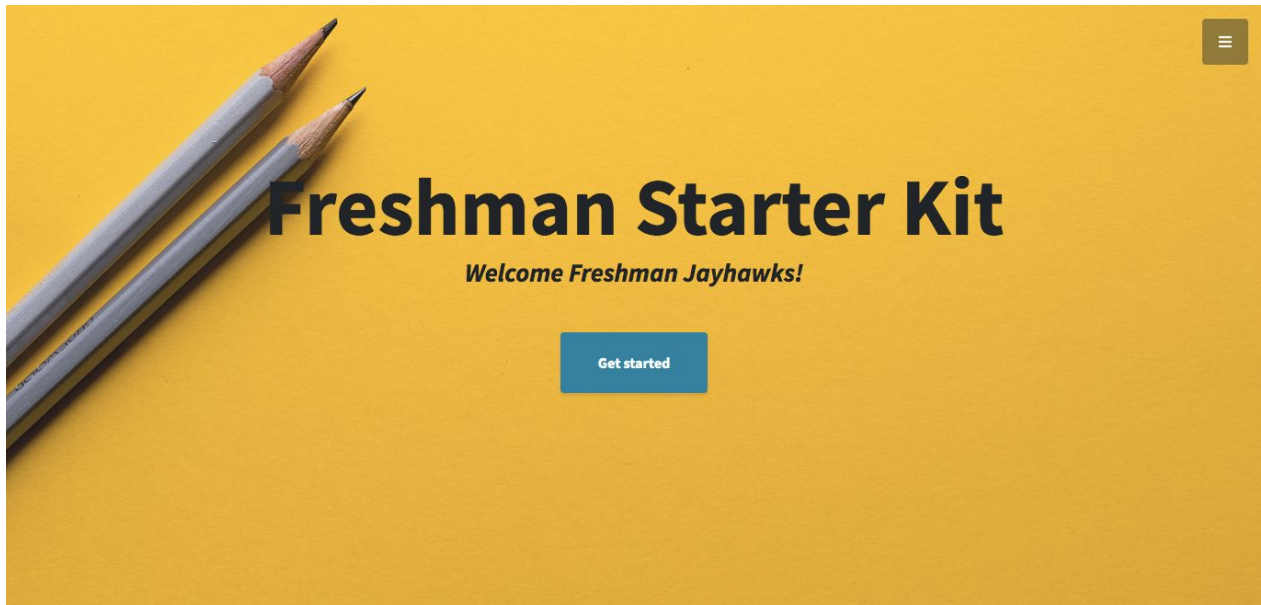


EECS 448 Project 4

Freshman Starter Kit, a Django Webapp



Codehers

Victoria Maldonado

Anjali Pare

Archana Ramakrishnan

Sindhu Shakamuri

Table of Contents

Product Backlog: Completed Featured	3
Product Backlog: Potential Future Additions	4
Standard Documentation	4
UML Diagrams	7
Use Case Diagram	7
State Diagram	8
Class diagram	9
Meeting Log	9
Gantt Chart	12
Bug List	12
Documentation: Division of Labor	13
Sources	13

Artifacts

Product Backlog: Completed Featured

1. Rate my class

- a. Upperclassmen rate EECS classes to help freshman decide their course load
 - i. **Task difficulty level:** Medium
 - ii. **Time and effort:** 1 week
- b. Will need a ratings form and a page to view all of the ratings in blog form
 - i. **Task difficulty level:** Medium
 - ii. **Time and effort:** 1 week
- c. Filter the ratings based on the class number or view all classes
 - i. **Task difficulty level:** Easy
 - ii. **Time and effort:** 2 days

2. Advice

- a. Upperclassmen can upload blog posts of advice (use tutorials from DjangoGirls workshop blog project to learn)
 - i. **Task difficulty level:** Medium
 - ii. **Time and effort:** 1 week
- b. Can like and view posts from others
 - i. **Task difficulty level:** Hard
 - ii. **Time and effort:**
 - 2 weeks.
 - It was difficult to get the “like” functionality to work without creating another html landing page. This is why the implementation took longer.

