DigiClips

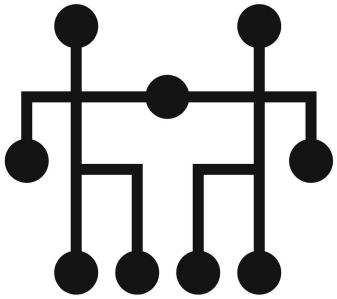
System Administrative Website

sddec24-Team03

Jan 16, 2024 - Dec 20, 2024

https://sddec24-03.sd.ece.iastate.edu/

Faculty Advisor: Dr. Ashfaq Khokhar



Team Members/Role:

Israel Sanchez Tellez - Client Interaction Lead

Derek Brandt - Architecture Lead

Tyler Orman - Security Lead

Aryan Rao - Programming Lead

Executive Summary

Development Standards and Practices Used

Our team uses an **Agile** approach to software development and a **Waterfall** approach to design development. We will have **code reviews** through *Github* pull requests and **automated testing** with expected 100% branch coverage. Code will be documented through **code comments** and a **readme**.

We will adhere to the **Angular Coding Style Guidelines**, the **Google Typescript Style Guidelines**, and the standard **IEEE 23026:2023** Engineering and Management of Websites for Systems, Software, and Services Information.

Summary of Requirements

Functional requirements

- The administrator portal will display all error messages from backend machines.
- The administrator portal will display statistics of the uptime of backend machines.
- The administrator portal can modify configuration files that control backend machines.
- The administrator portal will require login credentials.
- The administrator portal can modify user login information from the search engine.
- The administrator portal will allow super-administrators to manage administrators.

Aesthetic/UI Requirements

- Graphs will be labeled and titled clearly and interactable.
- UI Pages will be logical, well-structured, and visually appealing.

Constraints

- Each page will be accessible in less than or equal to three clicks. This will limit the complexity of the UI for the client.
- The software will have to run on *Amazon Lightsail* with a 99% uptime to allow the client to manage their users and machines effectively.

Applicable Courses

COM S 227 Object Oriented Programming, COM S 228 Introduction to Data Structures, COM S 309 Software Development Practices, COM S 319 Construction of User Interfaces, COM S 363 Introduction to Database Management, S E 317 Introduction to Software Testing, S E 329 Software Project Management, S E 339 Software Architecture and Design

New Skills and Knowledge Acquired

Gained experience with the Angular framework and Amazon Lightsail.

Table of Contents

Table of Contents....2
List of Figures and Tables....4
Figures....4
Tables....4
1. Introduction....5
1.1. Problem Statement....5
1.2. Intended Users....5
2. Requirements, Constraints, and Standards....8
2.1. Requirements & Constraints....8
2.2. Engineering Standards....8
3. Project Plan....10
3.1. Task Decomposition and Project Timeline....11
3.2. Milestones and Metrics....14
3.3. Risk and Mitigation....15
3.4. Personnel Effort Requirements....16
3.5. Resource Requirements....17
4. Project Design....19
4.1. Design Decisions and Ideation....19
4.2. Proposed Design....20
4.3. Technology Considerations....23
4.4. Trade-offs and Design Alternatives....24
4.5. Design Analysis....24
5. Testing....25
5.1. Unit Testing....25
5.2. Interface Testing....25
5.3. Integration Testing....26
5.4. System Testing....26
5.5. Regression Testing....27
5.6. Acceptance Testing....27
5.7. Security Testing....28
5.8. Result....28
6. Implementation....29
7. Professional Responsibility....30
7.1. Areas of Responsibility....30
7.2. Project Specific Professional Responsibility Areas....32
7.3. Most Applicable Professional Responsibility Area....33
8. Closing Material....34
8.1. Discussion....34
8.2. Conclusion....34

8.3. References..... 35
8.4. Appendices..... 35
9. Team.....36
9.1. Team Members..... 36
9.2. Required Skills for Project..... 36
9.3. Skill Sets Covered by Team..... 36
9.4. Project Management Style Adopted by Team..... 36
9.5. Initial Project Management Roles..... 36
9.6. Team Contract..... 37

List of Figures and Tables

Figures

- Figure 1. A Decomposition of Tasks Chart
- Figure 2. Research Related Tasks' Gantt Chart
- Figure 3. Design Related Tasks' Gantt Chart
- Figure 4. Implementation Related Tasks' Gantt Chart
- Figure 5. High-Level Block Diagram for Backend
- Figure 6. Administration Portal Inter Component Communication
- Figure 7. The UI decomposition of the administrative portal

Tables

- Table 1. A list of tasks for the research phase and their dependencies
- Table 2. A list of tasks for the design phase and their dependencies
- Table 3. A list of tasks for the implementation phase and their dependencies
- Table 4. Risks Severity Table and Affected Tasks
- Table 5. Task Hour Requirement Breakdown
- Table 6. Decision Matrix for Project Format

1. Introduction

DigiClips is a company that monitors news sources such as radio, television, and news articles. This is accomplished through various computers and subsystems running nonstop with complicated algorithms. This data is then provided to customers through a custom search engine hosted on a website.

1.1. Problem Statement

The computers, subsystems, search engines, and client information all need a form of administration. The computers and subsystems must be monitored for uptime, total time running without a complete system crash, and error checking/processing. The search engine website also needs to be checked for uptime and errors. Finally, the customer's information must be managed and verified. Accessing this data through each system individually is time-consuming. This causes data to be lost and customers to be delayed, which is costly for *DigiClips*.

To solve this, our team will develop a website accessible anywhere (given internet access). The website will have visual indications of server and computer status, display important statistics regarding computers and subsystems, remotely access configuration files for the video, radio, and image processing computers, and manage costumes' information. This will streamline the process of error detection and resolution and customer support.

1.2. Intended Users

There are four primary users we will target. Since this administration portal is not client-facing (only *DigiClips* employees will be able to use it), each user represents a different *DigiClips* employee.

DigiClips Managers / Owners

DigiClips managers are a group of users who want to broader system overview. These are people between the ages of thirty and fifty. Their backgrounds will be comprehensive of mainly business related fields with some software engineering. They will not know the technical details of computers running the algorithms; however, they will have the following needs. Their needs are as follows:

- They need to see the computer and server status to ensure their systems are up.
- They need to have statistics available for user traffic and machine uptime.
- They need to be able to manage the permissions of other users of the admin portal.

DigiClips managers benefit from the administrative portal by viewing statistics of their machines and user interactions. These statistics, combined with the server and computer status, allow the managers to allocate company resources more efficiently. The managers

need to be able to control the permissions of their employees so information abuse doesn't occur.

Customer Support Representative

The customer support representative is a group of users that will the client-facing for *DigiClips*. These users will be answering questions from clients regarding their accounts and the state of the service. They will primarily be people aged twenty to forty and likely have little to no software experience. Their needs are as follows:

- They need to see the server status to inform customers properly.
- They need to access the state of customers' accounts.
- They need to know the times when machines were in error.

The customer support representative benefits from the administrative portal by efficiently accessing customer accounts and machine information. Previously, this information was contained in a database that was difficult to access and nearly impossible to read. We will improve the efficiency with which the customers of *DigiClips* will receive their support.

Media Recording Engineer

The media recording engineers are a group of users who are software engineers. They work on *DigiClip*'s computer systems that process the video, audio, and image data. They are aged twenty to forty and have extensive knowledge of software. They will have technical knowledge of how the backend systems are functioning. Their needs are as follows:

- They need to know what computers are down and are in error states.
- They need in-depth knowledge of the errors occurring in each computer.
- They need to be able to manage the configurations of each computer remotely.

Media recording engineers benefit from the administration portal by quickly accessing error data regarding their computers and subsystems. Accessing error data through each machine and then each subsystem or through the database is labor-intensive and time-consuming. The administrative portal will reduce this overhead.

