**Sprint 2 Report**

• **Heading:**  Sprint 2 Report, Hungry Slugs, Team Hungry Slugs, 5/07/2024

• **Actions to stop doing:**

1. Stop holding standup meetings after class ends since it is too early and not everyone can make it to every class. Switch to a later time on the same days.
2. Starting tasks close to the end of a sprint or TSR.

• **Actions to start doing:**

1. Hold Zoom standup meetings after class instead so everyone can make it every time (10:00 am)
2. The team should be more accurate at estimating tasks since tasks were consistently underestimated every sprint.
3. If a team member is waiting on a task to be completed they can assist other members to complete the task or begin researching tasks.
4. Making the tasks more specific in the sprint plans.
5. Tasks should be made shorter so that they can be completed in between each standup meeting. Each person should have around 6 tasks to complete per sprint.
6. Make acceptance criteria for each task in the sprint plan.
7. Follow the definition of done for each task in the sprint.

• **Actions to keep doing:**

1. Keep updating the team on the progress of each given task so people waiting on a task to be completed can get started right away.

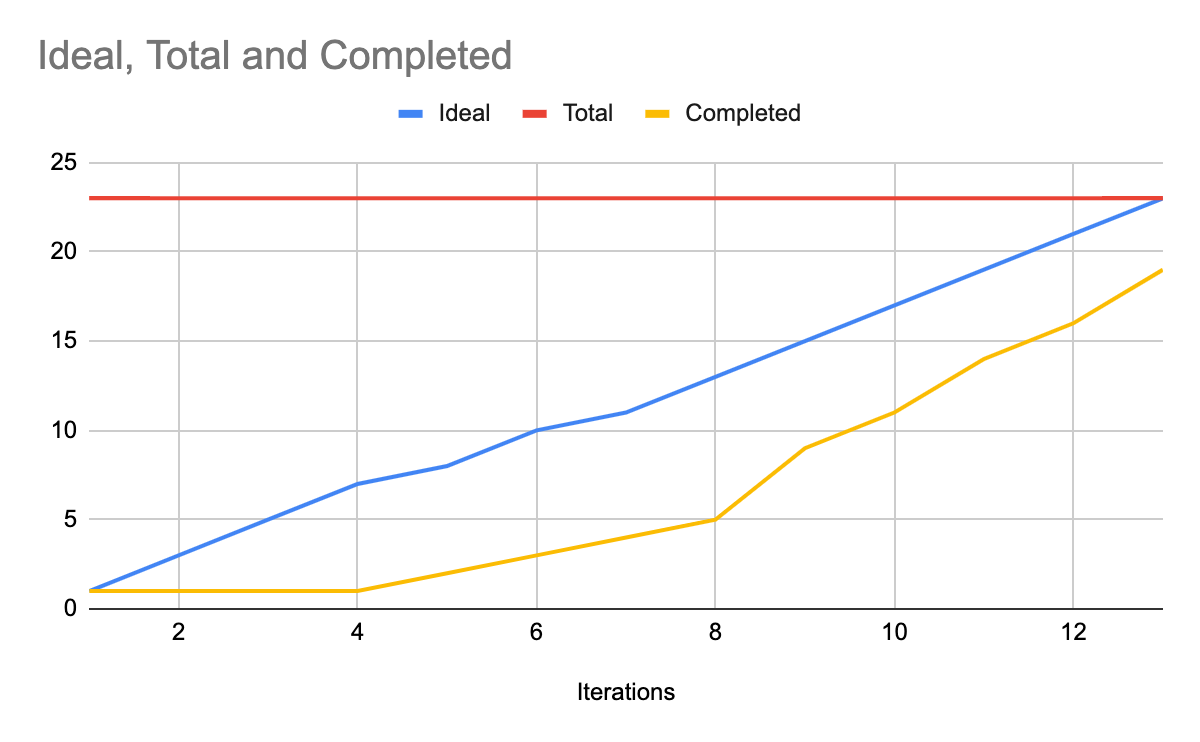
• **Work completed/not completed:**

Completed: US2

Not Completed: US1, US3

US1 and US3 are close to being completed however there were a couple of features that were not completed that led to the user stories not being finished.

• **Work completion rate:**



Task List:

* ~~Use~~ **~~Figma~~** ~~to figure out how we want the UI to look. (1 hr)~~
* ~~Create a way to rate meals on the backend by creating a foods collection on the database that stores food information and each user rating of that food. (2 hr)~~
* Create a login page where users log in/sign up using their UCSC Google account. Collect all user data into a python class. (3 hr)
* ~~Use~~ **~~MongoDB~~** ~~to keep track of all users that use the website. Create a user collection that stores all the users. Be able to pull usernames and favorite foods from a user. (2 hr)~~
* Create a user profiles page that displays current known information about the user. Use backend functions to get data on the user. (2 hr)
* ~~Improve how often the dining hall data is retrieved from the UCSC website. (2 hr)~~
* Add a UI to rate the dining hall foods and connect this to the backend. (2 hr)
* ~~Implement a search bar on the food page for each dining hall that filters food that matches the search query. (3 hr) [Search all dining halls for a specific food. There will be some sort of searching page]~~
* ~~Web scrape the main UCSC website for more data on features of food such as dietary restrictions and ingredients. (2 hr)~~
* ~~Add data from new web scraping features to the database structure for the dining halls. Basically store extra info in the food collection. (1 hr)~~
* Add more search options like a check box or drop down menu to filter types of food. (1-2 hr)
* ~~Check the time of day to hide and show certain meals to users. (1 hr)~~

Estimated/Ideal a total of 23 hours initially

Estimated/Ideal 1.5 user story every 7 days

Estimated/Ideal 1.6 hours of work a day

Actual hours worked on tasks was roughly 41 hours total

Actual 1 user story completed in 14 days

Actual ~3 hours of work a day

MORE DOCUMENTS BELOW:

