

Sprint 4 Plan

Product Name: CruzCal

Team name: CruzCal Developers

Sprint Completion Date: May 31, 2022

Revision number: 2.0

Revision date: May 30, 2022

Goal

The goal for sprint 4 is to complete the current backlog, add and/or update unit tests, prepare for final presentation, and optionally attempt our *stretch goal* once project release is ready for launch.

User Stories for Release

1. Sprint 4 (5/18/22 - 5/31/22) - *Stretch Goals*
 - a. Team goal: Create multiple calendars, query classes via class name rather than CRN, and merge calendars with other users
 - b. As a student, I want to be able to search for class via the class name rather than having to rely on what the CRN (class registration number)
 - c. As a student, I want to be able to view my friends' schedules and compare them with my own (similar to when2meet)
 - d. As a student, I want to be able to create multiple calendars for different quarters
 - e. Tasks:
 - i. Query classes using class name (versus CRN)
 - ii. Enable autofill for class search

Team Roles

- Tanmay Mittal: product owner, backend developer
- Kitana Toft: frontend developer
- Mario Reyes: **SCRUM Master**, backend developer
- Pablo Gaeta: backend developer, frontend developer
- Andrew Lim: frontend developer

SCRUM Times

Weekly Meetings

- **Tuesday: 10:30 - 11:30 am** meet on campus (or virtual) at LEEPs lab room 416
- **Thursday: 5:30 - 6:15 pm** with TA (remote through Zoom)
- **Friday: 9:05 - 9:20am** in-person meeting after class (weekly)
- **Sunday: 4:00pm - 4:15 pm** online check in for progress report (virtual)

Check in Agenda

1. What progress has been made?
2. What complications are we facing?

3. How should we move forward with these facts?
4. What are the next steps to work on the current or following task?

How to Run Each Meeting

- 2 minute intervals for people to express & summarize their ideas or findings
- Developers should meet & determine how to split up the given tasks based on user stories per sprint

Initial Task Assignment

Please reference figure 1.

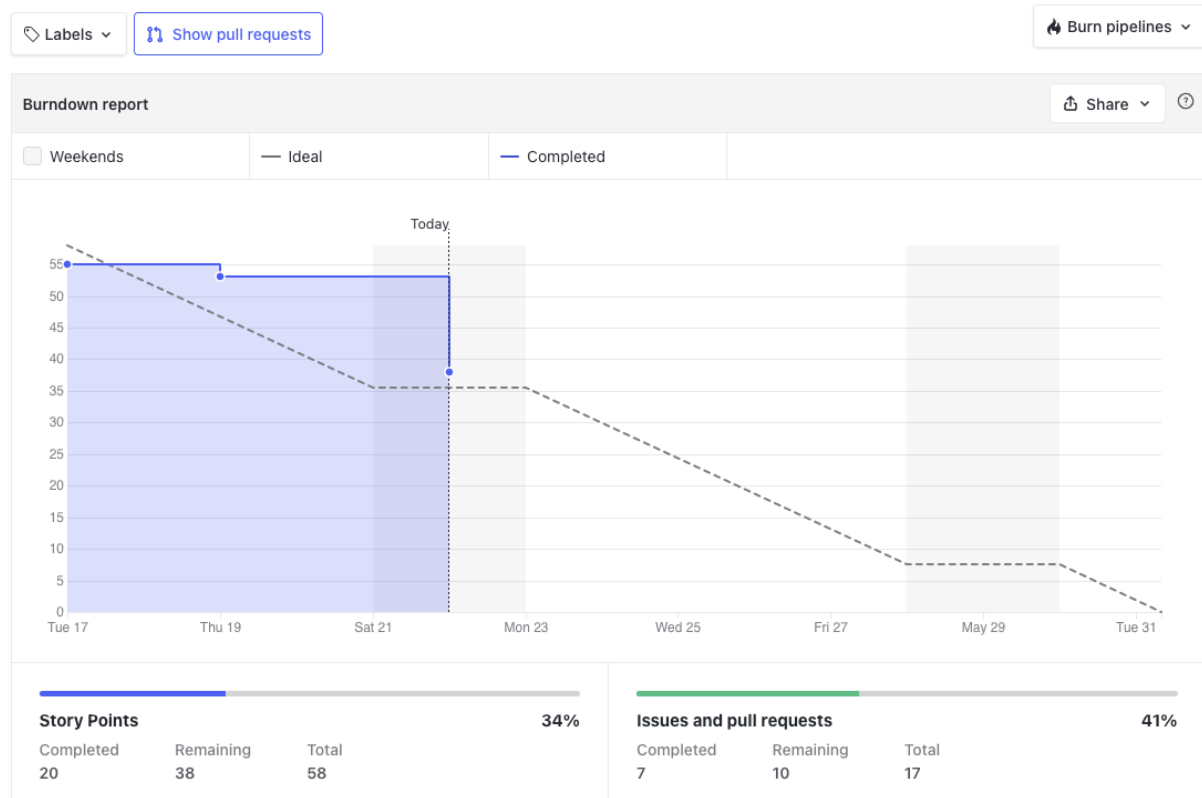
User stories	Tasks
As a student, I want to be able to search for class via the class name rather than having to rely on what the CRN (class registration number)	Query classes using class name (versus CRN)
	Enable suggestions for class search (Not maintained but may be easy to setup: https://github.com/moroshko/react-autosuggest#installation)
	^ Single text input, when selected would add new course selection row
As a student, I want to be able to view my friends' schedules and compare them with my own (similar to when2meet)	Decide how classes should be imported (import link to merge into current calendar or just initialize by URL)
	Above link format will likely have form: /api/calendar?termCode=:code&courseIDs=:id&...&courseIDs=:id
	Create button to merge link calendar (if valid) into course selections
	Implement history of calendar selection and allow undo/redo operations using an atom
As a student, I want to be able to create multiple calendars for different quarters	Calendar read: Is this just a dropdown? Should it display for non-logged in users? ID by term or user-given name?
	Calendar creation: If saved by term, just select given term. Otherwise we need to differentiate calendars of same term and have add button
	Calendar updates: Will imported user calendars automatically sync or require saving?
	Calendar deletion: Remove from db and set UI to default calendar
	Define Calendar table based on above decisions
Current workload	Over weekend, we each try to make simplest scheme for users storing, retrieving, and updating calendars (later planning poker)
	Get all backend tests working on Mario's branch
	Fix bugs in ics calendar generation and add testing to verify calendar dates
	Determining a means of saving user progress (undo/redo) in case user makes the mistake of change their term (wiping their progress)
	Finishing up a backlog of frontend testing for both components & atoms
	Finish google calendar api integration along with testing, make sure it creates a new calendar for the user - Mario
	Figure out how to use local storage to store array of course selections
	Test my atoms and schedule selection/submission components

Fig. 1

Burndown Chart

Sprint: May 17 - May 31, 2022

Start May 17, 2022 Due by May 31, 2022



SCRUM Board

