# SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

# MANY VOICES PUBLISHING PLATFORM

**Version 0.5**

**October 27, 2016**

**Prepared for:**

**Dr. Carlos Jensen, D. Kevin McGrath, and Dr. Kirsten Winters**

**CS461 Senior Capstone**

**Prepared by:**

**Commix**

**Josh Matteson, Steven Powers, and Evan Tschuy**

**Oregon State University**

**Corvallis, OR 97330**

Admix - Many Voices Publishing Platform

**[INSERT YOUR PROJECT NAME] SSRS**
**TABLE OF CONTENTS**

**Section  Page**

# 1. Introduction

## A. IDENTIFICATION

The software system being considered for development is referred to as the Many Voices Publishing Platform. The customer providing specifications for the system is Dr. Carlos Jensen that is reachable at his email cjensen528@gmail.com. The ultimate customer, or end-user, of the system will be any user that wants an easier way to combine materials into a book. This is a new | project effort, so the version under development is version 0.5 release 0.

## B. PURPOSE

The purpose of the system under development is to provide a collaborative authoring platform that allows for users to combine various materials into a book. While the system will be used by professors and instructors in majority, this document is intended to be read and understood by software designers and coders. The document will also be vetted or approved by Dr. Carlos Jensen, D. Kevin McGrath, and Dr. Kirsten Winters.

## C. SCOPE

This new project is sponsored by Dr. Carlos Jensen, being developed by Josh Matteson, Steven Powers, and Evan Tschuy for completion of CS461 Senior capstone project class. This project is planned to be operated via a website for wide use.

## D. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

| Term or Acronym | Definition |
|---|---|
| Alpha test | Limited release(s) to selected, outside testers |
| Beta test | Limited release(s) to cooperating customers wanting early access to developing systems |
| Final test | aka, Acceptance test, release of full functionality to customer for approval |
| DFD | Data Flow Diagram |
| SDD | Software Design Document, aka SDS, Software Design Specification |
| SRS | Software Requirements Specification |
| SSRS | System and Software Requirements Specification |

| | |
|---|---|
| Chapter | An ordered collection of Sections and Scraps |
| Section | An ordered collection of Scraps belonging to a chapter |
| Scrap | A section of a textbook, which can contain formatted text (markdown or latex), and media |
| Media | A standalone image, figure, or video. Can be embedded in a Scrap |
| Internet Application | An interactive program that can be accessed and is based through a web server instead of being stored on a user's desktop |
| Source Control | An element of software design management, version control, and is the management of changes to documents, large web sites, computer programs, and other collections of data |
| User Interface (UI) | The means by which the user and a computer system interact, in particular the use of input devices and software |

## E. REFERENCES

[1]C. Jeffery, Systems and Software Requirements Specification (SSRS) Template, 2nd ed. Moscow: University of Idaho, 2016, pp. 1-22.

## F. OVERVIEW AND RESTRICTIONS

This document is for limited release only to Oregon State University professors and associated staff of D. Kevin McGrath, Dr. Kirsten Winters, and Jon Dodge, personnel working on the project and the client Dr. Carlos Jensen.

## 2. **OVERALL DESCRIPTION**

### A. *PRODUCT PERSPECTIVE*

The main differentiator from traditional textbook publication methods are the products' micro-writing abilities. A traditional textbook is written by one expert, using a defined curriculum that fits the needs of the author. The curriculum is a generic version of a multitude of classes that is somewhat useful for many professors, rather than very useful for few professors. Instead, the platform gives individual professors working with their own curriculum to create the textbook they need from the same kinds of components. This allows for textbooks that more closely track the intricacies of how classes are taught by different individuals.

### B. *PRODUCT FUNCTIONS*

The product is a document construction service. The most basic usage is the creation of a textbook out of pre-existing chapters, by searching and selecting from a list. Alternatively, chapters can created using pre-existing sections and media. A chapter or section can be created using a simple to use input interface that allows input of text and latex markup, along with the upload of media like image or perhaps video content.

### C. *USER CHARACTERISTICS*

The average user of the service will be a professor. Professors in general have decent but by no means high technical skills. Thus, the interface of the product will be highly graphically driven, using modern interface paradigms as defined by industry leaders such as Google's Material Design. The interface will allow users to not think about the technical specificities of the product, but more-so on the content they are using, the authors of the content, and how they are using it.

### D. *CONSTRAINTS*

A regular web browser will be the means of access for users. For the backend server, the main requirement is availability - as a web application, it has no safety- or life-critical components. A high (99.99but otherwise availability, safety, and specific hardware are not main concerns.

The software can, as a web application, be written in a high-level language, with minimal inter-application communication. In the case that the software is built on top of other applications such as git, inter-process communication should be handled in a generic way that provides a high-security barrier preventing malicious users from accessing other, non-specified applications.

## E. ASSUMPTIONS AND DEPENDENCIES

The product will be designed to run on a generic Linux-with-web server environment. Ward Cunningham's Federated Wiki, a similar project being drawn upon for inspiration, can run on any system with Ruby and CouchDB (a popular NoSQL database). With the product being either partially or wholly based on the Federated Wiki, similar system requirements are anticipated.

