

Restaurant Assignment

Case details

In a popular London restaurant, the following system is required to speed up preparation of meals.

Each waiter is assigned a group of tables, after taking orders for a table the waiters enter the orders (a list of dishes and drinks ordered by the diner or group of diners) into the system at the PC. The waiter usually knows of any dishes that are unavailable before taking an order but occasionally one of the specials will sell out. The system must confirm the availability of dishes. Should an item not be available the system must allow the waiter to change or even delete a customers order. Dishes to be prepared are sent to the kitchen, drinks orders to the bar. Starters and main course orders are usually taken together. Drinks and desert orders may be taken separately.

Kitchen staff sees the dish orders on their screen, prepare them in an appropriate sequence and confirm preparation to the system when complete, similarly with the bar.

When a waiter sees the completion indications on his terminal he collects the items and takes them to the table. The waiter can also check on the status of dish and drink orders.

At the end of the meal the waiter will have the system print a bill, and he will enter the details of payment for it. The management can give discounts.

The system keeps track of the numbers of customers served by each waiter and the amount of money taken by each waiter. The management can view these statistics.

Successful scenario

- The Diner is seated
- The Waiter takes a drinks and food order
- The Waiter enters order into the system
- The system alerts the Kitchen staff of the order
- The Kitchen staff prepare food
- The system alerts the Bar staff of the order
- The Bar staff prepare drinks
- The system alerts the Waiter that the drinks are ready
- The Waiter delivers drinks to the table
- The system alerts the waiter that the dish is ready
- The Waiter delivers the dish to the table
- The Diner orders a Desert
- The Waiter enters the order into the system
- The system alerts the Kitchen staff of the order
- The Kitchen staff prepare the food
- The system alerts the waiter that the dish is ready
- The Waiter delivers the dish to the table
- The Management decide to issue a discount from the bill
- The Waiter uses the system to print out the bill
- The Diner offers payment
- The Waiter enters the payment details into the system
- The Management uses the system to view the statistics

Alternative scenario

- The Diner is seated
- The Waiter takes a drinks and food order
- The Waiter enters order into the system
- *The system alerts the waiter that the food is out of stock*
- *The Waiter takes an alternative order from the Diner*
- *The Waiter changes the order in the system*
- The system alerts the Kitchen staff of the order
- The Kitchen staff prepare food
- The system alerts the Bar staff of the order
- The Bar staff prepare drinks
- The system alerts the Waiter that the drinks are ready
- The Waiter delivers drinks to the table
- The system alerts the waiter that the dish is ready
- The Waiter delivers the dish to the table
- The Diner orders a Desert
- The Waiter enters the order into the system
- The system alerts the Kitchen staff of the order
- The Kitchen staff prepare the food
- The system alerts the waiter that the dish is ready
- The Waiter delivers the dish to the table
- The Management decide to issue a discount from the bill
- The Waiter uses the system to print out the bill
- The Diner offers payment
- The Waiter enters the payment details into the system
- The Management uses the system to view the statistics

Use case diagram

Actor description



Waiter

The Waiter uses the system to input orders taken from the diners. They will use the system to be able to tell them if there are any dishes not available from the menu. They will also use the system to inform them when the food is ready for serving to the table.

At the end of the meal the waiter will use the system to print the bill, and enter payment details.



Kitchen Staff

The Kitchen Staff use the system to view the orders on their screen so that they can prepare them in an appropriate time and confirm preparation when complete.



Bar Staff

The Bar Staff use the system to view orders on their screen so that they can prepare them and confirm preparation when complete.



