

Lecture 11: January 15

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



Fall Semester Feedback

- Really nice job overall!
- Clear improvements in presentation skills
- Majority of projects are on track
- Good use of sprint boards, code commits

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
 - More frequent sprints, with demos immediately after

Spring Schedule

<https://gw-cs-sd-24-25.github.io/>

Month	Expected Status	Monthly Focus	Deliverables
October	- Project selected & approved by instructors	- Begin technical investigations (services, apis, programming language, etc) - Flesh out project functionality & requirements - Coding should start (scaffolding, ci/cd, prototyping)	- Writing: Executive Summary - Updated Gantt Chart - Teamwork survey - Writing: Technical summary - Presentation: Elevator pitch - Presentation: Project design
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 demo	- Writing: PRD
December	- Code complete for alpha demo	- more coding for beta demo - Formalize design discussions into proper TDD	- Presentation: Alpha prototype - Writing: TDD
January	- Continued focus on project development	- continued development for demo 2 - focus on proper testing & integration	- Demo 2 (individual)
February	- Code complete for demo 2	- Refine code from a prototype into a fleshed out project -- testing, integration, polishing - continued development for demo 3 (get as close to finished as you can here)	- Presentation 4 - Demo 3 (team)
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 (individual)
April	- Code 99% complete for final demo	- finishing touches for final project submission - ideally you are done with coding by this point	- Final Demo - Final Presentation - R&D Showcase
May			- Final package due (team website)

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
 - 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**
- Link PRs to tickets if possible



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



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



Spring Sprint Progress Rubric

Full credit

- Tickets addressed as either “done”, “won’t do”, or moved to next sprint.
- Weekly standup updates & slack participation
- Code is PRed & merged to master. Branches & PRs are well-scoped. PRs are linked to tickets.
- Each individual has authored & reviewed a PR

Partial credit

- Majority of tickets addressed as either “done”, “won’t do”, or moved to next sprint.
- Occasional standup updates & moderate participation
- Code is committed, PRs are sometimes present and sometimes well-scoped. PRs are sometimes linked to tickets.

Minimal credit

- Few tickets addressed as either “done”, “won’t do”, or moved to next sprint.
- Minimal standup updates & rare participation
- Minimal code is committed, PRs are missing or not well-scoped.

No credit

- No sprint board activity
- No standup updates
- No slack participation
- No code committed to master/main

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 Sprint | + 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

opened on Nov 23, 2024 - edited by Edits ...

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: Subscribe

You're not receiving notifications from this thread.

Participants: Transfer issue, Lock conversation, Pin issue, Delete issue

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)

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 Sunday date on the course website
 - Epic should be **status update**
- 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



Status Update - Week of 11/3 #47

Closed OW-CS-SD-24-25 Private Edit 📄 🔗 ⋮ ×

opened on Nov 2, 2024

No description provided.

added this to **Board** on Oct 28, 2024

converted this from a draft issue on Nov 2, 2024

on Nov 4, 2024

- Completed:
 - Finished out October sprint and merged [1](#)-> [Static Bubble Chart - Prototype #43](#) to main
 - Planned November sprint
- Blocked: None
- Currently working on:
 - [Presentation 2 #42](#) Prep
 - Demo 0 Prep
 - [Debug Frontend/Smooth implementation of Chart/js #49](#)

on Nov 4, 2024

- Completed:
 - Finished October sprint and merged in [1](#)-> [WiFi probe request capture #45](#)
 - Planned my November/December sprint tasks
 - [Research Bluetooth/on-network WiFi sniffing #44](#) (specifically, the feasibility of using Bluetooth LE to track devices)
 - [Seed database with sample data #50](#) (=> [Mock Data Generation #53](#) opened)
- Blocked:
 - None
- Currently working on:
 - [Presentation 2 #42](#)
 - Demo 0 prep

on Nov 5, 2024

- completed: added data display toggle button to front-end prototype, added rough login page to prototype as well
- blocked: none
- currently working on: [Presentation 2 #42](#), want to create a plan for user workflow and visual plans for front-end appearance

Assignees
No one - [Assign yourself](#)

Labels
No labels

Projects

Status **Done**

Sprint November / December
Nov 3 - Dec 20

Epic [Status Update](#)

Due Date Nov 6, 2024

Milestone
No milestone

Development
[Create a branch](#) for this issue or link a pull request.

