Sprint Plan 2 - NutriHall - CSE 115-A – Intro to Software Engineering - 10/24/24

• **Goal:** Short, 1-2 sentence description of the high-level goal(s) for the sprint.

For sprint 2, we want to have a working UI that displays the meals from the dining hall. Furthemore, the website should be able to store user preferences, restrictions, past meals(aka store user data).

Task listing, organized by user story: This section lists the user stories, in priority order from most important (top) to least important (bottom). Within each user story, there needs to be a list of tasks required to implement the user story, along with the time estimate for each tasks (preferably less than or equal to 6 ideal hours). This should look like:

User Story 1:
As a student who has dietary restrictions, I want to be able to select preferences and restrictions so that I am not recommended meals I cannot eat.

- Task 1: Design the front-end UI for selecting preferences and restrictions (e.g., allergens, dietary preferences). (3 hours)
- Task 2: Create a database schema to store user preferences and dietary restrictions. (3 hours)
- Task 3: Modify menu.js to fetch the firebase database menu items and display the information in the frontend (3 hours)
- Task 4: Implement the recommendation algorithm that filters meals based on user preferences and restrictions. (4 hours)
- Task 5: Integrate the recommendation system into the meal suggestions on the front-end. (3 hours)
- Task 6: Design and implement the front-end UI to display meal details, including nutrition information. (4 hours)
- Task 7: Fetch nutrition information form the database and display that information in the frontend

Total: 23 hours

User Story 2:
As a student who pays attention to their diet, I want to view past selected meals.

- Task 1: Set up a front-end view to display a history of selected meals. (3 hours)
- Task 2: Create a back-end API endpoint to fetch past meal data from the database. (3 hours)

- Task 3: Update the database schema to store a record of meals selected by the user. (3 hours)
- Task 4: Integrate the API with the front-end to display past meal data. (3 hours)

Total: 12 hours

- **Team roles:** Give a listing of all team members. Next to the team member, list their role(s) for this sprint. Assign each person to at least one role (for example, this role might be "Developer"). This looks like:

  Aiden: developer

  Violet : Scrum Master

  Artem: developer

  Amish: developer

  Kevin: developer
- **Initial task assignment:** A listing of each team member, with their first user story and task assignment. This should look like:

  Violet : *Integrate web scraper to the front end by creating a react script & accessing the meal data from the database. Integrate user profile data to the front end by creating react script & accessing the data from the backend.* (scrum master)

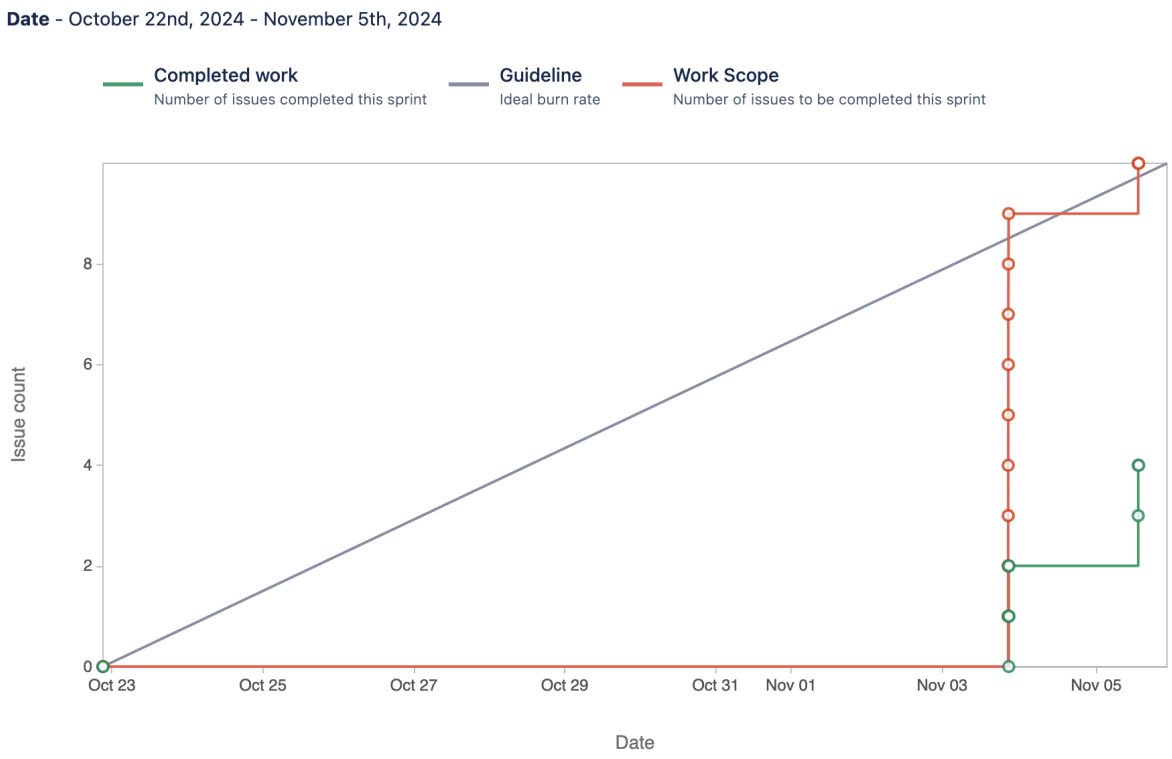
  Kevin: User story 2, modify web scraper tool to contain the allergen information along with the type of food. {Add database section for user profile information such as preferences and restriction. Get started on firebase authentication for logins on web page. Edit database to store past meals for each user profile.}

  Amish: - Add functionality to web-scraper. Code front-end web page in react to display website with all menus for dining halls.

