**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**for**

**<PROJECT NAME>**  
Version 1.0

***By***

**Student Name 1 CIIT/SP09-BCS-xxx/ATD**

**Student Name 2 CIIT/SP09-BCS-xxx/ATD**

***Supervisor*Supervisor Name**

***Bachelor of Science in Computer Science (20xx-20xx)***

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 6**

2.1 Product perspective 6

2.2 Operating environment 6

2.3 Design and implementation constraints 7

**3.** **Requirement identifying technique 8**

3.1 Use Case Diagram 8

3.2 Use Case Description **Error! Bookmark not defined.**

**4.** **Specific Requirements 39**

4.1 System feature X 39

**5.** **Quality attributes** Error! Bookmark not defined.

5.1 Usability 40

5.2 Performance **Error! Bookmark not defined.**

**6.** **External interface requirements 35**

6.1 User interfaces 35

6.2 Software interfaces 35

6.3 Hardware interfaces 35

6.4 Communications interfaces 35

**7.** **Project Gantt chart 35**

**8.** **References 41**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Introduction

The Software Requirements Specification (SRS) for the AR-based Online Furniture Store outlines the essential requirements for the development and deployment of the system. This document serves as a comprehensive guide for the project's stakeholders, including developers, designers, testers, and clients, providing a clear understanding of the functional and non-functional requirements necessary to achieve the project’s goals.

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to identify and detail the requirements for the AR-based Online Furniture Store application. This application is designed to revolutionize the online shopping experience by integrating Augmented Reality (AR) technology, enabling customers to visualize and interact with virtual 3D furniture models in their real-world environment before making a purchase.

## Scope

The AR-based Online Furniture Store application designed to enhance the furniture shopping experience by integrating Augmented Reality (AR) technology. This application allows customers to visualize and interact with virtual 3D models of furniture in their real-world environment using their mobile device cameras. By providing this immersive and interactive experience, the application aims to help customers make well-informed purchasing decisions and bridge the gap between online and offline shopping.

## Major Features:

## 3D Model Creation and Management:

* Creating, uploading, and managing 3D furniture models.
* Ensures accurate, high-quality representations.

## Augmented Reality Preview Mode:

* Visualizes 3D furniture models in real environments.
* Interactive features for rotation, scaling, and placement.

## Admin Panel:

* Manages product listings, user accounts, and orders.
* Tools for tracking inventory, sales, and managing feedback.

## User Registration and Authentication:

* Secure account creation and login.
* Supports user profiles and personalized experiences.

## Furniture Listing:

* Searchable, filterable catalog with descriptions, images, and 3D views.
* Categories and tags for easy product discovery.

## Customer Reviews Management:

* Features for leaving and viewing product reviews and ratings.
* Admin moderation tools to ensure quality and relevance.

## Secure Payment:

* Integration with reliable payment gateways.
* Supports multiple payment methods and ensures security compliance.

## Purpose:

* The main purpose of our application includes:
* Utilize AR technology for immersive shopping experience.
* Enable users to view and interact with 3D product models in their own space.
* Empower users to make informed purchasing decisions.
* Revolutionize online shopping experience.

**Overall description**

## Product perspective

The AR-based Online Furniture Store, inspired by international projects like IKEA Place, introduces advanced augmented reality capabilities to the Pakistani market, where such technology is currently unavailable in online furniture shopping. This platform enhances the shopping experience by allowing customers to visualize and interact with 3D furniture models in their physical environments using mobile device cameras. By bridging the gap between online and offline shopping, the project leverages global best practices and innovations to offer Pakistani consumers a novel, immersive, and informed way to purchase furniture.

## Operating environment

The AR-based Online Furniture Store application is designed for optimal performance across various environments:

* **Hardware Platform:** Compatible with desktops, laptops, tablets, and smartphones. AR features require devices with cameras supporting augmented reality.
* **Operating Systems:** Supports Windows, macOS, iOS, and Android, ensuring broad accessibility.
* **Geographical Locations:** Designed for global use, adapting to different locales and languages.
* **Servers and Databases:** Utilizes secure, scalable cloud platforms (AWS, Azure, Google Cloud) with Node.js and Express for the backend, and PostgreSQL for database management.
* **Web-Based Interaction:** Accessible via modern web browsers: Google Chrome (all versions), Mozilla Firefox (version 45+), Microsoft Edge (all versions), Apple Safari (version 10+).
* **Data Hosting:** Hosted by organizations adhering to stringent data security protocols and regulations (GDPR, CCPA).
* **Network Requirements:** Requires a stable internet connection for smooth operation, optimized for variable network conditions.

**Design and implementation constraints**

**CO-1**: The application must use Next.js for frontend development.

**CO-2**: Node.js and Express are required for backend development.

**CO-3**: PostgreSQL must be used as the database engine.

**CO-4**: AR functionality must be implemented using Unity.

**CO-5**: 3D models must be managed using Three.js.

**CO-6**: 3D models must be created using Blender.

**CO-7**: The development process must follow Agile methodology.

**CO-8**: All source code must be version controlled using Git and GitHub.

**CO-9**: The application must ensure compatibility with modern mobile and desktop browsers.

**CO-10**: Secure payment integration must use established and trusted payment gateways.

A diagram of a company

Description automatically generated**Requirement identifying technique.**

Figure 1: Use Case Diagram

|  |  |
| --- | --- |
| **Use Case ID:** | UC-001 |
| **Use Case Name:** | Register |
| **Actors:** | Visitor |
| **Description:** | Allows a visitor to create a new account by providing necessary details. |
| **Trigger:** | Visitor chooses to sign up for a new account. |
| **Preconditions:** | The visitor is not logged in. |
| **Postconditions:** | A new user account is created, and the visitor is logged in. |
| **Normal Flow:** | 1. Visitor navigates to the registration page. 2. Visitor enters required details (e.g., name, email, password). 3. Visitor submits the registration form. 4. System validates the details. 5. System creates a new user account. 6. System logs the user in and redirects to the home page. |
| **Alternative Flows:** | If the email is already in use, the system prompts the visitor to use a different email or recover the existing account. |
| **Exceptions:** | 1. Network failure during registration. 2. Validation errors (e.g., invalid email format, weak password). |
| **Business Rules** | Password must meet security criteria (e.g., minimum length, contains letters and numbers). |
| **Assumptions:** | Users will have access to a valid email address for registration. |

## Use Case: Register

## Use Case: Login

|  |  |
| --- | --- |
| **Use Case ID:** | UC-002 |
| **Use Case Name:** | Login |
| **Actors:** | Registered User |
| **Description:** | Allows a registered user to access their account by entering valid credentials. |
| **Trigger:** | User chooses to log in. |
| **Preconditions:** | User has a registered account. |
| **Postconditions:** | User is logged into their account and redirected to the home page. |
| **Normal Flow:** | 1. User navigates to the login page. 2. User enters email and password. 3. User submits the login form. 4. System validates the credentials. 5. System logs the user in and redirects to the home page. |
| **Alternative Flows:** | If credentials are incorrect, the system prompts the user to retry. |
| **Exceptions:** | Account locked due to multiple failed login attempts. |
| **Business Rules** | Password must be hashed and securely stored. |
| **Assumptions:** | Users will remember their login credentials or use the "Forgot Details" feature if needed. |

