



College of Engineering

CS CAPSTONE FINAL REPORT

JUNE 12, 2017

MANY VOICES PLATFORM

PREPARED FOR

OREGON STATE UNIVERSITY
CARLOS JENSEN

PREPARED BY

GROUP 61
REMIX

JOSH MATTESON
STEVEN POWERS
EVAN TSCHUY

Abstract

The culmination of documents created during the development of the Many Voices Publishing Platform by Team Remix; Josh Matteson, Steven Powers, and Evan Tschuy for Dr. Carlos Jensen. Details of the implementation, technologies, and requirements are included in this document.

4 APPENDIX A. GANTT CHART

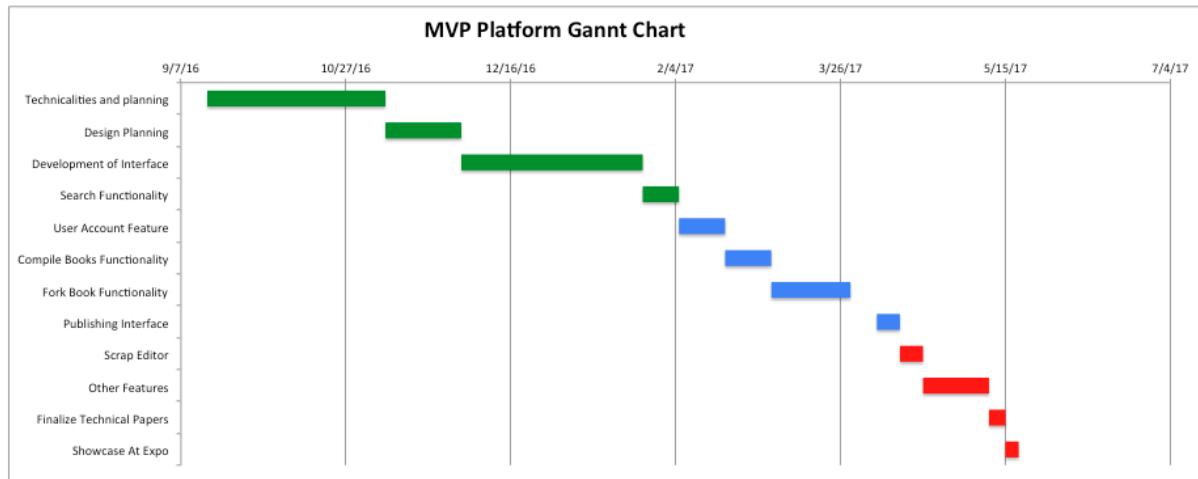


Figure 1. A simple preliminary Gantt Chart that outlines a rough sketch of our anticipated working time line

5 APPENDIX B. UNIFIED MODELING LANGUAGE DIAGRAM

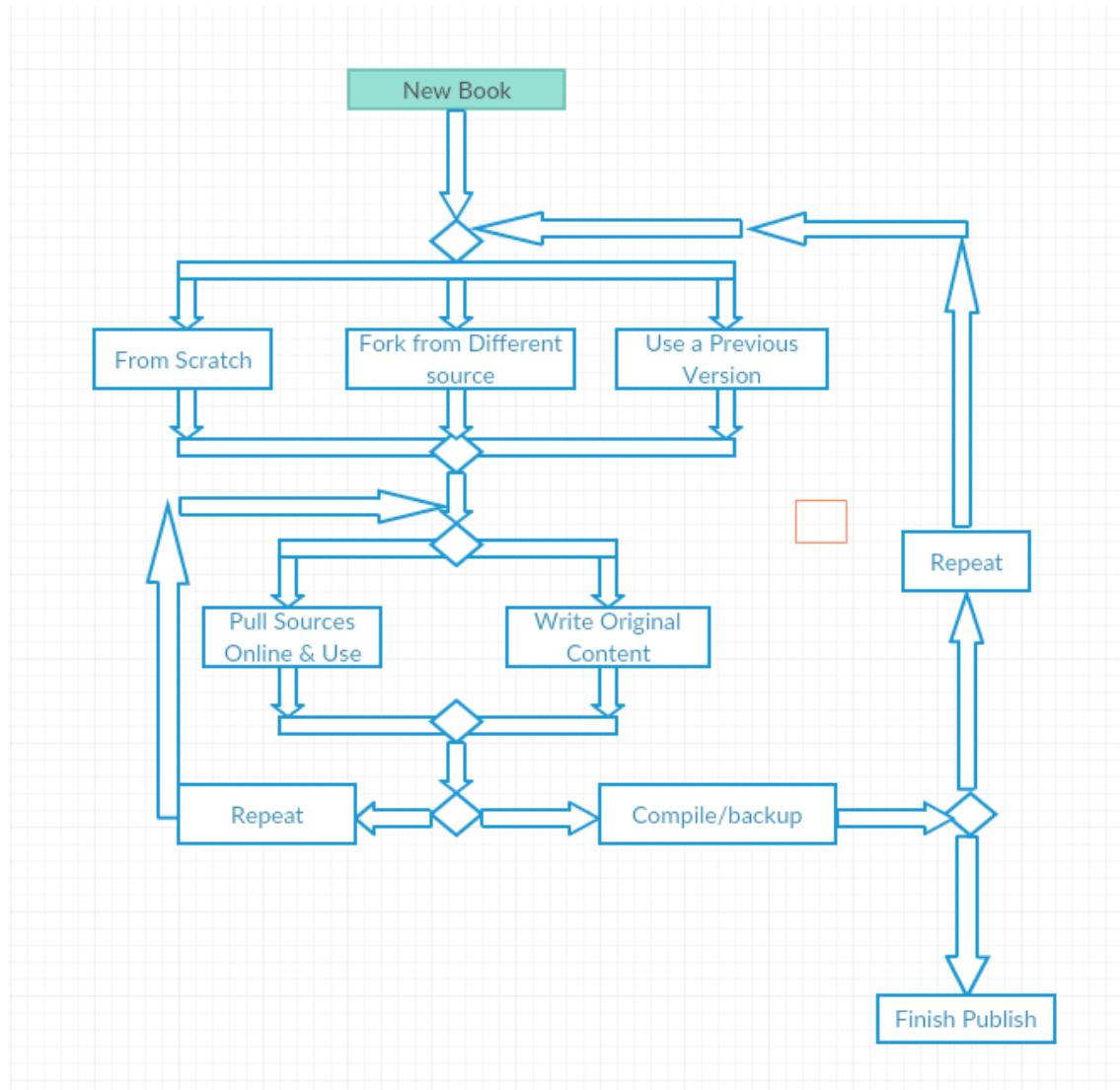


Figure 2. A preliminary UML Diagram that outlines the basic flow of information

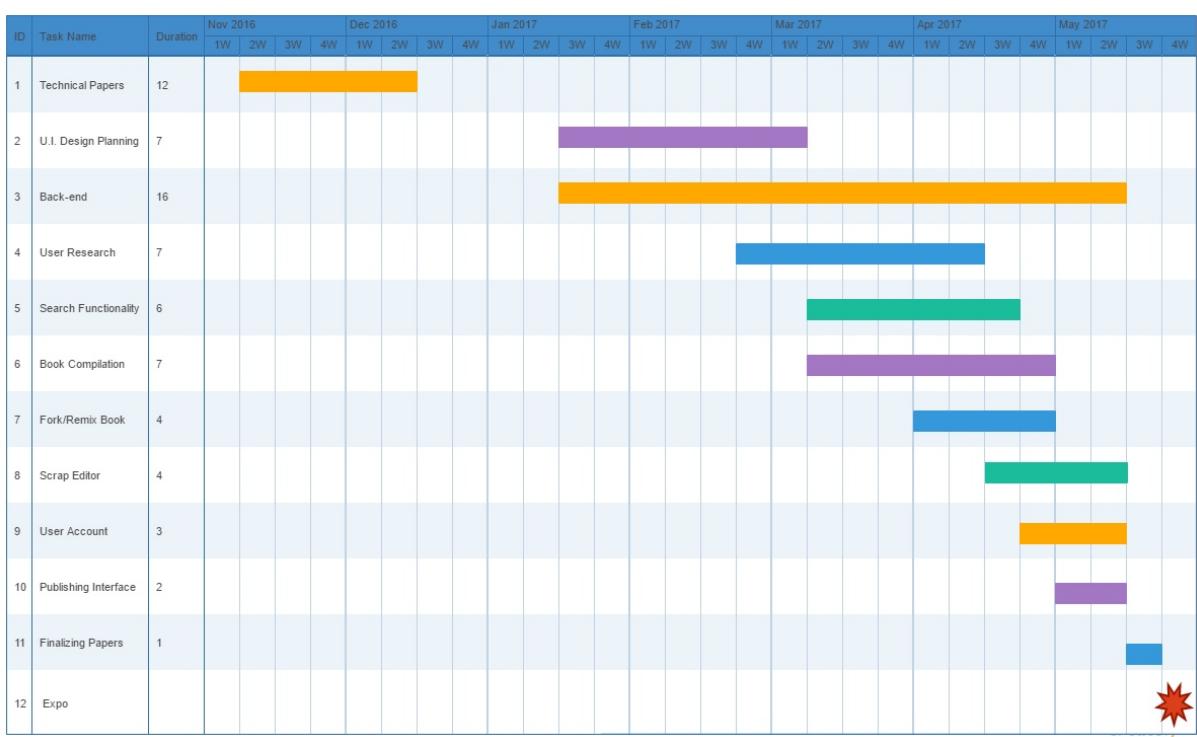


Fig. 1. The final Gantt Chart that outlines our working time line

Many Voices Publishing Platform

4.7 Design Rationale

For this project, design choices are made based on client specifications as well as development concerns due to technology availability and adaptability to the current system. Our client Dr. Carlos Jensen wants the project to allow for the easy creation of textbooks through what is called the Many Voices Publishing Platform. Design choices will be made to accommodate this requirement.

