Lecture 10: January 14

Spring Project Planning

Agenda

- **Fall Semester Feedback**
- Spring Semester
  - What's new
  - Spring Schedule
  - Grading Criteria
  - Sprint Expectations
  - Setting up Spring Sprints
- Upcoming Deadlines
- Project Goals
- Github Account Verification

Fall Semester Feedback

- Really nice job overall!
- Clear improvements in presentation skills
- Majority of projects are on track
- Majority of teams using PRs, some doing code reviews

Individual feedback will be provided during instructor meetings

Agenda

- Fall Semester Feedback
- **Spring Semester**
  - What's new
  - Spring Schedule
  - Grading Criteria
  - Sprint Expectations
  - Setting up Spring Sprints
- Upcoming Deadlines
- Project Goals

What’s New?

- Student feedback from fall semester
  - Not enough time to focus on projects
  - Grading is unclear & feedback isn’t frequent enough and/or actionable

What's New?

- Student feedback from fall semester
  - Not enough time to focus on projects
  - Grading is unclear & feedback isn't frequent enough and/or actionable

- Spring Changes
  - Less labs, more office hours*
  - Less presentations, with a focus on presentation skills instead of content
  - Less tangential deliverables
  - Sprint rubric on course website
  - Greater focus on setting sprint goals

Spring Schedule

https://gw-cs-sd-25-26.github.io/

Month	Expected Status	Monthly Focus	Deliverables
September	N/A	- Figure out teams - Brainstorm projects	- Create teams - resume
Mid-September	- Teams selected - Handful of project ideas	- Final project selection - Begin meeting w/ mentors	- Project proposal - Hardware/software request - Writing: Team Charter
October	- Project selected & approved	- Begin technical investigations (services, apis, language, etc) - Flesh out project functionality & requirements - Coding should start (scaffolding, ci/cd, prototyping)	- Writing: Technical summary - Presentation: Elevator pitch
November	- Main technologies selected - project is well-defined - Everyone is actively coding	- Answer all questions needed to complete TDD - Lot's of coding for alpha review	- Writing: PRD/TDD - Presentation: Project Design
December	- Code complete for alpha review	- more coding for demo 2 - Formalize design discussions into proper TDD	- Alpha review - Presentation: Alpha prototype - Writing: revised PRD/TDD
January	- Continued focus on project development	- continued development for demo 2 - focus on proper testing & integration	- demo 2
February	- Code complete for demo 2	- Refine code from a prototype into a fleshed out project -- testing, integration, polishing - continued development for prelim prototype (get as close to finished as you can here)	- Presentation: skill refinement - demo 3
March	- Code complete for demo 3	- final code polishing to wrap up project - complete any necessary integration work - add extra features if possible	- demo 4
April	**- Code 99% complete for final demo**	- finishing touches for final project submission - ideally you are done with coding by this point	- R&D Showcase - Final Presentation & Demo
May			- Final package due

Agenda

- Fall Semester Feedback
- Spring Semester
  - What's new
  - Spring Schedule
  - **Grading Criteria**
  - Sprint Expectations
  - Setting up Spring Sprints
- Upcoming Deadlines
- Project Goals

Grading Criteria

- Presentations **10%**
  - Presentation 4: Presentation Skills **5%**
  - Practice final presentation **5%**
- Sprint Progress & Demos **39%**
  - [3x] Demos **21%**
  - [3x] Sprints **18%**
    - Weekly updates
    - Sprint boards
    - Code commits & PR reviews
- Final Project Demo / Presentation **35%**
  - Final Presentation **10%**
  - Final Demo **25%**
- Final Package **10%**
- Participation **6%**
  - Attendance
  - Team surveys

Agenda

- Fall Semester Feedback
- Spring Semester
  - What's new
  - Spring Schedule
  - Grading Criteria
  - **Sprint Expectations**
  - Setting up Spring Sprints
- Upcoming Deadlines
- Project Goals

Sprint Schedule

Last Semester Sprints

September Sprint

October Sprint

November / December Sprint

Spring Semester Sprints

January Sprint (demo 2)

February Sprint (demo 3)

March Sprint (demo 4)

April Sprint* (2 weeks!)

