Software Requirements Specification

for

UPOD - Back End

**Version 0.5**

**Prepared by Clive Shen**

**Wilfrid Laurier University, CP317**

**June 17, 2016**

**Table of Contents**

1. Introduction 3

1.1 Scope 3

1.2 Definition, Acronyms, and Abbreviations 3

1.3 References 3

1.4 Overview 4

2. Overall Description 5

2.1 Product Perspective 5

2.2 Constraints 6

3. Specific Requirements 7

3.1 External Interfaces 7

3.2 Functions 7

3.3 Performance Requirements 8

3.4 Logical Database Requirements 8

3.5 Software System Attributes 9

3.6 Organizing the Specific Requirements 9

4. Supporting Information 10

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Clive Shen | 17/06/16 | Edit based on new template | v0.5 |
| Tom West | 31/05/16 | Aligned Introduction & Formatting | v0.4 |
| 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

## Scope

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 in the search bar, articles related to those keywords should be shown. With the use of server-side scripting, administrators and moderators of UPOD can easily modify the contents of the webpages.

## Definition, Acronyms, and Abbreviations

* Administrator: 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: login, logout, reset password, add/delete/modify categories, and add/delete/modify pages
* User: view categories/pages, search for categories/pages, and interact with graphs

## References

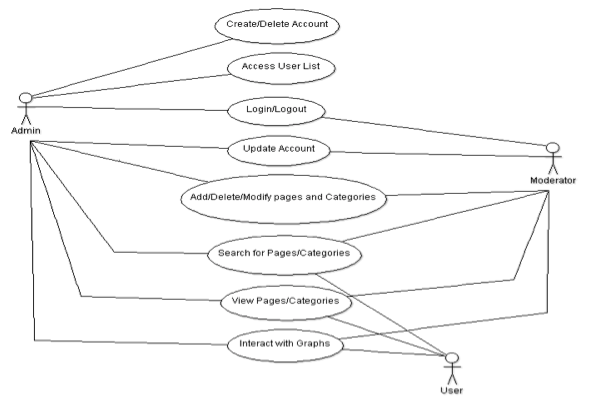
* IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

## Overview

The back end requirement discusses the following topics:

* Interfaces of the UPOD Back End
* Functions of the UPOD Back End
* Requirements of the UPOD Back End
* Constraints of the UPOD Back End

# 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.

* + 1. **System Interfaces**

UPOD operates on modern and standard browsers, including

* Safari
* Google Chrome
* Firefox
* Internet Explorer (Microsoft Edge)
  + 1. **User Interfaces**

N/A for back end

* + 1. **Software Interfaces**
       1. Database

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, and articles.

* Build connections between related articles.
* Ensure information is always stored only once.
* Improve the implementation of a search engine.
* Potential issues of scalability and reliability can be minimized.
  + - 1. User Management Portal (UMP)
* Allow administrators to manage moderators and modify permission of each moderator.
* Add, delete, or edit page layouts and contents.
  + 1. **Communication Interfaces**
* Hypertext Transfer Protocol (HTTP): The underlying protocol used by the World Wide Web, which is a set of rules for transferring files.
* Hypertext Transfer Protocol Secure (HTTPS): A protocol for secure communication, which consists of communication over HTTP within a connection encrypted by Transport Layer Security (TLS) or Secure Sockets Layer (SSL).
  + 1. **Memory Constraints**

N/A for Back-end

## Constraints

The budget of UPOD is $0, so any software must be free to use.

# Specific Requirements

## External Interfaces

N/A for Back-end

## Functions

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

* Login: In the User Management Portal, an administrator or a moderator needs to enter their username and password to authenticate their identity.
* Logout: To terminate their 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 their 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 their 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 their password in order to login to UPOD. Once the moderator finishes the registration process, their account status will change to verified, and the moderator can login to UPOD. A disabled account prevents a moderator from logging in 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.

## Performance Requirements

Will be added in at a later stage.

## Logical Database Requirements

Details for this section will be added in at a later stage, once information becomes more apparent.

* + 1. Types of information used by various functions
    2. Frequency of use
    3. Accessing capabilities
* Administrators can access all moderators’ attributes, except their passwords.
* Moderators can view their own permissions (add, edit, or remove articles).
  + 1. Data entities and their relationships
    2. Integrity constraints
    3. Data retention requirements

## Software System Attributes

* + 1. Reliability
* Database provides up-to-date, accurate information.
* No duplicate information is stored in the database.
* The search engine should provide a list of articles that are contains the key words or are related to the key words.
  + 1. Availability
    2. Security
* Account passwords of administrators and moderators accounts should be stored securely.
* Without appropriate permissions, no articles should be added, removed, or edited.
  + 1. Maintainability
* Additional features may be added to the website in the future, so the whole software should be well documented and easy to maintain.
  + 1. Portability

## Organizing the Specific Requirements

* + 1. System Mode
    2. User Class
    3. Objects
    4. Feature
    5. Stimulus
    6. Response
    7. Functional Hierarchy

# Supporting Information