Zewail City of Science, Technology and Innovation
University of Science and Technology
School of Computational Sciences and Artificial Intelligence
CSAI 202 - Fall 2023

CSAI 203 - Fall 2025
Introduction to Software Engineering

<Project Name: Cosmetics Store>

<Report Title: Design Document>
Team Number: 09

Team Members:

Rana Sherif Mohamed	**-**	**202401784**
Marwa Omar Youssef	**-**	**202401652**
Mariam Samir	**-**	**202200562**
Roaa Abdelghani Abdo	**-**	**202400980**

Design Document (SRS)

1. Introduction

1.1 Purpose

The purpose of this design document is to provide a detailed and structured description of the architecture, data structures, workflows, and user interface designs of the Online Cosmetics Store System. It serves as a guide for developers, testers, and stakeholders to understand how the system will be built and how components interact.

1.2 Scope

This phase covers the system's architecture design, module decomposition, MVC mapping, UML diagrams, detailed class structure, sequence diagrams, user interface wireframes, database schema, and data dictionary. The design focuses on both functional and structural blueprints required for implementation.

1.3 Intended Audience

This document is intended for:
- Software developers
- Quality assurance testers
- System analysts and designers

1.4 Overview of Contents

This document contains detailed descriptions of system components, architecture diagrams, MVC application, sequence diagrams, user interface wireframes, and database schema.

- # 2. System Overview

2.1 Brief Description of the System

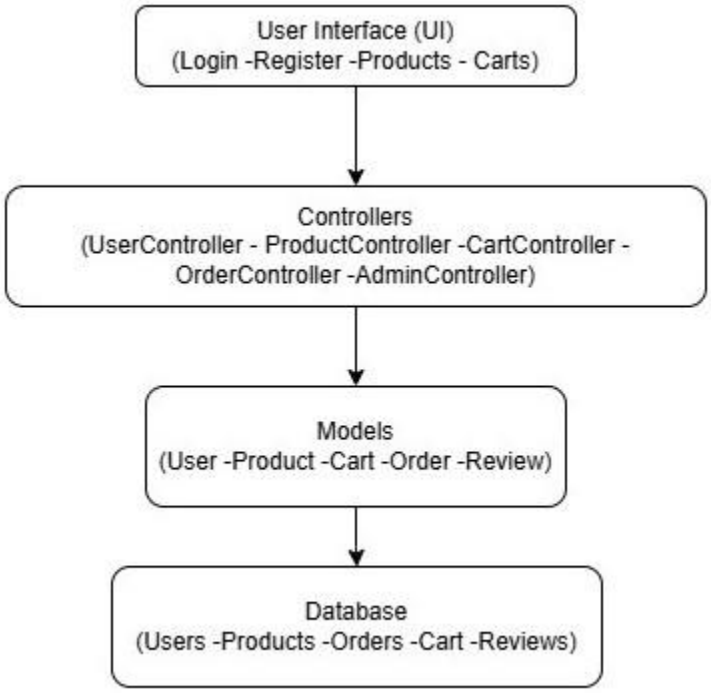
The Online Cosmetics Store is a web-based platform that enables customers to browse, search, and purchase cosmetic products. Users can manage their accounts, shopping cart, favorites, and orders, while administrators can manage products, inventory, and monitor sales analytics.

2.2 *Key Design Goals and Constraints*

- Scalability
- Maintainability
- High usability and intuitive interface
- Secure handling of user data and payment
- Efficient database design and optimized queries

3. Architectural Design

3.1 System Architecture Diagram



3.2 *Discussion of Architectural Style and Components*

The system uses a Monolithic Architecture with clear modular separation between the UI, controllers, models, and database. Components include:
- Presentation Layer (Views)
- Application Logic (Controllers)
- Data Layer (Models + Database)

3.3 Technology Stack and Tools

- Python (Flask)
- HTML, CSS, JavaScript
- CSV File Database
- Draw.io for diagrams
- Draw.io for UI mockups
- Draw.io for UML diagrams

