# Software Requirements and Design Document

**for**

# Productify

**Prepared by**
**Abdullah Mehmood**
**Syed Kalbe Raza**
**Aneeq Ahmed**

**Data Smiths**

**27th November 2024**

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the software requirements for **Productify**, a unified product scraper and aggregator. This document outlines the functionalities, goals, and scope of the system, which automates data aggregation from multiple e-commerce platforms into a single user-friendly interface. The document also details the system's features, constraints, and objectives.

## 1.2 Product Scope

**Productify** is a web-based application designed to streamline the online shopping experience by aggregating product data from various e-commerce platforms like Amazon, AliExpress, and Newegg. Its primary purpose is to allow users to compare products, check availability, analyze reviews, and explore second-hand products across platforms, all within a single interface. The system will cater to a wide audience of online shoppers looking for convenience, efficiency, and better decision-making tools.

The application differentiates itself from existing tools by offering broader product coverage, automated review analysis, targeted recommendations, and seamless user interactions like posting and updating advertisements.

## 1.3 Title

***Productify*:** A Unified Product Scraper and Aggregator for Enhanced Online Shopping

## 1.4 Objectives

The key objectives of **Productify** are:

- To provide a centralized platform where users can view and compare products from multiple vendors.
- To automate the scraping of product data, ensuring users access the most up-to-date information.
- To include detailed insights from review analysis and tailored product recommendations.
- To simplify the search process for second-hand products by gathering them in one place.

- To provide user-friendly features like cart management, ad posting, and email notifications.

### 1.5 Problem Statement

Online shoppers often face the hassle of navigating multiple e-commerce platforms to find and compare products, which is time-consuming and inefficient. Existing price comparison tools are either limited to specific categories (e.g., mobile phones) or lack coverage across vendors.

**Productify** addresses this issue by automating product data aggregation and offering a centralized view of all relevant information. It saves users time and enhances the shopping experience by providing recommendations and insights tailored to user preferences. The system's feasibility lies in its use of modern technologies, including web scraping tools and frameworks like Spring Boot and Angular.

## 2. Overall Description

### 2.1 Product Perspective

**Productify** is a standalone platform that serves as a next-generation e-commerce aggregation tool. It integrates components like web scrapers for multiple platforms, a backend for data processing and user management, and a frontend for a seamless user experience.

The system connects directly to e-commerce platforms through automated web scraping techniques to fetch product details and reviews. The aggregated data is processed, stored, and presented to users in an intuitive manner. This project introduces features like targeted recommendations, email notifications, and ad management, which are not commonly found in traditional tools.

### 2.2 Product Functions

The primary functions of **Productify** include:

- **Product Aggregation**: Scraping data (e.g., price, features, availability) from multiple e-commerce platforms.
- **Review Analysis**: Automatically analyzing product reviews to provide insights.
- **Product Recommendations**: Offering tailored recommendations based on user preferences and cart contents.
- **Ad Posting and Management**: Allowing users to post, update, and delete advertisements.
- **Cart Management**: Facilitating users to add and manage products in their cart.

- **Email Notifications**: Sending updates about user activity and recommendations.

## 2.3 List of Use Cases

The use cases of Productify are listed below:

1. Scrape Data
2. Redirect User
3. Notify User
4. Register User
5. Display Products
6. Add to Cart
7. Manage Ads
8. Search Products
9. Analyze Interests
10. Analyze Reviews
11. Feature Ads

## 2.4 Extended Use Cases

### 1. Use Case: Scrape Data

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** System

**Stakeholders and Interests:**
- **Productify:** Wants to display products scraped from other websites.
- **Customer:** Wants to view comprehensive product details.

**Preconditions:**
- The system must have access to target websites.

**Postconditions:**
- Product data is scraped, processed, and stored in the database.
- Scraped data is displayed on the Productify platform.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. System initiates a request to target websites. | |
| | 2. System fetches product data from the website. |
| | 3. System processes and stores the data in the database. |

**Extensions:**

**2a.** The target website is down or cannot be reached.
**2a1.** System logs the error and retries later.

**Special Requirements:**

- **Performance**: The system should scrape data within a specified time frame to ensure freshness.
- **Security**: All scraped data must comply with legal regulations regarding data use.
- Data must comply with legal regulations regarding data use.

**Technology and Data Variations List:**

- The system must support variations in data formats (HTML, JSON, XML) when scraping from different websites.
- Adaptability to changes in the HTML structure of target websites.

2. **Use Case: Redirect User**

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** Customer

**Stakeholders and Interests:**

- **Customer:** Wants to purchase a product.
- **Third-party website:** Wants users to buy products from their platform.

**Preconditions:**

- **Customer:** Wants to purchase a product.

**Postconditions:**

- The user is redirected to the third-party website for purchase.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. Customer clicks the "Buy" button | |
| | 2. The system redirects the user to the third-party website for purchase. |

**Extensions**:

**1a.** The third-party website is unavailable.
**1a1.** The system displays an error and prompts the customer to try again later.

**Special Requirements:**

- **Reliability**: The redirection should be reliable and should maintain session integrity if applicable.

**Technology and Data Variations List:**

- Handle various URL structures from different retailers.

### 3. Use Case: Notify User

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** Customer

**Stakeholders and Interests:**

- **Customer:** Wants to receive timely updates on cart items, price changes, or new products.
- **Productify:** Wants to enhance user engagement.

**Preconditions:**

- The system has access to customer data and notification preferences.

**Postconditions:**

- Notifications are sent to the customer via the selected channel.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. System Identifies events for notification. | |
| | 2. The system prepares and sends notifications to the user. |

**Extensions:**

**1a.** The customer has wrong email address.
**1a1.** The system logs an error message that email couldn't be sent.

**Special Requirements:**

- Support for multiple notification channels (email, SMS, in-app).

**Technology and Data Variations List:**

- Email (using SMTP services like Gmail, SendGrid).

### 4. Use Case: Register User

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **New User:** wants to get updated about sales and other news.

**Preconditions:**

- The user must have an email to register.

**Postconditions:**

- The user is registered and can log into the system.

- User details are stored in the database.

| Actor Action | System Responsibility |
|---|---|
| 1. The user opens the registration page. | |
| 2. The user fills in the required fields in the form and submits. | |
| 3. The user clicks on the "Feature Ad" button. | |
| | 3. The system validates the input data. |
| | 4. The system checks if the user already exists. |
| | 6. The system creates a new record in the database. |

**Main Success Scenario:**

**Extensions:**

**1a.** The user enters incomplete or invalid data.
**1a1.** The system displays an error message and prompts the user to correct it.

**2b.** The email already exists in the database.
**2b1.** The system informs the user and prompts them to log in or recover their password.

**Special Requirements:**

- **Data Validation**: The system should validate the email format and ensure strong passwords during registration.

**Technology and Data Variations List:**

- Support for different email validation standards (e.g., different top-level domains).

5. **Use Case: Display Product**

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** Customer

**Stakeholders and Interests:**

- **Customer:** wants to view available products to purchase.
- **Productify:** wants to display relevant and updated products to users.

**Preconditions:**

- Product data must be available in the system database.

**Postconditions:**

- The product is displayed on the Productify platform.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. User opens the product page. | |
| | 2. The system loads products from the database. |
| | 3. The system displays the product details to the customer. |

**Extensions:**

**1a.** The product data is unavailable or incomplete.
**1a1.** The system displays an error or prompts the user to try again later.

**Special Requirements:**

- **User Interface**: The product display should be visually appealing and responsive.

**Technology and Data Variations List:**

- Support for various product data structures depending on the source website.

## 6. Use Case: Add to Cart

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** Customer

**Stakeholders and Interests:**

- **Customer:** Wants to add products to their shopping cart for later purchase.
- **Third-party website:** Wants customers to engage with their platform and make purchases.

**Preconditions:**

- The product is available on the Productify platform.

**Postconditions:**

- The product is added to the customer's shopping cart.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. Customer clicks the "Add to Cart" button. | |
| | 2. The system adds the product to the customer's shopping cart. |

**Extensions:**

**1a.** The product is out of stock.

**1a1.** The system displays an out-of-stock message and prevents the customer from adding the product.

**Special Requirements:**

- **Usability**: The "Add to Cart" functionality should be intuitive and clearly visible.

**Technology and Data Variations List:**

- Support for various product types and quantities in the cart.

## 7. Use Case: Manage Ads

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **Customer:** Wants to manage ads effectively, including creating, editing, and featuring ads.
- **Productify:** Wants to generate revenue through featured ads and ensure user satisfaction.

**Preconditions:**

- The user must be registered and logged into the system.
- Ad details (title, description, media files) must be provided correctly.

**Postconditions:**

- The ad is successfully created, updated, or featured.
- Featured ads appear prominently on the platform.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. The user selects the "Manage Ads" option. | |
| | 2. The System displays a list of the user's current ads. |
| 3. The user creates a new ad or edits an existing one. | |
| | 4. The system validates and saves the ad details. |
| 5. User selects an ad to feature. | |
| | 6. The system checks if the user meets the conditions for featuring the ad. |
| 7. The user confirms the action. | |
| | 8. The system updates the ad's status and displays it as featured. |

**Extensions:**

**1a.** User enters incomplete or invalid ad details.

**1a1.** The system displays an error message and prompts the user to correct the data.

**2a.** The user attempts to feature an ad without meeting conditions.
**2a1.** The system notifies the user of the prerequisites (e.g., minimum number of active ads).

**Special Requirements:**

- The interface must allow users to easily manage ads.

**Technology and Data Variations List:**

- Logs for failed uploads and fallback mechanisms for rendering errors.

## 8. Use Case: Search Products

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **Customer:** Wants to find relevant products quickly and easily..
- **Productify:** Aims to provide accurate and efficient search functionality to enhance user experience.

**Preconditions:**

- Product data is available in the system's database.
- The user is on the search page.

**Postconditions:**

- Search results matching the query are displayed to the user.
- Results can be filtered or sorted based on user preferences.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. The user enters a search query. | |
| | 2. The system processes the query using search algorithms. |
| 3. The customer applies filters or sorting options. | |
| | 4. The system updates the search results dynamically. |
| 5. The user views the relevant product list. | |
| | 6. The system displays the results in a user-friendly layout. |

**Extensions:**

**1a.** No matching results are found for the query.
**1a1.** The system displays a "No products found" message and suggests related queries.
**2a.** Customer enters a malformed or incomplete query.
**2a1.** The system provides real-time suggestions to correct the query.

**Special Requirements:**

- Support for autocomplete, filters, and sorting.

**Technology and Data Variations List:**

- Use Fuzzy matching and TFIDF matching for search results.

## 9. Use Case: Feature Ad

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **Productify:** Wants to gain more audience and ad revenue.

- **User:** Wants to quickly sell their products and feature them prominently over other ads.

**Preconditions:**

- The user must be registered and must have posted at least 5 ads.

**Postconditions:**

- The user's ad is featured on the first page, at the top of all other ads.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. The user selects an ad they want to feature. | |
| | 2. The system retrieves the details of the selected ad. |
| 3. The user clicks on the "Feature Ad" button. | |
| | 4. The system checks if the user meets the preconditions (registered, at least 5 ads). |
| 5. The user confirms the action. | |
| | 6. The system updates the ad display settings to show the featured ad prominently. |

**Extensions:**

**1a.** User does not meet preconditions
**1a1.** The system displays an error message indicating the requirement to have posted at least 5 ads.

**2a.** Ad already Featured
**2a1.** The system displays an error message indicating that the ad is already featured.

**Special Requirements:**

- The ad must be features for a specific timeline.

**Technology and Data Variations List:**

- Ad differs from other ads, as a featured label is shown next to it.

### 10. Use Case: Analyze Reviews

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **Productify:** Aims to enhance user experience and improve product offerings by analyzing customer feedback.
- **User:** Wants insights from reviews to understand product performance and improve decision-making.

**Preconditions:**

- The system must have access to product data and reviews from various websites via web scraping.
- The system must be able to process and analyze reviews using NLP techniques.

**Postconditions:**

- Users can see scraped product listings and corresponding analysis of reviews automatically displayed on the product page.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. The user browses the product listings. | |
| 2. The user clicks on a product. | |
| | 3. The system analyzes the reviews for sentiment and key themes. |
| | 4. The system generates an analysis report automatically without requiring user action. |
| | 5. The system displays the analyzed review data alongside the product details. |
| 6. User views the insights of reviews and make decisions accordingly. | |

**Extensions:**

**1a.** No reviews available for a product
**1a1.** The system displays that there are no reviews available.

**Special Requirements:**

- The interface must be user-friendly, making it easy for users to interpret the analysis alongside product information.

**Technology and Data Variations List:**

- Different web scraping tools and libraries (e.g., Beautiful Soup, Selenium).
- Various NLP libraries for sentiment analysis (e.g., spaCy, NLTK).

### 11. Use Case: Analyze Interests

**Scope:** Productify

**Level:** User Goal

**Primary Actor:** User

**Stakeholders and Interests:**

- **Productify:** Aims to use data-driven insights to better understand user preferences and improve product recommendations.
- **User:** Wants personalized product suggestions based on their browsing history, review data, and preferences without manual input.

**Preconditions:**

- The system must have access to product data, user interaction data, and cart contents.
- The system must be capable of processing and analyzing interests using NLP or other analytical techniques.

**Postconditions:**

Users receive two types of recommendations:

- **Product-Specific Recommendations:** Displayed on the product page based on the current product being viewed.
- **Personalized Homepage Recommendations:** Displayed on the homepage based on items the user has in their cart.

**Main Success Scenario:**

| Actor Action | System Responsibility |
|---|---|
| 1. The user visits a product page. | |
| | 2. The system displays the product's details along with product-specific recommendations. |
| 3. The user adds products to their cart. | |
| | 4. The system tracks the items added to the cart and stores this data for further analysis. |
| 5. The user visits the homepage. | |
| | 6. The system displays recommendations based on the user's cart contents. |

**Extensions:**

**1a.** No related products for a specific product page:
**1a1.** The system displays a message indicating that there are no relevant recommendations for this product.

**2a.** Cart is empty:
**2a1.** The system generates product recommendations based on the user's past cart history, otherwise, displays "No recommendations"

**3a.** Error in analyzing cart or product data:
**3a1.** If there is a technical issue during interest analysis, the system logs the error and may display default or generic recommendations.

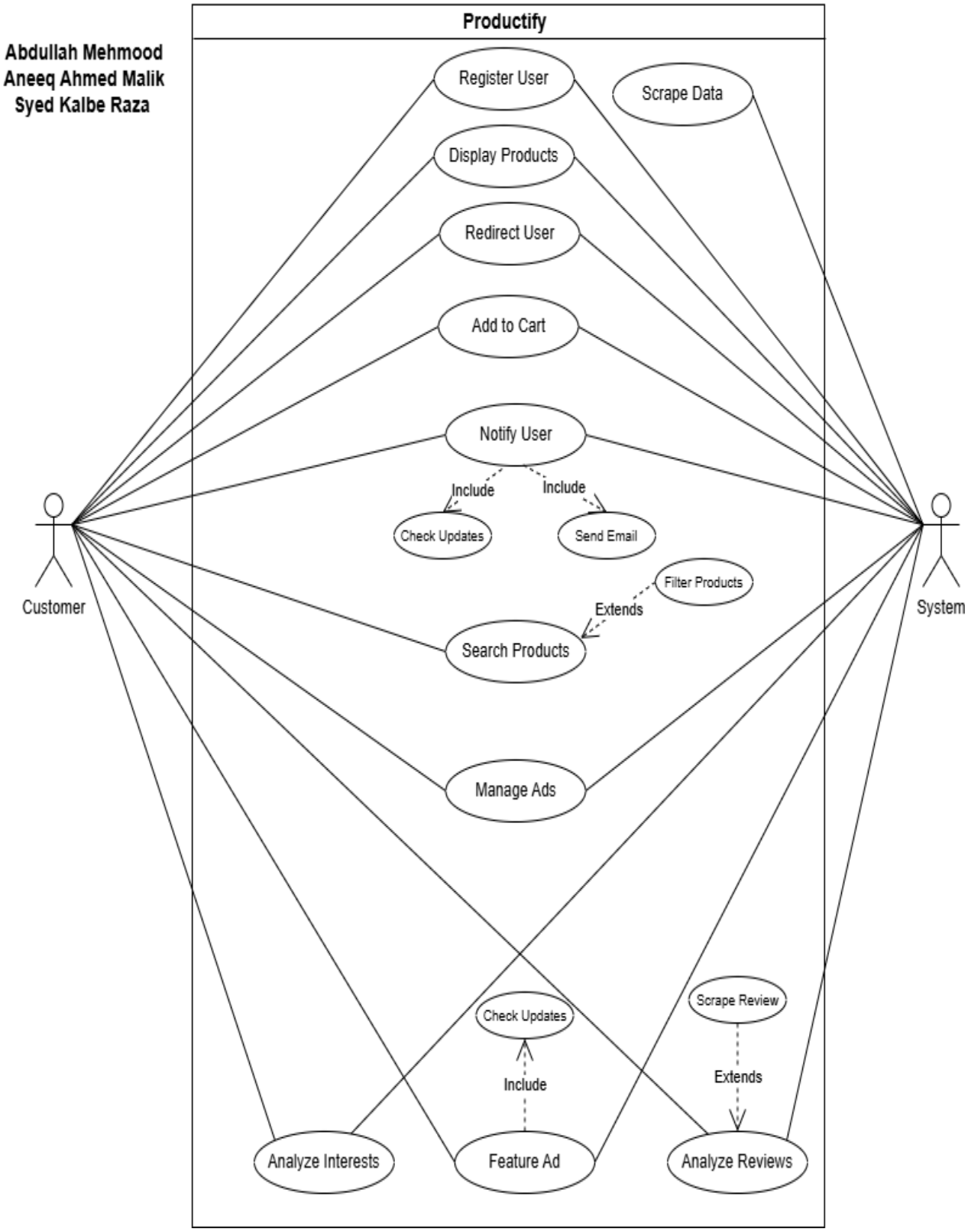
**Special Requirements:**

- The product page recommendations should be contextually relevant and personalized to the current product being viewed.
- The homepage recommendations should dynamically reflect the user's preferences based on cart contents.

**Technology and Data Variations List:**

- Different product types and categories that influence recommendations (e.g., electronics, clothing).

## 2.5 Use Case Diagram



# 3. Other Nonfunctional Requirements

## 3.1 Performance Requirements

- The system must respond to user actions (e.g., adding to cart, navigating, or searching) within **2 seconds**.
- Web scraping operations should be completed within **4 hours** for 1000 products to ensure data freshness.
- Recommendations and analytics should generate results in **real-time** (<1 second).
- System should handle **simultaneous access** by up to 10,000 users without performance degradation.

## 3.2 Safety Requirements

- Ensure safe handling of sensitive user information such as passwords and personal details to avoid leaks.
- Any failed transaction during ad-feature payment will not deduct the amount; the user will be notified.
- Follow GDPR and similar data protection regulations to avoid misuse of user data.
- Prevent unintended data deletion by requiring confirmations before any permanent action (e.g., deleting ads).

## 3.3 Security Requirements

- **Authentication:** All user actions must require authentication using a secure login mechanism.
- Passwords must be stored using encryption (e.g., salted hashing).
- Only registered users can create or manage ads.
- **Data Protection:** Encrypt user data in transit (HTTPS) and at rest.
- Regular security audits will be conducted to identify vulnerabilities.
- Compliance with security standards such as ISO/IEC 27001.

## 3.4 Software Quality Attributes

- **Adaptability:** The system should easily accommodate new features such as additional analytics or ad types.
- **Availability:** The platform must be operational **99.9% of the time**, including during peak hours.
- **Reliability:** All features, especially transactions and redirection, must perform consistently without failure.
- **Usability:** Interfaces must be intuitive, with clear navigation and accessible designs (WCAG 2.1 compliance).
- **Maintainability:** Codebase should follow industry best practices (e.g., modular design) for easy updates.
- **Testability:** Automated tests must cover at least **90%** of the codebase to ensure robustness.

## 3.5 Business Rules

- **Ad Management Rules:** Only users who meet predefined conditions (e.g., 5 active ads) can feature ads.
- **Search Rules:** Search results must align with user queries while adhering to relevance-based ranking.
- **Privacy Rules:** Users must explicitly opt in for email notifications and marketing content.
- **Recommendation Rules:** Recommendations should only consider products available in the database.

## 3.6 Operating Environment

- **Operating Systems:** Compatible with Windows, macOS, and Linux.
- **Browser Support:** Chrome (latest), Firefox (latest), Edge, and Safari.
- **Additional Software:** Database (e.g., MySQL/PostgreSQL), web servers (e.g., Nginx or Apache).

- ***Software Requirements***

  - ***Functional Software Requirements***

1. Database Management System:
   - MySQL or PostgreSQL for storing scraped product data, user accounts, and transactional details.

2. Web Scraping Tools:
   - **Selenium** for dynamic page interaction and JavaScript-heavy sites.
   - **BeautifulSoup** for parsing HTML and extracting static content.

3. Natural Language Processing (NLP):
   - **spaCy**, **NLTK openNLP** or **StanforfNLP** for sentiment analysis.

4. Recommendation System Frameworks:
   - **TFIDF Model** for building recommendation models.

5. Web Frameworks:
   - **Spring Boot** (Java) for backend development.

6. Browser Driver:
   - Latest **ChromeDriver** for Selenium.

  - ***Nonfunctional Software Requirements***

1. APIs and Libraries:
   - APIs for integrating with third-party websites (e.g., eBay, Amazon, etc.).
   - RESTful API for internal communication between the frontend and backend.

2. Authentication and Authorization:
   - OAuth 2.0 or custom authentication mechanisms for secure user access.

3. File Management:
   - **CSV** support for exporting scraped data and analytics reports.
   - File upload/download functionality for user profile pictures or ad attachments.

