CPEN 321: Software Engineering

System Design Document

Last updated:
27th November 2016

Group: Cyann

Chen Chen (3281 6143)
Howard Zhou (4898 6137)
Justin Toh (2954 9136)
Luvian Wang (1693 9134)
Xi Chu (1735 3137)
Yufei Qiao (4885 7130)

Table of Contents

1. INTRODUCTION

We plan to build Cyann as a mobile and web application which provides a platform for students and instructors to communicate. Students can use this app to post questions and answer/comment other students' questions. Instructors can use it to answer students' questions and post announcements and notes.

This document attempts to explain the architecture and design of the system.

2. SYSTEM ARCHITECTURE AND RATIONALE

Cyann, the student-teacher communication application with different versions for students and instructors, consists of the following major components:

1. A mobile application with only the student version, connected to a database for users to create, comment and manage posts.
2. A web version of the application with both student and instructor versions, connected to the same database, mainly for instructors to create courses and upload files.
3. A backend server to store, access and manage the resources the users uploaded.

2.1 React.js

The Web application is built with ReactJS. React makes it painless to create interactive UIs. Design simple views for each state in the application, and React will efficiently update and render just the right components when data changes.

Source: https://facebook.github.io/react/

2.2 React-Native

The mobile application is coded using a open source JavaScript framework called React Native. React Native allows us to build mobile apps using only JavaScript. It uses the same design as React, empowering us to compose a rich mobile UI from declarative components. React native works reliably on both iOS and Android devices, and combines smoothly with components written in Objective-C, Java, or Swift.

Source: https://facebook.github.io/react-native/releases/next/

2.3 Node.js & Express.js

Since our web and mobile clients are using JavaScript, we decide to use the same language to implement our backend infrastructure so that we can easily transfer people between subteams without worrying about any programming-language barrier. Using Node.js for our backend server was a clear choice for us as it's the leading server-side JavaScript framework used in the industry.

Express.js will be used to handle the routing on our server as it is the de facto standard server framework for Node.js. The framework was released 6 years ago, hence there's a wealth of online resources and a huge community for our team to learn its API and easily seek for support when we need it.

Express.js is a Node.js web application framework written in JavaScript and used for delivering web applications to mobile devices and building APIs. Express.js has the following features:

- Robust routing
- HTTP helpers
- Content negotiation
- Executable for generating applications quickly

