

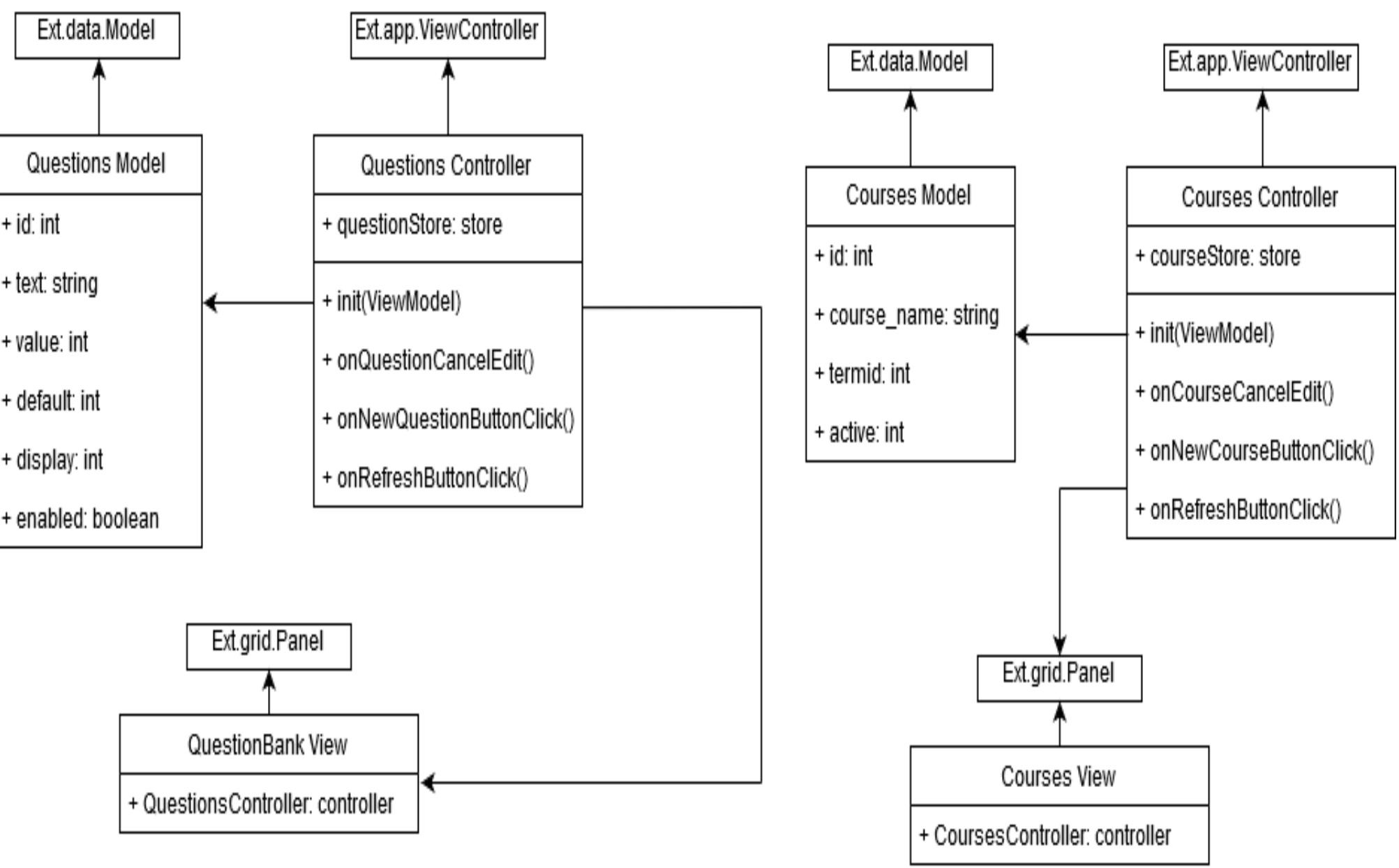
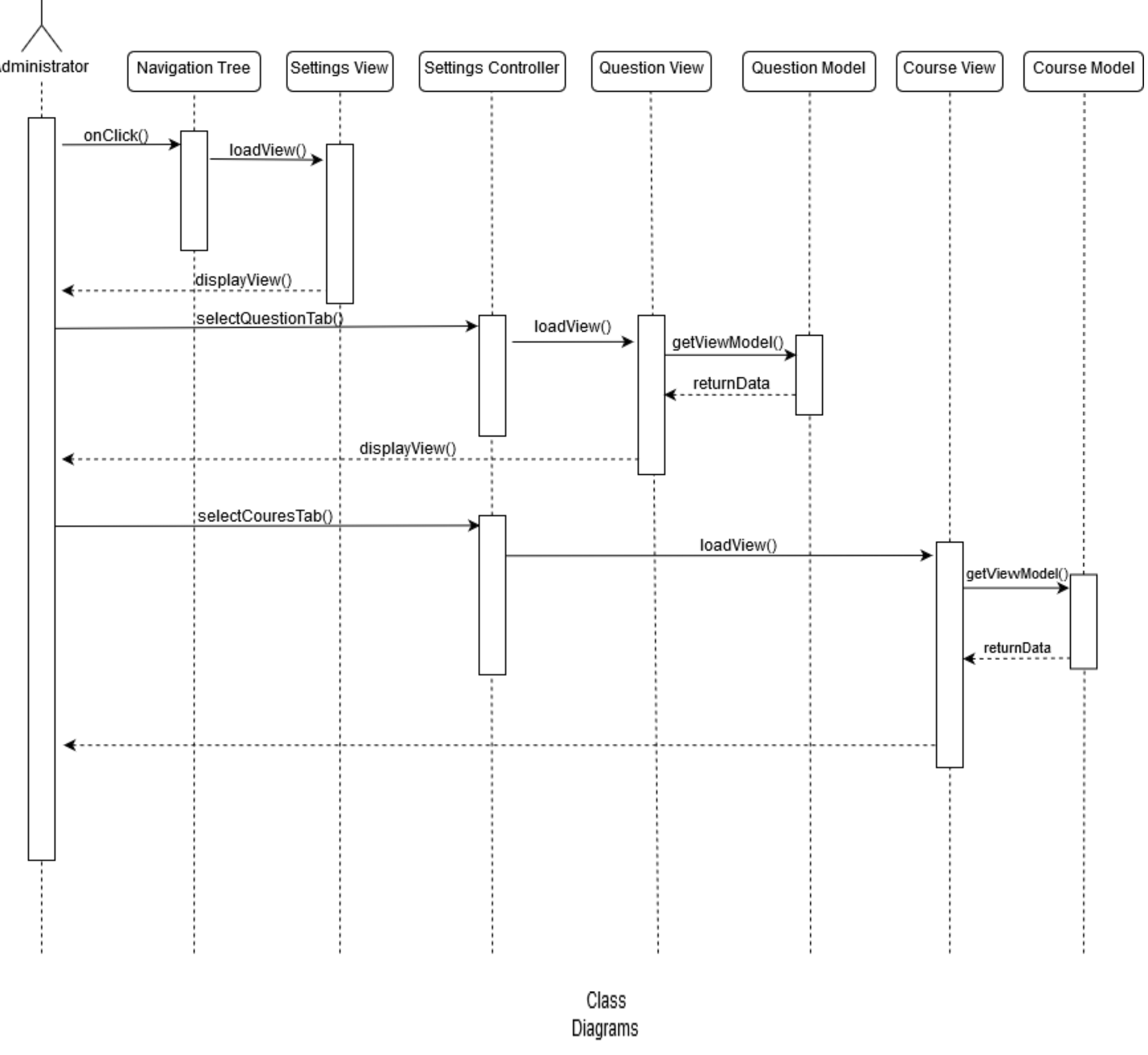
Problem

At the end of each semester students graduating and enrolled in the Senior Project course present their capstone projects at the Senior Showcase. Alumni, university professors, and industry professionals attend this showcase to judge and present feedback to the student's projects. An application is needed to centralize and streamline all the efforts required to coordinate such an event.