4. UI Development:
   - **ReactJS**, **VueJS**, or **Thymeleaf** for building responsive frontend pages.

- ***Additional Requirements***

1. Data Formats:
   - Support for **HTML**, **JSON**, and **XML** when scraping data from diverse sources.
   - Exporting data as **CSV** or **Excel** files for offline access or analysis.

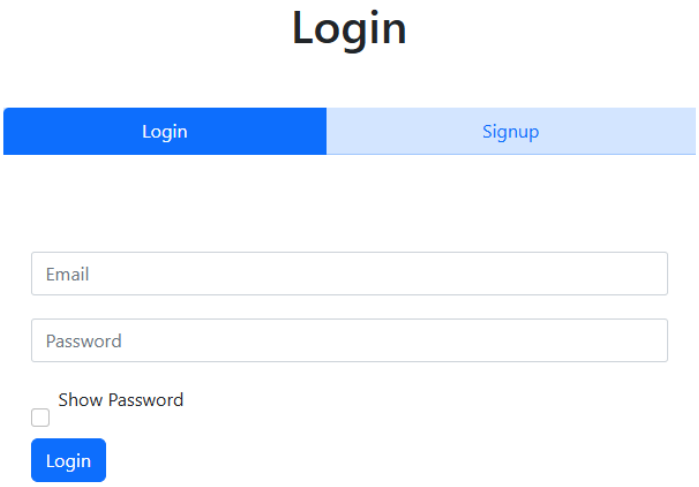
2. NLP Requirements:
   - Sentiment analysis and key theme detection for reviews using spaCy/NLTK.
   - Pre-trained models to improve the accuracy of recommendations and review analysis.

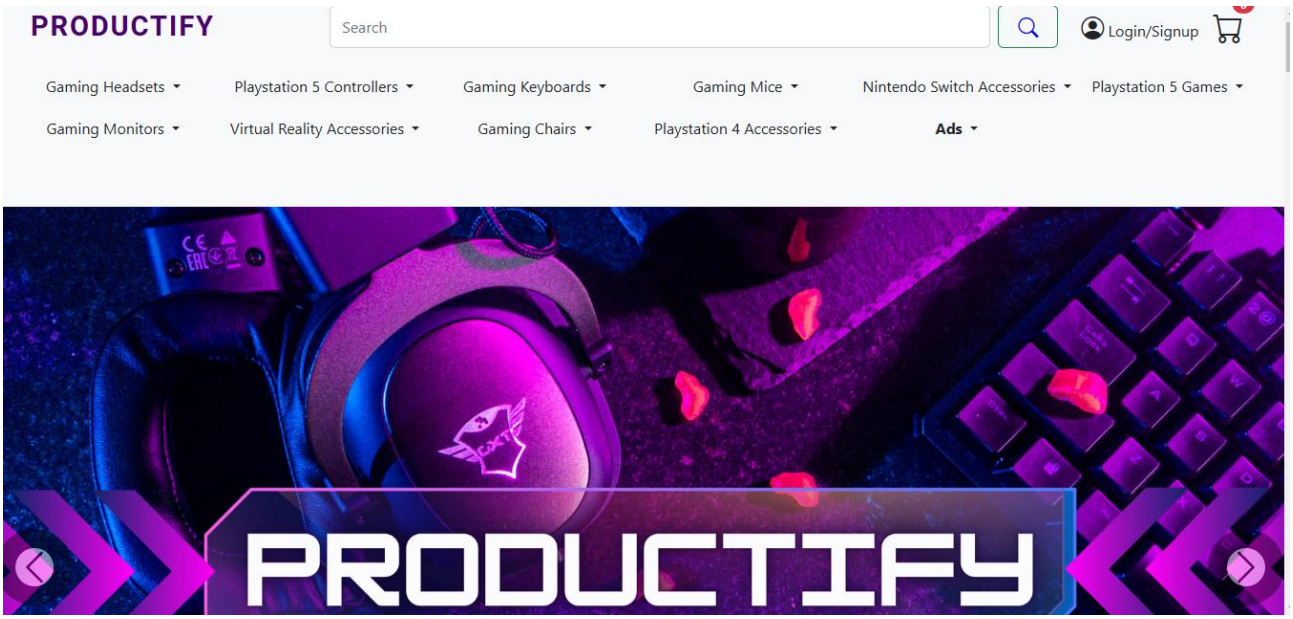
3. Email and Notification System:
    - Integration with email services (e.g., SendGrid) for sending notifications to users.

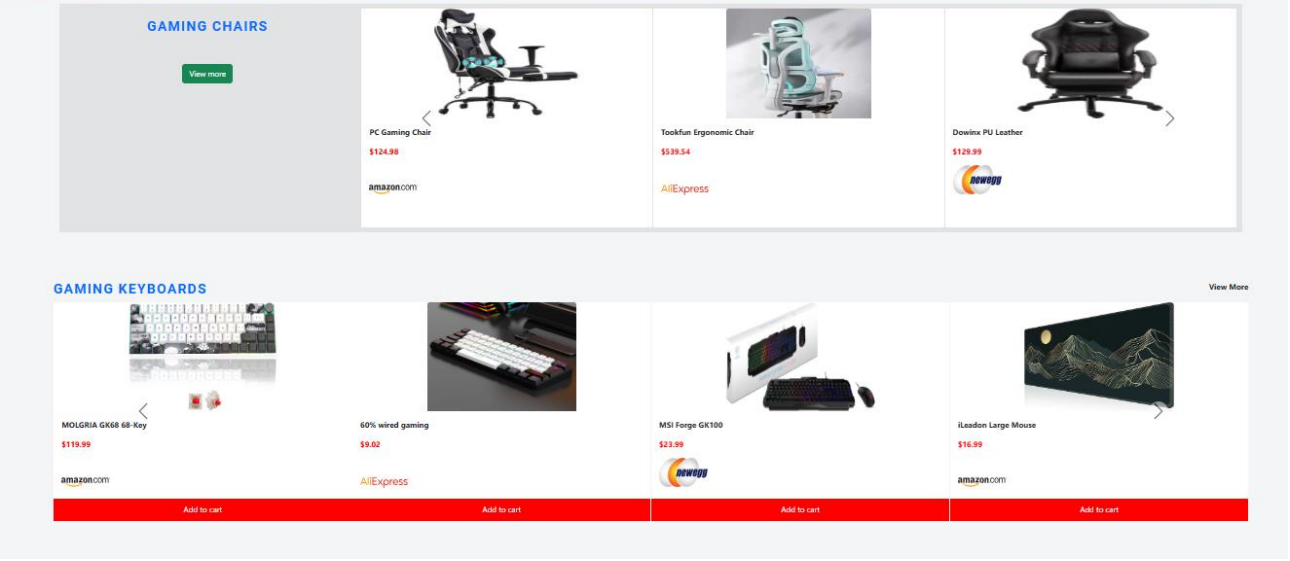
4. Testing and Debugging:
    - Tools like **Postman** to verify API functionalities.

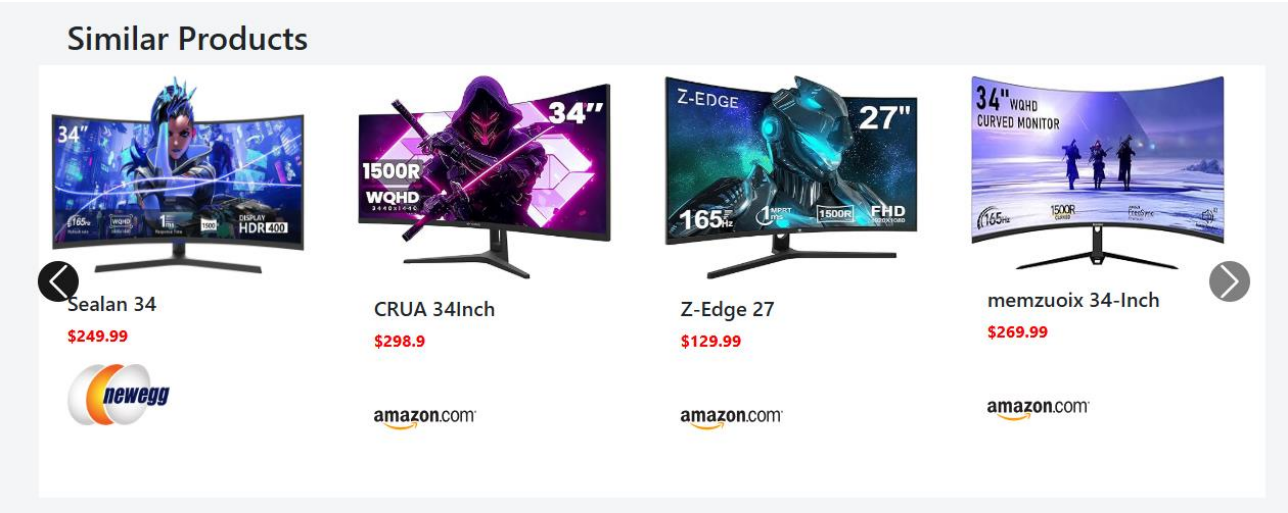
5. Hosting and Deployment:
    - **Heroku**, **AWS**, or **Google Cloud Platform** for deploying web applications.
    - Docker containerization for consistent environment setups.

## 3.7 User Interfaces

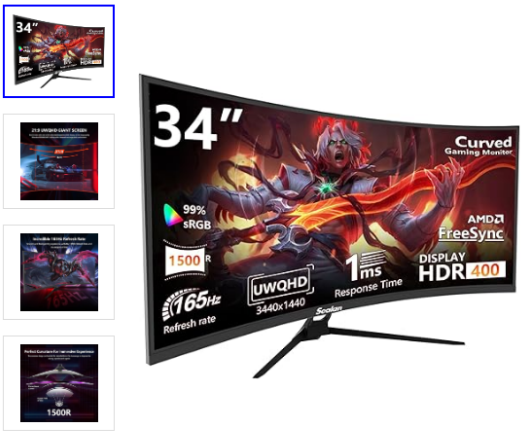



GAMING CHAIRS
View more
PC Gaming Chair
$124.98
amazon.com
Tookfun Ergonomic Chair
$539.54
AliExpress
Dowinx PU Leather
$129.99
newegg
GAMING KEYBOARDS
View More
MOLGRIA GK68 68-Key
$119.99
amazon.com
Add to cart
60% wired gaming
$9.02
AliExpress
Add to cart
MSI Forge GK100
$23.99
newegg
Add to cart
iLeadon Large Mouse
$16.99
amazon.com
Add to cart


Similar Products
Sealan 34
$249.99
newegg
CRUA 34Inch
$298.9
amazon.com
Z-Edge 27
$129.99
amazon.com
memzuoix 34-Inch
$269.99
amazon.com

## Product Details




amazon.com

**34 inch Ultrawide Curved Computer Gaming Monitor 165Hz UWQHD 3440x14401500R -PIP1ms (MPRT)HDR 300nits sRGB99%FreeSyncHDMI2.0(TMDS) x2DP1.4 x2Eye CareSupport Wallmount 75x75mm**

price : 199 $

4.1

Add to Cart | Buy Now

Description

### Sentiment Analysis

**Positive**

Playstation 5 Games


Resident Evil 3

$29.99

amazon.com

Add To Cart


Grand Theft Auto

$38.51

AliExpress

Add To Cart


Call of Duty

$69.99

newegg

Add To Cart


Avatar: Frontiers of

$58

amazon.com

Add To Cart

Sony Playstation 5

$89.98

AliExpress

Add To Cart

PlayStation 5 Digital

$699.99

newegg

Add To Cart

The Rumble Fish

$32.66

amazon.com

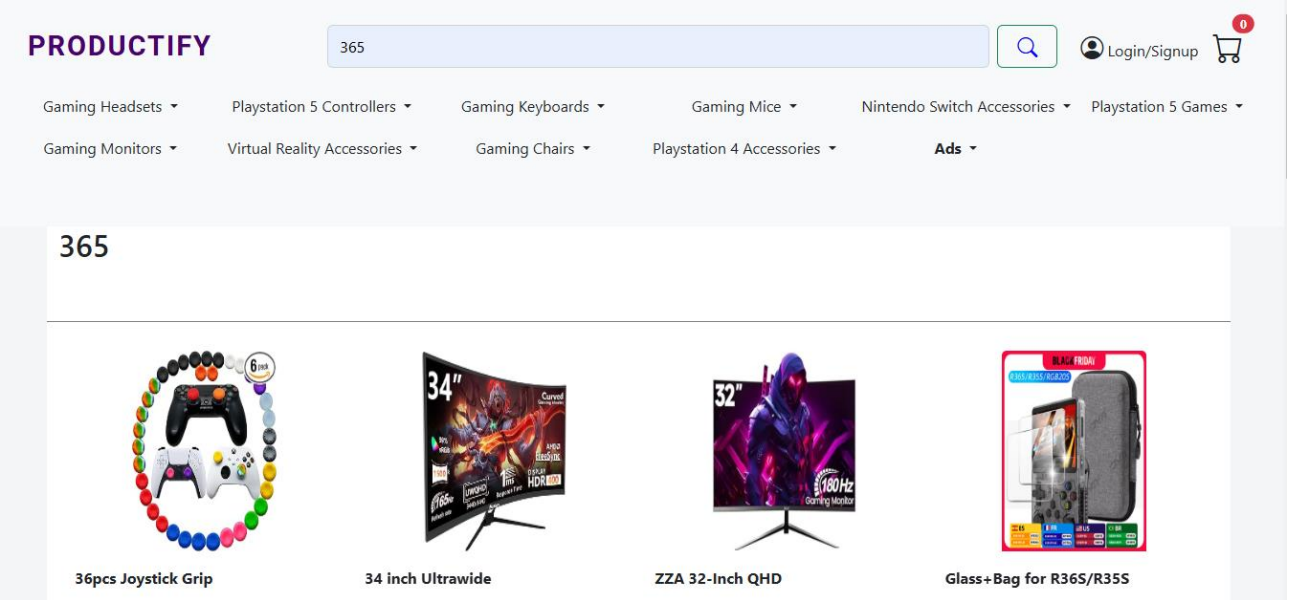
Add To Cart

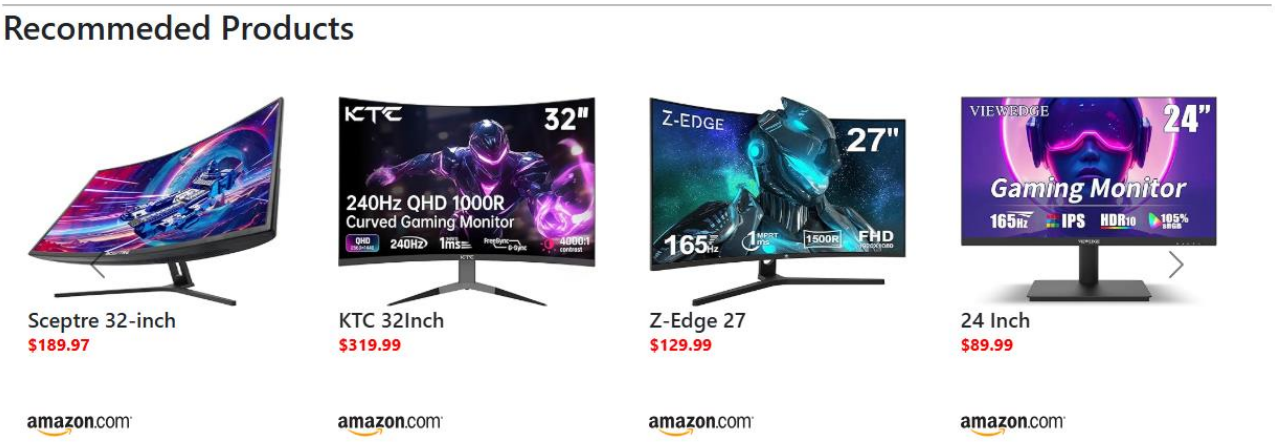
Kinhank Super Console

$154.73

AliExpress

Add To Cart


PRODUCTIFY
365
Login/Signup
0
Gaming Headsets
Playstation 5 Controllers
Gaming Keyboards
Gaming Mice
Nintendo Switch Accessories
Playstation 5 Games
Gaming Monitors
Virtual Reality Accessories
Gaming Chairs
Playstation 4 Accessories
Ads
365
36pcs Joystick Grip
34 inch Ultrawide
ZZA 32-Inch QHD
Glass+Bag for R36S/R35S


Recommeded Products
Sceptre 32-inch
$189.97
amazon.com
KTC 32Inch
$319.99
amazon.com
Z-Edge 27
$129.99
amazon.com
24 Inch
$89.99
amazon.com

## POST YOUR AD

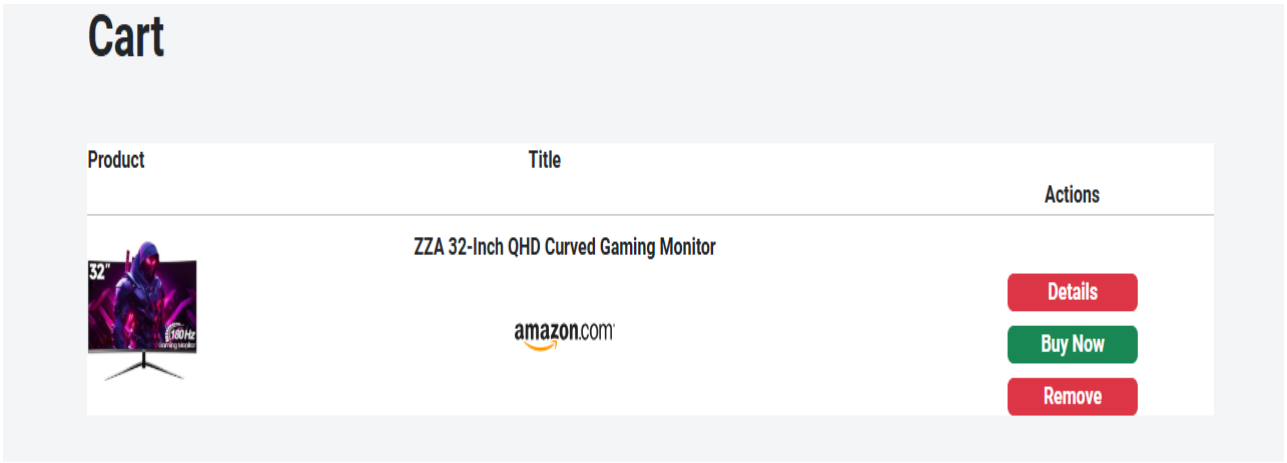
INCLUDE SOME DETAILS

Add Title

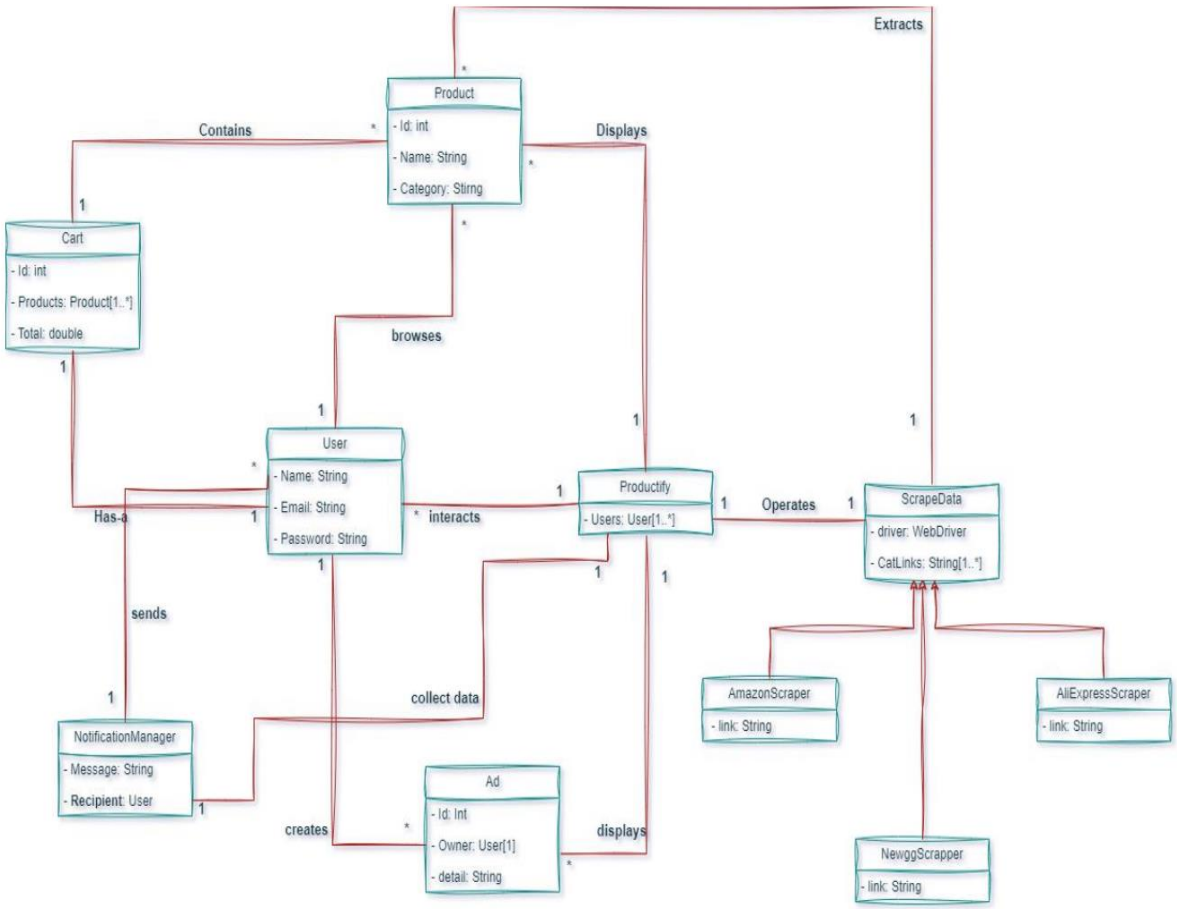
Description

## ADS

| Product | Title | Price | Actions |
|---|---|---|---|
|  | corolla | 1200000 | See Details |

