Shaikh Shawon Arefin Shimon, Nathan Lighthart, Yu Qiu, Caleb Tebbe

Abstract

This document contains Use case Analysis for A Pizza Ordering System: Use case diagram showing the actors and the use cases, and all the use cases in fully dressed essential form.

A2: Use Case Analysis

Team Project group cs414e

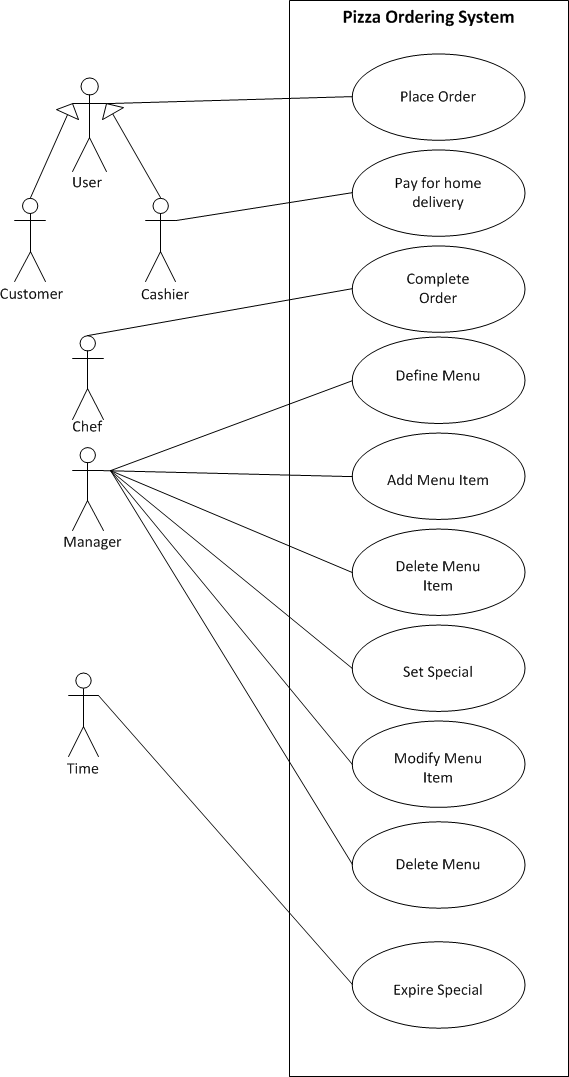


Figure 1: Use Case diagram for A Pizza Ordering System

Table of Contents

[Use case UC1: Place order 3](#_Toc399184877)

[Use case UC2: Define Menu: 7](#_Toc399184878)

[Use case UC3: Add Menu Item: 8](#_Toc399184879)

[Use case UC4: Modify Menu Item 10](#_Toc399184880)

[Use case UC5: Set Special 12](#_Toc399184881)

[Use case UC6: Delete Menu Item: 13](#_Toc399184882)

[Use case UC7: Delete Menu: 15](#_Toc399184883)

[Use case UC8: Complete Order: 16](#_Toc399184884)

[Use case UC9: Pay for Home delivery: 17](#_Toc399184885)

[Use case UC10: Expire Special 18](#_Toc399184886)

# 

# Use case UC1: Place order

**Primary Actor:** User

**Stakeholders and Interests:**

Cashier: Wants accurate, fast entry. Wants no payment errors, as cash drawer shortages will be deduced from his/her salary.

Customer: Wants purchase and fast service with minimal effort. Wants proof or order to claim order.

Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded.

Government Tax Agencies: Wants to collect tax from every successful order sale.

Payment Authorization Service: Wants to receive digital authorization request in the correct format and protocol. Wants to accurately account for their payables to the store.

Chef: Want accurate display of the orders placed.

**Precondition:**

User selects new order option from kiosk (User in this case is the Customer) or Pizza order Cashier POS screen (User In this case is the Cashier). If User selects new order option from Pizza order Cashier POS screen, user is asked for an ID and password. User enter ID and Password, Pizza store employee (Cashier) identified or authenticated.

**Success Guarantee (Post condition):**

Order is displayed to the chef. Sale is saved. Tax is correctly calculated. Receipt is generated.