Management

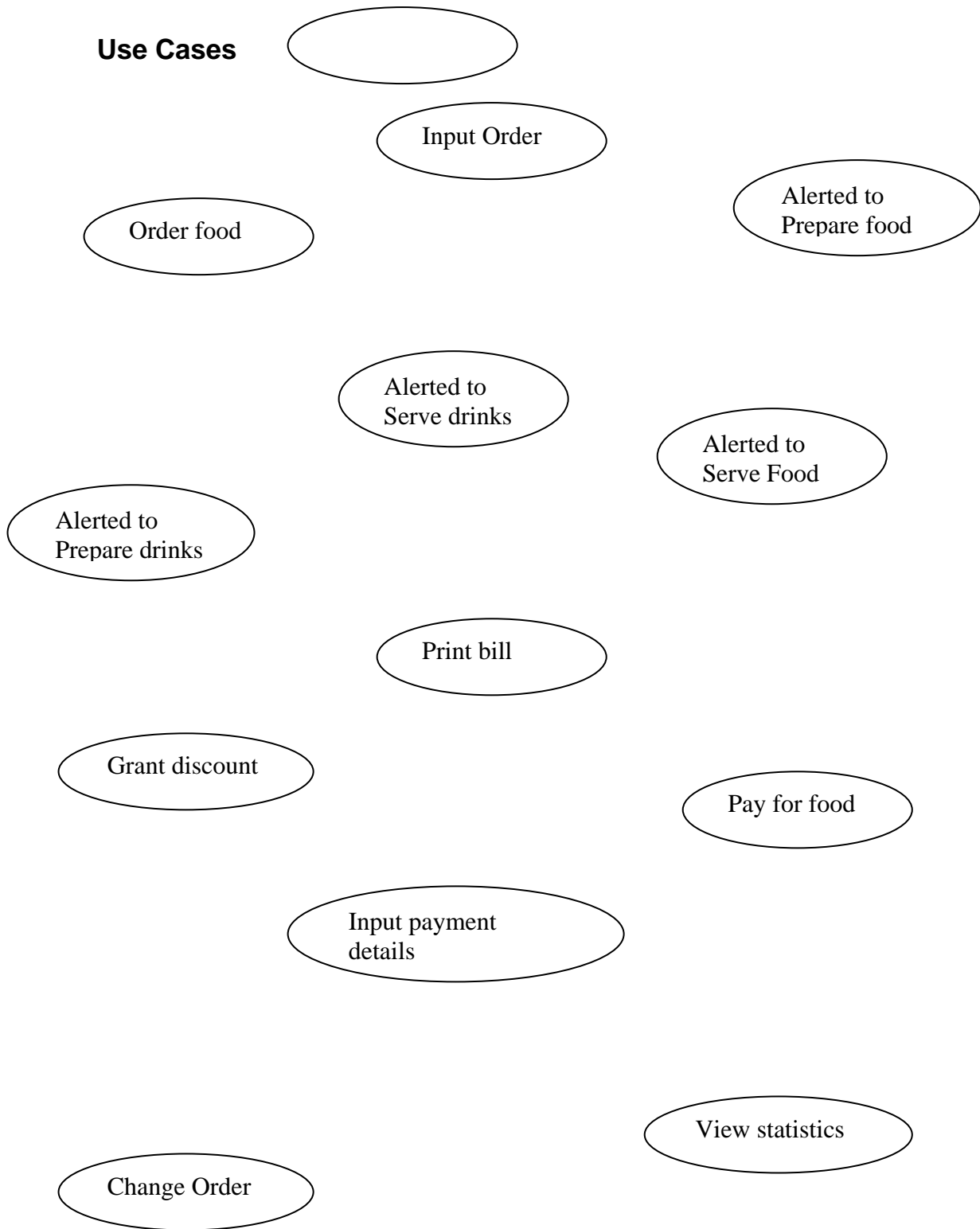
The Management use the system to view the statistics that are generated. They also use the system to grant discounts



