Project Horseshoe Farm   
 Cycle 1 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 5, 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........................................................................... 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. There will be a student, tutor, and a admin/team leader account. Students 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 students, and assign students to tutors. Team leaders should be able to log in, plan lessons for students, and award horseshoes. Tutors will be able to log in, wait to be assigned to a student 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 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 was no previous work for our group to build on. Last cycle our group developed a prototype of the website that had basic login functionality for students. While not having much functionality, this prototype acted as the root for all other development done this cycle. A great deal of time last cycle was also spent on clarification of requirements and expectations provided by the sponsors.

## 3.2 Intent This Cycle

This cycle ours group expanded the login functionality to allow both students and admins to login. Additionally, the functionality of the login page was polished such that the login checks credentials dynamically and includes standard features such as character masking for password entry. While not yet accessible via the login page, the tutor's page has been created and several of its features have also been implemented. The database has been created and some information has been entered into it to allow demonstration of functionality. A great deal of time this cycle has also been devoted to research into software such as TeamViewer and Talky.io. Additionally, a trip was conducted to Horseshoe Farm's base of operations to gain a firsthand perspective of the tutoring sessions.

## 3.3 Future Work

In the final cycle, we have a few major features to implement. First and foremost, we aim to allow students and tutors to connect for video sessions, ideally through talky.io. Of equal importance is moving all database and other relevant data onto GoDaddy servers so there is an actual web site up and running, though that will also entail discussing pricing options with the sponsors. Additionally, we aim to augment the student pdf viewer so they can resume reading from the last page they were on. Finally, by the end of next cycle admins should be able to designate an educational plan for each student.

# 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 (1)

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: Completed

##### 4.1.2 Student View Notifications (2)

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: Completed

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

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: Completed

4.1.4 Student/Tutor start video chat service (4)

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: Collaboration

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

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: Completed

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

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 (7)

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: Discarded

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

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: Collaboration

##### 4.1.9 Tutor view daily schedule (student and lesson) (9)

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: Collaboration

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

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: Completed

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

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: 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 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: Discarded

4.1.13 Admin plan student lesson (13)

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 recommendations as possible.

Initially, it was thought that the sponsors wanted the system to be split into student, admin, and tutor accounts. Upon further clarification, it is now understood that the account types are as follows: administrator, team leader, remote tutor, and student.

### 5.1.1 Administrator Use Case

Administrators have a high level view of the entire system. While logging into the website, administrators will be brought to the same log in screen used by students. 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 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 their corresponding remote tutors. The student list will indicate which students are currently online while the tutor list will indicate the match percentage of that student-tutor pairing. In future iterations of this project, the student-tutor pairings will be made automatically based on match percentages rather than recommendations of the administrator. Clicking on any one of the provided student names will bring the administrator to an information page regarding the student of interest. A student info page will display matching history and behavior tracking. From this page, an administrator may go back to the roster or move on to the next student.

### 5.1.2 Team Leader Use Case

Several team leaders are always on site during tutoring sessions. These leaders are each assigned a group of students to preside over and are the individuals responsible for making recommendations of study for each student. 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 student names. The team leader may then use this page to check students in or out and make recommendations. Clicking on a student name will bring up more details about that student and allow the team leader to leave comments about the student behavior for the day. Due to time restrictions, only the recommendations feature will be available to team leaders by the end of this semester.

### 5.1.3 Remote 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 the tutor to a waiting screen until he or she can be paired with a student. Upon pairing with a student, Talky.io will appear in the top left corner of the screen while the rest of the page will display the student's screen.

### 5.1.4 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 above 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 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 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. Several scenarios may occur here: math without tutor, reading without tutor, math with tutor, or reading with tutor. If it is a math without 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 tutor day will also provide a list of recommendations for the student to choose from. If it is a math with tutor day, the student will be connected to TeamViewer and Khan Academy. On a reading with tutor day, the student will be connected to Talky.io and read.amazon.com; due to licensing issues with some books and the preferences of some students, a portion of student-tutor pairs will read from physical books over Talky.io rather than reading from the Amazon website.

## 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 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 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

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

Goals for this cycle were to get PHP up and running, to communicate with the database for logging in all users, dynamically redirecting a student to a lesson, starting Talky.io during a session, and integrating PDFs. Once PHP was up and running, some modifications had to be made to the structure of the login set up. Where previously there had been data hardcoded in Javascript, an HTML form now has to submit to a PHP file which communicates with the database to obtain data. Some functions that were initially intended to be accomplished with Javascript, such as redirecting are now handled by PHP.

