

Instructional Dashboard for Learning Networks

Design Document

Team Number: 12

Clients: Rachel Smith & Michael Brown

Adviser: Nicholas Fila

Team Members/Roles:

Brett Bethke

James Carter

Josh Caplinger

Nick Matthews

Sam Reinart

Steven Ehler

Tyler O'Hare

Team Email: sdmay22-12@iastate.edu

Team Website: <http://sdmay22-12.sd.ece.iastate.edu>

Revised: 12/05/21

Executive Summary

Development Standards & Practices Used

Software Practices

Waterfall/Agile project management style

Using version control software (Git)

Continuous Integration and Continuous Deployment (CI/CD)

Hardware

Virtual Machine (Hosted within Iowa State Ece)

Engineering Standards

IEEE-CS/ACM - (<https://www.computer.org/education/code-of-ethics>)

CLIENT – Engineers shall act in a manner in the best interest of the client consistent with public interest.

PRODUCT - Engineers shall ensure the product and related items meet the highest of standards possible.

FERPA (§99.5)-

Family Educational Rights and Privacy Act, US federal law governing the access to educational information. Since our application will be dealing directly with students grades this will be an important standard to abide by

Only instructors and explicitly authorized parties will have access to student data

Student data must be securely stored

NIST (SP 800-52) - (<https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final>)

Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations.

Providing encryption via TLS/SSL will help to ensure confidentiality and integrity of data, a key concern for the client.

NIST (SP 800-95) - (<https://csrc.nist.gov/publications/detail/sp/800-95/final>)

NIST Guide to Secure Web Services.

Implementing recommendations from the National Institute of Standards and Technology will help ensure best practices are being followed to provide a secure web application for the client.

OAuth 2.0 - (<https://datatracker.ietf.org/doc/html/rfc6749>)

Providing an OAuth compliant single sign-on service like OKTA for user authentication and management will provide a simple, yet secure way of logging in to the web application.

W3C Web Content Accessibility Guidelines (WCAG) 2.1 AA - (<https://www.w3.org/TR/2018/REC-WCAG21-20180605/>)

WCAG 2.1 AA will allow the web application to be more accessible to users who may have disabilities. Following the WCAG 2.1 AA guidelines will make our web application ADA compliant.

Summary of Requirements

- Friendly, professional, and intuitive UI that doesn't take long to understand how to use
- Ability to change filters to display different information in the dashboard, related to student demographics and performance
- Network graphs section that builds said graphs based on the results of Network Canvas surveys
- Iowa State color scheme and/or branding
- ADA compliant (accessible to disabled users, e.g. interoperable with screen readers)
- Loading time for the dashboards must be less than 1 minute
- Users will need to sign in via Okta with the university
- Data that has not been de-identified needs to be securely stored and cannot be in any cloud environment. (constraint)

Applicable Courses from Iowa State University Curriculum

SE Courses: 329, 339

COM S Courses: 227, 228, 309, 319, 363

DS Courses: DS 202

Communication Courses: SP CM 212, ENGL 314

New Skills/Knowledge acquired that was not taught in courses

R Programming Language

Shiny apps and Shiny server

OAuth 2.0 and Okta

Table of Contents

1 Team	6
2 Introduction	8
3 Project Plan	10
3.1 Project Management/Tracking Procedures	10
3.2 Task Decomposition	10
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	11
3.4 Project Timeline/Schedule	11
3.5 Risks And Risk Management/Mitigation	13
3.6 Personnel Effort Requirements	13
3.7 Other Resource Requirements	15
4 Design	15
4.1 Design Context	15
4.1.1 Broader Context	15
4.1.2 User Needs	16
4.1.3 Prior Work/Solutions	16
4.1.4 Technical Complexity	16
4.2 Design Exploration	17
4.2.1 Design Decisions	17
4.2.2 Ideation	17
4.2.3 Decision-Making and Trade-Off	18
4.3 Proposed Design	18
4.3.1 Design Visual and Description	18
4.3.2 Functionality	21
4.4 Technology Considerations	22
4.5 Design Analysis	22
4.6 Design Plan	22
5 Testing	23
5.1 Unit Testing	23

5.2 Interface Testing	23
5.3 Integration Testing	23
5.4 System Testing	23
5.5 Regression Testing	24
5.6 Acceptance Testing	24
5.7 Security Testing (if applicable)	24
5.8 Results	24
6 Implementation	25
7 Professionalism	25
7.1 Areas of Responsibility	25
7.2 Project Specific Professional Responsibility Areas	26
7.3 Most Applicable Professional Responsibility Area	27
8 Closing Material	28
8.1 Discussion	28
8.2 Conclusion	28
8.3 References	28
8.4 Appendices	29
8.4.1 Team Contract	29

List of figures/tables/symbols/definitions

List of Figures

Figure 1: Network Canvas network graph example

Figure 2: Project Timeline

Figure 3: Component Diagram

Figure 4: Use Case Diagram

Figure 5: UI/UX Design of Network Graph Dashboard

Figure 6: UI/UX Components

List of Tables

Table 1: Personnel Effort Requirements Broken Down By Task

Table 2: Products Impacts In Each Societal Area

Table 3: IEEE vs. NSPE Code of Ethics

1 Team

1.1 TEAM MEMBERS

Brett Bethke
James Carter
Josh Caplinger
Nick Matthews
Sam Reinart
Steven Ehler
Tyler O'Hare

1.2 REQUIRED SKILL SETS FOR YOUR PROJECT

- Front-end: For users to view the web based dashboard.
- Back-end: Used to store the clients data.
- Data Science: R programming language specifically, for scripts used by the client.
- Data Security: Confidentiality of sensitive data.

1.3 SKILL SETS COVERED BY THE TEAM

- Front-end: Brett, James, Josh, Sam, Steven, Tyler
- Back-end: Brett, Josh, Nick, Sam, Steven, Tyler
- Data Science: James, Nick, Sam
- Data Security: Brett, Tyler

1.4 PROJECT MANAGEMENT STYLE ADOPTED BY THE TEAM

We are using the Waterfall/Agile Hybrid project management style. Due to the fact that the number of person-hours will be limited because work on the project will only occur during the school year, it makes sense to use an hybrid approach. Our project will need less than 200 person-days of effort, the overhead of a purely agile management system will be too high. The waterfall method allows us to easily decompose and schedule tasks to complete, and then the tasks will be able to be completed in an agile manner.