3. KU Maps

- a. Calculate shortest distance, walk time
 - i. **Task difficulty level:** Medium
 - ii. **Time and effort**
 - 1 week
 - Use haversine function
 - Make sure to add haversine to requirements.txt to avoid issues when publicly hosting the website
- b. Database and model object with location coordinates of buildings - use CSV, calculate the shortest distance between the two location coordinates, use average walking speed to determine walking time
 - i. **Task difficulty level:** Hard
 - ii. **Time and effort**
 - 2-3 weeks

- Get assistance on how to populate a drop down menu with existing fields from the Building model, and to send the calculated distance, times and building names to the output html page.
 - c. Iframe with Google Maps of KU campus
 - i. **Task difficulty level:** Easy
 - ii. **Time and effort:** 1 day
- 4. Time Management
 - a. Reminders to take break depending on how long they have been active on the website. "Enter Website" button on homepage triggers timer to start.
 - i. **Task difficulty level:** Medium
 - ii. **Time and effort:** 1 week
 - Getting the time to display on another page with the start of button triggered. Having alerts display anywhere on the website for every 10 minutes the user has been active on the page.
 - b. Track time spent on website using Django/Python built-in methods
 - i. **Task difficulty level:** Hard
 - ii. **Time and effort:** 2 weeks
 - Getting the time to not refresh when entering another page, and continuing the timer until "get started" is clicked.

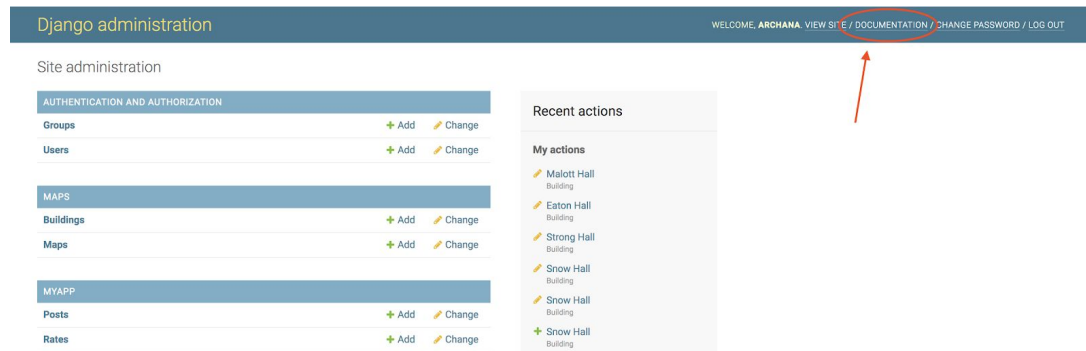
Product Backlog: Potential Future Additions

- 5. Login
 - a. Create account to navigate the website
 - b. Use Django authentication system and create user objects
 - c. Ambitious goal- give role based access based on if it is a freshman, upperclassman or admin
- 6. Advice
 - a. Upload videos
 - b. Because we couldn't do the login: gain beak - em - bucks or money for your student organization if your videos get more likes/comments
- 7. Google Maps API
 - a. To provide instant navigation based on the chosen classes

