Project Stratus
Sprint 2 Deliverable

Team P
Kyle Van Landingham, Alec Schule, Alec Fong, Andrew Zolintakis, Noah Bergman

Table of Contents

1. Executive Summary

This document describes the development process followed by team P through the second sprint of the project. This includes a review and retrospective of the first sprint, especially focussing on how the team will address issues brought up during the sprint 1 demonstration. Following that are the product backlog and sprint backlog sections showing which requirements the team will aim to complete in this phase of the project. Finally, this document describes extreme programming practices used during the sprint, and provides evidence that those practices were followed by the team.

2. Overview

2.1 Purpose

The purpose of this document is to track and document the work planned and completed in the second sprint of the project.

2.2 Intended Audience

The intended audience for this document is the product owners and the development team.

2.3 References

Reference	Relevant Information
GitHub Repository	https://github.com/kvanland/csci310TeamP
Product Backlog Excel Sheet	https://docs.google.com/spreadsheets/d/1IrTDg63ZMwNC10liWfWb5Cs50-KmELNTij1JuijRXHE/edit?usp=sharing

2.4 Definitions

Term	Definition
Product Backlog	A prioritized list of features or requirements for the system to have some time during development.
Sprint Backlog	A sublist of the Product Backlog that contains the features or requirements that the developers will tackle during the current sprint.

3. Sprint Review

Sprint 1 ended with a demonstration to the stakeholders of development progress so far. At that meeting, several issues were brought up that the team must fix within the next two sprints. The issues brought up were:

- Stop words should not be displayed in the wordcloud.
- The wordcloud generates far too slowly. A wordcloud made from 100 articles should generate in a reasonable amount of time.
- The status bar displaying the wordcloud generation progress must be a rectangle.
- The currently written white box tests do not test the progress of the status bar.

In addition to those issues with already-implemented requirements, it was revealed that the team failed to implement as many features as would be desirable by the end of sprint 1. The team only received 15/150 points for progress made as of the end of sprint 1. The team should aim to accomplish far more in sprint 2 in order to re-establish some semblance of a sustainable pace.

4. Sprint Retrospective

The team discussed that they should elaborate more during the stand up meetings and document the meeting responses immediately instead of at the end of the meeting. Another improvement to the stand ups would be more concrete reasoning for the tasks taken up between the meetings with respect to sustainable pace. It was also discussed that a guide to the documentation should be created to keep track of all the documentation needed for each pair programming session.

5. Product Backlog

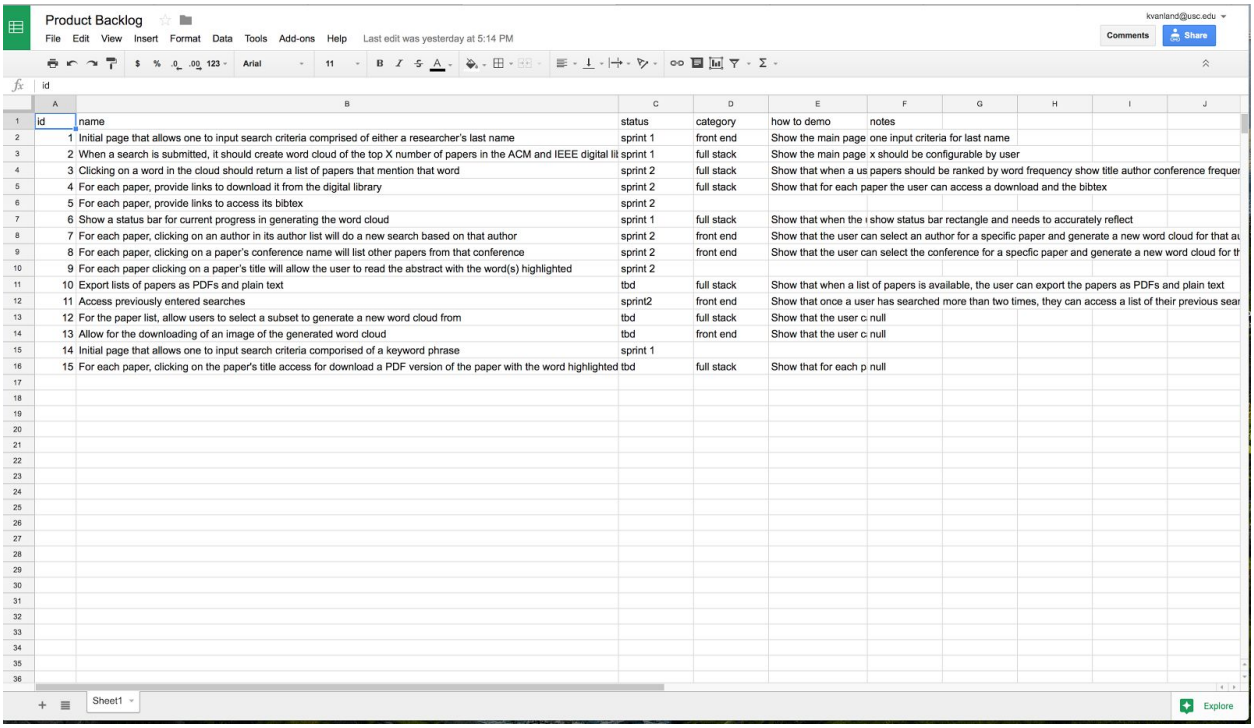
The ordering of the Backlog reflects the priority of each feature where number one is the highest priority and fifteen is the lowest priority.