Sprint Schedule

Sprint Components

1. **Define Sprint Goals (instructors)**
2. **Sprint planning (mentors)**
3. **Execution (code reviews)**
4. **Demo (individual & team)**

Spring Semester Sprints

January Sprint (demo 2)

February Sprint (demo 3)

March Sprint (demo 4)

April Sprint* (2 weeks!)

Expectations: Instructor Meetings

- **In the fall**: status updates, somewhat redundant with mentor meetings

- **In the spring**: focus on high level planning or technical details
  - Refine sprint goals the week before each sprint
  - Come with concrete questions

Expectations: Mentor Meetings

- **In the fall**: mentors came up with the agenda and led the meeting

- **In the spring**: students should come up with the agenda and lead the meeting
  - Sprint planning
  - PR reviews
  - Lead with weekly updates
  - Come with concrete questions

Expectations: Code

- All students should contribute code during each sprint
- Code should be pushed to feature branches and PRed to main
- **We will only evaluate code merged to main**
  - We will not look at code in feature branches
  - You will not receive full credit for code committed directly to main
- We use commit history for grading, so...
  - **The author of the PR should be the one to merge it**
  - **A single branch should contain only a single person's commits – squash-merge assigns all code to the PR merger**
  - **If your github account isn't associated with your commits, you will not get credit**
- Link PRs to tickets

Expectations: PR Reviews

- For the January & March sprints, we require **at least one** peer code review
  - Each team member must **author** and **review (leave comments)** on one PR before the end of the sprint
  - PRs should have descriptions describing the requested change

- For the February sprint, we require **at least one** mentor code review
  - During your weekly mentor meeting, the team should choose 1 PR to review with your mentor as a group. The mentor should leave at least one comment on the PR so we know they reviewed it with you.

Expectations: Demos

- In your instructor meeting at the end of the sprint, you'll demo what you accomplished and if you hit your goals
- We expect to see code from all students
- Demo grades are given individually, but the format of the demos alternate between individual & as a group
    - January: individual
    - February: group
    - March: individual
    - Final demo: group

Expectations: Sprint Board

- Create tickets to capture project-specific work.
    - Create project-specific epics to organize work
- Tickets should include:
    - Descriptions + definition of done
    - Assignees
    - Due dates
    - Sprint
    - Status
    - Linked PR (when there is code)
- While it is ok to have some tickets assigned to multiple team members, the majority of tickets should be assigned to one person
- **All tickets should be completed, moved to next sprint, or marked as “won’t do” by the end of the sprint**

