Project Kickoff!

CS 346: Application
Development

Welcome to Sprint 1!

You first iteration (aka sprint) starts today!

• Day 1	Planning	Deciding what to implement
• Days 2-9	Design/coding	Implementation, documentation
• Day 10	Demo	Demo to your TA, submit a release

See the Project Deliverables > Project iterations for details.

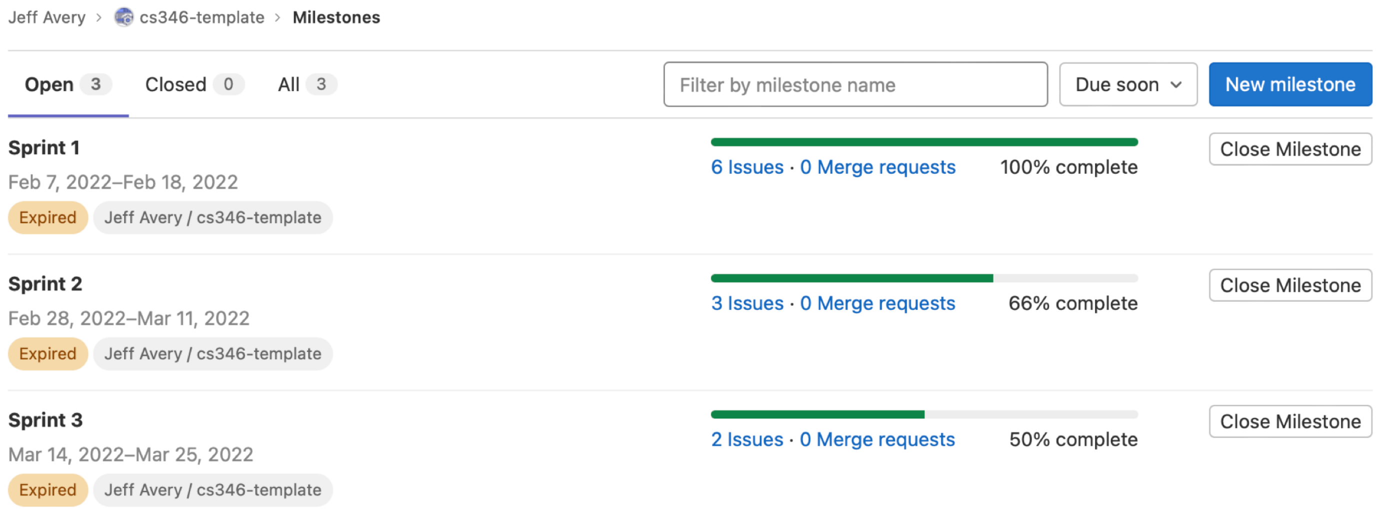
Setup

What you should do first.

Establish Milestones

Create **milestones** are for each of your four software releases.

`Plan > Milestones`



Sprint 1
Feb 7, 2022–Feb 18, 2022
Expired · Jeff Avery / cs346-template
6 Issues · 0 Merge requests · 100% complete

Sprint 2
Feb 28, 2022–Mar 11, 2022
Expired · Jeff Avery / cs346-template
3 Issues · 0 Merge requests · 66% complete

Sprint 3
Mar 14, 2022–Mar 25, 2022
Expired · Jeff Avery / cs346-template
2 Issues · 0 Merge requests · 50% complete

Create Issues

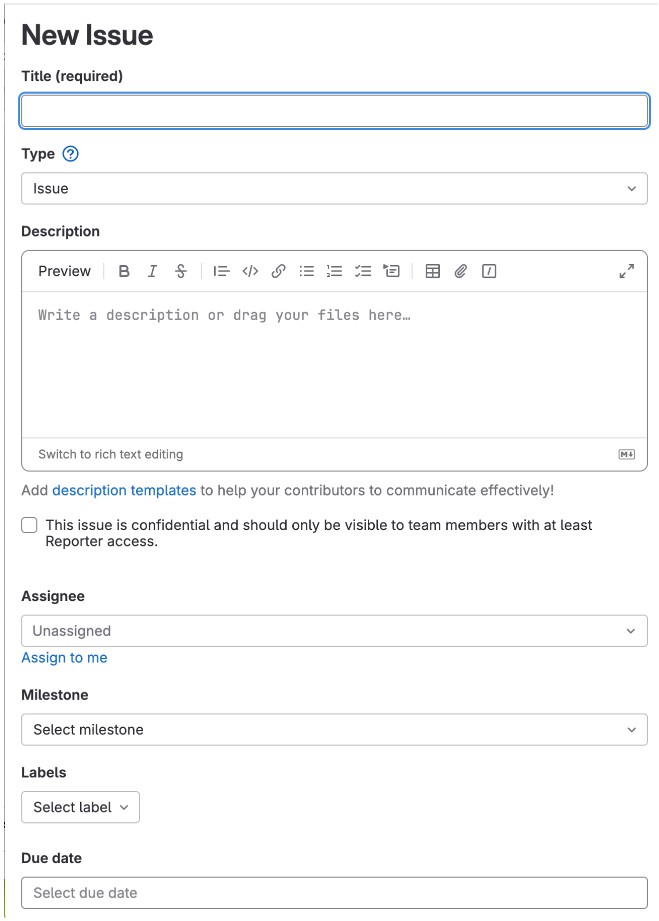
Issues represent the tasks that you need to complete for your project.

