Oregon State University
College of Engineering

CS CAPSTONE

FEBRUARY 17, 2017

MANY VOICES PLATFORM

PREPARED FOR

OREGON STATE UNIVERSITY

CARLOS JENSEN

PREPARED BY

GROUP 61

REMIX

JOSH MATTESON
STEVEN POWERS
EVAN TSCHUY

Abstract

An overview of the requirements for the Many Voices Publishing Platform. Introduces the Many Voices Publishing Platform, describes in detail the purpose, product functions, user characteristics, assumptions and dependencies, and system level (non-functional) requirements and lists specific requirements that are outside of the platform, system features, and an overview of the development life cycle.

CONTENTS

1 Revision Log 2

2 Introduction 2

2.1 IDENTIFICATION . . . 2
2.2 PURPOSE . . . 2
2.3 SCOPE . . . 2
2.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS . . . 2
2.5 REFERENCES . . . 3
2.6 OVERVIEW AND RESTRICTIONS . . . 3

3 OVERALL DESCRIPTION 4

3.1 PRODUCT PERSPECTIVE . . . 4
3.2 PRODUCT FUNCTIONS . . . 4
3.3 USER CHARACTERISTICS . . . 4
3.4 CONSTRAINTS . . . 4
3.5 ASSUMPTIONS AND DEPENDENCIES . . . 4
3.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS . . . 5
3.6.1 Site Dependencies . . . 5
3.6.2 Safety, Security and Privacy Requirements . . . 5
3.6.3 Performance Requirements . . . 5
3.6.4 System and Software Quality . . . 5
3.6.5 Packaging and Delivery Requirements . . . 5
3.6.6 Personnel-related requirements . . . 5
3.6.7 Training-related requirements . . . 5
3.6.8 Logistics-Related Requirements . . . 6
3.6.9 Other Requirements . . . 6
3.6.10 Precedence and Criticality of Requirements . . . 6

4 SPECIFIC REQUIREMENTS 7

4.1 EXTERNAL INTERFACE REQUIREMENTS . . . 7
4.1.1 Hardware Interfaces . . . 7
4.1.2 Software Interfaces . . . 7
4.1.3 User Interfaces . . . 7
4.1.4 Other Communication Interfaces . . . 7
4.2 SYSTEM FEATURES . . . 8
4.2.1 System feature 1: General Website Features . . . 8
4.2.2 System feature 2: Compile a book . . . 8
4.2.3 System feature 3: Textbook Compilation Interface . . . 8
4.2.4 System feature 4: Search Feature . . . 8

4.2.5 System feature 5: Publishing Interface 8
4.2.6 System feature 6: Scrap Editor 8

5 APPENDIX A. Gantt Chart 9

6 APPENDIX B. UML Diagram 10

References 11

1 REVISION LOG

Name	Change Number	Date	Description of Change
Steven Powers	1	2/13/2017	Updated document to clarify that we will not be using Ward Cunningham's Federated Wiki as the base, but instead as inspiration.
Steven Powers	2	2/16/2017	Updated Gantt Chart to feature an adjusted and updated project timeline.

2 INTRODUCTION

2.1 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 jensenca@engr.orst.edu 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.

2.2 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.

2.3 SCOPE

The 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. The product is to be operated via a website for wide availability and use.

2.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term or Acronym	Definition
Alpha Test	Limited release(s) to selected, outside testers (Friends and Family)
Beta Test	Limited release(s) to cooperating customers wanting early access to developing systems (Professors and other educators)
Final Test	Release of full functionality to customer for approval
SDD	Software Design Document
SSRS	System and Software Requirements Specification
PDF	Portable Document Format, that is able to combine text, graphics, and other information into a single document

PCI	Payment Card Industry, is a proprietary information security standard for credit cards in an effort to reduce credit card fraud
Scrap	A section of a textbook, which can contain formatted text (markdown or latex), and media
Section	An ordered collection of Scraps belonging to a chapter
Chapter	An ordered collection of Sections and Scraps
Media	A standalone image, figure, or video. Can be embedded in a Scrap
Web 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

2.5 REFERENCES

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

2.6 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.

3 OVERALL DESCRIPTION

3.1 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, our solution will give the tools necessary for individual professors to create the textbook they need to match the curriculum desired. This allows for textbooks that more closely track the intricacies of how classes are taught by different individuals.

3.2 PRODUCT FUNCTIONS

The product will be a document construction service. The most basic usage will be 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 input interface that accepts text and LaTeX markup, along with the upload of media like image or perhaps video content.

3.3 USER CHARACTERISTICS

The expected user of the service will be a professor. Professors, in general, have decent means 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 [1]. 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.

3.4 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 (99.99%) availability should be provided, but 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.

