Project Horseshoe Farm   
 Cycle 3 Report

By:

Amanda Bailey

John Carroll

Lanxin Ma

Allison Macdonald

Cesar Sanchez

Submitted in partial fulfillment of the requirements for  
 COMP 4710 Senior Design  
 to the Department of Computer Science and Software Engineering,  
 Samuel Ginn College of Engineering, Auburn University

Auburn, Alabama

April 26, 2016

# Table of Contents

Table of Contents..................................................................Allison Macdonald, John Carroll................................................. ii

2 Executive Summary (System Metaphor).........................Amanda Bailey............................................................................. 4

3 Project Introduction................................................Cesar Sanchez & John Carroll............................................................... 5

3.1 Previous Development........................................................................................................................................ 5

3.2 Intent This Cycle................................................................................................................................................. 5

3.3 Future Work......................................................................................................................................................... 5

4 Requirements / User Stories...............................................Lanxin (Mark) Ma...................................................................... 6

4.1 User Stories.......................................................................................................................................................... 6

4.1.1 Off-Screen Ghost Markers............................................................................................................................... 6

5 Design Documentation.......................................................Allison Macdonald & Amanda Bailey..................................... 12

5.1 Architecture..........................................................................................................................................................12

5.2 Structure...............................................................................................................................................................16

5.3 Interfaces..............................................................................................................................................................17

5.4 Assumptions and Dependencies ..........................................................................................................................17

5.5 Planned vs Actual Code ......................................................................................................................................17

6 Management Plan...............................................................John Carroll.............................................................................. 18

6.1 Tasks Under Development................................................................................................................................ 18

6.2 Task Assignments............................................................................................................................................. 18

6.3 Development Schedule..................................................................................................................................... 19

6.4 Planned Code / Feature Freeze......................................................................................................................... 19

7 Risk Mitigation....................................................................Cesar Sanchez......................................................................... 20

8 Test Plan and Test Procedures.............................................Amanda Bailey........................................................................ 21

8.1 Test Plan............................................................................................................................................................ 21

8.2 Test Procedures................................................................................................................................................. 21

9 Lessons Learned...................................................................Lanxin (Mark) Ma................................................................. 21

References.................................................................................Allison Macdonald................................................................. 23

Appendix A...............................................................................John Carroll............................................................................ 24

A.1 Status Reports.................................................................................................................................................. 24

A.2 Meeting Minutes.............................................................................................................................................. 24

A.3 Size Estimation Documentation....................................................................................................................... 24

A.4 Problem Reports / Change Requests................................................................................................................ 24

A.5 Correspondence................................................................................................................................................ 24

A.6 Source Code..................................................................................................................................................... 24

# 2 Executive Summary (System Metaphor)

Horseshoe Farms has requested a webpage which can help streamline their online tutoring process. There will be a Child, Remote Tutor, and a Admin/Team Leader account. Children will be able to log in, view notifications, and then be redirected to their lesson. Admin’s should be able to log in and view a roster of Children, and assign Children to Remote Tutors. Team Leaders should be able to log in, plan lessons for Children, and award horseshoes. Remote Tutors will be able to log in, wait to be assigned to a Child and then be redirected to their lesson.

# 3 Project Introduction

This project aims to create a website for the Horseshoe Farm charity to aid them in providing education in areas such as math and reading. Horseshoe Farm is a charity situated in Greensboro, Alabama that is committed to helping vulnerable kids and teens, such as those suffering from mental impairments. They provide in person tutoring in math and reading by connecting kids with volunteer Remote Tutors. Despite this, the current system could still be improved, hence our group was commissioned to build a standalone website that could connect Children with their Remote Tutors, supplement lessons, and assist organizers in matching kids with the Remote Tutor best suited for them. While it is unlikely that we will be able to implement the site in full by the end of this semester, we aim to provide a framework that can be utilized by Horseshoe Farm and easily expanded by the groups that will follow us.

## 3.1 Previous Development

This is the first semester this project has been developed, so there was no previous work for our group to build on. In our first cycle, our group developed a prototype of the website that had basic login functionality for Children. While not having much functionality, this prototype acted as the root for all other development done this cycle. A great deal of time that first cycle was also spent on clarification of requirements and expectations provided by the sponsors.

In cycle 2, the base was further fleshed out by expanding login capabilities to include Children and Admins. Furthermore, the login page was polished in terms of both appearance and features. The skeleton of the Remote Tutor's page and the database were both built as well. Finally, significant time was dedicated to researching software and the exact day-to-day activities that occur at Horseshoe Farms.

## 3.2 Intent This Cycle

This cycle our group implemented several of the more vital features in an effort to have a usable product available to the sponsors. The previously skeletal Remote Tutor page was implemented this cycle, directing them to a waiting page if they are not paired with a Child and to Talky.io if they are. This pairing can be done via the Admin pairing page we implemented, where a given Child's attendance can be marked, and they can be paired with and then connected to any Tutor that is online. If a Child is not paired with a Tutor, then they are able to do a solo day, as determined by the Team Leader, who can assign activities for a day via the newly implemented recommendations page. As mentioned previously, both the Child and Remote Tutor can be redirected to Talky.io at any time by an Admin to begin a session.

## 3.3 Future Work

While not a complete product, the baseline features are implemented to render it useable. Future groups should add additional core features (such as session timeout) as well as address any bugs that may be present in the user stories we worked on.

Below are notes and comments on user stories that are completed, but with known issues or possible complications. Also, below are user stories that are not completed.

### 3.3.1 User Story# 6: Child resume reading from last page

This user story was not started during this cycle or by the end of this cycle. All research is left up to the next group.

### 3.3.2 User Story# 8: Child /tutor screen share control

This user story involves the use of the program TeamViewer. Our group did research into ways to automate login into TeamViewer to fulfill this user story. From speaking with our TA, we got wind of a way that would involve command line. Our thoughts on it would be to create a batch file, an executable on the desktop that would auto login into TeamViewer for the user. Either the batch file would be customized per user, or it would pull in login information from the database through some means.

### 3.3.3 User Story# 14: Auto login into external websites

For user story# 14, John has done a lot of research on his own and spoke to two of his most credible sources for web development. One works for CWS(Campus Web Solutions), the other at ENS(Engineering Network Services). There are some options for the user story but, unfortunately, they do not lead to this user story being completed within the remaining time of this cycle.

John’s conversation notes with his contact who has worked at CWS:

It is possible, e.g. AuAccess and Canvas, however they would likely need a central authentication server to do so, or some sort of API to allow for that.

You would have to either set up a partnership with those sites and allow their authenticatation through your portal on login, have a central authentication server, or pray they have an API for that. Short of that, you can't use your cookies on their sites - it won't be set up to allow that.

You'd likely have to contact each place directly and see if they have anything that can support that. Cookies will be fine for your own authentication, but it won't allow you to access anyone else's sites

John researched forging cookies, and using the POST method, which duplicates the form and attempts to enter it in on the external site, as well as storing sessions for each user in the database.

POST method will not work, mostly due to anti-forgery, anti-hijacking implementations in place on the destination domain.

Session data is stored in cache and cookies. The issue is that a cookie is needed to have a session, and authentication is needed to have a cookie. Cookies come with an internal expiry. This expire date has a time of persistent, can set for indefinite, or whenever, but whenever it expires, it will no longer provide information

Authentication typically works like: Send out cookie, get a CAS cookie, then the user logs in with that.

From speaking with John’s sources and from research, here are the options, ranked with the best option first:

1. Central Authentication Server - need partnership with the other company/party

2. API

3. Some other arrangement.

1. CAS with partnership:

We can ask Khan for a partnership. If they say yes, they would work with us to get us up and running. They may provide us with an API if they do not decide to work with us directly, which will likely provide us with a way to tie into their system.

2. API

Google authentication API

Perhaps using this API for simply interfacing with Kindle and Khan for just logging in would be for the best since each Child/student has a Google account through their gmail account.

<https://developers.google.com/identity/>

<https://developers.google.com/api-client-library/javascript/features/authentication>

Khan authentication API

This API will allow this project to grab behavior data regarding the Child/student -- badges, exercises completed, analytics, etc.

<http://api-explorer.khanacademy.org/group/api/v1/user>

<https://github.com/Khan/khan-api/wiki/Khan-Academy-API-Authentication>

Google SAML (SSO)

<https://developers.google.com/google-apps/sso/saml_reference_implementation?csw=1>

SAML Single Sign-on for Khan

(not currently supported, but should be in the future)

<https://www.bitium.com/khan-academy-saml-provider>

3. Some other arrangement

One approach John looked into was the use of iMacros.

<http://www.hacktabs.com/how-to-automate-login-to-website/>

John had researched iMacros and drew conclusions to its limitations. iMacros has a desktop application as well as recently releasing a Firefox Add-on and a Google Chrome Extension.

Of the options, the iMacros desktop application was the most advanced and had the most documentation for using it. It is possible to run a recorded macro from javascript in Internet Explorer because of its use of ActiveX. Also, it was possible in Firefox, but buggy. As for Chrome, it was not possible and this is probably why they made an Add-on and an extension, respectively.

John spoke with Dr.Dorsey, our sponsor, in regards to this option which would need to rely on an app that is on the desktop. Because it is on the desktop and not web-based like the rest of our web-component, it is not portable, thus it is not meeting the customer's requirements. John expressed that a Child, a student, would need to be on the same computer each day they tutored for the auto-login to work for external websites through the use of iMacros on the desktop.

As for the Firefox Add-on, it was not looked at since our operations are centered around Google Chrome. John managed to try out the Google Chrome Extension. It works really well from within the browser, and it saves macros to the google account user's bookmarks. Macros cannot be ran from javascript, they must be ran from utilizing the iMacros extension. This would not be ideal for us, because it would need the user to not only install the extension but also log into their google chrome browser to save the extension for future use. Also, that macro could be recorded once and placed into each student's files/bookmarks in Chrome, but that is not easily scalable nor is it efficient. One more thing to add is that the amount of documentation present for use or manipulation of the iMacros Chrome extension is relatively low.

In conclusion, due to the lack of resources, as well as it being not ideal for this project, the use of iMacros should not be used due to its limitations for future development. A partnership is the most ideal for future development. The use of at least one API is also very promising for future development for user stories and functionality.

### 3.3.3 User Story# 15: Admin

It is possible to pair a Child with a Remote Tutor even though a Child is not present. This is an issue because this is not wanted behavior because it could cause issues if the Child is not present.

# 4 Requirements / User Stories

This section describes several user stories for this cycle. A user story is a description of the steps a user would take in performing various actions within the software.

## 4.1 User Stories

##### 4.1.1 Child Log In (1)

Summary: A Child login system that allows Children access to the website when they type in a username and password.

Description: The Child login box is placed in the middle of the screen where the Child can login to the system by typing in a requested username and password through keyboard. If the Child types in the right username and password, the Child will be able to access his/her account. If the Child types in the wrong username or password, he/she will be blocked out of their account and be required to retype their username and password.

Hours: Total planned: 25

Planned this cycle: 25

Total Actual: 9

Actual this cycle: 9

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Completed

##### 4.1.2 Child View Notifications (2)

Summary: A section in which Children can view progress made during last session.

Description: The page will greet the Child by name and display a section in the center of the screen which will include a bulleted list of the number of pages read, number of videos watched, and number of worksheets completed during the last session. The font will be large and simple to read so that young users will have a simple time understanding the notifications. The only way a user may interact with the page at this point is to click the "Okay!" button in order to move into the next section of the site.

Hours:

Total Planned: 15

Planned This Cycle: 15

Total Actual: 6

Actual This Cycle: 10

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Completed

##### 4.1.3 Child Auto Redirect to Lesson (3)

Summary: Children will be automatically redirected to their scheduled lesson once they are finished reading notifications.

Details: After logging in and being shown progress from the last session as well as the schedule for the day, the Child will select the final "Okay!" button to indicate that he or she is ready to continue. This button will cause the system to redirect the Child to the lesson on his or her schedule. It will also connect the Child to the specified Remote Tutor for the day.

Hours:

Total Planned: 10

Planned This Cycle: 5

Total Actual: 5

Actual This Cycle: 5

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Completed

4.1.4 Student/Tutor start video chat service (4)

Summary: Children and Remote Tutors will meet face to face while discussing about their lesson plans.

Details: The Child and Remote Tutor will meet on a video chat service such as "Talky.io" with "TeamViewer". During the duration of their onscreen meeting, the Remote Tutor will teach the Child the required lessons for that day, while the Child follows the Remote Tutor and is allowed to take notes or ask questions onscreen. After the Remote Tutoring session is over, or if the Child or Remote Tutor cannot continue, he/she is allowed to dismiss the session by logging out of their account.

Hours: Total planned hours: 25

Planned this cycle: 5

Total actual: 1

Actual this cycle: 1

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Completed

##### 4.1.5 Child View Math Worksheets (5)

Summary: Children will view embedded PDFs as worksheets within the site.

Details: Alternately to being redirected to either read.amazon.com or khanacademy.com, a Child may be redirected to a worksheet for the day. This worksheet will be built into the web site and the Child will be able to work from it.

Hours:

Total Planned Hours: 10

Planned This Cycle: 10

Actual Hours This Cycle: 3

Total Hours This Cycle: 3

Coder: Cesar Sanchez

Tester: Amanda Bailey

Reviewer: Team

Status: Completed

##### 4.1.6 Child Resume Reading From Last Page (6)

Summary: Children will be redirected back to the last page read on read.amazon.com.

Details: When a Child is redirected to a reading lesson which picks up from the last session, he or she will be redirected directly to the most recently read page rather than the beginning of the book.

Hours:

Total Planned Hours: 12

Planned Hours This Cycle: 0

Actual Hours This Cycle: 0

Total Hours This Cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Unstarted

##### 4.1.7 Child preview lesson for the day (7)

Summary: Children will be directed to a page that will contain a summary of topics that will be covered by the Remote Tutor for that particular session.

Details: After the Child has logged into his/her account, he/she will be directed to a page that contains a list of topics and subjects that will be covered for that daily lesson.

Hours:

Total planned hours: 10

Planned hours this cycle: 10

Actual hours this cycle: 0

Total hours this cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Discarded

##### 4.1.8 Student/Tutor Screen Share Tool (8)

Summary: The system will automatically connect the Child to a Remote Tutor once ready for the session.

Details: While connecting to Talky.io, the website will also drive a connection to the Remote Tutor via TeamViewer. This tool will allow the Remote Tutor to take control of the student's screen at appropriate times to aid in demonstrating lessons to the student.

Hours:

Total Planned Hours: 10

Planned Hours This Cycle: 0

Actual Hours This Cycle: 2

Total Hours This Cycle: 2

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Discarded

##### 4.1.9 Remote Tutor view daily schedule (Child and lesson) (9)

Summary: The system will display the list of Children and lessons that the Remote Tutor will need to teach for a particular day.

