

Software Requirements Specification

Software Engineering

The Interior House

An Online Marketplace for Interior Design Products

- | | | |
|----|----------------|----------------|
| 1. | Vibha Puthran | (01FB16ECS441) |
| 2. | Vishal S Rao | (01FB16ECS450) |
| 3. | Vishnu V Singh | (01FB16ECS451) |
| 4. | Aditya Lokesh | (01FB16ECS461) |
| 5. | Ayushi S Mehta | (01FB16ECS465) |
| 6. | Parul Tebriwal | (01FB16ECS475) |
| 7. | Sanjana U | (01FB16ECS478) |

Document Owner

Name: Aditya Lokesh

SRN: 01FB16ECS461

Project Name: Interior Marketplace

Email: adityapesu@gmail.com

Name: Parul Tebriwal

SRN: 01FB16ECS475

Project Name: Interior Marketplace

Email: parulslg@gmail.com

Index

Document Outline

1. Introduction

- Purpose
- Product Scope

2. Overall Description

- Project Perspective
- Project Functions
- User Classes and Characteristics
- Operating Environment
- Assumptions and Dependencies

3. External Interface Requirements

- User Interfaces
- Hardware Interfaces
- Software Interfaces
- Communications Interfaces

4. System Features

- Login Service
- Homepage Recommendation
- Search Feature
- Shopping Cart
- Checkout System
- View Order Details
- The Seller Lists Products
- Customer care chat service

5. Other Nonfunctional Requirements

- Performance Requirements
- Safety Requirements
- Security Requirements
- Software Quality Attributes
- Business Rules

Introduction

Purpose

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters, and its goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team, and audience see the product and its functionality.

The product is a web-application. The product scope ranges from buying products to selling products. The purpose of this document is to build an online interior design platform where customers and architects can easily find their requirements and have a simpler and faster shopping experience.

Project Scope

The product is a web application for interior design products, the specific application being enabling the buying and selling of interior design products on the website. The benefit of this application is that it aggregates multiple interior product suppliers into a single marketplace.

Overall Description

Project Perspective

This product is a self-contained product. The product will offer an abundance of products in all domains of interior designing and will be a one-stop solution for interior design products. The product also offers discounts on design products.