source:https://github.com/expressjs/express

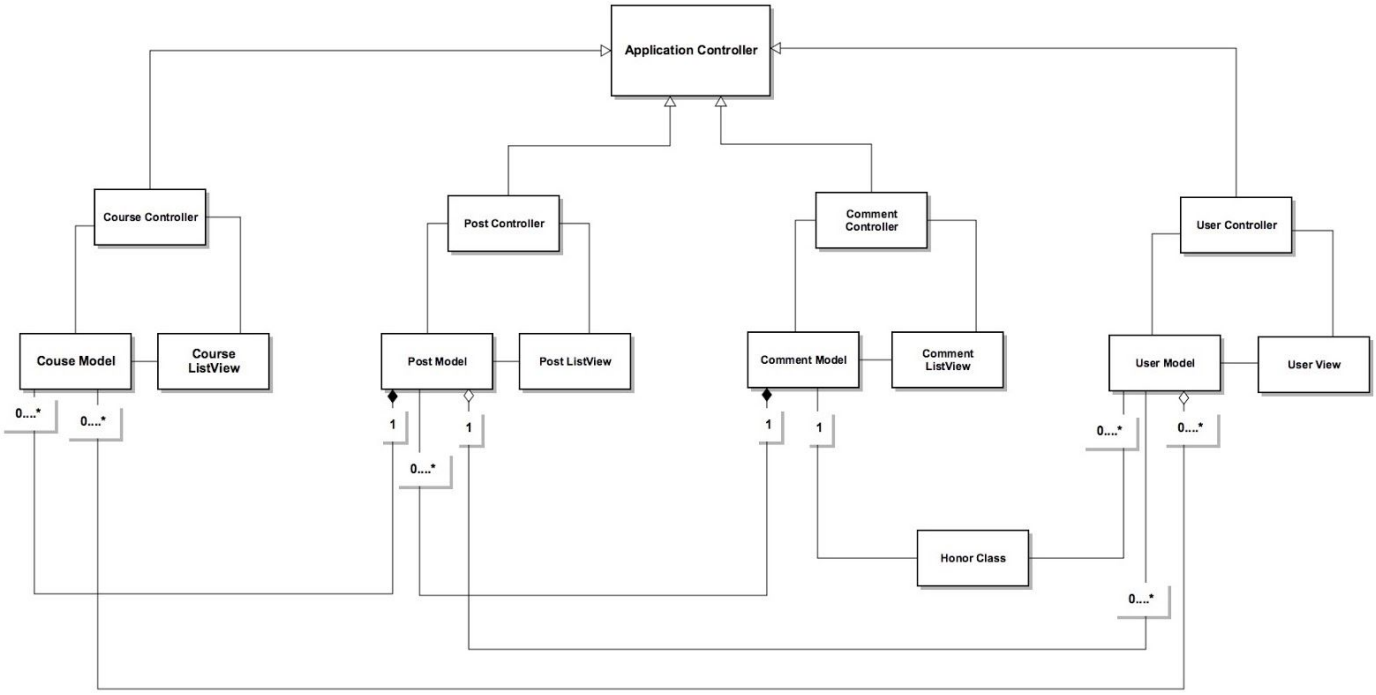
2.4 MongoDB & Mongoose

The backend database is built using MongoDB which is an open-source document database and leading NoSQL database written in C++. It allows us to create and deploy highly scalable and performance-oriented databases.

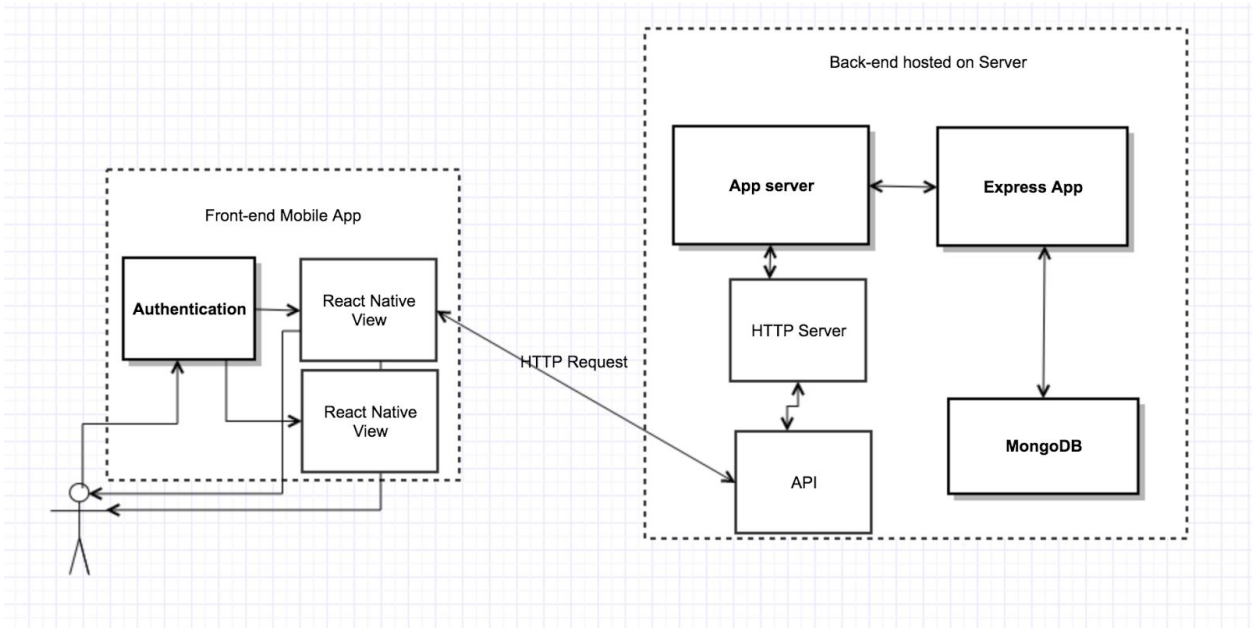
Since MongoDB does not enforce any form of database schema, we will be using the Mongoose framework as our ODM (Object Document Mapper) so that we have a built-in automatic validation of the data which we are inserting/updating.

The primary reason that MongoDB was chosen over traditional SQL databases because it has a far better integration with Node.js (uses JSON documents to store records). Mongoose's simple querying API also removes the need for our team members to learn SQL queries to perform database operations.

