Baylor Burgers

By Francis Boyle, Patrick Boyle, Timmy Frederiksen, Johnny Acosta

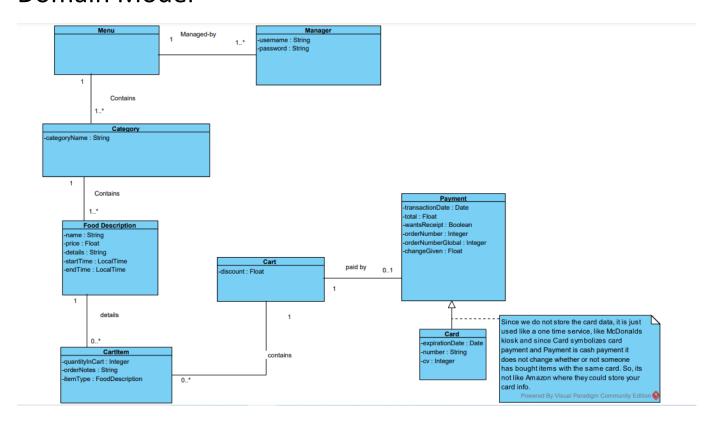
Project Vision

Our vision for Baylor Burgers is a self-order kiosk for restaurants. We plan to implement features such as allowing customers to add, delete and customize items for their order. We also plan to allow for ownership/managers to change the menu from the system. The system will also facilitate payment and allow customers to redeem deals. The system will also place orders to the restaurant so the employees can begin preparing the order as soon as possible. Our top priority is providing a simple, easy-to-understand user interface and allow for the quickest possible order time, streamlining the ordering process for the customer and the restaurant.

Gantt Diagram



Domain Model



List of Actors

Actors:

Customer – Orders food through system and should be able to add, delete, and customize items for their order. Also, should be able to pay for food and redeem special offers on the food.

Manager – Should be able to login to an unrestricted version of the system and add, delete, and edit items on the menu. Should be able to override certain functions and restart the system to fix issues.

Employee – Should be able to receive copies of customer orders to begin preparing the order as soon as possible.

List of Requirements

Requirements:

REQ-1	The system must keep a log of purchase transactions.
REQ-2	The system must handle a card payment and authenticate the transaction.
REQ-3	The manager must be able to update the inventory of items on the menu.
REQ-4	The customer must be able to contact the employee.
REQ-5	An employee needs to be able to be contacted.
REQ-6	The account information for each customer, employee, and manager must be
	verified.
REQ-7	The customer must be able to navigate/search menu and items.
REQ-8	The customer must be able to add/remove items from cart.
REQ-9	The customer must be able to pay for the items in cart.
REQ-10	The system must handle a cash payment.
REQ-11	The employee and manager must be able to make an account.
REQ-12	The manager must be able to add categories of items.
REQ-13	All employees and managers must be able to log into their accounts.
REQ-14	The system must hold a record of all employees and managers name,
	information, hours, and salary.
REQ-15	The customer and manager must be able to update personal information.

Traceability Matrix

		REQ-1	REQ-2	REQ-3	REQ-4	REQ-5
UC-1	Add Item to Cart					
UC-2	Remove Item from Cart					
UC-3	Search for Items				х	х
UC-4	Add Item to Menu			х		
UC-5	Edit Item to Menu			х		
UC-6	Remove Item in Menu			х		
UC-7	Pay for Items in Cart	х	х			
UC-8	Coupon/Rewards Discount					
UC-9	Manager Login					
UC-10	Display Menu					
UC-11	Get Help from Employee				х	х
UC-12	Manager Sign-up					

		REQ-						
		6	7	8	9	10	11	12
UC-1	Add Item to Cart	x		х				
UC-2	Remove Item from Cart			х				
UC-3	Search for Items	х	х					
UC-4	Add Item to Menu	х						х
UC-5	Edit Item to Menu	х						х
UC-6	Remove Item in Menu	х						
UC-7	Pay for Items in Cart	х			х	х		
UC-8	Coupon/Rewards Discount							
UC-9	Manager Login						х	
UC-10	Display Menu							
UC-11	Get Help from Employee							
UC-12	Manager Sign-up						х	

		REQ-13	REQ-14	REQ-15
UC-1	Add Item to Cart			
UC-2	Remove Item from Cart			
UC-3	Search for Items			
UC-4	Add Item to Menu			
UC-5	Edit Item to Menu			
UC-6	Remove Item in Menu			
UC-7	Pay for Items in Cart			
UC-8	Coupon/Rewards Discount			
UC-9	Manager Login	х	Х	х
UC-10	Display Menu			
UC-11	Get Help from Employee			
UC-12	Manager Sign-up		х	х

Use Case Diagrams and System Sequence Diagrams

ID: UC Add Item to Cart

Level: User Goal **Scope:** Kiosk system

Actors (Stakeholders) and interests:

Customer

Person interested in ordering food.

Employee

- Person responsible for preparing food.

Preconditions:

Customer is using the application.

Customer found an item to add to cart.

Postconditions:

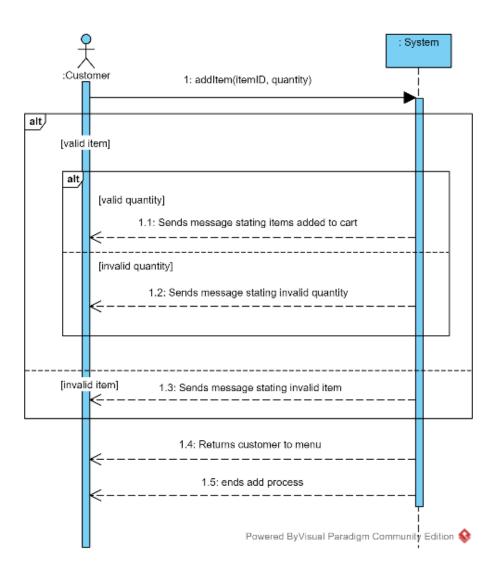
The item with given quantity is added to the cart.

