Lý do xây dựng phần mềm

Quản lý khách sạn là một trong những việc phải có trong kinh doanh mô hình khách sạn. Nó giúp cho người dùng có thể đặt phòng, trả phòng, thanh toán, đồng thời quản lý cả nhân viên mà cụ thể là tiếp tân. Đây là một công việc vô cùng quan trọng cho việc kinh doanh của khách sạn và cũng đảm bảo sự thoải mái cho khách hàng khi đến thuê phòng.

Phần mềm quản lý khách sạn giúp cho những việc nêu trên được thực thi một cách nhanh chóng. Giúp cho việc tính toán giá của hóa đơn được chính xác thay vì chúng ta phải tính toán thì giờ đã có phần mềm làm thay điều đó.

Ngoài ra phần mềm Quản lý khách sạn còn cho biết tình hình của khách sạn, tình trạng các phòng thuê.

## Mục tiêu của phần mềm

Các mục tiêu chính của phần mềm:

* + Xây dựng một ứng dụng có thể lưu trữ và cập nhật thông tin khách hàng, nhân viên, phòng và dịch vụ.
  + Xây dựng một ứng dụng có chức năng cho khách hàng đặt phòng, nhận phòng và xuất hóa đơn khi khách hàng thanh toán trả phòng.
  + Chức năng lưu thông tin khách hàng kèm theo phòng và dịch vụ mà khách hàng đó thuê được và tính ra giá tiền mà khách hàng đó phải trả theo từng mục tương ứng.
  + Chức năng thống kê doanh thu đặt được trong các tháng lân cận

## Mô tả phầm mềm :

Trong một khách sạn có một số lượng phòng nhất định, có thể thêm phòng tùy vào người quản trị viên, một phòng có một loại phòng, một loại phòng có thể thuộc nhiều phòng, một phòng sẽ có một mục tình trạng phòng để người sử dụng có thể biết phòng đó đã được thuê hay vẫn còn trống. Tương tự với mục nhân viên, một nhân viên có một loại nhân viên và một loại nhân viên có thể có nhiều nhân viên. Có một bảng đặt phòng, mỗi phòng trong 1 thời điểm chỉ có thể được đặt bởi 1 khách hàng. Bảng hóa đơn gồm mã phòng, mã nhân viên tạo hóa đơn, mã khách hàng cụ thể. Ngoài ra, có mục dịch vụ cho khách hàng sử dụng khi cần thiết. Khi thuê khách sạn, khách hàng có thể thuê nhiều dịch vụ để sử dụng. Khi nhận phòng thì hóa đơn sẽ lưu lại ngày vào của khách hàng, khi trả phòng, ngày hiện tại sẽ là ngày ra của khách hàng, căn cứ vào số ngày đó để tính giá tiền phòng cho khách, tiền dịch vụ sẽ tính và cộng vào sau đó xuất ra giá tiền tổng cho khách hàng phải chi trả.

**II. Chức năng cơ bản:**

* Đầu tiên hệ thống sẽ yêu cầu người dùng đăng nhập vào hệ thống, thông tin người dùng và mật khẩu được lưu trong bảng nhân viên. Sẽ có 3 đối tương người dùng chính: lao công (Labor), tiếp tân (Receptionist)và người quản lí (Manager).
* Với người quản lí (Manager) là toàn quyền sử dụng ứng dụng và nhân viên sẽ bị hạn chế một số quyền trong ứng dụng. Trong đó:

+ Manager: Có toàn quyền chỉnh sửa,thêm xóa, tác động đến các bảng trong cơ sở dữ liệu, có quyền được truy cập vào chức năng phân tích thống kê doanh thu của khách sạn, cũng như có quyền thực hiện các thao tác chính trong khách sạn bao gồm Booking, Checkin và Checkout.

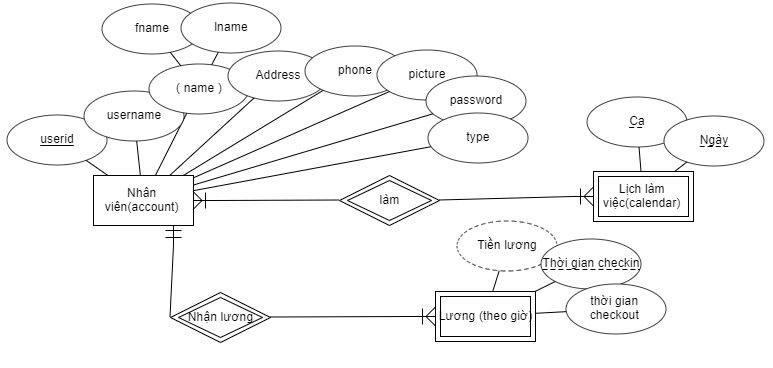
+ Receptionist: Chỉ có quyền thực hiện 4 thao tác Booking, Checkin và Checkout cho khách và Checkin và Checkout cho chính mình.

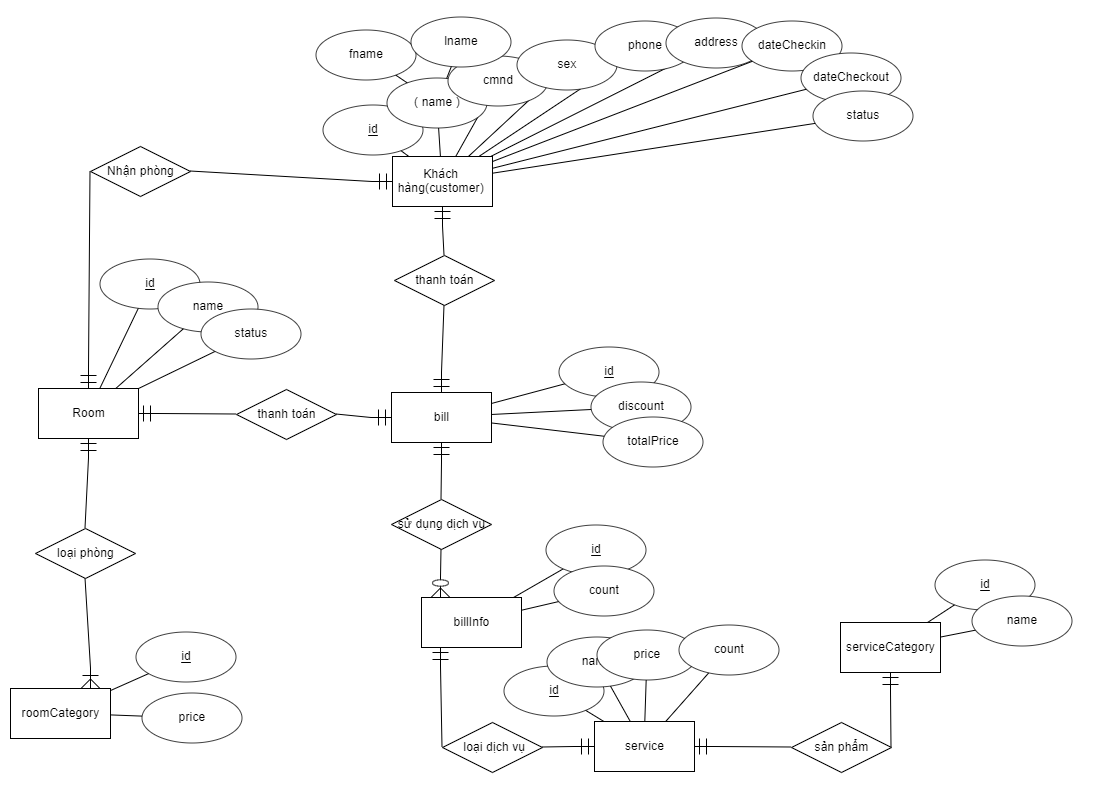
+ Labor : Chỉ có quyền thực Checkin và Checkout cho chính mình.

* Khi đăng nhập thành công, sẽ hiện ra các nút chức năng như xuất bảng phòng, bảng nhân viên, bảng dịch vị, bảng khách hàng,…
* Những chức năng cơ bản như thêm, sửa, xóa dòng trong bảng.
* Thống kê tổng doanh thu đạt được trong tháng và năm.

**III. Mô tả dữ liệu:**

1. **Mô hình ERD:**

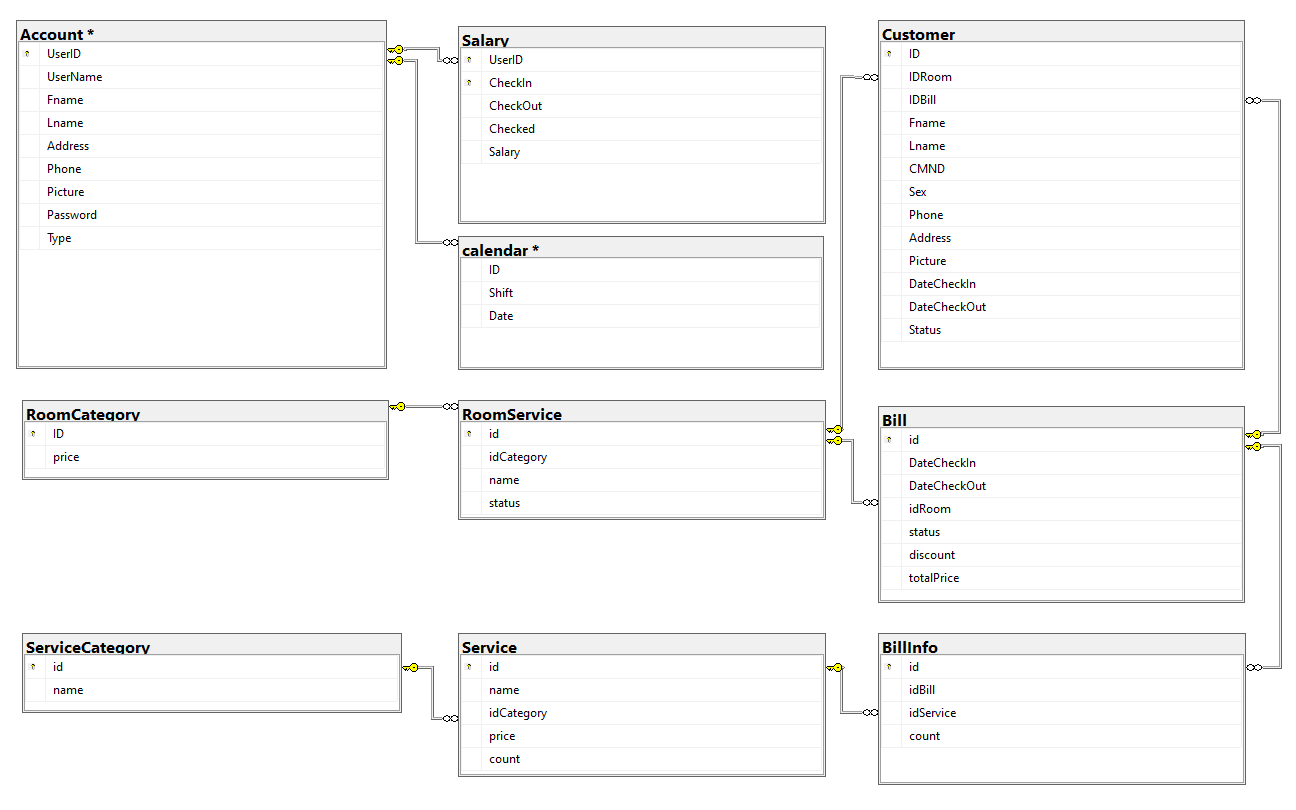




1. **Lược đồ quan hệ :**

* Account(UserID:integer,Username:string,Fname:string,Lname:string, Address:string,Phone:string,Picture:image,Password:string,Type:string)
* calendar(ID:integer, Shift:string, Date:date)
* salary(UserID:integer, Salary:real, CheckIn: date,CheckOut:date,Checked:integer)
* customer(ID:integer,IDRoom:integer,IDBill:integer,Fname:string,Lname:string,CMND:string,Sex:string,Phone:string,Address:string,Picture:image, DateCheckIn:date,DateCheckOut:Date,status:string)
* RoomService(id:integer, idCategory:integer,name:string,status:string)
* RoomCategory(ID:integer, price:real).
* Service(id:integer, name:string, idCategory:integer,price:real, count:integer)
* ServiceCategory(id:integer, name:string)
* BillInfo(id:integer,idBill,integer,idService:integer,count:integer)
* Bill(id:integer, DateCheckIn:date, DateCheckOut:date, idRoom:integer, status:string, discount:int, totalPrice:float)

**3.Mô hình Diagram:**

****

**CHƯƠNG 2: CẤU HÌNH VÀ CÀI ĐẶT HỆ THỐNG**

1. **Tạo cơ sở dữ liệu và ràng buộc (Constraints).**

Trong phần này áp dụng các ràng buộc như: unique, not null, ràng buộc khóa ngoại, ràng buộc khóa chính, đặt tên cho các ràng buộc.