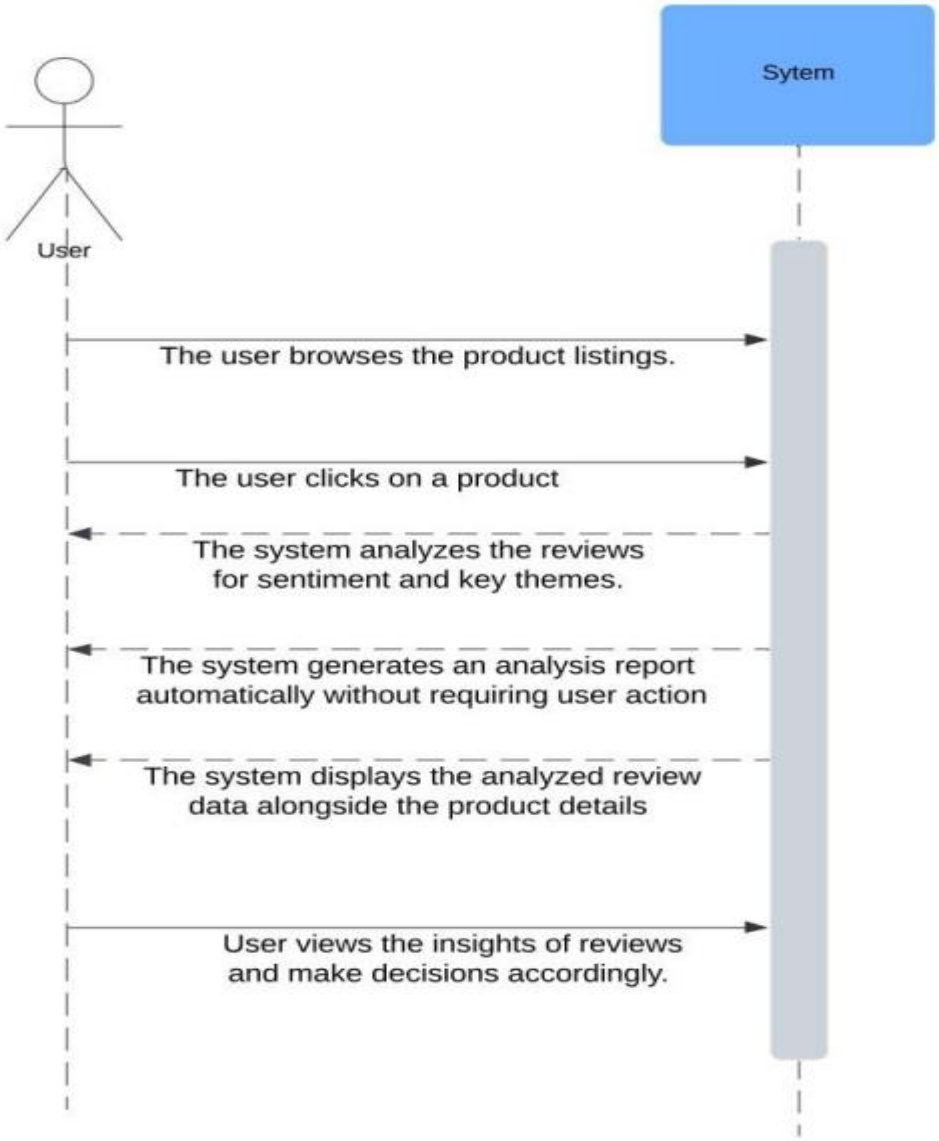


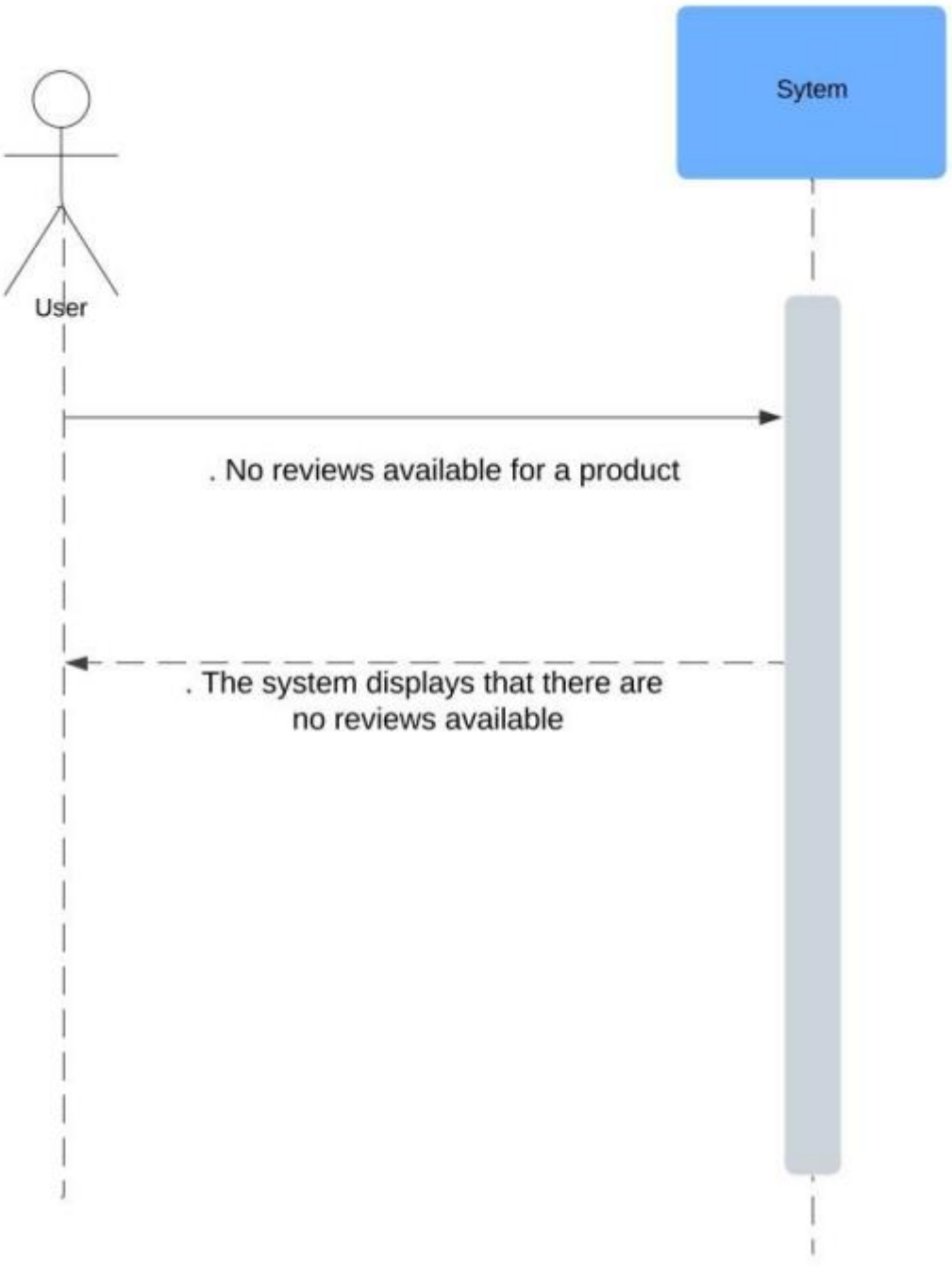

# 4. Domain Model



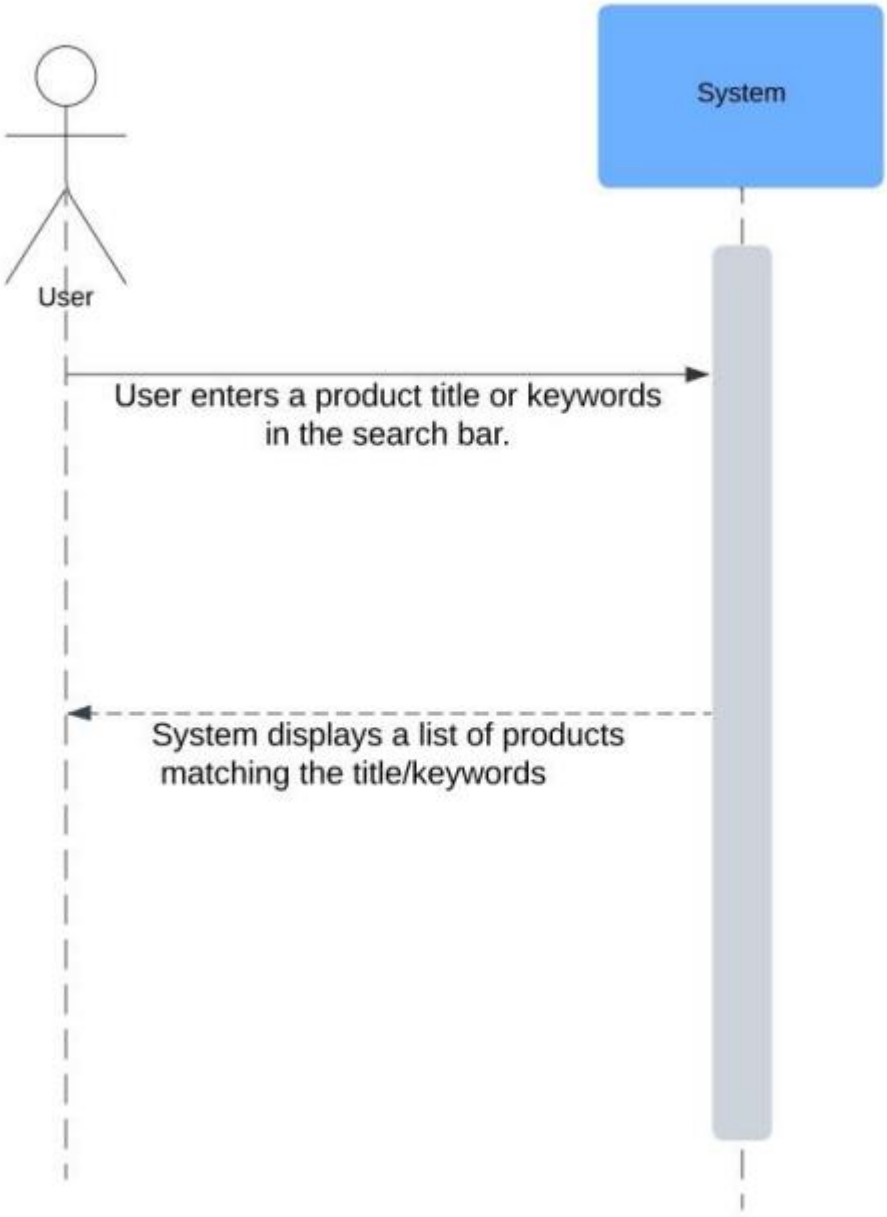
# 5. System Sequence Diagram

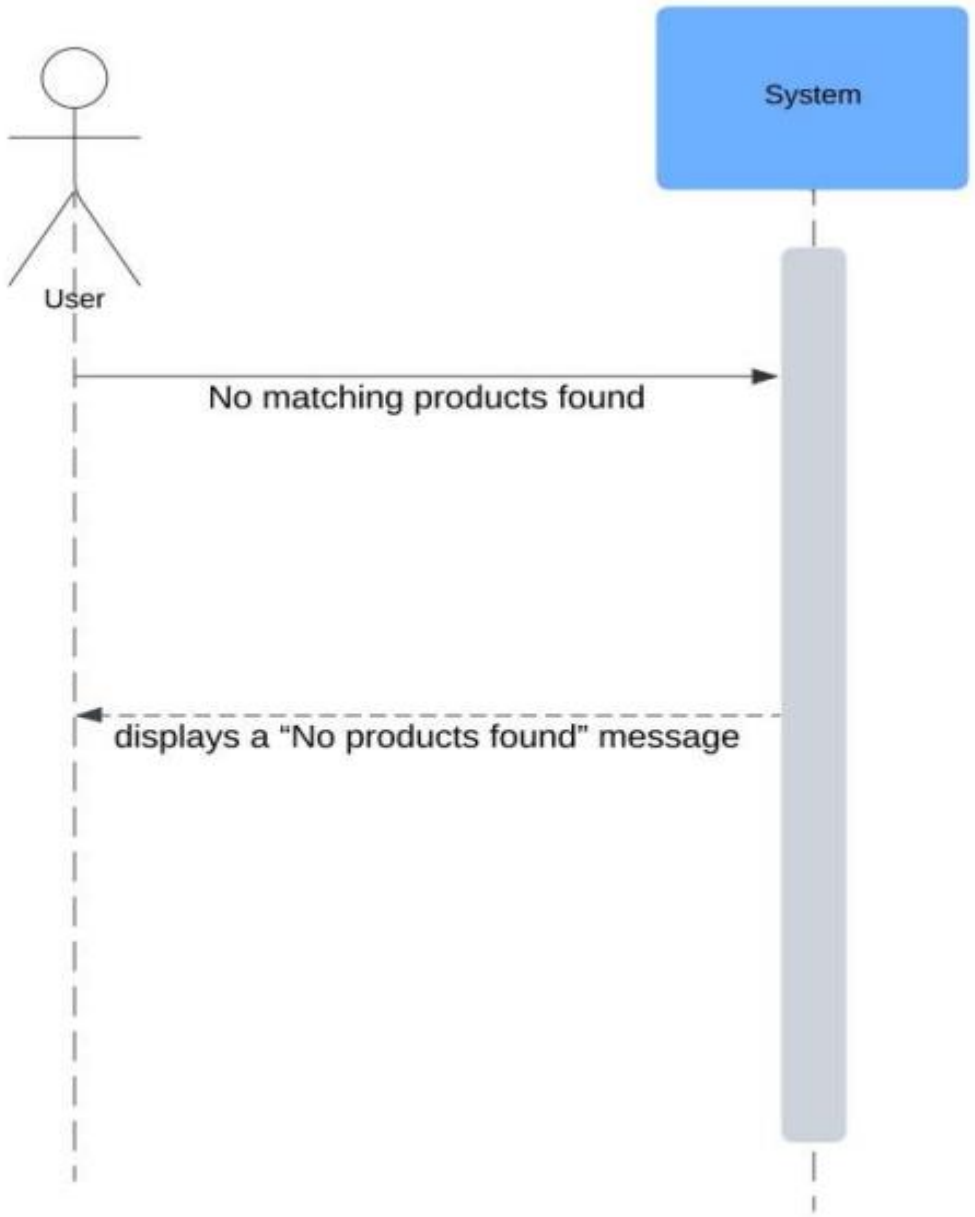
1) Analyze Reviews



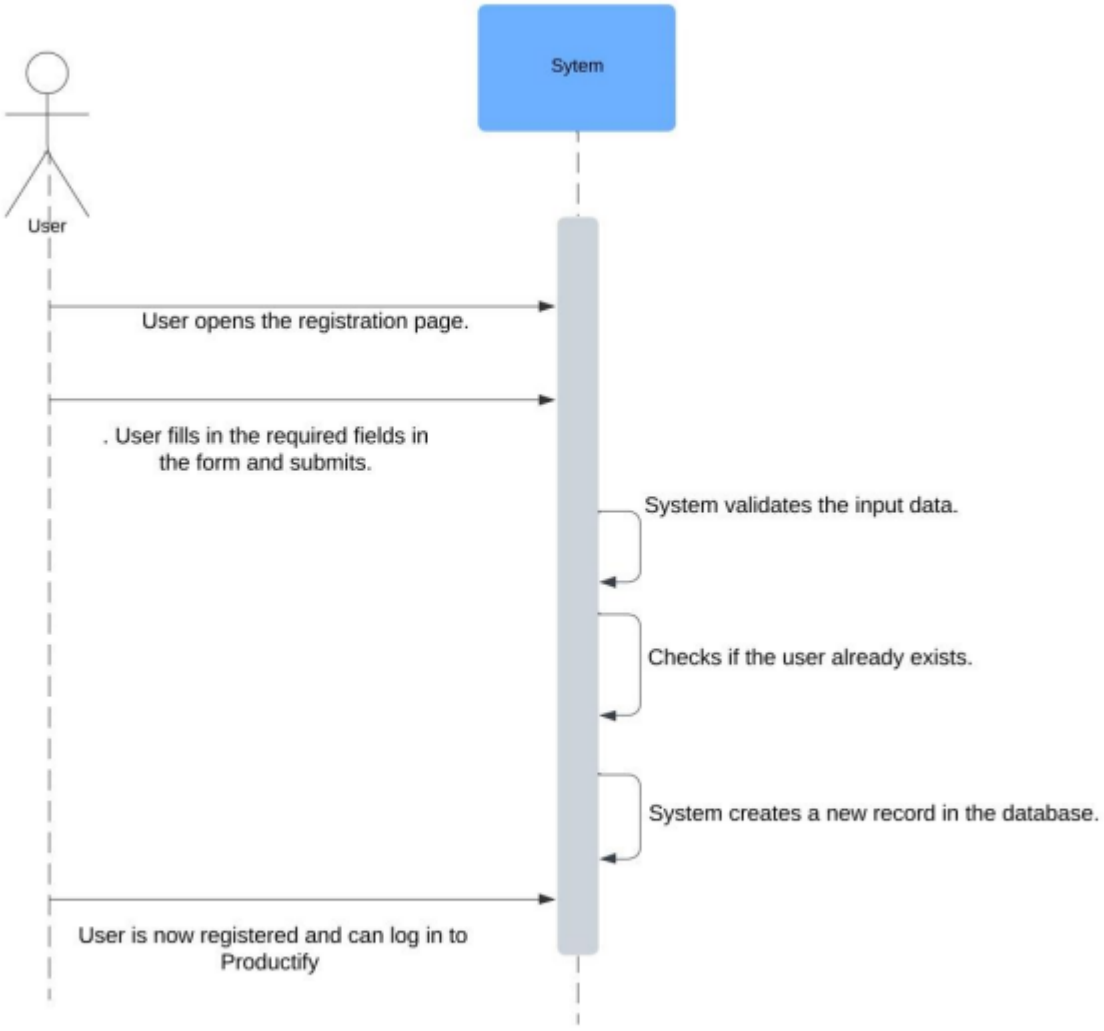

Sytem
User
. No reviews available for a product
. The system displays that there are
no reviews available

