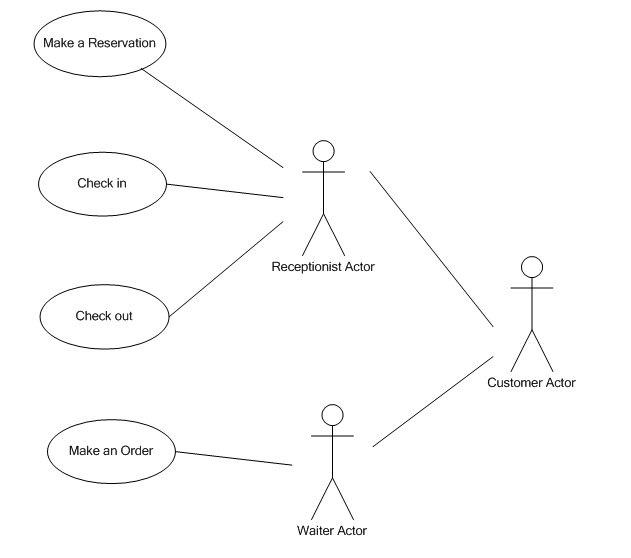
**Restaurant Management System**

**Part I**

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

**Use Case Diagram**



**Requirement Document**

**D: Functional requirement**

Check in  
1. Input: The system should accept customer name, cell phone number, group size

2. Output: The system should produce check-in result. (Exact waiter, seat or judge if it is valid reservation information)

3. Data should be stored: The customer name, cell phone number, group size

4. Computation: System should search the database to see if there are any waiters and seats are available. If yes, the system should select one of available seats and waiters.

Check out  
1.Input: The system should accept table number of customer and pay preference.

2. Output: The system should produce the result of total money a certain customer should pay.

3. Data should be stored: The total amount for a certain table.

4. Computation: System should add the price of each dish together and get total amount.

**E: Non-functional requirement**

Platform

Operating system: Windows, Mac OS

Process

Netbean/eclipse software (.java)

Quality

* Response requirement: The software should react to a user input in 1 second.
* Throughout requirement: The software can accomplish the task in 1 second.
* Availability requirement: The software is available for service when requested by end-users.
* Security: This software should be compatible with most anti-virus software.
* Usability: This software should be very easy to learn after reading rules.
* Reliability: The software needs to fail less than once a month.
* Recovery from failure: The software should automatically back up data before failure. The data can be restored after failure.

**1st Use case description**

Name of Use Case: **Check in**

Actor: Front desk receptionist

Goals: Assign an available waiter and seat to customer.

Preconditions: The receptionist must have the basic information of the customer.

Summary: When a certain customer comes to the front desk, receptionist needs to get basic information of this customer and use this software to find out an available seat and waiter to serve.

Steps:

|  |  |
| --- | --- |
| **Actor actions** | **System responses** |
| 1. Choose “Check in” command. | 2. Display a new interface to prompt receptionist to type basic information of customer. |
| 3. Type basic information (last name, cell phone number, group size) into computer.  4. Click “search” |  |
|  | 5. Display available waiter and seat condition result. |

6. Click “Check in” to confirm selection.

7. Go back to original interface.

Postcondition: The system has a record of the fact that a certain seat and waiter have been chosen as well as the change of available seats and waiters.

**Scenario of “Check in”**

1st Scenario

|  |  |
| --- | --- |
| **Actor actions** | **System responses** |
| 1. Choose “Check in” command. | 2.Display a new interface to prompt receptionist to type last name of customer, cell phone number and group size. |
| 3.Input “Chen” into “user name” text box  4.Input “123-345-3232” into “cell phone” text box  5.Input “4” into “group size” text box  6. Click “Search”  8. Click “OK” | 7. Display “David” (Waiter’s name), “3”(Table number)  9. Go back to the original interface |
| 2nd Scenario |  |

|  |  |
| --- | --- |
| **Actor actions** | **System responses** |
| 1. Choose “Check in” command. | 2.Display a new interface to prompt receptionist to type user name, cell phone number and the amount of people |
| 3. Input “Da” into “user name” text box |  |

4. Input “767-231-3211” into “cell phone” text box

5. Input “3” into “group size” text box

6. Click “Search”

7. A message box pops out and shows “Invalid information for reservation”

8. Click “OK”

9. Go back to the original interface

**2nd Use case description**

Name of Use Case: **Check out**

Actor: Front desk receptionist

Goals: To help receptionist calculate how much money customer in a certain table needs to pay.

