**DANH MỤC CÁC HÌNH VẼ**

[Hình 1. Form đăng nhập 24](#_Toc437347971)

[Hình 2. Form đăng ký 25](#_Toc437347972)

[Hình 3. Giao diện chính của người dùng 25](#_Toc437347973)

[Hình 4. Form chọn món của khách hàng 26](#_Toc437347974)

[Hình 5. Danh sách các Order đã đặt theo thời gian 26](#_Toc437347975)

[Hình 6. Xác nhận số lượng món và gửi Order 27](#_Toc437347976)

[Hình 7. Form cập nhật thay đổi lại các Order 27](#_Toc437347977)

[Hình 8. Form lựa chọn việc quản lý các nhà hàng sở hữu 28](#_Toc437347978)

[Hình 9. Form xóa địa chỉ của một nhà hàng 28](#_Toc437347979)

[Hình 10. Giao diện chính và form xử lý các Order từ khách hàng 29](#_Toc437347980)

[Hình 11. Giao diện quản lý các món ăn 30](#_Toc437347981)

[Hình 12. Giao diện của máy chủ 31](#_Toc437347982)

[Hình 13.Form nhập câu truy vấn. 32](#_Toc437347983)

[Hình 14. Form sơ đồ cơ sở dữ liệu 32](#_Toc437347984)

[Hình 15. Form hiển thị kết quả truy vấn 33](#_Toc437347985)

**MỤC LỤC**

[LỜI NÓI ĐẦU 2](#_Toc437347298)

[CHƯƠNG 1 : MÔ TẢ ỨNG DỤNG 3](#_Toc437347299)

[1.1 : Nhà hàng 3](#_Toc437347300)

[1.2 : Người dùng là khách hàng 3](#_Toc437347301)

[1.3 : Người dùng là quản lý nhà hàng 3](#_Toc437347302)

[1.4 : Tương tác giữa khách hàng và nhà hàng 4](#_Toc437347303)

[CHƯƠNG 2 : PHÂN TÍCH CƠ SỞ DỮ LIỆU 5](#_Toc437347304)

[2.1 : Các bảng 5](#_Toc437347305)

[2.2 : Sơ đồ thực thể liên kết 16](#_Toc437347306)

[2.3 : Các câu truy vấn sử dụng 17](#_Toc437347307)

[CHƯƠNG 3 : THIẾT KẾ GIAO DIỆN (Mô phỏng bằng Moqup) 25](#_Toc437347308)

[3.1 : Giao diện người dùng – khách hàng (Client) 25](#_Toc437347309)

[3.2 : Giao diện người dùng – quản lý nhà hàng (Client) 30](#_Toc437347310)

[3.3 : Giao diện máy chủ (Server) 31](#_Toc437347311)

[CHƯƠNG 4 : KẾT LUẬN 34](#_Toc437347312)

[4.1 : Ưu điểm 34](#_Toc437347313)

[4.2 : Nhược điểm 34](#_Toc437347314)

[4.3 : Hướng phát triển 35](#_Toc437347315)

[TÀI LIỆU THAM KHẢO 36](#_Toc437347316)

# LỜI NÓI ĐẦU

- Hiện nay các phần mềm công nghệ thông tin được sử dụng rộng rãi và được áp dụng để giải quyết nhiều vấn đề trong xã hội trong đó có nhu cầu của con người.

- Khi con người có nhu cầu ăn uống họ thường tìm đến những nhà hàng, việc phát triển một ứng dụng cho phép người dùng tiếp cận và đưa ra yêu cầu với nhà hàng sẽ giúp tiết kiệm được thời gian và chi phí.

► Vì vậy ứng dụng ResManager được viết bằng ngôn ngữ Java sẽ giải quyết được vấn đề về nhu cầu giữa khách hàng với nhà hàng và ngược lại giúp nhà hàng xử lý yêu cầu của khách hàng một cách nhanh chóng, hiệu quả.

# CHƯƠNG 1 : MÔ TẢ ỨNG DỤNG

## 1.1 : Nhà hàng

- Mỗi một nhà hàng đều có các cơ sở trên các địa bàn khác nhau.

- Mỗi cơ sở của nhà hàng sẽ cung cấp các món ăn theo quy định của nhà hàng đó.

- Nhà hàng tiếp nhận các yêu cầu đặt chỗ của khách hàng theo thời gian và các món.

## 1.2 : Người dùng là khách hàng

- Khách hàng đều có thông tin cụ thể, có tài khoản đăng nhập, số điện thoại,…

- Khách hàng yêu cầu đặt chỗ ăn ở một cơ sở nhà hàng cụ thể, lựa chọn món ăn tùy thích và đặt chỗ theo theo thời gian.

- Khách hàng có quyền hủy bỏ Order của mình khi Order trong quá trình cập nhật.

## 1.3 : Người dùng là quản lý nhà hàng

- Người dùng là quản lý nhà hàng cũng có thông tin cụ thể như khách hàng.

- Quản lý nhà hàng có quyền sửa đổi món ăn cung cấp, cập nhật giá tiền, xóa các món ăn.

- Quản lý nhà hàng có thể là quản lý của nhiều nhà hàng.Trong quá trình có thể tạo thêm một nhà hàng mới để quản lý cũng như xóa một nhà hàng khỏi danh sách quản lý.

- Quản lý nhà hàng có thể thêm các cơ sở mới của nhà hàng theo địa chỉ cụ thể.

- Quản lý nhà hàng chịu trách nhiệm xử lý các Order từ khách hàng.

