**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 to function seamlessly in a variety of environments to provide a robust and user-friendly experience. Key aspects of the operating environment include:

* **Hardware Platform:** The application is optimized to run on a range of devices, including desktops, laptops, tablets, and smartphones. The AR functionality requires devices with cameras capable of supporting augmented reality features.
* **Operating Systems:** The application is platform-agnostic, ensuring compatibility with major operating systems such as Windows, macOS, iOS, and Android. This broad compatibility allows users to access the platform from virtually any device.
* **Geographical Locations:** The software is intended for global use, catering to customers from various geographical regions. It is designed to adapt to different locales and languages to provide a personalized shopping experience.
* **Servers and Databases:** The application relies on secure, scalable server infrastructure and database systems. These components may be hosted on cloud-based platforms such as AWS, Azure, or Google Cloud, ensuring high availability and reliability. The backend is built using Node.js and Express, with PostgreSQL as the database management system.
* **Web-Based Interaction:** The user interface is web-based and accessible through modern web browsers. The application supports the following browsers:

Google Chrome (all versions)

Mozilla Firefox versions 45 and above

Microsoft Edge (all versions)

Apple Safari versions 10 and above

* **Organizations Hosting Data:** Depending on the implementation, various organizations or service providers may host the related databases, servers, and websites. These organizations are expected to adhere to stringent data security and privacy protocols to protect user information. Hosting providers must comply with relevant data protection regulations such as GDPR for European users and CCPA for users in California.
* **Network Requirements:** A stable internet connection is required to ensure smooth operation of the application, especially for loading 3D models and AR functionalities. The platform is designed to optimize performance even under 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.

**Requirement identifying technique.**

The requirement identifying technique for the AR-based Online Furniture Store project involves selecting methods tailored to the nature and scope of the application. Given the interactive nature of the end-user experience and the need to capture user interactions effectively, the following technique is chosen:

**Use Case Analysis:** Use cases are an effective technique for interactive end-user applications like the AR-based Online Furniture Store. They provide a structured approach to identifying functional requirements by focusing on user interactions and system behaviour. Use case analysis helps in defining various scenarios, user roles, and system responses, thereby facilitating a clear understanding of system functionality and user requirements.