1.5 INITIAL PROJECT MANAGEMENT ROLES

Brett: Client interaction
James: Testing
Josh: Team organization
Nick: Backend coordinator
Sam: Front-end coordinator, meeting runner
Steven: Individual component design
Tyler: Security coordination

2 Introduction

2.1 PROBLEM STATEMENT

In large classroom settings professors can have difficulties identifying how students are interacting in the classroom, and how those interactions affect the student. A network graph is a graph made up of nodes. A connection, shown as a line, between two nodes is called a branch and a single node can have multiple branches. Network graphs are a great way to visualize the connections between students in a class. An example network graph is shown below. This data is collected using the Network Canvas survey software application, but needs to be displayed in an intuitive and accessible way so instructors can effectively use it. Network Canvas outputs three different csvs which will be used to create network graph of classroom interactions filtered based on criteria (e.g. gender, attendance, exam grades, etc.) from the Canvas LMS API, needs to be displayed so instructors can understand how the students in their class work in order to teach them in the most effective manner.



Fig. 1. Network Canvas network graph example.

Our client is looking for us to build a prototype of this workflow and provide a proof of concept for research funding.

2.2 REQUIREMENTS & CONSTRAINTS

- Friendly, professional, and intuitive UI that doesn't take long to understand how to use
- Ability to change filters to display different information in the dashboard, related to student demographics and performance
 - Class grade, class attendance, lab attendance, exam grades, gender, time frame
- Network graphs section that builds said graphs based on the results of Network Canvas surveys (needs to use Network Canvas constraint)
 - Graphs load in a timely fashion (1 minute or less)
- Iowa State color scheme and/or branding
- ADA compliant (accessible to disabled users, e.g. interoperable with screen readers)
- Loading time for the dashboards must be less than 1 minute
- Users will need to sign in via Okta with the university
- Data that has not been de-identified needs to be securely stored and cannot be in any cloud environment (constraint)
- Canvas LMS graphs section with scatter plot graphs based on Canvas LMS class data (e.g. attendance and grades) (needs to use Canvas LMS APIs constraint)

2.3 ENGINEERING STANDARDS

- IEEE-CS/ACM - (<https://www.computer.org/education/code-of-ethics>)
 - CLIENT – Engineers shall act in a manner in the best interest of the client consistent with public interest.
 - PRODUCT - Engineers shall ensure the product and related items meet the highest of standards possible.
- FERPA (§99.5)- (<https://www.ecfr.gov/current/title-34/part-99>)
 - Family Educational Rights and Privacy Act, US federal law governing the access to educational information. Since our application will be dealing directly with students grades this will be an important standard to abide by
 - Only instructors and explicitly authorized parties will have access to student data
 - Student data must be securely stored
- NIST (SP 800-52) - (<https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final>)
 - Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations.
 - Providing encryption via TLS will help to ensure confidentiality and integrity of data, a key concern for the client.
- NIST (SP 800-95) - (<https://csrc.nist.gov/publications/detail/sp/800-95/final>)
 - NIST Guide to Secure Web Services.
 - Implementing recommendations from the National Institute of Standards and Technology will help ensure best practices are being followed to provide a secure web application for the client.
- OAuth 2.0 - (<https://datatracker.ietf.org/doc/html/rfc6749>)
 - Providing an OAuth compliant single sign-on service like OKTA for user authentication and management will provide a simple, yet secure way of logging in to the web application.

- W3C Web Content Accessibility Guidelines (WCAG) 2.1 AA - (<https://www.w3.org/TR/2018/REC-WCAG21-20180605/>)
 - WCAG 2.1 AA will allow the web application to be more accessible to users who may have disabilities. Following the WCAG 2.1 AA guidelines will make our web application ADA compliant.

2.4 INTENDED USERS AND USES

Our only users are Professors and TAs .

Professors would be able to login to our application via OKTA and be able to create or select an existing class. Once Professors have a class, they will be able to import their first survey data set taken with Network Canvas. Then the instructional dashboards would appear on the page with their most recent data shown. They would have the ability to filter the network graph via different criteria such as student attendance, grades, gender, etc to identify patterns and make inferences about student groups. Professors will also be able to import new sets of survey data to a class and be able to compare it to historical data from before. Using this tool they can make changes to their class and easily observe the effects using this tool.

Project Use Cases:

- Allow professors to identify groups of students struggling within their class and the patterns between them
- Allow TA's to coordinate better with their assigned sections
- Easy accessibility to a visual, organized representation of student groups
- Make classes modular by grouping students that all have something in common (same lab section, same lecture section, etc.).
- Allow instructors to collect information regarding classroom interactions and how those interactions affect student performance.

3 Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

We will use the Waterfall/Agile Hybrid project management style. Due to the fact that the number of person-hours will be limited because work on the project will only occur during the school year, it makes sense to use an hybrid approach. Our project will need less than 200 person-days of effort, the overhead of a purely agile management system will be too high. The waterfall method will allow us to easily decompose and schedule tasks to complete, and then the tasks will be able to be completed in an agile manner.

To keep track of our group's progress throughout the duration of the next semester we will use the following tools:

Git / Github Project: Project file collaboration, organization and task management

Discord: Team communication

Email: Client communication

Zoom: Advisor communication

3.2 TASK DECOMPOSITION

Research

- Learn about R, Shiny, and Network Canvas
- Research ADA, Security,
- Get initial client specifications and visions
- Learn about Canvas LMS API

Design

- Design app look and layout
- Design database schema
- Pick what information is shown to users and how
- Design filters for what data will be available to display

Setup

- Setup hosting server by installing Shiny server, R, and SQL database
- Properly setup git branches, tools and pipelines (CI/CD)
- Setup individual development environments

Development

- Implement database
- Create login and account system
- Secure with password encryption and Okta 2 factor authentication
- Create R scripts that, when run, will pull the necessary data from Network Canvas
- Shiny app UI and server

Testing

- Test functionality with client's "dummy" data
- Confirm ADA compliance

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- Schema designed with ability to input CSV
- Front-end and back-end communicating
- Application tests created and passed
- Working prototype of the application that requires the user to run a script to go from Network Canvas to the application
- Working prototype of the application that can handle going from Network Canvas to the application itself
- Application creates network graphs in time required by client (Loads in less than 1 minute)
- Application adheres to all accessibility requirements (ADA Compliance)
- Application is secure with password encryption and Okta 2 factor authentication.

3.4 PROJECT TIMELINE/SCHEDULE

Research:

- Research tools needed to create a Shiny app or related application. Date: 10/24/2021
- Research R programming language and Shiny app configuration. Date: 10/24/2021
- Create a simple Shiny app that can receive data from Network Canvas. Date: 10/24/2021
- Get information regarding Canvas LMS API and if it's feasible to use with this project. Date: 10/24/2021
- Check security measures that can be implemented to secure the web application. Date: 10/31/2021
- If Canvas LMS API implementation is feasible, research how to get Canvas LMS information onto Shiny app. Date: 11/19/2021

Design:

- Decide the front end visual aesthetic of the web app, research what's needed to make the web app ADA compliant. Date: 10/31/2021
- Task board and git set up. Date: 11/23/2021
- Decide database schema design. Date: 11/27/2021
- Determine statistics and graphs that will be shown on the Shiny app with the client. Date: 12/06/2021

Development:

- Create and configure a server to host the Shiny app. Date: 11/28/2021
- Creating a database to store web app information. Date: 11/28/2021
- Get a working Shiny app that can display some statistics and graph connections with Network Canvas data. (Frontend and backend implementation) Date: 03/13/2021
- Create and implement a login system for the web application. Date: 03/27/2022

Testing:

- Unit testing, manual testing, security testing of web app. Date: 04/24/2022

Demo:

- Demo of a working prototype Shiny web app for the client. Date: End of Spring Semester.