Solution

This problem has led to the development of Mobile Judge. Mobile Judge is a system that allows management of students and judges, as well as allows the showcase administrator to view all the necessary statistics and configure things such as: questions, showcase start-time and end-time, student's grades, and much more.

Object Design



Current System

Admins:

- Configure showcase information for a term.
- Invite Judges via the internal emailing system.
- Sync students via an API through VIP's Senior Project Website.
- View statistics on Judges and Students.
- Assign grades to students.

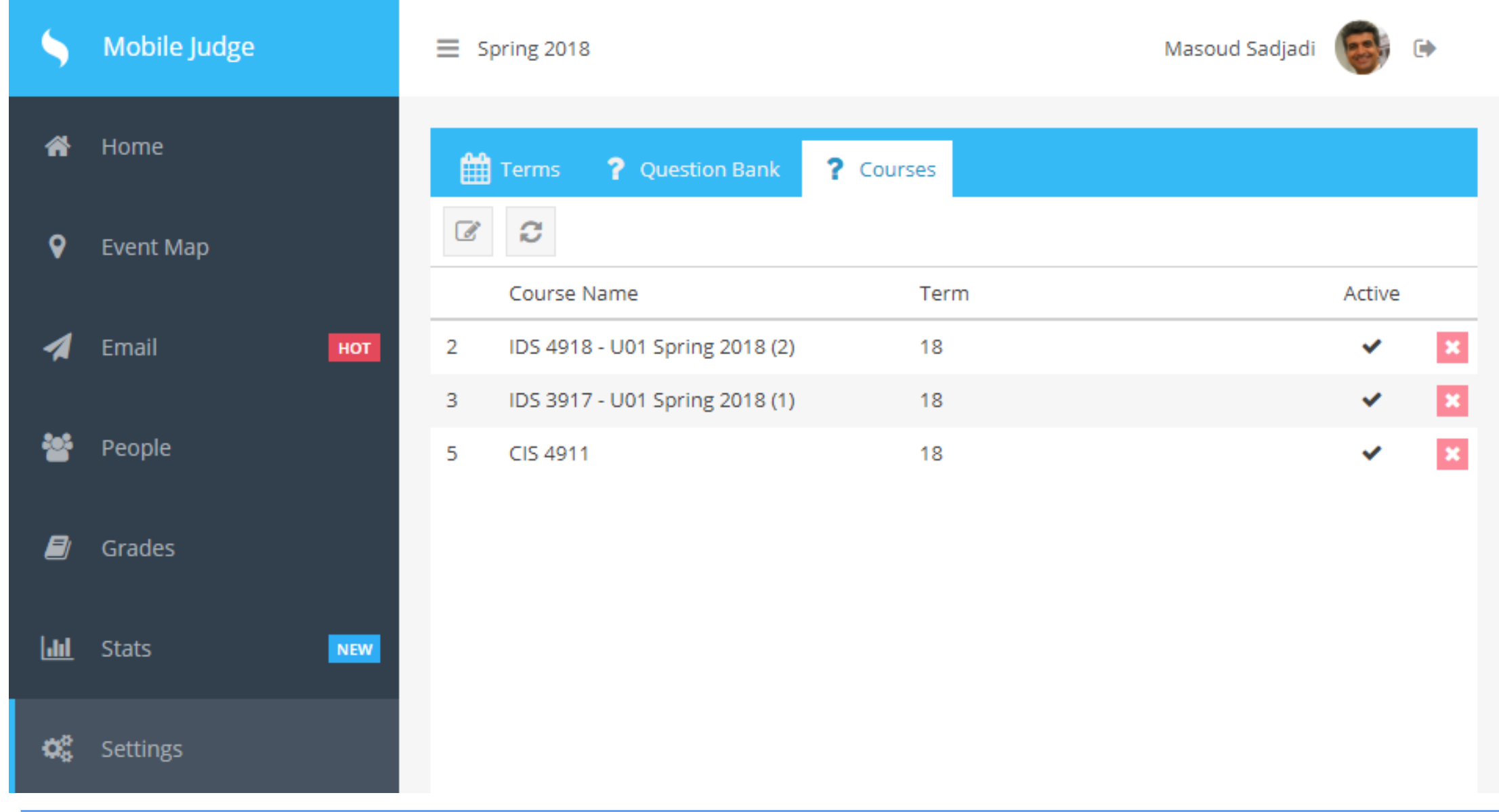
Judges:

- Register, view showcase map, comment on assigned students, and grade assigned students.

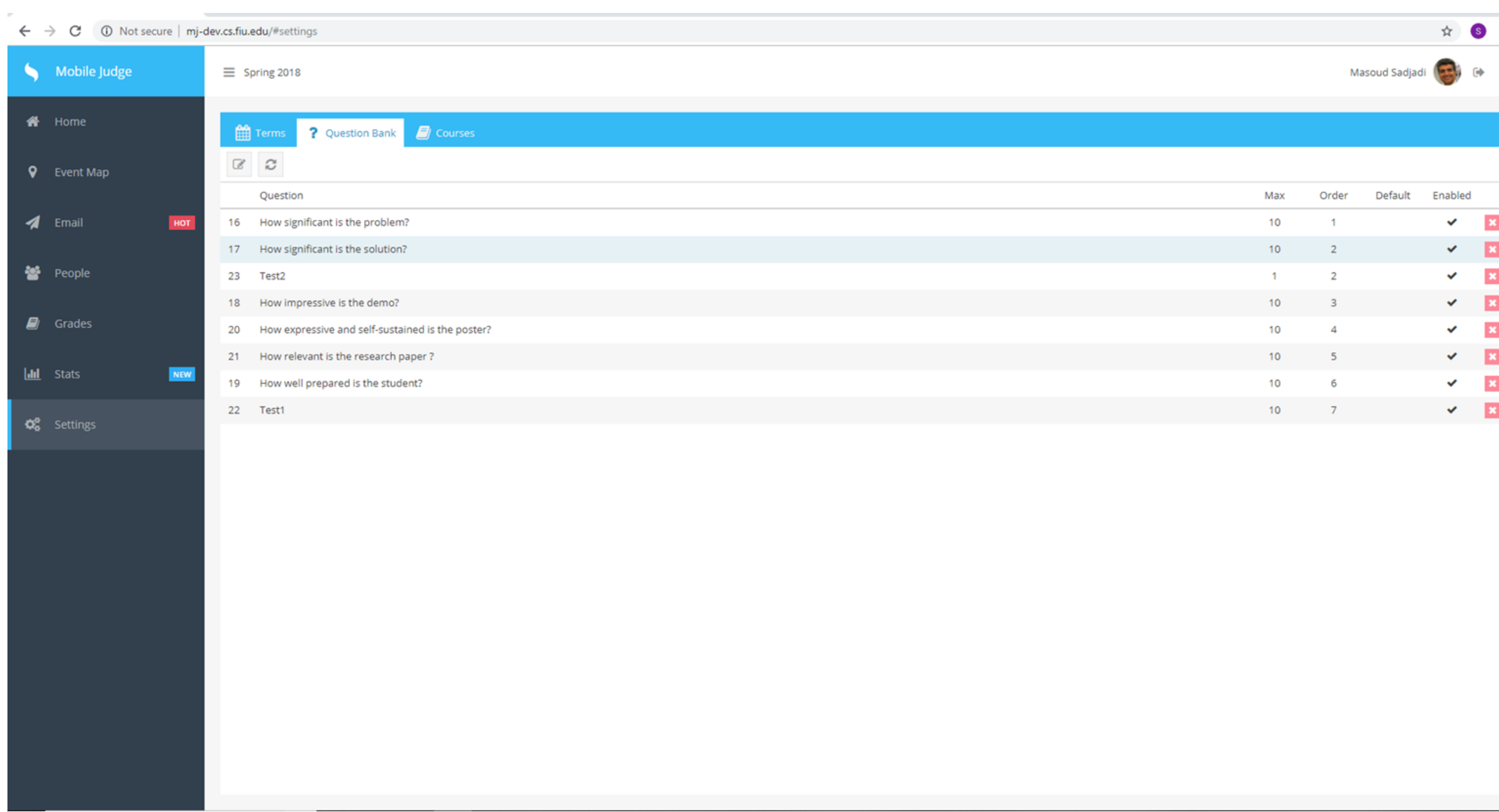
Students:

