

User Requirements

Emptio is a third-party e-commerce platform where merchants can sell products, shoppers can buy them, and advertisers can promote selected products. The platform also includes moderators and administrators responsible for maintaining service quality and enforcing platform rules. Emptio itself does not own or ship products. Payments are processed through Stripe, which creates transaction records stored in the system.

Shoppers

Shoppers can browse available products using the search feature or by exploring the main and listing pages. Each product includes a title, price, image, and description. Shoppers can either buy a product directly or add it to their cart for later checkout. The system prevents purchases that exceed available stock and notifies users if a product is unavailable. When a purchase is made, Stripe processes the payment and returns confirmation to Emptio. The corresponding transaction details are stored in the Transaction class, and the relevant merchant is informed to deliver the product to the shopper's shipping address. Shoppers can update their profile information and report products that appear fraudulent or inappropriate. Reported products are reviewed by merchant moderators.

Merchants

Merchants manage their inventory through a dedicated panel that allows them to add, edit, or remove products. Each listed item includes details such as title, price, description, and image. Merchants can also update their business information, including warehouse location and bank account details. Once a shopper purchases a product, the merchant receives notification and fulfills the delivery independently.

Advertisers

Advertisers create campaigns that promote selected products on shopper pages. When creating a campaign, the advertiser chooses its name and adds one or more products. Each product added to a campaign requires payment of a fixed advertisement fee. Payment is processed through Stripe, which generates a transaction link and stores the transaction information in the system. Advertisers can later add more products to an existing campaign by paying additional advertisement fees through new transactions linked to the original one. Campaigns can be viewed and managed from the advertiser's panel, where advertisers may update campaign details or

remove included products.

Moderators and Administrators

Emptio has two types of moderators: Support Moderators and Merchant Moderators.

Support Moderators handle user issues and manage support tickets, closing them when resolved.

Merchant Moderators oversee reported products and can remove items or restrict merchant accounts if they violate platform rules.

Each moderation type includes two permission levels:

Regular Moderators handle standard reports and support requests.

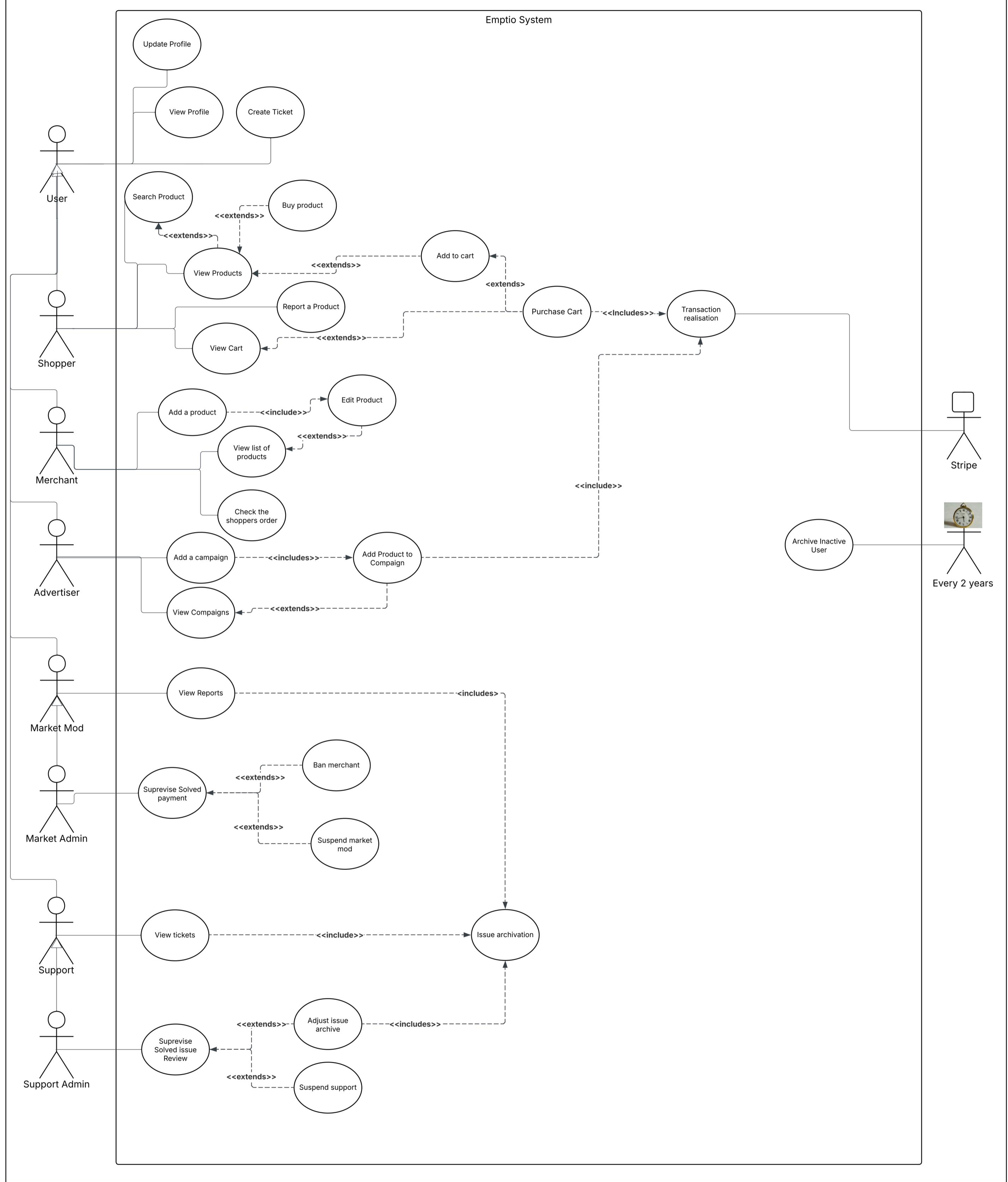
Admin Moderators serve as department heads who supervise their teams and can suspend members of their department.