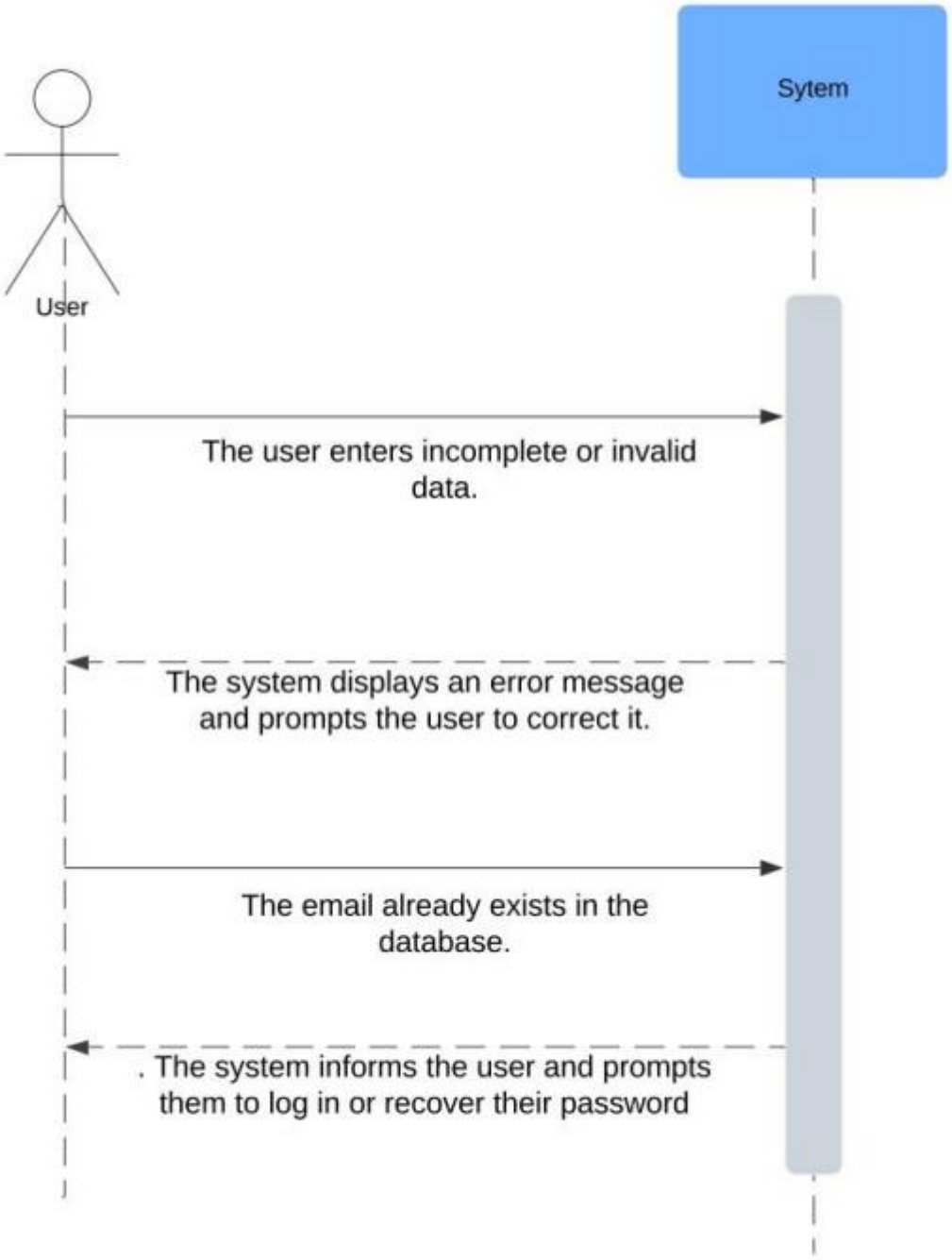
2) Search Products



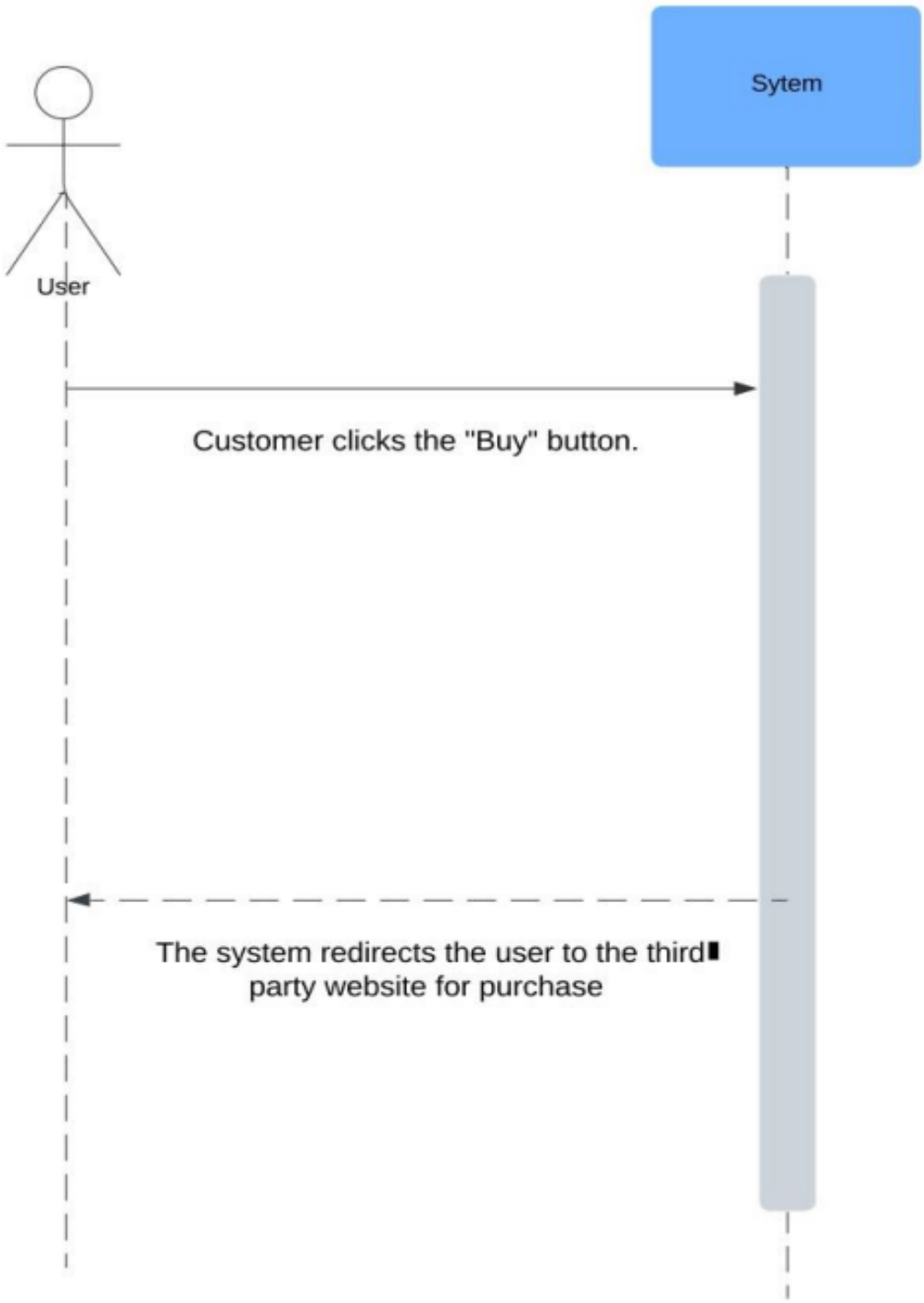

System
User
No matching products found
displays a “No products found” message

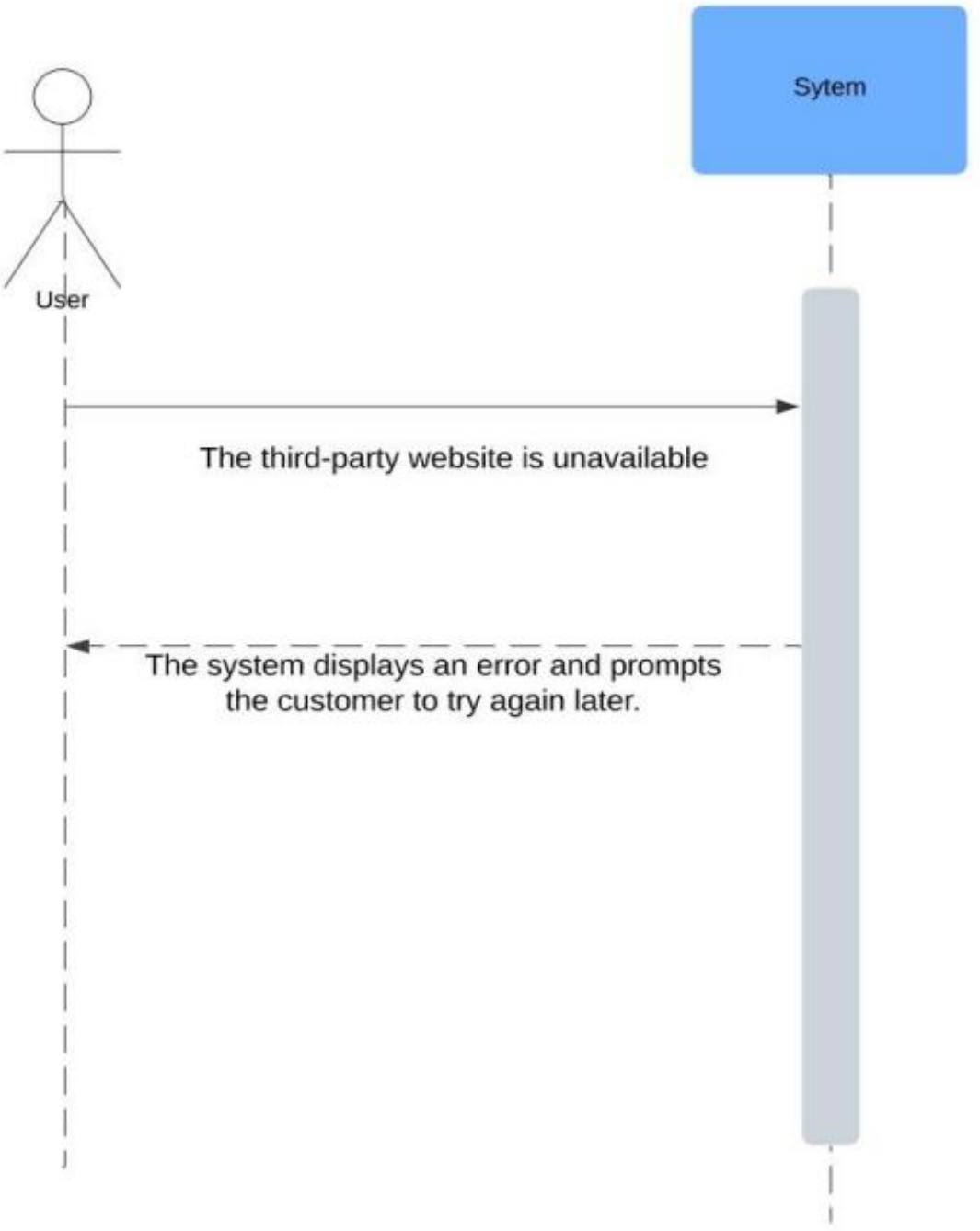
3) Register User



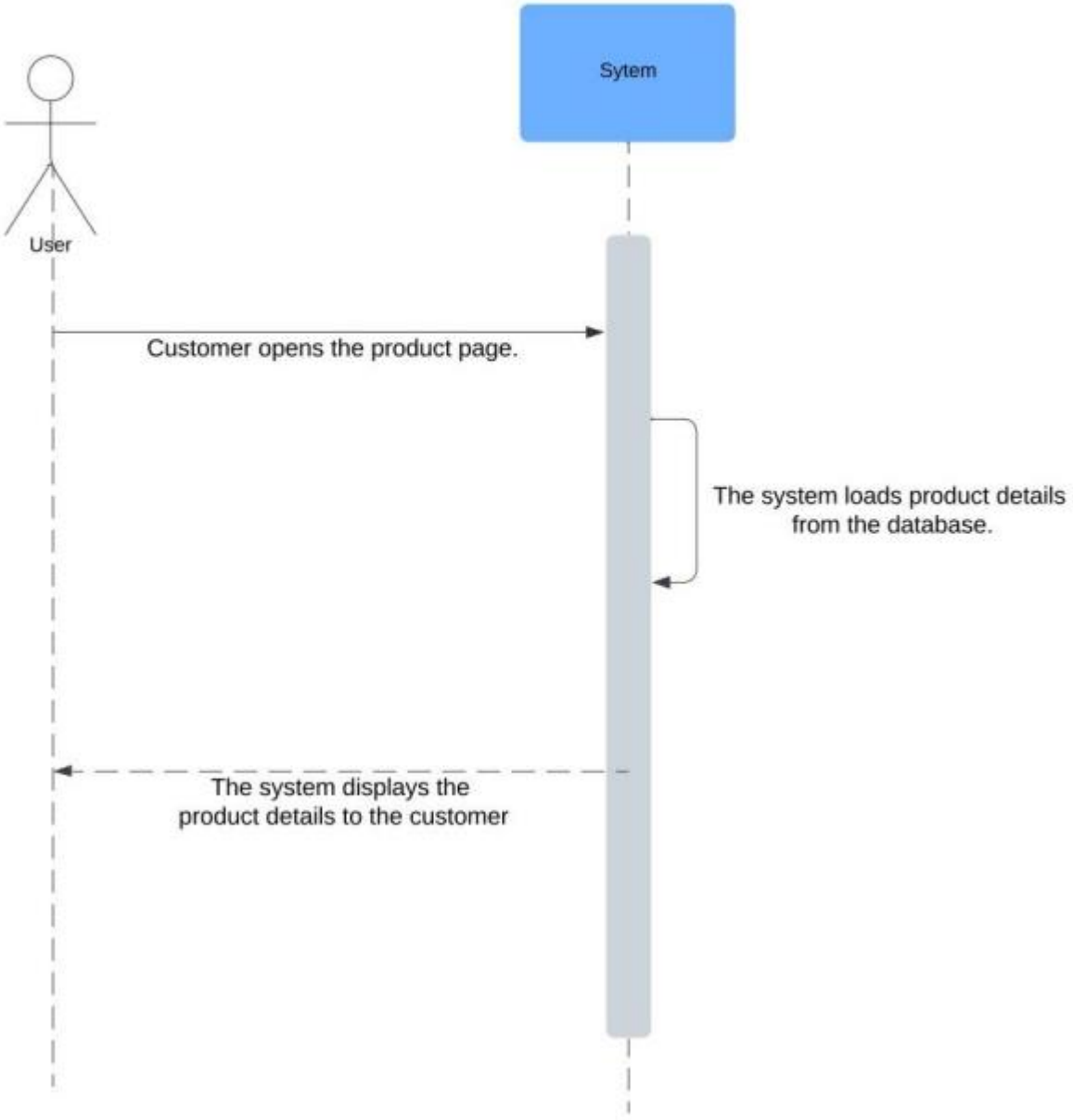

Sytem
User
The user enters incomplete or invalid data.
The system displays an error message and prompts the user to correct it.
The email already exists in the database.
. The system informs the user and prompts them to log in or recover their password

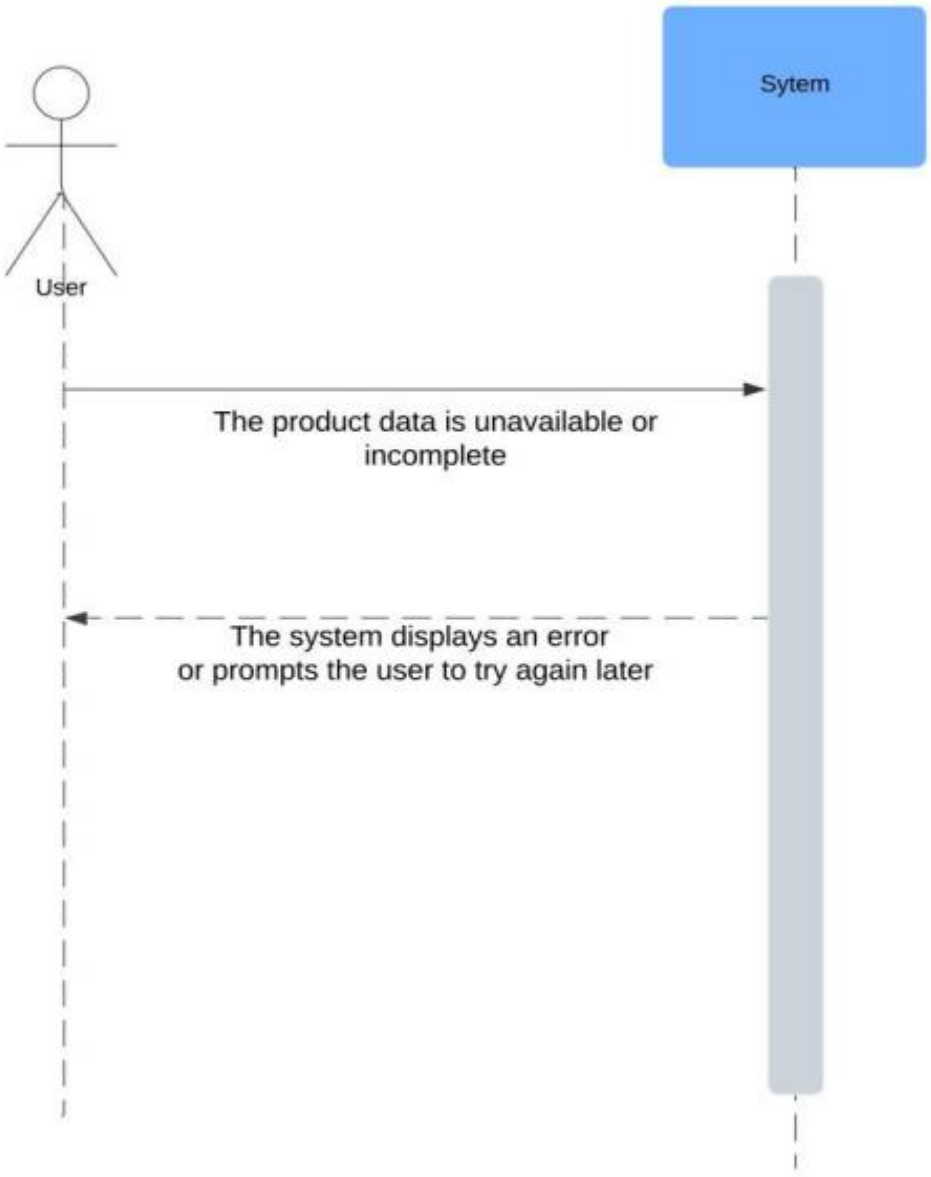
4) Redirect User



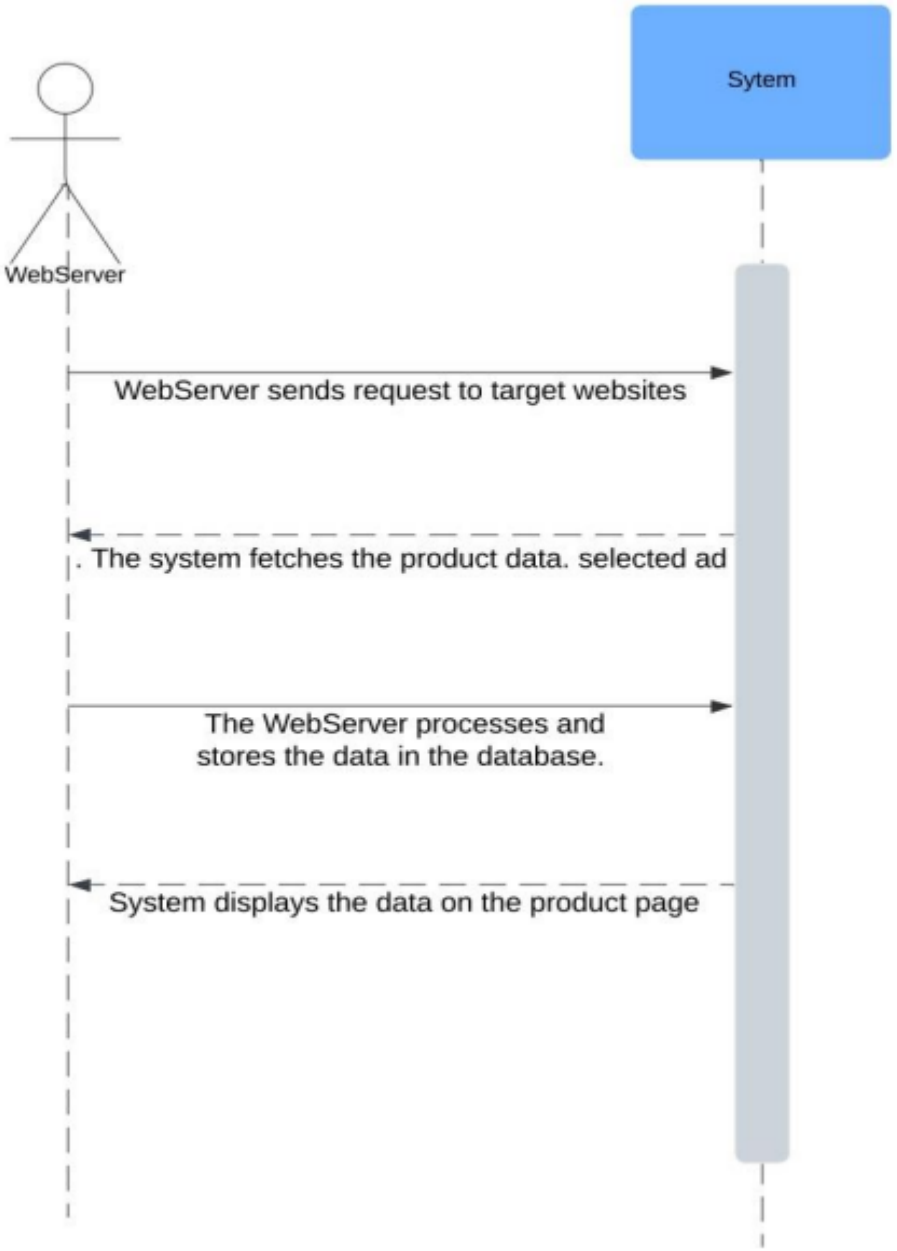

Sytem
User
The third-party website is unavailable
The system displays an error and prompts
the customer to try again later.