4.8 Design Timeline

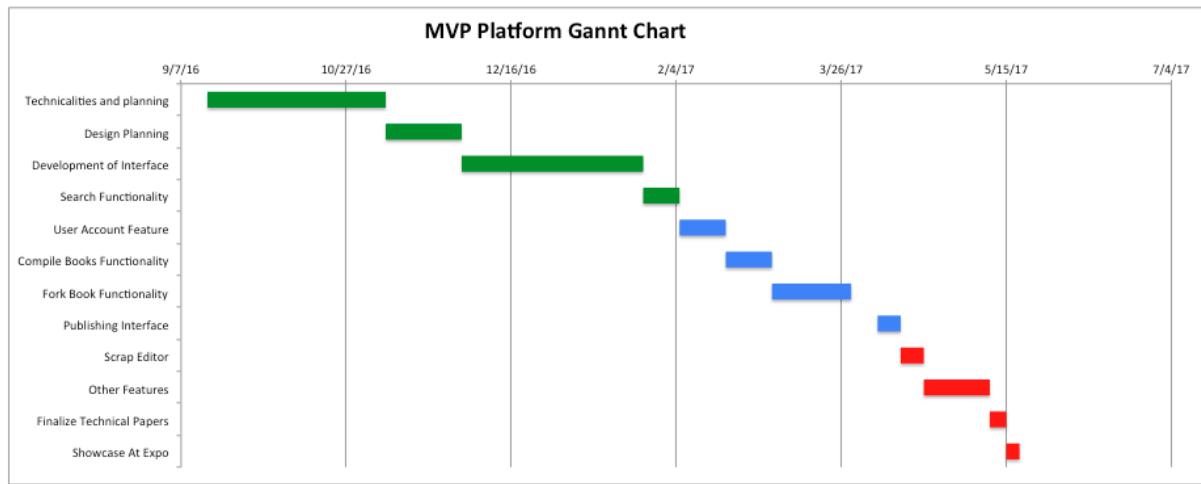


Fig. 1. A preliminary Gantt Chart that outlines a rough sketch of our anticipated working time line as of Fall term

4.9 Design Languages

This document will use UML as the design language.

Many Voices Publishing Platform

5.2.4 Function Attribute

This component provides the user interface for users to interact with while using the software. Handles display of information and provides the interface for input.

5.2.5 Relationship

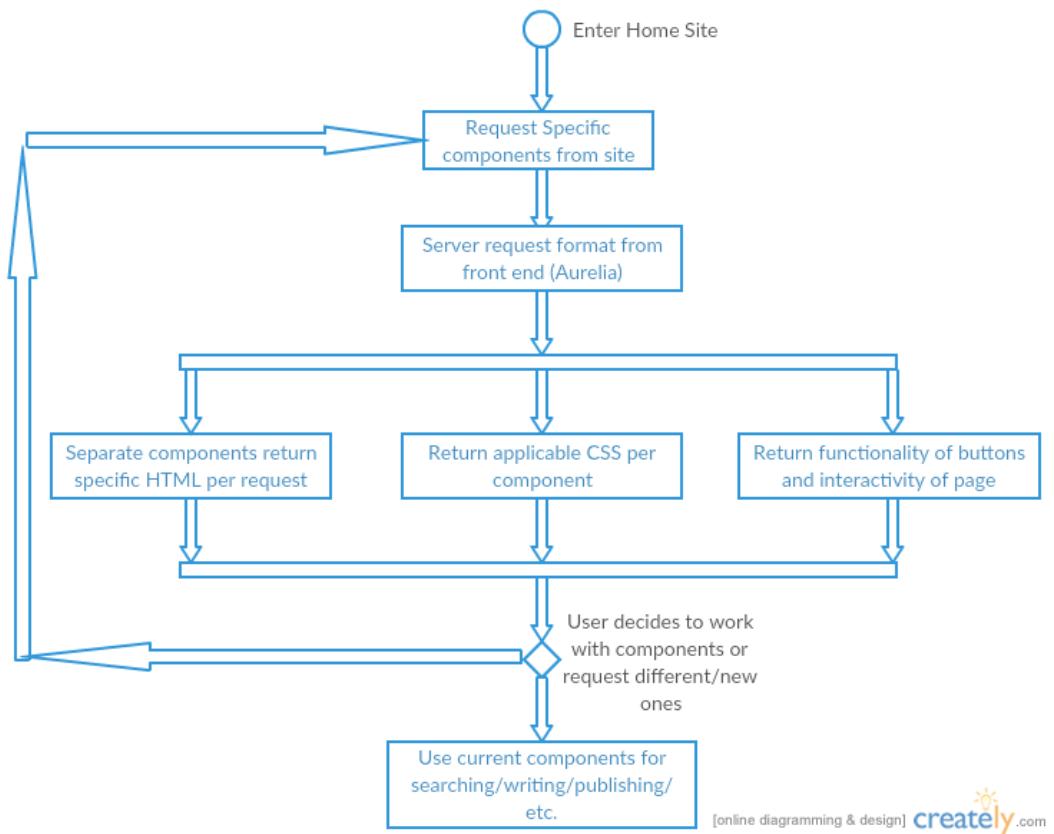


Fig. 2. A preliminary UML diagram of how the user interface tool of Aurelia will enable transmission of data to the screen or from the screen, up to date as of Fall term

Many Voices Publishing Platform

5.4.5 Relationship

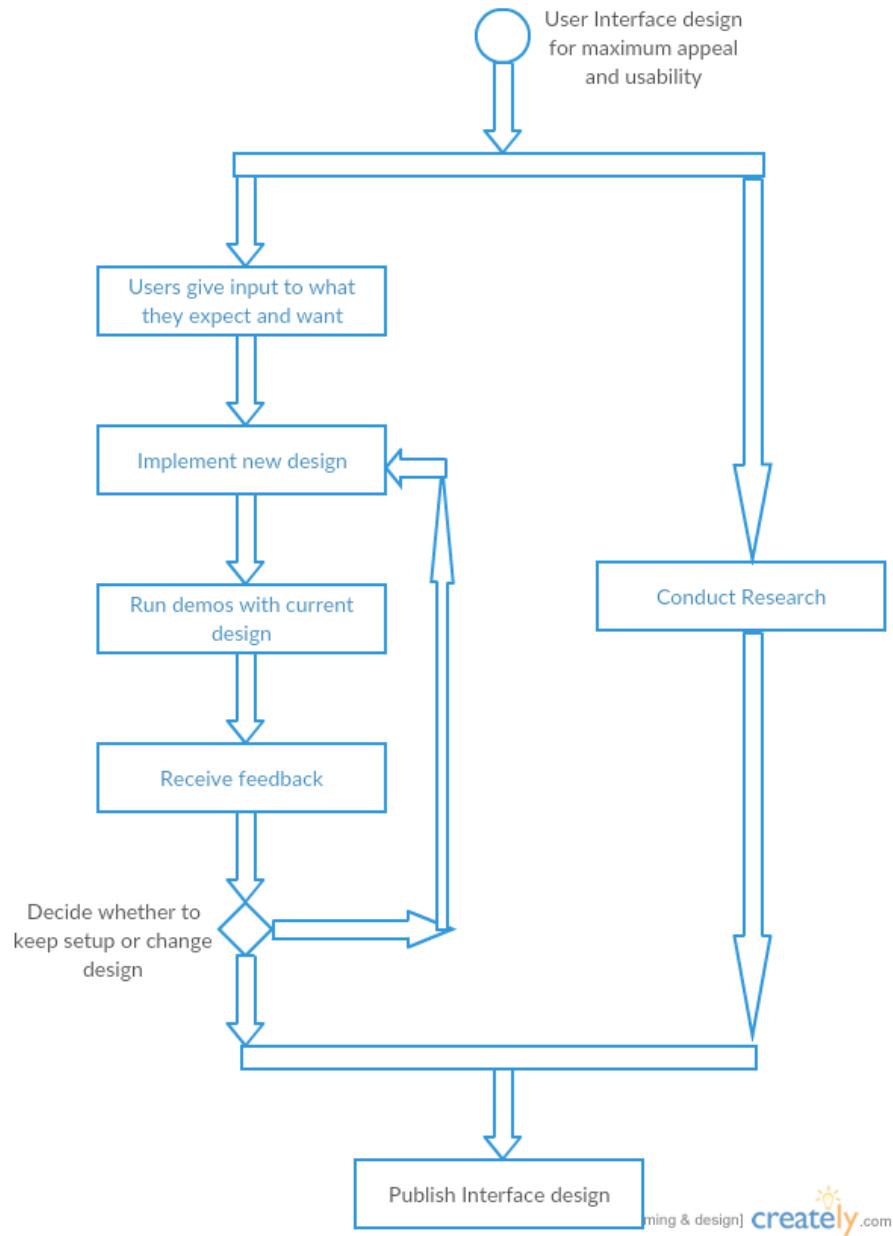


Fig. 3. A preliminary UML diagram of how user centered design & user research can improve a UI for increased affordance and usability for the target audience, up to date as of Fall term