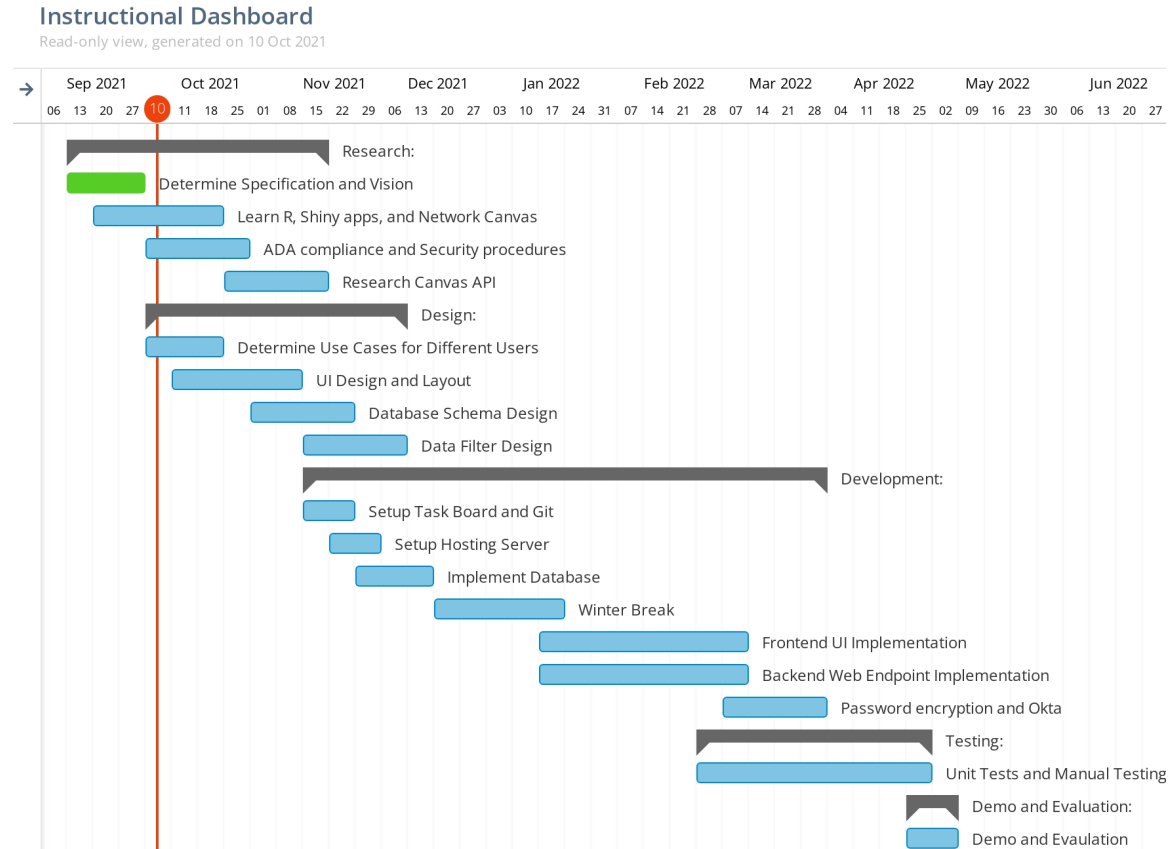


Fig. 2. Project Timeline

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

- Canvas LMS API incompatible/not feasible to use with project | risk: 0.6
 - Create alternative functionality that allows manual input of grades.
- Team inexperience with Shiny, R, and/or Network Canvas | risk: 0.7
 - Team members will learn about the tools and languages to be used in the project before development begins.
- Shiny and/or R not feasible to use with project | risk: 0.3
- App has security issue exposing student grades | risk: 0.5
 - Security will take high priority, making sure this doesn't become a problem. We plan on deidentifying the student data before it is stored on our application.
- Database retrieval too slow | risk: 0.2
- Product inefficiency because of methods to keep data secure | risk: 0.4
- Client not accepting graph visualization | risk 0.3

3.6 PERSONNEL EFFORT REQUIREMENTS

Task	Explanation	Projected Effort
Research R programming language	Become more familiar with R by experimenting with the language	120 hours
Research Shiny package	Become familiar with the R package Shiny by building a small application	180 hours
Determine feasibility of Canvas LMS API	Look deeper into Canvas LMS API to determine its effectiveness for the Shiny application	78 hours
Create software architecture diagrams	Architect the software system (use cases, components, interfaces, communication diagrams)	120 hours
UI design drafting	Mock up the front-end of the application using UI tools (Adobe XD, Sketch,...)	90 hours
Database design	Create ER diagrams and relational schemas for systems data	60 hours
Code repository setup	Initialize the source code repository to manage the versions of our application.	30 hours
Configure application server	set up and configure the server which will host the Shiny application.	78 hours
Implement database	Initialize, build, and host working SQL database	180 hours
Implement UI	Using the R Shiny package, construct the UI dashboard	210 hours
Backend authentication implementation	Using Okta, implement the authentication for accessing the app	126 hours
Write Tests	Create manual and unit tests to ensure the stability of the Shiny app	240 hours

Table. 1. Personnel Effort Requirements Broken Down By Task

Intermediate milestones:

We have 7 people working on this project with each person working approximately 9 hours a week. That allows us 63 working hours a week.

We broke down the amount of personnel hours needed into our 5 project phases.

Research: Around 378 Hours

Design: Around 378 Hours

Development: Around 756 Hours

Testing: Around 378 Hours

Demo: 2 Hours

3.7 OTHER RESOURCE REQUIREMENTS

Other Necessary Resources

- Git repository - should already be provided through this class
- Server space - For hosting SQL database and Shiny server
- Network Canvas accounts
- Canvas LMS API and data access permissions
- Computers with access to internet

4 Design

4.1 DESIGN CONTEXT

4.1.1 Broader Context

Area	Impact
Public health, safety, and welfare	<p>Product would help improve education for students by letting professors and teachers adapt their class room based on Network Canvas and Canvas LMS data.</p> <p>Product could lead to a breach in confidentiality or privacy of student data.</p>
Global, cultural, and social	<p>Product would change how professors and teachers adapt their courses to be more effective for their students leading to higher quality education. Professors often already survey students in the form of course evaluations so our product would simply have students take surveys during a course.</p> <p>Requires frequent and honest survey feedback from students.</p>

Environmental	Being a purely software based product, its environmental effects are minimal with only power consumption from server(s) hosting our application.
Economic	Product needs to be a free tool available for professors. Since this product is being designed for professors at Iowa State, development and maintenance costs would be covered by the University. This product will however affect and help our client in securing grant funding for their research

Table. 2. Products Impacts In Each Societal Area

4.1.2 User Needs

Teachers and TAs need a way to observe and filter data relevant to them in multiple ways to facilitate data analysis, because they want to identify patterns and issues with student groups.