- View grades and their assigned showcase location.

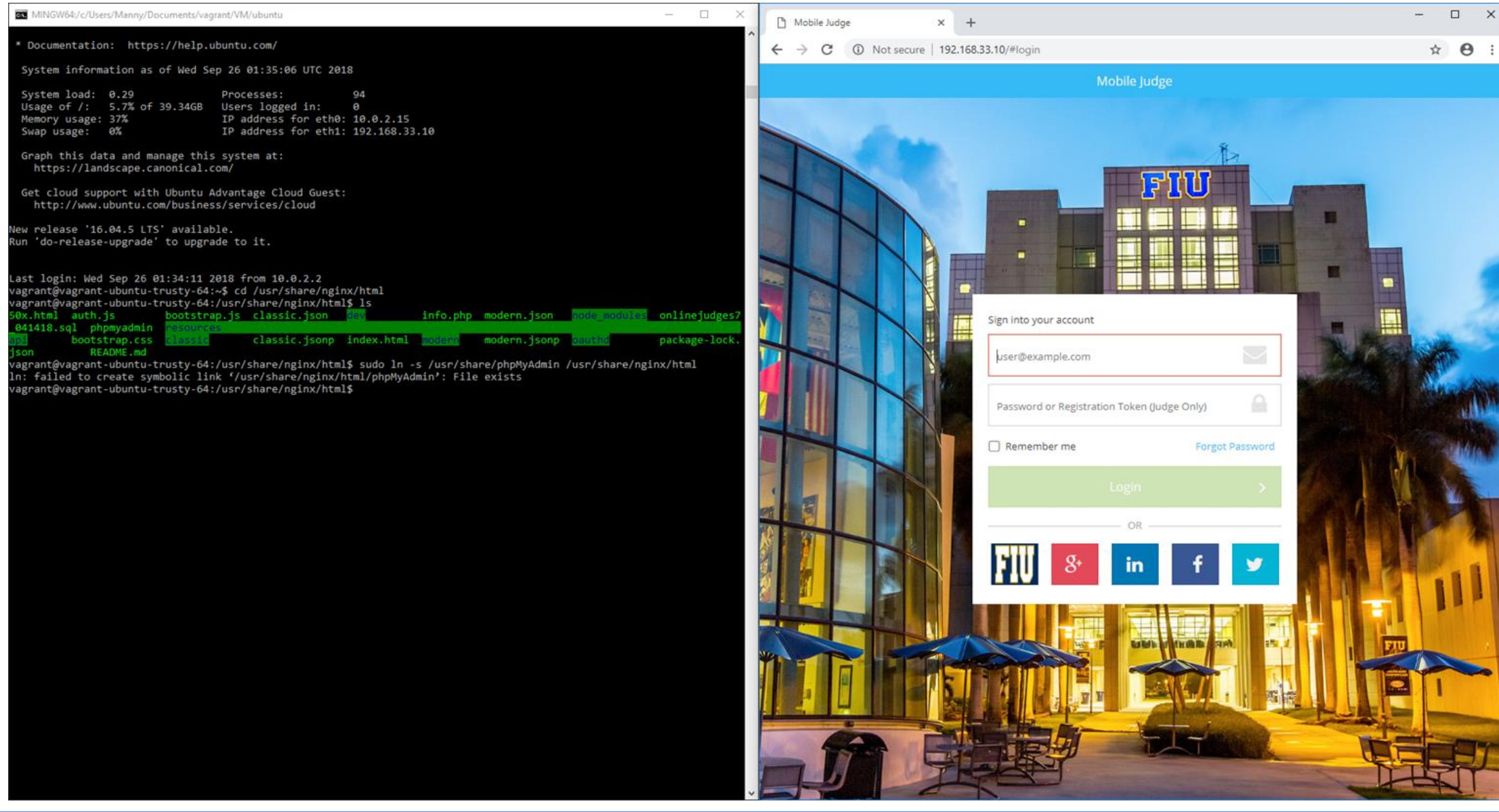
Screenshots



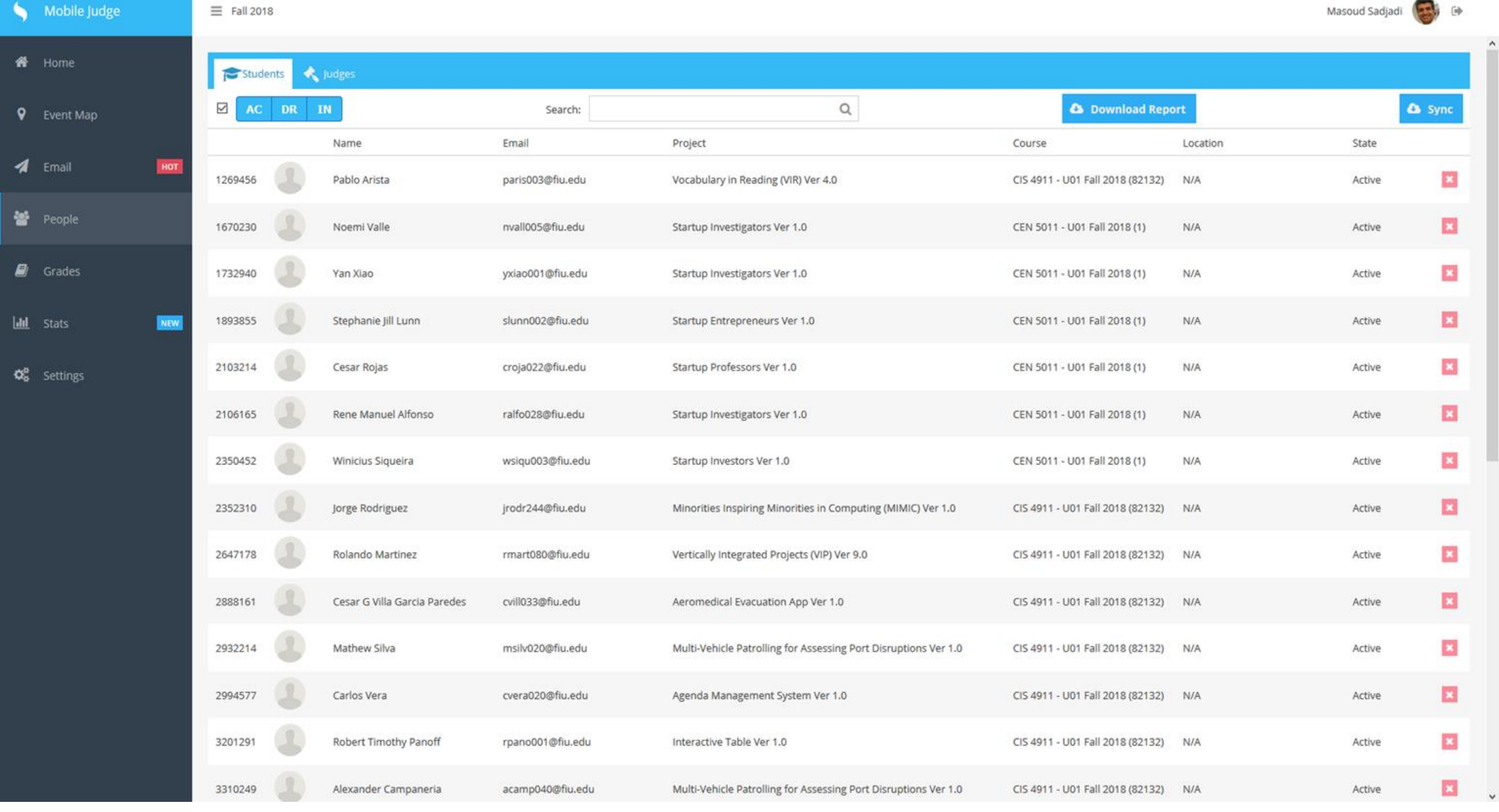
Clicking the Courses tab in the Settings menu will give you a list of available of courses and will allow you to perform CRUD operations.