Many Voices Publishing Platform

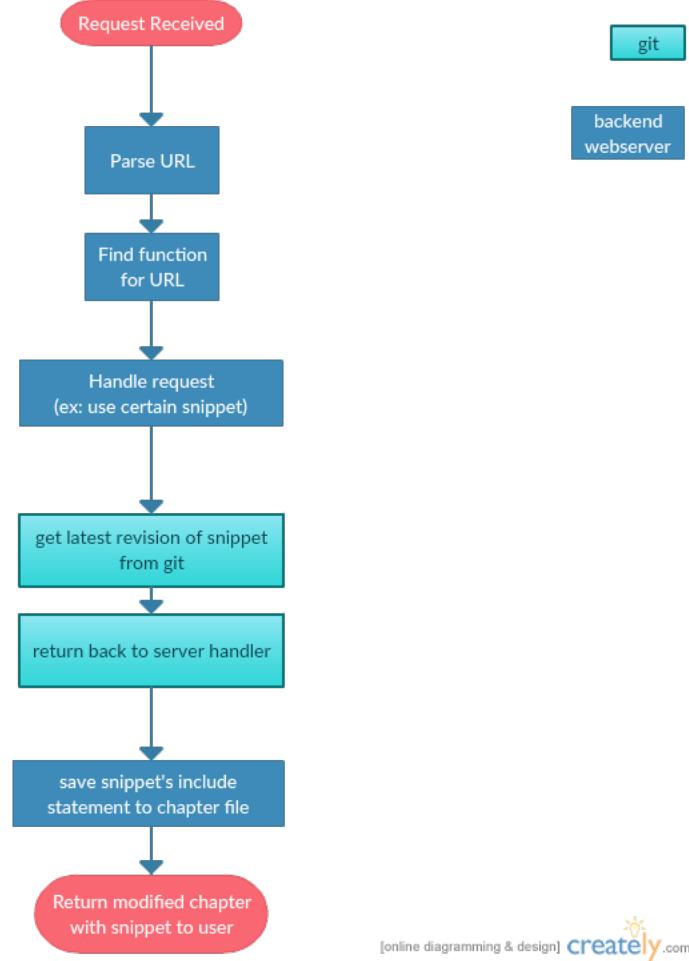


Fig. 4. An example backend handling flow. The application determines which function to call, and the function handles the request and sends any necessary file operations to the git module.

[online diagramming & design]  [creately.com](#)

5.6 Viewpoint: Text Formatting

By: Evan Tschuy

5.6.1 *Formatting*

The front end of the platform lets users interact with "snippets" of documents – a block of text that, paired with potentially dozens of other snippets, can make a chapter. Then, chapters need to be combined together to make a full textbook. Using a LaTeX backend, snippets can be related to an

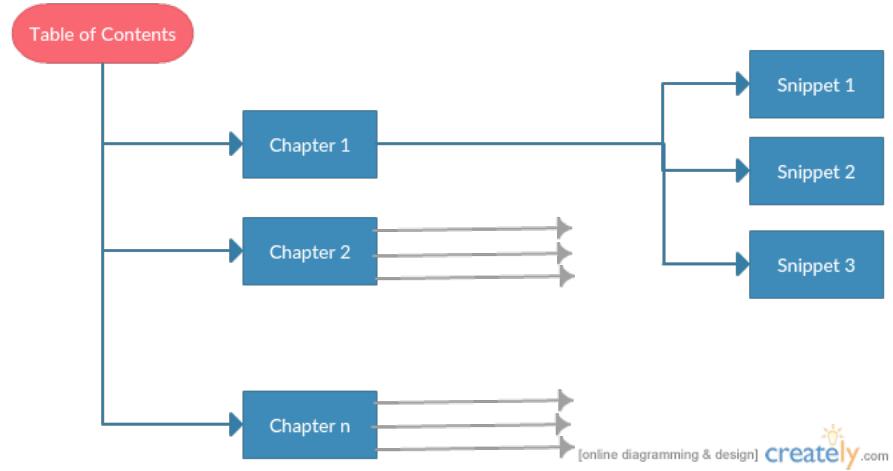


Fig. 5. A document is divided into chapters (which are included under a table of contents) and snippets (which are parts of a chapter).

5.7 Viewpoint: Password Storage

By: Evan Tschuy

5.7.1 *Password hashing*

Any time a password is being requested from a user, it should be securely and irreversibly hashed. By securely hashing a password, it becomes impossible for hackers with access to the user database to use stored credentials to compromise user accounts on other sites even if users reuse passwords between sites. A well designed cryptographic hash, such as bcrypt, includes a salt value, which is simply added to the beginning of the password at hash time, stored in the database in plain text, so that two uses of the same password result in different hashes.

5.7.2 *Design Concerns*

A secure password storage system always hashes passwords, and requires hackers to crack each password individually by guessing passwords one by one. Another alternative to storing user passwords at all is, as discussed above in the user authentication section, to use a third-party user authentication service such as those offered by Google, Facebook, etc. For a platform used by professors, who may

Many Voices Publishing Platform

not all have a Google or Facebook account, however, it should always be possible to create an account that does not require a third-party account.

5.7.3 Design Elements

A well-designed, secure system in which any password stored cannot be reversed without a slow cracking process. Preferably, users would not store any passwords at all with the service and instead would use third-party authentication.

5.7.4 Function Attribute

This component provides the functionality of secure password storage.

5.7.5 Relationship

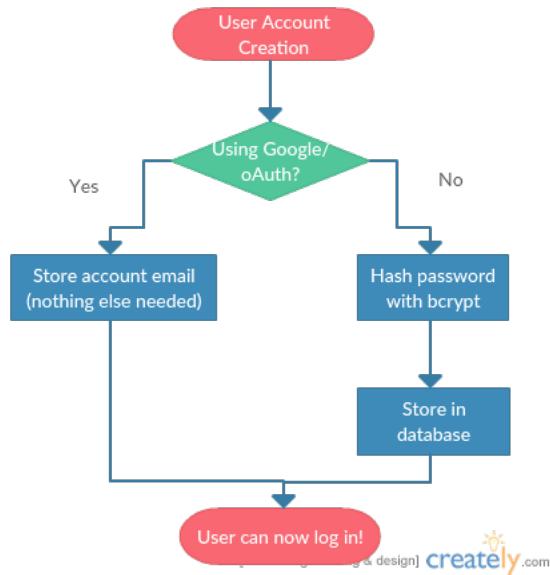


Fig. 6. Passwords will either be securely hashed, if the user is not using external authentication, or not needed at all, if the user is using external authentication.

