



Project of AAA Furnitures

Software Requirement Specification

Version
1.0

Status
DRAFT adapted from IEEE SRS Template

Proposed by:
Andreandhiki Riyanta Putra
Andrian Danar Perdana
M. Argya Vityasy

Instructor
Khabib Mustofa



Contents

1	Introduction	5
1.1	Purpose	5
1.2	Project Scope	5
1.3	Definitions, Acronyms, and Abbreviations	6
1.4	References	6
2	Overall Description	7
2.1	Product Perspective	7
2.2	Product Functions	7
2.3	User Classes and Characteristics	8
2.4	Operating Environment	8
2.5	Design and Implementation Constraints	9
2.6	Assumptions and Dependencies	9
3	External Interface Requirements	10
3.1	User Interfaces	10
3.1.1	Landing Page	10
3.1.2	Warehouse Page	11
3.1.3	Catalogue Page	12
3.1.4	Collection Page	12
3.1.5	Product Page	13
3.2	Hardware Interfaces	13
3.3	Software Interfaces	13
3.4	Communications Interfaces	14
4	System Features	15
4.1	Product Management	15
4.1.1	Description and Priority	15
4.1.2	Stimulus/Response Sequences	15
4.1.3	Functional Requirements	15

PROJECT OF AAA FURNITURES

Software Requirement Specification



4.2 Product Catalogue Display	16
4.2.1 Description and Priority	16
4.2.2 Stimulus/Response Sequences	16
4.2.3 Functional Requirements	16
4.3 Contact Form Submission	16
4.3.1 Description and Priority	16
4.3.2 Stimulus/Response Sequences	17
4.3.3 Functional Requirements	17
4.4 Warehouse Location Display	17
4.4.1 Description and Priority	17
4.4.2 Stimulus/Response Sequences	17
4.4.3 Functional Requirements	17
5 Other Nonfunctional Requirements	18
5.1 Performance Requirements	18
5.2 Safety Requirements	18
5.3 Security Requirements	19
5.4 Software Quality Attributes	19
5.5 Business Rules	19



Revision History

Name	Date	Reason For Changes	Version
21	22	23	24
31	32	33	34



CHAPTER **1**

Introduction

1.1 PURPOSE

The purpose of this document is to present the requirements for the AAA Furnitures Catalog Website. This web-based application allows customers to browse furniture products and contact the company via WhatsApp for purchases. The document describes the system's features, constraints, and interface specifications. It serves as a reference for stakeholders and developers to create the initial version of the product catalog platform.

1.2 PROJECT SCOPE

AAA Furnitures Catalog Website is a responsive web application that enables customers to:

- Browse furniture products with detailed information
- Filter and search through product listings
- Contact the company via WhatsApp for purchases

Administrators can manage product information through a dedicated interface. The system consists of a single product service with a database, eliminating the need for user accounts, payment processing, or order tracking features.



1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term	Definition
Visitor	Any user accessing the website to browse products
Admin	Authorized personnel managing product information
Product	Furniture item displayed in the catalog
API	Application Programming Interface for data management[1]
REST	Architectural style for API communication[2]
SQL	Language for relational database management[3]
LCP Test	Measures largest content load time (Ideal: 2.5s)
FCP Test	Measures first content load time (Ideal: 1.8s)

Table 1.1: Definitions, Acronyms, and Abbreviations

1.4 REFERENCES

- [1] J. Ofoeda, R. Boateng, and J. Effah, "Application programming interface (api) research: A review of the past to inform the future," *International Journal of Enterprise Information Systems*, vol. 15, pp. 76–95, Jul. 2019. DOI: 10.4018/IJEIS.2019070105.
- [2] N. Patel, "Representational state transfer (rest) application program interface (api)," Jun. 2018.
- [3] Y. Silva, I. Almeida, and M. Queiroz, "Sql: From traditional databases to big data," Feb. 2016, pp. 413–418. DOI: 10.1145/2839509.2844560.