Expectations: Sprint Board

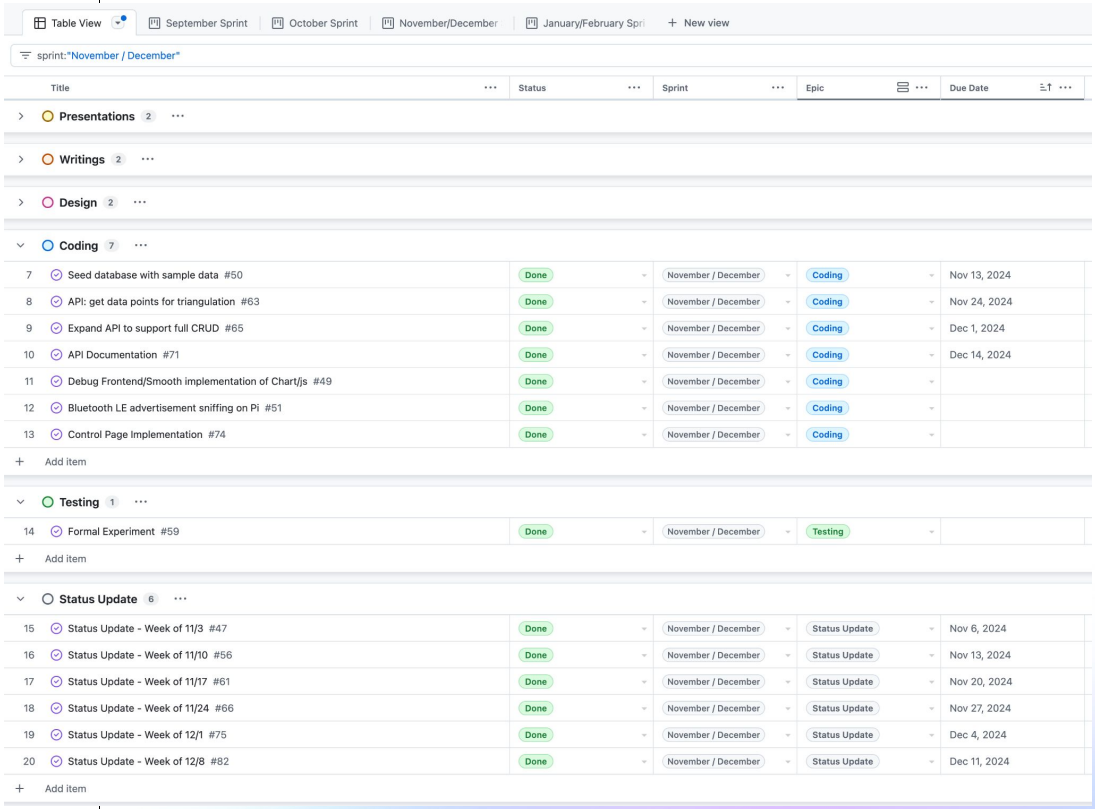


Table View | September Sprint | October Sprint | November/December | January/February Spri | + New view

sprint:"November / December"

	Title	Status	Sprint	Epic	Due Date
	Presentations 2				
	Writings 2				
	Design 2				
	Coding 7				
7	Seed database with sample data #50	Done	November / December	Coding	Nov 13, 2024
8	API: get data points for triangulation #63	Done	November / December	Coding	Nov 24, 2024
9	Expand API to support full CRUD #65	Done	November / December	Coding	Dec 1, 2024
10	API Documentation #71	Done	November / December	Coding	Dec 14, 2024
11	Debug Frontend/Smooth implementation of Chart/js #49	Done	November / December	Coding	
12	Bluetooth LE advertisement sniffing on Pi #51	Done	November / December	Coding	
13	Control Page Implementation #74	Done	November / December	Coding	
	+ Add item				
	Testing 1				
14	Formal Experiment #59	Done	November / December	Testing	
	+ Add item				
	Status Update 6				
15	Status Update - Week of 11/3 #47	Done	November / December	Status Update	Nov 6, 2024
16	Status Update - Week of 11/10 #56	Done	November / December	Status Update	Nov 13, 2024
17	Status Update - Week of 11/17 #61	Done	November / December	Status Update	Nov 20, 2024
18	Status Update - Week of 11/24 #66	Done	November / December	Status Update	Nov 27, 2024
19	Status Update - Week of 12/1 #75	Done	November / December	Status Update	Dec 4, 2024
20	Status Update - Week of 12/8 #82	Done	November / December	Status Update	Dec 11, 2024
	+ Add item				

Expand API to support full CRUD #65

Edit

Closed | #70 | GW-CS-SD-24-25/ | Private

opened on Nov 23, 2024 · edited by | Edits

Expand the API (and database) to support all endpoints listed in #62:

Endpoint	Method	Parameters (type)	Response
/device_history(add a new sensor reading)	POST	device_mac (string)sensor_id (int)strength (float)timestamp? (datetime)Note: can POST an array of these parameters to insert multiple at the same time	HTTP status code 200 if successful, appropriate code if not (with error message)
/layouts	GET	parent? (int)id? (int)	List of layouts, optionally filtered by parent ID, or specific layout
/layouts(to update a specific layout's details)	POST	name? (string)parent? (int)scale? (float)map? (string)lat? (float)long? (float)	HTTP status code 200 if successful, appropriate code if not (with error message)
/layouts	POST	name (string)parent? (int)scale? (float)map? (string)lat? (float)long? (float)	HTTP status code 200 if successful, appropriate code if not (with error message)
/layouts/{id}/devices	GET	id (int)	List of devices and locations currently seen in a layout (using triangulation)
/devices	GET	mac? (string)	List of identified devices with details, or specific device
/devices	POST	mac (string)name? (string)hide? (boolean)	HTTP status code 200 if successful, appropriate code if not (with error message)