Many Voices Publishing Platform

5.8.5 Relationship

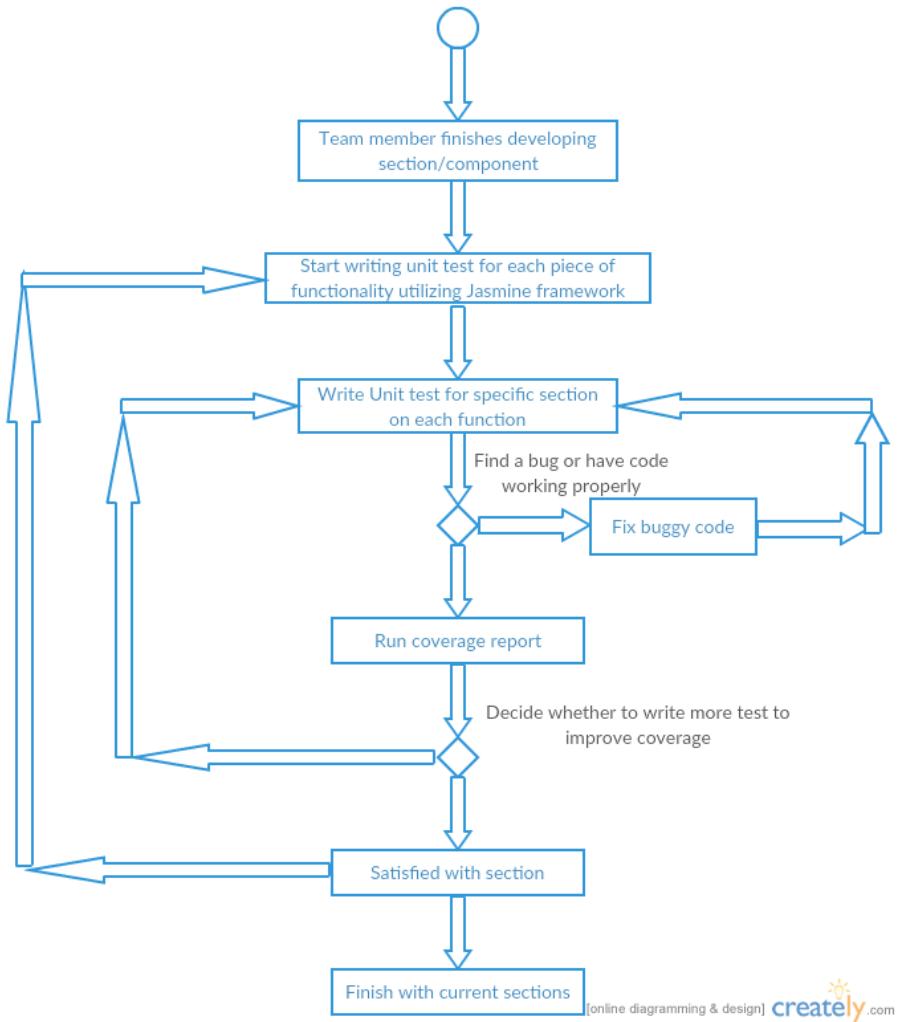


Fig. 7. A preliminary UML diagram demonstrating the flow of testing with Jasmine

Many Voices Publishing Platform

5.9.5 Relationship

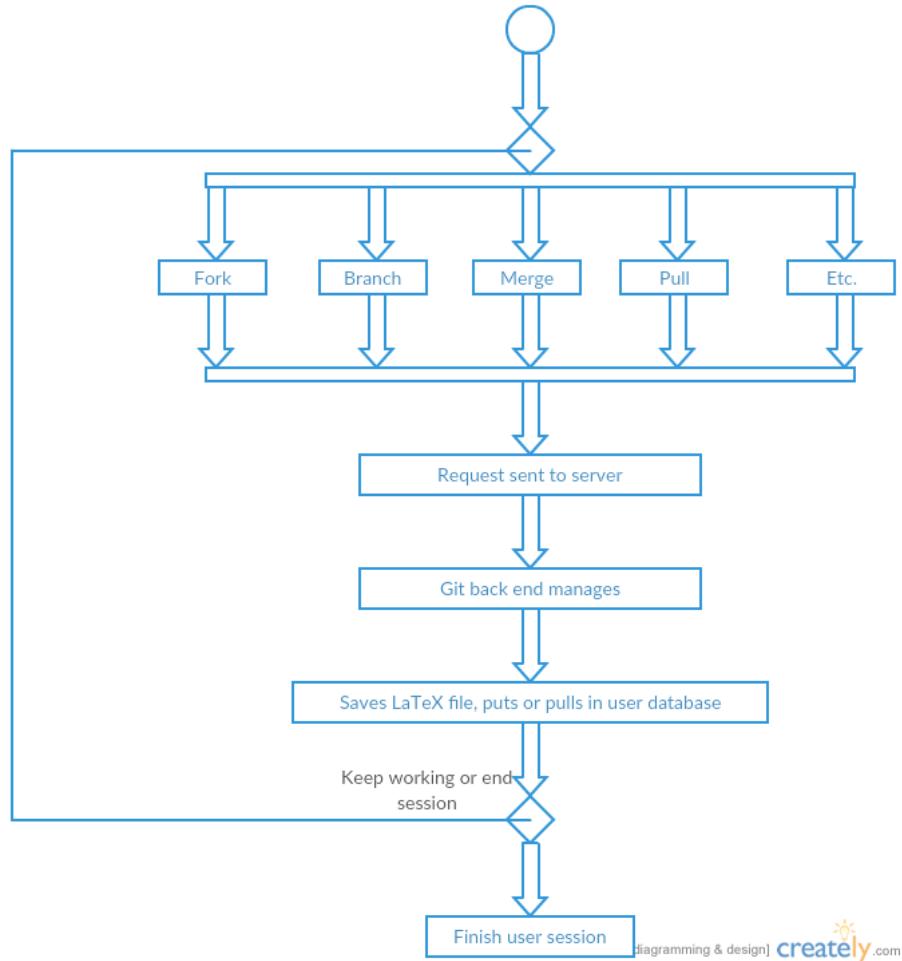


Fig. 8. A preliminary UML diagram demonstrating the relationship of Git to the request of the user



College of Engineering

CS CAPSTONE FINAL REPORT BLOG POSTS

JUNE 11, 2017

MANY VOICES PLATFORM

PREPARED FOR

OREGON STATE UNIVERSITY

CARLOS JENSEN

PREPARED BY

GROUP 61

REMIX

JOSH MATTESON

STEVEN POWERS

EVAN TSCHUY

Abstract

The culmination of blog posts created during the development of the Many Voices Publishing Platform by Team Remix; Josh Matteson, Steven Powers, and Evan Tschuy for Dr. Carlos Jensen.



College of Engineering

CS CAPSTONE FINAL REPORT FINAL POSTER

JUNE 11, 2017

MANY VOICES PLATFORM

PREPARED FOR

OREGON STATE UNIVERSITY

CARLOS JENSEN

PREPARED BY

GROUP 61

REMIX

JOSH MATTESON

STEVEN POWERS

EVAN TSCHUY

Abstract

The culmination of blog posts created during the development of the Many Voices Publishing Platform by Team Remix; Josh Matteson, Steven Powers, and Evan Tschuy for Dr. Carlos Jensen.

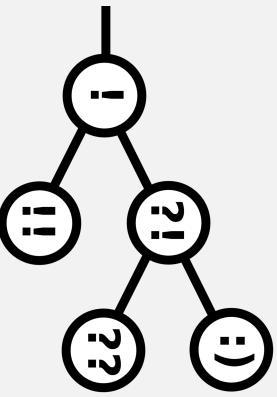
Why does it matter?

TEXTBOOKS NEED IMPROVING

Today, textbooks are filled with yearly updates that are often released with the minimum number of changes needed to justify a new edition.

Textbooks

- can cost hundreds of dollars
- are often outdated or contain incorrect information
- take a long time to write and review



The Many Voices Publishing Platform team logo representing the remixing of ideas and content

MANY VOICES PUBLISHING PLATFORM

Collaborative Textbook Publishing

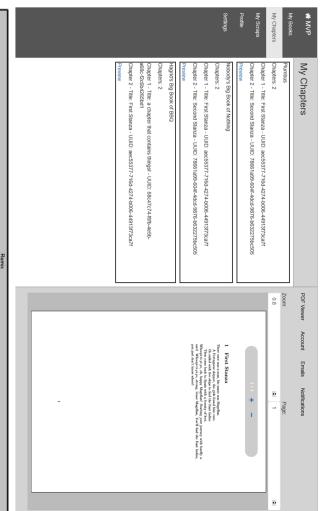
PROJECT OVERVIEW

A remixed textbook created with the Many Voices Publishing (MVP) Platform consists of a textbook object that contains chapters. These chapters are made up of a collection of scraps, which can be plaintext, LaTeX, or images.

TECHNOLOGIES USED

- Aurelia JavaScript Framework that provides a single page application through the use of a page router.
- Git backend for managing each scrap, chapter, and textbook
- NodeJS server interfacing between backend and user application to manage authentication and user relations

When a textbook has been created or remixed, the user is able to create a PDF of their textbook to distribute to their students or other intended audience.



WHAT IS REMIXING?

The idea of ‘remixing’ is that it is not always necessary to rewrite an entire document to cover what is needed for a course.

Instead, the MVP Platform encourages a more publicly available publishing concept where documents aren’t necessarily a single person’s writing, but “open source” documents edited by whomever wants to change them to fit their needs.

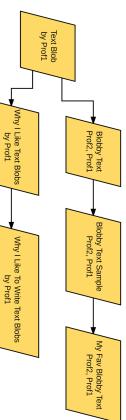


Figure: the “tree” of a paragraph being modified by multiple professors

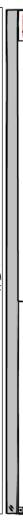
HOW DOES IT WORK?

The individual parts making up the books and their chapters are all stored separately.

Each individual paragraph is then individually modifiable - so if a professor wants to change a particular paragraph, it can be stored separately and then referenced in a new copy of that chapter.

This is done using a technology called “git”, which allows us to store a tree of changes. This means that we can keep old versions of the paragraph around for those who want them, and allow new users to modify the paragraph however they want.

- Content creators are able to submit materials of all different formats, including: images, small pieces of text, pages, or even entire chapters.
- A knowledgeable professor can contribute a chapter or unit on their specialty, and others are able to modify obtained materials in the way they see fit.
- Provide the ability to publish their textbook as a PDF when they desire.



A brief history of development (bottom to top): Low Level Prototype, Medium Level Prototype, and Alpha Implementation

CS Capstone Team #61 Team Remix



Team Members

Josh Matteson
mattesoj@oregonstate.edu

Steven Powers
powersst@oregonstate.edu

Evan Tschuy
tschuye@oregonstate.edu

Project Advisors
D. Kevin McGrath
Dr. Kirsten Winters
Jon Dodge

Project Sponsor
Dr. Carlos Jensen
jensenca@oregonstate.edu

The Many Voices Publishing Platform allows professors from all walks of life to have an equal opportunity to create the textbook they desire using textbook materials from peers around the world or create them from scratch.

Many Voices Publishing Platform User Guide

June 2017

Version 1.10

*Team Remix
Josh Matteson
Steven Powers
Evan Tschuy*

Table of Contents

1	Introduction	3
	<i>1.1.....Scope and Purpose.....</i>	<i>3</i>
	<i>1.2.....Process Overview.....</i>	<i>3</i>
2	New Textbook	5
	<i>2.1.....New Textbook Details.....</i>	<i>5</i>
3	Edit Textbook	6
	<i>3.1.....Edit Textbook Details.....</i>	<i>6</i>
4	Edit Chapter	8
	<i>4.1.....Edit Chapter Details.....</i>	<i>8</i>
5	New Scrap	10
	<i>5.1.....New Scrap Details.....</i>	<i>10</i>
6	Profile Settings	12
	<i>6.1.....Profile Settings Details</i>	<i>12</i>
7	Login.....	13
	<i>7.1.....Login Details.....</i>	<i>13</i>

1 Introduction

1.1 Scope and Purpose

The Many Voices Publishing Platform (MVP Platform) is a document creation tool aimed at professors looking to write a textbook for their courses but are short on availability. The MVP Platform allows a user to create their own **Textbooks**, **Chapters**, or **Scraps** or copy and modify existing textbooks, chapters, or scraps created by other users with the MVP Platform.

This user guide will provide basic guidelines for account login, textbook creation, editing a textbook, chapter creation, editing a chapter, scrap creation, creating an image scrap and profile settings.

