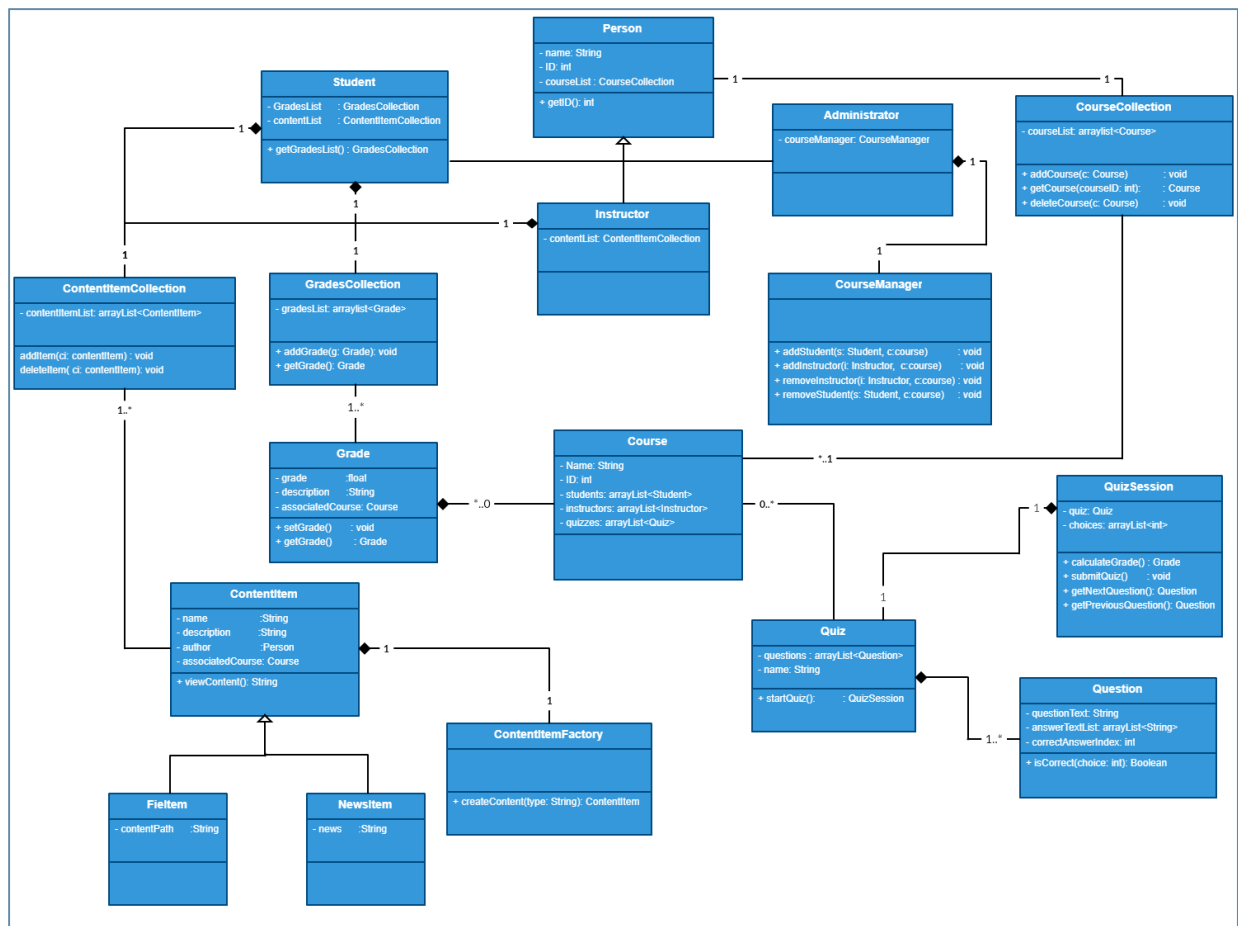


## Project Part 3: Progress Report

1. **Project:** Team 42: Erik Holbrook (github:zephyr1999) and Ahmed Al Hasani (github:AhmedAlHasani)

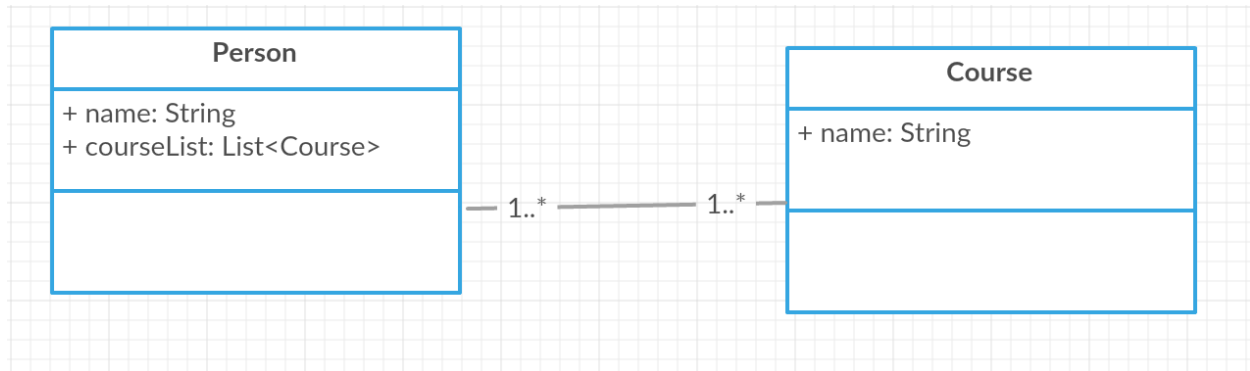
Schooodle is a lightweight, web-based college course management application. It allows students to view grades and course content, and complete assignments. Instructors can upload content and assign grades. Administrators manage enrollment of students, creation of courses, and assigning instructors to courses.