* 1. **Tạo Database :**

|  |
| --- |
| CREATE DATABASE HotelManagement  GO  USE HotelManagement  GO |

* 1. **Tạo bảng account:**

|  |
| --- |
| CREATE TABLE dbo.account  (  UserID INT IDENTITY(1,1) PRIMARY KEY,  UserName NVARCHAR(100) NOT NULL UNIQUE,  Fname NVARCHAR(100) NOT NULL,  Lname NVARCHAR(100) NOT NULL,  [Address] TEXT,  Phone [nvarchar](15) NOT NULL,  Picture [image],  [Password] [nvarchar](100) NOT NULL,  [Type] [nvarchar](100) NOT NULL,  )  GO |

* 1. **Tạo bảng calendar:**

|  |
| --- |
| CREATE TABLE dbo.calendar  (  ID INT NOT NULL,  [Shift] NVARCHAR(20) NOT NULL,  [Date] Datetime NOT NULL,  FOREIGN KEY(ID) REFERENCES account(UserID)  )  GO |

* 1. **Tạo bảng Salary:**

|  |
| --- |
| CREATE TABLE dbo.Salary  (  UserID INT NOT NULL,  CheckIn DATETIME NOT NULL,  CheckOut DATETIME NOT NULL,  Checked INT NOT NULL,  Salary BIGINT NOT NULL  PRIMARY KEY (UserID,CheckIn),  FOREIGN KEY(UserID) REFERENCES account(UserID)  )  GO |

* 1. **Tạo bảng RoomCategory:**

|  |
| --- |
| CREATE TABLE dbo.RoomCategory  (  ID INT IDENTITY(1,1) PRIMARY KEY,  price FLOAT NULL,  )  GO |

* 1. **Tạo bảng RoomService:**

|  |
| --- |
| CREATE TABLE dbo.RoomService  (  id INT IDENTITY(1,1) PRIMARY KEY,  idCategory INT NOT NULL,  [name] NVARCHAR(100) NOT NULL,  [status] NVARCHAR(100) NOT NULL,  FOREIGN KEY(idCategory) REFERENCES RoomCategory(ID)  )  GO |

* 1. **Tạo bảng ServiceCategory:**

|  |
| --- |
| CREATE TABLE dbo.ServiceCategory  (  id INT IDENTITY(1,1) PRIMARY KEY,  [name] NVARCHAR(100) NOT NULL,  )  GO |

* 1. **Tạo bảng Service:**

|  |
| --- |
| CREATE TABLE dbo.[Service]  (  id INT IDENTITY(1,1) PRIMARY KEY,  [name] NVARCHAR(100) NOT NULL,  idCategory INT NOT NULL,  price FLOAT NOT NULL,  [count] INT NULL,  FOREIGN KEY(idCategory) REFERENCES ServiceCategory(id)  )  GO |

* 1. **Tạo bảng Bill:**

|  |
| --- |
| CREATE TABLE dbo.Bill  (  id INT IDENTITY(1,1) PRIMARY KEY,  DateCheckIn DATE NOT NULL,  DateCheckOut DATE NULL,  idRoom INT NOT NULL,  [status] INT NOT NULL,  discount INT NULL,  totalPrice FLOAT NULL,  FOREIGN KEY (idRoom) REFERENCES RoomService(id),  )  GO |

* 1. **Tạo bảng BillInfo:**

|  |
| --- |
| CREATE TABLE dbo.BillInfo  (  id INT IDENTITY(1,1) PRIMARY KEY,  idBill INT NOT NULL,  idService INT NOT NULL,  [count] INT NOT NULL,  FOREIGN KEY (idService) REFERENCES [Service](id),  FOREIGN KEY (idBill) REFERENCES Bill(id),  )  GO |

* 1. **Tạo bảng Customer:**

|  |
| --- |
| CREATE TABLE dbo.Customer  (  ID INT IDENTITY(1,1) PRIMARY KEY,  IDRoom INT NOT NULL,  IDBill INT NULL,  Fname NCHAR(100) NOT NULL,  Lname NCHAR(100) NOT NULL,  CMND NCHAR(50) NOT NULL,  Sex NCHAR(20) NOT NULL,  Phone NCHAR(50) NOT NULL,  DateCheckIn DATETIME NOT NULL,  DateCheckOut DATETIME,  [Status] INT,  FOREIGN KEY (IDRoom) REFERENCES RoomService(id),  FOREIGN KEY (IDBill) REFERENCES Bill(id),  )  GO |

* **Ràng buộc khóa ngoại:**

|  |
| --- |
| ALTER TABLE [dbo].[Bill] WITH CHECK ADD FOREIGN KEY([idRoom])  REFERENCES [dbo].[RoomService] ([id])  GO  ALTER TABLE [dbo].[BillInfo] WITH CHECK ADD FOREIGN KEY([idBill])  REFERENCES [dbo].[Bill] ([id])  GO  ALTER TABLE [dbo].[BillInfo] WITH CHECK ADD FOREIGN KEY([idService])  REFERENCES [dbo].[Service] ([id])  GO  ALTER TABLE [dbo].[calendar] WITH CHECK ADD FOREIGN KEY([ID])  REFERENCES [dbo].[account] ([UserID])  GO  ALTER TABLE [dbo].[Salary] WITH CHECK ADD FOREIGN KEY([UserID])  REFERENCES [dbo].[account] ([UserID])  GO  ALTER TABLE [dbo].[Customer] WITH CHECK ADD FOREIGN KEY([IDBill])  REFERENCES [dbo].[Bill] ([id])  GO  ALTER TABLE [dbo].[Customer] WITH CHECK ADD FOREIGN KEY([IDRoom])  REFERENCES [dbo].[RoomService] ([id])  GO  ALTER TABLE [dbo].[RoomService] WITH CHECK ADD FOREIGN KEY([idCategory])  REFERENCES [dbo].[RoomCategory] ([ID])  GO  ALTER TABLE [dbo].[Service] WITH CHECK ADD FOREIGN KEY([idCategory])  REFERENCES [dbo].[ServiceCategory] ([id])  GO |

1. **Tạo các câu lệnh thêm, sửa, xóa trên Stored Procedure** 
   1. **Thêm, sửa, xóa Customer :**

* **Thêm khi Check in :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_InsertCustomerCheckIn]  @idroom int,@idbill int , @fname nvarchar(100),@lname nvarchar(100),@cmnd nvarchar(50),@sex nvarchar(20),@phone nvarchar(20) ,@address text,@picture image, @datecheckin datetime ,@status int  AS  BEGIN  INSERT dbo.Customer  (  IDRoom,  IDBill,  Fname,  Lname,  CMND,  Sex,  Phone,  Address,  Picture,  DateCheckIn,  Status  )  VALUES (  @idroom,  @idbill,  @fname,  @lname,  @cmnd,  @sex,  @phone,  @address,  @picture,  @datecheckin,  @status  )  END |

* **Thêm khi Check Out :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_InsertCustomerCheckOut]  @idroom int ,@date datetime , @status int  AS  BEGIN  update dbo.Customer  set  DateCheckOut = @date,  Status = @status  WHERE IDRoom = @idroom  END |

* **Sửa Customer :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_UpdateCustomer]  @id int,@fname nvarchar(100),@lname nvarchar(100),@cmnd nvarchar(50),@sex nvarchar(20),@phone nvarchar(20) ,@address text,@picture image  AS  BEGIN  UPDATE dbo.Customer  SET  Fname = @fname,  Lname = @lname,  CMND = @cmnd,  Sex = @sex,  Phone = @phone,  Address = @address,  Picture = @picture  WHERE ID = @id  END |

* **Lấy danh sách thông tin Customer :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_GetAllCustomer]  AS  BEGIN  SELECT RoomService.name as Room,Fname as [First Name],Lname as [ Last Name ],CMND,Sex,Phone,Address,DateCheckIn,DateCheckOut  FROM Customer inner join RoomService on Customer.IDRoom = RoomService.id  end |

* **Lấy thông tin Customer theo tên phòng :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_GetCustomer]  @idroom int  AS  BEGIN  SELECT Fname as [First Name] ,Lname as [Last Name],DateCheckIn as [Check In] ,DateCheckOut as [Check Out]  FROM Customer  WHERE IDRoom = @idroom  END |

* **Xóa Customer :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_DeleteCustommer]  @id int  AS  BEGIN  DELETE dbo.Customer  WHERE ID = @id  END |

* 1. **Thêm, xóa, cập nhật tình trạng Room:**
* **Thêm Room:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_InsertRoom]  @name NVARCHAR(20) ,@idCategory int ,@status NVARCHAR(20)  AS  BEGIN  INSERT dbo.RoomService  ( name,  idCategory,  status  )  VALUES (  @name ,  @idCategory,  @status  )  END |

* **Xóa Room :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_DeleteRoom]  @name NVARCHAR(20)  AS  BEGIN  DELETE dbo.RoomService  WHERE RoomService.name = @name  END |

* **Cập nhật tình trạng Room :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_UpdateRoom]  @id int ,@status NVARCHAR(20)  AS  BEGIN  UPDATE dbo.RoomService  SET  status = @status  WHERE  id=@id  END |

* **Xuất thông tin chi tiết về Room:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_GetRoomDetail]  @id int  AS  BEGIN  SELECT RoomService.status  FROM dbo.RoomService  WHERE RoomService.id =@id  END |

* **Xuất các Room:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_GetRoomList]  AS SELECT \* FROM dbo.RoomService |

* 1. **Thêm, cập nhật, xem thông tin Service:**
* **Thêm Service:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_InsertService]  @name NVARCHAR(100) ,@idcategory int,@price float,@amount int  AS  BEGIN  INSERT dbo.Service  (  name,  idCategory,  price,  count    )  VALUES (  @name ,  @idcategory,  @price,  @amount  )  END |

* **Cập nhật Service :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_UpdateService] 1  @id int ,@name nvarchar(100),@idcategory int, @price float  AS  BEGIN  UPDATE dbo.Service  SET  name = @name,  idCategory = @idcategory,  price=@price  WHERE  id=@id  END |

* **Xem thông tin về Service :**

|  |
| --- |
| CREATE PROC [dbo].[USP\_GetServiceDetail]  @id int  AS  BEGIN  SELECT Service.id ,Service.name,ServiceCategory.name as category ,Service.price ,Service.count  FROM Service inner join ServiceCategory on Service.idCategory = ServiceCategory.id  WHERE Service.id = @id  END |

* **Xem thông tin về Service theo Category :**

|  |
| --- |
| CREATE proc [dbo].[USP\_GetArchiveByIDCategory] 1  @idcategory int  AS  BEGIN  SELECT Service.name as [Name Service],ServiceCategory.name as [Category],price as Price ,count as Count  FROM Service inner join ServiceCategory on Service.idCategory = ServiceCategory.id  WHERE ServiceCategory.id = @idcategory  END |

* 1. **Thêm, cập nhật, xem thông tin Bill:**
* **Thêm Bill:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_InsertBill]  @idRoom INT  AS  BEGIN  INSERT dbo.Bill  ( DateCheckIn ,  DateCheckOut ,  idRoom ,  status,  discount  )  VALUES ( GETDATE() , -- DateCheckIn - date  NULL , -- DateCheckOut - date  @idRoom , -- idTable - int  0, -- status - int  0 )  END  CREATE PROC [dbo].[USP\_InsertBillInfo]  @idBill INT, @idService INT, @count INT  AS  BEGIN  DECLARE @isExitsBillInfo INT  DECLARE @ServiceCount INT = 1    SELECT @isExitsBillInfo = id, @ServiceCount = b.count  FROM dbo.BillInfo AS b  WHERE idBill = @idBill AND idService = @idService  IF (@isExitsBillInfo > 0)  BEGIN  DECLARE @newCount INT = @ServiceCount + @count  IF (@newCount > 0)  UPDATE dbo.BillInfo SET count = @ServiceCount + @count WHERE idService = @idService  ELSE  DELETE dbo.BillInfo WHERE idBill = @idBill AND idService = @idService  END  ELSE  BEGIN  INSERT dbo.BillInfo  ( idBill, idService, count )  VALUES ( @idBill, -- idBill - int  @idService, -- idService - int  @count -- count - int  )  END  END |

* **Cập nhật Bill:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_UpdateBillCheckOut]  @idbill int ,@date datetime , @status int ,@totalprice float  AS  BEGIN  update dbo.Bill  set  DateCheckOut = @date,  Status = @status,  totalPrice=@totalprice  WHERE id = @idbill  END |

* 1. **Thao tác để thống kê :**
* **Thống kê tất cả khoảng thu vào của Hotel theo khoảng thời gian :**

|  |
| --- |
| CREATE proc [dbo].[USP\_Static]  @date1 date ,@date2 date  AS  BEGIN  SELECT CAST(DateCheckOut AS DATE) as Date , Sum(totalPrice) as Price  FROM Bill  WHERE status = 1 and DateCheckOut >= @date1 and DateCheckOut <= @date2  GROUP BY CAST(DateCheckOut AS DATE)  end |

* **Thống kê tất cả khoảng thu vào của Hotel :**

|  |
| --- |
| CREATE proc [dbo].[USP\_StaticBill]  AS  BEGIN  SELECT CAST(DateCheckOut AS DATE) as Date , Sum(totalPrice) as Price  FROM Bill  WHERE status =1  GROUP BY CAST(DateCheckOut AS DATE)  END |

* **Thống kê tất cả khoảng chi tiêu của Hotel theo khoảng thời gian:**

|  |
| --- |
| CREATE proc [dbo].[USP\_StaticSalaryByDay]  @date1 date ,@date2 date  AS  BEGIN  SELECT CAST(CheckOut AS DATE) as Date ,Sum(Salary) as Cost  FROM salary  WHERE Checked=2 and CheckOut >= @date1 and CheckOut <= @date2  GROUP BY CAST(CheckOut AS DATE)  END |

* **Thống kê tất cả khoảng chi tiêu của Hotel :**

|  |
| --- |
| CREATE proc [dbo].[USP\_StaticSalary]  AS  BEGIN  SELECT CAST(CheckOut AS DATE) as Date ,Sum(Salary) as Cost  FROM salary  WHERE Checked=2  GROUP BY CAST(CheckOut AS DATE)  END |

1. **Trigger:**
   1. **Trigger update số lượng trong kho khi khách sử dụng dịch vụ**

|  |
| --- |
| CREATE or ALTER TRIGGER trigger\_update\_kho  ON BillInfo  AFTER insert  as  BEGIN  declare @id int = (SELECT idService FROM inserted)  declare @soluong int =(SELECT [count] FROM inserted)  update [Service]  set [Service].[count] = [Service].[count] - @soluong  where [Service].[id] = @id  END |

