

**Tunga Restaurant**

**Development’s Guide**

**Faculty: DUONG TRONG PHU SON**

**Batch No:**

**Group No: 4**

**Student:**

|  |  |  |
| --- | --- | --- |
| **No** | **Full Name** | **Roll No** |
| **1** | **Tran Huy Hieu** | **Student724546** |
| 2 | Nguyen Dac Thien | Student609111 |
| 3 | Truong Phuc Chau | Student609432 |
| 4 | Pham Anh Dung | Student524187 |
| 5 | Nguyen Ngoc Hoai Anh | Student601653 |

**REVIEW 1**

**1. Introduction**

**FUN ‘N’ FOOD** is a big name in the hospitality business (Hotels & Resorts) and has a very successful independent chain of restaurants, **TUNGA,** in the country. **TUNGA** restaurants are known for their wide range of continental cuisine and consummate style.

**TUNGA** provides a range of fully-equipped, private meeting rooms, and regal banquet halls which offer the perfect combination of efficiency and convenience. They are also popular among school & college students for range of bakery & fast food products.

**2. Existing Scenario**

With increasing popularity and the customer base, the organization has decided to go techno savvy to meet customer expectations with respect to its exclusive services in the market.

The organization has decided to reach out to its customers via online services through a web site on the internet.

The web site talks about other business aspects of the firm and has online reservation and home delivery service as one of its premium services.

This project will focus on the online booking and the home delivery aspect only and to design and build a site which deals with the same.

**2.1. Customer Requirement Specification**

The site should allow customer to choose the services offered by the restaurant.

2.1.1 Online table reservation in the restaurant for a specific date and time. This will

a. Provide a list of restaurants in the country so that the user can choose accordingly.

b. Provide a list of tables available for online reservation at different dining rooms in the restaurant.

c. Mention the number of guests that can be accommodated on the particular table.

d. Accept the booking for tables.

e. Online table reservations are done 6hrs in advance for the current date.

f. Table reservation can be done up to one month in advance.

g. Give a token number to the customer as an acknowledgement of booking.

2.1.2. Online order for food under restaurant’s exclusive home delivery service.

a. This facility is valid for 24hrs. from the time of order.

b. Accept the customer’s order so that the nearest restaurant can service the order in stipulated time (60 - 120 Minutes).

c. Record the customer’s name, residence address and contact numbers and other vital details if required.

d. Facilitate with the choice of food keeping Vegetarian and non-vegetarian customers in mind.

e. Provide the menu list with their price tags accordingly and also mention the number of persons that can be served with one unit of order. This will help the customer to decide the quantity to be ordered.

f. Display the list items select from the menu and exact amount (itemized bill) that the customer is supposed to pay (inclusive of all taxes applicable).

g. Give a token number as an acknowledgement to the customer that the order has been noted down by the restaurant.

h. Show the approximate time (in minutes) in which the ordered will be delivered to the customer.

2.1.3. The site should have the appropriate look and feel suitable to the restaurant’s reputation and status.

**2.2. Functional Requirement Specification**

2.2.1. Staff:

Allow employee to manage (view, update, delete) Takeaway, Order, Reservation, Feedback table.

2.2.2. Admin:

a. Allow admin to manage (view, create, update, delete) Restaurant, Dish, Table, Category, Staff table.

b. Allow admin to manage (view) Customer table.

2.2.3. Customer:

a. Allow user to view website information.

b. Allow user to add feedback and view reply.

c. Allow user to change password.

d. Allow user to edit account’s information.

e. Allow user to do reservation and cancel reservation.

f. Allow user to order takeaway.

2.2.4. Guest:

Allow guest to view website information.

Allow guest to register as a customer.

**2.3. System Requirement**

***2.3.1. Hardware***

**Web Server**

|  |  |
| --- | --- |
| Processor | Pentium IV or higher. |
| Memory | 1024MB RAM or greater. |
| Modem/ADSL | Connecting to internet 24/24 |

**Client**

|  |  |
| --- | --- |
| Processor | Core i3 or higher. |
| Memory | 1024MB RAM or greater. |
| Monitor | VGA (1024x768) or higher resolution with 32 bits. |
| Pointing Device | Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device. |
| Modem/ADSL | Connecting to internet |

***2.3.2. Software***

**Web Server**

|  |  |
| --- | --- |
| Operation System | Windows XP/7/linux |
| Browser | Internet Explorer 10 or higher. |
| Database | Microsoft SQL Server 2008 or higher |
| Software | Java SDK 1.7 or higher |
| Application server | Glassfish 4.0 |
| Framework | JSF 2.2, Prime Faces 5.1 or higher, EJB 3.0 |

**Client**

|  |  |
| --- | --- |
| Operation System | Windows XP or later. |
| Browser | Internet Explorer 10.0 or higher/Chrome/FireFox. |

**2.4. Development Software**

* Microsoft Windows 7
* Netbean 7.4
* Java SDK 1.7
* Microsoft SQL Server 2012
* Chrome
* JSF Framework
* Glassfish 4.0

**2.5. Technology**

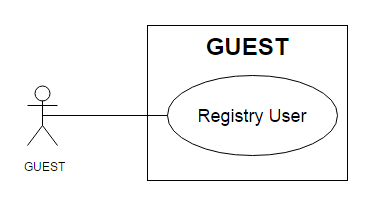
* JSF
* J2EE
* HTML, JQuery

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader |  |
| **Date** | 13-August-2015 |  |

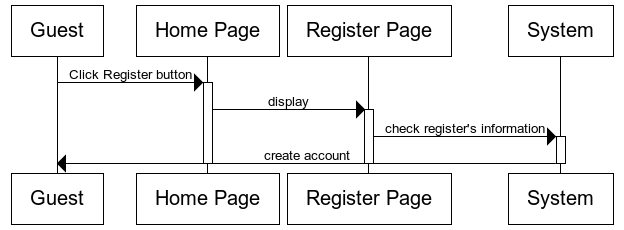
**Review 2**

**3. Use Case and Sequence Diagram**

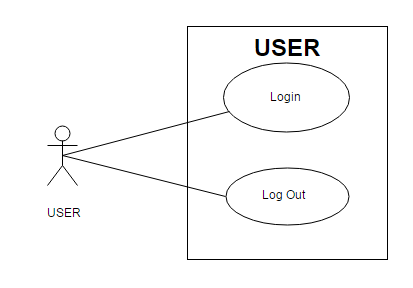
**3.1. Guest:**



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Register user | |
| **Actors** | Guest | |
| **Description** | Guest wants to register with the website to become a customer. | |
| **Requirements** | Guest provides information | |
| **Pre-conditions** |  | |
| **Post-conditions** | Success: Guest creates a account in database  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Register’ link in Home page  3. Actor inputs register’s information and click ‘Register’ button. | System Responses:  2. System redirects to Register page and displays Registration form with the following control :  - ‘UserName’ text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Name’ text field  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Register’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts the account into database, show message: ‘Your account has been created!’ and then redirects to Login page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank.  4. Actor lefts ‘Confirm Password’ text field blank.  5. Password and confirm password don’t match.  6. Actor lefts ‘Name’ text field blank.  7. Actor lefts ‘Address’ text field blank.  8. Actor lefts ‘Phone’ text field blank.  9.Input phone not in numeric or phone length not in range 10 – 11 character.  12. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’.  - ‘Confirm Password can’t be blank’.  - ‘Confirm password not match password.’  - ‘Name cannot be blank’.  - ‘Address cannot be blank’.  - ‘Phone cannot be blank’.  - ‘The phone number should be numeric only and must be between 10 and 11 digits’.  - ‘Email is invalid’. |

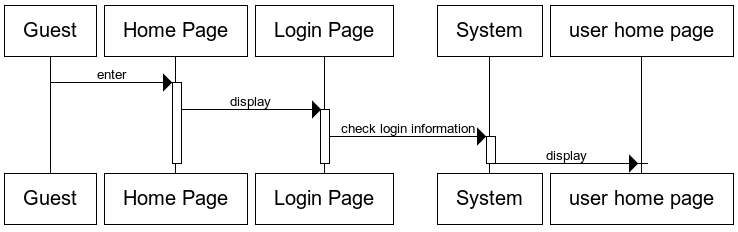


**3.2 Login/Log Out:**



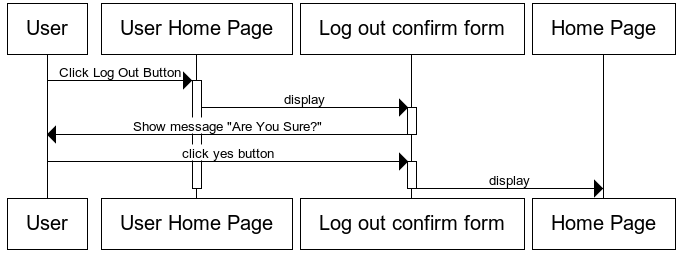
**3.2.1 Login:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Login | |
| **Actors** | Customer, Admin or Staff. | |
| **Description** | Customer, Admin or Staffwants to login to website. | |
| **Requirements** | Actor provides username and password | |
| **Post-conditions** | Success: User logged in to website.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor go to Home page of website.  3. Actor inputs UserName and Password, then click the Login button. | System Responses:  2. System redirects to Home page and displays Login form on the left with the following control:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System redirects to Actor’s Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor input invalid username and password. | System Responses:  - System redirect to Login page with the following controls:  - ‘UserName’ text field  - ‘Password’ text field  - ‘Login’ button.  - ‘Cancel’ button.  System show a message box: “Invalid Username and password”. |

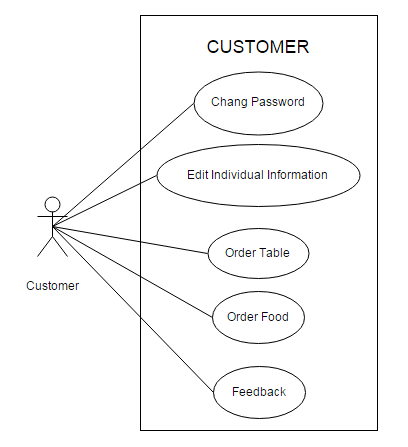


**3.2.2. Log out:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Log out | |
| **Actors** | Customer, Admin or Staff. | |
| **Description** | Customer, Admin or Staff wants to log out of the System. | |
| **Pre-conditions** | Actor has logged in. | |
| **Post-conditions** | Success: Actor Logged out. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Log out’ button on the Actor’s Home menu. | System Responses:  2. System redirect to Logout page, remove session, and then redirect to Login page. |