Teachers and TAs need these dashboards to be easily accessible for convenience.

Teachers and TAs need these graphs to load in a timely fashion (< 1 minute).

Teachers and TAs need their students' data to be secure to protect student information and because of FERPA.

4.1.3 Prior Work/Solutions

This project is focused on taking the functionality of Network Canvas and expanding and specializing it for use by college professors in active learning courses. Network Canvas is a tool for gathering data about personal relations and networks and then visualizing and organizing that data in a useful manner. Network Canvas however, is unable to integrate and display Canvas LMS data with it. Our application will take the data from Network Canvas and combine it with information about student grades and other academic data to be more relevant and useful to professors.

Advantages:

- Professors able to access data easier and make classroom decisions based of findings
- Adds class performance data to personal network visualization

Drawbacks:

- Learning curve for how to understand data and make classroom decisions
- Very specific audience (Only for those using Network Canvas and Canvas LMS)

4.1.4 TECHNICAL COMPLEXITY

Our web application will contain several complex components. Currently, the complexity in which we are facing is finding a way to link our two different data sources for each student. We will implement an R application with the Shiny package to create a web application. This web application will be hosted using a Shiny server. The web application will connect to multiple APIs including OKTA and Canvas LMS, and combine that data with survey data gathered using Network Canvas. Then, our web application client will use the egor R package to create network graph visualizations filtered according to user-selected filters. We will be designing a database schema for MySQL to store information relevant to the visualizations displayed in the web application. Along with this we will be implementing web application encryption and security standards from NIST to make the web application as secure as possible.

4.2 DESIGN EXPLORATION

4.2.1 Design Decisions

Key design decisions and why they were made:

1. What data should be visualized: Project is about giving more information for professors to analyze, this includes the relationship network between students combined with Canvas LMS grade and participation data.
2. What data visualization language should be used: Project requires a final web application that can assist in visualizing data so we decided on R since its primary focus is on data analysis and statistics.
3. What color scheme should we use for the design of the web application: Since this application will be used solely by Iowa State University Staff, it was decided that we'd use the Iowa State color scheme of red and yellow.
4. What database should be used: We decided to use a MySQL database because through our research it integrates well with shiny applications, it is lightweight and fast, and our team has the most experience with it.
5. What network visualization library should be used: We have decided to use the egor library in R as it allows for integration with Shiny applications and is one of the only network visualizers available for R that is not deprecated. The egor package will allow us to make customizable ego networks with Network Canvas data efficiently. In this context, an "ego" is the focal point of the network during data collection and analysis. The ego is surrounded by alters, where each alter is nominated only by the ego.

4.2.2 Ideation

For deciding which data visualization language to use we considered which languages were the most popular (Python, R, Java, C#, Node.js) and what additional libraries each has access to (Pandas for Python, Shiny for R, React for Node.js, etc.). Majority of these languages are ones in which we learned about during our own industry experiences or recommendations from our client.

4.2.3 Decision-Making and Trade-Off

One decision in which we made as a group was to use the R programming language. This language would be used to analyze the data and make it simple for data visualizations to be created. We came to this decision as our client had recommended it to us and he has experience with it. As well as one of our group members also has experience in it and is willing to take lead in R development. R is also known for its ability to work with data and perform data analysis.

4.3 PROPOSED DESIGN

So far we have researched tools available for us to use in creating our web application. This includes things like R, what Shiny applications are, how they work, and how we can connect our data from the Network Canvas output into our web application. Also, we have set up both a Canvas LMS server as well as a Shiny web server. Also, we have set up both a Canvas LMS server as well as a Shiny web server. Right now we are in the process of creating a Shiny application that will take in the mock Network Canvas data provided by our client and process it.

