Group 27

Fudo Hyoka

ECS506U Software Engineering Group Project

Requirements Elicitation Report

1. List of Requirements

Note: The requirements have been set according to general and specific actor interactions with the system. The hierarchy of the actors works accordingly: a manager can do everything a chef and a Receptionist can, a Receptionist can do everything a Waiter can. The term “access level” refers to the limitations to what tasks actors can perform, e.g. Manager has “high access” and Waiter has “low access”.

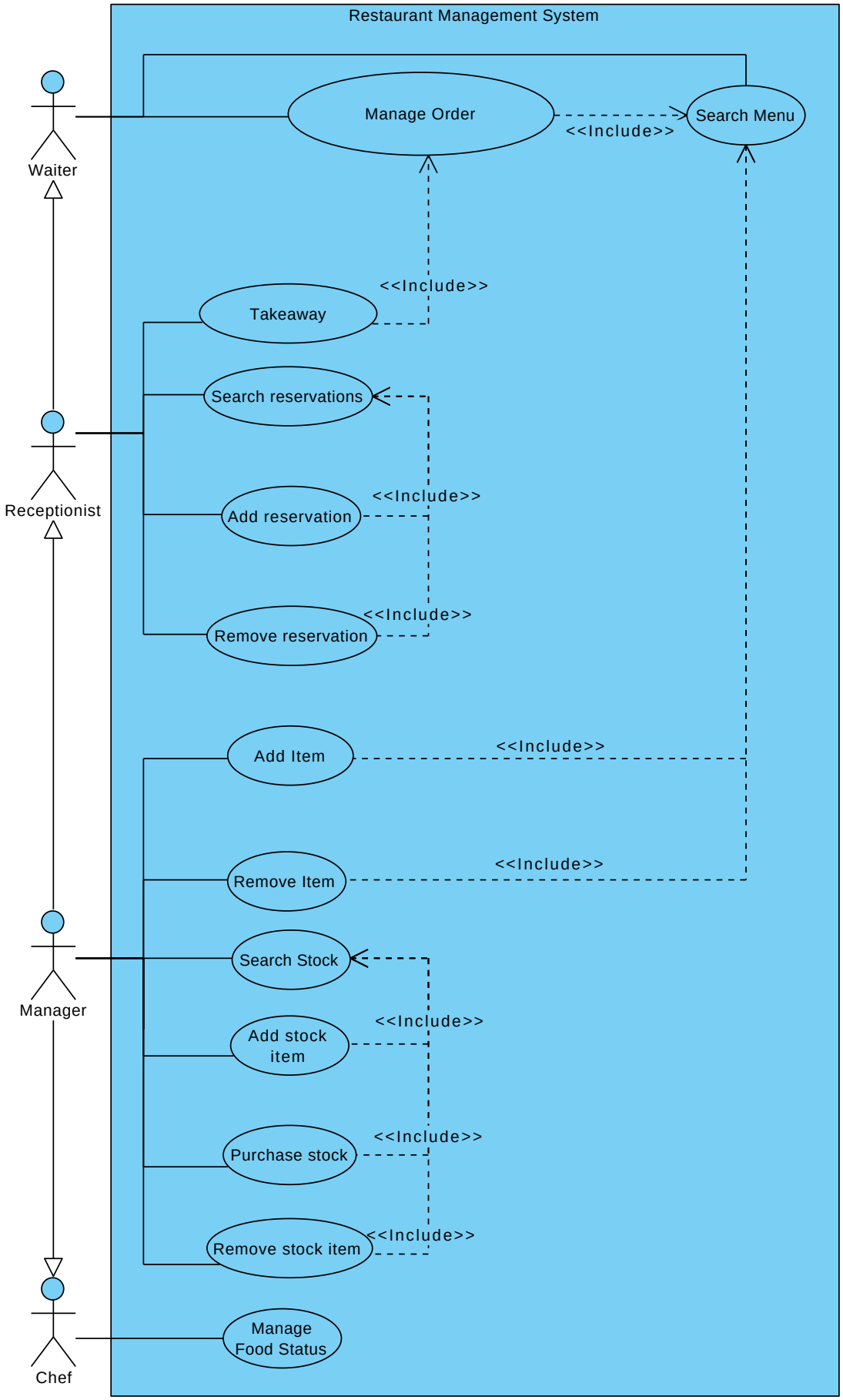
ID	Requirement	Type	Priority	Use Case
Functional requirements				
GENERAL				
1	The system must allow interaction from four types of actors: manager, receptionist, chef and waiter	Functional	Core	-
2	The system must contain a database of all actors	Functional (D)	Core	-
3	The system will restrict the actor’s interaction with the system based on their access level	Functional	Core	-
4	Each actor must be identified by unique username	Functional (D)	Core	-
5	The system must enable the adding and removal of staff accounts	Functional	Optional	-
6	The system must allow actors to log into the system, using a username and password	Functional (D)	Core	-
7	The system must allow actors to log out	Functional	Core	-
WAITER: Note: A table is associated to a customer				
1	The system must allow the creation of a new order	Functional	Core	Manage Order
2	The system must maintain a database for all orders	Functional (D)	Core	-
3	The system must associate an order to a table	Functional	Core	Manage Order
4	The system will allow adding/removal of items to the order	Functional	Core	Manage Order
5	The system allows the waiter to search menu items	Functional	Core	Search Menu
6	Menu items can be searched according to: Name, Allergies or price	Functional	Optional	Search Menu
7	The system must display an order summary for each table	Functional	Core	Manage Order
8	The system must display payment status for each table	Functional	Core	Manage Order
9	The system must store and allow manipulation of payment status for each table. Status include: Paid / Not paid	Functional (D)	Core	Manage Order
10	The system must display availability of tables	Functional	Core	-
11	The system must display order status, as either of the following: Waiting / Delivered	Functional	Core	Manage Order
12	The system must allow the manipulation of order status (i.e. when food can be delivered, mark as Delivered)	Functional (D)	Core	Manage Order