## F. SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

*1) Site dependencies:* As a web application, the product has no hard requirements other than high-speed network access and an instance of the CouchDB database management system.

- Hardware: standard virtual machine(s) running on cloud provider (ex: Amazon EC2 t2.medium)
- Database: CouchDB 2.0

*2) Safety, security and privacy requirements:* Private information shall be held to any relevant security standards. For instance, any credit card information processed by the product in the textbook creation process will be held to PCI compliance standards. Passwords shall be stored in a securely hashed, non-reversible method such that infiltration of the product by an unauthorized third party will not be able to result in the theft of user passwords or other private information.

*3) Performance requirements:* A standard user load shall be defined as 200 simultaneous users. User information will include things such as text documents, media uploads, and user account data.

Under a standard load,

- 95% of save operations (excepting media upload time) shall take under 500ms;
- 95% of initial page loads shall take under three seconds;
- 95% of user account operations (login, creation, deletion, etc) shall take under 500ms

*4) System and software quality:* The software shall maintain 99.99% accessibility from any modern web browser. As a non-safety-critical product, reliability shall be pushed for but not promised to end users. Testing shall be handled through a suite of automated unit and integration testing, as well as through manual checking of newly written and critical features.

*5) Packaging and delivery requirements:* The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD's and/or via email, as specified by the customer at time of delivery. Although document "drops" will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.

*6) Personnel-related requirements:* The system under development has no special personnel-related characteristics.

*7) **Training-related requirements:*** No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

*8) **Logistics-related requirements:*** The software shall run on a standard Amazon Web Services backend instance with another standard instance serving as the database.

*9) **Other requirements:*** No other system requirements are anticipated.

*10) **Precedence and criticality of requirements:*** The main critical requirements of the system are those relating to payment processing secrecy.

## 3.  SPECIFIC REQUIREMENTS

*A.  EXTERNAL INTERFACE REQUIREMENTS*

*1) Hardware Interfaces:*   [ insert your text here ]

*2) Software Interfaces:*   [ insert your text here ]

*3) User Interfaces:*   [ insert your text here ]

*4) Other Communication Interfaces:*   [ insert your text here ]

## B. SYSTEM FEATURES

*1)* *Use Case Diagrams:* [insert 1+ use case diagrams here]

*2)* ***General website features :***

- These may come free if we fork the federated wiki
- Material/otherwise nice looking design template
- User account management
- Create new account
- Edit existing account
- Login/logout functionality

*3)* ***Compile a book: :***

- Search functionality for finding existing textbooks
- Ability to fork existing textbooks
- Add a new blank textbook

*4)* ***Textbook compilation interface :***

- Table of Contents view, with access to New Chapter and Edit Chapter
- Edit/New Chapter:
- Button to Add New Section/Scrap
- Button to Search Sections/Scraps
- User-friendly interface to add Section/Scrap to desired location in Chapter

*5)* ***Search :***

- By keyword, author, tag, field; ranked by relevance and/or community voting
- Scrap/Section search
- Textbook search

*6)* ***Publishing Interface :***

- Indicate number of copies/digital access for students, determine price
- Implement payment processor – TODO far future
- Save textbook as PDF/send to printer

*7)* ***Scrap Editor:***

- "Pretty" view for adding formatted text, media, etc to a scrap
- "Raw" view for managing
- Tag management (add, delete, search? tags)
- Price management

*For each feature, you should either provide a Use Case Description **or** a Non-task feature description, whichever is more appropriate.*

| Use Case Description | a) *Non-task feature description:* |
|---|---|
| **Name** | |
| Actors | b) *Introduction/Purpose of this feature:* [ insert your text here ] |
| Goals | |
| Preconditions | c) *Input/Output sequence for this feature:* [ insert your text here ] |
| **Summary** | |
| Related use cases | d) *Design constraints of this feature:* [ insert your text here ] |
| **Steps** | |
|   1. ... | e) *Performance requirements of this feature:* [ insert your text here ] |
|   2. ... | |
| Alternatives | f) *Detailed functional requirements of this feature:* |
| Postconditions | g) *:* [ insert your text here ] |

8) *System feature 2: [ insert feature name here ]:*

  a) ***Introduction/Purpose of this feature:*** [ insert your text here ]

  b) ***Input/Output sequence for this feature:*** [ insert your text here ]

  c) ***Design constraints of this feature:*** [ insert your text here ]

  d) ***Performance requirements of this feature:*** [ insert your text here ]

  e) ***Detailed functional requirements of this feature:*** **System feature [m]: [ insert feature name here ]**

  f) ***Introduction/Purpose of this feature:*** [ insert your text here ]

  g) ***Input/Output sequence for this feature:***

  h) ***Design constraints of this feature:*** [ insert your text here ]

  i) ***Performance requirements of this feature:*** [ insert your text here ]

  j) ***Detailed functional requirements of this feature:*** [ insert your text here ]