SEADS Sprint Plan 2

User Story 1: As a user, I want a graph interface that is full of features so I can view or extract the information that I want.

- Implement a data exporting feature
- Select a well-developed graph visualization API so that we can minimize the development of basic graphing features
- Connect features such as granularity and different data sets (voltage, current) from the previous code
- Make graph professional looking so that our clients will have a good user experience.
- Implement dynamic graphing feature

User Story 2: As a developer, I want to be able to connect through an API to access data for each specific device

- There is already an API but it can't be called with Javascript. We need to change some server settings to get it to work.
- I want to be able to run the back-end server on multiple devices such as a Raspberry Pi.
- I want to have a program that can send test data to the backend.

User Story 3: As a developer, I want to record voltage, current, etc. data on some household appliances so I can have a dataset to test my software (front end and backend).

- Take home a PA1000 and record the electricity usage of 5 household appliances.

User Story 4: As a developer, I want to be able to test my graphs with all sorts of data so I don't have to manually insert it myself.

- Setup message formatting and exchanging successfully between a device, and the server.
- Seed meaningful pseudo-data by adding optional arguments to the program.
- Specify what type of data it is by adding optional arguments.

User Story 5: As a user, I want to be able to connect all kinds of devices up to the database so we can see generalized data from different devices from the same API.

- Make a general python script that gets numbers, formats it in a specific way, and sends it off to our server.
- This is a way we can gather real data from small linux embedded devices like raspberry pis or beaglebones.

Team Roles:

- Aiden Hoopes: Scrum Master/back end
- Dickson Tong: Scrum Master/front end

- Henry Crute: Server
- Andrew Ringer: Server
- Brendan Cicchi: front end