

Course Evaluation System

Student: Gerardo Parra

Project Owners: Dr. *Nagarajan Prabakar, Eric Johnson*, Florida International University

Instructor: *Dr. Masoud Sadjadi*, Florida International University

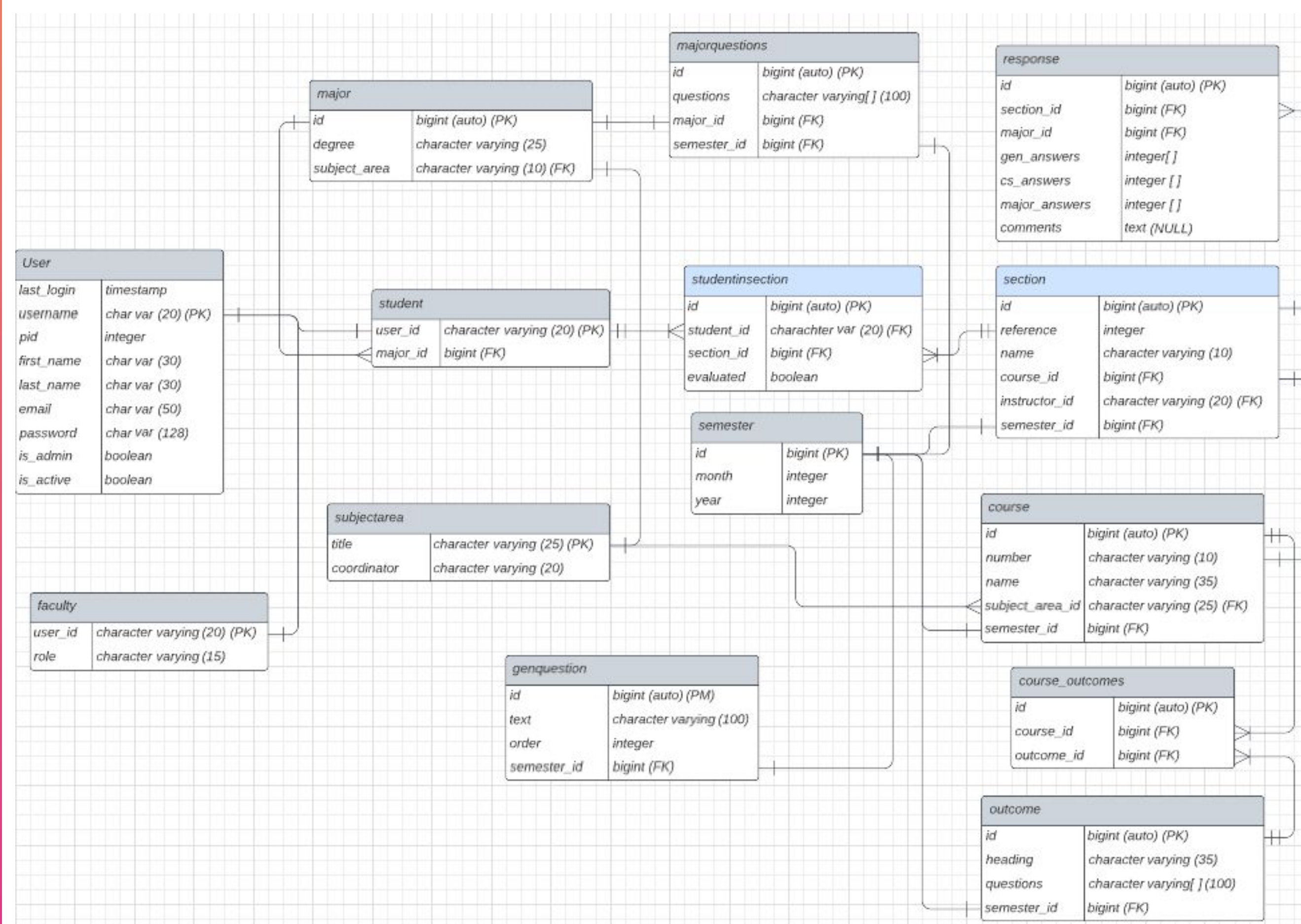
CURRENT SYSTEM

The Course Evaluation System (CES) is an online computer software system designed to conduct course evaluations at the end of every semester. The system provides students with a list of courses provided by the School of Computing and Information Sciences that are receiving evaluations and for which they are registered: after the evaluation period is over the system generates reports maintaining the participant's anonymity, and then it allows administrators to look at the generated reports.

NEW SYSTEM

The new system focuses on modularity in order to accelerate the creation of new features, the addition of new APIs to control individual aspects of the system and maintenance by future teams. By creating an individual database that holds all semester information, report generation is done dynamically rather than hard coding HTML pages and saving them for when a faculty member needs a report; this makes the system aware of any changes that could happen in current or previous semesters. Because the new system uses JSON to communicate to the database, frontend development is not tied to an individual framework and can evolve as the backend grows with new features.