* 1. **Trigger update trạng thái phòng khi khách checkin**

|  |
| --- |
| CREATE or ALTER TRIGGER trigger\_update\_room\_status  ON Customer  AFTER insert  as  BEGIN  declare @id int = (SELECT IDRoom FROM inserted)  update RoomService  set [status]= N'Đầy'  where id = @id  END |

* 1. **Trigger tính lương khi nhân viên checkout**

|  |
| --- |
| CREATE or ALTER TRIGGER trigger\_update\_salary  ON Salary  AFTER update  as  BEGIN  declare @id int =(SELECT UserID FROM inserted)  UPDATE salary  SET Checked=2, Salary =40000\*(ABS(CAST(DATENAME(HOUR,CheckOut) AS int)-CAST(DATENAME(HOUR,CheckIn) AS int)) -ABS(ABS(CAST(DATENAME(HOUR,CheckOut) AS int)-CAST(DATENAME(HOUR,CheckIn) AS float))-8))  WHERE Checked=1 and UserID =@id  END |

1. **Transaction :**
   1. **Transaction thực hiện đồng thời thanh toán và trả phòng khi khách checkout:**

|  |
| --- |
| CREATE PROC [dbo].[USP\_TransactionCheckOut]  @idroom int ,@date datetime , @status int  AS  BEGIN  BEGIN TRY  BEGIN TRANSACTION  update dbo.Customer  set  DateCheckOut = @date,  Status = @status  WHERE IDRoom = @idroom  update dbo.RoomService  set [status]= N'Trống'  WHERE id =@idroom  COMMIT TRANSACTION  END TRY  BEGIN CATCH  ROLLBACK TRANSACTION  END CATCH  END |

1. **Function :**
   1. **Function lọc ra những phòng có Bill chưa được thanh toán:**

|  |
| --- |
| Create function [dbo].[f\_GetUncheckBillIDByRoomID](@id int)  returns table  as  return (select \* from dbo.Bill  where dbo.Bill.idRoom=@id AND status = 0) |

* 1. **Function để tìm kiếm Customer theo thông tin First name, Last name, Address, Phone, CMND :**

|  |
| --- |
| CREATE function [dbo].[f\_SearchCustomer](@key text)  returns table  as  return (SELECT RoomService.name as Room,Fname as [First Name],Lname as [ Last Name ],CMND,Sex,Phone,Address,DateCheckIn,DateCheckOut  FROM Customer inner join RoomService on Customer.IDRoom = RoomService.id  WHERE CONCAT(Fname, Lname, Address, Phone, CMND) LIKE @key); |

* 1. **Function để hiển thị các Service của từng Room theo idRoom :**

|  |
| --- |
| CREATE FUNCTION [dbo].[f\_GetListMenuByRoom](@id int)  RETURNS TABLE  AS  RETURN (SELECT dbo.Service.name, dbo.BillInfo.count, dbo.Service.price, dbo.Service.price \* dbo.BillInfo.count AS TotalPrice  FROM dbo.Bill INNER JOIN  dbo.BillInfo ON dbo.Bill.id = dbo.BillInfo.idBill INNER JOIN  dbo.Service ON dbo.BillInfo.idService = dbo.Service.id  WHERE(dbo.Bill.status = 0 AND dbo.Bill.idRoom = @id)) |

* 1. **Function để tìm kiếm thông tin về User :**

|  |
| --- |
| CREATE FUNCTION [dbo].[f\_SearchAccount](@key TEXT)  RETURNS TABLE  AS  RETURN (SELECT \*  FROM dbo.Account  WHERE CONCAT(Username,Fname,Lname,Address,Phone,Type) LIKE @key) |

* 1. **Function để tìm kiếm về thông tin lịch trực của toàn bộ nhân viên :**

|  |
| --- |
| CREATE FUNCTION [dbo].[f\_FindCalendar](@id int)  RETURNS TABLE  AS  RETURN(SELECT ID,Fname AS [First Name],Lname AS [Last Name],Type AS [Position],Shift,CONVERT(varchar,Date,101) AS [Date],Phone  FROM calendar INNER JOIN Account ON calendar.ID=Account.UserID  WHERE CONVERT(varchar,Date,101) >=CONVERT(varchar,GETDATE(),101) AND ID=@id) |

* 1. **Function để tìm kiếm thông tin về lịch trực theo từng nhân viên :**

|  |
| --- |
| CREATE FUNCTION [dbo].[f\_getMyCalendar](@userID int)  RETURNS TABLE  AS  RETURN (SELECT Shift,CONVERT(varchar, Date,101) AS[Date]  FROM calendar  WHERE CONVERT(varchar,Date,101) >=CONVERT(varchar,GETDATE(),101) AND ID=@userID) |

1. **View :**
   1. **View để lấy dữ liệu về Service Category :**

|  |
| --- |
| CREATE VIEW [dbo].[ServiceCategory\_data]  AS  SELECT dbo.ServiceCategory.\*  FROM dbo.ServiceCategory |

* 1. **View để lấy dữ liệu về toàn bộ Service :**

|  |
| --- |
| CREATE VIEW [dbo].[Service\_data]  AS  SELECT \*  FROM dbo.Service |

* 1. **View để lấy tất cả thông tin về Account :**

|  |
| --- |
| CREATE VIEW [dbo].[Account\_data]  AS  SELECT dbo.Account.\*  FROM dbo.Account |

* 1. **View để lấy tất cả thông tin về Bill :**

|  |
| --- |
| CREATE VIEW [dbo].[Bill\_data]  AS  SELECT dbo.Bill.\*  FROM dbo.Bill |

* 1. **View để lấy thông tin chi tiết của Bill :**

|  |
| --- |
| CREATE VIEW [dbo].[BillInfo\_data]  AS  SELECT dbo.BillInfo.\*  FROM dbo.BillInfo |

* 1. **View để lấy tất cả thông tin về Salary của mỗi nhân viên :**

|  |
| --- |
| CREATE VIEW [dbo].[Salary\_data]  AS  SELECT dbo.Salary.\*  FROM dbo.Salary |

* 1. **View để lấy tất cả thông tin về Calendar của mỗi nhân viên :**

|  |
| --- |
| CREATE VIEW [dbo].[Calendar\_data]  AS  SELECT dbo.calendar.ID, dbo.Account.Fname AS [First Name], dbo.Account.Lname AS [Last Name], dbo.Account.Type AS Position, dbo.calendar.Shift, CONVERT(varchar, dbo.calendar.Date, 101) AS Date, dbo.Account.Phone  FROM dbo.calendar INNER JOIN dbo.Account  ON dbo.calendar.ID = dbo.Account.UserID |

1. **Phân quyền:**

* **Tạo các User Login :**

|  |
| --- |
| CREATE USER [Reception] FOR LOGIN [Reception] WITH DEFAULT\_SCHEMA=[dbo]  GO  CREATE USER [Manager] FOR LOGIN [Manager] WITH DEFAULT\_SCHEMA=[dbo]  GO  CREATE USER [Labor] FOR LOGIN [Labor] WITH DEFAULT\_SCHEMA=[dbo]  GO |

* **Tạo các Role :**

|  |
| --- |
| CREATE ROLE [ReceptionRole]  GO  CREATE ROLE [ManagerRole]  GO  CREATE ROLE [LaborRole]  GO |

* **Map giữa User và Role :**

|  |
| --- |
| ALTER ROLE [ReceptionRole] ADD MEMBER [Reception]  GO  ALTER ROLE [ManagerRole] ADD MEMBER [Manager]  GO  ALTER ROLE [LaborRole] ADD MEMBER [Labor]  GO |

1. **Kết nối cơ sở dữ liệu**

class DataProvider

{

private static DataProvider instance;

public static DataProvider Instance

{

get { if (instance == null) instance = new DataProvider(); return DataProvider.instance; }

private set { DataProvider.instance = value; }

}

private DataProvider() { }

string role = Global.globalUserType;

private string connectionManager = @"Data Source=DESKTOP-RSHOI2U\SQLEXPRESS;User ID=Manager;Password=123;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False";

private string connectionReception = @"Data Source=DESKTOP-RSHOI2U\SQLEXPRESS;User ID=Reception;Password=123;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False";

private string connectionLabor = @"Data Source=DESKTOP-RSHOI2U\SQLEXPRESS;User ID=Labor;Password=123;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False";

private string connectionSTR;

public DataTable ExecuteQuery(string query, object[] parameter = null)

{

DataTable data = new DataTable();

if(role == "Manager")

{

connectionSTR = connectionManager;

}

else if(role == "Reception")

{

connectionSTR = connectionReception;

}

else

{

connectionSTR = connectionLabor;

}

try

{

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

SqlDataAdapter adapter = new SqlDataAdapter(command);

adapter.Fill(data);

connection.Close();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

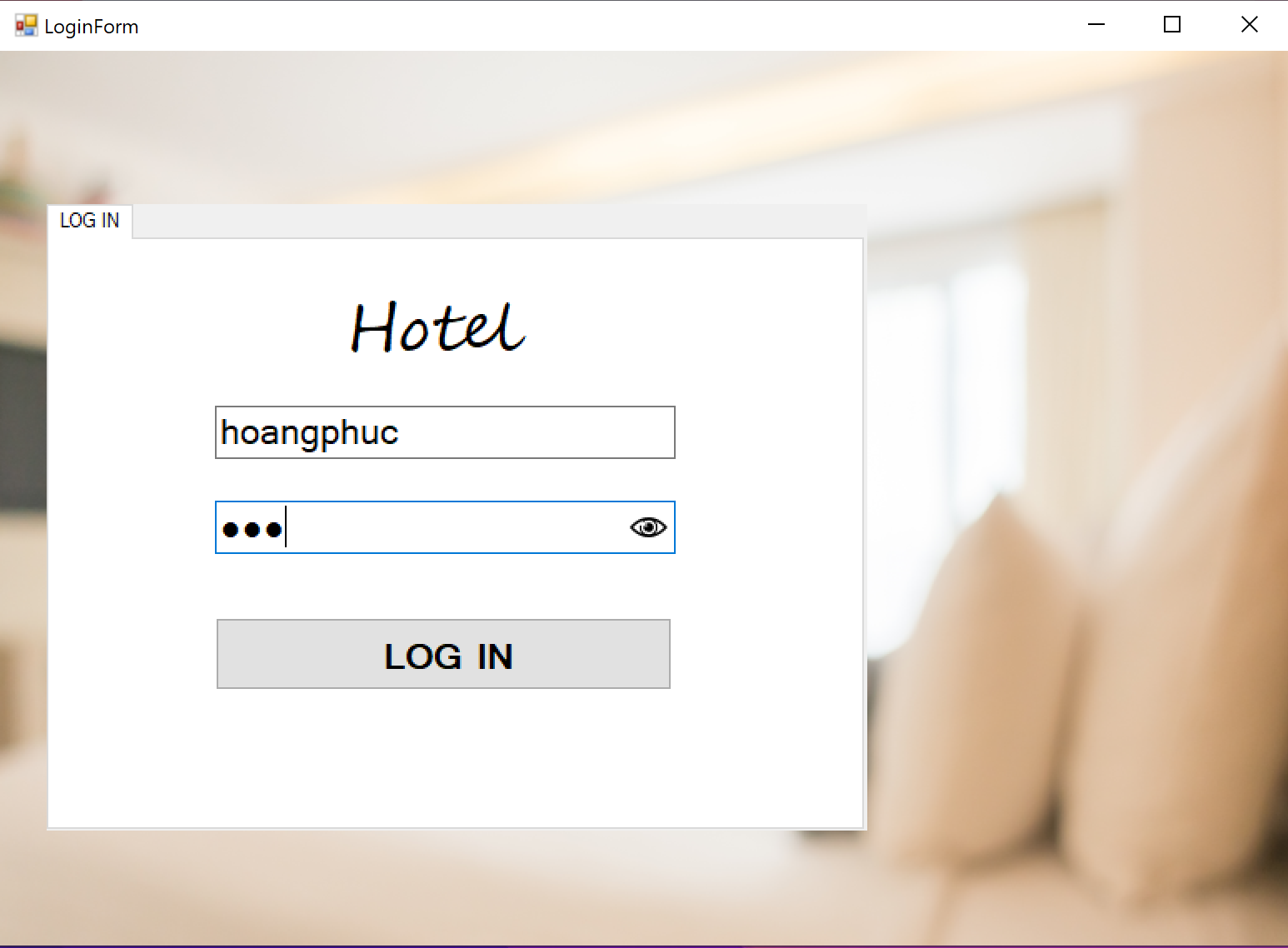
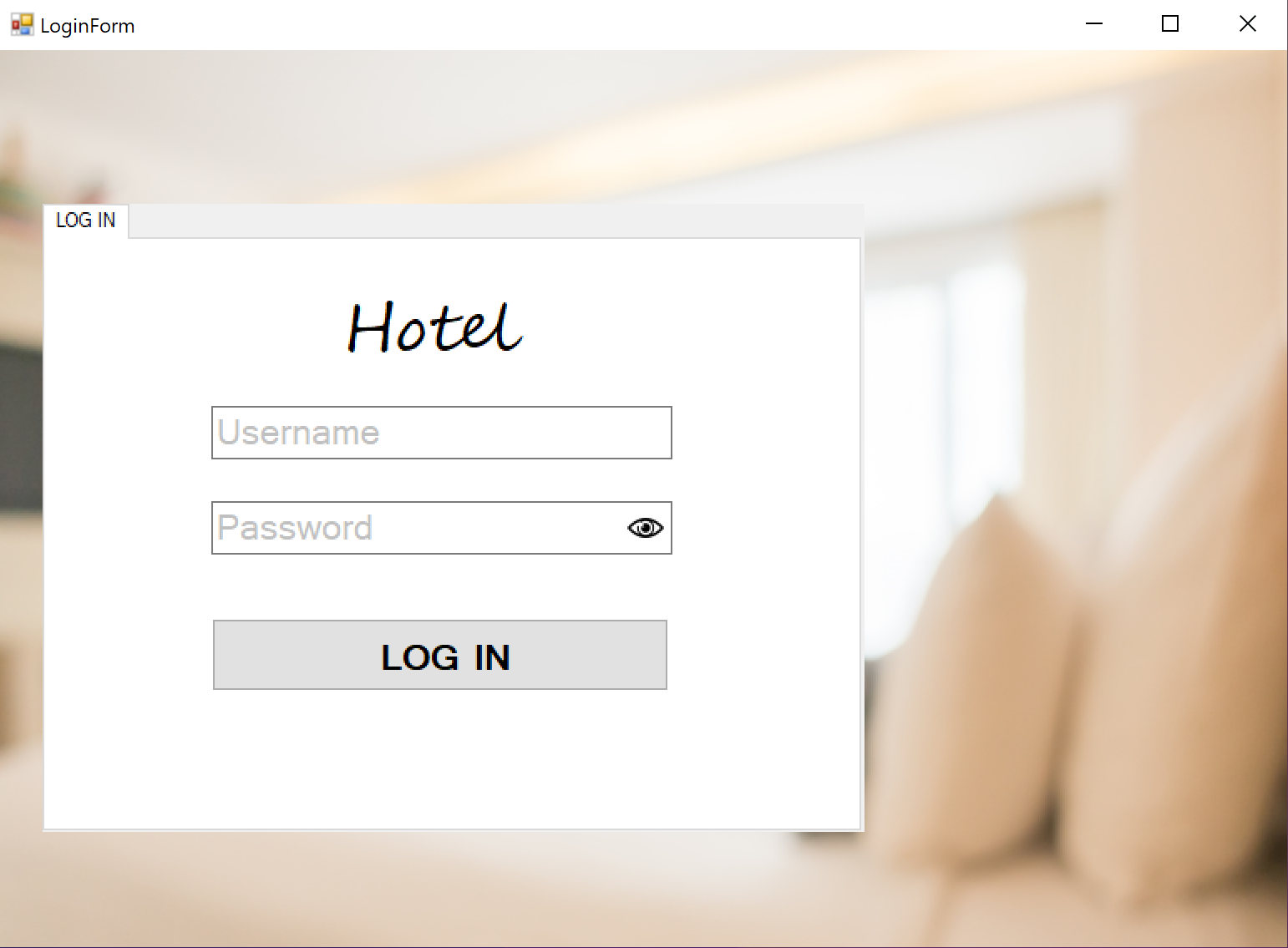
return data;