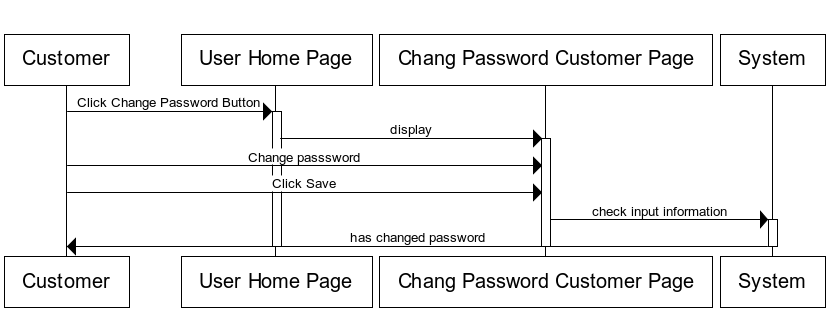


**3.3. Customer:**



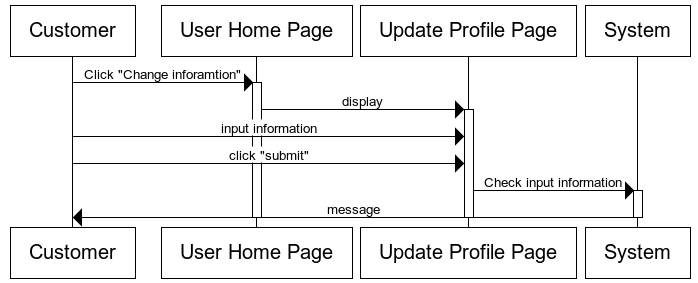
**3.3.1. Change Password:**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Change Password | |
| **Actors** | Customer | |
| **Description** | Change Password | |
| **Requirements** | Actor provides Username and Password | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: The password has changed. | |
| **Basic flow** | Actor actions:  1. From User Home page, actor clicks on to the ‘Change Password’ button.  3. Actor clicks on to ‘Save’ button to finish adding. | System response:  2. System redirects to ‘Change Password’ page and display form with the following controls:  ‘Old Password’ textfield.  ‘New Password’ textfield.  ‘Confirm New Password’ textfield.  ‘Save’ button.  4. System checks input information.  [Exception]  5. System adds new Administrator into the database. |
| **Exception** | [Exception]  1. Actor lefts ‘Old Password’ textfield.  2. Actor lefts ‘New Password’ textfield.  3. Actor lefts ‘Confirm New Password’ textfield.  4. ‘New Password’ textfield is not same as ‘Confirm New Password’ textfield. | System displays a message to notice actor:  “Old Password cannot be blank. Please input Old Password.”  “New Password cannot be blank. Please input New Password.”  “Confirm New Password cannot be blank. Please input Confirm Password.”  “New Password and Confirm New Password must be same.” |

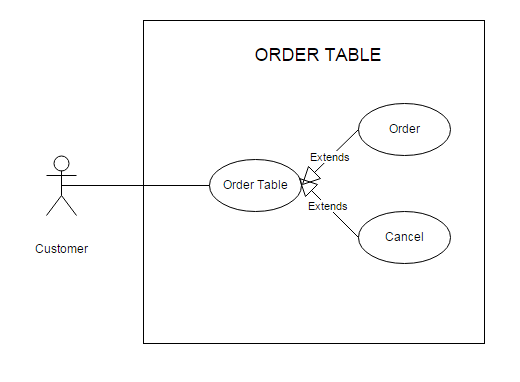
****

**3.3.2. Edit Individual Information:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Individual Information | |
| **Actors** | Customer | |
| **Description** | Customer wants to Edit Individual Information. | |
| **Requirements** | Customer provides Customer’s information | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: Change Individual Information success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Update Profile’ link in User Home page  3. Actor inputs individual information and click ‘Submit’ button. | System Responses:  2. System redirects to update Profile page and displays Update profile form with the following control :  - ‘UserName’ text field, read only.  - ‘Address’ text field  - ‘Phone’ text field  - ‘Email’ text field  - ‘Submit’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System edit individual information into database, show message: ‘Change Individual Information Success!’ and then redirects to User Homepage. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘Address’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3.Input phone not in numeric or phone length not in range 10 – 11 character.  4. Actor lefts ‘Email’ text field blank or input an invalid email address. | System Responses:  System displays message to Actor :  - ‘Address cannot be blank’.  - ‘Phone cannot be blank’.  - ‘The phone number should be numeric only and must be between 10 and 11 digits’.  - ‘Email is invalid’. |

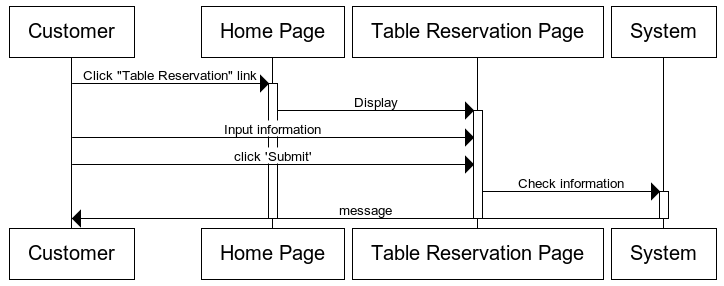
****

**3.3.3. Order Table:**

****

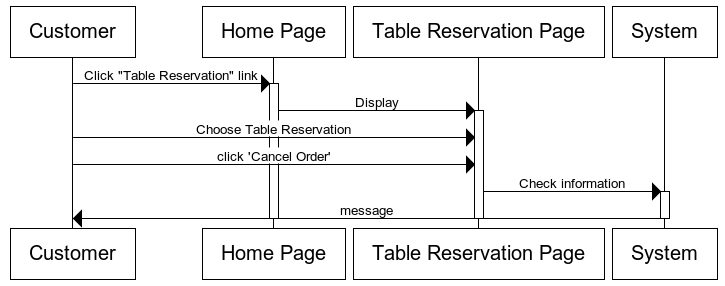
**3.3.3.1. Order:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Order Table | |
| **Actors** | Customer | |
| **Description** | Customer wants to Order Table. | |
| **Requirements** | Customer provides ordering table’s information | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: order table success  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘table reservation’ link in Home page  3. Actor inputs order information and click ‘Submit’ button. | System Responses:  2. System redirects to table reservationpage and displays table reservation form with the following control :  - ‘branch’ drop down list field  - ‘room’ drop down list field  - ‘table’ drop down list field  -‘banquet date’ text field  -‘banquet day’ drop down list field  - ‘Submit’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System ordering table information into database, show message: ‘order reservation Success!’ and then redirects to Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Input banquet date and banquet day near (order date time). | System Responses:  System displays message to Actor :  - ‘With dinner banquet, you have order before 11am of banquet date! With lunch banquet, you have order before 16pm the day before’.  - ‘Over table!’. |

****

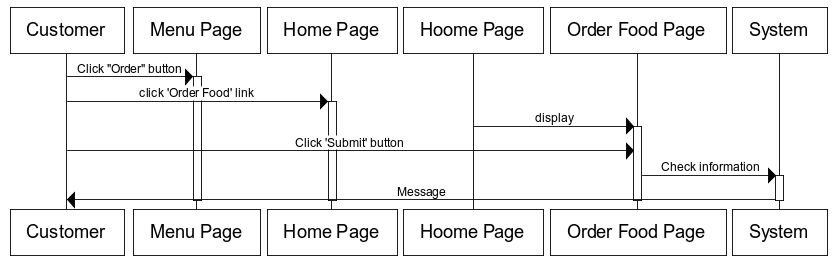
**3.3.3.2. Cancel Ordering Table:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Cancel Ordering Table | |
| **Actors** | Customer | |
| **Description** | Customer wants to Cancel Ordering Table. | |
| **Requirements** |  | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: Cancel ordering table success  Fail: Cancel ordering table faild. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘table reservation’ link in Home page  3. Actor choose ordering tableandclick ‘Cancel Order’ button. | System Responses:  2. System redirects to table reservation page and displays table reservationinformation form with the following control :  - ‘List Ordering table of customer.  - ‘Cancel Order’ button  4. System checks the information  [Exception]  5. System edit individual information into database, show message: ‘order reservation Success!’ and then redirects to Home page. |
| **Exception** | Actor Actions:  [Exception]  1.Actor choose Ordering table to cancel with conditional not valid. | System Responses:  System displays message to Actor :  - ‘With diner banquet, you have cancel before 11am of banquet date! With lunch banquet, you have cancel before 16pm the day before’.  - ‘you have confirmed this ordering table! PLZ… Contact with our staff!’. |



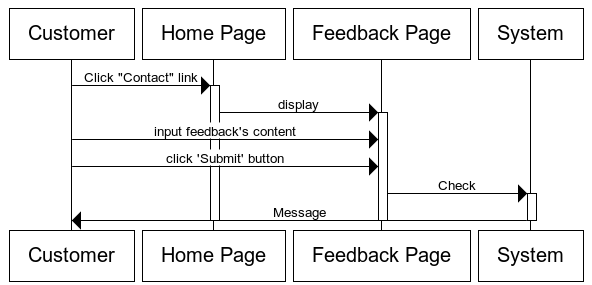
**3.3.4. Order Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Order Food | |
| **Actors** | Customer | |
| **Description** | Customer wants to Order Food. | |
| **Requirements** | Customer provides ordering food’s information | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: order food success  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Order’ button in Menu page.  2. Actor click ‘Order Food’ link in Home Page  4. Actor inputs order information and click ‘Submit’ button. | System Responses:  3. System redirects to Order Food page and displays Order Food form with the following control :  - List Of Food, which customer order.  - ‘Submit’ button  - ‘Cancel’ button  5. System checks the information  [Exception]  6. System ordering food information into database, show message: ‘order food Success!’ and then redirects to Home page. |
| **Exception** | Actor Actions:  [Exception]  1.Actor order food greater than 10 foods. | System Responses:  System displays message to Actor :  - ‘A Customer only order 10 foods for a day!’. |

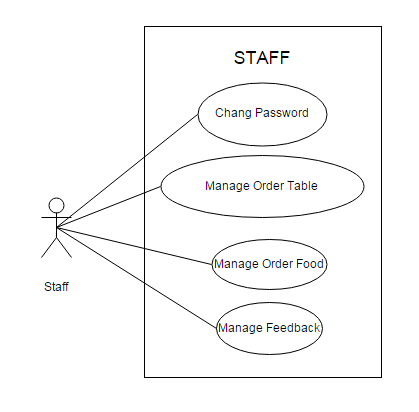


**3.3.5. Feedback:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Feedback | |
| **Actors** | Customer | |
| **Description** | Customer wants to Feedback. | |
| **Requirements** | Customer provides deedback’s content. | |
| **Pre-conditions** | Actor has logged in as Customer | |
| **Post-conditions** | Success: feedback success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘contact’ link in Home page  3. Actor inputs feedback’s content and click ‘Submit’ button. | System Responses:  2. System redirects to feedback page and displays feedback form with the following control :  -‘content’ text field  - ‘Submit’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System ordering table information into database, show message: ‘Send feedback Success!’ and then redirects to Home page. |
| **Exception** | Actor Actions:  [Exception]  1.Actor lefts ‘content’ text field blank. | System Responses:  System displays message to Actor :  - ‘content cann’t be blank! ’ |