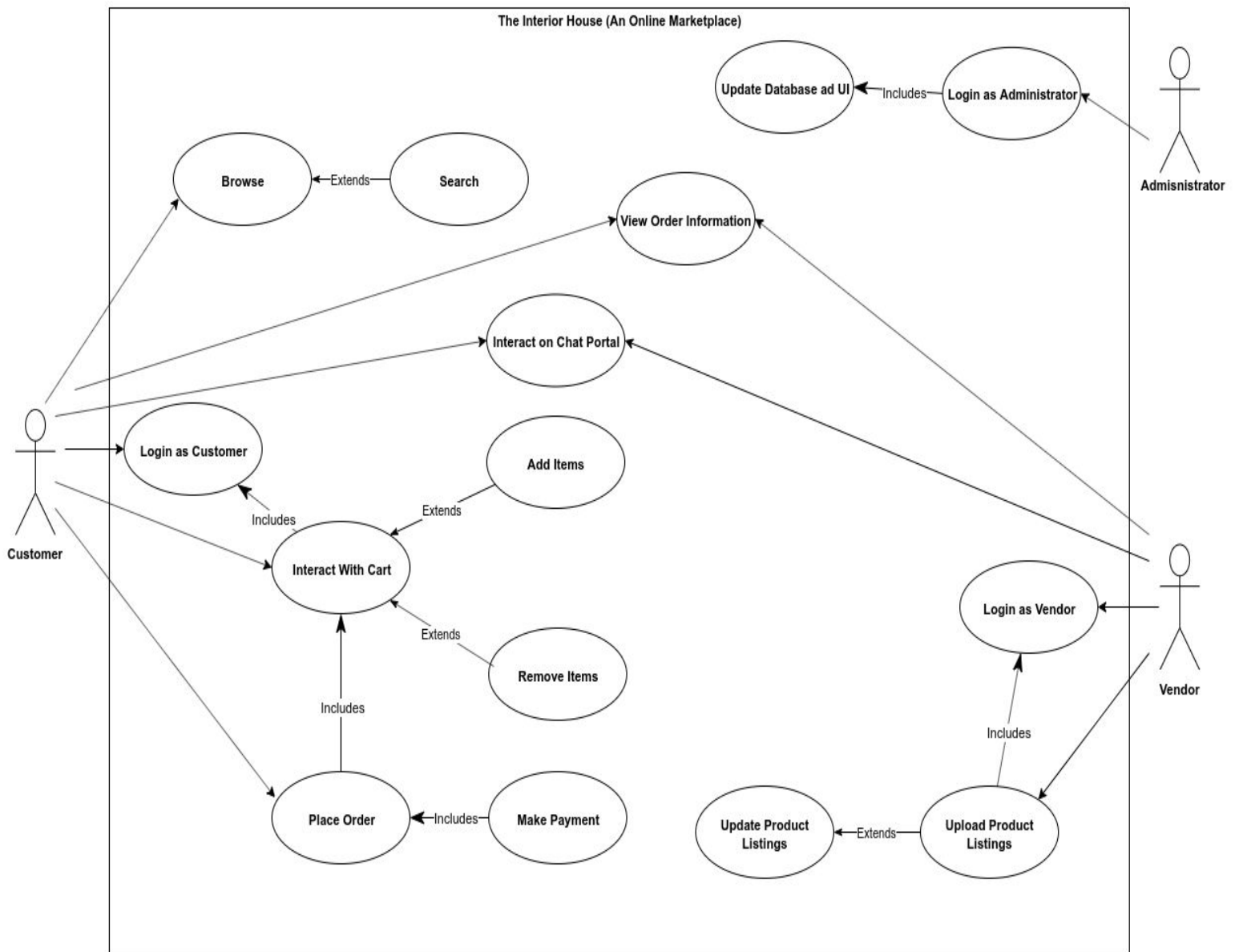
Project Functions

User Functions

- Login or signup
- Add products to the cart
- Remove a product from the cart
- Place order
- Cancel order
- Search products
- Edit or update personal information.

Product Functions

- Provide a simple interface to ease the process of buying and selling products online.
- Display account information of users as well as vendors.
- Display cart
- Take order
- Provide a secure payment gateway
- Provide customer care (FAQs, Chat)
- Notifying order and delivery reports to customers as well as sellers.



User Classes and Characteristics

Customer

The customer is a verified user of the website whose intent is to buy a product from the seller via the platform. The customer must have a username and a password in order to make a purchase. The customer is regularly updated with the latest offers and discounts according to their interests.

Administrator

They are responsible for monitoring functions and procedures on the platform. The administrator is responsible for providing valid information about a purchase to the concerned authority.

Sellers

Seller is a verified person who is allowed to sell products over the platform. Seller's details are stored in the database and all the products are listed under him are available. He is responsible to set product details, their prices, and their quantities.

Operating Environment

- Distributed database
- Client/Server Architecture
- Operating System: Windows/Linux
- Database Management System: MongoDB
- Backend PL: Python

Assumptions and Dependencies

- The architecture entails the use of MongoDB, Nginx, HTML, CSS, and Python assuming all components are well coordinated all the operations performed results in expected behavior the application works as expected.
- Internet connection is a must.
- Proper browsers should be installed in the user's system.

External Interface Requirements

User Interfaces

- Login or Signup page
- Homepage containing products according to the customer's interests.
- There will be a screen displaying information about all the products that user searches.
- If the customer selects any product, then a new page will open containing all the information about the product.
- After the transaction, the system makes the invoice will be generated.
- Customers should be able to add the items they like to their cart or wishlist (Add to Cart/ Wishlist).
- Clicking on the logo of the application displays the homepage.
- The user can search from the search bar the items he/she is looking for.
- The user can contact customer care via chat service.
- Users can ask for assistance or can give feedback on a particular aspect(Support).

Hardware Interfaces

- Since the application must run over the internet, all the hardware shall require to connect to the internet which will be the hardware interface for the system. Eg: Modem, WAN-LAN.
- The system also requires a database to store the transactions of the system like MongoDB.
- The system also requires a DNS (domain name system) for naming on the internet.
- At last, the user requires a web browser to interact with the system.
- The intranet communication protocol used in this application is TCP/IP protocol and HTTPS protocol for communication over the internet.

