# Sprint 3 Plan - Planetarium

Scrumbags

## Sprint Completion 11/22/2023, Revision 2, 11/19/2023

# Goal

Have complete user interfaces for essential parts of the application. Implement simple nonessential features into the task and event planner.

# Task Listing

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

As a user, I want to be able to…:

1. See my inputted information when I reopen the app so that I do not need to keep it open [13] - 3hrs
   1. Handle Google SSO errors and integrate Google username into database - 2hr
   2. Display a login screen for Android app - 1hr
   3. *Acceptance Criteria*
      1. Ensure the user can log in and their data gets successfully saved
      2. Once the user closes the app and reopens it, they can see all the data they had before in the app
2. Manage my tasks in various time frames so I can get the amount of detail I need to plan my schedule [5] - 4hr
   1. Create a new Widget on the UI to accept new tasks - 1hr
   2. Display tasks in daily view - 2hr
   3. Display tasks in weekly view – 0hr
   4. Display tasks in monthly view - 0hr
   5. Link the “add new task” UI Widget to the “Task” class - 0hr
   6. Widget to a UI that allows editing a task - 1hr
   7. Link the “edit task” UI Widget to the “Task” class - 0hr
   8. *Acceptance Criteria*
      1. Be able to see names of tasks in all views and be able to add tasks
      2. In any view, be able to select a task to expand it into a larger view
      3. be able to see more information about it in this view
      4. be able to edit the information in this view
3. Move my tasks around the timeline so I can assign them to other days [5] - 2hr
   1. Add UI gestures for task Widget - 2hr
   2. Add method to delete a task from the database - 0hr
   3. *Acceptance Criteria*
      1. Successfully use gestures to modify/move a task
      2. Be able to see the task is on a new day when the gesture is done
      3. be able to see the task is completed when the gesture is done
4. Manage my events in various time frames so I can get the amount of detail I need to plan my schedule [8] - 2hr
   1. Create a new Widget on the UI to create new events - 0hr
   2. Display events in daily view - 2hr
   3. Display events in weekly view - 0hr
   4. Display events in monthly view - 0hr
   5. Link the UI Widget to the new method - 0hr
   6. *Acceptance Criteria*
      1. Be able to see events in all views
      2. Be able to add an event to the planner in all views by bringing up a larger view
5. Edit an event so I can keep the planner up to date with changes to my schedule [3] - 2hr
   1. Add UI for seeing event and moving back to different times on the timeline - 1hr
   2. Add method to Event class for updating the fields on an event - 1hr
   3. *Acceptance Criteria*
      1. In the larger view of an event, be able to see and edit all fields of the event
      2. Open an event and make sure all the detail is visible
      3. Changes are reflected in the database
6. Look at different time windows so that I can plan for the short, medium and long term as needed [8] - 2hr
   1. Build the screen for the monthly view - 1hr
   2. Add UI buttons and swipe gestures to change time windows - 1hr
   3. *Acceptance Criteria*
      1. Able to move from daily view to monthly view with UI and gestures
      2. All screens display the correct corresponding tasks/events for each time window
7. As a student I want to see homework deadlines easily so I do not miss deadlines [5] - 3hr
   1. Make a new UI banner that shows the name of a task due on a day - 1hr
   2. Display the UI banner when the time window overlaps with the due date - 1hr
   3. Make a database function to fetch a list of due dates on a day - 1hr
   4. *Acceptance criteria*
      1. At the top of the daily view of tasks, be able to see a small list of tasks that are due on that day
      2. At the top of each day in the weekly view of tasks, be able to see a small list of tasks that are due on each day
      3. On each day in the monthly view of tasks, be able to see a small indicator of tasks that are due on each day

