

EE/CprE/SE 492 Bi-WEEKLY REPORT 4

Mar 23, 2023 – Apr 6, 2023

Group number: 40

Project title: *Interactive learning platform to engage large size classroom*

Client &/Advisor: *Md Maruf Ahamed (Client and Advisor)*

Team Members/Role:

- ***Tyler Miller: Backend***
- ***Alex Swenson: Backend***
- ***Jaden Ciesielski: Frontend***
- ***Brandon Burt: Frontend***
- ***Adam Walters: Frontend***
- ***Guan Lin: Frontend***

○ **Bi-Weekly Summary**

We have integrated PATCH endpoints on the backend, as well as API calls on the frontend to support the editing of messages and poll fields, as well as the deletion of messages by the professors. Documentation was added to all of our API functions. We added a readme and documented how to use our web socket server. We set a series of goals on the team with our client of what needs to get finished to be a complete application, and have been checking off the list of tasks over the past couple of weeks to prepare for our live demo to Maruf's classroom where all features will be available for use. Some good security measures were also added to the backend.

○ **Past 2 week accomplishments**

● **Backend Documentation (Tyler)**

- Used jsDoc to document all of our api functions so that parameters and return types are documented for all functions
- Documented how to use our websocket server and created a readme
 - Each payload type is specified
 - How to connect

- **Pipeline Correctly Stops on Error (Tyler)**
 - Pipeline will correctly stop and not deploy if an error occurs
 - Test failure
 - Pull failure
 - Installing
 - Building
- **Websocket Attachments (Tyler)**
 - Added ability for messages with attachments to be sent over our websocket connection
- **Various API enhancements (Tyler)**
 - Duplicate users cannot be created
 - Added extra fields to messages like messageType and fixed the lazy loading feature so the most recent 50 messages are grabbed
- **Security Enhancements (Tyler)**
 - Passwords are hashed before being stored so that those with DB access still cannot see users passwords
 - Updated all corresponding APIs and updated all existing users passwords
 - Permission checks added to all relevant apis
 - E.g you cannot access course content for a course you are not in, only course owners can give roles to others
- **Backend Logging (Tyler & Alex)**
 - Switched all logging on the server to use our custom tool and log files
 - Logs can now be viewed on the live server via ssh into the server
 - general.log
 - error.log
 - Debug.log
- **Pre Deploy Checks (Tyler)**
 - Added a linting check to pipeline to ensure code formatting
 - Added a 90 percent code coverage requirement on backend
 - If either check fails the pipeline will not deploy the code
- **PATCH Endpoints (Alex)**
 - PATCH endpoints and security validation for poll prompts, poll choices, poll responses
- **Media download Enhancements (Alex)**
 - Stored file type is now returned in header of GET request to download media
- **Created .csv download feature for poll results (Guan)**

- Allows professors or TA roles be able to download a .csv file of the results of a poll
- The .csv file holds headers for netid, first name, last name, and whether they got the question correct
- **Added unit tests (Guan)**
 - Finished unit tests for edit message, edit user, and patch service
- **Creating multiple API calls on frontend (Brandon)**
 - Creating API calls to new PATCH endpoints that were made on backend for editing messages, deleting messages, editing poll prompts, editing poll choices, and editing poll responses
- **Integrating editing/deleting feature into Poll and Message files (Brandon)**
 - Integrating editing, and deleting into Poll/Message component files by passing API params in as props, and then using statehooks to return the updated results of each action.
- **Adding user permissions for new features on frontend (Brandon)**
 - Adding elevatedUser permissions to some of the new features such that a message can only be edited by the user who created it, and only users in the professor role can delete messages, and edit poll params.
- **Adding hover feature to edit or delete messages and polls (Brandon)**
 - Adding hover feature to Poll and Message components using statehooks within the rendering of the UI. When a user with the proper permissions hovers over a poll or message the button to edit/delete will appear available to select.
- **Supported “Lazy” Course Addition (Adam)**
 - Allowed for frontend objects to add courses after a user has already signed in (e.g. after registration)
- **Created Generic Pop-up Component (Adam)**
 - Created a component that displays its children in a pop-up that smoothly moves into view and blurs the background to emphasize its content
 - Parent of the popup has control over when the popup appears and when it closes
- **Created Pop-up Form Component (Adam)**
 - Using the pop-up component, I created another generic pop-up that allows for input forms
 - Pop-up is entirely self-generated based on a list of desired inputs and input types to better enforce regular styling
- **Integrated Pop-up Component into Course View (Adam)**

