USCoffee

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 07/07/2023 | 1.0 | First version of Use-case specification | Group 109 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 5

2.1 Use-case: Login 5

2.2 Use-case: Logout 6

2.3 Use-case: Update user profile 7

2.4 Use-case: Searching products 8

2.5 Use-case: Calculate Charges 9

2.6 Use-case: Manage total revenue. 10

2.7 Use-case: Update a product. 11

2.8 Use-case: Add products to cart. 12

2.9 Use-case: Manage users. 13

2.10 Use-case: Manage application. 14

2.11 Use-case: Add a product 14

2.12 Use-case: Delete a product 16

2.13 Use-case: Make payment 17

2.14 Use-case: Assign Internet 18

2.15 Use-case: Request Internet 19

# Use-case Model

A diagram of a network

Description automatically generated

# Use-case Specifications

## Use-case: Login

|  |  |
| --- | --- |
| Use case Name | Login |
| Brief description | This use case describes how a user login into system to use other features. |
| Actors | Admin, Salesperson, Shop owner, diner |
| Basic Flow | 1. The user navigates to the login page. 2. The system presents a login form to the user, requesting their username/email and password. 3. The user enters their valid username/email and password. 4. The user submits the login form. 5. The system verifies the provided credentials against the stored user data. 6. The system confirms the user's identity and grants access to their account. 7. The system redirects the user to their account dashboard or a designated landing page. 8. The user can now interact with their account and perform related actions. |
| Alternative Flows | **Alternative flow 1: Users forgot their password.**   1. After Step 2, the user can click on a "Forgot Password" link. 2. The system prompts the user to provide their registered email address. 3. The user enters their email address and submits the form. 4. The system verifies the email address and sends a password reset link to the user's email. 5. The user receives the email and follows the instructions to reset their password. 6. The user can then proceed with logging in using their new password. |
| Pre-conditions | The user has a registered account.  The user has a valid username/email and password combination.  The user has an active internet connection. |
| Post-conditions | The user is successfully authenticated and gains access to their account. |

## Use-case: Logout

|  |  |
| --- | --- |
| Use case Name | Logout |
| Brief description | This use case describes how a user logout of system to use other features. |
| Actors | Admin, Salesperson, Shop owner, diner |
| Basic Flow | 1. The user clicks on the "Logout" or "Sign Out" button in the application interface. 2. The system receives the logout request from the user. 3. The system invalidates the user's session and clears any session-related data. 4. The system logs out the user and terminates their access to the account. 5. The system redirects the user to the login page or a designated landing page, indicating successful logout. |
| Alternative Flows | **Alternative flow 1: Automatic Logout (Inactivity Timeout)**   1. If the system has a feature to automatically log out users after a period of inactivity, the following flow can be included: 2. The system detects that the user's session has exceeded the inactivity timeout threshold. 3. The system automatically logs out the user, invalidates the session, and clears session-related data. 4. The system redirects the user to the login page or a designated landing page, indicating automatic logout due to inactivity. |
| Pre-conditions | The user is logged in and has an active session |
| Post-conditions | The user's session is terminated, and they are logged out of the account.  The user is redirected to the login page or a designated landing page. |

## Use-case: Update user profile

|  |  |
| --- | --- |
| Use case Name | Update profile |
| Brief description | This use case describes how a user update his/her profile on website. |
| Actors | Admin, Salesperson, Shop owner, diner |
| Basic Flow | 1. The user navigates to the profile settings or account settings page. 2. The system presents the user's current profile information for editing. 3. The user modifies the desired fields in the profile form (e.g., name, contact details, profile picture, etc.). 4. The user submits the updated profile form. 5. The system validates the updated profile information. 6. The system saves the validated changes to the user's profile. 7. The system displays a success message indicating that the profile update was successful. 8. The system updates the displayed profile information to reflect the changes made by the user. |
| Alternative Flows | **Alternative flow 1: Change password**   1. After Step 2 or Step 4, the user can choose to change their password. 2. The system provides a separate form specifically for changing the password. 3. The user enters their current password and a new password. 4. The system validates the entered passwords and updates the user's password accordingly. 5. The system displays a success message indicating that the password change was successful. |
| Pre-conditions | The user is logged in and has an active session.  The user has navigated to the profile settings or account settings page. |
| Post-conditions | The user's profile information is updated and saved in the system.  The updated profile information is displayed to the user.  The user can continue using the application with the updated profile information. |

