**Project Meeting 2 Report**

**Cohort 2 Group 9**

**Project Overview**

We will be developing a smart interactive classroom application that will help facilitate learning in SUTD. The main features we will be including are real-time questions, a feedback system, student performance reviews and a quizzing system.

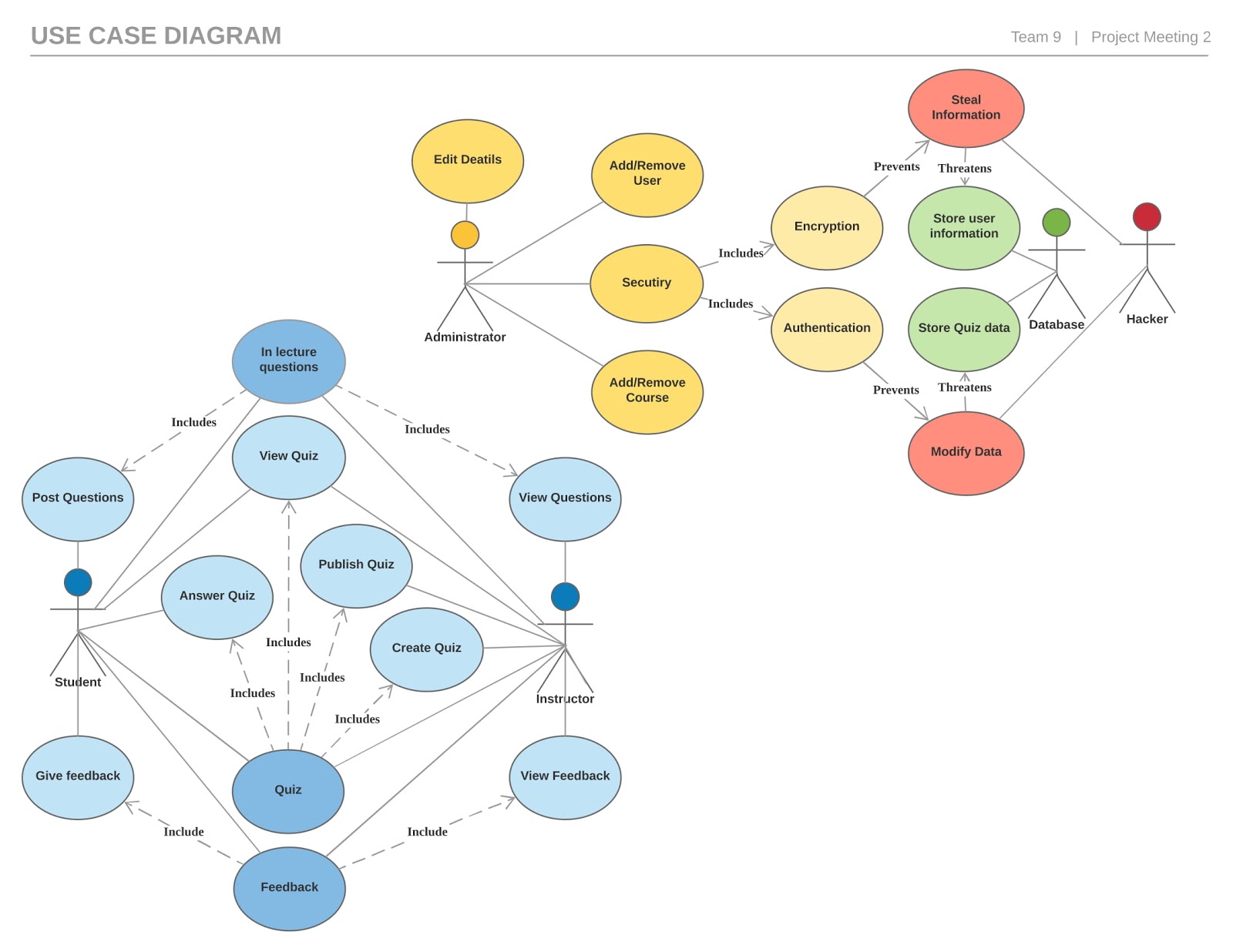
We will be developing the application on javascript with a MERN stack, which consist of mongoDB, express.js, react.js and node.js. This will allow the web-based application to be cross platform which does not restrict the users to a single device.

