

Beauty Product Catalog - Features Overview

1. Product Management

Create New Product

- Users can add new beauty products to the catalog.
- **Mandatory Fields:**
 - Brand
 - Name
 - Price
 - Category
- **Optional Fields:**
 - Image URL - Not yet implemented
 - Description
 - Rating
- Validation ensures that mandatory fields must be filled before adding a product.
- Validation ensures price>0 and rating>0
- Validation ensures no duplicates are added

Edit Product

- Users can modify the details of existing products.
- **Editable Fields:**
 - Brand
 - Name
 - Price
 - Description
 - Category
 - Rating
- Changes are saved instantly and reflected in the catalog.

Delete Product

- Users can remove a product from the catalog permanently.
- A confirmation prompt ensures that deletions are intentional.

2. Search & Filter Functionality

Search Feature

- Users can search for products using:
 - **Product Name**

- **Brand Name**
 - **Keywords in the Product Name**
- The search dynamically filters matching results in real time.

Filter Feature

- Users can refine product listings based on specific criteria.
 - **Category Filter:**
 - Options: All, Lipstick, Liquid, Powder, Palette, Pencil, Cream, Mineral
 - **Price Range Filter:**
 - Options: \$0-10, \$10-20, \$20-30, \$30-40, \$40+
 - **Rating Filter:**
 - Scale from **0 to 5**
 - Filters and search **can be used together** to refine product selection.
-

3. Product Viewing

Displayed Information

- **Mandatory Information (Always Visible):**
 - **Name**
 - **Price**
 - **Brand**
 - **Rating**
 - **Brief Description**
 - **Optional Information (Expandable Details):**
 - Ingredients
 - Usage Instructions
 - Tags (e.g., Paraben-Free, Vegan, Organic)
-

4. User Experience & Additional Features

- **Grid-Based Product Display:** Allows users to easily browse through multiple products.
 - **Fast Performance:** Optimized filtering, search, and image loading ensure a smooth experience.
 - **Interactive Details Panel:** Clicking on a product expands its details, showing both mandatory and optional information.
 - **Right-Click Context Menu:** Enables quick access to edit, delete, and view product details.
-

Presentation Summary for Customer

Create Products with validation for mandatory fields. **Edit and Update** product details seamlessly. **Delete Products** with confirmation. **Search by Name, Brand, and Keywords.** **Filter by Category, Price, and Rating (0-5).** **Combine Search & Filter** for refined results. **Display both mandatory & optional product details.**

Test cases :

Test Case ID	Test Scenario	Steps	Expected Outcome	Status
TC-001	Add a valid product	1. Open app 2. Enter details 3. Click Add	Product appears in the catalog and can be searched for	Pass
TC-002	Add a product with missing fields	1. Open app 2. Leave some fields empty 3. Click Add	Show error message and prevent addition	Pass
TC-003	Add duplicate product	1. Enter the same product twice 2. Click Add	Show error message and prevent addition	Pass
TC-004	Load products on startup	1. Open app	All previously saved products should appear	Pass
TC-005	Display empty catalog	1. Delete all products from JSON file 2. Open app	UI should display a message like "No current products"	Pending
TC-006	Edit an existing product	1. Select a product 2. Click Edit 3. Modify details 4. Click Save	Product details update in UI and JSON file	Pass
TC-007	Edit product with invalid input	1. Select a product 2. Click Edit 3. Enter invalid values (e.g., negative price) 4. Click Save	Show validation error	Pass
TC-008	Delete a product	1. Select a product 2. Click Delete	Product is removed from UI and JSON file	Pass
TC-010	Verify data persistence	1. Add products 2. Restart the app	Products should still be there	Pass

TC-011	Corrupt JSON file handling	1. Manually edit JSON file and insert invalid data 2. Restart app	Show error message or default to empty catalog	Pending
TC-012	Responsive UI test	Resize window, minimize, maximize	UI elements adjust properly	Pass
TC-013	Button click feedback	Click Add, Edit, Delete	Buttons provide visual feedback (e.g., highlight)	Pass
TC-014	Search for an existing product	1. Open app 2. Type product name in the search bar	Only matching products appear in the list	Pass
TC-015	Search for a non-existent product	1. Open app 2. Type a product name that does not exist	Display "No results found" message	Pending
TC-016	Case-insensitive search	1. Open app 2. Type the product name in lowercase/uppercase	Results should appear regardless of case	Pass
TC-017	Search by partial keyword	1. Open app 2. Enter part of a product name	All products containing that keyword appear	Pass
TC-018	Filter by category	1. Select a category from filter 2. Click Filter	Only products in the selected category appear	Pass
TC-019	Filter by price range	1. Set a min & max price range 2. Click Apply Filter	Only products within the range appear	Pass
TC-020	Filter by brand	1. Select a brand from filter 2. Click Apply Filter	Only products from selected brand appear	Pass
TC-021	Filter with no matching results	1. Select a category with no products 2. Click Filter	Display "No results found" message	Pending
TC-022	1. Click Clear Filters	All products reappear	Pending	Pass

Current Requirements List

Completed in iteration 1:

- Add, Update, Delete, products from catalog
- General UI and Product Display
- Produced test cases, recorded bugs, applied fixes accordingly
- Implemented basic error checking (text field validation, mandatory parameters)

