SEO Inventory System

Group 15

Cale Harms, Chris Larsen,

Jaehyeon Yang, Josiah Ngu,

and Thiam Kiat Sim

Software Requirements Specification Document

Version : 1.1 **Date:02/09/2018**

Date	Changes	Version
2/4/2018	Created Initial Document	1.0
2/9/2018	Update Initial Document using checklist	1.1

Table of Contents

1. Introduction

1.1. Purpose

1.1.1. The purpose of this document is to provide an overview and explain the functionality requirements for the SEO Inventory web software.

1.2. Scope

1.2.1. The intent of the current inventory system design is to allow users access to a functional, web-based and customizable inventory that provides simplicity and efficiency. This software will be used to organize inventory in a non-business specific way, thus allowing its use to span many domains. Users will be able to log inventory as well as manage current inventory through the software. Organizational tools such as categories and subcategories will be implemented to provide users with an inventory software that is easily customizable.

1.3. Definitions, acronyms, and abbreviations

1.3.1. SEO - Simplistic, Efficient, Organized

1.3.2. GUI - Graphical User Interface

1.3.3. Web-based Interface - Interaction between a user and software running on a Web server

1.3.4. HTML - Hypertext Markup Language - a language for creating web pages and web applications.

1.3.5. CSS - Cascading Style Sheets - a language used for describing the presentation of a document written in a markup language, such as HTML.

1.3.6. JavaScript - A high-level programming language.It is used to make webpages interactive and provide online programs, including video games.

1.3.7. Node.js - a JavaScript runtime environment for executing JavaScript code server-side.

1.3.8. MySQL - a relational database management system. SQL stands for Structured Query Language.

1.4. Overview

1.4.1. The rest of this document will go into the general design and feature requirements of the system. An overall description will be given followed by the specific requirements of the software.

2. Overall Description

2.1. Product Perspective

2.1.1. System Interface

2.1.1.1. The system interface will be used to connect the web-based interface to the database, which will be created using MySQL that stores and organizes the data. The system interface will be created with Javascript and Node.js to connect the database to the webpage.

2.1.2. User Interface

2.1.2.1. The user interface will consist of organizational tools that are featured purely in a web-based format. Account support as well as intuitive user interaction options will provide users maximum functionality. The user interface will be constructed with HTML/CSS.

2.1.3. Communications Interface

2.1.3.1. The system will provide direct communication between the account of the user and the database that stores information.

2.2. Product functions

2.2.1. This product's function is to serve as a web-based inventory system that can be used to store and manage a wide variety of inventory types.

2.3. User characteristics

2.3.1. The users of the system will be someone with a need for an inventory-like system. They will want an elegant solution. They will not want to or will not be able to hire someone to create a custom inventory

system for themselves, or purchase expensive software. The likely users will be a small business owner, or someone who wants to organize personal items, such as a collection.

2.4. Constraints

2.4.1. As this is a web-based application a reliable internet connection will be necessary. Functional support will not feature offline modes thus providing constraints if internet is not available.

2.5. Assumptions and dependencies

2.5.1. Besides reliable internet connection, a modern web browser will also be a dependency. The software is expected to run on top web browsers without any issue.

3. Specific Requirements

3.1. External interface requirements

We are using our cse accounts to host the MySQL database. In order that is continually running. We will be using JavaScript and Node.js to connect to this database remotely. The HTML/CSS page will interact with JavaScript to then connect to the database in order to dynamically show data from the database on the webpage.

3.2. System Features

3.2.1. Phase 1

3.2.1.1 Account

3.2.1.1.1 The user will create an account by providing their email address, username and password

3.2.1.1.2 The user can sign in to the system with their username and password and access their inventory.

3.2.1.1.3 Functional requirements

[3.2.1.1.3.1] Sign up function for user to sign up

[3.2.1.1.3.2] System check whether the user has fill up all the basic information that is necessary to create an account.

[3.2.1.1.3.3] Notify the user via email or text if the sign up was successful.

3.2.1.2 GUI

3.2.1.2.1 A basic GUI so that user can sign into their account, access inventory, and account setting.

3.2.1.2.2 After signing in, user can navigate to inventory, to access and manage their inventory, or account setting page to change their username, password,etc.

3.2.1.2.3 Functional requirements

[3.2.1.2.3.1] Account setting page for user to change their information.

[3.2.1.2.3.2] Sign in page for user to log in.

3.2.1.3 Inventory

3.2.1.3.1 Users will create their own custom inventory to track their products.

3.2.1.3.2 For each item, the user will provide general information, such as item's name, quantity, threshold value, date of creation and optionally, the price.

3.2.1.3.3 Functional requirements

[3.2.1.3.3.1] Inventory page

[3.2.1.3.3.2] Enter basic info about the item

[3.2.1.3.3.3] Function to create categories

[3.2.1.3.3.4] Multiple inventories with respective pages.

