Oregon State University
College of Engineering

CS Capstone Progress Report

February 27, 2017

Many Voices Platform

Prepared for

Oregon State University

Carlos Jensen _____ _____
Signature *Date*

Prepared by

Group 61

Remix

Josh Matteson _____ _____
Signature *Date*

Steven Powers _____ _____
Signature *Date*

Evan Tschuy _____ _____
Signature *Date*

Abstract

This document summaries the progress that the Remix team has made on the Many Voices Publishing Platform for the client Dr. Carlos Jensen. Additionally this document provides a week by week summary of work performed, as overviews of progress that has been made throughout Winter Term.

CONTENTS

1 Revision Log

Name	Change Number	Date	Description of Change
Steven Powers	1	2/17/2017	Finalized Progress Report for half way point through winter term. Fixed spelling mistakes, added code listings, adjusted content, added images.

2 Project Purpose

A modern textbook is updated frequently, perhaps even yearly, and can cost in the range of hundreds of dollars. Students are often left to attempt to understand poorly worded, even incorrect information from a textbook often chosen from those sent to a professor for review by the publisher. This can lead to better works with less aggressive sales tactics not being made available, or even known. Another choice would be for a professor to write their own textbook. However, this is a process that takes months of endless research and time spent, and on top of that will require peer review and publishing before it can be released.

The Many Voices platform offers to put an end to this massive, slow, expensive cycle. Instead of a textbook being a single document written by one professor, we seek to re-imagine the textbook as instead a collection of content written by professors from around the world that are useful for a particular class. A knowledgeable professor can contribute a few chapters on their specialty, without needing to write an entire textbook around it.

Professors wishing to use this content can then modify it for their uses in the classroom. The material will be hosted in such a way as to provide the ability to ”fork” content, or create content based off of it. The platform will provide a way to search for and find content, prioritized by relevance and credibility as determined by other users; the most popular material will be shown with the most prominence.

3 Winter Term Weekly Summary

Steven:

Here is our previous term Positives, Deltas, and Actions chart which we felt is still important to our progress report for Winter Term as it shows some additional history to our project development.

Week	Positives	Deltas	Actions
3	Got the initial draft of the problem statement done after meeting last week with our client.	Were unable to set up a meeting with our client, and thus had no way of getting feedback on the draft before submission.	We will talk to our client via email about possible meeting times, either in person or remote, that may possibly work for everyone.
4	Figured out a time for meeting with our TA.	Need to hear back from our client.	We will wait another day or two and then send a new polite email.
5	Finished revising the problem statement, got it signed. Began the requirements document.	N/A	N/A
6	Revised Requirements document via feedback from TA. Got client signature on Requirements. Attended writing workshop.	Need to be on top of things regarding contacting our client as he is difficult to contact.	We will send him another email asking for feedback on the requirements.
7	Began technology review. Revised requirements document for possible re-grading.	Need better team work schedule to get times to work on documents together rather than separately.	We will discuss possible extra meeting times in addition to our weekly meeting with our TA where we can work for several hours uninterrupted.
8	Finished technology review (after getting extension to Wednesday).	Need to be on top of documents to finish them well in advance of due dates!	We will personally push ourselves to get work done sooner rather than later.
9 & 10	Spent Thanksgiving Week getting rested. Finished design document. Finished progress update document/finished presentation.	Need to get client signature for design document!	We will be emailing Carlos. With the new term we hope to set up a weekly meeting with him.

4 Steven Term Progress

4.1 Weekly Recapping

For this first week we were getting back into the groove of things after Winter break. Josh and I were able to meet with Carlos the week prior and talked about arranging meetings going forward each week. We weren't able to get a whole lot of work done this week, but we planned for meeting each week at least a few hours a couple days of the week so we can continue making progress. We planned to meet next week and work on the framework and user stories.

The second week of the term we began our weekly meetings, though our client had to cancel meeting for the first two weeks. We also setup our meetings with Jon, which was actually pretty difficult to find a time that worked for all four of our schedules. We plan to continue working on the User Stories and Framework so we are able to make progress towards a working prototype.