We will be adopting the AGILE software development methodology. This means that although each person has an area in which they are incharge of, there will be a lot of cross interactions between the individuals. We aim to push out working software as fast as possible so that we can gather user feedback to improve our design

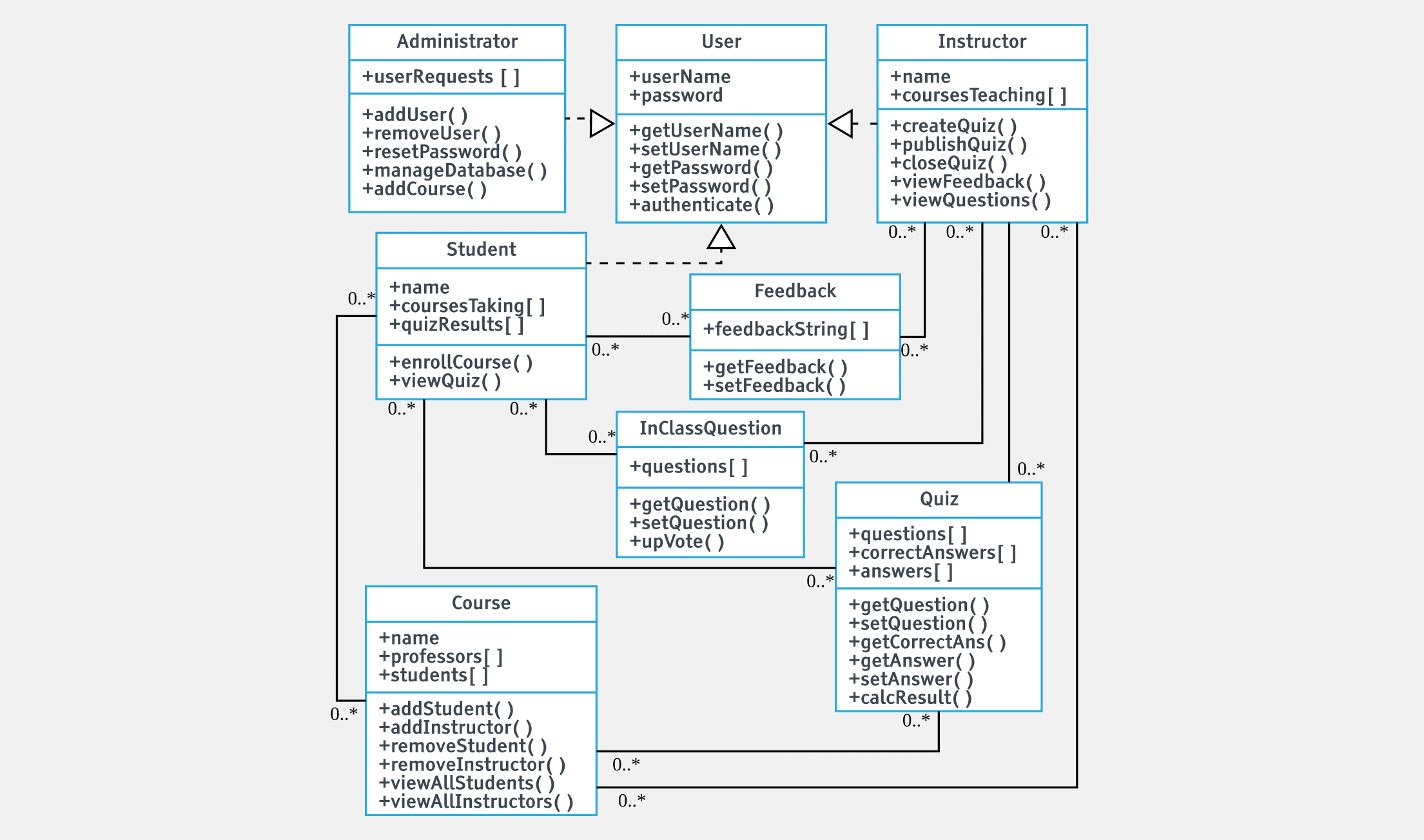
**Changes in Requirement (0.5)**

* Most of what our group had previously discussed remained with some exceptions to a few features.
* During project meeting 1, we asked the opinion of our professor on having a chatroom and to that, he believed that there should not be too many features as it may be distracting to the students during lecture. Hence, we modified the chatroom concept to make it into a forum type of platform, where students can post any queries that they may have regarding the lecture and/or the lecture slides. This makes it more formal and will be apt for a classroom setting. It is to also avoid any unnecessary messages, hence, possible distractions to students during lesson time as our professor stated.
* Second, we were debating whether we should display the posted questions beside the slides or in a separate tab. Some of us thought that it might be distracting and that it should only be shown to instructors. However, we decided to display it to the students as well so that other students will know the questions asked and when the instructor answers them, they will be able to follow along.
* Lastly, we felt the need to clearly distinguish this platform from the already existing platforms like Edimension, for instance Therefore, we decided that there will be a live stream of the lecture slides during lesson time. This enables the students to follow along better during lecture and also have a look at questions and answers posted on each slide by students which may be very helpful especially when doing exam revisions. The live stream, quizzes, and forum all in one platform is the one that greatly distinguishes our web app from any existing platforms that are all either required to be accessed in separately or mostly not in real time.

**Formal Documentation of use case (1.0)**



**Initial Design (1.0)**



**Testing Plan (1.0)**

1. Unit testing

* Is the testing of individual software modules or components that make up an application or system, to determine whether they are fit for use. One of the tools that or group will be using for unit testing will be Karma and QUnit.
