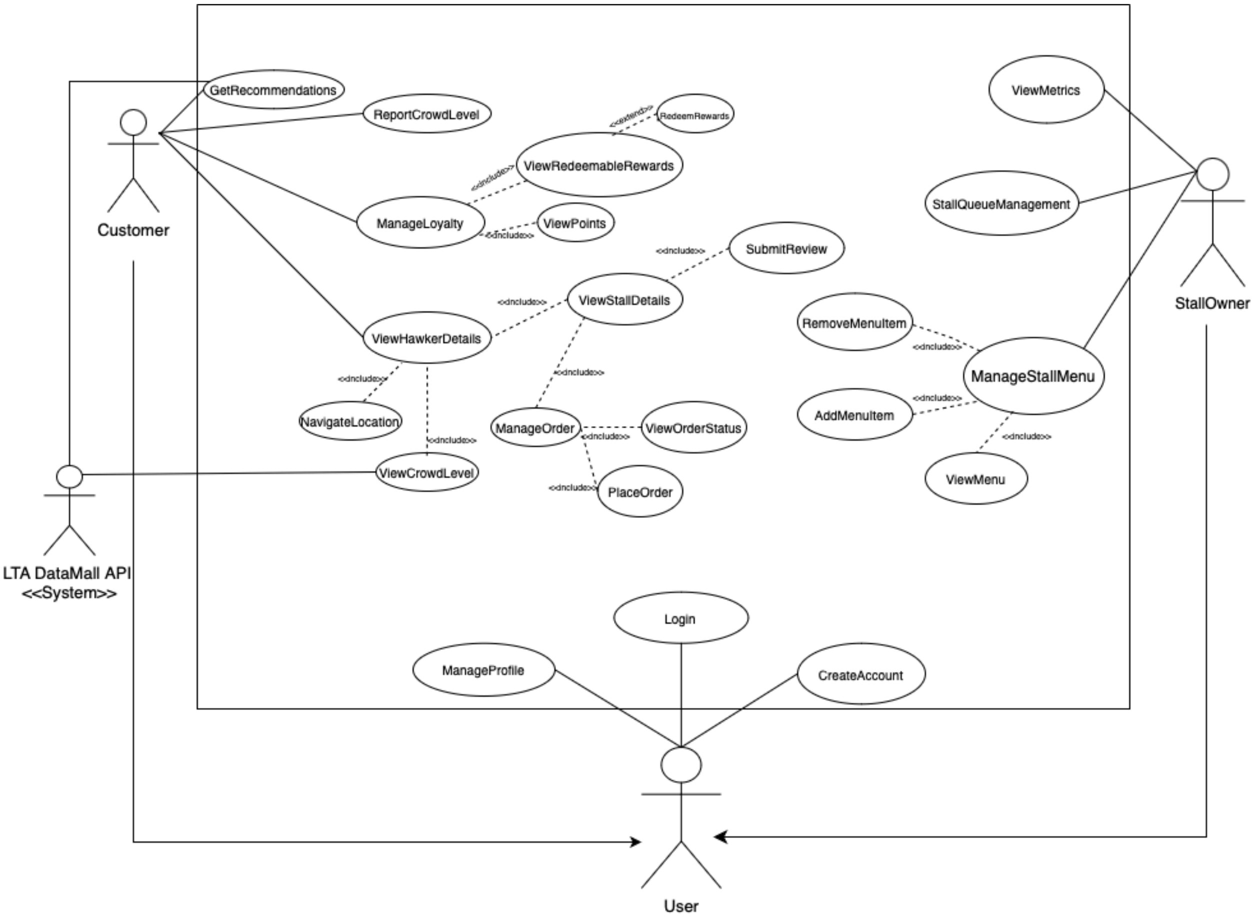
Table of Contents

Use Case Diagram:



Functional Requirement #1

1.0 Authorization

1.0.1 Login

Actor	User
Description	Allows the User to login to his/her HawkerGo account with their email and password
Preconditions	None
Postconditions	The User is logged into the HawkerGo application and is navigated to the application's home screen.
Priority	High
Frequency of Use	High
Flow of events	1. The system allows the User to login with email and password 2. The User chooses to log in with email and password 3. The User enters his/her email and password. The password is hidden as dots, but the User can choose to see it by selecting the eye icon 4. The User selects the "Login" button
Alternative flows	AF-S2 Social log-in: 1. Instead of entering email and password, the user selects a social login option (Google, Facebook,). 2. The app redirects to the selected social provider for authentication. 3. If authentication is successful, the app retrieves the user's profile information and creates an account if it doesn't exist. 4. The user is logged in and redirected to the home screen.
Exceptions	Information missing or invalid: 1. If any required information is missing or invalid, an error message is displayed 2. If the email and password do not match when the User tries to login into Step 4, the System shall display “Email and password do not match” to the User
Includes	None