Details: When the Remote Tutor logs into their account, he/she will be able to view a list of appointments that Children have made. The Remote Tutor will be able to view the name of the student, the subjects he/she needs help on, and the time that they want to meet the Remote Tutor with. This schedule list will allow the Remote Tutor to plan accordingly, so he/she will be able to help every single Child with their problems.

Hours: Total planned hours: 15

Planned hours this cycle: 0

Actual hours this cycle: 3.25

Total hours this cycle: 3.25

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Discarded

##### 4.1.10 Secured Log In (10)

Summary: Logging into either Admin or Remote Tutor accounts will require a password for logging in.

Details: All users log in at the same page, but if the system detects that the account belongs to either an Admin or a Remote Tutor, it will request a password. An incorrect password will not allow the user to log in.

Hours:

Total Planned Hours: 10

Planned Hours This Cycle: 10

Actual Hours This Cycle: 4

Total Hours This Cycle: 4

Coder: Allison Macdonald

Tester: Amanda Bailey

Reviewer: Team

Status: Completed

##### 4.1.11 Admin View Child Statistics (11)

Summary: Admin will be able to view a page containing Child statistics.

Details: After logging in, an Admin will be allowed to click on the name of a student. This will cause a Child information page to open which includes the student's most frequently matched Remote Tutor and the student's current schedule. It will also include an analysis of the student's recent activity.

Hours:

Total Planned Hours: 20

Planned Hours This Cycle: 0

Actual Hours This Cycle: 0

Total Hours This Cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Discarded

4.1.12 Admin View student/tutor pairing history (12)

Summary: The sponsors would be able to access a page that allows them to view the frequency of each student/tutor pair during each session.

Details: When a sponsor logs into the Remote Tutoring website, he/she will have access to the list of all the student/tutor pairings for every lesson taught since then. They will be able to view how frequently each Child is paired with a particular Remote Tutor. This way, the sponsors will be able to make arrangements to have a Child paired with his/her favorite Remote Tutor for future lessons.

Hours: Total planned hours: 15

Planned hours this cycle: 0

Actual hours this cycle: 0

Total hours this cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Discarded

4.1.13 Admin plan Child lesson (13)

Summary: Admin will be able to access a page where they can plan Child lessons.

Details: After logging in, the Admin will be allowed to make daily lesson plans for the Children. This page will be directed into the account of every single Child and Remote Tutor after the plans have been made, so that everyone will be aware of what to expect for that day, and plan accordingly based on the lessons issued by the Admin.

Hours: Total planned hours: 25

Planned hours this cycle: 10

Actual hours this cycle: 0

Total hours this cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Collaboration

4.1.14 Auto Login to external websites (14)

Summary: Child will be automatically logged into external websites once he/she logs into the main page.

Details: After logging in, Children will have access to any external website supported by the system.

Hours: Total planned hours: 18

Planned hours this cycle: 18

Actual hours this cycle: 3

Total hours this cycle: 3

Coder: John, Amanda

Tester: TBD

Reviewer: Team

Status: Collaboration

4.1.15 Admin pair Child to Remote Tutor (15)

Summary: Admin pairs each Child with their selected Remote Tutor.

Details: Once the Admin logs in, he will view the list of all the Children and their chosen Remote Tutors. The Admin will make the final judgements of which Children should be with which Remote Tutors based on their choices, and finally pairs each Child with a Remote Tutor.

Hours: Total planned hours: 15

Planned hours this cycle: 15

Actual hours this cycle: 15

Total hours this cycle: 15

Coder: Allison, Cesar

Tester: TBD

Reviewer: Team

Status: Completed

4.1.16 Remote Tutor waiting view (16)

Summary: A blank window where the Remote Tutor will wait for a particular Child to login.

Details: Once a Remote Tutor logs in and goes to the viewing page, the Remote Tutor will be directed to a blank screen. Once the Remote Tutor is connected to a student, talky.io will pop up on the screen and the Child lesson plan will start.

Hours: Total planned hours: 15

Planned hours this cycle: 15

Actual hours this cycle: 3.75

Total hours this cycle: 3.75

Coder: Amanda, Lanxin (Mark)

Tester: TBD

Reviewer: Team

Status: Completed

4.1.17 Log out/Auto log out (17)

Summary: User logout page.

Details: user will press a key that will log him/her out of their account. If the user does not log out after a certain period of time, the system will automatically log the user out.

Hours: Total planned hours:

Planned hours this cycle:

Actual hours this cycle:

Total hours this cycle:

Coder:

Tester:

Reviewer:

Status: Unstarted

# 5 Design Documentation

The overall intent behind designing this project was simplicity. Horseshoe Farm plans to give this system to Children as young as five years old by the end of the year. It was requested that the project be implemented with a minimalistic user interface that would allow even those who are not strongly computer literate to utilize the website. Therefore, we kept simplicity in mind when making decisions. Due to our limited timeframe and this being a new project, development on key features were the priority.

## 5.1 Architecture

In order to keep the user interface as simple as possible, we stuck to a 1:1 relationship. The Admin and the Team Leader have the ability to change what is viewed on the Child’s and the Remote Tutor’s session. The Child’s page will change based off of the recommendations made by the Team Leader. If it is a Remote Tutor day for a Child, the Admin will redirect the Child’s page to the talky.io of the Child’s Remote Tutor. If the Admin assigns a Child a reading day, then no additional pages will be loaded for the Child, however if the Admin assigns the Child a math day, then Khan Academy will load for the Child. If the Admin paired a Remote Tutor to a Child having a reading day, then the Kindle website will open for the Remote Tutor, as well as talky.io. If the Child is having a math day, then only talky.io will load for the Remote Tutor. Each Team Leader has amount of children they are in charge of. Depending on the recommendations made by each Team Leader for a Child, a list of links and or PDF's will appear on the Childs recommendation pages.

After each use case is a customer provided storyboard for what they had in mind for each page. Due to some issues with security, there were changes that had to be made for certain pages. One of these changes was giving a separate page for the password field, as opposed to having it appear underneath the username field. This change was made as a result of the method used for secured login. A more important difference between the UI Diagrams and what has been implemented was embedding talky.io and Kindle on to Project Horseshoe Farm's website. This feature was discarded due these websites preventing others from embedding their website onto other website due to security concerns. Instead, talky.io and kindle open in separate tabs.

### 5.1.1 Administrator Use Case

Admins have a high level view of the entire system. While logging into the website, Admins will be brought to the same log in screen used by Children. If the user enters an invalid username, an error message will appear. However, if a valid username is provided, the website will redirect to a screen asking for a password. An invalid password will result in an error message while a valid password will bring the Admin to a welcome screen where they can pair a Child with a Remote Tutor. This welcome screen will be populated with two lists on either side of the screen, one for Children, and one for their corresponding Remote Tutors. The Child list will indicate which Children are currently online while the Remote Tutor list shows which Remote Tutor the Child is paired with. In future iterations of this project, the Child-Remote Tutor pairings should be made automatically based on match percentages rather than recommendations of the Admin. Clicking on any one of the provided Child names will eventually bring the Admin to an information page regarding the Child of interest. A Child info page will display matching history and behavior tracking. From this page, an Admin may go back to the roster or move on to the next student. The automation feature and the ability to click on a Child's name has not been implemented yet, because they were not key features.

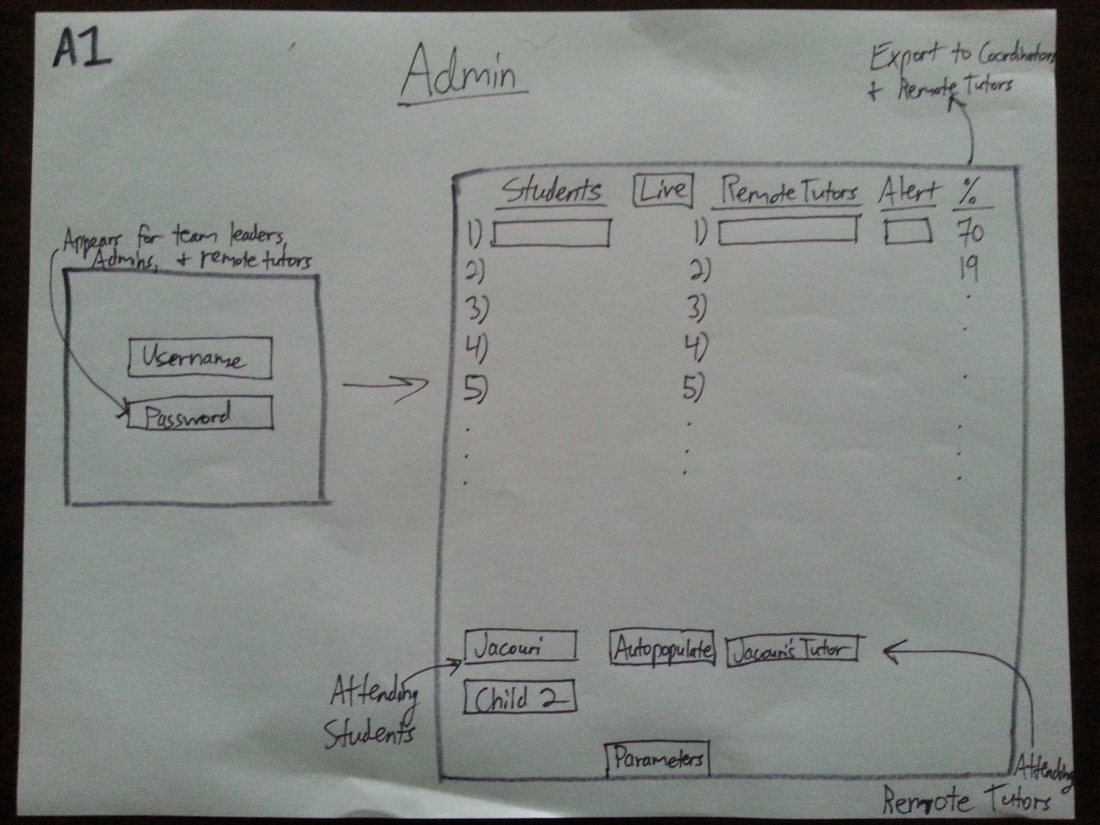


Image A1: Login and Child with Remote Tutor UI storyboards for the Admin, provided by Project Horseshoe.

From image A1, the key features that have been implemented from this page are showing a list of Children and a list of Remote Tutors. In the final implementation the Remote Tutor list is a drop down menu next to each Childs name( a decision made for simplicity), which shows only the Remote Tutors currently online (a decision made my Project Horseshoe).

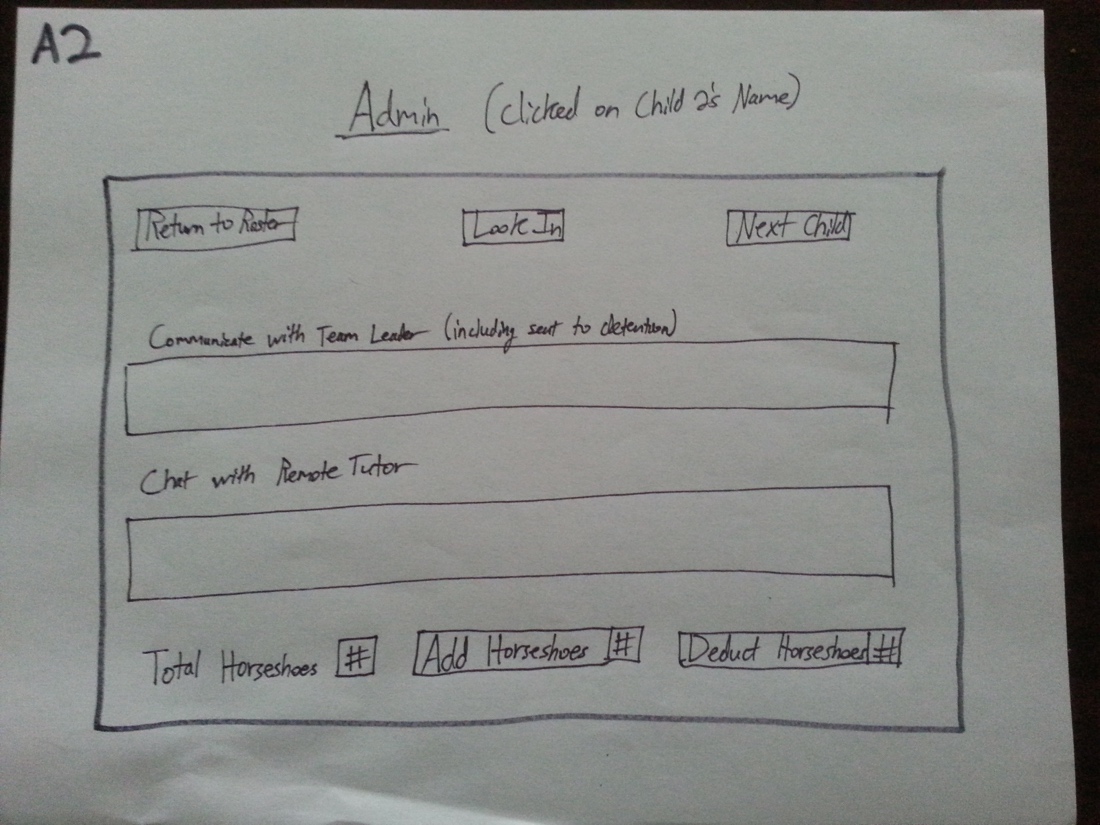


Image A2: When an Admin clicks on a Child's name UI storyboard provided by Project Horseshoe.

From Image A2, no development has been made. This is due to the nature of the page being communication and behavior based. Project Horseshoe instructed us that the communication feature would be a nice feature to have but currently is non-essential. All portions relating to behavior were considered not to be key features.