## Use-case: Searching products

|  |  |
| --- | --- |
| Use case Name | Search product |
| Brief description | This use case describes how a user search a product on website |
| Actors | Admin, Salesperson, Shop owner, diner |
| Basic Flow | 1. The user enters a search query or keyword in the search input field. 2. The user submits the search query by clicking a "Search" button or pressing the Enter key. 3. The system receives the search query and validates it. 4. The system searches the product database using the validated search query. 5. The system retrieves a list of products that match the search query. 6. The system presents the list of matching products to the user. 7. The user can scroll through the list of products and view summary information about each product (e.g., name, image, price, rating). 8. The user can click on a product to view detailed information about it. 9. The system displays the detailed product information, including additional details, images, customer reviews, and related products. 10. The user can add the product to their cart, wishlist, or perform other relevant actions. |
| Alternative Flows | **Alternative flow 1: Empty search query**   1. The system detects that the user submitted an empty search query. 2. The system displays an error message indicating that a valid search query is required. 3. The user is prompted to enter a valid search query and resubmit the search.   **Alternative flow 2: No Matching Products**   1. The system searches the product database but finds no products that match the search query. 2. The system displays a message indicating that no matching products were found. 3. The user can refine their search query and perform a new search. |
| Pre-conditions | The user has access to the website/application.  The user is on the page or section where the product search functionality is available. |
| Post-conditions | The user is presented with a list of relevant products based on their search query.  The user can view detailed information about the products and proceed with further actions. |

## Use-case: Calculate Charges

|  |  |
| --- | --- |
| Use case Name | Calculate Charges |
| Brief description | This use case describes how a shop owner/salesperson calculate charges |
| Actors | Shop owner, salesperson |
| Basic Flow | 1. The system receives the customer’s order. 2. The system applies any applicable discounts or promotions. 3. The system calculates the subtotal, taxes, and total amount. 4. The system presents the calculated charges to the shop owner or salesperson. 5. The shop owner/ salesperson provides the customer with an accurate invoice or receipt. |
| Alternative Flows | **Alternative flow 1: Step 4: Applying Discounts or Promotions**   1. Step 4: Applying Discounts or Promotions 2. The system checks if any discounts or promotions are applicable to the customer's order. 3. If applicable, the system applies the discounts or promotions to the relevant items/services in the order. 4. The system recalculates the subtotal considering the applied discounts or promotions.   **Alternative flow 2: Step 3: Data Unavailability**   1. The system encounters a situation where the necessary data for calculating charges is unavailable or incomplete. 2. The system displays an error message indicating the missing or incomplete data. 3. The shop owner is prompted to provide or update the necessary data before proceeding with charge calculation. |
| Pre-conditions | The shop owner/salesperson has access to the shop management system or interface.  The shop owner/salesperson has entered or imported the necessary data for calculating charges, such as product prices, taxes, and discounts. |
| Post-conditions | The shop owner/salesperson obtains the calculated charges for the customer's purchase.  The shop owner/salepersons can provide the customer with an accurate invoice or receipt. |

## Use-case: Manage total revenue.

|  |  |
| --- | --- |
| Use case Name | Manage total revenue |
| Brief description | This use case describes how a shop owner manage total revenue |
| Actors | Shop owner |
| Basic Flow | 1. The shop owner or authorized personnel accesses the financial management system. 2. The system calculates the total revenue from recorded sales and transactions. 3. The system generates reports and insights based on the total revenue. 4. The shop owner or authorized personnel reviews and analyzes the total revenue data. 5. Informed decisions regarding pricing, promotions, or business strategies are made based on the insights. |
| Alternative Flows | **Alternative flow 1: Step 3: Revenue Comparison**   1. Step 3: Revenue Comparison 2. The system compares revenue data across different time periods. 3. Reports or visualizations highlight revenue growth or decline between compared periods. 4. Insights from the comparison help optimize strategies and make data-driven decisions |
| Pre-conditions | The shop owner or authorized personnel has access to the financial management system or interface.  Sales and transaction data are recorded accurately in the system. |
| Post-conditions | The total revenue of the shop is managed and tracked.  Reports and insights on total revenue are available for analysis and decision-making. |