The third week we were to begin our weekly meetings with our client. Our client recently transferred positions and our weekly appointment was unfortunately lost. This resulted in our weekly time slot being given away, so we had o find a new time that worked for all of us. We decided for 9:00AM Mondays, which was unfortunate because I had planned on sleeping in these days. Our client also wanted to instead meet biweekly rather than every week. We talked over the past terms documents and possible upcoming documents that we would be seeing in the class and that we should be focusing on development throughout this term. During this week we also worked on the backend system and testing of the include and input commands with our setup which is working well at at this time. I looked into security features of LaTeX to protect our LaTeX documents from succumbing to common exploits which include (escaping the document and accessing the shell). Josh and I also worked on getting the framework up and running (TS Lint was giving us issues so we ended up switching to the ES Next variant of JavaScript).

On the fourth week of the term we met with Dr. Jensen and discussed the backend of the system as well as some plans for how to model the front end of the system. Dr. Jensen suggested using low fidelity / medium fidelity to receive feedback from users instead of focusing on getting to high fidelity and having users review that as they will be less critical of the whole system and instead focus on pixel alignment. Josh and I need to get more work done on Aurelia and get a basic user interface up and running and try working towards in page rendering of the resulting PDF from the back end system. I began working on some prototypes using a two column view, with our PDF rendering taking place to the right of the working area.

The fifth week of the term was a week we had off from meeting with Dr. Jensen, we used this time to work on developing some of the required features that we need. This included Josh and I working on the PDF in browser rendering with Aurelia. Josh finished this task up separately. I worked on Prototype development to have Dr. Jensen review and provide feedback on during next weeks meeting. I wasn't able to get as many prototypes drawn as I would have liked, but there is still a good chunk of design that Dr. Jensen could critique. Next week I planned to show Dr. Jensen our designs and take in any feedback he might have. I also plan to work on revising all of our documents and get them placed into our OneNote document that I have prepared for our group.

The sixth week of the term, the week that this progress report was due. We met with Dr. Jensen and he had lots of feedback about our designs, mainly in ways to simplify the interface for the users while also making it easier to develop. We talked about rendering the PDF only when needed (as in active editing) instead of constantly. We also talked about the possibility of using a tabbed system to only render when the user wants to, which will reduce our server calls.

I have continued working on our document revisions, as well as our Progress Report materials (video, presentation materials, and report), all of which are currently in near final forms as I write this. Something that will be better next week is availability, as this week has a lot of midterms for various classes. Josh and I plan to integrate our PDF viewer into our application and work on making progress on other areas of the application.

4.2 What's Left on the Frontend

Josh will go into greater detail on some of the below items in his section, but I will briefly list and expand upon those that he does not.

1) **Integration of PDF Viewer and Application**
   As of right now we have a functioning PDF viewer built with a special branch of PDF.JS that is made for Aurelia. We have implemented a separate PDF viewer page at the moment, and need to integrate this into our main application.

2) **Finalizing Layout Design & User Testing**
   We met with our client earlier this week with a few layout prototypes and were able to get good feedback on our designs, as Dr. Jensen is wanting to write a book, his feedback is valuable, though he said that we might come up with something better, or have other users desire something else. We plan to iterate on our designs post feedback and create medium fidelity prototypes with a tool similar to Balsamiq, a prototype editor.

3) **Working Scrap Editor**
   This item is one of the most important features, as the user will be editing all of their work so it is important for us to provide a familiar interface to what a user might expect. Formatting isn't a requirement for this application, but it is something we can hopefully integrate as a stretch goal. With this in mind, a rich text editor might be beyond what we need.

4) **Book/chapter/scrap view**
   One of the elements of feedback received from our client is to make the user interface simpler for the user, so being able to see what chapters are listed under a book, and what scraps are listed under each given chapter. Depending on how we choose to implement this, it might prove difficult to have something that is appealing to our users and efficient for our application.