# 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
* 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

6.1.1 User Stories:

1. Unsecured Student Log In
   1. Mark
      1. List of ProjectHSF's student's login ID's to use as the Usernames
      2. Get WAMP up so we no longer have to hard code the data structure
2. Student View Notifications
   1. John
      1. ~~# of Horseshoes earned, also in the top right corner at all times~~
      2. Previous Lesson
         1. How far the student got
3. Student Auto Redirect to Lesson
   1. Mark, Allison
      1. Dynamic, so pull from WAMP database to determine today's lesson
      2. Load lesson
4. Student/Tutor start video chat service
   1. Mark, Cesar
      1. Simply link to talky.io
      2. Speak with ProjectHSF regarding which talky.io sites they use
5. Student view math worksheets
   1. Cesar
      1. PDF.js - for pdf integration
6. Student resume reading from last page
   1. NOTE: Helpful but not quite necessary for it to run
   2. Save for Cycle 2
7. ~~Student preview lesson for the day~~
   1. ~~John~~
      1. ~~Similar to 2 and 4~~
8. Student/tutor screen share control
   1. NOTE: Can't even.
   2. BUT: IMPORTANT
   3. Amanda, John
      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
9. Tutor view daily schedule (student and lesson)
   1. John, Amanda
      1. Tutor can see students and their corresponding lessons
10. Secured log in
    1. Allison
       1. Requires a password
       2. Need to set up MySQL again
11. ~~Admin view student statistics~~
    1. ~~Saving for Cycle 2~~
12. ~~Admin view student/tutor pairing history~~
    1. ~~Saving for Cycle 2~~
13. Admin plan student lesson
    1. Allison

### 6.1.2 Architecture

* + Database Design – MySQL
    - Allison, John
      * Using Draw.io, designed the database

UML Diagram, ER Model

* + - * Created database using phpMyAdmin which came with WAMP Server
  + Development feasibility on GoDaddy
    - Allison, John
      * Purpose and objective:
        + Have one centralized database to develop on
        + Sponsors would be able to run with our demo.
        + Closer to the end of semester goal
      * Call GoDaddy
      * Get MySQL/PHP DB up and running for use

### 6.1.3 Testing

* + Amanda
* Student login (1)
* Student View Math WorkSheet(3)
* Student Reading with Kindle(3)
* Admin login (10)

## 6.2 Task Assignments

|  |  |
| --- | --- |
| Team Member | Team Member Assignments and Tasks Assigned This Cycle |
| Amanda Bailey | **User Stories**: 8, 9  **Cycle Repor**t: Executive Summary Documentation  **Team Role**: correspondence, team process management, UX Design, testing lead, architecture technology research |
| John Carroll | **User Stories**: 2, 8, 9  **Cycle Report**: Management Planning, Appendices  **Team Role**: team process management, correspondence, architecture technology research, UX Design |
| Lanxin (Mark) Ma | **User Stories**: 1, 3, 4  **Cycle Report**: User Stories Documentation, Lessons Learned  **Team Role**: technology research, UX Design |
| Allison Macdonald | **User Stories:** 3, 5, 10, 13  **Cycle Report:** Design Documentation, 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, 5  **Cycle Report:** Risk Mitigation, Project Introduction Documentation  **Team Role:** architecture technology research, tester, UX Design |

## 6.3 Development Schedule

Gantt chart

## 6.4 Planned Code / Feature Freeze

April 4th, 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. The testing below is for Firefox only. Chrome still needs to be tested. Testing the functionality of the code will include ensuring that the test procedures pass. In this cycle, this will include the student login with a name from the database, viewing a math worksheet, and being redirected to talky.io. In order to ensure the test procedures pass, there are multiple components that we will have to test. These components include the files for PHP, HTML, and JavaScript.