}

**CHƯƠNG 3 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM**

1. **Đăng nhập**

Người dùng nhập ip sever chứa cơ sở dữ liệu và sử dụng tài khoản của mình để đăng nhập vào hệ thống.  
Phần mền có 3 quyền như sau: Manager, Reception, Labor



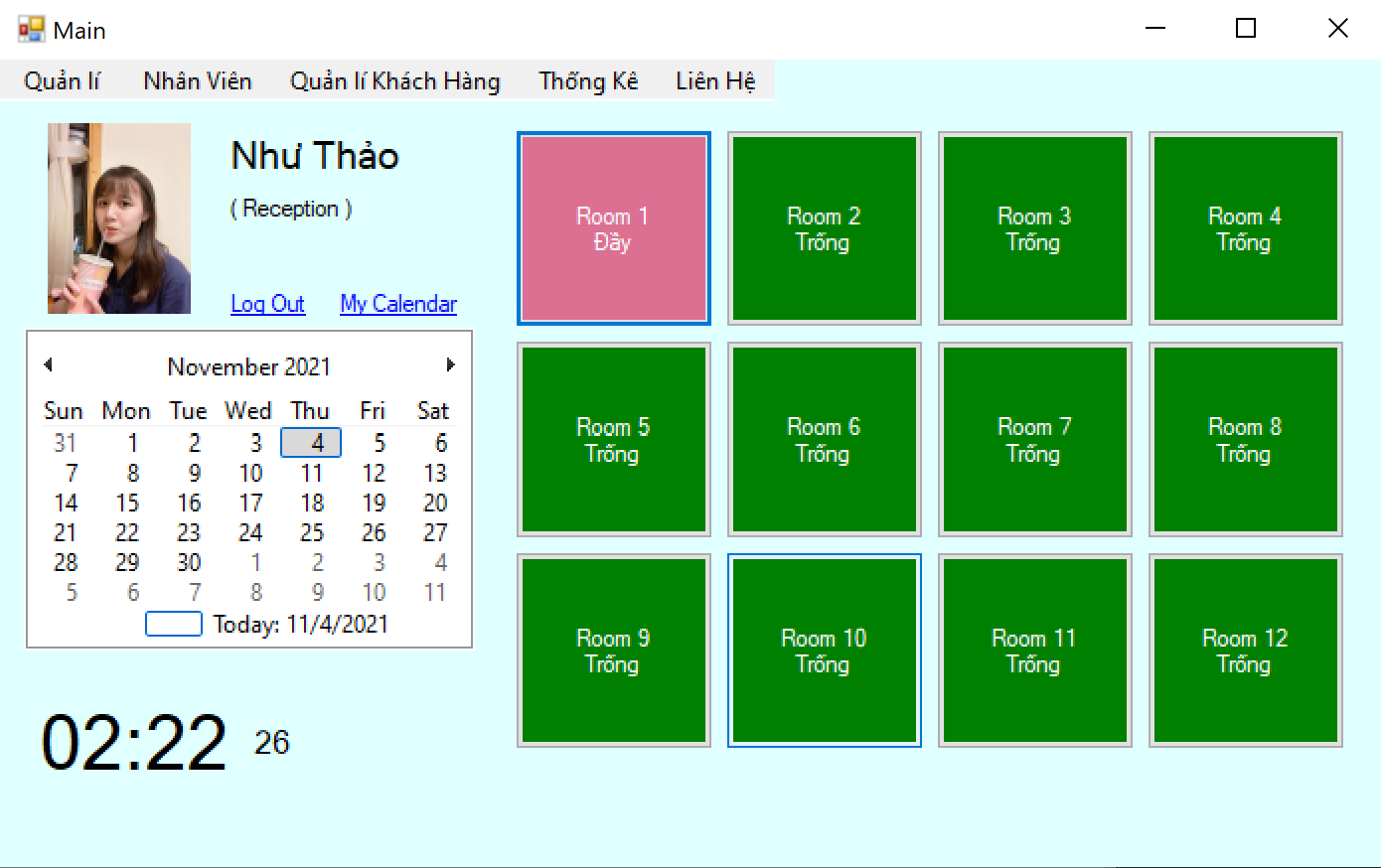
1. **Màn hình trang chủ:**

Sau khi đăng nhập thì trang chủ của mỗi quyền đăng nhập sẽ hiện lên:

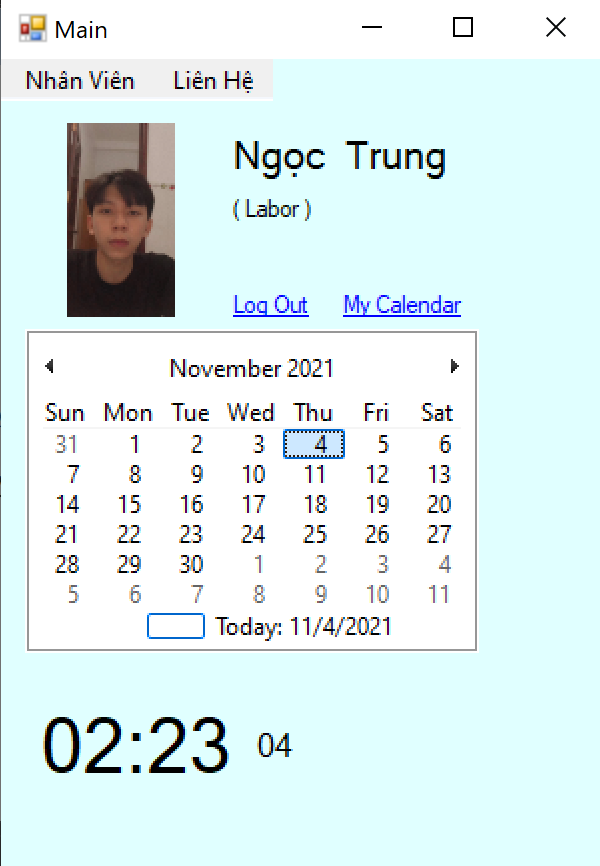
+ Manager :



+ Reception:

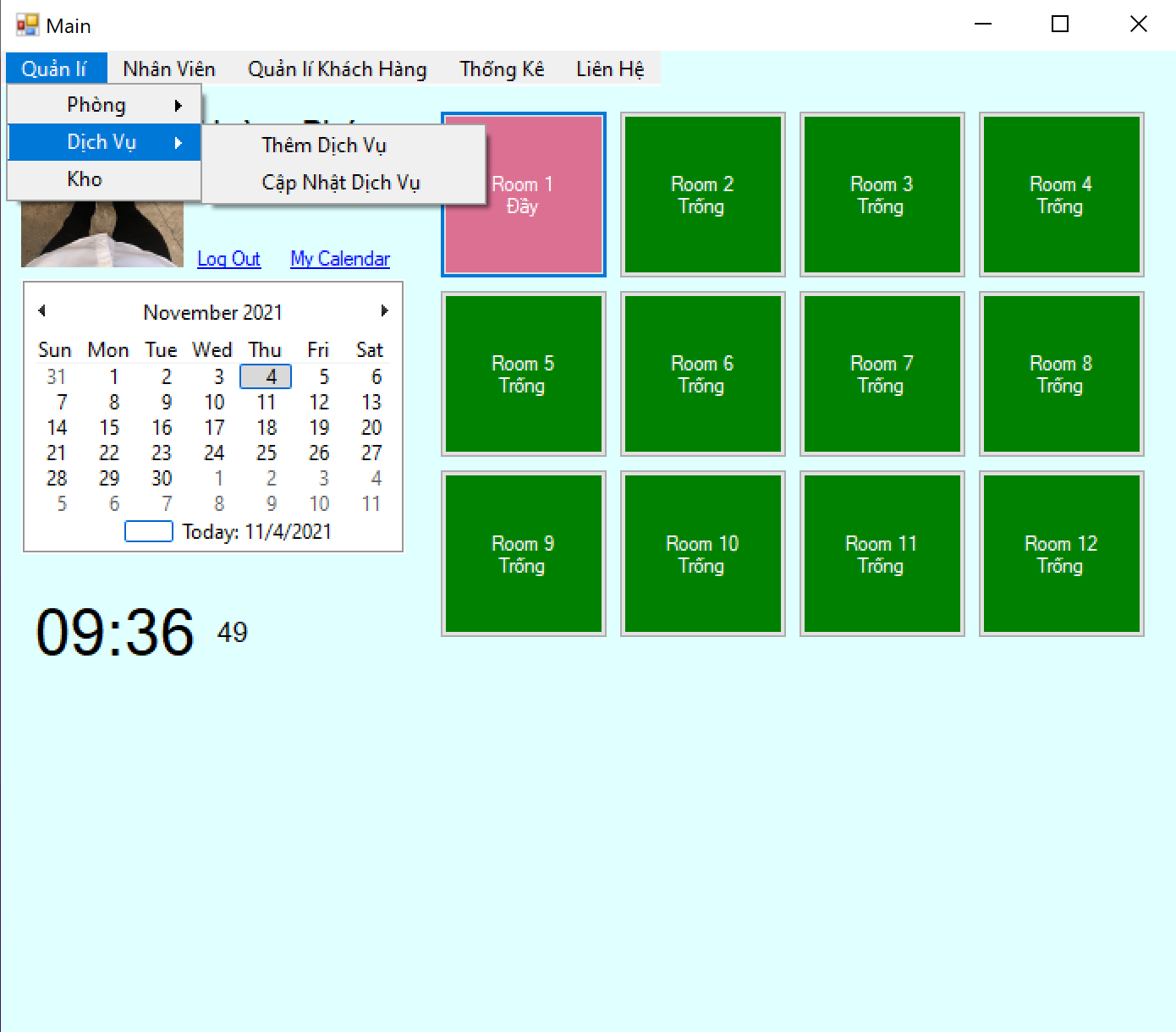
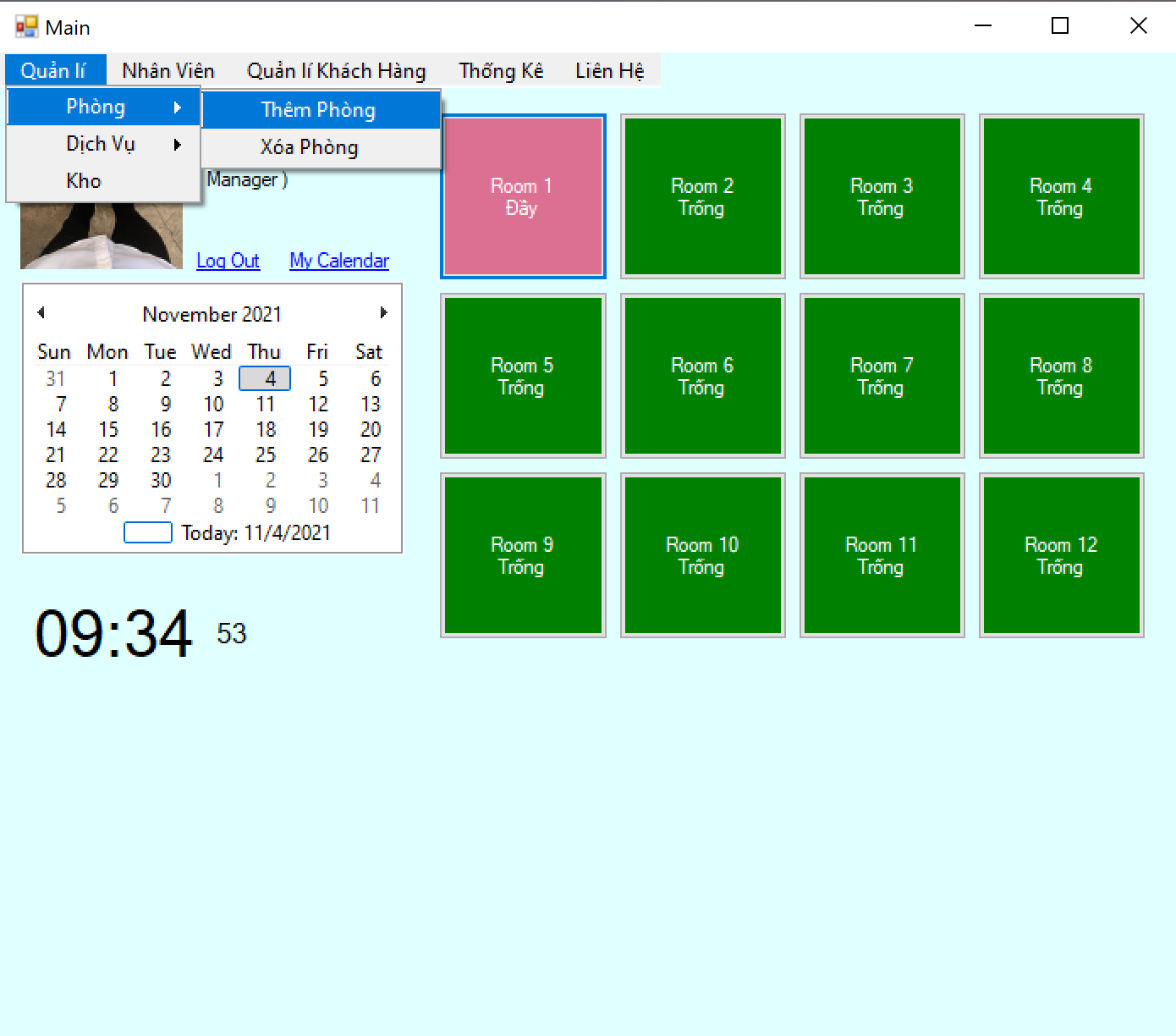