4.3.1 Design Visual and Description

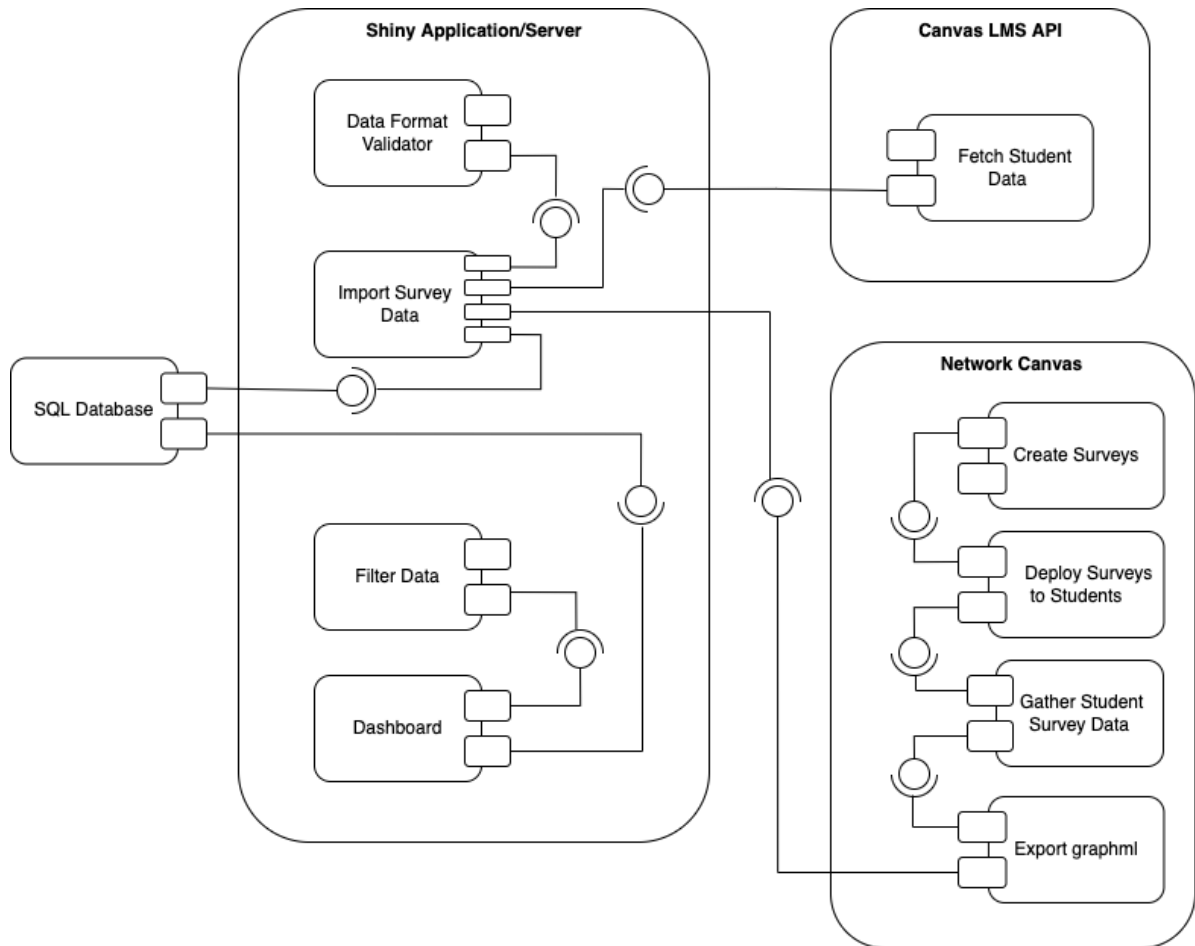


Fig. 3. Component Diagram

The system consists of 3 main components as seen in the component diagram (Figure 3): Shiny Application Server, Canvas LMS, Network Canvas, and an SQL database. The Shiny application server component contains the dashboard in which the user sees. The most important piece of this component is the “import survey data” module, which represents the users ability to import survey data from Network Canvas into an SQL database. Upon importing, the data is validated by a separate module, and relevant student data from Canvas LMS is fetched and added to the SQL database as well. The physical Shiny application dashboard interfaces with this SQL database to fetch and display data for the user to view.

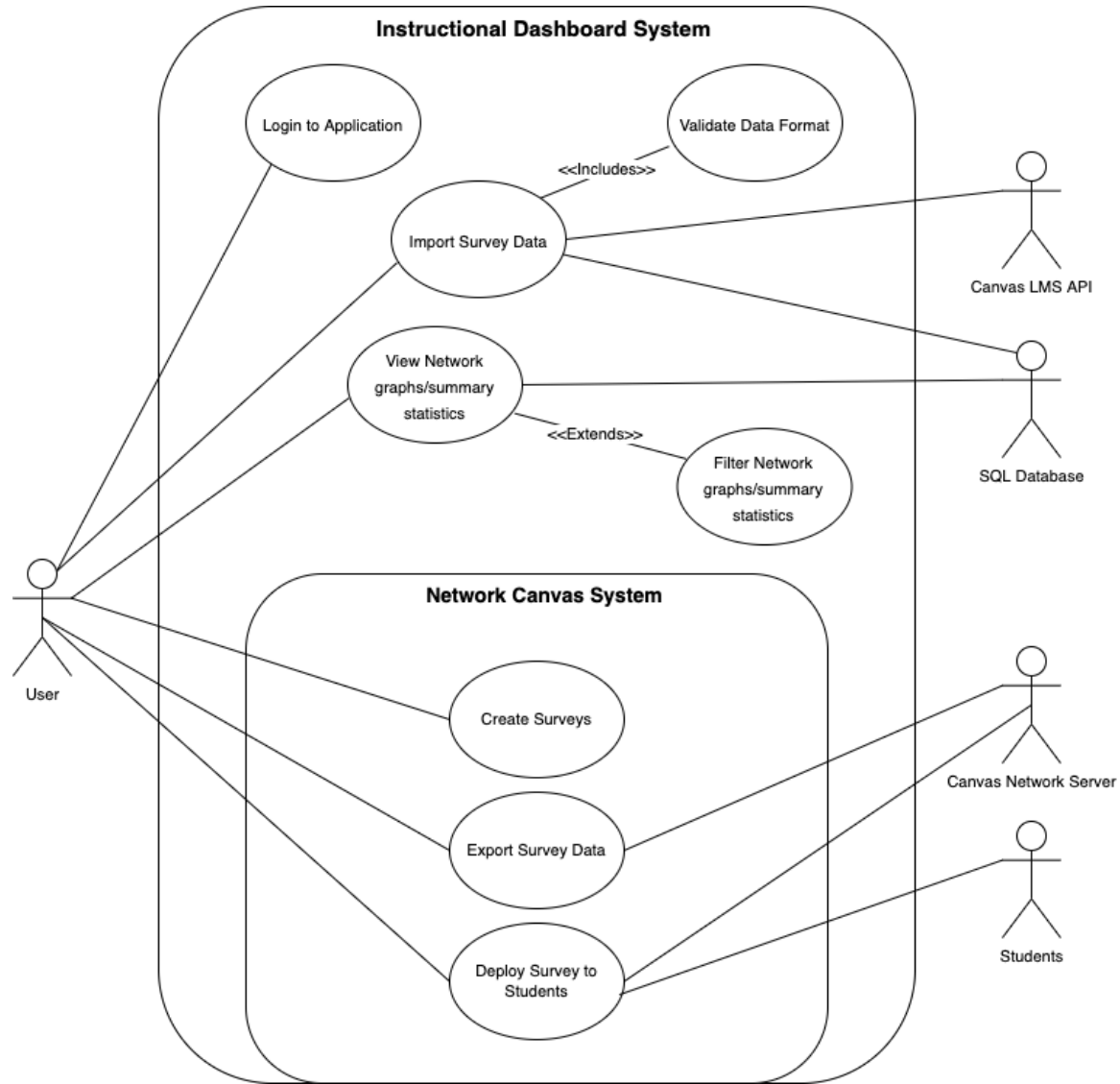


Fig. 4. Use Case Diagram

Figure 4 reveals a use case diagram that shows how each type of user will interact with the application and which pieces of functionality are available to them.

The user's job within the system is to make and collect survey data, then import it into the shiny application. Once that is complete, users can view and filter data according to their needs. As you can see in the figure above, validating imported data into the Shiny application is a required part of the process. Since the Network Canvas application is an already existing and complete system, it is separated into its own entity within the use case diagram.