1.2 Process Overview

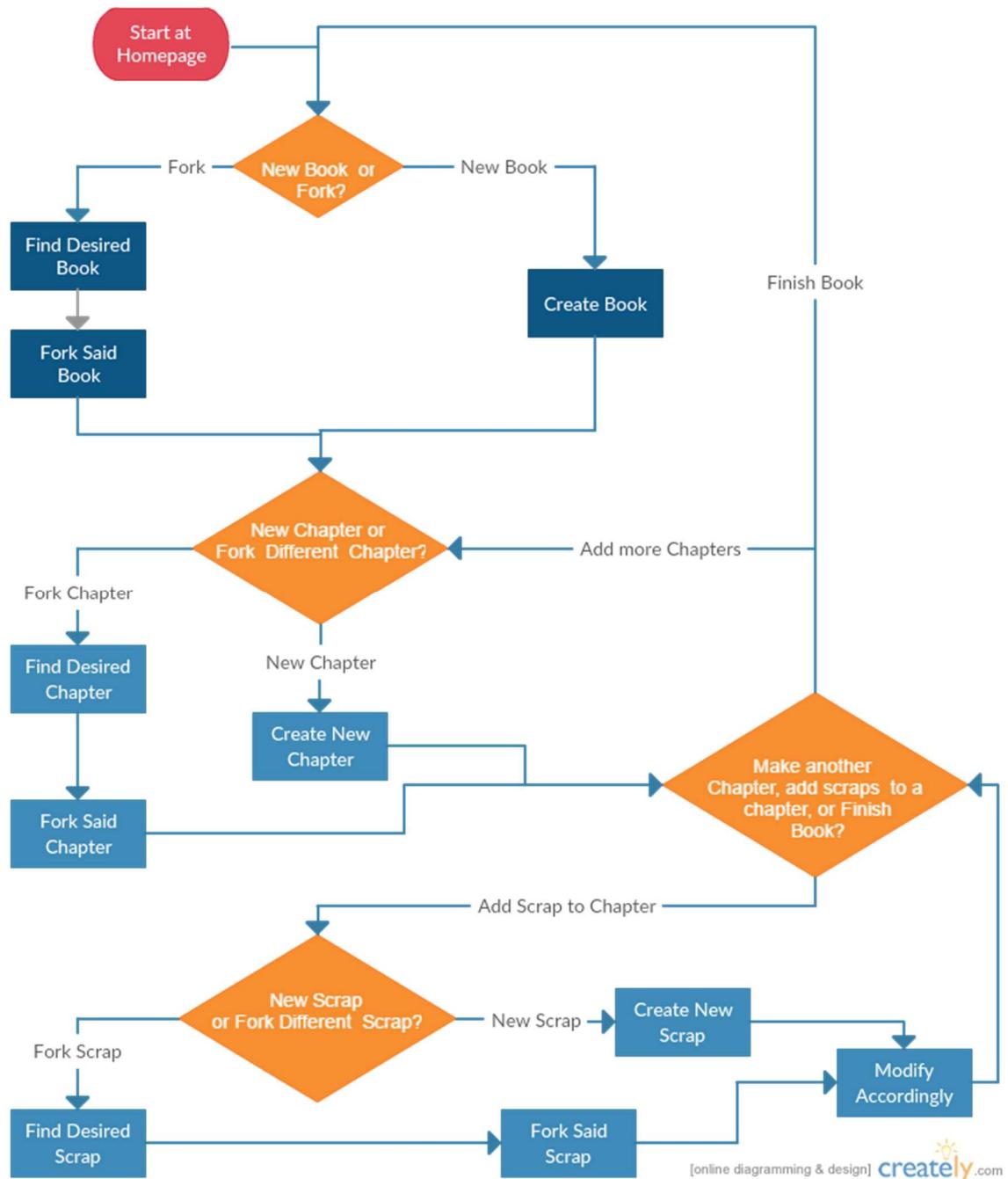
A user will typically follow the order of operations listed below, this user guide will cover these operations in this order.

1. New Textbook
2. Edit Textbook
3. New Chapter
4. Edit Chapter
5. New Scrap
6. New Image Scrap
7. Profile Settings

Many Voices Publishing Platform

USER GUIDE

Process Flow Diagram:



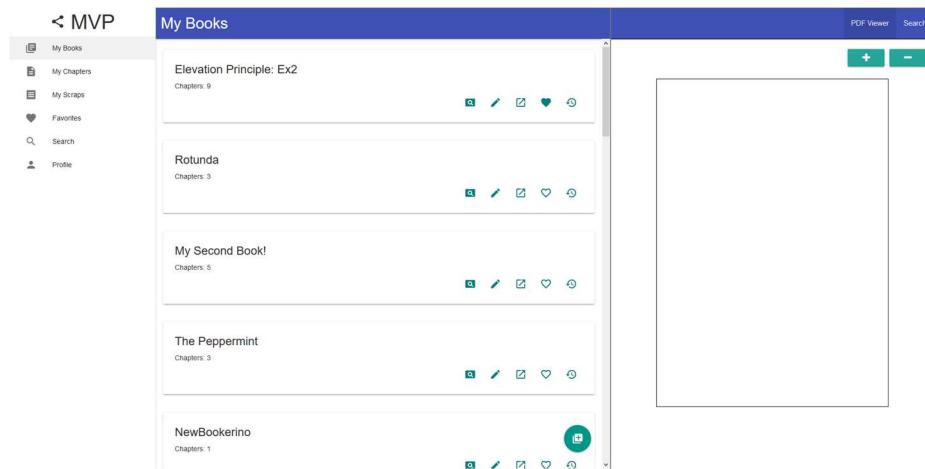
2 New Textbook

Create a new textbook for your text creation or modification to take place.

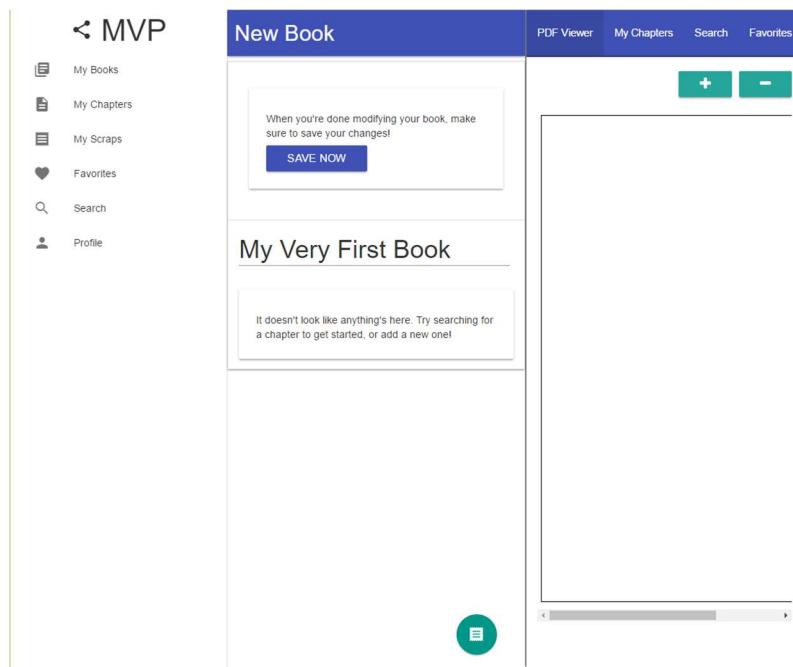
The new textbook operation is a simple press of a button which will prompt you for a new book name.

2.1 New Textbook

1. From the My Books page locate the Create New Book button



2. Click the Create New Book button
3. Input a book name



4. Click "Save Now"

3 Edit Textbook

Edit an existing textbook for your text creation or modification to take place.

The edit textbook operation is a simple press of a button which will allow the user to edit the title of their book, rearrange chapters, search and drag chapters into their book.

3.1 Edit Textbook

1. From the My Books page locate the Edit Book button

The screenshot shows the 'My Books' section of the platform. On the left is a sidebar with links: 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile'. Two book entries are listed: 'Elevation Principle: Ex2' (Chapters: 9) and 'Rotunda' (Chapters: 3). Each entry has a set of small icons at the bottom right. The 'Edit Book' icon for 'Elevation Principle: Ex2' is highlighted with a red box.

2. Click the Edit Book button
3. Options:
 - a. Drag in a new chapter from My Chapters / Searched chapters

The screenshot shows the 'Edit Book' interface for 'Elevation Principle: Ex2'. The main area displays the book title and three chapters: 'Chapter 1' (Scraps: 0), 'Chapter 1' (Scraps: 0), and 'Chapter 3' (Author: stevenbenjaminpowers). A 'SAVE NOW' button is visible above the chapters. To the right is a sidebar titled 'My Chapters' containing a list of scrap titles: 'Testing Toast !', 'asdfasdf', and 'Howdy YakIII d'. Each item in the list has a 'Scraps: 3' note below it.

- b. Rename the book
 - i. Click on the book title to rename the book
- c. Create a new chapter

The screenshot shows the platform's user interface. On the left, a sidebar titled 'MVP' contains links for 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile'. The main area is titled 'Edit Book' and shows a book named 'Example Book' with an author listed as 'joshuatravismatteson'. A 'New Chapter' section is present. On the right, a sidebar titled 'My Chapters' lists several chapters: 'New Chapter' (Scraps: 0), 'New Chapter' (Scraps: 0), 'How They're Made' (Scraps: 2), 'New Chapter' (Scraps: 0), and 'New Chapter' (Scraps: 0). A 'PDF Viewer' link is also visible at the top of the sidebar.

- d. You will be prompted to enter a name for the chapter
- e. For more details see section 4
4. Click "Save Now"

4 Edit Chapter

Edit an existing chapter for your text creation or modification to take place.

The edit chapter operation is a simple press of a button which will allow the user to edit the title of their chapter, rearrange scraps, search and drag scraps into their book.