Project description: Web application to display word clouds associated with researchers' papers that have been published in the ACM or IEEE digital libraries and allow navigation to inspect papers and authors.

1. Initial page that allows one to input search criteria comprised of either a researcher's last name.
2. When a search is submitted, it should create word cloud of the top X number of papers in the ACM and IEEE digital library that match the provided criteria. (X is configurable by the user)
3. Clicking on a word in the cloud should return a list of papers that mention that word
4. For each paper, provide link to download it from the digital library
5. For each paper, provide link to access its bibtex

6. Show a status bar for current progress in generating the word cloud (needs to reflect actual progress)
7. For each paper, clicking on an author in its author list will do a new search based on that author
8. For each paper, clicking on a paper's conference name will list other papers from that conference.
9. For each paper clicking on the paper's title will allow the user to read the abstract with the word highlighted.
10. Export lists of papers as PDFs and plain text
11. Access previously entered searches
12. For the paper list, allow users to select a subset to generate a new word cloud from
13. Allow for the downloading of an image of the generated word cloud
14. Initial page that allows one to input search criteria comprised of a keyword phrase
15. For each paper, clicking on the paper's title allows for download of a PDF version of the article with the word highlighted

Image of Product Backlog within an excel format. Link in the reference table.



Product Backlog

id	name	status	category	how to demo	notes
1	Initial page that allows one to input search criteria comprised of either a researcher's last name	sprint 1	front end	Show the main page	one input criteria for last name
2	When a search is submitted, it should create word cloud of the top X number of papers in the ACM and IEEE digital lit	sprint 1	full stack	Show the main page	x should be configurable by user
3	Clicking on a word in the cloud should return a list of papers that mention that word	sprint 2	full stack	Show that when a us	papers should be ranked by word frequency show title author conference frequer
4	For each paper, provide links to download it from the digital library	sprint 2	full stack	Show that for each	paper the user can access a download and the bibtex
5	For each paper, provide links to access its bibtex	sprint 2			
6	Show a status bar for current progress in generating the word cloud	sprint 1	full stack	Show that when the s	show status bar rectangle and needs to accurately reflect
7	For each paper, clicking on an author in its author list will do a new search based on that author	sprint 2	front end	Show that the user can	select an author for a specific paper and generate a new word cloud for that au
8	For each paper, clicking on a paper's conference name will list other papers from that conference	sprint 2	front end	Show that the user can	select the conference for a specfic paper and generate a new word cloud for th
9	For each paper clicking on a paper's title will allow the user to read the abstract with the word(s) highlighted	sprint 2			
10	Export lists of papers as PDFs and plain text	tbd	full stack	Show that when a list of papers is available,	the user can export the papers as PDFs and plain text
11	Access previously entered searches	sprint2	front end	Show that once a user has searched	more than two times, they can access a list of their previous sear
12	For the paper list, allow users to select a subset to generate a new word cloud from	tbd	full stack	Show that the user c	null
13	Allow for the downloading of an image of the generated word cloud	tbd	front end	Show that the user c	null
14	Initial page that allows one to input search criteria comporised of a keyword phrase	sprint 1			
15	For each paper, clicking on the paper's title access for download a PDF version of the paper with the word highlighted	tbd	full stack	Show that for each p	null

