

SCORE Milestone 1 Project Evaluation

Team Members:

Charlie Collins, ccollins2021@my.fit.edu
 Michael Komar, mkomar2021@my.fit.edu
 Logan Klapproth, lklapproth2021@my.fit.edu
 Tommy Gingerelli, tgingerelli2021@my.fit.edu

Faculty advisor/client:

- Dr. Mohan - rmohan@fit.edu

Milestone 1 Progress

Task	Completion	Charlie	Logan	Michael	Tommy	To Do
Compare and select technical tools	100%	25%	25%	25%	25%	N/A
Select collaboration tools	100%	25%	25%	25%	25%	N/A
Demos	100%	25%	25%	25%	25%	N/A
Resolve Technical Challenges	80%	25%	15%	40%	20%	Awaiting further conversation with IT over TRACKS CAS
Requirements	90%	30%	20%	20%	30%	Need to make requirements on containers
Design Document	100%	20%	20%	20%	40%	N/A
Test Document	100%	50%	20%	15%	15%	N/A

Discussion of accomplished tasks:

Task 1 (Technical Tools): The technical tools that we needed to select were for server implementation, file transfer, web development stack, and user authentication. For file transfer we were deciding between Secure File Transfer Protocol (SFTP) and Secure Copy Protocol (SCP), and ultimately decided on SFTP. While SCP is slightly faster, SFTP is more robust and allows for file transfer into and from the client. The web development stack that we chose is MongoDB, Express.js, React, Node.js (MERN). We felt this stack will provide us with the tools to make a modern web application. For user authentication, we have reached out to IT regarding access to the TRACKS API, but we have prepared for the possible use of Google authentication in the event we can't use TRACKS.

Task 2 (Select Collaboration Tools): For creating documentation, we have decided on using Google docs as it allows for concurrent editing. For our version control, we have decided to use GitHub. The editor we have chosen is Visual Studio Code as it is lightweight and allows all processes that we need.

Task 3 (Demos): For file transfer, we made a small SFTP demo in python, in which we were able to transfer a file from a local machine to a raspberry pi. Utilizing Tailscale VPN and a centralized home server running an Ubuntu virtual machine under a ProxMox cluster we were able to develop an environment that demonstrates the act of using ssh to access code01, which will eventually be used to run the client. We were able to get a simple demo of React.js up and running and an example database using MongoDB to display a MERN stack example.

Task 4 (Technical Challenges): For this milestone we had three main technical challenges. The first challenge was using the Canvas API. To resolve this challenge, we read the api documentation and consulted several other related sources. From this search we discovered that Canvas implements GraphQL, providing a GUI for us to create our JSON requests, and found the exact API endpoint to upload grades to Canvas. Using TRACKS CAS for the user authentication will require further development in the ongoing conversation between our team and the IT department to discern the feasibility of, or the benefits of using CAS over OAuth2. Once this milestone has been completed the conversation will continue by sharing the Design, Test, and Software Requirements documents with the IT department. Our decision to utilize containerization has possibly brought a new technical challenge to add to the list, as properly configuring this requirement will be difficult in production.

Task 5 (Requirements): Our requirement specification document outlines both functional and interface requirements. The functional requirements specify exactly what functionality each feature should provide. Included with our functional requirements are sample input and output to describe the expected behavior of our application. The interface requirements define the interface, through which the users will interact with the application.

Task 6 (Design Document): The design document for this project includes a UML diagram, an ER diagram, as well as Mockups of the web app interface. The UML diagram describes the classes as well as how they interact with each other. The Entity Relationship (ER) model provides a diagram of how the database for the project will be set up. Finally the Mockups were created as Google drawings and are a visual description of the interface.

Task 7 (Test Document): The test document that we created for this project details the test cases to validate that each requirement has been met. The test cases describe a procedure to follow within the application and end with the expected output from that procedure.

Discussion of member contribution:

Charlie: My contribution to this milestone was to first select the technical tool for file transfer. I then created a simple file transfer demo that transferred a text document between my PC and my raspberry pi. Then I worked on the Canvas API challenge, finding the endpoint that we need as well as how to access the GraphQL for Canvas to assist in writing the JSON requests. For the requirements document, I wrote the sections on assignment creation, assignment submission, assignment deletion, and MOSS integration. I then created the mockups for the design document, and finished the milestone by working on the testing document.