3.5 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) [2]. Though not being directly based upon the Federated Wiki, the project has similar levels of complexity and engineering requirements. Therefore it is a reasonable benchmark to aim for. [2].

3.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

3.6.1 Site Dependencies

As a web application, the product has no hard requirements other than high-speed network access and an instance of a chosen database management system.

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

3.6.2 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.

3.6.3 Performance Requirements

A standard user load shall be defined as up to 100 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 be responsive in nature.
- 95% of initial page loads shall shall be responsive in nature.
- 95% of user account operations (login, creation, deletion, etc) shall be responsive in nature.

3.6.4 System and Software Quality

The software shall maintain 99.99% accessibility from the major modern web browsers, including Mozilla FireFox, Google Chrome, and Microsoft Edge. As a non-safety-critical product, reliability shall be attempted 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.

3.6.5 Packaging and Delivery Requirements

The executable system and all associated documentation (i.e., SSRS, 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.

3.6.6 Personnel-related requirements

The system under development has no special personnel-related characteristics.

3.6.7 Training-related requirements

No training materials or expectations will be tied to this project other than the limited help screens built into the software and the accompanying user manual.

3.6.8 Logistics-Related Requirements

The software shall run on a standard Amazon Web Services backend instance with another standard instance serving as the database.

3.6.9 Other Requirements

No other system requirements are anticipated.

3.6.10 Precedence and Criticality of Requirements

The main critical requirements of the system are those relating to payment processing secrecy.

4 SPECIFIC REQUIREMENTS

4.1 EXTERNAL INTERFACE REQUIREMENTS

4.1.1 Hardware Interfaces

User should have a basic computer with no extensive requirements. The project should be able to run on a standard virtual machine instance.

4.1.2 Software Interfaces

Web front-end using JavaScript or JavaScript variant.

4.1.3 User Interfaces

Angular2 - Single Page Application, or other similar variant

4.1.4 Other Communication Interfaces

Git, email, comments

4.2 SYSTEM FEATURES

4.2.1 General Website Features

- These may come free if we duplicate and edit the Federated Wiki.
- Material/otherwise nice looking design template
- User account management
- Create new account
- Edit existing account
- Login / logout functionality

4.2.2 Compile a Book

- Search functionality for finding existing textbooks
- Ability to duplicate and edit existing textbooks
- Add a new blank textbook

4.2.3 Textbook Compilation Interface

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

4.2.4 Search

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

4.2.5 Publishing Interface

- Save textbook as PDF/send to printer
- Optional future feature: payment processor integration

4.2.6 Scrap Editor

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

5 APPENDIX A. GANTT CHART

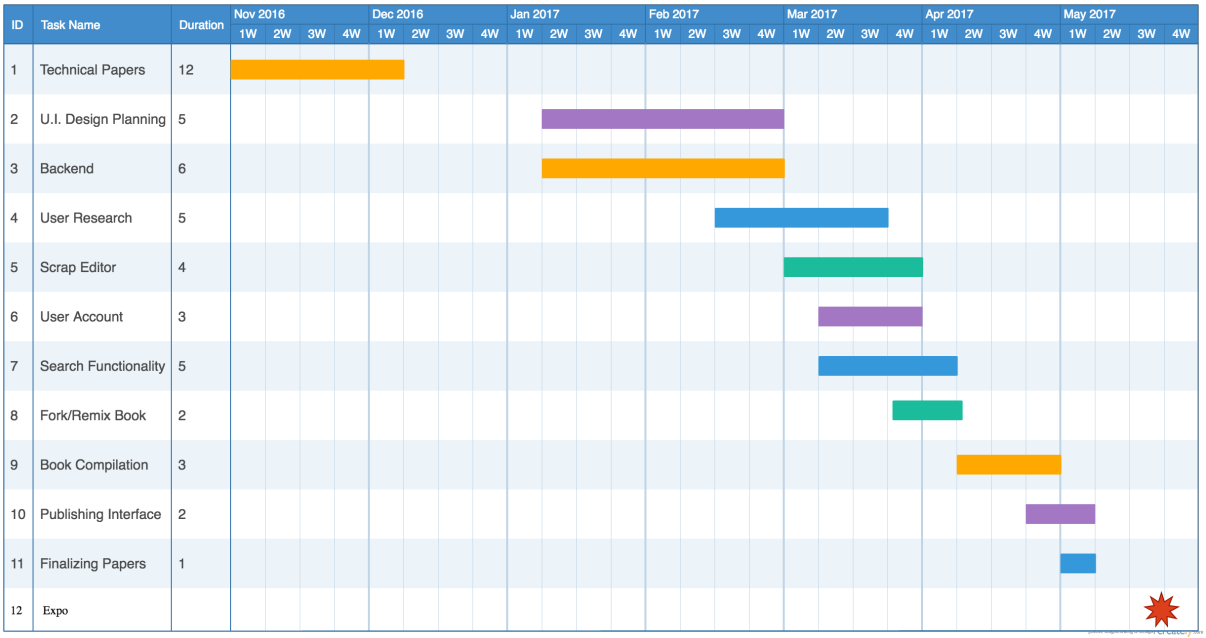

Fig. 1. Our planned Gantt Chart that outlines our planned working schedule for development of the MVP Platform

