|  |  |
| --- | --- |
| Adeleke Oyebode Adeola | **610516** |
| Amit Bhattarai | **610514** |
| Ny Aina Michael Andriantsoa |  |
| Lamhot Siagian | **610517** |

**E-Shopper: Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 1** | | | |
| **Name** | | Search products | |
| **Brief description** | | This use case allows the users to search for products | |
| **Actors** | | Customer | |
|  | | | |
| **Preconditions** | | | |
| All the users must be logged in | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.1.0 Search for product by keyword** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The user types a product keyword in the search box and hits search | | The system displays the products matching the keywords |
| 2 | The user types in a product number | | The system retrieves the corresponding product and displays it in the dashboard |
| **Postconditions** | | | |
| The product matching the keywords that was searched by the user will be given additional hit points. | | | |
|  | | | |
| **Business Rules** | | | |
| No special characters can be used to look for a product | | | |
|  | | | |
| **1.1.1 Browse through product catalog** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The user clicks to view a list of all the categories of products available | | The system displays all the categories of products |
| 2 | The user selects a category | | The system returns a list of all the products in the category |
| **Postconditions** | | | |
| The user will get a list of products sorted by most searched | | | |
| **Business Rule** | | | |
| The user cannot view multiple catalogs in the same dashboard | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 2** | | | |
| **Name** | | Cart CRUD | |
| **Brief description** | | This use case provides a functionality to add products to the shopping cart, modify the products already in the cart or delete them from the cart | |
| **Actors** | | Customer | |
|  | | | |
| **Preconditions** | | | |
| The customer must be logged in | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Add product to cart** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The customer searches for the product they want to add to the cart | | The system displays the products wanted by the customer |
| 2 | The customer selects the number of products they want, and presses add to cart | | The system checks if the item is in stock and adds the product to the shopping cart |
| **Postconditions** | | | |
| The product is added to the shopping cart and the availability of that product is decreased temporarily until customer checks out | | | |
| **Business Rule** | | | |
| Customers cannot order products more than their availability | | | |
|  | | | |
| **1.1 Update the number of products in the cart** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The user goes to the shopping cart | | The system returns a list of all the products in the shopping cart. |
| 2 | The user selects the number of products in the cart | | The system checks the product availability and puts constraints on how many products they can select |
| 3 | The user finishes updating the number of products | | The system updates the shopping cart |
|  | | | |
| **1.2 Delete product from the cart** | | | |
| 1 | The user goes to the shopping cart | | The system returns a list of all the products in the shopping cart. |
| 2 | The user presses the delete button | | The system removes the product and updates the shopping cart |
| **Postconditions** | | | |
| The shopping cart will be updated | | | |
| **Business Rule** | | | |
| The customer cannot select the products more than their number in the stock | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 3** | | | |
| **Name** | | View Order | |
| **Brief description** | | This use case provides a functionality for a customer to view all of their orders | |
| **Actors** | | Customer | |
|  | | | |
| **Preconditions** | | | |
| The customer must be logged in | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Viewing order** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The user goes to orders section | | The system displays the list of all the orders placed by the customer |
| 2 | The user selects a specific order | | The system displays the details about the order |
| **Postconditions** | | | |
| The orders remain unchanged | | | |
|  | | | |
| **Business Rules** | | | |
| Users cannot modify an already placed order | | | |
|  | | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 4** | | |
| **Name** Merchant Create Product | | |
| **Brief description** This use case allows the approved merchants to create products | | |
| **Actors** Merchants | | |
|  | | |
| **Preconditions** | | |
| The merchant must logged in to the system | | |
| **Flows of Events:** | | |
| **1. Basic Flows** | | |
| **1.1.0 Create Faculty Profile** | | |
| **Step** | **User Actions** | **System Actions** |
| 1 | The Merchant clicks the Add Product link on the dashboard | The system displays the **Add Product** form with the fields for product name, product code, brief description, full details, price, category, color, brand, size, discount in percentage, Number of available product. |