+Labor:



* Ở giao diện của Manager thì giao diện này hoàn toàn giống với giao diện của Reception. Nhưng Manager có thể thao tác được hết tất cả chức năng trong giao diện này. Do Manager được toàn quyền sử dụng cơ sở dữ liệu. Còn Reception không phải chức năng nào thì Reception cũng sử dụng được do Reception bị giới hạn một số quyền trong cơ sở dữ liệu.

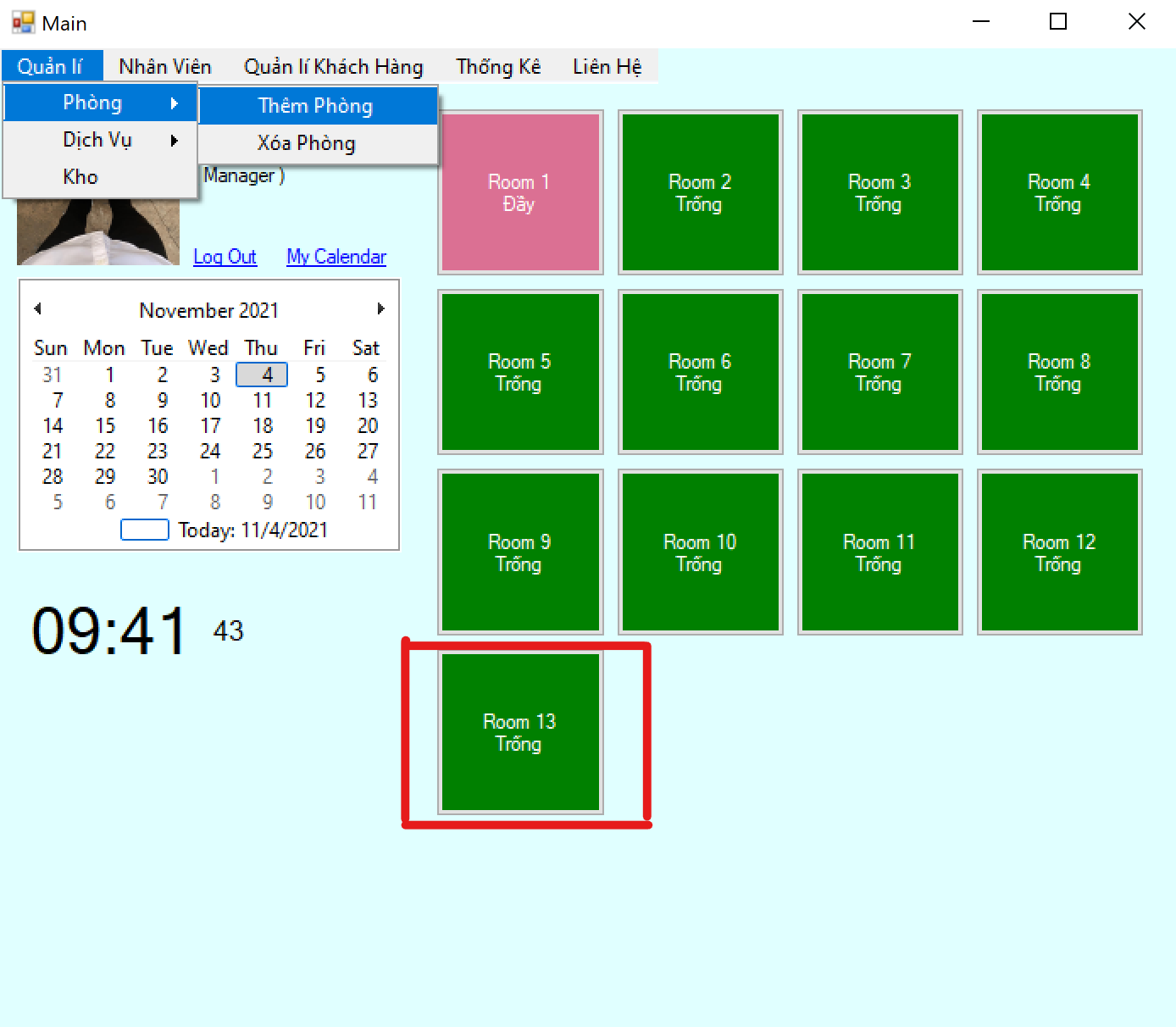
1. **Các chức năng có thể thực hiện ở quyền Manager:**
   1. **Chức năng Quản lí :**



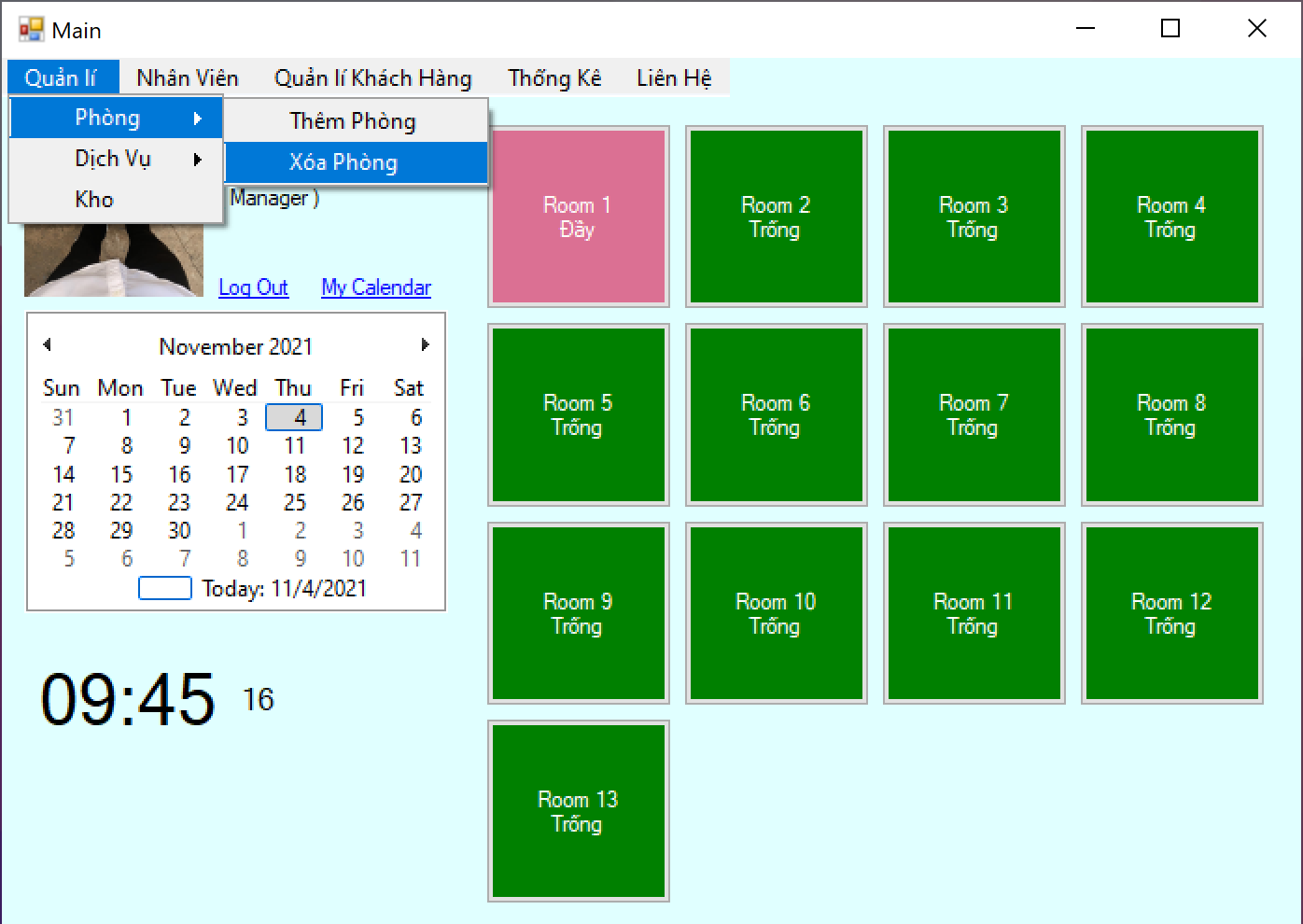
* Ở chức năng Quản lí với quyền đăng nhập của Manager thì Manager có thể :