Diner

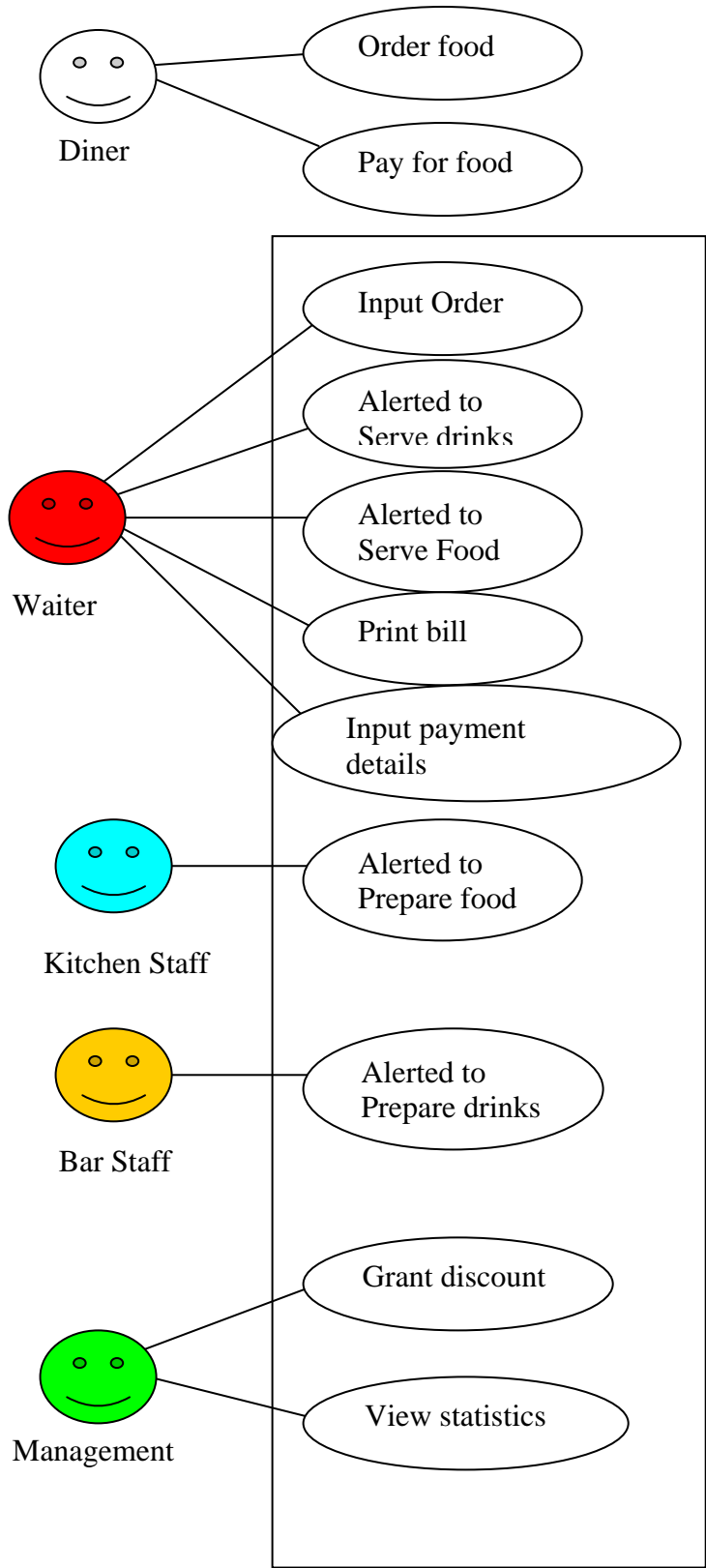
The Diner doesn't directly use the system.

