

Virtual Tutor Room

Project Report

The Brogrammers



Kenya Holland, Tatum Hackler, Rahjee Manuel, Jared Campisi

Capstone Project Spring 2022

The University of West Florida

April 29, 2022

CIS4592 Capstone Project

Dr. Bernd Owsnicki Klewe

Table of Contents

1 Executive Summary	3
2 Requirements	3
2.1 Initial Requirements	3
2.2 Final Requirements	3
2.3 Initial vs Final Requirements Comparison	4
3 Timeline	4
4 Project Results	5
5 Software Evaluation	5
6 Work to be Done	6

1 Executive Summary

Over the course of this semester we developed a Java desktop application that provides a virtual tutoring environment for users, with the goal of solving current online learning challenges. By improving students' accessibility to tutors and the ability to collaborate with other students and communicate effectively, we hoped to improve the overall quality of the academic experience comparable to that of the pre-pandemic time. This virtual tutoring environment has a directory of rooms. This directory lists the tutor and student names that are inside of a room. Inside of a room, there is a text chat that the tutors and students can use to communicate with each other. A zoom meeting is active in each room where the tutor can make breakout rooms for the students to work together.

2 Requirements

2.1 Initial Requirements

Our initial project requirements included the following user stories:

- 1) As a user, I can create an account and login with my credentials.
- 2) As a user, I want to be able to see a list of rooms in a directory and select an appropriate tutor room for what I am working on/tutoring on. Students can also see what students are in each room.
- 3) As a user, I can utilize a text chat to communicate with the tutor and join breakout rooms with other students to work on a specific class or assignment.
- 4) As a user, I want to be able to see other students' and tutors' statuses (tutors: available, busy with a student, etc. students: available to collaborate, do not disturb, hand raised, etc) and share my own.

Additionally, components of the system included an implementation of a client/server to allow for a directory of rooms, a display of users within each room, and a text chat. Furthermore, initial requirements included a database for storage of user login information and the use of the Zoom API for generating meetings for each room.

2.2 Final Requirements

Our final project requirements include the following user stories:

- 1) As a user, I can create an account and login with my credentials.

- 2) As a user, I want to be able to see a list of rooms in a directory and select an appropriate tutor room for what I am working on/tutoring on. Students can also see what students are in each room.
- 3) As a user, I can utilize a text chat to communicate with the tutor and join breakout rooms with other students to work on a specific class or assignment.

Additionally, components of the system include an implementation of a client/server to allow for a directory of rooms, a display of users within each room, and a text chat. Furthermore, a MySQL database is deployed for storage of user login information and the Zoom API is used to generate meetings for each room.

2.3 Initial vs Final Requirements Comparison

The final requirements for this project mostly align with our anticipated initial requirements. Throughout the course of the semester we successfully developed a functional user registration and login system with MySQL database for data persistence, a tutoring room directory, text chat system with client/server architecture, and Zoom meeting generation using the Zoom API. One thing that we had to drop from our initial requirements, however, was the implementation of user statuses.

3 Timeline

The initial timeline for our project, shown below, closely matched with the final timeline. We decided to break our project down into three sprints of roughly three weeks each and followed this throughout the course of the semester.

Week of	Project Goals/Milestones
2/7	Presentation 1
2/14	Start Sprint 1
2/21	
2/28	Sprint 1 Due
3/7	Presentation 2
3/14	Spring Break
3/21	Sprint 2 Start
3/28	
4/4	Sprint 2 Due

4/11	Sprint 3 Start
4/18	
4/25	Final Presentation, Sprint 3 Due

4 Project Results

As stated above, the one key feature that we were not able to implement is live statuses when a user is in one of the tutor rooms. Therefore, we also did not touch any of the ‘nice-to-have’ features that we were considering in the beginning, which were a virtual whiteboard, 2-D virtual room layout rather than just boxes on the user interface, and optional polls and quizzes to aid in the tutoring session.

Going into the project, we had anticipated a few problems, which we identified in our project plan document. We were hoping to be able to integrate the university Central Authentication Service as a security measure to avoid storing user credentials in the database. Unfortunately, we were unable to do so, even after reaching out to the IT department, so our software does store user account info in a central database, though the passwords are salted before being stored. A second problem that we anticipated was the Zoom API being challenging to implement because none of us had experience working with it, and this part of the project did take a larger amount of time due to the learning curve.

5 Software Evaluation

Our group initially planned on evaluating our software based on usability as well as code quality. We attempted to keep the user interface simple while remaining somewhat informative. In sprint one we had many issues with the user interface being coupled to the backend. This hardcoded implementation created more bugs than expected. It additionally slowed down the progress of the backend’s implementation. As of sprint two, we constructed an MVC architecture that separated the user interface from the backend logic. This made updating both the backend and frontend easier now that they could be compiled separately. As of right now there are a few known bugs and issues. Many of which can be fixed with our current knowledge. Although the user interface is easy to use, there are several delay issues. Loading and unloading data from the database is done on the same thread as the frontend. This causes the user interface to wait for user data to load. It will be better to separate the database logic from the frontend.

6 Work to be Done

There is some additional work to be done for this software to meet the goals we had for its development this semester. There are also ways in which this system could be extended for real-world use. The very next steps to be taken would be implementing user statuses, displaying tutor information, curing the lag caused by database operations by improving the efficiency of our queries, and getting the Zoom implementation cleaned up. To continue extending the functionality of this software, we would like to make the actual virtual tutoring room user interface look like a 2-D representation of a tutoring room or learning center type of environment. We would also need to deploy a separate server for Zoom because it cannot run on the local host as of right now. This would make the system more functional.