4. Detailed Design

4.1 Model–View–Controller (MVC) Design Pattern

- ### *4.1.1 Description of MVC Pattern*

  The MVC pattern separates the system into three main components: the Model, which handles data and business logic; the View, which manages the user interface; and the Controller, which processes user input and coordinates between the View and Model

- ***4.1.2 Mapping of Project Components to MVC***

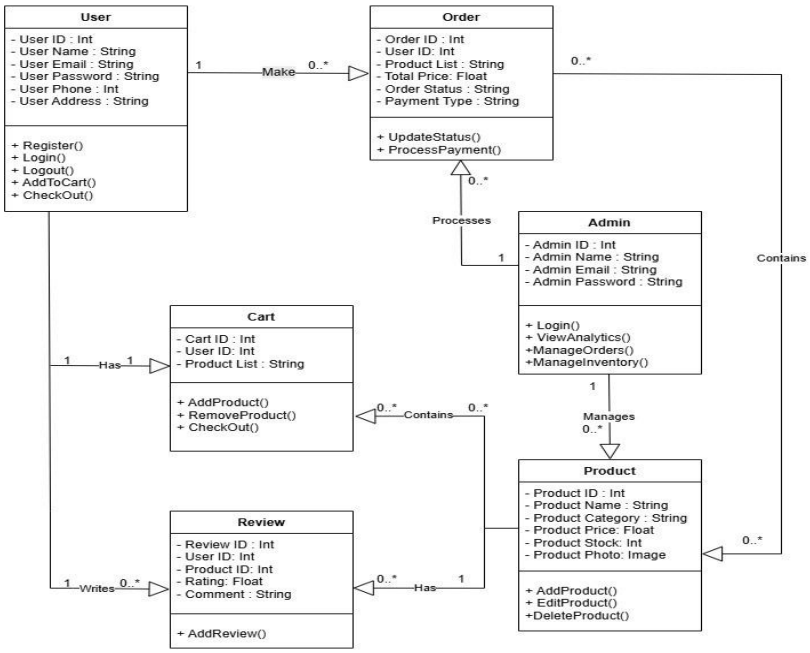
Component	**Modules / Classes**
Model	User, Admin, Product, Cart, Order, Review
View	All HTML Templates: Login, Product Page, Cart, Checkout, Admin Dashboard
Controller	AuthController,ProductController, CartController, OrderController, AdminController

- ***4.1.2 Responsibilities of Each Component***

- Model: Stores data, validates input, and applies business rules.
- View: Displays data and collects user interactions.
- Controller: Handles input, sends requests to Models, updates Views.