| 2 | The Merchant fills out the form and requests the system to save the details | The system verifies that there’s no other product with the same name under the same merchant in the database and all required form fields are filled, saves the form and returns the success message on success or a fail message in case of failure. In case another product exists with the product code, the system returns the message indicating a duplicate entry exists. |
| **Postconditions** | | |
| The new product is persisted in the system | | |
|  | | |
| **Business Rules** | | |
| No duplicate product allowed under the same merchant. For each product code, the Merchant code is appended to the product code for product identification | | |
|  | | |
| **1.1.1 Read/view Buyers Orders** | | |
| **Step** | **User Actions** | **System Actions** |

|  |  |  |
| --- | --- | --- |
| 1 | The Merchant selects to view a list of orders by buyers who have ordered his/her products | The system returns a list of all orders of the products sold by the merchant. The system shall present a list showing the product code, quantity ordered, amount, Payment status. |
|  |  |  |
|  | | |
| **1.1.2 Update Product** | | |
| **Step** | **User Actions** | **System Action** |
| 1 | The merchant selects to view a list of all his/her products | The system returns a list of all merchant products. |
| 2 | The merchant selects the product of the merchant he/she wants to update | The system displays an editable product form pre-populated with the product details. |
| 3 | The merchant updates the fields they want to update and requests system to save the new details | The system updates the record and returns the success message or a fail message on exception. |
| **Postconditions** | | |
| The merchant product will be updated | | |
| **Business Rule** | | |
| The Product code shall not be editable if the product is already order in the past. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 5** | | | |
| **Name** | | User Profile CRUD | |
| **Brief description** | | This use case allows the admin to manage profiles for merchant and buyer | |
| **Actors** | | Admin | |
|  | | | |
| **Preconditions** | | | |
| The admin must log in to the system | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.1.0 Create User Profile** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin calls the create user profile command | | The system displays the user profile form with the fields for first name, last name, email, username, password, role (merchant or buyer or admin). |
| 2 | The Admin fills out the form and requests the system to save the details | | The system verifies that there’s no other profile in the database with the same username and email address then saves the user and either returns a success message or a failure message. In case another profile exists with the same username or email address, the system returns the message indicating a duplicate entry exists. It may be a server-side verification or a client-side verification or both. |
| **Postconditions** | | | |
| The user profile is persisted in the system and that profile can now be used to browse through the application according to the roles it has. | | | |
|  | | | |
| **Business Rules** | | | |
| No duplicate username or email address. A unique profile is identified by username and email address. | | | |
|  | | | |
| **1.1.1 Read/view User Profile** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin selects to view a list of user profiles | | The system returns a list of all user profiles. |
| 2 | The admin selects to view a profile of one of the user from the list of profiles | | The system returns the profile of the user as a string with the names (first name and last name), email, roles, and other possible details. |
|  | | | |
| **1.1.2 Update User Profile** | | | |
| **Step** | **User Actions** | | **System Action** |
| 1 | The admin selects to view a list of user profiles | | The system returns a list of all user profiles. |
| 2 | The admin selects the profile of the user they want to update | | The system displays an editable user profile form pre-populated with the user profile details |
| 3 | The admin updates the fields they want to update and requests system to save the new details | | The system updates the record and returns the success message or a fail message on exception. |
| **Postconditions** | | | |
| The user profile will be updated in the database. | | | |
| **Business Rule** | | | |
| The email field and username fields can be changed but not into ones which already exist in the database otherwise it will be a duplication. | | | |
|  | | | |
| **1.1.3 Delete User Profile** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin selects to view a list of user profiles | | The system returns a list of all user profiles. |
| 2 | The admin selects to delete a user profile from a list of user profiles | | The system displays a confirmation dialogue window |