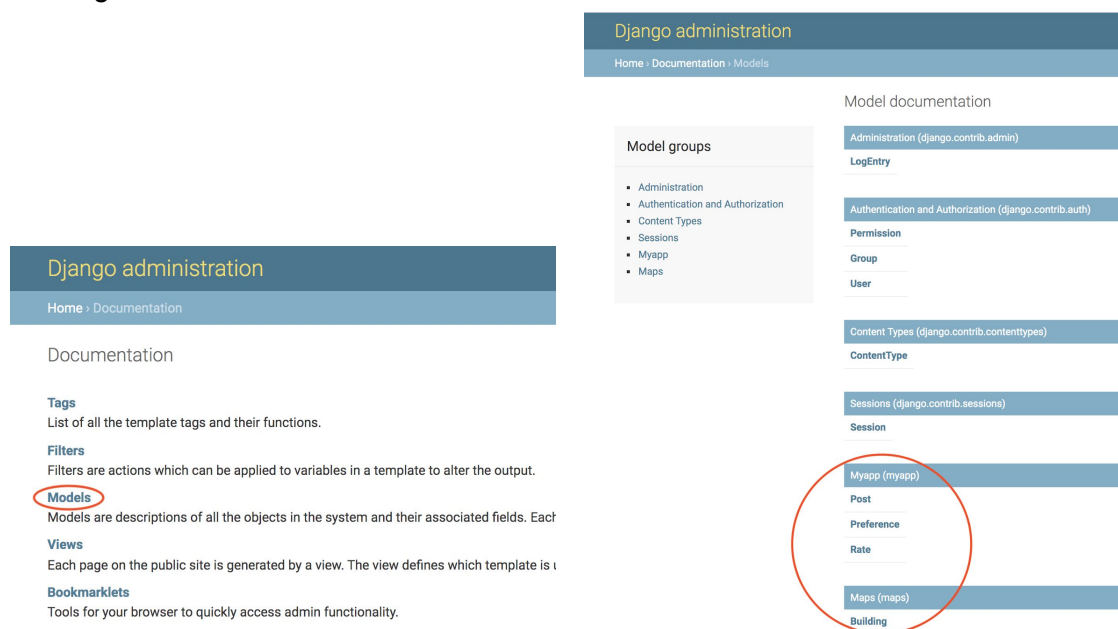
Standard Documentation

To view the standard documentation, navigate to <https://codehers.pythonanywhere.com/admin/>

Username: archana | **Password:** codehers



Click on 'Models'. This will take you to a page with all the 'Models', which are classes that enforce object oriented programming in the Django framework. The Models like Post, Rate, Buildings are ones we created.



The standard documentation for each model can be viewed by clicking on it. For example, Rate:

Django administration	
Home > Documentation > Models > myapp.Rate	
myapp.Rate	
A model that stores information needed to rate a class	
Fields	
FIELD	TYPE
class_comments	String (up to 300)
class_difficulty_level	String (up to 20)
class_exams_num	Integer
class_hours_spent	String (up to 20)
class_hw	String (up to 50)
class_overall	String (up to 20)
class_rated	String (up to 20)
id	Integer
rater_grade	String (up to 20)

If you go back to the Documentation page and click on 'Views', it will take you to a page with all the functions created that trigger the redirects to the needed pages within the webapp. These functions also parse and calculate the information required to be shown as output.

codehers.pythonanywhere.com/admin/doc/views/#ns	
/admin/doc/models/<app_label>\-<model_name>/	View function: django.contrib.admin.views.ModelDetailView. Name:
/admin/doc/tags/	View function: django.contrib.admin.views.TemplateTagIndexView.
/admin/doc/templates/<path:template>/	View function: django.contrib.admin.views.TemplateDetailView. Name:
/admin/doc/views/	View function: django.contrib.admin.views.ViewIndexView. Name: dj
/admin/doc/views/<view>/	View function: django.contrib.admin.views.ViewDetailView. Name: d
/advice.html	View function: myapp.views.advice. Name: .
/maps/<id>/	View function: maps.views.distance_output. Name: .
/post_new.html	View function: myapp.views.post_new. Name: post_new.
/rating_form.html	View function: myapp.views.rating_form. Name: .
/ratings_landing_page.html	View function: myapp.views.ratings_landing_page. Name: .
/ratings_view_class.html	View function: myapp.views.ratings_view_class. Name: .
/static/<path>	View function: django.contrib.staticfiles.views.serve. Name: .
/time.html	View function: myapp.views.time. Name: time.
/view_advice.html	View function: myapp.views.view_advice. Name: post_list.

The circled views are ones we created for the functions to navigate between pages, by matching URL patterns using regular expressions. For example, distance_output calculates the necessary information and redirects to map.html

