PM Tool: Waffle.io

We will be using waffle.io, which is built into GitHub. It's a form of Kanban project management that allows us to take tasks and move them from backlog, to in progress, to finished. It is integrated with our Github and our Slack to allow for easy use.

Project Requirements:

- Ability to upload / create recipes easily
- Google Authentication Sign in
- Ability to bookmark recipes to account
- Search / browse recipes
- Easy sharing (goo.gl implementation?)
- Minimalistic Design
- View others profiles & bookmarked recipes
- Differentiate between public and private uploads

Project Plan:

- Sprint 1 - Website Structure:
    - Set up Heroku
    - Structure branches in GitHub
        - Development / Production
    - Page Templates
        - Homepage
        - Profile page
        - Upload page (form)
        - Search Page
        - Explore Page
        - Recipe View
    - Standardize CSS for project & page structure (header, footer, sidebars, etc..)
    - Create database structure
    - Add "dummy" recipes to database
- Sprint 2 - Handling Uploads:
    - Making upload page funcional (not user-specific yet though)
    - Integrate Searching
    - Integrate Explore page
- Sprint 3 - Sign in / Users:
    - Allow for sign in through google
    - Profile pages
        - Show public created recipes
        - Show bookmarked recipes
    - Allow for bookmarking recipes
    - Integrate Recipe sharing and privileges

Agile Meeting:

Robert:

1. Created a dummy database to learn about PostgreSQL
2. Create a write-up for database structure and requirements
3. Will have to have time during a busy week.

Nas:

1. Learned about some of the tools we were going to have to use
2. Learn about PostgreSQL and work on database requirements
3. Finding time to work on it, and learning something new

Harsh:

1. Developed basic ideas for what front-end and integration layer may look like
2. Creating basic CSS outline, header and footer template
3. Creatively making it look nice and design aspects - making it presentable but not over excessive

Kyle:

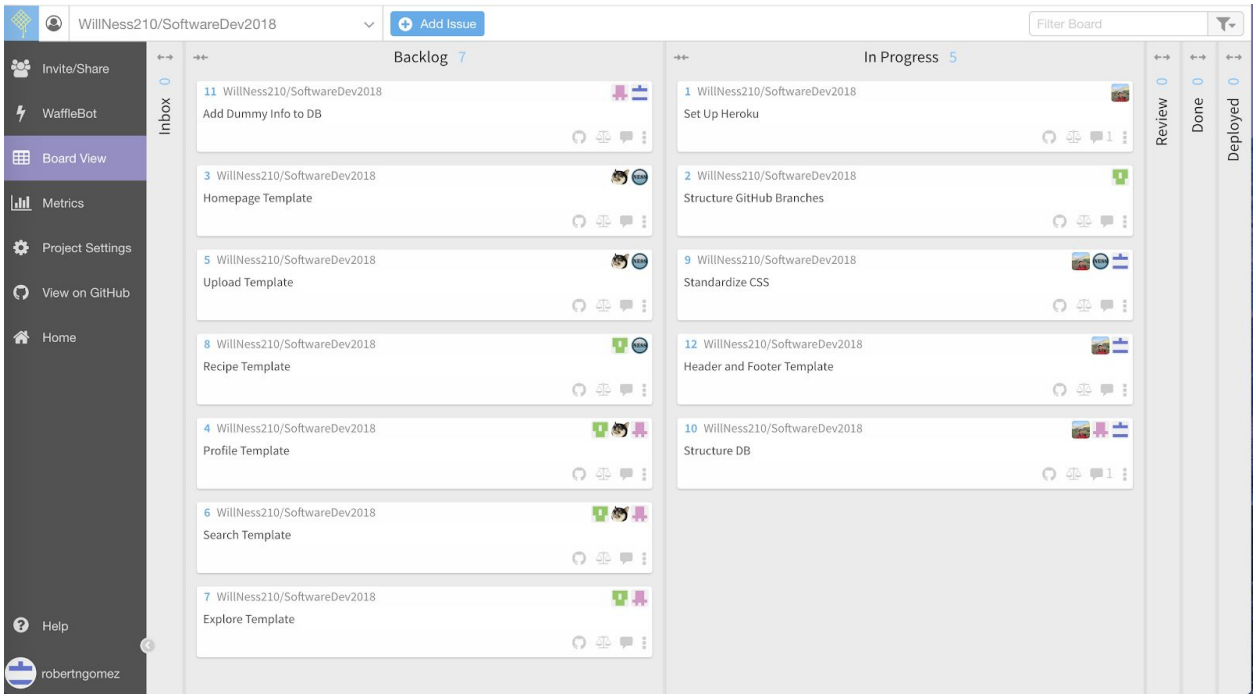
1. Set up waffle.io account and cloned github repositories.
2. Set up github branches and start templates if CSS outline is completed.
3. The creative aspect of the templates.

