Project Horseshoe Farm   
 Architectural Spike 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

March 7, 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.................................................................................. 5

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

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

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

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

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

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

5 Design Documentation.......................................................Allison Macdonald......................................................................... 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.

Students should be able to log into a website, see notifications, and navigate to tutoring materials easily. Notifications will include progress via horseshoes and what the students will be working on and, if they will doing remote tutoring, who the tutor will be. Based on the day, either reading or math materials will be presented .The math materials will include a khan academy based on recommendations from tutors and pdf worksheets. The reading materials will include the specific kindle book they will be reading.

Admins will be able to login and presented with a page that will include a table of the students and a table of the tutors. They should be able to click on a student’s name to pull up information on that student and be able to do the same for the tutors. Admins should be able to match the students and the tutors in these tables.

Tutors will be able to login and be presented with a page that will include which student the tutor will be tutoring and when the tutoring session will start. The will have an option to click on the students name in order to see the information on that student. When it is time for the session to start, the page will include a button to join the session.

# 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 tutors. Despite this, the current system could still be improved, hence our group was commissioned to build a standalone website that could connect students with their tutors, supplement lessons, and assist organizers in matching kids with the 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 is no previous work which we can build on. So far we have repeatedly conversed with representatives from Horseshoe Farm to get a design hammered out and have done some development of a prototype. Much of our time so far has been spent in researching options online to allow us to best meet the specifications given to us by Horseshoe Farm.

## 3.2 Intent This Cycle

This cycle our group intends to complete a prototype of the website in which a student, tutor or administrator can log into their account. The ability to log in will likely be the only feature implemented in this cycle, but doing so will pave the way for implementation of additional features such as notification and PDF support.

## 3.3 Future Work

In following cycles, we aim to implement PDF support to allow students to view reading and math related material. In addition, we want to develop the student interface such that they can view notifications and use links to sites such as Khan Academy. In addition, the implementation of some admin controls (such as being able to see individual student's progress) is a feature we would like to implement this semester. Other features such as connecting students and tutors via video will most likely be left for future groups to develop.

# 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 Student Log In

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

Description: The student login box is placed in the middle of the screen where the student can login to the system by typing in a requested username and password through keyboard. If the student types in the right username and password, the student will be able to access his/her account. If the student 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: Collaborative Development

##### 4.1.2 Student View Notifications

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

Description: The page will greet the student 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: 4

Actual This Cycle: 4

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Collaborative Development

##### 4.1.3 Student Auto Redirect to Lesson

Summary: Students 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 student 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 student to the lesson on his or her schedule. It will also connect the student to the specified tutor for the day.

Hours:

Total Planned: 10

Planned This Cycle: 5

Total Actual: 1

Actual This Cycle: 1

Coder: Allison Macdonald

Tester: Team

Reviewer: Team

Status: Collaborative Development

4.1.4 Student/Tutor start video chat service

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

Details: The student and tutor will meet on a video chat service such as "Talky.io" with "TeamViewer". During the duration of their onscreen meeting, the tutor will teach the student the required lessons for that day, while the student follows the tutor and is allowed to take notes or ask questions onscreen. After the tutoring session is over, or if the student or 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: Collaborative Development

##### 4.1.5 Student View Math Worksheets

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

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

Hours:

Total Planned Hours: 10

Planned This Cycle: 10

Actual Hours This Cycle: 0

Total Hours This Cycle: 0

Coder: Cesar Sanchez

Tester: Amanda Bailey

Reviewer: Team

Status: Collaborative Effort

##### 4.1.6 Student Resume Reading From Last Page

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

Details: When a student 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 Student preview lesson for the day

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

Details: After the student 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: Unstarted

##### 4.1.8 Student/Tutor Screen Share Tool

Summary: The system will automatically connect the student to a tutor once ready for the session.

Details: While connecting to Talky.io, the website will also drive a connection to the tutor via TeamViewer. This tool will allow the 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: 0

Total Hours This Cycle: 0

Coder: TBD

Tester: TBD

Reviewer: Team

Status: Unstarted

##### 4.1.9 Tutor view daily schedule (student and lesson)

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

Details: When the tutor logs into their account, he/she will be able to view a list of appointments that students have made. The 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 tutor with. This schedule list will allow the tutor to plan accordingly, so he/she will be able to help every single student with their problems.