Notifications Customize
[Subscribe](#)

You're not receiving notifications from this thread.

Participants

- Transfer issue
- 🔒 Lock conversation
- 📌 Pin issue
- 🗑️ Delete issue

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!

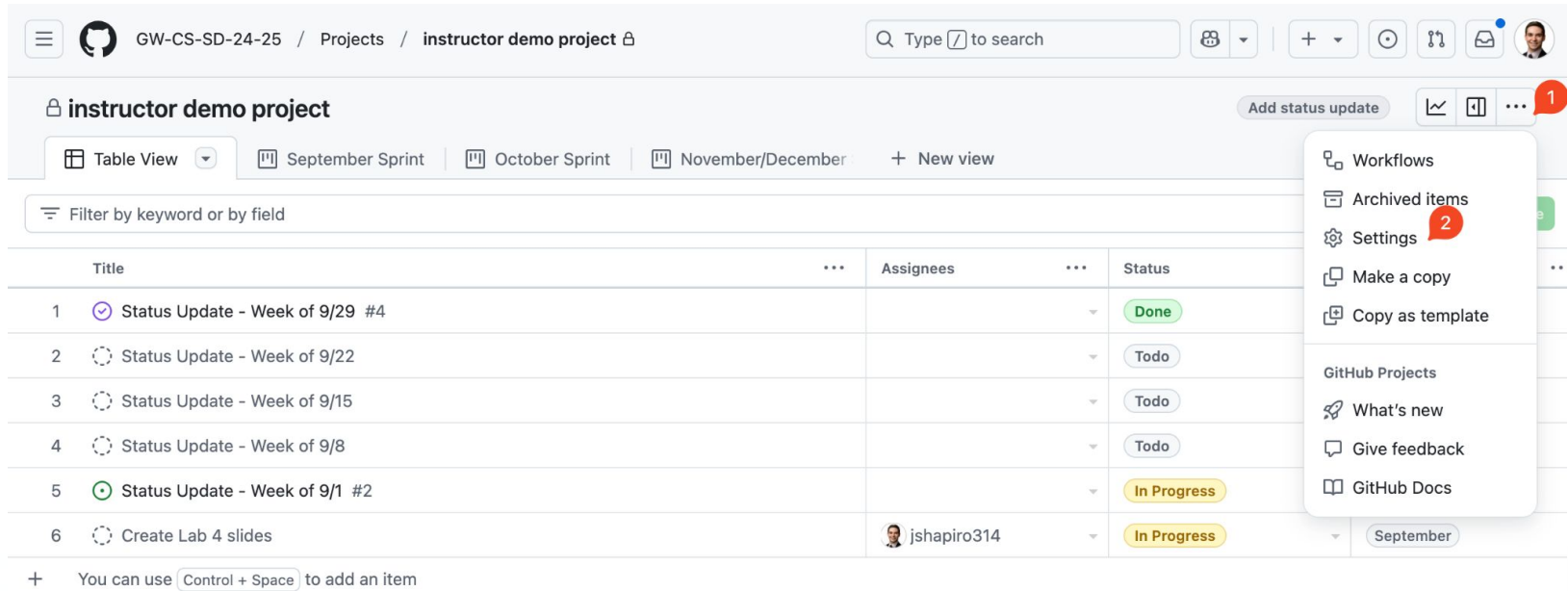
Agenda

- 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 Sprints



The screenshot shows the GitHub Projects interface for the 'instructor demo project'. The board is in 'Table View' and displays a list of tasks. A settings menu is open, showing options like 'Workflows', 'Archived items', and 'Settings'. The 'Settings' option has a red notification badge with the number '2'. The tasks are listed in a table with columns for Title, Assignees, and Status.

