

# **SEO Inventory System**

Group 15

Cale Harms, Chris Larsen,  
Jaehyeon Yang, Josiah Ngu,  
and Thiam Kiat Sim

## **Design Document**

**Version : 1.0**

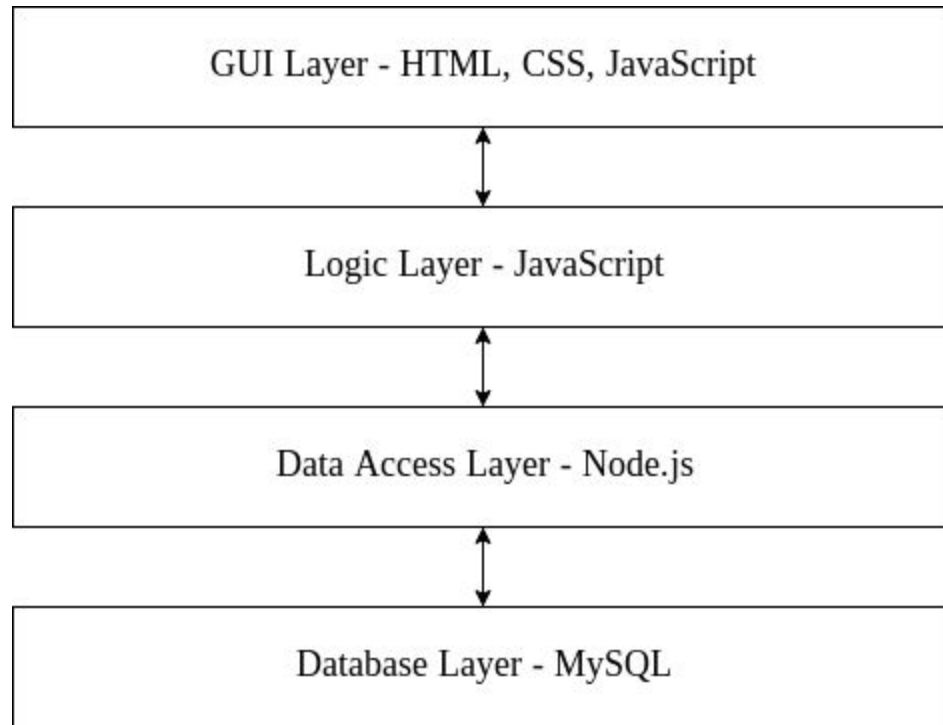
**Date:02/21/2018**

| Date      | Changes                  | Version |
|-----------|--------------------------|---------|
| 2/21/2018 | Created Initial Document | 1.0     |
| 2/23/2018 | Update Document          | 1.1     |

## Table of Contents

|        |                          |  |
|--------|--------------------------|--|
| 1.     | Introduction.....        |  |
| 2.     | Architecture .....       |  |
| 2.1.   | Introduction .....       |  |
| 2.2.   | Modules .....            |  |
| 2.2.1. | Database Layer .....     |  |
| 2.2.2. | Data Access Layer .....  |  |
| 2.2.3. | Logic Layer .....        |  |
| 2.2.4. | GUI Layer .....          |  |
| 3.     | Class Diagrams .....     |  |
| 3.1.   | Data Table Classes ..... |  |
| 3.1.1. | Schema .....             |  |
| 3.1.2. | Schema information ..... |  |
| 3.2.   | Class Information .....  |  |
| 3.3.   | GUI Layer .....          |  |

## 1. Introduction



### 1.1.

The system will consist of 4 layers, GUI layer, logic layer, data access layer and database layer. The Database Layer will store all of the information for the user including their account and their inventory information. The Database will be written using MySQL. The Logic layer will be written in JavaScript. This layer will handle useful the functions, classes, and objects of the system. Lastly, the GUI layer will be written in various languages including HTML, CSS and JavaScript. It will allow users of the SEO inventory system to interact with the system.

**1.2.** The purpose of this design is to lay out the high level architecture and database relationships for the SEO Inventory System. This document will firstly go over architecture followed by class diagrams, and then the GUI.

## 2. Architecture

### 2.1. Introduction

2.1.1. The graphical user interface (GUI) will be done in HTML5 and JavaScript. The logic layer will be done in JavaScript. The data access layer will be done using Node.js. Finally the database layer will be done using MySQL databases. By using a layered model it will allow the software to provide be efficiently created and maintained throughout the product's life cycle.

### 2.2. Modules

**2.2.1. Database Layer :** The database layer will be responsible for storing and managing all data for the inventory system. Since the inventory system is based on good organization, this layer will be fully optimized to cover all necessary features and storable information. By using MySQL it allows for a reliable database storage on the UNL CSE servers which all members have access to.

**2.2.2. Data Access Layer :** The database access layer will be the connection between the physical MySQL databases and the programs code. We will use Node.js to connect and query database edits and searches.

**2.2.3. Logic Layer :** The Logic layer and business level logic will be handled by the various JavaScript functions, classes, and objects.