## Use Case: Forget Details

|  |  |
| --- | --- |
| **Use Case ID:** | UC-003 |
| **Use Case Name:** | Forget Details |
| **Actors:** | Registered User |
| **Description:** | Enables users to recover or reset their forgotten login details. |
| **Trigger:** | User chooses the "Forget Details" option. |
| **Preconditions:** | User is unable to remember their login credentials. |
| **Postconditions:** | User receives instructions to recover or reset their details. |
| **Normal Flow:** | 1. User clicks on the "Forget Details" link. 2. User enters their registered email. 3. System sends a recovery email with instructions. 4. User follows the instructions to reset their password. |
| **Alternative Flows:** | If the email is not registered, the system prompts the user to check their email or register a new account. |
| **Exceptions:** | Email delivery failure. |
| **Business Rules** | Recovery email must contain a secure link with a limited validity period. |
| **Assumptions:** | User has access to their registered email. |

## Use Case: View Furniture

|  |  |
| --- | --- |
| **Use Case ID:** | UC-004 |
| **Use Case Name:** | View Furniture |
| **Actors:** | Visitor, Registered User |
| **Description:** | Allows users to browse and view different furniture items available in the store. |
| **Trigger:** | User navigates to the furniture listing page. |
| **Preconditions:** | System has furniture items listed in the database. |
| **Postconditions:** | User can see a list of furniture items with basic details. |
| **Normal Flow:** | 1. User navigates to the furniture page. 2. System displays a list of furniture items with images, names, and prices. 3. User selects an item to view detailed information. |
| **Alternative Flows:** | If no items are available, the system displays an appropriate message. |
| **Exceptions:** | Database connection failure. |
| **Business Rules** | Items must be displayed in a user-friendly manner, sorted by relevance or preference. |
| **Assumptions:** | Users will explore various furniture items before making a purchase decision. |

## Use Case: View AR Preview of Furniture

|  |  |
| --- | --- |
| **Use Case ID:** | UC-005 |
| **Use Case Name:** | View AR Preview of Furniture |
| **Actors:** | Registered User, Visitor |
| **Description:** | Enables users to visualize how the furniture would look in their physical space using AR technology. |
| **Trigger:** | User selects the AR preview option for a furniture item. |
| **Preconditions:** | User has a compatible mobile device with a camera. |
| **Postconditions:** | User can see a virtual representation of the furniture item in their physical environment. |
| **Normal Flow:** | 1. User navigates to a furniture item's details page. 2. User selects the AR preview option. 3. System accesses the mobile device's camera. 4. System overlays the 3D model of the furniture item onto the live camera feed. |
| **Alternative Flows:** | If the device is not compatible, the system displays an appropriate message. |
| **Exceptions:** | AR functionality fails due to software or hardware issues. |
| **Business Rules** | AR preview must be accurate and responsive to user movements. |
| **Assumptions:** | Users will use the AR feature to better visualize the furniture in their space. |

## Use Case: View Similar Furniture

|  |  |
| --- | --- |
| **Use Case ID:** | UC-006 |
| **Use Case Name:** | View Similar Furniture |
| **Actors:** | Visitor, Registered User |
| **Description:** | Shows products of same category like the one user is currently viewing. |
| **Trigger:** | User is viewing a specific furniture item. |
| **Preconditions:** | System has related products in the database. |
| **Postconditions:** | User can see a list of similar categories of furniture items. |
| **Normal Flow:** | 1. User views the details of a furniture item. 2. System displays a list of similar items below the main product details. |
| **Alternative Flows:** | If no similar furniture items are found, the system displays a relevant message or alternative suggestions. |
| **Exceptions:** | Database query failure. |
| **Business Rules** | Similar furniture items should be based on category, style, and price range. |
| **Assumptions:** | Users will be interested in exploring similar products for comparison. |

## Use Case: Augment Multiple Furniture Items from List

|  |  |
| --- | --- |
| **Use Case ID:** | UC-007 |
| **Use Case Name:** | Augment Multiple Furniture Items from List |
| **Actors:** | Registered User, Visitor |
| **Description:** | Enables users to visualize multiple furniture items simultaneously within their physical space. |
| **Trigger:** | User selects multiple furniture items to view in AR mode. |
| **Preconditions:** | User has initiated AR preview mode and has a compatible mobile device with a camera. |
| **Postconditions:** | User can see multiple virtual furniture items in their physical environment. |
| **Normal Flow:** | 1. User navigates to the furniture listing. 2. User selects multiple items for AR preview. 3. System accesses the mobile device's camera. 4. System shows similar category furniture items list in AR preview mode. 5. System overlays the 3D models of the selected furniture items onto the live camera feed. |
| **Alternative Flows:** | If the device is not compatible, the system displays an appropriate message. |
| **Exceptions:** | AR functionality fails due to software or hardware issues. |
| **Business Rules** | Multiple items must be displayed accurately and maintain spatial relationships. |
| **Assumptions:** | Users will use this feature to visualize how multiple pieces of furniture work together in their space. |

## Use Case: View Customer Reviews

|  |  |
| --- | --- |
| **Use Case ID:** | UC-008 |
| **Use Case Name:** | View Customer Reviews |
| **Actors:** | Visitor, Registered User, Admin |
| **Description:** | Displays reviews and ratings left by other customers for a particular furniture item. |
| **Trigger:** | User is viewing a furniture item’s details page. |
| **Preconditions:** | System has reviews stored for the furniture item. |
| **Postconditions:** | User can read and assess customer reviews. |
| **Normal Flow:** | 1. User navigates to a furniture item's details page. 2. System displays a list of reviews and ratings for the item. |
| **Alternative Flows:** | If no reviews are available, the system displays a message indicating there are no reviews yet. |
| **Exceptions:** | Database query failure. |
| **Business Rules** | Reviews must be displayed in chronological order with the most recent first. |
| **Assumptions:** | Users will read reviews to inform their purchasing decisions. |

## Use Case: View 3D Model

|  |  |
| --- | --- |
| **Use Case ID:** | UC-009 |
| **Use Case Name:** | View 3D Model |
| **Actors:** | Visitor, Registered User |
| **Description:** | Allows users to interact with a 3D model of the furniture item, including rotation and zoom functions. |
| **Trigger:** | User selects the 3D view option for a furniture item. |
| **Preconditions:** | System has a 3D model available for the furniture item. |
| **Postconditions:** | User can view and interact with the 3D model. |
| **Normal Flow:** | 1. User navigates to a furniture item's details page. 2. User selects the 3D view option. 3. System loads the 3D model. 4. User interacts with the model using rotation and zoom controls. |
| **Alternative Flows:** | If the 3D model fails to load, the system displays an error message and provides the option to view standard images. |
| **Exceptions:** | System failure in loading or rendering the 3D model. |
| **Business Rules** | 3D models must be accurate and high-quality representations of the actual products. |
| **Assumptions:** | Users will use the 3D model to get a better understanding of the product. |

