**Project Description**

The goal of this project was to design and implement several new functionalities for an existing website. The project required Ruby-on-Rails and the practice of Test Driven Development, Agile Development and Continuous Integration. The code coverage needed to be 100%.  We were able to complete two of these goals, discussions tab and student membership.

**Discussions Tab**

This feature allows a user of the site to create a discussion, or reply to an existing discussion. The user must be signed in to use this function, but anyone may view the discussions. AJAX calls are to be used throughout the tab. Emails are sent to the corresponding persons if there is a remark made about their presentation or discussion they are involved in. This goal has been completed.

**Student Membership**

This function allows users to apply for a student membership. Before applying the user must be signed in. The student is then allowed to fill in a form that contains two types of validations. Either a student chooses edu email, or a professor recommendation. After submitting the form, their status will change to pending. If the student chooses .edu email, they will receive an email will a link. Once they click the link they are asked to log in. The status changes to waiting after they sign in. If they choose the professor recommendation, an email will be sent with a link to the professor. The link will ask for the professors URL school website address. After the professor submits the URL, the student status is changed to waiting. Once a student status has changed to waiting, the admin will receive an email with a link. The link will direct the admin to a page where they may approve or deny a student membership. Once the admin has approved or denied membership, an email is sent to the student notifying them of their status. At any time in the process a student may view their status. This goal has been completed.

**Key Learnings**

Ruby was a new language for the entire team. Only two of the team members were familiar with test driven development and continuous integration. Creating AJAX calls was new to everyone. Learning to test the build before committing to Jenkins took some time. Working with the continuous integration and how to debug after failure became second nature to the team. Trusting team members became critical, as well as learning the programming strengths and weakness of each team member. This last item was the key ingredient in finishing the student membership.

**Challenges and Solutions**

Our most difficult challenge had nothing to do with the code. Control turned out to be our enemy. We spent time trying to find the perfect online task manager. We spent more time trying to assign tasks. We talked about what we were going to do. Nothing got done. This was a true waterfall experience. The less we tried to control the project, the more we got done. This is where we switched to agile development. If you don’t know where to start, just write something. We still wrote our tasks, but with less structure. Just having a running list of current tasks allowed for anyone to do what was next. Most of the code occurred after we learned this lesson.

Much thanks to our professors Shishir and Venkat!

        Obinna Ugwuzor

        Menglong Piao

        Mark van Esch

        Braden Duke

        Ravi Kiran

        Dawn Hughes

        Steven Avila

        Abhishek Koluguri