Team Name: WIP: Work-in-Progress
Project Name: Tibl
Spring 2018
Github Project Link: https://github.com/jagath-jaikumar/326-Project

Team Overview

Team Member Names	Github Usernames
Adam Rivelli	AdamRivelli
Ali Grace	aliaidengrace
Jagath Jai Kumar	jagath-jaikumar
Jeremy Kelleher	Jeremy6462
Sagar Arora	arorasagar

Overview

Our goal for the semester is to build a social networking application for students to use to develop a community within each classroom. TIBL, a play on the acronym for Team Based Learning (TBL), is a space where students can subscribe and contribute to a twitter-like news feed for every class they are enrolled in. In the class space, they can post short messages to the whole class, and see who else is enrolled in that particular section. In addition, a student user can see what classes other students have taken, and easily identify which classes have overlapped with the user - either currently, or in the past. They idea for our app has remained largely the same from our Project 1 Idea submission, but in the last submission we decided to remove the direct messaging element of our project.
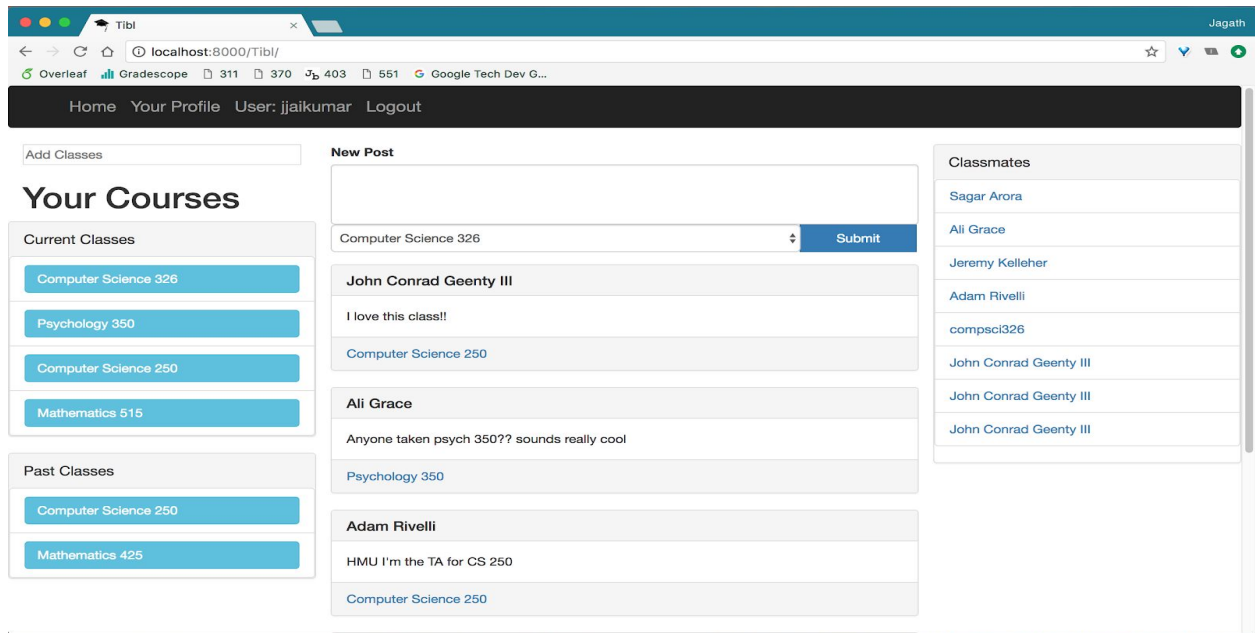
Design Overview