Special requirements	The system needs to validate User input data
Assumptions	The User has an existing HawkerGo Account. User doesn’t forget password.
Notes and Issues	None

1.0.2 CreateAccount

Actor	User
Description	Allows a person to create an Account in HawkerGo to become a User
Preconditions	None
Postconditions	An account is created
Priority	High
Frequency	High
Flow of Events	1. The system prompts the User to enter their name, address, password, email, contact number, profile picture, role and back navigation icon. 2. The User enters the required information and selects the "Create Account" button to confirm his/her inputs. 3. An Account is created 4. The app presents set preferences feature where the user can choose: a. Dietary restrictions (e.g., vegetarian, halal, no seafood) b. Favourite cuisines (e.g., Chinese, Indian, Malay, Western) c. Spiciness tolerance level (e.g., mild, medium, spicy) d. Preferred price range 5. The user sets preferences or skips this step. 6. The user is redirected to the home screen or login page.
Alternative Flow	AF-S1 Returning to the Login page: 1. If the User selects the back navigation icon, the system will navigate the User to the login page. AF-S2 Social Sign-up: 1. The user selects a social sign-up option (Google, Facebook.). 2. The app redirects to the chosen social provider for authentication. 3. If authentication is successful, the app retrieves user details (e.g., name, email). 4. If additional details (e.g., phone number) are required, the user is prompted to enter them. 5. The account is created, and the user is logged in automatically.

	AF-S4 Email verification required: 1. If the app enforces email verification, a confirmation email is sent. 2. The user must verify their email before they can log in. 3. If they attempt to log in before verification, the app displays a message: "Please verify your email to activate your account."
Exceptions	Missing or invalid information: 1. If any required information is missing or invalid, an error message is displayed.
Include	None
Special requirements	The system needs to validate User input data
Assumptions	None
Notes and Issues	None

1.0.3 ManageProfile

Actor	User
Description	This use case describes how Users (both hawkers and Customers) can manage their profiles within the HawkerGo app. This includes updating personal details, changing passwords, setting preferences, and managing contact information
Preconditions	1. The User must be registered in the HawkerGo app 2. The User must be logged into their account
Postconditions	1. Users profile is updated successfully 2. The updated information is stored successfully on system 3. User receives confirmation message
Priority	High
Frequency	Moderate
Flow of Events	1. The User navigates to the profile management section in the HawkerGo app 2. The system displays the User's current profile information 3. The User selects the field(s) they want to update (e.g., name, email, phone number, password, preferences) 4. The User enters the new information 5. The User submits the changes 6. The system validates the new information and updates the profile 7. The system confirms the successful update to the User
Alternative Flow	None
Exceptions	1. The system displays an error message if the User enters invalid information (e.g., incorrect email format or weak password) 2. If there is a system error, the User is notified, and the update is not saved 3. If the User tries to update with an email or phone number already in use, the system rejects the change
Include	None
Special requirements	None
Assumptions	Users provide accurate information
Notes and Issues	None

Functional Requirement #2

1.1 ReportCrowdLevel

Actor	Customer
Description	Allows the Customer to report crowd level according to 3 different tiers; low, medium or high. The Customer can toggle between the 3 levels while waiting in line
Preconditions	1. The Customer must have an active account i.e (logged into the app) 2. Customer must have their location services enabled to submit reports of nearby Hawker Centres 3. Should have access to an internet
Postconditions	Crowd level is stored in the system
Priority	High
Frequency of Use	High
Flow of Events	1. The System detects customers location and displays Hawker Centre which they are at. 2. The customer reports crowd level by selecting the appropriate crowd level from predefined levels: low, medium or high 3. Customer submits report 4. The System saves the report to the database 5. The System updates the hawker centres' real-time crowd estimate 6. The System displays a confirmation message to the User
Alternative Flows	AF-S1 Customer manually selects location of hawker centre: 1. The Customer manually selects the location of their desired hawker centre 2. The System checks the Customers' current location 3. If the Customer isn't at the location of the hawker centre, the system prevents input of crowd level 4. The system informs the Customer that reporting is restricted to Customers physically present at the location
Exceptions	Repeated Reporting: 1. If the Customer repeatedly reports crowd level more than 5 times consecutively, crowd reporting action will be suspended for 10 minutes
Includes	None