Use Cases

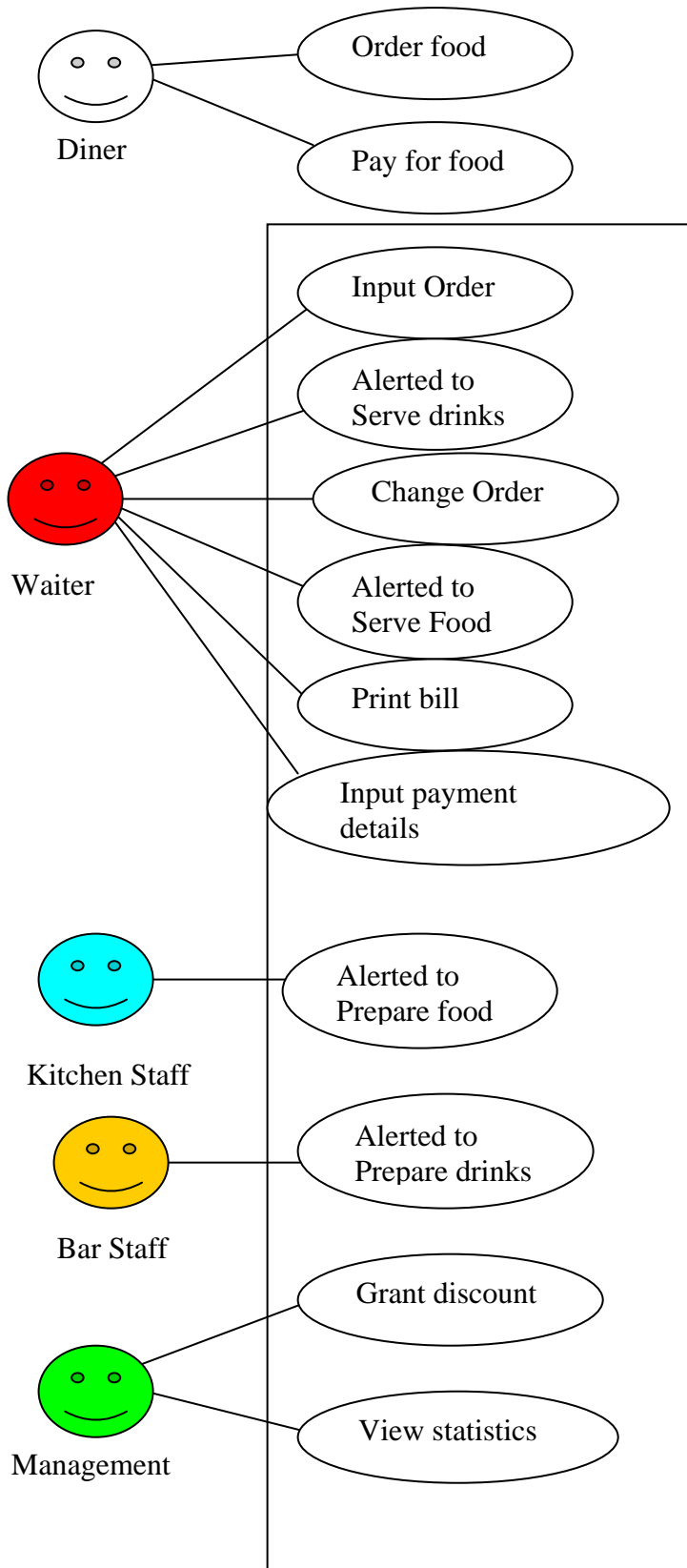


Use Cases in a Use Case Diagram

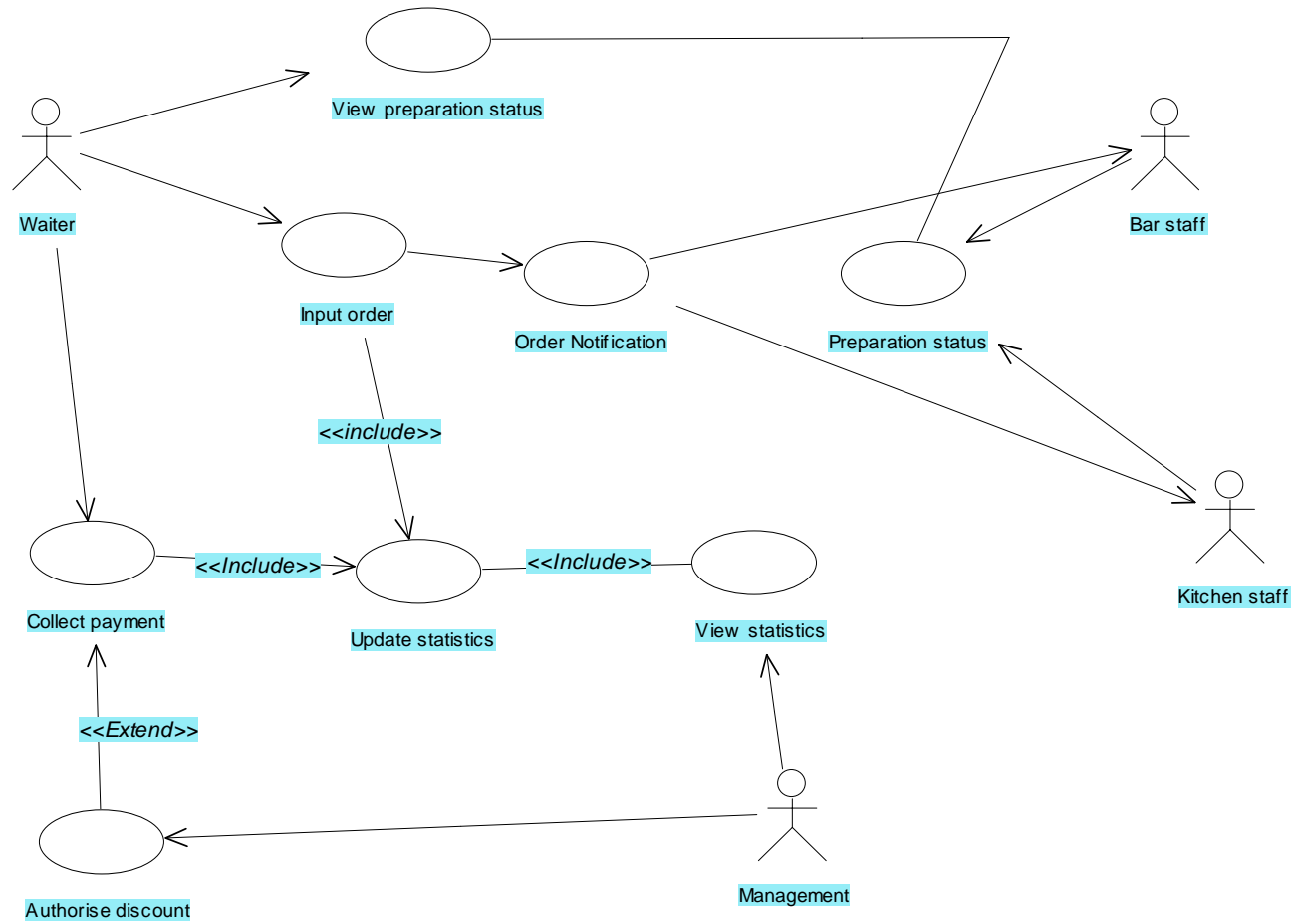
Successful scenario



Alternative scenario



Use Case Diagram – generated in Rational Rose



High Level Descriptions

Use case: Input Order
Actors: Waiter
Goal: To input an order for a meal

Description:

When a Diner orders a meal, the Waiter writes down the order and puts it into the system. The system presents this order to the Kitchen staff who prepare the food.

Use case: Prepare orders
Actors: Kitchen staff and Bar staff
Goal: To prepare the orders presented by the system

Description:

The system presents the meal order to the Kitchen staff to prepare the food. The system also presents the drinks order to the Bar staff to prepare the drink.

Use case: Announce order readiness
Actors: Kitchen staff and Bar staff
Goal: To Alert the Waiter to the readiness of the order

Description:

The Bar staff prepare the drinks and use the system to send an announcement to the Waiters terminal that they are ready to be served. The Kitchen staff prepare the dishes and use the system to send an announcement to the Waiters terminal that they are ready to be served.

Use case: Print Bill
Actors: Waiter
Goal: To print a bill for the Diner

Description:

Once the Diner has finished their meal, the Waiter uses the system to print a bill for presentation to the Diner

Use case: Input payment details
Actors: Waiter
Goal: To input payment details to the system

Description:

The Waiter takes the payment from the Diner and uses the system to input the details about the payment.

Use case: Grant discount
Actors: Management
Goal: To grant a discount from the bill

Description:
Management use the system to alter the bill that is to be presented to the Diner.

Use case: View statistics
Actors: Management
Goal: To view statistics

Description:
Management use the system in order to view various statistics about the Waiters and the amount of money that they have taken.

