SOFTWARE REQUIREMENTS SPECIFICATION

for

Decked Out

Version 1.1 approved

Prepared by:

Joshua Kelley
Michelle Hedrick
Christopher Sisk

CSE-4214-6214

October 3, 2025

Version: 1.1　　　　　Revision Date: October 3, 2025

Contents

1. Introduction

1.1. Purpose

The purpose of this SRS document is to provide a detailed description of the objectives, scope, and intended use for Decked Out, an e-commerce platform dedicated to providing customers with high-quality skateboarding gear and accessories. This document establishes a clear understanding of the system's objectives, features, and constraints to ensure that all stakeholders, including developers, designers, and testers share the same vision for the application.g developers, designers, and testers share the same vision for the application.

1.2. Document Conventions

Red text is used to denote high priority objectives/requirements.

   Blue text is used to denote currently implemented objectives/requirements.

1.3. Intended Audience and Reading

The intended audience of this document includes all stakeholders of the project, including but not limited to developers, project managers, marketing staff, users, testers, and documentation writers. Each section of this SRS is organized to provide a comprehensive understanding of the software requirements, starting with an overview and progressing to detailed specifications. Readers are encouraged to begin with the introduction and overall description before delving into specific system features and nonfunctional requirements relevant to their roles.

1.4. Product Scope

Decked Out is an e-commerce platform designed to provide skateboarders with an accessible, user-friendly, and secure online marketplace for buying skateboards, parts, apparel, and accessories. The system will allow customers to browse diverse product catalog, manage shopping carts, complete secure transactions, and track orders through an intutive web interface

1.5. References

- Python 3.11.4 documentation. https://docs.python.org/3/.

- mySQL. https://docs.oracle.com/en/database/mysql/otr-installation/index.html#GUID-B41949CF-5344-42E0-9736-B1F3DB1D3D63

- Django 4.2 documentation. https://docs.djangoproject.com/en/4.2/.

2. Overall Description

2.1. Product Perspective

Decked Out is a new, self-contained e-commerce platform and will be built from the ground up. It will be developed using Django for the backend, MySQL as the primary database, and HTML, CSS, and JavaScript for the frontend. Django provides rapid development, and built in security features, while MySQL offers reliable and efficient data management.

2.2. Product Functions

- User Registration and Login: Users can create accounts, log in, and log out securely.

- Product Browsing and Search: Users can browse product categories and search for specific items using various filters.

- Shopping Cart Management: Users can add, view, update, and remove items from their shopping cart.

- Checkout and Payment Processing: Users can enter shipping and payment information to complete purchases securely.

- Admin Management: Administrators can manage user accounts, product listings, and view site analytics.

- Updating Product: Sellers can manage their product listings by adding, updating, or removing items.

2.3. User Classes and Characteristics

Buyers are the primary users of the system. They will browse products, and purchase skateboards, accessories, and apparel. They are expected to have basic computer and mobile device literacy.

Sellers are users who list products for sale on the platform. They can be store employees or 3rd party sellers. They need moderate technical skills to manage their listings and view orders.

Administrators are responsible for managing the platform, including user accounts and product listings. They require advanced technical skills in SQL and web adminstration.

2.4. Operating Environment

Decked Out will operate as a web service that is supported on major web browser including Chrome, Firefox, Safari, and Edge. The system will be accessible from desktop and mobile devices. The backend will be hosted on an environment with Django and SQL. Django will manage the application logic, while SQL will store product, order, and customer data.

2.5. Design and Implementation Constraints

- Coding standards: Code must follow Python and Django practices.

- Performance: Must handle up to 50 concurrent users.

- Web Standards: Compatible with HTML5 and modern browsers (Chrome, Firefox, Safari, Edge).

3. System Features

3.1. Customer Account Creation

3.1.1. Description and priority

Allow customers to create, view, update, and delete their accounts. Priority: High **3.1.2. Stimulus/Response Sequences**

- Customer creates their account: The system stores customer information and confirms successful account creation
- customer logs in: the system verifies the customer's credentials and grants access to account features.
- customer updates account information: the system saves changes and confirms the update to the customers account information
- customer deletes their account: The system removes customer data and confirms account deletion.
- customer views past orders: the system displays a list of previous orders with details of the customer's past purchases.

3.1.3. Functional Requirements

- REQ-1: The system shall allow customer to create new accounts with a username, password, and email address.
- REQ-2: The system shall allow customer to log in and log out securely.
- REQ-3: The system shall allow customer to update their personal information.
- REQ-4: The system shall allow customer to delete their accounts.
- REQ-5: The system shall allow the customer to display past orders.

3.2. Sellers Account Creation

3.2.1. Description and priority

Allow sellers to create, view, update, and delete, and add products to their accounts. Priority: High **3.2.2. Stimulus/Response Sequences**

- Seller creates their account: The system stores Seller information and confirms successful account creation
- Seller logs in: the system verifies credentials and grants access to account features.