RESTful API	A RESTful API is an application program interface(API) that uses HTTP requests to GET, PUT, POST and DELETE data.
MongoDB	MongoDB database for storage for data and user information.
Nginx	A webserver

Software Interfaces

Software Used	Description
Operating System	Windows/Linux
Database	To save the transaction records, customer records and vendor records, MongoDB database is used.
HTML/CSS/Javascript	For implementation and interactive support.
Python Flask	For the implementation of the backend of the application.

Communications Interfaces

- Electronic forms will be required for users who haven't registered yet.
- The application shall use the HTTPS protocol for communication over the internet.
- Intranet communication shall be through TCP/IP protocol.
- The user must have an SSL certificate licensing for the registered web browser.

System Features

1. Login Service

Description and Priority

- The different users (buyers, sellers, administrators) should have a username and password using which they need to log in to the system so that they can perform their activities.
- This requires us to implement a login page for the application and a database to store the different usernames and passwords.
- Priority - High.

Stimulus/Response Sequence

- The user visits the login page.
- If the user does not have an account
 - Go to Sign up page
 - Create Account

- At the login page enter a valid username and password
- Click on the submit button

Functional Requirements

- Login UI
 - Form with Submit button
- Backend Database
 - Store as key-value pairs, i.e, {username: uname, password: pwd}
- Possible errors
 - Empty username or password - Do not check the database, display error message to the user and ask to retry.
 - Invalid username or password - Ask the user to retry through the UI

2. Homepage Recommendations

Description and Priority

- Every user has a homepage that displays customized product recommendations to the user based on their search history and ratings given by them to other products.
- This is done by using a recommendation engine algorithm that works on data collected over time by the application.
- Priority - Medium

Stimulus/Response Sequence

- A user uses the application (Search for products, rate products)
- User behavior is recorded
- Recommendation engine updates the recommended products for the user
- Recommended products are displayed on the user's homepage

Functional Requirements

- Recommendation Engine
 - Created using collaborative filtering
 - Dataset (user behavior) collected over time by the application
- Backend Database
 - Stores the recommended products for each user as key-value pairs, i.e, {username1: {product1: pid1, product2: pid2,},}
- Homepage UI
 - Recommended products are displayed here
- Possible errors

- User is inactive or a new user (No dataset available) - Display a default list of products on the homepage.

3. Search Feature

Description and Priority

- The search system feature enables the users to search for specific products from the list of available products in the database. It allows users to enter search text on the screen and display matching products based on the search.
- It also notifies the user when no matching product is found by the search.
- This requires us to implement a search engine for the application using client-side Javascript libraries.
- Priority- High

Stimulus/Response Sequence

- The user types in the name of the product he wants.
- If it is available, the list of items is displayed otherwise it shows that the particular item is not available.

Functional Requirements

- Search bar UI
- Backend database of all the products available
- Possible errors
 - Search string does not match any available products

4. Shopping Cart

Description and Priority

- The buyers should be able to add those products to their cart which they would like to purchase, right now or later on.
- This requires us to create a shopping cart feature that displays all the products in the cart. The backed database will save the cart information for every user.
- Priority - Medium

Stimulus/Response Sequence

- The user clicks on the 'Add to Cart' button.
- If the product is in stock, it gets added to the user's cart otherwise, it shows Out Of Stock.

Functional Requirements

- Shopping cart UI
- Backend database
 - Stores shopping cart information for every user
- Possible errors
 - A product in the cart may run out of stock.
 - The product quantity in the database might be wrong.

5. Checkout System

Description and Priority

- After the user has added products to their cart, they should be allowed to checkout by setting their delivery location and making payment by different means.
- This requires us to create a checkout system that is secure. We accomplish this by using a third party payment gateway.
- Priority- High

Stimulus/Response Sequence

- The buyer clicks on the Proceed to Buy button.
- The buyer inputs the shipping address and contact numbers. Any applicable vouchers can be chosen.
- Different payment options are displayed, after which the buyer gets redirected to a secure third-party payment gateway.
- After completion of payment, order details are generated and saved in the buyer's profile.

Functional Requirements

- Add to Cart UI
- Checkout UI
- Backend database
 - Stores shopping cart information for every user
- Secure payment gateway
- Possible errors
 - The connection gets terminated mid-checkout.
 - The buyer provides invalid payment details.