CHAPTER 2

Overall Description

2.1 PRODUCT PERSPECTIVE

The product is a responsive web catalog with direct WhatsApp integration for customer inquiries. Key components include:

- Frontend interface for product browsing
- Backend product management system
- Product database
- WhatsApp Business API integration

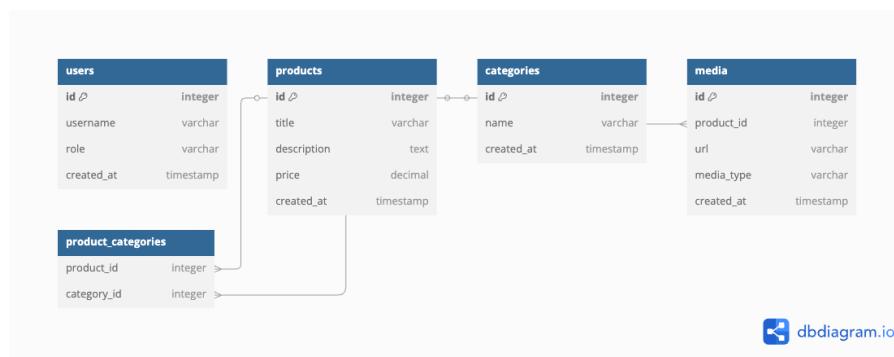


Figure 2.1: ERD

2.2 PRODUCT FUNCTIONS

- Product Presentation
 - Display product categories
 - Show product details (images, specifications, pricing)
 - Item search and filtering functionality
- WhatsApp Integration

PROJECT OF AAA FURNITURES

Software Requirement Specification



- Pre-filled message templates for product inquiries
- Direct click-to-chat functionality
- Admin Functions
 - Add/remove products
 - Update product information
 - Manage product categories
 - Manage communication with users

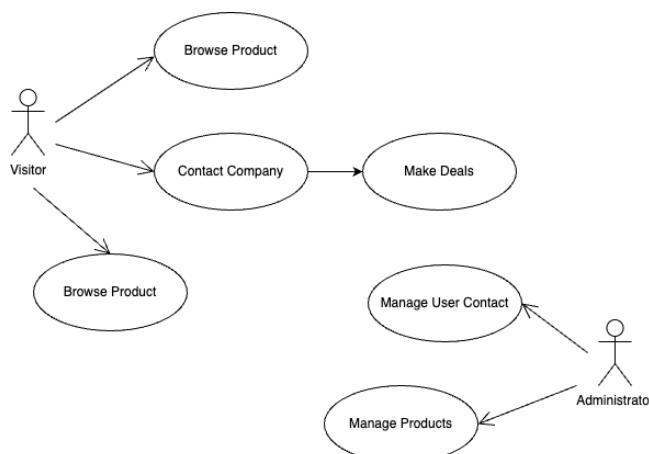


Figure 2.2: Updated Use Case Diagram

2.3 USER CLASSES AND CHARACTERISTICS

- Website Visitors
 - Frequency: Variable (casual browsing to serious buyers)
 - Functions: Product browsing, WhatsApp contact
 - Technical Expertise: Basic web navigation skills
- Administrators
 - Frequency: Daily updates
 - Functions: Full product catalog management
 - Security: Protected admin interface

2.4 OPERATING ENVIRONMENT

- Server Environment
 - Web server with product database
 - CMS for content management



- Client Environment
 - Modern web browsers
 - Mobile-first responsive design
- Database
 - Relational database (MySQL/PostgreSQL)
 - Cloud Bucket to store image blob (AWS S3 Bucket)

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

- Technical Constraints
 - Responsive design requirements
 - WhatsApp Business API integration
 - Optimized media loading for product images
- Performance Constraints
 - Page load time <3s for 95% of users
 - Support 500 concurrent users
- Security Constraints
 - Secure admin interface
 - Regular database backups

2.6 ASSUMPTIONS AND DEPENDENCIES

- Assumptions
 - Users have WhatsApp installed
 - Primary access via mobile devices
 - Product availability matches physical inventory
- Dependencies
 - WhatsApp Business API availability
 - Web hosting reliability
 - Image CDN for product photos
- Risks
 - WhatsApp API changes
 - Inventory discrepancies
 - High bounce rates from casual browsers



