Software Functional Requirements Document

For: Doompause Emporium Ecommerce Application

Daniel Rumfelt, Ryan Coon, and Christa Moncrief

Grand Canyon University: CST-326

09/17/2023

Document Version 1.4

09.17.2023

**AUTHORS**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Department** |
| Daniel Rumfelt | Backend and IoT Site Security | Software Engineering |
| Christa Moncrief | Frontend, GUI and Database Development | Web Design, GUI |
| Ryan Coon | Backend and Database Development | Software Engineering |

**DOCUMENT HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Document Revision Description** | **Document Author** |
| 09/14/23 | 1.0 | Document Created with Basic Details | Daniel Rumfelt |
| 09/15/23 | 1.1 | Added Application Architecture | Daniel Rumfelt |
| 09/16/23 | 1.2 | Added all Diagrams and Wireframes | Ryan Coon |
| 09/16/23 | 1.3 | Form Elements Created | Daniel Rumfelt |
| 09/17/23 | 1.4 | Added Form Rules and Buttons | Christa Moncrief |
|  |  |  |  |

**APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Approval Date** | **Approved Version** | **Approver Role** | **Approver** |
| 09/17/23 | 1.2 | Backend and IoT Site Security | Daniel Rumfelt |
| 9/17/23 | 1.4 | Backend and Database Development | Ryan Coon |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc134439848)

[1.1. Purpose of the Document 4](#_Toc134439849)

[1.2. Project Scope 4](#_Toc134439850)

[1.3. Related Documents 4](#_Toc134439851)

[1.4. Terms/Acronyms and Definitions 4](#_Toc134439852)

[1.5. Risks and Assumptions 4](#_Toc134439853)

[2. System/Solutions Overview 5](#_Toc134439854)

[2.1. Context Diagram/ Interface Diagram/ Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow 5](#_Toc134439855)

[3. Functional Specifications 5](#_Toc134439856)

[3.1. Architecture 5](#_Toc134439857)

[3.2. Purpose/Description 6](#_Toc134439858)

[3.3. Sitemap 6](#_Toc134439859)

[3.4. Wireframes 6](#_Toc134439860)

[3.5. Functional Requirements 6](#_Toc134439861)

[3.6. Field-Level Specifications 6](#_Toc134439862)

[3.7. Nonfunctional Requirements 8](#_Toc134439863)

[4. References 9](#_Toc134439864)

[5. Open Issues 9](#_Toc134439865)

[6. Appendix 9](#_Toc134439866)

# Introduction

This document will address is the **development of a web application for online sales**. The company, with its rich history in the trade of medieval weapons, is seeking to expand its reach to customers globally through an online platform.

**Background Information:** The company has been in business for hundreds of years, offering medieval weapons for both defensive and offensive combat. Over the years, it has built a reputation for quality and authenticity. However, with changing times and the rise of digital commerce, the company has found it increasingly difficult to reach customers.

The proposed web application will allow users to view items, read weapon descriptions, and see the prices of the items. It will also offer an online shopping cart where customers can add, remove, and edit the quantity of products. This digital transformation aims to make shopping more convenient for customers and expand the company’s customer base.

By addressing these needs, the company hopes to continue its tradition of excellence while adapting to the modern marketplace.

**Purpose of the Document**

A **Functional Specification Document (FSD)** is a formal document that details all features and functionalities of a software system that is to be developed. It serves as a guide for the development team and a reference for stakeholders to understand what the system will do.

The intended purpose of an FSD is multi-fold:

* It provides a detailed description of the system’s functionalities, including how the system will interact with users and other systems.
* It outlines the system’s performance requirements, interface design, and expected behaviors.
* It serves as a contract between the stakeholders and the development team, ensuring everyone has a clear understanding of what is to be built.
* It guides the development process and helps in estimating the resources, time, and cost involved in the project.

In essence, an FSD is a blueprint for building a software system. It ensures that all parties involved in a project have a shared understanding of what the software should do and how it should function before development begins. This helps prevent misunderstandings and conflicts later in the project lifecycle.

Please note that while this is a standard purpose of an FSD, it can be modified as needed to suit specific project requirements or organizational practices.

## Project Scope

The project scope is to develop a web application for the company to sell its medieval weapons to online customers. This application will address the business need of reaching a wider customer base and adapting to the changing shopping habits of customers.

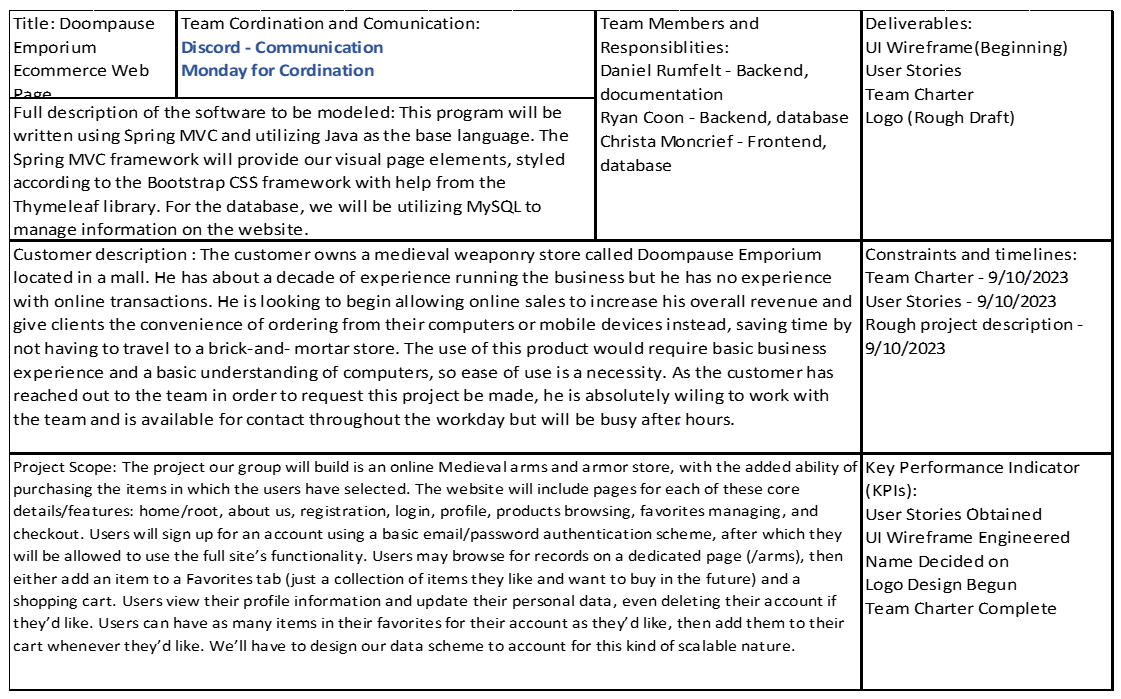
**High-Level Information of the Solution:**