3. Class Diagram



4. High-Level Design



5. GUI

5.1 Design Principle

Our UI design follows Google's material design guideline meaning all UI elements will be presented to users in an abstract yet realistic way complemented with simple animations.

5.2 Typography

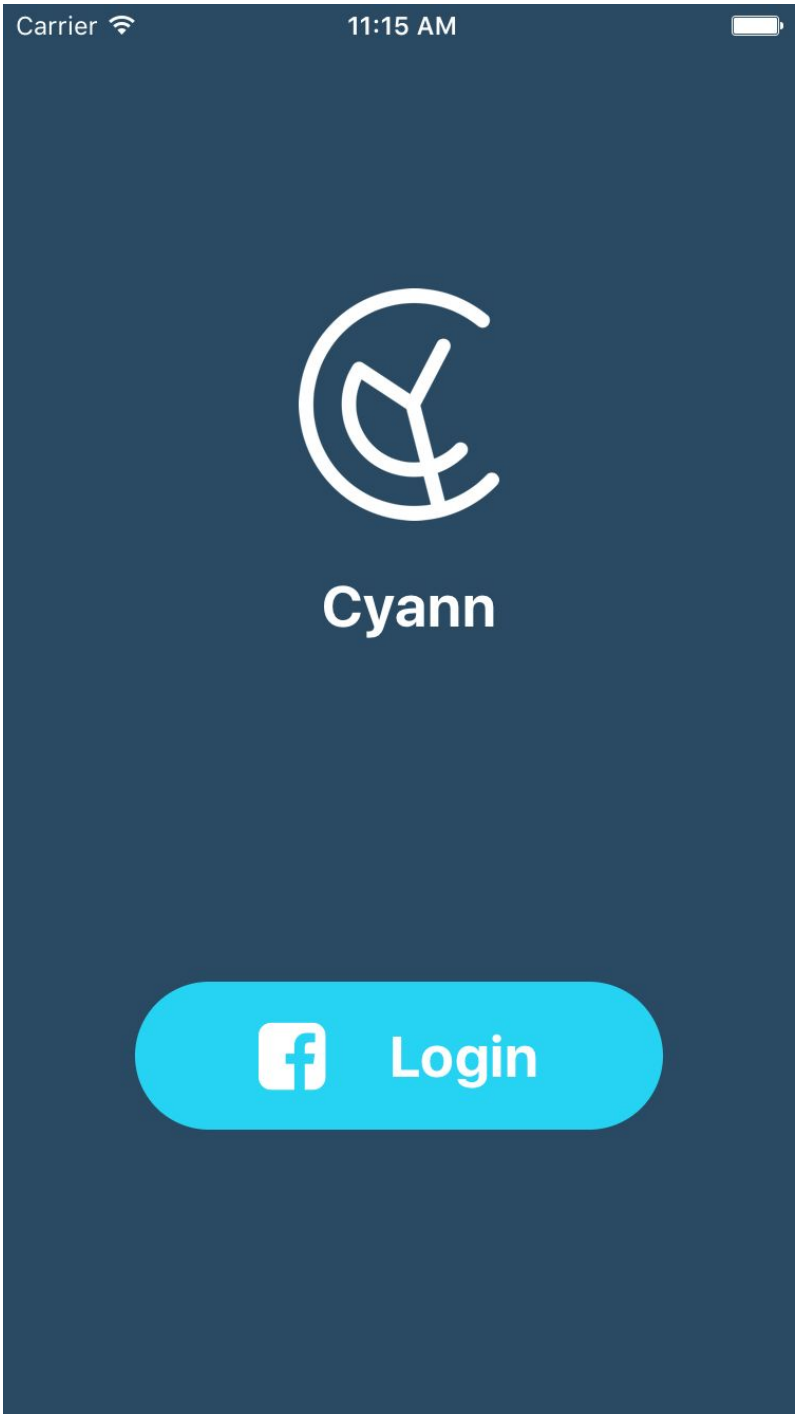
Helvetica was used on the mobile application to increase readability.

5.3 Icon set

To better accommodate for flat design principle, Font Awesome was used in both the mobile and web app.

