Online Cafeteria System
Requirements Models (Sample Solution)

October 2, 2022

1 Problem Statement

Big companies often offer a cafeteria service to their employees. Typically, each day, several meals are prepared in advance, which hungry workers can enjoy for a reduced price during lunch hours. Running a full-fledged cafeteria requires considerable investment, though, which smaller companies can often not afford.

Your development team is tasked to create an online cafeteria system, which allows companies that cannot afford their own fully staffed cafeteria with kitchen to nevertheless provide daily catered lunch service at a reasonable price to their employees.

2 Informal Requirements Description

Once a company signs up for the online cafeteria, users can consult the different available lunch menus online several days in advance. Each menu lists the ingredients, its price, and configuration options, if any. To place an order, the authenticated employee simply selects the desired meal and quantity, and optionally customizes the meal, if there are configuration options. Orders can be cancelled up to midnight on the previous day. Orders can also be changed, however, it is possible that some meals are not available anymore, because the caterer usually has to place the order for ingredients three days in advance.

Early in the morning of each day, the caterer consults the list of orders and starts preparation. Once ready, the meal is packaged, labelled with a sticker that shows a bar code, the price of the meal, and the name of the employee that placed the order. The meals are then delivered to the cafeteria on the company's premises. During lunch hours, employees can pickup their order at that cafeteria, where it is also possible to buy beverages, dessert, and other snacks. Before handing the order to the customer, the cashier scans the bar code on the label, after which the price for the meal is displayed on the rotatable customer display. Credit card and cash payment is accepted. It is also possible to simply charge the cost of the meal to the employee account, the balance of which is deducted from the employee's salary at the end of the month. Any orders that are not picked up by 2:30pm are offered at a discounted price to interested customers. The cost for meals that were not picked up are charged to the account of the person that originally placed the order at the end of the day.

3 Task 1: Use Case Model

The use case model of the Online Cafeteria has already been established. The use cases and their relationships are summarized in Figure 1. The individual descriptions of each use case are given below.