CHAPTER 3

External Interface Requirements

3.1 USER INTERFACES

3.1.1 Landing Page

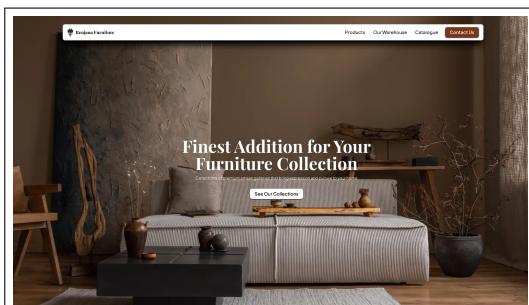


Figure 3.1: Hero Section

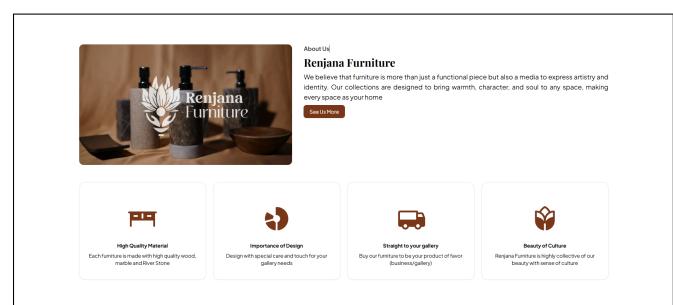


Figure 3.2: About Section

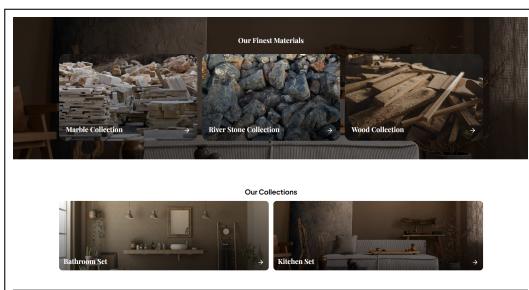


Figure 3.3: collections Section



Figure 3.4: Custom Design Section

PROJECT OF AAA FURNITURES

Software Requirement Specification

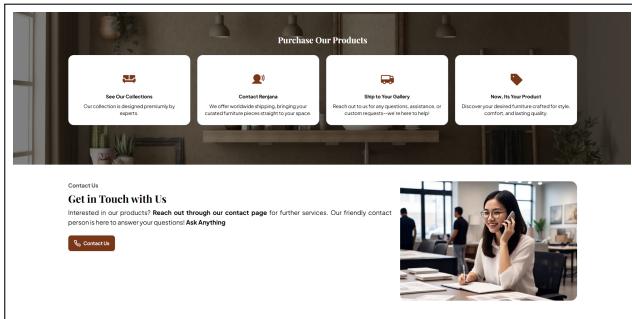


Figure 3.5: Timeline and Contact Section

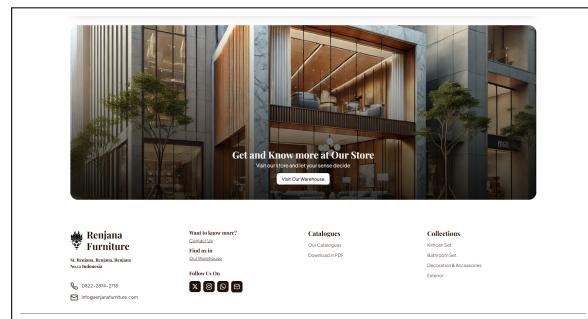


Figure 3.6: CTA Section

- First-time users accessing the website will land on this page.
- Displays an overview of the product catalog, key collections, and featured products.
- Includes navigation links to the catalog, collections, warehouse, and contact page.
- CTA buttons allow users to download the catalog, contact support, or browse collections.