## Use-case: Update a product.

|  |  |
| --- | --- |
| Use case Name | Update a product |
| Brief description | This use case describes how a user update information of a product |
| Actors | Shop owner, admin, salesperson |
| Basic Flow | 1. The user navigates to the product management section of the system. 2. The user searches for the specific product they want to update. 3. The system displays the product details for the selected product. 4. The user modifies the desired fields or attributes of the product, such as name, price, description, or stock quantity. 5. The user submits the updated information. 6. The system validates the updated data for accuracy and completeness. 7. The system saves the updated product information in the database. 8. The system confirms the successful update and displays a confirmation message. 9. The updated product details are reflected in the product listings or catalog. |
| Alternative Flows | **Alternative flow 1: Step 2: Product Not Found**   1. The user searches for a specific product, but the system does not find a matching product. 2. The system displays a message indicating that the product was not found. 3. The user may need to verify the product details or create a new product entry if necessary.   **Alternative flow 1: Step 6: Validation failed.**   1. The system detects that the updated data is invalid or incomplete. 2. The system displays an error message indicating the validation errors. 3. The user is prompted to correct the invalid fields and resubmit the update. |
| Pre-conditions | The user (admin, salesperson, or shop owner) has the necessary permissions to update products.  The user has access to the product management system or interface. |
| Post-conditions | The product information is successfully updated in the system.  The updated product details are reflected in the product listings or catalog. |

## Use-case: Add products to cart.

|  |  |
| --- | --- |
| Use case Name | Update a product |
| Brief description | This use case describes how a customers add a product to their cart |
| Actors | Diners (Customer) |
| Basic Flow | 1. The diner browses the available menu items or products. 2. The diner selects a specific product they want to add to their cart. 3. The system displays the product details, including name, price, description, and options if applicable. 4. The diner customizes the product by selecting any desired options (e.g., size, toppings, quantity). 5. The diner confirms their selection and chooses to add the product to their cart. 6. The system validates the availability of the selected product and its options. 7. The system adds the product to the diner's cart, associating it with their account or session. 8. The system confirms the successful addition of the product and displays a notification or summary of the updated cart. 9. The diner can continue browsing and adding more products to their cart or proceed to checkout. |
| Alternative Flows | **Alternative flow 1: Step 6: Product Unavailability**   1. The system detects that the selected product or options are currently unavailable (e.g., out of stock). 2. The system displays a message indicating unavailability. 3. The diner may be prompted to choose an alternative product or proceed without the unavailable item.   **Alternative flow 2: Step 2: No Product Selected.**   1. The diner does not select any specific product to add to the cart. 2. The system may display a message or provide suggestions to encourage the selection of a product. |
| Pre-conditions | The diner has access to the online ordering system or application.  The diner has logged into their account or provided necessary identification details. |
| Post-conditions | The selected product is successfully added to the diner's cart.  The diner can view and proceed to checkout with the items in their cart. |

## Use-case: Manage users.

|  |  |
| --- | --- |
| Use case Name | Manage users |
| Brief description | This use case describes how admin manage users |
| Actors | Admin |
| Basic Flow | 1. The administrator accesses the user management system. 2. The administrator selects a user and performs desired actions (create, modify, deactivate). 3. The system updates the user information and permissions accordingly. 4. Confirmation of the successful user management action is displayed. |
| Alternative Flows | **Alternative flow 1: Step 3: User not found**   1. The administrator searches for a specific user, but the system does not find a matching user account. 2. The system displays a message indicating that the user was not found. 3. The administrator may need to verify the user details or create a new user account if necessary. |
| Pre-conditions | The administrator has access to the user management system or interface.  User account information and related data are available in the system. |
| Post-conditions | User-related actions, such as creation, modification, and deactivation, are successfully performed.  User accounts and permissions are accurately updated in the system. |

## Use-case: Manage application.

|  |  |
| --- | --- |
| Use case Name | Manage application |
| Brief description | This use case describes how admin manage application |
| Actors | Admin |
| Basic Flow | 1. The administrator accesses the application management system. 2. The administrator performs desired actions (configure settings, update modules). 3. The system applies the requested changes or updates to the application. 4. Confirmation of the successful application management action is displayed. |
| Alternative Flows | **Alternative flow 1: Step 5: Invalid Input or Incomplete Information**   1. The system detects that the provided information for application configuration or update is invalid or incomplete. 2. The system displays an error message indicating the validation errors. 3. The administrator is prompted to correct the invalid fields or provide complete information before proceeding. |
| Pre-conditions | The administrator has access to the application management system or interface.  Application configuration settings and related data are available in the system. |
| Post-conditions | Application-related actions, such as configuration, settings, and updates, are successfully performed.  The application functions with the desired settings and updates in place. |