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: Unstarted

##### 4.1.10 Secured Log In

Summary: Logging into either administrator or 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 administrator or a 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: 1

Total Hours This Cycle: 1

Coder: Allison Macdonald

Tester: Amanda Bailey

Reviewer: Team

Status: Collaborative

##### 4.1.11 Admin View Student Statistics

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

Details: After logging in, an admin will be allowed to click on the name of a student. This will cause a student information page to open which includes the student's most frequently matched 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: Unstarted

4.1.12 Admin View student/tutor pairing history

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 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 student is paired with a particular tutor. This way, the sponsors will be able to make arrangements to have a student paired with his/her favorite 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: Unstarted

4.1.13 Admin plan student lesson

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

Details: After logging in, the admin will be allowed to make daily lesson plans for the students. This page will be directed into the account of every single student and 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: Unstarted

# 5 Design Documentation

The overall intent behind designing this project was simplicity. Horseshoe Farm plans to give this system to students 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.

## 5.1 Architecture

In order to keep the user interface as simple as possible, it was determined that the website will have only one function per page. For example, there is a log in page which has nothing on the screen except for a username box and a submit button. Similarly, any subsequent pages loaded by student accounts have only a simple message in large font and a single button. There is as much automation in moving students to the correct lessons as possible.

In talking with Horseshoe Farm, it became clear that the site would need to be structured into student accounts and admin accounts. Further planning revealed that for the site to encompass all requirements, tutor accounts would be needed as well. Each account has its own functionality and will experience the site as laid out in the story boards below (Figures 1-3).

### 5.1.1 Student 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 student 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 under the text box and the user will not be let into the site. Upon the submission of a valid user name, the student will be logged into the site. A screen will welcome the student by first name while displaying a summary of which tasks were accomplished during the last session, for example, number of pages read, number of videos watched, number of worksheets completed, and number of horseshoes earned, as well as the total number of horseshoes that the student has earned up to date. This running total will be kept at the top right of the screen for easy viewing. After viewing the summary, the student 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. The next display will show a quick overview of the student’s schedule for the day including the subject that will be studied and the matched tutor for the day. Once again, the student will click “Okay!” to initialize the Talky.io and TeamViewer connections as well as a redirect to the lesson page specified by the schedule.