The Customer returns to the page where he/she found the item.

Main success scenario:

- 1. Customer clicks to add item to cart.
- 2. Customer specifies the quantity of the item to add to cart.
- 3. The system verifies that the items can be prepared.
 - 3.1 Ingredients for the item are available.
- 4. The system sends message to restaurant employee.

- 3.a. If item(s) cannot be prepared.
 - 1. The maximum quantity available is displayed to the screen.
 - 2. The Customer returns to the page where he/she found the item.



ID: UC Remove Item from Cart

Level: User Goal **Scope:** Kiosk system

Actors (Stakeholders) and interests:

Customer

Person interested in ordering food.

Employee

Person responsible for preparing food.

Preconditions:

Customer found an item to remove from cart, on shopping cart page.

Postconditions:

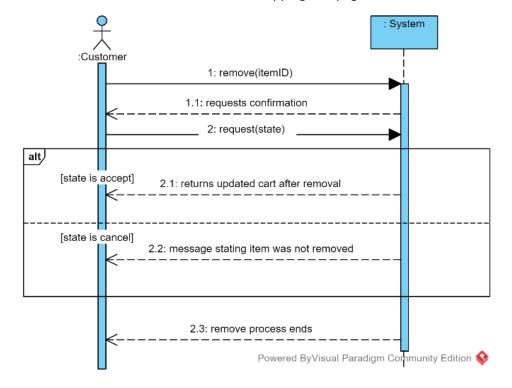
The item is removed from the cart.

The Customer stays on shopping cart page.

Main success scenario:

- 1. Customer clicks to remove item from cart.
- 2. The system asks for the quantity to be removed.
 - 2.1 Only options given to Customer are valid options. (i.e., not -1)
- 3. Customer selects the quantity he/she wants removed of the specific item.
- 4. The system removes the quantity/item and updates the information to screen.
- 5. The system sends message to update the employee.

- 3.a. If customer no longer wants to remove items.
 - 1. The customer may cancel the removal of the items.
 - 2. The customer is returned to the shopping cart page.



ID: UC Search for Item

Level: User Goal **Scope:** Kiosk system

Actors (Stakeholders) and interests:

Customer

Person interested in ordering food.

Employee

- Person responsible for helping Customers.

Preconditions:

Customer is using the application.

Postconditions:

Customer has found the item they were looking for.

Main success scenario:

- 1. Customer locates the navigation bar.
- 2. Customer chooses which menu category their item best fits.
- 3. Customer looks through the list of the category and locates the item.

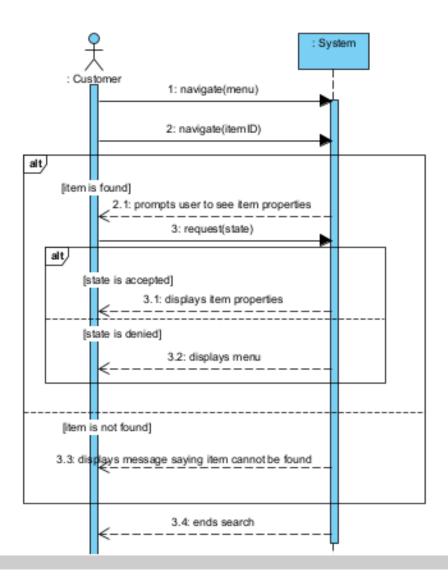
Extensions:

- 3.a. If item(s) is not in menu.
 - 1. No item will appear.
 - 2. Customer contacts employee for help.

Alternative success scenario:

- 1. Customer locates search bar.
- 2. Customer types in name of item they are looking for.
- 3. Customer selects item from search bar.

- 3.a. If item(s) is not in menu.
 - 1. No item will appear.
 - 2. Customer contacts employee for help.



UC Login - Timmy Frederiksen

ID UC Login

Scope Self-order kiosk

Level User-Goal

Stakeholders and interests

Manager

 This person should be able to login to the system to perform certain restricted actions and to override the normal system behavior.

Preconditions

· The system should have certain manager login to recognize on login attempt

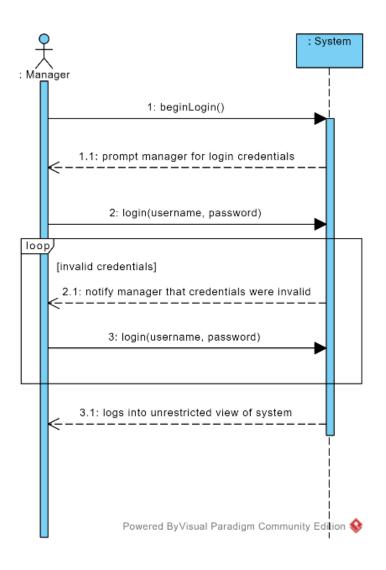
Postconditions

 The manager should be logged into the system and restricted action should now be unlocked

Main Success Scenario

- 1. Manager selects 'Manager Login' button
- 2. The system should then prompt the manager for credentials
- 3. The manager enters the credentials
- 4. The system should now unlock restricted actions and the manager is logged in

- *.a. At any point, the system may fail.
- 3.a. If incorrect credentials are entered, the system should then prompt the manager to try again.
- 3.a*. If too many incorrect attempts are made in a short period of time, the system should alert a manager, thus alerting them to a customer potentially trying to enter as a manager.



UC Pay - Timmy Frederiksen

ID UC Pay

Scope Self-order kiosk

Level User Goal

Stakeholders and interests

Customer

 The customer who wants to place an order, and will now pay for said order Manager

 The person who is able to help facilitate the payment process and can fix any issues with the system, as necessary.