## Use-case: Add a product

|  |  |
| --- | --- |
| Use case Name | Add product |
| Brief description | This use case describes how users add a new product to catalog |
| Actors | Admin, Shop owner, saleperson |
| Basic Flow | 1. The actor accesses the product management system. 2. The actor fills in the required product information in the provided form. 3. The actor submits the product details for addition. 4. The system validates the entered information and saves the new product in the catalog. 5. The system confirms the successful addition of the product. |
| Alternative Flows | **Alternative flow 1: Step 2: Adding an existing product**   1. The actor attempts to add a product with information matching an existing product. 2. The system detects the duplicate and prompts the actor to provide unique product details. 3. The actor resubmits the product details with distinct information. 4. The system successfully adds the new product to the catalog. |
| Pre-conditions | The actor has the necessary permissions to add products.  The actor has access to the product management system or interface. |
| Post-conditions | The new product is successfully added to the system's product catalog. |

## Use-case: Delete a product

|  |  |
| --- | --- |
| Use case Name | Delete product |
| Brief description | This use case describes how users delete a product from catalog |
| Actors | Admin, Shop owner, saleperson |
| Basic Flow | 1. The actor accesses the product management system. 2. The system displays a list of existing products and available product management actions. 3. The actor searches for the specific product they want to delete. 4. The system presents the details of the selected product. 5. The actor confirms their intention to delete the product. 6. The system prompts for confirmation to proceed with the deletion. 7. The actor confirms the deletion request. 8. The system removes the product from the product catalog. 9. The system displays a success message confirming the deletion of the product. |
| Alternative Flows | **Alternative flow 1: Step 3: Delete an unexisting product.**   1. The actor attempts to add a product with information matching an existing product. 2. The system detects the duplicate and prompts the actor to provide unique product details. 3. The actor resubmits the product details with distinct information. 4. The system successfully adds the new product to the catalog. |
| Pre-conditions | The actor has the necessary permissions to delete products.  The actor has access to the product management system or interface. |
| Post-conditions | The selected product is successfully removed from the system's product catalog |

## Use-case: Make payment

|  |  |
| --- | --- |
| Use case Name | Make payment |
| Brief description | This use case describes how customers pay for their cart |
| Actors | Diners |
| Basic Flow | 1. The customer proceeds to checkout and reviews the cart. 2. The customer selects a payment method and provides necessary details if required. 3. The system securely processes the payment and confirms the transaction. 4. The customer receives a payment confirmation and receipt. |
| Alternative Flows | **Alternative flow 1: Step 2:** **Payment Declined.**   1. The customer's payment is declined during checkout. 2. The system displays an error message and prompts the customer to try another payment method or verify payment details. 3. The customer resubmits the payment request with an alternative method or corrected details. 4. The payment is successfully processed, and the system confirms the transaction. 5. The customer receives a payment confirmation and receipt. |
| Pre-conditions | The customer has added products to their cart and is ready to proceed with the payment.  The customer has a valid payment method linked to their account or is willing to provide payment details. |
| Post-conditions | The payment for the cart is successfully processed.  The customer receives a payment confirmation and receipt. |

## Use-case: Assign Internet

|  |  |
| --- | --- |
| Use case Name | Assign internet |
| Brief description | This use case describes how admin assign internet for website |
| Actors | Admin |
| Basic Flow | 1. The administrator selects the user or location to assign the internet service. 2. The administrator chooses the appropriate internet plan or configuration. 3. The internet is successfully assigned, and the service becomes available to the specified user or location. |
| Alternative Flows | **Alternative flow 1: Step 2:** **Assignment Validation Failure.**   1. The system detects conflicting assignments or invalid details. 2. The system displays an error message indicating the validation errors. 3. The administrator is prompted to correct the conflicting details or resolve any constraints before resubmitting the assignment request. |
| Pre-conditions | The administrator has the necessary permissions and access to manage internet assignments.  The internet service is available and accessible to be assigned. |
| Post-conditions | The internet is successfully assigned to the specified user or location. |

## Use-case: Request Internet

|  |  |
| --- | --- |
| Use case Name | Request internet |
| Brief description | This use case describes how customers request for accessing internet |
| Actors | Customers |
| Basic Flow | 1. The customer selects the desired internet plan and provides contact and location details. 2. The customer submits the internet service request. 3. The request is successfully submitted to the internet service provider for further processing. |
| Alternative Flows | **Alternative flow 1: Step 2:** **No Available Internet Plans.**   1. The system does not display any available internet plans or options. 2. The customer may be informed that internet service is not currently available in their area. |
| Pre-conditions | The customer has access to the internet service provider's request system or interface.  The customer is eligible for requesting internet service. |
| Post-conditions | The customer's request for internet service is successfully submitted. |