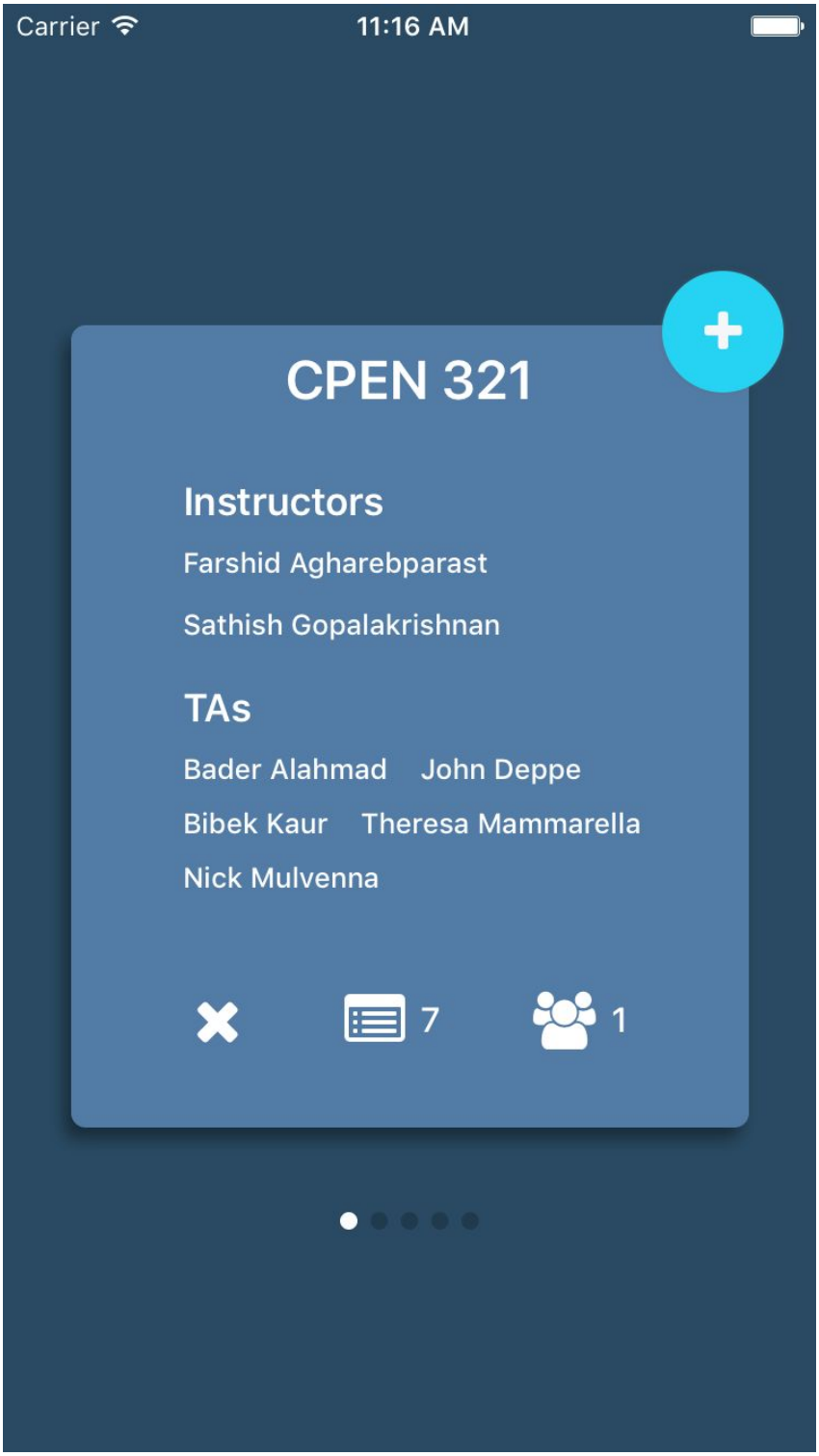
5.4 Login page

Cyann uses facebook login therefore the users don’t need to provide any extra information on the login page, making the page clean and intuitive.



5.5 Class selection

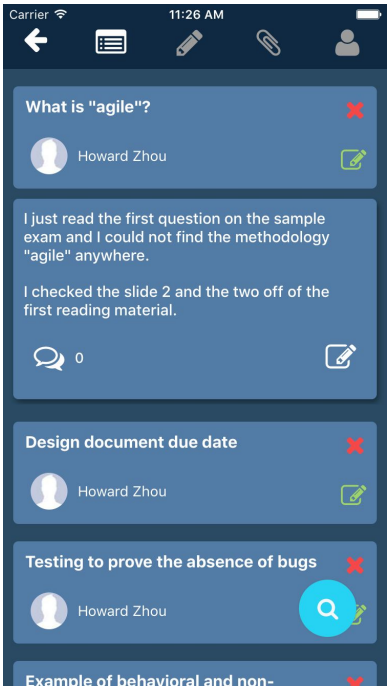
The class selection will present the user a horizontal scroll view and on each course card, the brief information about the course will be listed.

