# Sprint 4 Plan - Planetarium

Scrumbags

## Sprint Completion 12/5/2023, Revision 1, 11/19/2023

# Goal

Fix displaying events in a daily view to match acceptance criteria. Implement a cooler user interface and add more convenient features such as searching and sorting tasks and events.

# Task Listing

Check [here](https://plannertarium.atlassian.net/jira/software/projects/PLANNER/boards/1) for the most up to date view.

As a student, I want to be able to…:

1. ~~Manage my events in various time frames so I can get the amount of detail I need to plan my schedule [8] - 2hr~~
   1. ~~Display events in daily view - 2hr~~
   2. *Acceptance Criteria*
      1. Be able to see events in all views
      2. Daily event view shows events in continuous time blocks
      3. Be able to add an event to the planner in all views by bringing up a larger view
2. ~~Feel cool while navigating and using the app so that I do not get frustrated or bored while trying to plan [8] - 7hr~~
   1. ~~Add date picker to fields that take a date - 1hr~~
   2. ~~Make task/event menus scrollable - 1hr~~
   3. ~~Add the top bar to event views - 0hr~~
   4. ~~Make gesture behavior consistent with left to get smaller window, right to get larger window, switch at the top to change view - 2hr~~
   5. ~~Set delay task to new color and icon - 0hr~~
   6. ~~Add a new look to display delayed tasks - 2hr~~
   7. ~~Add a logo to the login screen - 1hr~~
   8. *Acceptance criteria*
      1. The various pages look pretty to the Product Owner
      2. The transitions between pages are approved by the Product Owner
      3. The overall interface is consistent across pages
      4. Have a good date picker for creating task/event
3. ~~Check my planner in the distant past and future so I can review my productivity and plan for future events [3] - 3hr~~
   1. ~~Display latest monday as the default weekly view - 1hr~~
   2. ~~Add back and forward arrows to weekly view to move the see the a previous or future week - 1hr~~
   3. ~~Add a button to the top bar to open a monthly calendar date picker - 0hr~~
   4. ~~Move the weekly view to the selected date from the date picker - 1hr~~
   5. *Acceptance criteria*
      1. Can navigate to an exact week
      2. Can go backwards/forwards by one week from the weekly view
      3. Can navigate to an exact month
      4. Can go backwards/forwards by one month from the monthly view
4. ~~Mark some events as recurring so that I do not have to put repeating classes or meetings each time they happen [3] - 2hr~~
   1. ~~Extend the create event menu to input recurrence information - 2hr~~
   2. *Acceptance criteria*
      1. Can set an event as recurring when creating or editing them
      2. Can see the recurring events on the weekly and monthly event view
5. ~~Search my tasks so that I can lookup my old items and use it to help me filter out the things I don’t need [3] - 3hr~~
   1. ~~Create database method to get tasks with a substring in the name - 1hr~~
   2. ~~Create database method to get tasks with a tag - 1hr~~
   3. ~~Create database method to get tasks with a location - 1hr~~
   4. ~~Show the result of database methods in the search menu - 0hr~~
   5. *Acceptance criteria*
      1. Be able to search for tasks based on name, description, location, or tag
      2. Be able to search with matching values or values in a range depending on the attribute
6. Organize my tasks and events into categories so that I can keep my classes separately [13] - 11hr
   1. ~~Add a text field to enter tags on tasks and events - 1hr~~
   2. ~~Make a function to parse input on tags - 0hr~~
   3. ~~Create database methods to get and set tags of tasks and events - 2hr~~
   4. ~~Create database methods to get tasks/events with certain tags - 2hr~~
   5. ~~Add a way to display tags to tasks - 3hr~~
   6. Add a way to display tags to events - 3hr
   7. *Acceptance criteria*
      1. Can assign tags to any task or event
      2. Tags are displayed with each task or event they are associated with
      3. Able to sort tasks and day or week by tag
      4. Able to search for tasks and events by tag

# Definitions of Done

Task is done when:

* Code pushed into PR and all requested team members have reviewed it (minimum 1)
* Pull request was accepted into main branch
* Documentation describing basic functionality (not too in depth)
* Code compiles and runs without errors

General user story acceptance criteria:

* All tasks are done
* All tests pass
* All acceptance criteria are met

# Team Roles

Steven Xue - Backend developer, Frontend developer

Liam Xu - Frontend developer

Andrew Hu - Backend developer, Frontend developer

Andrew Yegiayan - Backend developer, Frontend developer

Cheng Wai Chong - Frontend developer

# Initial Task Assignment

Cheng Wai Chong - 2f

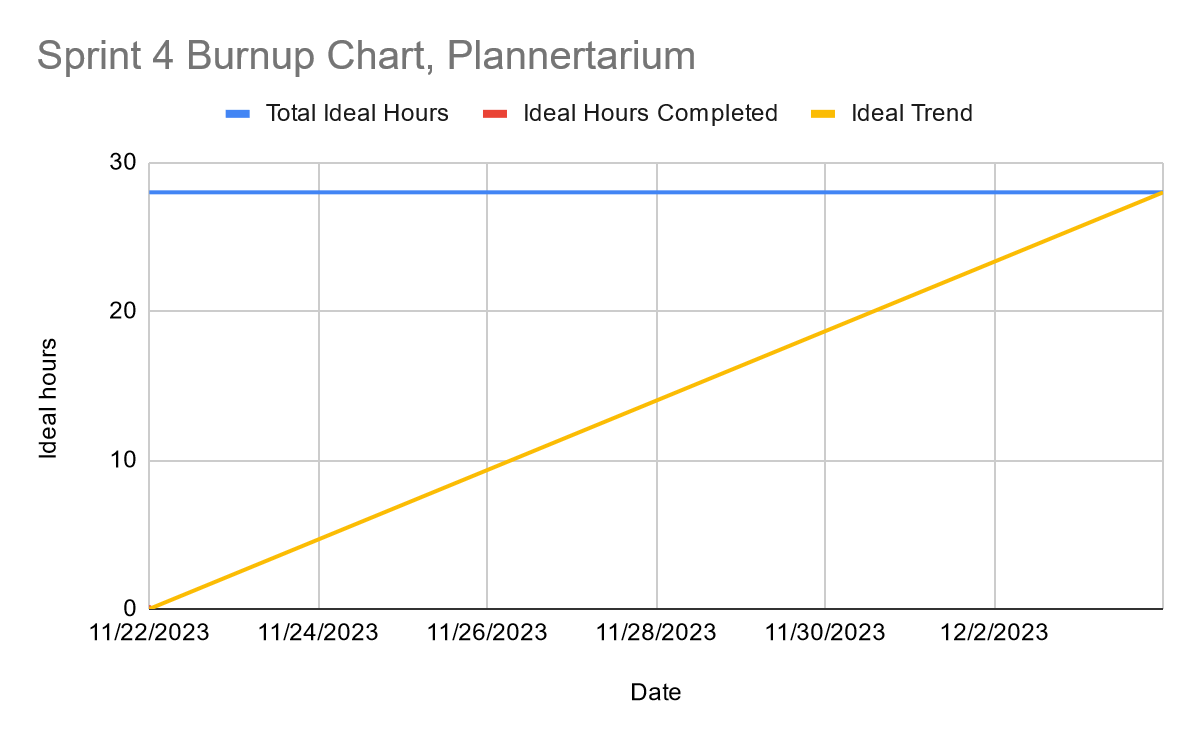
Liam Xu - 1a

Andrew Yegiayan - 6c

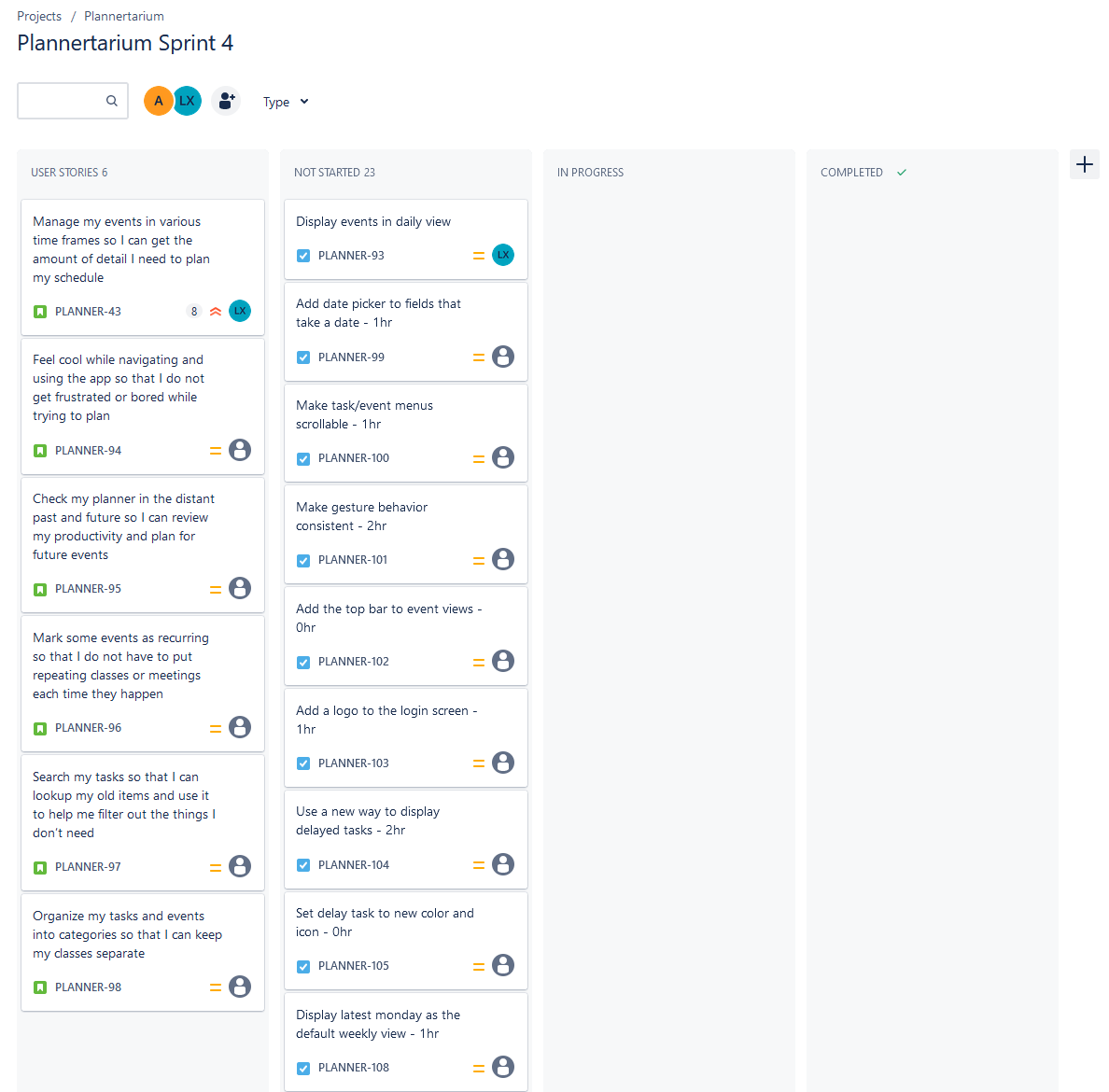