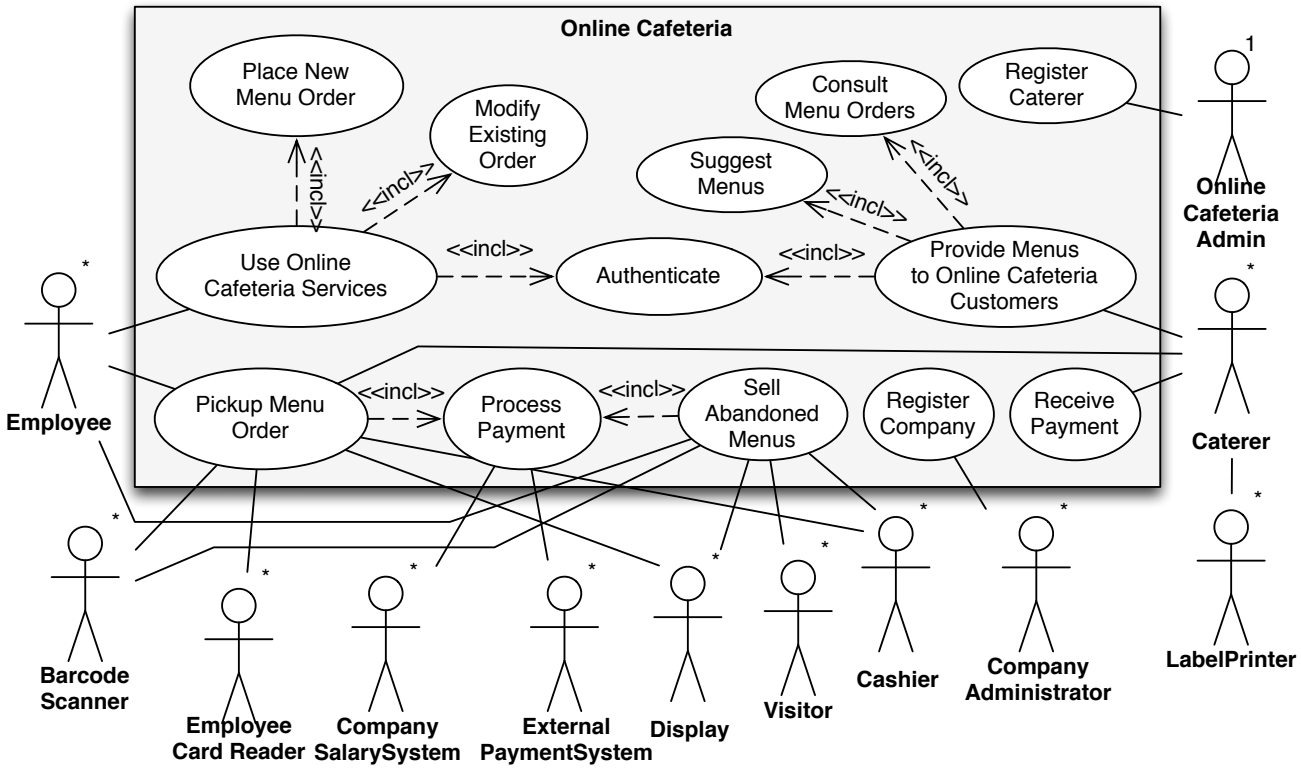

Figure 1: Online Cafeteria Use Case Diagram

Use Online Cafeteria Services

Use Case: Use Online Cafeteria Services

Scope: Online Cafeteria

Level: Summary

Intention in Context: The intention of the *Employee* is to manage his menu orders with the online cafeteria to make sure he will get catered lunch on workdays when he does not have time to prepare his own lunch.

Multiplicity: Multiple *Employees* can manage their menu orders with the online cafeteria at a given time. A given *Employee* can only have one active session with the online cafeteria.

Primary Actor: *Employee*

Main Success Scenario:

1. *Employee* authenticates with the system.
2. *System* provides the list of active orders of the employee to *Employee*.
3. *Employee* proceeds to place a new menu order or modify an existing menu order.

Step 3 can be repeated as many times as desired.

4. *Employee* informs *System* that he wishes to logout.
5. *System* informs *Employee* of successful logout.

Extensions:

2a. Authentication is not successful. Use case ends in failure.

3||a. *Employee* is idle for too long.

3||a.1 *System* informs *Employee* that his session expired because he was idle for too long. Use case ends in failure.

4a. *Employee* forgets to logout.

4a.1 *System* informs *Employee* that he was logged out due to inactivity. Use case ends in success.

Place New Menu Order Use Case

Use Case: Place New Menu Order
Scope: Online Cafeteria
Level: User-goal
Intention in Context: The intention of the *Employee* is to place a new menu order for a specific day.
Multiplicity: Multiple *Employees* can place new menu orders at a given time. A given *Employee* can only place one order at a time.
Primary Actor: *Employee*
Main Success Scenario:
1. Employee requests available menus from *System* for a specific date.
2. *System* provides available menus for a specific date to *Employee*.
3. *Employee* informs *System* what menu he wants to order for a specific date, and in what quantity, and what customization option he chooses.
Extensions:
3a. *Employee* does not find any menu he desires to order. Use case ends in failure.
3b. *Employee* works for several companies that are registered with the online cafeteria and employee did not specify a preferred cafeteria location previously.
3b.1 *System* requests *Employee* to provide preferred cafeteria location.
3b.2 *Employee* provides preferred cafeteria information to *System*. Use case continues at step 3.

Modify Existing Menu Order Use Case

Use Case: Modify Existing Menu Order
Scope: Online Cafeteria
Level: User-goal
Intention in Context: The intention of the *Employee* is to modify or cancel one of his existing menu orders.
Multiplicity: Many *Employees* can modify or cancel their existing menu orders at a given time. A given *Employee* can only modify one of his existing orders at a time.
Primary Actor: *Employee*
Main Success Scenario:
1. *Employee* informs *System* that he wants to modify an existing order (update the quantity or customization options of a menu, or cancel).
2. *System* acknowledges modification to *Employee*.
Extensions:
2a. *System* informs *Employee* that order cannot be modified / cancelled any more. Use case ends in failure.

Authenticate Use Case

Use Case: Authenticate
Scope: Online Cafeteria
Level: Subfunction
Intention in Context: The intention of the *Person* is to authenticate with the *System*.
Multiplicity: Multiple *Employees* can authenticate simultaneously. A given *Employee* can only authenticate once at a given time.
Primary Actor: *Person* (can be *Employee* or *Caterer*, but not *Visitor*)
Main Success Scenario:
1. *Person* provides user name and password to *System*.
2. System validates username and password.
3. *System* informs *Person* of successful login.
Extensions:
2a. System ascertains that the user name or password is unknown or wrong.
2a.1 *System* prompts *Person* to try again. Use case continues at step 1.
2a.1a. *System ascertains that Person entered a wrong password for the third time in a row.*
2a.1a.1 *System* informs *Person* that his/her account is now blocked. Use case ends in failure.

2b. System ascertains that the Person's account is blocked.
2b.1 *System* informs *Person* of situation. Use case ends in failure.

Provide Menus to Online Cafeteria Customers Use Case

Use Case: Provide Menus to Online Cafeteria Customers
Scope: Online Cafeteria
Level: Summary
Intention in Context: The intention of the *Caterer* is to regularly provide catered menus for the employees of companies that registered with the online cafeteria.
Multiplicity: Multiple *Caterers* can provide menus for online cafeteria customers concurrently.
Primary Actor: *Caterer*
Main Success Scenario:
1. *Caterer* authenticates with system.
Step 2 can be repeated as many times as desired.
2. *Caterer* suggests possible menus or consults the menu orders for a specific day.
3. *Caterer* informs *System* that he wishes to logout.
4. *System* informs *Caterer* of successful logout.