Software Development Engineer

The software development engineers are users continuing work on the administrative portal. They will be aged twenty to thirty and have extensive software and development knowledge. They will know how to read code and know technical terms related to databases, websites, and frameworks. Their needs are as follows:

- They need documentation on how to compile and execute the administrative portal.
- They need comments on the code explaining what the code is doing.

- They need design documentation and interaction diagrams to continue in-progress features.

The software development engineer does not benefit from the administrative portal themselves. They will not be using it but developing it. Thus, they benefit from the practices used to create the administrative portal. These practices make the code readable, easy to understand, and easy to compile, decreasing overall development time.

2. Requirements, Constraints, and Standards

2.1. Requirements & Constraints

There are three main groups of requirements: functional, aesthetic/UI, and constraints. Our functional requirements have been derived from the clients' specifications of the proposed project and are as follows:

- The administrator portal will display all error messages from backend machines.
- The administrator portal will display statistics of the uptime of backend machines.
- The administrator portal can modify configuration files that control backend machines.
- The administrator portal will require login credentials.
- The administrator portal can modify user login information from the search engine.
- The administrator portal will allow super-administrators to manage administrators.

The aesthetic/UI requirements have been derived from standard UI practices and team discussions:

- Graphs will be labeled and titled clearly.
- Graphs will be interactable to allow for data to be read.
- UI Pages will be logical and well-structured.
- UI Pages will be visually appealing.

The last section of requirements has been classified into constraints. These are direct quantifiable limiters of the UI and software environment. They are as follows:

- Each page will be accessible in less than or equal to three clicks. This will limit the complexity of the UI for the client.
- The software will have to run on *Amazon Lightsail* with a 99% uptime to allow the client to effectively manage their users and machines.

2.2. Engineering Standards

The implemented software will follow the *Angular Coding Style Guidelines* and *Google Typescript Style Guidelines*. The *Angular Coding Style Guidelines* ensure all code written in the Angular framework will look the same. This improves future readability and readability amongst the team. For the same reasons, the *Google Typescript Style* will be followed to ensure the rest of the code, written in TypeScript, will be readable amongst team members and future developers.

IEEE 23026:2023 Engineering and management of websites for systems, software, and services information

- HTTP/HTTPS: Our project will use the HTTP/HTTPS protocols for communication between the frontend Angular application and the backend server. These protocols ensure secure and reliable data transmission over the Internet.

- Security Standards: Implementing security best practices, such as input validation, authentication, and authorization, is crucial to protect the application from vulnerabilities.
- Performance Optimization: Standards for optimizing the performance of Angular applications, such as ‘lazy loading modules,’ minimizing bundle size, and optimizing rendering, will be adopted to ensure a smooth user experience.
- Testing Standards: Adhering to testing standards for Angular applications, such as writing unit tests and end-to-end tests, will ensure the reliability and robustness of the application.
- Security: Adhering to best practices for front-end applications, such as preventing SQL vulnerabilities and ensuring sensitive information is not leaked.

3. Project Plan

Our team will use a hybrid approach to project management. Three phases are identified: research, design, and development (See *Figure 1*). These phases will be discussed in detail in the following sections.

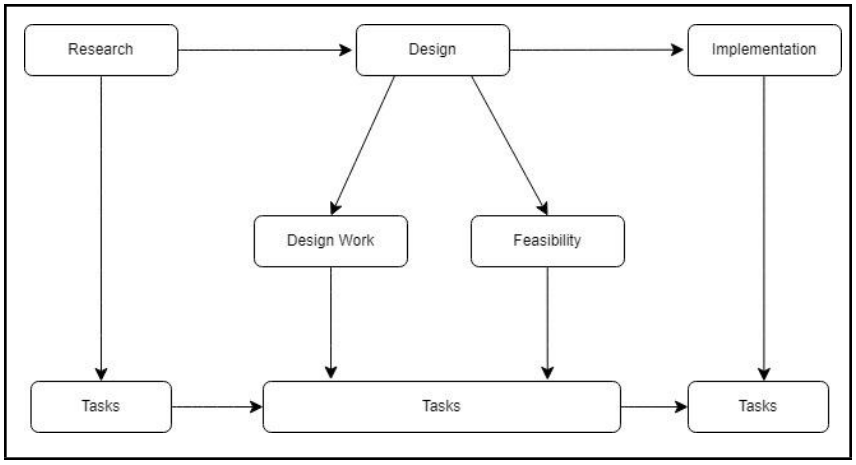


Figure 1. Decomposition of Tasks Chart

The phases' progression is linear; thus, they will be done in a *Waterfall* management style. This stems from the fact that the project requirements must be understood before any development or design can be completed. Also, due to an existing implementation, our team wants to fully understand the current state of the UI and codebase before planning our designs and implementation. Design and development are also linear. Before spending resources on the implementation, we need confirmation from the client on our team's designs.

These significant phases will be broken down into tasks/stories. This means we will use the *Agile* project management style. This is beneficial as different UI parts will take different lengths to implement; however, almost all UI can be designed and implemented in parallel. Breaking down the current implementation into sections will allow the research to be done in parallel.

A list of tasks will be compiled into a shared spreadsheet in which tasks will be marked as backlogged, in progress, in review, or completed. There will also be two dates kept: date started and date completed. The final information column will be who is assigned to the task.

3.1. Task Decomposition and Project Timeline

As shown in *Figure 1*, our phases act as our task groups. First, we will look at the Research tasks.

Task #	Description	Dependent On
1	Identify technologies used.	N/A
2	Map current implementation architecture.	N/A
3	Gather an estimate of the UI component count.	N/A
4	Identify missing ui components (Components still need to be implemented).	N/A

Table 1. A list of tasks for the research phase and their dependencies

As we can see, research-related tasks can be done in parallel. However, we have opted to prioritize identifying technologies being used and mapping the current implementation architecture to allow us to understand the code base we will be modifying and the UI components that will need to be implemented (See *Figure 2*).

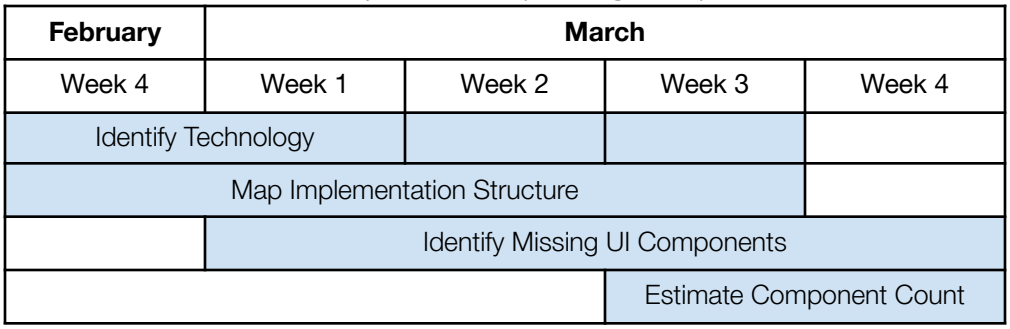


Figure 2. Research Related Tasks' Gantt Chart

Toward the end of the research phase, we will identify the components that will be used and work with the client to identify any remaining UI features that need to be implemented.

Now, we will look at the design-related tasks.

Task #	Description	Dependent On
5	Design Backend Dashboard	N/A
6	Design Customer Support Dashboard	N/A
7	Design Super Admin Dashboard	N/A
8	Design Machine Status Dashboard	N/A
9	Design User Management	N/A
10	Design Backend Dashboard Popups	5
11	Design Customer Support Dashboard Popups	6
12	Design Support Admin Dashboard Popups	7
13	Run feasibility check on UI components.	5-12

Table 2. A list of tasks for the design phase and their dependencies.

Most tasks in the design stage can be done in parallel; however, planning them all simultaneously would overload the team. Each design task follows a two-week-long design model (See *Figure 3*).