Assignees

Labels
No labels

Projects
Status: Done
Sprint: November / December • Nov 3 – Dec 20
Epic: Coding
Due Date: Dec 1, 2024

Milestone
No milestone

Development
Expanded API

Notifications | Customize
Subscribe
You're not receiving notifications from this thread.

Participants

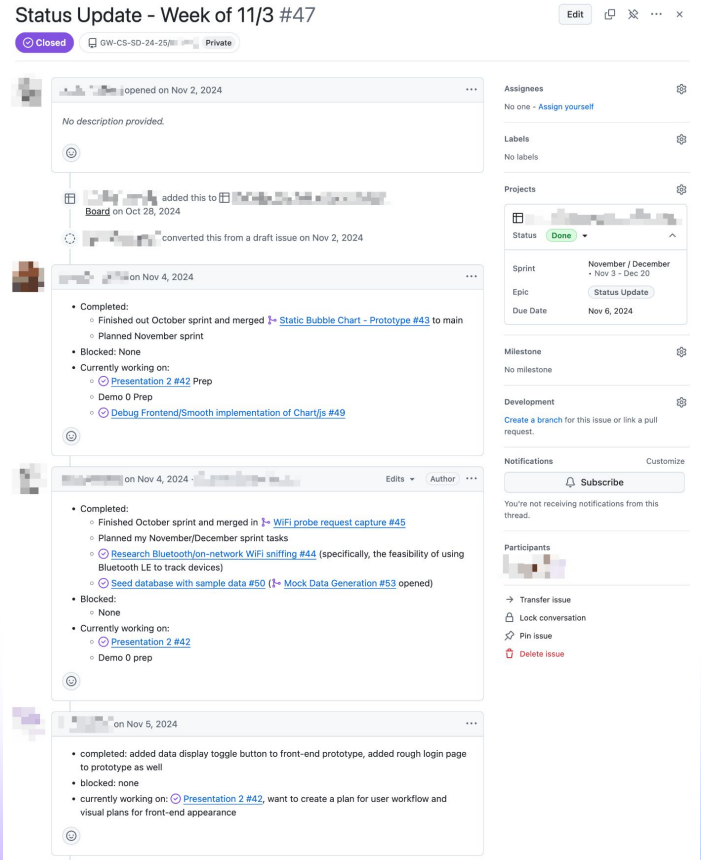
Transfer issue
Lock conversation
Pin issue
Delete issue

Expectations: Weekly Status Updates

- Create a new status update ticket for each week
  - Title should be **Status Update - Week of MM/YY** with the date matching the Monday date on the course website
  - Epic should be **status update**
  - Due date should be the following Sunday
- Move ticket from TODO to DONE as week progresses
- Students should post weekly status updates covering:
  - What they completed (can link to other tickets)
  - What they are blocked by
  - What they are currently working on
  - **Each student must leave their own comment (do not update the description) before the due date to receive full credit**

Recommendation: Create all status update tickets at the beginning of the sprint

Expectations: Weekly Status Updates



Sprint Grading Rubric

- **Sprint Boards (2 pts)**
  - **Full credit:**
    - All tickets have correct metadata (epics, due date, sprint assignee) & moved to done/won't do
    - individual has sufficient # of individual, project-specific tickets assigned (> 3)
    - Tickets have clear descriptions w/ acceptance criteria
  - **-0.1 - 0.25pt** if tickets left in progress / todo, missing metadata (epics, assignees, etc)
  - **-0.25 - 0.5pt** if tickets don't include clear descriptions & acceptance criteria
  - **-0.5 - 1pt** if not enough tickets in sprint
  - **0pt** if no tickets / not enough evidence of sprint progress