**3.1.1 Thêm/ Xóa các phòng :**

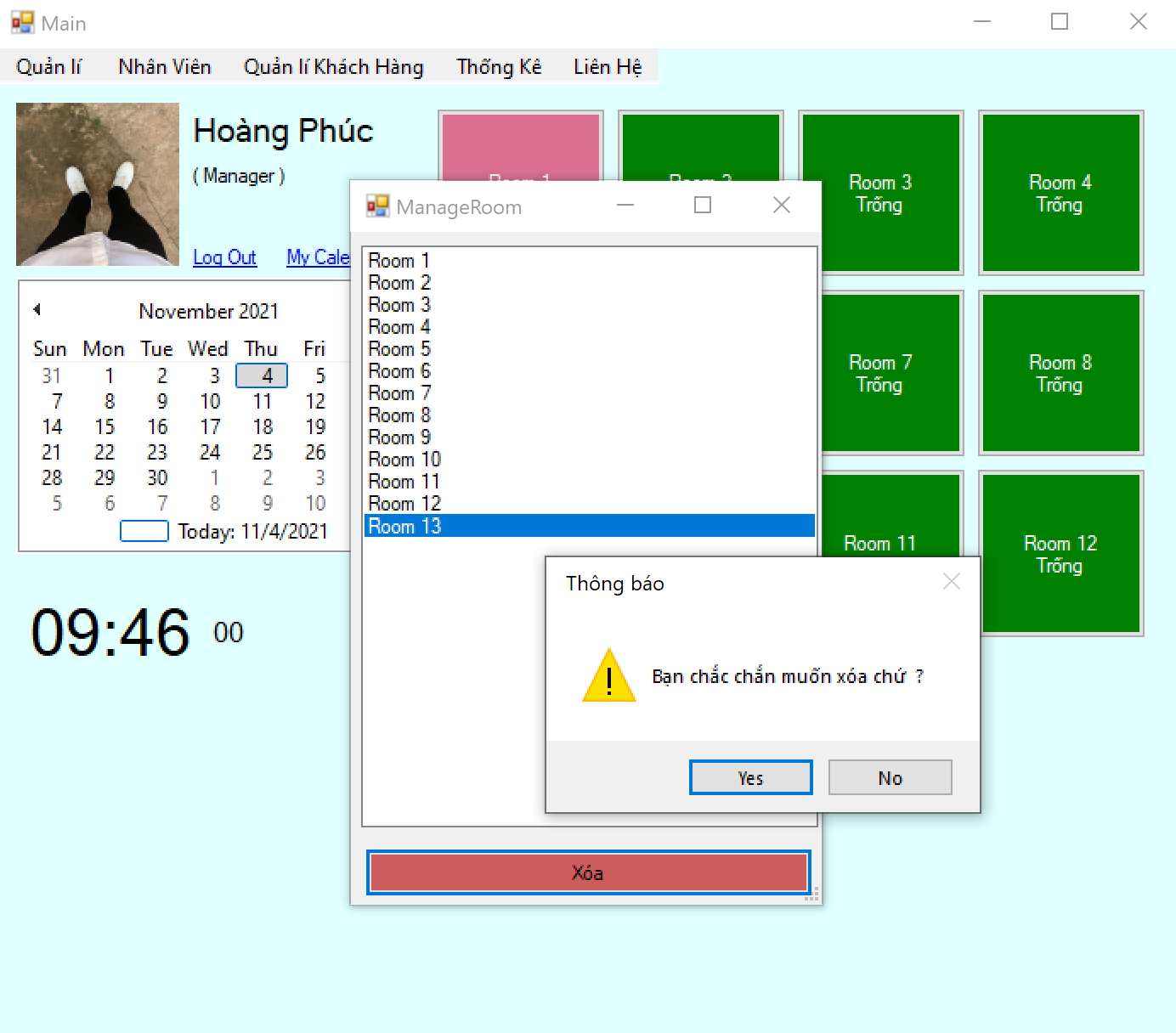
* **Quản lí -> Phòng -> Thêm phòng :**



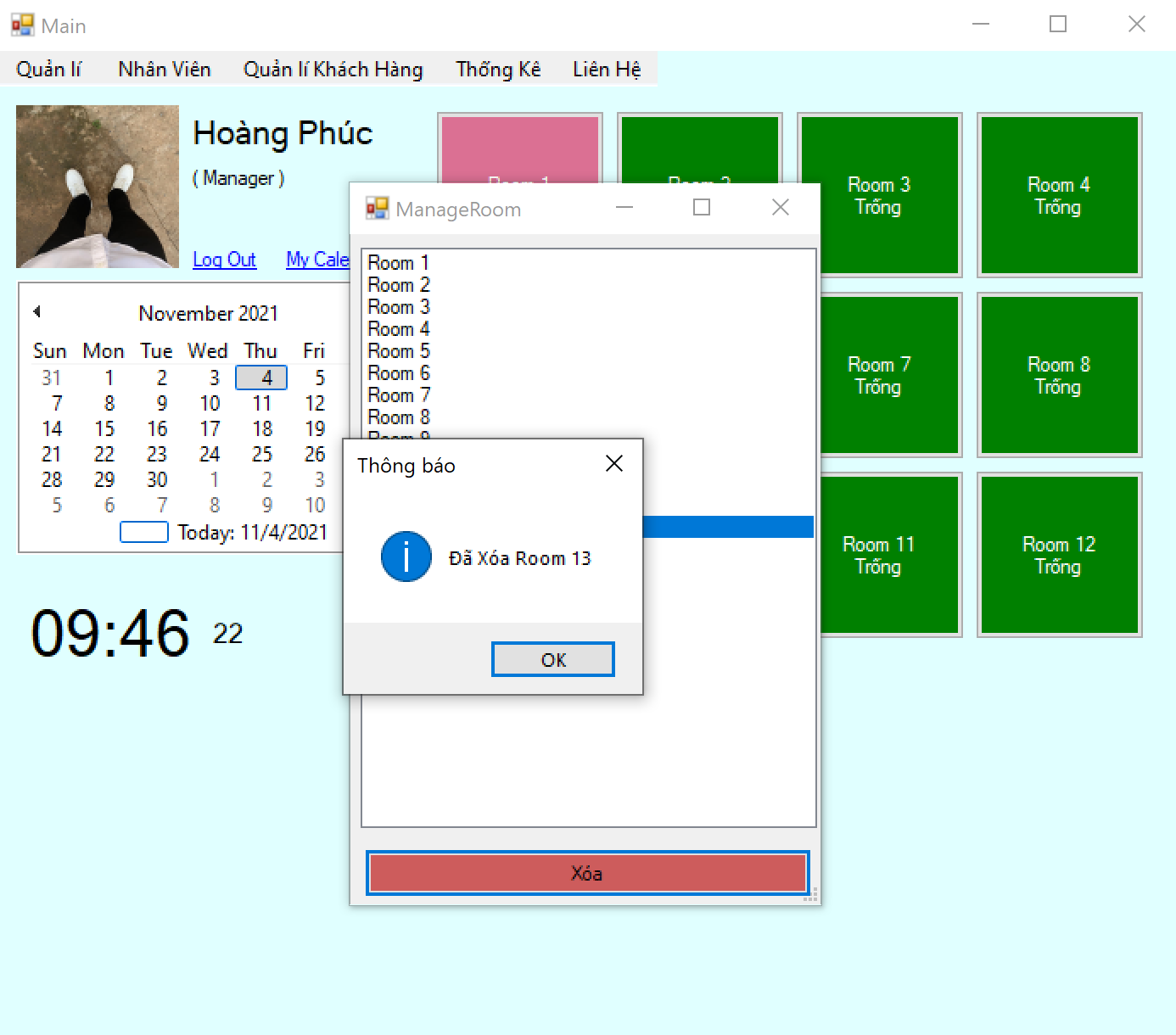
* **Quản lí -> Phòng -> Xóa phòng :**



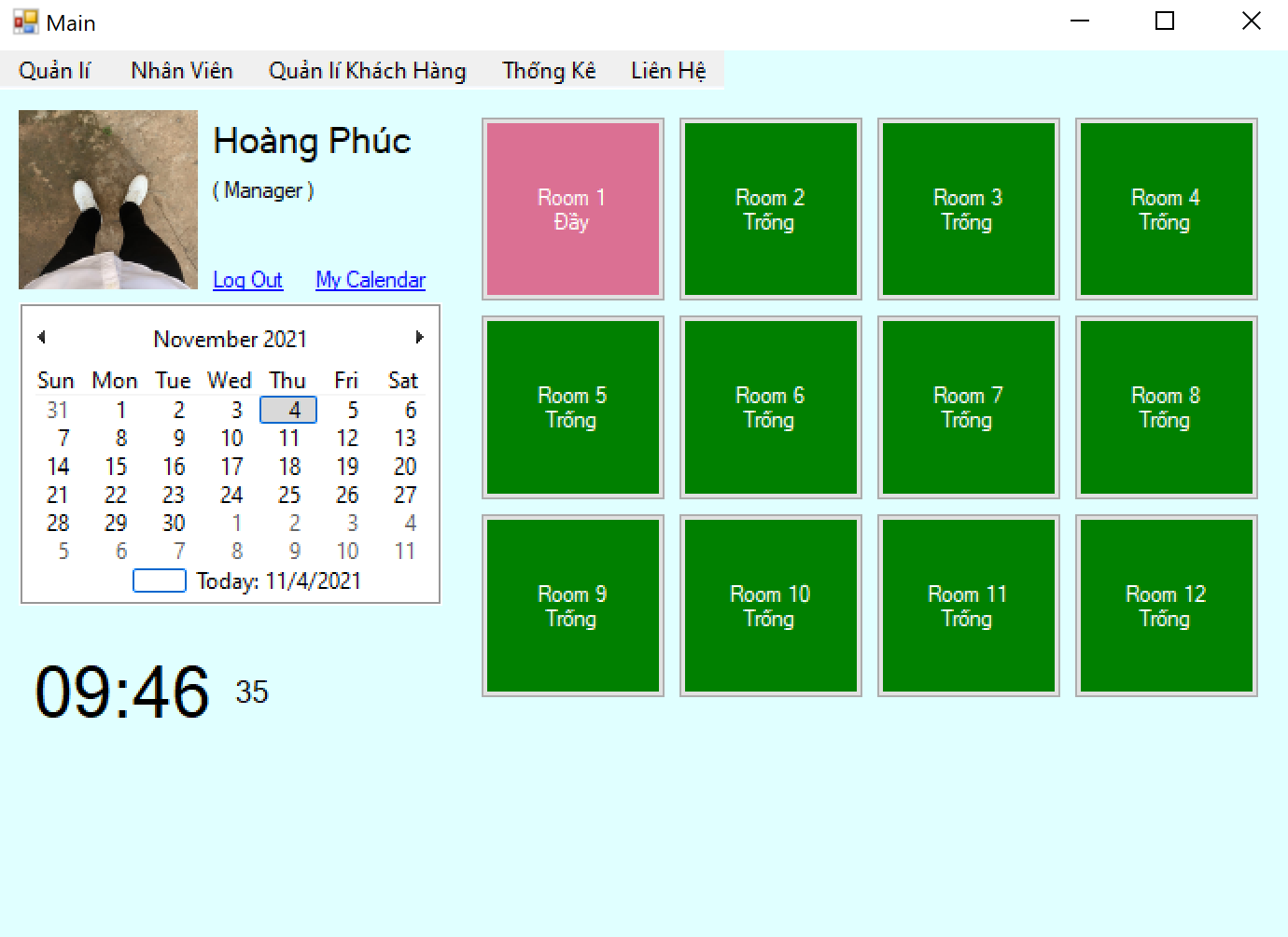
Cửa sổ ManageRoom sẽ hiện ra để Manager có thể chọn phòng cần xóa. Sau khi chọn được phòng cần xóa thì ấn nút **Xóa**. Sẽ có hộp thoại thông báo xuất hiện để xác nhận lại lần nữa trước khi xóa. Khi ta ấn **Yes** thì phần mềm sẽ tiến hành thao tác xóa.