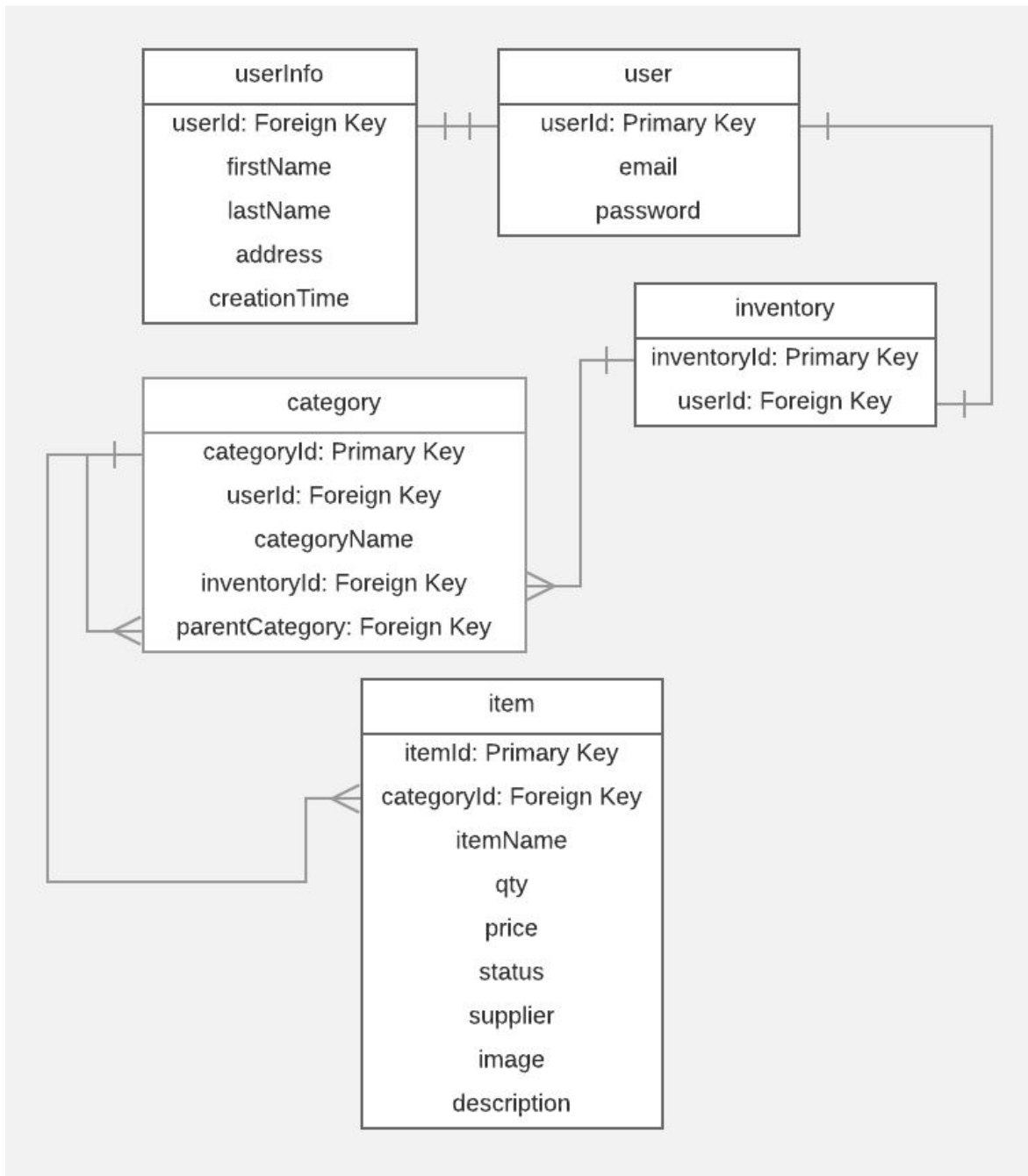
**2.2.4. GUI Layer :** The GUI layer will be the access point for all front end interactions with the software. Here the users of the system will be able to access their inventories as well as add new items to the inventories. By using Javascript and HTML5 the software will provide users with intuitive interfaces and efficient usage from the user's point of view. All functionality of the software will start here.

## 3. Class Diagrams

### 3.1. Data Table Classes

The system will use MySQL database to hold the data. This includes all the relevant user, item and inventory description. The figure below represents the database design and the relationship between each table.

### 3.1.1. Schema



### 3.1.2. Schema Information

**user:** This table stores user credentials. It contains information for the user's email, password, and whether the user is a customer or admin. Their email is used for their login name.

**userInfo:** This table stores relevant user information such as names, address, and account creation time.

**inventory:** This table will help join all of the categories in the user's inventory system.

**category:** This table categorized the items of users. One user can have many categories of items. There can be subcategories (such as an 'iPhone' belonging to the 'smartphone' category, and 'smartphone' belonging to the 'electronics' category). In this case, the table refers to category it belongs to through the category Id.

**item:** This table stores all the relevant information of an item such as item name, price, status, suppliers, image of the item, and item description. Each item that has the same category will be linked to the same category table.

## 3.2. Class Information

Most of the classes described in section 3.1 for the data table classes will be implemented using JavaScript. The information for these classes will be accessed from the MySQL Database using Node.js and then passed into the JavaScript classes to then be displayed via the GUI using HTML5 and CSS.

## 3.3. GUI Layer

The GUI Layer of the system is consists of login screen , sign up page and the homepage. The sign up page will prompt the user to enter information that is needed to create an account. The login screen will take the username and password of the user and check the data against the database. If the entered information is invalid, an error message

will be display to notify the user. When the user successfully login, his or her homepage page will be presented on screen.

The homepage have many different functions. It's main function is to show the user inventory with all the items info, this is considered the 'Inventory Section' of the homepage. The user has the ability to view his or her inventory in list or grid view. There is also a notification button where the user can see all of the notifications regarding his inventory. By navigating to their account settings page, the user has the ability to change his or her information such as password, username, etc..