Sprint Grading Rubric

- **Weekly Updates (1 pt)**
  - Total point value divided by number of weeks (factoring out holidays)
  - **½ credit** if weekly update late
  - **½ credit** if format is incorrect (ticket metadata incorrect, comments posted as description)
  - **¼ credit** if weekly update late AND format is incorrect
  - **0 - ½ credit** if weekly update content insufficient

Sprint Grading Rubric

- **Github (2 pts)**
  - Sufficient code merged to main, PRs used w/ clear descriptions
  - **-0.1 - 0.25 pt** if PRs missing descriptions
  - **-0.25 - 0.5 pt** if minimal/no PRs used, but code pushed to main
  - **-1 pt** if minimal code written / only documentation committed

Sprint Grading Rubric

- **Peer PR Review (1 pt)**
  - **+0.5 pt for authoring a PR (no partial credit)**
  - **+0.5 pt for reviewing a PR (must include comments & feedback, not just an approval)**
    - **-0.25 if review is late**

- **Mentor PR Review (1 pt)**
  - **+1 pt for mentor reviewing a PR (must leave at least 1 comment)**
  - **No partial credit**

Expectations: Passing the course

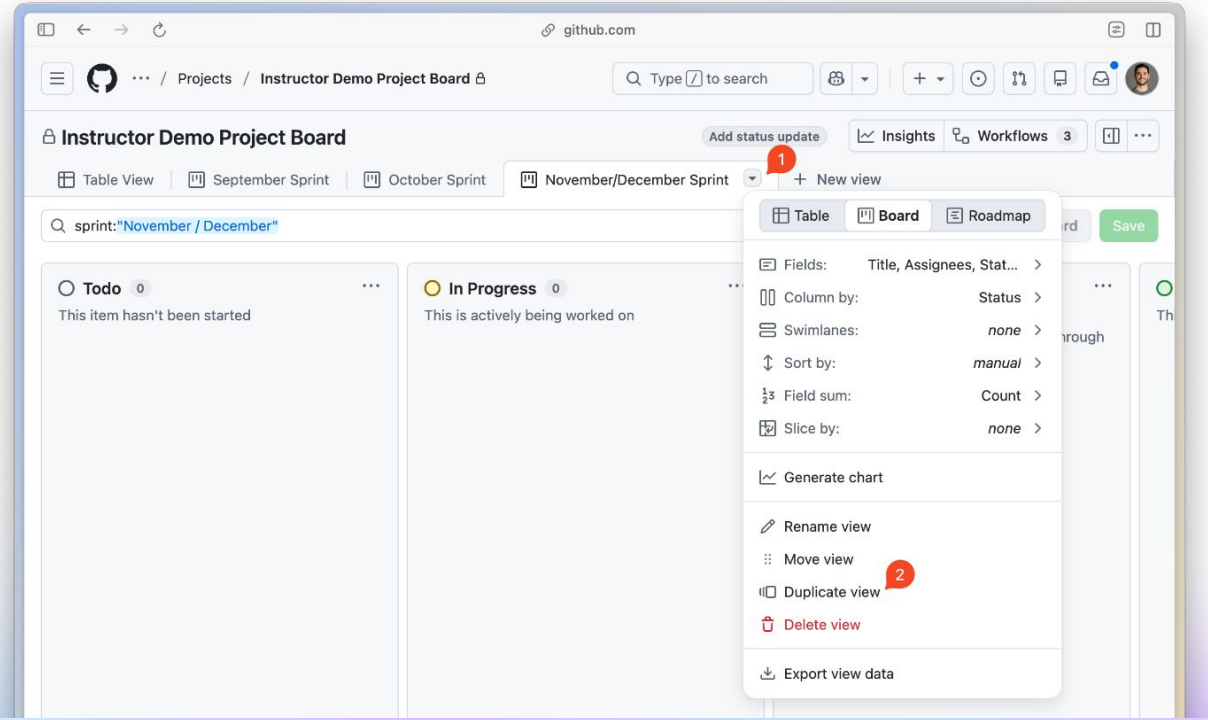
- **We really want you to graduate! But our job is also to act as gate keepers that ensure a GW CS degree is a meaningful certification.**
  - Evidence that you can: build a significant software project, work on a team, follow software engineering best practices, solve problems that go beyond simple "boot camp" programming
- Most students pass senior design, but it is not a guarantee
  - If we have concerns about your progress, we will try to discuss with you as early as we can, but sometimes we cannot catch things (especially if you are not being honest about your progress)
  - If we realize in April that your contribution to the project is trivial and not representative of two semester of work, **you will fail the course**.
  - Help yourself and teammates by alerting us of issues soon!

