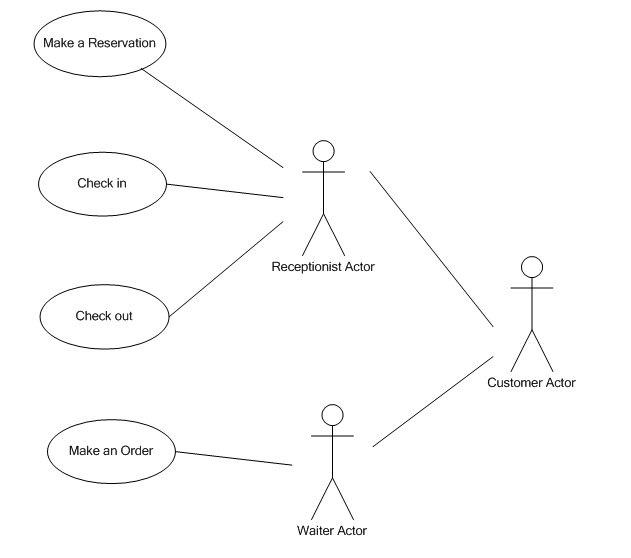
**Restaurant Management System**

**Part I**

**Rui Da, Yi Yang, Kun Chen**

**Use Case Diagram**



**Use Case**

**Name:** Make an order

**Actors:** Waiter

**Goals:** To make an order

**Preconditions:** Guests must have checked in successfully

**Summary:** Price will prompt up and total will be computed after each guest has finished an order

**Steps:**

|  |  |
| --- | --- |
| *Actor actions* | *System responses* |
| 1. Choose “Order” in main page. | 2. Direct to order page. |
| 3. Type in waiter name and table number, and then click “Submit” to initialize the order page. | 4. Display several rows according to the guest number, including name, order detail and a “Edit” button. |
| 5. Choose a guest and click “Edit” button. | 6. Direct to menu page. |
| 7. Click “Appetizer”, “Entrée” or “Dissert” to order. | 8. Prompt a list of menu. |
| 9. Sort the courses by hitting the sorting button. | 10. Display courses of the chosen category. |
| 11. Choose the courses and input the amount, click “Done” when the order is completed. | 12. Direct to order page again. Order detail should be displayed after guest name. If there is any other guest waiting to order, click “Edit” button and repeat step 5-12. |
| 13. Click “Done”. | 14. Direct to summary page. Display table number, order detail for the whole table and a “Print” button to print out the summary. |
| 15. Click “Print”. | 16. Print out an order summary of this table. |

**Post condition:** Order information should be recorded to database successfully.

**Scenarios**

**Steps:**

|  |  |
| --- | --- |
| *Actor actions* | *System responses* |
| Click on “Order” button. | Display the order page, ask for waiter name and table number. |
| Input “Alex” in waiter name, “1” in table number, click “Submit”. | Suppose there are two guest of this table. Display two rows. First row: “Guest 1”, “Blank” and a “Edit” button. Second row: “Guest 2”, “Blank” and a “Edit” button. |
| Click “Edit” button of guest 1. | Display the menu page for guest 1. |
| Click “Appetizer”, and then click “Chicken” button on the left side. | Display all the appetizer made of chicken. |
| Select “Chicken Wings” and type in “2”. |  |
| Click “Entrée”, select “Beef Ramen” and type in “1”. |  |
| Click “Done”. | Return to order page. After “Guest 1” should display the order detail: “Chicken Wings × 2, Beef Ramen × 1, Total Price $15.00”. |
| Click “Edit” button of guest 2. | Display the menu page for guest 2. |
| Click “Entrée”, check “Beef Ramen” and type in “1”. |  |
| Click “Dissert”, Select “Coconut Ice Cream” and type in “1”. |  |
| Click “Done”. | Return to order page. After “Guest 2” should display the order detail: “Beef Ramen × 1, Coconut Ice Cream × 1, Total Price $3.00”. |
| Click “Done”. | Direct to summary page. Display “Table 1: Chicken Wings × 2, Beef Ramen × 2, Coconut Ice Cream × 1”, and a “Print” button below the summary. |
| Click “Print”. | Print out order summary, return to main page. |

**Requirements**

**Functional**

* Inputs: number of guests, courses and number of courses.
* Outputs: a list of courses and the total price for each guest, a printed list of course for each table.
* Data store: a list of courses, the total price and the table number for each guest.
* Computation: the total price for each guest, and the whole table.

**Quality**

* Response time: results for actions should be completed in 2 seconds.
* Recovery from failure: data should recover to the last time when a guest’s order has made.
* Allowance for reusability: at least 50% measured in terms of lines of code, must be designed generically so that it can be reused.

**Platform**

* Operating system: Windows, Mac OS

**Process**

* Programming language: Java
* Delivery date: Nov, 2014