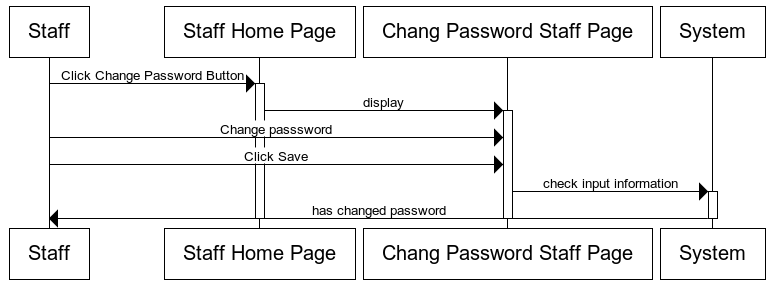


**3.4. Staff:**

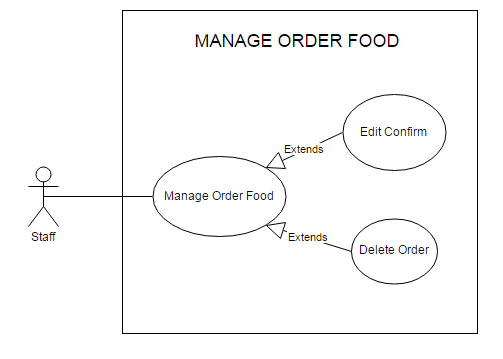


**3.4.1. Change Password:**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Change Password | |
| **Actors** | Staff | |
| **Description** | Change Password | |
| **Requirements** | Actor provides Username and Password | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: The password has changed. | |
| **Basic flow** | Actor actions:  1. From Staff Home page, actor clicks on to the ‘Change Password’ button.  3. Actor clicks on to ‘Save’ button to finish adding. | System response:  2. System redirects to ‘Change Password’ page and display form with the following controls:  ‘Old Password’ textfield.  ‘New Password’ textfield.  ‘Confirm New Password’ textfield.  ‘Save’ button.  4. System checks input information.  [Exception]  5. System adds new Administrator into the database. |
| **Exception** | [Exception]  1. Actor lefts ‘Old Password’ textfield.  2. Actor lefts ‘New Password’ textfield.  3. Actor lefts ‘Confirm New Password’ textfield.  4. ‘New Password’ textfield is not same as ‘Confirm New Password’ textfield. | System displays a message to notice actor:  “Old Password cannot be blank. Please input Old Password.”  “New Password cannot be blank. Please input New Password.”  “Confirm New Password cannot be blank. Please input Confirm Password.”  “New Password and Confirm New Password must be same.” |

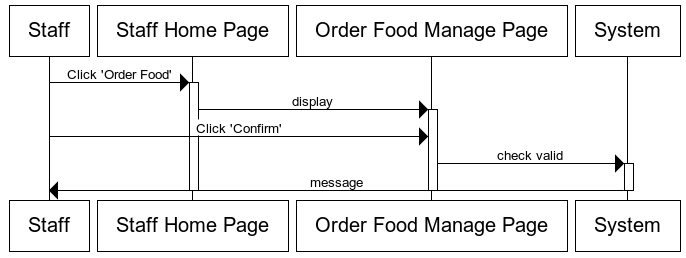


**3.4.2. Manage Order Food:**

****

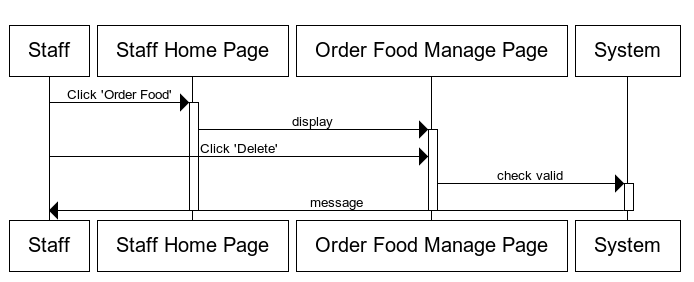
**3.4.2.1. Edit Confirm:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Confirm | |
| **Actors** | Staff | |
| **Description** | Staff wants to Edit Confirm. | |
| **Requirements** | Staffchoose Ordering food. | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: has confirm.  Fail: confirm fail. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Order Food’ link in StaffHome page  3. Actor click ‘Confirm’ button. | System Responses:  2. System redirects to Order Food Managementpage and displays list of ordering food and following control :  - ‘Confirm’ button  4. System checks the information  [Exception]  5. System confirm orderinto database, show message: ‘Order has just confirm!’ and then redirects to Staff Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor choose ordering food, which have banquet date before current date. | System Responses:  System displays message to Actor :  - ‘You cannot confirm this ordering food!’. |

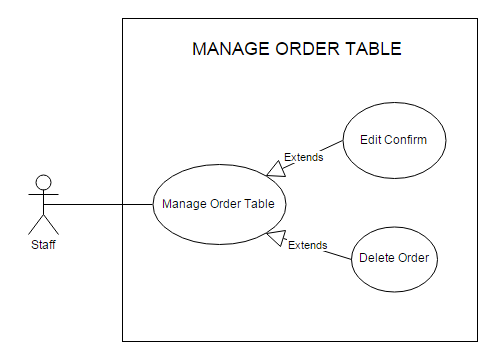
****

**3.4.2.2. Delete Ordering Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Delete Ordering Food | |
| **Actors** | Staff | |
| **Description** | Staff wants to Delete Ordering Food | |
| **Requirements** | Staff choose Ordering food. | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: delete success.  Fail: delete fail. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Order Food’ link in Staff Home page  3. Actor click ‘Delete’ button. | System Responses:  2. System redirects to Order Food Manage page and displays list of ordering food and following control :  - ‘Delete’ button  4. System checks the information  [Exception]  5. System confirm order into database, show message: ‘Order has just confirm!’ and then redirects to Staff Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor choose ordering food, which have banquet date before current date.  2.Actor choose ordering food, which has confirm. | System Responses:  System displays message to Actor :  - ‘You cannot delete this ordering food!’. |

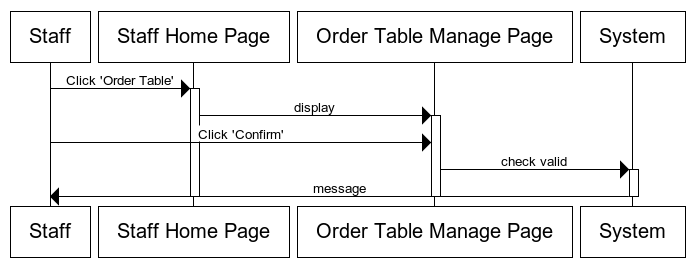


**3.4.3. Manage Order Table:**

****

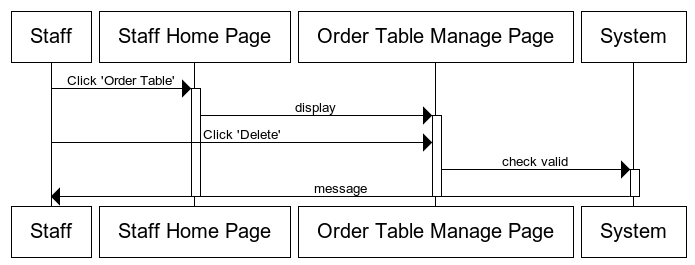
**3.4.3.1. Edit Confirm:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Confirm | |
| **Actors** | Staff | |
| **Description** | Staff wants to Edit Confirm. | |
| **Requirements** | Staff choose Ordering Table. | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: has confirm.  Fail: confirm fail. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Order Table’ link in Staff Home page  3. Actor click ‘Confirm’ button. | System Responses:  2. System redirects to Order Table Manage page and displays list of ordering Table and following control :  - ‘Confirm’ button  4. System checks the information  [Exception]  5. System confirm order into database, show message: ‘Order has just confirm!’ and then redirects to Staff Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor choose ordering table, which have banquet date before current date. | System Responses:  System displays message to Actor :  - ‘You cannot confirm this ordering food!’. |

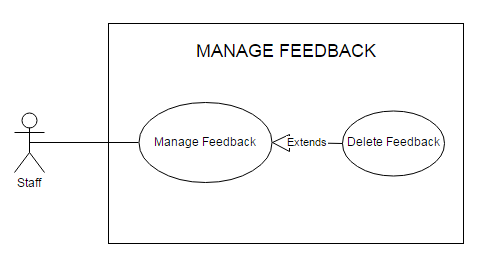
****

**3.4.3.2. Delete Ordering Table:**

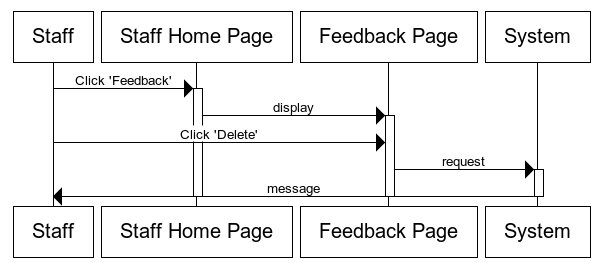
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Delete Ordering Table | |
| **Actors** | Staff | |
| **Description** | Staff wants to Delete Ordering Table | |
| **Requirements** | Staff choose Ordering table. | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: delete success.  Fail: delete fail. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Order Table’ link in Staff Home page  3. Actor click ‘Delete’ button. | System Responses:  2. System redirects to Order Table Manage page and displays list of ordering table and following control :  - ‘Delete’ button  4. System checks the information  [Exception]  5. System confirm order into database, show message: ‘Order has just confirm!’ and then redirects to Staff Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor choose ordering table, which have banquet date before current date.  2.Actor choose ordering table, which has confirm. | System Responses:  System displays message to Actor :  - ‘You cannot delete this ordering table!’. |

****

**3.4.4. Manage Feedback:**

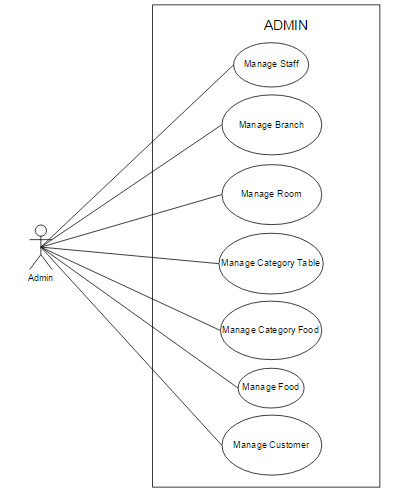
****

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Delete Feedback | |
| **Actors** | Staff | |
| **Description** | Staff wants to Delete Feedback | |
| **Requirements** | Staff choose Feedback. | |
| **Pre-conditions** | Actor has logged in as Staff | |
| **Post-conditions** | Success: delete success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Feedback’ link in Staff Home page  3. Actor click ‘Delete’ button. | System Responses:  2. System redirects to Feedback page and displays list of feedback and following control :  - ‘Delete’ button  4. System checks the information  5. System feedback into database, show message: ‘Delete feedback Success!’ and then redirects to Staff Home page. |

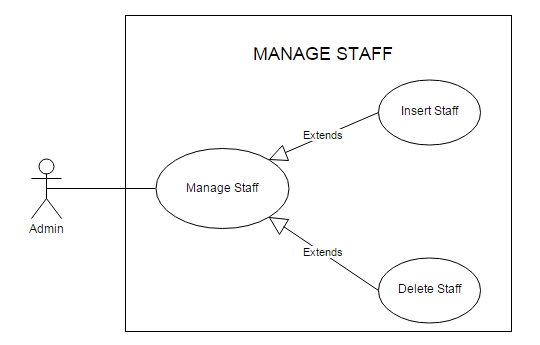
****

**3.5. Admin:**

(Note: Management of admin include: all management of staff and private management of admin).