## 1.4 : Tương tác giữa khách hàng và nhà hàng

- Khách hàng yêu cầu đặt chỗ ăn tại một nhà hàng, yêu cầu đó sẽ được nhà hàng chốt lại khi gần đến ngày yêu cầu, trong thời gian trước đó khách hàng có thể thay đổi và cập nhật Order của mình tùy thích.

- Nếu nhà hàng không chấp nhận Order khách hàng thì khách hàng phải hủy Order đó.

- Nếu khách hàng muốn hủy Order của mình thì gửi yêu cầu đợi nhà hàng hủy nó đi.

- Các Order đã chốt sẽ được phục vụ sau đó khách hàng có thể xem hóa đơn và Order đó sẽ bị xóa.

- Các Order tồn tại sau ngày đã yêu cầu quá 1 ngày sẽ tự động bị xóa.

# CHƯƠNG 2 : PHÂN TÍCH CƠ SỞ DỮ LIỆU

## 2.1 : Các bảng

a) ACCOUNT

Các thuộc tính:

+ AID : ID của mỗi người dùng, là thuộc tính khóa để nhận biết người dùng.

+ Username : tài khoản đại diện của người dùng để đăng nhập vào ứng dụng.

+ Password : mật khẩu cho phép người dùng đăng nhập thành công nếu nhập đúng.

+ Full Name : tên đầy đủ của người dùng.

+ Birthday : ngày tháng năm sinh của người dùng.

+ Phone Number : Số điện thoại của người dùng.

+ Sex : Giới tính của người dùng.

+ Status : Trạng thái online của người dùng.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Account` (*

*`AID` VARCHAR(8) NOT NULL,*

*`Username` VARCHAR(45) NOT NULL,*

*`Password` VARCHAR(45) NOT NULL,*

*`FullName` VARCHAR(45) NOT NULL,*

*`Birthday` DATE NULL DEFAULT NULL,*

*`PhoneNumber` CHAR(13) NOT NULL,*

*`Sex` VARCHAR(6) NOT NULL,*

*`Status` INT NULL DEFAULT 0,*

*PRIMARY KEY (`AID`),*

*INDEX `AIDIndex` (`AID` ASC),*

*UNIQUE INDEX `Username\_UNIQUE` (`Username` ASC),*

*UNIQUE INDEX `PhoneNumber\_UNIQUE` (`PhoneNumber` ASC))*

b) RESTAURANT

Các thuộc tính:

+ ResID : ID của mỗi nhà hàng, là thuộc tính khóa để nhận biết nhà hàng.

+ Resname : Tên của nhà hàng.

+ AID : là khóa ngoại để xác định người quản lý nhà hàng.

Câu lệnh tạo bảng:

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Restaurant` (*

*`ResID` VARCHAR(8) NOT NULL,*

*`Resname` VARCHAR(45) NOT NULL,*

*`AID` VARCHAR(8) NOT NULL,*

*PRIMARY KEY (`ResID`, `AID`),*

*INDEX `ResIDIndex` (`ResID` ASC),*

*INDEX `AIDIndex` (`AID` ASC),*

*UNIQUE INDEX `Resname\_UNIQUE` (Resname` ASC),*

*CONSTRAINT `fk\_AID\_Restaurant FOREIGN KEY (`AID`) REFERENCES `BT\_LON`.`Account` (`AID`))*

c) SEQUENCE RESTAURANT

Các thuộc tính :

+ Address : Địa chỉ, là khóa chính để xác định mỗi cơ sở của một nhà hàng.

+ ResID : ID của nhà hàng, cho biết chi nhánh thuộc nhà hàng nào.

Câu lệnh tạo bảng:

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`SequenceRestaurant` (*

*`Address` VARCHAR(45) NOT NULL,*

*`ResID` VARCHAR(8) NOT NULL,*

*PRIMARY KEY (`Address`, `ResID`),*

*INDEX `ResIDIndex` (`ResID` ASC),*

*INDEX `AddressIndex` (`Address` ASC),*

*UNIQUE INDEX `Address\_UNIQUE` (`Address` ASC),*

*CONSTRAINT `fk\_ResID\_SequenceRestaurant FOREIGN KEY (`ResID`)REFERENCES `BT\_LON`.`Restaurant` (`ResID`))*

d) FOODSET (Các món ăn )

Các thuộc tính:

+ Foodname: thuộc tính duy nhất, cũng là khóa chính xác định tên của món ăn.

Câu lệnh tạo bảng:

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`FoodSet` (*

*`Foodname` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`Foodname`),*

*INDEX `FoodnameIndex`(`Foodname` ASC)*

*)*

e) PROVIDE (Các món ăn của nhà hàng)

Các thuộc tính:

+ ResID : ID của nhà hàng cung cấp, là khóa ngoại.

+ Foodname: Tên món ăn , là khóa ngoại .

+ Cost : Giá tương ứng với mỗi món.

+ Description : Mô tả cho món ăn.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Provide` (*

*`ResID` VARCHAR(8) NOT NULL,*

*`Foodname` VARCHAR(45) NOT NULL,*

*`Cost` DOUBLE NULL DEFAULT NULL,*

*`DescribeFood` VARCHAR(45) NULL DEFAULT NULL,*

*PRIMARY KEY (`ResID`, `Foodname`),*

*INDEX `FoodnameIndex` (`Foodname` ASC),*

*INDEX `ResIDIndex` (`ResID` ASC),*

*CONSTRAINT `fk\_Foodname\_Provide`*

*FOREIGN KEY (`Foodname`)*

*REFERENCES `BT\_LON`.`FoodSet` (`Foodname`),*

*CONSTRAINT `fk\_ResID\_Provide`*

*FOREIGN KEY (`ResID`)*

*REFERENCES `BT\_LON`.`Restaurant` (`ResID`))*

f) RESERVATION (Yêu cầu đặt chỗ từ khách hàng)

Các thuộc tính:

+ AID :khóa ngoại, xác định khách hàng yêu cầu.

