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 user shall be able to search assets on every page based on the page context.
[PMP-3] The software shall display a maximum of 3 possible matches under the search bar as the user types their search.
[PMP-4] The user shall be able to filter out search results based on all attributes of the assets.

2.2 Asset Management

2.2.1 Asset Viewing

[PMP-5] The software shall provide a way to view all company assets in either a table or list format.
[PMP-6] 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-7] The software shall provide a means to edit only the assignment type, the assignee, the group tag, the owner, and the activity status of an asset.
[PMP-8] The software shall not allow editing of an asset's ID, serial, product type, or description.
[PMP-9] The user shall be able to edit an asset's information either individually or in bulk.
[PMP-10] The software shall keep a log of all editing events occurring to an asset.
[PMP-11] The changes made to an 'Assembly' type asset shall flow down and update the asset's children as well.
[PMP-12] The user shall be able to create a new asset in the system.

2.2.3 Asset Details

[PMP-13] The software shall provide a way for the user to view all information that pertains to a specific asset.
[PMP-14] The user shall be able to see a timeline of the history of the asset, including event dates and descriptions.

2.3 Shipping Management

2.3.1 Shipment Creation

[PMP-15] The user shall be able to create new shipments of any number of assets under a single identifying contract number.
[PMP-16] The software shall associate and track a contract number to a shipment.
[PMP-17] The user shall be able to select only staging facilities to ship to when creating an incoming shipment and only staging facilities to ship from when creating an outgoing shipment.
[PMP-18] The user shall be able to add, remove, and edit items in the shipment manifest individually and in bulk.
[PMP-19] The software shall check for errors regarding asset status prior to shipment but allow the user to acknowledge and ship regardless.
[PMP-20] The user shall be able to enter custom text notes to an item in the shipping manifest.
[PMP-21] The user shall be able to ship unserialized assets that are not in the system.
[PMP-22] The software shall automatically mark assets as unavailable for all other users while they are being worked with by a user creating a new shipment.

2.3.2 Shipment Management

[PMP-23] The software shall allow the user to send and receive shipments.
[PMP-24] The asset's location and checkout status shall be updated upon its shipment or receival.
[PMP-25] The software shall update all assets within a shipment when any change is made to the shipment.
[PMP-26] The user shall be able to complete or cancel a shipment.

[PMP-27] The shipment shall only be able to be edited prior to being completed or cancelled.
[PMP-28] The shipment's tracking data shall be used to update its manifest items upon completion.

3 External Interface Requirements

3.1 User Interfaces

[PMP-29] The UI shall be responsive and look consistent across screens sized 375x667, 1024x768, and 1920x1080.
[PMP-30] The UI shall follow Material Design guidelines.
[PMP-31] The UI shall utilize the Material-UI component library for ReactJS.
[PMP-32] The UI shall use Material Design icons for all iconography.
[PMP-33] The UI shall use the Roboto font for all typography.
[PMP-34] A search bar shall be displayed on every page.
[PMP-34] The software shall have a sidebar for navigation that appears on every page.
[PMP-36] The sidebar shall include a header that displays the current user's information.
[PMP-37] The sidebar shall contain links to all modules in the software.
[PMP-38] The software shall display a header on every page including the module title and any applicable breadcrumbs.
[PMP-39] 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-40] The software's front-end shall be written using v**16.13.1** of ReactJS.
[PMP-41] The software's back-end shall use v**4.17.1** of Express for serving the API.
[PMP-42] The software's back-end shall use v**14.13.0** of NodeJS for running Express and building the API.
[PMP-43] The software's back-end shall use v**4.4.1** of MongoDB Community Server for storing asset and event information.
[PMP-44] 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-45] The software shall run on all modern web browsers greater than Internet Explorer 11.
[PMP-46] The software shall use HTTPS to facilitate communication between the client browser and the API server.
[PMP-47] Both the client and the server shall send requests and responses in JSON format.

4 Other Nonfunctional Requirements

4.1 Performance Requirements

[PMP-48] The software shall sustain a minimum of 50 active users concurrently without page load times exceeding 2 seconds.

4.2 Security Requirements

[PMP-49] The software shall perform basic dummy authentication to limit page access to authenticated users.

4.3 Software Quality Requirements

[PMP-50] The front-end shall be tested using JEST.
[PMP-51] The NodeJS API shall be tested using a combination of Mocha for JavaScript testing and Postman for endpoint testing.

4.4 Maintainability Requirements

[PMP-52] The software shall have documentation for all API endpoints written using OpenAPI that describes the endpoints type, purpose, expected request and query format, and possible responses.
[PMP-53] The software shall be organized into a hierarchical folder structure that is logically grouped.
[PMP-54] Code structure shall be broken up into modular, reusable components wherever possible.

4.5 Usability Requirements

[PMP-55] The UI shall be similar to existing dashboard products such as OneDrive and CRM tools like Salesforce and therefore be familiar to the end-users.

5 Other Requirements

[PMP-56] The software shall depend only on free 3rd party service providers when dependencies are necessary.

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