Software Requirements Specification

Asset Management System

Team 10

Table of Contents

1. Introduction

1.1 Purpose

This is an Asset Management system created for a non-profit organization. The non-profit organization currently uses an out-of-date legacy asset management system that needs a custom AMS after being unable to find one on the market that fits their needs.

1.2 Scope

The asset management system will provide the nonprofit organization the ability to sort, store and search their assets. One of the current problems facing the nonprofit is that there are no existing AMSs that work for them. This AMS will allow approved users the ability to manage their assets and access them in an easy to use, efficient manner. Other functions provided by the AMS include reporting basic information on assets as well as generating printable labels.

1.3 Definitions, Acronyms, and Abbreviations.

AMS - Asset Management system
LDAP - Lightweight Directory Access Protocol
Fatal Error - Error that causes the website to entirely not work.
Retire - When an item is no longer active or being used, synonymous with delete but instead of deleting forever the item will remain in the database and labeled as retired.
QR Code - Quick Response code that acts as a bar code and stores data that when scanned directs users to some site or returns some data.

1.4 Overview

The remainder of this document will cover the overall description of the AMS as well as specific requirements and our appendix. Section 2 deals with the general functionality of the AMS. Section 3 provides specific design features.

2. The Overall Description

The AMS is an update to a legacy inventory system. The AMS tracks types of inventory, location, age. It must be web-based.

2.1 Product Perspective

This asset management system is self-contained and is not necessarily a component of a larger system. It can be incorporated into larger systems in order to provide users a more fluid experience.

The AMS is being designed for a not-for-profit. It is commissioned as a management tool to better track inventory information, especially as related to the company's information technology. The AMS was commissioned because available products were either found to be unsatisfactory for company needs, or were overpriced.

2.1.1 System Interfaces

The initial AMS will not have any need for system interfaces however we plan to expand the AMS in later phases. Later phases will interface with an existing website called "Pegboard" which is an employee in out board and ticket system. The pegboard allows users to check status of all employees as well as submit service tickets to specific IT users. The AMS will start as a stand alone site but eventually be integrated with the Pegboard and the Pegboard's authentication methods.

2.1.2 Interfaces

In future phases our application will be added to a pre-existing website and the design will keep with current design standards.

2.1.3 Hardware Interfaces

The AMS has no hardware interfaces.

2.1.4 Software Interfaces

2.1.4.1 Bootstrap 4. The AMS must use Bootstrap 4 (alpha.6) to design its front-end component. Bootstrap will provide many CSS elements to use in the interface. We will obtain Bootstrap from https://v4-alpha.getbootstrap.com/getting-started/download/.

2.1.4.2 jQuery 3.1.1. The AMS will benefit from using Jquery for front-end convenience. https://jquery.com/

2.1.4.3 Morris JS 0.5.1. The AMS must use Morris JS to display inventory. This will be used in conjunction with DataTables. http://morrisjs.github.io/morris.js/

2.1.4.4 DataTables 1.10.13. This plugin for jQuery will be used in conjunction with Morris JS. Obtained from https://datatables.net/

2.1.4.5 bcrypt 1.1. The AMS must use bcrypt for security and authentication. Obtained from http://bcrypt.sourceforge.net/

2.1.4.6 MySQL 5.7.16. The AMS must use MySQL to store and serve inventory data. This will be accessed and updated through the website. Obtained from https://downloads.mysql.com/archives/community/.

2.1.5 Communications Interfaces

Our AMS will use HTTP to communicate over the internet communication over TCP/IP protocol.

2.1.6 Operations

Operations occur solely through interacting with a web interface.

2.1.7 Site Adaptation Requirements

There are not any site adaptation requirements for the AMS.

2.2 Product Functions

The general functions of the AMS is to provide a method of tracking assets across an organization or an individual's possession. AMS will provide basic functionality which includes adding, editing and deleting assets.

2.3 User Characteristics

The main users will be IT employees. These user will almost certainly have a higher level of technical experience than the average user of most websites. The AMS will still be presented in a user friendly manner oriented for use from any person regardless of technical experience though we anticipate all users to have a fairly advanced technical background.

2.4 Constraints

Most of the libraries used may not have a compatibility with Internet Explorer.

2.5 Assumptions and Dependencies

This AMS relies on the user having an internet connection and a web page browser.