April				**May**
Week 1	Week 2	Week 3	Week 4	Week 1
Backend Dashboard				
Customer Support Dashboard				
Super Admin/Owner Dashboard				
	Machine Status Dashboard			
	User Management Dashboard			
		Backend Popups		
		Customer Support Popups		
		Super Admin Popups		
Run Feasibility Check				

Figure 3. Research Related Tasks' Gantt Chart

The first week is to complete the initial designs for the section of the UI being worked on. The team will meet with the client at the halfway point to discuss the designs. The following week will be focused on making any changes requested by the client and starting the design of another UI component. Throughout the design process, we will conduct feasibility checks. This ensures that the UI we are developing will be implementable during the Fall semester.

Finally, we will look at the implementation-related tasks.

Task #	Description	Dependent On
14	Implement Backend Dashboard	5
15	Implement Customer Support Dashboard	6
16	Implement Super Admin Dashboard	7
17	Implement Machine Status Dashboard	8
18	Implement User Management Dashboard	9
19	Design Backend Dashboard Popups	11, 14
20	Design Customer Support Dashboard Popups	12, 15
21	Design Support Admin Dashboard Popups	13, 16
22	Integration and System Testing	14-21

Table 3. A list of tasks for the implementation phase and their dependencies

The implementation phase follows a similar pattern as the design phase. Most of the tasks can be done in parallel; however, overlapping too many tasks would overload our team. Each task follows a four-week-long structure. (See *Figure 4*).

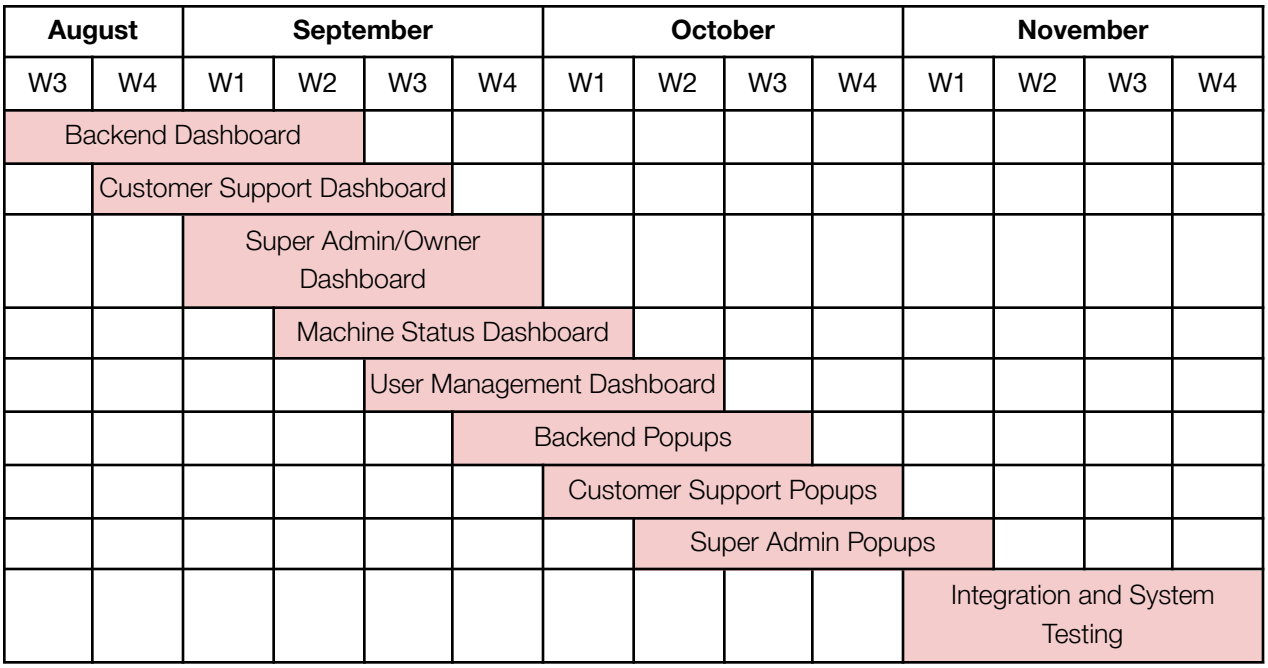


Figure 4. Implementation Related Tasks' Gantt Chart

The first week only contains one task. Since we will be using *GitHub*, the team has one week to set up the project and create the initial dashboard (each dashboard will stem from this one). Then, each following task can be branched off of this commit to allow parallel development. Finally, once the entire website has been implemented, we complete our system and integration testing to ensure proper connectivity and functionality.

3.2. Milestones and Metrics

We have identified seven milestones needed to complete the project successfully.

Research Phase

- Identify 90% of the components that need to be implemented.
- Identify 100% of the technologies needed to complete the project.
- Develop a map of the current implementation that covers 85% of the project.

Design Phase

- Design 100% of the components identified in the research phase.
- Identify 80% of the database relations needed for the dashboard.

Implementation Phase

- Implement 100% of the components identified in the research phase.
- Test 100% of the implemented components as they are integrated into the project.

3.3. Risk and Mitigation

Risk Description	Risk Probability and Severity	Tasks Affected
Failure to identify required components	20% - Low	1 - 4
Failure to get client support for designs	50% - High	5 - 12
Failure to implement code	5% - Medium	14 - 21
Failed database or server connections.	60% - High	14 - 22

Table 4. Risks Severity Table and Affected Tasks

As two of our risks are classified as “High Severity,” we have developed mitigation strategies to decrease the potential impact on the project. To mitigate the risk of our clients not approving our designs, we have regular meetings with our clients to ensure that our designs are approved. We also can communicate with our clients outside of meetings to get feedback on our designs.

For the risk of failing to implement code, we will implement our designs in modular components to decrease the complexity of individual pieces. This will ensure that progress can be made and deadlocks won’t arise.

The final risk is mitigated through multiple things. First, we allow for mock data. By allowing for mock data, we will be able to continue the implementation of code without functioning databases or servers. This removes the dependencies on other teams. At the end of the semester, we also allocated extra time to verify our system integrates with our dependent resources.

3.4. Personnel Effort Requirements

Task #	Hours Req.	Justification and Explanation for hours dedicated
1	4	Our team must look through the code to identify the technologies used. Identifying the leading technologies being used would take an hour of code analysis. However, we will have each number perform this task to ensure everything is noticed and the team comprehends the project.
2	8	Mapping the architecture of the current implementation of the administrative portal requires a comprehensive understanding of the system's front and back end. The backend code is short and will take approximately two hours to understand completely. The front end contains more components and requires a simulation of the code.
3	6	Identifying the components currently implemented will take an hour. However, our team will identify where each UI component is created in the code and how it is created. This will take an additional five hours of work
4	2	Missing components can be identified by looking through code and seeing where past teams put placeholders for future features. The client will also be contacted regarding any remaining features/components that must be designed.
5 - 12	8 Each	The UI components will take four hours for initial designs. Each UI component is expected to follow the aesthetic, and mimicking the aesthetic will take some time. Two hours will go towards other team members reviewing the designs and making recommendations. The final two hours will go towards making any updates the client requests and changes.

Table 5. Task Hour Requirement Breakdown

13	Undefined 16 Hours	A feasibility study will require team members to implement code in the Angular framework. This will take some time since most team members are new to Angular. Each team member is expected to spend four hours on a feasibility study to identify how long UI components will take to develop during development. However, the accurate amount of time can depend on the complexity of the designed UI.
14 - 21	22 Hours Each 6 Hours	Each task will take twenty hours. There will be fifteen hours allocated to the implementation of the software. We estimated Roughly three hours per task for unit testing and two hours for research. The final two hours will be allocated for documentation of the code. This is either commenting on the code or updating the readme. The second four hours are dedicated to three team members spending two hours reviewing the code.
22	8 Hours Each Teammate	The system and implementation testing need to verify all connections between *DigiClips* computers, servers, and databases. Writing the test cases will take around four hours per teammate. An estimated four additional hours are for any complication that might arise with the database or computers. This will create a waiting time in which we will have to work with other teams.

Table 5 Continued

3.5. Resource Requirements

GitHub Repository

We will use a GitHub Repository to store our project files and documentation. We will also use this to keep a record of completed tasks and for code reviews.