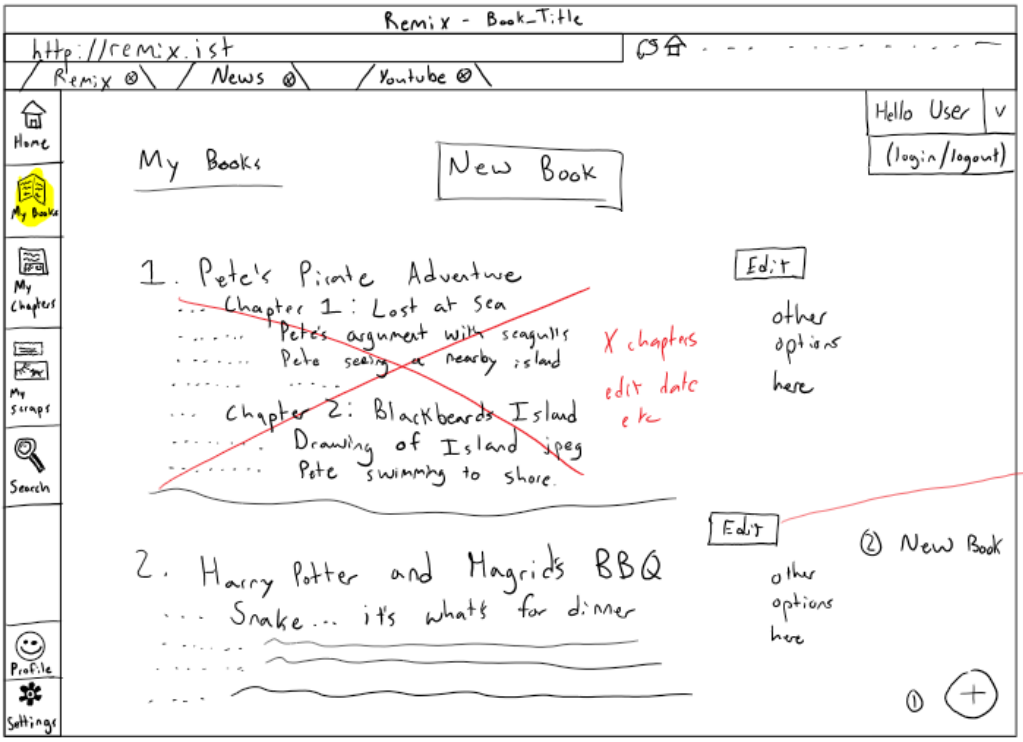

Fig. 1. Here is one low fidelity prototype that was shown to our client. Dr. Jensen suggested removing the preview of chapters and scraps to make the interface a little more simple.

5 Evan Term Progress

5.1 Weekly Recapping

For the first week of the term, we were mostly catching up on what we had done over the break. Steven and Josh had managed to meet with Carlos, which was great to arrange future meetings and to get his opinion on where we were going with the project. After a term of lower levels of contact than would have been preferable, it was good to get this initial meeting set up and done with no hassles.

Over the break, I began work on the book management backend library. This is the foundation of the entire project – the part that will do the actual work interacting between the git repositories and JavaScript objects with convenient APIs. In the second week, we got meetings set up with Jon, our TA. In the mean time I wrote a basic preliminary version of the API specification for the library. Barring any issues I ran into the plan was to implement this for the next several weeks.

The third week was to be our first meeting with Dr. Carlos Jensen. Unfortunately, due to a calendar migration relating to his change in job position, he had missed our appointment scheduling and overbooked. We were luckily able to work out a time to meet the next week with him. Week four was the first week where we were able to meet with Dr. Jensen as a group. We go to talk about the state of the backend, including its technical document storage design, and the frontend. For the frontend, Dr. Jensen asked us to draw some rough prototypes. Steven is planning on taking the lead with that.

We were unable to meet with Dr. Jensen during week five, as with his calendar migration reset we changed our meetings to every other week. Instead, we used that week to polish features we'd been working on already. For instance, I finally got textbook rendering working. It was a pain to get, because it involves descending into chapters and scraps (all asynchronous calls in JavaScript), but with it working now that is a major milestone out of the way. At our next meeting, Monday of week six, we primarily went over our frontend prototype designs with Dr. Jensen to get his feedback on user flow, interface direction, and the like. We were able to refine our vision for the interface, with his main feedback stressing making it simpler, and making it less duplicative. We now have an idea around how the user will go about editing a scrap, editing a chapter, etc., and when in the writing process to start rendering PDFs for layout preview.

This week, week six, has been all about capstone assignments – the revised documents, the video summary, and of course this progress report document.