To Complete for Iteration 2:

- Error Handling (duplicate products, more data validation)
- User login
- User logout
- User Sign up
- Enhance UI Aesthetics
- Upload pictures for adding new product
- Generate test case document for iteration 2

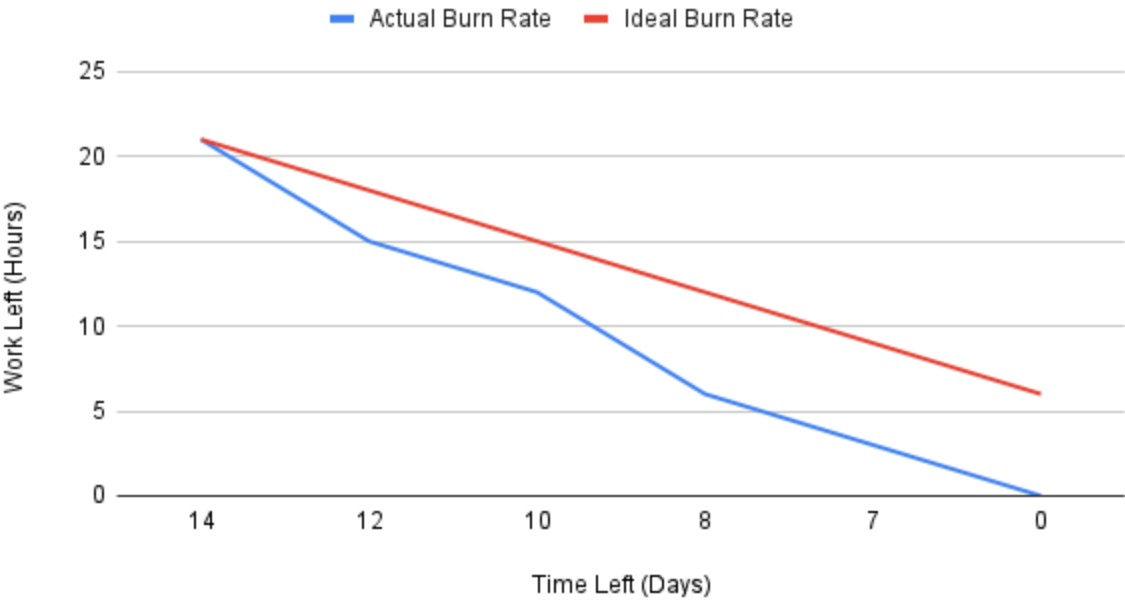
UML Diagram for Application

UML Diagram for - DELETE Product

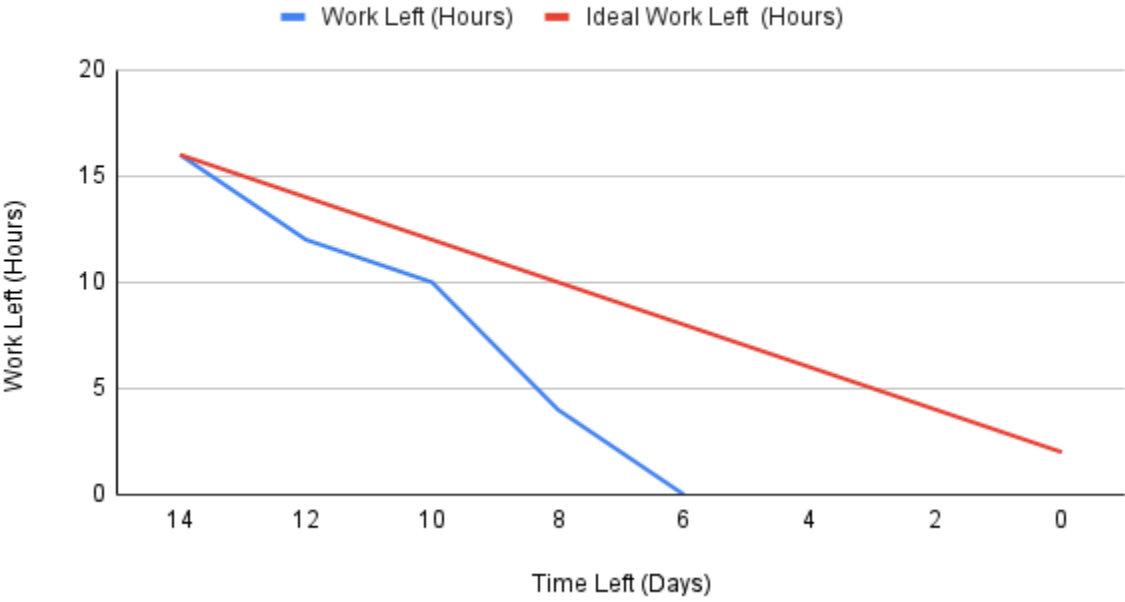
Trello Board - Riley

Burn Down Chart - Hanveer

Iteration #1 Burn Down Chart



Iteration #2 Burn Down Chart



Iteration 1 Retrospective

What went well:

- Generated accurate iteration plan
- Successful implementation of **add, edit, delete products, search, and filter.**
- Functional UI
- Quick fixes for bugs

What can be Improved:

- More clear iteration
 - Generate a requirements and parameters document before beginning development
 - Data Validation
 - Better tracking of work
-