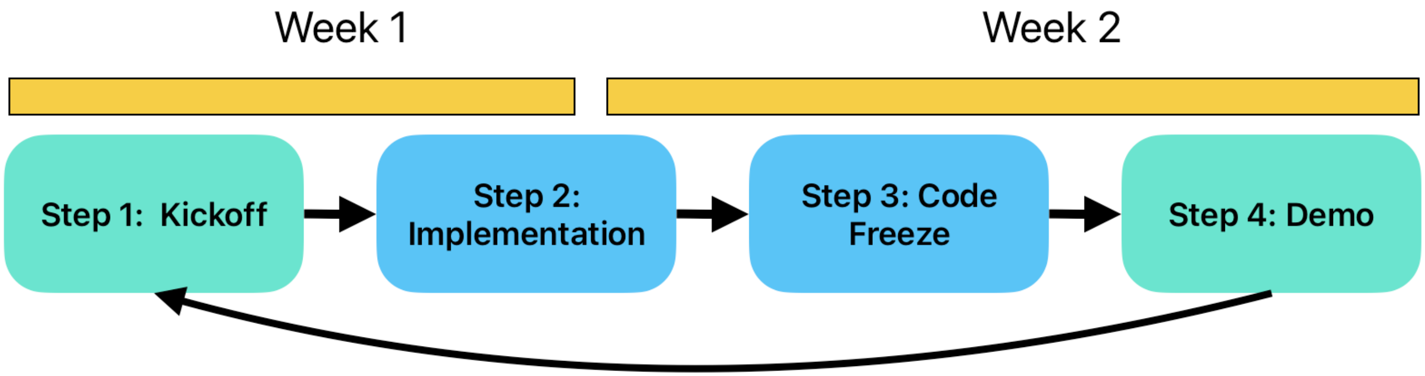
Sprint 1 Kickoff

CS 346 Application
Development

Sprint Structure



Today!

Goal: Decide What To Implement

You decide as a team what to include:

1. In GitLab, identify the user stories you wish to work on and create sub-tasks for each user story that represent the **features** you require.
   - Mix of mandated features + custom features from user stories.
2. Assign sub-tasks to Sprint 1 milestone. This is your **sprint backlog**.
3. Assign sub-tasks to team members. You can help each other & make adjustments.
4. You do not need to fully complete each user story; spread features across sprints.

https://student.cs.uwaterloo.ca/~cs346/1259/course/project/stages/sprints/

Mandated Sprint 1 Features

In GitLab, include these as issues, or sub-tasks of something else.

1. “You should have a Gradle project, with the correct structure for the type of project that you are building e.g., multi-project, or KMP.”
2. “You should have entity (business objects) classes relevant to your project e.g., recipe class for a Recipes application.”
3. “You should have an obvious layered architecture, with Domain and Model classes.”

Covered In-class.

4. “You should have a ‘reasonable number’ of unit tests in-place.”

Today

https://student.cs.uwaterloo.ca/~cs346/1259/course/project/stages/sprints/sprint1/

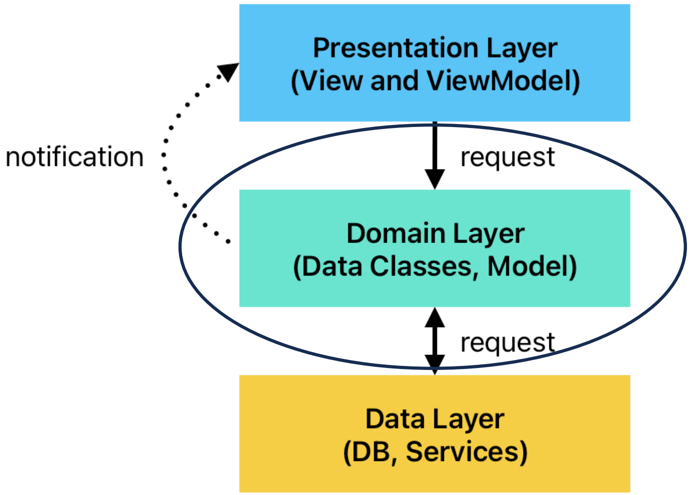
Custom Features?

For a starting application:

- The Presentation Layer will be pretty “thin” since we don’t have a GUI (yet).
- Service layer is probably undefined.
- Domain, Model can be built from user stories.

Suggestions:

- Get the build working (Gradle).
- Build out your Domain classes.
- Add unit tests that work with test data.
- Expand into other areas if you wish e.g., UI.



Each of these is a layer which may consist of multiple classes and/or functions.

Sprint Demo Format

From a single computer:

- Bring up the list of issues completed (Plan > Issue Boards) and describe what your main goals for the sprint.
- Each person on the team should demonstrate what they personally did. We want to see working functionality and passing units tests (no source code!)
- You are allowed to show GitLab documents, diagrams, unit tests results, or other work that helps to demonstrate what you accomplished.

https://student.cs.uwaterloo.ca/~cs346/1259/course/project/stages/sprints/