Extended use case description

Take order

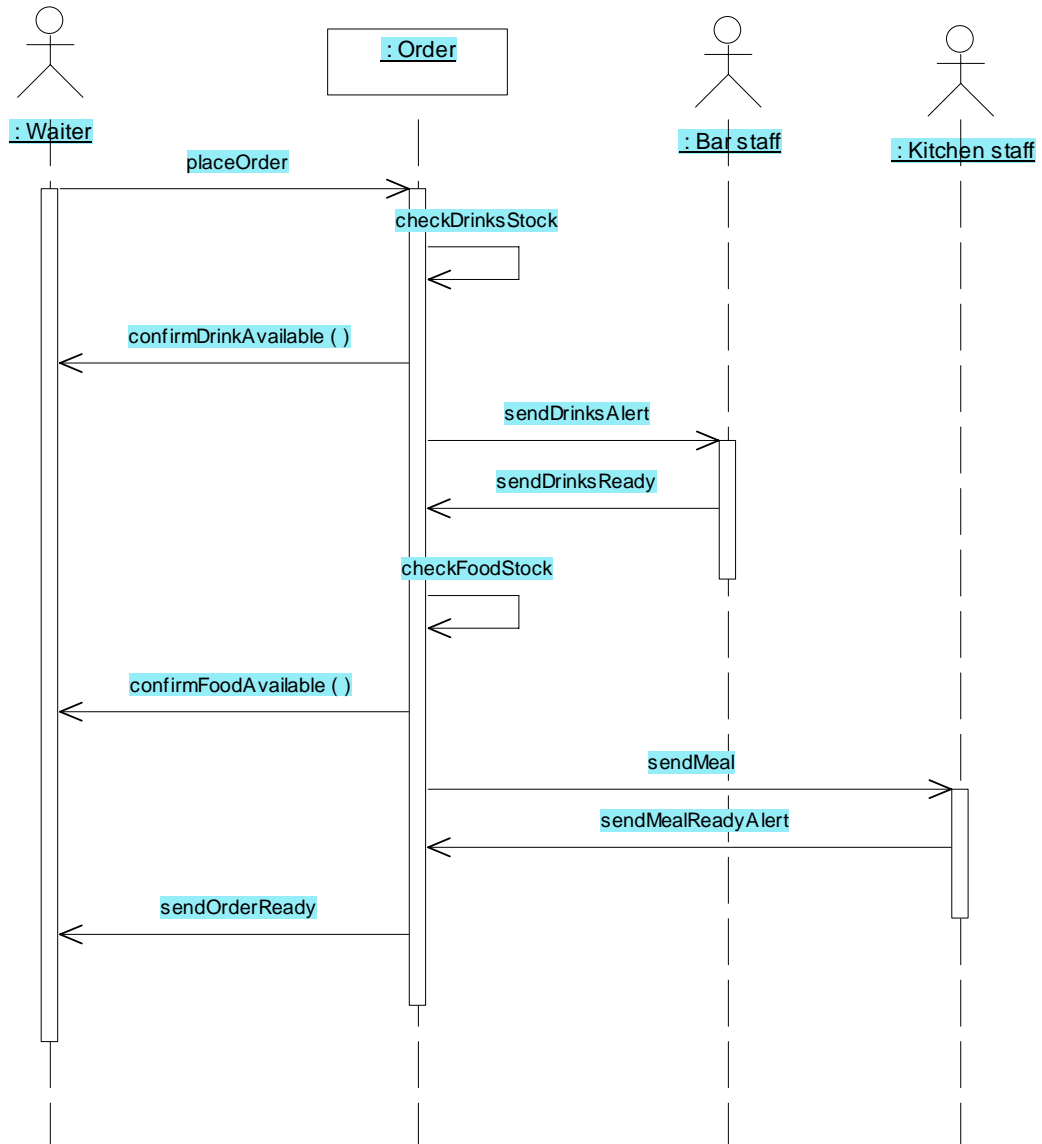
Use case:	Take order
Actors:	Waiter
Goal:	To input an order from the Diner
Overview	
When a Diner places an order with the Waiter, the Waiter inputs this into the system. The system will check the stock to ensure that the order is available. The details of the order (including an identifier for the table and Waiter) are confirmed.	
Typical course of events	
<i>Actor action</i>	<i>System response</i>
1. Begins the order process	2. requests table number
3. Inputs the table number	4. request drinks ordered
5. Inputs the drinks order	6. confirms drinks availability
	7. add selected drinks to the order
	8. request dishes ordered
9. Inputs the requested dishes	10. confirms dishes availability
	11. adds selected dishes to the order
	12. displays order
	13. requests confirmation
14. Confirm order	15. Drinks order sent to the bar
	16. Dishes order sent to the Kitchen
	17. Waiter statistics updated
Alternative courses	
Step 6. Drinks confirmation not successful. Prompt for an alteration to the order	
Step 10. Dish confirmation not successful. Prompt for an alteration to the order	