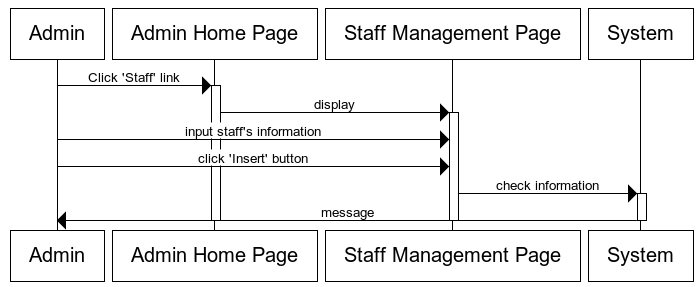


**3.5.1. Manage Staff:**

****

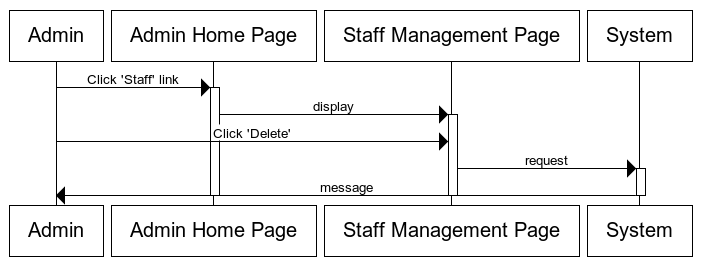
**3.5.1.1. Insert Staff:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Staff | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a staff. | |
| **Requirements** | Admin provides Staff’s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert Staff Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Staff’ link in Admin Home page  3. Actor inputs staff’s information and click ‘Insert’ button. | System Responses:  2. System redirects to Staff Management page and Insert Staff form with the following control :  - ‘UserName’ text field  - ‘Password’ text field  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts the Staff into database, show message: ‘Insert Staff Success!’ and then redirects to Admin Homepage. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘UserName’ text field blank.  2. UserName has already exists.  3. Actor lefts ‘Password’ text field blank. | System Responses:  System displays message to Actor :  - ‘UserName cannot be blank’.  - ‘This Username has already exist!’.  - ‘Password cannot be blank’’. |

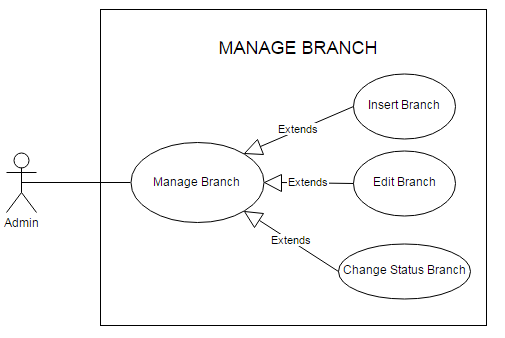
****

**3.5.1.2. Delete Staff:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Delete Staff | |
| **Actors** | Admin | |
| **Description** | Admin wants to Delete Staff | |
| **Requirements** | Admin choose Staff. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: delete success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Staff’ link in Admin Home page  3. Actor click ‘Delete’ button. | System Responses:  2. System redirects to Staff Management page and displays list of Staff and following control :  - ‘Delete’ button  4. System checks the information  5. System save into database, show message: ‘Delete Success!’ and then redirects to Admin Home page. |

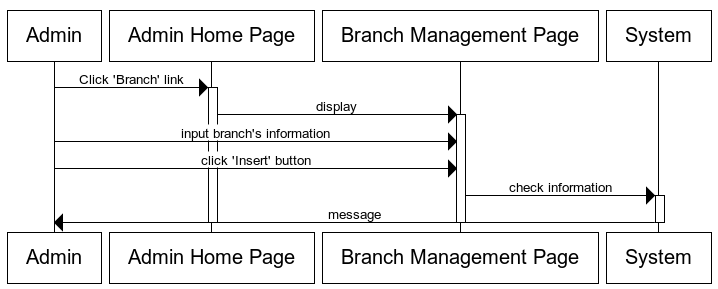
****

**3.5.2. Manage Branch:**

****

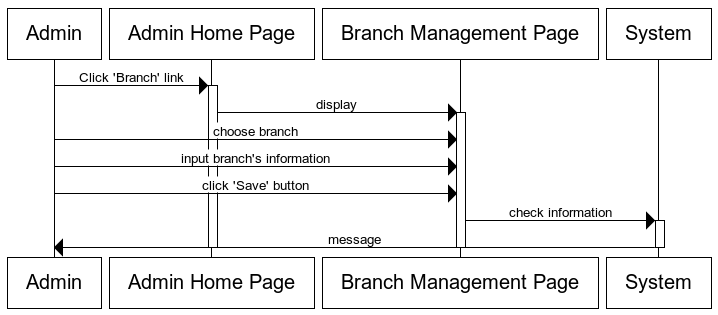
**3.5.2.1. Insert Branch:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Branch | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a branch. | |
| **Requirements** | Admin provides branch’s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert branch Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘branch’ link in Admin Home page  3. Actor inputs branch’s information and click ‘Insert’ button. | System Responses:  2. System redirects to Branch Management page and Insert Branchform with the following control :  - ‘BranchID’ text field  - ‘Name’ text field  - ‘Address’ text field  - ‘Phone’ text field  - Display List Of Rooms and quantity each room.  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts the Branch into database, show message: ‘Insert Branch Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘BranchID’ text field blank.  2. BranchID has already exists.  3. Actor lefts ‘name’ text field blank.  4. Name has already exists.  5.Name is not format ‘roomxx’  6. Actor lefts ‘Address’ text field blank.  7. Actor lefts ‘Phone’ text field blank.  8.Input phone not in numeric or phone length not in range 10 – 11 character. | System Responses:  System displays message to Actor :  - ‘BranchID cannot be blank’.  - ‘This BranchID has already exist!’.  - ‘name cannot be blank’’.  - ‘This name has already exist!’.  -‘Name must format branchxx with xx is numeric.’  - ‘Address cannot be blank’.’  - ‘Phone cannot be blank’.  - ‘The phone number should be numeric only and must be between 10 and 11 digits’. |

****

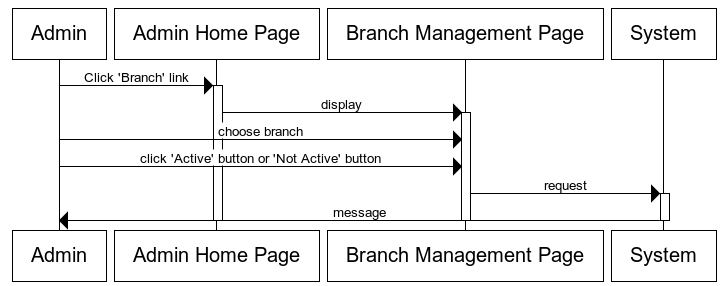
**3.5.2.2. Edit Branch:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Branch | |
| **Actors** | Admin | |
| **Description** | Admin wants to Edit Branch. | |
| **Requirements** | Admin provides Branch’s information | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: Edit Branch success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Branch’ link in AdminHome page  3. Actor inputs individual information and click ‘Save’ button. | System Responses:  2. System redirects to Branch Management Page and displays edit branch form with the following control :  - ‘BranchID’ text field, read only.  - ‘Name’ text field,read only.  - ‘Address’ text field  - ‘Phone’ text field  - Display List Of Rooms and quantity each room.  - ‘Save’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System save information into database, show message: ‘Edit Branch Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘Address’ text field blank.  2. Actor lefts ‘Phone’ text field blank.  3.Input phone not in numeric or phone length not in range 10 – 11 character. | System Responses:  System displays message to Actor :  - ‘Address cannot be blank’.’  - ‘Phone cannot be blank’.  - ‘The phone number should be numeric only and must be between 10 and 11 digits’. |

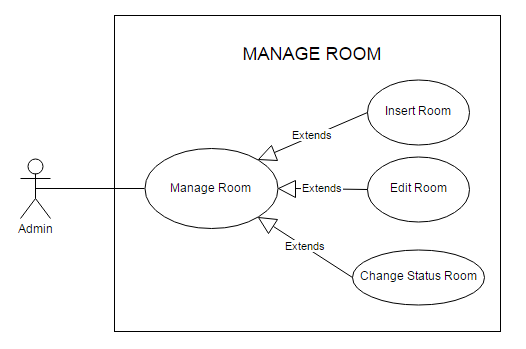
****

**3.5.2.3. Change Status Branch:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change Status Branch | |
| **Actors** | Admin | |
| **Description** | Admin wants to change status of Branch. | |
| **Requirements** | Admin choose branch. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: change status branch success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Branch’ link in Admin Home page  3. Actor click ‘Active’ button or ‘Not Active’ button. | System Responses:  2. System redirects to Branch Management Page and change status branch form with the following control :  - if branch is active, ‘not active’ button display.  - if branch is not active, ‘active’ button display.  4. System checks the information  5. System save information into database, show message: ‘Change Status Success!’ and then redirects to Admin Home page. |

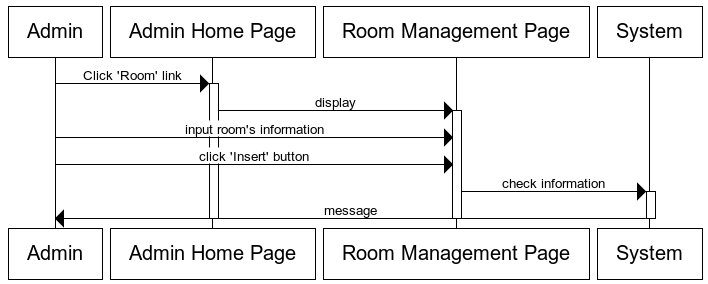
****

**3.5.3. Manage Room:**

****

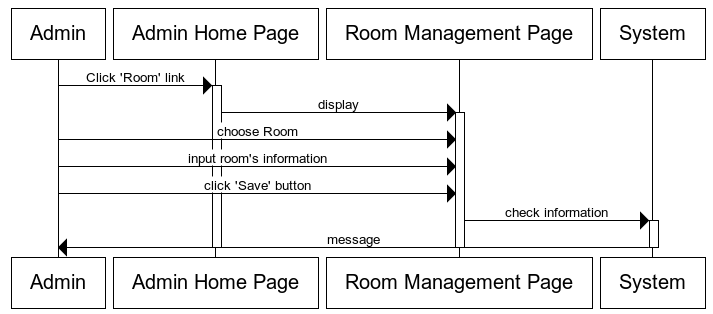
**3.5.3.1. Insert Room:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Room | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a room. | |
| **Requirements** | Admin provides room’s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert room Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Room’ link in Admin Home page  3. Actor inputs room’s information and click ‘Insert’ button. | System Responses:  2. System redirects to Room Management page and Insert Roomform with the following control :  - ‘RoomID’ text field  - ‘Name’ text field  - Display List Of Category table and quantity each category table.  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts Room into database, show message: ‘Insert Room Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘BranchID’ text field blank.  2. BranchID has already exists.  3. Actor lefts ‘name’ text field blank.  4. Name has already exists.  5.Name is not format ‘roomxx’ | System Responses:  System displays message to Actor :  - ‘BranchID cannot be blank’.  - ‘This BranchID has already exist!’.  - ‘name cannot be blank’’.  - ‘This name has already exist!’.  -‘Name must format roomxx with xx is numeric.’ |