Employee

 The employees should be able to see orders that have been placed in order to begin preparation of the order

Preconditions

- . The system has been used to add the customer's desired items into their order.
- · The customer wants to pay and place their order

Postconditions

- · The payment has been made and the order has thus been placed.
- · The customer should have their order number and a receipt (if requested)

Main Success Scenario

- 1. Customer selects 'pay' button on the screen
- 2. System should then prompt the user for payment method
- 3. Customer determines method of payment
 - a. If the customer opts for card payment, the system should prompt them to insert card
 - If the user selects cash or check, the system should notify the manager to facilitate physical payment
- 4. The system should confirm payment has been received
- 5. The system prompts the customer whether they would like a receipt
 - a. If the user wants a receipt, the system should send it. Otherwise, do nothing
- The system should place the order so that the employees can see it and begin preparing the order

- 7. The system should give the user an order number
- The transaction has been completed, and the customer can retrieve their order once it has been prepared

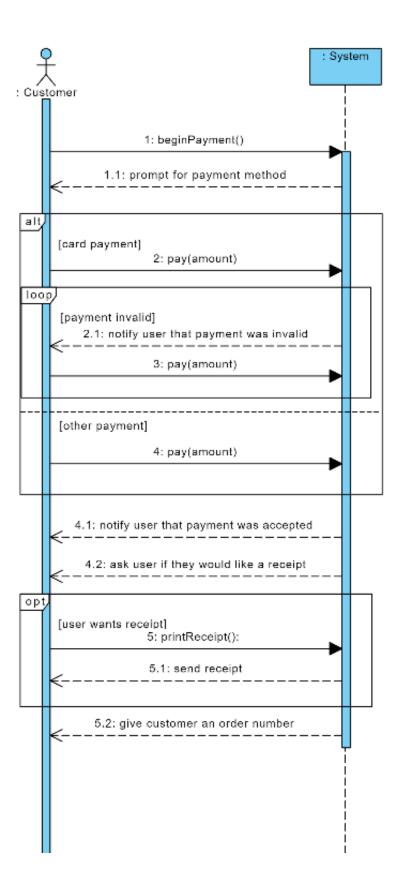
Extensions

- *.a. At any point, the customer may select the 'request help from manager' button, and the manager should be notified of the request.
- *.b. At any point, the system may fail. If this occurs, a manager should be notified by the customer.

3a.a. If the card payment is invalid, the system will notify the user of the failure and prompt again.

3b.a If the user has another payment method not specified here, proceed by step 3b, and the manager should determine whether said payment method is valid.

4.a If the user did not pay with a card and the manager facilitated payment, the manager should enter the manager code into the system. This will tell the system that physical payment has been received.



UC Redeem Offer - Timmy Frederiksen

ID UC Redeem Offer

Scope Self-order kiosk

Level User Goal

Stakeholders and interests

Customer

- The customer who wants to redeem an offer to apply to their order Manager
 - The person who is able to help facilitate the offer redemption process and can fix any issues with the system, as necessary.

Preconditions

- The system has been used to add the customer's desired items into their order.
- . The customer has an offer to be redeemed that they wish to apply to their order.

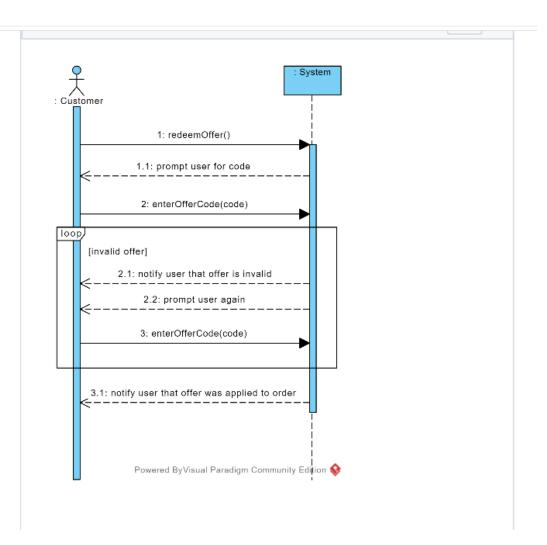
Postconditions

- · The offer has been redeemed
- . The customer may now continue to make their order

Main Success Scenario

- 1. The customer selects 'Redeem Offer' button on the system
- 2. The system prompts the user for a code for the offer
- 3. The user enters their code for the offer
- The system then applies the offer to the customer's order and returns the customer to the order menu to continue ordering or to checkout.

- *.a. At any point, the customer may select the 'request help from manager' button, and the manager should be notified of the request.
- *.b. At any point, the system may fail. If this occurs, a manager should be notified by the
- 3a.a. If the offer is invalid, the system will notify the user of the failure and prompt again.



ID UC Add an Item to Menu

Scope Menu Item System

Level Manager requirements

Stakeholders and Interest

Manager

- person that is interested in adding a new item to menu

Precondition:

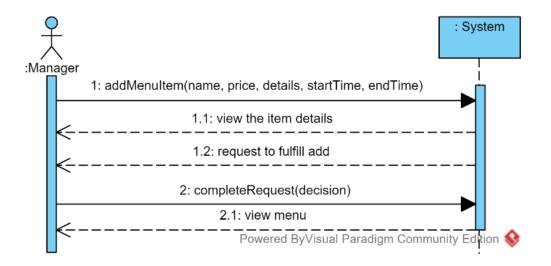
- 1. Manager is logged in
- 2. Manager is on menu view page

Postcondition: New item is added to menu

Main success scenario:

- 5. Manager clicks to add item to menu.
- 6. System asks for product specification attributes
- 7. Manager enters in product specification attributes.
- 8. System displays the new menu page

- 3.a* If manager navigates away without finishing
- 1. stop and do not preform specified action
- 3.b* Manager wishes to not add an item
- 1. do not add and return to view menu page



ID UC Edit an Item in Menu

Scope Menu Item System

Level manager capabilities

Stakeholders and Interest

Manager

-person that is interested in editing item in menu

Precondition:

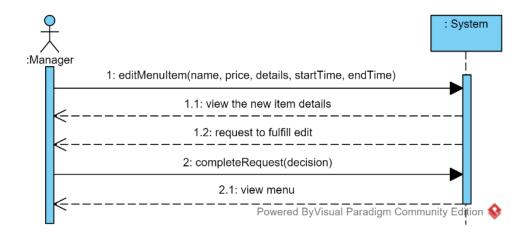
- 1. Manager is logged in
- 2. Manager is on menu view page

Postcondition: Item specifications are edited

Main success scenario:

- 1. Manager clicks the Edit button on item to be edited
- 2. System shows attributes of item specification
- 3. Manager edits the desired specification attributes
- 4. System asks for confirmation to edit
- 5. Manager confirms
- 6. System displays the new menu page

- a.* If manager navigates without finishing
- 1. stop and do not preform specified action
- 3.a* Manager wishes to not edit attribute
- 1. leave the attribute alone
- 3.b* Manager wishes to not edit the item
- 1. do not edit and return to view menu page
- 5.a Manager wishes to not confirm edit
- 1. do not edit the item



ID UC Remove an Item from Menu

Scope Menu Item System

Level manager capabilities

Stakeholders and Interest

Manager

-person that is interested in removing a new item from menu

Precondition:

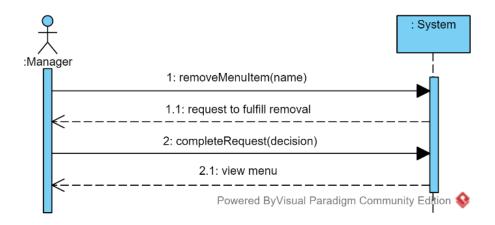
- 3. Manager is logged in
- 4. Manager is on menu view page

Postcondition: Item is removed from menu

Main success scenario:

- 7. Manager clicks the Delete button on item that wishes to be deleted
- 8. System prompts manager about her decision
- 9. Manager confirms
- 10. System displays the new menu page

- 2.a* If manager navigates away without finishing
- 1. stop and do not preform specified action
- 3.a* Manager wishes to keep the item
- 1. do not remove and return to view menu page



ID UC Display

Scope Self-Order Kiosk

Level User Goal

Stakeholders and interests

Customer

• Wants to see items on the menu

Preconditions

• Customer is at the kiosk

Postconditions

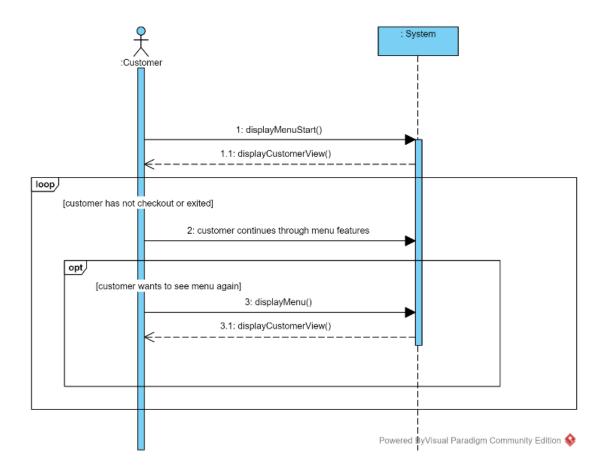
• Customer can view all items on the menu

Main Success Scenarios

- 1. Customer selects Start
- 2. The system gathers each food on each gather to put into a menu screen
- 3. The system frames all the information in an orderly fashion
- 4. The system displays the menu for the customer
- 5. The customer is able to browse the selection of items by category
- 6. If the customer wants to return to the main menu at any point
 - a. The customer presses the main menu button
 - b. The system repeats steps 2-5

Extensions

*.a. At any point the customer can leave the kiosk.



ID UC Call Help

Scope Self-Order Kiosk

Level User Goal

Stakeholders and interests

Customer

• Needs assistance while navigating through the kiosk

Manager

Has resources and knowledge to make the customer's experience satisfactory

Preconditions

Customer is at the kiosk

Postconditions

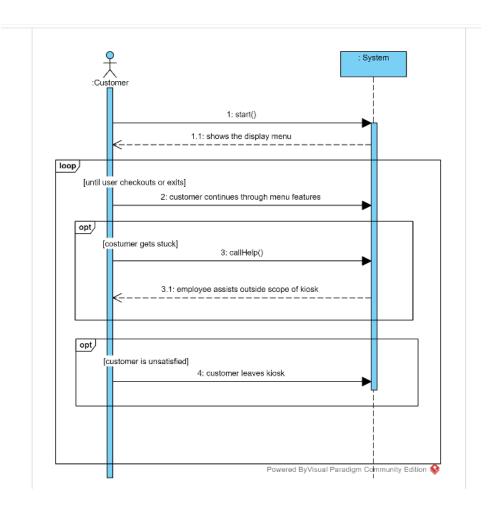
Customer has received help

Main Success Scenarios

- 1. Customer starts the kiosk
- 2. The system starts up
- 3. Customer proceeds to order through the system
- 4. The customer runs into a problem
- 5. The customer presses the help button
- 6. The system alerts the manger or employee and displays a message
- 7. Manager comes and assists the customer
- 8. If the manager is unable to help the customer at the kiosk
 - a. The manager helps the customer manually, out of scope of the system
- 9. The customer has received help
- 10. If the customer needs help again
 - a. Repeat steps 5-9

Extensions

*.a. At any point the customer can leave the kiosk.



