

# Software Requirements Specification (SRS)

for

Urban Style – Online Clothing Ecommerce Platform

## 1. Introduction

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to describe the functional and non-functional requirements of the Urban Style ecommerce system. This document provides a foundation for the development, validation, and quality assurance of the platform. It is intended for stakeholders, developers, testers, business analysts, and project evaluators.

### 1.2 Scope

Urban Style is an online fashion shopping platform that allows customers to browse clothing products based on their body shape, add items to a cart, securely checkout using Stripe, and manage their user profile.

The system includes:

User Registration & Login (Firebase Authentication)

Product Display

Shopping Cart Management

Online Payments (Stripe Integration)

Order Tracking

User Profile Management

The goal is to provide a secure, user-friendly experience available on both desktop and mobile devices.

### 1.3 Definitions, Acronyms & Abbreviations

| Term | Meaning |
|------|---------|
|------|---------|

|          |  |
|----------|--|
| SRS      | Software Requirements Specification                        |
| UI       | User Interface   |
| DB       | Database   |
| Firebase | Cloud backend platform used for Authentication & Firestore |
| Stripe   | Payment Processing Platform                                |
| CRUD     | Create, Read, Update, Delete                               |

#### 1.4 References

IEEE 830 Standard — Software Requirements Documentation

Firebase Documentation (Auth + Firestore)

Stripe Developer Documentation

React JS & Vite Official Documentation

#### 1.5 Overview

This document outlines the system context, detailed requirements, constraints, design implications, diagrams, and validation approach for the successful implementation of Urban Style.

### SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Urban Style – Online Clothing Shopping Platform

## 2. Overall Description

### 2.1 Product Perspective

Urban Style is a standalone ecommerce web application developed using:

| Component | Technology |
|-----------|------------|
|-----------|------------|

|                 |  |
|-----------------|--|
| Frontend        | React (Vite + Tailwind CSS)                  |
| Backend         | Firebase Authentication + Firestore Database |
| Payment Gateway | Stripe                                       |
| Hosting         | Firebase Hosting (Future Scope)              |

The platform interacts with:

Firebase Authentication → Secure user login/registration

Firestore → Stores user profiles & orders

Stripe → Secure card payments

## 2.2 Product Features

Create an account and login securely

Browse and search clothing items

Add/remove items from cart

Pay securely using Stripe

View order history

Update personal information

## 2.3 User Classes & Characteristics

| User Type            | Description                                      | Technical Expertise          |
|----------------------|--|------------------------------|
| Customer             | Primary users who browse and shop clothing items | Basic computer/mobile skills |
| Admin (Future Scope) | Manages products and orders                      | Intermediate                 |

## 2.4 Operating Environment

|                    |                                 |
|--------------------|---------------------------------|
| Category           | Specification                   |
| Supported Browsers | Chrome, Edge, Firefox, Safari   |
| Devices            | Desktop, Tablets, Mobile Phones |
| OS                 | Windows, macOS, iOS, Android    |
| Internet Speed     | Stable connection required      |

## 2.5 Design & Implementation Constraints

| Constraint              | Explanation                                 |
|-------------------------|---|
| Stripe Payments         | Only real card details in production mode   |
| Firebase Service Quotas | Limited by free tier for development        |
| Internet Connection     | System can't function offline               |
| Security Compliance     | Sensitive data handled by Firebase & Stripe |

## 2.6 User Documentation

Screens and guidance will be provided through:

Help/FAQs (Future Enhancement)

User-friendly UI with error messages (Future Enhancement)

## 2.7 Assumptions & Dependencies

Users have valid email addresses and payment details

Stripe API key must remain secure

Browsers must support modern JavaScript and HTTPS

### 3. System Features

Urban Style includes multiple functional modules.

We will document them separately:

| Feature ID | Feature Name               | Status       |
|------------|----------------------------|--------------|
| SF-01      | User Registration & Login  | Current      |
| SF-02      | User Profile Management    | Current      |
| SF-03      | Product Browsing & Search  | Current      |
| SF-04      | Shopping Cart              | Current      |
| SF-05      | Secure Stripe Payment      | Current      |
| SF-06      | Order Management           | Current      |
| SF-07      | Admin Product Management   | Future Scope |
| SF-08      | Admin Order Administration | Future Scope |

#### SF-01: User Registration & Login

##### Description

Allows customers to create accounts and log in securely.