## 8.2 Test Procedures

8.2.1 Student 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. |  | 4/3/2016 |

8.2.2 Student Login: Name in database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid student name from the database into the text box. Click ‘Okay!’ | Should be redirected to a notifications page. |  | 4/3/2016 |

8.2.3 Student Auto Redirect to Lesson & Start Video Chat: With kindle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Enter a valid student name from the database into the text box. Click ‘Okay!’ | Should be redirected to a notifications page. |  | 4/3/2016 |
| 2 | Click ‘Begin Today's Lesson!’ | Should be taken to a page with a pdf of a math worksheet. |  | 4/3/2016 |

8.2.4 Student View Math Worksheets: No remote tutor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Required Actions | Expected Results | Comments | Dates |
| 1 | Sign in as student who will be having a reading day/tutoring session. | View notifications page. |  | 4/3/2016 |
| 2 | Click ‘Begin Today's Lesson!’ | Should be directed to the kindle page and a talky.io chat. |  | 4/3/2016 |

8.2.5 Admin Login: Admin 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. |  | 4/3/2016 |

8.2.6 Admin Login: Amin 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. |  | 4/3/2016 |
| 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. | 4/3/2016 |

8.2.7 Admin Login: Amin 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. |  | 4/3/2016 |
| 2 | Enter an invalid password into the password box and hit enter. | After hitting enter, an error message should appear. | Password should be hidden. | 4/3/2016 |

# 9 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 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. 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

## 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 880 minutes (14.67 hours)  
Total of 3 TA Meetings for 60 minutes (1 hour)

Total of 2 Sponsor Meetings for 240 minutes (4 hours)

Below is a breakdown of the meeting minutes.

### A.2.1 Team Meetings

3/9/2016: (120 min)

[Full team present]

Meeting Synopsis:

Discussed that Dr. Chapman would like our project to be in a working state for ProjectHSF by the end of our Cycle 2, which is around the end of April, 2016.

Through this meeting, we created a "Tasks" tab in our shared OneNote (name may be subject to change). In it, the information below will be stored, updated, and easily locatable for everyone to view. This will help us all remain informed on what components and roles are assigned to whom.

3/20/2016: (100 minutes)

[Full team present]

Meeting Synopsis:

* John took over as document organizer
* Added current tasks board
* Gave up on using swim lanes
* Noted that our process is very Agile
  + We may have a 4 week cycle but we speak with our Sponsors about once every 2 weeks, so it is about 2 sprints per 1 cycle
  + We show the sponsor/client what we have at the end of our sprint, and then see if we met requirements
    - They see if it is what they wanted, they also see if they wish to add features or feedback to component.

3/23/2016: (120 minutes)

[Cesar, Amanda, Allison, John]

Meeting Synopsis: Email Austin regarding PHP issues. All research ways to open teamviewer from a browser. Also formulate questions for PHSF fellows.

4/2/2016: (180 minutes)

[Allison, Cesar]

Meeting Synopsis:

Work on user story: 5

4/3/2016: (360 minutes)

[Full team present]

Meeting Synopsis: Worked on finalizing the status report for Cycle 2's final week, along with Cycle 2's report, as well as the starting the PowerPoint presentation for Cycle 2.

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

3/11/2016: (15 minutes)

[Full team present]

Meeting Synopsis: Met on uncovering what component integrations were deemed possible for the cycle/semester. Also, talked about what to discuss with our sponsor in the next hour over video conference.

3/25/2016: (30 minutes)

[Full teeam present]

Meeting Synopsis: Austin helped Allison with injecting mySQL queries through PHP. We were successful in getting to a point where we could query the mySQL database.

Discussed logging into teamviewer automatically and how it may require a command line/terminal component for that.  Vague, but it's part of our research now.

4/1/2016: (15 minutes)

[Cesar, John, Amanda, Mark]

Meeting Synopsis: Discussed with Austin in regards to our team meeting the demo's expectations.

### A.2.3 Sponsor Meetings

3/11/2016: (120 minutes)

[Cesar, John, Allison, Amanda]

Meeting Synopsis: Video conference with Dr. Dorsey, James and Josh. Gathered remote tutoring details, made plans to gather diagrams as well as user scenario details from the PHSF fellows.

3/25/2016: (120 minutes)

[Full team present]

Meeting Synopsis: James and Dr. Dorsey, we met regarding Diagrams: <https://drive.google.com/folderview?id=0B0uknCFcvncFeXdzdXMxVURBSG8&usp=sharing_eid&ts=56f3e588>

Purchase order for new hardware, PHSF would like to discuss minimum specifications for new computers. Also raised questions regarding the facilities’ network capabilities.

Dr. Dorsey called. We spoke about possibly going out to Greensboro, AL some Tuesday to check out and witness the Project Horseshoe Farm's tutoring process and everything else.

## 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 2 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.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