Extensions:
1a. *Caterer* does not have an account with the system.
1a.1 *Caterer* informs *System* that he wishes to register, providing registration details.
1a.2 *System* confirms registration to *Caterer*. Use case continues at step 2.
2a. Authentication is not successful. Use case ends in failure.
2||a. *Caterer* is idle for too long.
2||a.1 *System* informs *Caterer* that his session expired because he was idle for too long. Use case ends in failure.
3a. *Caterer* forgets to logout.
3a.1 *System* informs *Caterer* that he was logged out due to inactivity. Use case ends in success.

Suggest Menus Use Case

Use Case: Suggest Menus
Scope: Online Cafeteria
Level: User Goal
Intention in Context: The intention of the *Caterer* is to suggest possible menus for the online cafeteria.
Multiplicity: Multiple *Caterers* can suggest menus at a given time. A given caterer cannot suggest menus concurrently.
Primary Actor: *Caterer*
Main Success Scenario:
Step 1 and 2 can be repeated as many times as desired.
1. *Caterer* informs *System* of a menu (name, and possible configuration options, including a description of the ingredients, the price and the amount of days the order needs to be placed in advance) that he offers to prepare, and until when the offer is valid.
2. *System* acknowledges receipt of the offer to *Caterer*.

Consult Orders Use Case

Use Case: Consult Orders
Scope: Online Cafeteria
Level: User Goal
Intention in Context: The intention of the *Caterer* is to consult the number of orders of his menus that have been placed for a specific day to determine whether or not he needs to order additional ingredients.
Multiplicity: Multiple *Caterers* can consult orders at a given time. A given caterer consults orders sequentially.
Primary Actor: *Caterer*
Main Success Scenario:
1. *Caterer* communicates to the *System* the date for which he wants to consult the orders.

2. *System* informs *Caterer* about the orders (number, and configuration options) that were placed for the Caterers menus on the specified date.

Pickup Menu Order Use Case

Use Case: Pickup Menu Order
Scope: Online Cafeteria
Level: User Goal
Intention in Context: The intention of the *Employee* is to pickup his customized catered lunch on a specific day.
Multiplicity: Only one *Employee* can pickup his lunch at a given time.
Primary Actor: *Employee*
Secondary Actor: *EmployeeCardReader, BarcodeScanner, CompanySalarySystem, ExternalPaymentSystem, Display*
Main Success Scenario:
1. In the early morning, *System* sends *Caterer* the detailed list of menus (including quantities and configuration options) that have been ordered for the current day.

Caterer prepares the menu orders, packs them in lunch boxes, labels them, and delivers them to the location of the cafeteria on the companies premises by 11am. Employee goes to cafeteria to pickup his menu order, and passes his card over the EmployeeCardReader.

2. *EmployeeCardReader* informs *System* about the employee who wants to pickup his menu order.
3. *System* displays employee name and name of ordered menu to cashier on *Display.*

Cashier finds the lunch box with the employee's name on it.

4. *BarcodeScanner* informs *System* about the ordered menu that is being picked up.
5. *System* displays price of menu order on *Display.*
6. *System* processes payment.

Extensions:
2a. *Employee* does not pickup his menu order by the end of the day.
2a.1 *System* instructs *CompanySalarySystem* to subtract the cost for the not picked up menu from the employee's salary. Use case ends in failure.
3a. *Employee* has not placed an order for the current date.
3a.1 System notifies Cashier and Employee by means of the *Display* that no menu orders had been placed for the current date. Use case ends in failure.
4a. *Cashier* can not find the customers lunch box.
4a.1 *Cashier* informs *System* that menu order has disappeared. Use case ends in failure.

Process Payment Use Case

Use Case: Process Payment
Scope: Online Cafeteria
Level: Subfunction
Intention in Context: The system needs to process payment for a menu order.
Multiplicity: Multiple payments might have to be processed concurrently.
Primary Actor:
Secondary Actor: *BarcodeScanner, Display, EmployeeCardReader, Person*
Main Success Scenario:
1. *System* notifies *ExternalPaymentSystem* of the amount due.

At this point, the employee/visitor might want to buy additional drinks, snacks, etc. for his lunch. The handling of those items is done by the external payment system.

2. *ExternalPaymentSystem* notifies *System* that the full amount has been paid.

Extensions:
2a. *ExternalPaymentSystem* notifies *System* that external payment for the lunch (and any additional amount due to purchases of drinks, snacks, etc.) was not completed successfully in its entirety, and about the remaining amount due.
2a.1 *System* requests *CompanySalarySystem* to debit the remaining amount from the employee's salary. Use case ends in success.
2a.1a *Person* has not identified himself as an employee.