2.6 Apportioning of Requirements.

2.6.1 First Release
- 2.6.1.1 Add, edit and retire assets
- 2.6.1.2 Basic user heirarchy with authenication

2.6.2 Second Release - Scrum
- 2.6.2.3 Add printable labels for each asset
- 2.6.2.4 Add reporting based off various fields

2.6.3 Subsequent Release
- 2.6.3.1 Authenticate using LDAP (active directory)
- 2.6.3.2 Integration with Pegboard Site

3. Specific Requirements

3.1 External Interfaces

The AMS does not use any external interfaces.

3.2 Functions

3.2.1 Admin Users

3.2.1.1 An admin user has the most authority with the involvement of assets and their creation, deletion, and modification. Admin users will stem off of the current site that the AMS will be integrated with and as such will be admin level users of that site.

3.2.1.2 An admin user shall have unlimited access to modify asset data and manage user base.

3.2.1.3 Additional Functional Requirement

3.2.1.3.1 User management

3.2.1.3.1.1 Admin users may add a username and password

3.2.1.3.1.2 Admin users may distribute user credentials to general users

3.2.1.3.1.3 Admin users may modify or delete user credentials

3.2.1.3.2 Update Asset Data

3.2.1.3.2.1 An admin user shall modify existing asset data for an asset or set of assets

3.2.1.3.2.2 An admin user can modify the description, office location, primary user of the asset, and retirement status.

3.2.1.3.3 Manually Add Assets

3.2.1.3.3.1 An admin user shall add assets to the database

3.2.1.3.3.2 Admin user shall manually specify asset details to be entered into database.

3.2.1.3.3.3 Additional Functional Requirements

3.2.1.3.3.3.1 An admin user may specify type of asset, purchase date, purchase value, primary user, office location, kll model, and a text description of the physical qualities of the asset.

3.2.1.3.4 Synchronize Asset data

3.2.1.3.4.1 Assets shall be synchronized with existing LDAP system.

3.2.1.3.4.2 Requirement for future iterations

3.2.2 General Users

3.2.2.1 General users will be lower level user admin accounts from the main site that will serve as general users for this page of the site. They still have elevated permissions as opposed to most users as normal users of the Pegboard site will have no authority to even view the asset side of the site.

3.2.2.2 A general user can view records and make no modifications.

3.2.2.3 A user can navigate to the inventory and view information about items.

3.2.2.4 Additional Functional Requirements

3.2.2.4.1 General user actions are limited to Querying data [3.2.3]

3.2.3 Query Data

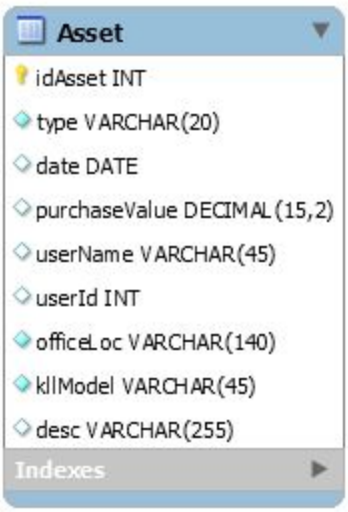
3.2.3.1 A user shall be able to access data regarding the assets

3.2.3.2 A user may choose parameters by which sets of data is viewed

3.2.4 Printable labels

3.2.4.1 A user shall be able to print an asset-specific label

3.2.4.2 Labels will have a generated QR code as well as asset names and company logo. This feature will come at a later phase of the site and will not be built into the original site.



3.3 Performance Requirements

The AMS does not have performance requirements.

3.4 Logical Database Requirements

The AMS will communicate with a database in the first iteration. The database will be responsible for storing the assets as well as users. In

later iterations we will add use for LDAP for authentication and querying the active directory server.

Fig shows asset model to be stored in database

3.5 Design Constraints

There are no design constraints for the AMS.

3.6 Software System Attributes

3.6.1 Reliability

The AMS will be reliable enough to add, edit and retire items without encountering fatal errors.

3.6.2 Availability

This application is available via the web, and must run 24/7.

3.6.3 Security

The AMS will utilize bcrypt to encrypt information such as passwords. The website itself will run SSL to secure traffic and ensure communication between the server and client is safe.

3.6.4 Maintainability

The AMS will be modular so that we can work on certain modules at a time to without affecting the rest of the site.

3.6.5 Portability

The AMS will be very portable so that if it needs to be moved to a new host or new machine it can be done quickly and without much down time.