Use Cases

for

Take Out Application

Version 1.0 approved

Prepared by Kevin Robles

Team One

10/6/2022

Revision History

Name	Date	Reason For Changes	Version
Dawson	10/9/2022	Update browse and order use case	1.0a

1. Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

2. Use Case Identification

2.1. Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.

2.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

2.3. Use Case History

2.3.1 Created By

Supply the name of the person who initially documented this use case.

2.3.2 Date Created

Enter the date on which the use case was initially documented.

2.3.3 Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

2.3.4 Date Last Updated

Enter the date on which the use case was most recently updated.

3. Use Case Definition

3.1. Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

3.2. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

3.3. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

3.4. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

3.5. Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.

3.6. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5.

3.7. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form “X.Y.E.Z”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

3.8. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

3.9. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

3.10. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

3.11. Business Rules

List any business rules that influence this use case.

3.12. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

3.13. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

3.14. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case List

Primary Actor	*Use Cases*
Customer	Browse and order food items
Kitchen staff	Update menu
Kitchen staff	Update kitchen stock
Management	Determine restaurant hours

Use Case Template

Use Case ID:	1		
Use Case Name:	Order Food		
Created By:	Kevin Robles	Last Updated By:	Dawson Sires
Date Created:	10/6/2022	Date Last Updated:	10/9/2022

Actors:	Customer (undecided, decided)
Description:	A customer enters the system and views the current available menu, selects items, and submits the order to the restaurant for preparation. Customers will visually receive confirmation and receive a notification for when the food is out on delivery.
Preconditions:	1. Customer enters the application and selects dine in or take out.
Postconditions:	1. Order is stored in the system for the kitchen to process. 2. Kitchen stock is updated.
Normal Flow:	1. Customer initializes the application. 2. System checks the time. 3. The menu for the day is displayed as well as predetermined food specials. 4. Customer determines which food items he or she would like to order. 5. Customer enters cart to review order. 6. Pre-payment information is displayed. 7. Customer reviews order. If correct, customer submits order as correct to kitchen.
Alternative Flows:	**1.1 Modify incorrect order** 1. Customer determines the cart doesn't contain the right items. 2. Customer enters modify UI to remove or edit food items. 3. After making changes, the customer confirms edits are correct. 4. The correct changes are reflected in the cart UI. 5. The customer can submit an order with reflected changes (step 7 in normal flow).
Exceptions:	**1.1 Current time is outside of operating hours** (step 2) 1. System informs the customer that the establishment is closed. 2. System terminates. **1.2 Food items are out of stock** (step 3) 1. Upon selecting out of stock item, the system informs the user that item is out of stock. 2. Menu is displayed for customer to select different items.
Includes:	Modify menu use case
Priority:	High.
Frequency of Use:	Hundreds of users, multiple times a day
Business Rules:	
Special Requirements:	1. Customer can cancel order. 2. Customer can edit order. 3. Customer receives receipt.
Assumptions:	1. Menus are updated beforehand 2. Stock information is updated according to kitchen inventory
Notes and Issues:	1. A special flag can be included to determine if the customer

	would like to dine in or receive delivery. If an establishment only delivers, dine in can be disabled.

Use Case ID:	2		
Use Case Name:	Update Menu		
Created By:	Dawson Sires	Last Updated By:	Dawson Sires
Date Created:	10/9/2022	Date Last Updated:	10/9/2022

Actors:	Kitchen Staff
Description:	Use case for creating the menu to include different food items.
Preconditions:	Kitchen account is logged in
Postconditions:	Menu is available for presentation to the user.
Normal Flow:	1. Kitchen account initiates menu creation process. 2. Kitchen selects a date to add food to. 3. Kitchen staff selects a food item. 4. Kitchen staff confirms that the food item should be added to the day. 5. Process repeats from step 2 until all food items are added. 6. Kitchen staff manually reviews the menu draft. 7. Kitchen staff confirms menu ok. 8. Menu is stored until customers access it when the day comes.
Alternative Flows:	**1.1 Menu draft is incorrect upon review (step 5).** 1. Kitchen staff determines menu is incorrect and enters menu modification. 2. Kitchen staff can add a food item or modify the existing one. 3. Kitchen staff reviews changes made. 4. Resume step 6 of normal flow. **1.2**
Exceptions:	**1.1 Staff selects out of stock food item (step 3).** **1.2 Staff makes multiple menus for one day (step 2-3).** **1.3 Staff adds food item out of stock.**
Includes:	Food Item Creation
Priority:	Critical
Frequency of Use:	Usually once a week or as often as needed (to update menu)
Business Rules:	
Special Requirements:	1. Staff can add menu item
Assumptions:	Kitchen account exists Manager determined a priori the days in which the restaurants are closed are marked as such in the system.
Notes and Issues:	

Use Case ID:	3		
Use Case Name:	View and update kitchen stock		
Created By:	Dawson	Last Updated By:	Dawson
Date Created:	10/11/2022	Date Last Updated:	10/11/2022

Actors:	Manager, kitchen staff
Description:	Restaurant personnel review and update stock information.
Preconditions:	Restaurant has received stock to be inventoried.