Special Requirements	None
Assumptions	The Customer reports accurate crowd-level estimates
Notes and Issues	None

1.2 GetRecommendations

Actor	Customer
Description	Customers get food suggestions based on their dietary preferences, meal type, and other criteria
Preconditions	1. The Customer is logged into the System 2. The System has access to all the food options available in real time
Postconditions	1. The Customer can view food options and receives a list of food recommendations based on their inputs
Priority	Moderate
Frequency	Moderate
Flow of events	1. Customer selects GetRecommendations 2. The system processes the inputs and filters out the hawker options also taking into consideration the location and reviews that match the Customers inputs 3. The System displays a list of food recommendations from the hawker stalls, based on Customer inputs
Alternative Flows	AF-S3 The Customer wants more information: 1. The Customer requests details on the meal, eg calories, ingredients, etc AF-S3 No suitable Options available: 1. The System displays messages and suggests broader alternatives
Exceptions	Invalid Customer input: 1. The System prompts the Customer to change input or enter input again
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

1.3 ManageLoyalty

Actor	Customer
Description	Helps Customers track loyalty points and rewards
Preconditions	The Customer should have an account and must be logged in
Postconditions	Loyalty points are updated and stored in the system.
Priority	High
Frequency	Moderate
Flow of events	1. The Customer logs into the HawkerGo app 2. The Customer selects the ManageLoyalty option 3. Customers can choose to either ViewPoints, ViewRedeemableRewards, or RedeemRewards 4. If the Customer selects Viewpoints, the use case ViewPoints displays Customers' points which have been currently accumulated 5. If the Customer selects ViewRedeemableRewards, the use case ViewRedeemableRewards displays rewards which can be redeemed, given the Customer has sufficient points to redeem them 6. If the Customer selects RedeemRewards, the use case RedeemRewards will update the Customer’s points balance if a reward is redeemed 7. The System confirms transactions
Alternative Flows	None
Exceptions	None
Includes	1. Viewpoints 2. ViewRedeemableRewards
Special Requirements	The HawkerGo app supports push notifications for rewards
Assumption	Users get points based on interactions
Notes and Issue	None

1.3.1 ViewPoints

Actor	Customer
Description	Customers can view accumulated points
Preconditions	1. The Customer is logged in 2. The System has accumulated the points for the Customer currently
Postconditions	1. The Customer can view the current point balance
Priority	Moderate
Frequency	Moderate
Flow of events	1. The Customer selects ViewPoints in the app 2. The System retrieves and displays the Customer's current point balance 3. Customers can see additional details such as how many points were earned/expired
Alternative Flows	AF S2 The Customer has no points: 1. The System displays a message informing the Customer they have no points AF S2 Points display error 1. Issue with retrieving points, the system displays an error message
Exceptions	None
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

1.3.2 ViewRedeemableRewards

Actor	Customer
Description	Customer can view the list of redeemable rewards
Preconditions	1. The Customer is logged into the system 2. There are rewards available to be redeemed
Postconditions	1. Customers view a list of rewards they can redeem using points
Priority	Moderate
Frequency	Moderate
Flow of events	1. The Customer selects ViewRedeemableRewards on the app 2. Based on the Customer's current point balance, the system finds all the rewards the Customer can redeem 3. The System displays a list of redeemable rewards 4. If the Customer selects RedeemRewards, the use case RedeemRewards allows the Customer to redeem rewards successfully 5. The Customer can view additional details on these rewards i.e terms/description
Alternative Flows	AF S3 The Customer has no redeemable rewards: 1. The System displays a message saying there are no redeemable rewards
Exceptions	None
Includes	1. RedeemRewards
Special Requirements	None
Assumption	None
Notes and Issue	None

1.3.3 RedeemRewards

Actor	Customer
Description	Customers can use points to redeem any rewards from redeemable reward listing
Preconditions	The Customer is logged on to the System Rewards are available on the System
Postconditions	Customers can earn rewards
Priority	Moderate
Frequency	Moderate
Flow of events	1. The Customer selects RedeemRewards on the app 2. The System displays a list of rewards to be redeemed. 3. The Customer selects a reward to redeem 4. The System verified Customer has sufficient points in balance to redeem the reward 5. The System confirms redemption and deducts points from the balance, respectively. 6. The Customer receives a reward and is notified.
Alternative Flows	AF-S5 The Customer wants to cancel the redemption: 1. Customers may cancel the redemption process before completing it, and the System returns them to the rewards catalogue without changing their points balance.
Exceptions	Insufficient points: 1. The System displays insufficient points message and advises the Customer to earn more points A reward is unavailable: 1. If the reward is no longer available or has expired, the System informs the Customer and prompts them to choose a different reward.
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