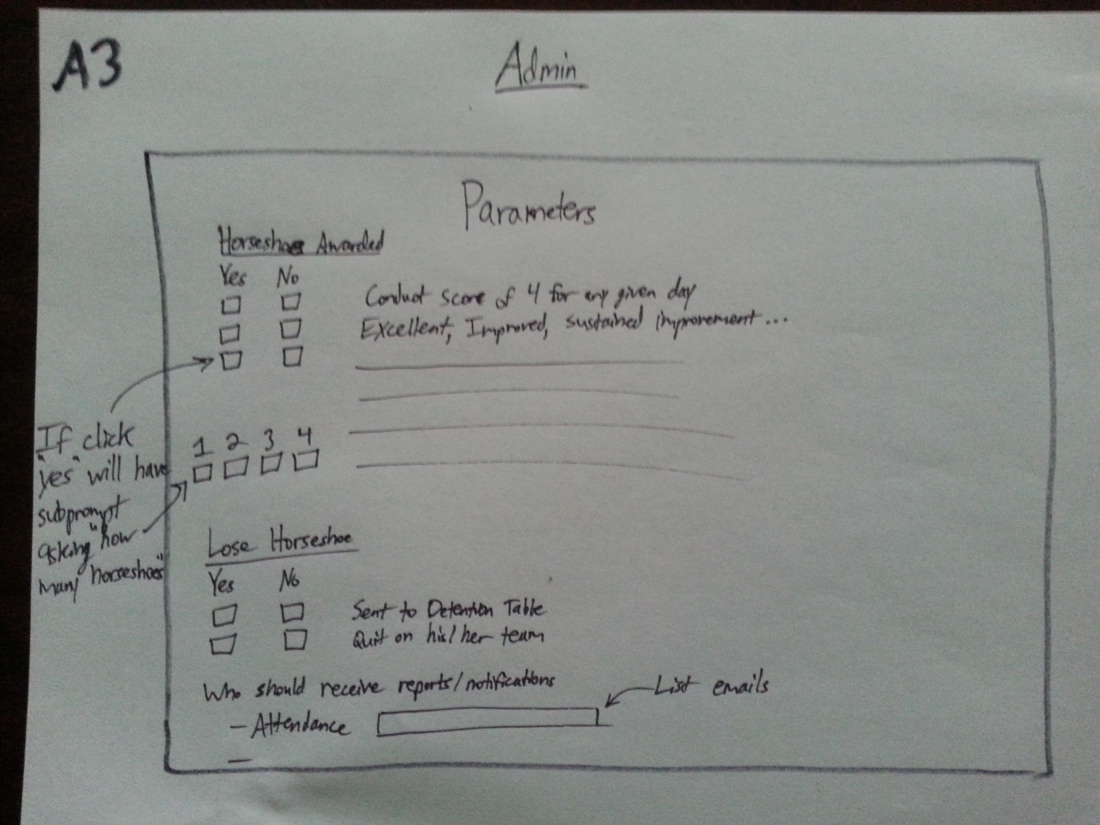


Image A3: Admin award Horseshoes to a Child UI storyboard, provided by Project Horseshoe.

### From Image A3, no development has been made due to the page being mostly behavioral in nature. All portions relating to behavior were considered not to be key features.

### 5.1.2 Team Leader Use Case

Several Team Leaders are always on site during Remote Tutoring sessions. These leaders are each assigned a group of Children to preside over and are the individuals responsible for making recommendations of study for each Child. After being brought to the main log in screen, the user will provide a username. Invalid usernames will result in an error message appearing while valid usernames will move the user to password screen. An invalid password will show an error message, but a valid password will bring a Team Leader to a page populated with Child names. The Team Leader may then use this page to check Children in or out and make recommendations. Eventually clicking on a Child name will bring up more details about that Child and allow the Team Leader to leave comments about the Child behavior for the day. Due to time restrictions and all the other features labeled as non-key features, only the recommendations feature will be available to Team Leaders by the end of this cycle.

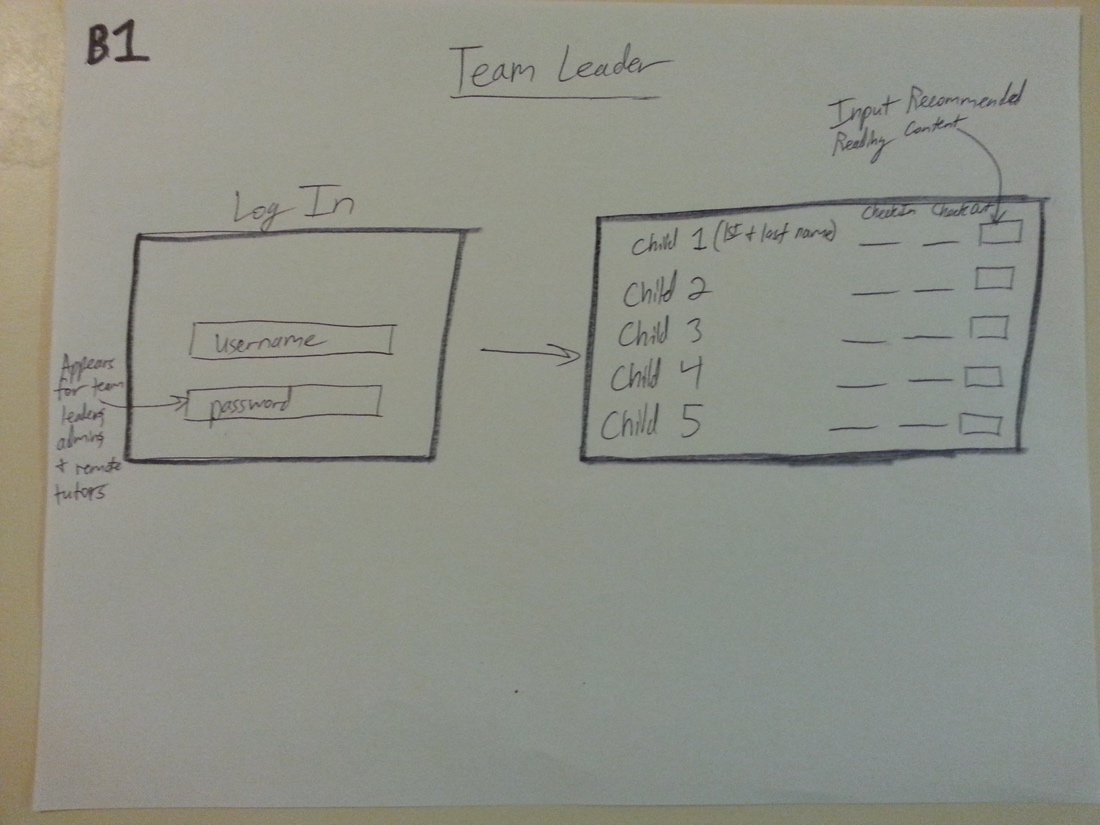


Image B1: Team Leader Login and Team Leader make recommendations UI storyboard, provided by Project Horseshoe

From Image B1, we have kept the Childs name on the left but have changed the layout of the rest. Since a Team leader can recommend reading or math content the field on the right most side labeled 'Input Recommended Reading Content' is also for math content. In order to allow for different types of recommendations, a PDF or a link, a column has been added to the left of the 'Input Recommended Reading Content' which is a drop down menu with the option of either PDF or link. If a PDF is selected, then the input field on the right will be a drop down menu containing options of PDFs that have been uploaded to the site. There is also a drop down menu at the top of the page in order to toggle between whether the PDF's shown are for math or for reading. Currently this drop down menu is not operational. Due to the time restrictions, a decision was made to leave the implementation to future development.

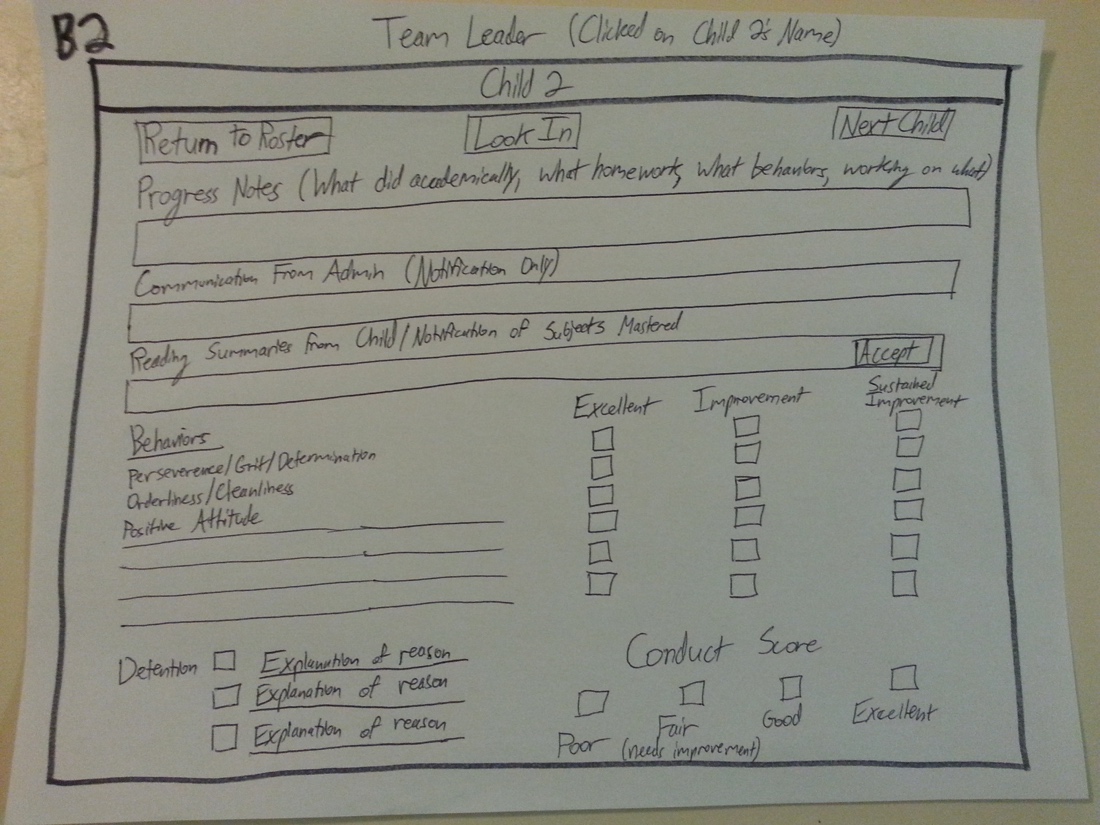


Image B2: Team Leader clicking on a Child UI storyboard, provided by Project Horseshoe.

From Image B2, this page was not implemented because behavioral features were considered to not be key features.

### 5.1.3 Remote Tutor Use Case

Remote Tutors will have a log in situation similar to that of an Admin. The Remote Tutor will be presented with the same log in page as other users, and similarly to the Admin will be asked for a password if he or she has entered a valid username. Upon receiving a valid password, the page will direct the Remote Tutor to a waiting screen until he or she can be paired with a student. Upon pairing with a Child, talky.io will load. If the Child has a reading day, then an additional page for the Kindle website will also load.

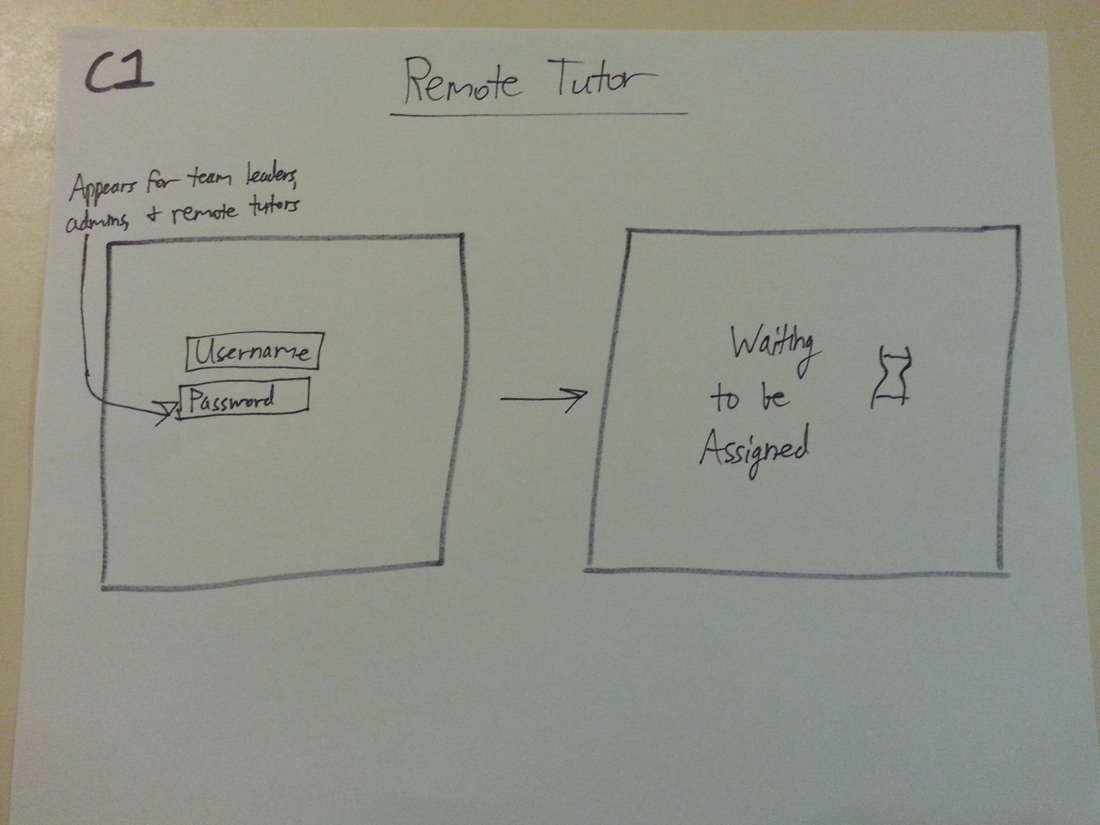


Image C1: Remote Tutor Login and Waiting View UI storyboard, provided by Project Horseshoe.

As of Cycle 3, the functions shown in Image C1 are fully implemented. The Remote Tutor can log in with a password and will be sent to a waiting page.

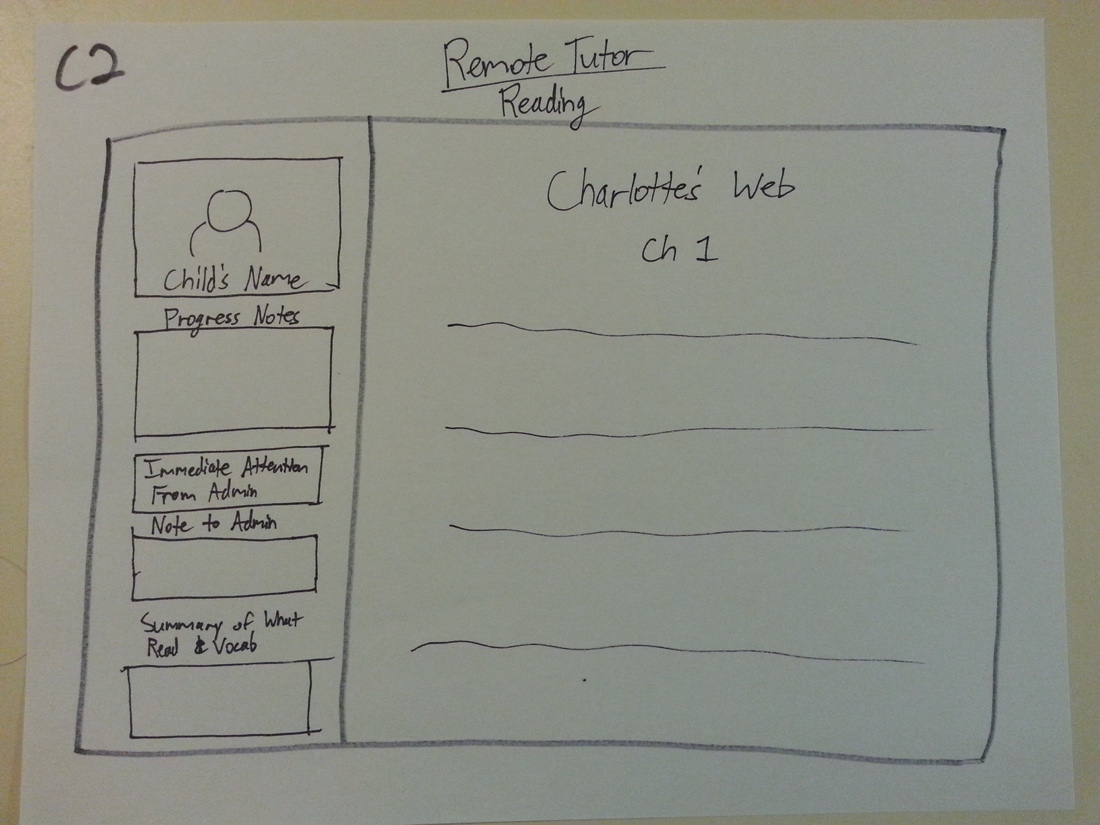


Image C2:Remote Tutor Session when Child has a Reading Day UI storyboard, provided by Project Horseshoe.