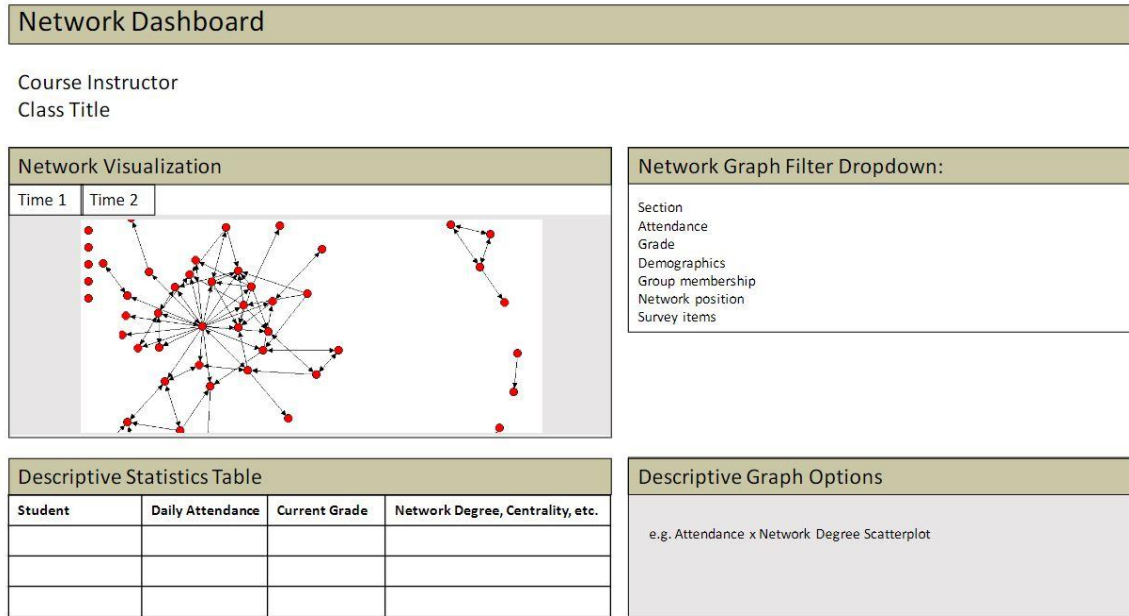


Fig. 5. UI/UX Design of Network Graph Dashboard

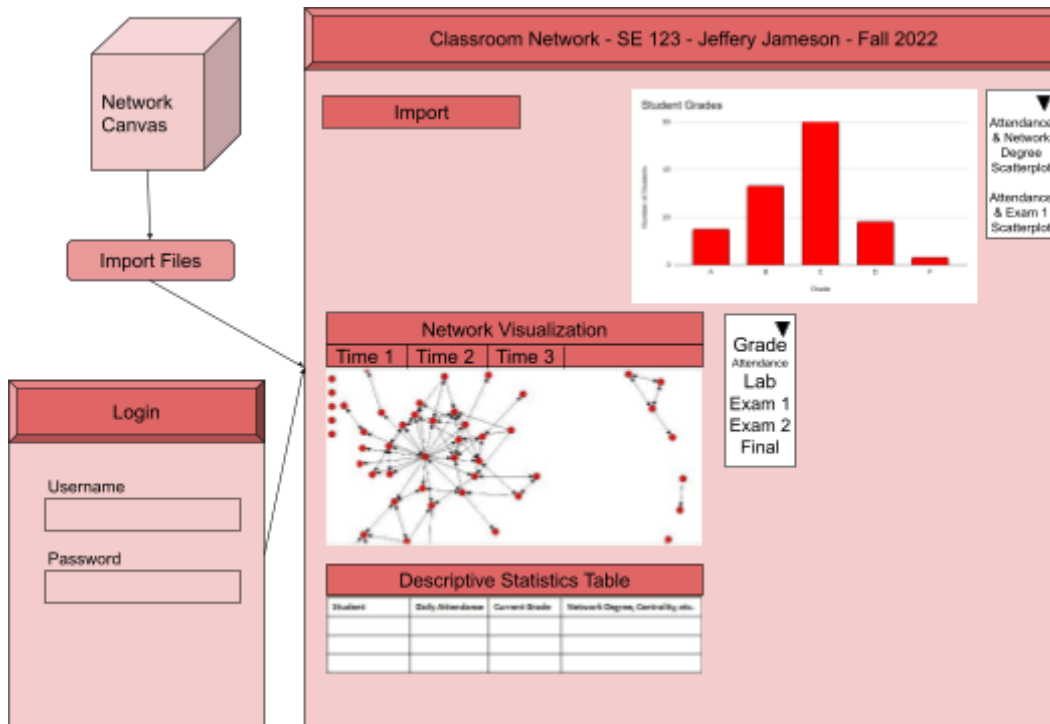


Fig. 6. UI/UX Components

Figure 5 and 6 display a diagram of the various user interface components to be included in our final design, including the ability for users to upload data into our web application from Network Canvas. The UI displays various summary statistics as well as network graphs, all in one page and controlled by layers of filters. As seen in figure 5, time-based filters change the network graph to display the selected semester. Furthermore, multiple dropdown options are available for the user to select class specifics such as section, attendance, grades, etc... Below the network graph is a table of descriptive statistics, giving the user insight into each student.

Figure 6 is a component based UI diagram which includes a login page and a Network Canvas module.

4.3.2 Functionality

Our design is intended to be run on a public facing web application server. Instructors and TAs will be able to log into the web application using OKTA to gain access to past and present statistics about their classes. The instructor will create a Network Canvas survey for their students to take, then our web server will automate the process of combining survey data with student Canvas LMS data and creating informational graphs and statistics about students and connections being made in the classroom. The instructors or TAs will then be able to view and filter the compiled data by student attendance, grades and other Canvas LMS data to provide them information about student connections compared to how students are doing in the class.

4.3.3 Areas of Concern and Development

One of the concerns is a lack of collective knowledge of the suggested language choice (R) and its subsequent package (Shiny). Other items like accessing OKTA and software interactions with ISU systems are also of concern. We are mitigating our lack of knowledge by setting high expectations for the amount of time spent in development and through the amount of research we have already done on R, Shiny and technologies we are using. The best solution for our concerns with integrating software interactions with ISU is to keep in constant communication with our client and with ISU IT about the subject.

4.4 TECHNOLOGY CONSIDERATIONS

Pros and cons of the technology chosen.

The strengths of R and RStudio are that they're widely used and easily accessible with plenty of tools available online to help with learning. R is best known for its use as a language used for statistical analysis, which aligns with the goals of our project.

One weakness of R and RStudio is that most of the content we need to build our application isn't standard (visualizations) and we had to find the libraries we felt would fit our client's needs best.

Another weakness of using R and Shiny for our project comes from the team's lack of knowledge and experience using R and Shiny. This does mean the team will require a lot more time in development.

The Shiny package includes many different functions that make it easy to create the different components of a dashboard used for interactive data analysis.

Strengths we determined from using a MySQL database include, being easily integrated into Shiny applications with readily available documentation, MySQL is fast and lightweight, and our team is most proficient in using MySQL.

4.5 DESIGN ANALYSIS

With the amount of prototype implementation done, we know that our proposed design does work and the libraries that we chose work as expected from our research. Shiny applications have been confirmed to have the functionality needed to cover our clients requirements as they properly visualize network graph data, are easily accessible and can be manipulated to suit the user's needs. Some difficulties that we encountered were with the initial installation of the shiny server and the packages we needed, but not an issue with our design.