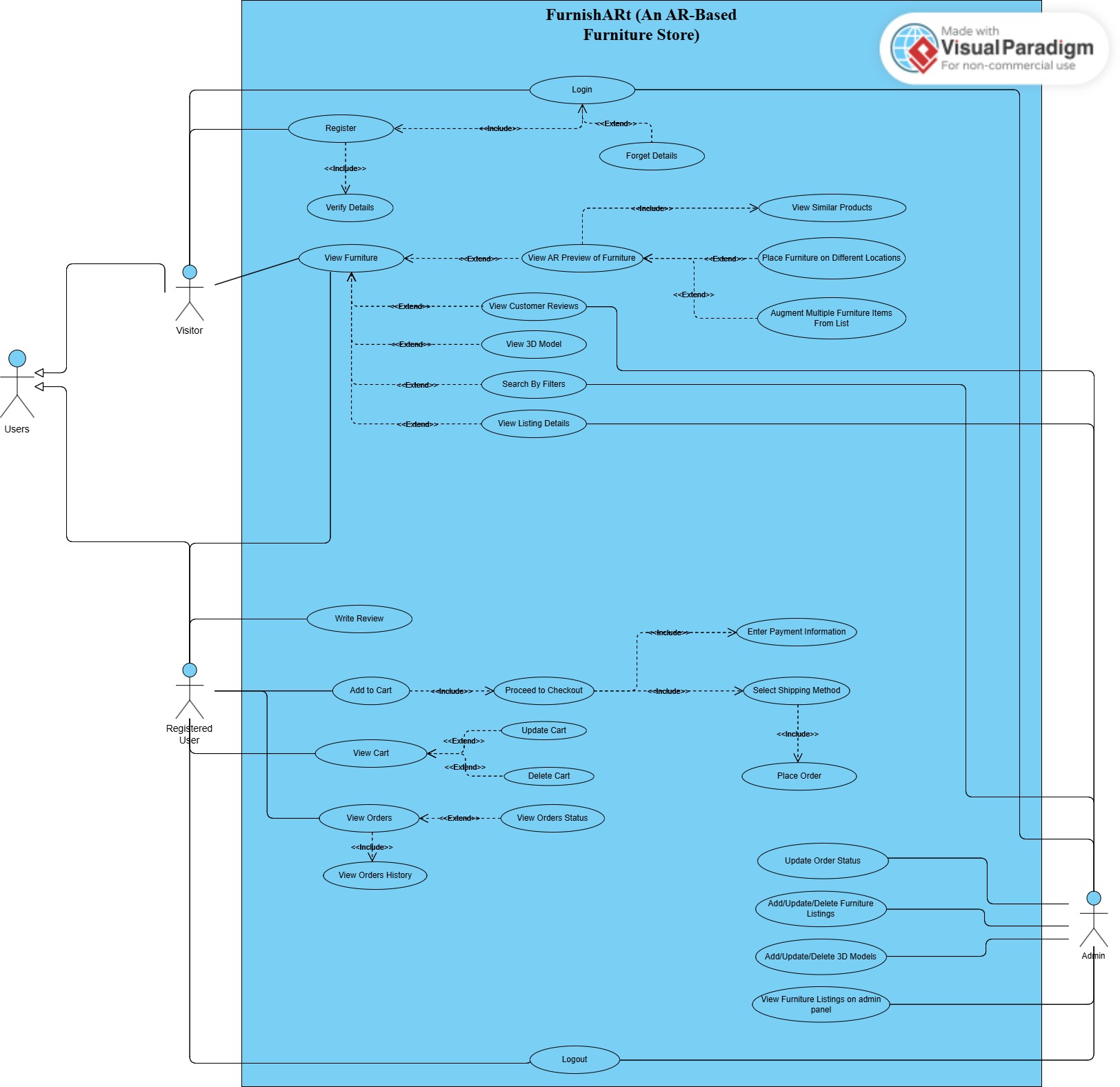


Figure 1: Use Case Diagram

|  |  |
| --- | --- |
| **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: Verify Details

|  |  |
| --- | --- |
| **Use Case ID:** | UC-002 |
| **Use Case Name:** | Verify Details |
| **Actors:** | Visitor |
| **Description:** | Ensures the provided registration details are correct and valid. |
| **Trigger:** | Visitor submits registration details. |
| **Preconditions:** | Visitor has filled out the registration form. |
| **Postconditions:** | Visitor receives feedback on the validity of the registration details. |
| **Normal Flow:** | 1. System checks if all required fields are filled. 2. System validates the email format. 3. System checks password strength. 4. System confirms details are unique (e.g., email is not already registered). |
| **Alternative Flows:** | If any detail is invalid, the system prompts the visitor to correct the errors. |
| **Exceptions:** | Unexpected server error during validation. |
| **Business Rules** | 1. Email must be unique. 2. Password must meet security criteria. |
| **Assumptions:** | Visitor will correct any invalid details before resubmitting. |

## Use Case: Login

|  |  |
| --- | --- |
| **Use Case ID:** | UC-003 |
| **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 an existing 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-004 |
| **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-005 |
| **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-006 |
| **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 Products

|  |  |
| --- | --- |
| **Use Case ID:** | UC-007 |
| **Use Case Name:** | View Similar Products |
| **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 products are found, the system displays a relevant message or alternative suggestions. |
| **Exceptions:** | Database query failure. |
| **Business Rules** | Similar products should be based on category, style, and price range. |
| **Assumptions:** | Users will be interested in exploring similar products for comparison. |

## Use Case: Place Furniture on Different Locations

|  |  |
| --- | --- |
| **Use Case ID:** | UC-008 |
| **Use Case Name:** | Place Furniture on Different Locations |
| **Actors:** | Registered User, Visitor |
| **Description:** | Allows users to place the furniture item in different locations within their environment in AR mode. |
| **Trigger:** | User selects the option to move the furniture item in AR mode. |
| **Preconditions:** | User has initiated AR preview mode. |
| **Postconditions:** | User can see how the furniture fits in different locations. |
| **Normal Flow:** | 1. User activates AR preview mode. 2. User selects the option to move the furniture item. 3. User moves the device to place the item in different locations. 4. System updates the AR display to reflect the new placement. |
| **Alternative Flows:** | If the placement is not feasible, the system provides feedback to the user. |
| **Exceptions:** | AR functionality fails due to software or hardware issues. |
| **Business Rules** | Placement must be intuitive and accurate in the AR display. |
| **Assumptions:** | Users will want to see how the furniture fits in various spots within their space. |

