Sprint 4 Schedule

Table of Contents

Introduction

This is the document used to prioritize which subtasks must be completed first in each of the user stories. It helps us identify which subtasks have room for individual troubleshooting, and which ones require immediate team help if there is a block / ones that cannot be delayed at all. As developers, it essentially lets us know how we should be handling each of our assigned user stories.

We have one additional developer story/chore this sprint, [PP-93]. The implementation influences the final product release, so it has still been assessed for Critical Path. The Notifications feature needs to be adjusted for Project due dates (added last sprint); and the points display system needs to accommodate widgets that will subtract points (non-free widgets) – which will be added this sprint. Although both user stories were written and completed in their respective sprints, this new functionality was added recently. Thus, it is a good idea to test against them to ensure everything works, now that they have become testable.

Note that the “Earliest Finish” for each user story is denoted by the **EF** tag to the right of the table at the corresponding subtask row.

Design Decisions

The idea behind not using the JIRA Ticket Number as the Subtask ID itself is that a → b → c → d etc. flow in alphabetical order, reflecting the logical order of subtasks that the developer will likely tackle in. The JIRA Ticket Number on the other hand may have non-increasing numbers depending on the time they were created, which could potentially confuse the reader if it was also the Subtask ID.

While not shown because the information is readily available in sprint4.md and JIRA itself, the Criteria of Satisfaction for each User and Developer Story was used to determine the true end state of the respective critical path.

Note that the time estimates for each subtask are dependent on the specific developers responsible. Everyone codes and learns new technologies at a different rate.

Jira Stories and Critical Paths

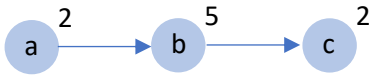
User Story 11

[PP-11] As Treyvon, an average user, I want to be able to check the weather for the day, so that I know if I will need an umbrella later on.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-100]	Research simple weather APIs to embed in websites	2	none	✔
b	[PP-101]	Implement a weather API that is refreshing accurately to show location-relevant data.	5	a	✔
c	[PP-102]	Position the weather widget appropriately on front end in dashboard and widget store.	2	b	✔

EF

Network Diagram:



Critical Path: a → b → c. This user story is sequential, like many from this sprint. A standalone widget, like the rest below, require step-by-step development generally in the form of research, implementation, and finally frontend tweaking.

Total Time: 9 hours. This is the total time of all the user stories, as required for sequential subtasks.

User Story 15

[PP-15] As Treyvon, an average user, I want to be able to tag tasks with priority levels; this way I can better prioritize my important tasks from my lesser ones. I want to also be able to sort them based off these priority levels.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-96]	Implement frontend: Sort the tasks by their priority levels in descending order.	4	none	✔

EF

Network Diagram:

Critical Path: a. This user story is purely filtering task display on the frontend; the backend need not be altered by filtering here.

Total Time: 4 hours.

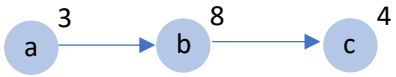
User Story 16

[PP-16] As Aayan, a power user, I want to be able to rearrange widgets so I can put the most important ones to me in a more accessible place.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-103]	Research how "Beautiful DnD (Drag 'n Drop)" component works.	3	none	✔
b	[PP-104]	Implement the Beautiful DnD inside bar for widgets.	8	a	✔
c	[PP-105]	Ensure apps purchased from widget store appear in the re-done sidebar with drag 'n drop functionality.	4	b	✔

EF

Network Diagram:



Critical Path: a → b → c. This user story is sequential, as all subtasks cannot be completed until the prior one is.

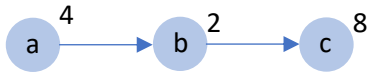
Total Time: 15 hours.

User Story 20

[PP-20] As Treyvon, an average user, I want a visual indicator of how many projects I've completed, so I can remind myself how far I have come.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-106]	Research live graph-modelling visual displays	4	none	✔
b	[PP-107]	Store in back-end all the completed/deleted projects of a user with title information.	2	a	✔
c	[PP-108]	Create graphic visual using stored projects on front end.	8	b	✔

EF

Network Diagram: 

Critical Path: a → b → c. All three subtasks are dependent on the one prior. This is a sequential user story, with most of the work being done towards the end (creating the actual, visual live-feed graphic for purchase in the widget store/sidebar display).

Total Time: 14 hours (4 + 2 + 8 = 14).

User Story 92

[PP-92] As Treyvon, an average user, I want to be able to delete projects manually on my own time, rather than having them auto-delete on completion. This lets me keep track of what I've completed so far recently.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-94]	Implement backend: Delete button should remove Projects from user's stored list. Upon login the state should remain consistent.	2	none	✘
b	[PP-95]	Implement frontend: Ensure delete button appears per each Project, and let completed projects remain.	3	none	✔

EF

Network Diagram: 

Critical Path: b. **a** can be worked on simultaneously as **b**, as neither have dependencies. As **b** is longer, it is the critical path. The frontend is not reliant on the backend for displaying already present projects prior to this user story.

Total Time: 3 hours.

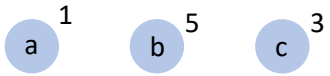
Developer Story 93

[PP-93] As a developer, I want to refine existing features in the new UI layout: specifically points display and notification popups.

Subtask ID	JIRA Ticket Number	Task Description	Estimated Time (hrs.)	Dependencies	Required for Critical Path?
a	[PP-97]	Use non-free widget to subtract points via backend.	1	none	✗
b	[PP-98]	Make notifications appear in the centre of the new UI layout.	5	none	✓
c	[PP-99]	Make notifications appear via newly added Projects due dates.	3	none	✗

EF

Network Diagram:



Critical Path: b. **a** and **c** can be worked on simultaneously as **b** as these subtasks don't have dependencies. **b** will take the longest to complete. This is a developer story tweaking already-implemented features from previous sprints, to newer features added recently; each subtask has background work already setup, and therefore is not dependent on the next subtask.

Total Time: 5 hours.