4.6 DESIGN PLAN

The project's design goal is to create a tool for teachers and TAs to be able to analyze group relations and patterns and compare them to student performance. Our project will take data from Network Canvas and Canvas LMS, use R to process that data, create data visualizations with the processed data, and then use Shiny to create an interactive dashboard of the data visualizations from R, including GUI. The project will also store historical data for professors to compare to newer data.

5 Testing

In order to thoroughly test our complete product, we will use a variety of testing methods to ensure that each component of our project functions according to our specifications. In order to make this testing process practical and streamlined, we have divided our project into components that can be first tested individually, and then working together.

5.1 UNIT TESTING

Units to be tested

- Shiny Web Application: Test R Snapshot-based tests for testing user-interaction with the web application
- Shiny Server: Test server functions behave as expected by testing their inputs and outputs with unit tests

5.2 INTERFACE TESTING

Interfaces to be tested

- Network Canvas Data
 - Network Canvas data that is imported into the Shiny application will need to be consistent. Therefore, on importing data, we will need to test the format before adding to our SQL database
- Canvas LMS API
 - Postman for quick testing during development. Jest framework will be used to test interactions (fetching) from Canvas LMS

5.3 INTEGRATION TESTING

The critical integration paths of our design include the following:

1. Shiny server backend receiving data from the Canvas LMS API
2. Shiny server backend storing and retrieving data from the SQL database
3. Shiny web application displaying data stored in the SQL database
4. Shiny web application using Shiny server backend to store an uploaded Network Canvas .csv file in our SQL databases

Our R code can be tested using R's built-in testing features. Our backend can be tested using the shinytest package. We will verify our testing results by ensuring all necessary data is stored in our database and displayed correctly on our web application.

5.4 SYSTEM TESTING

For system testing we plan to test the importing of Network Canvas data, converting it into a usable format, storing it in our MySQL database, and displaying it properly on the web application dashboard. Logging into and out of the application will also be tested. Our system testing will be a combination of integration, interface, and acceptance tests.

5.5 REGRESSION TESTING

While making changes to the live application we can implement two GitLab projects in which will merge master to master through the both of them while having one for testing and making changes purposes and the other for live use. This will allow us to have regression testing within our application.

As bugs are something which can never be avoided so regressions are going to be needed at some point. One amazing feature in which we can take advantage of regressions is GitLab's rollback feature. This will allow us to have the best experience for users with minimal to no downtime while we debug and error trace.

Finally, one thing that will help with ensuring functionality as additional features are implemented is extensive documentation. This will ensure that future developers can understand how everything was implemented in our version.

5.6 ACCEPTANCE TESTING

We will apply tests that make sure our project produces the desired outcome from an input, and ensure that the non-functional requirements hold true. As functions and non-functional requirements are fulfilled, we will demo the related features to the client to make sure they are happy with them. Potentially at the end of the project we may test the project with a user that is not our client.

5.7 SECURITY TESTING (IF APPLICABLE)

The first type of security tests we will perform are access control tests. Making sure each type of user: (Developer, Instructor, TAs) have only the permissions they are intended to. This can be tested manually by making clearly defined rules for each user and attempting to bypass those rules. If we are unable to bypass each rule for each user, the test is passed.

Another set of security tests we will perform are input and data validation tests. Fuzzing any areas of input to test for unintended behaviors, making sure the possibility for things like SQL injections are not possible. Testing things like edge cases, unintended data types, and data that's not properly formatted as inputs. If our web app can correctly handle all the cases it passes.

We will also perform sniffing tests on our web application. Using a packet sniffing application we will test that all sensitive data that enters and leaves the application is encrypted. The test is passed if no sensitive data is collected in plaintext.

5.8 RESULTS

Results of our testing will not be available until we have finished at least one unit of our design. We will ensure compliance with the requirements by testing that the requirements are met. Our system tests will ensure that our design works as intended. If every critical integration path in our design works, then we will have met our client's requirements.

6 Implementation

Our team plans to start implementing our design by building the UI design agreed upon by us and our client. This will involve several iterations where we will confirm with our client that the design includes all their needs. We already have a Shiny server and Shiny application running on our virtual machine server, so all we need to do is change the UI to the design we have created. The features available through the user interface can be found in Section 2.2 of this document. A visual representation of the planned interface can be found in Section 4.3.1.

We will also set up the back end by designing and implementing database schema with MySQL to store data. Then we will deploy an instance of Canvas LMS to generate student data to be consumed on the front end. This data will be accessible via the Canvas LMS API.

Once both of these steps are complete, the final step is to connect our front end UI to the back end, both through API calls to our instance of Canvas LMS and a direct connection to our MySQL database. The design will be considered complete upon approval from the client and the conclusion of our testing.

7 Professionalism

7.1 AREAS OF RESPONSIBILITY

	IEEE ethics	NSPE ethics	IEEE vs NSPE
Work Competence	Improve ourselves, focus on tasks we are competent at, and disclose skill limitations.	Leave work to those who know how to do it.	IEEE allows working on areas that one doesn't have full competency in if that is disclosed and accepted by others.
Financial Responsibility	Reject bribery. Avoid conflicts of interest or disclose them.	Use employer and client resources responsibly.	The NSPE is more strict on finances and includes rules against double billing.

Communication Honesty	Make honest and realistic claims and estimates.	Be honest in public statements.	Both discuss being honest with your employer, but the NSPE goes further, compelling engineers to advise their employers in certain situations and not to mislead.
Health, Safety, Well-Being	Be responsible with the health, safety and welfare of the public. Disclose factors that may endanger the public. Avoid injuring others.	Prioritize the health, safety and welfare of the public.	The NSPE encourages engineers to not only protect health, safety, and well-being, but also work for its advancement in their community.
Property Ownership	Avoid damaging others property or reputation by neglect or maliciousness.	Respect the property and ideas of employer and client.	The NSPE mentions that the property used and created while working on an employer's project is wholly owned by the employer and may not be used elsewhere without consent.
Sustainability	Disclose factors that may endanger the environment. Understand the potential consequences of used technology	Prioritize sustainable development to protect the environment.	The IEEE includes rules about disclosing the factors that may bring harm to the environment.
Social Responsibility	Avoid discrimination and treat others fairly.	Act in a way that reflects well on your profession.	The IEEE specifically mentions that engineers are not to discriminate against others, while the NSPE only says to act in a way that reflects well.

Table. 3. IEEE vs. NSPE Code of Ethics

7.2 PROJECT SPECIFIC PROFESSIONAL RESPONSIBILITY AREAS

Work Competence - This will apply to all aspects of our project as this area will allow for us to ensure the best possible outcome of our project and without this the project would fail.