maps.views.distance_output

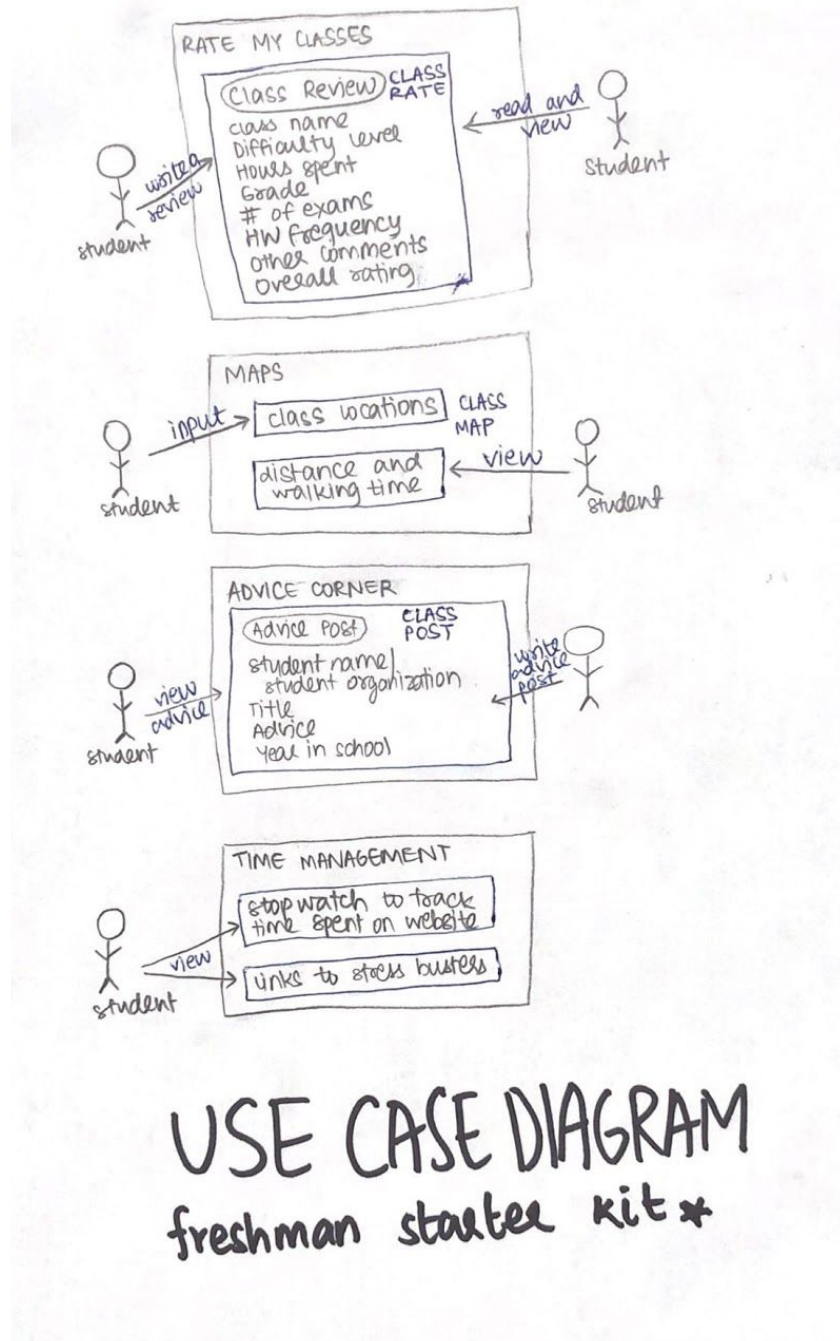
Directs to page with calculated distances and walking times, along with a google maps iframe of the KU campus Template:

maps/map.html

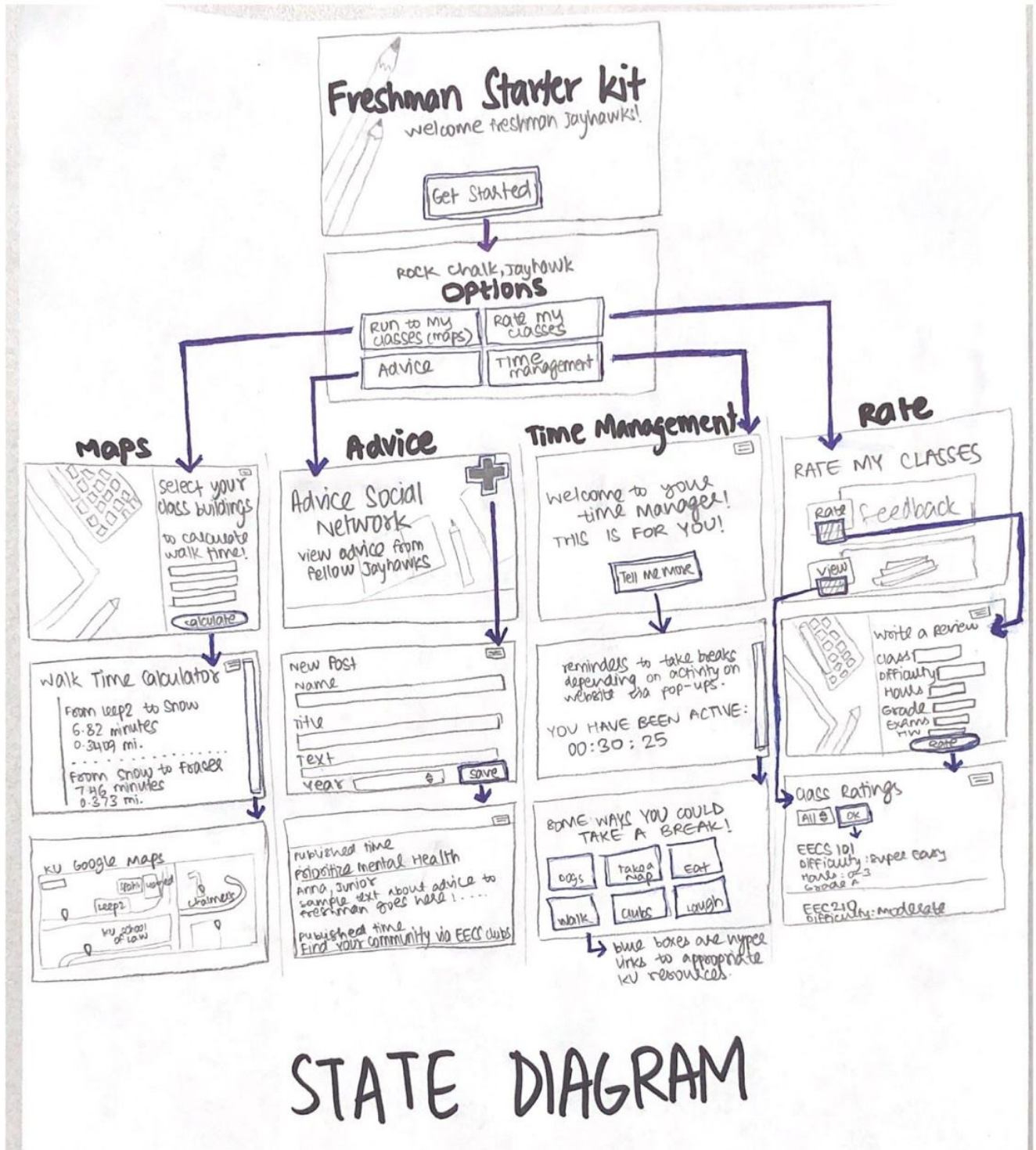
[Back to View documentation](#)