##### Functional Requirements

| ID | Requirement |
|----|-------------|
|----|-------------|

|      |  |
|------|--|
| FR-1 | The system shall allow users to register with email and password.          |
| FR-2 | The system shall store user details in Firestore.                          |
| FR-3 | The system shall validate password length (minimum 6 characters).          |
| FR-4 | The system shall allow users to log in using email and password.           |
| FR-5 | The system shall return appropriate error messages for invalid login info. |

#### Use Case – UC01: Account Registration

| Item           | Description   |
|----------------|---|
| Actor          | Customer  |
| Precondition   | User is not logged in   |
| Main Flow      | <ol style="list-style-type: none"> <li>1. User enters full name, email, password and address</li> <li>2. System registers user</li> <li>3. User profile saved in Firestore</li> <li>4. System redirects to profile</li> </ol> |
| Post-condition | User account exists in Firebase Auth  |

#### SF-02: User Profile Management

| Requirement ID | Requirement  |
|----------------|--|
| FR-6           | Users shall be able to update profile details          |
| FR-7           | Users shall be able to view past orders                |
| FR-8           | Users shall be able to logout and clear stored session |

#### Use Case UC02: Update Profile

|         |                                 |
|---------|---------------------------------|
| Actors  | Customer                        |
| Outcome | Updated data saved to Firestore |

#### SF-03: Product Browsing & Search

| Requirement ID | Requirement   |
|----------------|---|
| FR-9           | Users shall browse product categories                             |
| FR-10          | Users shall filter/search items by keywords                       |
| FR-11          | System shall show product details including price, images & sizes |

#### SF-04: Shopping Cart

| Requirement ID | Requirement                                     |
|----------------|---|
| FR-12          | Users shall add products to cart                |
| FR-13          | Users shall update/remove cart items            |
| FR-14          | System shall calculate total amount dynamically |
| FR-15          | Cart must persist during navigation             |

#### SF-05: Secure Stripe Payment

| Requirement ID | Requirement                                 |
|----------------|---|
| FR-16          | System shall allow payments via Stripe      |
| FR-17          | Users shall receive error on failed payment |

|       |  |
|-------|--|
| FR-18 | System shall store successful payment + order details in Firestore |
|-------|--|

#### SF-06: Order Management

| Requirement ID | Requirement  |
|----------------|--|
| FR-19          | System shall allow users to view their order history |
| FR-20          | Order record shall include purchased items and price |

#### SF-07: Admin Product Management (Future Scope)

| Requirement ID | Requirement   |
|----------------|---|
| FR-21          | Admin shall upload product details including images           |
| FR-22          | Admin shall edit or delete products                           |
| FR-23          | System shall restrict admin privileges to authorized accounts |

#### SF-08: Admin Order Administration (Future Scope)

| Requirement ID | Requirement  |
|----------------|--|
| FR-24          | Admin shall view all orders placed                                 |
| FR-25          | Admin shall update order status (Processing, Delivered, Cancelled) |

### 4. Non-Functional Requirements (NFRs)

These define how the Urban Style system should perform and behave.



#### 4.1 Performance Requirements

| Req ID | Requirement   |
|--------|---|
| PR-01  | The system shall load all pages within 3 seconds on a stable internet connection.     |
| PR-02  | The system shall handle up to 500 simultaneous users without performance degradation. |
| PR-03  | The system shall process Stripe transactions within 5 seconds on average.             |
| PR-04  | The cart updates (add/remove items) shall reflect immediately without page reload.    |

#### 4.2 Security Requirements

| Req ID | Requirement   |
|--------|---|
| SR-01  | The system shall use Firebase Authentication for secure login and registration. |
| SR-02  | Passwords shall not be stored in plaintext anywhere in the system.              |
| SR-03  | The system shall enforce secure HTTPS connections for all payment operations.   |
| SR-04  | Only authenticated users may add items to the cart or place orders.             |
| SR-05  | Admin-level features must be role-restricted (Future Scope).                    |

#### 4.3 Usability Requirements

| Req ID | Requirement  |
|--------|--|
| UR-01  | The UI shall be responsive and mobile-friendly across all devices.           |
| UR-02  | The system shall provide clear error messages for invalid actions.           |
| UR-03  | The system navigation shall be intuitive with a visible Navbar on all pages. |

|       |  |
|-------|--|
| UR-04 | Users shall be able to complete a purchase within 5 clicks after adding items to cart. |
|-------|--|

#### 4.4 Reliability & Availability

| Req ID | Requirement  |
|--------|--|
| RA-01  | The system shall be available 99% of the time, excluding planned maintenance.        |
| RA-02  | Urban Style shall automatically reconnect if network connection is temporarily lost. |
| RA-03  | User carts shall remain intact even if the browser refreshes during checkout.        |

#### 4.5 Maintainability Requirements

| Req ID | Requirement  |
|--------|--|
| MR-01  | The system's code shall be modular and separated into reusable components.     |
| MR-02  | Updates to a component must not break others due to React component isolation. |
| MR-03  | System documentation shall be maintained for future improvements.              |

#### 4.6 Scalability Requirements

| Req ID | Requirement  |
|--------|--|
| SC-01  | The product database and user accounts shall scale automatically using Firebase.             |
| SC-02  | System architecture shall support future expansion including admin dashboards and analytics. |

#### 4.7 Payment Compliance

| Req ID | Requirement  |
|--------|--|
| PC-01  | Stripe integration shall comply with PCI DSS security standards. |
| PC-02  | No card details shall be stored in Urban Style systems.          |