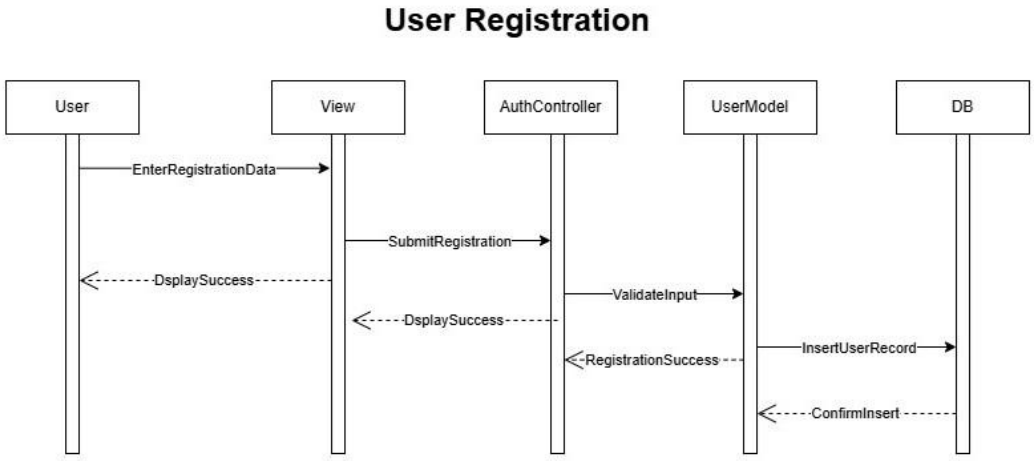
4.2 UML Diagrams

- ### *4.2.1 Detailed Class Diagram*

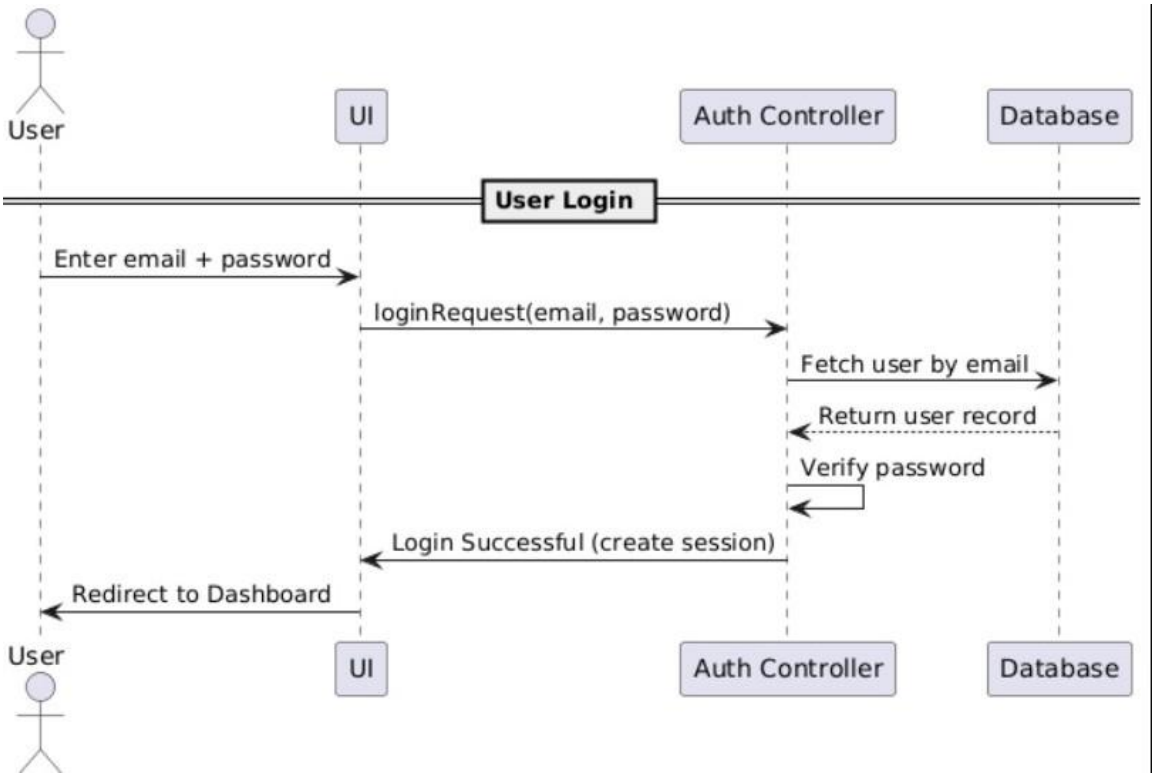


- ### *4.2.2 Sequence Diagrams*

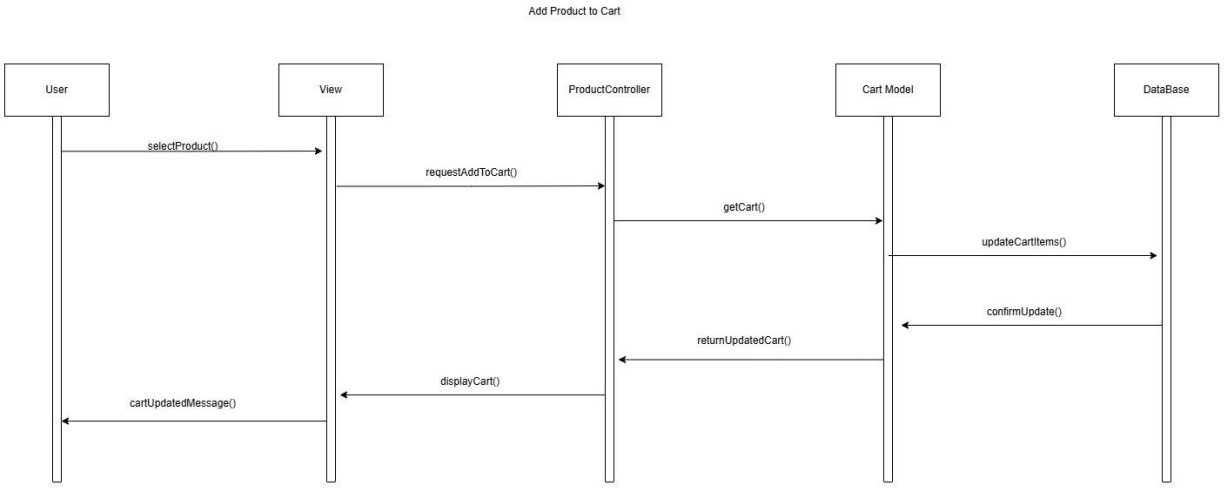