Title	Assignees	Status
1 Status Update - Week of 9/29 #4		Done
2 Status Update - Week of 9/22		Todo
3 Status Update - Week of 9/15		Todo
4 Status Update - Week of 9/8		Todo
5 Status Update - Week of 9/1 #2		In Progress
6 Create Lab 4 slides	jshapiro314	In Progress

Below the table, there is a note: '+ You can use `Control + Space` to add an item'

Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprints

The screenshot shows the Jira 'Settings' page for the 'instructor demo project'. The left sidebar contains navigation options: Project settings, Manage access, Custom fields (+ New Field), Status, **Sprint** (with a red notification bubble '1'), Epic, and Due Date. The main content area is titled 'Sprint field settings' and includes a 'Delete field' button. Under 'Field name', there is a text input containing 'Sprint'. Under 'Field type', there is a dropdown menu with 'Iteration' selected. Below this, a summary bar shows '0 Active' and '3 Completed' items, with a '+ Add iteration' button (marked with a red notification bubble '2') and a 'More options' dropdown. A message states: 'You're out of iterations. Create an iteration using the "+ Add iteration" button above.' At the bottom, there are 'Save' and 'Reset' buttons.



Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprints

The screenshot shows the Jira 'Settings' page for the 'instructor demo project'. The left sidebar lists 'Project settings', 'Manage access', 'Custom fields', 'Status', 'Sprint', 'Epic', and 'Due Date'. The 'Sprint' field is selected, showing its configuration: 'Field name' is 'Sprint', 'Field type' is 'Iteration', and it has '1 Active' and '3 Completed' iterations. A 'Break' iteration is shown for '22 days Dec 22, 2024 - Jan 12'. The 'Current' iteration is 'January', which is '3 weeks Jan 13 - Feb 02'. A calendar pop-up is open, showing 'January 2025' and 'February 2025'. The calendar highlights the dates from Jan 13 to Feb 2, with 'Jan 13' selected. The 'Apply' button is visible at the bottom of the calendar.

GW-CS-SD-24-25 / Projects / instructor demo project

Settings

Project settings
Manage access

Custom fields + New Field

Status
Sprint
Epic
Due Date

Sprint field settings [Delete field](#)