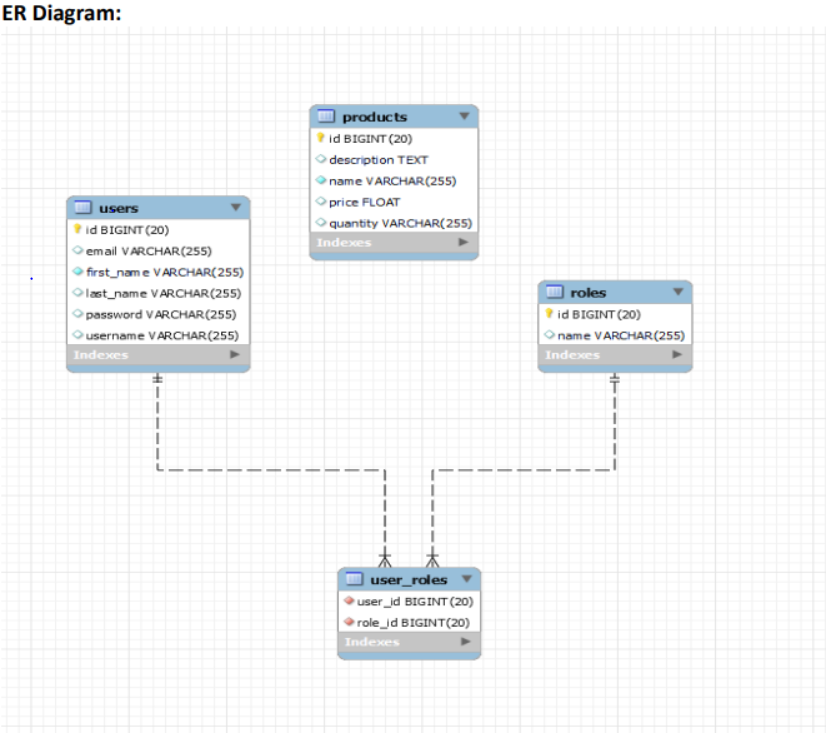
The web application will have the following features:

1. **Product Display:** The application will display a variety of medieval weapons. Each item will have a detailed description and price.
2. **Shopping Cart:** Customers will be able to add, remove, and edit the quantity of products in their shopping cart. This feature will provide customers with a flexible and convenient shopping experience.
3. **User-Friendly Interface:** The application will have an intuitive and easy-to-navigate interface, making it easy for customers to browse products and make purchases.
4. **Secure Payment Gateway:** The application will integrate a secure payment gateway for safe and secure transactions.

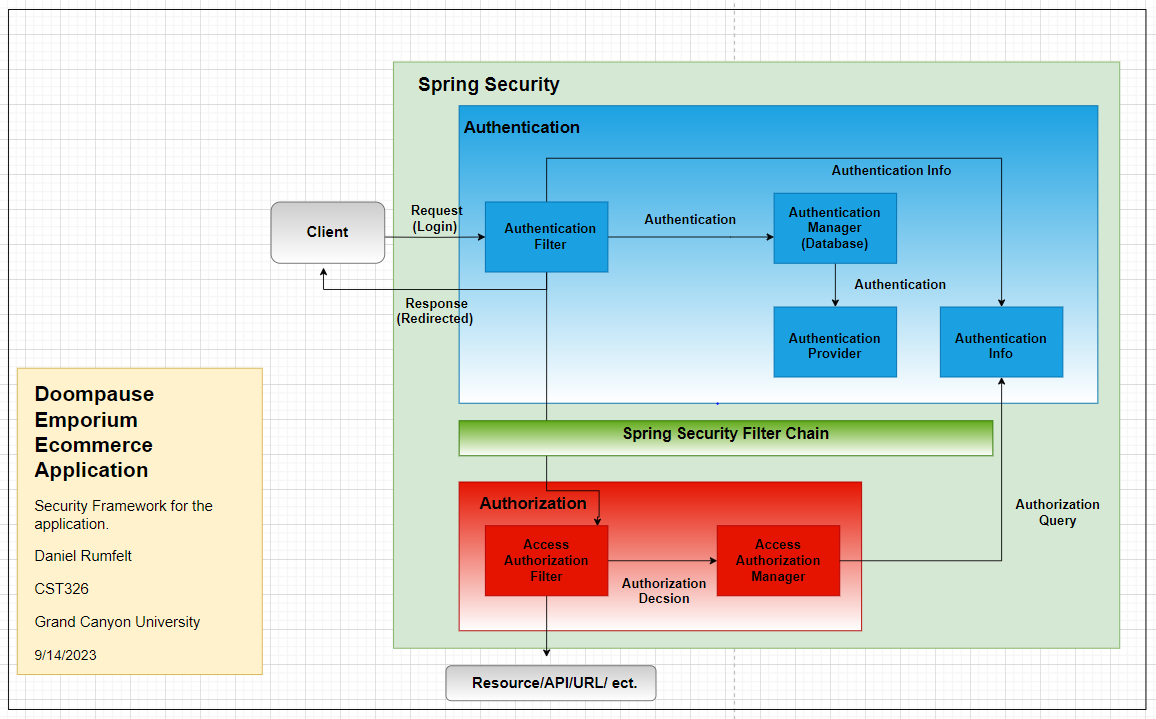
This solution aims to bring the company’s hundreds of years of business into the digital age, making its unique products easily accessible to customers around the globe. By doing so, it hopes to continue its tradition of excellence while adapting to modern market demands.

## Related Documents





*Spring Security Framework for Application:*



## Terms/Acronyms and Definitions

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definition** | **Description** |
| FSD | Functional Spec Document | A Functional Specification Document (FSD) is a formal document that details all features and functionalities of a software system that is to be developed. It serves as a guide for the development team and a reference for stakeholders to understand what the system will do. |
| ER Diagram | Entity Relationship (ER) How entities relate to each other. | An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. |
| Data flow | Data flow is the movement of data through a system comprised of software, hardware or a combination of both. | Data flow is often defined using a model or diagram in which the entire process of data movement is mapped as it passes from one component to the next within a program or a system, taking into consideration how it changes form during the process. |
| Order flow | Term used to describe the flow of buy and sell orders in a market. It is the rawest form of data that shows the activity and interest of buyers and sellers at different price levels. | Order flow traders use various tools and indicators to analyze the order flow and identify market imbalances, trends, reversals, and significant price levels. Order flow trading is a type of trading strategy that aims to profit from the order flow information. |
| Profile flow | The way of describing how users interact with an application or a website, such as when they sign in, sign up, edit a profile, or reset a password. | Design aid for a user-friendly and secure experience for your users, as well as collect and manage their data. |

## Risks and Assumptions

**Assumed Factors:**

1. **Third-Party Components:** The application may use third-party components for certain functionalities, such as the payment gateway. It is assumed that these components will be compatible with the rest of the application and will function as expected.
2. **Operating Environment:** The application is expected to operate in a stable and secure environment. Any issues with the operating environment, such as server downtime or security breaches, could affect the functional design of the application.

**Identified Risks:**

1. **Integration Risks:** There could be risks associated with integrating third-party components or commercial components into the application. These components may not function as expected or may not be fully compatible with the rest of the application.
2. **Security Risks:** As the system will handle sensitive customer data and financial transactions, there is a risk of security breaches. This could affect the integrity and confidentiality of data.
3. **Performance Risks:** The application is expected to handle a high volume of traffic and transactions. There is a risk that the application may not perform optimally under heavy load.
4. **Compliance Risks:** The application must comply with all relevant laws and regulations, including data protection and privacy laws. Non-compliance could result in legal penalties.