6. Sprint Backlog
    1. Fix issues brought up in Sprint Review

2. Clicking on a word in the cloud should return a list of papers that mention that word
3. For each paper, provide link to download it from the digital library
4. For each paper, provide link to access its bibtex
5. For each paper, clicking on an author in its author list will do a new search based on that author
6. For each paper, clicking on a paper's conference name will list other papers from that conference.
7. For each paper clicking on the paper's title will allow the user to read the abstract with the word highlighted.
8. Access previously entered searches

7. XP evidence

7.1 Develop Tests First (Test-Driven Development)

This XP practice was central to development this sprint. For each new feature or class, pair programming groups first implemented tests for the the feature or class to-be-implemented. After the tests were completed, the feature was then implemented and committed to our repository in a separate github commit. The list below provides evidence of test-driven development by listing separate test commits and implementation commits for each feature. This proves that testing led the development process and that testing was completed prior to implementation.

Feature: Clicking on author searches that author.
Github Test Commit: `f972d896ed5b89277613fb47789891fb41f79215 4/9/17`
Github Feature Commit: `af998fa5ff5cc7cd8faf222e820e41089118f197 4/9/17`

Feature: Viewing Article List
Github Test Commit: `44829ccaf606941ce970a370b5c0c5addb2ac562 4/6/17`
Github Feature Commit: `af998fa5ff5cc7cd8faf222e820e41089118f197 4/9/17`

Feature: Search History
Github Test Commit: `44829ccaf606941ce970a370b5c0c5addb2ac562 4/6/17`
Github Feature Commit: `cd6792621d550bc43f993b66e28058b43e9c3acb 4/12/17`

Feature: View Abstract
Github Test Commit: `b2357fd6e51a4bf209cf7e4953d957d712c38ad7 4/11/17`
Github Feature Commit: `cd6792621d550bc43f993b66e28058b43e9c3acb 4/12/17`

Feature: Conference Article List
Github Test Commit: `b2357fd6e51a4bf209cf7e4953d957d712c38ad7 4/11/17`
Github Feature Commit: `cd6792621d550bc43f993b66e28058b43e9c3acb 4/12/17`

Feature: Sorting Article List
Github Test Commit: `b2357fd6e51a4bf209cf7e4953d957d712c38ad7 4/11/17`
Github Feature Commit: `cd6792621d550bc43f993b66e28058b43e9c3acb 4/12/17`

Unit Test: getListofArticles
Github Test Commit: `cc046abcdc85b3db38ee3407219256d2c14cfa73 4/6/17`
Github Feature Commit: `d753787fe7efadec8fce45f04473b92eb9fabe39 4/7/17`

Unit Test: WordCloudTest
Github Test Commit: `cc046abcdc85b3db38ee3407219256d2c14cfa73 4/6/17`
Github Feature Commit: `8fd148dea3807f4f69fd794555fdf4e09e32e5e9 4/12/17`

Unit Test: getAbstract
Github Test Commit: `41e6e8d4113dd6cba1d53a2e5b6a0fc25d654b1e 4/9/17`
Github Feature Commit: `ffb2fb45f17bc1f4241f1633d42248cedefd9e1f5 4/13/17`

Unit Test: getConferenceArticleList
Github Test Commit: `8fd148dea3807f4f69fd794555fdf4e09e32e5e9 4/12/17`
Github Feature Commit: `fb2fb45f17bc1f4241f1633d42248cedefd9e1f5 4/13/17`

