Final Documentation

Group Name: Phase 4

Project Name: Cybersecurity Training 2.0

Group Members: Robert Hand (Team Leader), Valeria Gamboa, Matthew Sexton, Brandy Jordan

Table Of Contents

[**Executive Summary**](#_wbleqqmdj5jk) **3**

[**Project/System Requirements**](#_cbgqrsz4mnr) **4**

[**Timeline**](#_y8us5n3mcfwg) **5**

[**Design Specs**](#_2jsvnhcns38w) **6**

[Relational Schema](#_b1rsqemsbcw8) 6

[Database Diagram](#_ksk50xc3nfts) 8

[Website structure and navigation](#_qixt4k1fpvy1) 8

[Color scheme](#_odyll3qqa2ld) 10

[**Project evaluation/critique**](#_d8e971mldv1h) **10**

[Potential next steps](#_7cbd1fr9of22) 10

[What did/didn’t go as expected](#_fpeg6vjcby31) 11

[Good/Bad decisions](#_3p2szk3e6prx) 11

[Good](#_edd6mgjs76aq) 11

[Bad](#_6jtuxck7rj2b) 11

[What was learned](#_gngvpwnnmr2q) 11

# Executive Summary

Overall, this project was a mix of highs and lows. In the conception stages of the project, we all set out to improve upon the existing education methods for cyber security training. As a group, we found that the current methods of teaching cyber security concepts (especially in the business sector) left much to be desired. Therefore, we decided to create a web application that would take the user through an enjoyable, non-tedious curriculum that would genuinely improve the user’s knowledge of cyber security subjects.

We believe that we fell short of our goal. However, we did make a quite functional web application, that does teach cyber security subjects with reasonable efficiency. The system runs in a user’s browser, and a user can create an account with the system in order to track their progress. There is a login/logout feature, letting users enter and leave communication with the system whenever they need to. Also, the modules have readings on cyber security subjects that culminate in short quiz games to test their knowledge. Users can then choose to complete a course whenever they feel that they have learned it. At any point, users can view how much of the course they have completed.

Once all of the modules have been completed, the user can take a final exam consisting of 20 questions (that are stored in the database), if they make a 90% or higher, they complete the course! With that completion comes a personalized certificate that they can either print, download, or email to a recipient of their choice.

We originally decided on making a fully fledged game for each of the four modules. Part of the reason we fell short of that was that we were all unfamiliar with the languages and technologies we would be using to make the system. Another cause of difficulty was the juggling of other responsibilities outside of the course itself. In spite of falling short of our original goals, we believe that we have made something at the very least valuable, if not good or great. We also learned quite a bit about making web applications as a bonus.

# Project/System Requirements

(Old, phased out requirements are in parenthesis). **Justification of modifications are in bold text.**

1. The system will have a client-server architecture.
2. The client portion of the system will run in the user’s browser.
3. The system will have a website.
4. The server will contain user information.
5. (The system will educate users in cyber security subjects using games.) **The games method of education was found to be too intensive for the time that we had to work on the project.**
6. (The games will be educational in nature, teaching the user about a particular cyber security subject.) **The games method of education was found to be too intensive for the time that we had to work on the project.**
7. The system will educate users through readings related to the cyber security subjects. **This is part of the education method we chose to replace the games.**
8. The system will test a user’s knowledge of the individual cyber security subjects through quick quiz games. **This is part of the education method we chose to replace the games.**
9. (The server will contain information related to the games.) **During development, it was discovered that it would be better if the game ran in the client.**
10. (The system should have a pre-assessment feature that structures the curriculum based on the user’s performance on the pre-assessment.) **Cut for time.**
11. The system will have a way to verify that the user has indeed learned the subjects contained within the curriculum.
12. The system will have a login feature that allows the system to store information related to particular users.
13. The system will store a user’s progress, allowing them to resume where they left off.
14. (The system should be secure, protecting user privacy.) **This requirement was too general.**
15. The system will encrypt the user’s password for storage.
16. The system will not allow passwords with less than fifteen characters.
17. The system will only allow unique usernames.
18. The system will not allow usernames and passwords to be the same strings.
19. (The system should always respond to user inputs in less than 3 seconds.) **This requirement was too general as well.**