+ ResAddress : khóa ngoại, xác định chi nhánh nhà hàng được yêu cầu đặt chỗ.

+ Foodname : tên món yêu cầu.

+ Time : thời gian đến ăn của khách hàng , là khóa chính.

+ Cost : là giá tương ứng của món yêu cầu.

+ Quantity : số lượng mỗi món.

+ StatusReser : trạng thái hiện tại của Yêu cầu.

+ CustomerRight : quyền tác động từ phía khách hàng.

+ RestaurantRight : quyền tác động từ phía nhà hàng.

+ NextStatusForCustomer : trạng thái có thể xảy ra tiếp theo khi bị khách hàng tác động.

+ NextStatusForRestaurant : trạng thái có thể xảy ra tiếp theo khi bị nhà hàng hàng tác động.

Câu lệnh tạo bảng:

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Reservation` (*

*`AID` VARCHAR(8) NOT NULL,*

*`ResAddress` VARCHAR(45) NOT NULL,*

*`Foodname` VARCHAR(45) NOT NULL,*

*`Time` DATETIME NULL DEFAULT NULL,*

*`Quantity` MEDIUMTEXT NULL DEFAULT NULL,*

*`StatusReser` VARCHAR(10) NOT NULL,*

*`Cost` DOUBLE NULL,*

*`CustomerRight` VARCHAR(45) NOT NULL,*

*`RestaurantRight` VARCHAR(45) NOT NULL,*

*`NextStatusForCustomer` VARCHAR(45) NOT NULL,*

*`NextStatusForRestaurant` VARCHAR(45) NOT NULL,*

*KEY(`Time`),*

*PRIMARY KEY (`AID`, `ResAddress`, `Foodname`, `Time`),*

*INDEX `ResAddressIndex` (`ResAddress` ASC),*

*INDEX `AIDIndex` (`AID` ASC),*

*INDEX `FoodnameIndex` (`Foodname` ASC),*

*INDEX `TimeIndex` (`Time` ASC),*

*CONSTRAINT `fk\_ResAddress\_Reservation`*

*FOREIGN KEY (`ResAddress`)*

*REFERENCES `BT\_LON`.`SequenceRestaurant` (`Address`),*

*CONSTRAINT `fk\_AID\_Reservation`*

*FOREIGN KEY (`AID`)*

*REFERENCES `BT\_LON`.`Account` (`AID`),*

*CONSTRAINT `fk\_Foodname\_Reservation`*

*FOREIGN KEY (`Foodname`)*

*REFERENCES `BT\_LON`.`FoodSet` (`Foodname`))*

g) ROLE (vai trò của người dùng)

Các thuộc tính :

+ RLID : ID của Role, là khóa chính xác định mỗi vai trò.

+ RoleName : Tên của vai trò.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Roles` (*

*`RLID` VARCHAR(8) NOT NULL,*

*`Rolename` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`RLID`))*

h) OBJECTS (Các đối tượng trong cơ sở dữ liệu)

Các thuộc tính:

+ OBJID : ID của mỗi đối tượng, là khóa chính.

+ ObjectName : tên của đối tượng.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Objects` (*

*`OBJID` VARCHAR(8) NOT NULL,*

*`Objectname` VARCHAR(45) NOT NULL,*

*PRIMARY KEY (`OBJID`))*

i) OPERATIONS (các thao tác của người dùng trên dữ liệu)

Các thuộc tính :

+ OPID : ID xác định mỗi thao tác, là khóa chính.

+ Operation Name : tên của thao tác.

+ReadOp : quyền đọc (kiểu BOOLEAN).

+InsertOp : quyền thêm dữ liệu mới (kiểu BOOLEAN).

+UpdateOp : quyền thay đổi dữ liệu (kiểu BOOLEAN).

+DeleteOp : quyền xóa dữ liệu (kiểu BOOLEAN).

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Operations` (*

*`OPID` VARCHAR(8) NOT NULL,*

*`OperationName` VARCHAR(45) NOT NULL,*

*`ReadOp` TINYINT(1) NOT NULL,*

*`InsertOp` TINYINT(1) NOT NULL,*

*`UpdateOp` TINYINT(1) NOT NULL,*

*`DeleteOp` TINYINT(1) NOT NULL,*

*PRIMARY KEY (`OPID`))*

j) PERMISSIONS (các quyền cụ thể)

Các thuộc tính:

+ PRMID : ID xác định quyền,là khóa chính.

+ PermissionName : tên của quyền.

+ OBJID : ID của 1 đối tượng bị tác động (Object),là khóa ngoại.