7.2 Pair Programming

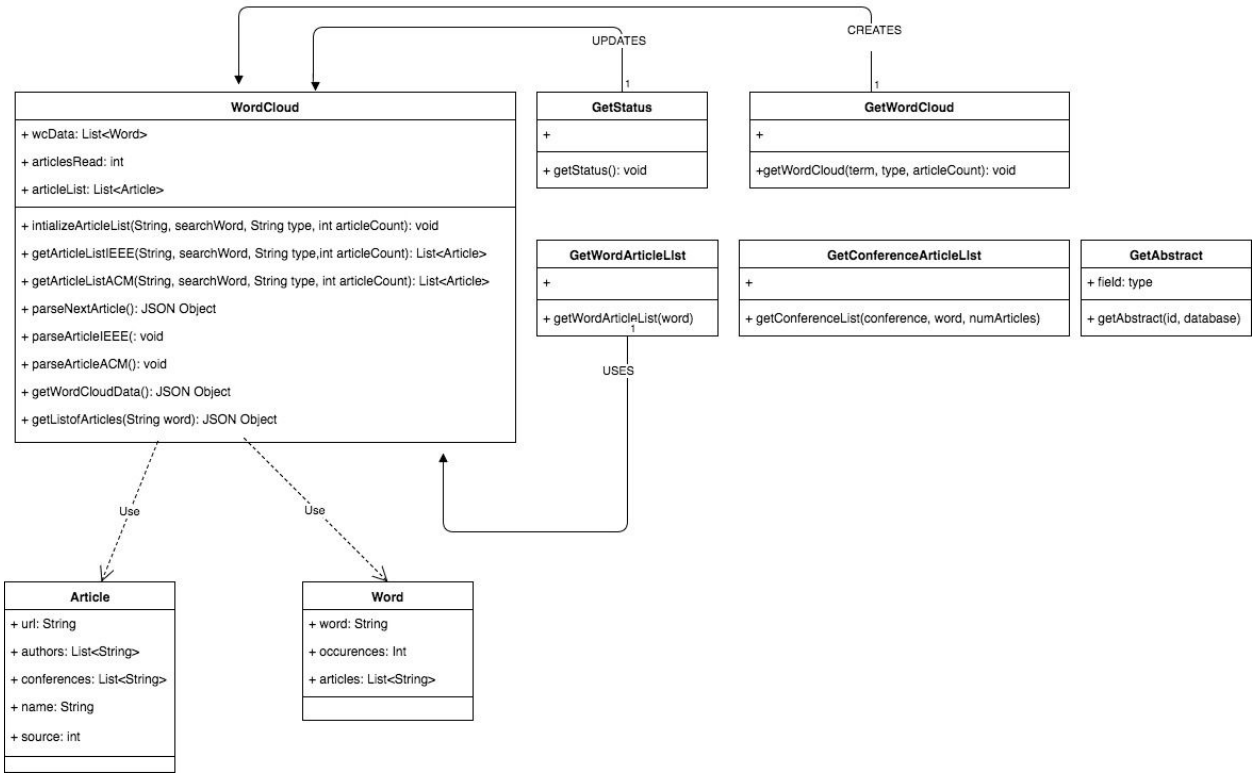
To view the evidence of pair programming during this sprint navigate to sprint2_documentation/pairProgramming in the github repository listed in the references section of this document. Inside each folder within that directory is a picture of the pair programming team with the tests that they ran after their session. The folders are intended to be created in the same commit as the code that was tested was committed but it is possible a few commits of the pair programming evidence were committed after the code was committed that is associated with the pair programming session but those would be no more than a day later in that case.

7.3 Simple Design

The system is designed to be simple in the sense that all the logic for the system is mostly centered within the php backend scripts. All the scripts that are called from the frontend via AJAX access data from a shared session object called WordCloud. WordCloud has all the data

related to the word cloud and any script that needs to access or manipulate that data has access to that data via the shared session instance of WordCloud. For each function in the frontend there are corresponding php backend scripts that provide and manipulate the data for that specific function. There are no general or multifunction scripts that have ambiguous functionality, there are only well defined scripts that accomplish specific functions. This design is simple because it is easy to access the word cloud data from any script and all the data related to the system is centered within the word cloud object.