****

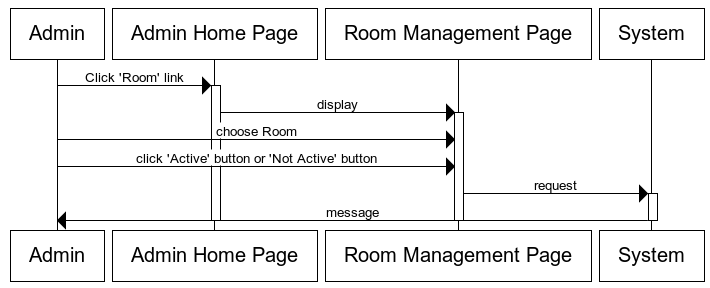
**3.5.3.2. Edit Room:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Room | |
| **Actors** | Admin | |
| **Description** | Admin wants to Edit Room. | |
| **Requirements** | Admin provides Room’s information | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: Edit Room success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Room’ link in Admin Home page  3. Actor inputs Rooml information and click ‘Save’ button. | System Responses:  2. System redirects to Room Management Page and displays edit room form with the following control :  - ‘roomID’ text field, read only.  - ‘name’ text field,read only.  - Display List Of Category table and quantity each category table.  - ‘Save’ button  - ‘Cancel’ button  4. System checks the information  5. System save information into database, show message: ‘Edit Room Success!’ and then redirects to Admin Home page. |

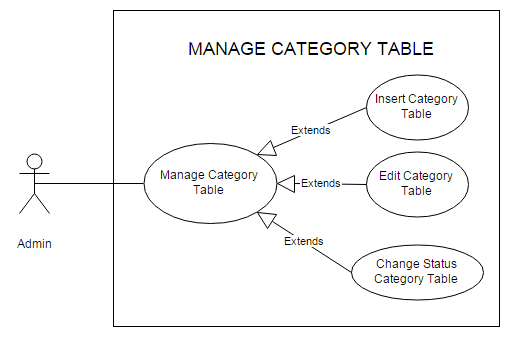
****

**3.5.3.3. Change Status Room:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change Status Room | |
| **Actors** | Admin | |
| **Description** | Admin wants to change status of Room. | |
| **Requirements** | Admin choose room. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: change status room success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Room’ link in Admin Home page  3. Actor click ‘Active’ button or ‘Not Active’ button. | System Responses:  2. System redirects to Room Management Page and change status roomform with the following control :  - if room is active, ‘not active’ button display.  - if room is not active, ‘active’ button display.  4. System checks the information  5. System save information into database, show message: ‘Change Status Success!’ and then redirects to Admin Home page. |

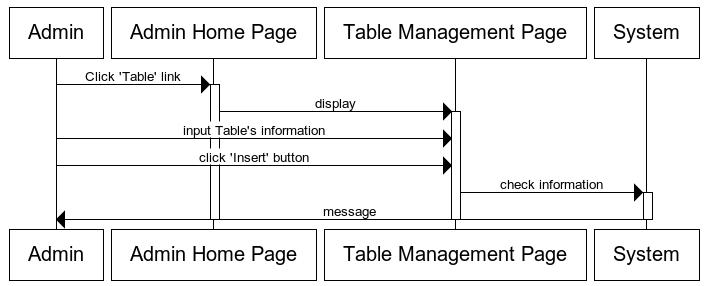
****

**3.5.4. Manage Category Table:**

****

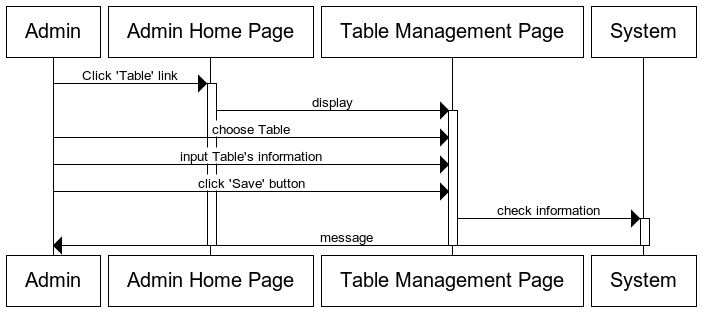
**3.5.4.1. Insert Category Table:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Category Table | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a category Table. | |
| **Requirements** | Admin provides category table‘s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert category table Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Table’ link in Admin Home page  3. Actor inputs table’s information and click ‘Insert’ button. | System Responses:  2. System redirects to Table Management page and Insert Tableform with the following control :  - ‘TableID’ text field  - ‘Name’ text field  - ‘Quantity Chairs’ text field  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts Table into database, show message: ‘Insert Table Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘TableID’ text field blank.  2. TableIDhas already exists.  3. Actor lefts ‘name’ text field blank.  4. Name has already exists.  5.Name is not format ‘roomxx’  6.Input quantity Chairs not in numerics | System Responses:  System displays message to Actor :  - ‘TableID cannot be blank’.  - ‘This TableID has already exist!’.  - ‘name cannot be blank’’.  - ‘This name has already exist!’.  -‘Name must format tablexx with xx is numeric.’  -‘quantity chairs’ must numeric. |

****

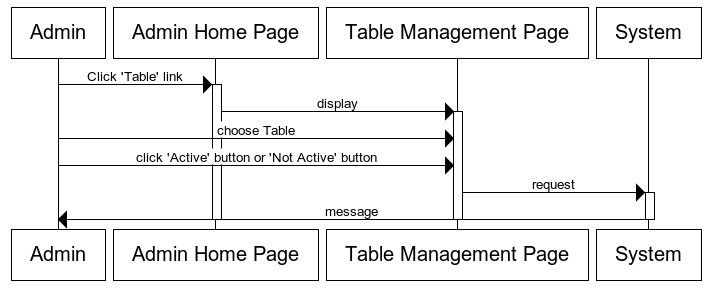
**3.5.4.2. Edit Room:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Table | |
| **Actors** | Admin | |
| **Description** | Admin wants to Edit Table. | |
| **Requirements** | Admin provides Table’s information | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: Edit Table success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Table’ link in Admin Home page  3. Actor inputs Table information and click ‘Save’ button. | System Responses:  2. System redirects to Table Management Page and displays edit Table form with the following control :  - ‘Table ID’ text field, read only.  - ‘name’ text field,read only.  -‘quantity chairs’ text.  - ‘Save’ button  - ‘Cancel’ button  4. System checks the information  5. System save information into database, show message: ‘Edit Table Success!’ and then redirects to Admin Home page. |

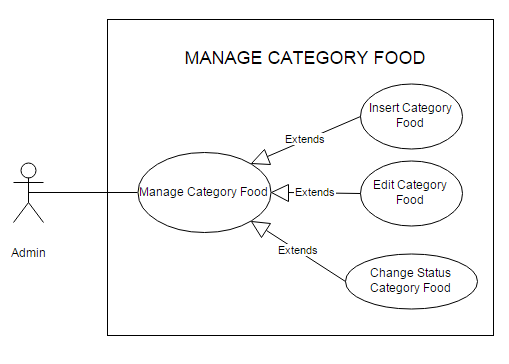
****

**3.5.4.3. Change Status Table:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change Status table | |
| **Actors** | Admin | |
| **Description** | Admin wants to change status of table. | |
| **Requirements** | Admin choose table. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: change status table success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Table’ link in Admin Home page  3. Actor click ‘Active’ button or ‘Not Active’ button. | System Responses:  2. System redirects to Table Management Page and change status Tableform with the following control :  - if Table is active, ‘not active’ button display.  - if Table is not active, ‘active’ button display.  4. System checks the information  5. System save information into database, show message: ‘Change Status Success!’ and then redirects to Admin Home page. |

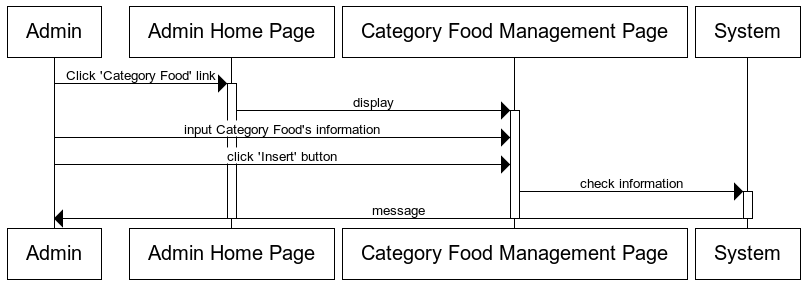
****

**3.5.5. Manage Category Food:**

****

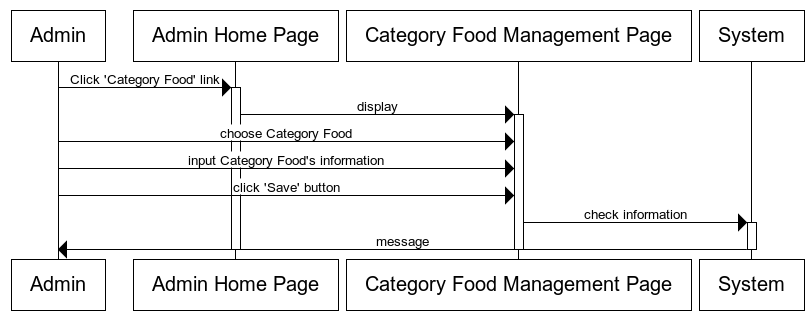
**3.5.5.1. Insert Category Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Category Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a category Food. | |
| **Requirements** | Admin provides category Food‘s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert category Food Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Food’ link in Admin Home page  3. Actor inputs category Food’s information and click ‘Insert’ button. | System Responses:  2. System redirects to category FoodManagement page and Insert category Foodform with the following control :  - ‘categoryFoodID’ text field  - ‘Name’ text field  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts category Food into database, show message: ‘Insert category Food Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  3. Actor lefts ‘name’ text field blank.  4. Name has already exists. | System Responses:  System displays message to Actor :  - ‘name cannot be blank’’.  - ‘This name has already exist!’. |

****

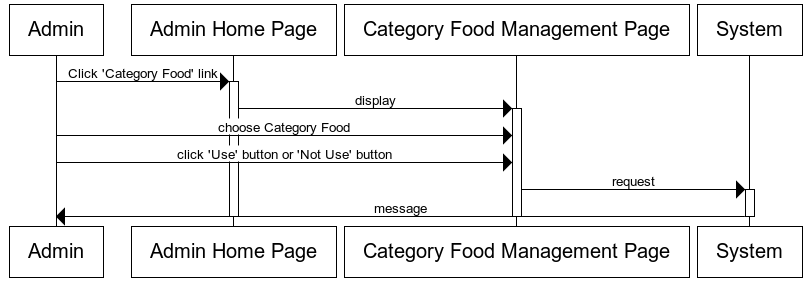
**3.5.5.2. Edit Category Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Category Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to Edit Category Food. | |
| **Requirements** | Admin provides Category Food’s information | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: Edit Category Food success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Category Food’ link in Admin Home page  3. Actor inputs Category Food information and click ‘Save’ button. | System Responses:  2. System redirects to Category Food Management Page and displays edit Category Food form with the following control :  - ‘Category Food ID’ text field, read only.  - ‘name’ text field.  - ‘Save’ button  - ‘Cancel’ button  4. System checks the information  5. System save information into database, show message: ‘Edit Category Food Success!’ and then redirects to Admin Home page. |