**Constraints:**

1. **Budget Constraints:** The budget for the project may limit the scope of the functional design. For example, it may not be possible to include all desired features due to budget constraints.
2. **Time Constraints:** The project timeline may also constrain the functional design. The system must be designed and developed within the specified timeline.
3. **Technical Constraints:** There may be technical constraints related to the development platform or technology stack used for the project. These constraints could limit the functionalities that can be included in the application.

# System/Solutions Overview

The software being specified is a web application designed to facilitate online sales for a Doompause Emporium specializing in medieval weapons. The application will be built using **JavaScript**, **HTML**, and **CSS** for the frontend, ensuring a user-friendly interface and seamless user experience.

The backend will be powered by **Spring Boot** and **Spring Boot Security**, providing robust, secure, and scalable server-side operations. **MySQL** will be used for database management, ensuring efficient data storage and retrieval.

**Purpose:** The purpose of this web application is to expand the company’s reach to customers globally, adapting to the changing shopping habits of customers in the digital age.

**Benefits:**

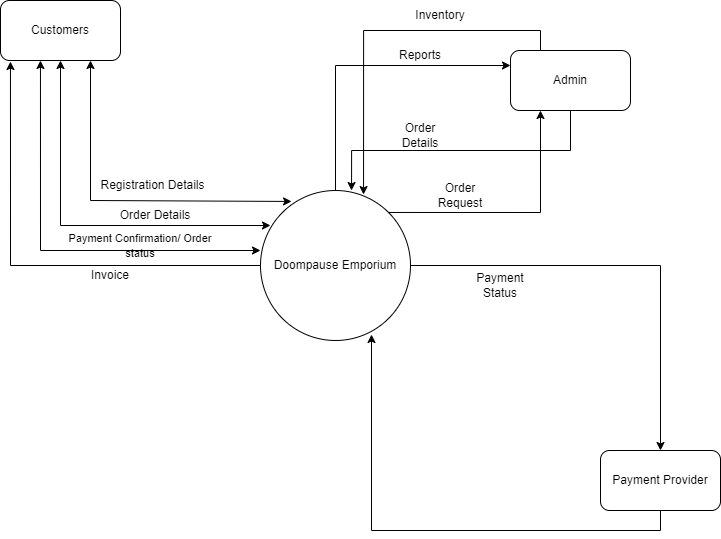
* **Increased Reach:** The web application will allow the company to reach a global audience, increasing its customer base.
* **Improved Customer Experience:** With features like detailed product descriptions, a shopping cart, and secure online transactions, customers will have a convenient and enjoyable shopping experience.
* **Efficient Operations:** The use of Spring Boot and MySQL will ensure efficient and reliable backend operations.

**Objectives and Goals:**

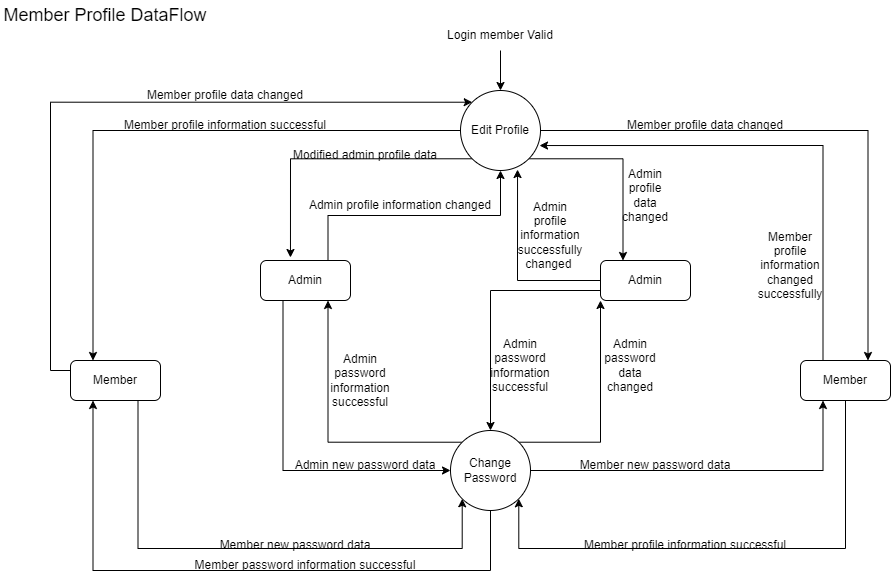
* Develop a user-friendly web application with an intuitive interface.
* Ensure secure online transactions with Spring Boot Security.
* Efficiently manage data with MySQL.
* Launch the web application within the specified timeline and budget.