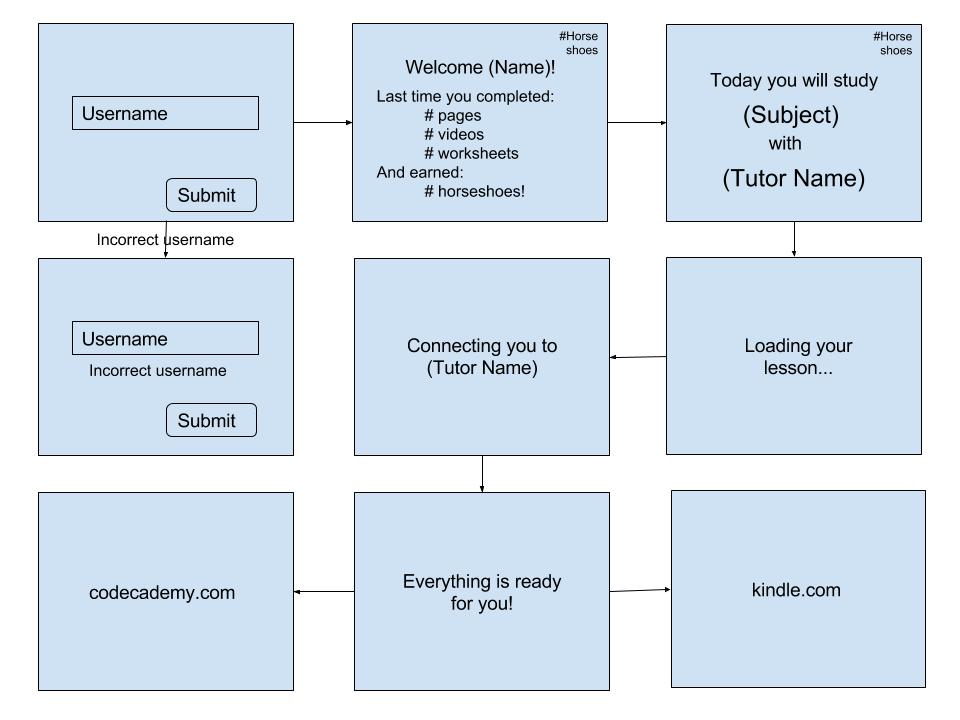


Figure 1: Student Story Board

### 5.1.2 Administrator Use Case

When the website is being utilized by an administrator, typically the on-site tutor, the same display screen that the students are brought to will load. Once again, if the administrator enters an invalid username, an error message will appear. However, this time upon entering a valid name, the website will not immediately log the user in. Instead, a hidden text field will appear prompting the user for a password. An invalid password will result in an error message while a valid password will bring the administrator to a screen welcoming him or her by name. This welcome screen will be populated with two lists on either side of the screen, one for students, and one for teachers. Clicking on any one of the provided names will bring the administrator to an information page regarding the student or tutor of interest. For tutors, an information page will display stats such as most frequently matched students as well as the tutor’s upcoming schedule. A student info page will display the student’s most frequently matched tutor as well as schedule and a summary of statistics such as total horseshoes, completed tasks, etc. From this page, an administrator will be able to make changes to the student’s schedule.

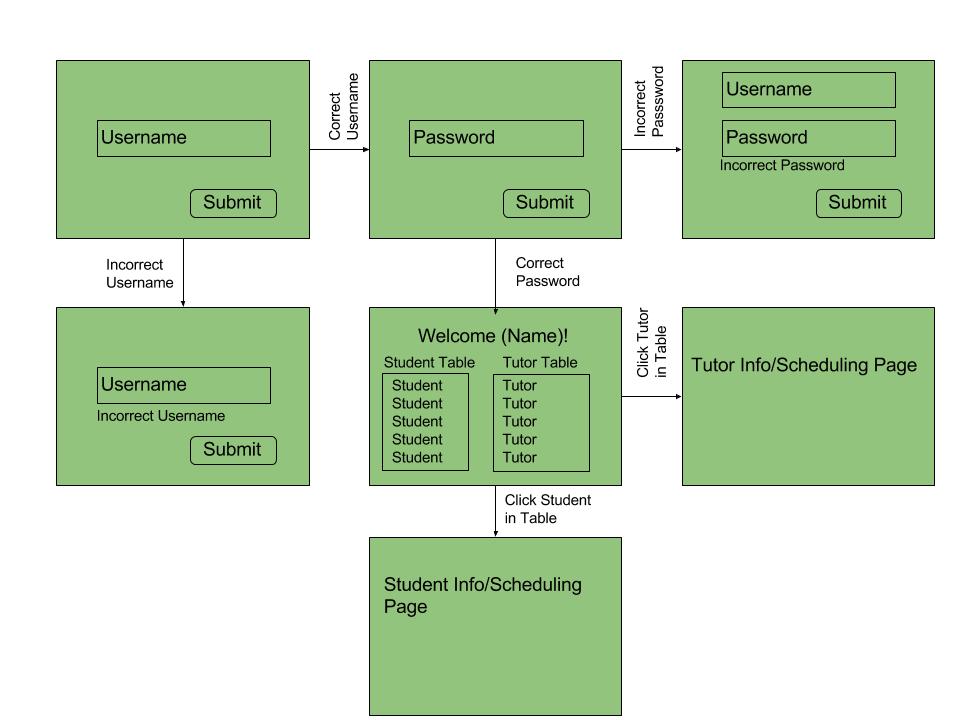


Figure 2: Admin Story Board

### 5.1.3 Tutor Use Case

Tutors will have a log in situation similar to that of an administrator. The 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 tutors to a welcome screen which greets the tutor by name and displays a countdown until the next scheduled session with a student. Upon clicking on the name of their next student, a tutor will be able to view the student’s info page without the schedule editing permissions of an admin. If the tutor arrives at the welcome page at some point after the countdown, the website will display a “Join!” button in place of the timer. Clicking this button will connect the tutor to his or her scheduled student via Talky.io and TeamViewer.