4.1 Edit Chapter

1. From the Edit Book page locate the Edit Chapter button

The screenshot shows the 'Edit Book' interface. On the left is a sidebar with links: 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile'. The main area has a blue header 'Edit Book' and a message 'When you're done modifying your book, make sure to save your changes!' with a 'SAVE NOW' button. Below this is a chapter titled 'Elevation Principle: Ex2' by 'Chapter 3' (Author: stevenbenjaminpowers). To the right of the chapter title is a set of icons: a magnifying glass, a red pencil (highlighted with a red box), a green arrow, a lock, a square, a heart, and a circular arrow.

2. Click the Edit Chapter button

3. Options:

- a. Drag in a new scrap from My Scraps / Searched scraps

The screenshot shows the 'Edit Chapter' interface. The sidebar is identical to the previous screenshot. The main area has a blue header 'Edit Chapter' and a chapter titled 'Chapter 1'. A message says 'It doesn't look like anything's here. Try searching for a scrap to get started, or add a new one!'. To the right is a sidebar with 'Add New Scrap' and a text area 'Write your new scrap here'. Below it is a note 'This is a new scrap'. At the bottom are buttons for 'BOLD', 'ITALIC', 'UNDERLINE', 'SUBS', 'PARAGRAPH', 'VERTICAL SPACE', and 'N'. There is also a 'UPLOAD IMAGE' button.

- b. Rename the chapter

- i. Click on the chapter title to rename the chapter

- c. Create a new scrap

The screenshot shows two main sections of the platform:

- Edit Chapter:** This section is titled "Edit Chapter" and contains a sub-section titled "Chapter 1". A message says "It doesn't look like anything's here. Try searching for a scrap to get started, or add a new one!" Below this is a search bar and a list of file icons. At the top right are links: PDF Viewer, My Scraps, Search, Favorites, and Edit Timeline.
- My Scraps:** This section is titled "My Scraps" and shows a list of files from a folder named "final_report". The files include:
 - encapsulated Postscript (EPS) files: "cov_v_spot1.eps" and "cov_v_spot1-eps-converted-to.pdf"
 - LaTeX style files: "comment.sty", "finalreport.aux", "finalreport.log", "finalreport.tex", and "finalreport.toc"
 - other files: "finalreport.pdf", "finalreport.gz", "IEETran.bst", and "IEETran.cls"

Both sections share a common header bar at the top and a "Scrap" editor interface on the right side, which includes buttons for BOLD, ITALIC, UNDERLINE, SUBSECTION, SUBSUBSECTION, PARAGRAPH, VERTICAL SPACE, NEWPAGE, and ENDPAGE. It also has fields for "Write your new scrap here" and "Submit your Scrap Tags here, separated by commas", along with checkboxes for "Enable LaTeX?", "Submit Tags?", and a "SUBMIT" button.

- d. You will be prompted to enter either text or upload an image for the scrap
- e. For more details see section 5

4. Click "Save Now"

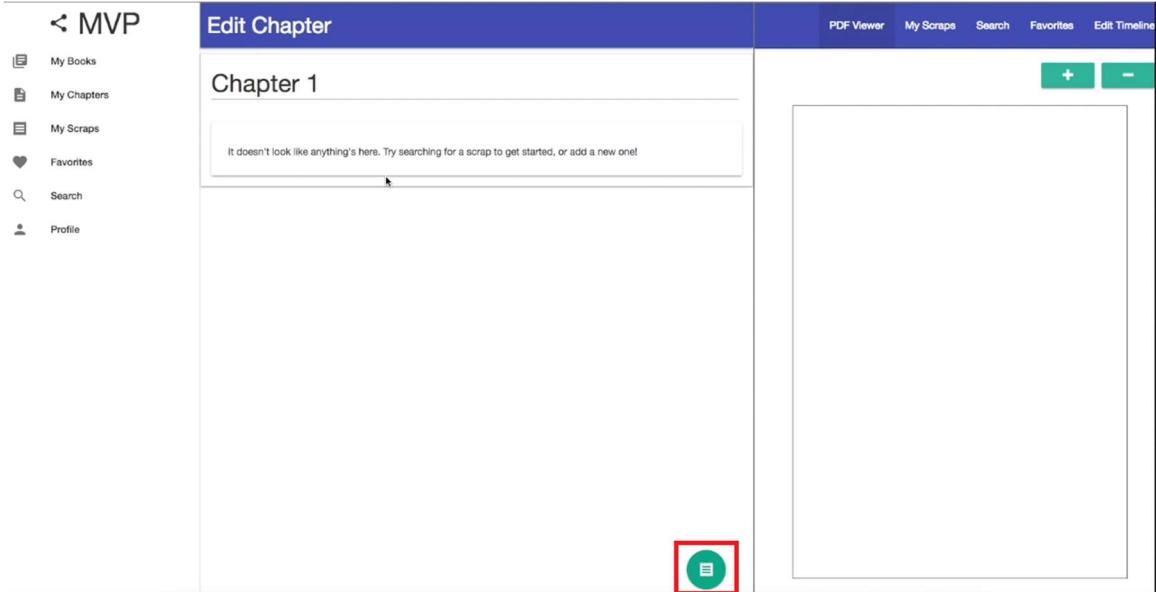
5 New Scrap

Create a new scrap that chapter for your text creation or modification to take place.

The edit chapter operation is a simple press of a button which will allow the user to edit the title of their chapter, rearrange scraps, search and drag scraps into their book.

5.1 New Scrap

1. From the Edit Chapter page locate the New Scrap button



2. Click the New Scrap button

3. Options:

- Create a new scrap using text, enable LaTeX mode, or add tags

The screenshot shows the 'Add New Scrap' interface. At the top, there are buttons for PDF Viewer, New Scrap, and Search. Below that is a title 'Add New Scrap'. A toolbar with various styling options (BOLD, ITALIC, UNDERLINE, SUBSECTION, SUBSUBSECTION, PARAGRAPH, VERTICAL SPACE, NEWPAGE, ENDPAGE) is visible. The main area contains a text input field with placeholder text: 'Write your new scrap here' and 'This is a new scrap that I am creating! I can add `\text{[LaTeX]}`!'. Below the text input is a 'UPLOAD IMAGE' button. There are two checkboxes: one checked ('Enable LaTeX?') and one unchecked ('Submit Tags?'). A text input field for 'Scrap Tags' is present, followed by a 'SUBMIT' button.

- Or create a new image scrap

The screenshot shows the 'My Scraps' interface with tabs for 'ALL SCRAPS' and 'UNUSED SCRAPS'. At the top, there are buttons for PDF Viewer, New Scrap, and Search. Below that is a title 'Add New Scrap'. A file selection dialog is open, showing a list of files from a 'final_report' folder. The dialog includes columns for Name, Date modified, Type, and Size. The selected file is 'finalreport.pdf'. The 'Add New Scrap' form is partially visible on the right, showing the same styling toolbar and text input fields as the previous screenshot.

4. Click "Save Now"

6 Profile Settings

Profile settings is a way to alter the name that appears on compiled documents or logout.

The profile settings page allows the user to view their account information, modify the name that appears on their compiled documents, or logout for another

6.1 Profile Settings

1. On the left navigation, locate the Profile Settings button.

2. Click the Profile Settings button
3. Options

- a. Modify your display name and click "Submit" or logout by clicking "Logout"

Username (You are unable to change this)

stevenbenjaminpowers

Display Name (How your name will appear on publications)

Steven Powers

SUBMIT

Logout

If you would like to logout, click the button below.

LOGOUT

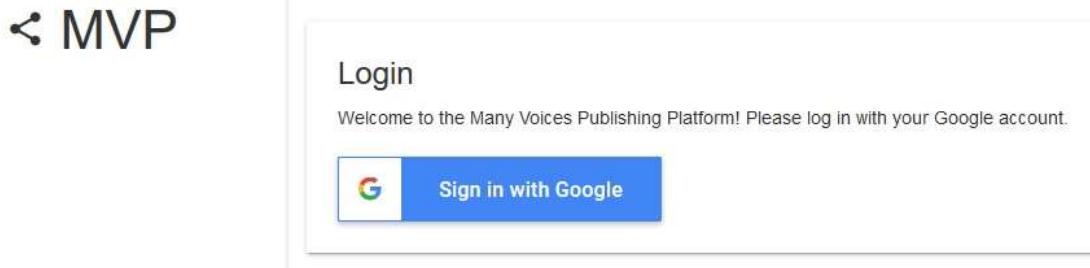
7 Login

Login functionality is an essential part of any web application.

The profile settings page allows the user to view their account information, modify the name that appears on their compiled documents, or logout for another

7.1 Profile Settings

1. When a user navigates to the website for the first time, or when no login is saved the user will be prompted with the following.



2. Click the "Sign in with Google" button
3. You will be prompted to select the account you want to use or you can create a new account with Google.



- a.
4. Once logged in, you will be presented with the MyBooks view.

Many Voices Publishing Platform

USER GUIDE

If you encounter issues not addressed by this user guide, please contact your account manager for additional support.