## Context Diagram/Interface Diagram/Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow

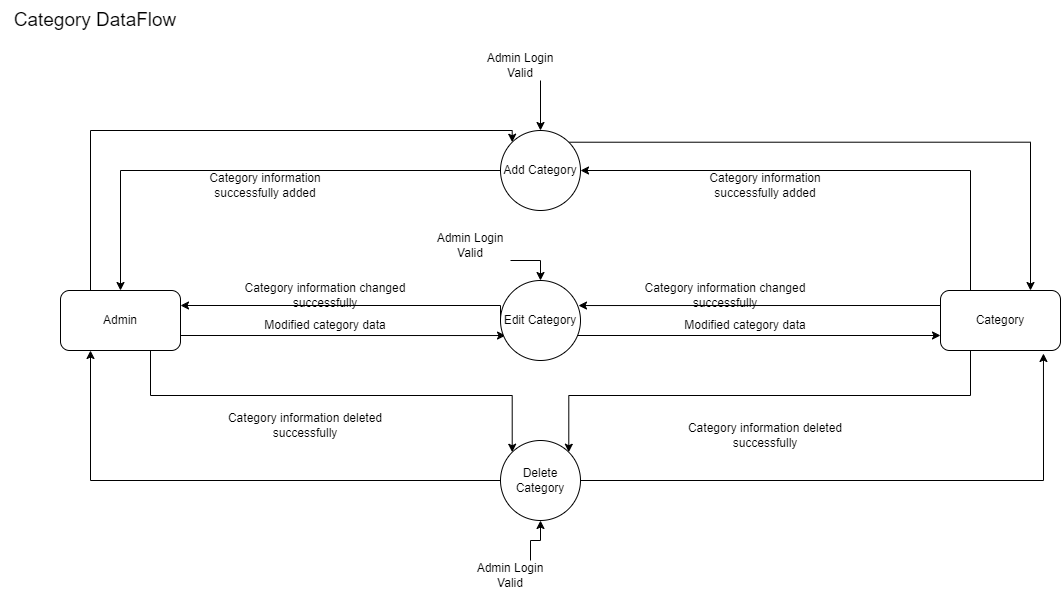
Context Diagram



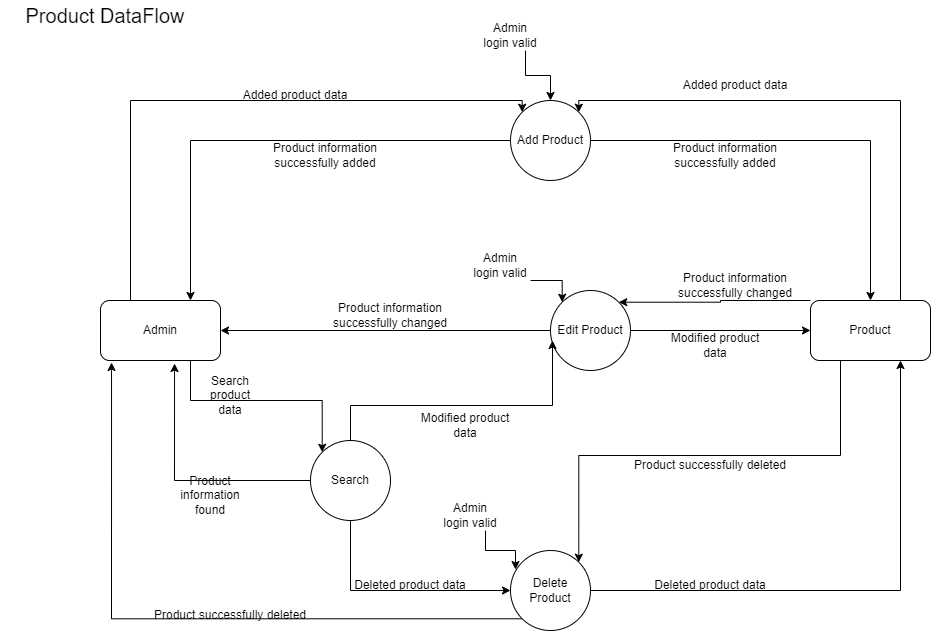
Member Profile DataFlow



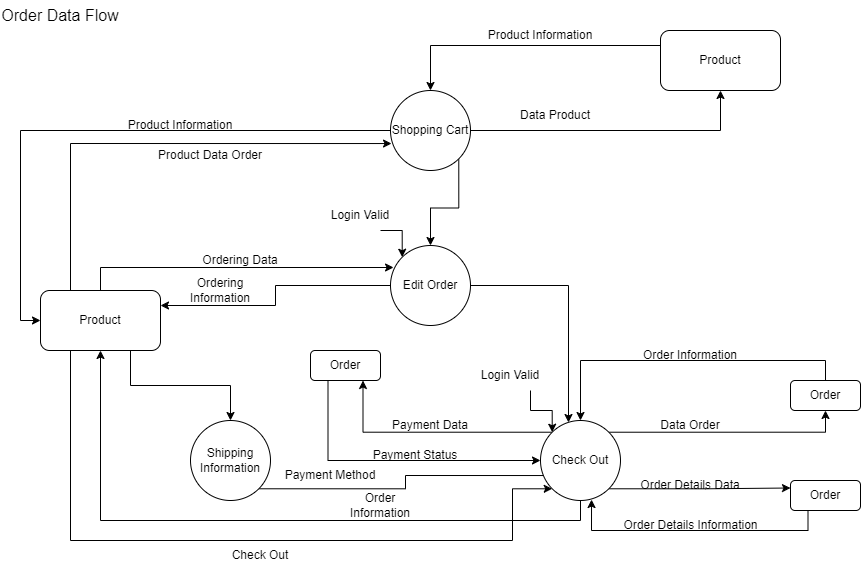
Category DataFlow



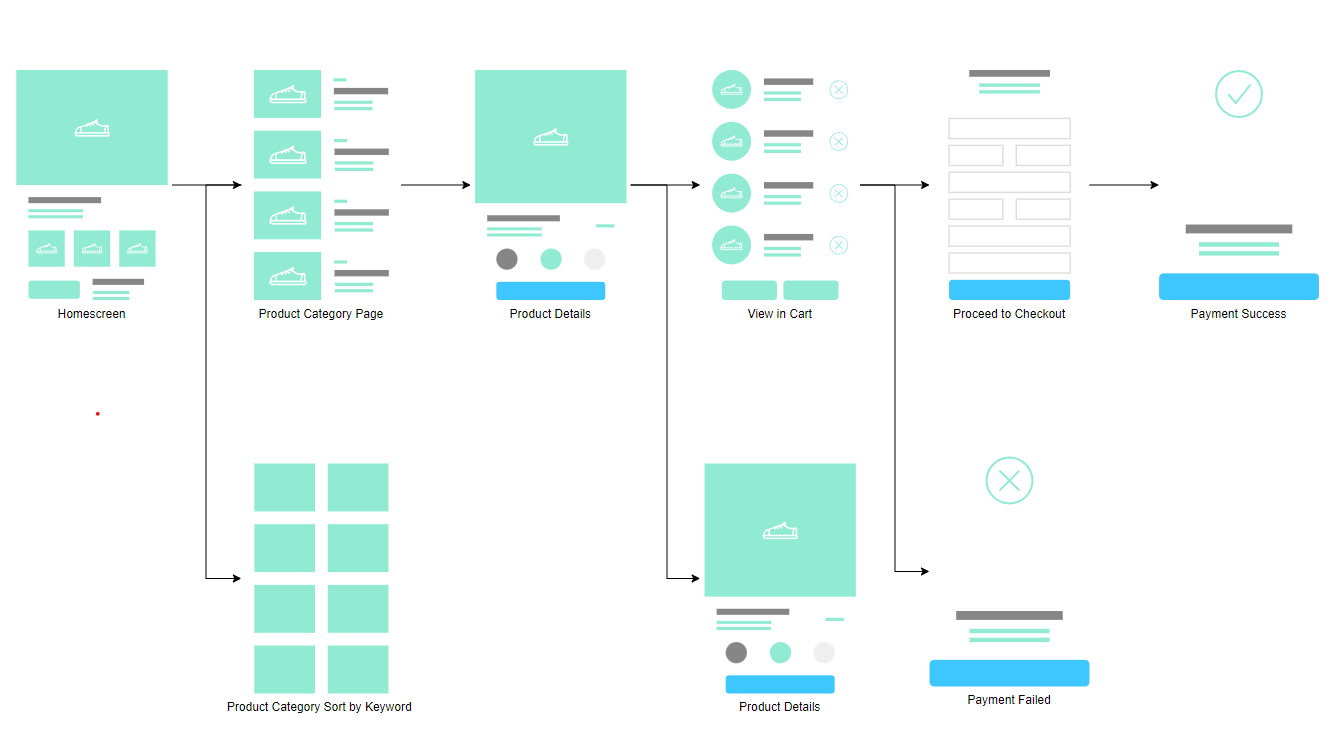
Product DataFlow



Order DataFlow

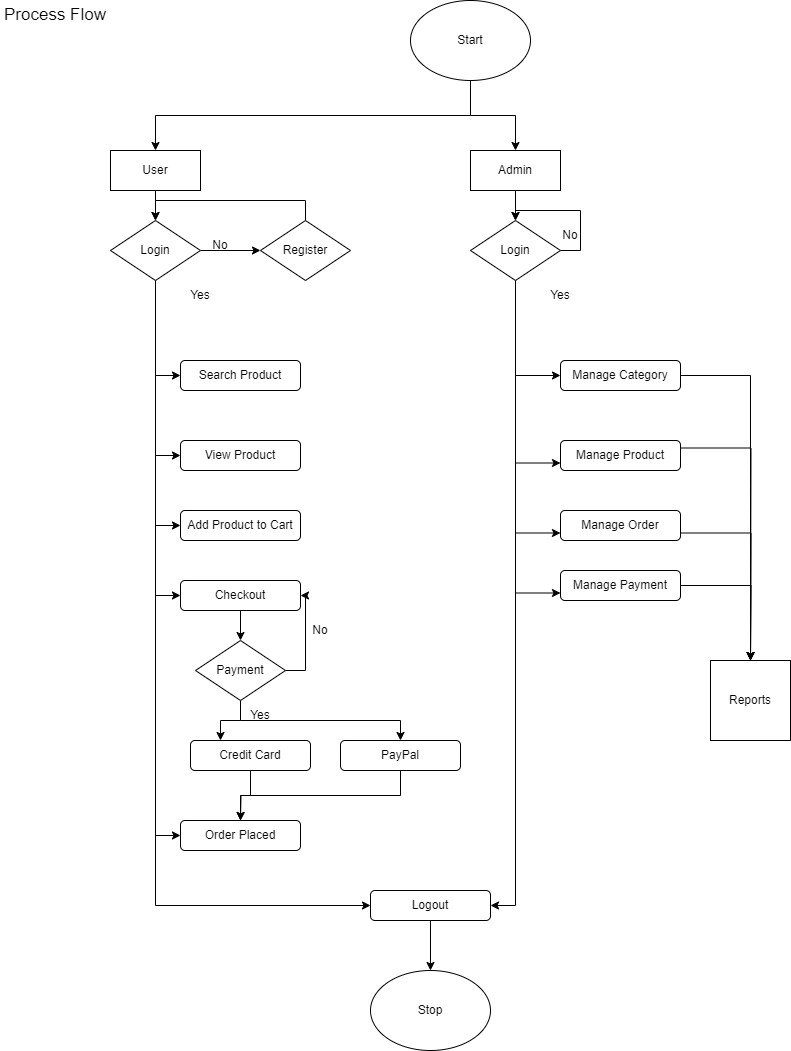


Member Screenflow



Process Flow Diagram

Process Flow Diagram



# Functional Specifications

The overall system will consist of several key functionalities to ensure a seamless and secure online shopping experience for the customers. Here is an index of the functionalities that will be explained in detail in the sections below:

1. User Registration and Login
2. Product Browsing
3. Shopping Cart Management
4. Checkout and Payment Processing
5. Order History and Tracking
6. Customer Support

Each of these functionalities plays a crucial role in the overall system, contributing to a comprehensive and user-friendly online shopping platform.

**User Registration and Login:** This functionality allows users to create an account and log in to the web application. It ensures that each user’s shopping cart and order history are saved and accessible across different sessions.

**Product Browsing:** This functionality enables users to browse through the various products offered by the company. It includes detailed descriptions and pricing information for each item.

**Shopping Cart Management:** This functionality allows users to add, remove, and edit the quantity of products in their shopping cart. It provides users with a flexible and convenient shopping experience.

**Checkout and Payment Processing:** This functionality enables users to check out their shopping cart and make secure online payments. It ensures a smooth and secure transaction process.