Postconditions:	Stock information is updated
Normal Flow:	1. Staff opens the menu of listed stock items at the end of the day. 2. Staff see that each list item is either listed in stock or out of stock 3. Staff has knowledge and can now resupply on low items.
Alternative Flows: (subject to change)	1. Customers opens the menu to see the food items 2. Once customer selects their item, or prior to selecting, there will be a notification that the food item is low. 3.
Exceptions:	None
Includes:	View Menu, Update Menu
Priority:	Medium
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	
Assumptions:	Manager and kitchen users exist in the system. The stock information that is verified to be correct actually reflects the stock.
Notes and Issues:	-Inventory is a state set to true or false, staff must be able to alter it -

Use Case ID:	4		
Use Case Name:	Employee Login		
Created By:	Dawson	Last Updated By:	
Date Created:	10/11/2022	Date Last Updated:	

Actors:	Kitchen Staff, Managers
Description:	Restaurant Employee Login Process
Preconditions:	Application has started
Postconditions:	Employee accesses respective pages (kitchen staff access kitchen page, managers access manager page)
Normal Flow:	1. Employee enters user id. 2. Employee enters password. 3. Employee submits credentials. 4. System verifies the credentials. 5. Employees are sent to the correct page.
Alternative Flows:	None
Exceptions:	**1.1 User credentials are invalid.** 1. User is notified invalid credentials were entered. 2. Jump to normal flow step 1.
Includes:	None
Priority:	Critical
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	None
Assumptions:	Employee accounts exist in the first place.
Notes and Issues:	A first “admin” account must be supplied for the system to bootstrap.

Use Case ID:	5		
Use Case Name:	Customer Create Account		
Created By:	Nick	Last Updated By:	
Date Created:	10/13/2022	Date Last Updated:	

Actors:	Customer
Description:	Customer Create Account
Preconditions:	Application has started
Postconditions:	Customer can create an account
Normal Flow:	1. Customer enters desired username 2. Customer enters desired password 3. Customer re-enters desired password 4. Customer clicks create account button 5. Customer is returned to the login page
Alternative Flows:	1. Customer clicks return to login page 2. Customer returns to login page
Exceptions:	**1.1 User credentials are invalid.** 1. Incorrect string length 2. Invalid characters 2. Error message
Includes:	None
Priority:	Critical
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	None
Assumptions:	None
Notes and Issues:	A first "admin" account must be supplied for the system to bootstrap.

Use Case ID:	6		
Use Case Name:	Manager Create Account		
Created By:	Nick	Last Updated By:	
Date Created:	10/13/2022	Date Last Updated:	

Actors:	Manager
Description:	Manager Create Account
Preconditions:	Is logged in as a manager
Postconditions:	Manager can create an employee account or a manager account
Normal Flow:	1. Manager can enter the username they would like their employee to use or manager account to use 2. Manager can enter a temporary password for the account. 3. Manager can re-enter the temporary password. 4. Manager selects whether the account will be a manager account or an employee account. 5. Manager selects create account 6. The fields reset and a message will read account reset.
Alternative Flows:	1. manager clicks return to menu 2. manager returns to menu
Exceptions:	**1.1 User credentials are invalid.**

	1. Incorrect string length 2. Invalid characters 2. Error message
Includes:	None
Priority:	Critical
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	None
Assumptions:	None
Notes and Issues:	A first “admin” account must be supplied for the system to bootstrap.

Use Case ID:	6		
Use Case Name:	The queue of customer orders		
Created By:	Nick	Last Updated By:	
Date Created:	10/13/2022	Date Last Updated:	

Actors:	Manager, Employee
Description:	The queue of customer orders
Preconditions:	Is logged in as employee or manager
Postconditions:	Employees can fulfill order requests
Normal Flow:	1. Orders come into the queue. 2. Orders are filled by employee and leave the queue.
Alternative Flows:	1. Customer clicks return to login page 2. Customer returns to login page
Exceptions:	**1.1 User credentials are invalid.** 1. Incorrect string length 2. Invalid characters 2. Error message
Includes:	None
Priority:	Critical
Frequency of Use:	Daily
Business Rules:	
Special Requirements:	None
Assumptions:	None
Notes and Issues:	A first “admin” account must be supplied for the system to bootstrap.

Classes:

Food Item Class:

Title:

- Food Item

Fields:

- Food name
- Food price
- Food description

Operations:

- Set/get price
- Set/get description
- Set/get name

Menu Class:

Title:

- Menu

Fields:

- Array_Of_Food_Item_Objects

Operations:

- Order (food item)
- Set food item
- Remove food item
- GetFoodItems
-

*Has food items

User Class:

*Use menu

Employee Interface:

Title:

- Employee Interface

Fields:

- Employee ID
- Employee name
- Employee Position

Operations:

-

*Use menu

*Is employee

Manager Class:

*** Implements Employee interface***

Title:

Fields:

Operations:

- Assign employee account

- Edit prices
- Edit menu

*Uses Menu
*Has Employee
*Is restaurant staff

Test plans:

Testing for order food:
Go through the order process. Press each button to make sure an order will go through.

Testing for update menu:
Press buttons to add items to the menu as a manager. Input strings. Input a very large string. Input nothing in input fields in the form.

Testing for view and update kitchen stock:
Add to stock. Add very large number. Add negative number add zero to stock. Make sure that the stock updates properly. Order food and make sure stock decrements by the order count.

Employee login and create account login:
Test login for shorter strings. Test login for longer strings in the password and username fields. Test login for empty strings. Test fields for uncommon characters.
For create account you will want to also test mismatching passwords.
Click each button.
Make sure passwords are hashed.
Make unique ID for each user account.