ID UC Manager Sign Up

Scope Self-Order Kiosk

Level User Goal

Stakeholders and interests

Manager

• Needs to register themselves into the system

Preconditions

Manager has the necessary credentials

Postconditions

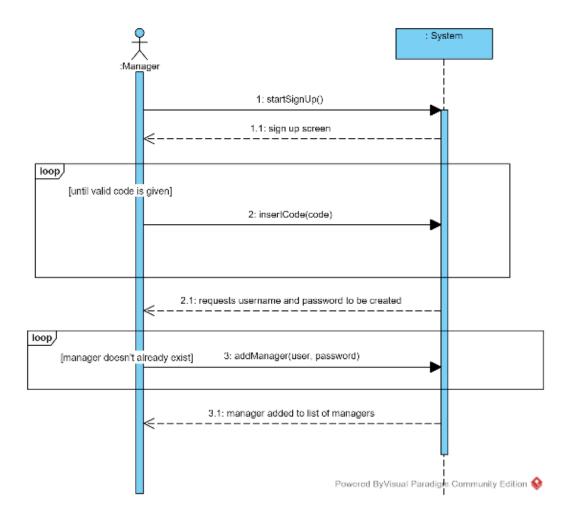
Manager has successfully registered themselves into the system

Main Success Scenarios

- 1. Manager clicks register option
- 2. The system displays a register screen
- 3. The system asks the user for a manager code
- 4. Manager inputs a code
- 5. If the system detects an incorrect code

- a. Repeats steps 3-4 until a correct code is entered
- 6. The system asks the user to input a username and password
- 7. Manager inputs their information
- 8. If the system detects duplicate information in the system
 - a. Repeat steps 6-7
- 9. The system stores the managers' information to a database
- 10. The manager is now registered in the system

- *.a. At any point the customer can leave the kiosk.
- 2.a The manager is unable to log in and leaves the system



Operation Contracts

Name: addItem(itemID, quantity)

Cross-references: UC Add Item to Cart

Precondition: The customer has an item that they want to add to cart.

Postcondition: An item or items (depending on quantity) is added to cart.

Name: removeItem(itemID, quantity)

Cross-references: UC Remove Item from Cart

Precondition: The customer has an item that they want to remove from their cart.

Postcondition: The item is removed from the cart.

Name: navigate(page)

Cross-references: UC Search for an Item

Precondition: The customer is on a page, and the customer wants to visit another page.

Postcondition: The customer is now at another page, the page chosen by the customer.

Name: callHelp(message)

Cross-references: UC Search for an Item and UC Call Help

Precondition: The customer needs help on their order.

Postcondition: An employee is notified that a customer needs help.

Name: addMenuItem(name, price, details, startTime, endTime)

Cross References: AddItemToMenu

Preconditions:

1. The manager is logged in and on the menu page

2. The manager has an item that needs to be added to the menu

Postconditions: The menu now contains the new item

Name: editMenuItem(name, price, details, startTime, endTime)

Cross References: RemoveMenuItem

Preconditions:

1. The manager is logged in and on the menu page

2. An item exists in the menu that the manager would like to edit

Postconditions: The item is updated with the new attributes

Name: removeMenuItem(name)

Cross References: RemoveMenuItem

Preconditions:

1. The manager is logged in and on the menu page

2. An item exists in the menu that the manager would like to delete

Postconditions: The menu no longer has that item available

Name: completeRequest(decision)

Cross References: RemoveMenuItem and EditMenuItem

Preconditions: The manager has made a decision that will make a major change so it must be

verified

Postconditions: The decision was verified with the manager

Name: start()

Cross-references: UC Display Menu

Precondition: The customer is at the kiosk

Postcondition: The customer is able to see the kiosk starting

Name: displayMenu()

Cross-references: UC Display Menu

Precondition: The customer is at the start screen

Postcondition: The customer is able to view the menu selection

Name: editItem()

Cross-references: UC Manager Sign-In

Precondition: The manager is on admin view of the menu

Postcondition: The manger changes the menu accordingly

Operation Name: redeemOffer()

Cross-References: UC Redeem Offer

Pre-Conditions: The user has a code to redeem a specific offer.

Post-Conditions: The system has prompted the user to enter the offer code.

Operation Name: enterOfferCode(code)

Cross-References: UC Redeem Offer

Pre-Conditions: The system has prompted the user to enter the offer code.

Post-Conditions: The user has entered the code and the system has applied the offer to the

order.

Operation Name: beginPayment()

Cross-References: UC Pay

Pre-Conditions: The user wants to pay for their order.

Post-Conditions: The payment process has begun, and the system is prompting the user for

payment info.

Operation Name: pay(amount)

Cross-References: UC Pay

Pre-Conditions: The user has entered payment information into the system.

Post-Conditions: The system has either authorized the payment or the payment is invalid, and

the user has been re-prompted for payment.

Operation Name: beginLogin()

Cross-References: UC Manager Login

Pre-Conditions: The manager wants to login.

Post-Conditions: The login process has begun, and the manager has been prompted for

credentials.

Operation Name: login(username, password)

Cross-References: UC Manager Login

Pre-Conditions: The manager has entered a username and password into the system.

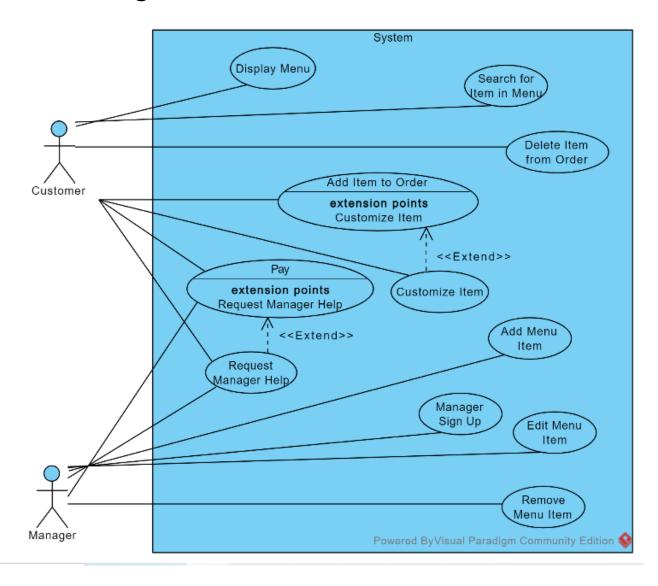
Post-Conditions: The manager has either been logged into the system successfully or the

credentials are invalid, and the manager is then re-prompted for login credentials.