# Timeline

**Justification of modifications are in bold text.**

|  |  |
| --- | --- |
| Week of: | What was accomplished during the week: |
| September 23: | Completed research on software and hardware that will be used, learned how to use Github, requirements document. **Other obligations prevented us from getting started right away.** |
| September 30: | Technology document, initial implementation of login/registration features (with database support). **Curriculum development got pushed back while we focused on coding the website and research.** |
| October 7: | Project Definition presentation, produced a list of cyber security subjects that the system needs to teach, first project report. |
| October 14: | Initial implementation of the layout (look) of the website, second project report. |
| October 21: | Created the sidebar (a chief navigation tool of the system), implemented the certificate display, print, download, and email functions, updated general appearance of website, and third project report. |
| October 28: | Finished certificate features by adding database and file system integration, completed the “view progress” feature, fourth project report. **Unable to complete a vertical slice due to timing problems.** |
| November 4: | Intermediate presentation, curriculum development, fifth project report. |
| November 11: | Added the contents of module two into the system, sixth Project report. **The original estimation of having most functionality complete by this time was too optimistic. The rest of the changes after this reflect that.** |
| November 18: | Implemented the final exam and module one, refactored the system for modularity, general bug fixing, seventh project report. |
| November 25: | Finished vertical slice of system, implemented the quiz game for module two, implemented module four, general refactoring, upgraded database interactions, implementing security features (i.e. username and password validation). |
| December 2: | Creation of config files for the database, implemented module three, implemented the rest of the games, fully implemented the final exam, final project artifacts, individual questionnaire, exit interview. |
| Week of Dec 10: | Final Presentation. |

# Design Specs

The design of this system has three major components: the front-end web pages, the code responsible for personalizing those webpages, and the back-end database. The specifications that are necessary as a result have to do with how the database operates, how the web pages look, and how the user goes through the system.

The questions and answers for the tests are contained within the documentation folder on the github repository.

## Relational Schema

DB name: cyber\_sec2

users

* id INT 11 (primary key, auto increment)
* username VARCHAR 255 (unique)
* email VARCHAR 100
* password VARCHAR 255

course progress

* id INT 11 (primary key, foreign key)
* module1 BOOLEAN (not null, default false)
* module2 BOOLEAN (not null, default false)
* module3 BOOLEAN (not null, default false)
* module4 BOOLEAN (not null, default false)
* final BOOLEAN (not null, default false)
* FK\_USER\_PROGRESS (On Delete/Update: Restrict) column: id
  + Database: registration
  + Table: users
  + Column: id

certificates

* id INT 11 (primary key, unique, foreign key)
* filename VARCHAR 255 (not null)
* FK\_USER\_CERTIFICATE (On Delete/Update: Restrict) column: id
  + Database: registration
  + Table: users
  + Column: id

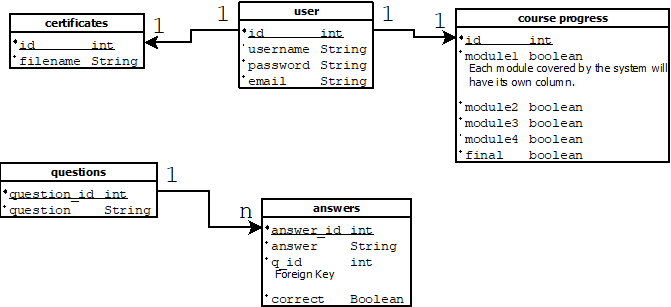
questions

* question\_id INT 10 (primary key, auto increment)
* question VARCHAR 200 (not null)

answers

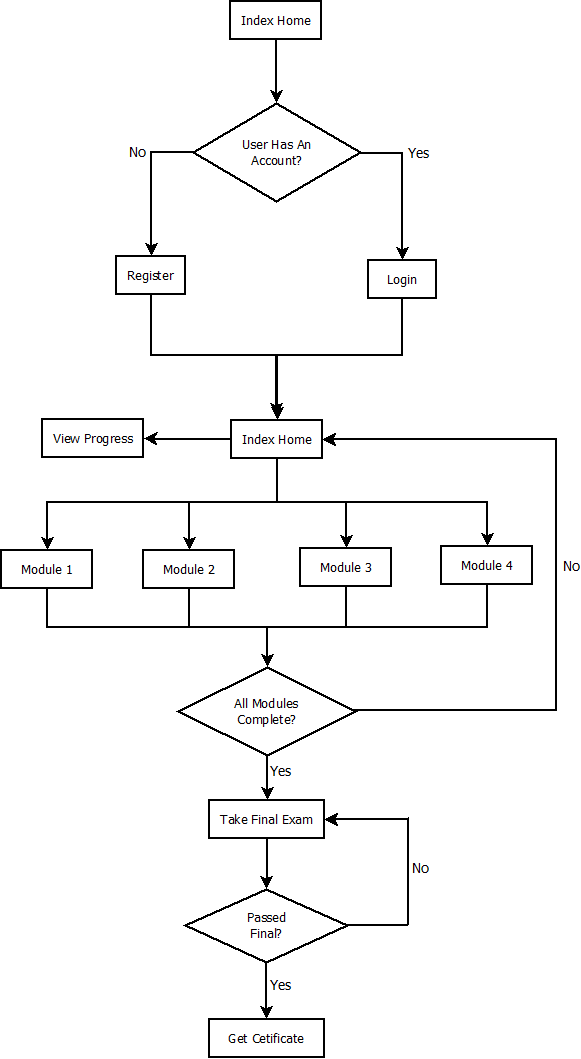
* answer\_id INT 10 (primary key, auto increment)
* answer VARCHAR 200 (not null)
* q\_id INT 10 (foreign key)
* correct BOOLEAN
* FK\_AssocQuestion (On Delete/Update: Restrict) column: q\_id
  + Database: registration
  + Table: question
  + Column: question\_id