Field name
Sprint

Field type
Iteration

1 Active 3 Completed + Add iteration More options

Break 22 days Dec 22, 2024 - Jan 12

Current January 3 weeks Jan 13 - Feb 02

Save Reset

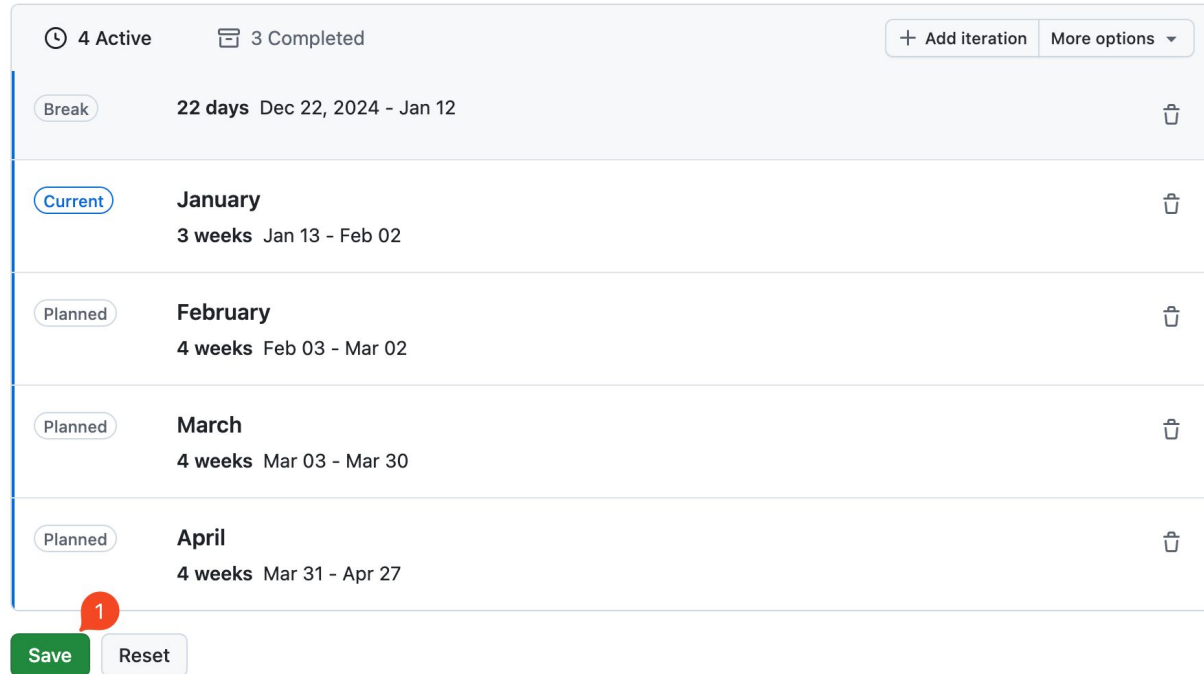
January 2025 February 2025

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4							1
5	6	7	8	9	10	11	2	3	4	5	6	7	8
12	13	14	15	16	17	18	9	10	11	12	13	14	15
19	20	21	22	23	24	25	16	17	18	19	20	21	22
26	27	28	29	30	31		23	24	25	26	27	28	

Today Jan 13 - Feb 2 Apply

Exercise: Prepare for upcoming sprints

1. Create Spring Semester Sprints



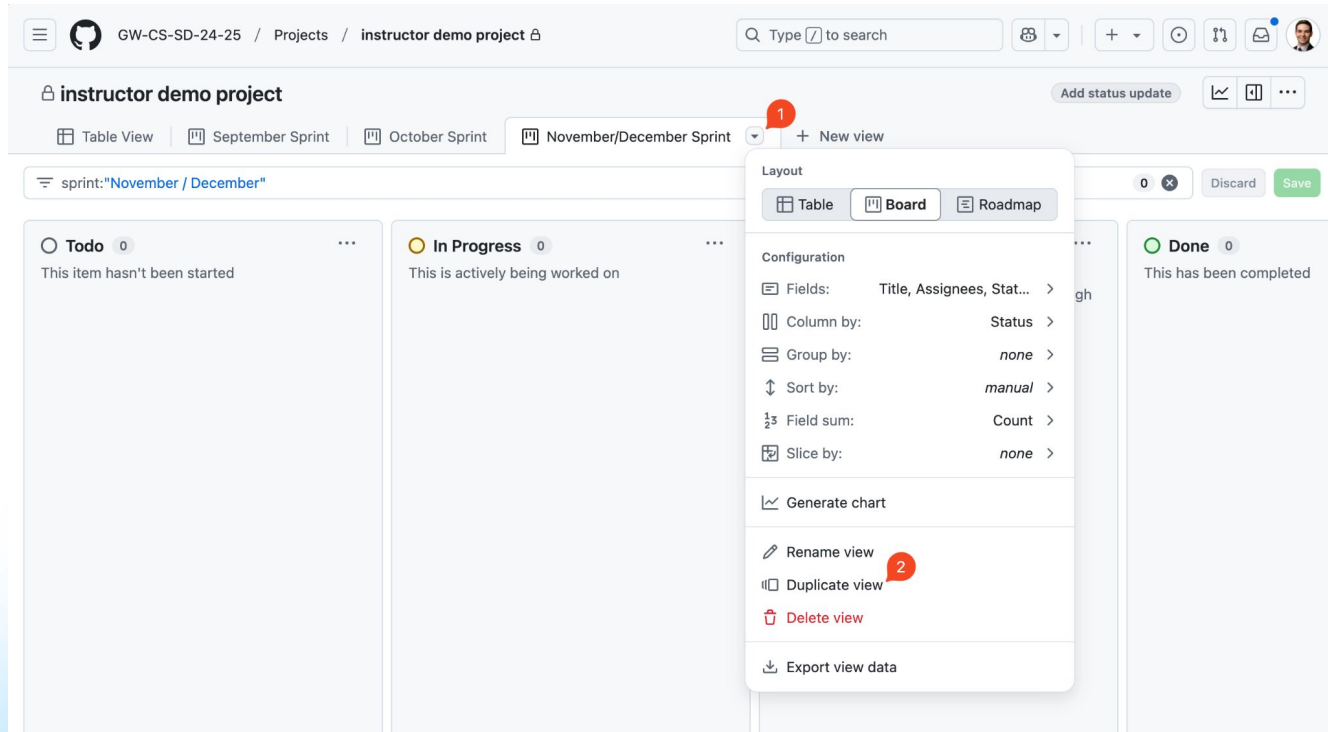
The screenshot displays a sprint planning interface with the following elements:

- Summary: 4 Active (clock icon), 3 Completed (calendar icon). Buttons: + Add iteration, More options (dropdown arrow).
- Sprint List:
 - Break**: 22 days, Dec 22, 2024 - Jan 12. Status: Break. Action: Delete (trash icon).
 - January**: 3 weeks, Jan 13 - Feb 02. Status: Current (blue outline). Action: Delete (trash icon).
 - February**: 4 weeks, Feb 03 - Mar 02. Status: Planned. Action: Delete (trash icon).
 - March**: 4 weeks, Mar 03 - Mar 30. Status: Planned. Action: Delete (trash icon).
 - April**: 4 weeks, Mar 31 - Apr 27. Status: Planned. Action: Delete (trash icon).
- Bottom: Save (green button), Reset (grey button). A red notification bubble with the number '1' is positioned above the Save button.



Exercise: Prepare for upcoming sprints

2. Create Spring Semester Sprint Boards



The screenshot shows a Jira project page for "instructor demo project". The top navigation bar includes the project name, a search bar, and user profile. Below the navigation, the project name is repeated, followed by a "Add status update" button and a menu icon. The main content area displays a Kanban board for the "November/December Sprint". The board has three columns: "Todo" (0 items), "In Progress" (0 items), and "Done" (0 items). A configuration menu is open over the board, showing options for "Layout" (Table, Board, Roadmap), "Configuration" (Fields, Column by, Group by, Sort by, Field sum, Slice by), "Generate chart", "Rename view", "Duplicate view", "Delete view", and "Export view data". A red circle with the number "1" is placed over the "Board" layout option, and another red circle with the number "2" is placed over the "Duplicate view" option.

GW-CS-SD-24-25 / Projects / instructor demo project

Type [Z] to search

instructor demo project

Add status update

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

sprint:"November / December"

0 x Discard Save

Layout

- Table
- Board
- Roadmap

Configuration

- Fields: Title, Assignees, Stat... >
- Column by: Status >
- Group by: none >
- Sort by: manual >
- Field sum: Count >
- Slice by: none >

Generate chart

- Rename view
- Duplicate view
- Delete view
- Export view data

Exercise: Prepare for upcoming sprints

2. Create Spring Semester Sprint Boards

The screenshot shows a Jira project page for "instructor demo project". The breadcrumb trail is "GW-CS-SD-24-25 / Projects / instructor demo project". A search bar is present with the text "Type [Z] to search". The project name "instructor demo project" is displayed with a lock icon and an "Add status update" button. Below this, there are view tabs: "Table View", "September Sprint", "October Sprint", "November/December", and "January Sprint". A "+ New view" button is also visible. A search filter "sprint: 'January'" is applied, with a red notification bubble containing the number "2".