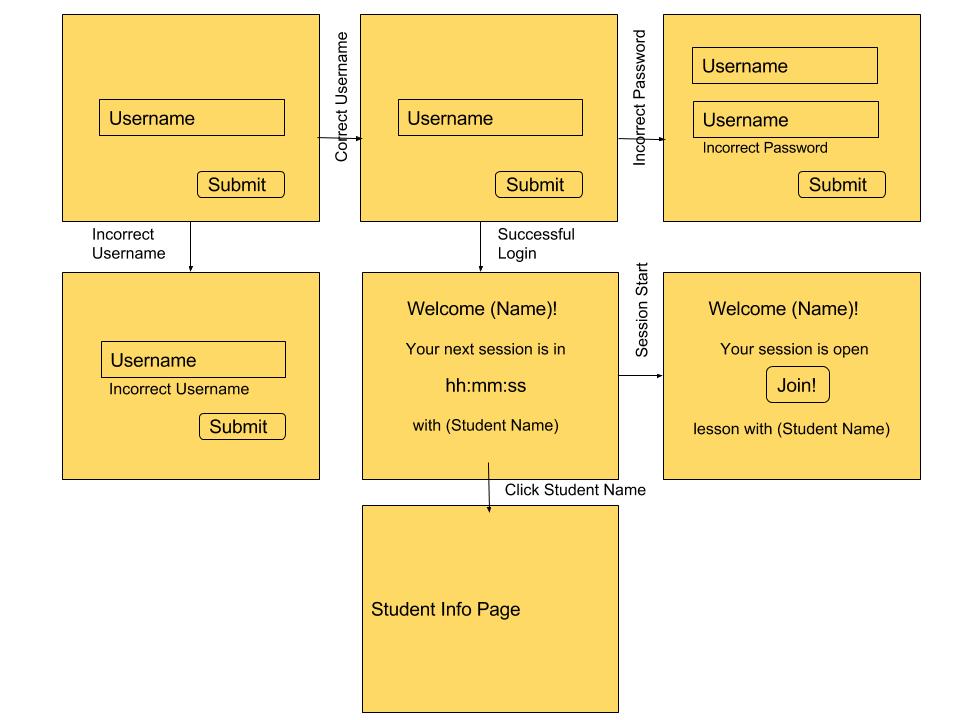


Figure 3: Tutor Story Board

## 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 login.html file only handles logging in while the welcome.html only handles welcoming users. Due to the varying user interfaces experienced by each user, each user type has its own folder containing HTML (ie. Student/welcome.html is a different file than Tutor/welcome.html). In order to keep all elements organized, each HTML file corresponds to exactly one JavaScript file which is located in the js folder at the project root. These files use jQuery to listen for events that occur on HTML elements in order to validate, redirect, and transition. 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 files are maintained in their own folder as well but are not yet implemented.

## 5.3 Interfaces

The main external components in this project are PHP, MySQL, and Apache. The three work together to create a local development environment while interfacing with front end code. While for the architectural spike each member of the team participated in setting up these back end components, there has not yet been any interaction between them and the front end, so no performance metrics have been taken.

## 5.4 Assumptions and Dependencies

After experiencing frequent failures with setting PHP on localhost, WAMP was installed on all Windows machines while MAMP was installed on Macbooks. These come with prepackaged versions of PHP, MySQL, and Apache meaning that the versions used by Windows developers and Apple developers may differ. The team is making the assumption that there will not be significant nuances between the versions of each tool provided across different platforms. A secondary and perhaps more serious 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

During the architectural spike, it is a goal to develop some sort of prototype that can be approved by the sponsor. It was the intention of the team to have a prototype that communicated with a database, but difficulties with setting up PHP took up a significant portion of time. The current prototype is still functional for the purposes of clarifying if our vision matches up with the vision of the sponsors, but it only functions on hard-coded values within JavaScript.