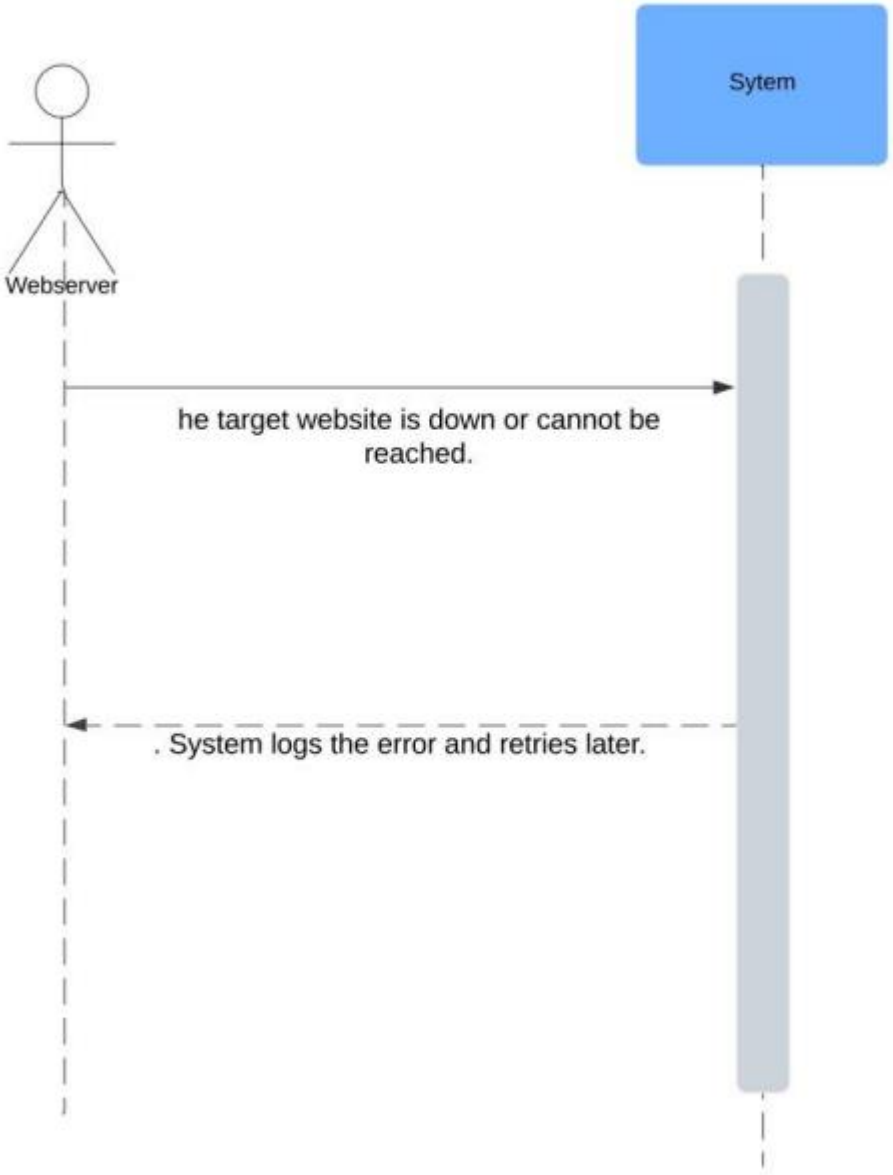
5) Display Products



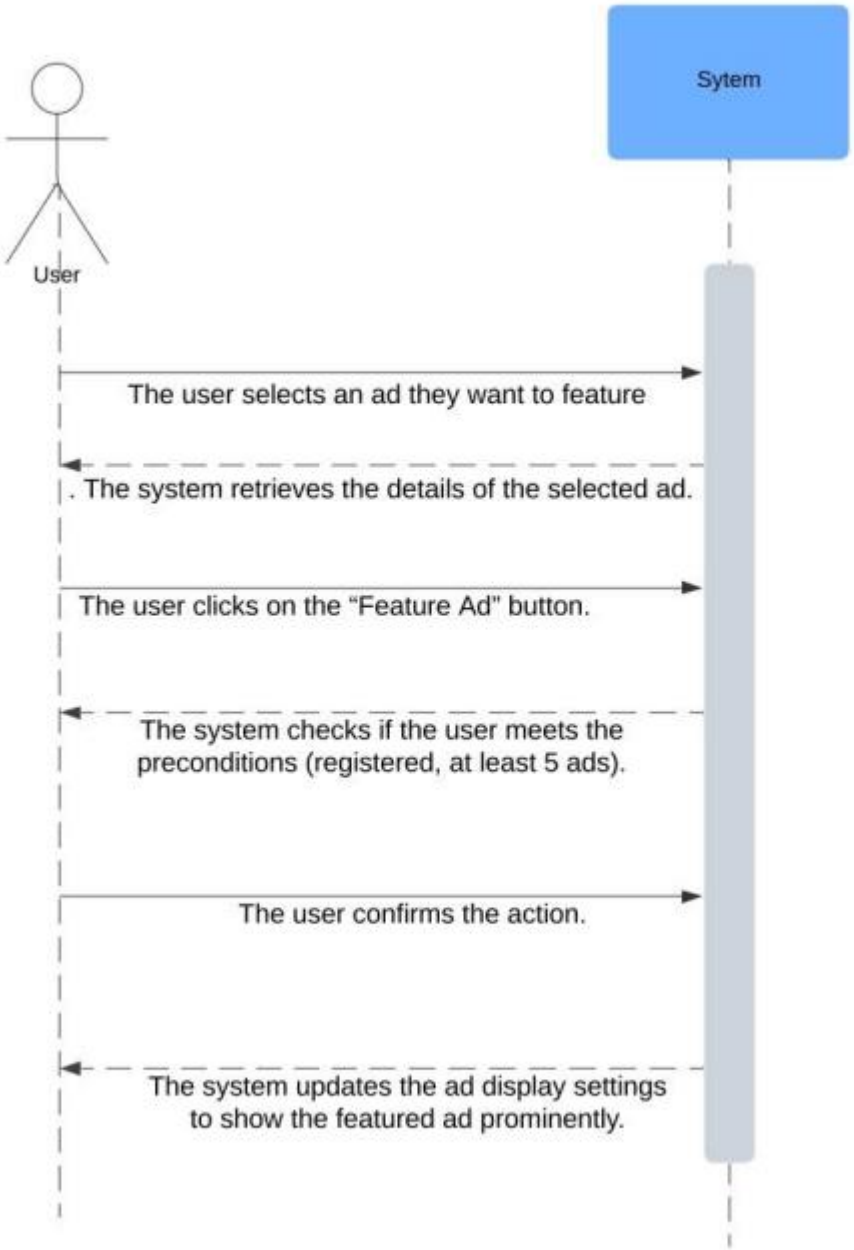

Sytem
User
The product data is unavailable or incomplete
The system displays an error or prompts the user to try again later

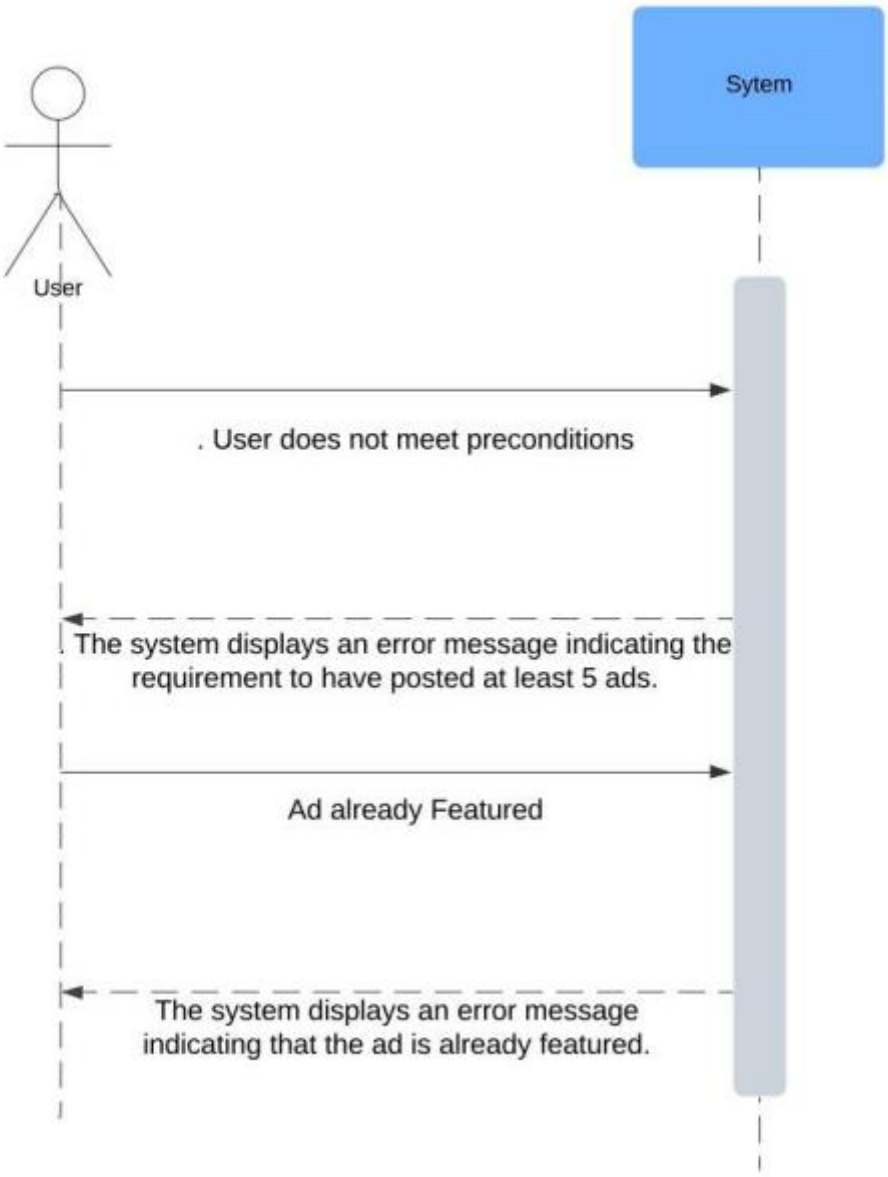
6) Scrape Data



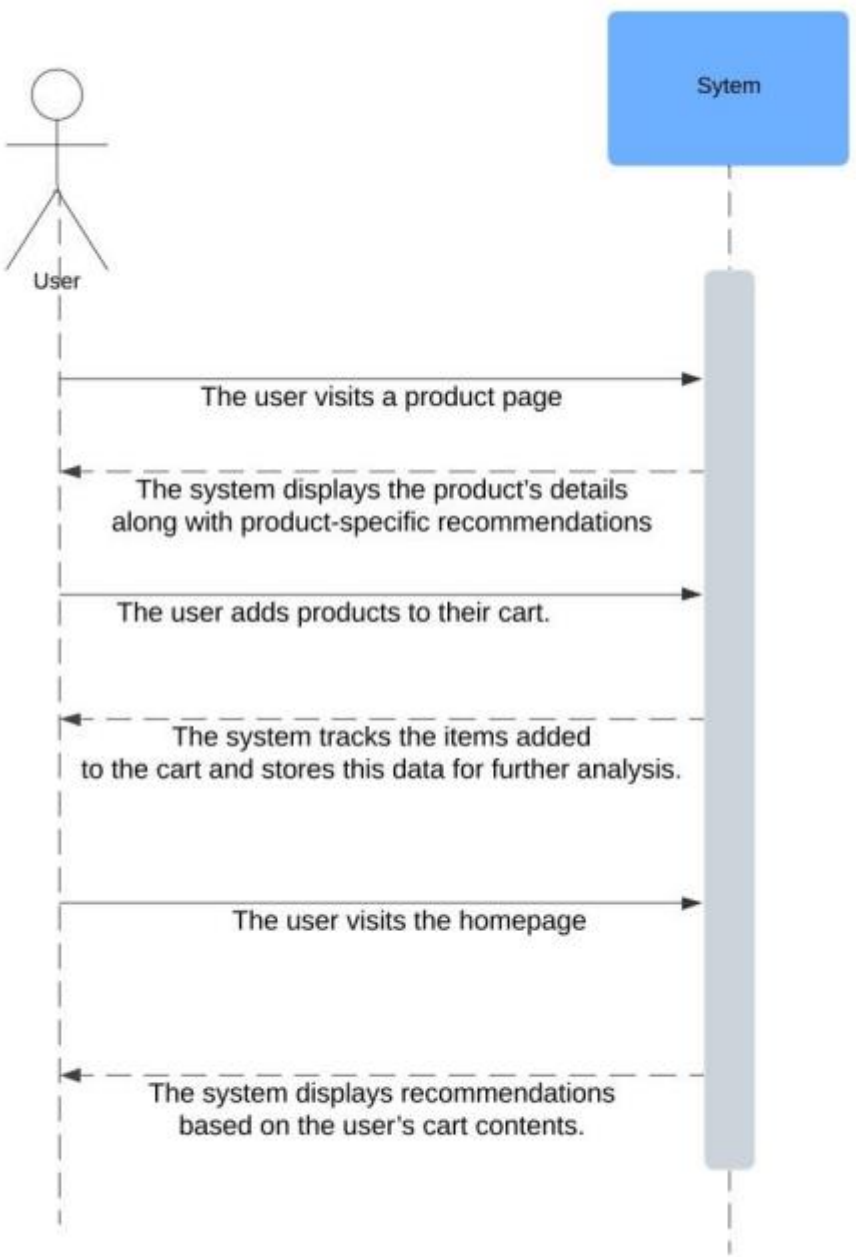

Sytem
Webserver
he target website is down or cannot be reached.
. System logs the error and retries later.

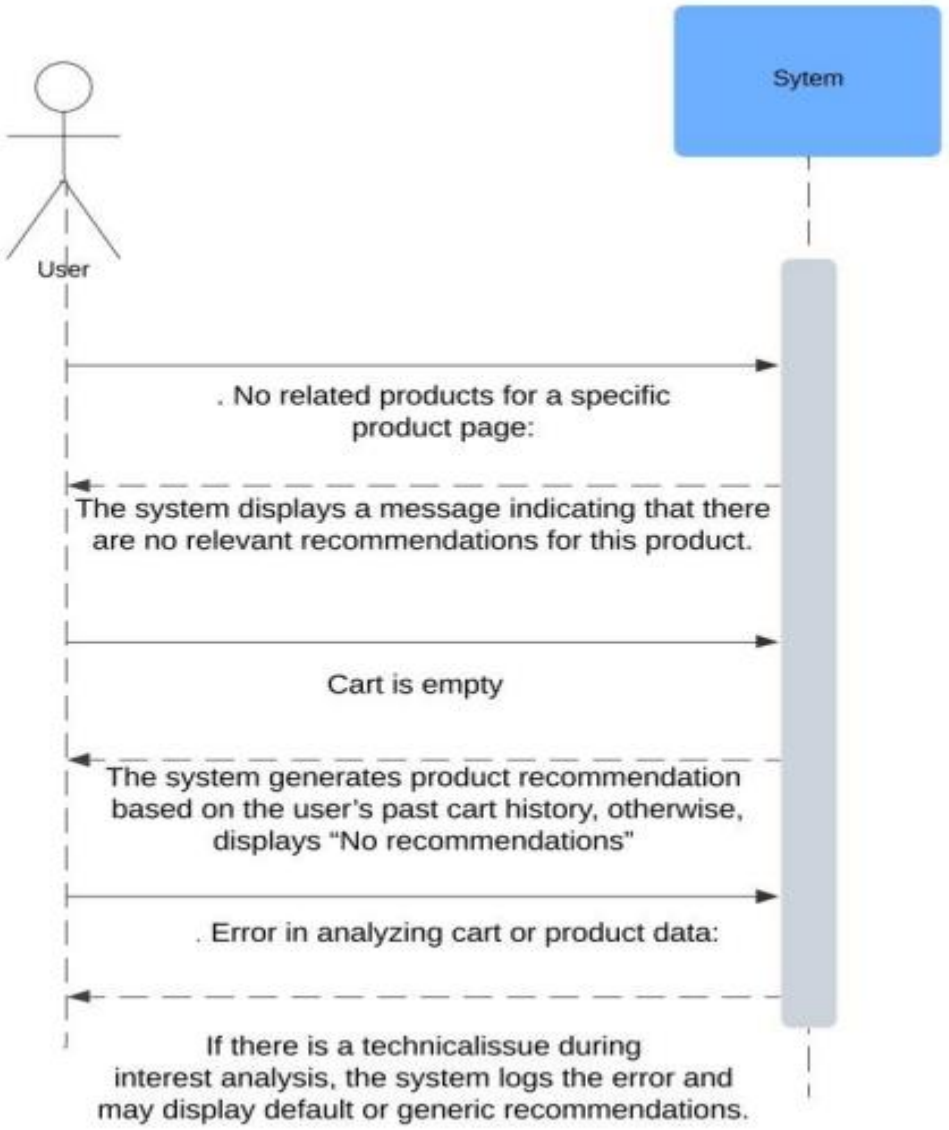
7) Feature Ad


Sytem
User
The user selects an ad they want to feature
. The system retrieves the details of the selected ad.
The user clicks on the “Feature Ad” button.
The system checks if the user meets the preconditions (registered, at least 5 ads).
The user confirms the action.
The system updates the ad display settings to show the featured ad prominently.

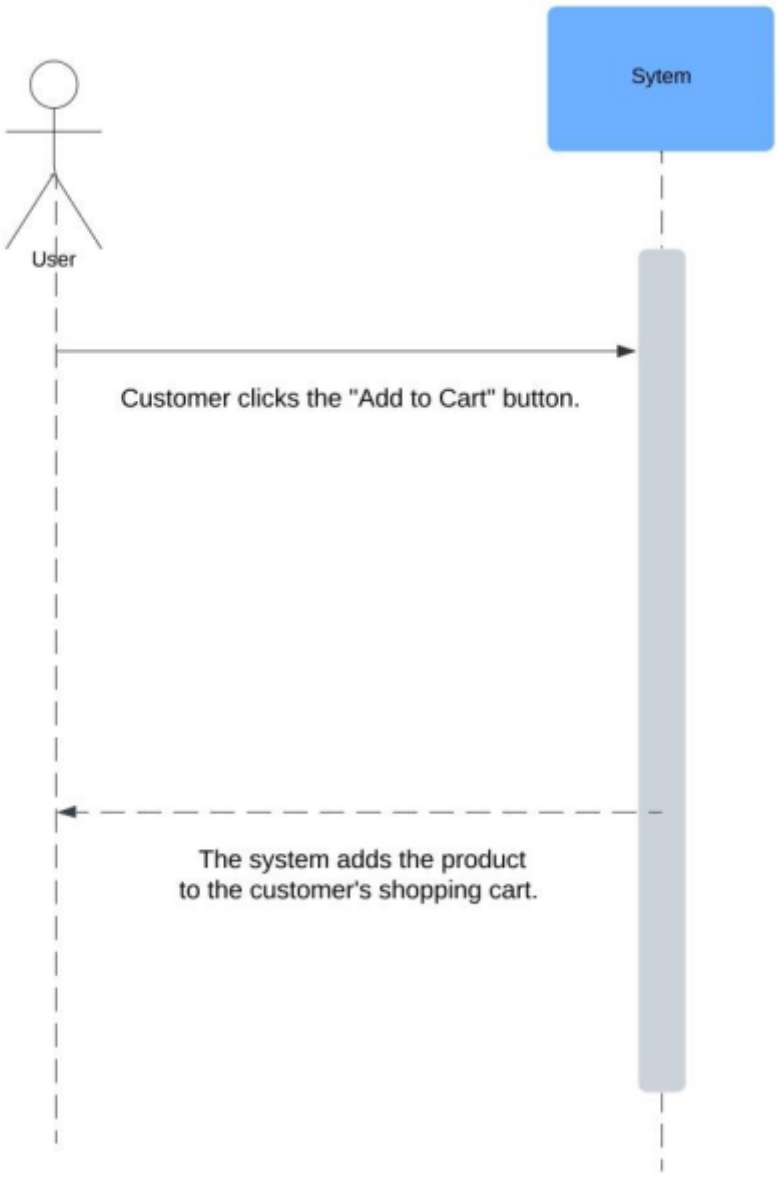

Sytem
User
. User does not meet preconditions
. The system displays an error message indicating the requirement to have posted at least 5 ads.
Ad already Featured
The system displays an error message indicating that the ad is already featured.

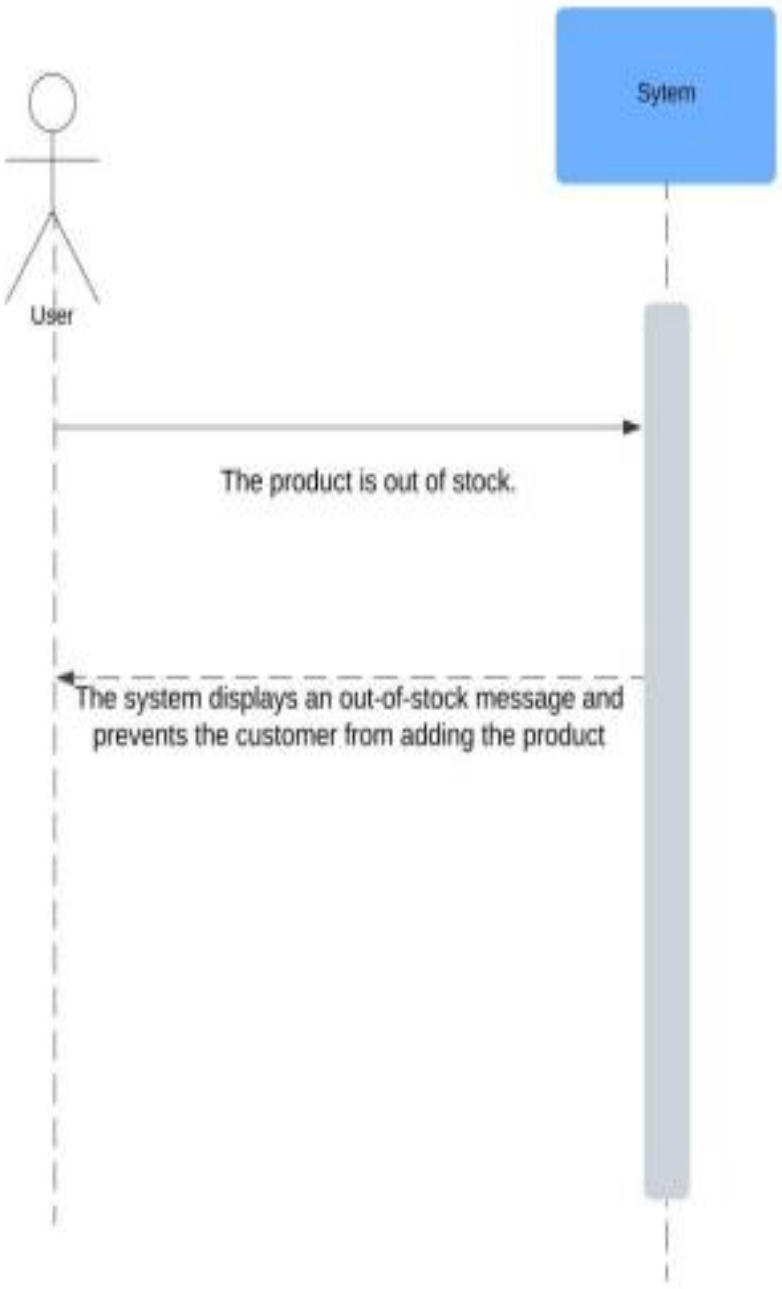
8) Analyze Interests



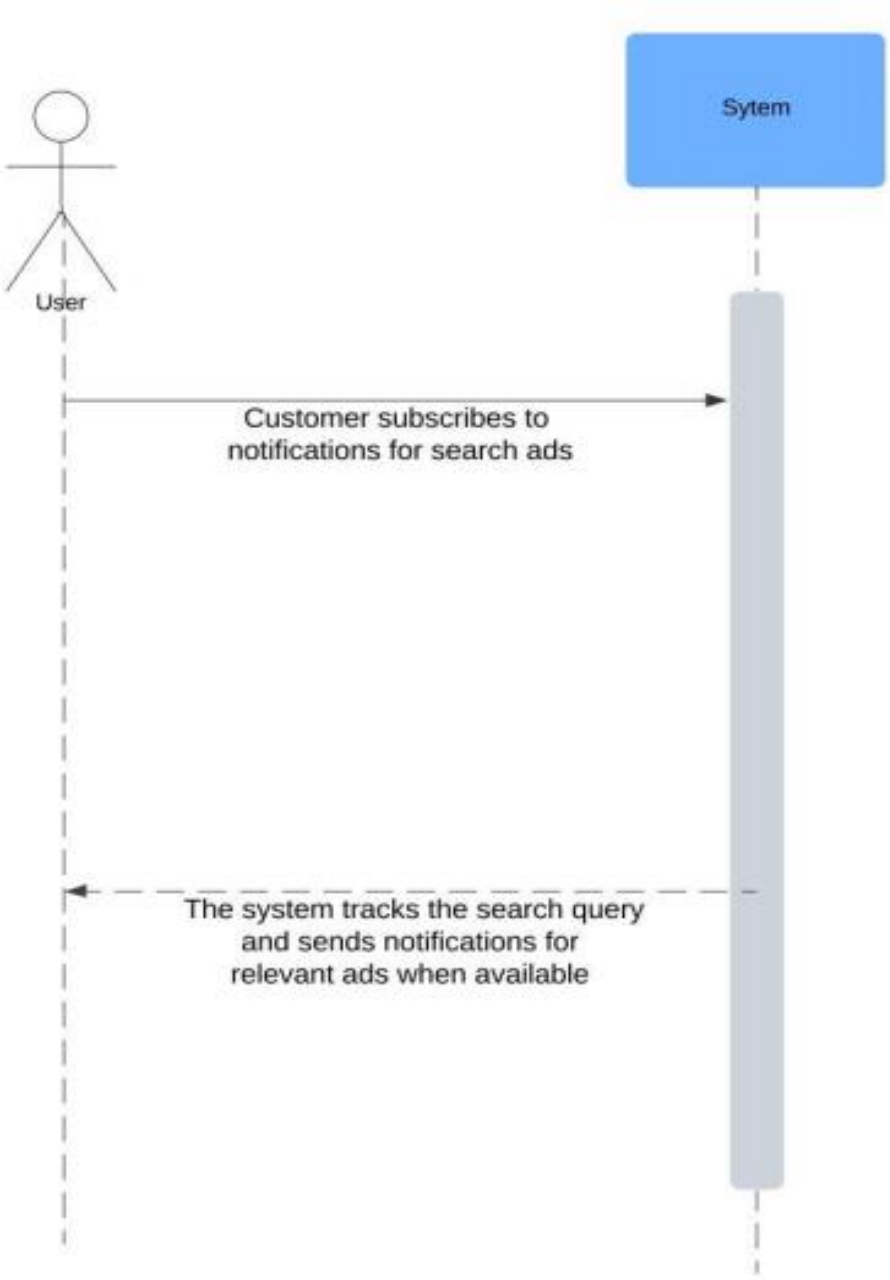

Sytem
User
. No related products for a specific product page:
The system displays a message indicating that there are no relevant recommendations for this product.
Cart is empty
The system generates product recommendation based on the user's past cart history, otherwise, displays "No recommendations"
. Error in analyzing cart or product data:
If there is a technicalissue during interest analysis, the system logs the error and may display default or generic recommendations.

