CollabFast

Bryant Anderson, Chris Odom, John Aye, Sam Alston

CollabFast [Fall, 2019]

The University of West Florida

12/2/2019

CEN4595 Capstone Project

Dr. Bernd Owsnicki-Klewe

Abstract

The Final Report summarizes the CollabFast project. It analyzes the the general overview of the project. It analyzes the final requirements in comparison to the initial requirements. It discusses the final timeline in comparison to the initial timeline. We discuss the project expectations with the actual project results. We review the project development process and discuss the work to be done.

Table of Contents

Table of Contents

1 Executive Summary

Collabfast is a collaboration application for small projects, teams, and any other type of thing that requires users to help one another. Examples could be teachers grading a number of tests, or Historians working on a dig site. Firstly, you start by creating a new project, and inviting members to it. In the project, you can create tasks to assign to other users, and keep track of them. Calendar dates to keep deadlines in perspective. Chat's for users to communicate with one another, and keep up with the project. You can also use it for a file transfer system, dropping in necessary files, and/or timesheets. Users would also be able to create, and manage multiple projects.

2 Final Requirements and initial requirements

So initially we planned to use AWS for our server, but this was scrapped after it was discovered that it would cost a lot of money. We then moved to Microsoft Azure for our database instead. The AWS also required us to rebuild the whole project from the ground up. For the duration of the project, we did stay with the agile process, there was no need to shift to another style. Not too mention, shifting suddenly in the middle of a project would be more trouble than it's worth. We also did not change roles during the project, as this was not needed, and everyone did what was required of them to the best of their abilities. As for testing, we managed to test as much as we could. Not much changed from the final requirements, to the initial requirements.

3 Final timeline and comparison with the initial timeline

The timeline was kept to the fullest extent that we could. Somethings we did shift, such as extending a number of sprints, since we needed the extra time to work. The things we needed more time to work on were, the implementation of the chat function, and installing the calendar function.

4 Project results compared with expectations

We did not get the full results that we expected. The program is only partially functional at this point in time. The chat function needs more work, and the calendar has not been installed. The project as it stands does not look like it should from an aesthetic point. We did get the tasks portion set up, and the project view is also finished. The UI is partially created, but since we lack some major functions, it is relatively incomplete.

5 Project process review

In retrospect, time could have been better managed to ensure we were able to meet more project requirements. We planned to meet weekly in Discord, but had few meetings in person. We feel that pair programming multiple times a week would have produced better results. Also, we should have done daily, or every other day SCRUM standups to make sure everyone was on task. Since we do not hold each other accountable, often we went a week or so without discussing the project. Also, we could have used weekly class time when we did not have class to work on the project. Finally, we should be planned less for each sprint and should have ensured we had the knowledge to implement all features before adding them as requirements. This would have allowed for better time management as well.

6 Work to be done

The following functionality still needs to be implemented into the CollabFast application. The chat functionality needs to be implemented. We found a few options for chat functionality but realized that it was a time wise, implementation was out of scope. We also need to implement the calendar functionality. We determined that we would use Fullcalendar API for calendar functionality. We found that we were unable to successfully implement this functionality within the given timeframe. We also need to clean up the user interface to make it more user friendly. Given another sprint, we could also finish implementing document sharing and have the invitation system fully functional.