* Karma
  + - is a JavaScript test-runner built with Node.js and meant for unit testing.
* Mocha
  + - is a feature-rich JavaScript test framework running on **Node.js as well**. It maps uncaught exceptions to the correct test cases.

Examples:

|  |  |  |
| --- | --- | --- |
| Feature | Test Condition | Expected Result |
| Login (Both Professors & Students) | Test for registered user logging in with username and password. | Login successful. |
| Answer Quiz (Students) | Test for submission of correct answer. | Answer correct! |
| Lecture Rating (Students) | Test for selecting number of stars for the rating. | All the stars from the first star to the star that is clicked on is highlighted. |
| Post Questions (Students) | Test for questions being delivered. | Questions are uploaded without any system hangs. |
| Register (Students and Professors) | Test for password with the correct requirements. | Password field check. |

Features that should be unit tested:

* Add/Remove User
* Quiz
  + Create Quiz
  + Publish Quiz
  + View Quiz
  + Answer Quiz
* Feedback
  + Give Feedback
  + View Feedback
  + Lecture Rating
* Login
  + Register new user
  + Change profile information
  + Login page
* Questions
  + Post Questions
  + View Questions
  + Receive reply

1. System Testing

* tests the entire system for errors and bugs. This test is carried out by interfacing the hardware and software components of the entire system and testing it as a whole. This testing can be categorized under the black-box testing method, where the software is tested by user study for potential exceptions and edge conditions. The methods our group is going to use for system testing is User Study and Crowdsourced Testing.
* Crowdsourced Testing
* User Study

Examples:

|  |  |  |
| --- | --- | --- |
| Feature | Test Condition | Expected Result |
| Main Page | Test navigation to all links in main page. | All successful, none of them is broken. |
| Give Feedback | Test for delivery of feedback to the professor. | Feedback successfully delivered to the professor to view. |
| Live stream | Test for viewing live stream. | The slide on student’s screen corresponds to the slide displayed on classroom screen by professor. |
| Login | Test for correct login of registered member. | Login successful and the user directed to main page. |
| Courses | Test for ease of access into courses. | Courses easily identifiable, accessible and no lags in between selecting and viewing the course materials. |