9) Add to Cart

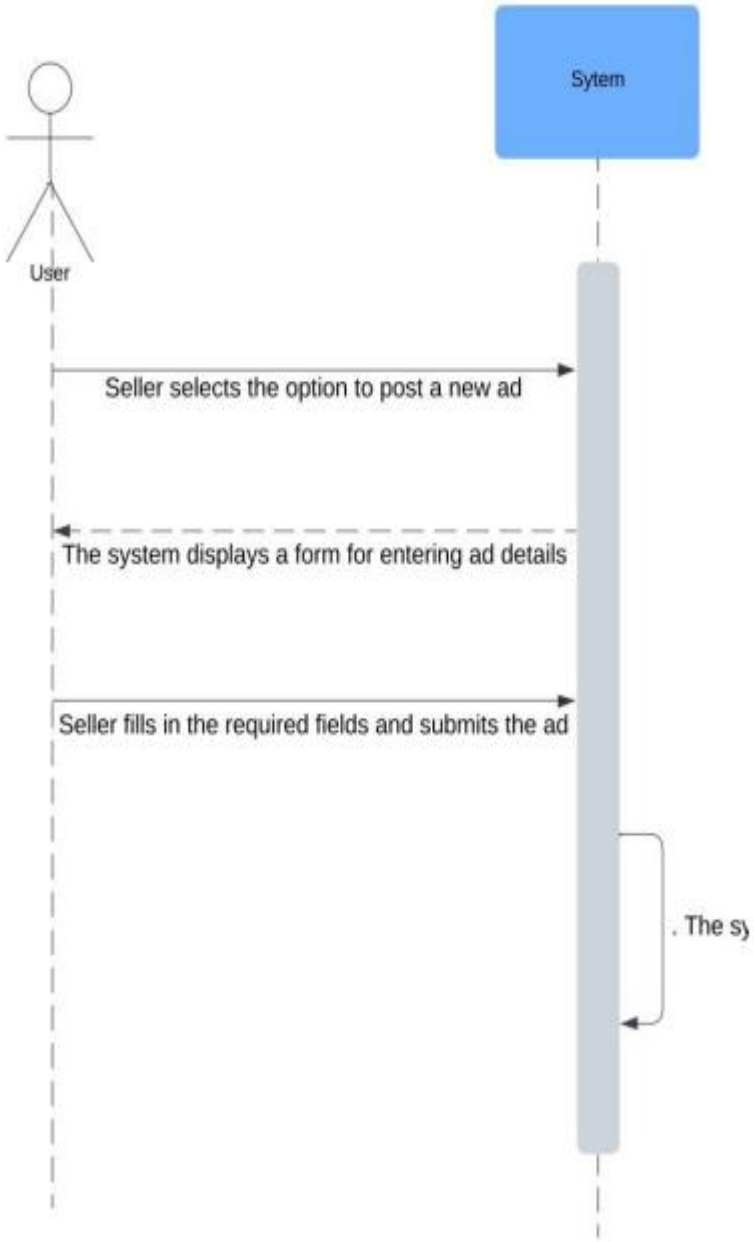

Sytem
User
Customer clicks the "Add to Cart" button.
The system adds the product
to the customer's shopping cart.

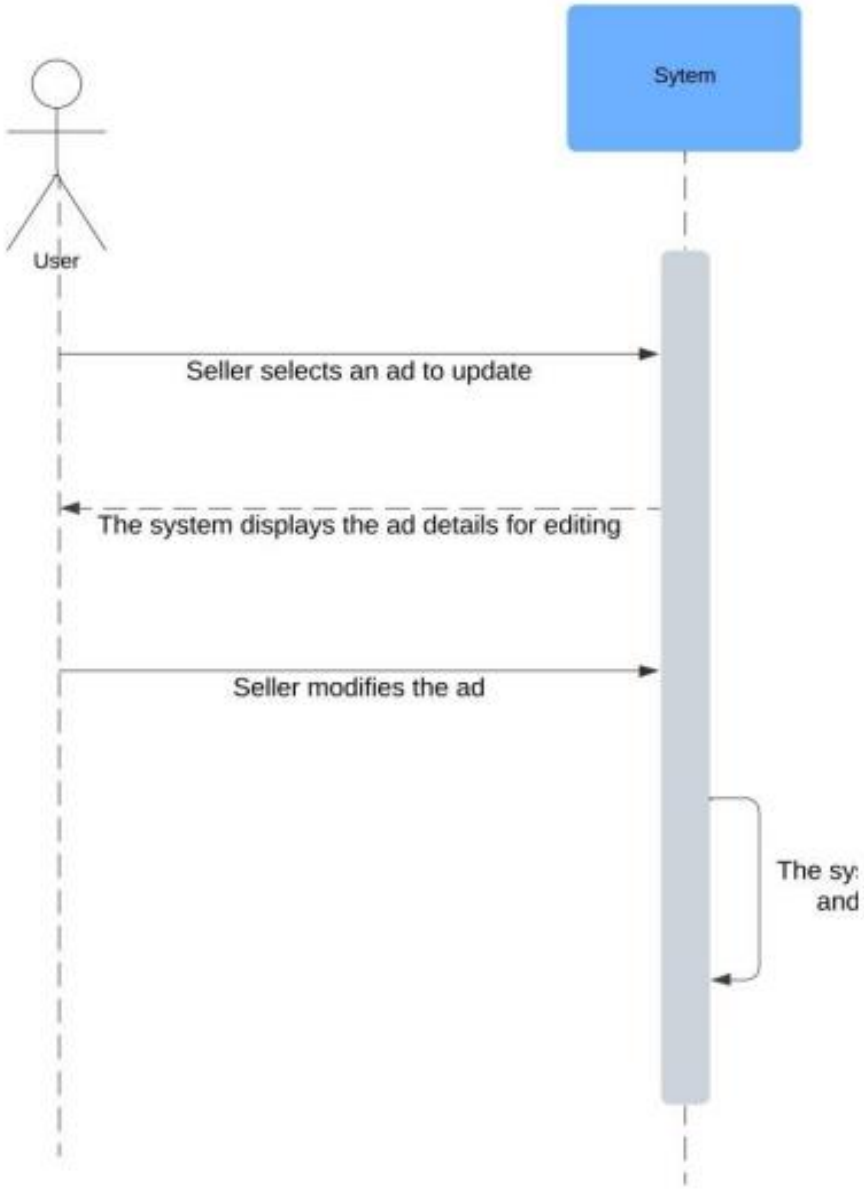

Sytem
User
The product is out of stock.
The system displays an out-of-stock message and prevents the customer from adding the product

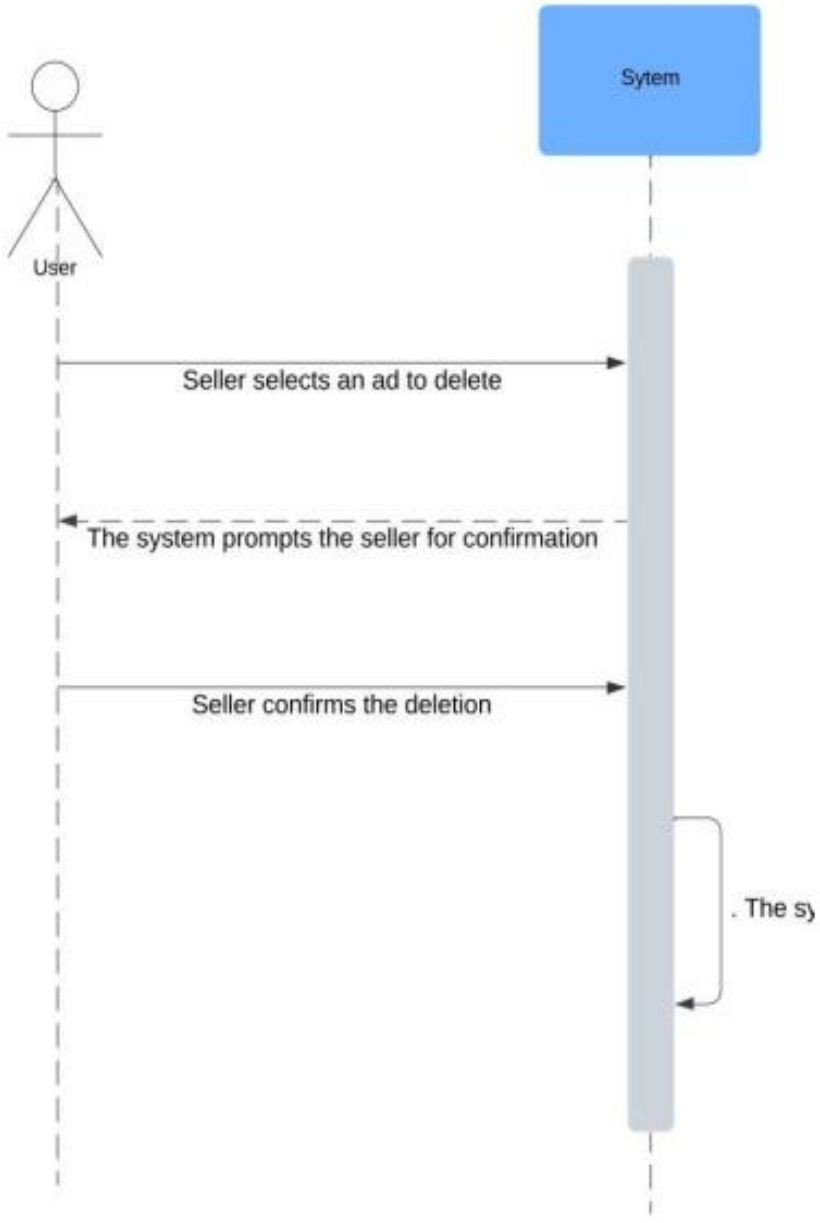
10) Notify User

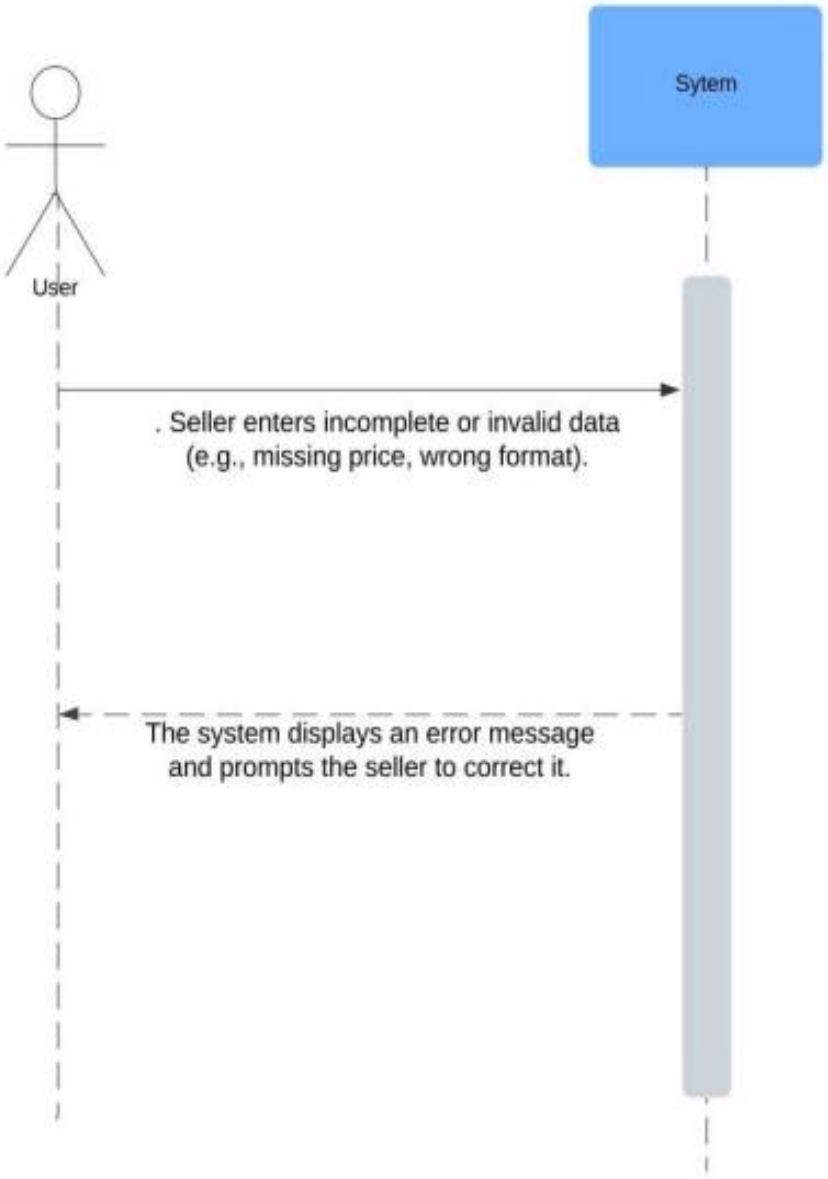


11) Manage Ads



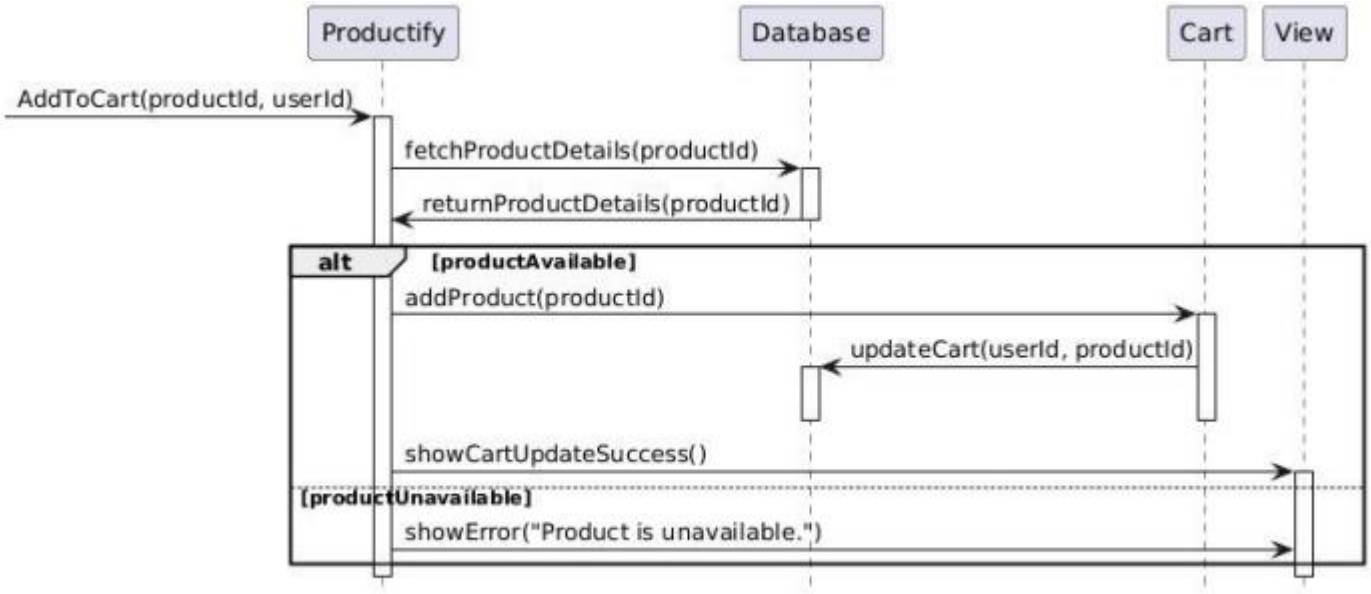

Sytem
User
Seller selects an ad to update
The system displays the ad details for editing
Seller modifies the ad
The sy
and


Sytem
User
Seller selects an ad to delete
The system prompts the seller for confirmation
Seller confirms the deletion
. The sy

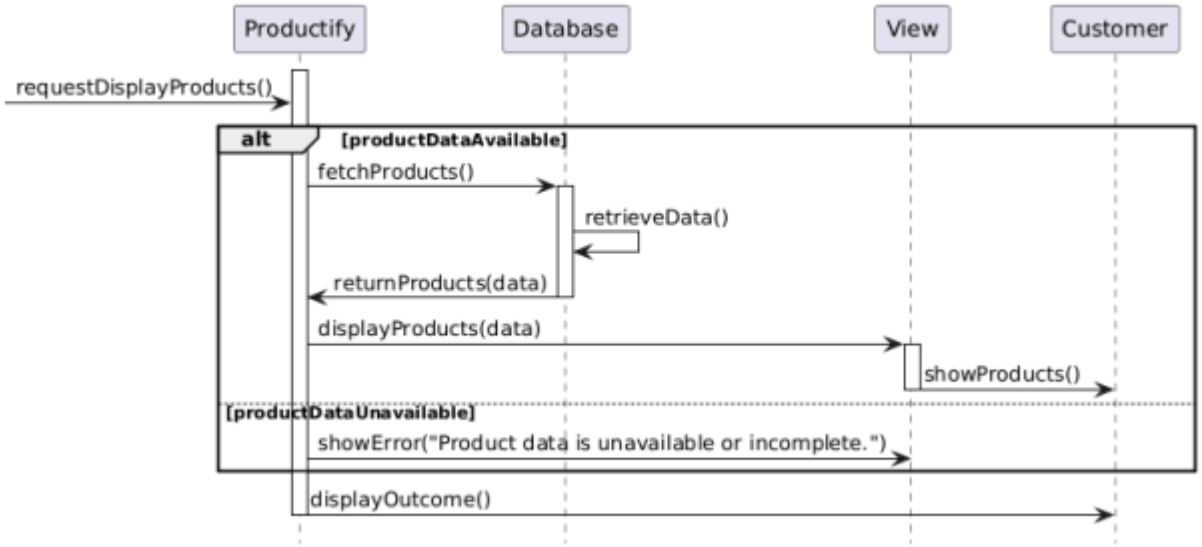

Sytem
User
. Seller enters incomplete or invalid data
(e.g., missing price, wrong format).
The system displays an error message
and prompts the seller to correct it.