- `Plan > Issues`

Every feature from your solution (design proposal) should be logged as an issue.

All new issues should be

- Not assigned to a milestone.
- Not assigned to a person.



New Issue

Title (required)

Type

Issue

Description

Write a description or drag your files here…

Add description templates to help your contributors to communicate effectively!

☐ This issue is confidential and should only be visible to team members with at least Reporter access.

Assignee

Unassigned

Assign to me

Milestone

Select milestone

Labels

Select label

Due date

Select due date

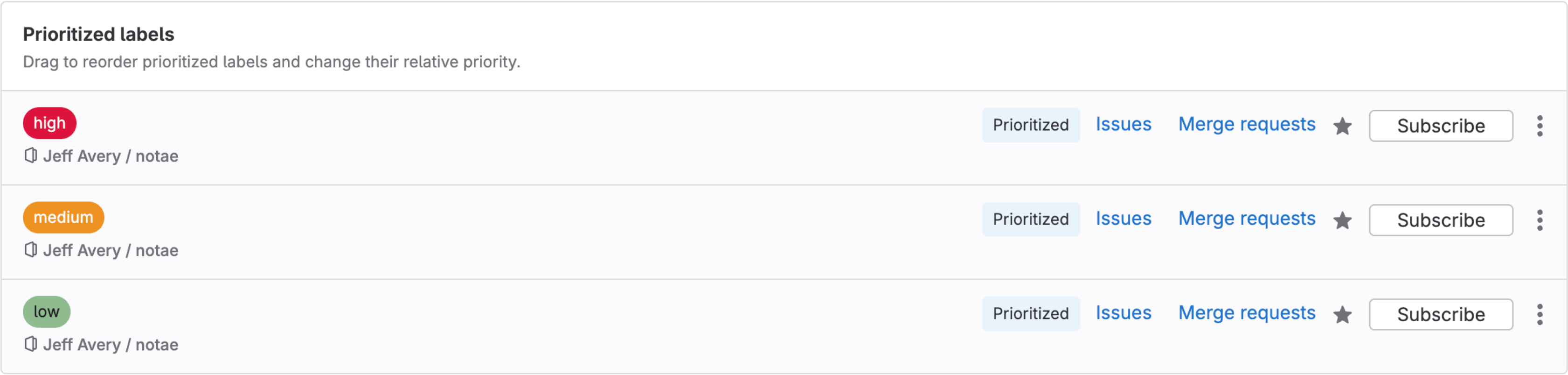
Guidelines for creating issues

- Issues should “fall out” from your proposed solution.
  - Clearly link your issues to your user stories and your design proposal.
  - Issues should be created AFTER the design proposal is submitted.
  - **You may need multiple issues per feature!**
- Each issue should represent about 0.5 days of work (or less).
- Each issue can be implemented and tested independently.
- Include as much detail as you know. This is the main (only?) source of information on what this work entails.

Recommended: Labels

Create tags for priority that you can use to label and sort issues.

`Manage > Labels`



Prioritized labels

Drag to reorder prioritized labels and change their relative priority.

high
Jeff Avery / notae

medium
Jeff Avery / notae

low
Jeff Avery / notae

Toolchain setup

Check that everyone has the toolchain installed.

- Make sure that everyone in your team is using the same IDE!

Create your project

- Use the project wizard in your IDE.
  - Android Studio for Android
  - IntelliJ IDEA for desktop
- Commit and push the project to Git.
- Check that everyone can open/build/run it!

Planning

How to get started!

Planning instructions

Make sure everyone agrees on the project direction.

- Review the feedback from the design proposal and address it.
- You can make changes to the plan as the project progresses, but they need to be made as a team (in meetings like this).

Make sure that you have issues logged for each your features (1:n).

