# Sluglet: Sprint 4 Review

## Completion Date: Tues, Dec 5
## Goal: Location Services, routing to classes, and keeping track of previous quarters.

| Priority | User Story | Tasks | Story Points |
|---|---|---|---|
| 10 | As a user, I want to be notified when it's time for class so I don't forget | ☑ ~~Set up Alarm Manager to wake up the app and perform a function at a designated time - 7 SP~~<br>a. Handle dateTime events. (Probably need to parse the dateTime field of the CourseData object) | 7 |
| 5 | As a user, I want the app to have multiple settings to customize my experience and make it my own. | ☑ ~~Create an empty Settings Page with navigation and scrolling capabilities - 2 SP~~<br>□ Create a Calendar theme composable that can change the color of the calendar - 1 SP<br>☑ ~~Create a composable settings option to sign out the current user (AccountService) - 1 SP~~<br>□ Create a composable to choose between light/dark mode or use the system default - 1 SP<br>□ Create a composable to enable/disable push notifications - 2 SP<br>☑ ~~Combine composables into a final settings screen - 2 SP~~<br>☑ ~~Link between Sign Up page and the Settings menu - 3 SP~~ | 9 |
| 8 | As a user, I want to easily remove courses from my schedule so I can better organize my time. | ☑ ~~Change coursebox to take in an Icon as a parameter ( + or - ) - 1 SP~~<br>☑ ~~Implement onRemoveClick() function in Schedule Screen view model - 1 SP~~<br>☑ ~~Add a removeCourse function to AccountService - 2 SP~~<br>☑ ~~Optional icons on composables~~ | 4 |
| 4 | As a user, I want to have a profile where I can see my previous quarter's classes and schedules so I can keep track of them | □ Add a method to retrieve current quarter and a list of existing quarters from a service module - 2 SP<br>□ Create a settings composable to select the user's desired quarter from existing quarters. - 1 SP | 7 |

| | | | |
|---|---|---|---|
| | | ☐ Update "courses" in StorageService to refer only to the current quarter's courses - 2 SP<br>☐ Update user course storage to store multiple quarters of courses - 1 SP<br>☐ Update user course retrieval to retrieve the currently selected quarters courses - 1 SP | |
| 7 | As a user, I want to see my location on the map so I know where I am on campus | ☑ ~~Require location service permissions to be enabled - 1 SP~~<br>☑ ~~Create a user location marker (so that it is distinguishable from other markers) - 1 SP~~<br>☑ ~~Implement MyLocation overlay for OSM to get current location (Link with example) - 4 SP~~ | 6 |
| 5 | As a user, I want to see a route from my current location so I can know how to get somewhere easily | ☑ ~~Create a RoutingService module to interface with routing API - 4 SP~~<br>☑ ~~Use a routing API to find the shortest path on OSM between two points - 3 SP~~<br>☑ ~~Use coordinates from routing API to draw a polyline on map - 2 SP~~<br>☑ ~~Create Polyline.kt file in OSMaps package for drawing polylines - 3 SP~~<br>☑ ~~Create UI element on CourseMarkers that draws a route when selected - 2 SP~~ | 14 |

## Continue

- Maintaining our Known Bug List to identify and fix known issues.
  - Having an easily referenced document helps identify what issues have been created from new implementations and what issues are existing problems.
  - Gives a source of tasks to work on if Sprint tasks are completed early
- Following revised coding standards
  - Using a more specific list of standards allows for easier integration and helps ensure that code is clean and well-written
- Starting the sprint from a fully merged branch
  - Having each Sprint start from the same codebase is an essential part of Scrum, otherwise deliverables become too fragmented and extremely hard to integrate.
- Pair programming
  - Pair programming has allowed for more work to be completed, and helped resolve common issues quickly and efficiently