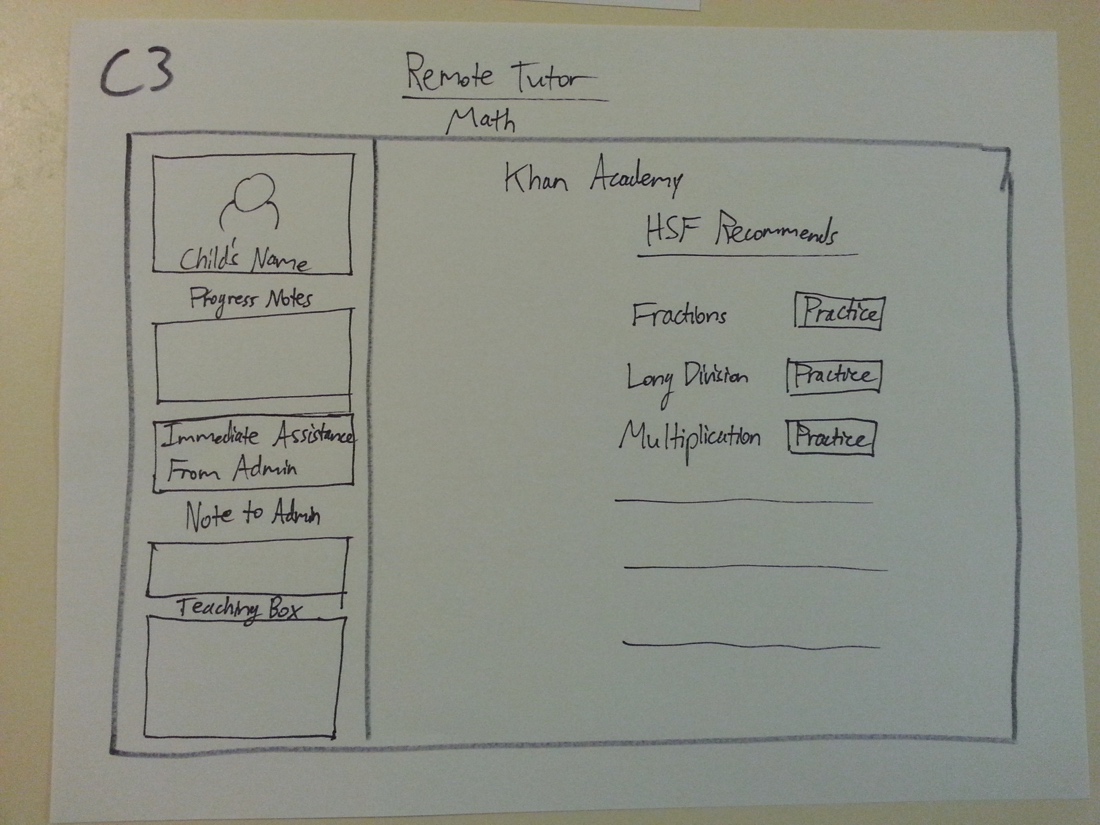


Image C3: Remote Tutor Session when Child has a Math Day UI storyboard, provided by Project Horseshoe.

The idea to embed talky.io onto Project Horseshoe's website, as shown on Image C2 and Image C3 was completely removed due to the aforementioned security concerns.

### 5.1.4 Child Use Case

In the case of a student, the user will open the page and be directed to the login page. This page contains only the site name across the top along with a single text box followed by a submit button. The text box will accept a username from the Child and check against a valid list of names when the submit button is clicked. If the user name is not valid, an error message will appear above the text box and the user will not be let into the site. Upon the submission of a valid user name, the Child will be logged into the site. A screen will welcome the Child by first name while displaying a summary of horseshoes earned during the last session. Due to the time restriction on this project, there are no plans to implement the feature which will keep track of horseshoes during this semester. After viewing the summary, the Child will only be allowed to click an “Okay!” button in order to indicate that he or she has read the message and is ready to move forward. Several scenarios may occur here: math without Remote Tutor, reading without Remote Tutor, math with Remote Tutor, or reading with Remote Tutor. If it is a math without Remote Tutor day, the user will encounter a list of recommended math activities along with buttons which will allow for selection of the most preferred activity for the day. A reading without Remote Tutor day will also provide a list of recommendations for the Child to choose from. If it is a math with Remote Tutor day, the Child will be connected to talky.io and Khan Academy. On a reading with Remote Tutor day, the Child will be connected to Talky.io; due to licensing issues with some books and the preferences of some Children, a portion of student-tutor pairs will read from physical books over Talky.io rather than reading from the Amazon website.

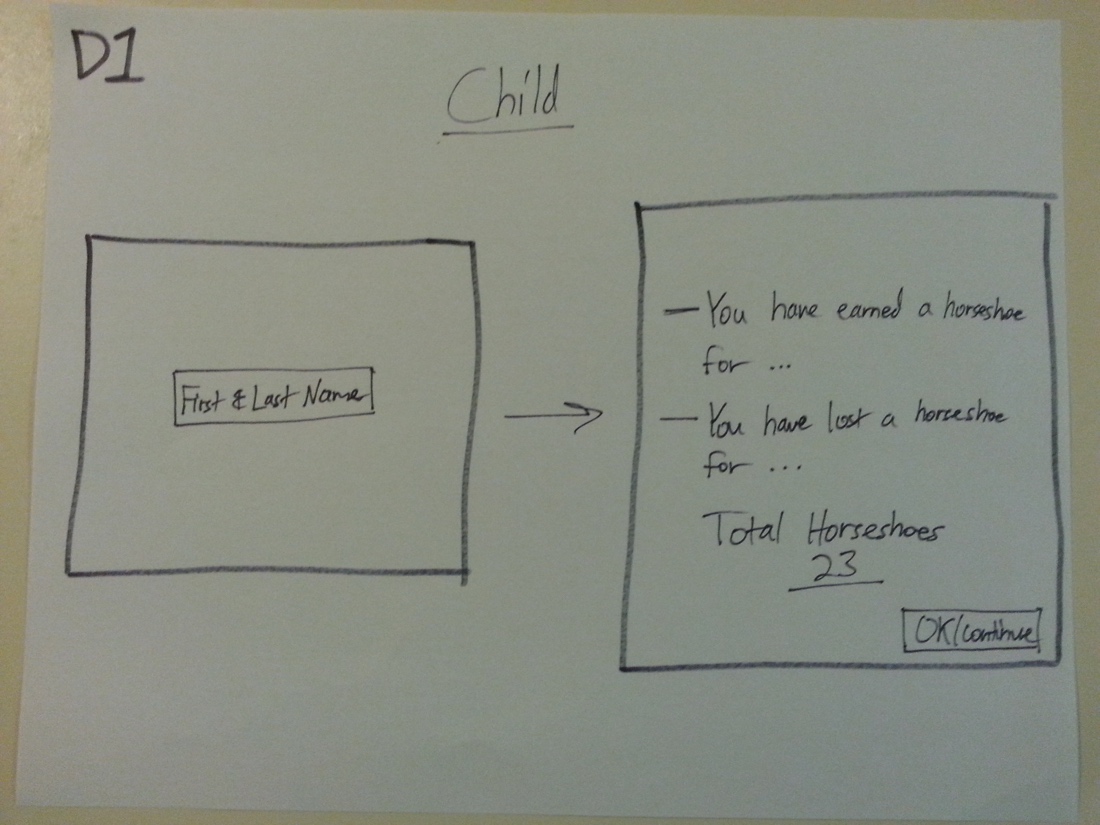


Image D1: Child Login and a page showing their horseshoes UI storyboard, provided by Project Horseshoe.

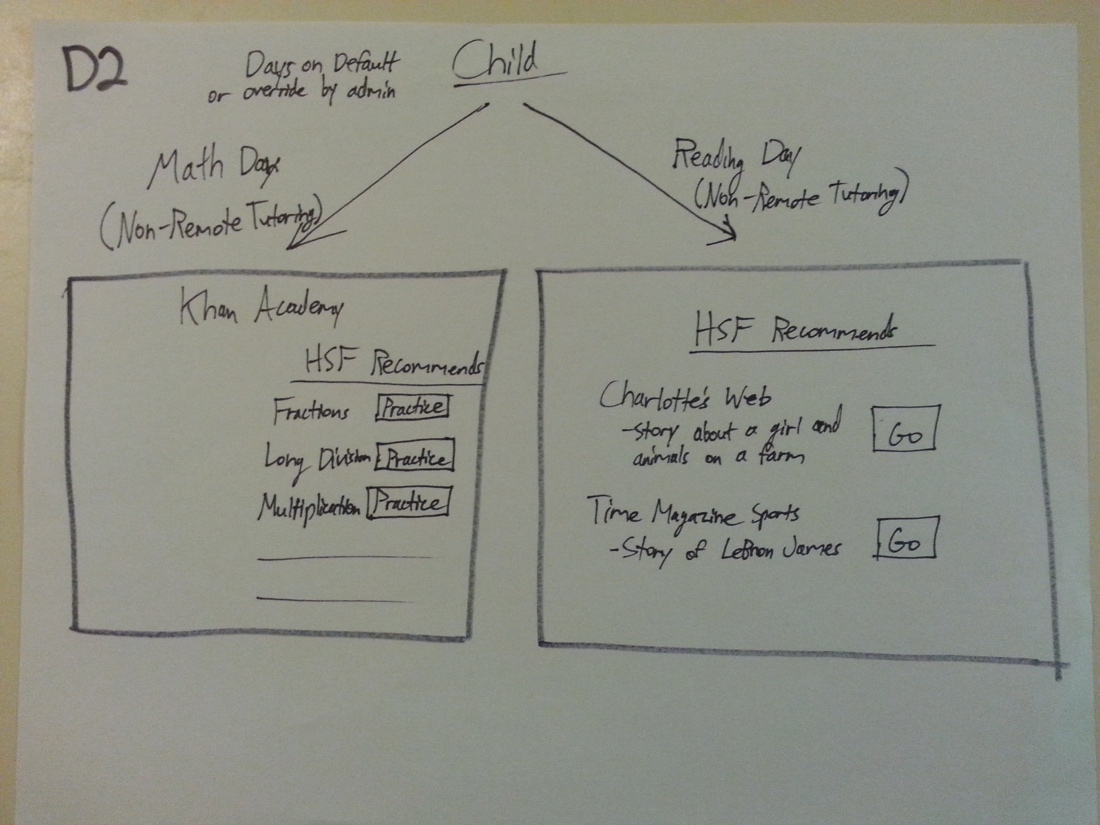


Image D2: Child Math and Reading without Remote Tutor UI storyboard, provided by Project Horseshoe.

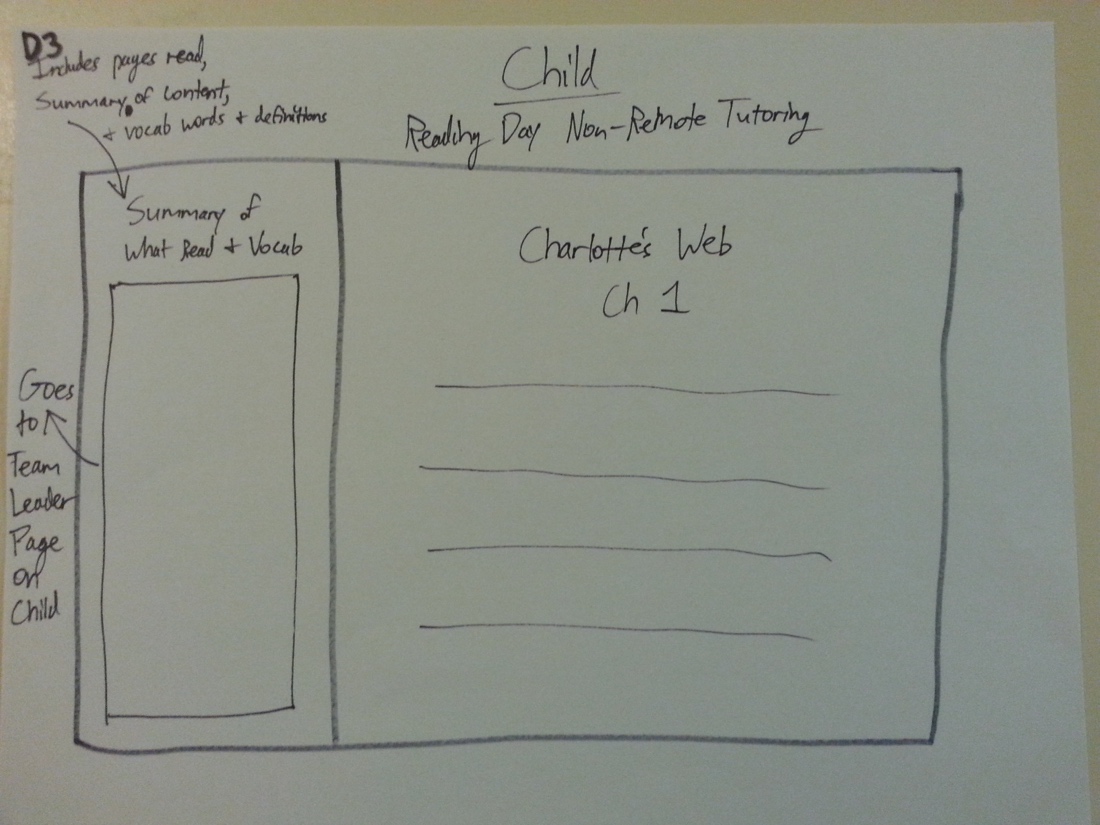


Image D3: Reading Dar for Child without a Remote Tutor UI storyboard, provided by Project Horseshoe.