Computer Environment

The code we will modify will have to be modified locally before it can be deployed onto a server. Thus, we will have personal machines (Mac, Windows, or Linux) to simulate the code.

Node.js and Angular

Node.js and Angular will compile the code and create the runtime environments.

Amazon Lightsail Cloud Server

For deployment, the website will be hosted on Amazon Lightsail so it can be accessed worldwide.

4. Project Design

Our project is primarily comprised of designs. Our client has asked us to build a web application to monitor and manage their system. Thus, we will primarily be presenting website designs in the following section. This will include UI designs of the user facing site and architectural designs of the code base: frontend code and backend code.

4.1. Design Decisions and Ideation

We have identified the following design decisions by primarily looking at current market solutions and meeting with our client regarding their wants and needs.

Design Decision #1: The administrative portal will be a web application with user-specific dashboard layouts.

This design decision is comprised of two parts: a web application and user-specific dashboards. The choice of developing a web application was at our client's request. A web application can be accessed from any device and doesn't require updating on multiple platforms in multiple (programming) languages.
The reason for choosing dashboards is primarily for information density. A lot of information could be displayed to the user; however, creating a dashboard allows us to select the vital information for each user scenario, i.e., machine status for machine analysts.

A dashboard wasn't the only option considered. The following list is comprised of the options we identified.

- Dashboard website that shows only need-to-know information about the dashboard subject. Information is targeted at the subject. Presented in a minimalist manner, showing only status and statistics.
- Display all website that shows all information on one page. The same information would be displayed as on the dashboard website but it would be consolidated.
- Multiple Websites that chunk information and separate it into websites, i.e., information regarding managing users would be located on one website.
- Excel spreadsheet that queries the database and displays preprogrammed charts and graphs.
- Tableau is an already-implemented service that uses a custom language to make database queries to display statistics in a dashboard format.

The following decision matrix weights the previously mentioned options in the following categories on a scale of [0,5]: Updatability, Usability, Information Density, and Cost. The one exception is that information density is measured inversely. Ideas with a ton of information in one place would make it hard to read.

Idea	Updatability	Usability	Information Density	Cost	Total
Dashboard	2	4	5	5	16
Display All	3	5	1	5	14
Multi Website	1	1	5	4	11
Excel	3	3	2	3	11
Tableau	4	4	3	2	13

Table 6. Decision Matrix for Project Format

The decision matrix led us to creating a dashboard website. The main benefits is that the dashboard website can be hosted locally on machines already purchased and it minimizes the amount of information that a user must consume at once.

Design Decision #2: The administrative portal frontend code will be divided into modules with each module pertaining to one and only one UI element.

This decision targets maintainability or updatability of the UI code for future developers. Breaking the code down into individual modules, I.e. a single graph or cards, allows the code to be easily modifiable. This will make future development and code modifications easier. This design decision will improve the development process as it will allow for parallel development of non dependent modules.

Design Decision #3: The user's selections will be saved into a configuration table stored in the MySQL server.

To reduce the overhead of the user each time they login to the site, their selections will be saved. This means that each UI element that contains a toggle, switch, or typed input will be saved so the next time they log in, the settings will already be applied. Saving it into a configuration file stored in the MySQL server leverages currently used technologies and improves security.

4.2. Proposed Design

The team's proposed design for the *DigiClips* Administration project is to produce a user-friendly interface for the system administrators and super administrators to efficiently oversee various aspects of the whole system. The key components of the proposed design include dashboards for handling error messages, system analytics, machine configuration, user accounts and user permissions. Additionally, there will be a dashboard dedicated for customer support and super administrator functionalities.

Detailed Design and Visual(s)

DigiClips Administration Portal backend architecture is made up of several interconnected components. *Figure 5* shows a block diagram of components that make up the backend for the Administration Portal.The backend has three event handlers: event.js, eventsDashboard.js and auth.js. These three components communicate with a MySql Database through its own module.

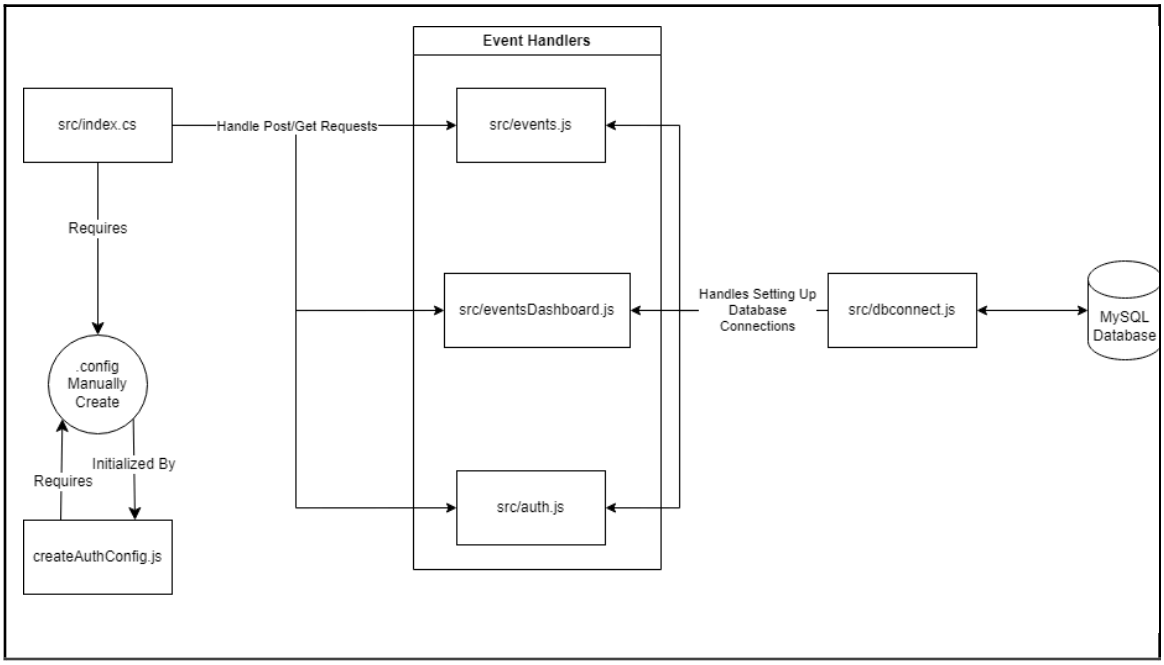


Figure 5. High-Level Block Diagram for Backend

The frontend part of the Administration Portal is the interface for client access communication to the backend through HTML Requests. Figure 6 shows a component diagram of how the whole Administration Portal inter component communication. The diagram shows that the Administration Portal is able to get the necessary information from servers or machines using a communication bridge composed of HTML requests.

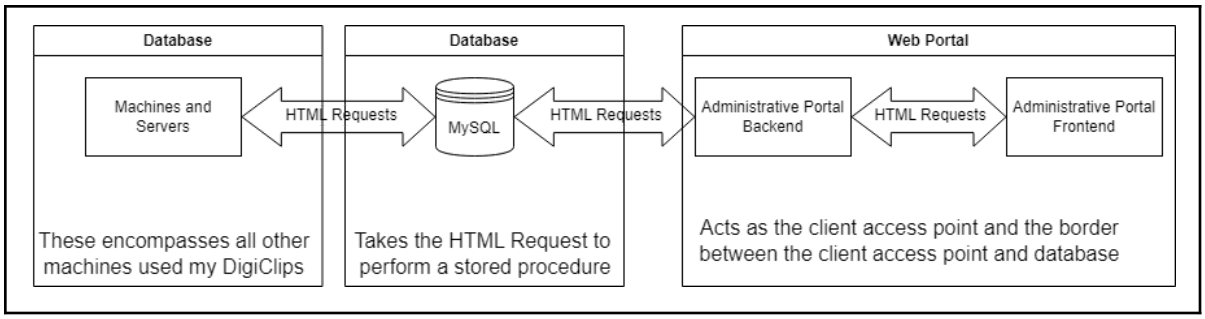


Figure 6. Administration Portal Inter Component Communication

The following is the list of the components that were discussed above.