**Main Success Scenario (or Basic Flow):**

1. User starts a new order in Pizza Ordering System.

2. Pizza Ordering System enters item entry mode and shows User the available menu on Screen with description and price.

3. User selects item from the screen.

4. Pizza Ordering System adds the selected item to current Order. Running total is updated by Pizza Ordering System for current order according to a set of price rules. User is shown his updated order list.

User repeats step 3-4 until indicates done.

5. User orders the selected items through Pizza Ordering System.

6. Pizza Ordering System asks the User if the orders are final.

7. User responds in agreement and finalizes order.

8. Pizza Ordering System asks the User if the orders are in-house, take-away or home-delivery.

9. User selects in-house.

10. Pizza Ordering System marks order as in-house.

11. Pizza Ordering System displays total and asks for payment.

12. User pays and Pizza Ordering System handles payment.

13. Pizza Ordering System log completes sale, updates pending order list and updates pending order list.

14. System presents receipt.

15. User takes receipt and waits.

**Extensions (or Alternative Flows):**

\*a. At any time in this use case, User can cancel the whole order (before final payment is done). The Pizza Ordering System discards the order and initializes to take a new order.

3-4a. Customer changes their mind and decides to delete items from the current order.

1. User selects items to be deleted off the current order.

2. Pizza Ordering System responds by deleting items from the current order, updating the running total and showing the User the updated list. Flow goes back to step 3.

7a. User decides to add or delete some items from the current order.

1. User selects modify order.

2. Pizza Ordering System responds by going back to Item entry mode.

9-10a. User decides to order take-away.

1. User selects take-away.

2. Pizza Ordering System responds by marking this order as a take-away. Flow goes to step 11.

9-13b. User decided for home-delivery.

1. User selects home-delivery.

2. Pizza Ordering System marks order as home-delivery and asks the User for home address for package to be delivered.

3. User enters home address.

4. Pizza Ordering System updates information of delivery address for current order.

5. Pizza Ordering System marks order as Cash-on-Delivery, updates pending order list and sends information about new order to the chef. Flow goes to System presenting receipt.

12a. Paying by Cash (To Cashier):

1. Customer gives cashier the cash amount.

2. Cashier (User) enters the cash amount tendered.

2. System presents balance due, and releases the cash drawer.

3. Cashier (User) deposits cash tendered and returns balance in cash to customer.

4. System records the cash payment.

12b. Paying by Cash (To Cash taking Machine):

1. Cash taking machine displays cash amount to tender.

2. Customer (User) inserts cash to input.

3. Machine displays cash amount entered.

4. Customer (User) hits payment button.

5. Machine calculates amount to return to the Customer (User) and drops changes in the output. Pizza Ordering System records the cash payment.

6. Pizza Ordering System goes to log sale complete step.

12c. Paying by Credit (Card Swipe):

1. Customer enters their credit account information.

2. Pizza Ordering System displays the payment for verification.

3. Customer confirms.

3a. Customer cancels payment step:

1. Pizza Ordering System goes back to item entry mode.

4. Pizza Ordering System sends payment authorization request to an external Payment Authorization System and requests payment approval.

5. Pizza Ordering System receives payment approval, signals approval to Cashier and releases cash drawer (to insert credit payment receipt).

5a. Pizza Ordering System receives payment denial:

1. System signals denial to Cashier.

2. Cashier asks customer for alternate payment.

5b. Payment timeout occurs:

1. System signals timeout to Cashier.

2. Customer tries again, or tries for alternate payment.

6. Pizza Ordering System records credit payment, which includes the payment approval.

7. System presents credit payment signature input mechanism.

8. Cashier asks customer for a credit payment signature.

9. Customer enters signature on credit swipe touchscreen.

9a. Customer signature on paper receipt:

1. Customer enters credit card signature on paper receipt and gives it to cashier.

2. Cashier places receipt in cash drawer and closes it.

10. Pizza Ordering System goes to log sale complete step.

12d. Paying by Debit (Card Swipe):

1. Customer enters their credit account information.

2. Pizza Ordering System displays the payment for verification.

3. Customer confirms.

3a. Customer cancels payment step:

1. Pizza Ordering System goes back to item entry mode.

4. Pizza Ordering System asks for Debit card Pin.

5. Customer enters Debit card Pin number and selects to proceed.

5a. Customer enters wrong pin and wants to cancel:

1. Customer selects to cancel entered pin and re-enters pin number.

6. Pizza Ordering System sends payment authorization request to an external Payment Authorization System and requests payment approval.