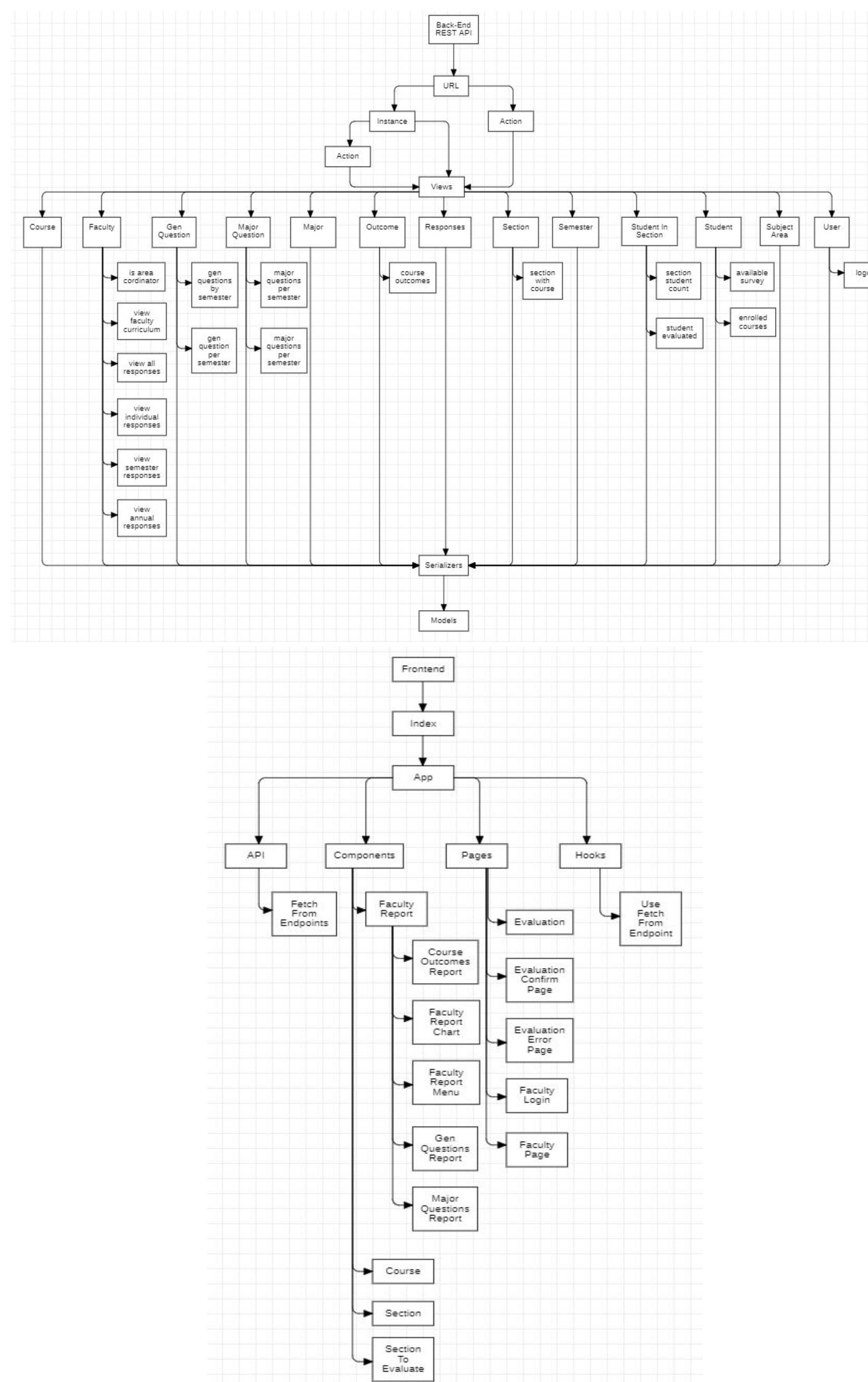
OBJECT DESIGN



PROBLEM

The Course Evaluation System (CES) was first developed in 2006 by the KFSCIS web development team and was used for more than fifteen years at the end of each term. This classic package was built on a Postgres database with several Perl modules to collect student course evaluation data and to generate reports. This package is not modular enough to classify the collected data according to student majors and to generate reports specific for each KFSCIS degree program.

SYSTEM DESIGN



REQUIREMENTS



IMPLEMENTATION



Django and the REST Framework were used in conjunction to create a web API that is modular, more secure, and can support multiple frontend architectures. With the excellent support from the REST Framework, we were able to integrate PostgreSQL as our main database. We used React to develop our frontend because it is flexible, and allows for fast development iterations that were required when modifying the backend. We used GitHub as our version control system.

INDIVIDUAL CONTRIBUTIONS

As the team lead, I wrote tutorials, reading materials, and python tutorials to start developing. I arranged meetings, created Trello Boards to keep track of goals and created our Discord server to communicate with the team. I submitted health reports weekly.

As a developer, I setup a VPS with a PostgreSQL instance so that we could all use it for developing; I created the initial backend framework with sample models, views and serializers for the team to learn; I designed and implemented the database schema with the help of Shaoting and Daniel and implemented the initial required models; implemented the User, UserManager, and Student models and respective serializers and views; created and contributed to multiple actions needed on the Faculty view such as `is_area_coordinator`, `view_faculty_curriculum` and `view_individual_responses`.

In the frontend, I implemented the Faculty page and all its logic and created helper hooks to aid us when communicating to the backend. I configured and deployed the app in the server provided by Eric Johnson.

I created the User Manual and Installation and Maintenance video. I wrote the System Architecture portion of the Documentation, and helped Daniel create the diagrams needed on the System Architecture.