Take Payment

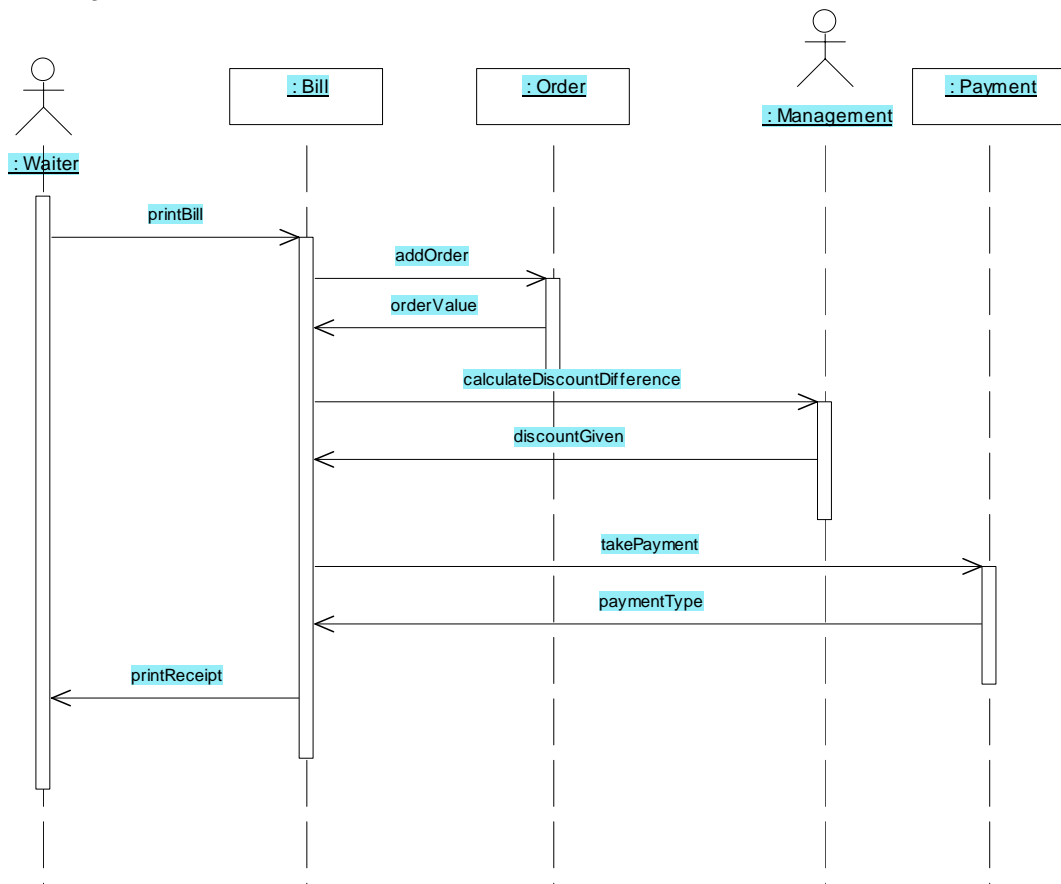
Use case:	Take payment
Actors:	Waiter
Goal:	To take payment from the customer
Overview	
When the Diner has finished their meal, the waiter will present them with their bill and take payment. These details will be entered into the system	
Typical course of events	
<i>Actor action</i>	<i>System response</i>
1. Request a bill printout	2. requests table number
3. Inputs the table number	4. print the bill requested
5. Enter payment details	6. request payment type (Cash or Card)
7. Select payment type	9. confirm card details and authorization
8. Swipe card	10. request authorizing identifier
11. Agree authorizing identifier presence	12. Verify payment
	13. Accept payment
	14. Print receipt
	17. Waiter statistics updated
Alternative courses	
Step 9. Payment authorization refused. Customer asked to provide alternative payment means.	
Step 10. Authorizing identifier not presented. Customer asked to provide alternative	