3.1.2 Warehouse Page

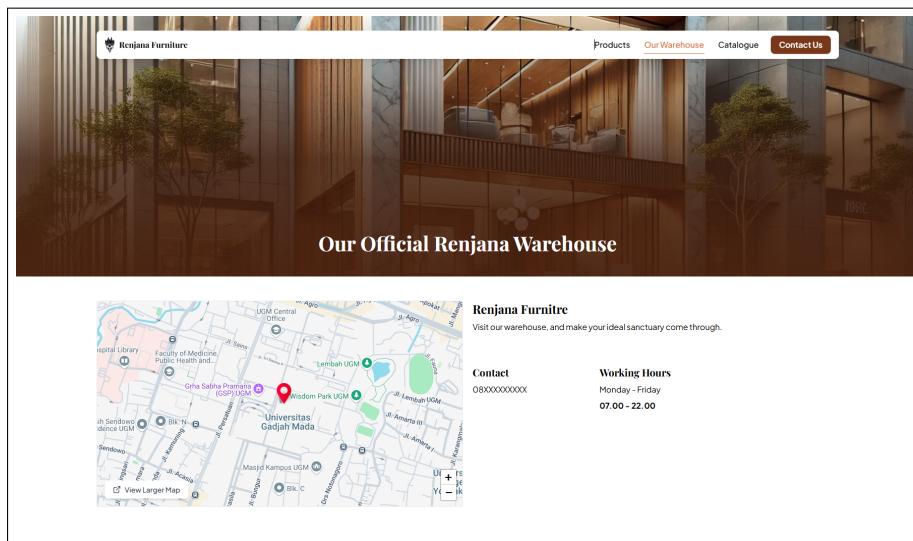


Figure 3.7: Warehouse Page

- Shows the business's warehouse/store location using Google Maps API.
- Displays the address and contact details.
- A CTA button allows users to open the location directly in Google Maps for navigation.



3.1.3 Catalogue Page

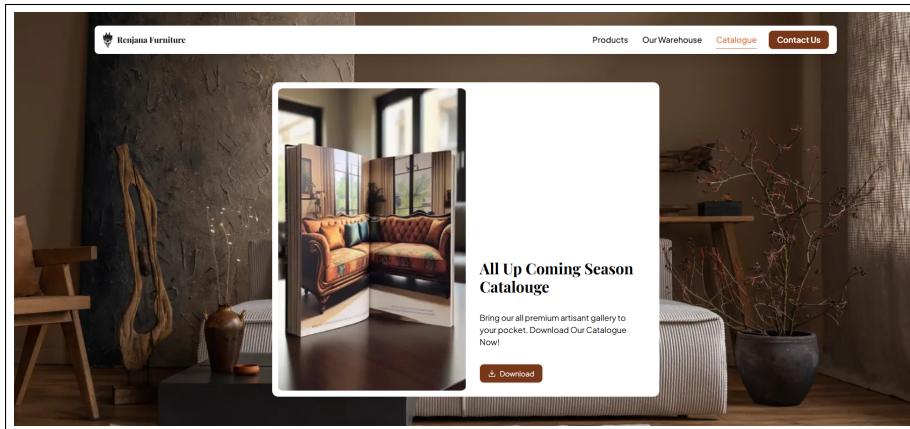


Figure 3.8: Catalogue Page

- Provides a dedicated section for users to download the latest catalog in PDF format.
- Includes a clear Download CTA button.