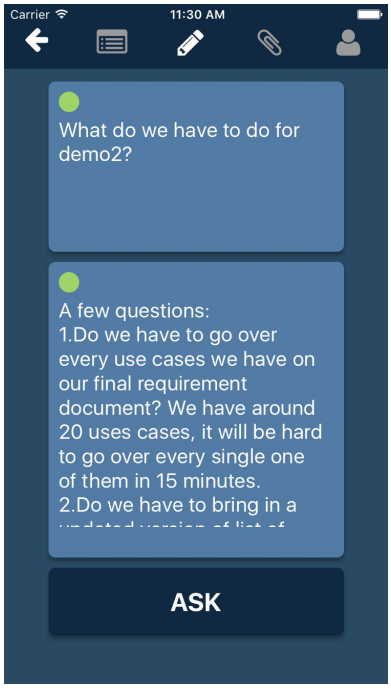


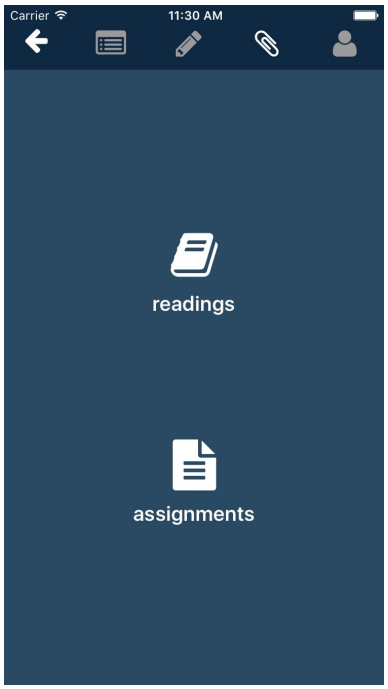
5.6 Main page

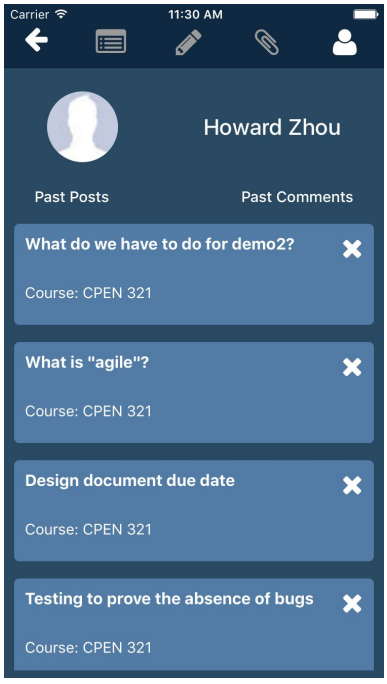
The users will use a scrollable tab view to navigate between four pages. The pages from left to right are a “Post Page” which contains all post within the course, a “Ask Page” for user to

compose and post their question, a “File Page” which list all the files within a course, and a “User History Page” for user to see all their past post.