## Use Case: Search By Filters

|  |  |
| --- | --- |
| **Use Case ID:** | UC-010 |
| **Use Case Name:** | Search By Filters |
| **Actors:** | Visitor, Registered User, Admin |
| **Description:** | Enables users to filter and search for furniture items based on various criteria like category, price, etc. |
| **Trigger:** | User accesses the search functionality. |
| **Preconditions:** | System has a diverse range of furniture items listed. |
| **Postconditions:** | User receives a refined list of furniture items matching the selected filters. |
| **Normal Flow:** | 1. User navigates to the search or filter section. 2. User selects desired filters (e.g., category, price range). 3. User submits the filter options. 4. System displays a list of items that match the criteria. |
| **Alternative Flows:** | If no items match the filters, the system displays a message and suggests clearing filters or adjusting criteria. |
| **Exceptions:** | System failure in applying filters or displaying results. |
| **Business Rules** | Filters must be comprehensive and user-friendly. |
| **Assumptions:** | Users will utilize filters to narrow down their search and find suitable furniture items. |

## Use Case: View Listing Details

|  |  |
| --- | --- |
| **Use Case ID:** | UC-011 |
| **Use Case Name:** | View Listing Details |
| **Actors:** | Visitor, Registered User, Admin |
| **Description:** | Provides detailed information about a specific furniture item, including dimensions, materials, and other specifications. |
| **Trigger:** | User selects a furniture item from the list. |
| **Preconditions:** | System has detailed information about the furniture item. |
| **Postconditions:** | User can see detailed information about the selected furniture item. |
| **Normal Flow:** | 1. User selects a furniture item. 2. System displays detailed information about the item, including dimensions, materials, and specifications. |
| **Alternative Flows:** | If detailed information is not available, the system displays a basic overview with a note about missing details. |
| **Exceptions:** | System failure in retrieving or displaying item details. |
| **Business Rules** | Information must be accurate and comprehensive to aid the user in making an informed decision. |
| **Assumptions:** | Users will review detailed listings to understand the product better before purchasing. |

## Use Case: Write Review

|  |  |
| --- | --- |
| **Use Case ID:** | UC-012 |
| **Use Case Name:** | Write Review |
| **Actors:** | Registered User |
| **Description:** | A registered user writes a review for a purchased furniture item. |
| **Trigger:** | The user is logged in and has purchased the furniture item. |
| **Preconditions:** | The user is logged in and has purchased the furniture item. |
| **Postconditions:** | The user has submitted a review for the furniture item. |
| **Normal Flow:** | 1. The user selects the option to write a review for a purchased furniture item. 2. The application displays a review form. 3. The user enters their review and rating. 4. The user submits the review. 5. The system saves the review and displays it in the customer reviews section. |
| **Alternative Flows:** | If the user cancels the review process, the application returns to the previous screen. |
| **Exceptions:** | System failure |
| **Business Rules** | Only users who have purchased the furniture item can write a review. |
| **Assumptions:** | The user has purchased the furniture item and is logged in. |

## Use Case: Add to Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-13 |
| **Use Case Name:** | Add to Cart |
| **Actors:** | Registered User |
| **Description:** | A registered user adds a selected furniture item to their shopping cart. |
| **Trigger:** | The user selects the option to add a furniture item to the cart. |
| **Preconditions:** | The user is logged in and viewing a furniture item. |
| **Postconditions:** | The selected furniture item is added to the user's shopping cart. |
| **Normal Flow:** | 1. The user selects the option to add a furniture item to the cart. 2. The system confirms the addition of the item to the cart. 3. The system updates the cart with the selected item. |
| **Alternative Flows:** | If the user cancels the process, the system returns to the previous screen. |
| **Exceptions:** | System failure |
| **Business Rules** | Only logged-in users can add items to their cart. |
| **Assumptions:** | The user is logged in and viewing a furniture item. |

## Use Case: Proceed to Checkout

|  |  |
| --- | --- |
| **Use Case ID:** | UC-14 |
| **Use Case Name:** | Proceed to Checkout |
| **Actors:** | Registered User |
| **Description:** | A registered user proceeds to checkout to purchase the items in their cart. |
| **Trigger:** | The user selects the option to proceed to checkout. |
| **Preconditions:** | The user is logged in and has items in their cart. |
| **Postconditions:** | The user has initiated the checkout process. |
| **Normal Flow:** | 1. The user selects the option to proceed to checkout. 2. The system displays the checkout form with shipping and payment options. 3. The user enters the required information and submits the form. |
| **Alternative Flows:** | If the user cancels the checkout process, the system returns to the cart view. |
| **Exceptions:** | System failure. |
| **Business Rules** | Only logged-in users can proceed to checkout. |
| **Assumptions:** | The user is logged in and has items in their cart. |

## Use Case: View Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-15 |
| **Use Case Name:** | View Cart |
| **Actors:** | Registered User |
| **Description:** | A registered user views the items in their shopping cart. |
| **Trigger:** | The user selects the option to view their cart. |
| **Preconditions:** | The user is logged in and has added items to the cart. |
| **Postconditions:** | The user has viewed the items in their cart. |
| **Normal Flow:** | 1. The user selects the option to view their cart. 2. The system displays the items in the user's cart. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can view their cart. |
| **Assumptions:** | The user is logged in and has added items to the cart. |

## Use Case: Update From Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-016 |
| **Use Case Name:** | Update From Cart |
| **Actors:** | Registered User |
| **Description:** | A registered user updates the quantity or details of items in their shopping cart. |
| **Trigger:** | The user selects the option to update their cart. |
| **Preconditions:** | The user is logged in and has items in its cart. |
| **Postconditions:** | The user's cart is updated with the new item quantities or details. |
| **Normal Flow:** | 1. The user selects the option to update their cart. 2. The system displays the current items in the cart with editable fields for quantity and other details. 3. The user updates the desired fields and submits the changes. 4. The system updates the cart with the new information. |
| **Alternative Flows:** | If the user cancels the update process, the system returns to the cart view without making changes. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can update their cart. |
| **Assumptions:** | The user is logged in and viewing their cart. |

## Use Case: Delete From Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-017 |
| **Use Case Name:** | Delete From Cart |
| **Actors:** | Registered User |
| **Description:** | A registered user deletes items from their shopping cart. |
| **Trigger:** | The user selects the option to delete items from their cart. |
| **Preconditions:** | The user is logged in and has items in its cart. |
| **Postconditions:** | The selected items are removed from the user's cart. |
| **Normal Flow:** | 1. The user selects the option to delete items from their cart. 2. The system displays the current items in the cart with delete options. 3. The user selects the items to delete and confirms the deletion. 4. The system removes the selected items from the cart. |
| **Alternative Flows:** | If the user cancels the deletion process, the system returns to the cart view without making changes. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can delete items from their cart. |
| **Assumptions:** | The user is logged in and viewing their cart. |