The design of the home page of our app comes a combination of the Twitter news feed and the Facebook home page. We have a navigation bar at the top of every page that allows users to either go to the home page or go to their personal profile page. From the home page, users can see what classes they are enrolled in on the left sidebar. In the middle of the page, they can see the main news feed, and on the right, they can see all of their classmates names. The user is also able to search for specific classes using the search bar in the top left. If the user clicks on a classmate, the user will view that classmate's profile page, and will be able to see which classes that classmate has taken previously and which classes he or she is currently enrolled in. Finally, every class section has a page that shows the class title, number, instructor, and the list of currently enrolled students. In addition to these design elements, we also have a functioning user registration page and working login/logout views.
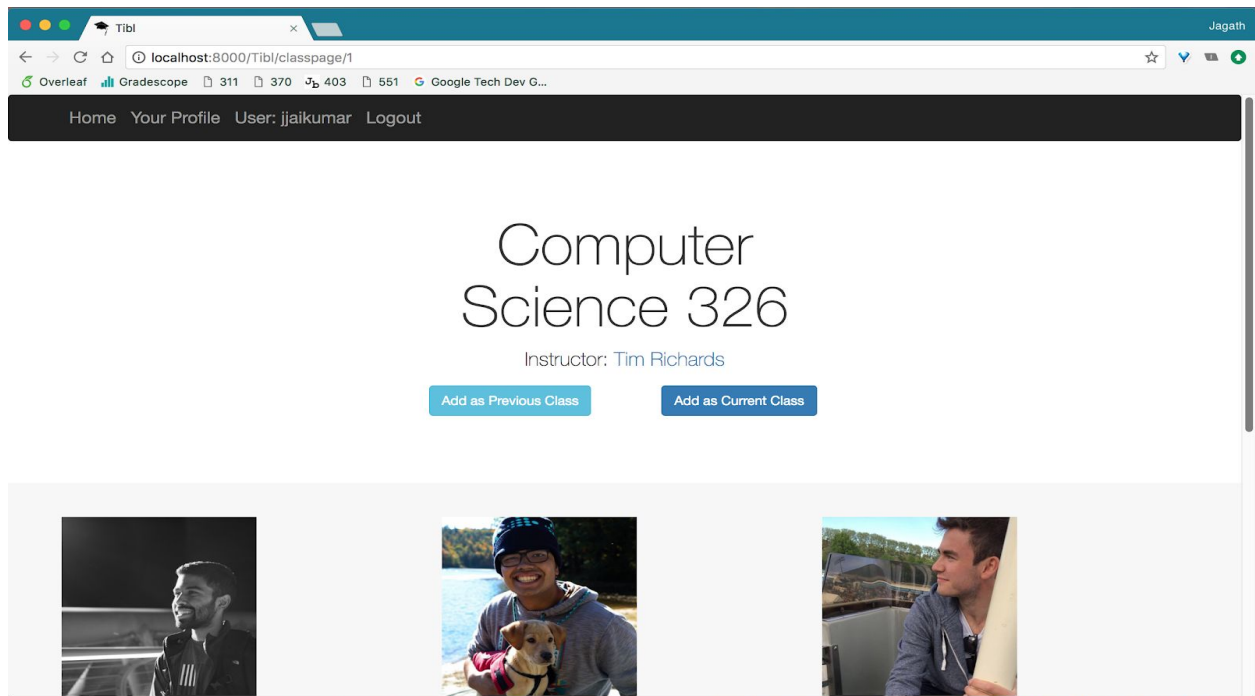
User Interface

Table describing all views in our application.

View	Description
Homepage	The main landing page for our application. From here, users can navigate to the classes they're in and to the profiles of their classmates. Users make posts from this page, and can view posts for all of the classes they're in. Users can also search for a new class to add from the homepage.
Classpage	The classpage displays all of the information related to a class, including its professor and all of the students in the class. Users can add a class from this page as well.
Profile	From the profile page, users can see what classes that person is currently in and which classes they've taken in the past.
Messaging	Messaging is not currently implemented besides the view. In its current state, it does not have any functionality, but it does display all of the messages that you have with anyone. However, there is not any way besides through the admin console to send new messages.
Login	This is a login page for our website. It has text boxes for users to input their username and password. It also has a link to a register page for new users.
Register	This is the page that allows people to create an account.

The following are screenshots of each of our views.
Homepage

Classpage

Profile

Login

Registration

Overleaf | Gradescope | 311 | 370 | J₅ 403 | 551 | G Google Tech Dev G...