Clicking the Questions tab in the Settings menu will return a grid of Judge questions and will allow you to perform CRUD operations.



This shows the deployment of a Vagrant environment and initiation of a Node server to set-up a local development environment.

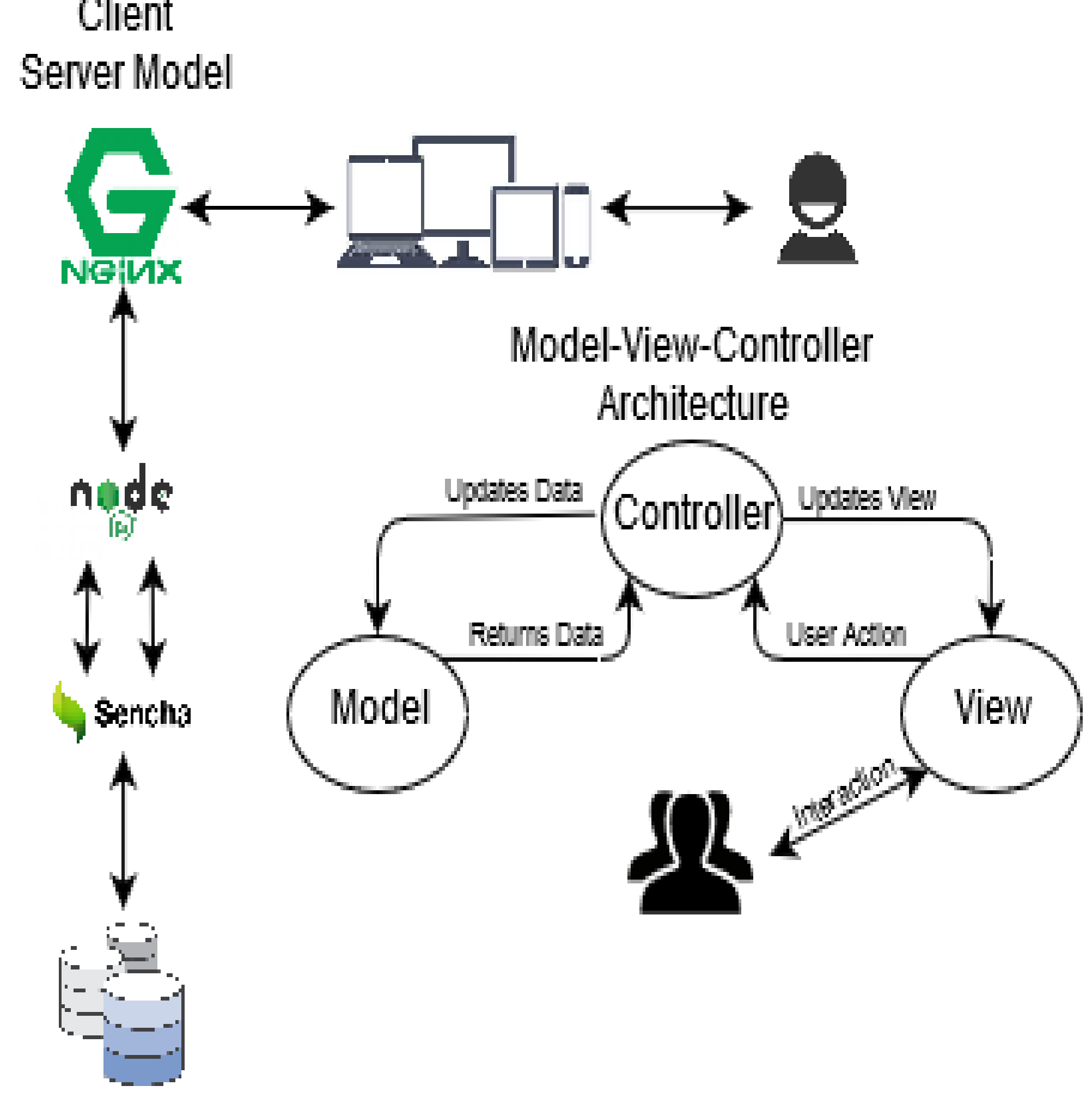


The result of clicking the Sync button triggers an event which will return student and project information retrieve from the VIP website via an API.

Requirements

- Create a view that displays Course information and allows update, inserts, and deletes.
- Enhance the Questions view.
- Fix the API Sync functionality from v10.1
- Create the backend infrastructure to allow questions to be assigned based on the course a student is enrolled in.
- Create a Vagrant box to streamline the deployment of a local development environment.

System Design



Implementation

- Sencha & ExtJS:** Used for cross-browser and cross-platform support, compiler, and framework classes for quick implementation.
- NodeJS:** Used for handling website routes and server requests.
- NGINX:** is used as the web server to run NodeJS and Oauth.
- Oauth:** is used for third-party user authentication.
- MySQL, PHPMyAdmin, MariaDB, & Redis:** Used as the application's database and database management system.
- PM2:** is used in the production server as our process management system and monitor.
- Vagrant:** Used to quickly deploy development environments with different configurations without having to create a completely new instance of a virtual machine.

Verification

- Test ID:** MJ-49 T1
- Description/Summary of Test:** An administrator clicks the Settings option and the Courses tab. The administrator wants to see all courses displayed with their corresponding Terms and be able to modify, delete, or add course.
- Pre-Conditions:** Course data must exist in the database.
- Expected Results:** Administrator is able to see Courses within the grid and is able to add, modify, and delete courses.
- Actual Result:** The administrative user was able to see all the data and add, modify, delete the records.
- Status:** PASS

Summary

Fall 2018 marks the 11th iteration of the Mobile Judge application. Mobile Judge continues to be an integral part of the Senior Project Showcase and thus a very important project in the senior project line-up. In version 11.0, my focus was on streamlining the deployment of the local development environment and setting up the infrastructure to allow the administrator to assign questions per courses. Throughout the semester many other features and issues took time and attention away from these goals. The challenges of learning the web technologies that make up this application, picking up where the other team left off, along with becoming familiar with the application's flow proved to be difficult but fulfilling experience.

Acknowledgement

The material presented in this poster is based upon the work supported by Dr. Masoud Sadjadi. I thank Dr. Sadjadi for assistance, cooperation and mentorship that I received throughout this process as well as my group members and fellow students Andrew Sanzetenea, Tobechukwu Ezewike, Pedro Ramos, and Carlos Larrauri.