# 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

Liam Xu - Frontend developer

Andrew Hu - Frontend developer, Backend developer

Andrew Yegiayan - Backend developer

Cheng Wai Chong - Frontend developer

# Initial Task Assignment

Cheng Wai Chong - US2, Create a new Widget on the UI to accept new tasks

Liam Xu - US4, Display events in weekly view

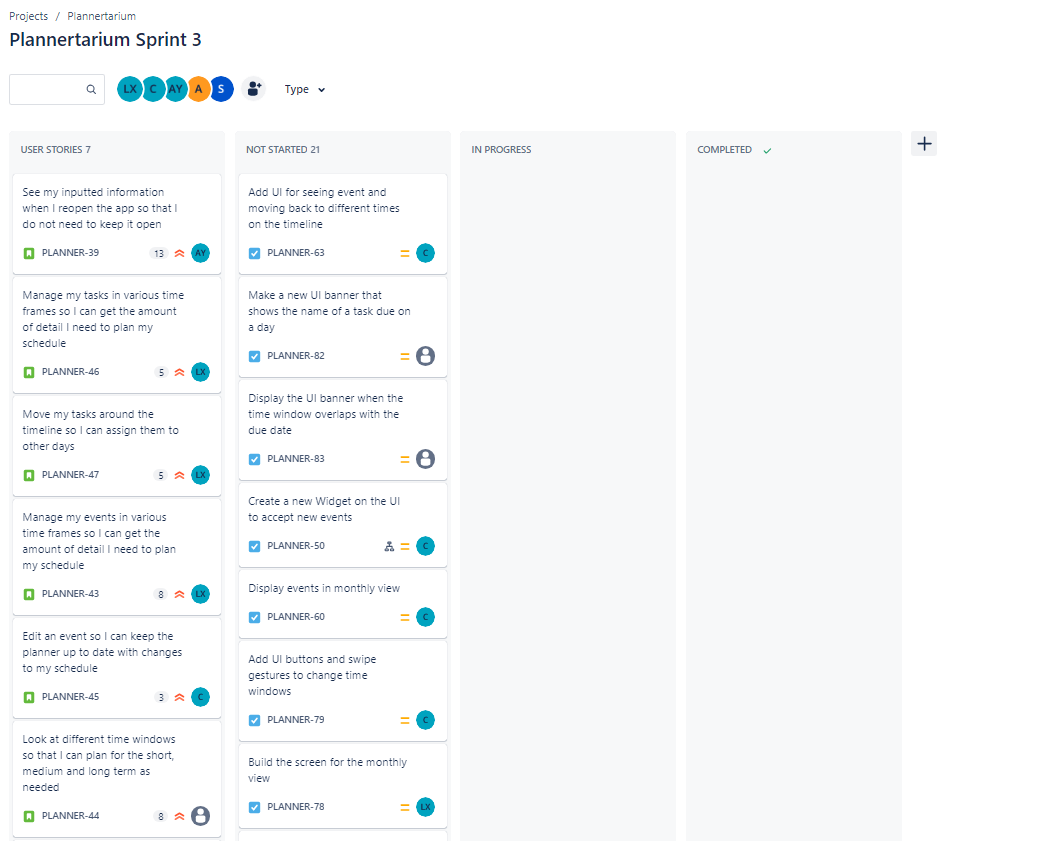
Andrew Yegiayan - US1, Handle Google SSO errors and integrate Google username into database

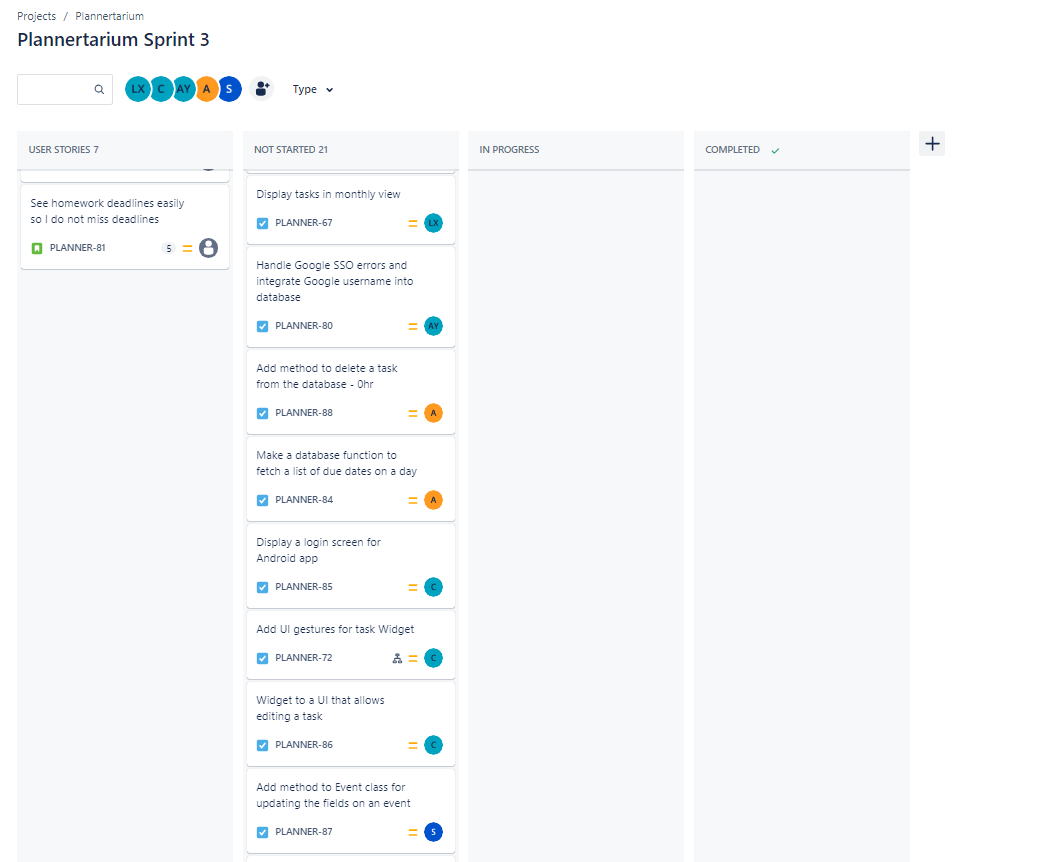
Steven Xue - US5, Add method to Event class for updating the fields on an event - 1hr