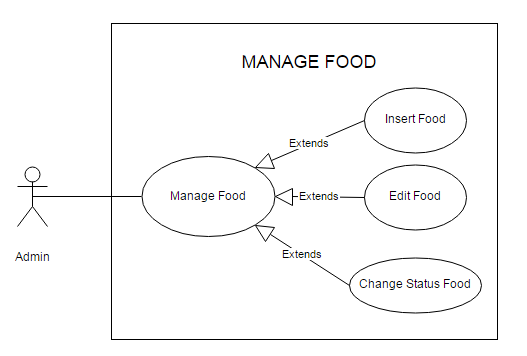
****

**3.5.5.3. Change Status Category Table:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change Status Category Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to change status of Category Food. | |
| **Requirements** | Admin choose Category Food. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: change status Category Food success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Category Food’ link in Admin Home page  3. Actor click ‘Use’ button or ‘Not Use’ button. | System Responses:  2. System redirects to Category Food Management Page and change status Category Foodform with the following control :  - if Category Food is use, ‘Not Use’ button display.  - if Category Food is not use, ‘Use’ button display.  4. System checks the information  5. System save information into database, show message: ‘Change Status Success!’ and then redirects to Admin Home page. |

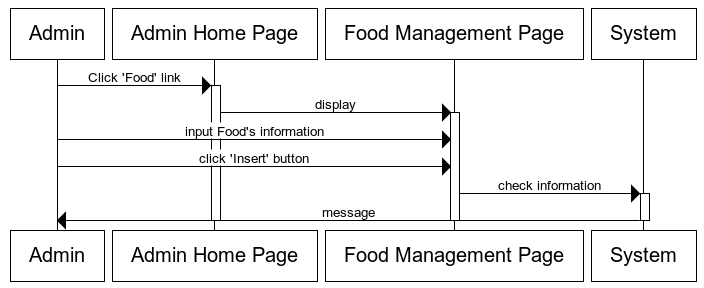
****

**3.5.6. Manage Food:**

****

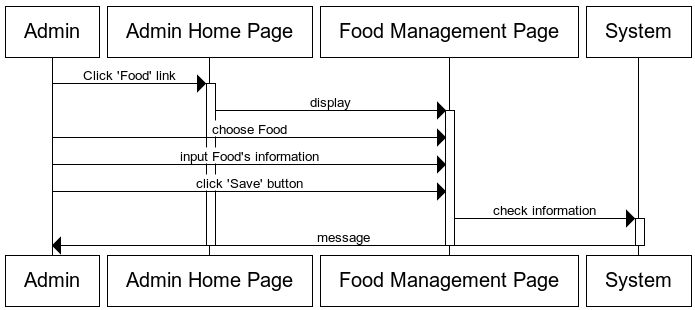
**3.5.6.1. Insert Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Insert Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to insert a Food. | |
| **Requirements** | Admin provides Food ‘s information | |
| **Pre-conditions** | Actor has logged in as Admin. | |
| **Post-conditions** | Success: Insert Food Success.  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Food’ link in Admin Home page  3. Actor inputs Food’s information and click ‘Insert’ button. | System Responses:  2. System redirects to Food Management page and Insert Foodform with the following control :  - ‘categoryFood ID’ dropdownlist  - ‘Name’ text field  - ‘Image’ text field  - ‘Description’ text field  - ‘Price’ text field  - ‘Insert’ button  - ‘Cancel’ button  4. System checks the information  [Exception]  5. System inserts Food into database, show message: ‘Insert Food Success!’ and then redirects to Admin Home page. |
| **Exception** | Actor Actions:  [Exception]  1. Actor lefts ‘name’ text field blank.  2. Name has already exists.  3. Actor lefts ‘image’ text field blank.  4. Actor lefts ‘price’ text field blank.  5.Input price not in numeric | System Responses:  System displays message to Actor :  - ‘name cannot be blank’’.  - ‘This name has already exist!’.  - ‘image cannot be blank’’.  - ‘price cannot be blank’’.  -‘price should be numeric’ |

****

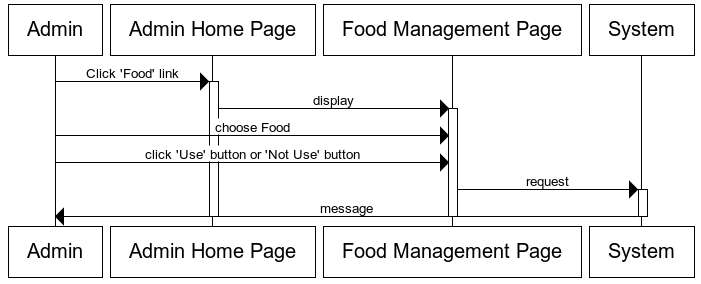
**3.5.6.2. Edit Category Food:**

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Edit Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to Edit Food. | |
| **Requirements** | Admin provides Food’s information | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: Edit Food success.  Fail: Refill information. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Food’ link in Admin Home page  3. Actor inputs Category Food information and click ‘Save’ button. | System Responses:  2. System redirects to Food Management Page and displays edit Food form with the following control :  - ‘Food ID’ text field, read only.  -‘category food id’ dropdownlist  - ‘name’ text field.  - ‘image’ text field.  - ‘description’ text field.  - ‘price’ text field.  - ‘Save’ button  - ‘Cancel’ button  4. System checks the information  5. System save information into database, show message: ‘Edit Food Success!’ and then redirects to Admin Home page. |

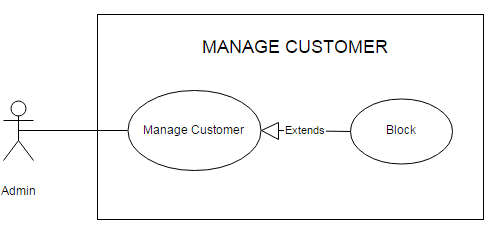
****

**3.5.6.3. Change Status Food:**

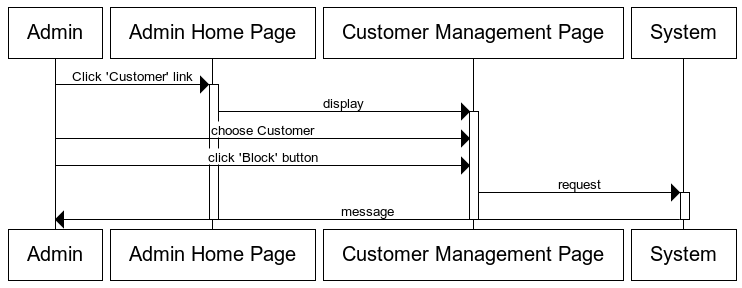
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Change Status Food | |
| **Actors** | Admin | |
| **Description** | Admin wants to change status of Food. | |
| **Requirements** | Admin choose Food. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: change status Food success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Food’ link in Admin Home page  3. Actor click ‘Use’ button or ‘Not Use’ button. | System Responses:  2. System redirects to Food Management Page and change status Food form with the following control :  - if Food is use, ‘Not Use’ button display.  - if Food is not use, ‘Use’ button display.  4. System checks the information  5. System save information into database, show message: ‘Change Status Success!’ and then redirects to Admin Home page. |