## Use Case: View Orders

|  |  |
| --- | --- |
| **Use Case ID:** | UC-018 |
| **Use Case Name:** | View Orders |
| **Actors:** | Registered User |
| **Description:** | A registered user views the list of their past and current orders. |
| **Trigger:** | The user selects the option to view their orders. |
| **Preconditions:** | The user is logged in. |
| **Postconditions:** | The user has viewed their order history. |
| **Normal Flow:** | 1. The user selects the option to view their orders. 2. The system displays a list of the user's past and current orders. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can view their order history. |
| **Assumptions:** | The user has placed orders in the past. |

## Use Case: View Order Status

|  |  |
| --- | --- |
| **Use Case ID:** | UC-019 |
| **Use Case Name:** | View Order Status |
| **Actors:** | Registered User |
| **Description:** | A registered user views the status of their current orders. |
| **Trigger:** | The user selects the option to view the status of their orders. |
| **Preconditions:** | The user is logged in and has placed orders. |
| **Postconditions:** | The user has viewed the status of their current orders. |
| **Normal Flow:** | 1. The user selects the option to view the status of their orders. 2. The system displays the status of the user's current orders. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can view their order status. |
| **Assumptions:** | The user has placed current orders. |

## Use Case: View Orders History

|  |  |
| --- | --- |
| **Use Case ID:** | UC-020 |
| **Use Case Name:** | View Orders History |
| **Actors:** | Registered User |
| **Description:** | A registered user views the history of their past orders, including details and status of each order. |
| **Trigger:** | The user selects the option to view their order history. |
| **Preconditions:** | The user is logged in. |
| **Postconditions:** | The user has viewed the details of their past orders. |
| **Normal Flow:** | 1. The user selects the option to view their order history. 2. The system retrieves the list of past orders for the user. 3. The system displays the order history, including order details and status. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Business Rules** | Only logged-in users can view their order history. |
| **Assumptions:** | The user has placed orders in the past and is logged in. |

## Use Case: Enter Payment Information

|  |  |
| --- | --- |
| **Use Case ID:** | UC-021 |
| **Use Case Name:** | Enter Payment Information |
| **Actors:** | Registered User |
| **Description:** | A registered user enters their payment information to complete the purchase of items in their cart. |
| **Trigger:** | The user proceeds to checkout and is prompted to enter payment information. |
| **Preconditions:** | The payment information is entered and validated. |
| **Postconditions:** | The user has viewed the details of their past orders. |
| **Normal Flow:** | 1. The user proceeds to checkout. 2. The system prompts the user to enter payment information. 3. The user enters the required payment details. 4. The system validates the payment information. |
| **Alternative Flows:** | If the user cancels the process, the system returns to the cart without saving the payment information.  If the payment information is invalid, the system displays an error message and prompts the user to re-enter the details. |
| **Exceptions:** | None. |
| **Business Rules** | Payment information must be valid and conform to standard formats. |
| **Assumptions:** | The user has valid payment information ready to enter. |

## Use Case: Select Shipping Method

|  |  |
| --- | --- |
| **Use Case ID:** | UC-022 |
| **Use Case Name:** | Select Shipping Method |
| **Actors:** | Registered User |
| **Description:** | A registered user selects a shipping method for the delivery of their purchased items. |
| **Trigger:** | The user proceeds to the shipping method selection step during checkout. |
| **Preconditions:** | The user is logged in, has items in their cart, and has entered valid payment information. |
| **Postconditions:** | The shipping method is selected and saved. |
| **Normal Flow:** | 1. The user proceeds to the shipping method selection step during checkout. 2. The system displays available shipping options. 3. The user selects their preferred shipping method. 4. The system saves the selected shipping method. |
| **Alternative Flows:** | If the user cancels the process, the system returns to the cart without saving the shipping method. |
| **Exceptions:** | None. |
| **Business Rules** | Only available shipping methods are presented to the user. |
| **Assumptions:** | The user has valid shipping information and options available. |

## Use Case: Place Order

|  |  |
| --- | --- |
| **Use Case ID:** | UC-023 |
| **Use Case Name:** | Place Order |
| **Actors:** | Registered User |
| **Description:** | A registered user places an order for items in their cart, completing the purchase process. |
| **Trigger:** | The user confirms the purchase at the end of the checkout process. |
| **Preconditions:** | The user is logged in, has items in their cart, and has entered valid payment and shipping information. |
| **Postconditions:** | The order is placed and confirmed. |
| **Normal Flow:** | 1. The user reviews the order summary and confirms the purchase. 2. The system processes the payment. 3. The system confirms the order and generates an order number. 4. The system sends an order confirmation to the user. |
| **Alternative Flows:** | If the payment fails, the system displays an error message and prompts the user to retry or enter new payment information.  If the user cancels the process, the system returns to the cart without placing the order. |
| **Exceptions:** | None. |
| **Business Rules** | The order can only be placed if all required information is valid and complete. |
| **Assumptions:** | The user has valid payment and shipping information, and the system is able to process the order. |

## Use Case: Add Furniture Listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-024 |
| **Use Case Name:** | Add Furniture Listings |
| **Actors:** | Admin |
| **Description:** | The admin adds new furniture listings to the system, making them available for users to view and purchase. |
| **Trigger:** | The admin chooses to add a new furniture listing through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The new furniture listing is saved and displayed in the inventory and available for users. |
| **Normal Flow:** | 1. The admin navigates to the add furniture listings page. 2. The admin enters the necessary details (name, description, price, etc.). 3. The system validates the entered details. 4. The system saves the new furniture listing. 5. The system confirms the addition of the new listing to the admin. |
| **Alternative Flows:** | If validation fails, the system prompts the admin to correct the errors |
| **Exceptions:** | None |
| **Business Rules** | The listing must include all mandatory fields. |
| **Assumptions:** | The admin has accurate and complete information to add the listing. |

## Use Case: Update Furniture Listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-025 |
| **Use Case Name:** | Update Furniture Listings |
| **Actors:** | Admin |
| **Description:** | The admin updates details of existing furniture listings. |
| **Trigger:** | The admin selects a furniture listing to update through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The updated furniture listing is saved and displayed in the inventory. |
| **Normal Flow:** | 1. The admin navigates to the list of furniture listings. 2. The admin selects a listing to update. 3. The admin modifies the necessary details. 4. The system validates the updated details. 5. The system saves the updated listing. 6. The system confirms the update to the admin. |
| **Alternative Flows:** | If validation fails, the system prompts the admin to correct the errors |
| **Exceptions:** | None |
| **Business Rules** | The listing must include all mandatory fields. |
| **Assumptions:** | The admin has accurate and complete information to update the listing. |