Agenda

- Fall Semester Feedback
- Spring Semester
  - What's new
  - Spring Schedule
  - Grading Criteria
  - Sprint Expectations
  - **Setting up Spring Sprints**
- Upcoming Deadlines
- Project Goals

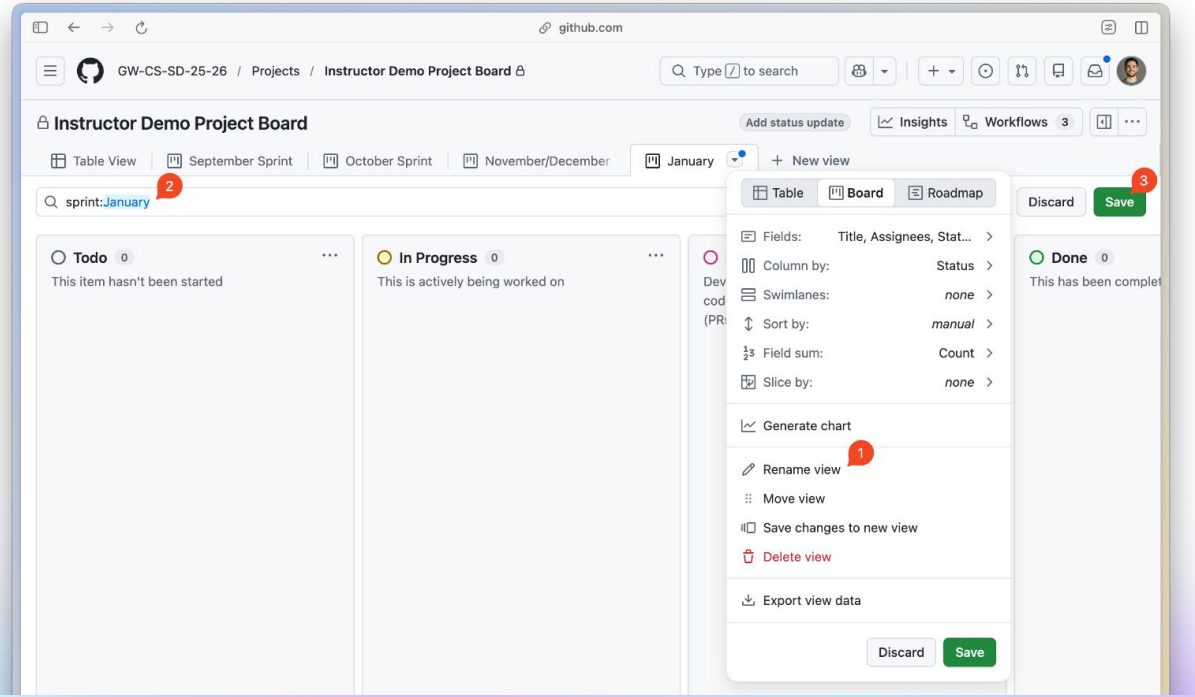
Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprint Boards



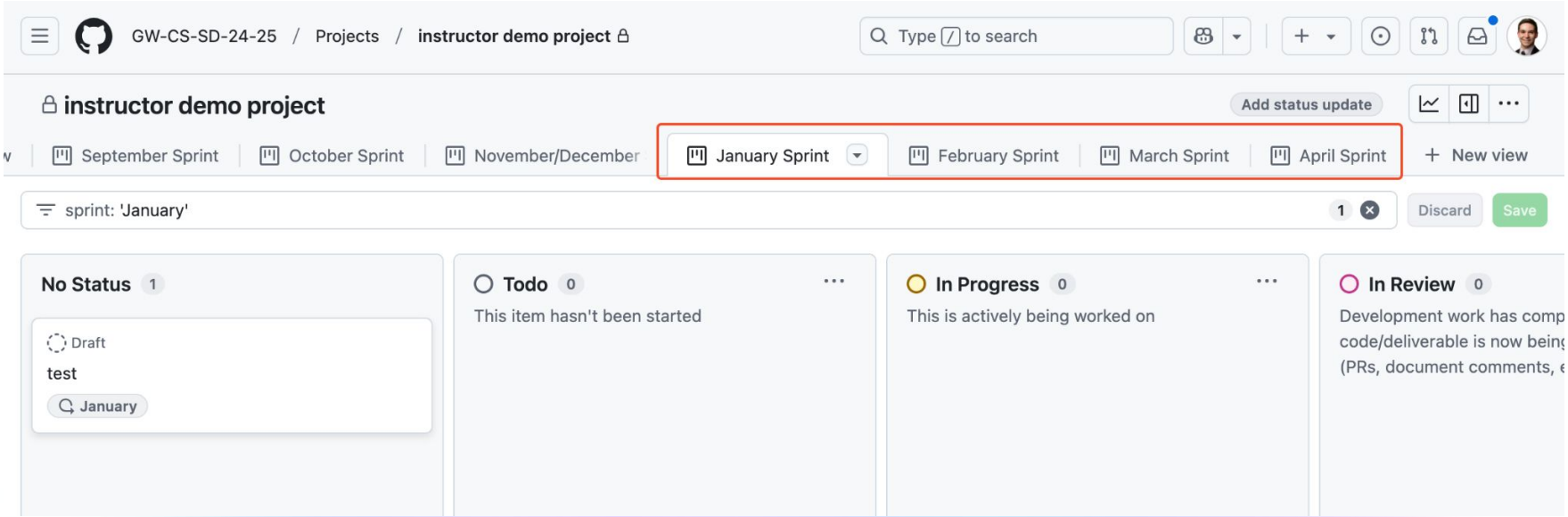
Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprint Boards



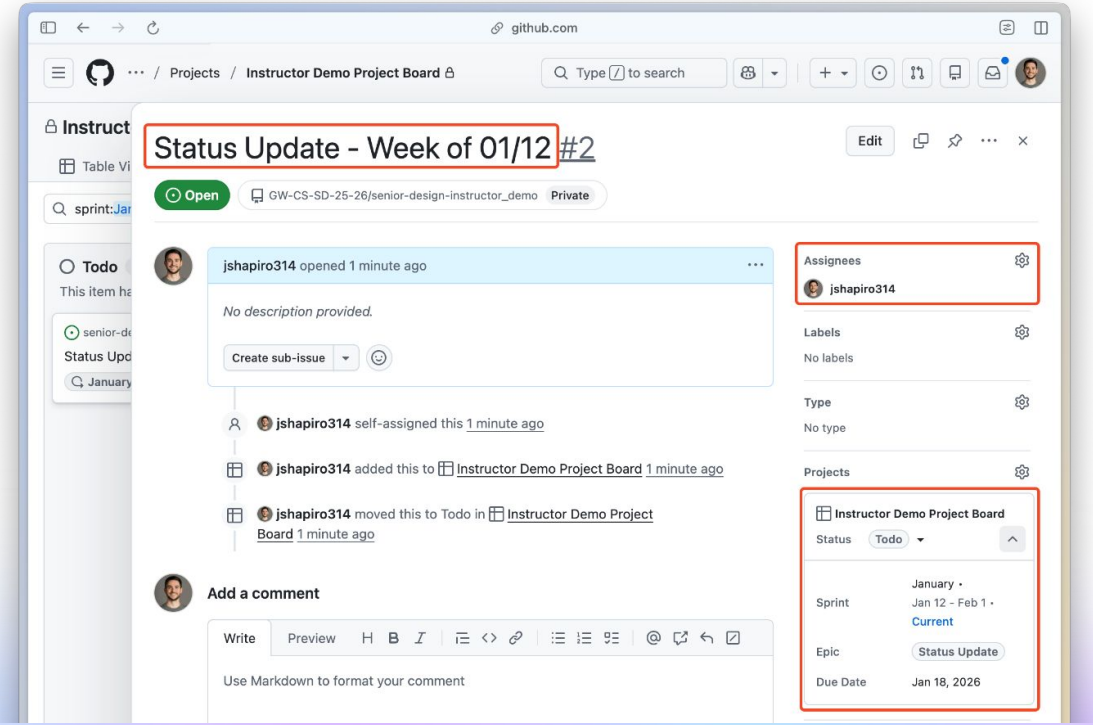
Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprint Boards