# 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.
* 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
* Prototyping of component
* End of cycle assessment of delivery by Sponsor
* Acquisition of changes in requirements from sponsor
* Planning of next cycle's features
* User Experience and User Story additions or changes

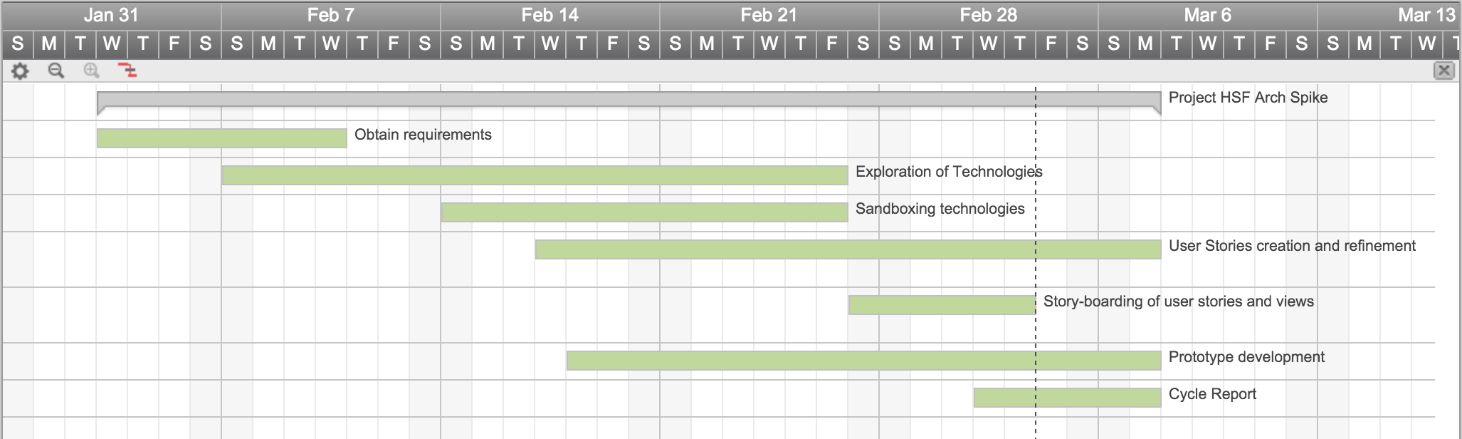
User Stories:

* Student Log In - Mark
* Student View Notifications - John
* Student View Worksheets - Cesar
* Secured Log In - Allison
* Testing – Amanda

## 6.2 Task Assignments

|  |  |
| --- | --- |
| Team Member | Team Member Assignments and Tasks Assigned This Cycle |
| Amanda Bailey | **User Stories**: Testing  **Cycle Repor**t: Executive Summary Documentation  **Team Role**: Correspondence Lead, Team Management, UX Design, Lead Tester |
| John Carroll | **User Stories**: Student View Notifications  **Cycle Report**: Management Planning, Appendices  **Team Role**: Team Project Manager, Correspondence, Technology Researcher, UX Design |
| Lanxin (Mark) Ma | **User Stories**: Student Log In  **Cycle Report**: User Stories Documentation, Lessons Learned  **Team Role**: Technology Research Lead, UX Design |
| Allison Macdonald | **User Stories:** Secured Log In  **Cycle Report:** Design Documentation, Lessons Learned, References  **Team Role:** Team Project Manager Lead, Meeting Minutes Documentation Lead, Web Development Lead, Correspondence Lead, Technology Research Lead, UX Design |
| Cesar Sanchez | **User Stories:** Student View Worksheets  **Cycle Report:** Risk Mitigation, Project Introduction Documentation  **Team Role:** Technology Research, Tester, UX Design |

## 6.3 Development Schedule

Gantt chart:  


## 6.4 Planned Code / Feature Freeze

March 7th, 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 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

Because this cycle acted as our architectural spike, we were not able to design a formalized test plan, however we did do some informal testing in order to make sure the prototype worked properly. The front end was tested by hard coding values into the pages, until it was possible to use PHP. This was done in order to make sure the pages were linking to the next appropriate page. The page was also tested in different browsers in order to make sure they viewed consistently.