Advertising and Display System

Products are displayed on shopper pages in a grid layout. Advertised products from active campaigns receive more visible placement, while regular items are shown according to availability

and category. The display logic ensures a fair mix of sponsored and regular listings while maintaining the platform's usability

Emptio UC diagram



Act...

Pur...

Ass...

Pre...

Bas...

1. T...

2. "...

3. T...

4. A...

5. A...

link...

Alte...

Pro...

The...

4a1...

Pos...

Log...

Stock...

Use Case Scenario -Add Product

Actor: Merchant

Purpose and Context: This use case allows the merchant to add new products to their inventory on Emptio. It enables merchants to expand their online catalog by entering product details such as name, description, price, category, quantity, and images.

Assumptions:

1. The Merchant has all the required product information (name, price, description, stock quantity).

Pre-Conditions:

1. The Merchant's account is verified and active

Initiating Business Event(Trigger): 1. The Merchant clicks on the "**Add Product**" option in the merchant panel.

Basic Flow of Events:

- 1.The Merchant press on the "**Add Product**" option in the merchant panel
- 2.The System displays the Edit Product page with a form to input product details
- 3.The Merchant enters all required product information - name, price, category, description, quantity, and upload images
- 4.The System validates the entered data
- 5.The Merchant clicks "**Submit**" button
6. The System saves the product details in the database and adds the product to the merchant's list
- 7.The System displays success message - "Product added successfully"
- 8.The Merchant views the new product in inventory list