2. **Previous Class Diagram:** Updated with fixes/corrections based on feedback from Project Part 2:



See Appendix A for class diagram from part 2, included for your convenience.

## 3. Completed Class Diagram



**4. Summary:** Our team has spent the last two weeks focused heavily on improving the design of our project with feedback from part 2 (see updated class diagram in part 2). Relationships between classes and composition has been completely redone; the design goals remain the same. Literally zero classes from our original design remained untouched. We have added over a dozen new classes, and rearranged the structure of the data entirely.

Besides redoing the class architecture, we have focused on initializing the Django framework for our application. The intent is for Schoodle to be a lightweight web application, and the progress of integrating Django to our object-oriented design has been tough going. Object-oriented-ness is an afterthought in the Django framework, so implementing the design has required in-depth study of the Django Framework.

Only 2 classes are given in the completed class diagram – the bulk of the work has not been on implementing the classes but rather setting up the Django project properly.

## 5. Breakdown:

Erik's contributions:

- Install Django
- Configure Django models settings
- Configure Django administration settings
- Initialize SQLite database with test objects
- Connect Django Views to models
- Configure URL templates in Django
- Write Person and Course class wireframes
- Write initial html templates for displaying information

Ahmed's contributions:

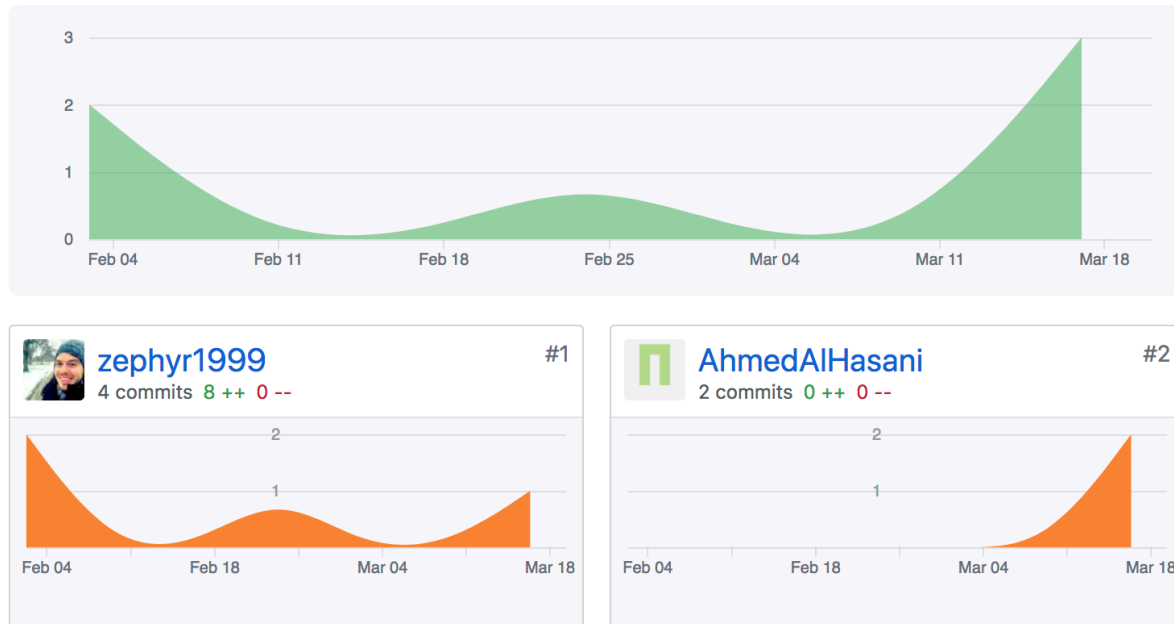
- Completely redraw class diagram from the ground up
- Rework main classes (Person, Course)
- Redesign quiz to not inherit from ContentItem
- Integrate QuizSession class with Quiz to control access to Quiz
- Create collections classes to encapsulate lists of Grades, Courses, and ContentItems
- Refactor Administrator class and design courseManager class

## 6. Github Graph

Feb 4, 2018 – Mar 21, 2018

Contributions: **Commits** ▼

Contributions to master, excluding merge commits



## 7. Estimate remaining effort:

There are two primary areas of work to be done now that the design is improved and the Django project initialized. First we must continue implementing the class structure from the class diagram above. Most of the classes are unimplemented, though the design is mostly set, so this is just a translation of the class diagram into Python Code. This is roughly 20 classes.