7. Pizza Ordering System receives payment approval, signals approval to Cashier and releases cash drawer (to insert credit payment receipt).

7a. Pizza Ordering System receives payment denial:

1. System signals denial to Cashier.

2. Cashier asks customer for alternate payment.

7b. Payment timeout occurs:

1. System signals timeout to Cashier.

2. Customer tries again, or tries for alternate payment.

8. Pizza Ordering System records Debit payment, which includes the payment approval.

9. Pizza Ordering System goes to log sale complete step.

**Special Requirements:**

1. Touch screen UI for Kiosk. Text has to be visible from 1.3 meters.
2. Screen on customer side of Cashier POS booth displaying current items and current running total for ongoing order.
3. Payment authorization response within 30 seconds 90% of the time.
4. Instantaneous update of ordered item to chef screen after sale completion is logged.

**Technology and Data Variation List:**

1. The Pizza Ordering System must be implemented by Java.
2. Card information can be entered by keyboard or card scanner.
3. Credit card payment signature can be captured on paper receipt or using digital signature capture

**Frequency of Occurrence:**

Could be nearly continuous. Orders can be given in parallel.

**Open issues:**

What are the tax law variations?

What about tips for Pizza store employees?

How can cash taking machine interact with kiosks?

# Use case UC2: Define Menu:

**Primary Actor:** Manager

**Stakeholder and Interests:**

**Manager:** Wants securely control the menu, so that people who don’t have access cannot define the menu.

**Precondition:**

1. There is a new menu to be defined.
2. Manager has a super-user account to log in Pizza Ordering System.
3. Manager logs into the system successfully.

**Success Guarantee (Post condition):**

A new empty menu is defined and manager can add menu items to the menu.

**Main Success Scenario (or Basic Flow):**

1. Manager opens up manage menu option in the Pizza Ordering System and selects “define Menu”.
2. The Pizza Ordering System responds by asking manager for the name of the new menu.
3. Manager types in the name of the new menu.
4. The Pizza Ordering System responses by a prompt of confirmation.
5. Manager confirms the prompt.
6. The Pizza Ordering System creates a new empty menu.

**Extension (or Alternative Flows):**

3a. Manager decides not to create a new menu:

1. Manager selects cancel create the menu.
2. The Pizza Ordering System back to the manage menu page.

4a. Manager provides repetitive name of the new menu:

1. The Pizza Ordering System detects the name manager provided is repetitive.
2. The Pizza Ordering System shows a notification to tell manager the name is repetitive.
3. The system goes back to step 2 in the main success scenario.

5a. Manager decides not to create a new menu:

1. Manager selects cancel create the menu.
2. The Pizza Ordering System back to the manage menu page.

**Frequency of Occurrence:**

Not frequent

# Use case UC3: Add Menu Item:

**Primary Actor:** Manager

**Stakeholders and Interests:**

Cashier: Wants clear and unambiguous menu item description, so as to provide accurate information to the customer.

Customer: Wants to find the food he/she wants as soon as possible. Wants to know the detailed information of the food, including the exact price.

Manager: Wants to securely control the menu, so that people who don’t have access cannot add a new menu item to a menu.

**Precondition:**

1. Manager have a super-user account to log in Pizza Ordering System.
2. Manager log in to the super-user account in the Pizza Ordering System.
3. The system must have at least one Menu. The manager selects the Menu that the menu item is going to be added to.

**Success Guarantee (Post condition):**

New menu item entry is successfully added to the menu that the manager selects.

**Main Success Scenario (or Basic Flow):**

1. Manager selects the menu that is going to contain the new menu item.
2. Manager opens up manage menu option for that menu in the Pizza Ordering System and selects “Add new Menu item”.
3. The Pizza Ordering System responds by showing the manager a form to fill in details for the new menu entry.
4. Manager fills in the details of new menu entry and sets a price for that entry.
5. Manager decides to save the newly created menu entry and tells the system to save the new entry.
6. The Pizza Ordering System checks if the essential fields like name and price have been filled in. If they have been entered correctly, then responds by giving a prompt to manager asking if he wants to confirm.
7. Manager confirms the prompt.
8. The Pizza Ordering System responds by saving the new Menu entry to the current menu.