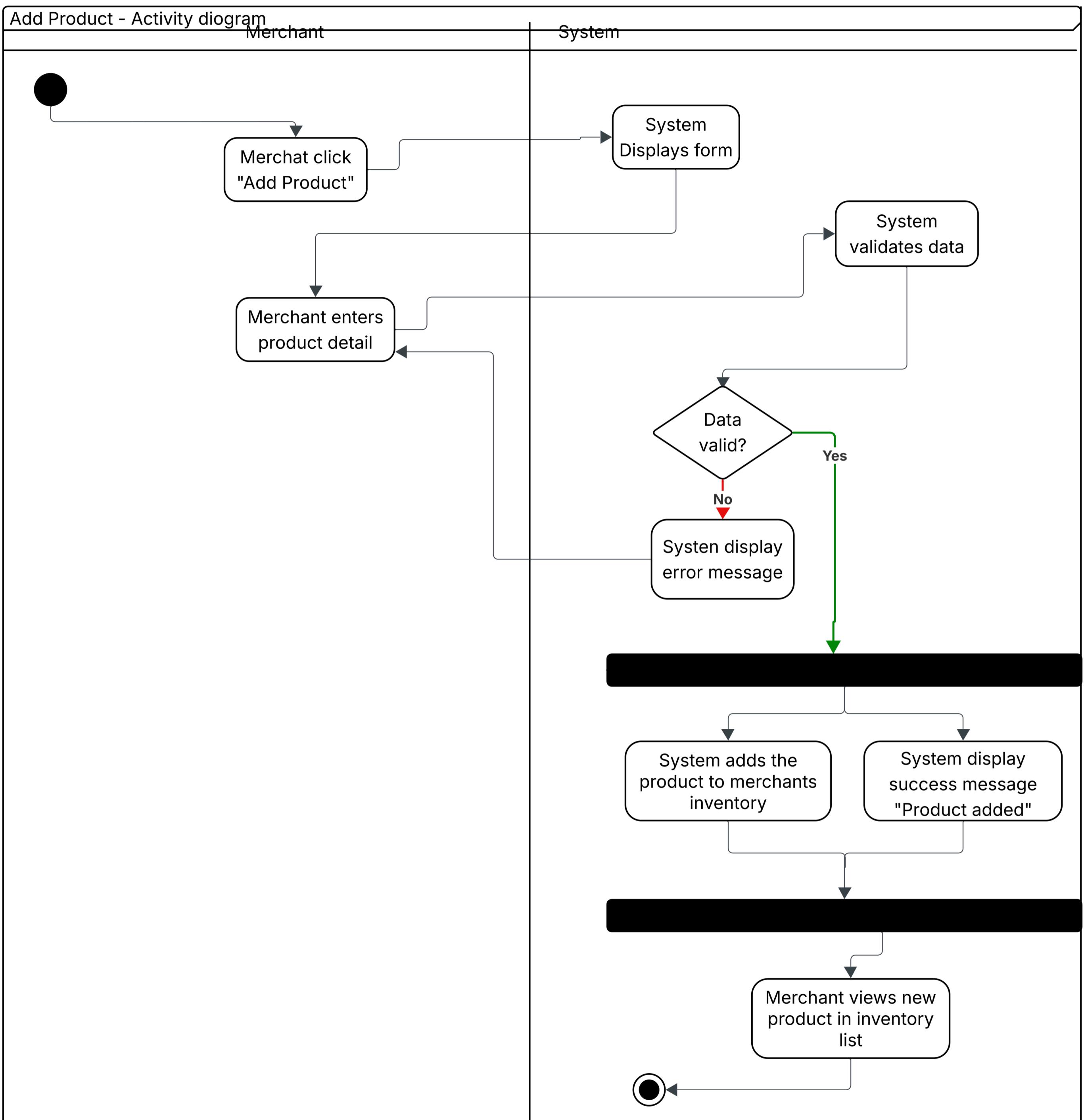
Alternative Flow of Events:

Missing Required Fields:

- 4A1.The System detects that one or more required fields are empty or contain invalid values
- 4A2.The System display an error message specifying which fields must be corrected.
- 4A3.The Merchant corrects the errors and resubmits the form

Post-Condition:

- 1.The new product is successfully added to the merchant's inventory and becomes visible in Inventory page
- 2.The system updates the product database with new entry



Use Case Scenario - Add Campaign

Actor: Advertiser

Purpose and Context: The use case is used by the advertiser to create a new advertising campaign. It allows the advertiser to define campaign details such as name, description, duration, and linked products, so that the campaign can be activated and managed within the system.

Assumptions:

- 1.The advertiser has the necessary permissions to create new campaigns.
- 2.The products intended to be included in the campaign already exist in the system.
- 3.The campaign name chosen by the advertiser is unique.