3.1.4 Collection Page

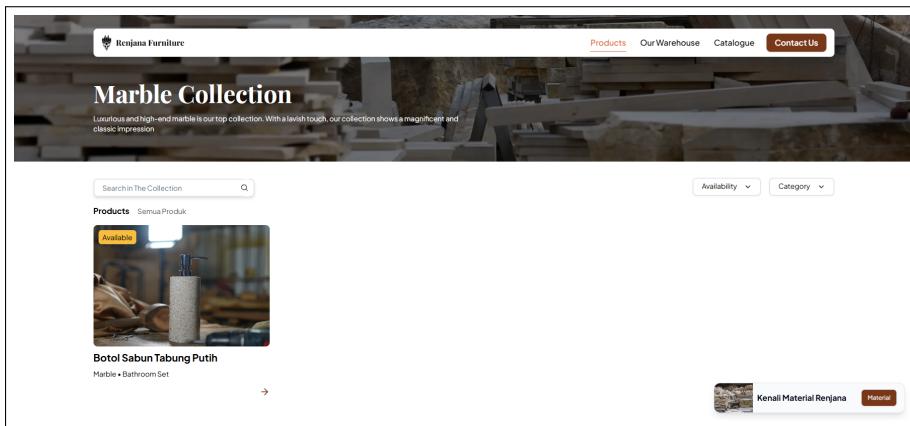


Figure 3.9: Collection Page

- Displays product categories in two main sections:
 - (a) Materials Collection (Wood, Marble, River Stone).
 - (b) Other Categories (Kitchen Set, Bathroom Set, Decoration Accessories, Exterior).
- Users can filter and browse products based on selected categories.



3.1.5 Product Page

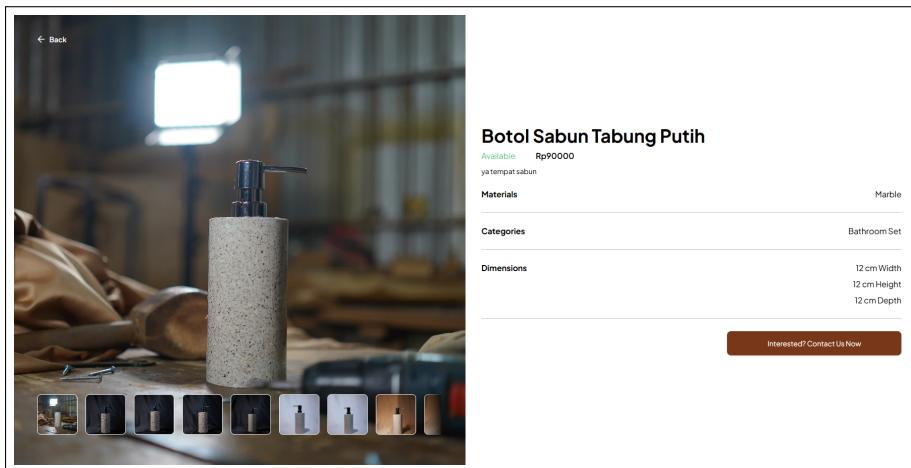


Figure 3.10: Catalogue Page

- Displays product details, including:
 - (a) Product images (stored in Supabase/Vercel Postgres).
 - (b) Name, description, dimensions (x, y, z), availability, and optional price.
- Users can contact the business via WhatsApp for further inquiries.

3.2 HARDWARE INTERFACES

The catalog website and the admin dashboard do not depend on specific hardware components, and therefore, no direct hardware interfaces are required. All product-related data, including images, descriptions, and other relevant details, are managed within the web platform. Payment transactions are facilitated through direct communication via WhatsApp rather than a dedicated payment processing hardware.

The database server connection is handled by the underlying infrastructure provided by the hosting service, ensuring reliable data storage and retrieval for both visitors and administrators. This approach abstracts hardware dependencies, allowing the system to operate efficiently across various environments.

3.3 SOFTWARE INTERFACES

The catalog website interacts with various software components to ensure seamless functionality and efficient data management. It is developed using Next.js as the primary frontend framework, with Tailwind CSS for styling and responsive design. The backend is managed through Payload CMS, which facilitates content organization and retrieval.

The system utilizes Supabase (Vercel Postgres) as its primary database for storing product details, user interactions, and other relevant data. Additionally, Supabase storage is used to handle and serve product images, ensuring optimized retrieval and delivery. Communication between the frontend and backend occurs via API requests, allowing structured data exchange and synchronization.



For transactions, the website does not include an integrated payment gateway but instead facilitates purchases through direct WhatsApp messaging. Furthermore, third-party libraries and services are integrated for image optimization, SEO enhancements, and performance monitoring to improve user experience and website visibility.

Data sharing between software components follows an API-driven approach, ensuring security, consistency, and scalability. If required, additional implementation constraints, such as authentication mechanisms and rate limiting, can be applied to protect data integrity.