UML Diagrams

Use Case Diagram

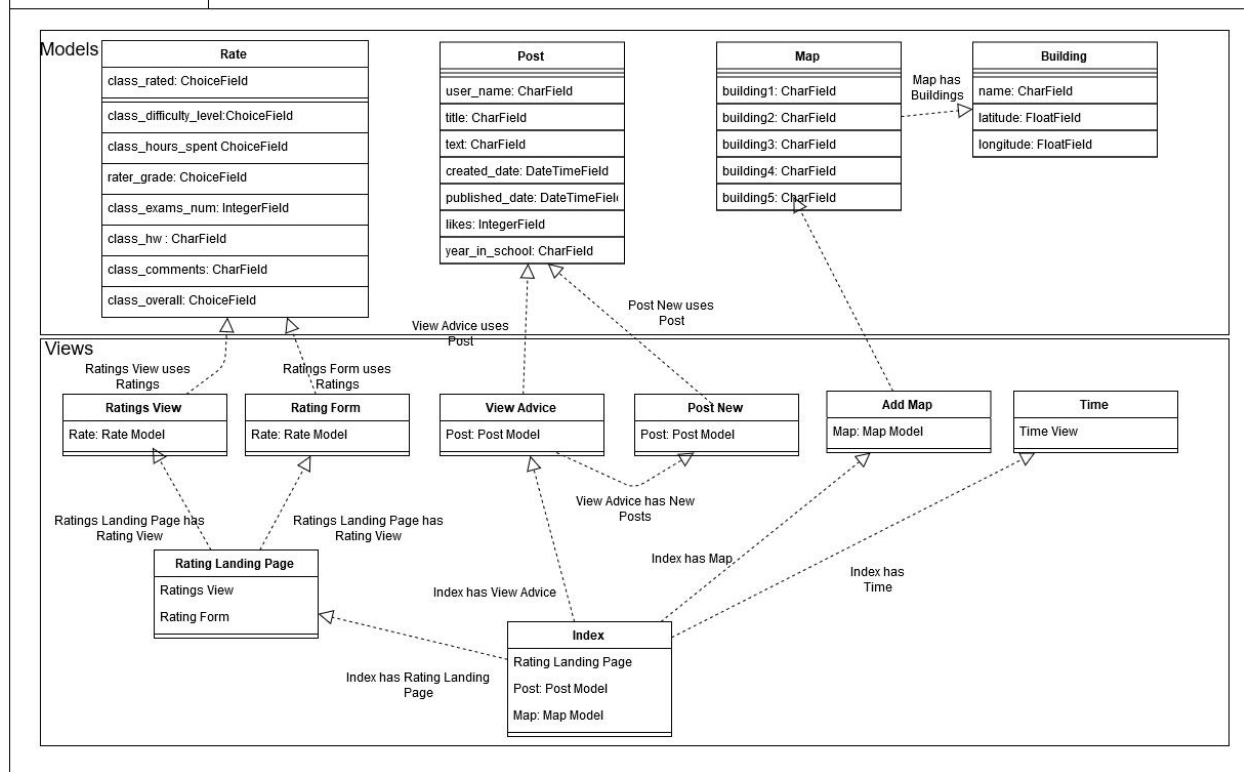


State Diagram



Class diagram

Freshman starter kit



Meeting Log

Sunday 1st December (10 am - 2 pm)

Leap 2

- As a group, we continued to update the UI and finish documentation.
- Got rid of minor bugs in some pages of the website (Sindhur)

Tuesday 26th November (6 pm to 9 pm)

Leap 2

- As a group, we continued to update the UI
- Started working on the Bug list and updated Meeting log (Anjali)
- Got rid of dead code in the files (Archana)
- Worked on the Class Diagram (Victoria)
- Publically deployed website (Archana)
- Continued to work on the test suite (Victoria)

Monday 25th November (4 pm to 7 pm)

Leep 2

- Made a decision as a group to not go forward with the “like” functionality for the posts.
- Continued working on the test suite (Anjali and Archana)
- Finished the menu bars for all the webpages (Sindhu)
- Spent time getting devel up to date with every group members work (Victoria)
- As a group, we updated the UI
- Completed the State and Use Case Diagram (Archana)

Saturday 23rd November (4 pm to 7 pm)

Leep 2

- Started working on the Maintenance and Deployment plan (Anjali and Victoria)
- Tried to get the “like” functionality for the posts to work (Anjali with help from Archana)
- Continued to work on setting the menu bars for all the pages (Sindhu)
- Spent about an hour merging all the files and specifically merging the “Maps” option into the developer branch (Victoria)
- Started working on the test suite for Maps (Archana)

Friday 22nd November (4 pm - 8 pm)

Leep 2

- Started working on documentation for Django, meeting log, spoke about and distributed documentation among group members.
- Looked into adding a Like feature for the posts (Anjali, Archana)
- Got the menu bar working on a couple of pages and converted the time counter from milliseconds to seconds. (Sindhu)
- Fixed minor bugs with rate my class page (Victoria)

Wednesday 20th November (9:00 am - 11:00 am)

Spahr Library

- Solved issues with the database table not found (Anjali, Victoria, Archana). Git no longer complains :)
- Worked on menu bar for all the pages (Sindhu)

Monday 18th November (4:00 pm - 5:45 pm)

Leep 2

- Continued to resolve git merge conflicts and database “not found”, could not fix :(

Monday 18th November (9:00 am - 11:00 am)

Spahr

- Tried to merge our individual progress and started to resolve merge conflict issues with git
- Finished rate my class form (Victoria)