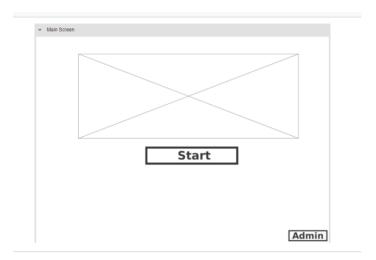
System Operations

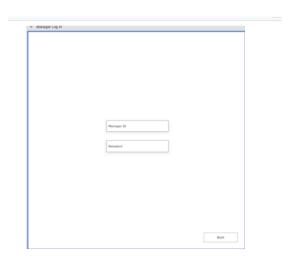
```
System
+navigate(page)
+message(help)
+remove(itemID, quantity)
+addItem(itemID, quantity)
+beginLogin()
+login(username, password)
+redeemOffer()
+enterOfferCode()
+beginPayment()
+pay(amount)
+start()
+callHelp()
+displayMenu()
+insertInfo(info)
+editItem()
+removeMenuItem(name)
+addMenuItem(name, price, details, startTime, endTime)
+editMenuItem(name, price, details, startTime, endTime)
+completeRequest(decision)
                             Powered ByVisual Paradigm Community Edition
```

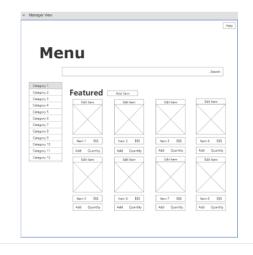
Use Case Diagram

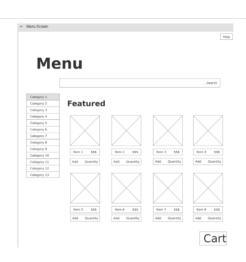


Wireframes











Timecards Report

Francis Boyle – 12½ Hours

- Attended Team Meetings
- Use Cases, System Sequence Diagrams, and Operation Contracts
 - Add Item to Cart
 - Remove Item from Cart
 - Search for Item
- Requirements (Functional and Non-Functional)
- Traceability Matrix
- Assembled PDF

Patrick Boyle – 11 Hours

- Attended Team Meetings
- Use Cases, System Sequence Diagrams, and Operation Contracts
 - o Add Item to Menu
 - o Remove Item from Menu
 - o Edit Item from Menu
- Domain Model

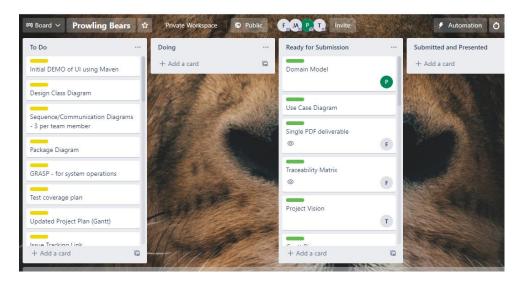
Timmy Frederiksen – 9 hours

- Attended Team Meetings
- Website, GitHub, Tello Links
- Use Case Diagram
- Use Cases, System Sequence Diagrams, and Operation Contracts
 - Manager Login
 - o Redeem Offer
 - o Pay

Johnny Acosta – 10 Hours

- Attended Team Meetings
- Use Cases, System Sequence Diagrams, and Operation Contracts
 - Manager Sign-up
 - o Call Employee for Help
 - o Display Menu
- Wireframes

Issue Tracking



Issues Resolved: 19

Trello Board Linked Below:

Trello: https://trello.com/b/xjHC0LGq/prowling-bears

Website, Trello, GitHub Links

Website: https://csi3471-kiosk-project.weebly.com

Trello: https://trello.com/b/xjHC0LGq/prowling-bears

GitHub: https://github.com/timmyFrederiksen/GroupProjectFall2021

Trello History

9/29/21, 8:00 PM Francis_Boyle1 | Trello

Francis_Boyle1 added Patrick_Boyle1 to Fix Domain Model 3 minutes ago

Francis_Boyle1 added Timmy_Frederiksen1 to Fix SSDs an hour ago

Francis_Boyle1 added Patrick_Boyle1 to Fix SSDs an hour ago

Francis_Boyle1 added Jonathan Acosta to Fix SSDs an hour ago

F Francis_Boyle1 joined Fix SSDs an hour ago

Francis_Boyle1 added Timmy_Frederiksen1 to <u>Use Case Diagram</u>

- Francis_Boyle1 moved Fix SSDs from Doing to Ready for Submission an hour ago
- Francis_Boyle1 moved Fix Domain Model from Doing to Ready for Submission an hour ago
- F Francis_Boyle1 added Timmy_Frederiksen1 to <u>Timecards Report</u> an hour ago
- Francis_Boyle1 joined <u>Timecards Report</u> an hour ago
- F Francis_Boyle1 left Project Vision an hour ago
- Francis_Boyle1 added Timmy_Frederiksen1 to Actors an hour ago
- Francis_Boyle1 joined <u>Single PDF deliverable</u> an hour ago
- Francis_Boyle1 moved <u>Single PDF deliverable</u> from Doing to Ready for Submission an hour ago
- Francis_Boyle1 moved Operations & Contracts from Doing to Ready for Submission an hour ago
- Francis_Boyle1 added Jonathan Acosta to Operations & Contracts an hour ago