# 6. Sequence Diagram

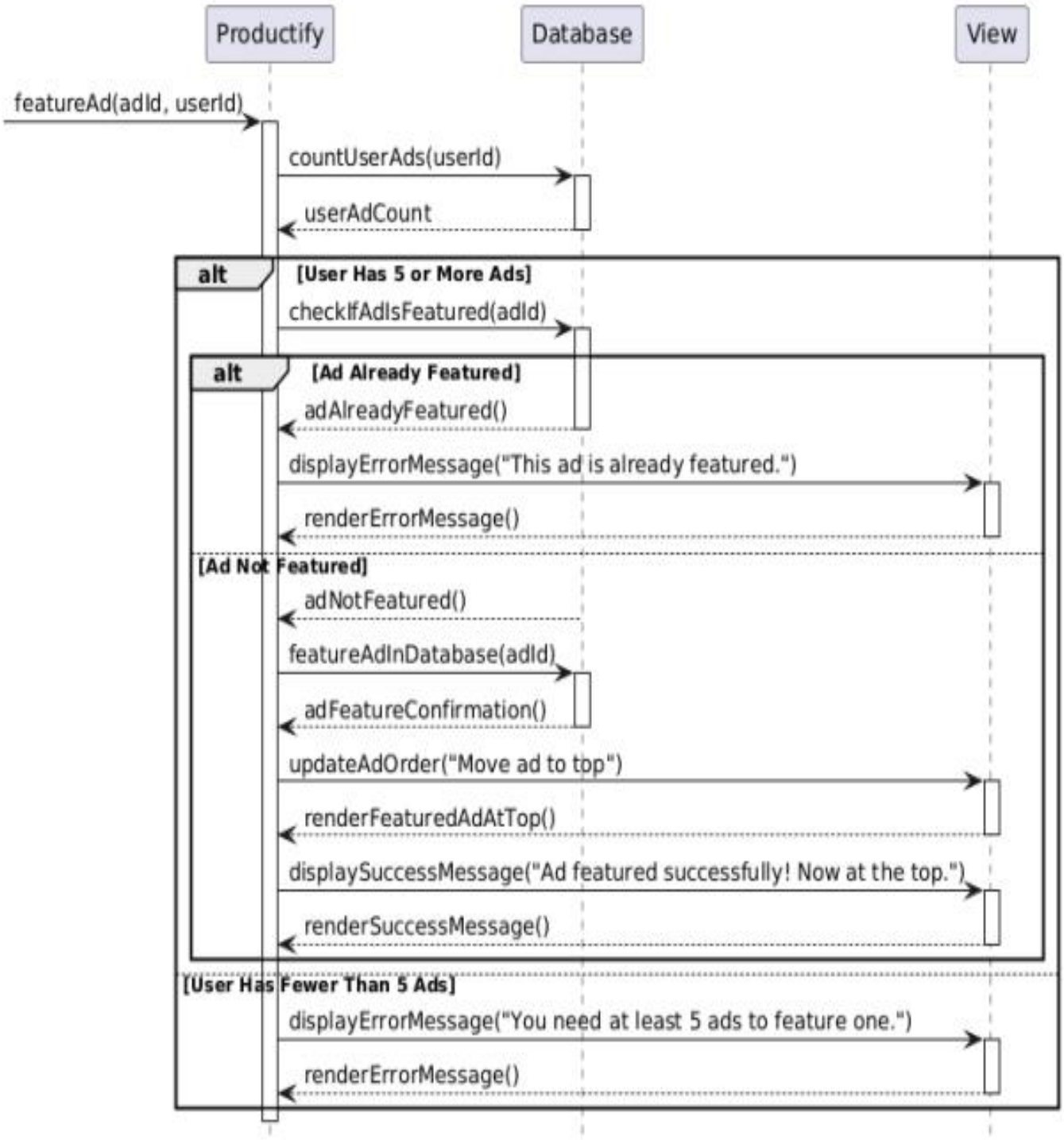
1) Add to Cart



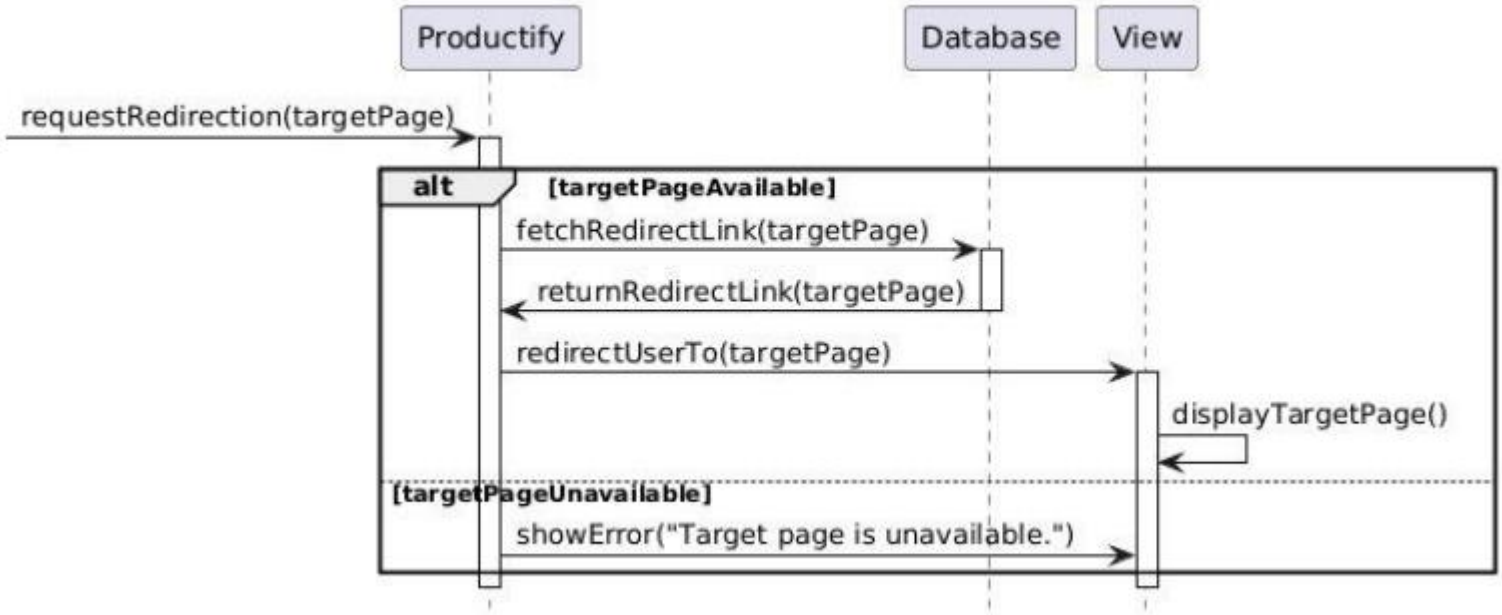

2) Display Products



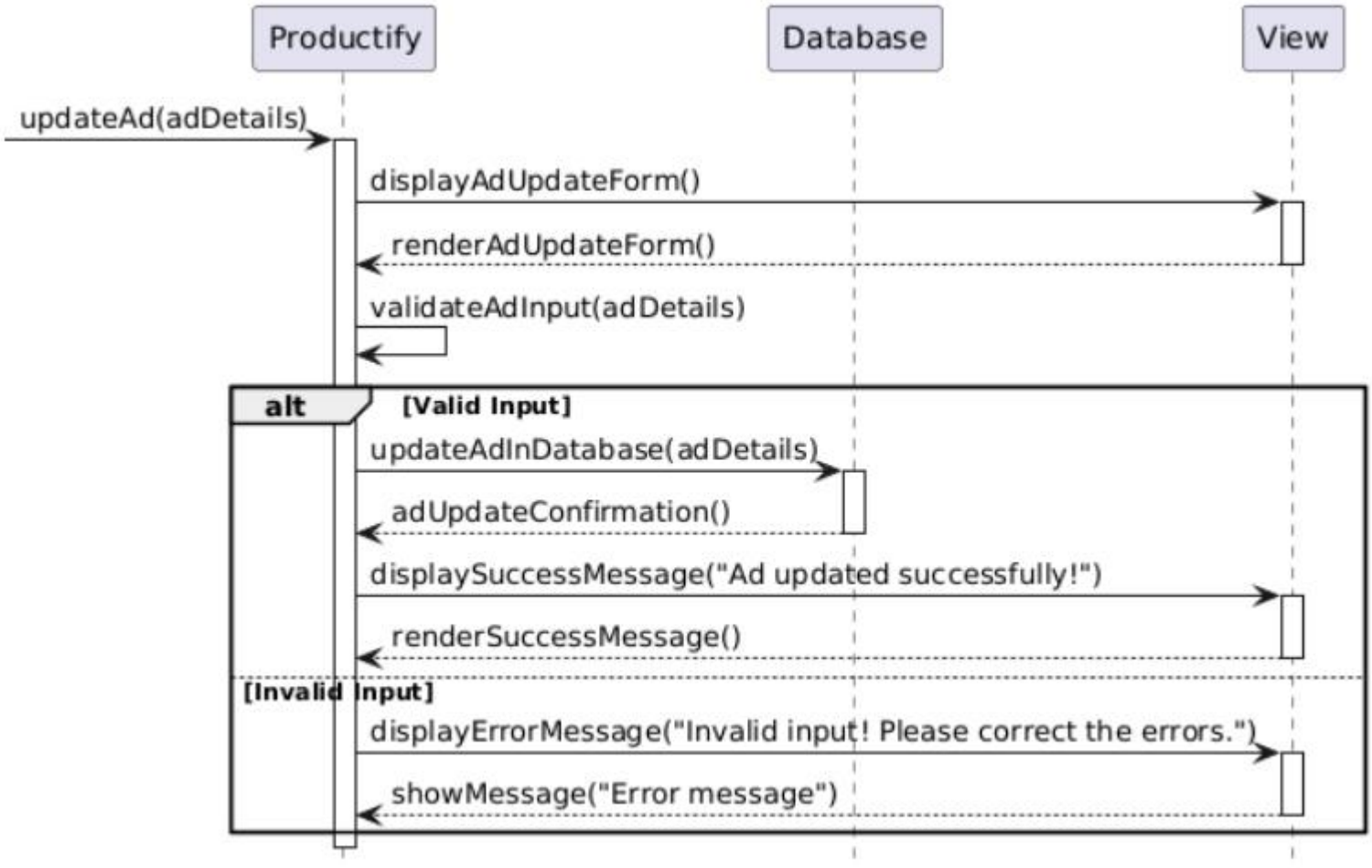
3) Feature Ad


Productify
Database
View
featureAd(adId, userId)
countUserAds(userId)
userAdCount
alt
[User Has 5 or More Ads]
checkIfAdIsFeatured(adId)
alt
[Ad Already Featured]
adAlreadyFeatured()
displayErrorMessage("This ad is already featured.")
renderErrorMessage()
[Ad Not Featured]
adNotFeatured()
featureAdInDatabase(adId)
adFeatureConfirmation()
updateAdOrder("Move ad to top")
renderFeaturedAdAtTop()
displaySuccessMessage("Ad featured successfully! Now at the top.")
renderSuccessMessage()
[User Has Fewer Than 5 Ads]
displayErrorMessage("You need at least 5 ads to feature one.")
renderErrorMessage()

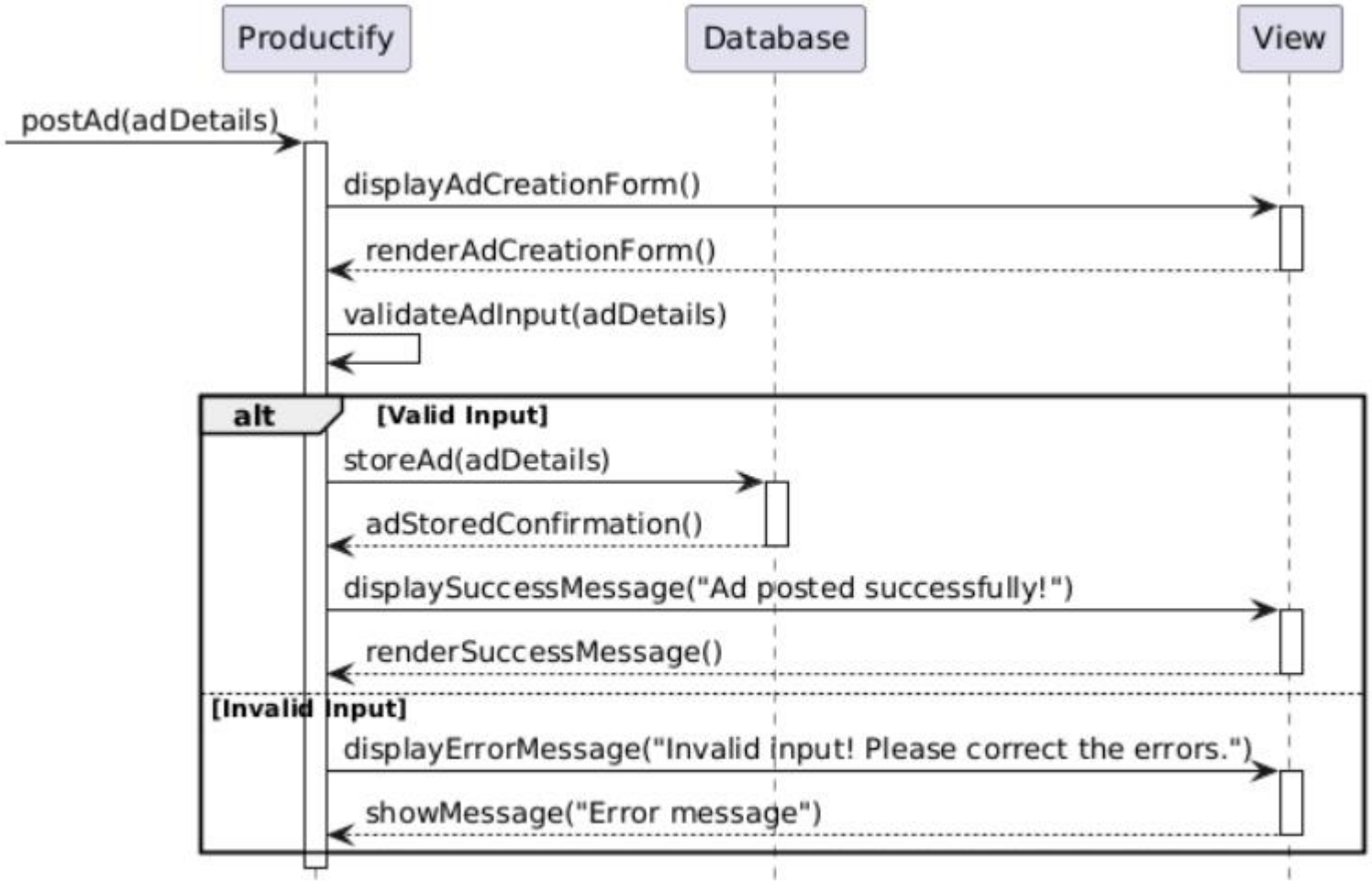
4) Redirect User



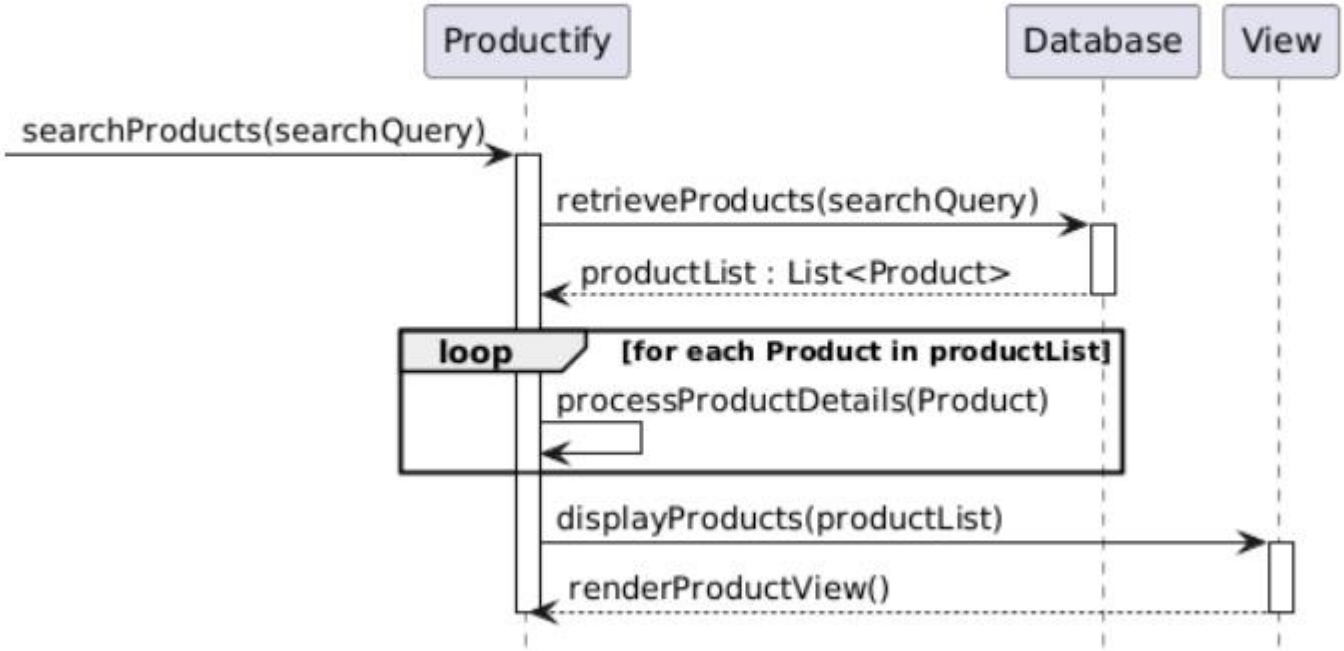

5) Update Ad



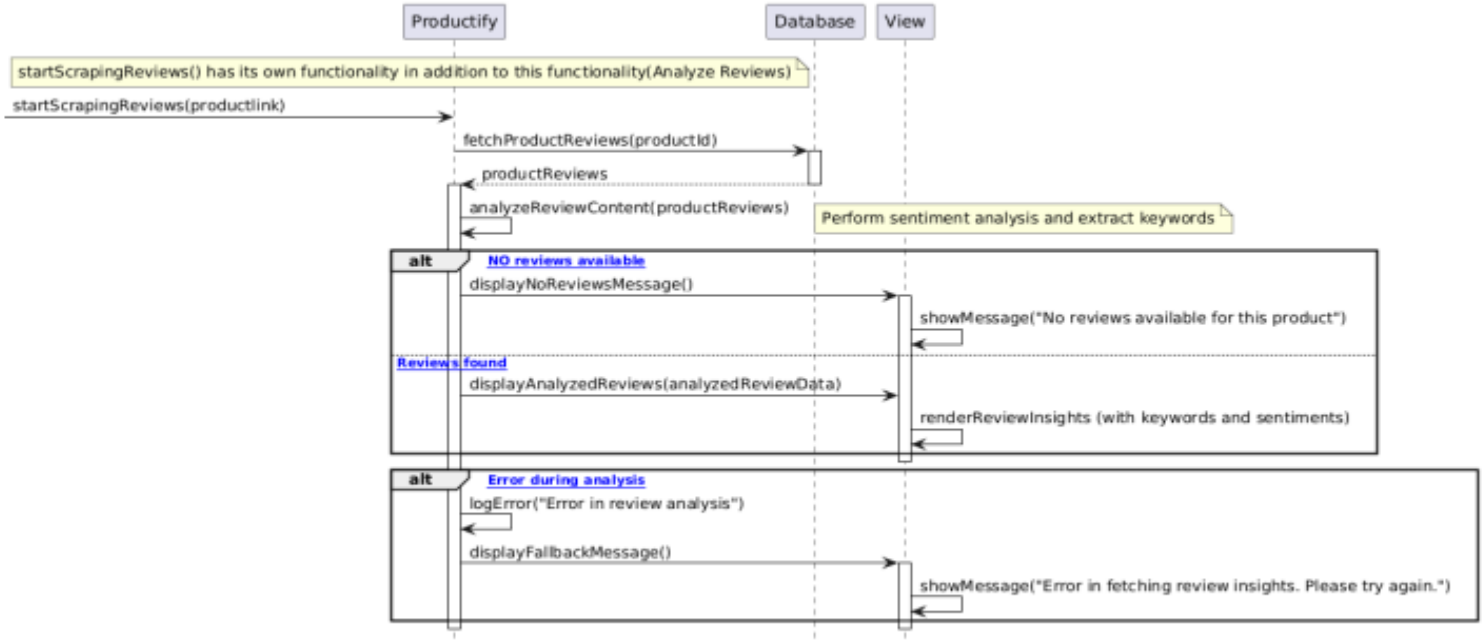
6) Post Ad


Productify
Database
View
postAd(adDetails)
displayAdCreationForm()
renderAdCreationForm()
validateAdInput(adDetails)
alt
[Valid Input]
storeAd(adDetails)
adStoredConfirmation()
displaySuccessMessage("Ad posted successfully!")
renderSuccessMessage()
[Invalid Input]
displayErrorMessage("Invalid input! Please correct the errors.")
showMessage("Error message")