Home   Your Profile   Login

Username: [   ]   Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: [   ]

- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation: [   ]   Enter the same password as before, for verification.
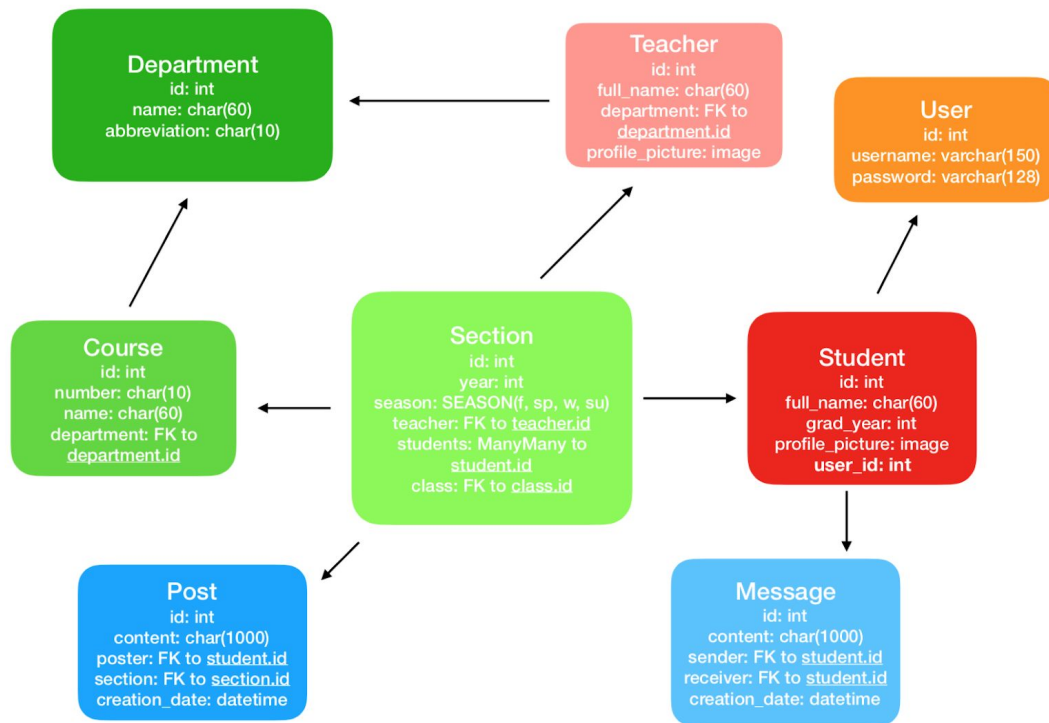
Full name: [   ]

Grad year: [   ]

Profile picture:

[Choose File] No file chosen

[Submit]

WIP: Work-in-Progress

Data Model

URL Routes

Route	Description
/	Homepage
/classpage/<int:num_>	Visit a specific class page
/messaging	Shows all your private messages (not fully implemented for final submission)
/myprofile	Shows your current profile
/friendprofile/<int:profile_id>	shows a specific friend's profile
/register **	Creates a new user
/classpage/<int:num_>/register	Register a course for this current semester
/classpage/<int:num_>/register_previous	Register a course for a previous semester
/class_search	Search for a specific course

** - denotes Team Choice element route

Authentication and Authorization

Our application implements the default Django User classes and functionalities. All users have the same permissions, with the exception of the Django superuser, who has access to the admin site. As an extension of this project, it would be smart to add a moderator permission level, because there currently isn't any way of adding a class to the website except through the admin page. It would make sense for trusted users to be able to add a class when the new semester starts.

Team Choice

We have discussed several options for our Team Choice element, and decided on one main Team Choice feature, and one feature we needed to develop to improve the functionality of our application. Our main Team Choice element is a new user registration page, where students will be routed to if they do not have an account set up with our app. We developed this using Django's built in registration functionality. The /register route is specifically for our main team choice element.

In addition to the user registration page, we also developed a search bar for the home page that students can use to search for specific classes. While this search bar is functional, we ultimately decided on the user registration as our team choice because we had developed it more.

Conclusion

Working together on this project was overall a great experience. We found success in dividing tasks amongst the five of us by using the Git Issues tab as a checklist to monitor our progress. Because some members of our team were inexperienced with using Git, we had to work collaboratively to resolve conflicts and handle different issues. Finally, we all gained experience in working with Django and implementing views, routes, and data models.