5.2 What's Left on the Backend

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 tex document to render from them.

```
var a = new scrap.Scrap('First sentence of a chapter', 'rambourg');
await a.save('initial save');

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);

await ch1.save('initial save');

var c = new scrap.Scrap('First sentence of another chapter', 'rambourg');
await c.save('initial save');
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');
myBook.addChapter(ch1);
myBook.addChapter(ch2);

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

We can create a new scrap with the scrap.Scrap constructor. We simply have to pass it the text to include in the scrap, and the author. This allows us to simply have some sort of API call that provides these two things and pushes them straight into a scrap. It's necessary to save an object every time. After creating the scraps we want, it's then time to create a chapter. It's worth noting that normally, a user would more likely create a book first, then a chapter, then some scraps, but since scraps, chapters, and books can exist freely from each other – a scrap doesn't always need to be associated with a chapter, for example – it's easiest to demonstrate here by creating them in the order we associate them in.

After creating the chapter and saving it, we can another create chapter as wanted, a new scrap, et cetera. We can then just add these to a book by calling the addChapter method. When we're done and want to get the text of the book in its entirety, we call the getText method. What this method does is, more or less, recursively read all chapters, and then all of their scraps, and then reconstruct the final latex file by concatenating the values of the scraps.

Let's take a closer look at the scrap object.

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

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

newCh.addScrap(scr)
await newCh.save('add new chapter')
```

We have today a working implementation to load chapters, scraps, and books from disk, modify them, and save them. As shown here, it's as easy as calling "reconstitute" with the author's name and the object's ID number. In the future, another user will be able to fork an object, which will create a copy of the object in their space that they can modify. They'll then be able to do things like add scraps to a chapter, chapters to a book, modify scraps, et cetera. It's worth noting that forking is not currently implemented. However, we do not anticipate any problems with its implementation, as it integrates easily into the existing git backend.

So what is currently left on the backend? As mentioned before, we have two main parts. The library is the unique part of the project, the part that controls the actual object storage, modification, forking, and saving. Of these functionalities, the only parts left are forking, and getting previous versions. Getting a previous version is around 80from the disk and retrieve the old version. It is now only a matter of creating a new object from this old version and returning it.

The other main part of the backend is the API. This API is a simple app written in Node that wraps the backend library in web-available JSON requests. The API is going to be the easier of the two pars to implement, as I already have ample experience creating web APIs, including in NodeJS. The API will include user authentication, allowing us to control who modifies which objects, and keep track of the owner.

In addition, we will be needing some kind of search backend. We have yet to develop our exact strategy for search, but will most likely be using a prebuilt search backend and simply loading objects into it at save time. This will allow for easy referencing back and forth between search results, which will return the object id and its owner, and the object stored on disk, which is referred to by its id and owner.

6 Josh Term Progress

6.1 Weekly Recapping

Towards the very end of winter break, we were able to get a meeting with Carlos to clear up some possible fallout from the previous term. While we were able to get confirmation on some of the technologies we were using, we weren't able to sort out some questions on the backend because Evan is the main group member handling that side of the application. We were able to, however, figure out that we could meet with him again the following week after he moved into his new office. This meant we weren't able to meet with him for at least two weeks, however.

After our meeting, we decided to delegate some time to developing a game plan of how we could divide and conquer the majority of it. The following week, our whole group managed to meet and accomplish setting up the basic architecture blueprints for the application. This meant that Steven and I would forward with Aurelia, and do everything we could get a working version up and running as soon as possible.

Our client ended up needing more time to delegate to other task on hand, and this meant switching our meeting frequency to a biweekly basis. This hasn't caused for any problems, and actually allows for us to bring more deliverable content to our meetings. Two week springs tend to be more ideal in that regard, however, we'll be working a little less close with our client in the future because of this.

While the backend had been progressing with Evan, Steven and I focused our time trying for a semi working version of a local Aurelia project that would run in browser. It wasn't too difficult to follow an example online and get an express version of the application up and running. Our group was then able to meet that week to briefly discuss progress and impediments.

During our 14th week, we finally managed to make some real progress and gain momentum, as we were able to meet with Dr. Jensen. This led to discussion about work flow designs with a desired low fidelity goal. This outlined adequate deliverables for our next meeting two weeks from then, and gave us expectations as to what our client wanted to see from us. During the meeting, Steven and I managed to demo a fully working version of a basic Aurelia project to Dr. Jensen. This was useful to to keep him updated with our current progress. After the meeting, we moved forward with the goal of having a few basic layouts prepared, as well as some further technology integration.