- Finished map-distance calculator (Archana)
- Made progress with the time manager (Sindhu)
- Fix formatting bugs for better UI

Monday 11th November (4:00 pm - 5:45 pm)

Leep 2

- Got the post/view advice working (Anjali)
- Made progress with the map-distance calculator, time manager, rate my class (Archana, Victoria, Sindhu)

Sunday 3rd November (4:00 pm - 7:00 pm)

Eaton

- Completed UML Class Design
- Finished Documentation using Admin docs

Wednesday 30th October (9:00 am to 11:00 am)

Eaton

- Put together documentation pieces
- Clean up home page
- Division of labor-task assignments, and deciding the scope of the project

Monday 28th October (4:00 pm - 6:00 pm)

Spahr Library

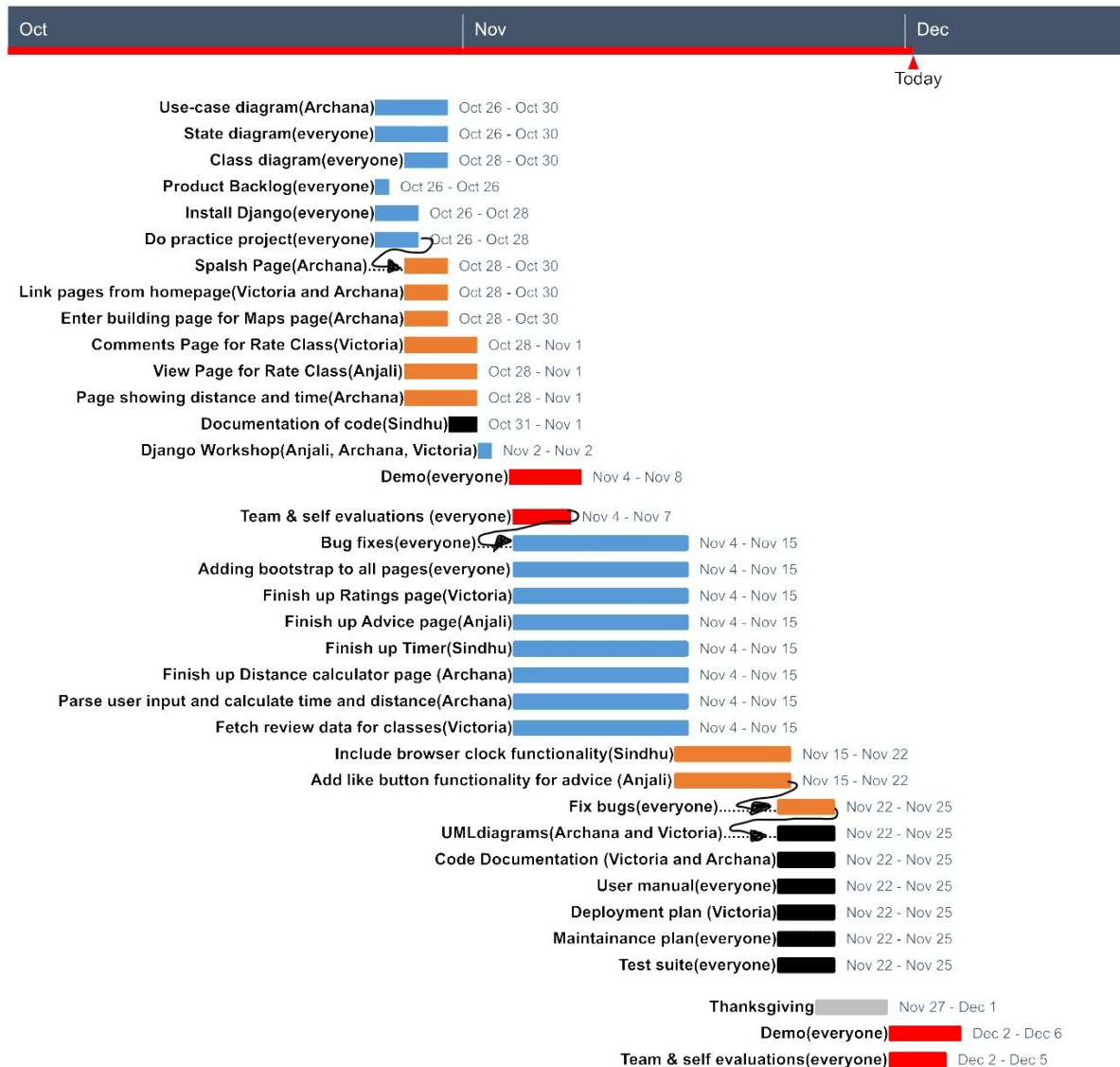
- Reinstalling django and virtual environment to troubleshoot
- Admindocs- documentation generator (install docutils)
- Linking other landing pages from the homepage