1.4 ViewHawkerDetails

Actor	Customer
Description	Permits the Customer to view the hawker details, including location, crowd level and menu
Preconditions	1. The Customer must be logged into the app 2. Hawker centre data must be available to the System
Postconditions	1. The Customer obtains relevant stall information
Priority	High
Frequency	High
Flow of events	1. The Customer logs into the app 2. Customer selects ViewHawkerDetails 3. The System prompts the Customer to choose a desired activity: NavigateLocation, ViewCrowdLevel, ManageQueue 4. If the Customer selects NavigateLocation, the Customer uses the included use case NavigateLocation to search for nearby hawkers 5. If the Customer selects ViewCrowdLevel, the Customer uses the included use case ViewCrowdLevel to gauge the crowd level of hawker centres 6. If the Customer selects ManageQueue, the Customer uses the included use case to view the queue levels 7. If the Customer selects ViewStallDetails, the Customer uses the use case ViewStallDetails to view the menus and the attributes of the stalls registered on the system.
Alternative Flows	None
Exceptions	None
Includes	1. NavigateLocation 2. ViewCrowdLevel 3. ViewStallDetails
Special Requirements	None
Assumption	None
Notes and Issue	None

1.4.1 NavigateLocation

Actor	User
Description	Provides Users with directions to their selected hawker centre or stall
Preconditions	1. The Customer has selected a hawker stall 2. GPS and maps services are enabled
Postconditions	The customer receives real-time directions to a hawker stall
Priority	High
Frequency	High
Flow of events	1. The Customer selects NavigateLocation after selecting the desired hawker stall 2. GPS detects Customer's current location 3. GPS provides a route from the customers' current location to the hawker stall 4. The customer follows the route and the system concurrently tracks the User's location
Alternative Flows	AF-S4 Customer goes the wrong way: 1. If the customer follows the wrong direction, mapping services will automatically, recalculate the distance, and provide an adjusted route accordingly
Exceptions	GPS is disabled: 1. If GPS is disabled, the System prompts the customer to enable location services
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

1.4.2 ViewCrowdLevel

Actor	Customer
Description	Allows Customers to view crowd levels of a selected location
Preconditions	1. The customer should be logged into the System 2. The system should have access to real-time crowd data
Postconditions	1. The customer successfully views crowd-level data information for the selected location
Priority	High
Frequency	Moderate
Flow of events	1. The customer navigates to the ViewCrowdLevel feature 2. The system prompts the Customer to select the desired location 3. The System uses real-time data, historical trends and Customer reports to calculate and display crowd levels based on predefined terms ie: low, medium, high 4. Customer makes decisions based on crowd-level data
Alternative Flows	AF-S3 The system can't retrieve data: 1. The customer is informed that the System can't retrieve real-time data 2. The system displays the last known crowd-level data along with a timestamp
Exceptions	The system fails to fetch data: 1. The System fails to fetch data due to connectivity issues, and an error message is displayed 2. The selected location has no available crowd data, prompting the Customer to check another place.
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

1.4.3 ViewStallDetails

Actor	Customer
Description	Customers can view menu details of selected hawker stalls.
Preconditions	1. The customer is logged into the system 2. The hawker stall exists in the database and has a menu available.
Postconditions	1. The menu is displayed to the customer
Priority	High
Frequency	High
Flow of Events	1. The customer launches the HawkerGo app. 2. The customer navigates to the list of hawker stalls. 3. The customer selects a specific hawker stall. 4. The customer can view the stall's specific attributes: a. specific dietary restrictions b. spice level range c. price range 5. The system displays the option to SubmitReview. If a Customer wants to submit a review, they can use this use case. 6. The system displays the option to ManageOrder. If a customer wants to place the order or view the order status, they can use the use case ManageOrder. 7. The app also retrieves the stall's menu from the database. 8. The app displays the menu, including item names, descriptions, prices, and images (if available). 9. The customer can browse the menu and view details for each item.
Alternative Flows	None
Exceptions	Menu unavailable: 1. If the selected hawker stall has no menu, the app displays a message to the customer. Stall not found: 1. If a stall no longer exists on the system, the app notifies the customer and suggests similar stalls.