Steven and I started to work on a PDF viewer that we could integrate into our application, and managed to find an online guide that would walk us through setting one up with Aurelia as our framework. While we managed to get a PDF viewer running on a separate instance, Steven and I learned quite a lot from the experience and furthered our knowledge on working with Aurelia in general. We had our meeting with Dr. Jensen later this week, and we wanted to have this accomplished going into our meeting. From this point, Steven worked on prototypes for the layout design and I branched off into finishing the pdf viewer demo.

Our next meeting with Dr. Jensen went very well, he gave us constructive feedback on our low fidelity prototype designs and we were able to demo the front of the application to him. Our group also started work towards our progress report,

Which will slightly hinder the time dedicated to pushing our application forward.

6.2 Overview

The first real development our team started to work on was the user interface. We have already dedicated a fair amount of time to this section, as the user interface is one of the most integral parts of our application, and to any application for that matter. Studies have shown that the average user will stay on a new website for as little as one minute, which means that making the user interface appealing to the user as well as easy to use is crucial to the overall success.

Our goal was to start on this a couple of weeks before winter term started, that way we could get a good lead and gain momentum. However, previous impediments had led us to wait on this aspect, this way, however, we could get insight into what our client had envisioned for this part of the application. We're planning to be working on this for around 7 weeks total, which should give us plenty of time to move forward on this major part of the project. More than just these 7 weeks, we'll be adding continuously to the U.I. as our application continues to grow.

6.3 What's Left on the Frontend

For this next section, I'll briefly go over some of the future implementations and tweaks that I'll be taking on personally or with the help of Steven.

1) **Integration of PDF Viewer and Application**
   We currently have a working instance of an Aurelia project with a basic front page layout, as well as a different project instance that contains the PDF viewer. While both of these seem to work well on their own, integrating them together will take some time and massaging in order to obtain the desired functionality. The pdfviewer is an essential part of the application, which means that this needs to be integrated in such a way that it won't cause difficulty when trying to work with it later on.

2) **Finalizing layout design**
   Last week, we went into our client meeting with a few layout prototypes, and with this we were able to narrow down what was expected. We have the basic layout, and our next still will be taking it from a low/mid level fidelity prototype to a high fidelity layout.

3) **Working scrap editor**
   This section will take a little bit more work, as the combination between backend and frontend will merge in some aspects. On the frontend, this won't be such a challenge. We need to either find an open source rich text editor/ or build one from scratch. Making one from scratch would be more time consuming, but we would have more control and knowledge as far as how it works. This will look like a word document or any other basic text editor or formatter. Seeing as this is one of the more important aspects of our application, we're approximating it to take around 4 weeks. If it ends up taking more than this, we'll delegate more time appropriately.

4) **Book/chapter/scrap view**
   This part might be more complicated. Seeing as we're a single page application, we might have to look into

reformatting a good amount of the front end design. Implementing a router would be the only way to ensure multiple page views, but shouldn't cause too much drawback.

7 Current Project Status

Steven:
Currently our project is a little behind schedule. We are attempting to make up lost time by meeting more frequently and having sprints for given tasks. We have updated our gantt chart that has moved up various tasks so that if we take longer on a given task that it will have started sooner as well. We have revised all of our written documentation, reflecting any changes that we have made after they were written. We are still partially in the planning phase for both the frontend and backend systems and continue to iterate on ideas as we progress through the term.

Our next few goals are to have an integrated PDF viewer within our main Aurelia application, backend and frontend integration so we can begin working on sending and receiving information, and making more progress with User Testing and associated prototypes.

We will continue with weekly meetups, work individually, or in groups so we will be able to progress even when one individual team member is blocked on either something relating to the project, or on other work.

8 Conclusion

Steven:
The Many Voices Publishing Platform has been a great project to work on, bringing each team member outside their comfort zone. A lot of planning had taken place over the previous term, sometimes resulting in shifts in direction of how to manage and develop the platform. Now that we are in Winter Term, the Remix team feels more comfortable as we develop the platform. Part of this comfort comes from the the biweekly our client, and weekly meetings with our TA Jon Dodge who is providing us guidance as we move onto different project tasks and class requirements.