This is an updated Class Diagram

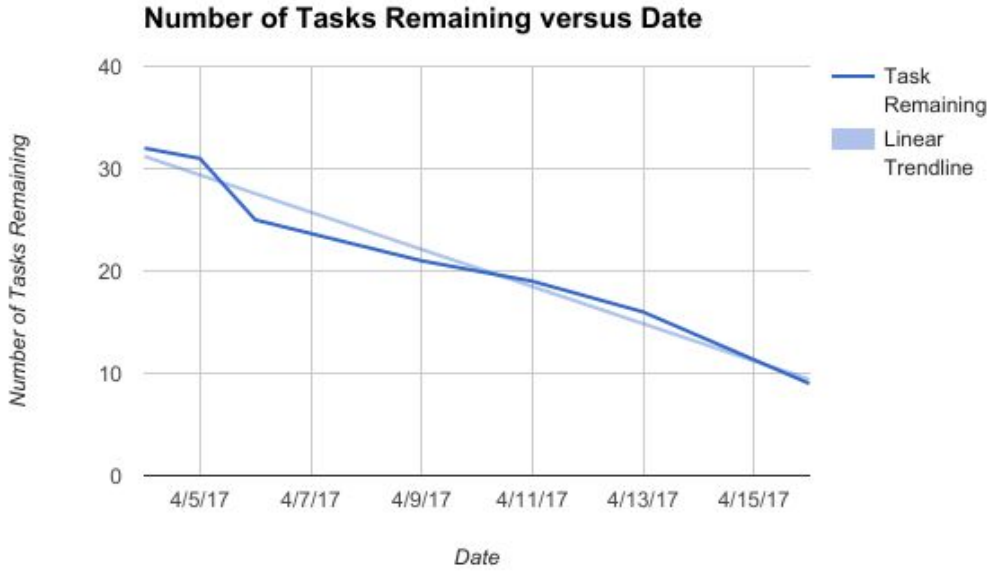


The class diagram represents the logic behind the backend part of the system. The WordCloud class represents a global shared object between all the php files which contains all the information needed to generate a word cloud and the article list associated with that word cloud. The reason for the lack of front end diagram is that majority of the computing logic resides in the backend section of the system.

The interaction between the frontend and backend that is used in the system developed during this sprint is limited to ajax calls to GetStatus, GetWordCloud, GetWordArticleList, GetConfereneArticleList, and GetAbstract. The parameters of these requests correspond to the parameters of the methods in the class diagram. Out of these five endpoints, GetWordCloud, GetWordArticleList, and GetStatus use the global object wordcloud. This is why the classes have an arrow to the WordCloud class in the diagram. GetConferenceArticleList and GetAbstract make calls to third-party APIs to get their data.

7.4 Sustainable Pace

The visual of the burndown chart shows that the team attempted to stay on a sustainable pace during the second sprint of the project. There are frequent screenshots of the burndown chart in sprint2_documentation/burndownChart and screenshots of the task chart in sprint2_documentation/taskChart. It is noted that the team was unable to finish the tasks in the task chart due to a problem with obtaining valid cookies from the Shibboleth login system which will hopefully be resolved in the final sprint.



7.5 Collective Code Ownership

This XP practice is easily documented in the pair programming logs. For the first sprint Kyle Van Landingham, Alec Schule, and Noah Bergman worked on the front end code as in index.html, style.css, and javascript.js. Andrew Zolintakis and Alec Fong worked on the back end code as in all the php code written. Looking at the Pair Programming section of the documentation it can be noted that most of the pair programming sessions took place between two team member that worked on different parts of the project or for the instance of Noah and Kyle they worked on the backend for one session. To clarify, some of the common pair programming groups were Kyle and Andrew, Alec and Alec F, Noah and Alec F. These groups have a mix of team members who worked on different parts of the project and are in compliance with the clarification asked in piazza post 320.

7.6 Refactoring

Refactoring was done for each commit that contributed a good amount of code to the system. Refactoring the code in the system was meant to improve readability or simplify the code structure. Documentation for refactoring can be found in sprint2_documentation/refactoring where there is a text document name log.txt which contains a log of the refactoring done by each pair programming group during each session that was large enough to accommodate a refactoring.

7.7 Continuous Testing

Continuous testing was adhered to during the sprint by running the entire test suite after each pair programming session and submitting the results of the test with the code that was tested. These test results can be found within sprint2_documentation/pairProgramming and within each folder in that directory there are files designated as the test results for that commit.

7.8 Create a test for every bug found

A log of creating a test for every bug found during the sprint can be found in sprint2_documentation/testing in the document testsForBugs.txt. It is noted that there were very few tests for bugs found, a possible explanation for this is that many bugs were found during the running of the tests and test suite was adapted to check for those bugs but were not documented due to preoccupation with getting the test suite to run correctly and documentation was forgotten among the rest of the required documentation. It is possible to observe the code changes in the different commits however due to time constraints the team was not able to track each change of the test suite to accommodate for testing specific bugs. It is not expected that the graders go through the repository themselves either, this is just an attempt at explaining the low number of test cases listed in the testsForBugs.txt document.

8. Scrum Documentation

Pictures of the stand up meets can be found in sprint2_documentation/scrumMeetings.

Here is a log of the stand up meetings.
4/5/17
What have you done since the last meeting? (Sprint Planning)
Everyone: Sprint retrospective
Kyle: Pair programmed with Noah and refactored parts of WordCloud.php
Noah: Refactored some parts of Wordcloud.php and finished implementing Word.php

Alec F: spiked on acm api's for optimizing performance speed
Andrew: spiked on acm and ieee api's for getting pdfs
Alec S: Sprint review
What will you accomplish before the next meeting?
Kyle: Get rid of stop words in the wordcloud data and limit word cloud to top 250
Noah: A few tasks related to providing more info from the backend
Alec F: I will spike getting app to run on XAMPP
Andrew: Make the frontend changes from sprint 1 review (i.e. making status bar a rectangle)
Alec S: Remove stop words from word cloud
Are there any obstacles that stand between you and getting your plan done?
Kyle: It may take some large refactoring to get the desired results
Noah: Nah,
Alec F: XAMPP Apache server shutting down unexpectedly
Andrew: Havn't worked on frontend much this project
Alec S: Need time to meet up to pair program