The other area remaining is front-end implementation. Currently, there are only wireframe templates. Many small bits of html remain; these will comprise the bulk of the user interface. This effort will be dynamic in nature and depend heavily on the progress of backend implementation. We estimate this will be the most time-consuming portion of the project, as is often the case with web-design. Wireframes and simple html will suffice though development of the backend, but a refined look will take time.

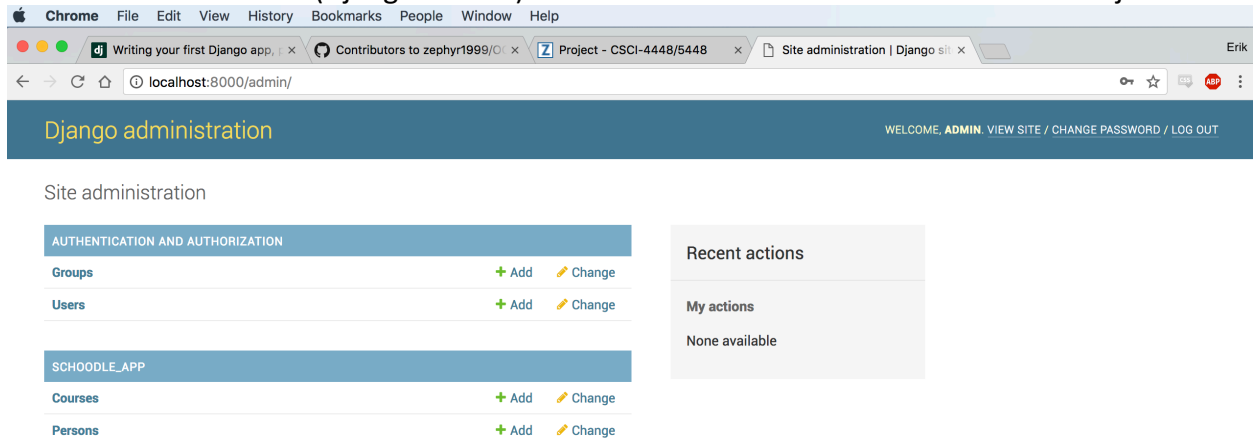
## BONUS Screenshots of current implementation

The output of the Django server (from terminal log)

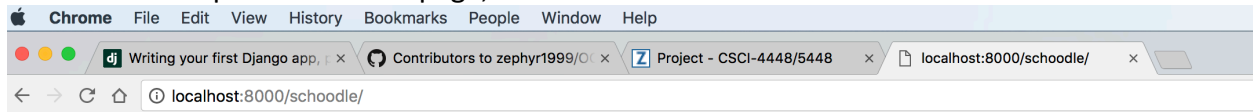
```
[Eriks-MacBook-Pro:00D_schoodle erikholbrook$ cd Schoodle/
[Eriks-MacBook-Pro:Schoodle erikholbrook$ python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
March 21, 2018 - 15:24:48
Django version 2.0.3, using settings 'Schoodle.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
█
```

The administration site (Django default) that allows creation of Person and Course objects:



Placeholder templates for homepage, Person and Course Detail:

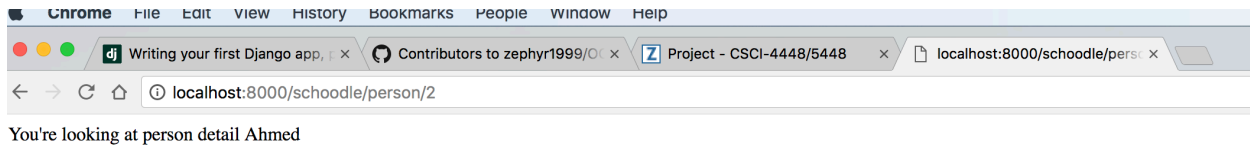
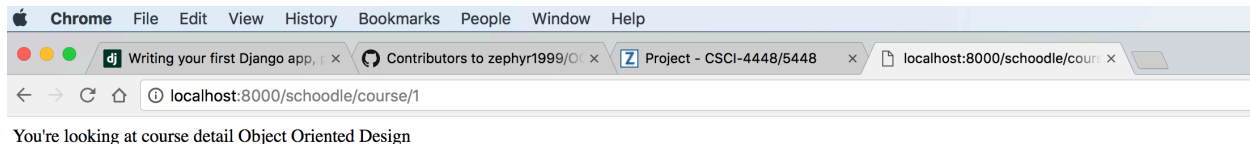


**Index: People**

- [Ahmed](#)
- [Erik](#)

**Index: courses**

- [Object Oriented Design](#)



Note the correct URL routing in the screenshots.

## 8. Next Iteration

The primary goal for the next iteration is a completed backend, i.e. all classes in the class diagram are completed and functional. Our focus is on functionality over design, thus once the backend is complete, we will focus simple implementations of the use cases and not the interface. This way, we will have as much time as possible to devote solely to interface design and “pretty-ing up” the application, without adding any more behavior.

## Appendix A:

For referencing fixes in the class diagram above, see below for the class diagram as submitted in part 2.