## 8.2 Test Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | √ |
| 1 | Type in the name “Maria” | The student logs in and is presented with notifications. |  | √ |
| 2 | Click the “Okay” button that is below the notifications. | The student is presented with the lesson plan. |  | √ |

# 9 Lessons Learned

Lessons learned is an invaluable section of your report for teams that come after you. Anything you tried that didn’t work, any technologies or solutions you considered or attempted and then abandoned, any problems with parts, components, vendors, software APIs, etc., should be documented here.

We are not interested in lessons such as “we learned how to better communicate as a team” or “we learned how to set up a database using Microsoft SQL Server.” That is pointless – you’re expected to develop team skills in this course and you’re expected to develop or improve your skills with new tools.

Instead, this section, which is arguably one of the most important in your report, should serve as a roadmap for future work and help future engineers avoid some of the problems or roadblocks you encountered.

Lessons learned:

In this cycle, we have learned four 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 on 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 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. Finally, we have learned about the importance of branching before making any changes to code pulled from GitHub, in order to avoid losing or modifying any important code.

# References

# A Appendix A Supporting Documents

## A.1 Status Reports

XXXXXX

## A.2 Meeting Minutes

Total of 6 Full Team Meetings for 720minutes (12.67 hours)  
Total of 2 Partial Team Meetings for 540 minutes(9 hours)

=====================================================================2/5/2016: (120 minutes)

[Amanda, John, Mark, Allison, Cesar]

It was on this Friday when we met in Dr.Chapman's office to host a video conference with the fellows of Project Horseshoe Farm. This was our first meeting with the sponsors where we received insight into who they are, what they do, and where we come into their operations.

2/10/2016 (200 minutes)

[Amanda, John, Mark, Allison, Cesar]

Meeting Synopsis:

The meeting's intent concerned exploring technologies to apply to our project's architectural spike.

Meeting Notes:

* *How are we paying for this Amazon thing?*
* *Do we want to work on:* 
  + *\*\*the start-up framework\*\**
  + *look for a video chat service*
  + *work on reading app*
* *Need to get a cost estimate for hosting via traffic.* 
  + *30-50 people 5 days a week*
  + *Max 50 users estimated.* 
    - [*https://www.godaddy.com/help/how-many-visitors-can-view-my-site-at-once-3206*](https://www.godaddy.com/help/how-many-visitors-can-view-my-site-at-once-3206)
    - *Economy tier should be fine.*
  + *Microsoft vs Linux Servers* 
    - [*https://www.godaddy.com/hosting/web-hosting*](https://www.godaddy.com/hosting/web-hosting)
  + *Shared price is comparable but… shared.* 
    - [*https://www.dreamhost.com/hosting/*](https://www.dreamhost.com/hosting/)
* *Do we have any ideas for signing in or should we learn more about their process before speculating?* 
  + *Student or tutor? Different levels of accounts.*
  + *Username and password*
  + *Names stored in database*
  + *MySQL issue for Mark in the past*

2/17/2016 (200 minutes)

[Amanda, John, Mark, Allison, Cesar]

Meeting Synopsis:

Met to look into technologies to create our project's baseline. Devised beginnings of user stories.

Meeting Notes:

* *Tools:* 
  + *PHP & MySQL*
  + *Node.js*
  + *LINQ to SQL w/ C#*
  + *Jade.js vs HTML*
  + *It will be best to go with PHP&MySQL because some of our members have experience with this. This set up is also commonly used for database to web applications and we will likely be able to find plenty of support online.*
* *Generate user stories and prioritize functionality for prototype.* 
  + *The existing stories are okay for the prototype, but we will develop more as we delve more deeply into the project.*
* *Anyone good at UXD?* 
  + *Mark and Amanda will be the primary front end designers.*
  + *Discussed design for handling admin users* 
    - *Conditional redirect or conditional hidden password field*
* *Can I move this stuff over to Visual Studio or should we stay away from a .NET project?* 
  + *Using .NET will allow us to implement Partial Views, but has little else to offer. It would not support PHP and we would have to use LINQ and SQL Server.*
* *Assign Tasks for the Week* 
  + *Log In - Allison*
  + *Splash Screen Design - Mark*
  + *Look into Video Chat Service - Amanda*
  + *Test PDF Integration - Cesar*
  + *Admin Functionality or View Notifications - John*
* *We need more details on* 
  + *How to obtain a horseshoe*
  + *Are the students doing anything once they log in, or will everything (including notifications) be automated based on what the tutor has assigned them?*
  + *Can you elaborate on the general process? Do students all have different individual assignments?*
  + *Would this table design be an accurate way to express what is incorporated into a lesson?*