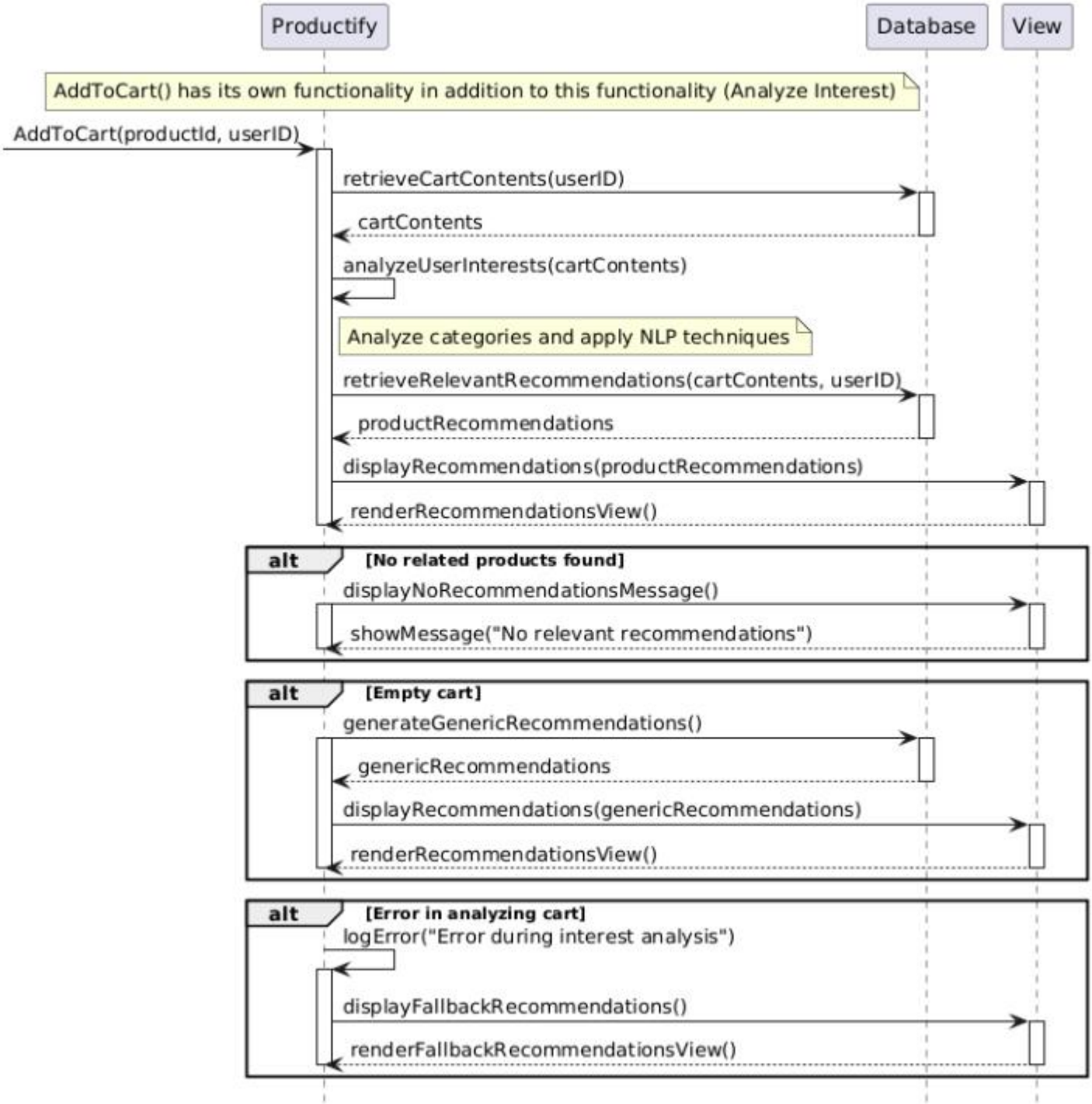
7) Search Products



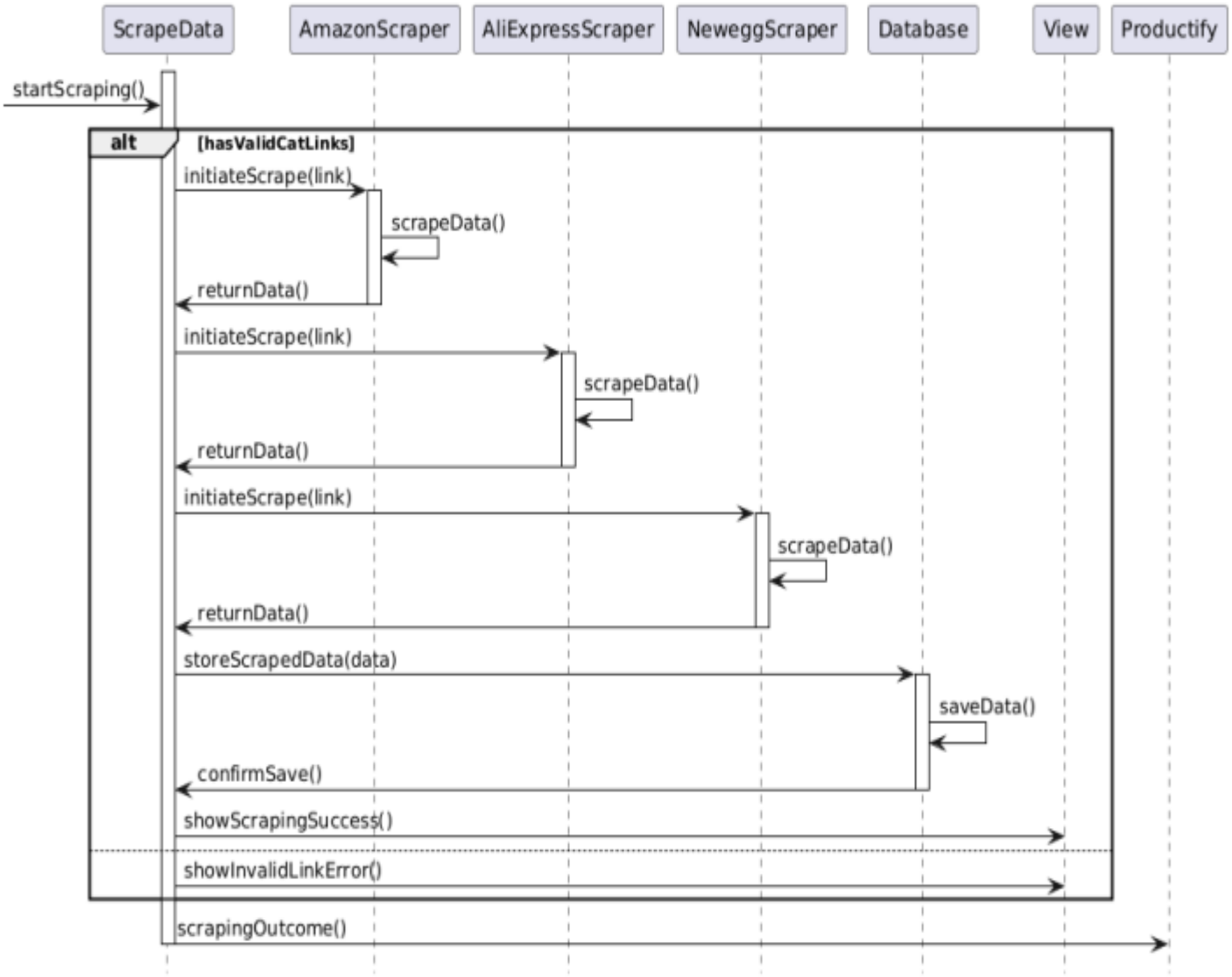

8) Analyze Reviews



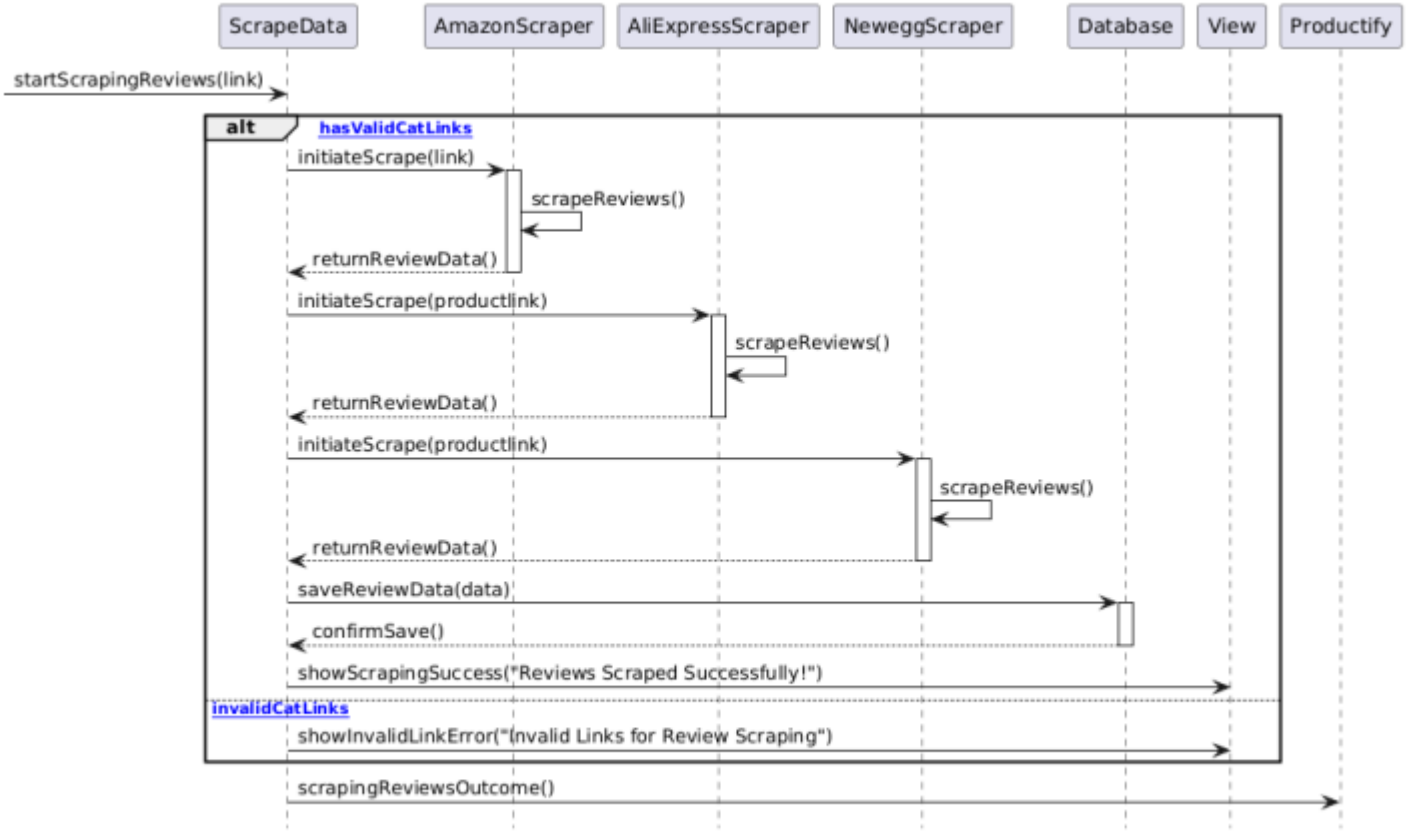
9) Analyze Interests



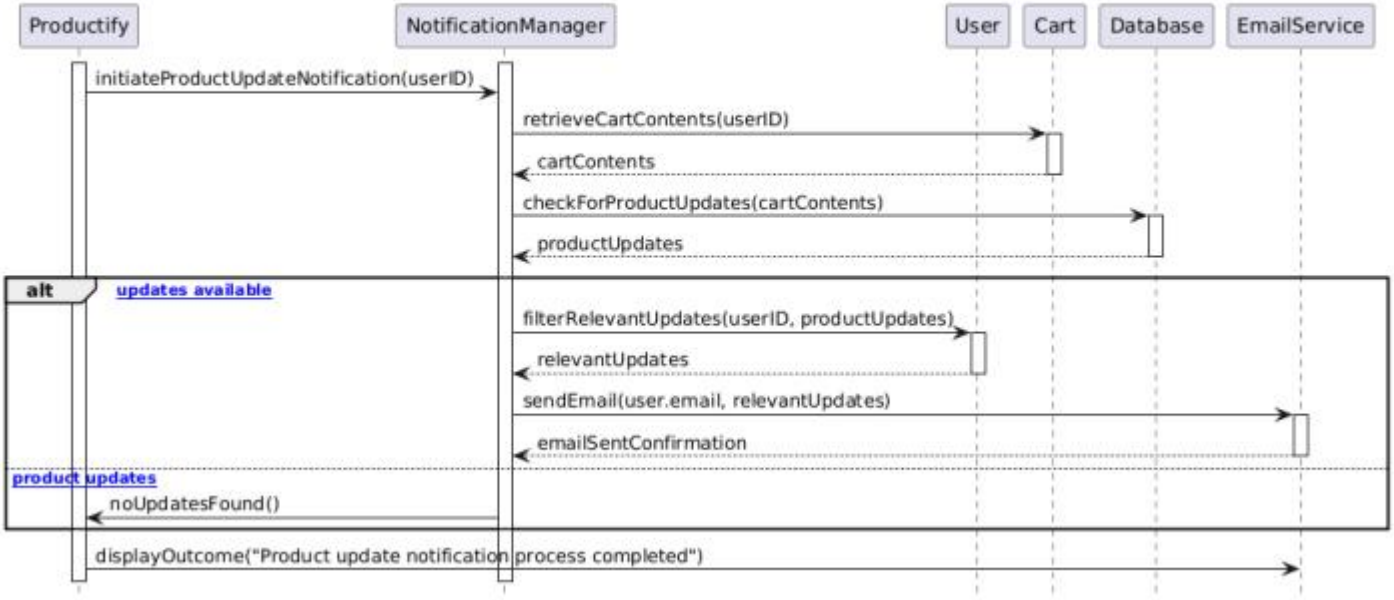
10) Scrape Data



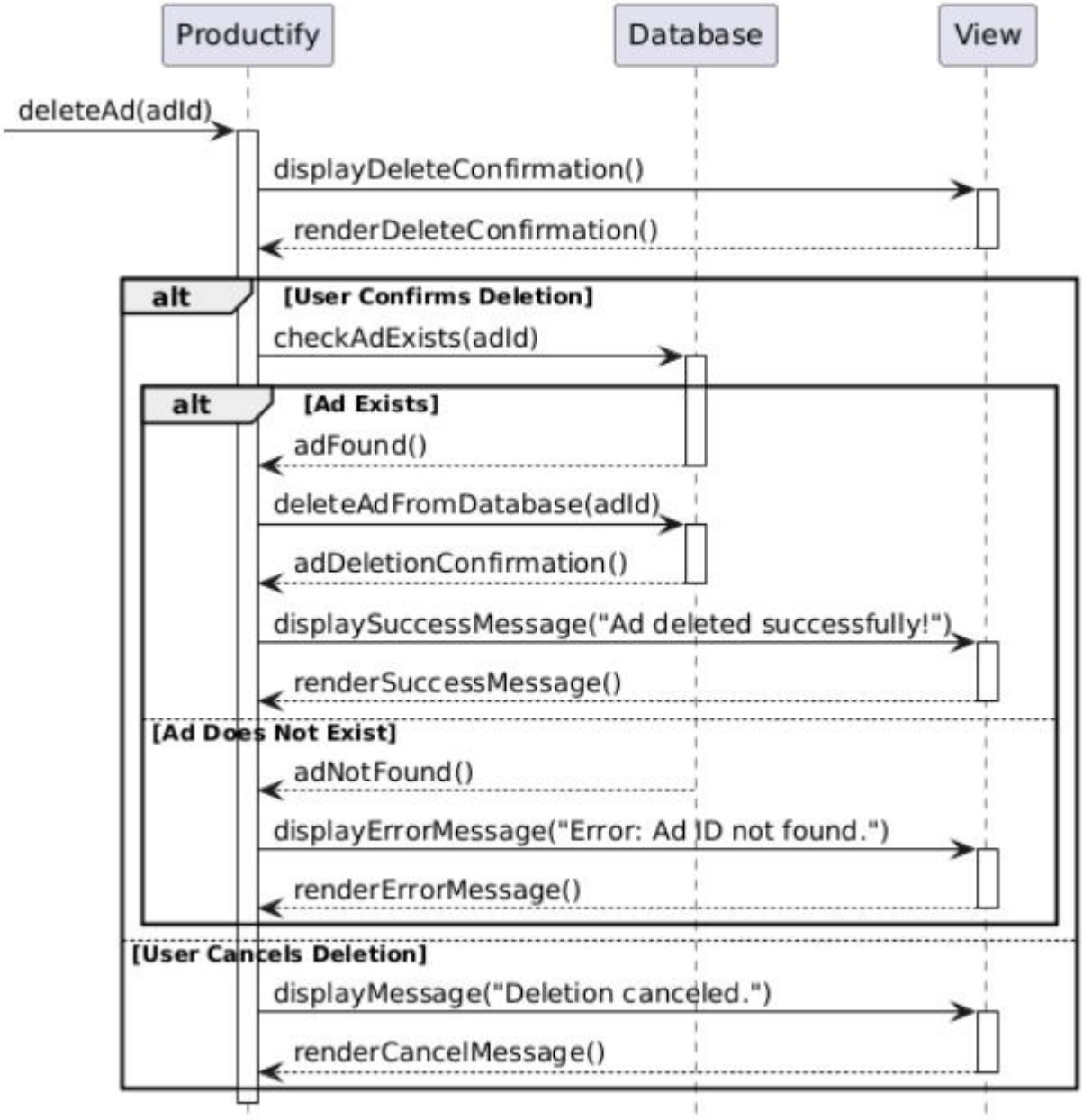
11) Scrape Reviews



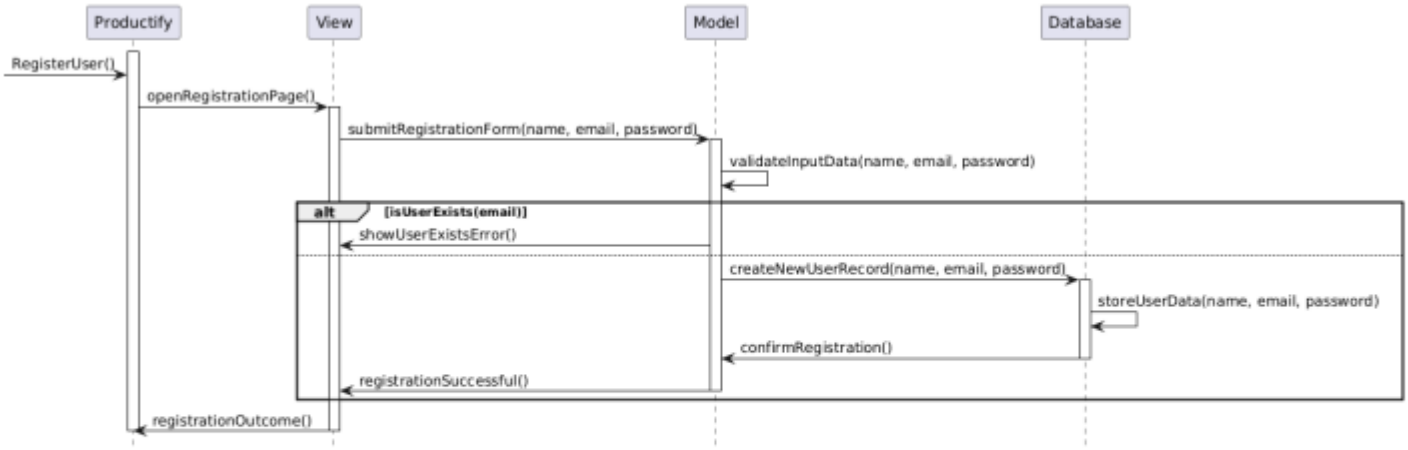

12) Notify User



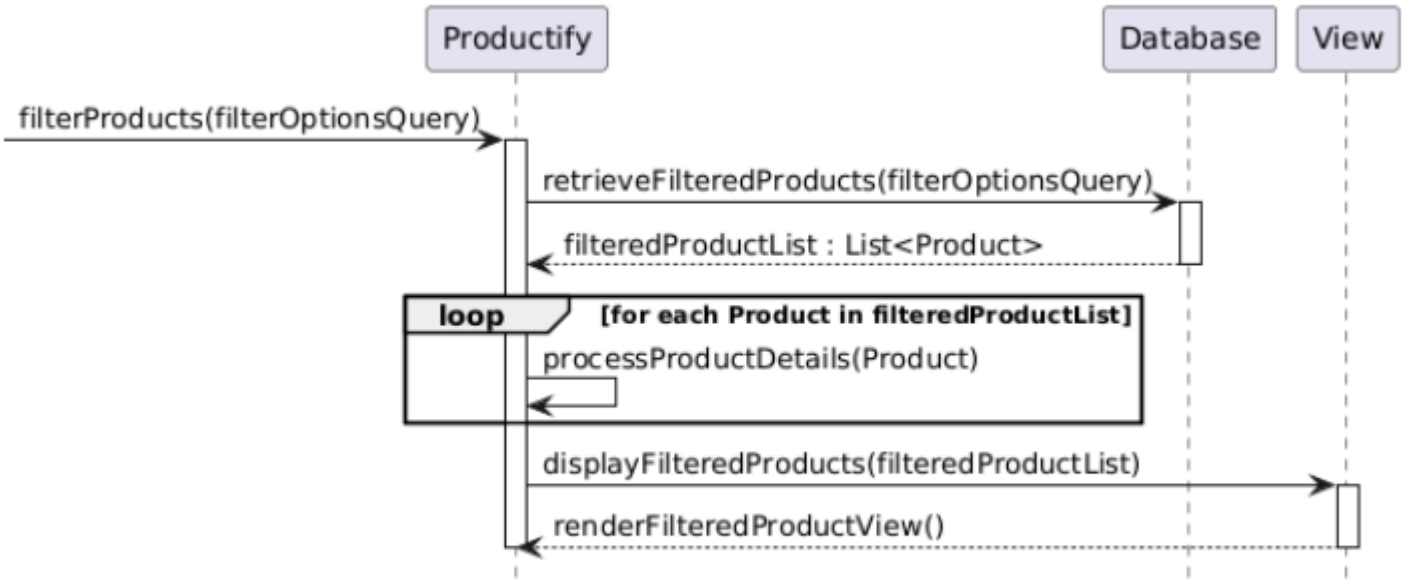
13) Delete Ad


Productify
Database
View
deleteAd(adId)
displayDeleteConfirmation()
renderDeleteConfirmation()
alt
[User Confirms Deletion]
checkAdExists(adId)
alt
[Ad Exists]
adFound()
deleteAdFromDatabase(adId)
adDeletionConfirmation()
displaySuccessMessage("Ad deleted successfully!")
renderSuccessMessage()
[Ad Does Not Exist]
adNotFound()
displayErrorMessage("Error: Ad ID not found.")
renderErrorMessage()
[User Cancels Deletion]
displayMessage("Deletion canceled.")
renderCancelMessage()

14) Register User




15) Filter Products



# 7. Class Diagram