Pre-Conditions:

- 1.The advertiser has access to the campaign management panel.
- 2.The advertiser has at least one active product available to associate with the campaign.
3. The system has campaign creation functionality active and accessible

Initiating Business Event(Trigger): 1.The advertiser clicks on the “**Add Campaign**” button in the campaign management panel.

Basic Flow of Events:

- 1.The advertiser clicks on the “**Add Campaign**” button to start creating a new campaign.
- 2.The system displays the campaign creation form.
- 3.The advertiser enters the campaign name, description, start date, and end date.
- 4.The system records the entered information temporarily.
- 5.The advertiser selects one or more products to associate with the campaign.
- 6.The system displays a list of available products.
- 7.The advertiser confirms the selected products.
- 8.The system validates all entered campaign details and selected products.
- 9.The advertiser clicks “**Create**” to confirm the campaign creation.
- 10.The system creates a new campaign record linked to the advertiser and selected products.
- 11.The system displays a confirmation message indicating the campaign was successfully created.

Alternative Flow of Events:

Missing required information

- 3A1. The advertiser leaves one or more required fields (e.g., campaign name or dates) empty.
- 3A2. The system displays an error message prompting the advertiser to fill in all mandatory fields.
- 3A3. The advertiser completes the missing information.
- 3A4. The flow returns to step **4** of the basic flow

Invalid data range

- 8A1. During validation, the system detects that the campaign end date is earlier than the start date.
- 8A2. The system notifies the advertiser that the date range is invalid.
- 8A3. The advertiser corrects the dates.
- 8A4. The flow returns to step **8** of the basic flow.