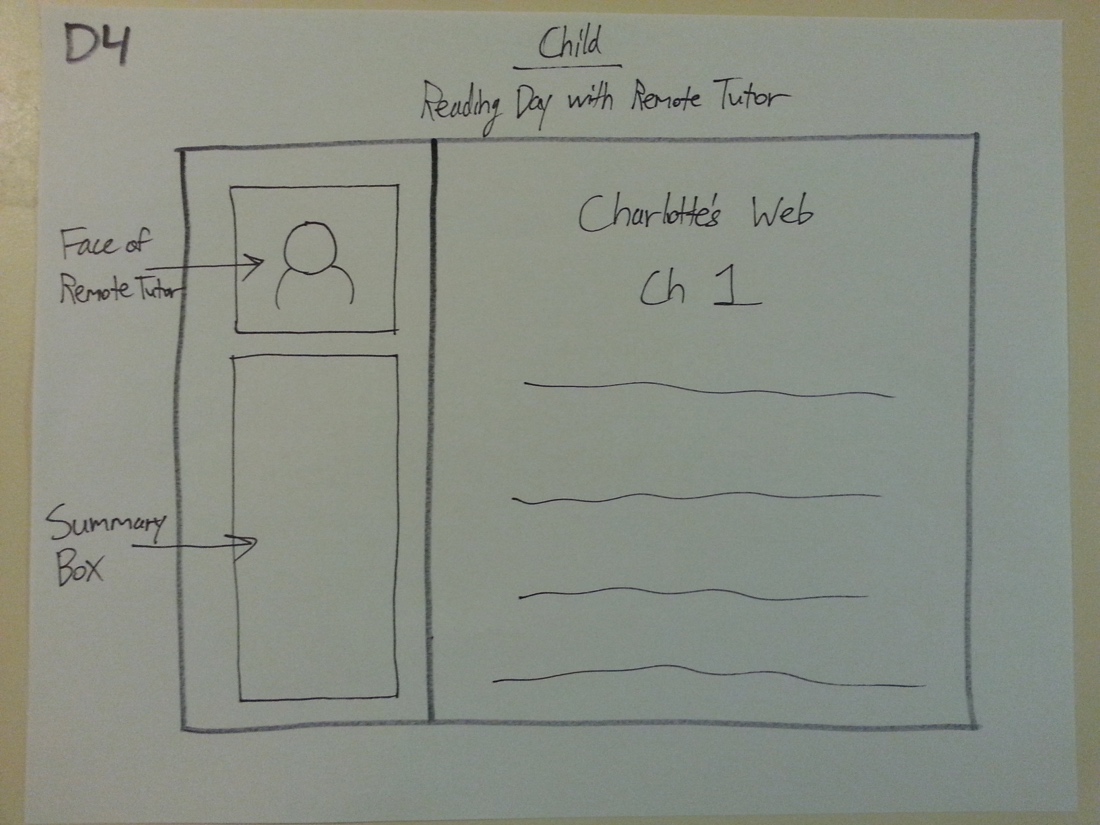


Image D4: Reading Day for Child with Remote Tutor UI storyboard, provided by Project Horseshoe.

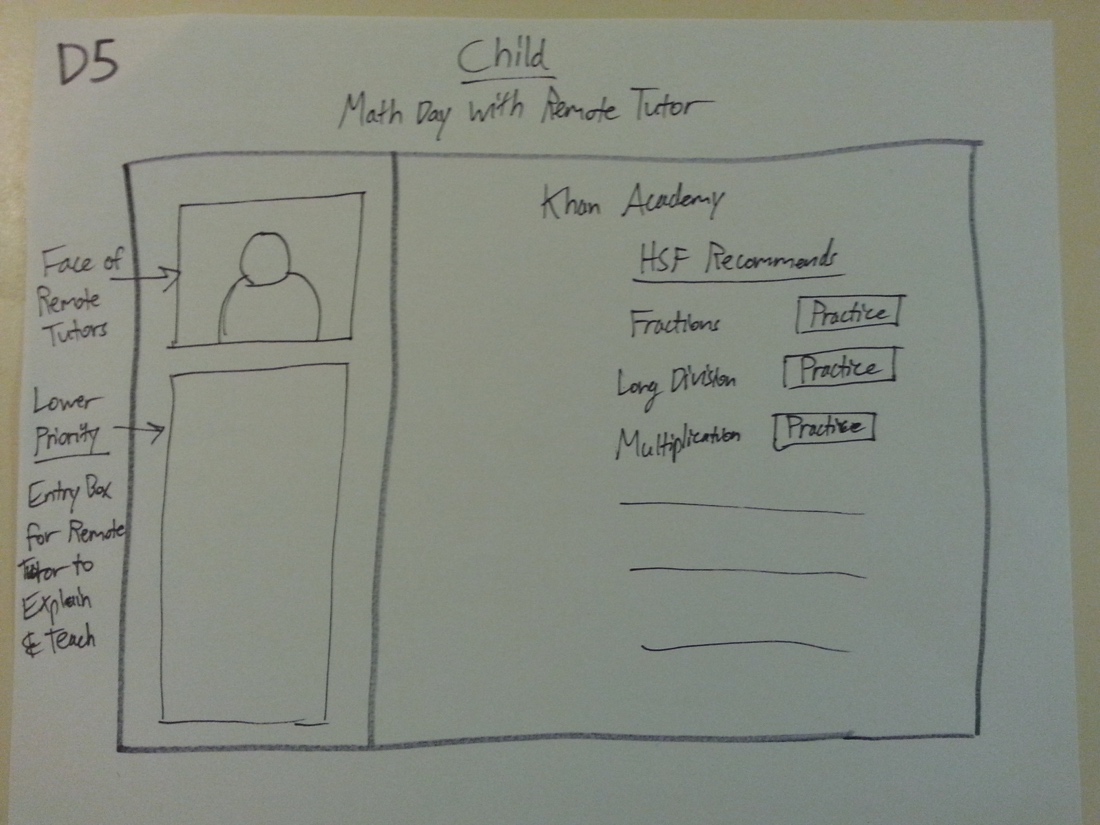


Image D5: Math Day for Child working with Remote Tutor UI storyboard, provided by Project Horseshoe.

The idea to embed talky.io onto Project Horseshoe's website, as shown on Image D4 and Image D5 was completely removed due to the aforementioned security concerns.

## 5.2 Structure

The components chosen for this project are the Bootstrap framework, Hyper Text Mark-Up Language (HTML), Cascading Styling Sheets (CSS), JavaScript, the jQuery library, and Hypertext Preprocessor (PHP).

In order to maintain a separation of concerns and cater to the requirement for a minimal user interface, each HTML file is designed for a single task. For example, the index.html file only handles logging in while the welcome.html only handles welcoming users. These files use jQuery to listen for events that occur on HTML elements in order to validate, redirect, and transition. The HTML files are their intended purpose are as follows: index.html(this is the login page. It is the first page that should be shown), loading.html(this is the page shown when loading a Child's page), math.html(this is the page that a Child will see that contains recommendations made by their Team Leader. It loads after the Child leaves their welcome page when a Child is not connected to a Remote Tutor), secured.html(this is the page that contains a field for the user to input their password), waiting.html(this is the page Remote Tutors will see while they are waiting to be paired), welcome.html(this is the welcome page that is viewed by the Child).

The Bootstrap framework was selected for this website primarily for its well-known simplicity in implementing responsive design – a requirement due to the varying nature of hardware used by Horseshoe Farm. All components provided by Bootstrap are in the bootstrap folder at root.

CSS files are maintained in the root css folder and include stylesheets from Bootstrap, Bootswatch.com, and custom design.

PHP is used for the database component. The files and their intended purpose are as follows: connectPairs.php(used for matching a Child to a Remote Tutor), connection.php(used for loading the correct talky.io for Children and Remote Tutors), lesson.php(used for loading recommendations), lessonHistory.php(), login.php(used for username login), populateTutors.php(used for populating the drop down menu for Admins to select only Remote Tutors that are online), secured.php(used for logging in with a password), test.php(used for testing purposes), updateRoster.php(used for updating who is currently online), welcomeAdmin.php(used for the Admin welcome page), and welcomeStudent.php(used for the Child welcome page). PHP files are maintained in their own folder and have now been implemented.

Due to the transition from WAMP to x10hosting, the HTML files had to be moved to the root and the name of login.html has been renamed to index.html. Instead they may be reached from the main folder. Instead of having separate HTML files for the different welcome pages for different users, there is now one HTML welcome page folder. A folder called web was added. It was added during the implementation of PDFs. It's purpose is unknown, and can likely be deleted.

## 5.3 Interfaces

The external components selected for this project are PHP, MySQL, and Apache. The three work together to create a local development environment while interfacing with back-end code.

### 5.3.1 PHP: Hypertext Preprocessor (PHP)

Due to Horseshoe Farm's current hosting arrangement with GoDaddy, technologies were restricted to languages that could be supported on a Linux server such as PHP, PERL, or Python. PHP had previously been worked with by two members while no one else had experience with either of the other two languages. PHP also has a long standing reputation as a traditional server-side language which would allow for wide-spread support online whenever problems were encountered.

### 5.3.2 MySQL

After settling on PHP, MySQL was selected as a compatible database for the language. Once again, this was restricted based upon support from GoDaddy. The classic Linux server from GoDaddy exclusively supports MySQL. MySQL and PHP are often used in development together, therefore using these two technologies in conjunction could be considered standard.

### 5.3.3 Apache

After determining that PHP and MySQL had to successfully work on local machines, the team began to search for services that could run localhost. Apache was free, reliable, and well supported, so an attempt was made to set up all three individual components.

## 5.4 Assumptions and Dependencies

A previous dependency for our past cycles were WAMP for Windows and MAMP for Macbooks. Because of issues with WAMP, an alternative was found and WAMP is no longer used. What our team is now using is called x10hosting, a website that offers free hosting. One of the drawbacks of using x10hosting is the time it takes to update any changes made to the site.

We decided it would be best to use talky.io as the communication platform for the days there are remote tutoring, because it is a free service that Project Horseshoe Farm is already familiar with and there is an option for the Remote Tutor to share his/her screen with the child on reading days. This decision prevents our website from being compatible with Internet Explorer, therefore, our assumption is that the end users will be using Chrome or Firefox as their browser.

Another assumption is that it will be possible to interface with websites such as read.amazon.com or khanacademy.com. Depending on the security levels of these popular websites, it may be difficult to capture meta-data such as most recently read page number storage or last viewed video. Similarly, because this website is built on so many components, it is dependent on these components remaining reliable, up to date, and available. For example, if the Bootstrap framework or jQuery library was suddenly removed, the team would be left to develop with the last version downloaded or find new tools.

## 5.5 Planned Versus Actual Code

Goals for this cycle were to get our component on the GoDaddy website and that key features were implemented. Due to time constraints and being able to move to x10hosting, the focus was shifted to ensuring the key features were implemented. The remaining features that were completed this cycle were Admin pairing a Child with a Remote Tutor, Team Leader making recommendations, the Remote Tutor Waiting View, and the Child and Remote Tutor connecting to talky.io.

# 6 Management Plan

The management plan is a high-level schedule indicating tasks and task assignments. It includes the following:

* User Story or tasks under development in this cycle.
* Architecture Development
* Testing
* Team member assignments.
* Planned start and end dates for each user story and/or task this cycle. A Gantt chart is included as a supporting figure for the start and end dates.
* Planned code/feature freeze date.

## 6.1 Tasks Under Development

Tasks for this cycle currently under development or soon to be under development:

* Web-hosting solutions for component
* User Experience and User Story additions or changes
* Hand-off of web-component to sponsor

6.1.1 User Stories:

1. ~~Student Log In~~
   1. ~~Mark~~
      1. **~~Completed~~**
2. ~~Student View Notifications~~
   1. ~~John~~
      1. **~~Completed~~**
3. ~~Student Auto Redirect to Lesson~~
   1. ~~Mark, Allison~~
      1. **~~Completed~~**
4. Student/Tutor start video chat service
   1. Mark, Cesar, Allison
      1. Simply link to talky.io
      2. Speak with ProjectHSF regarding which talky.io sites they use
      3. If math day,  Student goes to math worksheet/khan first, talky.io is second view
      4. If Reading day, Student goes only to talky.io
      5. Use sockets to push notification from database to student and tutor
         1. Timeout method is not at all efficient or desirable, but it works.
5. ~~Student view math worksheets~~
   1. ~~Cesar, Allison~~
      1. **~~Completed~~**
6. Student resume reading from last page
   1. Team
      1. Can research, at the very least, for the next team.
7. ~~Student preview lesson for the day~~
8. Student/tutor screen share control
   1. John, Amanda
      1. First research ways to implement
      2. Currently looking into opening executable from javascript function
      3. Executable can be batch, python, etc.; needs to use command line to:
         1. Log user into teamviewer
            1. Desktop application dependent
         2. Use Chrome Remote Desktop
            1. Browser dependent
            2. Test it!
9. ~~Tutor view daily schedule (student and lesson)~~
   1. ~~Amanda, Mark, John~~
      1. ~~Tutor can see students and their corresponding lessons~~
10. ~~Secured log in~~
    1. ~~Allison~~
       1. **~~Completed~~**
11. ~~Admin view student statistics~~
12. ~~Admin view student/tutor pairing history~~
13. Team Leader plan student lesson
    1. Allison, Cesar
       1. Diagram B1, recommended lesson for the day for the student
       2. Date of lesson, Remote tutor day?, worksheets or links to recommendation
       3. Team leader will only be able to view his/her students
       4. <http://tutorialzine.com/2014/09/cute-file-browser-jquery-ajax-php/>
14. Auto login into external websites
    1. John, Amanda
       1. Student View (User story 3)uses this
       2. Log in into KhanAcademy and Kindle using credentials
          1. Each student has their own gmail account to get into KhanAcademy
          2. Master account for Kindle
       3. Different Days
          1. Math day opens on a students computer
          2. Reading day opens on tutors computer
15. ~~Admin pair student to tutor~~
    1. ~~Allison, Cesar~~
    2. ~~THIS WEEK~~
       1. Diagram A1
16. Tutor Waiting View
    1. Amanda, Mark
       1. Diagram C1
          1. Have it work with the Database
          2. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>
          3. Php modify table when logged in
          4. Talk first, reading pdfs/kindle second view
       2. This waiting view needs a redirect. It redirects if true when it pings
          1. Pings for student's name = 'null' which is 'false'
          2. Pings for student's name = 'not null' which is 'true'
17. Log-out/Auto log-out
    1. Allison, Cesar
       1. This user story arose due to the necessity for having users exit the system, so that the database can be up to date, and admins can only pair users