Và đây là kết quả sau khi xóa

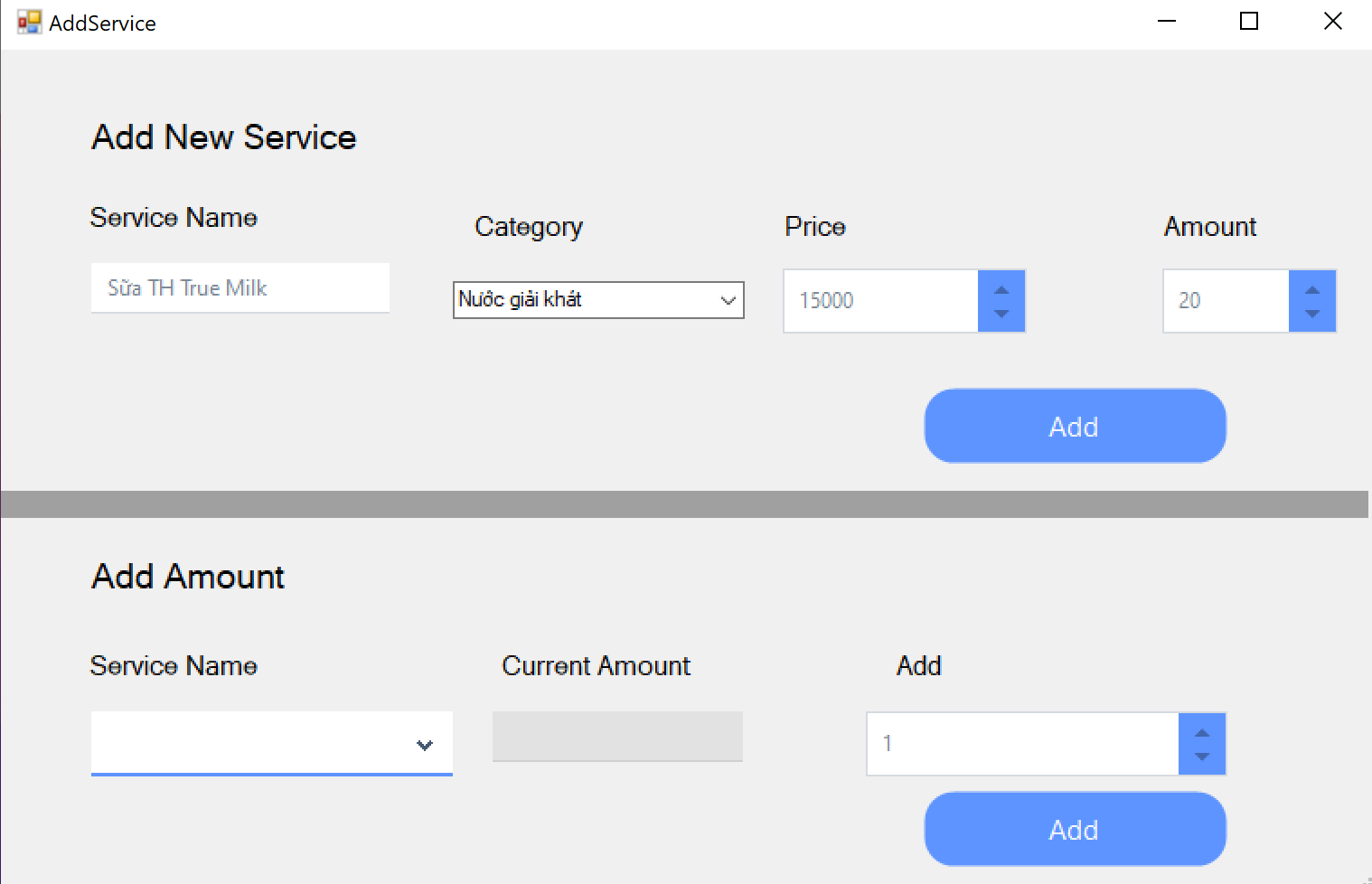


Phòng 13 đã được xóa

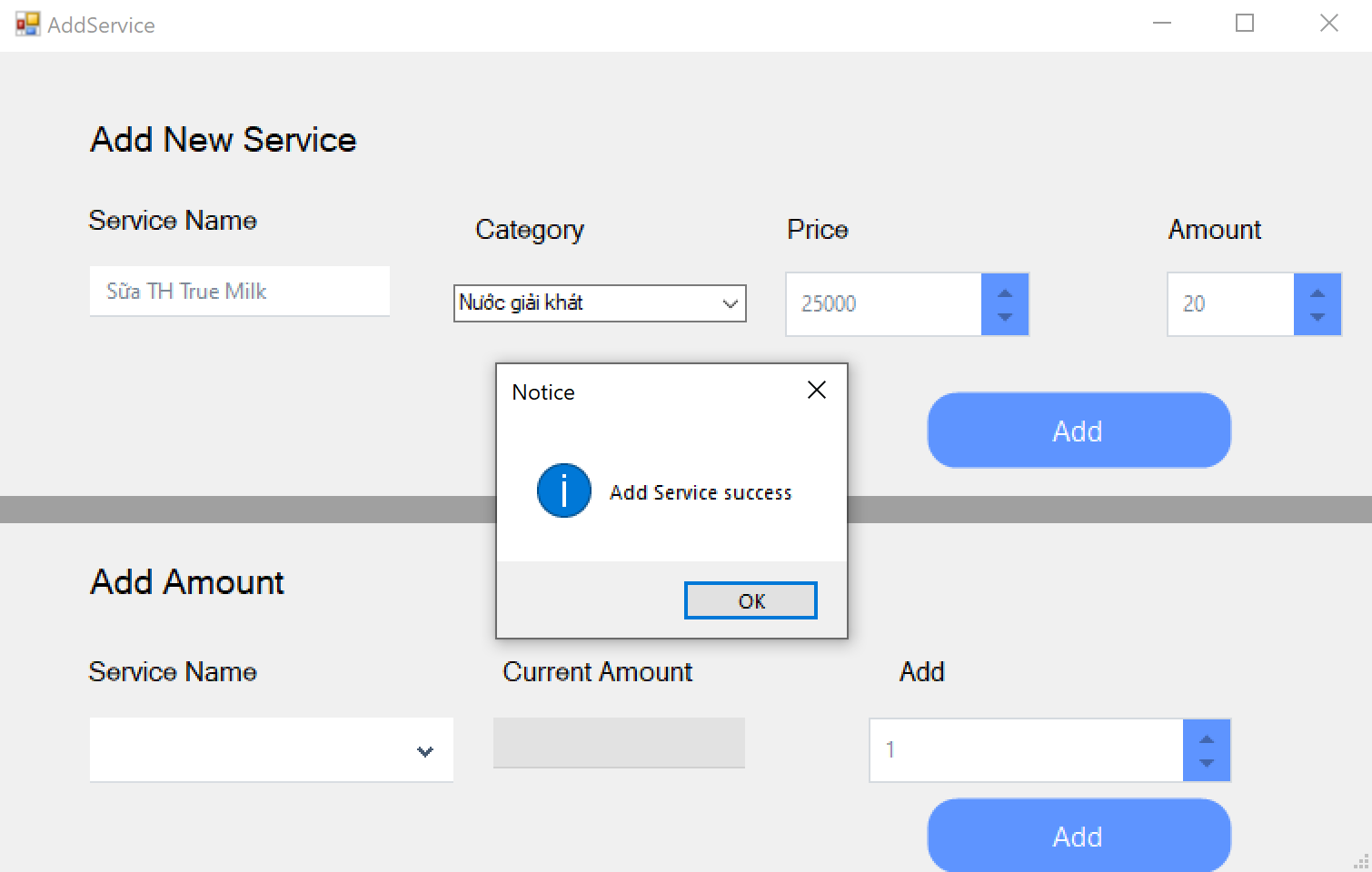


**3.1.2 Thêm/ Cập nhật các dịch vụ :**

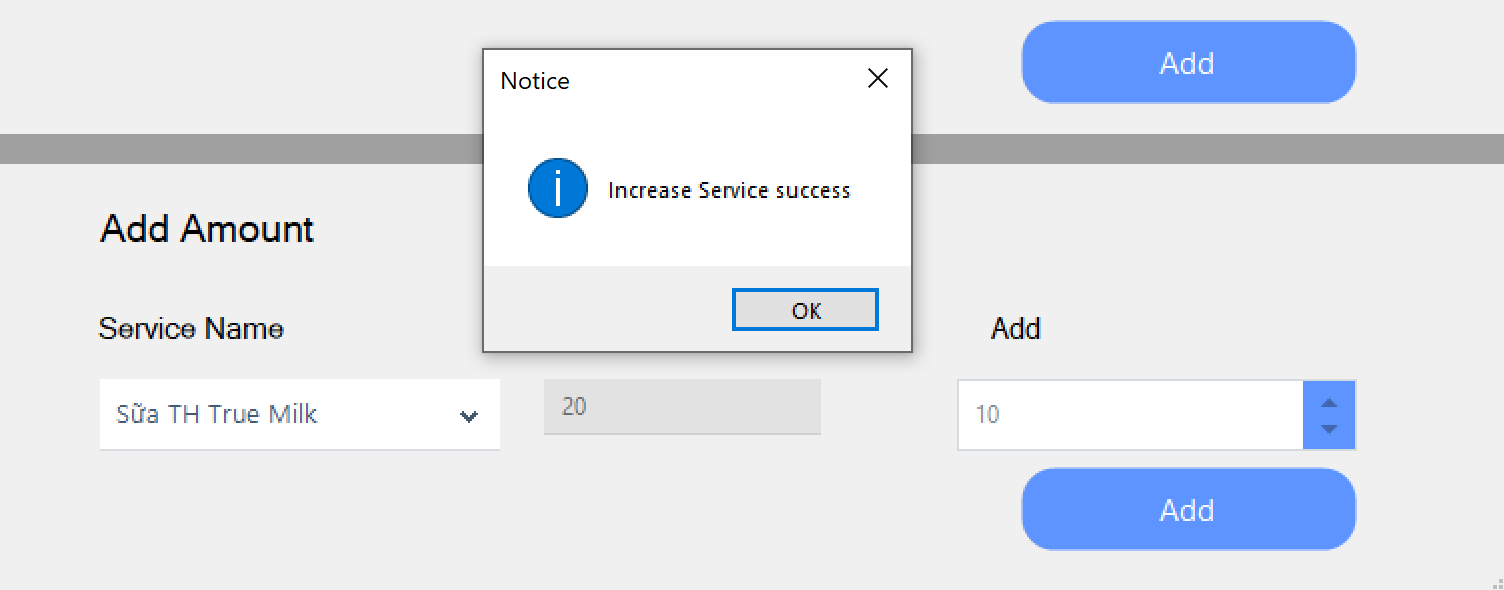
* **Quản lí -> Dịch vụ -> Thêm dịch vụ :** cửa sổ AddService sẽ hiện ra.Ta có thể thêm dịch vụ ở mục Add New Service và cập nhật thêm số lượng của các dịch vụ hiện có ở mục Add Amount



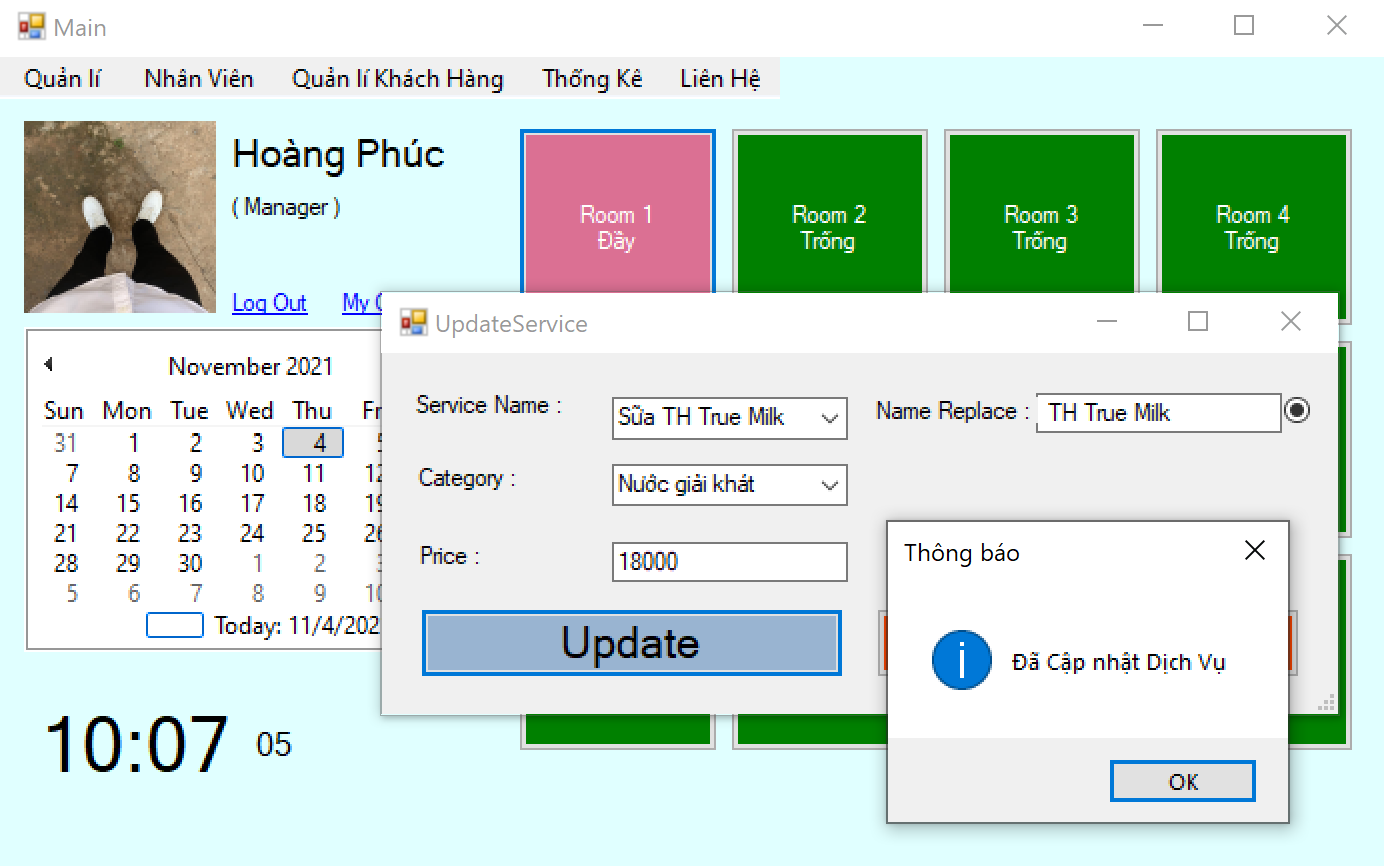
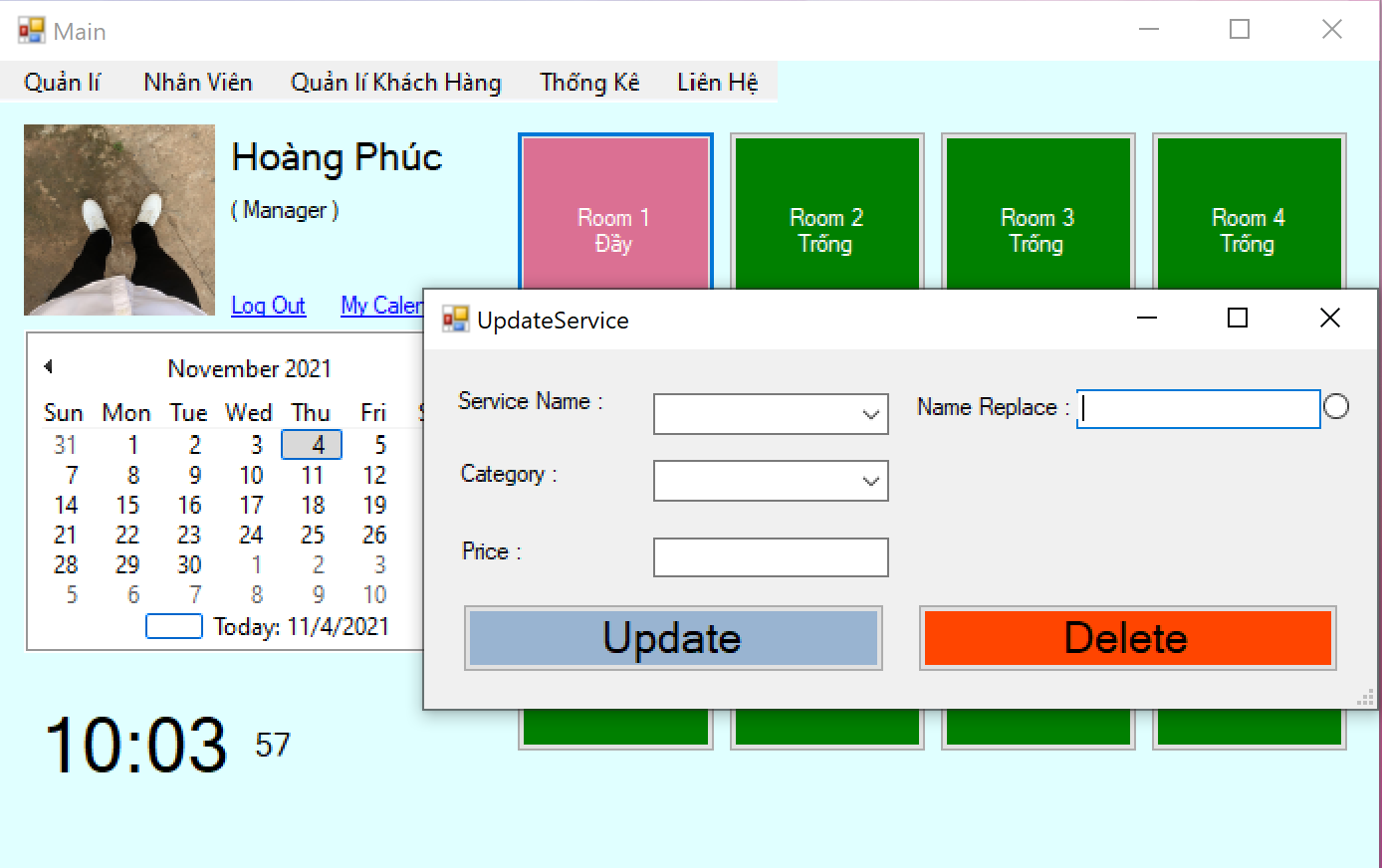
Khi Manager đã chọn xong các thiết lập ở mục Add New Service thì click nút **Add** để tiến hành thêm dịch vụ