- Backend Event Handlers
  - Event.js - Main focus is error handling and pinging machines through OS commands.
  - EventsDashboard.js - Queries database for analytical information.
  - Auth.js - Responsible for user's account management.
- MySQL Database Interaction
  - Connect.js - Module for managing connections to the MySql database. Facilitates query execution and result retrieval.
  - Stored Procedures - These are executed in response to an HTML request from the frontend of the Administration Portal.
- Administration Portal Frontend
  - Provides a user interface for client interaction.
  - Communicates with the backend through HTML requests.

Functionality

The administration portal will allow users to access only the relevant dashboard they require based on their permissions. Administrators will be able to view error messages and machine statistics and configure backend settings. Customer support personnel can manage user accounts. Super Administrators have additional privileges to manage permissions and assign administrators to users. The administration portal will respond to users' interactions by updating relevant information and executing necessary backend operations in real time (response time negligible to users).

Areas of Concern and Development

- Ensure there is seamless integration of the subsystems with the proposed changes. Thorough testing of integration points will need to be conducted.

- There are multiple university teams also working within the Administration Portal. Constant interaction between the other teams is necessary to facilitate the proposed design.
- The proposed design needs to adhere to best practices regarding security for frontend applications. The team will need to implement input validation, authentication and authorization as well as preventing SQL vulnerabilities.

4.3. Technology Considerations

The Digiclips project incorporates several distinct technologies to achieve its goals. These technologies have been selected based on their compatibility with the project requirements and standards. Here is an overview of the technologies used, along with their strengths, weaknesses, and trade-offs.

Angular Framework: Angular is being used for the front-end development of the administrative website. It provides a robust framework for building dynamic and responsive web applications. Its use of TypeScript ensures strong typing and better code organization, enhancing maintainability. However, Angular has a steep learning curve, especially for developers new to the framework.

Node.js: Node.js is used for the backend development of the project. It offers a non-blocking, event-driven architecture that is well-suited for handling asynchronous operations, such as handling API requests and database interactions. Node.js has a large ecosystem of packages available through npm (Node Package Manager), which can accelerate development. However, Node.js may not be as performant as other server-side technologies for CPU-intensive tasks.

MySQL Database: MySQL is used as the database management system for storing media information and user data. Its relational database model provides strong data consistency and supports complex queries, making it suitable for applications requiring structured data storage. However, MySQL may require more careful schema design compared to document-oriented databases like MongoDB. Additionally, MySQL's performance and scalability can be optimized through proper indexing and query optimization strategies.

CSS/JavaScript: These core web technologies are used for frontend development, along with Angular. They provide the foundation for building interactive and visually appealing user interfaces. While these technologies are widely supported and easy to use, they may require additional effort to ensure cross-browser compatibility and responsiveness.

4.4. Trade-offs and Design Alternatives

Technology Selection: The choice of Angular for frontend development was made to leverage its component-based architecture and the support for building single-page applications. However, alternatives like React or Vue.js could also have been considered, each with its strengths and weaknesses.

Database Selection: While MySQL was chosen for its relational capabilities and robustness, alternatives like PostgreSQL or MariaDB could have been considered. Each database system has its strengths and weaknesses, and the choice depends on the project's specific requirements, such as data consistency, scalability, and ease of use.

Overall, the technology stack chosen for the Digiclips project aims to balance performance, scalability, and developer productivity, ensuring that the project meets its goals effectively. They also aim to meet the requests of DigiClips effectively.

4.5. Design Analysis

We are running feasibility studies to ensure that designs have the potential to be developed. Due to the design work needed, most of it being web design, we won't have a functioning prototype until next semester. However, running these feasibility tests will ensure we can implement the designs. We must redesign the front end if the feasibility tests present a negative outlook. Our feasibility tests are done through identifying potentially challenging designs and implementing a low level mock to validate the possibility of implementation. This means size, color, size of objects, text, etc., are not done during mocking. Only rough shape placement and size logic will be present.

5. Testing

Testing Strategy and Philosophy

Our testing strategy for the Digiclips project is fundamental to ensuring the functionality, reliability, and security of the system. We believe in early and continuous testing throughout the development process to identify and address issues promptly. Our testing plan is unique to our project, tailored to its specific requirements and design. The philosophy behind our testing approach is to catch and resolve issues as early as possible, minimizing the impact on the overall project timeline and ensuring a high-quality final product.

Challenges to Testing

One unique challenge to testing for our system is the complex integration of multiple media sources and the need to ensure seamless data recording and presentation. Another challenge is ensuring that the system is user-friendly for administrators with varying technical backgrounds. These challenges require a thorough and comprehensive testing approach to ensure that all aspects of the system are functioning correctly and meeting user expectations.

5.1. Unit Testing

The DigiClips Administration Portal is built using the Angular framework, which provides a robust set of tools for unit testing. We will conduct unit tests for each individual component, service, and pipe within the application. These tests will verify the correctness of the component's internal logic, input/output handling, and adherence to the established code standards.

Our team will leverage the Jasmine testing framework and the Karma test runner to write and execute the unit tests. The tests will cover all the core functionalities of the components (See *Figure 7*), such as data handling, event handling, and state management. This will allow us to identify and fix any issues early in the development process, improving the overall code quality.

5.2. Interface Testing

The DigiClips Administration Portal features several key interfaces between the frontend and backend components. We will conduct comprehensive interface testing to ensure the seamless flow of data and interactions between these components.
The interface testing will focus on verifying the correctness of the API contracts, request/response formats, and error handling mechanisms. We will use tools like Postman to simulate API calls, inspect the responses, and validate the expected behavior.

Additionally, we will employ contract testing techniques to ensure that changes in one component do not inadvertently break the functionality of the connected components. This will help us maintain the integrity of the overall system architecture.

5.3. Integration Testing

The DigiClips Administration Portal is composed of several interconnected modules, such as the Error Dashboard, Machine Status Dashboard, and User Management. We will perform integration testing to verify the proper functioning of these modules when they are combined together.

The integration testing will focus on validating the end-to-end workflows, data persistence, and error handling across the different modules. We will create realistic user scenarios and test cases that exercise the interactions between the modules, ensuring that the overall system behaves as expected.

In addition to testing the interactions between the frontend and backend components, we will also verify the integration with the MySQL server. This will involve testing the data flow, including the creation, retrieval, update, and deletion of records in the database. We will ensure that the application's interactions with the MySQL server are functioning correctly and that the data is being persisted and retrieved as expected.
The integration testing will be conducted using a combination of tools, including Cypress for end-to-end testing and custom scripts for data validation and scenario-based testing.

5.4. System Testing

At the system level, we will perform comprehensive testing to verify that the DigiClips Administration Portal meets all the functional and non-functional requirements. The system testing will include the following aspects:

Functional Testing: Verifying that all the required features and user stories are implemented correctly, including error handling, edge cases, and user interactions.

Performance Testing: Evaluating the system's response times, scalability, and resource utilization under various load conditions, using tools like JMeter.

Usability Testing: Evaluating the user interface and user experience, collecting feedback from representative users, and making necessary improvements.

The system testing will be conducted in a staged approach, starting with a development environment, and progressing to a production-like staging environment, to ensure the seamless migration of the application to the live environment.

5.5. Regression Testing

To maintain the integrity of the DigiClips Administration Portal, we will implement a robust regression testing strategy. Whenever new features or bug fixes are introduced, we will run a comprehensive suite of tests to ensure that the existing functionality remains intact.

The regression test suite will include a combination of unit tests, integration tests, and system-level tests. We will leverage the CI/CD (Continuous Integration/Continuous Deployment) pipeline to automatically trigger the regression tests, ensuring that any regressions are identified and addressed early in the development process.

Additionally, we will establish a set of critical user flows and features that will be included in the regression test suite. These will be identified based on the project's requirements and the feedback from the client and end-users.

5.6. Acceptance Testing

The acceptance testing for the DigiClips Administration Portal involves a collaborative process between the development team and the client representatives, Henry Bremers and Bob.
Rather than conducting formal acceptance testing sessions, we have opted for a more iterative approach, where the designs and implemented code are reviewed on a regular basis before being integrated into the main codebase.

Design Acceptance

For the design aspects of the project, we have established bi-weekly meetings with Henry and Bob. During these sessions, we present the latest design proposals, mockups, and prototypes. The client representatives provide valuable feedback, identify any gaps or issues, and approve the designs before we proceed with the implementation.
This iterative design review process has been instrumental in ensuring that the final UI and user experience align with the client's expectations. By incorporating their input early and often, we have been able to make adjustments and refinements to the designs, resulting in a more polished and user-friendly administration portal.

Code Acceptance

For the code implementation, we have implemented a rigorous code review process before merging any changes into the main codebase. Each pull request is reviewed by at least one other team member, who checks the code for correctness, code quality, and adherence to the project's coding standards.

Additionally, before a pull request is approved and merged, Henry Bremers, as the client representative, reviews the changes and provides his final approval. This code review process allows us to catch any issues or deviations from the requirements early, ensuring that the implemented functionality meets the client's expectations.

The combination of the design review meetings and the code review process has been an effective approach for the acceptance testing in the DigiClips Administration Portal project. By involving the client representatives throughout the development lifecycle, we have been able to maintain a close alignment between the delivered system and the client's needs, minimizing the risk of costly rework or late-stage changes.

5.7. Security Testing

Security testing will be conducted to identify and mitigate potential vulnerabilities in the system. Tools like OWASP ZAP will be used with the clients approval to perform security scans to check for SQLi and other common web application vulnerabilities. Testing will also be done to ensure that confidential information is not leaked through the frontend web page. Security testing is essential for ensuring that the system is secure and protected against potential threats and attacks

5.8. Result

At this point, we do not have any test results to report.

6. Implementation

Our team will follow the plan laid out by the Gantt charts referenced in *Figure 4* and *Appendix B*. We will maintain our weekly meetings to ensure that we adapt our plan weekly to meet the times specified in the Gantt chart. Our team will break down each UI to be implemented into smaller components (See *Figure 7*) and backend components. These tasks will be delegated to our team members.

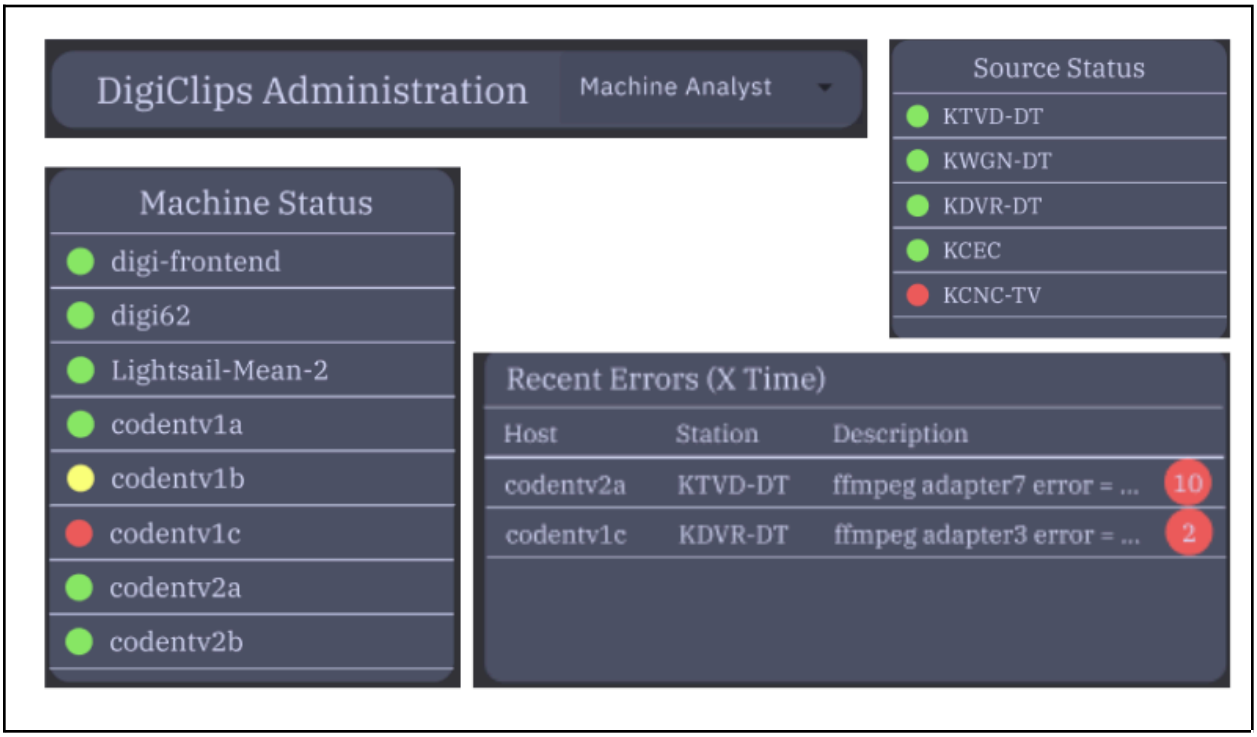


Figure 7. The UI decomposition of the administrative portal

Each team member will concurrently write unit tests to verify their work. Before each code review, the updated code will be run against all previously implemented unit tests. When merging into our team's branch, all tests must be passed. Before merging into the team's branch, a code review must be conducted with each team member as a reviewer.

7. Professional Responsibility

This discussion is with respect to the paper titled “Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, International Journal of Engineering Education Vol. 28, No. 2, pp. 416–424, 2012

7.1. Areas of Responsibility

Area of Responsibility	Definition	NSPE Canon	IEEE Code of Ethics	NSPE vs IEEE Differences
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	(6) To undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations (3) To be honest and realistic.	While the NSPE focuses primarily on only doing work within ones area of competence, the IEEE code allows for work outside of their competence as long as there are proper disclosures.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	(7) To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;	The IEEE code does not specifically cover the how one should act for clients, although it does mention to correct errors and accept criticism.

Table 6. Area of Responsibility Chart

Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	(3) To be honest and realistic in stating claims or estimates based on available data.	The two codes are similar on this value. They both state to be honest and realistic without deceit.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	(1) To accept responsibility in making decisions consistent with the safety, health, and welfare of the public. (9) To avoid injuring others, their property, reputation, or employment by false or malicious action.	While the codes are similar on this value, the NSPE holds safety paramount while the IEEE code is more focused on accepting the responsibility to make a safe product and avoid injuring others.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	(7) To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors.	While the IEEE code does not specifically mention how to act for clients, it does focus on accepting criticism and correcting errors.

Table 6 Continued

Sustainability	Protect environment and natural resources locally and globally.	Adhere to sustainable principles to protect the environment.	(1) To disclose promptly factors that might endanger the public or the environment.	The NSPE code focuses on adhering to sustainable practices while the IEEE code focuses on reporting environmental dangers.
Social Responsibility	Produce products and services that benefit society and communities.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	(5) To improve the understanding of technology; its appropriate application, and potential consequences.	While both codes talk about furthering the profession, the NSPE code focuses more on the personal conduct as opposed to improving understanding with the IEEE code.

Table 6 Continued

7.2. Project Specific Professional Responsibility Areas

Work Competence - High: Work Competence plays a large role in our project as the team is developing a piece of software that will be in use within DigiClips. We are performing highly in this area because we are producing our project within our deadlines and to the highest quality with the knowledge gained from our coursework.

Financial Responsibility - High: The area of Financial Responsibility applies partially to the project, while the team is striving to create a high quality product with a realizable value, the costs of the project are not applicable as we are using non-commercial software and internal tools. Our team has been able to design a high-quality product with zero financial cost to the client.