****

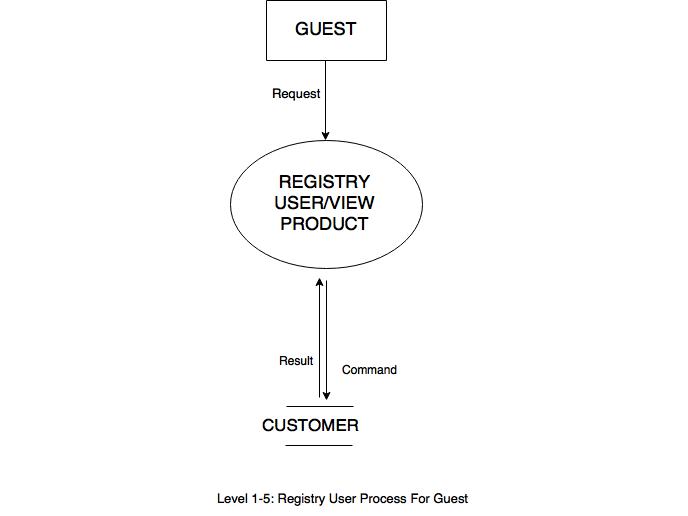
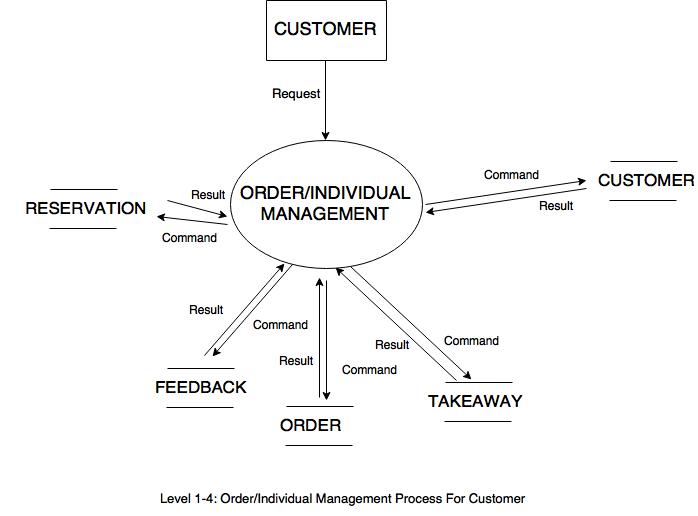
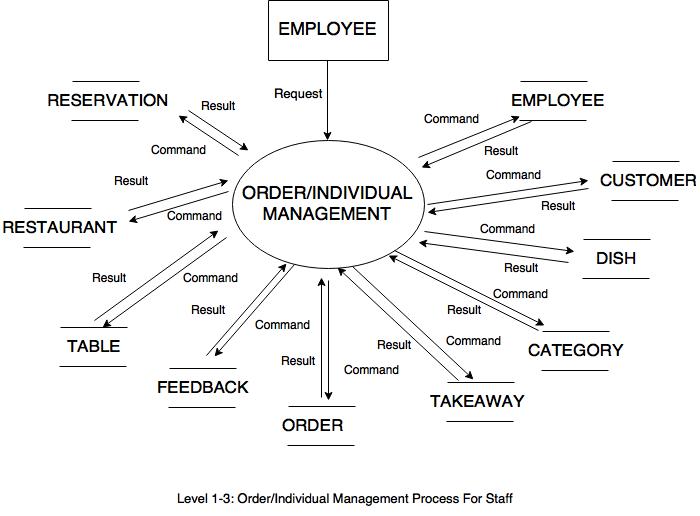
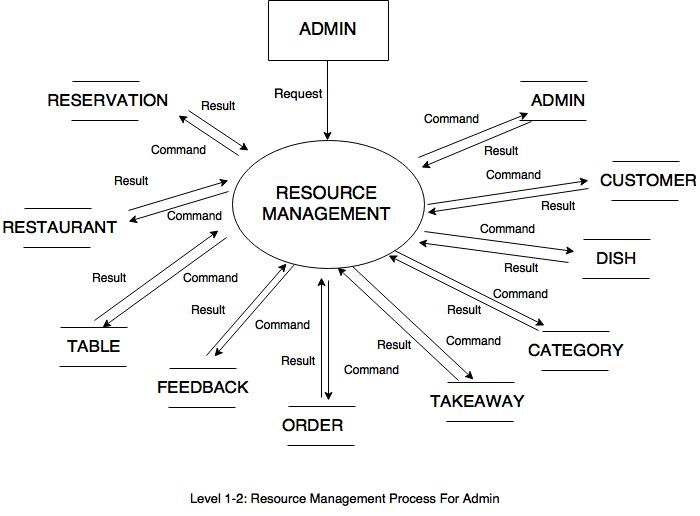
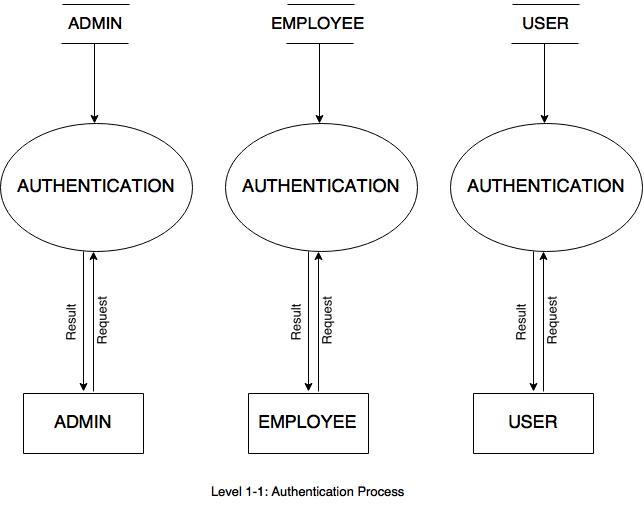
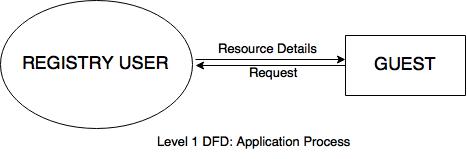
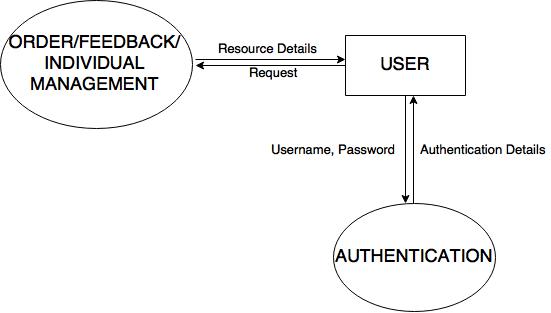
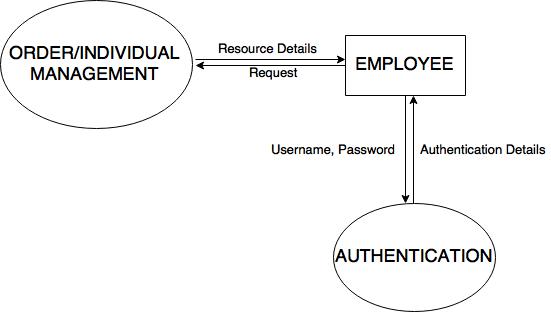
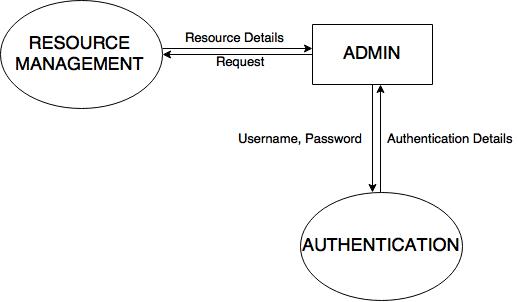
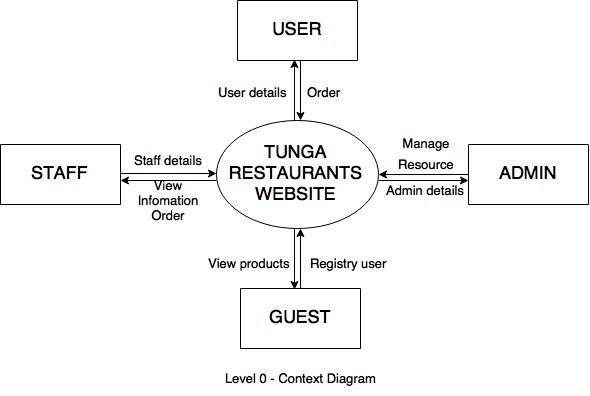
**3.5.7. Manage Customer:**

****

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | Block Customer | |
| **Actors** | Admin | |
| **Description** | Admin wants to Block Customer. | |
| **Requirements** | Admin choose Customer. | |
| **Pre-conditions** | Actor has logged in as Admin | |
| **Post-conditions** | Success: block customer success. | |
| **Basic flow** | Actor Actions:  1. Actor clicks ‘Customer’ link in Admin Home page  3. Actor click ‘Block’ button | System Responses:  2. System redirects to Customer Management Page and block customer form with the following control :  -‘Block’ button.  4. System checks the information  5. System save information into database, show message: ‘Block Customer Success!’ and then redirects to Admin Home page. |