## Use Case: Delete Furniture Listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-026 |
| **Use Case Name:** | Delete Furniture Listings |
| **Actors:** | Admin |
| **Description:** | The admin removes furniture listings from the system. |
| **Trigger:** | The admin chooses to delete a furniture listing through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The furniture listing is removed from the inventory. |
| **Normal Flow:** | 1. The admin navigates to the list of furniture listings. 2. The admin selects a listing to delete. 3. The system prompts for confirmation. 4. The admin confirms the deletion. 5. The system removes the listing. 6. The system confirms the deletion to the admin. |
| **Alternative Flows:** | If the admin cancels the confirmation, the listing is not deleted. |
| **Exceptions:** | None |
| **Business Rules** | The listing can only be deleted if it is not associated with any active orders. |
| **Assumptions:** | The admin has the authority to delete listings. |

## Use Case: Add 3D Models

|  |  |
| --- | --- |
| **Use Case ID:** | UC-027 |
| **Use Case Name:** | Add 3D Models |
| **Actors:** | Admin |
| **Description:** | The admin adds 3D models to furniture listings. |
| **Trigger:** | The admin chooses to add a 3D model through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The 3D model is saved and associated with the selected furniture listing. |
| **Normal Flow:** | 1. The admin navigates to the 3D models section. 2. The admin selects a furniture listing to associate the 3D model with. 3. The admin uploads the 3D model. 4. The system validates the 3D model file. 5. The system saves the 3D model. 6. The system confirms the addition of the 3D model to the admin |
| **Alternative Flows:** | If validation fails, the system prompts the admin to correct the errors. |
| **Exceptions:** | None |
| **Business Rules** | The 3D model must meet specified format and size requirements. |
| **Assumptions:** | The admin has the necessary 3D model file available for upload. |

## Use Case: Update 3D Models

|  |  |
| --- | --- |
| **Use Case ID:** | UC-028 |
| **Use Case Name:** | Update 3D Models |
| **Actors:** | Admin |
| **Description:** | The admin updates 3D models associated with furniture listings. |
| **Trigger:** | The admin selects a 3D model to update through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The updated 3D model is saved and associated with the selected furniture listing. |
| **Normal Flow:** | 1. The admin navigates to the 3D models section. 2. The admin selects a 3D model to update. 3. The admin uploads the updated 3D model. 4. The system validates the 3D model file. 5. The system saves the updated 3D model. 6. The system confirms the update of the 3D model to the admin. |
| **Alternative Flows:** | If validation fails, the system prompts the admin to correct the errors. |
| **Exceptions:** | None |
| **Business Rules** | The 3D model must meet specified format and size requirements. |
| **Assumptions:** | The admin has the necessary 3D model file available for upload. |

## Use Case: Delete 3D Models

|  |  |
| --- | --- |
| **Use Case ID:** | UC-029 |
| **Use Case Name:** | Delete 3D Models |
| **Actors:** | Admin |
| **Description:** | The admin removes 3D models from furniture listings. |
| **Trigger:** | The admin chooses to delete a 3D model through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The 3D model is removed from the selected furniture listing. |
| **Normal Flow:** | 1. The admin selects a 3D model to delete. 2. The system prompts for confirmation. 3. The admin confirms the deletion. 4. The system removes the 3D model. 5. The system confirms the deletion to the admin. |
| **Alternative Flows:** | If the admin cancels the confirmation, the 3D model is not deleted. |
| **Exceptions:** | None |
| **Business Rules** | The 3D model can only be deleted if it is not associated with any active listings. |
| **Assumptions:** | The admin has the authority to delete 3D models. |

## Use Case: Generate Inventory Reports

|  |  |
| --- | --- |
| **Use Case ID:** | UC-030 |
| **Use Case Name:** | Generate Inventory Reports |
| **Actors:** | Admin |
| **Description:** | The admin generates reports on the inventory, including stock levels, stock movements, and reorder levels. |
| **Trigger:** | The admin chooses to generate an inventory report through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The inventory report is generated and available for viewing or download. |
| **Normal Flow:** | 1. The admin navigates to the reports section. 2. The admin selects the type of inventory report to generate. 3. The system compiles the relevant data. 4. The system generates the report. 5. The system presents the report to the admin. |
| **Alternative Flows:** | None. |
| **Exceptions:** | If there is insufficient data to generate the report, the system notifies the admin. |
| **Business Rules** | The report can only include data the admin has permission to view. |
| **Assumptions:** | The system has accurate and up-to-date inventory data. |

## Use Case: Generate Inventory Reports

|  |  |
| --- | --- |
| **Use Case ID:** | UC-030 |
| **Use Case Name:** | Generate Inventory Reports |
| **Actors:** | Admin |
| **Description:** | The admin generates reports on the inventory, including stock levels, stock movements, and reorder levels. |
| **Trigger:** | The admin chooses to generate an inventory report through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The inventory report is generated and available for viewing or download. |
| **Normal Flow:** | 1. The admin navigates to the reports section. 2. The admin selects the type of inventory report to generate. 3. The system compiles the relevant data. 4. The system generates the report. 5. The system presents the report to the admin. |
| **Alternative Flows:** | None. |
| **Exceptions:** | If there is insufficient data to generate the report, the system notifies the admin. |
| **Business Rules** | The report can only include data the admin has permission to view. |
| **Assumptions:** | The system has accurate and up-to-date inventory data. |

## Use Case: Update Order Status

|  |  |
| --- | --- |
| **Use Case ID:** | UC-031 |
| **Use Case Name:** | Update Order Status |
| **Actors:** | Admin |
| **Description:** | This use case allows the admin to update the status of a customer's order in the system. |
| **Trigger:** | The admin decides to update the status of an order based on its current state in the fulfillment process. |
| **Preconditions:** | The admin is logged into the system.  There are existing orders in the system. |
| **Postconditions:** | The order status is updated in the system.  The customer is notified of the status change. |
| **Normal Flow:** | 1. The admin selects the order management option from the admin dashboard. 2. The admin views the list of existing orders. 3. The admin selects a specific order to update. 4. The admin changes the order status to the appropriate state (e.g., 'Processing', 'Shipped', 'Delivered', 'Cancelled'). 5. The admin confirms the status update. 6. The system saves the updated status and sends a notification to the customer. |
| **Alternative Flows:** | If the selected order does not exist, an error message is displayed.  If the status update fails due to a system error, the admin is prompted to try again later. |
| **Exceptions:** | System error occurs during the status update process. |
| **Business Rules** | Only authorized Admin users can update order statuses.  The system must log all status changes for audit purposes. |
| **Assumptions:** | The admin has the necessary permissions to update order statuses.  The system is operational and accessible when the admin attempts to update an order status. |