- Changed the existing rigid pop-up to use the new component, simplifying the course view page's logic
- **Implemented Raise Hand Feature on Frontend (Adam)**
 - Used the backend's live "question" system to add the ability for users to virtually "raise their hand"
 - Questions are automatically streamed to the professor and pinned to his screen
 - Questions are saved locally in the professor's browser so he can get back to them later if need be
- **Updated Poll API to Support PollType (Jaden)**
 - Added the course id and lecture id for additional params
 - Added the api call to take pollType
- **Added the Add Course Pop Up (Jaden)**
 - Created the api call on the frontend to add a person to a course
 - Connected the call to the frontend, supplying correct params
- **Updated Components (Jaden)**
 - Updated the profile icon to support a filled view when true
 - Updated the Live Lecture Title to be centered
 - Updated the current Poll PopUp to supply the correct type
- **Pending issues**
 - None
- **Individual contributions**

<u>NAME</u>	<u>Individual Contributions</u> <i>(Quick list of contributions. This should be short.)</i>	<u>Hours these Past 2 weeks</u>	<u>HOURS cumulative</u>
Tyler Miller	Documented function level code for all of the apis on the backend. Added various security enhancements like hashing passwords and permission checks on apis. Added ability to send attachments via websocket, and to send message and poll updates. Various pipeline enhancements to help prevent breaking code from entering the live environment. Switched the entire backend to use our custom logging tool.	13	77

Brandon Burt	Created multiple API calls on frontend, integrating edit/delete features into Polls and Messages, integrating hovering feature, integrating user permissions for editing/deletions to assure nothing bad can happen when using these abilities.	18	68
Adam Walters	Supported lazy course addition, added generic pop-up component, added generic pop-up form using inputs, integrated pop-up in course view page, and implemented the raise hand feature. Additionally helped out some teammates work through issues.	12	66
Alex Swenson	PATCH endpoints, security, server-side logging, media download enhancement	12	62
Jaden Ciesielski	Finished the add course popup, updated the profile icon to support multiple views, updated multiple apis to match with backend.	10	64
Guan Lin	Created .csv download feature for professors and TA's for poll results, and added unit tests for edit message, edit user, and patch service	16	51

o Plans for the upcoming weeks

- Frontend:
 - Finish up final tasks needed to complete base goals of our project, including:
 - Filter live lecture messages
 - Update polls to support multiple-choice
 - Update polls to auto-close
 - Implement more detailed lecture info on course view page
 - Hover over messages to edit/delete (*nearly done*)

- Download participation as .CSV (*nearly done*)
 - Polish up the UI for various parts of the application to clean up its aesthetic.
- Backend:
 - Finish up PATCH endpoints
 - Start storing filenames
 - Scan backend for any potential security issues
 - Make changes to APIs based on Front-end discoveries as they work
 - Update integration tests to include the changes made in the past few weeks
- Team:
 - Load test first completed version of application with one of Marufs classrooms.
 - Work out any discovered issues if any arise during our demo with the classroom.
- **Summary of bi-weekly advisor meeting**
 - We showed off the ability to upload attachments
 - Locked down to certain file types
 - Downloadable
 - Need to support (txt, mp4)
 - Poll participation will need downloadable csv
 - Discussed when we will be going to one of his lectures to have his class use our application
 - April 19th
 - Discussed the backend enhancements made
 - Mostly Testing, logging, and security work
 - Front end join code functionality will be available soon
 - Discussed final requirements left needed to complete original goals set and created our final sprint of tasks.