3.4 COMMUNICATIONS INTERFACES

The catalog website relies on standard web communication protocols to ensure seamless data exchange between the frontend, backend, and external services. The system primarily uses HTTPS for secure communication and is accessible through modern web browsers.

Data transactions between the frontend and backend are managed through GraphQL API provided by Payload CMS, allowing efficient retrieval and management of product information stored in Supabase (Vercel Postgres).

For customer inquiries via the Contact Us form, the system integrates with Google Sheets API to store submitted form data in a centralized Google Sheet. Additionally, Nodemailer is used to send email notifications based on form submissions, ensuring that administrators are promptly informed of customer inquiries.

To enhance the user experience, the website also integrates Google Maps API, which provides an interactive map displaying the warehouse location. This allows users to easily locate the company's physical store or storage facility.

Since transactions are conducted outside the platform via WhatsApp messaging, no payment gateway integration is required. WhatsApp's built-in encryption ensures secure communication between customers and the business.

To maintain data integrity and security, all API communications follow industry best practices, including authentication mechanisms, rate limiting, and encrypted data transmission where applicable.



CHAPTER **4**

System Features

4.1 PRODUCT MANAGEMENT

4.1.1 Description and Priority

This feature enables administrators to efficiently manage the product catalog by adding, editing, and deleting products. Each product entry includes essential details such as name, description, availability status, dimensions (X, Y, Z), category, and optional pricing. Additionally, product images are stored securely in Supabase (Vercel Postgres). Given its critical role in maintaining the catalog, this feature is assigned high priority.

4.1.2 Stimulus/Response Sequences

1. The admin logs into the dashboard or admin panel.
2. The admin selects Add Product, fills in the product form, and submits it.
3. The system stores the data in the database via Payload CMS.
4. The admin selects Edit Product, modifies the details, and saves the changes.
5. The system updates the product data in the database.
6. The admin selects Delete Product, and the system removes the product from the database.

4.1.3 Functional Requirements

- REQ-1: The system must provide an admin dashboard to manage products.
- REQ-2: The system must support CRUD (Create, Read, Update, Delete) operations for products.
- REQ-3: Product images must be stored and retrieved from Supabase (Vercel Postgres).
- REQ-4: The system must display error messages for invalid inputs.



4.2 PRODUCT CATALOGUE DISPLAY

4.2.1 Description and Priority

This feature ensures that products in the catalog are displayed in an organized manner based on predefined collections. The catalog is structured into two main categories:

1. Materials Collection: Includes products categorized by material, such as Wood, Marble, and River Stone.
2. Other Categories: Groups products based on their function or placement, including **Kitchen Set, Bathroom Set, Decoration Accessories, and Exterior.

This feature is high priority, as it directly impacts the browsing experience and usability of the catalog for customers.

4.2.2 Stimulus/Response Sequences

1. User Action: Opens the catalog landing page.
 - System Response: Displays collections section and navigation bar with various predefined collections with a categorized product listing.
2. User Action: Selects a collection (e.g., "Wood" or "Kitchen Set").
 - System Response: Filters and presents products from the selected collection.
3. User Action: Clicks on a product to view its details.
 - System Response: Displays the product's name, description, dimensions, availability status, images, and optional price.

4.2.3 Functional Requirements

- REQ-5: The system must categorize products into Materials Collection and Other Categories.
- REQ-6: The catalog page must display collections in a structured manner.
- REQ-7: Users must be able to filter products by collection.
- REQ-8: The system must provide a detailed product view when a user selects a product.

4.3 CONTACT FORM SUBMISSION

4.3.1 Description and Priority

This feature allows users to submit inquiries or information requests via the contact form. Data is stored in Google Sheets API, and the admin receives email notifications via Nodemailer. This feature has a medium priority.



4.3.2 Stimulus/Response Sequences

1. User Action: Fills out the contact form and clicks Submit.
 - System Response: Stores the data in Google Sheets API.
2. System Action: Sends an email notification to the admin using Nodemailer.
3. User Action: Receives a confirmation message indicating that the inquiry was successfully submitted.