| 3 | The admin selects OK on the confirmation dialog window to confirm deleting the profile | | The system confirms the user is not coupled to any products and deletes the profile. The system returns message a success message on success or a failure message. The system should return a message indicating the profile could not be deleted because it still has some payment left, or orders or anything of this kind. In fact, it should be very rare to delete an user and should depend on how long the user was active or not. |
| **Postconditions** | | | |
| The user profile will be moved to another table which just keeps track of all deleted user just to avoid losing data or may be for future restoration of the account. | | | |
| **Business Rule** | | | |
| Only a user profile that is not coupled to any actions or transactions in the system may be deleted. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 6** | | | |
| **Name** | | Order CRUD | |
| **Brief description** | | This use case provides a functionality for managing orders. It should be very rare for the administrator to interfere in managing the orders. | |
| **Actors** | | Admin | |
|  | | | |
| **Preconditions** | | | |
| The admin must log in to the system | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Create Order** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin calls the create order command | | The system displays the order form with description, products, payment plan, etc.… |
| 2 | The Admin fills out the form and requests the system to save the details | | The system verifies there’s no missing information for the execution of the order then persists it to the database and returns a success message or a fail message in case of an exception. It should return a specific message when a duplicate order ID exists. |
| **Postconditions** | | | |
| The order is persisted in the system and is only visible to the specific buyer, the specific merchant and admin | | | |
| **Business Rule** | | | |
| No two orders can have the same order ID. | | | |
|  | | | |
| **1.1 Read/view Order** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin requests for a list of orders | | The system returns a list of all orders. |
| 2 | The admin selects to view an order they want from the list. | | The system returns the order object and display it to the user. |
|  | | | |
| **1.2 Update Order** | | | |
| 1 | The admin requests a list of orders | | The system returns a list of orders. |
| 2 | The admin selects the order they want to update | | The system displays an editable form with the order details |
| 3 | The admin changes the parts they want to update and requests system to save the new details | | The system updates the order and returns a success message or a fail message on exception. |
| **Postconditions** | | | |
| The order details will be updated | | | |
| **Business Rule** | | | |
| The order ID should be unwritable. The admin should not change anything from a totally completed order. The admin should not change an ongoing order unless cancel it and leave the change to the specific user. | | | |
|  | | | |
| **1.1.3 Delete Order** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin requests for a list of orders | | The system returns a list of all orders. |
| 2 | The admin selects to delete an order from a list of orders | | The system displays a confirmation dialogue window |
| 3 | The admin selects OK on the confirmation dialog window to confirm deleting the order | | The system confirms the order is not an ongoing and non-completed order then deletes and returns a success message or a fail message on exception. |
| **Postconditions** | | | |
| The order will be moved to another table for keeping track of all deleted orders for any future professional intent. | | | |
|  | | | |
| **Business Rule** | | | |
| The order must not be an ongoing and non-completed order. Admins should not interfere to any orders unless requested by specific user | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 7** | | | |
| **Name** | | User Approval | |
| **Brief description** | | This use case provides a functionality for approving user’s request of selling products | |
| **Actors** | | Admin | |
|  | | | |
| **Preconditions** | | | |
| The admin must log in to the system | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Approving User** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | The admin receives requests from users and click the review button to verify the product | | The system displays the user and the product, the price and number of piece available in stock. |
| 2 | The Admin accept or refuse the request | | If accepted, the information of the product is persisted to the database then the system sends acceptance notification to the user. Otherwise, the system sends refusal notification to the user with the reasons. |
| **Postconditions** | | | |
| The product is persisted in the system if approved. | | | |
|  | | | |
| **Business Rules** | | | |
| Any essential information about the product should not be missing. | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Number: 8 | | | |
| **Name** | | Place Order | |
