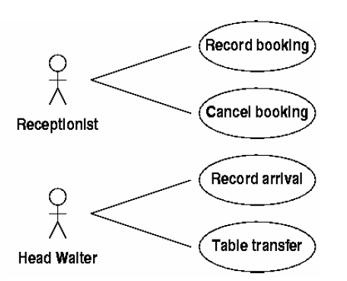
Project Requirements

* Scenario
  + The following is a sample scenario describing a customer reserving table for party from a Restaurant Booking System.
    - It’s Friday afternoon and Joe wants to hold a party tomorrow so he needs to invite friends and booking a table for it.
    - He calls his friends and counts the numbers of party.
    - He runs the Booking System and searches for a table suit on party.
    - He reserves a table with 6 seats.
    - He finds that more people would come than his expect.
    - So, he changes the table to a 8 seats one.
  + General event
  + i.e. RecordBooking
    - input date
    - System shows all booking in current date
    - Accept a new booking sheet (include table no. donated by system)
    - System takes booking sheet and update new booking info
  + Optional event flow (no available table)
  + i.e. RecordBooking
    - input date
    - System shows all booking in current date
    - No available table today
    - end use case
  + Exceptional event flow (table are too small)
  + i.e. RecordingBooking
    - input date
    - System shows all booking in current date
    - input a new booking sheet
    - system shows no available table to take enough seats
      * Respond No: End case
      * Respond Yes: Keep recording the booking sheet but set a warning mark.
* Roles identified
* Use case
  + A booking System should:
    - Record a new booking information (also called booking sheet)
    - Cancel a booking information
    - Record arrival of a customer
    - Table transfer
  + 
  + Also, system should collect those info:
    - Name, Phone number and covers. (on booking sheet)
    - For each booking, occupies an available table.
    - Record walk-ins and booking but not taking table customers. (only for counting, no phone number and name record)
    - Update the information on booking sheet
    - Whether has changed the table
    - Whether arrived or not
    - Cancellation
  + So, the basic functions are:
    - Record booking
    - Updating booking sheet (on time)
* Customer
  + Make bookings (receives booking confirmation if reference number)
  + Cancel bookings
  + Change bookings (Reschedule or change size of party)
* Staff (Front of house/Receptionist)
  + Log in to system
  + Can do everything a customer can do
  + Checking customers in or out
* Manager
  + Can do everything the staff can do
  + Print up to date list of bookings
  + Creating seating plan
  + Modify seating plan

Optional Additions

* Customer
  + Choose a table
  + Pre-order food
* Staff
  + Submit food orders

New Edition:

* Customer
  + - * Login
      * Record
      * Edit
      * Cancel
* Waiter
  + - * Record check in
      * Record change
      * Cancel check in
* New Cust
  + - * Register
      * Login as guest (cannot booking)

Business Logic

* Changing Tables
* Booking
* Location (if other restaurants has suitable table)
* Reword System
* Waiting List

In all:

* We have a booking system and it is for Chain stores like Supermacs.
* An administrator could manage all bookings and stuffs.
* Stuff has two types one is waiter another is front house.
* New customer needs to register first otherwise they can only look the booking that has already exist.
* Registered customer could have a booking on restaurant, and once it completed, the customer could have a feedback and a unique number (or random number) as an evidence.
* Booking sheet needs contains email, first name, phone number etc.
* Waiter needs to write the cost of food and name of food on it, and how many people are actually sits here on the booking sheet.
* Front house needs to receive and deal the booking (customer only face front house), checking available table and give a table number to waiter. Also, front house needs to know the show up rate (how many people has booking but not show up without cancel).
* Customer’s booking operations (include cancel, change or other) are only sent to front house then when they actually show up then send it to waiter.
* Administrator could do all things like stuff.
* A record system works on show up rate.
* (possible) After eating at restaurant, customer could give marks on front house and waiter, record system also records that.
* If not enough table, the system needs to give an option to customer:
  + If other restaurant has available table
  + Else if Using 2 or more tables to make a big table.
  + Else if Setting in waiting list.
  + Cannot order.