Sequence Diagram

Take Order



Take Payment



Class design

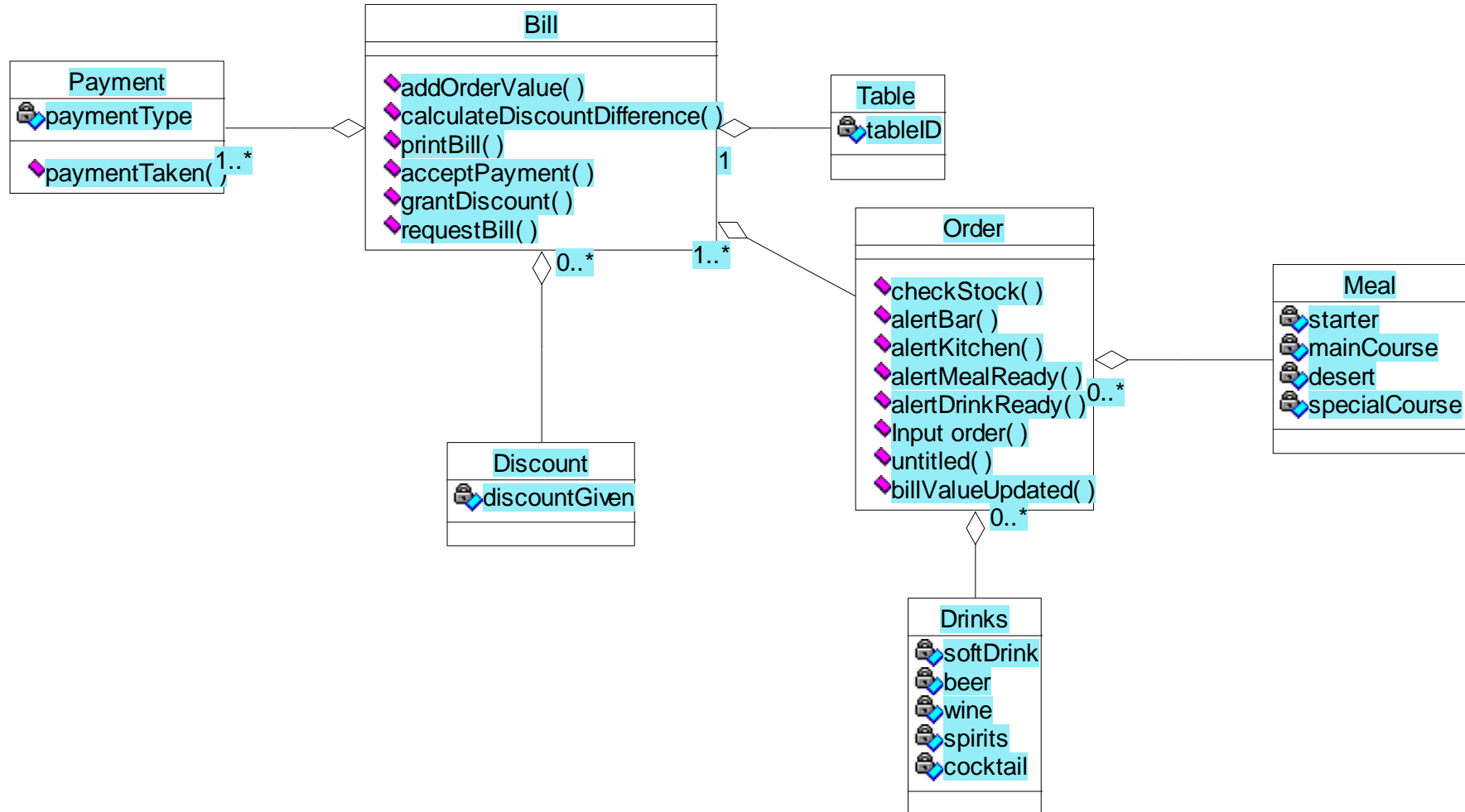
Nouns

- bar – Irrelevant – Outside scope
- bill – Class
- completion Indications – Attribute of Order
- desert – Attribute – Could be inheritance with Meal
- diner - Class
- discount – Class – Could be inheritance of Bill
- drinks – class – Could be inheritance with order
- kitchen – Irrelevant – Outside scope
- kitchen Staff – Role, user of system
- meal – class – Could be inheritance with order
- main Course – Attribute – Could be inheritance with Meal
- management – Role, user of system
- money – Attribute – Could be inheritance with payment
- order – Class
- orders – Redundant – same name as Order
- payment - Class
- restaurant – Irrelevant, too general
- screen – Output of system - Operation
- specials – Attribute – Could be inheritance with Meal
- starters – Attribute – Could be inheritance with Meal
- statistics – Output of system - Operation
- system – Too general we are modelling the classes that make up the system
- table – Class
- tables – Redundant – same name as Table
- terminal – Redundant – same as Screen
- waiter – Role, user of system
- waiters – Redundant – same name as Waiter

Classes to use

- Bill
- Discount
- Drinks
- Meal
- Order
- Payment
- Table

Class Diagram



Data Dictionary

Order = {Meal} + (Drinks)

Meal = { start | mainCourse | desert | specialCourse }

Drinks = { softDrink | beer | wine | spirits| cocktail }

Bill = Order + (discountGiven)

Activity diagram

This activity diagram models the flow of the order input to the serving of the order.