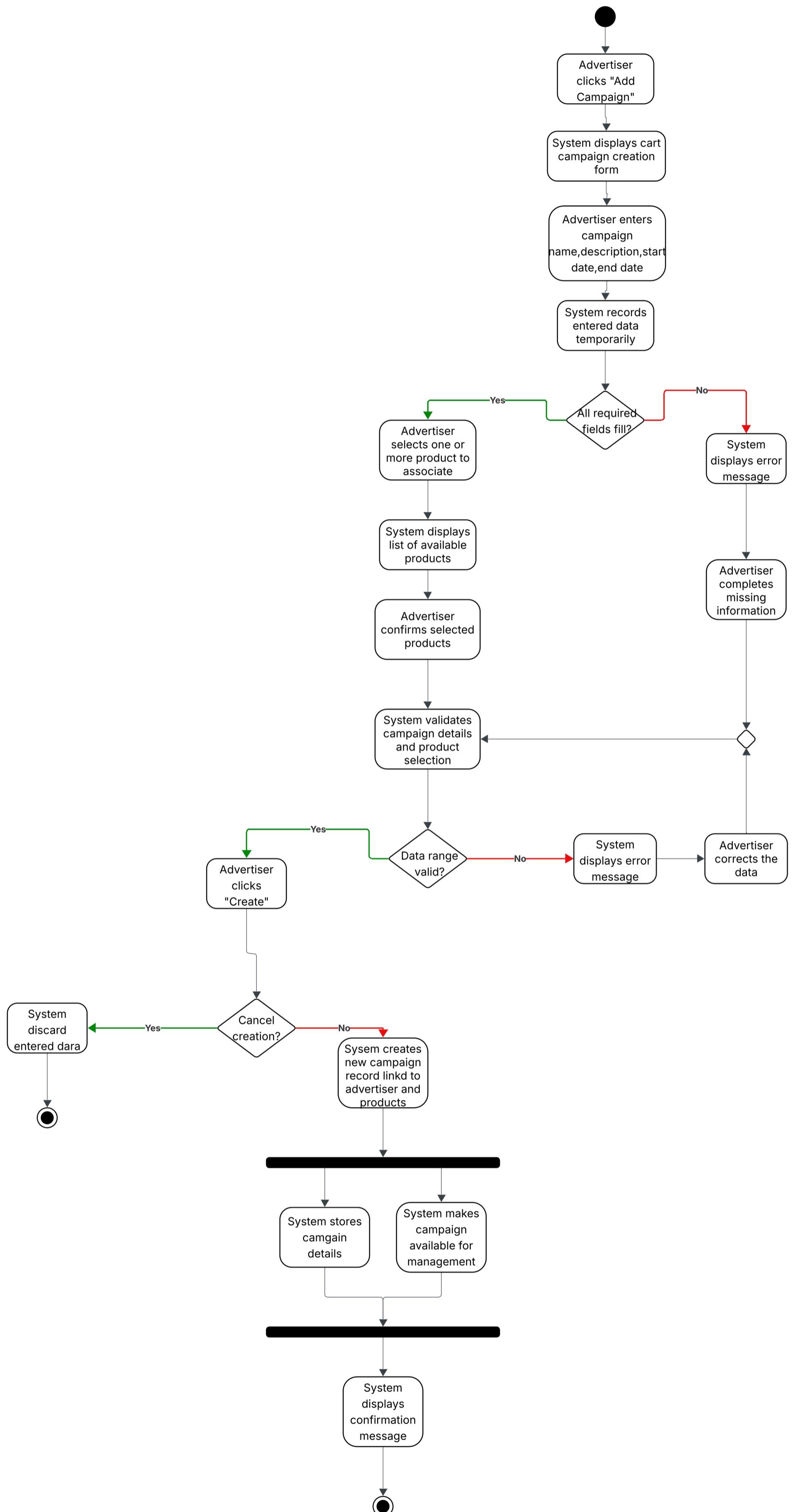
Campaign creation cancelled

Alternative scenario: Campaign creation canceled

- 9A1. The advertiser decides to cancel before confirming campaign creation.
- 9A2. The system discards all entered data.
- 9A3. The use case ends.

Post-condition:

- 1.The system creates a new campaign record linked to the advertiser and the selected products.
- 2.The system stores all campaign details, including name, description, and duration.
- 3.The campaign becomes available for further management or activation by the advertiser.
- 4.If the advertiser cancels the creation, no campaign record is created.



Use Case Scenario "Purchase Cart"

Actor: Shopper

Purpose and context: A shopper adds needed items to the cart and after completing that proceeds to payment and delivery options.

Assumption: 1. The user is satisfied with the selected cart items also with their number, prices

.

Precondition: 1. The user is viewing the cart page with the list of selected items for PURCHASE.
2. There should be at least one item in the cart

.

Basic flow of events:

1. The shopper clicks on "Proceed to payment" button
2. "Select delivery address" page
 - a. the shopper already has an information of delivery address in the user account and selects it
 - b. In the other case the shopper fills the form correctly in the page of delivery address
3. Then payment page is shown , where the shopper selects a preferred payment method (not by cash)
4. After selecting that payment method generates a link for a transaction
5. After a successful purchase the message about it is displayed,**the check is sent to the email** and a link to track the order status,also it clears the cart of the shopper

Alternative flow of events:

Product in the cart are unavailable

2b1. The system displays the list of such products and return to the main page

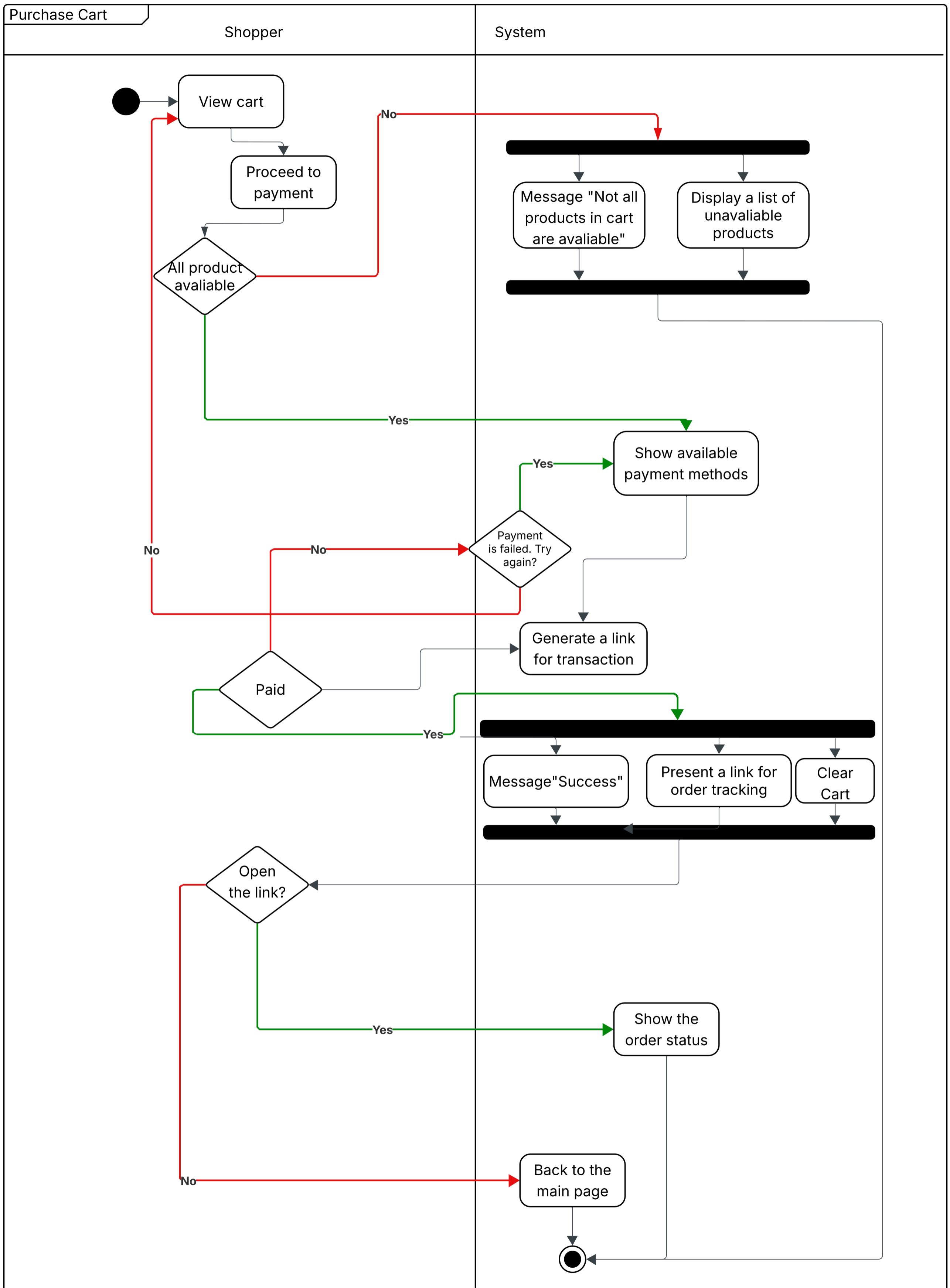