Exercise: Prepare for upcoming sprints

2. Create weekly status update ticket(s)



Agenda

- Fall Semester Feedback
- Spring Semester Expectations
  - What's new
  - Spring Schedule
  - Grading Criteria
  - Sprint Expectations
  - Setting up Spring Sprints
- **Upcoming Deadlines**
- Project Goals

For Next Week

Weekly Focus

- Write code for demo 2
- Peer PR reviews

Mentor Meetings

- [Team]: Sprint planning for January

Deadlines

- [Team]: Peer PR reviews (**2/2)**
- [Team]: End of January Sprint (**2/2)**
- [Individual]: Demo 2 (**week of 2/3**)

Reminders

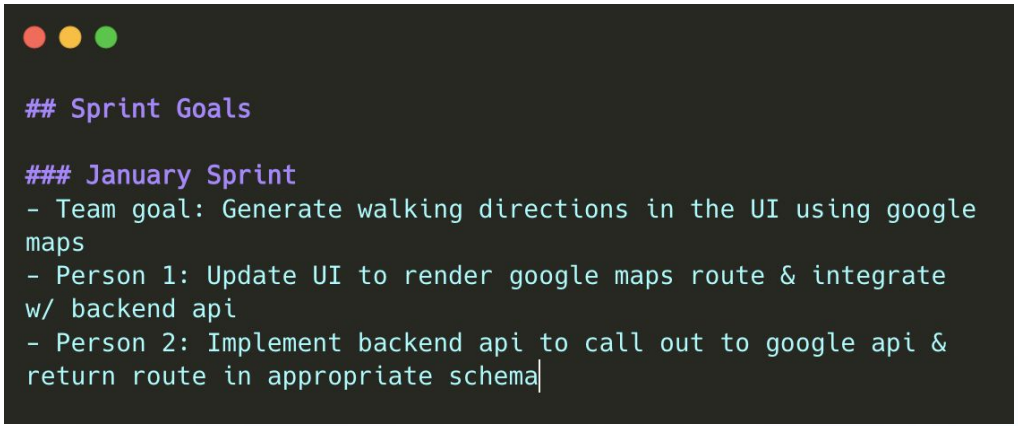
- Don’t forget to post weekly updates, including for this week!

Agenda

- Fall Semester Feedback
- Spring Semester Expectations
  - What's new
  - Spring Schedule
  - Grading Criteria
  - Sprint Expectations
  - Setting up Spring Sprints
- Upcoming Deadlines
- **Project Goals**
- **Github Account Verification**

Defining Sprint Goals

1. As a team, come up with sprint goals between now and the final demo (April)
2. List these goals in your github readme



```
## Sprint Goals

### January Sprint
- Team goal: Generate walking directions in the UI using google
maps
- Person 1: Update UI to render google maps route & integrate
w/ backend api
- Person 2: Implement backend api to call out to google api &
return route in appropriate schema
```

3. Review goals with instructor before leaving lab
4. Revisit & update these goals at the beginning of each sprint