Saturday 26th October (1:00 - 4:00 pm)

Spahr Library

- work on planning documentation
- UML for Web UI
- Gantt Chart
- Django installation and tutorial

Gantt Chart



Bug List

1. Timer keeps going unless you hit "Get started". Once you hit "Get started," it will start again, but will not stop. Even when you open another browser window, it keeps going. Even when the laptop is switched off, it just keeps going. That is something we will have to look into. However, even if it keeps going, you will not receive a "time spent" notification unless you visit the page again. This ensures that this website does not come in the way of any other work you are doing.

2. Since we do not have a user login, anyone can create posts and add comments about the classes as well. Anyone can view the posts and the class ratings. Had a user login been made, we would be able to restrict access to all the pages to KU students only. This would get rid of most of the security issues. Although, there is still a need to have someone take a look at the posts before making them publicly available for university students.
3. The font styles and sizes are not consistent across all of the sidebar navigation. Implementing the functionality and positioning it in the right spot was in itself a difficult task and we were glad that we were able to achieve what we did.
4. In the Advice section, when we add css for an Add New Post (Glyphicon-plus Sign) on a css page, the orientation of the sign is not in the proper format. The orientation and size of the sign is only positioned correctly when we add css to the html file.

Documentation: Division of Labor

- Product Backlog (Archana)
- UML Diagram Edits based on project 4 scope
 - State (Archana)
 - Use Case (Archana)
 - Class (Victoria)
- Meeting Log (Anjali)
- Gantt Chart
 - Input: Sindhu, Anjali, Archana, Victoria
 - Make edits: Victoria
- Bug List
 - Team effort on Friday 11/22
- User Manual
 - Team effort on Friday 11/22
- Deployment Plan (Victoria starts, everyone helps)
- Maintenance Plan
 - Team effort on Friday 11/22
- Test Suite
 - Input: Sindhu, Anjali, Archana, Victoria

Sources

Average walking speed:

<https://www.healthline.com/health/exercise-fitness/average-walking-speed#average-speed-by-age>

CSS for time calculator:

- <https://codepen.io/alanhouser/pen/aErrQJ>
- <https://freefrontend.com/css-timelines/>

Bootstrap:

- Advice page: <https://startbootstrap.com/previews/clean-blog/>
- Time management: <https://startbootstrap.com/themes/agency/>
- Ratings form:
https://startbootstrap.com/snippets/sign-in-split/?fbclid=IwAR3OAKj_Z0Ah-ggOSDp4E9FuH5QZnl1r0AUFERMoQITOIhdzMIfrGeoRpGI
- Homepage <https://startbootstrap.com/previews/stylish-portfolio/>
- Advice landing page <https://startbootstrap.com/themes/business-casual/>
- View class ratings: <https://startbootstrap.com/templates/blog-home/>

Images used from:

- <https://www.pexels.com>
- <https://unsplash.com>

Tutorials Used:

- Initial Setup: https://youtu.be/PqeAvFf_HDI
- Form for maps: https://youtu.be/3XOS_UpJirU
- Forms for ratings:
 - https://tutorial.djangogirls.org/es/django_forms/
 - <https://docs.djangoproject.com/en/2.2/topics/forms/>
- Django Workshop from KC Women in Technology: <https://tutorial.djangogirls.org/en/>

Gantt chart: <https://online.officetimeline.com>

Class diagram: <https://www.draw.io/>

HTML Emoji Unicode

- <https://www.w3schools.com/charsets/tryit.asp?deci=128077>
- <https://www.w3schools.com/charsets/tryit.asp?deci=11013>

Maintenance plan, deployment plan and user manual have been attached as separate PDF files in the GitHub repo.