Michael: My primary contribution to this milestone was to begin solving the technical challenges that we initially realized would halt progress. As of right now, the long-term storage for submission files has not been decided upon. User authentication is currently an ongoing conversation between the IT and Security department, Dr. Mohan, and our team in order to consider the cost and benefits of using CAS over the Oauth2 protocol. Other contributions that I made towards this milestone includes setting up the environment for developing the server and client applications on a centralized server hosted from my home and accessed through Tailscale VPN. This interaction between remote PC and the server is considered a demo for the purposes of documentation. In terms of the design document, my main contribution was the Entity Relationship diagram, and for the requirements document I outlined the specifics of the client-server connection and the functional and interface requirements associated.

Tommy: My biggest contribution to this milestone was creating the UML diagram. This included defining the classes and relationships between them. I also created the hello world demos for user authentication and containerization. As we progressed through this milestone, containerization became an increasingly important concern for us, so I also created a technical demo of containerization. Finally, I wrote a portion of the requirements including auto grading, immediate feedback, grading portal, and user authentication.

Logan: My primary contribution to this milestone involved re-familiarizing myself with the tools and software needed to develop a MERN stack website. I collaborated with the other members on discussions involving the layout of the website interface and how we would maintain the integrity of the server and safeguard against malicious code being submitted. I spent time coding in React.js and MongoDB to not only create a demo of the full stack working but also to prepare for the full stack website that the user will interact with.

Task Matrix for Milestone 2

Task	Charlie	Logan	Michael	Tommy
Implement the Shell Application	20%	15%	50%	15%
Implement Assignment Creation	15%	35%	15%	35%
Implement Assignment Submission	40%	20%	20%	20%

Discussion of Milestone 2 Tasks

Task 1: Our first task of milestone 2 is to implement the shell application. We need a user interface to start implementing the other features on top of, and we felt the shell application would be the easier of the two interfaces to create first.

Task 2: Our second task of milestone 2 is to implement the assignment creation feature. Once we have completed the first task, assignment creation is the logical next step. For this milestone, assignment creation will include creating a new assignment and giving it a name, description, due date, and number of attempts. Test cases and auto testing configuration will be implemented in the next milestone.

Task 3: Our third and final task for this milestone is to implement the assignment submission feature. Once we have assignment creation implemented, this was the next logical feature to implement. For this milestone, assignment submission will include uploading a file into the application, receiving confirmation that the assignment has been submitted, and being able to view the submitted file. This will happen locally, however, as file transfer will not be implemented until the next milestone.

Dates of meetings with the client/advisor:

9/17/2024 at 11 am

9/24/2024 at 11 am

Client/Advisor feedback

Task 1 (Technical Tools): “The tools that you all have selected seem reasonable. However, I would ask that you document why those tools were selected. For example, include pros and cons to the alternatives, and why the advantages outweigh the disadvantages in this instance.”

Task 2 (Select Collaboration Tools): No feedback needed

Task 3 (Demos): “The demos look good, they demonstrate that you can use the tools.”

Task 4 (Technical Challenges): “The resolution of your challenges look good.”

Task 5 (Requirements):

Auto testing: Auto testing should also be able to use a professor submitted verifier for the solution of an assignment. Not every assignment will give an output that can just be checked using diff. Some assignments, like Towers of Hanoi, for example, could have multiple solutions, so the professor might instead want to upload code that can verify if the solution is correct or not.

Test cases: Consider implementing weighted test cases. Some test cases might be harder to account for, or cover more of the requirements, so each test case should have an adjustable weight. Visually, the students will see the test case as either passed failed, the numeric score would be affected by the weight.

MOSS: Professors should be allowed to submit files to the MOSS algorithm. This might be to include starter code given to the student that should be ignored, or include certain answers from the internet to check against. Additionally, you should consider adding some sort of visualization to the MOSS report. Since this would essentially be a graph of pairwise similarities, it might be difficult to read, so a graphical representation could be helpful. Finally, see if you could add clustering of pairwise similarities to detect if groups of students worked together on an assignment.

Shell Client Interface: Make PDF files be a link that can be opened in a browser or PDF viewer.

Task 6 (Design Document):

Mockups: Make the visualization of detected similarities be more obvious. Perhaps change the entire row of the flagged submission to be red instead of just the name of the student.

Task 7 (Test Document): No feedback given.

Faculty Advisor Signature: _____ **Date:** _____

Evaluation by Faculty Advisor

- **Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu**
- **Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)**

Charlie Collins	0	1	2	3	4	5	5.5	6	6.6	7	7.5	8	8.5	9	9.5	10
Tommy Gingerelli	0	1	2	3	4	5	5.5	6	6.6	7	7.5	8	8.5	9	9.5	10
Michael Komar	0	1	2	3	4	5	5.5	6	6.6	7	7.5	8	8.5	9	9.5	10
Logan Klapproth	0	1	2	3	4	5	5.5	6	6.6	7	7.5	8	8.5	9	9.5	10