Francis_Boyle1 joined Operations & Contracts

an hour ago

Francis_Boyle1 added Patrick_Boyle1 to Operations & Contracts

an hour ago

Francis_Boyle1 moved Website from Doing to Ready for Submission

Francis_Boyle1 joined Project Vision

yesterday at 1:13 PM

Francis_Boyle1 left Project Vision

yesterday at 1:13 PM

Francis_Boyle1 added Timmy_Frederiksen1 to Gantt Diagram

Sep 27 at 8:20 PM

Francis_Boyle1 added Patrick_Boyle1 to Gantt Diagram

Sep 27 at 8:20 PM

Francis_Boyle1 added Jonathan Acosta to Gantt Diagram

Sep 27 at 8:20 PM

Francis_Boyle1 added Timmy_Frederiksen1 to Project Vision

Sep 27 at 8:20 PM

Francis_Boyle1 joined Gantt Diagram

Sep 27 at 8:06 PM

Francis_Boyle1 joined Project Vision

Sep 27 at 8:06 PM

Francis_Boyle1 moved Traceability Matrix from Doing to Ready for Submission

Sep 27 at 8:05 PM

Francis_Boyle1 moved Requirements from Doing to Ready for Submission

Sep 27 at 8:05 PM

Francis_Boyle1 joined Requirements

Sep 27 at 6:01 PM

Francis_Boyle1 joined Traceability Matrix

Sep 27 at 6:01 PM

Francis_Boyle1 moved Use Cases and SSDs - Francis Boyle from Doing to Ready for Submission

Sep 27 at 4:56 PM

Francis_Boyle1 joined Use Cases and SSDs - Francis Boyle

Sep 26 at 6:39 PM

9/29/21, 8	00 PM Francis_Boyle1 Trello
F	Francis_Boyle1 left <u>User Interface Wireframes</u> Sep 26 at 6:39 PM
F	Francis_Boyle1 joined <u>User Interface Wireframes</u> Sep 26 at 12:47 PM
F	Francis_Boyle1 joined Fully Dressed Use Cases Sep 26 at 12:46 PM
F	Francis_Boyle1 moved Single PDF deliverable from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>Suggested Point distribution</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>Timecards Report</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved Actors from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>Domain Model</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved Gantt Diagram from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>User Interface Wireframes</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>SSDs</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved <u>Traceability Matrix</u> from To Do to Doing Sep 25 at 8:16 PM
F	Francis_Boyle1 moved Requirements from Ready for Submission to Doing Sep 25 at 8:16 PM
	Load more activity.

More Activity on Trello (Links Above)

≔ Jonathan Acosta (jonathanacosta51)

- Francis_Boyle1 added Jonathan Acosta to Fix SSDs
- F Francis_Boyle1 added Jonathan Acosta to Operations & Contracts an hour ago
- F Francis_Boyle1 added Jonathan Acosta to Gantt Diagram
 Sep 27 at 8:20 PM
- JA Jonathan Acosta moved <u>User Interface Wireframes</u> from Doing to Ready for Submission Sep 27 at 6:49 PM
- JA Jonathan Acosta moved <u>Use Cases and SSDs Johnny Acosta</u> from Doing to Ready for Submission Sep 27 at 6:49 PM
- Timmy_Frederiksen1 added Jonathan Acosta to <u>Use Cases and SSDs Johnny Acosta</u> Sep 26 at 6:56 PM
- T Timmy_Frederiksen1 added Jonathan Acosta to <u>User Interface Wireframes</u> Sep 26 at 6:51 PM

- F Francis_Boyle1 added Timmy_Frederiksen1 to Fix SSDs an hour ago
- F Francis_Boyle1 added Timmy_Frederiksen1 to <u>Use Case Diagram</u> an hour ago
- T Timmy_Frederiksen1 moved <u>Use Case Diagram</u> from Doing to Ready for Submission an hour ago
- F Francis_Boyle1 added Timmy_Frederiksen1 to Timecards Report an hour ago
- F Francis_Boyle1 added Timmy_Frederiksen1 to Actors an hour ago
- T Timmy_Frederiksen1 added <u>Fix Domain Model</u> to Doing 7 hours ago
- T Timmy_Frederiksen1 joined Operations & Contracts
 7 hours ago
- Timmy_Frederiksen1 archived Operations & Contracts Francis 7 hours ago
- T Timmy_Frederiksen1 archived Operation & Contracts Patrick
 7 hours ago
- T Timmy_Frederiksen1 archived Operations & Contracts Johnny 7 hours ago
- Timmy_Frederiksen1 added Operations & Contracts Johnny to Doing 7 hours ago
- Timmy_Frederiksen1 added Operation & Contracts Patrick to Doing 7 hours ago
- T Timmy_Frederiksen1 added Operations & Contracts Francis to Doing 7 hours ago
- T Timmy_Frederiksen1 added Operations & Contracts to Doing 7 hours ago
- T Timmy_Frederiksen1 added Fix SSDs to Doing 7 hours ago
- T Timmy_Frederiksen1 added <u>Use Case Diagram</u> to Doing 7 hours ago

Timmy_Frederiksen1 made <u>Prowling Bears</u> visible to the public Sep 27 at 8:21 PM

Francis_Boyle1 added Timmy_Frederiksen1 to Gantt Diagram

Sep 27 at 8:20 PM

Francis_Boyle1 added Timmy_Frederiksen1 to Project Vision

Sep 27 at 8:20 PM

Timmy_Frederiksen1 moved <u>Suggested Point distribution</u> from Doing to Ready for Submission Sep 27 at 8:06 PM

Timmy_Frederiksen1 moved Timecards Report from Doing to Ready for Submission

Sep 27 at 8:06 PM

Timmy_Frederiksen1 moved Website from Ready for Submission to Doing

Sep 27 at 5:42 PM

Timmy_Frederiksen1 archived <u>Documentation (Gantt Diagram)</u>

Sep 27 at 5:32 PM

Timmy_Frederiksen1 moved <u>Use Cases and SSD - Timmy Frederiksen</u> from Doing to Ready for Submission Sep 27 at 5:32 PM

