Team Buff Buddies
Recitation 112 - Team #2
Members: Owen Carlson, Aileen Ma, Briana Griffin, Alex Mazur

Jupiter - Milestone 2

Feature List -

- **User Accounts:**
    - Each user will have the ability to login to the site with an email and password. Logging in will allow a user to save location data, location comparison lists, etc.

- **Weather Cards:**
    - Each location will have a weather card that displays a quick overview of current conditions, and predicted weather over the next few days (2 or 3). These cards will also contain a predicted trip time section if the user enters a starting location. Weather cards do not lead to more in depth weather reporting. On mobile weather cards might display slightly less info, and could be clicked and turned over, to save some screen real estate.

- **Trip Time Predictor:**
    - This feature will hook-up to the Google Maps API and will give users a predicted time of travel based on traffic. This information will be displayed on a weather card. If the user has a location, but has not specified a time, the trip time will be calculated using current time. The user will be able to enter a time they would like to leave and see trip times based on predicted traffic for that time.

- **Weather Comparison Lists:**
    - A user will be allowed to compare different locations at the same time, side-by-side. Each location is represented by a weather card. If the amount of locations entered exceeds the size of the screen then they will overflow into a grid pattern. If a user is logged in then they will be allowed to save any list that they have made.

- **Dynamic Route Viewer (DRV):**
    - Beneath the list of weather card(s) there will be a map pulled from Google Maps API that displays a route to the location. The user will be able to change which route is loaded and displayed on the map. If possible, we might be able to overlay multiple routes on the map at the same time, must look into this.

- **Database:**
    - This is the software that will store user generated data. Mostly user created lists and user account settings will be stored here.

- **Node Server:**
    - This is the backbone to the entire website. The backend server will make all API calls, handle user login, and connect to the database.

Requirements -

- **User Accounts:**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Allow user to login to their account	Save/update user login to database
F/NF 2	User is able to change their account info	Verify login information with database

- **Weather Cards:**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Display current weather	Hookup to backend to receive weather information
F/NF 2	Display weather from yesterday, current, and	Hookup to backend to receive Google Maps API

	predicted weather for tomorrow next to each other	information on travel.
F/NF 3	Display the trip time predictor	Hookup to backend to send user selected data to server
F/NF 4	Allow users to select departure time and (date?)	Have the ability to be a member of a list of other weather cards
F/NF 5	When clicked change temp to C or F	

- **Trip Time Predictor:**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Display estimated travel time to destination	Hook up to backend to receive data
F/NF 2	Display on each weather card	Pull from user location data and weather card trip time selector

- **Weather Comparison Lists:**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Specific to each user, saveable and loadable	Must be able to load multiple weather cards at once in a timely manner
F/NF 2	User should be able to add and remove cards from the list	Save/Remove weather cards in user account
F/NF 3	Contains multiple weather cards	

- **Dynamic Route Viewer (DRV):**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Display at least one of the destinations currently represented in the list	App should be able to retrieve location data
F/NF 2		Should hook to Google Maps API for route information

- **Database:**

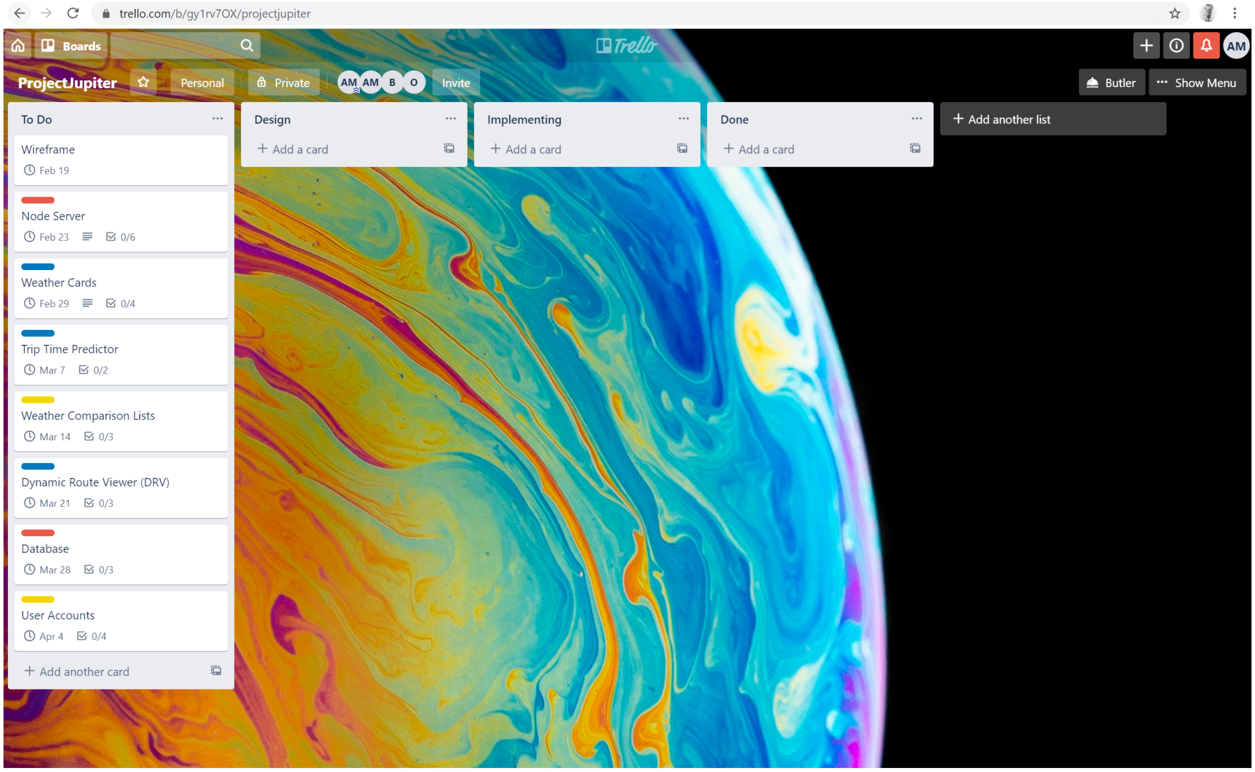
Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Store user login information	User login information will be encrypted
F/NF 2	Store user weather lists	

- **Node Server:**

Function / Non-Functional Requirement #	Functional Requirement	Non-Functional Requirement
F/NF 1	Get information on weather	Call weather API (which one specifically will be determined, as there are many)
F/NF 2	Get route/travel time information	Call Google Maps API
F/NF 3	Feed weather information to front end	Read and Write to database

F/NF 4	Feed route/travel time information to front end	
F/NF 5	Verify user login status	

This is a screenshot of our project management tool, Trello, and the board that we have so far. Each feature has a due date and requirements checklists. The order of each spring is clearly laid out as Design -> Implementing -> Done.



To Do
- Wireframe — Feb 19
- Node Server — Feb 23 — 0/6
- Weather Cards — Feb 29 — 0/4
- Trip Time Predictor — Mar 7 — 0/2
- Weather Comparison Lists — Mar 14 — 0/3
- Dynamic Route Viewer (DRV) — Mar 21 — 0/3
- Database — Mar 28 — 0/3
- User Accounts — Apr 4 — 0/4
- + Add another card

Design
- + Add a card

Implementing
- + Add a card

Done
- + Add a card