Sequence Diagram 1: User Registration:



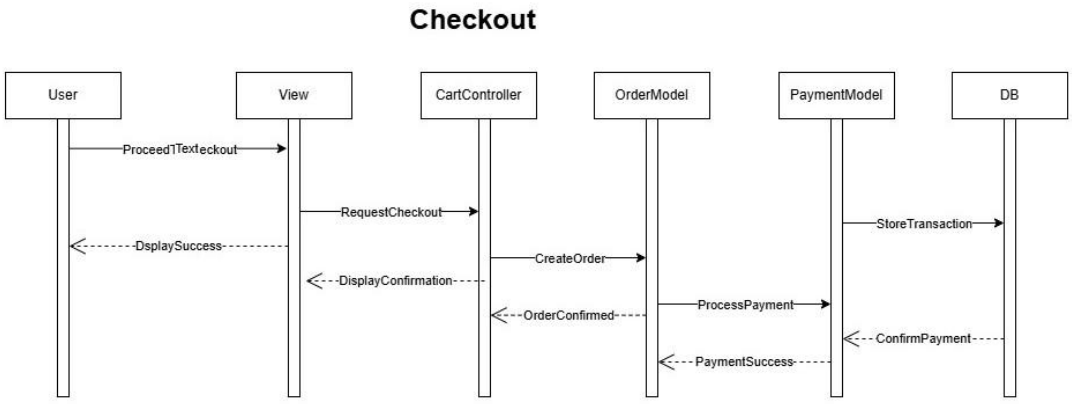
Sequence Diagram 2: User Login:



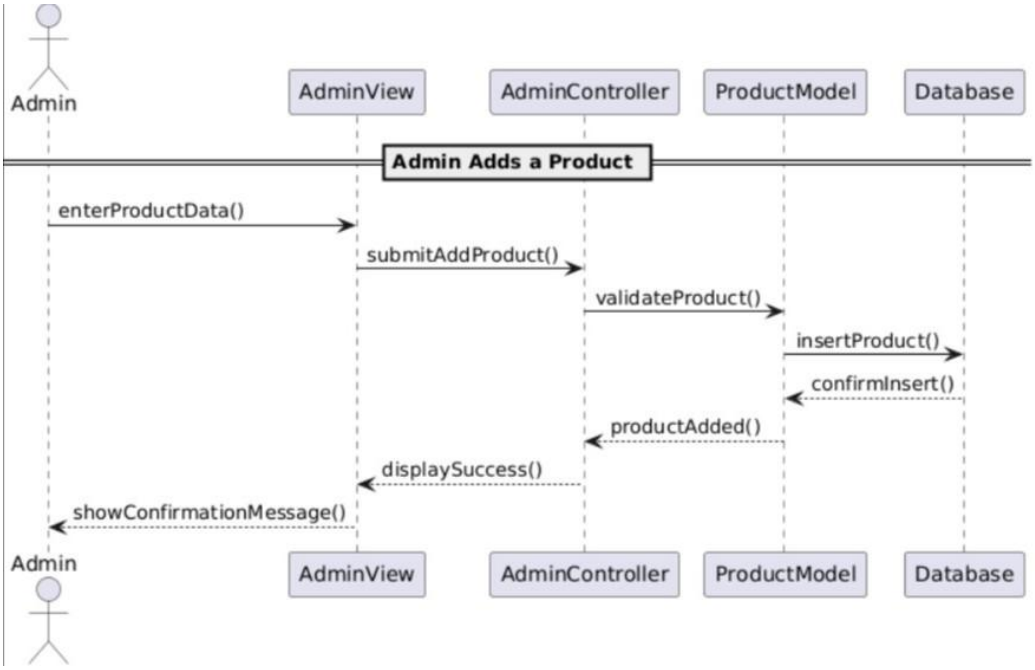
Sequence Diagram 3: Add Product to Cart:



Sequence Diagram 4: Checkout:



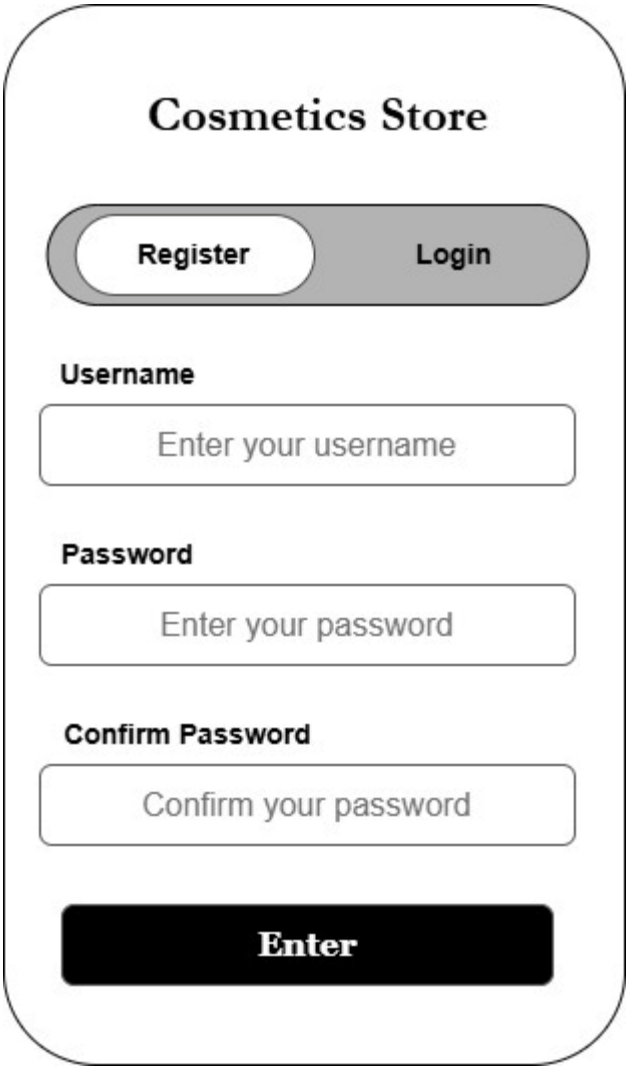
Sequence Diagram 5: Admin Adds a Product:



4.3 UI/UX Design

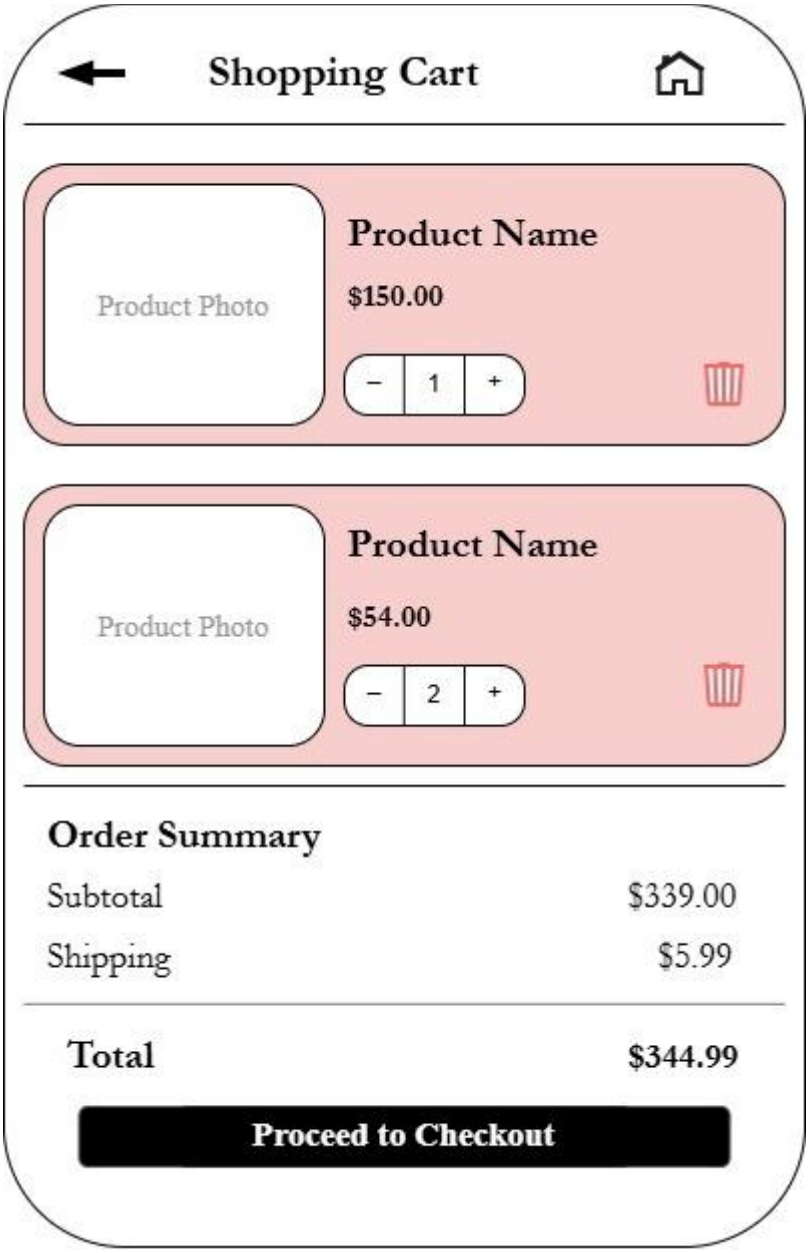
- ***4.3.1 Wireframes / Mockups***

Login page:

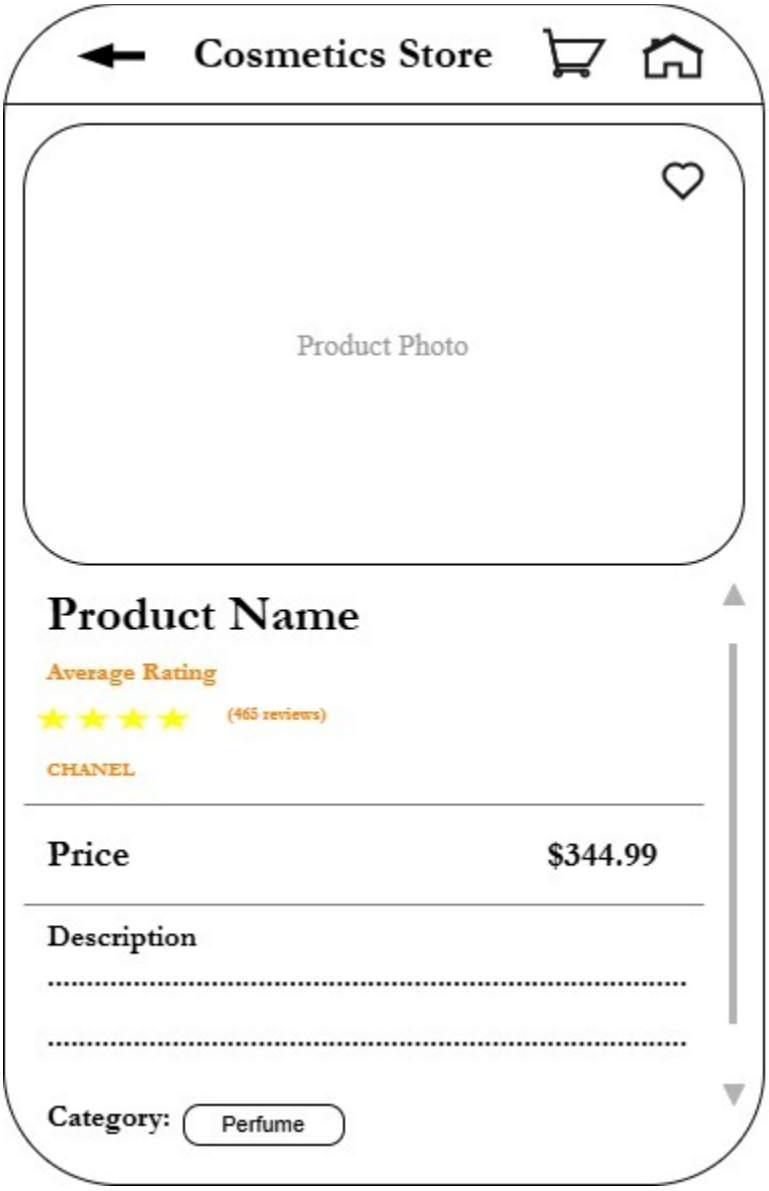


Product listing page frame:

Cart Page Frame:



Product details page frame:



- ***4.4.2 File Structure / Data Storage Model***

Users.CSV File

Column	Data Type	Purpose
product_id	Integer	Product identifier
name	String	Product name
category	String	Product category
description	String	Detailed info
price	Float	Price of product
stock	Integer	Quantity available
image	String	File path of product image
created_at	String	Date product was added

Products.CSV File

Column	Data Type	Purpose
product_id	Integer	Product identifier
name	String	Product name
category	String	Product category
description	String	Detailed info
price	Float	Price of product
stock	Integer	Quantity available
image	String	File path of product image
created_at	String	Date product was added

Cart.csv File

Column	Data Type	Purpose
cart_id	Integer	Unique cart ID

user_id	Integer	User who owns the cart
created_at	String	Timestamp

Cart_items.CSV File

Column	Data Type	Purpose
item_id	Integer	Unique item record
cart_id	Integer	Reference to cart
product_id	Integer	Product added
quantity	Integer	Quantity selected

Orders.CSV File

Column	Data Type	Purpose
order_id	Integer	Order identifier
user_id	Integer	Customer who made the order
total_amount	Float	Total price
payment_type	String	"cash" or "credit"
status	String	Order status
address	String	Delivery address
created_at	String	Order timestamp

Favorites.csv File

Column	Data Type	Purpose
fav_id	Integer	Entry ID
user_id	Integer	User who favorited
product_id	Integer	Favorited product

Reviews.CSV File

Column	Data Type	Purpose
review_id	Integer	Review ID

user_id	Integer	Reviewer
product_id	Integer	Reviewed product
rating	Integer	Rating (1–5)
comment	String	Optional user review
created_at	String	Review date

Admin_notifications.CSV File

Column	Data Type	Purpose
noti_id	Integer	Unique notification ID
message	String	Notification text
product_id	Integer	Related product (optional)
created_at	String	Notification date

- ***4.4.3 Data Dictionary***

Field	Type	Description

user_id	INT	Primary key for user.
name	VARCHAR	Customer full name.
email	VARCHAR	Unique email address.
password	VARCHAR	Encrypted password.
phone	VARCHAR	User contact number.

5. Conclusion

This design document provides a comprehensive representation of the system architecture, UML diagrams, MVC structure, data design, and user interface wireframes required for the development of the Online Cosmetics Store.