## Use Case: View Furniture Listings on Admin Panel

|  |  |
| --- | --- |
| **Use Case ID:** | UC-032 |
| **Use Case Name:** | View Furniture Listings on Admin Panel |
| **Actors:** | Admin |
| **Description:** | This use case allows an admin to view the available furniture listings in the store for administrative purposes. |
| **Trigger:** | The admin decides to review the furniture listings for inventory management or other administrative tasks. |
| **Preconditions:** | The system contains a catalog of furniture item. |
| **Postconditions:** | The admin can access detailed information about furniture items for administrative purposes. |
| **Normal Flow:** | 1. The admin navigates to the furniture listings section. 2. The system retrieves the list of furniture listings. 3. The system displays the list to the admin. |
| **Alternative Flows:** | If no furniture items are available, a message indicating no items are available is displayed. |
| **Exceptions:** | System error occurs while retrieving the furniture listings. |
| **Business Rules** | The admin has full access to view all furniture listings and associated details. The system must update the furniture listings in real-time to reflect current availability. |
| **Assumptions:** | The system is operational and can retrieve furniture listings without delays. |

## Use Case: Shows Sales & Stock Dashboard

|  |  |
| --- | --- |
| **Use Case ID:** | UC-033 |
| **Use Case Name:** | Shows Sales & Stock Dashboard |
| **Actors:** | Admin |
| **Description:** | The admin views a dashboard that displays sales performance and stock levels. |
| **Trigger:** | The admin chooses to view the sales and stock dashboard through the admin panel. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | The sales and stock dashboard is displayed. |
| **Normal Flow:** | 1. The admin navigates to the dashboard section. 2. The system retrieves sales and stock data. 3. The system compiles the data into a dashboard format. 4. The system displays the dashboard to the admin. |
| **Alternative Flows:** | None. |
| **Exceptions:** | If there is insufficient data, the system notifies the admin. |
| **Business Rules** | The dashboard can only include data the admin has permission to view. |
| **Assumptions:** | The system has accurate and up-to-date sales and stock data. |

## Use Case: Secures Sales and Updates Stocks

|  |  |
| --- | --- |
| **Use Case ID:** | UC-034 |
| **Use Case Name:** | Secures Sales and Updates Stocks |
| **Actors:** | Admin |
| **Description:** | The admin secures sales transactions and updates stock levels accordingly. |
| **Trigger:** | Sales transactions occur and stock levels need updating. |
| **Preconditions:** | The admin is logged in and has the necessary permissions. |
| **Postconditions:** | Sales transactions are secured and stock levels are updated. |
| **Normal Flow:** | 1. Sales transactions are recorded in the system. 2. The system updates stock levels based on the transactions. 3. The system secures the sales data. 4. The system confirms the stock update to the admin. |
| **Alternative Flows:** | If stock levels cannot be updated, the system notifies the admin. |
| **Exceptions:** | None. |
| **Business Rules** | Stock levels must be accurately updated based on sales. |
| **Assumptions:** | Sales transactions are accurately recorded. |

## Use Case: Logout

|  |  |
| --- | --- |
| **Use Case ID:** | UC-035 |
| **Use Case Name:** | Logout |
| **Actors:** | Registered User, Admin |
| **Description:** | A registered user and admin log out of their account. |
| **Trigger:** | The user selects the option to log out. |
| **Preconditions:** | The user is logged in. |
| **Postconditions:** | The user is logged out, and their session is terminated. |
| **Normal Flow:** | 1. The user selects the option to log out. 2. The system terminates the user's session and returns to the home page. |
| **Alternative Flows:** | None. |
| **Exceptions:** | None. |
| **Business Rules** | The system must securely terminate the user's session. |
| **Assumptions:** | The user is logged in. |

## Functional Requirements

## Functional Requirement for UC-1: Register

|  |  |
| --- | --- |
| **Identifier** | FR-01 |
| **Title** | Register |
| **Requirement** | The system shall allow visitors to create a new account by providing necessary details such as name, email, and password. |
| **Source** | Visitor |
| **Rationale** | To enable visitors to become registered users and access additional functionalities. |
| **Business Rule (if required)** | Password must meet security criteria (e.g., minimum length, contains letters and numbers). |
| **Dependencies** | None |

## Functional Requirement for UC-2: Login

|  |  |
| --- | --- |
| **Identifier** | FR-02 |
| **Title** | Login |
| **Requirement** | The system shall allow registered users to log in using their email and password. |
| **Source** | Registered User |
| **Rationale** | To provide access to personalized features and functionalities. |
| **Business Rule (if required)** | Valid email and password combination required. |
| **Dependencies** | User must be registered. |

## Functional Requirement for UC-3: Forgot Details

|  |  |
| --- | --- |
| **Identifier** | FR-03 |
| **Title** | Forgot Details |
| **Requirement** | The system shall allow users to recover their account details if they forget them. |
| **Source** | Registered User |
| **Rationale** | To ensure users can regain access to their account. |
| **Business Rule (if required)** | Verification through email or security questions. |
| **Dependencies** | User must have a registered account. |

## Functional Requirement for UC-4: View Furniture

|  |  |
| --- | --- |
| **Identifier** | FR-04 |
| **Title** | View Furniture |
| **Requirement** | The system shall allow visitors and registered users to browse available furniture. |
| **Source** | Visitor/Registered User/Admin |
| **Rationale** | To provide users with access to the product catalog. |
| **Business Rule (if required)** | None |
| **Dependencies** | None |

## Functional Requirement for UC-5: View AR Preview of Furniture

|  |  |
| --- | --- |
| **Identifier** | FR-05 |
| **Title** | View AR Preview of Furniture |
| **Requirement** | The system shall allow users to view an augmented reality preview of furniture. |
| **Source** | Visitor/Registered User |
| **Rationale** | To help users visualize furniture in their space. |
| **Business Rule (if required)** | Requires device with AR capabilities. |
| **Dependencies** | None |

## Functional Requirement for UC-6: View Similar Furniture

|  |  |
| --- | --- |
| **Identifier** | FR-06 |
| **Title** | View Similar Furniture |
| **Requirement** | The system shall suggest similar furniture based on the user's selection. |
| **Source** | Registered User |
| **Rationale** | To assist users in finding alternatives. |
| **Business Rule (if required)** | Suggestions based on category and attributes. |
| **Dependencies** | User must be viewing a furniture item. |

## Functional Requirement for UC-7: View Customer Reviews

|  |  |
| --- | --- |
| **Identifier** | FR-07 |
| **Title** | View Customer Reviews |
| **Requirement** | The system shall allow users to view reviews from other customers. |
| **Source** | Registered User, Admin, visitor |
| **Rationale** | To help users make informed decisions. |
| **Business Rule (if required)** | Display all reviews and ratings. |
| **Dependencies** | None. |

## Functional Requirement for UC-8: View 3D Model

|  |  |
| --- | --- |
| **Identifier** | FR-08 |
| **Title** | View 3D Model |
| **Requirement** | The system shall allow users to view 3D models of the furniture. |
| **Source** | Registered User |
| **Rationale** | To provide a detailed visual representation. |
| **Business Rule (if required)** | Requires device with 3D model viewing capabilities. |
| **Dependencies** | User Registration and Authentication |

## Functional Requirement for UC-9: Search by Filters

|  |  |
| --- | --- |
| **Identifier** | FR-9 |
| **Title** | Search by Filters |
| **Requirement** | The system shall allow users to filter furniture based on various criteria (e.g., price, category). |
| **Source** | Admin/User/Visitor |
| **Rationale** | To help users find furniture that meets their preferences. |
| **Business Rule (if required)** | Filter options based on product attributes. |
| **Dependencies** | None. |

## Functional Requirement for UC-10: View Listing Details

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | View Listing Details |
| **Requirement** | The system shall display detailed information about a furniture listing. |
| **Source** | Admin/User/Visitor |
| **Rationale** | To provide users with all necessary information about the product. |
| **Business Rule (if required)** | Must include price, description, dimensions, and available stock. |
| **Dependencies** | None. |

## Functional Requirement for UC-11: Augment Multiple Furniture Items from List

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | Augment Multiple Furniture Items from List |
| **Requirement** | It enables users to visualize multiple furniture items simultaneously within their physical space. |
| **Source** | Registered User, Visitor |
| **Rationale** | To enable users to visualize how multiple pieces of furniture will look together in their space. |
| **Business Rule (if required)** | Multiple items must be displayed accurately and maintain spatial relationships. |
| **Dependencies** | User must be logged in and have a compatible device for AR. |

## Functional Requirement for UC-12: Write Review

|  |  |
| --- | --- |
| **Identifier** | FR-12 |
| **Title** | Write Review |
| **Requirement** | The system shall allow registered users to write reviews for purchased furniture. |
| **Source** | Registered User |
| **Rationale** | To enable user feedback and ratings. |
| **Business Rule (if required)** | Review linked to verified purchase. |
| **Dependencies** | User must be logged in and have purchased the item. |

## Functional Requirement for UC-13: Add to Cart

|  |  |
| --- | --- |
| **Identifier** | FR-13 |
| **Title** | Add to Cart |
| **Requirement** | The system shall allow registered users to add furniture items to their shopping cart. |
| **Source** | Registered User |
| **Rationale** | To enable users to prepare for purchase. |
| **Business Rule (if required)** | Item must be in stock. |
| **Dependencies** | User must be logged in. |

## Functional Requirement for UC-14: View Cart

|  |  |
| --- | --- |
| **Identifier** | FR-14 |
| **Title** | View Cart |
| **Requirement** | The system shall allow users to view items in their cart. |
| **Source** | Registered User |
| **Rationale** | To review selected items before purchase. |
| **Business Rule (if required)** | Display all items, quantities, and total price. |
| **Dependencies** | User must have items in the cart. |

## Functional Requirement for UC-15: Update from Cart

|  |  |
| --- | --- |
| **Identifier** | FR-15 |
| **Title** | Update From Cart |
| **Requirement** | The system shall allow users to update quantities or remove items from the cart. |
| **Source** | Registered User |
| **Rationale** | To allow users to modify their purchase selection. |
| **Business Rule (if required)** | Must update total price accordingly. |
| **Dependencies** | User must be logged in and have items in the cart. |

## Functional Requirement for UC-16: Proceed to Checkout

|  |  |
| --- | --- |
| **Identifier** | FR-16 |
| **Title** | Proceed to Checkout |
| **Requirement** | The system shall allow users to proceed to checkout with items in their cart. |
| **Source** | Registered User |
| **Rationale** | To initiate the purchase process. |
| **Business Rule (if required)** | Must have valid items in the cart. |
| **Dependencies** | User must be logged in. |

## Functional Requirement for UC-17: Enter Payment Information

|  |  |
| --- | --- |
| **Identifier** | FR-17 |
| **Title** | Enter Payment Information |
| **Requirement** | The system shall allow users to enter their payment details. |
| **Source** | Registered User |
| **Rationale** | To complete the purchase transaction. |
| **Business Rule (if required)** | Secure payment gateway required. |
| **Dependencies** | User must be proceeding to checkout. |

## Functional Requirement for UC-18: Select Shipping Method

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Select Shipping Method |
| **Requirement** | The system shall allow users to choose a shipping method. |
| **Source** | Registered User |
| **Rationale** | To provide options for delivery. |
| **Business Rule (if required)** | Display available shipping methods and costs. |
| **Dependencies** | User must be proceeding to checkout. |

## Functional Requirement for UC-19: Place Order

|  |  |
| --- | --- |
| **Identifier** | FR-19 |
| **Title** | Place Order |
| **Requirement** | The system shall allow users to place an order after entering payment and shipping information. |
| **Source** | Registered User |
| **Rationale** | To complete the purchase process. |
| **Business Rule (if required)** | Confirm order details and deduct from stock. |
| **Dependencies** | User must have valid payment and shipping information. |

## Functional Requirement for UC-20: View Orders

|  |  |
| --- | --- |
| **Identifier** | FR-20 |
| **Title** | View Orders |
| **Requirement** | The system shall allow users to view their current and past orders. |
| **Source** | Registered User |
| **Rationale** | To track purchase history and order status. |
| **Business Rule (if required)** | Display order details, status, and history. |
| **Dependencies** | User must be logged in. |

## Functional Requirement for UC-21: View Order Status

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | View Order Status |
| **Requirement** | The system shall allow users to view the status of their orders. |
| **Source** | Registered User |
| **Rationale** | To provide updates on the progress of their orders. |
| **Business Rule (if required)** | Display real-time status updates. |
| **Dependencies** | User must have orders placed. |

## Functional Requirement for UC-22: View Orders History

|  |  |
| --- | --- |
| **Identifier** | FR-22 |
| **Title** | View Orders History |
| **Requirement** | The system shall allow users to view the history of all their past orders. |
| **Source** | Registered User |
| **Rationale** | To provide a record of all previous transactions. |
| **Business Rule (if required)** | Show complete order history with details. |
| **Dependencies** | User must be logged in. |

## Functional Requirement for UC-23: Delete from Cart

|  |  |
| --- | --- |
| **Identifier** | FR-23 |
| **Title** | View Orders History |
| **Requirement** | The system shall allow users to remove items from their shopping cart. |
| **Source** | Registered User |
| **Rationale** | To enable users to manage their cart contents before checkout. |
| **Business Rule (if required)** | Item should be removed from the cart immediately upon user confirmation. |
| **Dependencies** | User must be logged in and have items in the cart. |