**User Story Priority (Numerical Ranking by priority, and using MoSCoW)**

|  |  |  |
| --- | --- | --- |
| **M** | **4, 17** | on admin view(15) |

|  |  |  |
| --- | --- | --- |
| **M** | **13** | the Team Leader view needs to be made |

|  |  |  |
| --- | --- | --- |
| **S** | **14** | does not seem feasible |

|  |  |  |
| --- | --- | --- |
| **W** | **6** | definitely not feasible. |

### 6.1.2 Testing

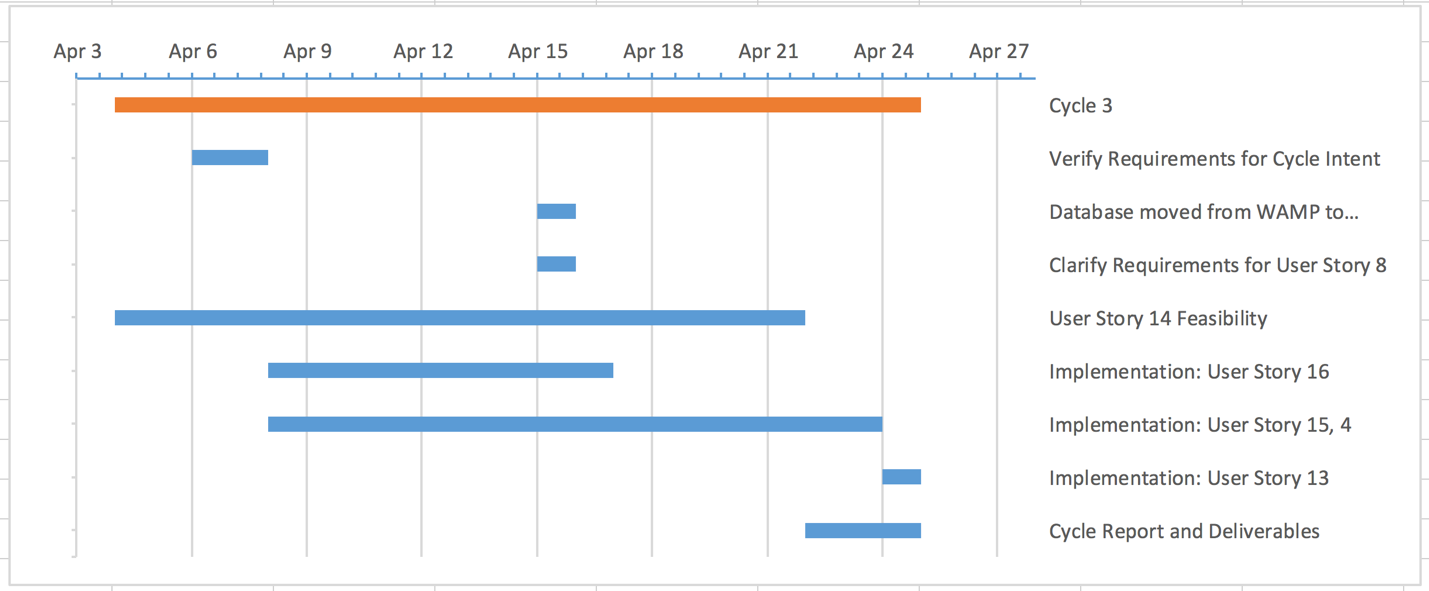
* + Amanda
* 4, 13, 14, 15, 16, 17

## 6.2 Task Assignments

|  |  |
| --- | --- |
| Team Member | Team Member Assignments and Tasks Assigned Throughout This Cycle |
| Amanda Bailey | **User Stories**: 8, 14, 16  **Cycle Repor**t: Executive Summary Documentation, Design Documentation  **Team Role**: correspondence, team process management, UX Design, testing lead, architecture technology research |
| John Carroll | **User Stories**: 8, 14  **Cycle Report**: Project Introduction Documentation: Future Work, Management Planning, Appendices  **Team Role**: team process management, correspondence, architecture technology research, UX Design |
| Lanxin (Mark) Ma | **User Stories**:  **Cycle Report**: User Stories Documentation, Lessons Learned  **Team Role**: technology research, UX Design |
| Allison Macdonald | **User Stories:** 4, 13, 15, 17  **Cycle Report:** Lessons Learned, References  **Team Role:** project lead, team process management, web development lead, correspondence lead, architecture technology research, UX Design |
| Cesar Sanchez | **User Stories:** 4, 13, 15, 17  **Cycle Report:** Risk Mitigation, Project Introduction Documentation  **Team Role:** architecture technology research, tester, UX Design, co-lead web developer |

## 6.3 Development Schedule

### 6.3.1 Gantt chart:



## 6.4 Planned Code / Feature Freeze

April 25th, 2016 at 11:59PM

# 7 Risk Mitigation

In order to minimize risk in this project, after each stage of development we will analyze our changes, perform thorough testing, and attempt to locate any potential vulnerabilities or avenues of attack. There will also be security measures pre-emptively implemented to prevent common web attacks such as cross-site scripting (XSS).

Security measures:

* Specify the UTF-8 charset in our HTML pages to prevent a UTF-7 attack. If the encoding style is not declared to be UTF-8 and a certain or out-of-date browser is used, then an attacker could insert a UTF-7 byte sequence and execute a script on the page.
* HTML-escape user input (characters such as <, >, and &) as well as URLs. In addition, we need to ensure that we do not allow user input as an unquoted attribute or as an attribute that is interpreted as JavaScript. If we did not place safeguards against character escaping, then an attacker could insert their own commands/scripts into the HTML of the page and execute them, possibly stealing information such as Admin login credentials.
* Validation of URLs and CSS values is important for the above reasons, as they can also be used to insert an attacker's code into a page and possibly steal information.
* Disallow user-provided HTML. Not disallowing this is an easy way to introduce many different XSS problems, including user-injected malicious HTML code. User-provided HTML should only be allowed for cases such as formatting of text, but in the context of this project, there is no clear reason to allow user-generated HTML.
* Prevent DOM-based XSS by not including user input in JavaScript-generated HTML code. To prevent this, we only need to use proper DOM methods to ensure that the input is read as text rather than HTML.

8 Test Plan and Test Procedures

8.1 Test Plan

There are two types of testing that will be necessary for the website: compatibility and functionality. The compatibility portion will include testing the website in Chrome and in Firefox. Testing the functionality of the code will include ensuring that the test procedures pass. In this cycle we moved our project onto x10host website all of the tests were performed to ensure the progress so far is still operable. New features that were implemented that required testing were Remote Tutor login, Remote Tutor waiting page, and Remote Tutor connect to lesson and talky.io, Admin plan lesson, and Child connect to talky.io.

8.2 Test Procedures

8.2.1 Child Login: Name not in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an invalid name from the database into the text box. Click 'Okay!' | An error message should appear. |  | 4/24/2016 |

8.2.2 Child Login: Name in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid Child name from the database into the text box. Click ‘Okay!’ | Should be redirected to a page showing what the Child worked on last time. |  | 4/24/2016 |

8.2.3 Child View Math Worksheets: No Remote Tutor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid Child name from the database into the text box. Click ‘Okay!’ | Should be redirected to a page showing what the Child worked on last time. |  | 4/24/2016 |
| 2 | Click ‘Begin Today's Lesson!’ | Should be taken to a page with a pdf of a math worksheet. |  | 4/24/2016 |

8.2.4 Child Auto-redirect to Lesson & Start Video Chat: Reading Day

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Sign in as Child who will be having a reading day with a Remote Tutor. | Should be redirected to a page showing what the Child worked on last time. |  | 4/24/2016 |
| 2 | Click ‘Begin Today's Lesson!’ | Should be directed to a talky.io chat. | This step requires the Admin to pair Child with a Remote Tutor. Kindle should load on the Remote Tutorwho is paired with this Child. | 4/24/2016 |

8.2.4 Child Auto-redirect to Lesson & Start Video Chat: Math Day

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Sign in as Child who will be having a reading day/tutoring session. | Should be redirected to a page showing what the Child worked on last time. |  | 4/24/2016 |
| 2 | Click ‘Begin Today's Lesson!’ | Should be directed to the Kahn Academy and a talky.io chat with Remote Tutor. | This step requires the Admin to pair Child with a Remote Tutor. | 4/24/2016 |

8.2.5 Admin Login: Name not in Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an unacceptable Admin name from the database into the text box. Click ‘Okay!’ | After hitting 'Okay!' an error message should appear. |  | 4/24/2016 |

8.2.6 Admin Login: Name in Database with Incorrect Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Admin name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter an invalid password into the password box and hit enter. | After hitting enter, an error message should appear. | Password should be hidden. | 4/24/2016 |

8.2.7 Admin Login: Name in Database with Correct Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Admin name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting enter, should log in to view the Admin welcome page | Password should be hidden. | 4/24/2016 |

8.2.8 Remote Tutor Login: Name not in Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an unacceptable Remote Tutor name from the database into the text box. Click ‘Okay!’ | After hitting 'Okay!' an error message should appear. |  | 4/24/2016 |

8.2.9 Remote Tutor Login: Name in Database & Incorrect Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter an invalid password into the password box and hit enter. | After hitting ‘Okay!’ an error message should appear | Password should be hidden. | 4/24/2016 |

8.2.10 Remote Tutor Login: Name in Database & Correct Password: Waiting Page

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to a waiting page. In the database the Remote Tutor should be marked as being online. | Password should be hidden. | 4/24/2016 |
| 3 | No action on part of Remote Tutor. Requires action from Admin. | After being paired with a Child, the Remote Tutor’s pages should automatically load. | In order for the tutor to be redirected to their lesson, an Admin has to pair a Child to a Remote Tutor | 4/24/2016 |

8.2.11 Remote Tutor paired with Child: Math Day

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to a waiting page. In the database the Remote Tutor should be marked as being online. | Password should be hidden. | 4/24/2016 |
| 3 | No action on part of Remote Tutor. Requires action from Admin. | After being paired with a Child, talky.io should automatically load on the Remote Tutors page, while Kahn Academy will open on the Child’s page. | In order for the tutor to be redirected to their lesson, an Admin has to pair a Child to a Remote Tutor | 4/24/2016 |

8.2.12 Remote Tutor paired with Student: Reading Day

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to a waiting page. In the database the Remote Tutor should be marked as being online. | Password should be hidden. | 4/24/2016 |
| 3 | No action on part of Remote Tutor. Requires action from Admin. | After being paired with a Child, talky.io and kindle should load. | In order for the tutor to be redirected to their lesson, an Admin has to pair a Child to a Remote Tutor | 4/24/2016 |

8.2.13 Remote Tutor Logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to a waiting page. In the database the Remote Tutor should be marked as being online. | Password should be hidden. |  |
| 3 | While you are signed in as a Remote Tutor, hit the log out button. | Remote Tutor should be logged out and the online status in the database should be switched to false |  |  |

8.2.14 Admin Logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Admin name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting enter, should log in to view the Admin welcome page | Password should be hidden. |  |
| 3 | While you are signed in as a remote Admin, hit the log out button. | Admin should now be logged out. |  |  |

8.2.15 Remote Tutor Auto-logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Remote Tutor name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to a waiting page. In the database the Remote Tutor should be marked as being online. | Password should be hidden. |  |
| 1 | Stay signed in as a Remote Tutor for 90 minutes. | Remote Tutor should be logged out and the online status in the database should be switched to false |  |  |

8.2.16 Admin Auto-logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Admin name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting enter, should log in to view the Admin welcome page | Password should be hidden. |  |
| 3 | Stay signed in as an Admin for 90 minutes. | Admin should be logged out and the online status in the database should be switched to false |  |  |

8.2.17 Child Auto-logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid Child name from the database into the text box. Click ‘Okay!’ | Should be redirected to a page showing what the Child worked on last time. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Stay signed in as a Child for 90 minutes. | Child should be logged out and the online status in the database should be switched to false |  |  |

8.2.18 Admin Pair Child and Remote Tutor: Child and Remote Tutor Online

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Admin name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting enter, should log in to view the Admin welcome page | Password should be hidden. | 4/24/2016 |
| 3 | Click the drop down menu for a Child that is present and select a remote tutor. Hit the “Connect” button that is on the same row as the Child and Remote Tutor that were just paired. | The Child that was just paired with Remote Tutor should be redirected to talky.io and, if it is a math day, to Khan Academy.  The Remote Tutor that was just paired with a Child should be redirected from their waiting page to the same talky.io session and, if it is a reading day, to Kindle. | The dropdown menu of Remote Tutors should only show they ones currently online. | 4/24/2016 |

8.2.19 Team Leader Login: Name not in Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an unacceptable Team Leader name from the database into the text box. Click ‘Okay!’ | After hitting 'Okay!' an error message should appear. |  | 4/24/2016 |

8.2.20 Team Leader Login: Name in Database with Incorrect Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Team Leader name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter an invalid password into the password box and hit enter. | After hitting enter, an error message should appear. | Password should be hidden. | 4/24/2016 |

8.2.21 Team Leader Login: Name in Database with Correct Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid Team Leader name from the database into the text box. Click ‘Okay!’ | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting enter, the user should be logged in to view the Team Leader recommendations page. | Password should be hidden. | 4/24/2016 |

8.2.22 Team Leader Make Recommendations

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable team leader name into the database. | A password box should appear. |  | 4/24/2016 |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to the team leader recommendations page. | Password should be hidden. | 4/24/2016 |
| 3 | Make recommendations for a student | Student should now have recommendations |  | 4/24/2016 |

8.2.23 Team Leader Logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Team Leader name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to the Team Leader recommendations page. | Password should be hidden. |  |
| 3 | While you are signed in as a Team Leader, hit the log out button. | Team Leader should now be logged out. |  |  |

8.2.24 Team Leader Auto-logout

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter an acceptable Team Leader name into the database. | A password box should appear. | Because this has not been implemented, no test date has been provided. |  |
| 2 | Enter the valid password into the password box and hit enter. | After hitting ‘Okay!’ the user should be redirected to the Team Leader recommendations page. | Password should be hidden. |  |
| 3 | Stay signed in as and Team Leader for 90 minutes. | Team Leader should now be logged out. |  |  |

# 9 Lessons Learned

In this cycle, we have learned five important lessons. Firstly, we have learned how precious time is and the importance of time management. We have to use all the time we have wisely so we can keep up with the project. Secondly, our team has learnt to plan things in advance, so we would know what to do next and stay on track with the project. All our team members have very busy weekly schedules, and we have learned to carefully plan our meetings in a way such that we can both discuss about the progression of this project as well as avoid interfering with every team member's daily schedule. Thirdly, because this is a team project, we have learned the importance of teamwork and communication among team members and sponsors. Based on experience, we have learned that staying in touch with the sponsor gives us the motivation to progress and succeed, since our sponsor was very helpful and willing to give us ideas to complete the project. Fourth, we learned to gain more experience with using web development, since having a good knowledge of web development gives us a good estimate of how long it takes for a user story or process to be completed. If we know how long it takes to complete a process, we would have better judgement and better planning on when and how to get things done. Finally, our group strongly grasped the concept of learning from failure, and felt like everywhere we lost points on is a reminder that there is a lesson to be learned from it.

# References

## References

[1] Bootswatch.com,. "Bootswatch: Free Themes For Bootstrap". N.p., 2016. Web. 27 Feb. 2016.

[2] Codecademy,. "Learn To Code". N.p., 2016. Web. 18 Feb. 2016.

[3] CSS-Tricks,. "CSS-Tricks". N.p., 2016. Web. 27 Feb. 2016.

