Formative SAGE Assessments: Midterm Progress Report

Ruicong Xie Tianci Zhong

COMS E6901, Section 14

Fall 2016

1.0 Introduction

This is the midterm progress report for SAGE frontend dashboard for students and instructions. We will summarize the all the work completed up to this point, and all future work that we planning to complete for the rest of the semester. In section 2.0, we describe the mock server that hosts students and instructors' dashboard and serving mock data. In section 3.0 we discuss the design and implementation of the student's dashboard.

2.0 Test Server

We build the test server in NodeJS to host and serving mock data to render dashboard built with AngularJS and Bootstrap to connect all the comment of Model-View-Controller. We are still in the progress of integrating it with SAGE backend that the other teams are working on. We have a share documents showing all the endpoints that data format for API calls. Figure 1 illustrate the structure of the project. The mock data are exposed through two endpoint, localhost:3000/stats/students/:id and localhost:3000/stats/instructors/:id.

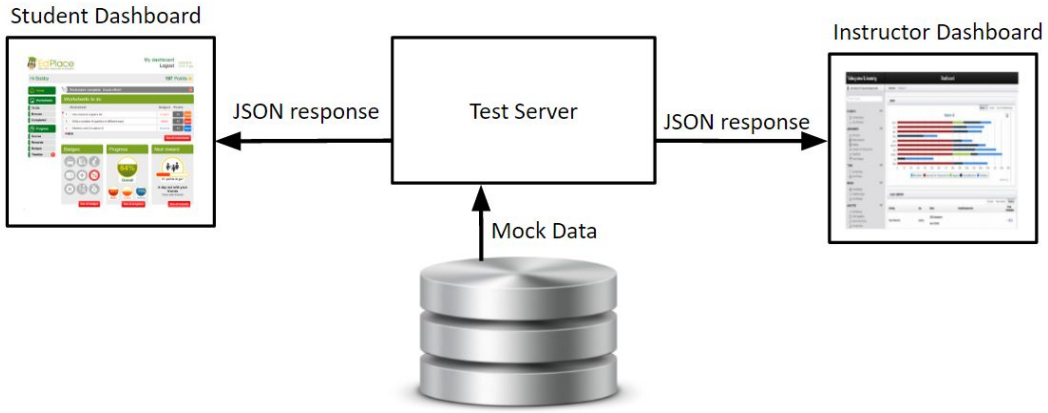
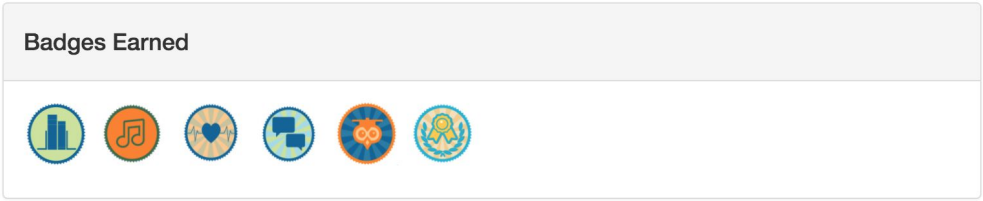


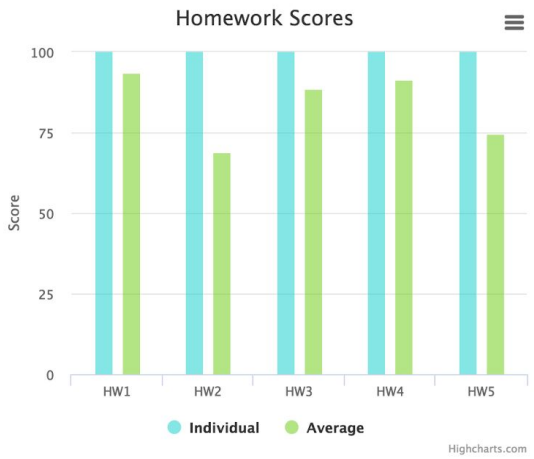
Figure 1 Structure of Dashboards and Test Server

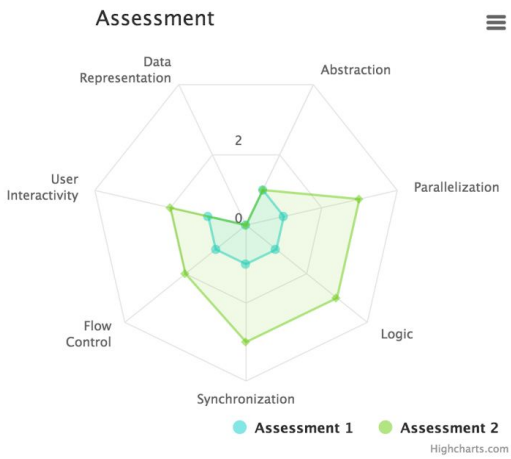
3.0 Student Dashboard

The student dashboard use the single page website design, but the student dashboard and the instructor board will be two website page. In the student dashboard, there are a couple of mechanics we built to encourage student to take ownership of their learning progress including displaying student's Avatar on their dashboard view; introduced a badges system that awards students' behaviors like starting assignment early, adding comments etc.



The dashboard also contains a round progress bar for current assignment with animation. In addition, we added bar chart and radar chart showing students' assignment's computational thinking assignment result. The attributes include data representation, abstraction, parallelization, logic, synchronization, flow control, user interactivity. We adopt the hairball mastery, a plugin for scratch static analyzer that used to assess computational thinking of Scratch projects. It is also the same assessment tool used in Dr. Scratch, a free online that evaluate Scratch projects in a variety of computational areas.