1 CODE EXAMPLES

The backend is split into two primary parts. The first, and most difficult part of the backend, is the interface library being used to store objects into the git repositories and to get them back out. Shown below is a code sample, which is fully working today, showing a simple script that creates new scraps, chapters, and books, and gets a valid latex document to render from them. Below are two interesting pieces of backend code that allow the user to have a seamless experience combining different scraps, chapters, and books together.

```
1  var a = new scrap.Scrap('First_sentence_of_a_chapter', 'rambourg');
   await a.save('initial_save');

5  var b = new scrap.Scrap('Second_sentence_of_a_chapter', 'rambourg');
  await b.save('initial_save');

  var ch1 = new chapter.Chapter('First_Chapter_Name', 'rambourg');
  ch1.addScrap(a);
  ch1.addScrap(b);

10  await ch1.save('initial_save');

  var c = new scrap.Scrap('First_sentence_of_another_chapter', 'rambourg');
  await c.save('initial_save');
15  var ch2 = new chapter.Chapter('Another_Chapter_Title', 'rambourg');
  ch2.addScrap(c);
  await ch2.save('initial_save');

  var myBook = new Book('My_Fav_Book_Title', 'rambourg');
20  myBook.addChapter(ch1);
  myBook.addChapter(ch2);

  var bookText = await myBook.getText();
```

Another piece of interesting code that we want to highlight is the creation of a new scrap. In the code example below, we determine whether this new scrap is going to be appended to an existing chapter, and if so, we need to first return and collect all of the existing scraps associated with that chapter. Once we have the new list of scraps, we need to update the chapter object with the new information.

```
1  var scr = new scrap.Scrap('Brody', 'abc123-def456')

  /* reconstitute a chapter from author and id */
  var ch = chapter.reconstitute('Patrick_Rambourg', '56c2c2ee')
5   var newCh = await ch.fork('Brody')

  newCh.addScrap(scr)
  await newCh.save('add_new_chapter')

10  submitNewScrap() {

  if (this.selectedFile) {
    this.submitImageScrap();
    return;
  }

15  }

  var requested = this.userText;
  var enableLatex = this.enableLatex;
  var theAuthor = Cookies.get('username');
  var theChapter = this.chapters[1];
20  var authToken = "Token_" + Cookies.get('token');

  var submitTags = this.submitTags;
  var submittedTags = this.submitTagsText;

25  if (theChapter === undefined || theChapter === null || theChapter === "") {
    if (submitTags) {
      submittedTags = submittedTags.toString();
      submittedTags = submittedTags.replace(/(^,)|($)/g, "");
    }

30  submittedTags = submittedTags.replace(/\s/g, '');
  submittedTags = submittedTags.split(",");
  }

35  let request = {
    author: theAuthor,
    text: requested,
    latex: enableLatex,
    tags: submittedTags
  };
}
```

The next code listing to be shown is the new scrap creation process. This is a particularly interesting section of code because it determines what to do for the user. Based on what page the user's request originated the new scrap function will either simply create the scrap and place it within the user's MyScrap page. If the user's request originated from the edit chapter page, it will dynamically build the list of all existing scraps and their ordering, and then append the new scrap and submit the new list to the database.

```
1 //CREATE THE NEW SCRAP AND GET THE NEW SCRAP ID
2 var scrapID = '';
3 httpClient.fetch('https://remix.ist/scraps/new', {
4   method: 'post',
5   body: JSON.stringify(request),
6   headers: {
7     'Content-Type': 'application/json',
8     'Authorization': authToken
9   }
10 })
11 .then(response => response.json())
12 .then(data => {
13   var scrapID = data.uuid;
14   this.toast.show('Scrap saved successfully!', 5000);
15   this.ea.publish('new-scrap', data);
16 });
17 } else { //The Chapter Was Provided, so create the scrap and then add it to a chapter.
18   if (submitTags) {
19     submittedTags = submittedTags.toString();
20     submittedTags = submittedTags.replace(/(^,)|($)/g, "");
21     submittedTags = submittedTags.replace(/\s/g, '');
22     submittedTags = submittedTags.split(",");
23   }
24
25   let request = {
26     author: theAuthor,
27     text: requested,
28     latex: enableLatex,
29     tags: submittedTags
30   };
31
32   //CREATE THE NEW SCRAP AND GET THE NEW SCRAP ID
33   var scrapID = '';
34   httpClient.fetch('https://remix.ist/scraps/new', {
35     method: 'post',
36     body: JSON.stringify(request),
37     headers: {
38       'Content-Type': 'application/json',
39       'Authorization': authToken
40     }
41   })
42   .then(response => response.json())
43   .then(data => {
44     var scrapID = data.uuid;
45     this.toast.show('Scrap saved successfully!', 5000);
46     this.ea.publish('new-scrap', data);
47   });
48 }
```

```

40      }
    })
    .then(response => response.json())
    .then(data => {
      var new_scrap = data;
      var scrapID = data.uuid;
50
      this.scrapss = [];
      var test = [];
      //GET CHAPTERS SCRAPS
      httpClient.fetch('https://remix.ist/chapters/' + theAuthor + '/' + theChapter)
      .then(response => response.json())
      .then(data => {
        var scrapss;
        this.scrapss.push(data);
        scrapss = data.scrapss;
55
        var newScrapRequest = [];

        for (var i = 0; i < scrapss.length; i++) {
          newScrapRequest.push([scrapss[i][0], scrapss[i][1], scrapss[i][2]]);
        }
        newScrapRequest.push([theAuthor, scrapID, null]);
        let request2 = {
          scrapss: newScrapRequest
        };

        // UPDATE CHAPTER WITH NEW SCRAPS
        httpClient.fetch('https://remix.ist/chapters/' + theAuthor + '/' + theChapter, {
70
          method: 'post',
          body: JSON.stringify(request2),
          headers: {
            'Content-Type': 'application/json',
            'Authorization': authToken
          }
        })
        .then(response => response.json())
        .then(data => {
          this.toast.show('Scrap saved successfully!', 5000);
          this.ea.publish('new-scrap', new_scrap);
        });
      }
    }
  }

```

2 PROJECT IMAGES

This section is dedicated to showing some of the progress that has been made through iteration and user feedback. Starting with our low level prototypes, medium fidelity prototypes, high fidelity prototypes, and finally our finished product. The designs were influenced by user feedback throughout the development process.

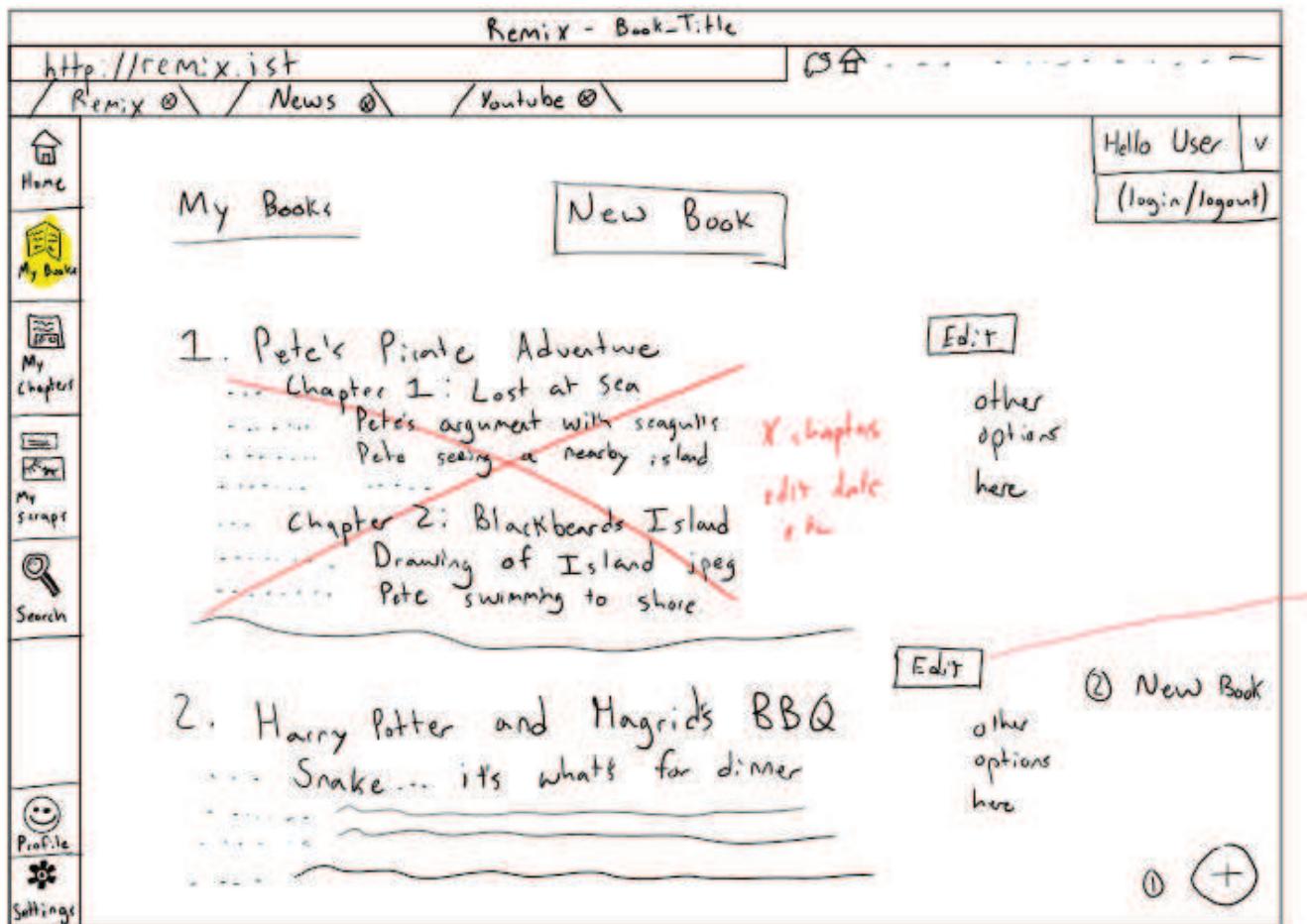


Fig. 1. Here is one low fidelity prototype that was shown to our client. Dr. Jensen suggested removing the preview of chapters and scraps to make the interface a little more simple.