- Seller uddates account information: the system saves changes and confirms the update
- Seller deletes their account: The system removes Seller data and confirms account deletion.
- Seller adds products: The system is able to add their products to the database to be viewed on the site.
- Seller removes products: The system is able to remove their products that are listed on the site.

3.2.3. Functional Requirements

- REQ-1: The system shall allow sellers to create new accounts with a username, password, and email address.
- REQ-2: The system shall allow sellers to log in and log out securely.
- REQ-3: The system shall allow sellers to update their personal stock.
- REQ-4: The system shall allow sellers to delete their products/stocks.
- REQ-5: The system shall allow sellers to display products and accessories.

3.3. Admin Account Creation

3.3.1. Description and priority

Allow Admin to create, view, update, and delete their accounts. Also manage user and seller accounts and databases Priority: High

3.3.2. Stimulus/Response Sequences

- Admin creates their account: the system creates an admin account from the admin exclusive page and is assigned admin previlages.
- admin logs in: the system verifies admin credentials and grants access to account features.
- Admin views account information for other accounts: the system allows admin accounts to make changes to customer and seller accounts.

3.3.3. Functional Requirements

- REQ-1: The system shall allow admin to create new admin accounts with a username, password, and email address.
- REQ-2: The system shall allow admin to log in and log out of their admin account securely.
- REQ-3: The system shall allow admin to view seller and users account information.
- REQ-4: The system shall allow admin to manage seller and user accounts.
- REQ-5: The system shall allow admin to manage the product database.

3.4. Shopping Cart

3.4.1. Description and priority

Allow Buyers to add, view, update, and remove items from their shopping cart. Priority: Medium **3.4.2. Stimulus/Response Sequences**

- User adds an item to the cart: The system updates the cart and displays the new item count, item details, and total price.
- User views the cart: The system displays all items in the cart with their details.
- User updates item quantity: The system adjusts the item quantity and recalculates the total price.
- User removes an item: The system deletes the item from the cart and updates the total price.

3.4.3. Functional Requirements

- REQ-6: The system shall allow users to add items to their shopping cart.
- REQ-7: The system shall display the contents of the shopping cart, including item details and total price.
- REQ-8: The system shall allow users to update the quantity of items in their cart.
- REQ-9: The system shall allow users to remove items from their shopping cart.
- REQ-10: The system shall recalculate the total price whenever items are added, updated, or removed.

3.5. Product Listing

3.5.1. Description and priority

Allow sellers to add, view, update, and remove products from their product listings. Priority: High **3.5.2. Stimulus/Response Sequences**

- Seller adds an item to their product listings: The system updates the product listings and displays the new added product.
- Seller removes an item from their product listing: The system deletes the item from the product listings.
- Seller adds/updates product price or description: The system adjusts the product details and updates the product listing.
- Seller adds product image: The system uploads the image and associates it with the product listing.

3.5.3. Functional Requirements

- REQ-1: The system shall allow sellers to add items to their product listings.
- REQ-2: The system shall allow sellers to remove items from their product listings.
- REQ-3: The System shall allow sellers to add/update product price or description.
- REQ-4: The system shall allow sellers to add product image.

3.6. Return products

3.6.1. Description and priority

Allow Buyers to return products they have purchased. Priority: Medium

3.6.2. Stimulus/Response Sequences

- User initiates a return: The system verifies the purchase and processes the return request.
- User views return status: The system displays the current status of the return request.
- User receives refund: The system processes the refund and updates the user's account balance.

3.6.3. Functional Requirements

- REQ-1: The system shall allow users to initiate a return for purchased items.
- REQ-2: The system shall verify the purchase before processing the return.
- REQ-3: The system shall allow users to view the status of their return requests.
- REQ-4: The system shall process refunds for returned items and update user account balances accordingly.

4. Other Nonfunctional Requirements

4.1. Performance Requirements

- Clicking any link for a item should take less than 1 second
- Adding an item to the car should take less than 1 second.

- The system should be able to handle 50 concurrent users.
- The system should be able to handle up to 10,000 products and 5,000 users accounts without performance lost

4.2. Safety Requirements

- The user login information should be secure enough so that anyone except for the developers will know that user's username and password.
- The user personal information should be secure enough so that no 3rd party can access it.
- The payment information should be secure enough so that no 3rd party can access
- The system shall implement account lockout mechanisms after multiple failed login attempts to prevent brute-force attacks.
- The system shall regularly update and patch all software components to address known vulnerabilities.

4.3. Security Requirements

- authentication: the system shall require users to input username and password to access their accounts.
- session management: the system shall automatically log out users after 15 minutes of inactivity.
- Data privacy: the system shall allow users to delete their accounts and personal data upon request.

4.4. Software Quality Attributes

- Usability: The system shall have an intuitive user interface that allows users to easily navigate and perform tasks with minimal training.
- Reliability: The system shall have an uptime of 99.9% and be able to recover from failures within 5 minutes.
- Maintainability: The system's codebase shall be modular and well-documented to facilitate future updates and maintenance.
- Scalability: The system shall be designed to handle increased load by adding resources without significant performance degradation.