**Order History and Tracking:** This functionality allows users to view their past orders and track the status of their current orders. It enhances transparency and allows users to stay updated on their order status.

**Customer Support:** This functionality provides users with assistance regarding any issues or queries they may have. It ensures a positive user experience by addressing user concerns promptly and effectively.

## Architecture

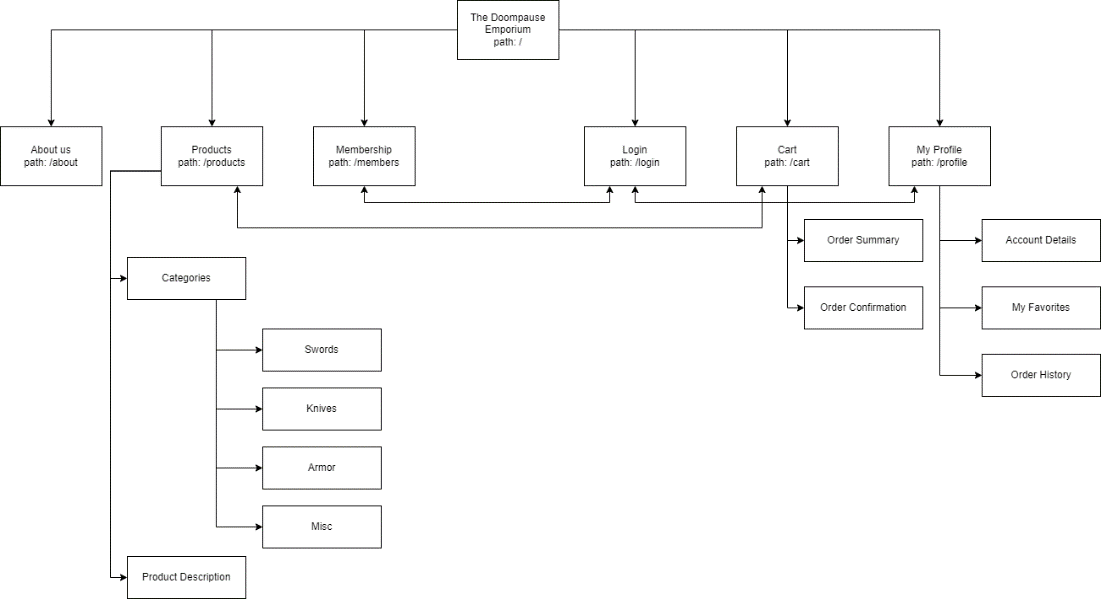
As you can see, the diagram shows four containers: a **user interface** that runs on the user’s computer or tablet, a **frontend** that handles the presentation logic and user interactions using JavaScript, HTML, and CSS, a **backend** that provides the business logic and data access using Java, Spring Boot, MySQL, Thymeleaf, and Node.js, and a **database** that stores the persistent data using MySQL. The arrows indicate the direction of communication between the containers. For example, the user interface communicates with the frontend using HTTP requests and responses, and the frontend communicates with the backend using RESTful APIs. The backend communicates with the database using JDBC. A diagram of a software application

Description automatically generated

## Purpose/Description

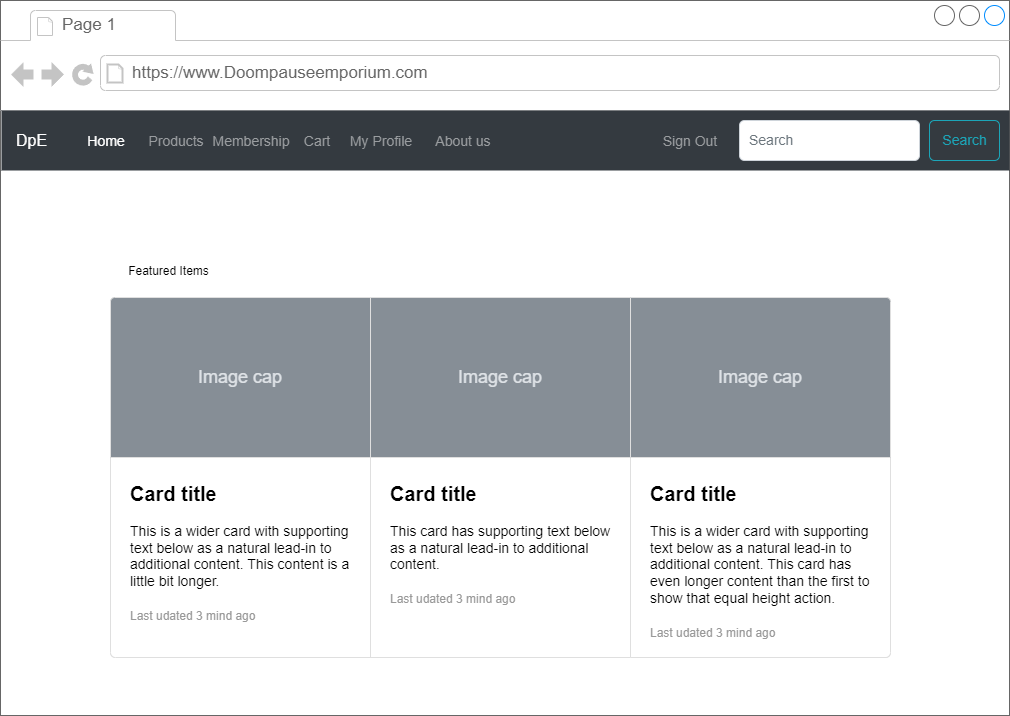
Provide a clear and detailed description of the functions that a system or component must perform in order to meet the client’s objectives. Functional specifications help developers plan and implement each stage of the product development cycle, as well as communicate the design and expectations to other stakeholders. Functional specifications also reduce risks, improve communication, and ensure quality and alignment of the final product.