Communication Honesty - High: Communication Honesty is a high priority for our team. Working closely with the clients ensures that we are creating a product that will suite their needs and be useful to their company. Our team conduct weekly meetings with our clients in order to review designs and give general status updates on the project.

Health, Safety, Well-Being - NA: As our team's project is entirely software based, Health, Safety, and Well-Being are not overly relevant. There are no significant Health, Safety, or Well-Being risks or concerns with our project.

Property Ownership - Medium: Property Ownership is relevant to our project and plays an important role with our team stores information regarding the clients of our application. Our team is working to ensure that our designs will properly handle any client information contained within the application.

Sustainability - NA: Due to our project being entirely software based, the Sustainability value is not relevant to our project. Our project does not have a positive or negative impact on the environment or natural resources.

Social Responsibility - NA: Similarly, since our project is to be used as an internal tool, the Social Responsibility value is also not relevant to our project. Our project does not have a positive or negative impact on society or communities.

7.3. Most Applicable Professional Responsibility Area

Communication Honesty is the most applicable area to our project. Our team is focused on maintaining consistent communication with the clients to ensure a project is developed that suits their specifications. In addition to client communications, our team aims to develop and maintain proper code documentation throughout the implementation phase of the project. This will ensure that future developers are able to understand the code and work with it. Finally, easy to understand instructions will be created in order to explain the proper uses and features of the application.

8. Closing Material

8.1. Discussion

Our design for the Administration Portal project addresses the multiple requirements provided by the client. With detailed conversations with the client, we identified key functionalities, such as server status monitoring, user account management, and configurable privileges for different roles, which were incorporated into the proposed design.

While the client had predetermined the use of Angular, Node.js and MySql in the Administration Portal, we were able to produce a design that will be able to take full advantage of the capabilities that these technologies provide.

8.2. Conclusion

Our primary objective throughout the project has been to design an efficient and user-friendly internal tool tailored to Digiclips' managers, customer support representatives, media recording engineers, and software development engineers. The requirements for this internal tool were gathered directly from the owner of DigiClips and the lead software engineer. After the requirements were gathered, the team started analyzing the frontend and backend code to identify the key components and locations where both interfaces meet. Once we had all the important information, the team was able to start creating preliminary designs. The designs were then presented to the client where a specific aesthetic was selected. All designs now resembled the selected aesthetic.

Going forward into the implementation phase, the team will switch from a waterfall mentality to agile implementation. We are choosing an agile development methodology since most of the frontend components can be implemented in parallel. The proposed changes will be created into user stories that will be broken down into manageable tasks. Following agile traditions, the team will present the new changes to the client during our weekly meetings and also request feedback.

The team's testing strategy will also be a vital action during the implementation phase. By performing unit testing, interface testing, integration testing, and regression testing, the team aims to validate our proposed design's performance and reliability. Through constant client communication, adhering to the IEEE standard and coding best practices, and having a reliable testing suite, the team will achieve its goal of delivering the Administration Portal internal tool to DigiClips.

8.3. References

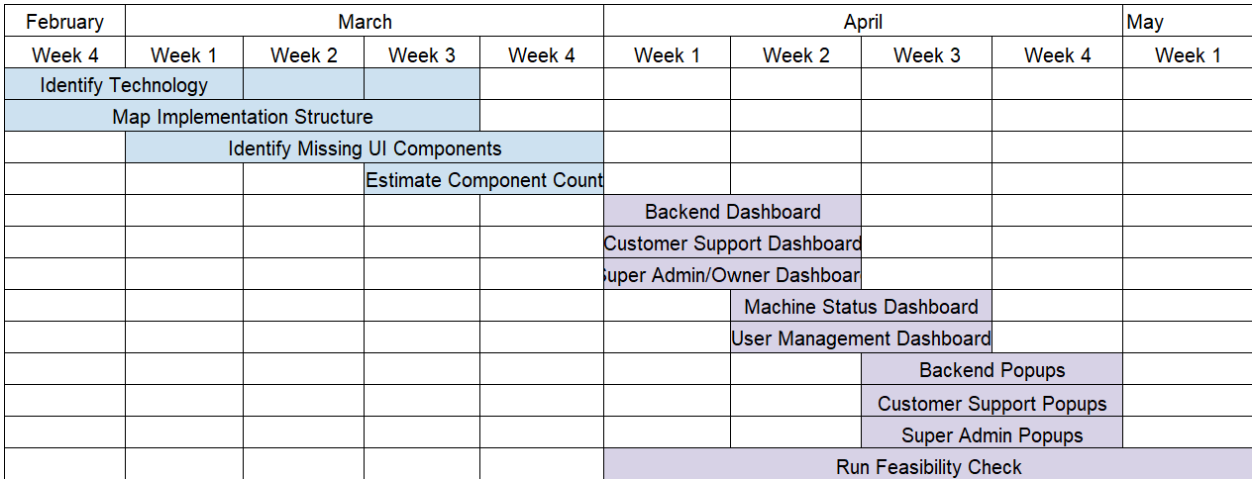
[1] IEEE Standards Association “IEEE SA - IEEE/ISO/IEC P23026: Systems and Software Engineering -- Engineering and Management of Websites for Systems, Software, and Services Information” [Online]. Available: https://standards.ieee.org/ieee/23026/10425/ [Accessed Apr. 16, 2024]

[2] Angular “Angular coding style guide” [Online]. Available: https://angular.io/guide/styleguide [Accessed Apr. 16, 2024]

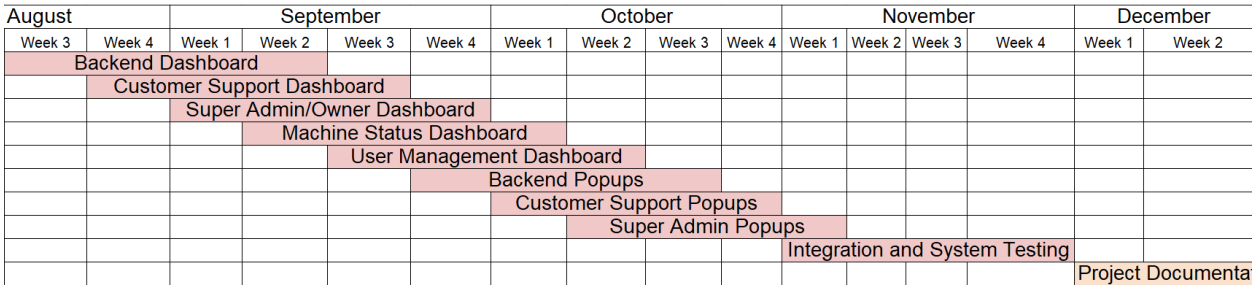
[3] Google “Google TypeScript Style Guide,” [Online]. Available: https://google.github.io/styleguide/tsguide.html [Accessed Apr. 16, 2024]

8.4. Appendices

Appendix A - Gantt Chart Senior Design



Appendix B - Gantt Chart Senior Design Implementation



9. Team

9.1. Team Members

Derek Brandt
Tyler Orman
Aryan Rao
Israel Sanchez-Tellez

9.2. Required Skills for Project

The frontend interface of the Administration Portal is written using Angular and Typescript.
The backend uses the technologies Express and MySql to perform tasks.
The overall goal of DigiClips is to be able to host this project on AWS Lightsail.

9.3. Skill Sets Covered by Team

For the front end, Aryan has experience using the Angular framework.
Also, Israel has used Typescript in a previous Co-op for the front end.
The whole team has experience with MySql.
Aryan, Derek, and Tyler have brief experience with Express.
The part where the whole team lacks experience is with AWS Lightsail.

9.4. Project Management Style Adopted by Team

Overall, we use a hybrid approach to project management. For design, we chose the Waterfall style. This is because the project requirements must be understood before any development or design can be completed. For the implementation phase, we will switch to an agile style where the changes will be made into user stories that can be broken down into manageable tasks.