The payment is failed

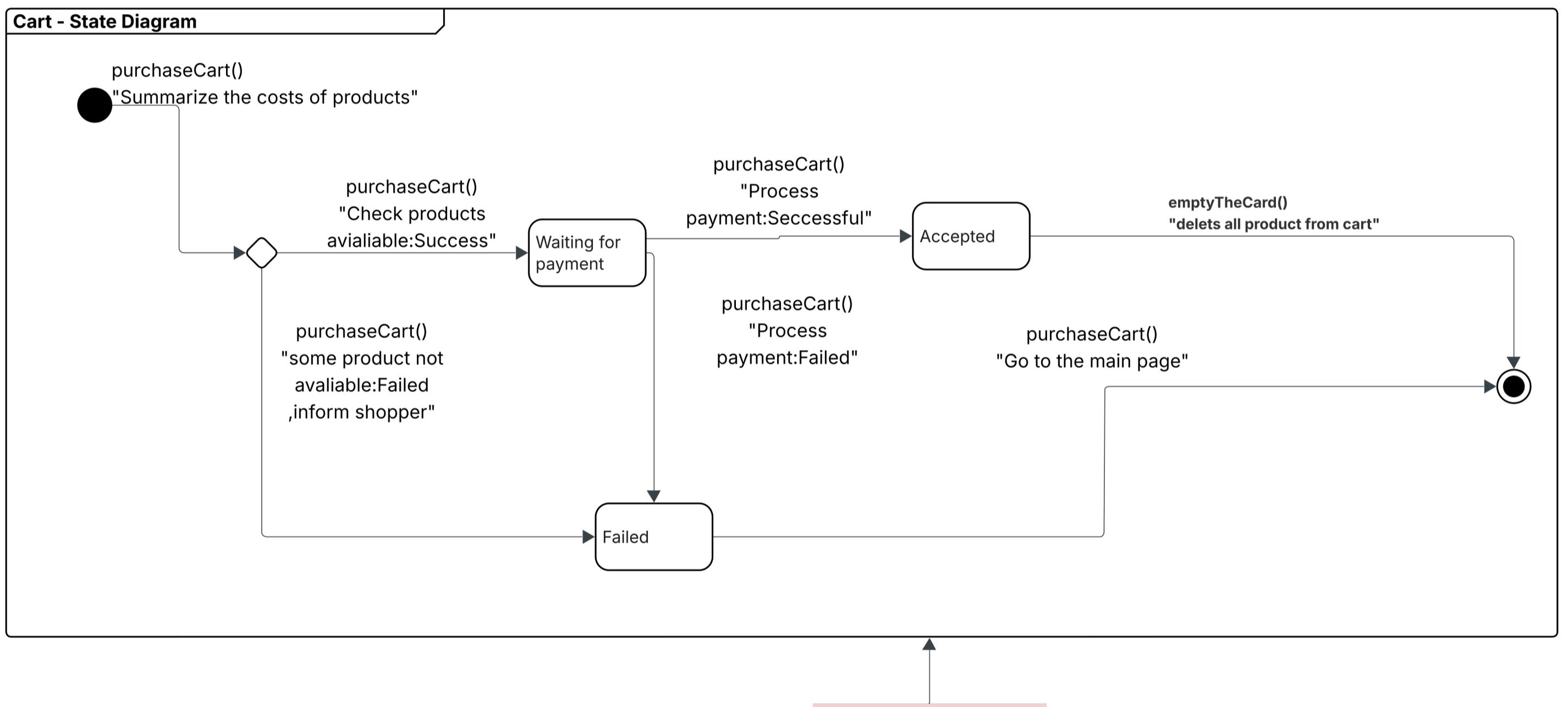
4a1. The system displays the info and a button "Retry", the transaction link is regenerated

