Software Requirements Specification

Product Management Portal

Team 24
Charles Dawkins
Eric Fritts
Maija Kingston
Shawn Stawiarski
Gabriel Udoette

1 Introduction

1.1 Purpose

The purpose of this document is to clearly define and describe the requirements of the product management portal. This document is meant to serve as the source of truth for both the developers and the project sponsor in order to facilitate a clear understanding of the necessary functionality of the project.

1.2 Scope

The proposed software is a product management portal. The main objective of this product is to allow the company to efficiently manage asset and shipment information. Employees of the company will use the software to store, view, and update information about the assets the company is managing, as well as create, receive, and send shipments of these assets to and from customers. All assets are tracked alongside their modification history, any applicable parent assets, and the entities that own or rent them. This information will be used by the company to efficiently manage its physical products and the logistics associated with them.

2 System Features

2.1 Searching

[PMP-1] The software shall allow the user to search for an asset based on its description or its serial number.
[PMP-2] The software shall display a maximum of 3 possible matches under the search bar as the user types their search.

2.2 Asset Management

2.2.1 Asset Viewing

[PMP-3] The software shall provide a way to view all company assets in either a table or list format.
[PMP-4] The user shall be able to filter the asset list based on its activity status, assignment status, type, creation date, update date, and group tag.

2.2.2 Asset Editing

[PMP-5] The software shall provide a means to edit the assignment type, the assignee, the group tag, the owner, and the activity status of an asset.

[PMP-6] The software shall not allow editing of an asset's ID, serial, product type, or description.
[PMP-7] The user shall be able to edit an asset's information either individually or in bulk.
[PMP-8] The software shall keep a log of all editing events occurring to an asset and store it alongside that asset.
[PMP-9] The changes made to an 'Assembly' type asset shall flow down and update the asset's children as well.
[PMP-10] The user shall be able to create a new asset in the system.

2.2.3 Asset Details

[PMP-11] The software shall provide a way for the user to view all information that pertains to a specific asset.
[PMP-12] The user shall be able to see a timeline of the history of the asset, including event dates and descriptions.

2.3 Shipping Management

[PMP-13] The user shall be able to create new shipments of any number of assets under a single identifying contract number.
[PMP-14] The software shall allow the user to send and receive shipments.
[PMP-15] The asset's location and checkout status shall be updated upon its shipment or receival.
[PMP-16] The software shall update all assets within a shipment when any change is made to the shipment.

2.4 User Management

[PMP-17] The software shall store user account information in an encrypted format for use with authentication.
[PMP-18] The user shall be able to edit their associated email and password.
[PMP-19] The authenticated user's session shall persist for 7 days without logging in again.

3 External Interface Requirements

3.1 User Interfaces

[PMP-20] The UI shall be responsive and look consistent across screens sized 375x667, 1024x768, and 1920x1080.
[PMP-21] The UI shall follow Material Design guidelines.
[PMP-22] The UI shall utilize the Material-UI component library for ReactJS.
[PMP-23] The UI shall use Material Design icons for all iconography.

[PMP-24] The UI shall use the Roboto font for all typography.
[PMP-25] The software shall have a sidebar for navigation that appears on every page.
[PMP-26] The sidebar shall include a header that displays the current user's information.
[PMP-27] The sidebar shall contain links to all modules in the software.
[PMP-28] The software shall display a header on every page including the module title and any applicable breadcrumbs.
[PMP-29] The UI's color scheme shall be limited to neutral colors for main content with no limitations for accent colors.

3.2 Hardware Interfaces

3.3 Software Interfaces

[PMP-30] The software's front-end shall be written using v**16.13.1** of ReactJS.
[PMP-31] The software's back-end shall use v**4.17.1** of Express for serving the API.
[PMP-32] The software's back-end shall use v**14.13.0** of NodeJS for running Express and building the API.
[PMP-33] The software's back-end shall use v**4.4.1** of MongoDB Community Server for storing asset and event information.
[PMP-34] The NodeJS API application shall query MongoDB to retrieve information and send it as a response to a client using Express.

3.4 Network Interfaces

[PMP-35] The software shall run on all modern web browsers greater than Internet Explorer 11.
[PMP-36] The software shall use HTTPS to facilitate communication between the client browser and the API server.
[PMP-37] Both the client and the server shall send requests and responses in JSON format.
[PMP-38] The server shall limit API endpoint access only to authenticated users.

4 Other Nonfunctional Requirements

4.1 Performance Requirements

[PMP-39] The software shall sustain a minimum of 50 active users concurrently without page load times exceeding 2 seconds.

4.2 Security Requirements

[PMP-40] The software shall not allow an unauthenticated user to access any page other than the login page.
[PMP-41] All unsuccessful login attempts shall be logged in a text file.

4.3 Software Quality Requirements

[PMP-42] All logical functions on the front-end shall be testing using JEST.
[PMP-43] The NodeJS API shall be tested using a combination of Mocha for the JavaScript testing and Postman for endpoint testing.

4.4 Maintainability Requirements

[PMP-44] The software shall have written documentation for all API endpoints that describes the endpoints type and functionality.
[PMP-45] The software shall be organized into a hierarchical folder structure that is logically grouped.
[PMP-46] Code structure shall be broken up into modular, reusable components wherever possible.

4.5 Usability Requirements

[PMP-47] The UI shall be similar to existing dashboard products and therefore be familiar to the end-users.

5 Other Requirements

[PMP-48] The software shall not depend on any paid 3rd party service providers.

Appendix A: Glossary

Term	Definition
CRM	Customer Relationship Management
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
UML	Unified Modeling Language
PMP	Product Management Portal
REST	REpresentational State Transfer, a popular architectural style for APIs
UI	User Interface
JSON	JavaScript Object Notation, a lightweight data-interchange format
HTTPS	Hypertext Transfer Protocol Secure, an extension of the Hypertext Transfer Protocol (HTTP) used for secure communication