+ OPID : ID của 1 thao tác (Operation), là khóa ngoại.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`Permissions` (*

*`PRMID` VARCHAR(8) NOT NULL,*

*`PermissionName` VARCHAR(45) NOT NULL,*

*`OBJID` VARCHAR(8) NOT NULL,*

*`OPID` VARCHAR(8) NOT NULL,*

*PRIMARY KEY (`PRMID`, `OBJID`, `OPID`),*

*INDEX `PRMIDIndex` (`PRMID` ASC),*

*INDEX `OPIDIndex` (`OPID` ASC),*

*INDEX `OBJIDIndex` (`OBJID` ASC),*

*CONSTRAINT `fk\_OBJID\_Permissions`*

*FOREIGN KEY (`OBJID`)*

*REFERENCES `BT\_LON`.`Objects` (`OBJID`),*

*CONSTRAINT `fk\_OPID\_Permissions`*

*FOREIGN KEY (`OPID`)*

*REFERENCES `BT\_LON`.`Operations` (`OPID`))*

k) PERMISSIONSASSIGNMENT (Phân quyền cho các vai trò (ROLE))

Các thuộc tính :

+ RLID : ID của ROLE được phân quyền, là khóa ngoại.

+ PRMID : ID của PERMISSION xác định quyền được phân định, khóa ngoại.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`PermissionsAssignment` (*

*`RLID` VARCHAR(8) NOT NULL,*

*`PRMID` VARCHAR(8) NOT NULL,*

*PRIMARY KEY (`RLID`, `PRMID`),*

*INDEX `PRMIDIndex` (`PRMID` ASC),*

*INDEX `RLIDIndex` (`RLID` ASC),*

*CONSTRAINT `fk\_RLID\_PermissionsAssignment`*

*FOREIGN KEY (`RLID`)*

*REFERENCES `BT\_LON`.`Roles` (`RLID`),*

*CONSTRAINT `fk\_PRMID\_PermissionsAssignment`*

*FOREIGN KEY (`PRMID`)*

*REFERENCES `BT\_LON`.`Permissions` (`PRMID`))*

l) ACCOUNTASSIGNMENT (Phân vai trò cho mỗi người dùng)

Các thuộc tính :

+ AID : ID của người dùng, là khóa ngoại.

+ RLID : ID của ROLE , xác dịnh vai trò cảu người dùng,là khóa ngoại.

Câu lệnh tạo bảng :

*► CREATE TABLE IF NOT EXISTS `BT\_LON`.`AccountAssignment` (*

*`AID` VARCHAR(8) NOT NULL,*

*`RLID` VARCHAR(8) NOT NULL,*

*PRIMARY KEY (`AID`, `RLID`),*

*INDEX `AIDIndex` (`AID` ASC),*

*INDEX `RLIDIndex` (`RLID` ASC),*

*CONSTRAINT `fk\_AID\_AccountAssignment`*

*FOREIGN KEY (`AID`)*

*REFERENCES `BT\_LON`.`Account` (`AID`),*

*CONSTRAINT `fk\_RLID\_AccountAssignment`*

*FOREIGN KEY (`RLID`)*

*REFERENCES `BT\_LON`.`Roles` (`RLID`)*

## 2.2 : Sơ đồ thực thể liên kết

1

1

m

1

1

1

FoodSet

k

1

m

1

n

1

n1

n1

n

1

Reservation

SequenceRestaurant

Permissions

Account assignment

Permissions Assignment

Objects

Restaurant

Account

Roles

Operations

n

1

Provide

m

1

m

m

## https://scontent-hkg3-1.xx.fbcdn.net/hphotos-xft1/v/t34.0-12/12357951_971233506280517_2069906269_n.jpg?oh=a21be741ba5a7a0fe272466326462c22&oe=56695340

## 2.3 : Các câu truy vấn sử dụng

► SELECT Address from sequencerestaurant WHERE ResID = ?

► SELECT DISTINCT FullName,Birthday,PhoneNumber,Time,AID

FROM reservation NATURAL JOIN account

WHERE ResAddress = ?

► SELECT Foodname,Cost,DescribeFood FROM provide WHERE ResID = ?

► SELECT

Foodname,Cost,Quantity,StatusReser,NextStatusForRestaurant,RestaurantRight

FROM reservation WHERE ResAddress = ? AND AID = ? AND Time = ?

► SELECT SUM(Cost \* Quantity)

FROM reservation

WHERE ResAddress = ? AND AID = ? AND Time = ?

► UPDATE reservation

SET StatusReser = 'Lock', CustomerRight = 'Change Status',RestaurantRight =

'Delete After Order Day 1 Day', NextStatusForCustomer ='OK' ,

NextStatusForRestaurant = 'Nothing' WHERE AID = ? AND ResAddress = ?

AND Time = ? AND Foodname = ?

► DELETE FROM reservation

WHERE AID = ? AND ResAddress = ? AND Time = ? AND Foodname = ?

► UPDATE reservation

SET StatusReser = 'Deny', CustomerRight = 'Delete',

RestaurantRight = 'Nothing', NextStatusForCustomer='Nothing' ,

NextStatusForRestaurant = 'Nothing'

WHERE AID = ? AND ResAddress = ? AND Time = ? AND Foodname = ?

► DELETE FROM reservation

WHERE StatusReser = 'OK' AND ResAddress = ?

► SELECT DISTINCT TIME

FROM reservation

WHERE StatusReser IN('Lock','Updating') AND ResAddress = ?

► DELETE FROM reservation

WHERE Time = ? AND StatusReser = 'Lock' AND ResAddress = ?

► INSERT INTO foodset values( ? )

INSERT INTO provide

values( ? , ? , ? , ? )

► UPDATE provide

SET Cost = ? , DescribeFood = ?

WHERE ResID = ? AND Foodname = ?

► DELETE FROM provide

WHERE ResID = ? AND Foodname = ‘myfoodname’

► INSERT INTO sequencerestaurant

values (‘mynewaddress’, ?)

