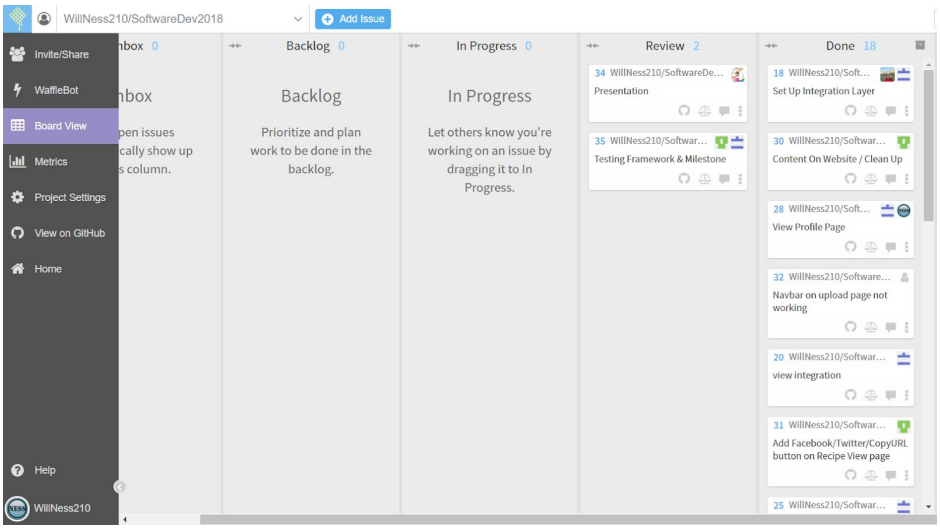
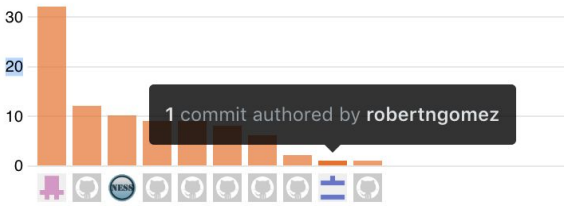
- Title
    - Let's Code Djibouti Recipe Sharing Website
- Who
    - William Ness
    - Robert Gomez
    - Kyle Bremont
    - Jeff Lucca
    - Nasurudin Furi
    - Harsh Deshpande
- Project Tracker
    - https://waffle.io/WillNess210/SoftwareDev2018
        - (should be able to login and view as long as you're a member of the GitHub repo, if not please email me @ William.Ness@colorado.edu)
    - 

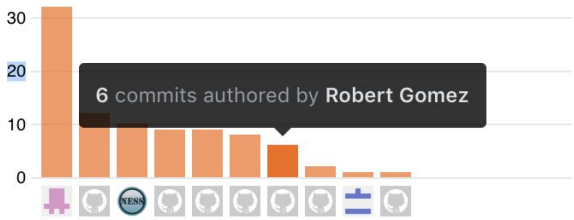
- GitHub Links
    - MAIN CODE
        - https://github.com/WillNess210/SoftwareDev2018
    - MILESTONES
        - https://github.com/WillNess210/SoftwareDev2018Milestones
    - MEETING NOTES
        - https://github.com/WillNess210/SoftwareDev2018Logs
    - Looking for?
        - Source Code
            - https://github.com/WillNess210/SoftwareDev2018
        - Test Cases / Test Plans
            - https://github.com/WillNess210/SoftwareDev2018Milestones
        - README.md in GitHub explaining to others what your project is about.
            - https://github.com/WillNess210/SoftwareDev2018
        - Project Milestone 7 document
            - You're reading it!

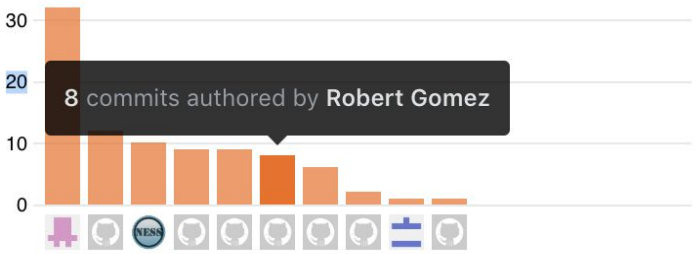
- Member Contributions
    - William Ness

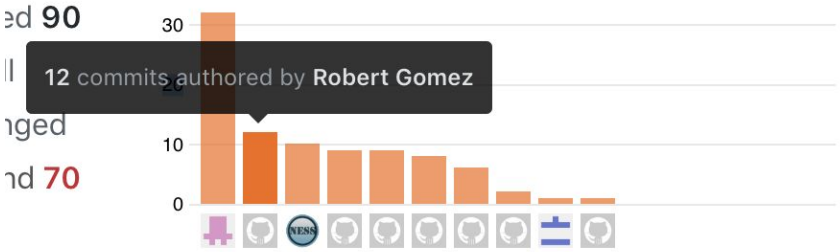
        -
        - Set up front end w/ bootstrap and some custom CSS
        - Set up Navbar
        - Created & added functionality for recipe view, explore, homepage, profile page,upload page w/ jQuery
        - Created login/register pages
        - Managed waffle.io
    - Robert Gomez



        -







- Github was incorrectly tracking my commit history: Total commits = 31
- Designed, implemented, and maintained the postgres database
- Helped with API and login
- General testing and bug tracing

- Kyle Bremont
- Jeff Lucca
- Nasurudin Furi

-
- Deployed website on Heroku
- Helped with upload page
- Created recipes
-

- Harsh Deshpande

-
- Set up API layer using JavaScript
- Wrote a queries page that uses the pg module and the Heroku Postgres CLI link - this would send different queries to the database and retrieve back data in JSON data
- After setting up the query commands, used the router feature of the express module to tie these queries to api endpoints
- Input information for POST and PUT calls would be put in JSON format and then passed to the appropriate api endpoint
- Also set up persistent session using cookies with the CookieSession module - the cookie would carry the value of the current user logged in (this was on the back-end/server side), cookie would get destroyed after user logged out
- Converted the Upload, Dashboard pages to ejs, served them up using the ejs module
- Made a python script that has API call testing using the unittest and requests libraries, deployed this onto Travis CI for continuous integration

- Deployment
  - https://djibouti-recipe.herokuapp.com/
- Continuous Integration System
  - https://travis-ci.com/WillNess210/SoftwareDev2018Milestones