2a.1a.1 *System* requests identification of employee by means of the *Display.*

2a.1a.2 *EmployeeCardReader* informs *System* about the employee who wants to pay. Use case continues at step 2a.1.

2a.1a.2a *Cashier* requests *System* to cancel the sale. The menu order and snacks, if any, are put back on display. Use case ends in failure.

Sell Abandoned Menus Use Case

Use Case: Sell Abandoned Menus

Scope: Online Cafeteria

Level: User Goal

Intention in Context: The intention of the *Cashier* after 2:30pm each day is to sell the menus of employees who forgot to pick up their orders in order to not let food go to waste for half price to visitors or employees that forgot to order their menus in advance.

Multiplicity: Multiple *Cashiers* can sell abandoned menus simultaneously. One *Cashier* can take care of at most one sale at a time.

Primary Actor: *Cashier*

Secondary Actor: *Person, BarcodeScanner, Display*

Precondition: Some employees did not pick up their menus.

Main Success Scenario:

After 2:30pm, Visitor or Employee tells Cashier that he would like to buy an abandoned menu. Cashier picks up the abandoned menu and scans its barcode.

1. *BarcodeScanner* informs *System* which menu order is being bought.
2. *System* displays discounted price on *Display.*
3. *System* processes payment.

Register Company Use Case

Use Case: Register Company

Scope: Online Cafeteria

Level: User Goal

Intention in Context: The *CompanyAdministrator* wants to register his company with the online cafeteria in order to offer catered lunch service to the companies employees.

Pay Caterer Use Case

Use Case: Pay Caterer

Scope: Online Cafeteria

Level: User Goal

Intention in Context: The *Caterer* wants to receive payment for the menu orders he prepared and delivered to the companies cafeteria.

4 Task 2: Domain Model

Figure 2 shows the domain model that has been established during requirements elicitation to accompany the use case model.

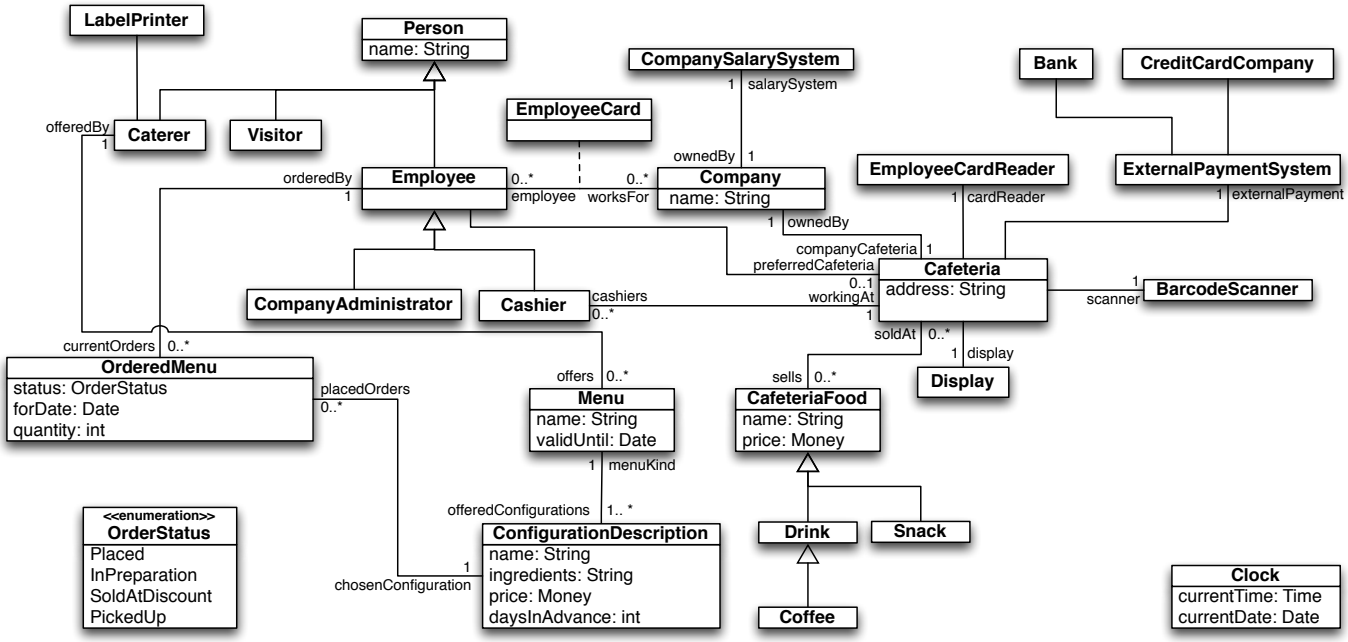

Figure 2: Online Cafeteria Domain Model