► DELETE FROM FoodSet

WHERE Foodname NOT IN (SELECT Foodname FROM reservation)

AND Foodname NOT IN (SELECT Foodname FROM Provide)

► SELECT Resname,Address FROM Restaurant NATURAL JOIN SequenceRestaurant

► SELECT \* FROM Account

WHERE Username = ? AND `Password` = ?

► SELECT PhoneNumber FROM Account

WHERE AID = (SELECT AID FROM Restaurant WHERE Resname = ? )

► SELECT FoodName,Cost from Provide

WHERE ResID = (SELECT ResID FROM SequenceRestaurant WHERE

Address = ? )

► INSERT INTO Reservation value (? , ? , ? , ? , ? ,'Updating', ?)

► SELECT ResID,Resname FROM Restaurant

WHERE AID = ?

► SELECT FoodName,Cost FROM Provide

WHERE ResID = (SELECT ResID FROM SequenceRestaurant WHERE

Address =?) AND FoodName = ?

► SELECT DISTINCT `Time` FROM `Reservation`

WHERE AID = ? AND `ResAddress` = ?

► SELECT Foodname,Quantity,Cost,StatusReser FROM `Reservation`

WHERE AID = ? AND ResAddress = ? AND `Time` = ?

► UPDATE `Reservation`

SET `StatusReser` = 'Delete'

WHERE AID = ? AND `ResAddress` = ? AND `Time` = ? AND `Foodname`

= ?

► UPDATE `Reservation`

SET `Time` = ?

WHERE AID = ? AND `ResAddress` = ? AND `Time` = ? AND `StatusReser`

= 'Updating'"

► SELECT DINSTINCT `Time` FROM `Reservation`

WHERE AID = ? AND ResAddress = ?

► UPDATE `Reservation`

SET `Quantity` = ?

WHERE AID = ? AND `ResAddress` = ? AND `Time` = ? AND

`StatusReser` = 'Updating' AND Foodname = ?

► DELETE FROM `Reservation`

WHERE ResAddress IN (SELECT Address FROM SequenceRestaurant

WHERE ResID = ? )

► DELETE FROM `Reservation`

WHERE ResAddress IN (SELECT Address FROM SequenceRestaurant

WHERE ResID = ? );

► DELETE FROM `Provide` WHERE ResID = ? ;

► DELETE FROM `SequenceRestaurant` WHERE ResID = ? ;

► DELETE FROM `Restaurant` WHERE ResID = ?

► INSERT INTO Restaurant VALUE ( ? , ? , ? );

► INSERT INTO SequenceRestaurant VALUE( ? , ?)

► DELETE FROM Reservation

WHERE AID = ? AND ResAddress = ? AND Foodname = ? AND Time = ?

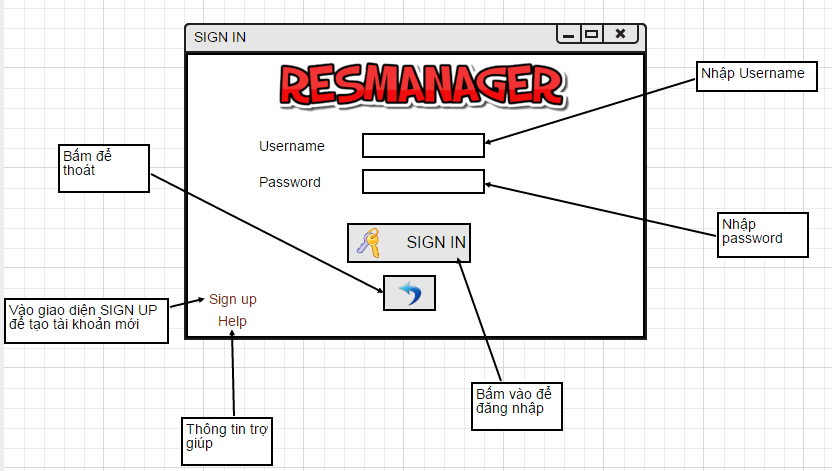
► UPDATE `Reservation`

SET `StatusReser` = 'OK'

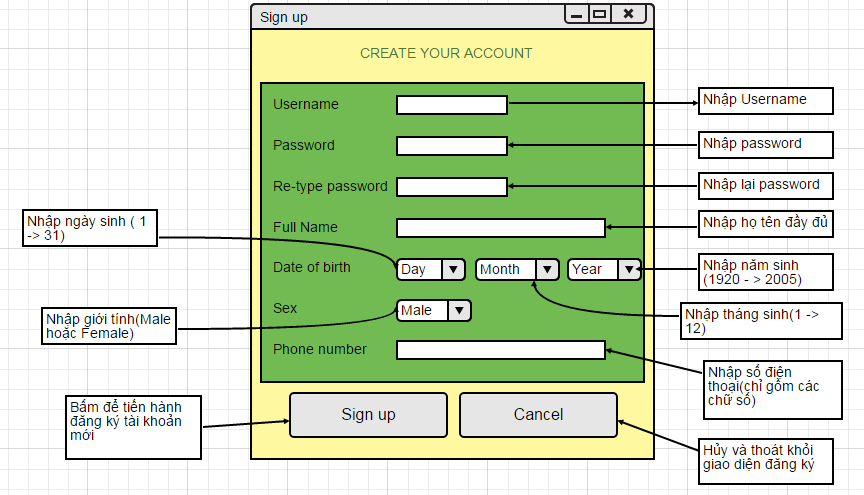
WHERE AID = ? AND `ResAddress` = ? AND `Time` = ? AND `Foodname` = ?