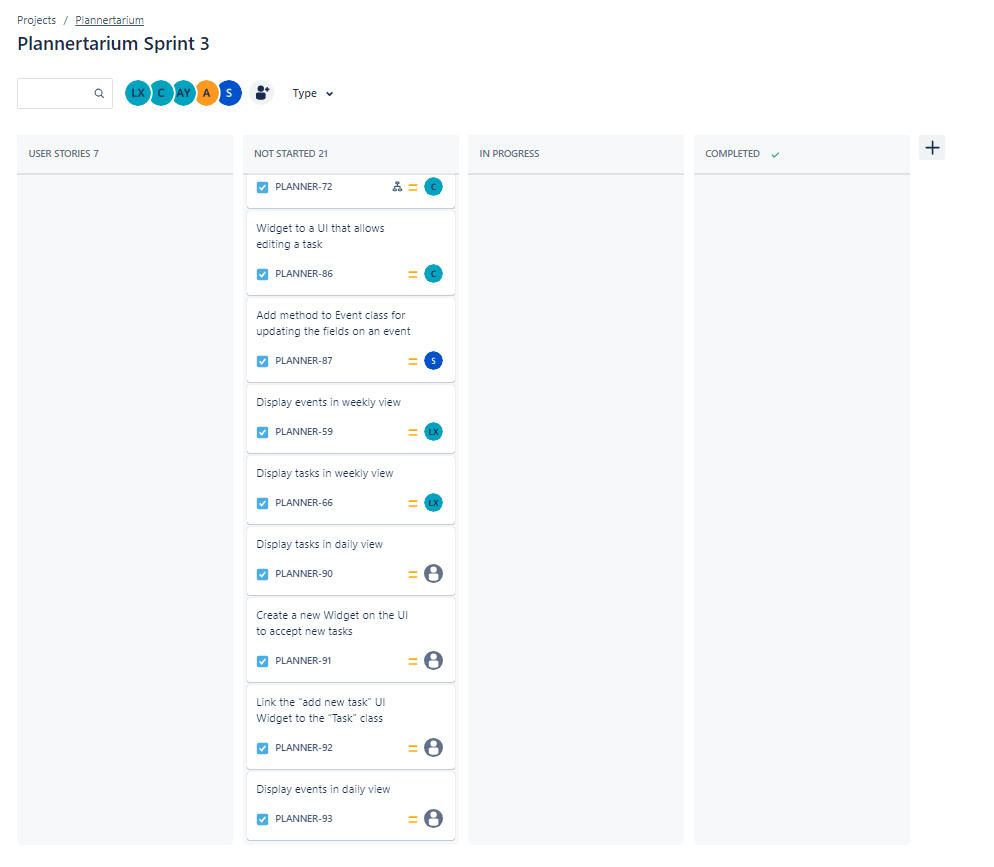
Andrew Hu - US7, Make a database function to fetch a list of due dates on a day - 1hr

# Initial Burnup ChartChart

# Initial Scrum Board







# Scrum Times

Monday 12:15pm

Wednesday 6pm

Friday 6pm