Includes	1. SubmitReview 2. ManageOrder
Special Requirements	None
Assumptions	1. Every hawker has a basic menu listed on the app 2. Customers prefer digital menus over physical ones
Notes and Issues	None

1.4.3.1 SubmitReview

Actor	Customer
Description	Customers can submit ratings for stalls, or specific food items within the HawkerGo app.
Preconditions	1. The customer must be logged into the system 2. The customer must have visited the stall to submit a rating
Postconditions	1. Rating is successfully submitted and updated on the system 2. The overall rating is recalculated and stored in the database 3. The customer receives a confirmation message once the rating is stored
Priority	Moderate
Frequency	Moderate
Flow of Events	1. The customer navigates to the SubmitReviews feature in the HawkerGo app 2. The system displays a list of currently visited hawker centres or stalls 3. Customers select the stall or food item they wish to review 4. The customer inputs a rating (1-5) stars and can leave an optional comment 5. The customer submits the review 6. The system validates the rating and updates the overall review 7. The system sends a confirmation message to the customer.
Alternate Flow	AF-S1 If the customer has not visited the stall: 1. The system does not allow the customer to submit a review. 2. The system prompts the customer to visit a hawker centre before submitting a review.
Exceptions	The system is down: 1. If the system is down, the customer gets a warning message
Includes	None
Special Requirements	None
Assumptions	The customer has visited the stall before reviewing
Notes and Issues	None

1.4.3.2 ManageOrder

Actor	Customer
Description	Allows Customer to place and view order details for desired hawker stall
Preconditions	1. The Customer has access to the queue management System 2. The System has an active queue for a specific location/service
Postconditions	1. Customer successfully joins, view, or manages their position in the queue
Priority	Moderate
Frequency	Moderate
Flow of events	1. Customer selects desired location 2. Customer selects manage queue 3. The system displays current queue status 4. The Customer can choose to a. Place order b. View order status 5. System updates order and notifies Customer for any changes 6. System alerts Customer when their order is ready 7. System changes order status when order is collected
Alternative Flows	None
Exceptions	None
Includes	1. PlaceOrder 2. ViewOrderStatus
Special Requirements	None
Assumption	Customers don't cancel orders once placed.
Notes and Issue	None

1.4.3.2.1 PlaceOrder

Actor	Customer
Description	Customers can place an order through our HawkerGo app
Preconditions	1. Customer has logged into the System 2. Hawker stall supports placing order through the app
Postconditions	1. The customer places order and is assigned an order ID
Priority	High
Frequency	High
Flow of Events	1. The Customer selects a hawker stall in the app 2. The Customer selects the "PlaceOrder" button 3. The System validates the request 4. The system assigns an order ID to the Customer 5. The system updates the Customer's order status in real-time 6. The System notifies the Customer when their order is ready
Alternative Flow	AF-S3 Stall stops taking orders: 1. If the stall stops taking orders, the System notifies the Customer and prevents more Customers from placing orders.
Exceptions	The stall closes before the Customer places an order: 1. The System prevents the Customer from joining and notifies the Customer
Include	None
Special requirements	The system should update the order status dynamically
Assumptions	Customers will check the app for order status updates
Notes and Issues	None

1.4.3.2.2 ViewOrderStatus

Actor	Customer
Description	Customers can check their order status using the HawkerGo app
Preconditions	1. The Customer is logged in 2. The Customer has placed an order
Postconditions	1. The Customer can see their current order status 2. The system should update the order status dynamically
Priority	High
Frequency	High
Flow of Events	1. The System retrieves and displays the current order status 2. The System updates the order status in real-time 3. The System sends notifications when the order status changes
Alternative Flow	None
Exceptions	The System fails to retrieve the order status: 1. Error message is displayed
Include	None
Special requirements	Order status must refresh in real-time
Assumptions	Customer regularly checks app for order status
Notes and Issues	None