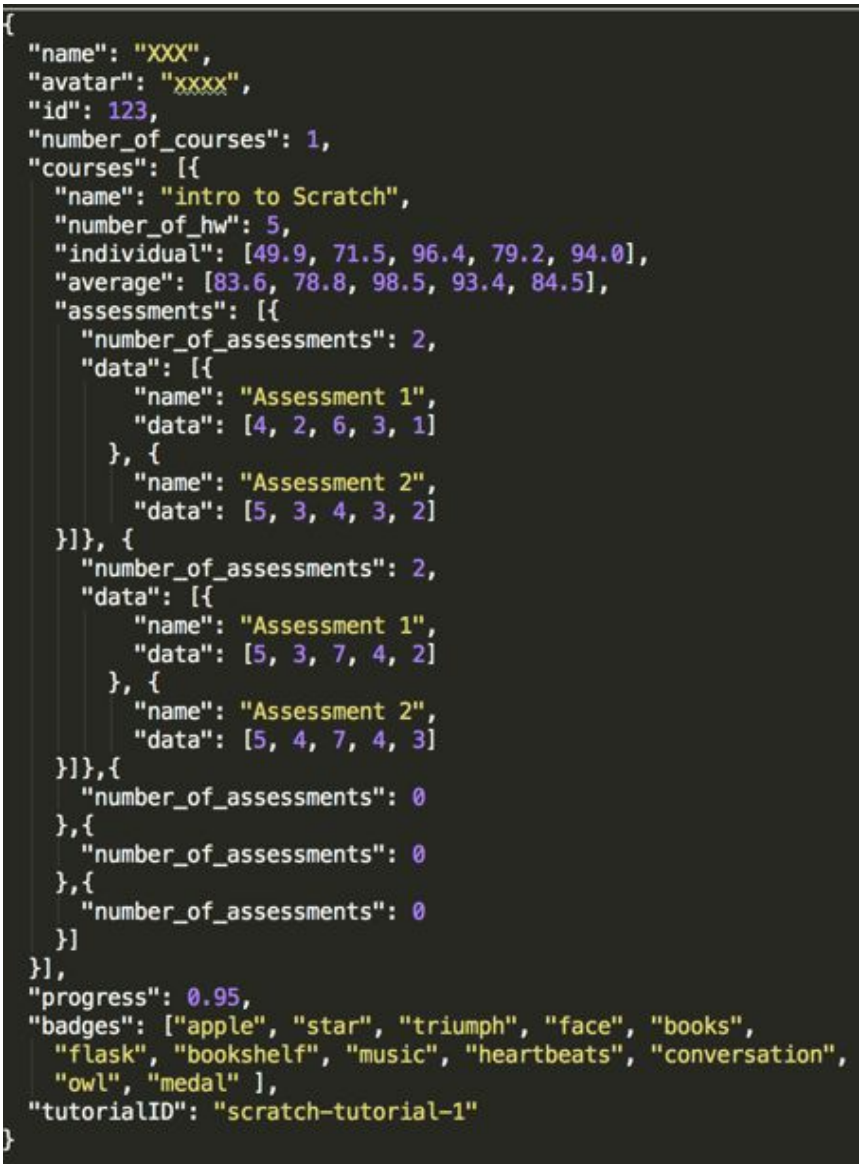




In the class overview section, we will put student's current progress in context of the entire class by showing a histogram on how they are performance comparing with their fellow classmates. We also added a assignment instruction panel, which include a .flv flash video file that instructor can record upload as a visual instruction tool helping students better understanding the assignment requirements. Currently the video files is hosting on AWS S3 buckets. We used angular-videojs to help brower decode flash video. We added sliding effect and hide and show animation for the dashboard, so later on we can easily integrate our design of the dashboard with scratch editor.

3.1 Student Data

The mock student is saved as Json file. Here is how the current student data looks like.



4.0 Pending Work

The following section describes pending technical work.

4.1 Student Dashboard

for student dashboard there are couple of tasks we still need to do. First complete the click animation for homework assignments and hover effect for badge system. When clicked on a homework assignment the computational thinking assessment radar charts will show the the corresponding assessment results. Secondly, implement leadership board for students to encourage positive class performance. Last and the most important one, integrating our current dashboard design with the existing SAGE scratch editor.

4.2 Detailed Documentation of all API calls

We need to add a detailed documentation for all the api calls to render our dashboards and sample dataset in JSON format.

4.3 Teacher Dashboard

Teacher dashboard, on the other hand, will have a more comprehensive view. All the class will be listed on the right. When clicked a specific class, all the assignment related to that class is presented in tabs on the right side of the page. Under each tab we will show a class progress bar at the top and a bar chart of top and bottom 10 performances for that assignment. Below the bar charts is a table view of each student with their id and aggregate data such as assignment starting time, completion time, and solution length that can be sorted in ascending or descending order. This design will help teachers identify struggling and overachieving students. All the students' entry will also have their photo next to their id, so instructor can quickly

identify their students. When instructors click on a student the server will redirect the browser to student's assignment page and show a list of error student made in time order, so instructors can more easily understand what students are struggling with and provide help accordingly. We need to also add the functionality to upload .flv file instructors recorded explaining assignment requirement to AWS S3 buckets.

5.0 Summary

Due to some compatibility issues we have to circumvent between the chart library, AngularJS and flash video decoding library, we are currently one week behind our inital schedule. However, we believe we will be able to catch up with our milestone and deadlines in the coming weeks, since there are enough similarities and overlays between students and instructors dashboard.