# CHƯƠNG 3 : THIẾT KẾ GIAO DIỆN (Mô phỏng bằng Moqup)

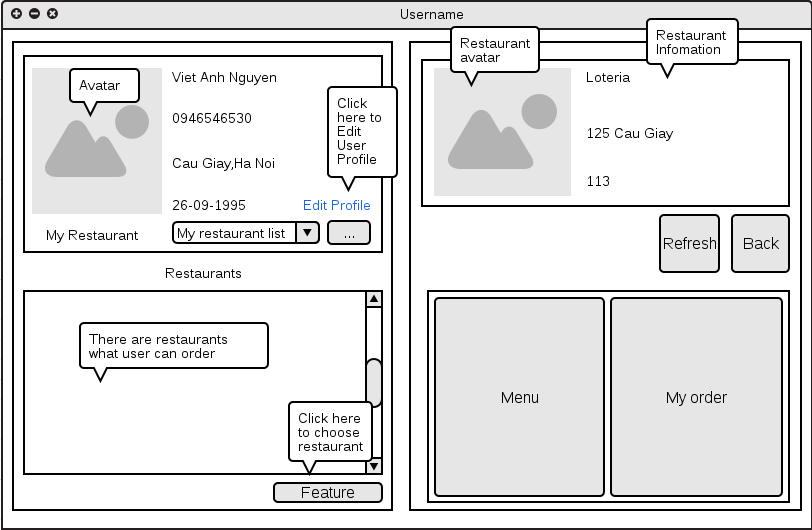
## 3.1 : Giao diện người dùng – khách hàng (Client)



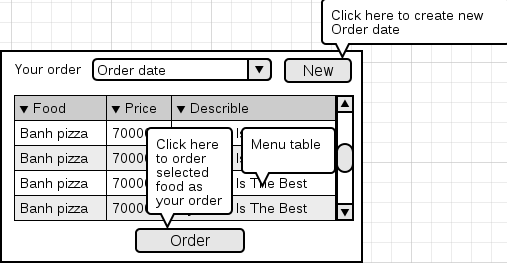
Hình 1. Form đăng nhập



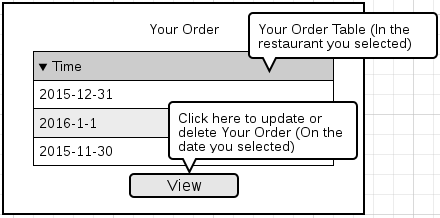
Hình 2. Form đăng ký



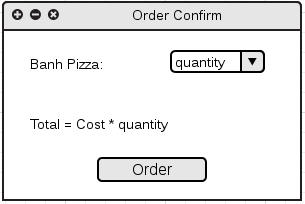
Hình 3. Giao diện chính của người dùng



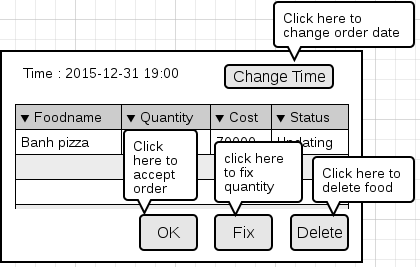
Hình 4. Form chọn món của khách hàng



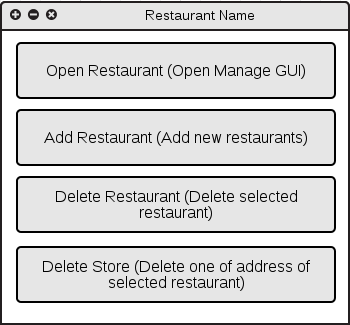
Hình 5. Danh sách các Order đã đặt theo thời gian