Steven Xue - 4a

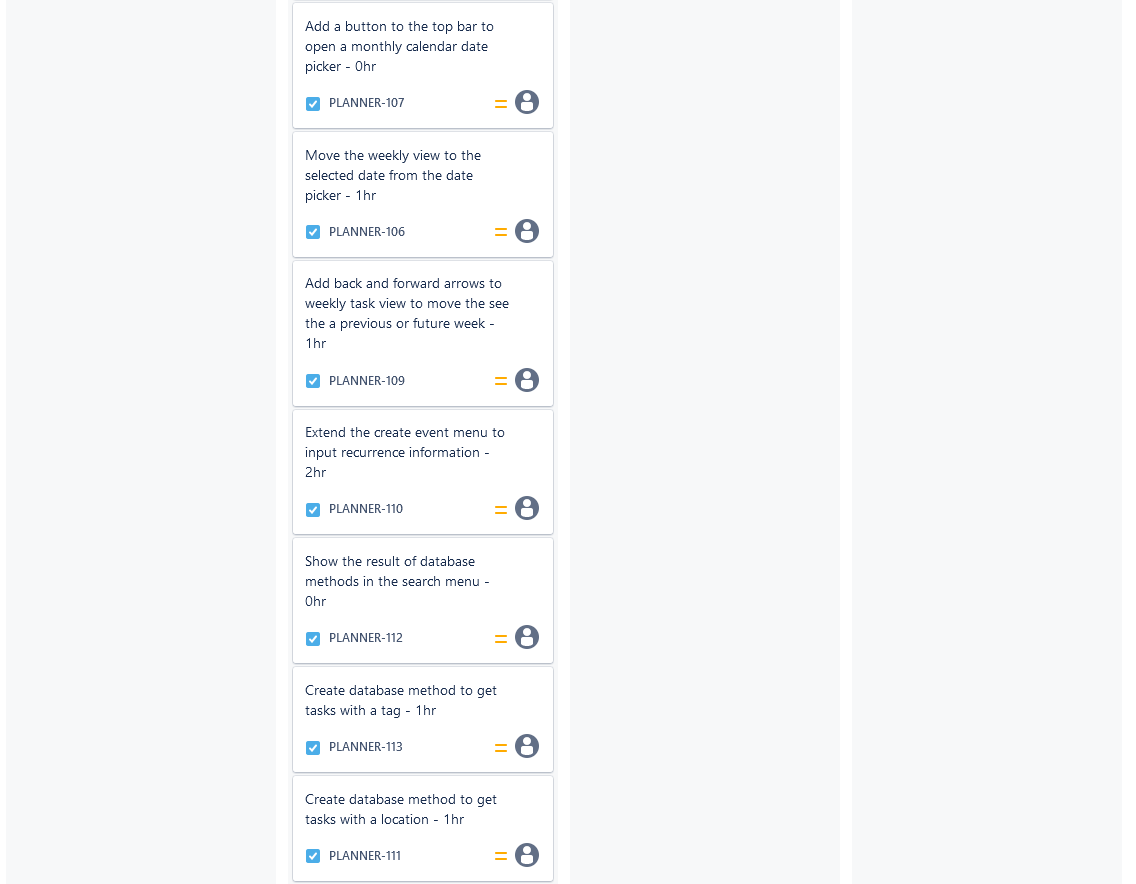
Andrew Hu - 2g

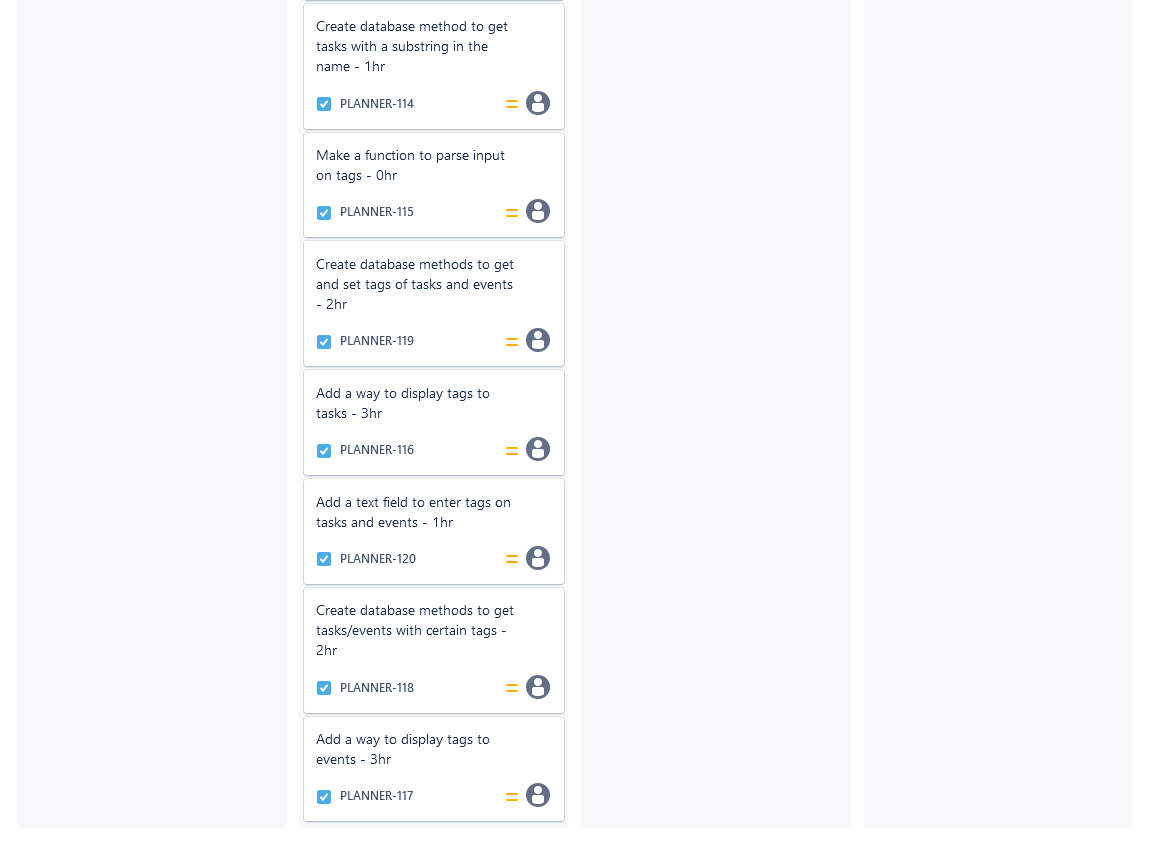
# Initial Burnup Chart



# Initial Scrum Board







# Scrum Times

Monday 12:15pm

Wednesday 6pm

Friday 6pm