Functional Requirement #3

2.0 ManageStallOperations

2.0.1 StallQueueManagement

Actor	StallOwner
Description	Allow Stall Owners to manage the queue for their stalls
Preconditions	The Stall Owner is logged into the system
Postconditions	Queue information is updated with the latest status
Priority	High
Frequency	High
Flow of Events	1. The Stall Owner selects the ManageQueue option 2. The system retrieves the stall's current queue 3. The system displays the current queue status to the Stall Owner 4. The Stall Owner can choose to: a. ViewQueue (see all queued orders) b. ResolveOrder (update the status of an order) 5. If the Stall Owner selects ViewQueue, the system fetches and displays all queued orders 6. If the Stall Owner selects ResolveOrder, the system prompts the Stall Owner to select an order to update 7. The Stall Owner selects an order and updates its status (e.g., completed, cancelled) 8. The system updates the queue and reflects the changes 9. The system confirms the status update to the Stall Owner
Alternative Flow	None
Exceptions	None
Include	None
Special requirements	None
Assumptions	Queue status changes are handled in real time
Notes and Issues	None

2.0.2 ManageStallMenu

Actor	Stall Owner
Description	Allow Stall Owners to manage their stall's menu
Preconditions	Stall Owner is logged into the System
Postconditions	Menu is updated with the latest items that are available
Priority	Moderate
Frequency	Moderate
Flow of events	1. Stall Owner will be presented with 3 use cases a. RemoveMenuItem b. AddMenuItem c. ViewMenu 2. If Stall Owner selects RemoveMenuItem, then the Stall Owner uses the included use case RemoveMenuItem to remove items from the menu or mark them as unavailable 3. If Stall Owner selects AddMenuItem, then Stall Owner uses the included AddMenuItem use case to add more items into their menus 4. If Stall Owner selects ViewMenu, then Stall Owner uses the included ViewMenu use case to view the current items on their menu
Alternative Flows	None
Exceptions	None
Includes	1. RemoveMenuItem 2. AddMenuItem 3. ViewMenu
Special Requirements	None
Assumption	None
Notes and Issue	None

2.0.2.1 RemoveMenuItem

Actor	Stall Owner
Description	Allows Store Owner to remove unwanted items or mark items as unavailable
Precondition	1. Store Owner is logged in and authenticated 2. There exists at least 1 item on the Stall Owner’s menu
Postconditions	1. The menu is updated with the latest list of available items
Priority	Moderate
Frequency	Moderate
Flow of events	1. If Stall Owner selects Remove_Item 2. Stall Owner selects which specific item to remove from the menu 3. The System deletes that item form the menu 4. The System will display the updated menu 5. If Stall Owner selects to mark an item as unavailable 6. Stall Owner specifies which item to mark 7. That item will be marked temporarily unavailable 8. The System will display the updated menu
Alternative Flows	None
Exceptions	None
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

2.0.2.2 AddMenuItem

Actor	Stall Owner
Description	Allows Store Owner to add an item to their menu
Preconditions	1. Store Owner is logged in and authenticated
Postconditions	2. The menu is updated with the latest list of available items
Priority	Moderate
Frequency	Moderate
Flow of events	1. Store Owner selects AddMenuItem 2. Store Owner enters the name, price and includes a picture of that item 3. The System will display the updated menu
Alternative Flows	AF-S2 Doesn't provide sufficient information: 1. If Store Owner does not provide either the name, price or picture, they will be prompted to provide the missing information
Exceptions	None
Includes	None
Special Requirements	None
Assumption	Store Owner has the name, price and picture of the menu item
Notes and Issue	None

2.0.2.3 ViewMenu

Actor	Stall Owner
Description	Allows Stall Owner to view their menu
Preconditions	1. Store Owner is logged in and authenticated
Postconditions	1. Store Owner sees menu items
Priority	High
Frequency	High
Flow of events	1. Store Owner selects View_Menu 2. System will display the name, price and picture of the items on the menu
Alternative Flows	AF-S1 Menu is empty: 1. If the menu does not contain any items, System will display "Menu is Empty"
Exceptions	None
Includes	None
Special Requirements	None
Assumption	None
Notes and Issue	None

2.0.3 ViewMetrics

Actor	Stall Owner
Description	Allows Stall Owner to view their sales volume over a specific time period
Preconditions	1. Stall Owner is logged in and authenticated 2. Stall has past sales record
Postconditions	1. Stall Owner can view analytics of their sales volume
Priority	High
Frequency	High
Flow of events	1. Stall Owner selects ViewAnalytics 2. Stall Owner chooses a specific time range for the sales data 3. The System displays the corresponding sales volume
Alternative Flows	AF-S2 No time specified: 1. If no time period was specified, the time range will default to the last 30 days
Exceptions	None
Includes	None
Special Requirements	None
Assumption	Sales volume is updated accurately
Notes and Issue	None