Jeff:

1. Learned more about waffle.io and CSS
2. Start templates if CSS outline completed, otherwise draw out/design pages to visualize what they can look like, also work on writeup for project requirements
3. I'm not super familiar with CSS and templates, or page design

Will:

1. Shared GitHubs with instructors, learned some about css flexbox
2. Help on CSS standardization & implementing CSS Flexbox, as well as page templates.
3. I need to get more familiar with Flexbox and front-end technologies for delivering the pages to the user.



In-depth Project Requirements:

The first requirement, as the basis of our project, is the ability for a user to create and upload their own recipes. Users should be able to input their own ingredients, change the quantity and measurements of those ingredients, and collect that information into a single recipe that is stored in a database so that it can be revisited at any point in the future. In the event that a user makes a typo or finds a way to improve their recipe, they will need to edit it, so all information inside the recipe should be able to be changed at any time. Next, because we want to build a recipe-sharing app, and not just a recipe-storing app, the second requirement of our project is the ability for a user to bookmark others' recipes to their account. Bookmarking a recipe will place it in a dedicated tab or section along with the rest of the user's bookmarks, allowing frequently used or favorite recipes to be easily accessible. In order to store this data along with any other account-specific information, our project must incorporate Google Sign-in. This will allow users to register using the Google account they already have rather than having to make a new account just for our site. The database needs to link to and differentiate between Google accounts, making sure that recipes, bookmarks, and other information are all displayed to the users they belong to.

For the first of our app's general features, we want to make it easy for users to discover new recipes and be able to find the recipes that they want. So, one requirement of the app is to be able to present users with recipes created and shared by other users that they will be able to explore through. This should be done through the form of a 'feed', similar to social media, where the user can keep scrolling through shared recipes. In addition, our project needs to implement a search feature that returns recipes to a user based on an input of specific recipe names or ingredient keywords. Because each of these recipes will have its own unique page, we also need to create a way to share these links easily. Goo.gl link shorteners or even something like light social media integration (share buttons) should be present to make spreading specific recipes easier.

Because this is an app centered on user-to-user interactions, we also have to mandate some requirements based around those interactions. First, users will have their own profile pages consisting of basic information about them as well as all of the recipes that they have created and bookmarked. Users should be able to view each other's profiles and see that information. However, there must also be privacy settings set in place for users who wish to share less about themselves. For example, if a user has a secret family recipe or some other reason to keep their uploaded recipes private, they must be able to have the option to make some or all of their recipes visible only to them. Because the app will also have a browsing and searching section, the database must be able to differentiate between public and private recipes so that only the people that are authorized to see certain recipes are able to see them.

Finally, the general design of our app should be minimalistic. It needs to present information in a clean and easy to understand manner. It also needs to be easy to navigate through each of the different sections of the app, so that the user can quickly switch between tasks and get where they want to go without hassle. The app's color palette needs to reflect this philosophy by putting a clear emphasis on the elements that matter most while not overloading the screen with too many bright or distracting tones.

Retrospective Meeting: The previous sprint was completed well, we were able to go through all the ideas we had and all decided on our recipe sharing site. We all worked on our goals of learning some of the programs and languages needed for the project and setting up everything required. We got most of what we needed to do done, but had troubles over deciding what tools to use. However, slowly we managed to decide what was the best for the job. We all agreed that we just need to improve on making time for the project and our sprint tasks.

- PDF/Graphic of timeline for sprint 1

  -
  - (people assigned to each task can be seen above in the screenshot of our waffle.io page) (made on teamweek.com)