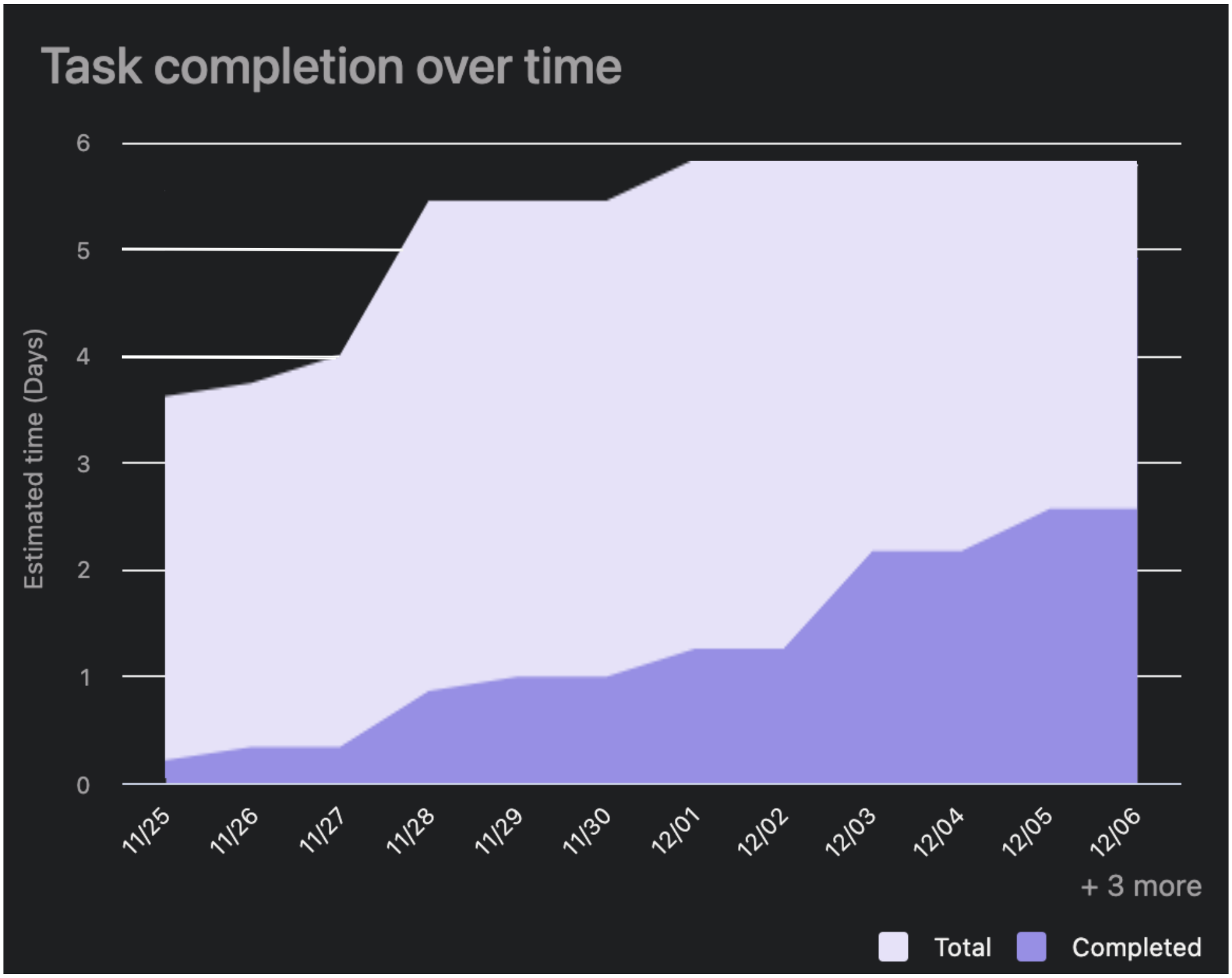
## Start Doing

- Automated Tests for integration
  - Having an automated way to verify that features are functional would greatly reduce overhead reviewing branch pull requests
  - Provides an easy metric to understand integration criteria at a glance
- Unit testing suite
  - More defined unit tests help ensure functionality in edge cases, and would catch the issues that have fallen through the cracks

## Stop Doing

- Ignoring the burnup chart during the Sprint
  - The burnup chart was not always up to date by each Scrum, and as a result it became hard to estimate the amount of work done and remaining in the Sprint.
- Not filling out the Test Plan and Report document with functional tests for user stories as their functionality is completed.
  - Not having the knowledge for testing other modules makes it difficult to determine when functionality breaks with new implementations.
- Retroactive time tracking in Asana
  - Marking time committed a day or more after it has been completed makes constructing the burnup chart difficult, as time is often missing.

## Burnup Chart:


Task completion over time
Estimated time (Days)
6
5
4
3
2
1
0
11/25
11/26
11/27
11/28
11/29
11/30
12/01
12/02
12/03
12/04
12/05
12/06
+ 3 more
Total
Completed