4.3.3 Functional Requirements

- REQ-8: The contact form must store submissions in Google Sheets API.
- REQ-10: The system must send email notifications to the admin via Nodemailer.
- REQ-11: The system must display a confirmation message after form submission.

4.4 WAREHOUSE LOCATION DISPLAY

4.4.1 Description and Priority

This feature displays the warehouse/store location using Google Maps API, allowing users to easily find the business's physical address. This feature has a low priority, but it enhances the user experience.

4.4.2 Stimulus/Response Sequences

1. User Action: Opens the warehouse location page.
 - System Response: Displays the warehouse address along with a map preview.
2. User Action: Clicks the CTA button to open the location.
 - System Response: Redirects the user to Google Maps, opening the coordinates in the app or web version.

4.4.3 Functional Requirements

- REQ-12: The system must display the warehouse location on a map.
- REQ-13: The system must provide a CTA button that allows users to open the location in Google Maps.
- REQ-14: Clicking the CTA button must redirect users to Google Maps with the correct coordinates.



CHAPTER 5

Other Nonfunctional Requirements

5.1 PERFORMANCE REQUIREMENTS

The AAA Furnitures web app must ensure efficient performance under various conditions to enhance the user experience and support business operations.

- **Page Load Speed:** All web pages should achieve a Largest Contentful Paint (LCP) of under 2.5 seconds and a First Contentful Paint (FCP) of under 1.8 seconds under standard network conditions.
- **Search and Filtering:** The product search and filtering functions should return results within 2 seconds to ensure seamless navigation.
- **Image Optimization:** Product images must be served in WebP format and lazy-loaded to ensure they load in under 2 seconds.
- **API Performance:** Backend APIs should respond within 200ms to ensure smooth interaction between frontend and backend components.
- **Database Queries:** Queries should be optimized to execute within 100ms to prevent bottlenecks.

5.2 SAFETY REQUIREMENTS

Safety measures are essential to protect user data and ensure proper handling of catalogue data.

- **Error Handling:** The system should provide informative error messages and prevent crashes or unresponsive states when encountering unexpected user input. A dedicated 404 Error page is required.
- **Backup and Recovery:** Regular backups must be maintained to prevent data loss due to system failures.
- **Product Descriptions and Pricing:** The administrators must verify that product descriptions and pricing updates do not result in discrepancies that may mislead customers.



5.3 SECURITY REQUIREMENTS

Security measures focus on protecting user data and preventing unauthorized access.

- **Admin Panel Security:** The admin panel is only accessible via a designated URL path (/admin) and requires authentication to prevent unauthorized modifications.
- **HTTPS Compliance:** The web app enforces HTTPS to encrypt all communications between users and the server, ensuring secure transactions.
- **Database Security:** Supabase policies are implemented to restrict unauthorized access and ensure only authenticated users can perform certain operations.
- **Rate Limiting:** API rate limits should be enforced to mitigate abuse and prevent denial-of-service attacks.

5.4 SOFTWARE QUALITY ATTRIBUTES

The AAA Furnitures web app should adhere to high-quality software standards to enhance usability, maintainability, and reliability.

- **Usability:** The interface should be intuitive, ensuring users can browse, search, and complete purchases with minimal effort.
- **Reliability:** The system should achieve at least 99.5% uptime to ensure availability for customers.
- **Scalability:** The application must be scalable to accommodate increased traffic and inventory expansion without performance degradation.
- **Maintainability:** The codebase should be modular and well-documented to allow for future enhancements and bug fixes.
- **Compatibility:** The web app should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and mobile devices.

5.5 BUSINESS RULES

The AAA Furnitures web app must follow operational principles that dictate how different roles interact with the system.

- **User Roles:** The system supports different roles such as customers and administrators. Customers do not need to create an account and can browse products and place orders, while administrators can manage inventory, orders via the admin panel.
- **Inventory Management:** Only authorized administrators can add, update, or remove products from the inventory.
- **Customer Support:** Users should be able to submit inquiries or complaints through a dedicated support line, either through WhatsApp or email linked on the footer of the website.