2/24/2016: (120 minutes)

[Amanda, John, Mark, Allison, Cesar]

Meeting Synopsis:

Met to gather a consensus for technologies platforms used, to plan out the coming objectives as well.

Meeting Notes:

*Technologies used:*

*PHP/MySQL*

*JQueryCSS/HTML5 - Bootstrap*

*Things we can do:*

*Story board*

*User stories*

*Start on Status Report*

*Start on Cycle Document*

*Split up User Stories*

*Set up PHP*

Stuff:

DBUnit for testing

2/28/2016: (120 minutes)

[Amanda, John, Mark, Allison, Cesar]

Meeting Synopsis:

Met to assign user stories to be focused by team members.

Meeting Notes:

* Let's split up user stories for this cycle!
  + Student Log In - Mark
  + Student View Notifications - John
  + Student View Worksheets - Cesar
  + Secured Log In - Allison
  + Testing - Amanda
* Video sharing
* Skype and talky can share but can't let the other user take control
* Google remote desktop requires an access code and to be preinstalled on the computer

3/2/2016: (120 minutes)

[Amanda, John, Mark, Allison, Cesar]

Meeting Synopsis:

Met to assign Cycle 1 report's sections to team members to complete.

Meeting Notes:

***Table of Contents ii***

***2******Executive Summary (System Metaphor) 1 -- Amanda***

***3******Project Introduction 2 -- Cesar***

*3.1 Previous Development 2*

*3.2 Intent This Cycle 2*

*3.3 Future Work 2*

***4******Requirements / User Stories 3 -- Mark***

*4.1 User Stories 3*

*4.2 User Stories 3*

*4.2.1 Off-Screen Ghost Markers 3*

***5******Design Documentation 5 -- Allison***

***6******Management Plan 6 -- John***

*6.1 Task Assignments 6*

*6.2 Development Schedule 6*

*6.3 Planned Code / Feature Freeze 6*

***7******Risk Mitigation 7 -- Cesar***

***8******Test Plan and Test Procedures 8 -- Amanda***

*8.1 Test Plan 8*

*8.2 Test Procedures 9*

*8.2.1 Procedure 1: Sample Test Procedure 9*

***9******Lessons Learned 10 -- Mark***

***References 11 -- Allison***

***Appendix A 12 -- John***

*A.1 Status Reports 12*

*A.2 Meeting Minutes 12*

*A.3 Size Estimation Documentation 12*

*A.4 Problem Reports / Change Requests 12*

*A.5 Correspondence 12*

*A.6 Source Code 12*

*Roadblock - Setting up*

*AMANDA H*

[*http://stackoverflow.com/questions/23665064/project-links-do-not-work-on-wamp-server*](http://stackoverflow.com/questions/23665064/project-links-do-not-work-on-wamp-server)

3/4/2015: (180 minutes)

[Allison, Cesar]

Meeting Synopsis:

Finalize Cycle 1 report documents.

3/6/2015: (360 minutes)

[Cesar, John, Amanda]

Meeting Synopsis:

Worked on finalizing the status report for Cycle 1's final week, along with Cycle 1's report, as well as the starting the PowerPoint presentation for Cycle 1.

## 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 1 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:

* 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.TXT)
  + A Version Description is the “README” for the delivered product.
  + A Version Description should contain the following:
    - Version number
    - Description of the application
    - Key features
    - Known bugs/issues