Preconditions: The receptionist must have order information of the customer.

Summary: When a costumer finishes dinner, the receptionist needs to input the customer’s table number given by waiter into software to find out the total money he needs to pay for order.

Steps:

|  |  |
| --- | --- |
| **Actor actions** | **System responses** |
| 1.Choose “Check out” command. | 2. Display a new interface to prompt receptionist to type table number and paying preference of customer |
| 3.Type table number and pay preference (separate or together) into computer  4.Click “Search”  6. Review check  7. Click “Check out” to confirm | 5. Display the total money and order details  8. Print order check  9. Go back to the original interface |

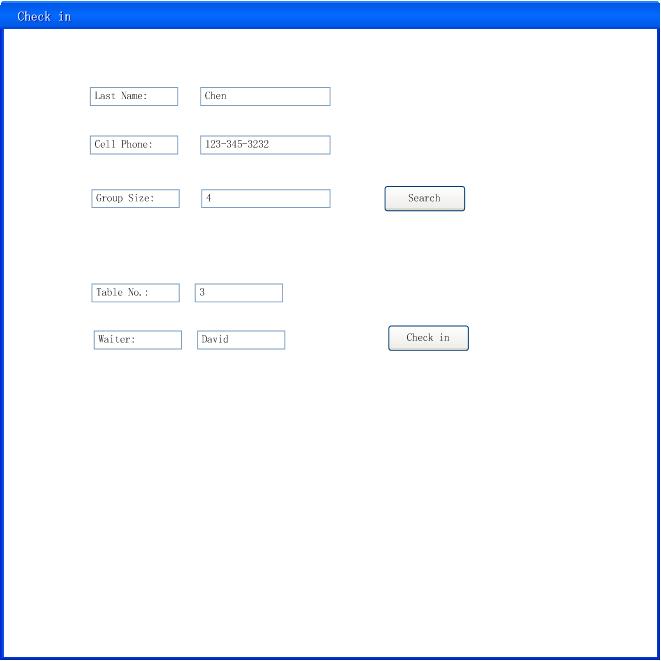
Postconditions: The system has a record of the fact that customers in a certain table have checked out.

**Scenario of “Check out”**

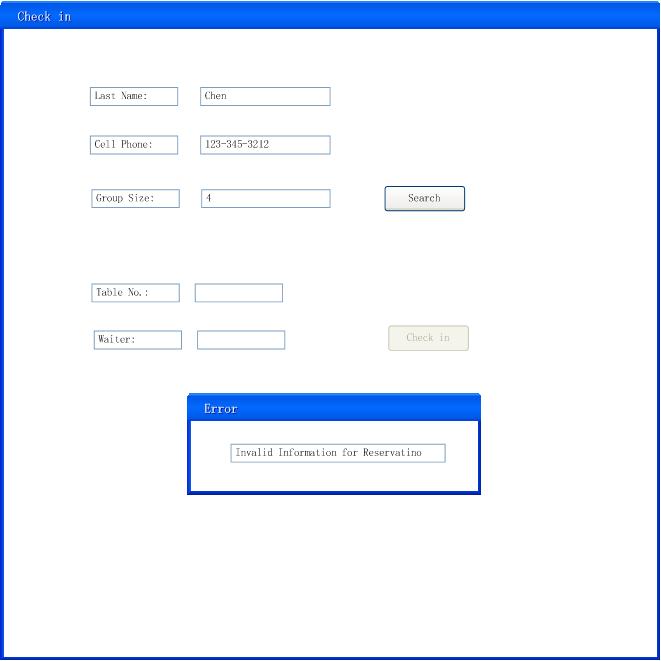
|  |  |
| --- | --- |
| **Actor actions** | **System responses** |
| 1.Choose “Check out ” command | 2.Display a new interface to prompt receptionist to type table number and choose paying preference of customer |
| 3.Input “4” |  |
| 4.Choose “Combine” |  |
| 5.Click “Search”  7. Click “check out” to confirm | 6.Display “3 Coca Cola”, “2 Chicken Sandwich”and “1 French Fries”. Also, Display “$16.30” as Total Price.  8. Print order check  9. Go back to the original interface |
|  |  |
|  |  |

**Sketches**

1st scenario of “check in”



2nd scenario of “Check in”



3. Check out