[3.2.2.1.3.5] Categories can have any number of subcategories, as well as the subcategories themselves.

3.2.2 Phase 2

3.2.2.1 Notification

3.2.2.1.1 The user can choose to get notifications for various events

3.2.2.1.2 User will either get an email or a notification directly from the web app about running low on items, adding or subtracting items from the inventory, etc.

3.2.2.1.3 Functional requirements

[3.2.2.1.3.1] Options to choose how the user wants to receive notifications

[3.2.2.1.3.2] Options to choose which events the user want to get notification.

3.2.2.2 Updated GUI

3.2.2.2.1 The website will be more presentable and easy-to-use.

3.2.2.2.2 Tser will enter the system and understand

3.2.2.1.3 Functional requirements

[3.2.2.1.3.1] Better look.

[3.2.2.1.3.2] Easy-to-use style.

[3.2.2.1.3.3] Multiple ways to view inventory, such as list and grid views.

3.2.2.3 Updated Account

3.2.2.3.1 User will have the ability to reset their password

3.2.2.3.2 Users click ‘forgot password’ button which sends them a link to a page where they can change their password.

3.2.2.3.3 Functional requirements

[3.2.2.3.3.1] forgot password button

[3.2.2.3.3.2] send email.

3.3. Performance requirements

Any response provided from the system should be given within the range of 0 to 1 second.

3.4. Logical database requirements

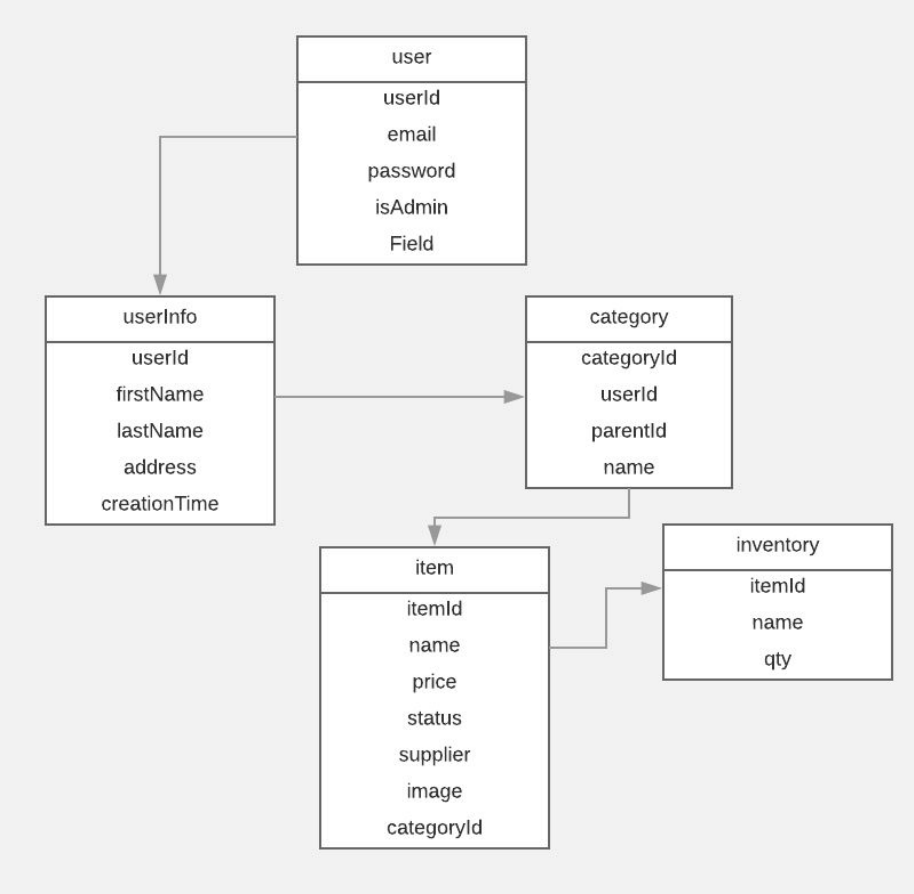

Figure 1: diagram representing database design

3.5. Design constraints

The web system was designed to be displayed on personal computers, mobile device with smaller screen might have difficulty viewing.

3.6. Software system attributes

3.6.1 Reliability

The system should be reliable enough to provide and update customers information without any bugs or error.

3.6.2 Availability

The web system should be online for most of the time with minimal downtime.

3.6.3 Security

Users will need to type in the correct password to log in to the account, if incorrect passwords were given more than 5 times within 10 minute, the account will be temporarily locked and an email will be sent to the user's account. Data of one customer can only access by him/herself. No data is shared between multiple users.

3.6.4 Maintainability

Core functionality and the user interface will be implemented separately. Advance functionality could be added without breaking both user interface and website functionality.

3.6.5 Portability

The system will be written in popular languages and therefore supported by majority of systems including mobile devices.

4. Appendix