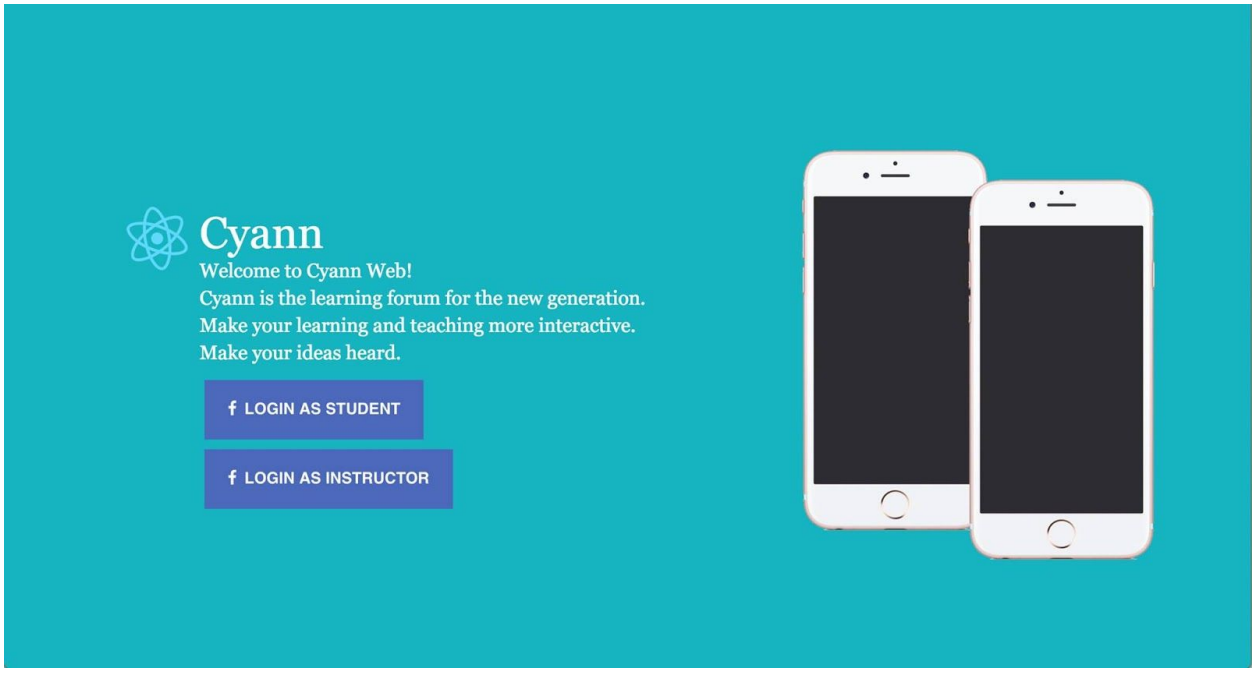




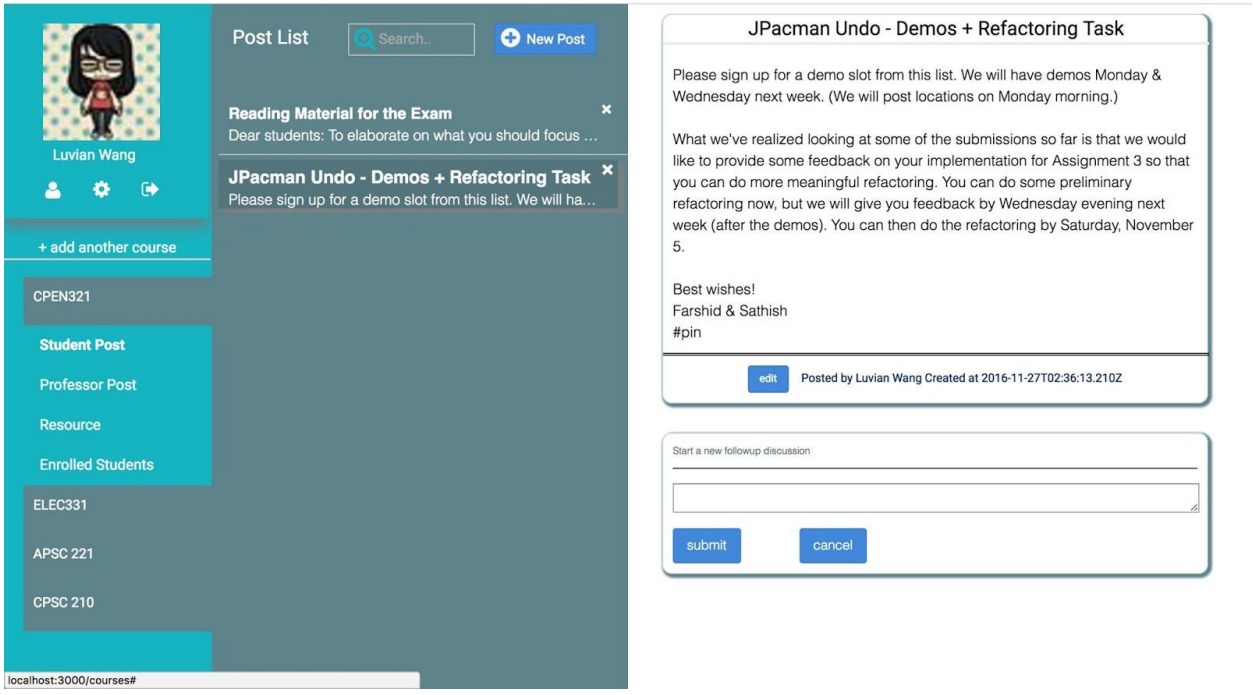


Web UI:

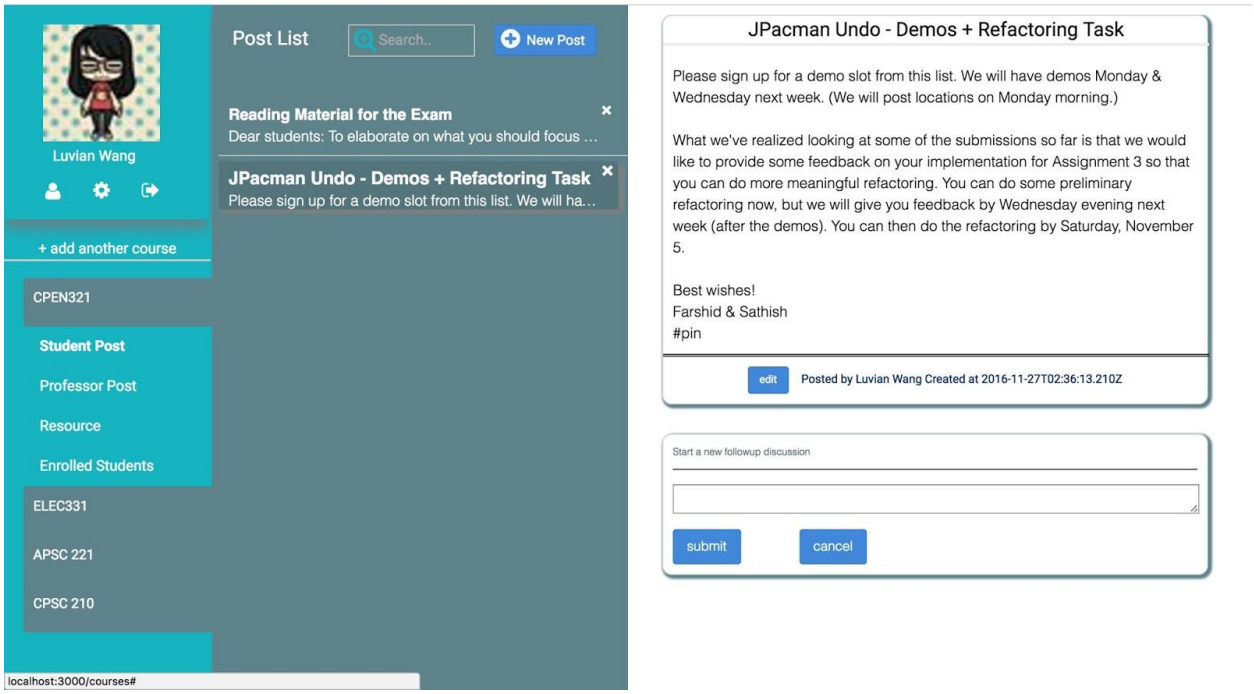
5.7 Landing page:



5.8Main page:



5.9 Course Selection page:



6. Validation

A	B	C	D
First Name (or Anonymous)	Date of Test	Feedback	
Alicia	Nov 25, 2016	1. Keyboard will mistakenly pop up when I swipe from question posting to file display.	
		2. In file reading mode, "back" button should be at the top left corner, consistent with other scenarios.	
		3. There is a "x" botton not working.	
John	Nov 25, 2016	1. Search button should be placed at the bottom of the page as a bar and can swipe up.	
		2. On the first page, put "+" button on the top right corner of each course, instead of a "menu" button.	
		3. Display animation for each page not just for one time.	
Juliana	Nov 25, 2016	1. Not sure which tab I'm on	
		2. Not sure if looking at past post and past comment	
		3. Past comments doesn't show title	
		4. Navigating is a little confusing	
		5. Button should just be a plus sign or repositioned	
Fred	Nov 25, 2016	1. Make course cards bigger	
		2. Backbutton	
		3. Search bar on main page	
		4. New post notification(backend)	
		5. Unread tag(backend)	
	The app has made adjustments accordingly		
	The feedback has not been considered yet		

In order to test the user friendliness of our UI, a hallway usability test was performed and several UI component was changed regrading to user's opinion.