| **Brief description** | | This use case provides a functionality for checkout after adding product to cart | |
| **Actors** | | Buyer | |
| Preconditions | | | |
| Buyer must login to system | | | |
| Flows of Events: | | | |
| 1. Basic Flows | | | |
| 1.0 Place Order | | | |
| Step | User Actions | | System Response |
| 1 | Buyer view product detail | | 1.1. System displaying product detail |
| 2 | Buyer input quantity and click “Add to Cart” button | | 2.1. System checking stock  2.2. System store to database |
| 3 | Buyer click “Checkout” button | | 3.1. System verify quantity before reducing stock  3.2. Check availability (Status) of the product  3.3. Reduce product quantity and update in database  3.4. System display success message, order successfully  3.5. System will be rendered to order detail |
| Postconditions | | | |
| Product stock will be reduced and render to order details. | | | |
| Business Rules | | | |
| Stock can’t be minus  Product Order Quantity should be validated  Non available product can’t be bought (products status) | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 9** | | | |
| **Name** | | Login | |
| **Brief description** | | This use case provides a functionality for longing to system | |
| **Actors** | | Admin, Buyer, and Merchant | |
|  | | | |
| **Preconditions** | | | |
| 1. User must be already registered in system | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Login** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | Buyer click login menu | | 1.1 System display login form |
| 2 | Buyer input username and password | | 2.1. If username and password correct as written in database, then will be succeed  2.2. if username and password mismatch along with database, will give error message  2.3. If username not found in database, will give error message “user not found”  2.4. if user leave one of form blank will give error message can’t empty |
| **Postconditions** | | | |
| User will be redirect to homepage | | | |
|  | | | |
| **Business Rules** | | | |
| * User only have 3 limit attempts if password wrong, in order to prevent attack | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 10** | | | |
| **Name** | | Register | |
| **Brief description** | | This use case provides a functionality for registering to system as merchant or buyer | |
| **Actors** | | Merchant, Buyer | |
|  | | | |
| **Preconditions** | | | |
|  | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.0 Register** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | User click register menu | | 1.1 System display register form |
| 2 | User input username, password, second password, email, address, phone number, user type (merchant/ buyer). | | * 1. System validate phone number, if not valid will give error message “phone number not valid”   2. System validate email, if not valid will give error message “email invalid”   3. System will check phone and email in database before registering, if already register then will give error message “email or phone already registered”   4. If one of field is empty will give error message can’t empty (required message)   5. If password and second password mismatch will give error message “password mismatch”   6. If all validation correct registration will be succeeded   7. Check password criteria |
| **Postconditions** | | | |
| User will get email; registration already succeed and email confirmation | | | |
|  | | | |
| **Business Rules** | | | |
| * User can’t register with same phone number or email address | | | |
| * “Use a minimum password length of 8 or more characters if permitted.” * “Include lowercase and uppercase alphabetic characters, numbers and symbols if permitted.” | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 11** | | | |
| **Name** | | Update Profile | |
| **Brief description** | | This use case provides a functionality for updating user profile | |
| **Actors** | | Buyer, Merchant and Admin | |
|  | | | |
| **Preconditions** | | | |
| 1. Buyer must be login | | | |
| **Flows of Events:** | | | |
| **1. Basic Flow** | | | |
| **1.0 Update Profile** | | | |
| **Step** | **User Actions** | | **System Actions** |
| 1 | User click menu update profile | | * 1. System will display update profile form |
| 2 | User update one more of their detail information | | 2.1. System check if the email already in database or not, if already will be reject  2.2. System will check phone, if already registered will be rejected  2.3. System will check if user leave one field empty, then will give error message “can’t empty”  2.4. If user want to change password, new password and second new password need to be match, otherwise will be error  2.5. if every validation pass, then will give success message  2.6. Check password criteria |
| **Postconditions** | | | |
| Detail information will be changed in form | | | |
|  | | | |
| **Business Rules** | | | |
| * User can’t change phone number or email if, already registered in system | | | |
| * “Use a minimum password length of 8 or more characters if permitted.” * “Include lowercase and uppercase alphabetic characters, numbers and symbols if permitted.” | | | |