Hình 6. Xác nhận số lượng món và gửi Order



Hình 7. Form cập nhật thay đổi lại các Order

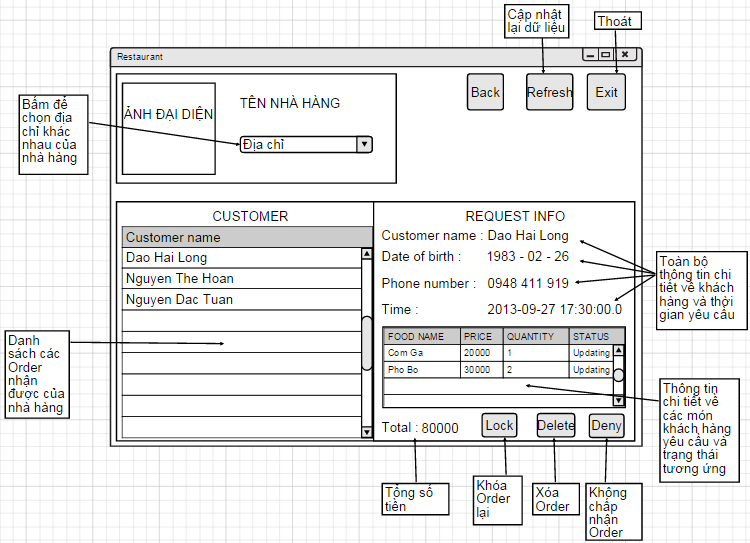


Hình 8. Form lựa chọn việc quản lý các nhà hàng sở hữu

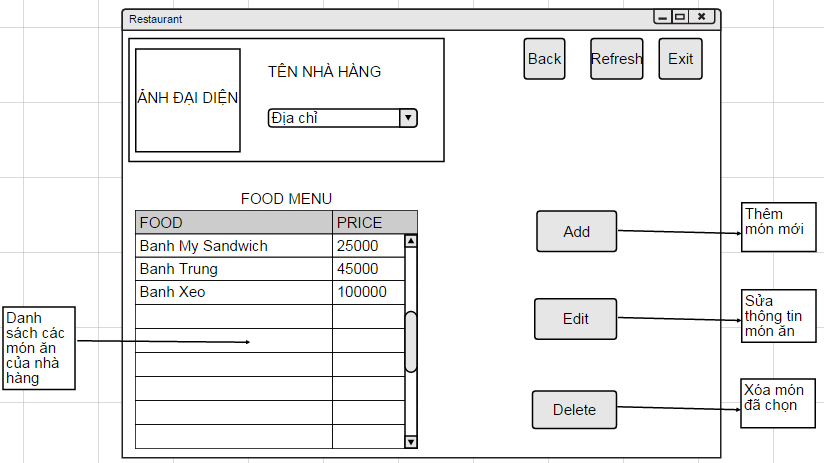
## https://fbcdn-sphotos-e-a.akamaihd.net/hphotos-ak-xlt1/v/t34.0-12/12335916_971228119614389_728485534_n.jpg?oh=18b01858a7b2e33e7afa30b65aa648db&oe=566855DF&__gda__=1449725644_3a661128d78291cc97804be5d1ac90ab