6. View Order Details

Description and Priority

- The buyers should be able to view the order details of all their previous and ongoing orders.
- Each order detail contains shipping information, vendor information, and payment information.
- Priority - High

Stimulus/Response Sequence

- User clicks on My Orders option present on the menu bar
- A page with the user's orders (previous and ongoing) are listed
- The user selects an order from the list
- The selected order details are displayed

Functional Requirements

- Tracking UI
 - Order tracking facility should be provided
- Backend Database
 - Stores order details for each user

7. Seller lists Products

Description and Priority

- The Vendors should be able to add a list of all the products that are available for sale.
- This requires us to implement an input feature using which the vendor can list out all the products and the respective product details such as quantity and price.
- Priority - High

Stimulus/Response Sequence

- The vendor clicks on the 'Add New Product' button.
- The vendor is prompted for product details, which they then input.
- The vendor clicks on the 'Confirm New Product' button.

- The new product is then added to the list of products that are sold by the vendor in the Vendor Database.
- The new product is displayed to the customer.

Functional Requirements

- Add to Product Roster UI.
 - Form with prompts for Product Description and details.
- Backend database containing product information for every vendor.
 - Stored as Key-Value Pairs. [Ex. Name: XYZ, Price: Rs. 3300, Qty: 10, Desc: ABC]
- Possible Errors
 - Invalid Quantity Entered [Ex. -1]
 - Duplicate Product details entered by the Vendor may ruin DB Consistency.

8. Customer Care Chat Service

Description and Priority

- If any user is facing issues with using the application, they can use the chat service on the application to speak to a customer care executive.
- Priority - Low

Stimulus/Response Sequence

- User opens the chat service
- User waits for a customer care executive to be assigned
- Customer care executive is assigned
- Messages are sent and received from both sides
- Chat is closed

Functional Requirements

- Chat UI
 - Implemented using AJAX
- Customer care executive assignment algorithm
- Possible errors
 - No customer care executive - the user is notified and chat is closed

Other Nonfunctional Requirements

Performance Requirements

- The product shall be based on the web and must be run from a web browser.
- The product shall take initial load time depending on the internet connection which also depends on the media from which the product is run.
- The performance shall depend upon the hardware components of the client/customer.

Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

Security Requirements

1) Data Transfer

- The system shall use secure sockets in all the transactions that include any confidential customer information.
- The system shall confirm all the transactions with the user's browser.
- The system shall not leave any cookies on the customer's computer containing user password or confidential information.

2) Data Storage

- The customer's web browser shall never display the customer's password. It shall always be echoed with special characters representing typed characters.
- The customer's web browser shall never display a customer's credit card number after retrieving it from the database. It shall always be shown with the last 4 digits of the credit card number.
- The system's backend servers shall only be accessible to the system administrators.
- The system's backend databases shall be encrypted and within the company's perimeter.

Software Quality Attributes

Reliability

- The system should be backed up which is continuously maintained and updated to reflect the most recent changes.
- The reliability of the overall application depends on the reliability of the separate components.

Maintainability

- A commercial database is used for maintaining the database.
- In case of failure, a re-initialization of the program will be done.

Other Non-functional requirements of the application are:

- Portability
- Extensibility
- Reusability
- Compatibility
- Resource Utilization
- Robustness
- Testability
- Usability

Business Rules

1. Administrator

The administrator is the superuser and has complete control over the activities that can be performed. The application notifies the administrator of all the incoming requests which he/she can accept or reject. He also manages the list of available product categories. He can also view or delete entries from the guestbook.

2. Customer

The customer can browse through the application and choose the products to place in the virtual cart. The shopping cart details can be viewed and items can be removed from the cart. To proceed with the purchase, the customer is prompted with a login. Also, the customers can modify their personal information, such as phone number and shipping address stored by the application. The customers can also view any of their previous orders and cancel any order that hasn't been shipped yet.

3. Vendors

The vendors can post the items they want to sell on the platform. They can modify the prices of the items and also remove items from the list. They can also view the transactions related to their items. Customer feedback can also be reviewed.

4. Employees

- Purchase Department under a purchase manager to overlook purchasing activities if any warehousing needs arise.
- Sales Department under a sales manager who will look after the sale of products and services.
- Accounts Department under an accounts manager to look after the accounting activities.