

Functional Requirements

Use Case 1: Display All Cosmetic Products on the Home Page

Actors: User (Visitor)

Preconditions: The user accesses the website's home page.

Main Flow:

1. The user navigates to the home page.
2. The system retrieves a list of all cosmetic products from the products-service.
3. The system displays the following information for each product on the home page:
 - Product image
 - Product name
 - Product price
 - Product category
4. The system implements pagination to limit the number of products displayed per page.

Postconditions: The user sees a paginated list of all available cosmetic products with the necessary details

Use Case 2: View Details of a Single Product

Actors: User (Visitor)

Preconditions: The user is on the home page and a list of products is displayed.

Main Flow:

1. The user selects a product by clicking on it.
2. The system retrieves detailed information about the selected product from the products-service.
3. The system navigates to a new page/view displaying the following information about the selected product:
 - Product image
 - Product name
 - Product price
 - Product description
 - Product usage

Postconditions: The user is able to view detailed information about the selected product on a new page

Use Case 3: Create a New Product

Actors: Admin

Preconditions: The admin is logged into the website and has access to the product management interface.

Main Flow:

1. The admin navigates to the product creation page/form.
2. The admin enters the following product details:
 - Product image
 - Product name
 - Product price
 - Product category
 - Product description
 - Product usage
3. The admin submits the form.
4. The system sends a request to the products-service API to create a new product.
5. The products-service API processes the request and adds the new product to the database.

Postconditions: The new product is added to the product list and displayed on the home page.

Use Case 4: Retrieve Product Information

Actors: User (Visitor) / Admin

Preconditions: The user or admin wants to view product information.

Main Flow:

1. The system sends a request to the products-service API to retrieve product information.
2. The products-service API processes the request and retrieves the necessary product data from the database.
3. The system displays the retrieved product information.

Postconditions: The user or admin is able to view the requested product information.

Use Case 5: Update an Existing Product

Actors: Admin

Preconditions: The admin is logged into the website and has access to the product management interface.

Main Flow:

1. The admin selects a product to update.
2. The system retrieves the current product details from the products-service.
3. The admin modifies the necessary product details.
4. The admin submits the updated information.
5. The system sends a request to the products-service API to update the product information in the database.
6. The products-service API processes the request and updates the product details in the database.

Postconditions: The product information is updated in the product list and displayed on the home page

Use Case 6: Delete an Existing Product

Actors: Admin

Preconditions: The admin is logged into the website and has access to the product management interface.

Main Flow:

1. The admin selects a product to delete.
2. The system sends a request to the products-service API to delete the product from the database.
3. The products-service API processes the request and removes the product from the database.

Postconditions: The product is removed from the product list and no longer displayed on the home page.

Non Functional Requirements

Performance:

- The system should load the home page within 3 seconds under normal conditions.
- The product details page should load within 2 seconds under normal conditions.

Scalability:

- The system should handle up to 10,000 concurrent users without performance degradation.
- The products-service API should support up to 1,000 requests per second.

Availability:

- The system should have an uptime of 99.9% over a year.
- Scheduled maintenance should not exceed 2 hours per month and should be performed during off-peak hours.

Security:

- The system should use HTTPS for all communications.
- User data should be encrypted at rest and in transit.
- The system should implement role-based access control (RBAC) for admin functions.

Usability:

- The system should be usable and accessible according to WCAG 2.1 guidelines.
- The user interface should be intuitive and require no more than 3 clicks to reach any product detail page from the home page.

Compatibility:

- The website should be fully functional on major browsers (Chrome, Firefox, Safari, Edge) and their latest three versions.
- The website should be responsive and functional on various devices, including desktops, tablets, and smartphones.

Maintainability:

- The codebase should follow standard coding practices and be well-documented.
- The system should have automated tests covering at least 80% of the codebase.