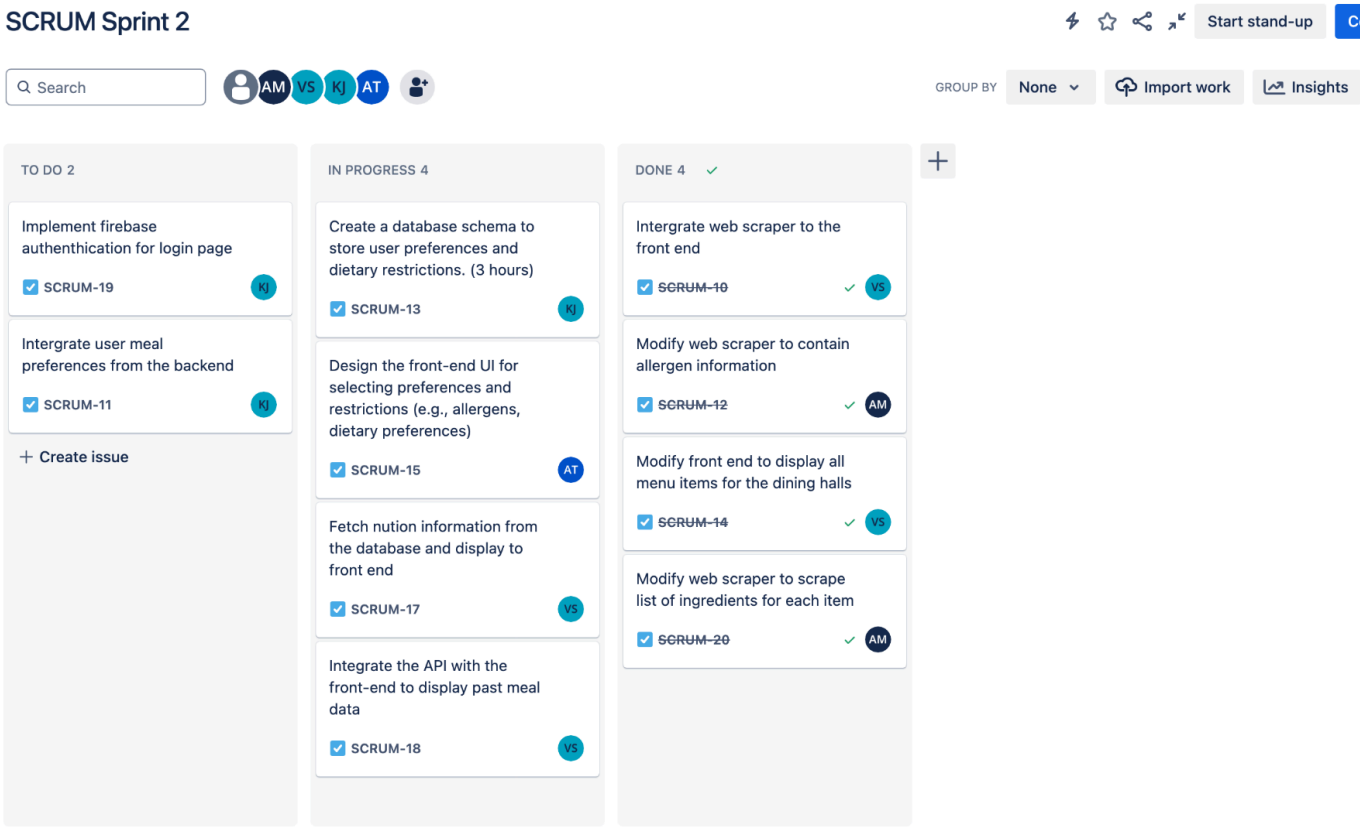
  Aiden: Work to get the front-end able to accept preferences and user settings and store them in the database.

  Artem: Worked on the login and sign in implementation.

- **Initial burnup chart:** A graph giving the initial burnup chart for this sprint and is labeled as such with sprint number and project name and is located in the lab.



• **Initial scrum board:** Also known as a task board, the scrum board is a physical board and labeled as such with sprint number and project name and located in the lab. This board has four columns, titled user stories, tasks not started, tasks in progress, and tasks completed. Index cards or post-it notes representing the user stories and the tasks for this sprint should be placed in the user stories, tasks not started, and tasks in progress columns. Tasks associated with a user story should be placed in the same row as the user story.



- **Scrum times:** Monday @ 2:30pm, Saturday @ 12pm, and Thursday 1:45pm