System goes back to Manage menu option for the menu under which the new menu item was added.

**Extensions (or Alternative Flows):**

2-7a. Manager decides to add an existing item in another menu to current menu:

1. Manager opens up manage menu option for that menu in the Pizza Ordering System and selects “Add Existing Menu item”.
2. The Pizza Ordering System responds by showing the manager a list of all items in all other Menus.
3. Manager selects the item he wants to add to the current item and tells the System to add it to the current Menu.
4. System asks for confirmation.
5. Manager Confirms.
   1. Manager Denies
      1. System goes back to showing all items in all other menus.
6. System removes the selected item from the previous shown list and adds it to the current menu. System goes back to showing the list of all items in all other menus.

The manager repeats step 3-6 for all items he wants to add to the current menu.

1. Manager selects cancel in the shown list to get back to Manage Menu option.

3-5a. Manager decides not to create a new entry:

1. Manager selects cancel in the form.
2. The Pizza Ordering System shut down the form window and no new entry in the current window is created.
3. The system goes back to the general list of menus.

6a. Essential fields not filled:

1. The Pizza Ordering System detects that one or more essential fields are not filled.
2. The Pizza Ordering System goes back to the step 5 of the main success scenario at the partially filled list and marks which fields needs to be filled.

6b. Manager filled in repetitive entry:

1. The Pizza Ordering System detects the entry that the manager try to create has the repetitive entry name as an existing entry in the menu.
2. The Pizza Ordering System shows which existing entry has the repetitive name.
3. The Pizza Ordering System goes back to the step 5 of the main success scenario at the partially filled list and marks the unique field id that caused the repetition (Menu item name).

7a. Manager decides to cancel the new entry:

1. Manager cancels his new entry creation at the prompt process.
2. The Pizza Ordering System back to the previous form again at step 5 in the main success scenario.

The Pizza Ordering System keeps the information manager has filled.

**Frequency of Occurrence:**

Will not be too frequent.

**Open issues:**

How many fields should the system provides to the manager to fill?

How many of them are essential fields?

When should the Pizza Ordering System update the menu for the new entry? (Immediately or the next day?)

# Use case UC4: Modify Menu Item

**Primary Actor(s):** Manager

**Stakeholders and Interested parties:**

Cashier: Needs to be informed on the current menu to guide customers.

Customer: Has an interest in knowing the current menu to place a valid order.

Manager: Must keep the menu up to date depending on materials available.

Chef: Must prepare the items available on the menu.

**Preconditions:**

1. Manager has a super user account to log in to the pizza ordering system
2. Manager logs in to the system.
3. There is a menu defined which contains at least 1 item.
4. The menu selected contains the menu item to be edited.

**Post-conditions:**

1. A menu item is edited to the Manager's liking / new business rule
2. The new menu item is saved in the system with the edit done.

**Main Success Scenario (or Basic Flow):**

1. Manager selects a menu that contains the item to be modified.
2. Manager opens up manage menu option for that menu in the Pizza Ordering System and selects “Modify Menu”.
3. The Pizza Ordering System responds by showing all existing menu items in the current menu.
4. Manager selects the desired menu item and hits “Modify” option.
5. The Pizza Ordering System responds by showing the manager a form showing details of selected item.
6. Manager changes either name, price, or description of item. Manager decides to save the newly created menu entry and tells the system to save the new entry.
7. The Pizza Ordering System checks if the essential fields like name and price have been filled in. If they have been entered correctly, then responds by giving a prompt to manager asking if he wants to confirm.
8. Manager confirms the prompt.
9. The Pizza Ordering System responds by saving the edited Menu item entry to the current menu. System goes back to step 3.

Step 3 to 9 can be repeated by the manager as many times as he wants to until all the items edited are edited to his satisfaction.

**Alternative Flows:**

3-5a. Manager decides not to create a new entry:

1. Manager selects cancel option in the current list screen.
2. The Pizza Ordering System shut down the list screen and no new entry in the current window is created.
3. The system goes back to the general list of menus.

7a. Essential fields not filled:

1. The Pizza Ordering System detects that one or more essential fields are not filled.
2. The Pizza Ordering System goes back to the step 5 of the main success scenario at the partially filled list and marks which fields needs to be filled.