[4] Farm, Project. "Project Horseshoe Farm". *Projecthorseshoefarm.com*. N.p., 2016. Web. 12 Feb. 2016.

[5] GmbH, appsolute. "MAMP & MAMP PRO". *Mamp.info*. N.p., 2016. Web. 2 Mar. 2016.

[6] GoDaddy,. "Domain Names | The World's Largest Domain Name Registrar - Godaddy". N.p., 2016. Web. 24 Feb. 2016.

[7] Mark Otto, and Bootstrap contributors. "Bootstrap · The World's Most Popular Mobile-First And Responsive Front-End Framework.". *Getbootstrap.com*. N.p., 2016. Web. 27 Feb. 2016.

[8] Stackoverflow.com,. "Stack Overflow". N.p., 2016. Web. 4 Mar. 2016.

[9] WampServer,. "Wampserver". N.p., 2016. Web. 2 Mar. 2016.

# A Appendix A Supporting Documents

## A.1 Status Reports

Status Reports are located one the CD in separate files.

## A.2 Meeting Minutes

Total of 5 Team Meetings for 1210 minutes (20.17 hours)  
Total of 3 TA Meetings for 245 minutes (4.08 hours)

Total of 2 Sponsor Meetings for 225 minutes (3.75 hours)

Below is a breakdown of the meeting minutes along with relevant notes for each meeting.

### A.2.1 Team Meetings

|  |
| --- |
| 4/6/2016: 80 minutes |
| Full Team |
| Updating user stories and taskboard. Assigning user stories.      Talked about godaddy hosting, getting our component on the website, and how it should be a part of a user story.   * User story: 14 - User access to website       Ask Sponsor's about structure in regards to student views, talky and lesson on one page?      Thought about how PHSF would add new PDF's for math worksheets.  How we should make the student view more dynamic, BUT time will not allow for that this cycle.  The same goes for a lot of more dynamic functionality for the website – time is not granted.    John will start copying the updated taskboard to the team meeting minutes, here:  Taskboard:  **Assignments/Progress:**   * **User Story Priority (Numerical Ranking)**   1. **14**   2. **15**   3. **16, 4, 8**      + **16 uses 4**      * **Research notes asked by Chapman**   + Alternatives to talki.io and teamviewer   + Purpose: pass on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   + Amanda     - 14, 4, 8, 13      * **Current User Stories:**   + ~~Student Log In~~     - 1. ~~Mark~~          1. **~~Completed~~**   + ~~Student View Notifications~~     - 1. ~~John~~          1. **~~Completed~~**   + ~~Student Auto Redirect to Lesson~~     - 1. ~~Mark, Allison~~          1. **~~Completed~~**   + Student/Tutor start video chat service     - 1. Mark, Cesar, Allison          1. Simply link to talky.io          2. Speak with ProjectHSF regarding which talky.io sites they use   + ~~Student view math worksheets~~     - 1. ~~Cesar, Allison~~          1. **~~Completed~~**   + Student resume reading from last page     - 1. Team          1. Can research, at the very least, for the next team.   + ~~Student preview lesson for the day~~   + Student/tutor screen share control     - 1. John, Amanda          1. First research ways to implement          2. Currently looking into opening executable from javascript function          3. Executable can be batch, python, etc.; needs to use command line to:   Log user into teamviewer  Desktop application dependent  Use Chrome Remote Desktop  Browser dependent  Test it!   * + ~~Tutor view daily schedule (student and lesson)~~     - 1. ~~Amanda, Mark, John~~          1. ~~Tutor can see students and their corresponding lessons~~   + ~~Secured log in~~     - 1. ~~Allison~~          1. **~~Completed~~**   + ~~Admin view student statistics~~   + ~~Admin view student/tutor pairing history~~   + Team leader plan student lesson     - 1. Allison, Cesar          1. Diagram B1, recommended lesson for the day for the student          2. Date of lesson, Remote tutor day?, worksheets or links to recommendation          3. Team leader will only be able to view his/her students   + User access to website     - 1. John & Team          1. Architectural/Development Stack          2. Ability to access database and web component located on a sub domain which is hosted through GoDaddy web-hosting   + Admin pair student to tutor     - 1. Allison, Cesar       2. THIS WEEK          1. Diagram A1   + Tutor Waiting View     - 1. Amanda, Mark          1. Diagram C1          2. Have it work with the Database |

|  |
| --- |
| 4/10/2016: 120 minutes |
| Amanda, Allison, John were present |
| Briefed John on Friday's meeting, because he was gone due to a job interview.  Changed 14 to a new userstory.    The new Taskboard:    **Assignments/Progress:**   * **User Story Priority (Numerical Ranking)**   1. **15 -> 16, 4**   2. **14**   3. **8, 13**      1. Nice but not super needed (MoSCoW)   4. **6**      * **Research notes asked by Chapman**   1. Alternatives to talki.io and teamviewer   2. Purpose: pass on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   1. Amanda      1. 4, 8, 13, 14, 15, 16      * **Current User Stories:**   1. ~~Student Log In~~      1. ~~Mark~~         1. **~~Completed~~**   2. ~~Student View Notifications~~      1. ~~John~~         1. **~~Completed~~**   3. ~~Student Auto Redirect to Lesson~~      1. ~~Mark, Allison~~         1. **~~Completed~~**   4. Student/Tutor start video chat service      1. Mark, Cesar, Allison         1. Simply link to talky.io         2. Speak with ProjectHSF regarding which talky.io sites they use   5. ~~Student view math worksheets~~      1. ~~Cesar, Allison~~         1. **~~Completed~~**   6. Student resume reading from last page      1. Team         1. Can research, at the very least, for the next team.   7. ~~Student preview lesson for the day~~   8. Student/tutor screen share control      1. John, Amanda         1. First research ways to implement         2. Currently looking into opening executable from javascript function         3. Executable can be batch, python, etc.; needs to use command line to:            1. Log user into teamviewer   Desktop application dependent   * + - * 1. Use Chrome Remote Desktop   Browser dependent  Test it!   * 1. ~~Tutor view daily schedule (student and lesson)~~      1. ~~Amanda, Mark, John~~         1. ~~Tutor can see students and their corresponding lessons~~   2. ~~Secured log in~~      1. ~~Allison~~         1. **~~Completed~~**   3. ~~Admin view student statistics~~   4. ~~Admin view student/tutor pairing history~~   5. Team leader plan student lesson      1. Allison, Cesar         1. Diagram B1, recommended lesson for the day for the student         2. Date of lesson, Remote tutor day?, worksheets or links to recommendation         3. Team leader will only be able to view his/her students   6. Auto login into external websites      1. John, Amanda         1. Student View (User story 3)uses this         2. Log in into KhanAcademy and Kindle using credentials            1. Each student has their own gmail account to get into KhanAcademy            2. Master account for Kindle         3. Different Days            1. Math day opens on a students computer            2. Reading day opens ons tutors computer   7. Admin pair student to tutor      1. Allison, Cesar      2. THIS WEEK         1. Diagram A1   8. Tutor Waiting View      1. Amanda, Mark         1. Diagram C1         2. Have it work with the Database         3. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>         4. Php modify table when logged in |

|  |
| --- |
| 4/13/2016: 90 minutes |
| Full Team |
| Talked about what did 14 need.  Then reviewed what needed to work for 16 and 4 (tutor and student login views)  Cesar, Amanda, and John reviewed the process for 15, and it would affect the behavior of 16 and 4.  14 would be a part of 16 and 4's use of 3 which auto redirects to lesson resources along with Talky.io.    **Assignments/Progress:**   * **User Story Priority (Numerical Ranking)**   1. **15 -> 16, 4**   2. **14**   3. **8, 13**      1. Nice but not super needed (MoSCoW)   4. **6**      * **Research notes asked by Chapman**   1. Alternatives to talki.io and teamviewer   2. Purpose: pass-on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   1. Amanda      1. 4, 8, 13, 14, 15, 16      * **Current User Stories:**   1. ~~Student Log In~~      1. ~~Mark~~         1. **~~Completed~~**   2. ~~Student View Notifications~~      1. ~~John~~         1. **~~Completed~~**   3. ~~Student Auto Redirect to Lesson~~      1. ~~Mark, Allison~~         1. **~~Completed~~**   4. Student/Tutor start video chat service      1. Mark, Cesar, Allison         1. Simply link to talky.io         2. Speak with ProjectHSF regarding which talky.io sites they use         3. If math day,  Student goes to math worksheet/khan first, talky.io is second view         4. If Reading day, Student goes only to talky.io   5. ~~Student view math worksheets~~      1. ~~Cesar, Allison~~         1. **~~Completed~~**   6. Student resume reading from last page      1. Team         1. Can research, at the very least, for the next team.   7. ~~Student preview lesson for the day~~   8. Student/tutor screen share control      1. John, Amanda         1. First research ways to implement         2. Currently looking into opening executable from javascript function         3. Executable can be batch, python, etc.; needs to use command line to:            1. Log user into teamviewer   Desktop application dependent   * + - * 1. Use Chrome Remote Desktop   Browser dependent  Test it!   * 1. ~~Tutor view daily schedule (student and lesson)~~      1. ~~Amanda, Mark, John~~         1. ~~Tutor can see students and their corresponding lessons~~   2. ~~Secured log in~~      1. ~~Allison~~         1. **~~Completed~~**   3. ~~Admin view student statistics~~   4. ~~Admin view student/tutor pairing history~~   5. Team leader plan student lesson      1. Allison, Cesar         1. Diagram B1, recommended lesson for the day for the student         2. Date of lesson, Remote tutor day?, worksheets or links to recommendation         3. Team leader will only be able to view his/her students   6. Auto login into external websites      1. John, Amanda         1. Student View (User story 3)uses this         2. Log in into KhanAcademy and Kindle using credentials            1. Each student has their own gmail account to get into KhanAcademy            2. Master account for Kindle         3. Different Days            1. Math day opens on a students computer            2. Reading day opens on tutors computer   7. Admin pair student to tutor      1. Allison, Cesar      2. THIS WEEK         1. Diagram A1   8. Tutor Waiting View      1. Amanda, Mark         1. Diagram C1            1. Have it work with the Database            2. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>            3. Php modify table when logged in            4. Talk first, reading pdfs/kindle second view         2. This waiting view needs a redirect. It redirects if true when it pings            1. Pings for student's name = 'null' which is 'false'            2. Pings for student's name = 'not null' which is 'true' |

|  |
| --- |
| 4/17/2016: 100 minutes |
| Full Team |
| Assessed the need for a log out, and auto log out for user's sessions. That way the Admin view(15) can pair using only the currently active users in the database, namely the people in the database.    **Assignments/Progress:**   * **User Story Priority (Numerical Ranking by priority, and using MoSCoW)**  |  |  |  | | --- | --- | --- | | **M** | **4, 17** | on admin view(15) |  |  |  |  | | --- | --- | --- | | **M** | **14** | is in research |  |  |  |  | | --- | --- | --- | | **S** | **13** | the Team Leader view needs to be made |  |  |  |  | | --- | --- | --- | | **W** | **6** | definitely not feasible. |  * **Research notes asked by Chapman**   1. Alternatives to talki.io and teamviewer   2. Purpose: pass on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   1. Amanda      + 4, 13, 14, 15, 16, 17      * **Current User Stories:**   1. ~~Student Log In~~      + ~~Mark~~        1. **~~Completed~~**   2. ~~Student View Notifications~~      + ~~John~~        1. **~~Completed~~**   3. ~~Student Auto Redirect to Lesson~~      + ~~Mark, Allison~~        1. **~~Completed~~**   4. Student/Tutor start video chat service      + Mark, Cesar, Allison        1. Simply link to talky.io        2. Speak with ProjectHSF regarding which talky.io sites they use        3. If math day,  Student goes to math worksheet/khan first, talky.io is second view        4. If Reading day, Student goes only to talky.io        5. Use sockets to push notification from database to student and tutor           1. Timeout method is not at all efficient or desirable, but it works.   5. ~~Student view math worksheets~~      + ~~Cesar, Allison~~        1. **~~Completed~~**   6. Student resume reading from last page      + Team        1. Can research, at the very least, for the next team.   7. ~~Student preview lesson for the day~~   8. Student/tutor screen share control      + John, Amanda        1. First research ways to implement        2. Currently looking into opening executable from javascript function        3. Executable can be batch, python, etc.; needs to use command line to:           1. Log user into teamviewer   Desktop application dependent   * + - * 1. Use Chrome Remote Desktop   Browser dependent  Test it!   * 1. ~~Tutor view daily schedule (student and lesson)~~      + ~~Amanda, Mark, John~~        1. ~~Tutor can see students and their corresponding lessons~~   2. ~~Secured log in~~      + ~~Allison~~        1. **~~Completed~~**   3. ~~Admin view student statistics~~   4. ~~Admin view student/tutor pairing history~~   5. Team Leader plan student lesson      + Allison, Cesar        1. Diagram B1, recommended lesson for the day for the student        2. Date of lesson, Remote tutor day?, worksheets or links to recommendation        3. Team leader will only be able to view his/her students        4. <http://tutorialzine.com/2014/09/cute-file-browser-jquery-ajax-php/>   6. Auto login into external websites      + John, Amanda        1. Student View (User story 3)uses this        2. Log in into KhanAcademy and Kindle using credentials           1. Each student has their own gmail account to get into KhanAcademy           2. Master account for Kindle        3. Different Days           1. Math day opens on a students computer           2. Reading day opens on tutors computer   7. ~~Admin pair student to tutor~~      + ~~Allison, Cesar~~      + ~~THIS WEEK~~        1. Diagram A1   8. Tutor Waiting View      + Amanda, Mark        1. Diagram C1           1. Have it work with the Database           2. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>           3. Php modify table when logged in           4. Talk first, reading pdfs/kindle second view        2. This waiting view needs a redirect. It redirects if true when it pings           1. Pings for student's name = 'null' which is 'false'           2. Pings for student's name = 'not null' which is 'true'   9. Log-out/Auto log-out   Allison, Cesar   * + - 1. This user story arose due to the necessity for having users exit the system, so that the database can be up to date, and admins can only pair users |