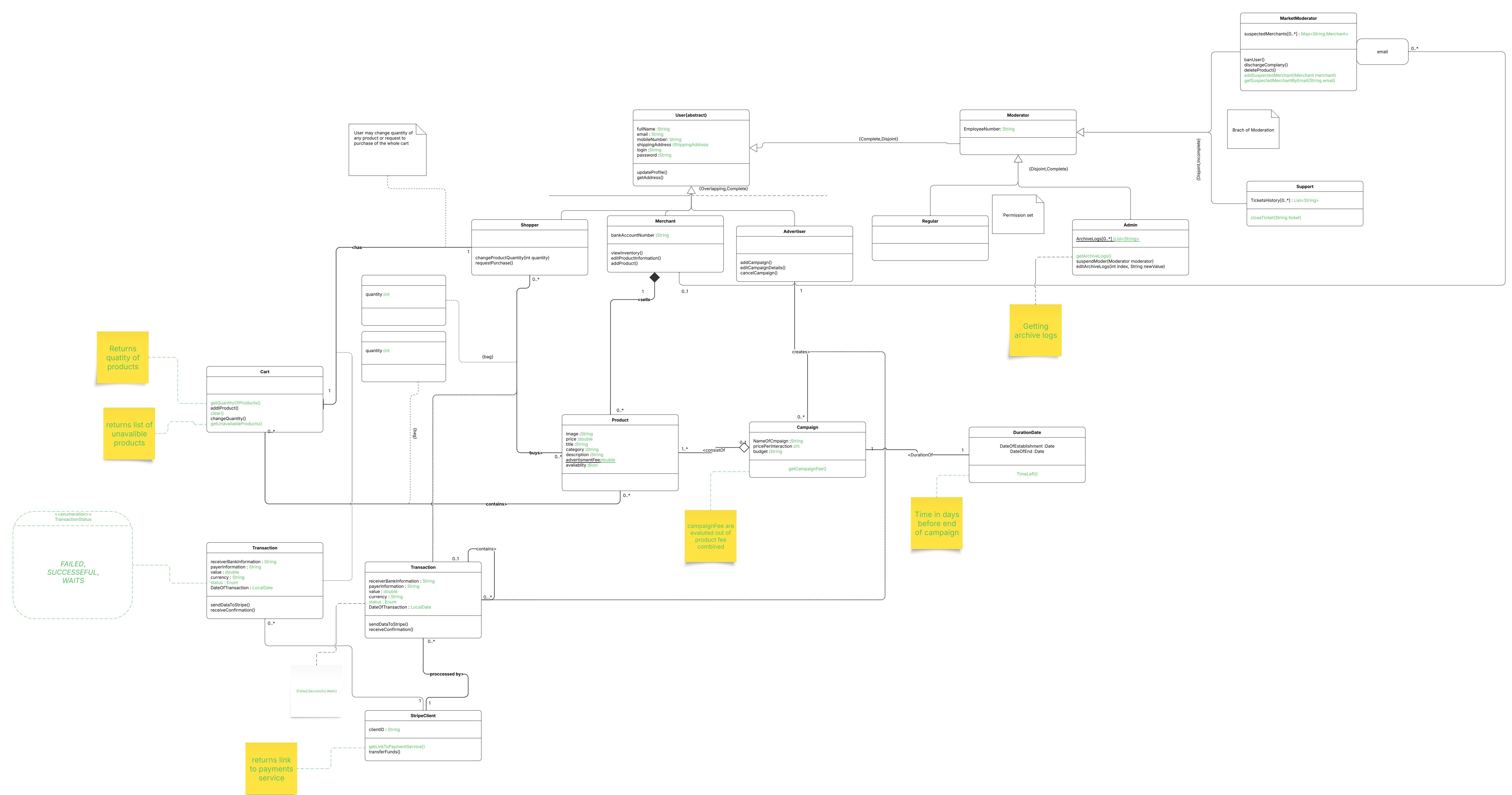
Post condition:

Log info: Update on history of transactions

Stock: Some items will be reduced or even out of stock







Dynamic Analysis on "Add Product","Add compaign","Purchase Cart" use cases & "Cart class" state diagram.

Following the above-presented diagrams, a dynamic analysis was performed. The result of this analysis indicated a need for further expansion on the design class diagram to encompass newly identified requirements and behaviors of the system. All introduced changes can be seen in the below presented final design class diagram.

"Cart class" state diagram.

1. First of all transaction might be failed so we need to add a status of transaction in transaction class
2. Card is cleared every time shopper use purchaseCart() method successfully so we need to add a method to clear the card so the shopper can cleared it if he wanted without buying the whole thing it might be a good method to make a money but not quite handy and fair.
3. Since in our card might be unavailable products we need to show and evaluate which products are unavailable in the card in do it dynamically so we gonna add field like /unavailableProducts do display it to the shopper in case he want buy a cart and his cart contain such unavailable products

"Add Product"

No comments its gonna work with implementation we have now at least i hope its gonna work.

"Add compaign"

1. Compaign now have validation date and date of declaration and date of end which is logical but with simple implementation that is unachievable we need to add class Duration class for associating it with the compaign since its gonna have field like dateOfDeclaration,dateOfEnd or smth similar and evaluating the gap between to know then compaign needed to be deleted

"Purchase Cart"

1 Have pretty much same problems as Cart class in general and they already described

/UnavalibaleProduct-its list of products which is currently in cart and have unvailable status.

Method clear() - clears out the cart out of products

Status in transaction is reffers to is transaction failed seccesful or waiting to be paid

DurationDate is new class explicitly for campaign its contain startDate and EndDate of the compaign also it evalute dimanicly how much time this compaign left in the future might be used for other implementations