OOM PROJECT- HOTEL SYSTEM

Sahil Rathi(LCS2020018), Manas Agarwal(LCS2020021),
Kumar Aryan(LCS2020067), Shashwat Pandey(LCS2020010)

Goal:

Our goal is to create a simulation of a Hotel system which includes components like the people involved(customer and different types of employees), the different departments like the hotel which is often accompanied by a restaurant. Every component is intended to have their unique functionality and interact with one another. A billing system for the hotel and restaurant is also needed.

LIST OF REQUIREMENTS

Classes and Subclasses:

1. Person
    a. Employee
        - Receptionist
        - Hotel Manager
        - Waiter
        - Chef
        - Room Service
    b. Customer
2. Rooms
3. List
    a. Menu
    b. Order
    c. Billing Receipt

3. Errors

Fields and behaviour of Classes (behaviour elements are in *italics*)

Person:name, email, phone number

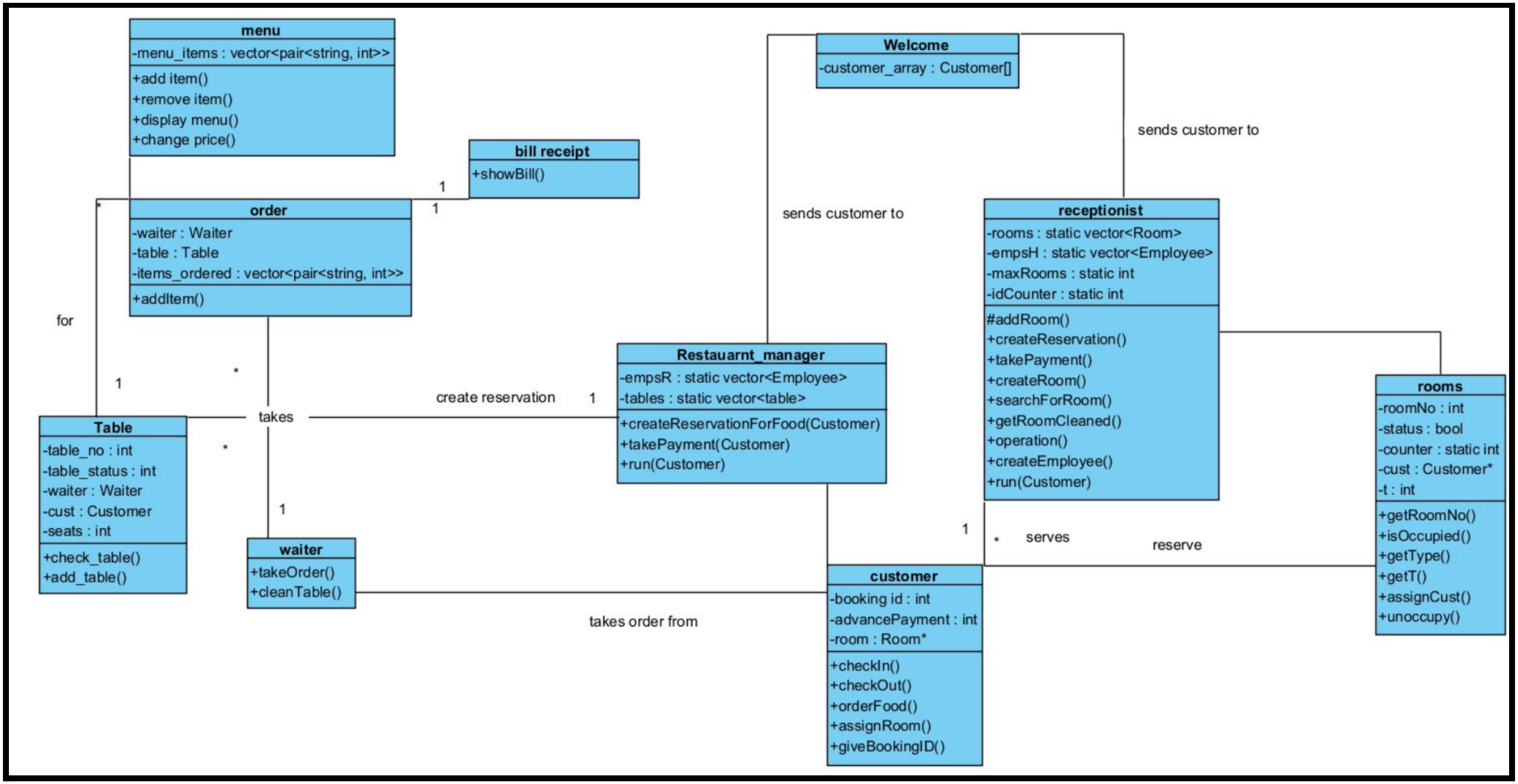
a. Employee: employee ID, employee type
    i. Receptionist: rooms(the list of rooms), employee list for hotel, maxRooms, *addRoom, add employee, take payment, print bill, search for rooms*
    ii. Waiter: *take order, clean Table*
    iii. Restaurant Manager:list of tables, employee list for restaurant, max tables, *add employee, createReservation, take payment, print bill*
    iv. Chef:
    v. Room Service:
b. Customer:booking id, advance payment, customer id, assigned room, bill, *order food, get bill, add to bill*