4/6/17
What have you done since the last meeting?
Kyle: Removed stop words from word cloud in backend
Noah: Removed stop words from word cloud in backend
Alec F:
Andrew: Created black box test feature files for search history, showing article list, and showing, article abstract and also changed the status bar to be a rectangle
Alec S: Pair programmed with Andrew on black box tests
What will you accomplish before the next meeting?
Kyle: Parse the PDF instead of the abstracts for word cloud generation and display article list when word is clicked on the front end
Noah: Will work on the backend and finish some tasks
Alec F:
Andrew: Finish the rest of the black box test feature files
Alec S: Pair program with Andrew to finish the rest of the black box test feature files
Are there any obstacles that stand between you and getting your plan done?
Kyle: It will take a while to understand the new way we obtain the ACM articles but IEEE will be more manageable
Noah: Need hidden ACM api
Alec F:
Andrew: No
Alec S: Will be out of town.

4/9/17
What have you done since the last meeting?
Kyle: Worked with Andrew to fix the word class, sort words, implement getArticleList, getAbstract, and parse pdfs all in the backend
Noah: I worked with Alec F to complete several tasks on the frontend. Frontend now displays a table that can be sorted and searches authors when clicked.
Alec F: Worked on using python to emulate browser in order to get cookies and download pdf
Andrew: Worked on word class, sorting word cloud, and get Article List based off a word
Alec S: Nothing
What will you accomplish before the next meeting?
Kyle: Work on sending the conference list to the frontend
Noah: Continue to complete frontend tasks. At minimum, displaying abstract on frontend
Alec F: Ability to download ieee and acm pdfs
Andrew: Get the conference list
Alec S: Continue finishing black box tests
Are there any obstacles that stand between you and getting your plan done?
Kyle: The requirements were ambiguous for this feature so we might have to clarify with the product owner
Noah: Waiting on some backend tasks to finish
Alec F: IEEE hides their link for the pdf
Andrew: It might be difficult to search just for conferences
Alec S: No

4/11/17
What have you done since the last meeting?
Kyle: Implemented Get Articles based on a conference with Andrew
Noah: Worked on documentation
Alec F: Worked on figuring out the Shibboleth login system
Andrew: Implemented Get Articles based on a conference
Alec S: Nothing
What will you accomplish before the next meeting?
Kyle: I will work to get the abstract and send it to the frontend and get bibtex done
Noah: Get the search history done
Alec F: Figure out the Shibboleth login system
Andrew: Getting the abstract for an article and parsing pdfs for the word cloud
Alec S: Get the blackbox tests working
Are there any obstacles that stand between you and getting your plan done?
Kyle: It might be hard to get abstracts quickly
Noah: nah

Alec F: It's a complicated system that I figured out in python but using php will be much harder
Andrew: Getting the abstract and pdfs requires a usc login
Alec S: None

4/13/17
What have you done since the last meeting?
Kyle: Got the abstracts in the backend and bibtex
Noah: Search history was finished
Alec F: Working with noah and still tried to figure out the login system
Andrew: Finished sending the abstracts to the frontend and bibtex
Alec S: Finished the feature files
What will you accomplish before the next meeting?
Kyle: Finish documentation
Noah: Finish documentation and blackbox feature context
Alec F: Get login system working
Andrew: Finish documentation
Alec S: Get blackbox feature context finished and login to Shibboleth with php
Are there any obstacles that stand between you and getting your plan done?
Kyle: none
Noah: none
Alec F: I may not be able to finish this task this sprint
Andrew: none
Alec S: Getting the login with php may be more difficult than anticipated

4/16/17
What have you done since the last meeting?
Kyle: Documentation
Noah: Documentation
Alec F: Made progress on the login system but not completely finished
Andrew: Documentation
Alec S: Worked with Alec on Shibboleth
What will you accomplish before the next meeting?
Kyle: n/a
Noah: n/a
Alec F: n/a
Andrew: n/a
Alec S: n/a
Are there any obstacles that stand between you and getting your plan done?
Kyle: n/a

Noah: n/a
Alec F: n/a
Andrew: n/a
Alec S: n/a