## Sitemap

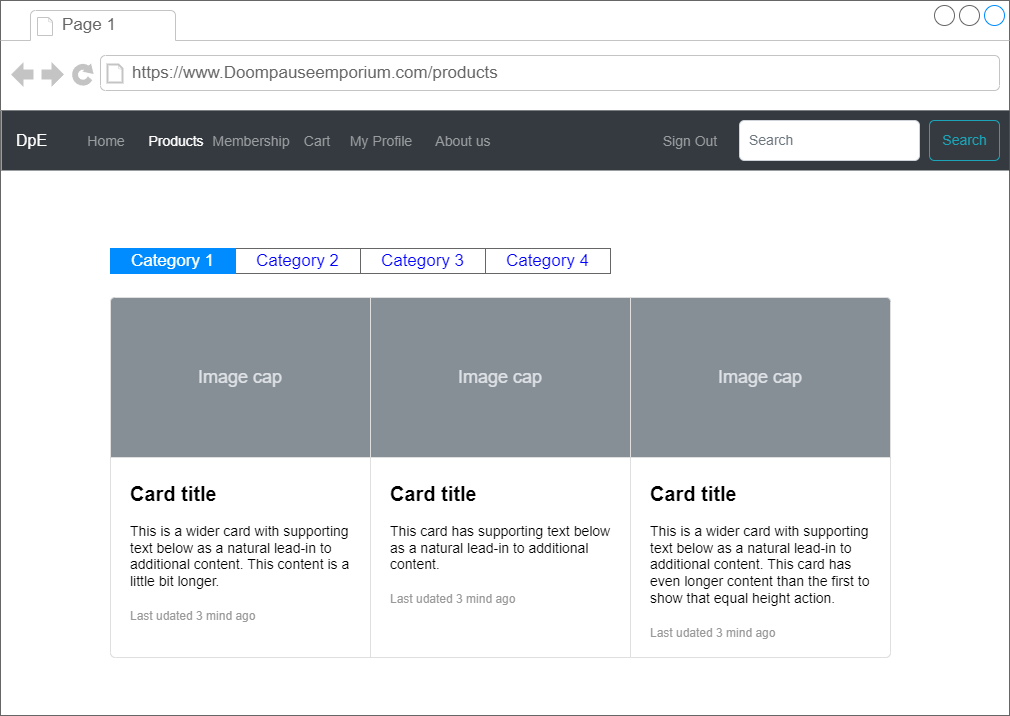


## Wireframes

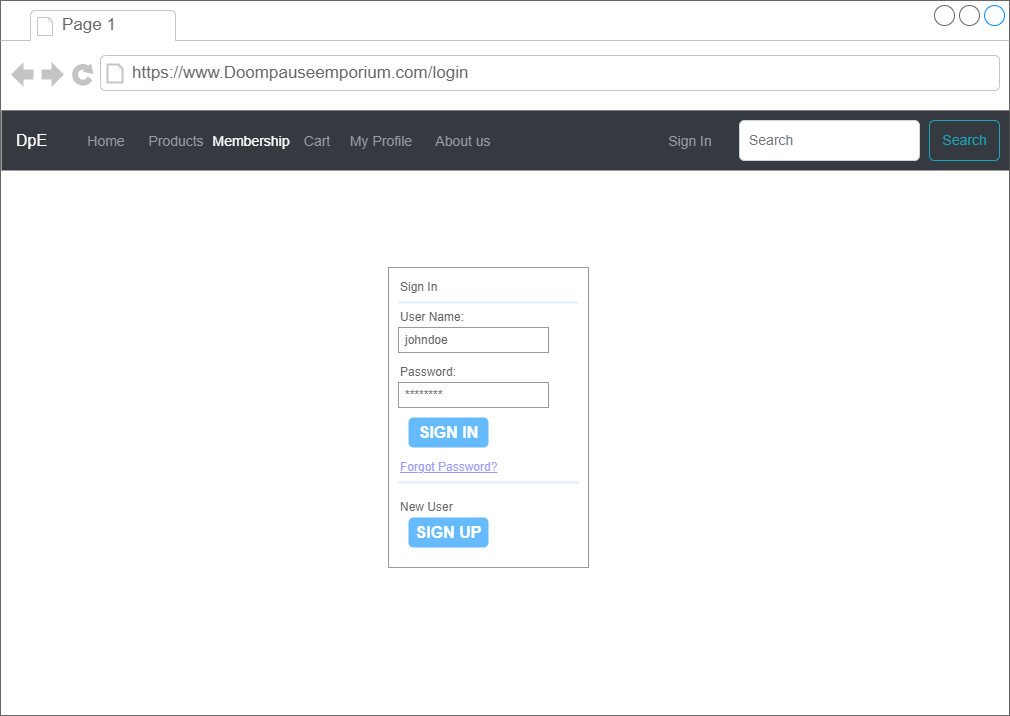
Homepage



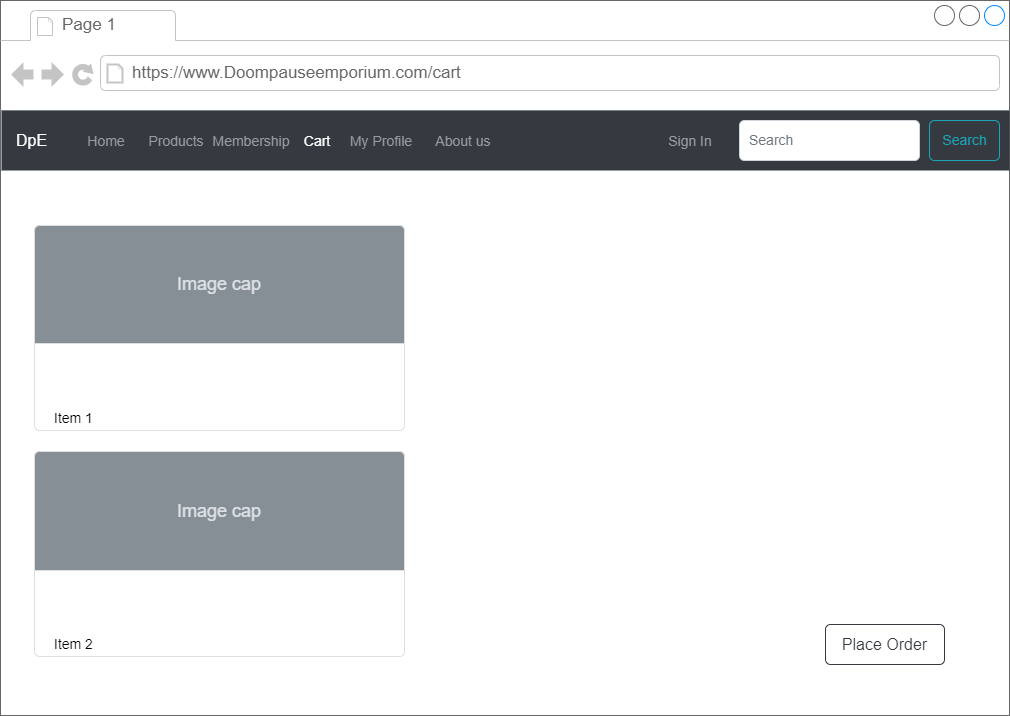
Products Page



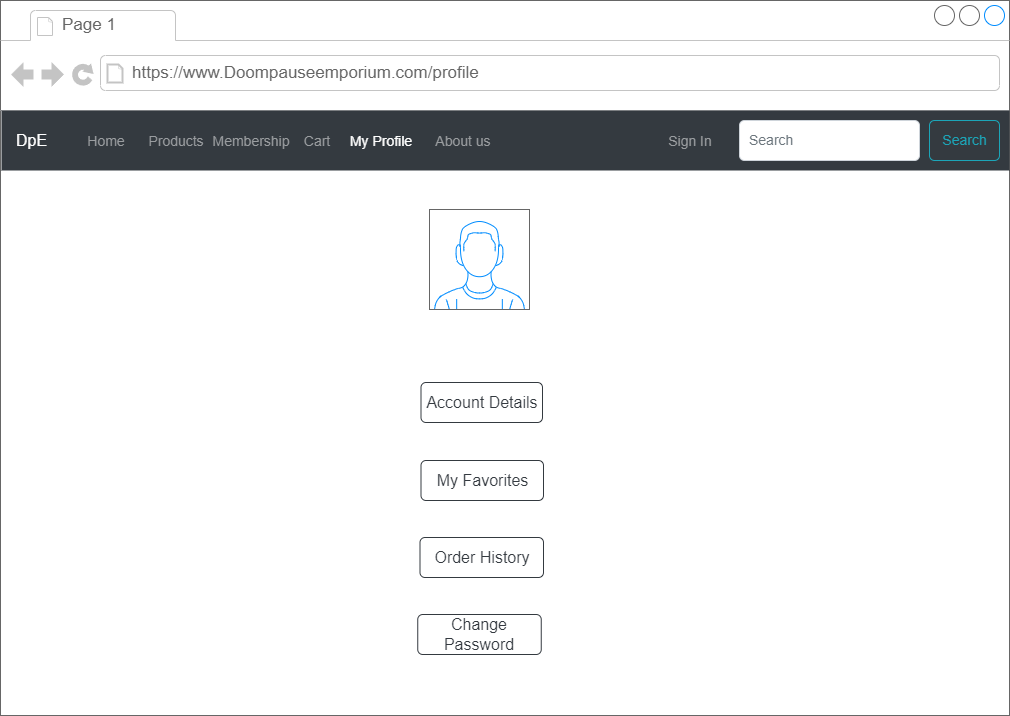
Login Page



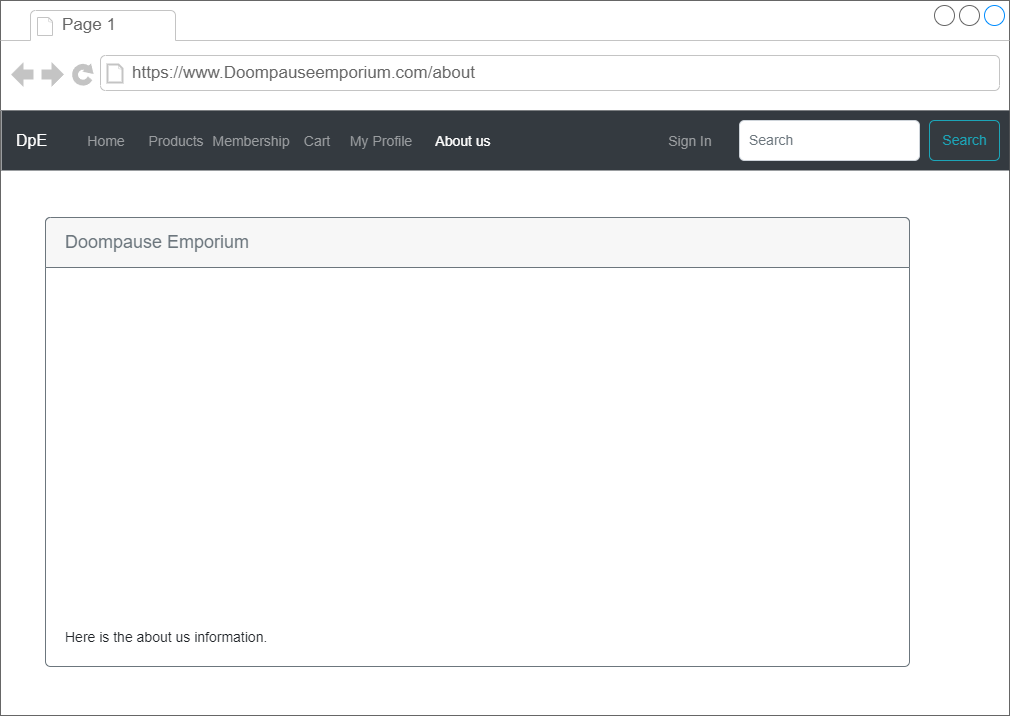
Cart Page



Member Profile Page



About Us Page



## Functional Requirements

| **Use**  **Case ID** | **ID** | **Functional Requirement** |
| --- | --- | --- |
|  | FR-1 | The user should be able to login and logout with their information and the use of clickable buttons. |
|  | FR-2 | The user shall be able to browse the inventory that is available |
|  | FR-3 | The user shall be able to search for a product/s by use of a keyword |
|  | FR-4 | The user shall be able to search through their orders placed by keyword or order number. |
|  | FR-5 | The user shall be able to filter products by way of categories |
|  | FR-6 | The user shall be able to add items to their wish list |
|  | FR-7 | The user shall be able to add items into their cart |
|  | FR-  8 | The user shall be able to check out (place order) from the items in their cart |
|  | FR-  9 | The user shall be able to browse through their orders in a list |
|  | FR-  10 | The user shall be able to view the selected product description/details |
|  | FR-  11 | The user shall be able to view category details |
|  | FR-  12 | The admin shall be able to add products into inventory |
|  | FR-  13 | The admin shall be able to delete products from inventory |
|  | FR-  14 | The admin shall be able to edit products in inventory |
|  | FR-  15 | The admin shall be able to browse all orders placed in a list |
|  | FR-  16 | The admin shall be able to add a new category |
|  | FR-  17 | The admin shall be able to delete a category |
|  | FR-  18 | The admin shall be able to edit a category |
|  |  |  |