Rooms:id, status, type, price, *unoccupy*
Table:no, number of seats, status, *change status*
list:

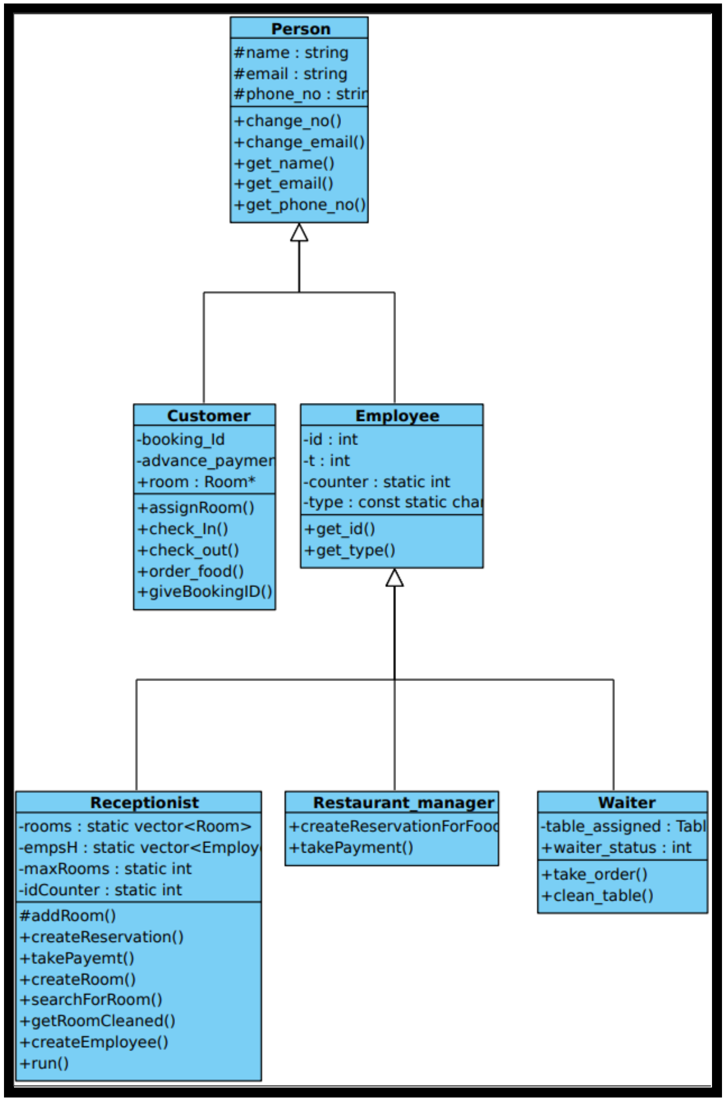
1. Menu:menu items, *add item, remove item, display menu, change price*
2. Order: id, waiter, chef , table no, items array
3. Bill receipt:room no, order id, price list from Order

DESIGN

-Main Class Diagram



-Person Class Diagram



Few more diagrams are available in the Docs directory of Hotel-System.

Current State:

- As of now we have been successful in implementing the basics of the hotel side of the project. Some employee types still need to be added.
- The billing system needs to be implemented. A very basic billing system exists.
- Currently the welcome class has been designed to create only a single customer but all the other parts of the system are completely capable of handling multiple instances of them.
- The code for the Restaurant side of things is ready but its integration with the Welcome class is not yet done. The controller function for the restaurant manager needs to be written.
- Some progress has been made on the lists class.