6 APPENDIX B. UNIFIED MODELING LANGUAGE DIAGRAM

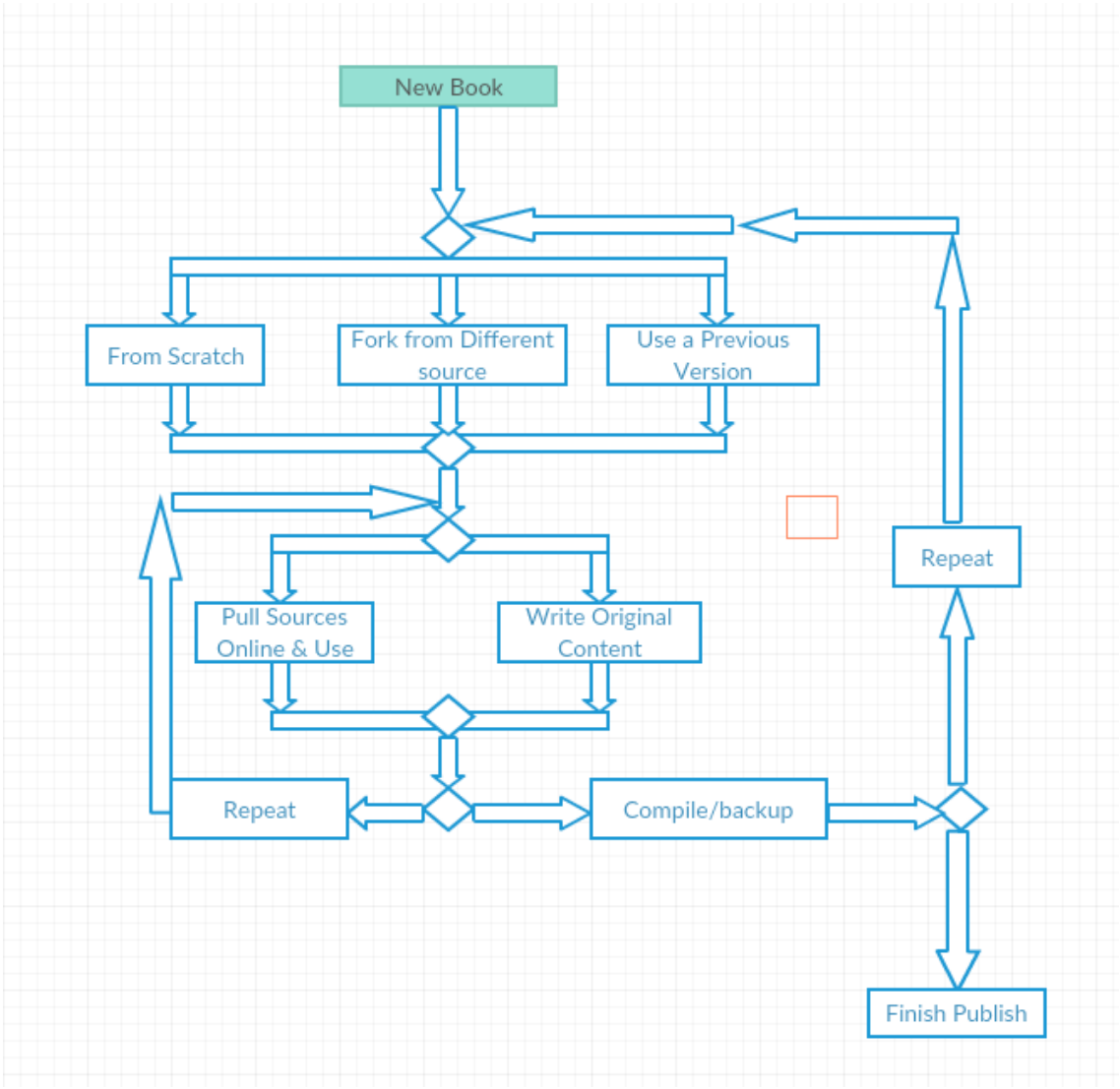

Fig. 2. A preliminary UML Diagram that outlines the basic flow of information

REFERENCES

[1] Google, “Material design,” https://material.google.com//.

[2] W. Cunningham, “Smallest federated wiki,” http://wardcunningham.github.io/.