Sprint 2 Report

The Team

- Sam Lichlyter (Product Owner)
- Kang Li (Scrum Master)
- Shannon Ernst (Developer)
- Deqing Qu (Developer)
- Jianchang Bi (Developer)
- Teng Li (Developer)
- Chengxi Yang (Developer)

Project GitHub

https://github.com/slichlyter12/CS561-Project

Sprint Overview

During this sprint Sam Lichlyter was the product owner and therefore maintained the sprint backlog as well as conducting sprint reviews and led sprint retrospective. Kang Li was the scrum master and led daily scrums and documented what was discussed and took photos for proof of each meeting. Teng Li worked on making sure our software could handle multiple classes as well as working with Deqing on TA stories. Those of which included picking which class the TA was assigned to, which question the TA was assigned to, details about the question asked and viewing all the questions in a list format. Jianchang Bi worked on the features to allow students to be able to add themselves to an already written question. Chengxi Yang finished up the algorithm as well as built a welcome page. He also performed a spike in which he researched technologies for animating the questions when the algorithm updates the order of them. Shannon Ernst worked on the Admin pages which included allowing them to add courses, add TAs to those courses, logging in, and allowing all users to logout.

Daily Scrum

Scrum Photos:
https://github.com/slichlyter12/CS561-Project/tree/master/docs/sprint_2/photo

Scrum Record:
https://github.com/slichlyter12/CS561-Project/blob/master/docs/sprint_2/Progress.xlsx

Stories Completed

Software needs to be able to host multiple classes: Need to be able to be a portal for multiple courses. Keep track of which TAs are in which course, which students are in which course. (Teng)

Student should be able to add themselves to an already written question: Questions should have a button when they are displayed to allow students to add themselves to the question. This button should add the name of the student to the list of interested students and increment the count of students wanting an answer to this question. (Bi)

Main Entrance: There should be a front login page for students and TAs with a brief description of the product. (Chengxi)

List Animation Spike: (Chengxi)

Software should be able to group students based on question and order priority based on number of students/time passed: Two stories: first sprint is the algorithm design, second is implementing and testing (Chengxi)

TA needs to pick which class they are working on: When a TA logs in, if they have multiple courses assigned to them, they need to be able to select which one they are currently working for. (Teng, Deqing)

TA needs to assign themselves a question: TAs should be able to view the question queue and assign themselves questions to answer so that other TAs and students know someone will be answering the question. (Teng, Deqing)

TA needs to be able to view the timestamp, contributors name, question text, number of students and their names, and if a TA has been assigned to it when viewing the question: Display of the question in the TA queue (Teng, Deqing)

TA should view questions in a list format: Display of the question in the TA queue (Teng, Deqing)

Admin should be able to add a course: (Shannon)

Admin should be able to add TAs to the course: (Shannon)

Logout: (Shannon)

Admin Login: (Shannon)

Sprint Retrospective

What Went Well	What didn't go well	What we could do to improve
We met consistantly	Misunderstanding of each other requirements. Stories were not written clearly to indicate who was responsible for what.	Familiarize with PHPUnit. We will create a read me so everyone knows how to do this.
Completed all user stories for this sprint	Unfamiliar with unit testing frameworks (PHPUnit), inconsistent test driven development and how the files were set up for testing. This caused much confusion between the developers. There was also a lot of deviation in environment set ups which impacted how unit tests were run.	Refactor PHP code into classes together. We will do this as a code jam so that everyone will know how the code has been refactored and how to write unit tests for it. A read me will be written to give a guide on how to set up a testing environment, how to write a unit test in php, how to write a unit test in javascript, how to add a test to the test script and how to run the test to generate a report. This will resolve the issue going forward of how to do test driven development.
Software is usable	Moved away from REST	We will rewrite remaining user stories to be more clear as to what is expected and what needs to be tested.
Separated front-end and back-end code	Relies too heavily on session	Needs more security, limit usage to only those with permission for each page. Student shouldn't be able to access admin page

Sprint Backlog (yellow indicates completion)

https://docs.google.com/spreadsheets/d/11aYoXOcj0m9VNJueFHJUlagau9j0Bd6CzYutle8b5YM/edit#gid=0

Testing

Unit tests:

(Web): http://web.engr.oregonstate.edu/~lichlyts/cs561/tests/test_report/unit_test/report/

(GitHub unit tests): https://github.com/slichlyter12/CS561-Project/tree/master/tests/test_report/unit_test