## Functional Requirement for UC-24: Generate Inventory Report

|  |  |
| --- | --- |
| **Identifier** | FR-24 |
| **Title** | Generate Inventory Report |
| **Requirement** | The system shall allow admins to generate reports on inventory status. |
| **Source** | Admin |
| **Rationale** | To monitor and manage stock levels. |
| **Business Rule (if required)** | Reports must include current stock levels and sales data. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-25: Secure Sales and Update Stocks

|  |  |
| --- | --- |
| **Identifier** | FR-25 |
| **Title** | Secure Sales and Update Stocks |
| **Requirement** | The system shall securely process sales transactions and update stock levels accordingly. |
| **Source** | Admin |
| **Rationale** | To ensure accurate inventory management and secure transactions. |
| **Business Rule (if required)** | Must integrate with payment gateway and inventory system. |
| **Dependencies** | Sale transaction initiated. |

## Functional Requirement for UC-26: Show Sales and Stock Dashboard

|  |  |
| --- | --- |
| **Identifier** | FR-26 |
| **Title** | Show Sales and Stock Dashboard |
| **Requirement** | The system shall provide a dashboard for admins to view sales and stock levels. |
| **Source** | Admin |
| **Rationale** | To provide a comprehensive overview of business metrics. |
| **Business Rule (if required)** | Dashboard must be updated in real-time. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-27: Add Furniture Listings

|  |  |
| --- | --- |
| **Identifier** | FR-27 |
| **Title** | Add Furniture Listings |
| **Requirement** | The system shall allow admins to add new furniture listings. |
| **Source** | Admin |
| **Rationale** | To expand the product catalog. |
| **Business Rule (if required)** | Must include all required details (e.g., name, price, description). |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-28: Update Furniture Listings

|  |  |
| --- | --- |
| **Identifier** | FR-28 |
| **Title** | Update Furniture Listings |
| **Requirement** | The system shall allow admins to update existing furniture listings. |
| **Source** | Admin |
| **Rationale** | To keep product information current. |
| **Business Rule (if required)** | Must allow updates to all listing details. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-29: Delete Furniture Listings

|  |  |
| --- | --- |
| **Identifier** | FR-29 |
| **Title** | Delete Furniture Listings |
| **Requirement** | The system shall allow admins to delete furniture listings. |
| **Source** | Admin |
| **Rationale** | To manage the product catalog by removing outdated or unavailable items. |
| **Business Rule (if required)** | Must ensure confirmation before deletion. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-30: Add 3D Models

|  |  |
| --- | --- |
| **Identifier** | FR-30 |
| **Title** | Add 3D Models |
| **Requirement** | The system shall allow admins to add 3D models for furniture listings. |
| **Source** | Admin |
| **Rationale** | To provide enhanced visualization of products. |
| **Business Rule (if required)** | Models must meet specified format requirements. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-31: Update 3D Models

|  |  |
| --- | --- |
| **Identifier** | FR-31 |
| **Title** | Update 3D Models |
| **Requirement** | The system shall allow admins to update existing 3D models for furniture listings. |
| **Source** | Admin |
| **Rationale** | To ensure accurate representation of products. |
| **Business Rule (if required)** | Models must meet specified format requirements. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-32: Delete 3D Models

|  |  |
| --- | --- |
| **Identifier** | FR-32 |
| **Title** | Delete 3D Models |
| **Requirement** | The system shall allow admins to delete 3D models for furniture listings. |
| **Source** | Admin |
| **Rationale** | To manage the 3D model repository. |
| **Business Rule (if required)** | Must ensure confirmation before deletion. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-33: Update Order Status

|  |  |
| --- | --- |
| **Identifier** | FR-33 |
| **Title** | Update Order Status |
| **Requirement** | The system shall allow admins to update the status of customer orders. |
| **Source** | Admin |
| **Rationale** | To provide accurate order tracking. |
| **Business Rule (if required)** | Status updates should reflect real-time changes. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-34: View Furniture Listings on Admin Panel

|  |  |
| --- | --- |
| **Identifier** | FR-34 |
| **Title** | View Furniture Listings on Admin Panel |
| **Requirement** | The system shall allow admins to view all furniture listings. |
| **Source** | Admin |
| **Rationale** | To manage the product catalog. |
| **Business Rule (if required)** | Display all listings with details. |
| **Dependencies** | Admin must be logged in. |

## Functional Requirement for UC-35: Logout

|  |  |
| --- | --- |
| **Identifier** | FR-35 |
| **Title** | Logout |
| **Requirement** | The system shall allow users to log out from their account. |
| **Source** | Registered User |
| **Rationale** | To secure the account when not in use. |
| **Business Rule (if required)** | None |
| **Dependencies** | User must be logged in. |

## Non-Functional Requirements

## Usability

**Usability-1: User-Friendly Interface**

* **Description:** The system must have a user-friendly interface that is easy to navigate and understand.
* **Rationale:** To ensure a positive user experience and easy adoption.
* **Verification:** User testing and feedback.

**Usability-2: Clear Instructions**

* **Description:** The system must provide clear and concise instructions for using each feature.
* **Rationale:** To facilitate user understanding and reduce the learning curve.
* **Verification:** Documentation and user feedback.

**Usability-3: Intuitiveness**

* **Description:** The system must be intuitive and allow users to easily perform tasks and access information.
* **Rationale:** To enhance user productivity and satisfaction.
* **Verification:** User feedback and usability testing.

**Usability-4: Accessibility**

* **Description:** The system must be accessible to users with disabilities, complying with WCAG 2.1 Level AA guidelines.
* **Rationale:** To ensure inclusivity and usability for all users.
* **Verification:** Accessibility testing and compliance checks.

**Usability-5: Error Handling**

* **Description:** The system must provide clear error messages with guidance for resolving issues.
* **Rationale:** To help users recover from errors quickly and easily.
* **Verification:** User testing and feedback on error recovery processes.

## Performance

**PER-1: Fast Loading and Execution**

* **Description:** The system must be optimized for fast loading and execution.
* **Rationale:** To provide a responsive user experience.
* **Verification:** Measured system response times.

**PER-2: Scalability**

* **Description:** The system must be able to handle a high volume of requests without experiencing significant delays.
* **Rationale:** To accommodate potential growth in usage.
* **Verification:** Load testing and response time measurements.

**PER-3: Large Data Handling**

* **Description:** The system must be able to handle large data sets efficiently.
* **Rationale:** To ensure efficient processing of substantial amounts of data.
* **Verification:** Performance testing with large datasets.

**PER-4: Concurrent Users**

* **Description:** The system must support at least 1000 concurrent users without performance degradation.
* **Rationale:** To ensure the website can handle peak usage times.
* **Verification:** Stress testing and load testing.

**PER-5: Transaction Processing**

* **Description:** The system must handle up to 200 transactions per minute during peak shopping times without issues.
* **Rationale:** To ensure reliable performance during high-traffic periods.
* **Verification:** Transaction load testing.

By adhering to these non-functional requirements, FurnishARt will ensure a robust, user-friendly, and high-performing AR-based online furniture store.

**References**

List any documents or other resources to which this SRS refers, if any. These might include user interface style guides, standards, system requirements specifications, interface specifications, or the SRS for a related product.