## Use Case: Augment Multiple Furniture Items from List

|  |  |
| --- | --- |
| **Use Case ID:** | UC-009 |
| **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-010 |
| **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-011 |
| **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-012 |
| **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-013 |
| **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-14 |
| **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-15 |
| **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-16 |
| **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-17 |
| **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 Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-18 |
| **Use Case Name:** | Update 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 Cart

|  |  |
| --- | --- |
| **Use Case ID:** | UC-19 |
| **Use Case Name:** | Delete 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-20 |
| **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-21 |
| **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-22 |
| **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-23 |
| **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-24 |
| **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-25 |
| **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/Update/Delete Furniture Listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-26 |
| **Use Case Name:** | Add/Update/Delete Furniture Listings |
| **Actors:** | Admin |
| **Description:** | An admin adds, updates, or deletes furniture listings in the system. |
| **Trigger:** | The admin selects the option to manage furniture listings. |
| **Preconditions:** | The admin is logged in. |
| **Postconditions:** | The furniture listings are updated accordingly. |
| **Normal Flow:** | 1. The admin selects the option to add, update, or delete a furniture listing. 2. The system displays the management interface for furniture listings. 3. The admin performs the desired action and submits the changes. 4. The system processes the changes and updates the listings. |
| **Alternative Flows:** | If the admin cancels the process, the system returns to the previous screen without making changes. |
| **Exceptions:** | None. |
| **Business Rules** | Only admins can manage furniture listings. |
| **Assumptions:** | The admin is logged in and has the necessary permissions. |

## Use Case: Add/Update/Delete 3D Models.

|  |  |
| --- | --- |
| **Use Case ID:** | UC-27 |
| **Use Case Name:** | Add/Update/Delete 3D Models |
| **Actors:** | Admin |
| **Description:** | An admin adds, updates, or deletes 3D models of furniture items in the system. |
| **Trigger:** | The admin selects the option to manage 3D models. |
| **Preconditions:** | The admin is logged in. |
| **Postconditions:** | The 3D models are updated accordingly. |
| **Normal Flow:** | 1. The admin selects the option to add, update, or delete a 3D model. 2. The system displays the management interface for 3D models. 3. The admin performs the desired action and submits the changes. 4. The system processes the changes and updates the 3D models. |
| **Alternative Flows:** | If the admin cancels the process, the system returns to the previous screen without making changes. |
| **Exceptions:** | None. |
| **Business Rules** | Only admins can manage 3D models. |
| **Assumptions:** | The admin is logged in and has the necessary permissions. |

## Use Case: Update Order Status

|  |  |
| --- | --- |
| **Use Case ID:** | UC-28 |
| **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

|  |  |
| --- | --- |
| **Use Case ID:** | UC-29 |
| **Use Case Name:** | View Furniture Listings on admin |
| **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 in the admin dashboard. 2. The admin views the list of available furniture items, including details such as descriptions, dimensions, prices, and current availability. The admin can filter and search through the furniture listings based on various criteria such as price, size, material, etc. |
| **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: Logout

|  |  |
| --- | --- |
| **Use Case ID:** | UC-30 |
| **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**

This section describes the functional requirements of the system expressed in natural language style. This section is typically organized by feature as system feature name and specific functional requirements associated with this feature. It is just one possible way to arrange them. Other organizational options include arranging functional requirements by use case, process flow, mode of operation, user classes Chapter 10 “Documenting the requirements”. Let consider feature scheme as an example.

**Functional Requirement X**

Itemize the specific functional requirements associated with each feature. These are the software capabilities that must be implemented for the user to carry out the feature’s services or to perform a use case. Describe how the product should respond to anticipated error conditions and to invalid inputs and actions. Uniquely label each functional requirement, as described earlier. You can create multiple attributes for each functional requirement, such as rationale, source, dependencies etc. The following template is required to write functional requirements. For further detail see Chapter 11” Writing excellent requirements”.

**Table 1 Show the functional requirement template**

|  |  |
| --- | --- |
| **Identifier** | Requirement ID |
| **Title** | Title of requirement |
| **Requirement** | Description of requirement which may be written either from user or system perspective e.g.  If written in **user perspective**  The [user class or actor name] shall be able to [do something] [to some object] [qualifying conditions, response time, or quality statement].  If written in **system perspective**  [optional precondition] [optional trigger event] the system shall [expected system response] |
| **Source** | Where this requirement is come from (who originate it) |
| **Rationale** | Motivation behind the requirement |
| **Business Rule (if required)** | Any restriction, policy, rule that the particular requirement must be fulfilled through its functional behavior |
| **Dependencies** | Requirements ID that are dependent on this requirement |
| **Priority** | High/Medium/Low |

## 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.