Hình 9. Form xóa địa chỉ của một nhà hàng

## 3.2 : Giao diện người dùng – quản lý nhà hàng (Client)

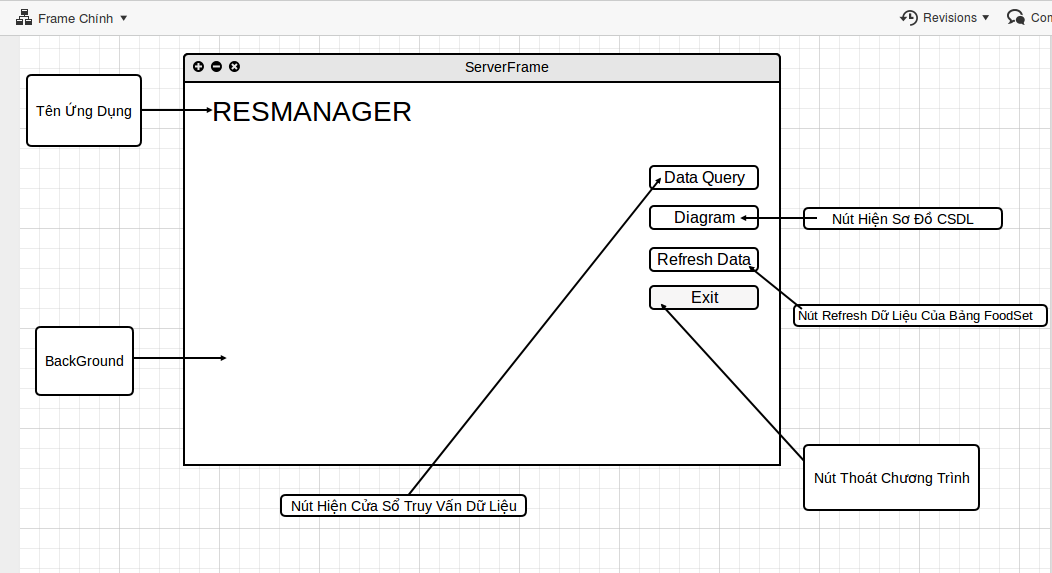


Hình 10. Giao diện chính và form xử lý các Order từ khách hàng

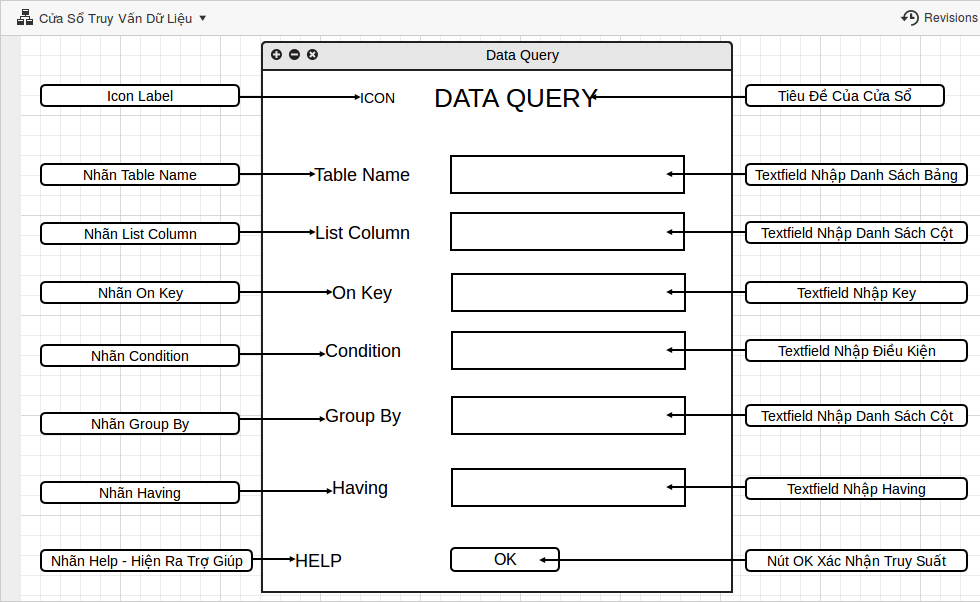


Hình 11. Giao diện quản lý các món ăn

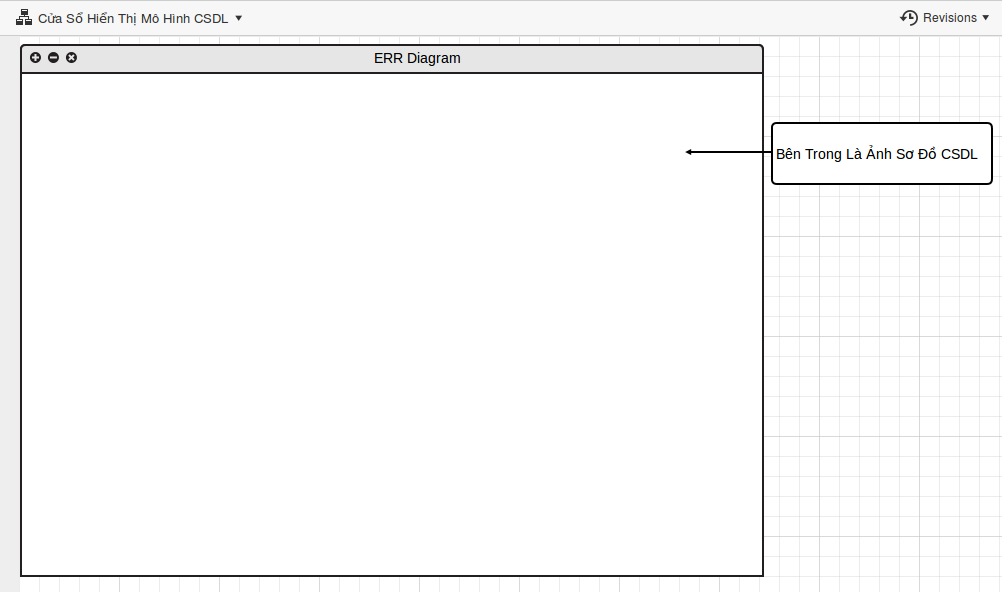
## 3.3 : Giao diện máy chủ (Server)



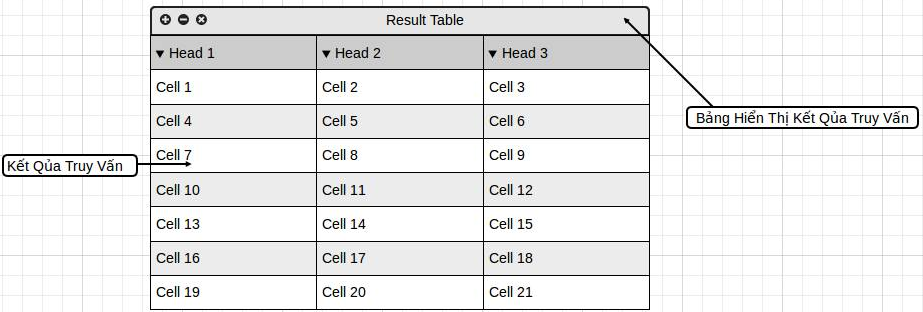
Hình 12. Giao diện của máy chủ



Hình 13.Form nhập câu truy vấn.



Hình 14. Form sơ đồ cơ sở dữ liệu



Hình 15. Form hiển thị kết quả truy vấn

# CHƯƠNG 4 : KẾT LUẬN

## 4.1 : Ưu điểm

+ Ứng dụng tích hợp được chức năng của cả khách hàng và người quản lý nhà hàng.

+ Ứng dụng theo mô hình Client – Server giúp người dùng chỉ cần cài đặt và sử dụng ứng dụng. Dữ liệu được lưu trữ ở máy chủ Server.

+ Tiếp cận mô hình dữ liệu RBAC (Role Base Access Control) để phân hoạch chức năng và vai trò cho mỗi người dùng.

+ Giải quyết và xử lý được xung đột dữ liệu giữa khách hàng và nhà hàng trong quá trình tương tác qua ứng dụng, tôn trọng khách hàng.

+ Giao diện thân thiện, gọn gàng, giúp người dùng thao tác một cách đơn giản và trực quan.

## 4.2 : Nhược điểm

+ Việc đưa chức năng người quản lý nhà hàng và khách hàng vào cùng một giao diện sẽ gây nguy cơ về vấn đề bảo mật.

+ Cơ sở dữ liệu và các chức năng vẫn còn khá đơn giản, đặc biệt là phần chức năng của nhà hàng trong việc quản lý khách hàng.

+ Mô hình RBAC (Role Base Access Control) vẫn còn sơ sài , chưa rõ ràng, mỗi nhà hàng vẫn chỉ do duy nhất một người quản lý và điều hành.

+ Sử dụng ngôn ngữ lập trình Java gặp nhiều hạn chế trong việc xây dựng ứng dụng online.

+ Ứng dụng online nên phụ thuộc vào tốc độ truyền mạng.

## 4.3 : Hướng phát triển

+ Ứng dụng cần tập trung và phát triền về phần quản lý nhà hàng một cách chi tiết hơn.

+ Đưa thêm các vai trò quản lý nhà hàng khác một cách đa dạng, rõ ràng qua mô hình RBAC.

+ Cập nhật, mở rộng cơ sở dữ liệu cho phép người dùng có nhiều lựa chọn hơn trong chương trình.

## TÀI LIỆU THAM KHẢO

[1] Giáo Trình Lập Trình Java – Đoàn Văn Ban, Đoàn Văn Trung

[2] http://docs.oracle.com/javase/7

[3] http://stackoverflow.com

[4] http://codebrach.com

[5] http://java2s.com