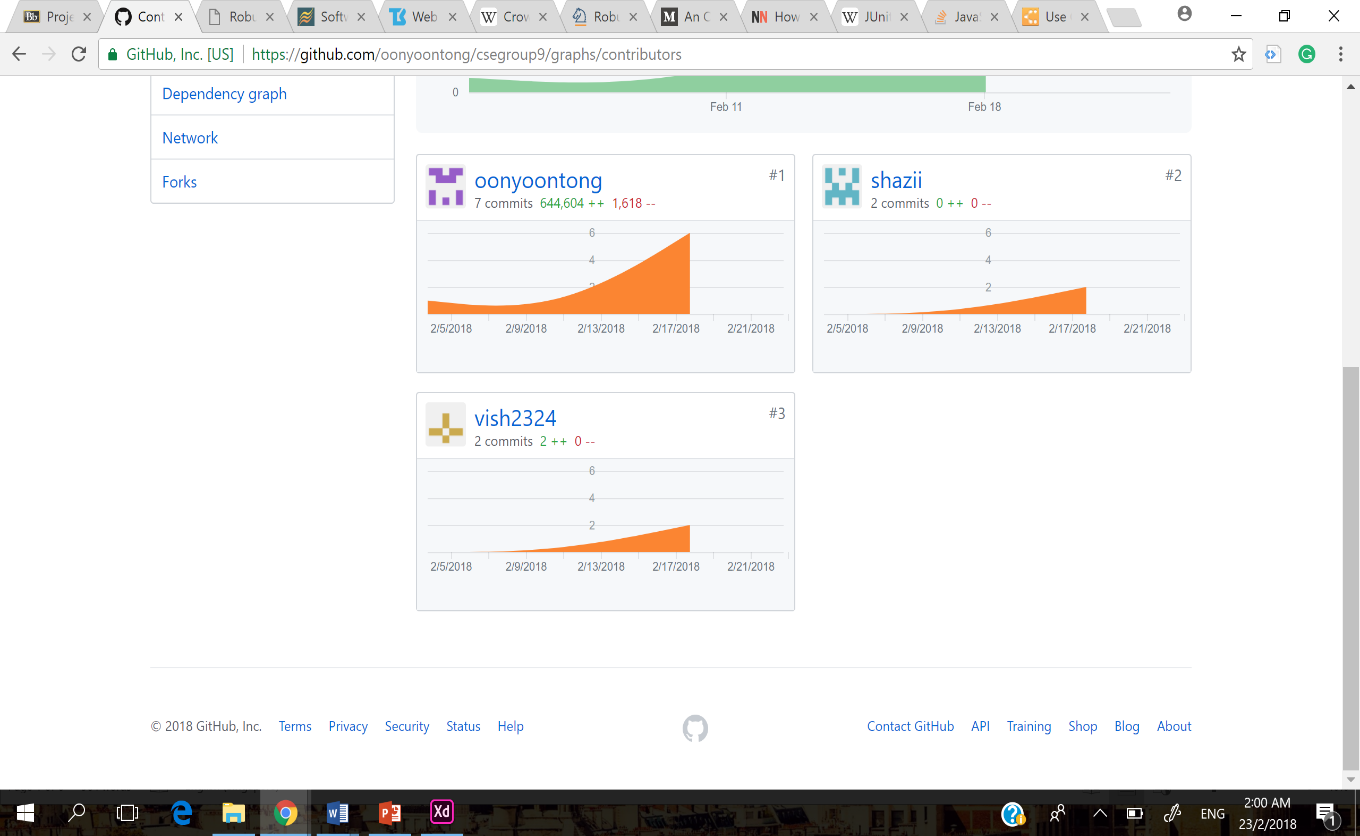
1. Robustness Testing

* the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions, through black box testing, equivalence class partitioning and boundary value analysis.

Examples:

|  |  |  |
| --- | --- | --- |
| Feature | Test Condition | Expected Result |
| Login (Both Professors & Students) | Test for invalid username and Password. | Login failed. |
| Answer Quiz (Students) | Test for no answer selected or typed. | Submission failed. |
| Lecture Rating (Students) | Test for no stars selected. | No rating given. Please rate the lecture/ |
| Post Questions (Students) | Test for the longest possible question allowed. | Question uploaded successfully without any crash. |
| Register (Students and Professors) | Test for already existing username. | Username already exists. Please select another username. |

1. **Workload distribution (0.5)**



URL Link to GitHub: <https://github.com/oonyoontong/csegroup9>