## Database Diagram



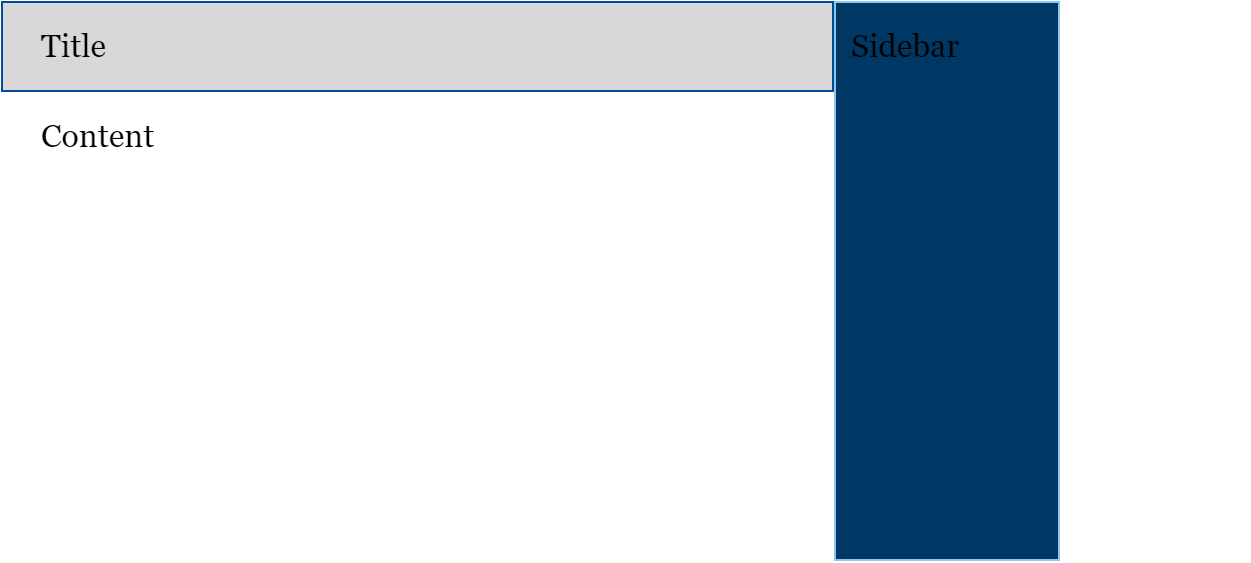
## Website structure and navigation

The overall structure of the website is a rather hierarchical system of progression, completing one part allows access to the next part of the system and so on. The general progression through the system would look something like:



As you can see, the system follows a somewhat linear flow, some things can be done in parallel, but there is a general flow through the system. These pages can be visited using the sidebar, but the functions associated with the pages cannot be accessed until the user completes the previous section of the system. For example, a user cannot take the final exam until they complete all of the modules.

A typical page of the system looks like:



The sidebar is used to navigate through the system.

### Color scheme

The color scheme we used for the website is the UWF argo color pallet, found [here](https://uwf.edu/brand/color/). Almost all colors in this system are drawn from that source.

# Project evaluation/critique

## Potential next steps

There are many different directions this project can go. Adding more modules would be a good first step. Keeping with the design of the website and just adding more would be easy. We could also make improvements with the game at the ends of the modules and try and innovate more with merging education and games.

As always, the system could always use general refactoring and bug fixing.

## What did/didn’t go as expected

None of us had any experience with building a project like this. So anything going as expected was exciting. Building the database was straightforward, but adding security measures proved more challenging. There were more precautions we need to take when building a project that holds private data we didn’t realize. We originally planned on making larger scope games in unity but it turned out harder than we thought. We wanted to make a unique game for each module and not only was it hard to bring education and gaming together effectively but it was even more time consuming to learn unity to build games that achieved what we wanted.

## Good/Bad decisions

### Good

Remaining flexible, when things weren’t working we were willing to change things up rather than head on.

### Bad

We did not do enough to adhere to plans we originally made. As an example, we wanted to do a scrum like development cycle, but that fell through. Maybe things could have gone better if we better communicated our schedules with each other.

## What was learned

We learned a lot during this project. We all got outside of our comfort zones doing this project and really challenged ourselves. I think we all got better at communicating relative to where we were individually. It has been very important for us to clearly deliver what we are thinking about to each other to make sure we are all on the same page. This was especially so when we made any changes in the project.