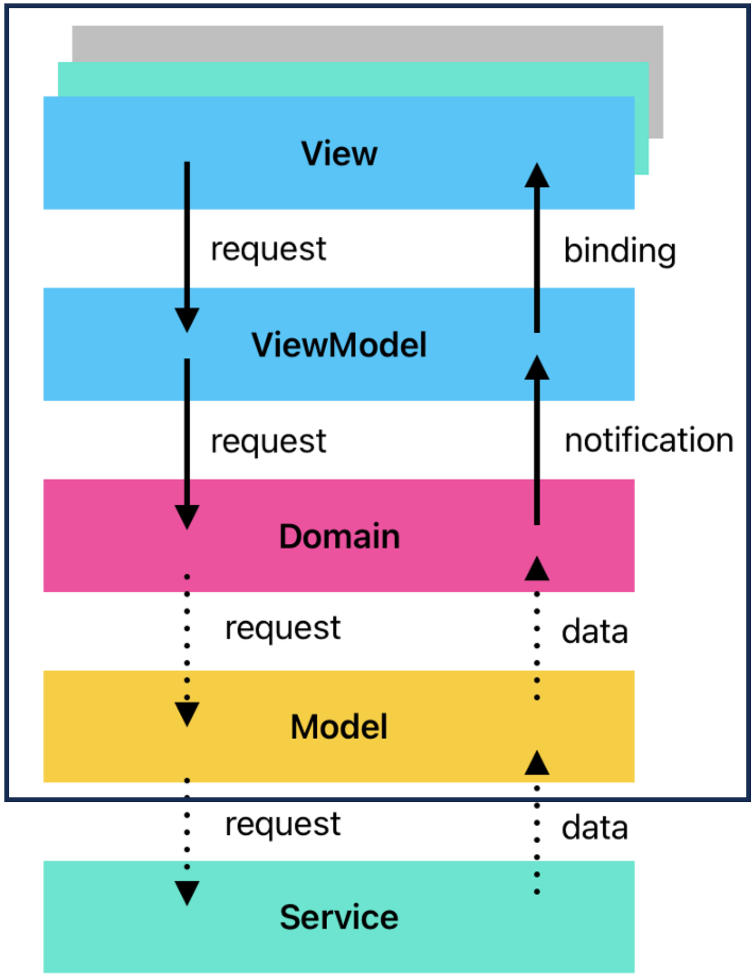
- An issue should describe the problem adequately for someone to investigate/design/implement the solution.

In this meeting, choose which features to implement for this iteration.

- Assign issues to your team.

What to build first?

- Build what you know how to build. You can:
  - Structure your code into layers.
  - Build user interface screens and navigate between them.
  - Build Domain objects e.g., recipe classes, note classes to model your data.
  - Write unit tests for domain and lower.
- As much as possible, build working features. If necessary, "fake" the data.
  - e.g., a Recipe App might launch with a list of recipes and images that are hard-coded (ideally in a mock Model class). You can replace the fake data later when the DB works.



What is the outcome of a kickoff?

At the end of the kickoff:

- All the features (issues) that you intend to complete for the first milestone are assigned to the Demo 1 milestone.
  - Other issues should be logged but should be unassigned.
- Each person should have some of these issues assigned to them.
- You should know when you will next meet.

https://student.cs.uwaterloo.ca/~cs346/1251/course-project/deliverables/project-iterations.html

During the Sprint

Work is assigned. Now what?

Coordinate your work

Work together as much as you can

- You are required to meet at least twice each sprint in-person.
- Meetings can be in-class, but everyone should attend.
- Take meeting minutes when you meet – record who attended, date/time, what was discussed/any decisions that were made.

Keep issues up-to-date and write detailed commit messages.

- We will review Git commit logs and GitLab issues to see what each person did.

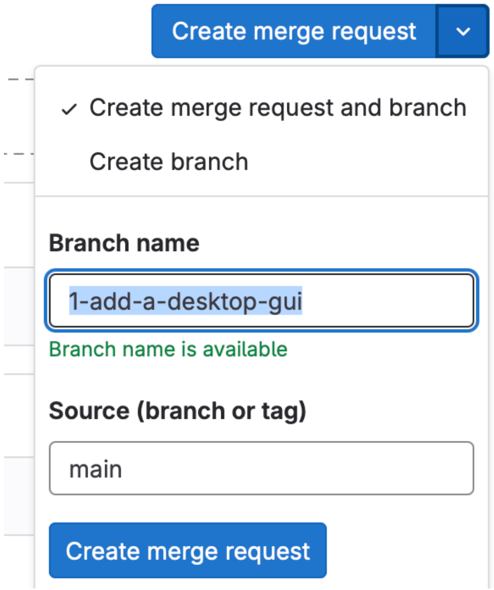
Use feature branches

Create a working branch for each feature (i.e. issue).

- These type of branches are called **feature branches**.
- Once we have a feature implemented and tested, merge our changes back into the trunk (“integration”).
- GitLab: `Create merge request` in the issue screen.

A typical workflow would be:

1. Create a feature branch for that feature.
2. Make changes on your branch only. Test everything.
3. (Optional) Have changes code reviewed by a teammate.
4. Switch to `main`; merge your feature branch into `main`.



Branch name: 1-add-a-desktop-gui

Branch name is available

Source (branch or tag): main

“Create merge request” will create a feature branch for you.

Update your issues!

- Keep your issues up-to-date.
- Add comments (to yourself, your team) to issues
  - Pretend that you are writing a note to your “future self”. What would you like to know about this issue in 6 months? A year?
- When work is complete, close the issues!
  - At the demo, all issues need to be updated (and closed if the work is complete).