Software Requirements Specification

for

UPOD - Back End

**Version 0.2**

**Prepared by Clive Shen**

**Wilfrid Laurier University, CP317**

**May 17, 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 | 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

## Product Functions

## User Classes and Characteristics

## Operating Environment

## Design and Implementation Constraints

## User Documentation

## Assumptions and Dependencies

# 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 have 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 hierachy 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 spefific 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 user name and password are matched.

4.3.3.2: Adminstrator interface gives adminstrators the options to list all current users, create an acccount for a user, and alter user permission as well as account status.

4.3.3.3: User name and password are stored in a database.

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