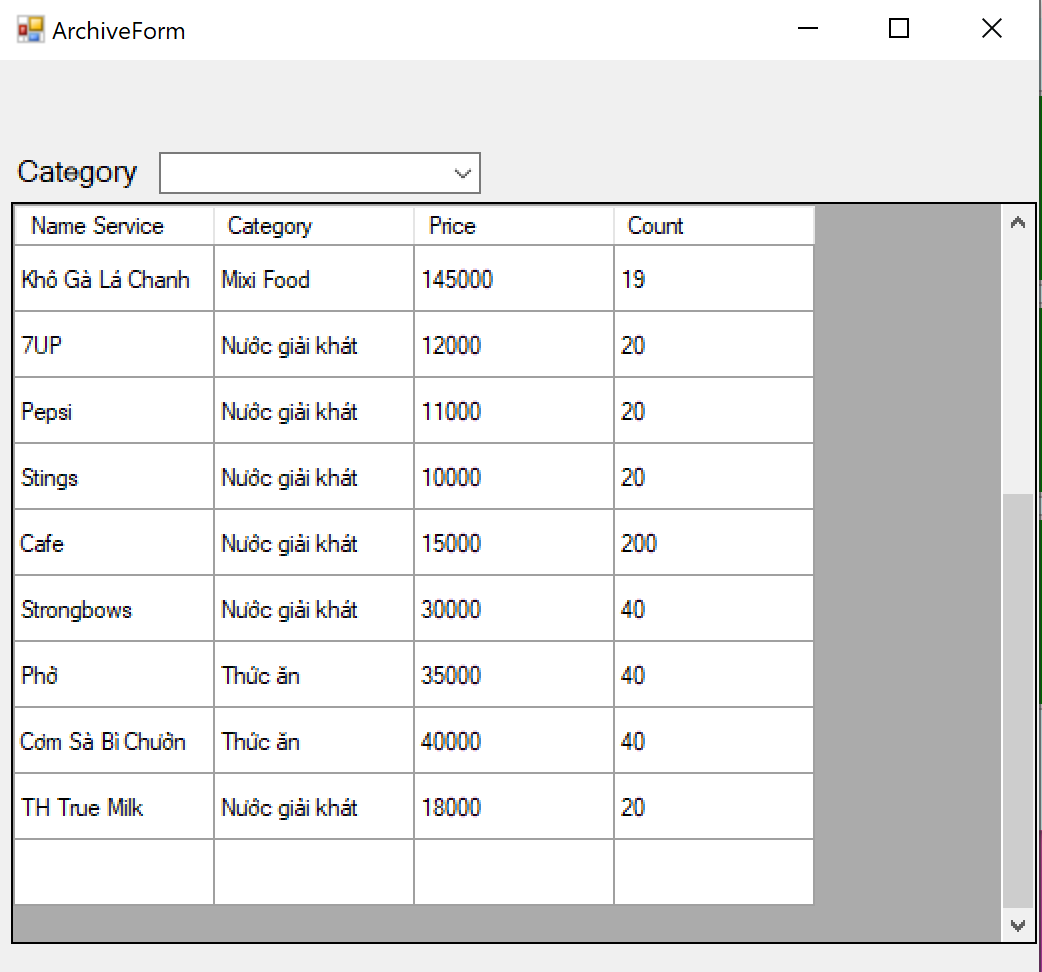
Ta có thể cập nhật số lượng của Dịch vụ ở mục Add Amount



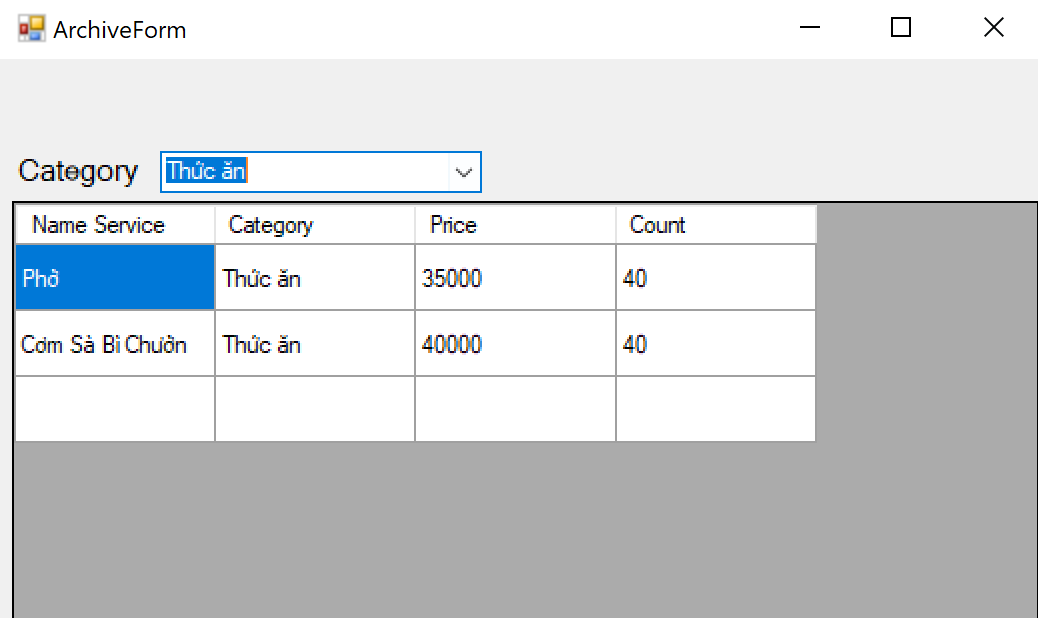
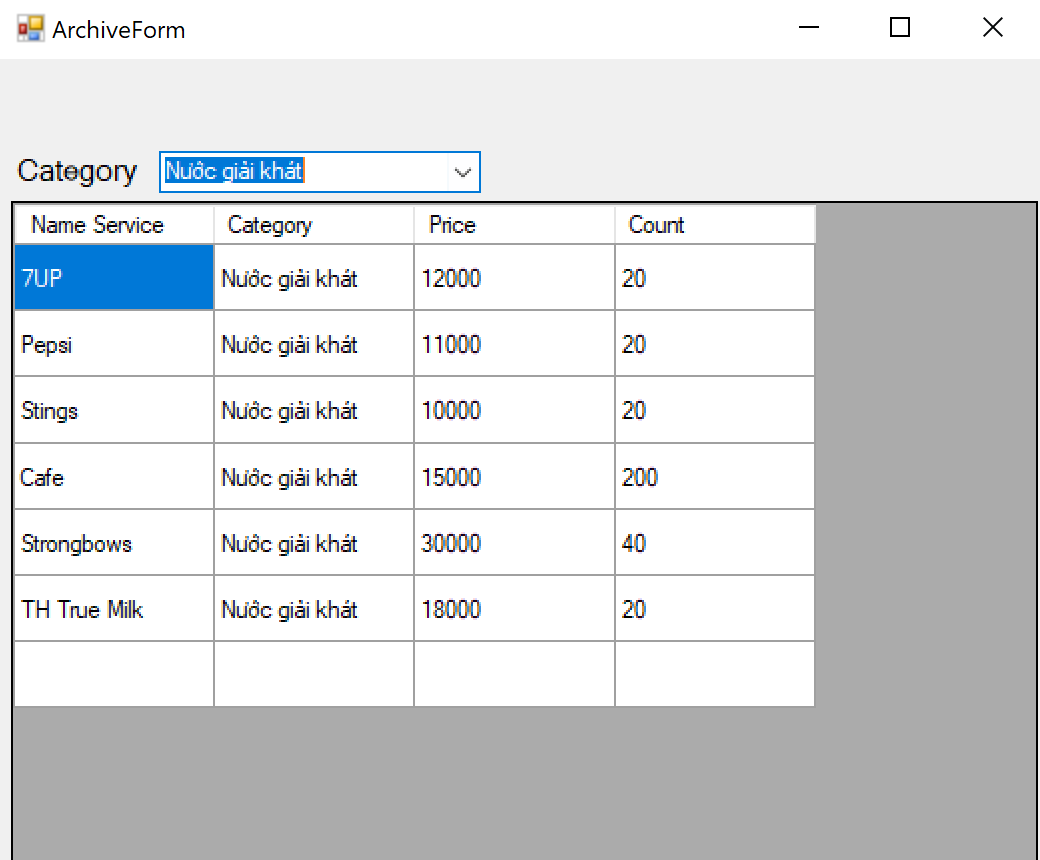
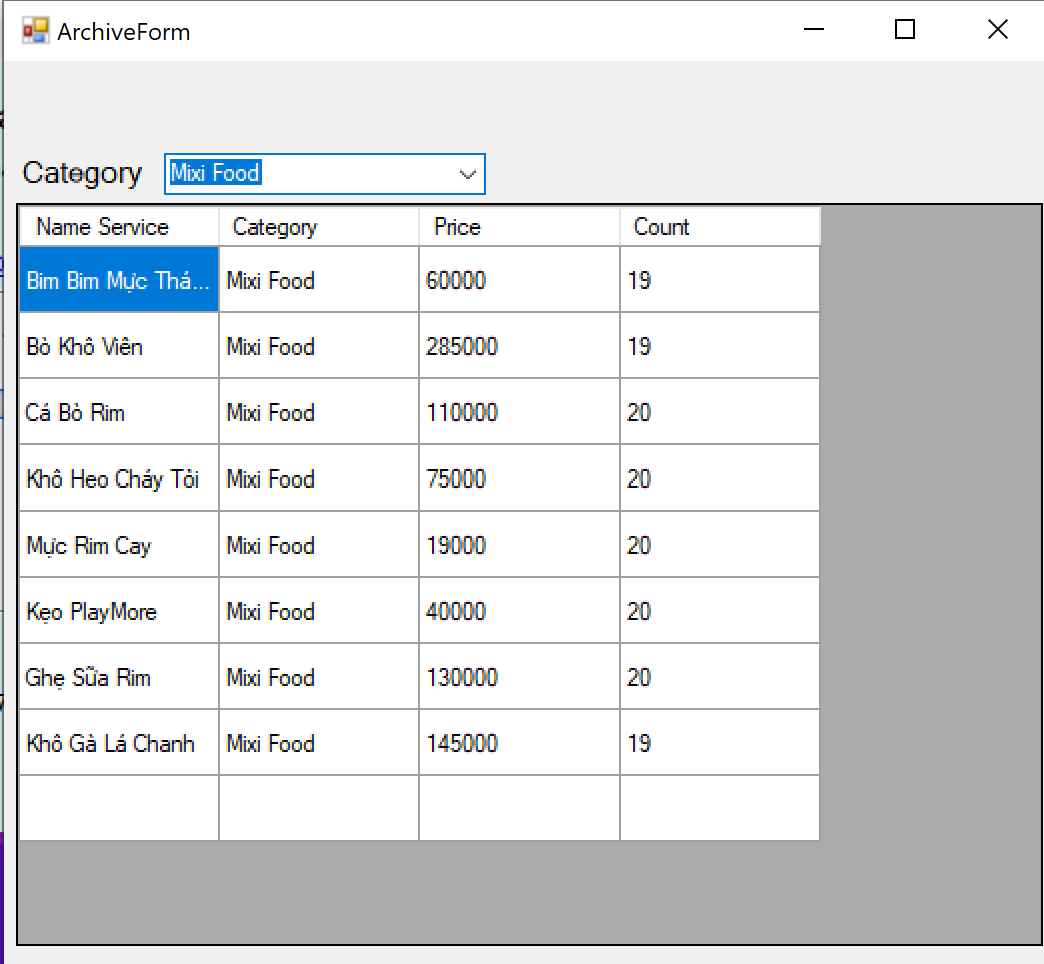
* **Quản lí -> Dịch vụ -> Cập nhật dịch vụ :** với chức năng này ta có thể cập nhật lại tên, giá và phân loại của dịch vụ.



* + 1. **Quản lí số lượng sản phẩm còn tồn kho :**
* **Quản lí -> Kho :** ở đây ta có thể xem được số lượng các sản phẩm đang hiện có :



Ta có thể xem theo loại của sản phẩm



* + 1. **Thêm/Xóa/Sửa User :**
    2. **Check In / Check Out :**
    3. **Phân công lịch làm việc:**
    4. **Xem danh sách khách hàng :**

1. **Các chức năng có thể thực hiện ở quyền Reception:**

* Các chức năng có thể thực hiện được ở quyền Reception tương tự như ở quyền Manager nhưng bị hạn chế một vài tính năng như :

+ Thêm/Xóa phòng

+ Thêm/Cập nhật dịch vụ

+Thêm/Xóa/Sửa nhân viên

+Phân công

1. **Các chức năng có thể thực hiện ở quyền Labor :**

* Các chức năng có thể thực hiện được ở quyền Labor :

+ CheckIn/Out khi đi làm

+Xem lịch của cá nhân