****

**4. Data Flow Diagram**



**5. Process Diagram**

5.1. Login:

5.1.1. Staff Login:

Input username and password

Check in Staff

Check Role

Admin

Employee

Resource Management

View Order / Reservation

Individual Management

Not Exist

Exist

Role=0

Role=1

Start

End

Login Failed

5.1.2. Customer Login:

Exist

Login Failed

Check Status

Deny Login

Status=True

Status=True

Check in Customer

Customer

Order/Reservation/Feedback

Management

Not Exist

Start

Input username and password

End

5.2. Register new Customer:

Start

Input Individual Information

Validation

Save in database

Register Failed

Register Successfully

End

False

True

5.3. View / Edit Customer’s information (for Customer)

5.3.1. View Customer’s information

Start

End

Display Individual Information

5.3.2. Edit Customer’s information

Start

Input data

Save in database

Edit Successfully

End

Check Valid data

Edit Failed

True

False

5.3.3. Change Customer’s password

Start

Input old and new password

Save in database

Edit Successfully

End

Check old password

Old password incorrect

True

False

5.3.4. Reserve Table

Start

Input Reservation Information

Check Valid data

Save in database

Reserve Failed

Reserve Successfully

End

False

True

5.3.5. View Reservation

Start

End

Display Reservation Information

5.3.6. Cancel Reservation

Start

Save in database

Cancel Successfully

End

Choose Reservation

Confirm cancel

Cancel Failed

False

True

Check Conditional

Cancel Failed

True

False

5.3.7. Feedback

Start

Input

Feedback

Check Valid data

Save in database

Feedback Failed

Order Successfully

End

False

True

5.3.8. Order Takeaway

Start

Input Order Information

Check Valid data

Save in database

Order Failed

Order Successfully

End

False

True

5.3.9. View Order

Start

End

Display Takeaway Order

5.4. Takeaway Order / Reservation Management (for Employee)

5.4.1. Change password

Start

Input old and new password

Save in database

Edit Successfully

End

Check old password

Old password incorrect

True

False

5.4.2. View Reservation

Start

End

Display Reservation Information

5.4.3. Cancel Reservation

Start

Save in database

Cancel Successfully

End

Choose Reservation

Confirm cancel

Cancel Failed

False

True

Check Condition

Cancel Failed

True

False

5.4.4. View Feedback

Start

End

Display

Feedback

5.4.5. View Takeaway Order

Start

End

Display

Takeaway Order

5.4.6. Cancel Takeaway Order

Start

Save in database

Cancel Successfully

End

Choose Takeaway

Confirm cancel

Cancel Failed

False

True

Check Condition

Cancel Failed

True

False

5.5. Resource Management (for Admin)

5.5.1. Change password

Start

Input old and new password

Save in database

Edit Successfully

End

Check old password

Old password incorrect

True

False

5.5.2. View staff

Start

End

Display Staff

5.5.3. Add staff

Start

Input New Staff

Check Valid data

Save in database

Add Failed

Add Successfully

End

False

True

5.5.4. Disable Staff

Start

Save in database

Disable Successfully

End

Choose Staff

Confirm disable

Delete Failed

False

True

5.5.5. View Dish

Start

End

Display Dish

5.5.6. Add Dish

Start

Input Dish Information

Check Valid data

Save in database

Add Failed

Add Successfully

End

False

True

5.5.7. Edit Dish

Start

Input data

Save in database

Edit Successfully

End

Choose Dish

Check Valid data

Edit Failed

True

False

5.5.8. View Restaurant

Start

End

Display Restaurant

5.5.9. Add Restaurant

Start

Input Restaurant

Check Valid data

Save in database

Add Failed

Add Successfully

End

False

True

5.5.10. Edit Restaurant

Start

Input data

Save in database

Edit Successfully

End

Choose Restaurant

Check Valid data

Edit Failed

True

False

5.5.11. View Category

Start

End

Display Category

5.5.12. Add Category

Start

Input Category

Check Valid data

Save in database

Add Failed

Add Successfully

End

False

True

5.5.13. Edit Category

Start

Input data

Save in database

Edit Successfully

End

Choose Category

Check Valid data

Edit Failed

True

False

5.5.14. View Table

Start

End

Display Table

5.5.15. Add Table

Start

Input Table

Check Valid data

Save in database

Add Failed

Add Successfully

End

False

True

5.5.16. Edit Table



Start

Input data

Save in database

Edit Successfully

End

Choose Table

Check Valid data

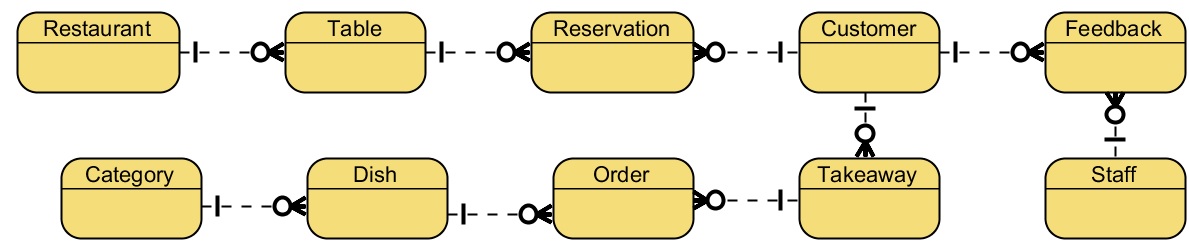
Edit Failed

True

False

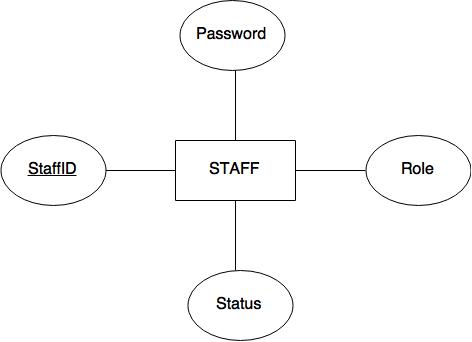
**6. Entity Relationship Diagram**

**6.1. Logical model**

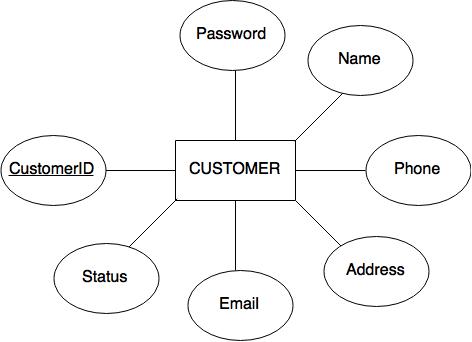
****

**6.2. Entity**

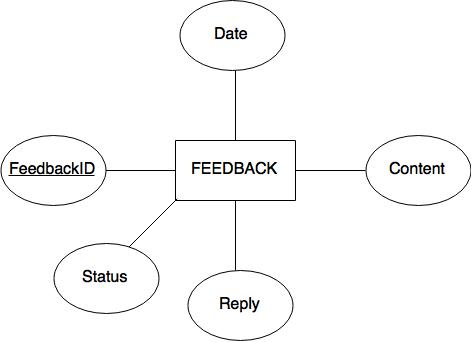
**6.2.1. Staff Table**

****

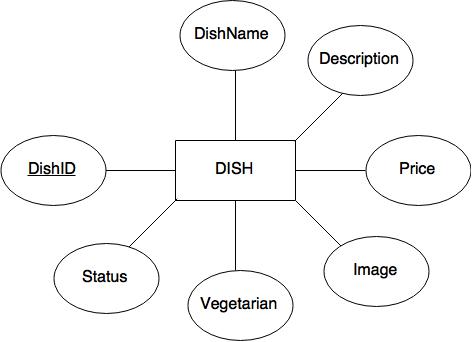
**6.2.2. Customer Table**

****

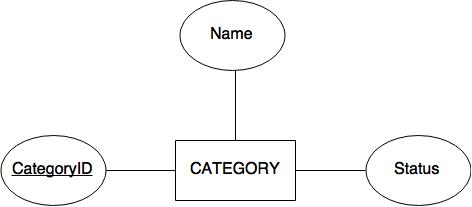
**6.2.3. Feedback Table**

****

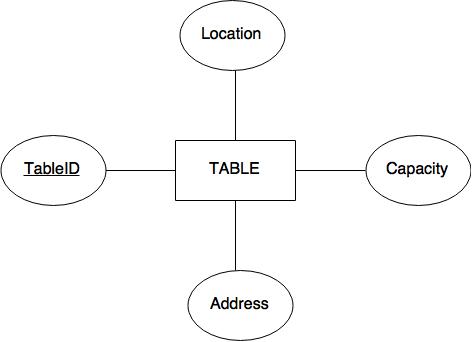
**6.2.4. Dish Table**

****

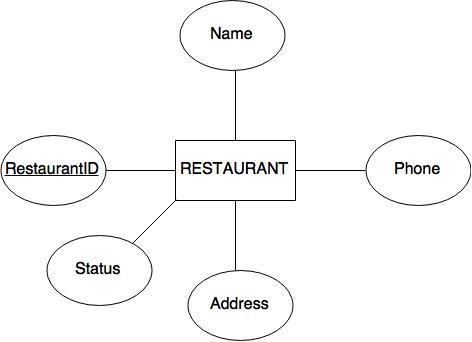
**6.2.5. Category Table**

****

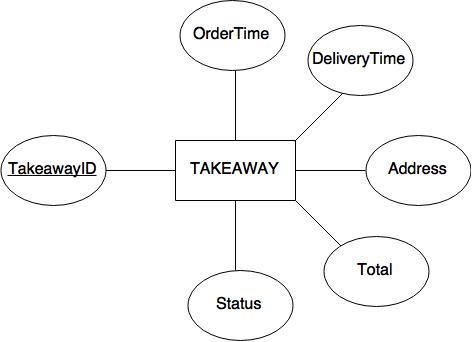
**6.2.6. [Table] Table**

****

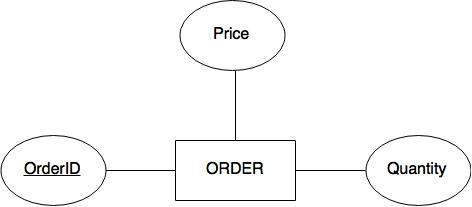
**6.2.7. Restaurant Table**

****

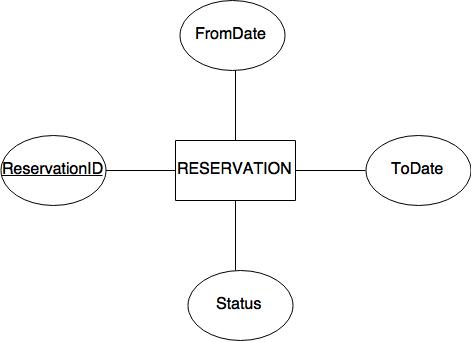
**6.2.8. Takeaway table**

****

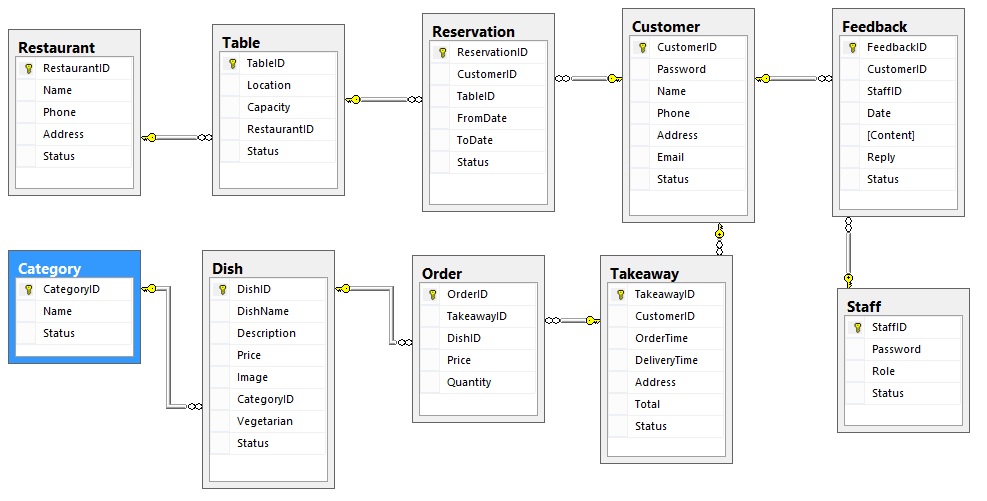
**6.2.9. Order Table**

****

**6.2.10. Reservation Table**

****

**6.3. E-R Diagram**

****

**7. Database Design / Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Staff** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| StaffID | nvarchar(20) | No | PK | Staff ID |
| Password | nvarchar(30) | No |  | Staff Password |
| Role | bit | No |  | Staff role  True: Admin  False: Employee |
| Status | bit | No |  | Account Status  True: Active  False: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Customer** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| CustomerID | nvarchar(30) | No | PK | Customer’s ID |
| Password | nvarchar(30) | No |  | Customer’s Password |
| Name | nvarchar(40) | No |  | Customer’s Name |
| Phone | nvarchar(15) | No |  | Customer’s Phone |
| Address | nvarchar(200) | No |  | Customer’s Address |
| Email | nvarchar(50) | No |  | Customer’s Email |
| Status | bit | No |  | Account status  True: Active  False: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Feedback** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| FeedbackID | nvarchar(10) | No | PK | Feedback’s ID |
| CustomerID | nvarchar(30) | No | FK  (Customer) | Customer’s ID |
| StaffID | nvarchar(20) | Yes | FK  (Admin) | Staff ID |
| Date | datetime | No |  | Date of feedback |
| Content | nvarchar(MAX) | No |  | Feedback content |
| Reply | nvarchar(MAX) | Yes |  | Reply content |
| Status | tinyint | No |  | Feedback status  1: Pending  2: Active  3: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Dish** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| DishID | nvarchar(10) | No | PK | Dish ID |
| DishName | nvarchar(50) | No |  | Dish name |
| Description | nvarchar(MAX) | No |  | Dish description |
| Price | float | No |  | Dish price |
| Image | nvarchar(50) | No |  | Dish image (url) |
| CategoryID | nvarchar(10) | No | FK  (Category) | Category ID |
| Vegetarian | bit | No |  | True: Non-Vegetarian dish  False: Vegetarian dish |
| Status | bit | No |  | Status of food  True: Active  False: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Category** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| CategoryID | nvarchar(10) | No | PK | Category ID |
| Name | nvarchar(30) | No |  | Category name |
| Status | bit | No |  | Status of category  True: Active  False: Block |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Takeaway** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| TakeawayID | nvarchar(10) | No | PK | TakeawayID |
| CustomerID | nvarchar(30) | No | FK  (Customer) | Customer’s ID |
| OrderTime | datetime | No |  | Time of order |
| DeliveryTime | datetime | No |  | Time of delivery |
| Address | nvarchar(200) | No |  | Delivery address |
| Total | float | No |  | Bill total |
| Status | tinyint | No |  | Takeaway Status  1: Pending  2: Active  3: Block |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Order** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| OrderID | nvarchar(10) | No | PK | Order ID |
| TakeawayID | nvarchar(10) | No | FK  (OrderFood) | Takeaway ID |
| DishID | nvarchar(10) | No | FK  (Food) | Dish ID |
| Price | float | No |  | Dish price |
| Quantity | int | No |  | Dish quantity |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Restaurant** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| RestaurantID | nvarchar(10) | No | PK | Restaurant ID |
| Name | nvarchar(50) | No |  | Restaurant name |
| Phone | nvarchar(15) | No |  | Restaurant phone |
| Address | nvarchar(200) | No |  | Restaurant address |
| Status | bit | No |  | Restaurant status  True: Active  False: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Table** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| TableID | nvarchar(10) | No | PK | Table ID |
| Location | nvarchar(50) | No |  | Table location |
| Capacity | tinyint | No |  | Maximum customer per table |
| RestaurantID | nvarchar(10) | No | FK  (Restaurant) | Restaurant ID |
| Status | bit | No |  | Status of Table  True: Active  False: Inactive |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Reservation** | | | | |
| **Field Name** | **Data Type** | **Allow Null** | **Key** | **Description** |
| ReservationID | nvarchar(10) | No | PK | Reservation ID |
| CustomerID | nvarchar(30) | No | FK  (Customer) | Customer ID |
| TableID | nvarchar(10) | No | FK  (Table) | Table ID |
| From | datetime | No |  | Reservation start |
| To | datetime | No |  | Reservation end |
| Status | tinyint | No |  | Status of Reservation  1: Pending  2: Active  3: Inactive |

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
| **Signature** | Team leader |  |
| **Date** | 13-August-2015 |  |