The main content area is divided into four columns representing sprint stages: "No Status" (1 item), "Todo" (0 items), "In Progress" (1 item), and "Review" (0 items). The "No Status" column contains a card for a "Draft" item titled "test" with a "January" filter. The "In Progress" column contains a card for an "In Progress" item with the text "This is active".

A configuration menu is open over the "In Progress" column. It has a red notification bubble with the number "1". The menu is titled "Layout" and includes the following options:

- Table
- Board (selected)
- Roadmap

Under the "Configuration" section, the following settings are visible:

- Fields: Title, Assignees, Stat... >
- Column by: Status >
- Group by: none >
- Sort by: manual >
- Field sum: Count >
- Slice by: none >

At the bottom of the menu, there are options for "Generate chart", "Rename view", "Duplicate view", "Delete view", and "Export view data".

On the right side of the page, there are "Discard" and "Save" buttons, with a red notification bubble containing the number "3" above the "Save" button.



Exercise: Prepare for upcoming sprints

2. Create Spring Semester Sprint Boards

The screenshot shows the GitHub project dashboard for 'instructor demo project'. The breadcrumb navigation is 'GW-CS-SD-24-25 / Projects / instructor demo project'. A search bar contains 'Type / to search'. The project name 'instructor demo project' is displayed with an 'Add status update' button and icons for a line graph, a calendar, and a menu. Below this, a row of sprint boards is shown: 'September Sprint', 'October Sprint', 'November/December', 'January Sprint' (highlighted with a red box), 'February Sprint', 'March Sprint', and 'April Sprint'. A '+ New view' button is on the right. A filter bar shows 'sprint: 'January'' with a count of '1' and 'Discard' and 'Save' buttons. The main content area has four columns: 'No Status' (1 item, 'Draft test', 'January'), 'Todo' (0 items, 'This item hasn't been started'), 'In Progress' (0 items, 'This is actively being worked on'), and 'In Review' (0 items, 'Development work has comp code/deliverable is now being (PRs, document comments, e

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 (due EOD!)



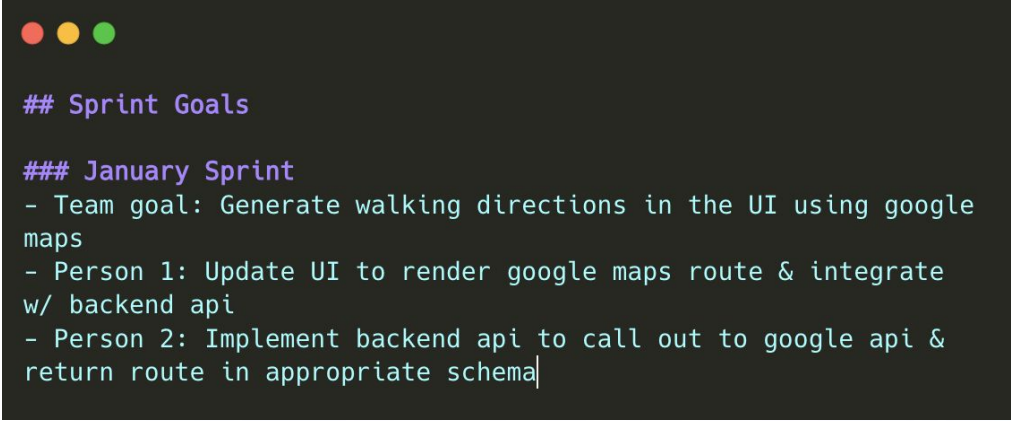
Agenda

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



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