9.5. Initial Project Management Roles

Derek Brandt - Architecture Lead
Tyler Orman - Security Lead
Aryan Rao - Programming Lead
Israel Sanchez Tellez - Client Interaction Lead

9.6. Team Contract

Team Members:

1) Tyler Orman
2) Derek Brandt
3) Israel Sanchez-Tellez
4) Aryan Rao

Team Procedures

1. Meeting Time and Location:

This team will meet virtually every week on Tuesday at 5 pm unless otherwise agreed upon by the ground members.

2. Team Communication:

Discord will be the primary communication means via chat or voice call. Deadlines and scheduling will be done in the respective Discord channel.

3. Decision-Making Policy:

The primary decision-making policy will fall to a decision matrix. These types of decisions will be brought up during the weekly team meeting. Before the meeting, the individual is responsible for creating the categories of the decision matrix. The team will fill out the decision matrix to reach a group consensus. A majority rule can override, change, or remove any category in the decision matrix during the group meeting. Due to the length of the process, this method should be reserved for software architecture changes, team structure changes, and software structure changes.

Decisions on software implementation can be brought up during team meetings; however, they should be reserved for code reviews. Between two code reviewers and the submitter, a consensus on implementation should be reached by the majority rule. It should be escalated to a team meeting decision matrix if it cannot.

If any team member has a decision that does not fall under any previous category, it should be brought up to the team during team meetings. If the decision requires a decision matrix due to complexity, the person responsible must submit categories for analysis.

4. Record-Keeping Procedure:

At the beginning of each meeting, a group member will be chosen to write meeting notes. These meeting notes will be sent to the respective Discord channel at the end of each meeting. The responsibility of the recording will fall under a round-robin schedule, in the following order: Derek, Israel, Tyler, and Aryan. The schedule will move to the next person if

a member misses a week. The person who missed will be required to record keep following week.

Participation Expectations

1. Meeting Expectations:

Each team member must attend all weekly meetings with the team and the client. During this, future deadlines will be discussed and agreed upon. If a member has hit a roadblock in, it is expected that they bring it up during the weekly team meeting. If a member has to miss a meeting for any reason, they likely inform the team; however, they only need to include why if repeatedly missing meetings.

2. Assignment Expectations:

Team members are expected to complete their assigned weekly tasks by the next meeting. If a team member is unable to complete their weekly task, the rest of the team should be promptly notified so that plans can be made to complete the task.

3. Response Expectations:

It is expected that each team member responds within 24 hours of a given message. Also, to ensure that communications are seen, it is expected that each member performs a daily Discord check.

4. Weekly Hours Expectations:

There is no maximum or minimum amount of time for each member to put into the project. However, it is expected that a team member finishes their weekly assigned tasks. If the weekly assigned tasks exceed 12 hours of total work, the member will not required to finish them. During the next team meeting, they should mention this excessive work so future weekly loads can be reduced.

Leadership

1. Team Roles:

- Israel - Client Interaction Lead
- Aryan - Programming Lead
- Derek - Architecture Lead
- Tyler - Security Lead

2. Support and Guiding Strategies:

The primary method to guide team members will be the weekly to-do list. This list will contain all tasks that need to be done and those that will be completed during the week. Researching technology is a valid to-do item. This will encourage team members to

explore new technologies and obtain more efficient and elegant results before implementing solutions. Each team member is expected to be adaptable and flexible. This means that team members are expected to help each other when hitting roadblocks, even if it means sacrificing meeting deadlines considered a lower priority. Team members are encouraged to participate in an open dialog during team meetings and throughout the week about roadblocks. Finally, all code will be subject to code reviews to allow members to learn about others' work and provide input.

3. Recognizing Contributions:

Each team member will report their accomplished tasks during the weekly meeting. This will be noted in the to-do as a visual representation of the work completed. Secondly, accomplishments will expected to be notarized in each progress report submitted.

Collaboration and Inclusion

1. Team Skills and Expertise:

- I (Israel) completed a Software Developer Co-op last year, where I worked on frontend code and the installer for the application. The frontend code consisted of Javascript and React components. I used the React Testing Library to test the frontend code. For the installer, I injected a dependency check using Wix# (WixSharp) for certain .NET Runtimes needed to run the application. This semester, I'm also doing a part-time co-op with the same company to continue gaining real-world experience.
- I (Derek) am a Software Engineer. I have experience writing embedded software in C++ and using Gmock, a testing framework that I have developed working at *John Deere*. This is my current job as a part-time student there. I have experience using Python machine-learning libraries such as Pytorch and Tensorflow in conjunction with GymAPI. I have experience programming games and game AI in Unity using C#. Finally, I have experience with backend and frontend development for web apps and Android apps that I have obtained through Iowa State courses.
- I (Tyler Orman) am a Cybersecurity Engineer. I have experience with networking as well as various programming languages, including C, Python, Java, and a bit of Javascript, HTML, and PHP. Through courses here at Iowa State, I have gained experience with Backend development and have also built various applications for different jobs. I currently work part-time for Iowa State as an intern with Hunter Strategy.
- I (Aryan) bring some hands-on experience in software development to the team. With a strong background in coding and problem-solving, I have honed my skills through various internships and work experience. This experience not only provided exposure to diverse technological landscapes but also fostered adaptability and a keen understanding of industry best practices. We, as a team, will be ensuring that we stay at the forefront of cutting-edge software development.

2. Team Encouragement and Support Strategies:
The primary strategy the team will use for encouraging and supporting contributions and ideas will be to have open communications. This means that as a team, we will hold meetings where any idea or perspective is valuable and will be heard by all. The team has a Discord server where chats and virtual meetings will occur.

Since DigiClips uses its own GitHub repository, the team will conduct code reviews. By doing code reviews, not only will the team ensure that minimal amounts of bugs reach production, but it will also allow team members to provide feedback to other team members.

Another strategy that will not be enforced but is highly encouraged is for team members to conduct pair programming sessions. These sessions include two or more team members working on the same task simultaneously. Pair-programming sessions create the opportunity for knowledge sharing and ensure everyone is actively involved.

3. Collaboration and inclusion Strategies:
All team concerns will be addressed with respect. These concerns should be raised during weekly meetings or mentioned in the Discord. We will implement an Open-door Policy to encourage each team member to vocalize concerns immediately. To ensure team members are collaborating, the team will use the weekly to-do list as primary evidence of the contributions of each team member. Any concerns regarding under or over-contributing should be addressed during the weekly team members.

Goal-Setting, Planning, and Execution

1. Team Goals:
The most fundamental goal of the team is to complete the course. However, the primary goal is to work with the given team to create a successful deliverable for Digiclips. Other objects include gaining experience handling functional and non-functional requirements, working with a client to create a deliverable and experience learning unknown technologies in short periods.

2. Assigning Team and Individual Work:
During each weekly team meeting, the team will assess what the client's needs are for the upcoming and which tasks to prioritize. All tasks will be recorded in the to-do list. At the end of each weekly meeting, tasks will be divided up amongst team members and will be recorded as in progress until otherwise completed.

3. Keeping on Task:
During weekly meetings, we will utilize a chime when the team gets off task. If a team member is absent at a meeting, it is their responsibility to read the meeting notes and check the to-do list to see what needs to be done. Since tasks will be assigned individually, it will be the individual's responsibility to keep themselves on task during work hours and adhere to prior expectations on week completion.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?
Infractions of the contract will be addressed directly with the person responsible. If the person has a reason for a breach of contract, I.e., family or personal issues, then the breach of contract will be understood. However, a nonvalid breach of contract will be addressed during the team's weekly meetings.

2. What will your team do if the infractions continue?
If the infractions continue, the infractions will be escalated to the TA, then our academic advisor, and finally the professors of the class.

Team Member Signatures

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.
b) I understand that I am obligated to abide by these terms and conditions.
c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

Signatures	Date Signed
1) Tyler Orman	01/30/2024
2) Derek Brandt	01/30/2024
3) Israel Sanchez-Tellez	01/29/2024
4) Aryan Rao	01/29/2024