Timmy_Frederiksen1 moved Git Link from Doing to Ready for Submission

Sep 27 at 5:32 PM

Timmy_Frederiksen1 moved Website from Doing to Ready for Submission

Sep 27 at 5:32 PM

Timmy_Frederiksen1 added Jonathan Acosta to Use Cases and SSDs - Johnny Acosta

Sep 26 at 6:56 PM

Timmy_Frederiksen1 added Jonathan Acosta to User Interface Wireframes

Sep 26 at 6:51 PM

Timmy_Frederiksen1 joined User Interface Wireframes

Sep 26 at 6:51 PM

Timmy_Frederiksen1 joined Website

Sep 26 at 6:51 PM

Timmy_Frederiksen1 moved Gantt Diagram from Doing to Ready for Submission

Sep 26 at 6:42 PM

Timmy_Frederiksen1 joined Use Cases and SSD - Timmy Frederiksen

Sep 26 at 6:33 PM

Timmy_Frederiksen1 joined Git Link

Sep 26 at 6:33 PM

9/29/21, 8:12 PM

Timmy_Frederiksen1 | Trello

Timmy_Frederiksen1 archived <u>Fully Dressed Use Cases</u> Sep 26 at 6:14 PM

Timmy_Frederiksen1 archived SSDs

Sep 26 at 6:14 PM

Timmy_Frederiksen1 added <u>Use Cases and SSDs - Francis Boyle</u> to Doing Sep 26 at 6:14 PM

Timmy_Frederiksen1 added <u>Use Cases and SSDs - Patrick Boyle</u> to Doing Sep 26 at 6:13 PM

Timmy_Frederiksen1 added <u>Use Cases and SSDs - Johnny Acosta</u> to Doing Sep 26 at 6:13 PM

T Timmy_Frederiksen1 added <u>Use Cases and SSD - Timmy Frederiksen</u> to Doing Sep 26 at 6:13 PM

T Timmy_Frederiksen1 made Patrick_Boyle1 an admin of Group Project SE fall2021.
Aug 26 at 2:06 PM

Timmy_Frederiksen1 made Jonathan Acosta an admin of Group Project SE fall2021

Aug 26 at 2:06 PM

T Timmy_Frederiksen1 made Francis_Boyle1 an admin of Group Project SE fall2021 Aug 26 at 2:06 PM

Timmy_Frederiksen1 added list Ready for Submission to Group Project SE fall2021.
Aug 26 at 2:02 PM

Timmy_Frederiksen1 added timecards report to To Do Aug 26 at 2:01 PM

Timmy_Frederiksen1 deleted card #35 from To Do

Aug 26 at 2:00 PM

Timmy_Frederiksen1 added code quality matters (remember comments/formatting/coupling/cohesion) to To Do

Aug 26 at 2:00 PM

Timmy_Frederiksen1 added <u>Single PDF deliverable for Iteration 3</u> to To Do Aug 26 at 1:59 PM

T Timmy_Frederiksen1 added Revised Iteration 2 (separate PDF) to To Do Aug 26 at 1:59 PM

T Timmy_Frederiksen1 added <u>Code documentation (JavaDoc)</u> to To Do Aug 26 at 1:58 PM

Timmy_Frederiksen1 added <u>Single PDF deliverable (excluding revised iteration 1)</u> to To Do Aug 26 at 1:55 PM

https://trello.com/b/xjHC0LGq/prowling-bears/member/timmy_frederiksen1

9/29/21, 8:12 PM

Timmy_Frederiksen1 | Trello

- T Timmy_Frederiksen1 added <u>Timecards Report</u> to To Do Aug 26 at 155 PM
- Timmy_Frederiksen1 added <u>Suggested point distribution</u> to To Do Aug 26 at 1:55 PM
- T Timmy Frederiksen1 added Revised Iteration 1 (separate PDF) to To Do
 Aug 26 at 155 PM
- Timmy_Frederiksen1 added Git Link to To Do Aug 26 at 1:55 PM
- Timmy_Frederiksen1 added <u>Issue Tracking Link</u> to To Do Aug 26 at 1:55 PM
- T Timmy_Frederiksen1 added <u>Single PDF deliverable</u> to To Do Aug 26 at 1:49 PM
- T Timmy_Frederiksen1 added <u>Suggested Point distribution</u> to To Do Aug 26 at 1:49 PM
- T Timmy_Frederiksen1 added Jonathan Acosta to Group Project SE fall2021
 Aug 26 at 1:45 PM

Load more activity

Patrick_Boyle1 (patrick_boyle1)

- Francis_Boyle1 added Patrick_Boyle1 to Fix Domain Model 17 minutes ago
- Patrick_Boyle1 joined <u>Timecards Report</u> an hour ago
- Patrick_Boyle1 joined Actors an hour ago
- F Francis_Boyle1 added Patrick_Boyle1 to Fix SSDs an hourago
- F Francis_Boyle1 added Patrick_Boyle1 to Operations & Contracts an hour ago
- Patrick_Boyle1 moved <u>Actors</u> from Doing to Ready for Submission Sep 27 at 8:21 PM
- F Francis_Boyle1 added Patrick_Boyle1 to Gantt Diagram
 Sep 27 at 8:20 PM
- Patrick_Boyle1 moved <u>Use Cases and SSDs Patrick Boyle</u> from Doing to Ready for Submission Sep 27 at 8:06 PM
- Patrick_Boyle1 moved <u>Domain Model</u> from Doing to Ready for Submission Sep 27 at 8:03 PM
- Patrick_Boyle1 left <u>Traceability Matrix</u> Sep 26 at 6:51 PM
- Patrick_Boyle1 joined <u>Domain Model</u> Sep 26 at 6:47 PM
- Patrick_Boyle1 joined <u>Traceability Matrix</u> Sep 26 at 6:45 PM
- Patrick_Boyle1 joined <u>Use Cases and SSDs Patrick Boyle</u> Sep 26 at 6:43 PM