Software Requirements Specification

for

UPOD - Back End

**Version 0.3**

**Prepared by Clive Shen**

**Wilfrid Laurier University, CP317**

**May 31, 2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Product Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 User Documentation

2.7 Assumptions and Dependencies

**3. External Interface Requirements**

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

3.4 Communications Interfaces

**4. System Features**

4.1 System Feature 1

4.2 System Feature 2 (and so on)

**5. Other Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

5.5 Business Rules

**6. Other Requirements**

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Clive Shen | 31/05/16 | Update product perspective and functions, as well as actors and use cases. | v0.3 |
| Clive Shen | 17/05/16 | Description and functional requirements of system features related to the back end | v0.2 |
| Tom West | 09/05/16 | Initial Draft | v0.1 |

# Introduction

## Purpose

This document outlines the requirements for the database, functionalities provided by the server-side scripting, and user management portal.

# Overall Description

## Product Perspective

UPOD’s back end implementation needs the collaboration of both front end and back end teams, as the functionalities provided by the database and scripting rely on the front-end design.

## Product Functions

Having a well-organized database and an efficient search engine help users to find the specific information they need in the shortest amount of time. By entering certain keywords, articles related to those keywords should be shown. With the use of server-side scripting, administrators of UPOD can easily modify the contents of the webpages.

## Actors and Use Cases

2.3.1 Actors

In UPOD, there are 3 types of actors.

* *Administrator*

An administrator can login, logout, reset password, create an account for a moderator, update a moderator’s account status, alter permissions of moderators, list administrators, list moderators, add/delete/modify categories, and add/delete/modify pages.

* *Moderator*

A moderator can login, logout, reset password, add/delete/modify categories, and add/delete/modify pages.

* *User*

A user can view categories, view pages, search for categories/pages, and interact with graphs.

2.3.2 Use Cases

The following is a list of use cases that are available for actors.

* **Login**

In the User Management Portal, in order to authenticate his/her identity, an administrator or a moderator needs to enter his/her username and password.

* **Logout**

To terminate his/her process, an administrator or a moderator can click the Logout option.

* **Reset Password**

By inputting the email address that is used to register the account, an email is sent with a link to the Reset Password Page where an administrator or a moderator can reset his/her password.

* **Create an Account for a Moderator**

By inputting user name, email address, and permissions, an administrator can create an unverified account for a moderator. Then, an email is sent to the given email address with a link to the Account Registration page, which allows the moderator to verify his/her account and set the password.

* **Update Moderator’s Account Status**

By entering a username, an administrator can update the account status of a particular moderator. There are three types of account status: unverified, verified or disabled. An unverified account requires a moderator to set his/her password in order to login to UPOD. Once the moderator finishes the registration process, his/her account status will change to verified. A verified account allows a moderator to login to UPOD. A disabled account prevents a moderator to login to UPOD.

* **Alter a Moderator’s Permissions**

By entering a username, an administrator can alter the permissions of a particular moderator. With a given permission, a moderator can add, delete, or modify the corresponding categories and pages. Note that an administrator has the right to add, delete, or modify all the categories and pages in UPOD.

* **List Administrators**

An administrator has the option to list all the administrators. The displayed table contains usernames, email addresses, and corresponding account status as well as permissions.

* **List Moderators**

An administrator has the option to list all the moderators. The displayed table contains usernames, email addresses, and corresponding account status as well as permissions.

* **Add/Delete/Modify Categories**

An administrator or a moderator can add, delete, or modify categories in UPOD.

* **Add/Delete/Modify Pages**

An administrator or a moderator can modify the content displayed on webpages, and add/delete pages in UPOD.

* **View Categories**

A user can view categories that are available in UPOD.

* **View Pages**

A user can view pages that are available in UPOD.

* **Search for Categories/Pages**

By inputting keywords in the search bar, a user can view the related categories/pages in the search results page.

* **Interact with Graphs**

A user can interact with graphs that are in SVG format.

# External Interface Requirements

## User Interfaces

## Hardware Interfaces

## Software Interfaces

## Communications Interfaces

# System Features

## Database-Driven Web Pages

4.1.1 Description

Database-driven web pages extract information from a database, and inserts that information into the corresponding web page. A database is used to store and organize data. In the database of this website, the types of data stored include administration information table, formula variables, learning modules, and articles. By using database, articles from two different tables can be related, information of an article is always stored only once, as well as potential issues of scalability and reliability can be minimized. Having a well-designed database schema could improve the implementation of a search engine.

4.1.2 Stimulus/Response Sequences

4.1.3 Functional Requirements

4.1.3.1: Each article at least has one table entry in the article table.

4.1.3.2: Learning modules are stored as SVG format.

4.1.3.3: Keep track of some key data elements so that users could revert changes

4.1.3.4: There is a strict data hierarchy for each data element.

(e.g. Variables → Formulas / Learning Modules → Articles)

## Server-Side Scripting

4.2.1 Description

Having scripts stored on a server can generate customized responses by selecting corresponding information from a database that resides on the server. Depending on a client’ request, the script can provide a customized interface for the client. The goal of server-side scripting is to provide additional functionalities to clients, which improve the overall user experience.

4.2.2 Stimulus/Response Sequences

4.2.3 Functional Requirements  
 4.2.3.1: Modify content displayed on webpages.

4.2.3.2: Allow users to search for specific content

4.2.3.3: Correlate articles that have similar or related physics content

4.2.3.4: Given a specific physics topic, practice problems are randomly generated, and a step-by-step solution is provided.

4.2.3.5: Generate correct files for each page on the front end, based on the given page URL

## User Management Portal (UMP)

4.3.1 Description and Priority

The UMP allows administrators to manage different types of user groups and modify user status of specific accounts.

4.3.2 Stimulus/Response Sequences

4.3.3 Functional Requirements

4.3.3.1: Administrator login checks whether the entered username and password are matched.

4.3.3.2: Usernames and passwords are stored in a database.

4.3.3.3: Once a user is registered, an email is sent from a mail server that authenticate the user’s identity.