RECEPTIONIST				
1	The system must allow the addition and removal of reservations	Functional	Core	Add reservation & Remove reservation
2	The system must store a database of all reservations	Functional (D)	Core	-
3	Addition of reservations must contain: customer name, contact number and reservation date & time	Functional (D)	Core	Add reservation
4	The system must allow reservations to be searched by: customer name, contact number, date & time	Functional	Core	Search reservations
5	The system must allow an option for takeaway	Functional	Core	Takeaway
MANAGER				
1	The system must allow the addition and removal of stock items	Functional	Core	Add stock item & Remove stock item
2	The system allows the manager to add and remove items to the menu	Functional	Core	Add item & Remove item
3	The system must keep a database of all stock items available	Functional (D)	Core	-
4	All menu items must contain the following information: Name and price	Functional (D)	Core	-
5	Each menu item is assigned a number of ingredients from stock	Functional (D)	Optional	-
6	Each stock item must contain the following information: name, quantity, minimum and supply status: Waiting/Delivered	Functional (D)	Core	-
7	The system must dynamically adjust stock based on orders	Functional	Core	-
8	A collection of stock items can be selected to be purchased, changing their supply status	Functional	Optional	Purchase stock
9	The system must flag stock items that reach their minimum, alerting manager to restock)	Functional	Optional	-
10	The system must allow the searching of items in stock by: Name, quantity and those flagged	Functional	Core	Search stock
CHEF				
1	The system must allow food status to be stored and manipulated as: Ready/ Delayed/ Cancelled	Functional	Core	Manage Food Status
Non-functional requirements				
1	User Manager should be able to order re-stocking of ingredients in max of 8 clicks	Non-functional Usability	Core	Purchase stock
2	User Staff should be able to pick which food or drinks to serve with max of 3 clicks	Non-functional Usability	Core	Manage Order
3	All payments must be correct to 2 decimal places	Non-functional Usability	Core	Manage Order

4	The system should be easy to use for users with visual impairment/difficulties (e.g. option to increase size of font)	Non-functional Usability	Optional	-
5	The system must be able to process operations within 2-3 seconds	Non-functional Usability	Core	-
6	The system must contain user friendly GUI, e.g. for distinguishing reserved tables from those available	Non-functional Usability	Core	-
7	The system must run on any web browser	Non-functional Platform	Core	-
8	The system must perform comparably on different hardware devices.	Non-functional Platform	Optional	-
9	The system must be capable of handling approx.200 active orders	Non-functional Platform	Core	Manage Order
10	The system must always be available to use during operating times (restaurant opening/closing times)	Non-functional Platform	Core	-
11	The whole system must be secured. Only Manager (Admin) can access all data	Non-functional Security	Core	-
13	The system must not be accessible without login and password	Non-functional Security	Core	-
14	The system must not allow one user do the role of the other	Non-functional Security	Core	-
15	The system will use HTTPS protocol	Non-functional Security	Core	-

2. Use Case Diagram

Visual Paradigm Standard(Catherine(Queen Mary University of London))



3. Use Case Description

Note: The word staff here refers to the actors allowed to process orders: Waiter, Receptionist & Manager.

Name – Manage Order

Brief Description – Customer placing an order (either eating in or out). Staff creates a new order. Searches items in the menu and adds items to the order. Items are delivered to customer at the table, or at the till for takeaway orders. A bill is generated before finalising order payment.

Actors – All actors EXCEPT Chef

Preconditions –
Assuming the staff member has logged into the system AND has required access level. i.e. Chef does not.
Stock items must exist for requested menu items.

Basic Flow –
1. Staff member assigns customer to available table
2. Customer requests to place an order
3. Staff member creates a new order
4. Customer request items to be added to the order (food or drink).
5. REPEAT 4 until all requested items added
6. Staff member double checks with customer the items ordered.
7. Customer indicates end of order
8. Chefs receive order details and prepare food
9. Chef notifies system that food is ready
10. Staff delivers food to table
11. System computes current bill
12. Customer proceeds to pay and order is marked as paid
13. Order finalised

Alternate Flows –

1.1 No tables are available (Full) or all reserved
   a. Staff informs customers to wait or unfortunately leave

1.2 Customer requests takeaway
   a. If staff attended customer is a waiter, they refer customer to Receptionist or Manager

4.1. If item is not available, customer can:
   a. Request alternative item

6.1. Customer may want item removed
   a. Staff removes the item

8.1. Chef is unable to produce menu item
   a. Chef marks item as cancelled
   b. Waiter alerts customer

10.1 Order is takeaway
   a. Staff delivers food to customer at the counter

Post Conditions –
Order is created and associated to table
Order status changed to Paid IF payment received