## Field-Level Specifications

**Form Elements:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Call-Out** | **Field Label** | **UI Control** | **Mand?** | **Editable** | **Data Type** | **Value Set** | **Default Value** | **Data Example** | **Data Source** |
| User Info | User name | textbox | Yes | Yes | Alpha-numeric | none | NA | agujar | User entry |
| User Info | Password | textbox | Yes | Yes | Alpha-numeric | none | NA | Swampfox | User entry |
| User Info | Email | textbox | Yes | Yes | Alpha-numeric | none | NA | [raj@inbox.com](mailto:raj@inbox.com) | User entry |
| Product Info | Product Id | Text | Yes | No | Alpha-numeric | none | NA | 8211 | database |
| Product Info | Product name | Text | Yes | No | Alpha-numeric | none | NA | Sword | Database |
| Product Info | Product desc. | Text | Yes | No | Alpha-numeric | none | NA | Sharp blade | Database |
| Product Info | Product price | Text | Yes | No | Alpha-numeric | none | 0.00 | 49.99 | Database |
| Product Info | Product qty | Text | Yes | No | Alpha-numeric | none | 0 | 7 | Database |
| Shopping cart | Cart ID | text | Yes | No | Alpha-numeric | None | 0000 | Ae32g | Database |
| Shopping cart | User ID | text | Yes | No | Alpha-numeric | None | NA | Rag84 | Database |
| Shopping cart | Order ID | text | Yes | No | Alpha-numeric | None | NA | 774g890 | Database |
| Shopping cart | Order qty | text | Yes | No | Alpha-numeric | none | 0 | 5 | Database |
| Order Info | Order Status | text | Yes | No | Alpha-numeric | none | Waiting for Order | Shipped | Database |
| Order Info | Order ID | text | Yes | No | Alpha-numeric | None | NA | 774g890 | Database |
| Order Info | User ID | text | Yes | No | Alpha-numeric | None | NA | Rag84 | Database |
| Order Info | Cart ID | text | Yes | No | Alpha-numeric | None | 0000 | Ae32g | Database |

**Form Business Rules and Dependencies:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Label** | **Validation / Business Rules** | **Error Messages** | **Data Dependencies** | **Additional Info/ Notes** |
| *Username* | *Cannot be Void*  *Must be at least 6 characters* | *Username Must Be At Least 8 Characters* | *None* | *String* |
| *Password* | *Cannot Be Void*  *Must be at least 8 characters* | *Username Must Be At Least 8 Characters* | *None* | *String* |
| *Email* | *Cannot Be Void*  *Must include @ symbol followed by valid email domain* | *Email Must Be Valid* | *None* | *String* |
| *Product ID* | *Cannot be void*  *Limited to 4 digits* | *None* | *None* | *Int* |
| *Product Name* | *Cannot be void* | *Please Enter a Product Name* | *None* | *String* |
| *Product Description* | *Cannot be void* | *Please Enter a Product Description* | *None* | *String* |
| *Product Price* | *None* | *None* | *None* | *Double* |
| *Cart ID* | *Cannot be void*  *Limited to 5 digits* | *None* | *None* | *String* |
| *User ID* | *Cannot be void*  *Limited to 5 digits* | *None* | *None* | *String* |
| *Quantity* | *Must be more than 0* | *Quantity Must Be More Than Zero* | *None* | *Int* |
| *Order ID* | *Cannot be void*  *Limited to 7 digits* | *None* | *None* | *String* |
| *Order Status* | *Limited to “pending”, “shipped”, “delivered”, or “canceled”* | *None* | *None* | *String* |

**Buttons, Links, and Icons:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Button, Link, Icon Label** | **OnClick Event** | **Other Event** | **Visible** | **Enabled Vs Disabled** | **Navigate To** | **Validation** | **Dependencies** |
| *logInButton* | *OnClick – Redirected to Login Page* | *Other Events – None* | *Visible When Not Logged In, Not Visible when Logged In* | *Enabled When Not Logged In, Disabled When Logged IN* | *Navigate To - DoompauseEmporium.com/login* | *Validation – Verify Username and Password* | *None* |
| *logOutButton* | *OnClick – Redirected to Logout Page* | *Other Events – None* | *Visible When Logged In, Not Visible when Not Logged In* | *Enabled When Logged In, Not Enabled when Not Logged In* | *DoompauseEmporium.com/* | *None* | *None* |
| *aboutButton* | *Redirected to About Page* | *None* | *Always* | *Always* | *DoompauseEmporium.com/about* | *None* | *None* |
| *productsButton* | *Redirected to Products Page* | *None* | *Always* | *Always* | *DoompauseEmporium.com/products* | *None* | *None* |
| *membersButton* | *Redirects to Login Page if Not Logged in, Shows “You’re already a member” Message if Logged In* | *None* | *Always* | *Always* | *DoompauseEmporium.com/Login (if not logged in) or DoomPauseEmporium.com/members (if logged in)* | *None* | *None* |
| *profileButton* | *Redirects to Profile Page If Logged In, Redirects to Login Page if Not Logged In* | *None* | *Always* | *Always* | *DoompauseEmporium.com/login (if not logged in) or DoompauseEmporium.com/profile (if logged in)* | *None* | *None* |
| *cartButton* | *Redirects to Cart Page* | *None* | *Always* | *Always* | *DoompauseEmporium.com/cart* | *None* | *None* |
| *addToCartButton* | *Adds Item to Cart if Logged In, Redirects to Login Page if Not Logged In* | *None* | *Always* | *Always* | *None if Logged In, Login Page if not Logged In* | *None* | *None* |
| *categoriesButton* | *Expands Categories Menu To Show Categories* | *None* | *Always* | *Always* | *None* | *None* | *None* |
| *searchButton* | *Searches for Item Entered Into Search Bar* | *None* | *Always* | *Always* | *Search Results* | *None* | *None* |
| *categoryButton<categoryID> (For each category)* | *Redirects to search results for specific category* | *None* | *Always* | *Always* | *Search Results* | *None* | *None* |

## Nonfunctional Requirements

<All nonfunctional requirements are detailed here.

Make sure each specification has a reference number that can be traceable back to the user story or use case.

These must be detailed enough so the software developer can design the website.>

| **ID** | **Nonfunctional Requirement** |
| --- | --- |
| NFR-1 | Easy to use interface |
| NFR-2 | Unique yet attractive design |
| NFR-3 | Easy to understand product descriptions |
|  |  |
|  |  |
|  |  |

# References

*Functional Specification Documents: your complete guide*. (n.d.). Copyright © Justinmind 2023. All Rights Reserved. https://www.justinmind.com/blog/functional-specification-documentation-quick-guide-to-making-your-own/

GeeksforGeeks. (2022). Unified Modeling Language UML activity diagrams. *GeeksforGeeks*. https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/

# Open Issues

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue ID** | **Issue** | **Raised By** | **Raised On** | **Solution/ Decision** | **Resolved By** | **Resolved On** | **Status** |
|  |  |  |  |  |  |  |  |

# Appendix