7b. Manager filled in repetitive entry:

1. The Pizza Ordering System detects the entry that the manager try to create has the repetitive entry name as an existing entry in the menu.
2. The Pizza Ordering System shows which existing entry has the repetitive name.
3. The Pizza Ordering System goes back to the step 5 of the main success scenario at the partially filled list and marks the unique field id that caused the repetition (Menu item name).

9a. The manager cancels selected option.

1. Manager cancels his selection
2. System responds by going back to step 3 in Basic flow.

**Frequency of Occurrence:**

May be either frequent or infrequent depending on the establishment.

# 

# Use case UC5: Set Special

**Primary Actor(s):** Manager

**Stakeholders and Interested parties:**

**Customer:** Has an interest in knowing the current special to make an informed order.

**Preconditions:**

1. Manager has a super user account to log in to the pizza ordering system
2. Manager logs in to the system.
3. There is a menu defined which contains at least 1 item.
4. The menu selected contains the menu item to be set as special.

**Post Conditions:**

1. An item has been selected and it's special status has been set.

**Basic Flow:**

1. Manager selects a menu that contains the item to be set as special.
2. Manager opens up manage menu option for that menu in the Pizza Ordering System and selects “Modify Menu”.
3. The Pizza Ordering System responds by showing all existing menu items in the current menu.
4. Manager selects the desired menu item and hits “Set Special for today” option.
5. System responds by giving a prompt to the manager asking to confirm his selection.
6. Manager confirms his selection.
7. The system saves the selected menu item as the special for the day. System goes back to showing the general list of menus.

**Alternative Flows:**

3a. Manager decides not to set special for the day.

1. Manager selects cancel option in the current list screen.
2. The Pizza Ordering System shut down the list screen and no special menu is created for current day.
3. The system goes back to the general list of menus.

6a. The manager cancels selected option.

1. Manager cancels his selection
2. System responds by going back to step 3 in Basic flow.

**Frequency of Occurrence:**

Every day at-least once.

**Open Issues:**

1. Is there a restriction on the number of specials that can be set on a menu?

# 

# Use case UC6: Delete Menu Item:

**Primary Actor:** Manager

**Stakeholders and Interests:**

Cashier: Wants to offer valid entries to the customer.

Customer: Wants his order to be valid. If he orders entries the store cannot offer, he will feel unpleasant.

Manager: Wants securely control the menu, so that people who don’t have access cannot define the menu.

**Precondition:**

1. Manager has a super-user account to log in Pizza Ordering System.
2. Manager log in to the super-user account in the Pizza Ordering System.
3. At-least one menu exists in the system.
4. The menu that the manager selects contains the menu item to be deleted.

**Success Guarantee (Post condition):**

The menu item deleted is no longer in the menu.

**Main Success Scenario (or Basic Flow):**

1. Manager selects a menu that contains the item to be deleted.
2. Manager opens up manage menu option in the Pizza Ordering System and selects “Modify Menu”.
3. The Pizza Ordering System responds by showing all existing menu items in the current menu.
4. Manager selects entries to be deleted and hits the delete option.
5. The Pizza Ordering System lists all the entries that manager selects and asks for a delete confirmation.
6. Manger confirms his selection.
7. The Pizza Ordering System deletes the selected entries. Those entries won’t be on the menu.

**Extensions (or Alternative Flows):**

3-4a. The Manager decides not to delete any item in the current menu, and wants to exit this option.

1. Manager selects cancel in the form.
2. The Pizza ordering System closes the current list shown and goes back to the general list of menus.

3b. The Pizza Ordering System provides a filter for manger to locate the entry:

1. Manager types in the name or select other information in the check box about the entry he wants to delete.
2. Manager selects the entry he wants to delete in the filtered list. Flow goes back to step 4 in the Main success Scenario.

4a. If manager does not select any entry and hits the delete option:

1. The Pizza Ordering System will show that no entries have been selected.
2. The Pizza Ordering System will back to the menu list in step 3 in the main success scenario.

6a. Manager wants to cancel/edit his selection:

1. Manger cancels his deletion when the Pizza Ordering System asks him for confirmation.
2. The Pizza Ordering System responds by back to the menu list again.
3. The Pizza Ordering System keeps the previous selection that manger has made at step 4 of the main success scenario.