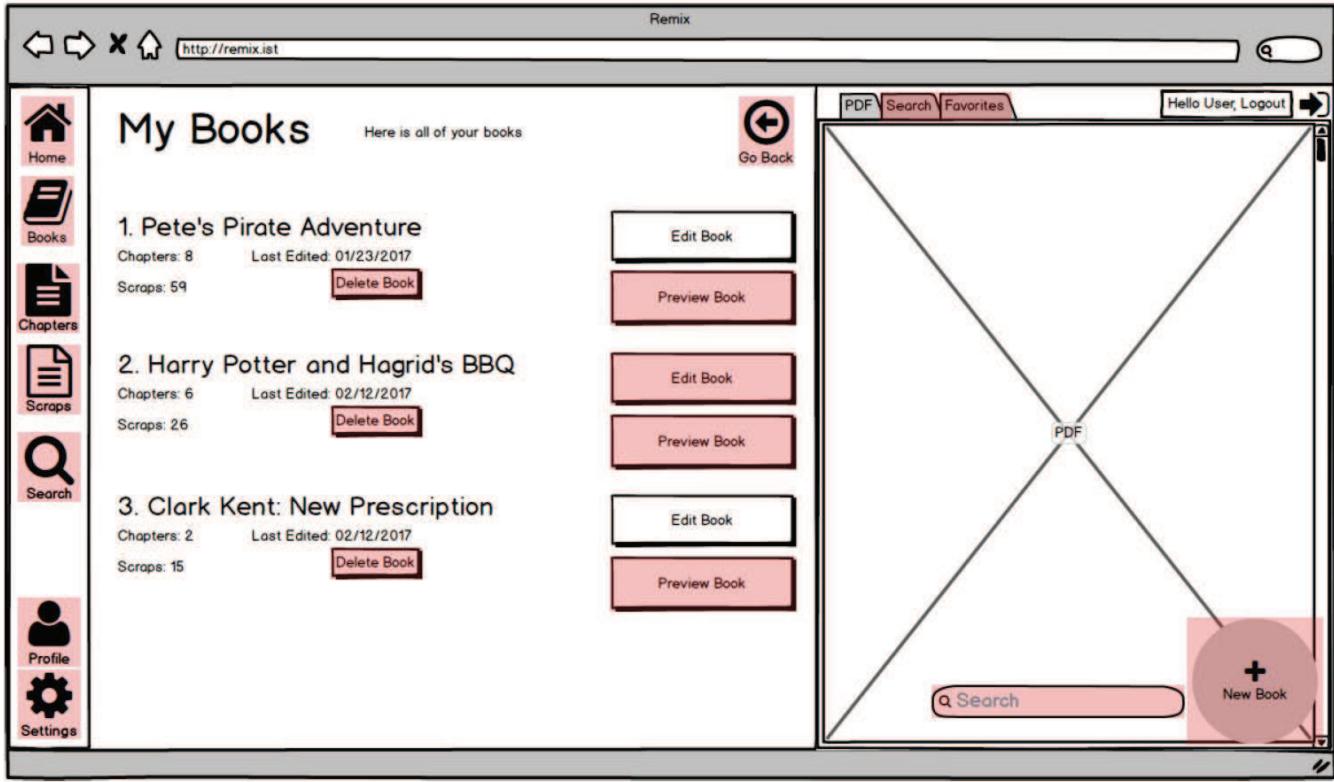


Fig. 2. Here is one medium fidelity prototype, created in balsamiq, that allowed for users to provide feedback about the basic ideas of placement without worrying about finalized plans.

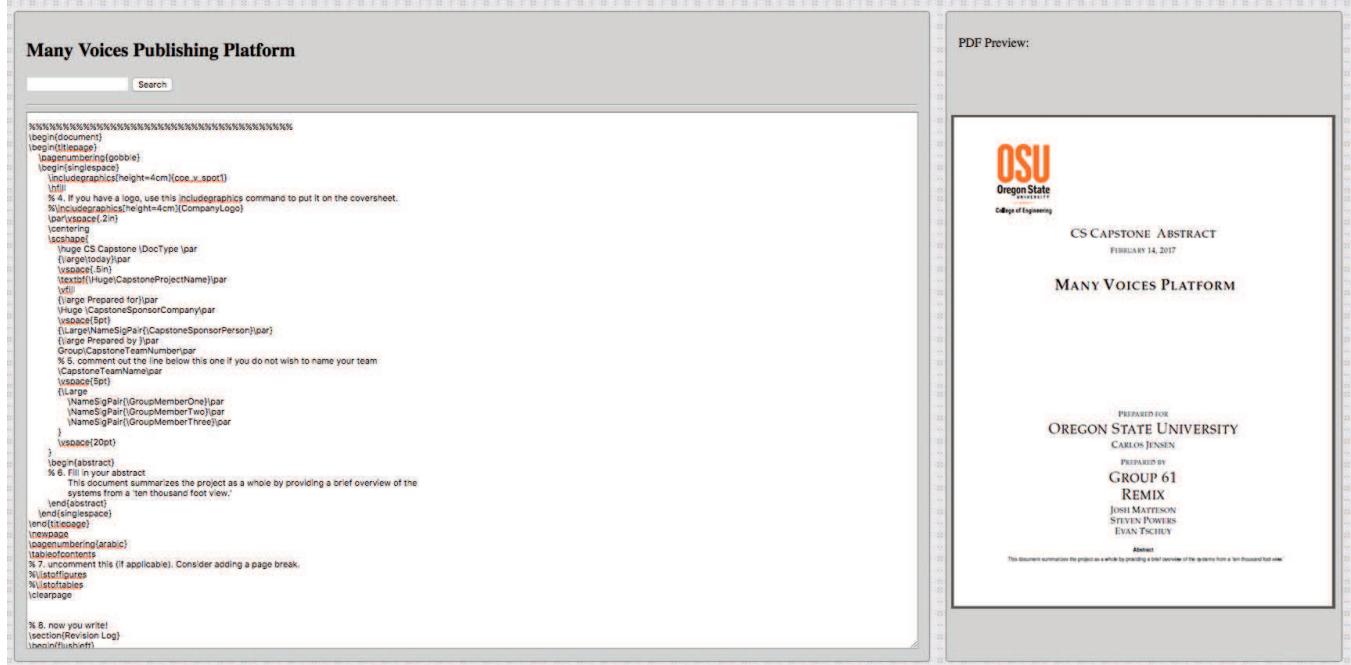


Fig. 3. Here is one high fidelity prototype. This was created using basic web HTML and our Aurelia framework and the use of a hard coded PDF to display a document. This was used as a learning exercise.

< MVP

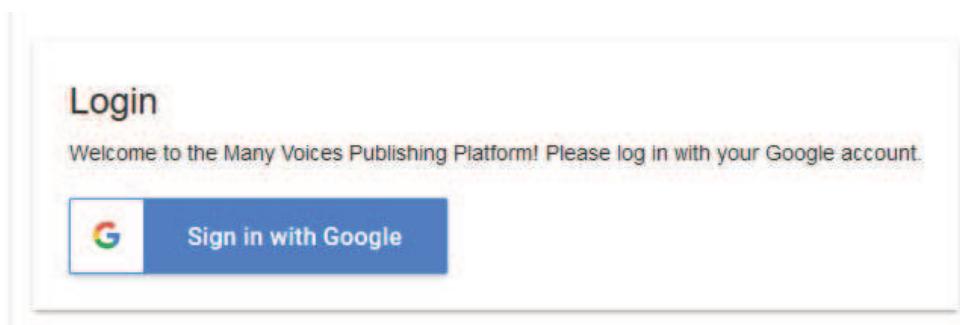


Fig. 4. Here is the final product login page, which uses Google OAuth to abstract our need for security.

< MVP

The image shows the 'My Books' page of the platform. The header includes the '< MVP' logo, a sidebar with links for 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile', and a top right corner with 'PDF Viewer', 'Search', and '+/-' buttons. The main content area displays five book entries in a grid:

- Elevation Principle: Ex2**
Chapters: 9
- Rotunda**
Chapters: 3
- My Second Book!**
Chapters: 5
- The Peppermint**
Chapters: 3
- NewBookerino**
Chapters: 1

Each book entry has a set of small icons below it, likely for editing or deleting.

Fig. 5. Here is the final product my books page, showing sample textbook data.

The screenshot shows the 'My Books' page. On the left, a sidebar menu includes 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile'. The main area displays five book entries:

- Elevation Principle: Ex2**: Chapters: 9. Preview shows a table of contents.
- Rotunda**: Chapters: 3. Preview shows a table of contents.
- My Second Book!**: Chapters: 5. Preview shows a table of contents.
- The Peppermint**: Chapters: 3. Preview shows a table of contents.
- NewBookerino**: Chapters: 1. Preview shows a table of contents.

On the right, there is a 'PDF Viewer' section with a '+' button and a search bar.

Fig. 6. Here is the final product my books page, previewing a textbook.

The screenshot shows the 'Edit Book' page. On the left, a sidebar menu includes 'My Books', 'My Chapters', 'My Scraps', 'Favorites', 'Search', and 'Profile'. The main area displays the chapters of the book 'Elevation Principle: Ex2':

- Chapter 3**: Author: stevenbenjaminspowers. Preview shows a table of contents.
- My Newest Masterpiece**: Author: stevenbenjaminspowers. Preview shows a table of contents.
- Chapter 2**: Author: stevenbenjaminspowers. Preview shows a table of contents.

On the right, there is a 'PDF Viewer' section with a '+' button and a search bar.

Fig. 7. Here is the final product edit books page, allowing the user to edit chapters within a book.

Fig. 8. Here is the final product edit books page, showing the use of drag and drop from a user's chapters.

Fig. 9. Here is the final product new scrap page, showing how easy it is for a user to create a new scrap and even enable more advanced features by using LaTeX..