Classroom Connect

Project Management Plan

Kyle Wright, Anthony Chan, Jacob Wyner, Joseph Cao

1. Project Summary

1.1 Project Overview

The Classroom Connect software is meant to serve as a platform for connecting students and professors in the same classroom via a poll-result interface that gauges students understanding of course material.

1.2 Project Scope

The project scope is primarily to collect and provide easy to interpret, one dimensional poll results. Providing a solution for one-on-one contact between peers, or between a student and their professor, is out of the scope of this project. Additionally, collecting or propagating any information outside o f lecture hours is outside of the scope of this project.

1.3 Development Process

The development process and steps of execution that best suit our schedule and work flow would be the Iterative and Incremental development process. This process best suits our team's workflow because the specific requirements are not entirely understood yet, and the scope of the project - effort wise - is not yet fully known on a technical scale. This lack of knowledge in understanding the requirements makes the Waterfall process unusable since that process relies on requirements being known and solidified early within the development phase. Our team would better rely on a process which helps to incrementally complete major aspects of the project and to revisit and optimize in future weeks of the process. The Iterative and Incremental process provides modular development in that software is developed and combined in small nodes which are 'complete'. This will ensure that we will have a working minimal viable product (MVP) by a certain date, while we continue to revisit steps of the development. Our belief is that we cannot take a chance in going over time, and this process will help us keep on track to the overall goal.

1.4 Effort, Schedule, and Team

Our team is comprised of the following 4 people:

Anthony Chan
Jacob Wyner
Joe Cao
Kyle Wright

Total Effort: 2.5 person months (59 persondays) Project Duration: 13 weeks (3.25 months)

1.5 Assumptions Made

There were several assumptions which were made which led to some difficulties. For instance, it was assumed that Apache and MySQL had within themselves asynchronous capabilities to update user information based on 'triggers'. It was later realized in mid thanksgiving that Client to Client trigger interactions via Websockets were required to obtain the type of communication wanted for this project

2. Detailed Effort and Schedule

The estimates laid out within this table are based on previous predictions of workflow from other process plans and outlines. In summary, with the Iterative and Incremental software development process, the timeline provided below gives the outline of start and 'soft' due dates of general tasks. Testing phases are not necessarily provided in the table since every code module is assumed to require testing prior to completion.

#	Task	Estimated Effort (person days)	Start Date	End Date	Person	Actual Effort (man hours)
1	System Research & Design	9	Sept. 26	October 6	Chan, Wyner, Wright, Cao	
2	Detailed Design	5	Oct 6	Oct 12	Chan, Wyner, Wright, Cao	

3a	Database Setup	13	Oct 12	Oct 28	Chan, Wright	
3b	Client Setup (API Integration)	13	Oct 12	Oct 28	Wyner, Cao, Chan	
3c	Client UI Design	13	Oct 12	Oct 28	Cao, Wyner	
4a	Database Integration	5	Oct 28	Nov 3	Chan, Wright, Cao	
4b	Client DB integration	5	Oct 28	Nov 3	Wyner, Cao, Wright, Chan	
5a	DB / Server Deployment	7	Nov 3	Nov 11	Chan, Wright, Cao	
5b	Client Deployment	7	Nov 3	Nov 11	Cao, Wright, Chan	
6	Client / DB Integration with Websocket	5	Nov 11	Nov 17	Cao, Chan	
7	Code Module Testing	7	Nov 17	Nov 25	Cao, Chan	
8	Rework, Retouches, Final	8	Nov 25	Dec 14	Wyner, Cao, Wright, Chan	

The total effort in person days is: 59 person days

3. Team Organization

Team organization is a pivotal aspect of project success. Our roles are generally refined and specified here, however w e work with the mindset of an agile start up - our roles are defined,

however at certain times we will all be wearing all kinds of ‘hats’. Below is our team’s organization:

a. *Overall Project Manager:* Kyle Wright

Kyle’s ability to initiate conversation and facilitate conversation around the idea makes him best suited to the r ole of our Overall Project Manager. His skills in both C, Java, and more generally soft skills of discussion facilitation will make him a pivotal partner in this project. Kyle will assist Anthony and Jacob in the networking between the server and clients, and will give Joseph a hand in UI/UX Development.

b. *Databasing Developer Lead:* Anthony Chan

Anthony has a wide variety of knowledge of the task flow and processes of software development, having recently worked at a scratch startup as the primary app developer on that team. Anthony has worked alongside primary database developers, and has a general understanding of the networking process required to create a successful server - client software service. Anthony will work closely with Jacob to ensure a solid and secure http connection between the server and the client.

c. *Front End Developer Lead:* Jacob Wyner

Jacob is a bright minded individual with knowledge of client development. Jacob will work closely on the development of the client, deciding - with the help of joseph - visualization APIs, P2P APIs, etc. to use to provide information to the users. Jacob will also work on the development of user authentication and registration.

d. *Quality Assurance & UI/UX Lead:* Joseph Cao

Joseph has extremely useful knowledge of the use of Git and will help Anthony, Jacob, and Kyle all stay on track with the best practices of such. Joseph will oversee the development of the user interface and experience - with collaboration with everyone as well. In addition, Joseph will help to keep everyone on track with the best practices of code development, documentation, and testing phases. While everyone should be acting as their own QA in some ways, Joseph will be the final decider of the modular software iterations created each week.

4. Hardware and Software Resources Required

The only hardware resource required is a workstation for each group member capable of the following:

- Connecting to the internet, and thus Google’s networked services
- Localhost testing with MAMP & Apache servers
- Testing of Websocket via port 9000

5. Quality Plan

The quality control process of this project will consist of the following:

- *Software Requirement Specifications:* Review completed by our team & external groups not working on the project
- *Unit Testing:* All programmers create and complete unit tests for his / her modules. In addition, our dedicated QA will contribute other unit tests to enhance code specifications.
- *System Testing:* After deployment, our servers must be tested for security & data leaks, based on a System Test Plan which will be created & reviewed.
- *Design Review:* Document will be created and critiqued by our team and focus groups.

6. Risk Management Plan

User information is a sensitive issue that is of the utmost priority. If the team decides to implement & integrate APIs for user registration based on other services, encryption of user data and clarification of data scopes will need to be reviewed and checked during implementation. Other than safety of user information, there are no other known risks with this project that may need overview.

7. Project Tracking

Project tracking will occur weekly and monthly via individual, team, and weekly logs, weekly meetings with team members and faculty. Each project member will be in charge of tracking his / her progress in tandem with the project outline, and to report hours for each module completed for later processing.

8. Post v1.0 Release Reflection

Throughout the development of the Classroom Connect software, team organization was a key aspect to successful completion of the project. By effectively breaking up and assigning tasks in the early development stages, while tracking the progress of key aspects of the project throughout its development, our team was able to stay on track and deliver a stable and working version of the software by the deadline.

Difficulties which arose were primarily due to the issue that teammates did not know what they did not know - and this applied to all on the team. Teammates ran into issues with deployment to the Google Cloud Services, or with integrating PHP Websockets, due to the lack of

experience with such. This would have been impossible to avoid, and the only direction was to continue to research, learn, and implement.