|  |
| --- |
| 4/20/2016: 100 minutes |
| Full Team |
| -John researched Auto login US #14 solo, and also consulted his friend, Brandon Hurler, a CWS developer about it. Mostly feasibility research regarding options for user story.  -Allison worked on testing the POST method for US# 14, along with working with US# 4 which uses the Admin view(US# 15). She concluded that x10hosting is very slow when updating the database, so testing for active users is cumbersome and tests our patience.    John's notes regarding US# 14:  So, for US#: 14, I have done a lot of research on my own and have spoken to two of my most credible sources for web development (two of my friends). One works for CWS, the other at ENS. I have the options, and they do not lead to this user story being completed within the remaining time.    1. Central Authentication Server (need partnership with the other company).  2. API (unlikely).  3. Some other arrangement.    "It is possible, e.g. auaccess and canvas, however they would likely need a central authentication server to do so, or some sort of API to allow for that.  You would have to either set up a partnership with those sites and allow their auth through your portal on login, have a central authentication server, or pray they have an API for that. Short of that, you can't use your cookies on their sites - it won't be set up to allow that.  You'd likely have to contact each place directly and see if they have anything that can support that."  "Cookies will be fine for your own auth but it won't allow you to access anyone else's sites"    -Brandon      "I have researched forging cookies, and using the POST method, which duplicates the form and attempts to enter it in on the external site, as well as storing sessions for each user in the database." All i need is concise answers to that. Haha To be able to tell my sponsors, and for future groups to know    Awaiting concise answers: phone call with Brandon    POST method will not work, mostly due to anti-forgery, anti-hijacking implementations in place on the destination domain.    Session data is stored in cache and cookies. The issue is that a cookie is needed to have a session, and authentication is needed to have a cookie. Cookies come with an internal expiry. This expire date has a time of persistent, can set for indefinite, or whenever, but whenever it expires, it will no longer provide information    Typically works like: Send out cookie, get a CAS cookie, then you login with that.    1. CAS with partnership:  Ask Khan for a partnership, if they said yes, they would work with us to get us up and running. They may provide us with an API if they do not decide to work with us directly. They may provide us a way to tie into their system.    2. API  Perhaps using Google authentication API.  <https://developers.google.com/identity/>  <https://developers.google.com/api-client-library/javascript/features/authentication>    Khan authentication API  -http://api-explorer.khanacademy.org/group/api/v1/user  -https://github.com/Khan/khan-api/wiki/Khan-Academy-API-Authentication    Google SAML (SSO)  -https://developers.google.com/google-apps/sso/saml\_reference\_implementation?csw=1    SAML Single Sign-on for Khan  -https://www.bitium.com/khan-academy-saml-provider (not currently supported, but should be in the future)    3. Another Arrangement  Using iMacros:  <http://www.hacktabs.com/how-to-automate-login-to-website/>          We updated the priority list based off of user stories:    **Assignments/Progress:**   * **User Story Priority (Numerical Ranking by priority, and using MoSCoW)**  |  |  |  | | --- | --- | --- | | **M** | **4, 17** | on admin view(15) |  |  |  |  | | --- | --- | --- | | **M** | **13** | the Team Leader view needs to be made |  |  |  |  | | --- | --- | --- | | **S** | **14** | does not seem feasible |  |  |  |  | | --- | --- | --- | | **W** | **6** | definitely not feasible. |  * **Research notes asked by Chapman**   1. Alternatives to talki.io and teamviewer   2. Purpose: pass on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   1. Amanda      + 4, 13, 14, 15, 16, 17      * **Current User Stories:**   1. ~~Student Log In~~      + ~~Mark~~        1. **~~Completed~~**   2. ~~Student View Notifications~~      + ~~John~~        1. **~~Completed~~**   3. ~~Student Auto Redirect to Lesson~~      + ~~Mark, Allison~~        1. **~~Completed~~**   4. Student/Tutor start video chat service      + Mark, Cesar, Allison        1. Simply link to talky.io        2. Speak with ProjectHSF regarding which talky.io sites they use        3. If math day,  Student goes to math worksheet/khan first, talky.io is second view        4. If Reading day, Student goes only to talky.io        5. Use sockets to push notification from database to student and tutor           1. Timeout method is not at all efficient or desirable, but it works.   5. ~~Student view math worksheets~~      + ~~Cesar, Allison~~        1. **~~Completed~~**   6. Student resume reading from last page      + Team        1. Can research, at the very least, for the next team.   7. ~~Student preview lesson for the day~~   8. Student/tutor screen share control      + John, Amanda        1. First research ways to implement        2. Currently looking into opening executable from javascript function        3. Executable can be batch, python, etc.; needs to use command line to:           1. Log user into teamviewer   Desktop application dependent   * + - * 1. Use Chrome Remote Desktop   Browser dependent  Test it!   * 1. ~~Tutor view daily schedule (student and lesson)~~      + ~~Amanda, Mark, John~~        1. ~~Tutor can see students and their corresponding lessons~~   2. ~~Secured log in~~      + ~~Allison~~        1. **~~Completed~~**   3. ~~Admin view student statistics~~   4. ~~Admin view student/tutor pairing history~~   5. Team Leader plan student lesson      + Allison, Cesar        1. Diagram B1, recommended lesson for the day for the student        2. Date of lesson, Remote tutor day?, worksheets or links to recommendation        3. Team leader will only be able to view his/her students        4. <http://tutorialzine.com/2014/09/cute-file-browser-jquery-ajax-php/>   6. Auto login into external websites      + John, Amanda        1. Student View (User story 3)uses this        2. Log in into KhanAcademy and Kindle using credentials           1. Each student has their own gmail account to get into KhanAcademy           2. Master account for Kindle        3. Different Days           1. Math day opens on a students computer           2. Reading day opens on tutors computer   7. ~~Admin pair student to tutor~~      + ~~Allison, Cesar~~      + ~~THIS WEEK~~        1. Diagram A1   8. Tutor Waiting View      + Amanda, Mark        1. Diagram C1           1. Have it work with the Database           2. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>           3. Php modify table when logged in           4. Talk first, reading pdfs/kindle second view        2. This waiting view needs a redirect. It redirects if true when it pings           1. Pings for student's name = 'null' which is 'false'           2. Pings for student's name = 'not null' which is 'true'   9. Log-out/Auto log-out      + Allison, Cesar        1. This user story arose due to the necessity for having users exit the system, so that the database can be up to date, and admins can only pair users |

|  |
| --- |
| 4/24/2016: 360 minutes |
| Full Team |
| Started at 6  Meeting ended at 12 AM    Initial start of meeting:  John is working on addressing Cycle 2 grade,  Allison and Cesar are working on development,  Mark is working on his time sheet,  Amanda and I are working on User Documentation and the Cycle Report    Cesar and Allison main focus was on pair programming on user stories.  Amanda, Mark, and John focused on the Cycle 3 Report, User Documentation      We updated the priority list based off of user stories:    **Assignments/Progress:**   * **User Story Priority (Numerical Ranking by priority, and using MoSCoW)**  |  |  |  | | --- | --- | --- | | **M** | **4, 17** | on admin view(15) |  |  |  |  | | --- | --- | --- | | **M** | **13** | the Team Leader view needs to be made |  |  |  |  | | --- | --- | --- | | **S** | **14** | does not seem feasible |  |  |  |  | | --- | --- | --- | | **W** | **6** | definitely not feasible. |  * **Research notes asked by Chapman**   1. Alternatives to talki.io and teamviewer   2. Purpose: pass on research and notes to hasten the next group of developers acclimation and keep them from doing work that has been done before.      * **Testing**   1. Amanda      + 4, 13, 14, 15, 16, 17      * **Current User Stories:**   1. ~~Student Log In~~      + ~~Mark~~        1. **~~Completed~~**   2. ~~Student View Notifications~~      + ~~John~~        1. **~~Completed~~**   3. ~~Student Auto Redirect to Lesson~~      + ~~Mark, Allison~~        1. **~~Completed~~**   4. Student/Tutor start video chat service      + Mark, Cesar, Allison        1. Simply link to talky.io        2. Speak with ProjectHSF regarding which talky.io sites they use        3. If math day,  Student goes to math worksheet/khan first, talky.io is second view        4. If Reading day, Student goes only to talky.io        5. Use sockets to push notification from database to student and tutor           1. Timeout method is not at all efficient or desirable, but it works.   5. ~~Student view math worksheets~~      + ~~Cesar, Allison~~        1. **~~Completed~~**   6. Student resume reading from last page      + Team        1. Can research, at the very least, for the next team.   7. ~~Student preview lesson for the day~~   8. Student/tutor screen share control      + John, Amanda        1. First research ways to implement        2. Currently looking into opening executable from javascript function        3. Executable can be batch, python, etc.; needs to use command line to:           1. Log user into teamviewer   Desktop application dependent   * + - * 1. Use Chrome Remote Desktop   Browser dependent  Test it!   * 1. ~~Tutor view daily schedule (student and lesson)~~      + ~~Amanda, Mark, John~~        1. ~~Tutor can see students and their corresponding lessons~~   2. ~~Secured log in~~      + ~~Allison~~        1. **~~Completed~~**   3. ~~Admin view student statistics~~   4. ~~Admin view student/tutor pairing history~~   5. Team Leader plan student lesson      + Allison, Cesar        1. Diagram B1, recommended lesson for the day for the student        2. Date of lesson, Remote tutor day?, worksheets or links to recommendation        3. Team leader will only be able to view his/her students        4. <http://tutorialzine.com/2014/09/cute-file-browser-jquery-ajax-php/>   6. Auto login into external websites      + John, Amanda        1. Student View (User story 3)uses this        2. Log in into KhanAcademy and Kindle using credentials           1. Each student has their own gmail account to get into KhanAcademy           2. Master account for Kindle        3. Different Days           1. Math day opens on a students computer           2. Reading day opens on tutors computer   7. ~~Admin pair student to tutor~~      + ~~Allison, Cesar~~      + ~~THIS WEEK~~        1. Diagram A1   8. Tutor Waiting View      + Amanda, Mark        1. Diagram C1           1. Have it work with the Database           2. <http://stackoverflow.com/questions/31313849/force-logout-on-web-application>           3. Php modify table when logged in           4. Talk first, reading pdfs/kindle second view        2. This waiting view needs a redirect. It redirects if true when it pings           1. Pings for student's name = 'null' which is 'false'           2. Pings for student's name = 'not null' which is 'true'   9. Log-out/Auto log-out      + Allison, Cesar        1. This user story arose due to the necessity for having users exit the system, so that the database can be up to date, and admins can only pair users |

|  |
| --- |
| 4/25/2016: 360 Minutes |
| Full Team |
| FINAL MEETING, Finalize deliverables |

### A.2.2 TA Meetings (Friday’s at 12:00)

|  |
| --- |
| 4/8/2016: 15 min |
| Full Team |
| Austin Wants to see:  -Process, communication  -Include storyboard stuff in esgin  -Godaddy not a user story  -Hashtable key-value for room and tutor  -Teamviewer & command line |

|  |
| --- |
| 4/15/2016: 15, then 200 minutes |
| Full Team |
| Austin talked with us about:  -We are finishing up on the admin view(15)  -Need to get clarification from our sponsor about  -Who connects the tutor and the student, the admin, or do the student and tutor simply go straight to their specified talky.io room and any relevant webpages or resources also gets load.  -Is it okay if we place a program on the desktop for teamviewer. This would make opening teamviewer tons more easier to do, we would think.  Austin, back in his office after Team meeting with  Dr.Dorsey  -Troubleshooting our issues with WAMP -> decided to move to x10hosting  -Other viable solutions for database development uses  -Converting to Node.js  -Using LAMP on a virtual machine on local linux virtualbox  -Uploading to GoDaddy, but that would require Sponsor to upgrade, and we potentially foresee that upgrading Godaddy and such would take more time than we have left in the semester.  -Austin has helped us move our files through ftp(fileZilla) onto x10hosting.  -He has helped us manage our views, as to get us closer to a point where we go to our subdomain on x10hosting, it will bring us to our login page.  -He has also helped us with phpMyAdmin for x10hosting  -Troubleshooting phpMyAdmin on Allison's local MySQL database. Objective is to get it up so we can export it and then import it into x10hosting's phpMyAdmin.  -Exported it from local, imported into phpMyAdmin on x10hosting.com |

|  |
| --- |
| 4/22/2016: 15 minutes |
| Full Team |
| Friday:  Met with Austin, said for John to try out iMacros to at least have something for US#14, John did more research and Amanda and John tried implementing iMacros and sought to see if it could be incorporated into our web component.    We talked about x10hosting being slow, and that being negligible for the component since it will not be as slow when properly hosted and it worked fine when being on the localhost. |

### A.2.3 Sponsor Meetings

|  |
| --- |
| 4/8/2016: 120 minutes |
| Full Team |
| Dr. Dorsey  -We talked about timeout, push student to wherever they need to go if they do not click OK on certain page  -Cannot embed things, so we must use two windows  -Drag and drop for student tutor match  -Chrome Desktop thing  -Admin vs. Team Leader Login    -Remote tutor controls reading while student controls  -Generate list of all students  -roster of students, roster of students  -remote based on log in  -student team leader checks in student  -dropdown is okay  -must repopulate – AJAX?  -Teamviewer – only on math days  -admin takes over screen  -not as high a priority  -Child vs student  -Tutor vs Remote Tutor  -Team Leader – assign recommendation + behaviour  -Administrator – remote matching + screen control  Admin subs for team leader    Name Scheme  Team leader# 30  Admin # 5  Child first+last name or something  Tutor #    Recommendations -> content recommendations  Math recommendations from  Khan Academy Page  Straight to page -> Important!!!  Kindle or PDF or lectio |

|  |
| --- |
| 4/15/2016: 90 Minutes |
| Full Team |
| PHSFAdmins Talked with:  -Originally, they wanted Child needs a waiting view just like the tutor waiting view until the Admin connects them  -If the child is not setup with a remote tutor for that day, they will be taken to a recommendation page  -We are settled on child non waiting screen.  -Talky.io room is tied to tutor – key,value    -So we decided that if the child is not yet paired with a tutor yet, they will log in as if it was a solo day.  -As soon as they say get paired, they will be redirected to the resources and talky.io room that is shared with their remote tutor  -To add to this, child's progress in solo will not transfer if they get redirected to a remote tutoring session.    Teamviewer(8)/screen control should be put off for later when needed for math days, with Khan academy. Another use of screen control would be for troubleshooting from the remote tutor's end over the child's computer.  Dr. Dorsey and the fellows explicitly said it was not of the utmost importance. Screen sharing is by far way more important than screen controlling. |

|  |
| --- |
| 4/22/2016: 15 Minutes |
| Full Team |
| Cancelled our phone conference meeting with Dr.Dorsey. Instead, we emailed him our demo and also had a few phone conversations with him. We rescheduled our meeting with him to be the final day of school, being 2/29/2016. |

## A.3 Size Estimation Documentation

XXX

## A.4 Problem Reports / Change Requests

XXXX

A.5 Correspondence

Correspondence is located in a separate file, " Senior Design Cycle 3 Report Correspondence." It is on CD.

Included are all types of our Team's correspondence.   
Email, chat logs, message boards, etc., between:

* The team and the customer.
* The team and the instructor(s)/manager(s).
* Individual team members.

## A.6 Source Code

Full source code is on CD.

It includes:

* User Documentation
* Sponsor provided diagrams and notes
* All source code
* Database create scripts, stored procedures, etc.
* Administration info (IP addresses, server/machine names, user names, passwords, gmail lists, dropbox or sharepoint accounts, etc.)
* Version information (e.g. README.MD)
  + Version Description is the “README” for the delivered product.
  + Version Description contains the following:
    - Version number
    - Description of the application
    - Key features
    - Known bugs/issues