**Frequency of Occurrence:**

Will not be too frequent.

# 

# Use case UC7: Delete Menu:

**Primary Actor:** Manager

**Stakeholders and Interests:**

Manager: wants to remove the entire menu because it is not going to be used any more.

**Precondition:**

1. Manager has a super-user account to log in Pizza Ordering System.
2. Manager log in to the super-user account in the Pizza Ordering System.
3. At-least one menu exists in the system.

**Success Guarantee (Post condition):**

The menu is completely removed from the system.

**Main Success Scenario (or Basic Flow):**

1. The manager signals to the system that he or she would like to delete a menu.
2. The system displays a list of all the menus known to the system.
3. The manager selects the menu that should be deleted.
4. The system shows a confirmation prompt.
5. The manager confirms the delete.
6. The system deletes the menu.

**Extensions (or Alternative Flows):**

\*a. Manager can exit this operation

1. The manager signals that no more menus should be deleted.

2. The system no longer allows the menus to be deleted at this time.

5a. Manager does not confirm delete.

1. The manager signals that no menu should be deleted.
2. The system does not delete the menu and prompts for another menu item to be selected.

**Frequency of Occurrence:**

Not very often.

**Open issues:**

None

# 

# Use case UC8: Complete Order:

**Primary Actor:** Chef

**Stakeholders and Interests:**

Chef: Wants clear out completed orders, so they do not mix with the uncompleted orders.

Customer: Wants to know that his or her food is ready to eat.

Server/Delivery man: Wants that the order should be given to the customer.

**Precondition:**

Order list must contain some items that are not marked as complete.

**Success Guarantee (Post condition):**

The order is marked as completed.

**Main Success Scenario (or Basic Flow):**

1. The chef tells the system to display the orders that are pending.
2. The system displays list of orders that are marked as pending.
3. As soon as the chef finishes preparing the food it tells the system to mark the order as completed.
4. The system marks the order as complete and updates the display list.
5. If the order is in-house, the system displays the order as complete in the customer area.

**Extensions (or Alternative Flows):**

5a. the system displays the order as complete in the delivery man area since the order needs to be delivered.

**Frequency of Occurrence:**

Could be nearly continuous.

**Open issues:**

How should the chef tell the system that an order would like to be completed?

How does the chef know what orders to cook since the chef only knows when he or she wants to complete an order?

# 

# Use case UC9: Pay for Home delivery:

**Primary Actor:** Cashier

**Stakeholders and Interests:**

Cashier: Wants to get the cash payment for the ordered pizza as well as a tip.

**Precondition:**

The system has at least one order marked as complete and is supposed to be cash is expected on delivery.

**Success Guarantee (Post condition):**

The payment for the order is completed.

**Main Success Scenario (or Basic Flow):**

1. The cashier tells the system that cash has been received for a cash on delivery order.
2. The system displays a list of orders that are cash on delivery that have not been paid.
3. The cashier selects the specified order that the cash is for.
4. The system marks the order as paid.

**Extensions (or Alternative Flows):**

No extensions

**Frequency of Occurrence:**

Not very often.

**Open issues:**

What the cashier does with the cash?

# 

# Use case UC10: Expire Special

**Primary Actor:** Time

**Stakeholders and Interests:**

Manager: Manager want a system that does not have a special item set at the beginning of the day, and wants the Special item that was set for the day to expire after the day has passed.

**Precondition:**

System recognizes the beginning of the new day or it boots up for the first time in a certain day. There should be some menu items that are set as special of the day.

**Success Guarantee (Post condition):**

The Items that were marked as Special of the day are unmarked as special of the day. The system does not contain any special items of the day.

**Main Success Scenario (or Basic Flow):**

1. System Clock (Time) generates an event to the system when a new day has started, or the system has booted for the first time in a day.
2. System catches the event and runs a query on the list of items in all menus to find the specials of the day. System gets a list of all special items of the day. System resets those items to their default values as not enabled as special items of the day.

**Extensions (or Alternative Flows):**

No extensions

**Frequency of Occurrence:**

Every day, once.

**Open issues:**

How can system clock trigger an event in the system to start the Use case?

How can the query be implemented?