Performance: Medium, we have been meeting our clients expectations and keeping pace with course timeline and project timeline.

Financial Responsibility - Little to no financial responsibility in which we are in charge of. The only financial responsibility of this project would be the server space in which we are using, which is currently being handled by the university's senior design fund. Performance: Not applicable

Communication Honesty - This area of professional responsibility will primarily apply to our relationship and communication with our client. Communication honesty is vital for setting appropriate expectations and trust/reputation. Performance: High, we have been very honest with our client and have been sharing all details in our progress on the project. Even sharing information that may not even pertain to the client.

Health, Safety, Well-Being - No health aspects which should matter on this project. Performance: Not applicable

Property Ownership - This professional responsibility applies by utilizing server space and data responsibly as well as respecting our clients ideas. Performance: High, we have a VM and dummy data for development. We also have always been open and respectful to our client's ideas in fact their ideas have been very helpful in the design of the project.

Sustainability - This project has no danger to the environment since it is a purely software based product. Performance: Not applicable

Social Responsibility - Social responsibility will always apply where there is interaction between people, whether that be our client or users, and data pertaining to people which our project does work with. Performance: High, we will be de-identifying data to protect individuals and we as individuals are very respectful to each other within our team and our client.

7.3 MOST APPLICABLE PROFESSIONAL RESPONSIBILITY AREA

Social responsibility is one of the most important professional responsibilities for our project. The product we are creating will benefit society and the communities that utilize it in various ways. Our instructional dashboard will allow its users (instructors and teaching assistants) to gain more valuable insights into their classroom environments. Users will be able to see connections being formed between students over the course of a semester and how those connections are directly impacting their performance in the classroom. These insights will provide the users the ability to identify students in need of support within their classes much more easily.

It is our job to be proficient in social responsibility for our project so that the project will be a good example for what is achievable in future iterations of the project. Our proficiency in social responsibility is demonstrated in our attention to detail of our clients' needs as they've communicated them to us. We will be working with sensitive data so we must handle it both ethically and lawfully so that it will make our project a feasible candidate for full implementation in

the future. Along with this, we are following standards regarding web application security and accessibility that will be in the best interest of those who will be in contact with our project both directly and indirectly. We are conducting ourselves in a socially professional manner by putting in the utmost effort into ethically, lawfully and responsibly implementing our project in a way that we know will be of benefit not only to our client, but to future developers and users of the project.

8 Closing Material

8.1 DISCUSSION

As we approach our second semester, we are finished with the research and design stage of our project and have enough knowledge and tools to begin the development stage. As of right now, We have:

- Secured a virtual machine for our web server with proper resources.
- Installed a Canvas LMS server to create mock classes we can pull data from.
- Installed rstudio for Shiny app development and deployment on the server.
- Configured the team's github and CI/CD tools so they are ready for immediate use.
- Researched and acquired a knowledge base in R, Shiny applications, R libraries such as egor, APIs for Canvas LMS, and OKTA.

This is inline with our goals in our Gantt chart and will continue to use the Gantt chart as a reference to where we should be progress wise on the project. We are not currently facing any issues with our development.

8.2 CONCLUSION

Our goals at the beginning of the semester were to get a better understanding of the R programming language and Shiny applications. Along with this, we had the goals of setting up our environment to begin development at the beginning of next semester. We accomplished our goal to accumulate the knowledgebase and toolsets necessary to set up a Shiny web server and Canvas LMS server in our virtual environment. Along the way, we had some difficulties achieving our goals such as technical difficulties with setting up our virtual machines and their services. In the end we were able to successfully implement the services. Our goals are now to begin the development stage of our project. Upon the start of next semester we plan to begin implementing our Instructional dashboard Shiny app. The best way for us to stay on track with the progress of our project is to make sure we are staying inline with our Gantt chart.

8.3 REFERENCES

A. Singhal, T. Winograd, and K. Scarfone, "Guide to secure web services," *nist.gov*, 29-Aug-2007.

[Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-95/final>. [Accessed: 30-Nov-2021].

"Code of Ethics: IEEE Computer Society." *IEEE Computer Society Code of Ethics Comments*,

<https://www.computer.org/education/code-of-ethics>.

D. Hardt, "The OAuth 2.0 Authorization Framework," *datatracker*, Oct-2012. [Online]. Available:

<https://datatracker.ietf.org/doc/html/rfc6749>. [Accessed: 30-Nov-2021].

"Family Educational Rights and Privacy Act (FERPA)." *Home*, US Department of Education (ED), 25

Aug. 2021, <https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html>.

K. McKay and D. Cooper, "Guidelines for the selection, configuration, and use of Transport Layer

Security (TLS) implementations," *nist.gov*, 29-Aug-2019. [Online]. Available: <https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final>. [Accessed: 30-Nov-2021].

"Web content accessibility guidelines (WCAG) 2.1," *W3C*. [Online]. Available:

<https://www.w3.org/TR/2018/REC-WCAG21-20180605/>. [Accessed: 30-Nov-2021].

8.4 APPENDICES

Network visualization package docs: <https://cran.r-project.org/web/packages/egor/egor.pdf>

Network Canvas tutorial: <https://documentation.networkcanvas.com/tutorials/working-with-data/>

Installing and deploying Canvas LMS: <https://github.com/instructure/canvas-lms/wiki/Quick-Start>

Shiny Server Admin Guide: <https://docs.rstudio.com/shiny-server/#stopping-and-starting>

GitLab Runner Documentation (CI/CD): <https://docs.gitlab.com/runner/>

8.4.1 Team Contract

Team Name: sdmay22-12

Team Members:

1) Samuel Reinart	2) Josh Caplinger
3) Brett Bethke	4) Jimmy Carter
5) Steven Ehler	6) Tyler O'Hare
7) Nicholas Matthews	

Team Procedures

- 1. Day, time, and location (face-to-face or virtual) for regular team meetings:**
Virtual (over Discord): Wednesdays at 8:00pm CST
- 2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):**
Use the Discord server to communicate to other team members, @ing them if necessary.
For scheduling we will use when2meet to show the team's availability.
- 3. Decision-making policy (e.g., consensus, majority vote):**
We will make decisions upon a consensus. If a consensus cannot be reached after a reasonable amount of time, then majority voting will be used.
- 4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):**
We will use a shared Google Drive folder to store notes and any other relevant documents.
Every team member will take notes relevant to their role at each meeting.

Participation Expectations

- 1. Expected individual attendance, punctuality, and participation at all team meetings:**
All team members that are able to come to a meeting are required to attend and participate. If you are unable to attend, you are required to let the other team members know and get caught up.
- 2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:**
When the team divides up work to be completed by specific team members, they will be required to do it before the deadline. If they know they will be unable to complete their required task, they need to let the other team members know as soon as possible.