**Epic Progress Report assignment EPR1 – Writing User Stories**

Assignment authored by Prof. Grant Braught,  
with edits by John MacCormick

**Introduction:**

In this activity you will be writing user stories, breaking an epic into smaller stories, breaking those stories into tasks and writing acceptance criteria for your tasks. This activity will also be the Writing in the Discipline exercise for this course.

**Getting Started:**

Each team will use one FarmData2 fork for their development work. This will allow you to collaborate more closely on the parts of FarmData that your team is developing.

1. Within your team, decide on whose FarmData2 fork will be used. It doesn’t really matter whose, since you will be building off the main branch, not any of the previous activities.

2. The owner of that fork should invite all other teammates to be collaborators on the repository with Admin privileges. See “Manage access” in the repository’s “Settings.”

3. All other team members should accept the invitation to the repository.

4. If you have accepted an invitation:

a. Rename the clone of your own FarmData2 origin (i.e. just change the directory name).

b. Clone the team FarmData2 fork to your local development machine (e.g. into VirtualBox).

c. Unzip the database file as described in the INSTALL.md file.

d. Set the upstream of your clone to be the DickinsonCollege FarmData2 repository:

* https://github.com/DickinsonCollege/FarmData2.git

**Creating a Project Kanban Board:**

5. Add a blank project to the team’s FarmData2 fork. This can be done from the repo Projects tab, then New Project.

[Note: at the time of writing, January 2022, GitHub is providing a new Project Beta as well as their traditional Projects. It’s better to stick with traditional style if possible. Make sure to select the **Kanban-style** project, not the spreadsheet-style Beta. Make sure to create your project inside the repository. There is a new feature to create projects that span the entire organization including multiple repositories, but we don’t want that.]

6. Create columns on your Kanban board for: Epic, Stories, Tasks, Scheduled, In Progress, Done.

[Note: feel free to improvise or ignore the options for each column. These are features that you will learn to use later and are not too important right away.]

**Writing an Epic:**

Read the following resource on writing Epics:

* <https://www.linkedin.com/pulse/4-steps-writing-good-epics-stories-randy-chevalier-csm>

Beyond that resource you should feel free to search for other resources on writing Epics and examples of Epics as you feel is necessary to develop enough background to complete the following activity. You might start writing your Epic and find you are a little stuck. That should not be unexpected as you are likely new to this. So, search for some more information or examples to help move you forward.

7. **As a team,** write an Epic for your assigned FarmData2 feature as a note in the Epic column of your Kanban board. Use the text of the issue(s) and an exploration of the original FarmData application in order to write your Epic. Your Epic should capture the entirety of the desired functionality.

Note: the GitHub interface sometimes refers to a *note* and sometimes to a *card*. The words *note* and *card* mean the same thing in this context.

Your Epic should include a Label, a Narrative, and Acceptance Criteria, as outlined in class and in the above resource. The Label should be concise and useful in referring to the Epic. Your Narrative should use the standard template for user stories (from the article and from class). Your Acceptance Criteria should be written as a “Task List” in markdown (<https://docs.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax>).

**Writing User Stories:**

Read the following resources on writing User Stories:

* <https://ardalis.com/invest-in-good-user-stories/>
* <https://stormotion.io/blog/how-to-write-a-good-user-story-with-examples-templates/>

Then read the following resource on Splitting an Epic into User Stories:

* <https://www.mountaingoatsoftware.com/blog/two-examples-of-splitting-epics>

Then read the following resources on writing Acceptance Criteria:

* <https://www.seguetech.com/what-characteristics-make-good-agile-acceptance-criteria/>
* <https://agileforgrowth.com/blog/acceptance-criteria-checklist/>

As with writing the Epic, you should feel free to seek out additional resources at any time. If you feel stuck, search for and skim a few pages with more information or different examples that will help you move forward.

8. **As a team,** discuss your Epic and *identify* at least 3 user stories per teammate. (Note that in this step you identify each story with a Label but you do not write the Narrative or Acceptance Criteria.) These stories do not have to be sufficient to complete the Epic but they should be clearly connected to the Acceptance Criteria for the Epic. They should also be stories that you think you can begin implementing and could complete within a 2-3 week window. As much as possible, they should be independent of the other stories. Or if they do depend on others, then the stories on which they depend have also been identified. For each user story, create a card in the Stories column and give it a draft Label. Each card should have only a Label at this point.

9. **Individually** each teammate should pick one user story and complete it on the card in the Stories column. If possible, you should pick a story that is independent: you can implement it without first implementing other stories. You should revise the Label as necessary, provide a Narrative (in the standard form) and give Acceptance Criteria as a Task List. You should make sure that your User Story follows the INVEST acronym.

10. **Individually**, in your WiD repository in the Dickinson-COMP-WiD organization on GitHub:

a. Create a COMP290 folder.

b. Place a text file in markdown format, named UserStoryDraft.md, into the COMP290 folder. This file should contain a copy of the contents of the cards for the Epic and for your completed User Story.

11. **As a team,** meet, discuss and collaboratively revise each of the individually completed User Stories from question #9. You should evaluate and revise each User Story using the INVEST criteria and the 7 Tips for Writing Acceptance Criteria.

12. **Individually** each teammate should repeat question #9 for a second User Story.

13. **Individually**, in your WiD repository in the [Dickinson-COMP-WiD](https://github.com/Dickinson-COMP-WiD) organization on GitHub:

a. Place into your COMP290 folder a text file in markdown format named UserStoriesFinal.md that contains your revised User Story from #11 and your new user story from #12.

b. Place into your COMP290 folder a text file in markdown format named Reflection.md that responds in a few sentences to the following prompts:

* How did you revise your first User Story?
* How is what you learned from that revision reflected in what you have written for the second User Story?

**Prepare for a Client Meeting:**

Matt Steiman, our client at the farm, will likely be joining us at the start of the class when this assignment is due to answer questions that you have about FarmData2 or the particular features that your team will be working on. Each team will get about 10 minutes.

14. **As a team**, be prepared to:

* Share your Kanban board.
* Talk about the Epic you have written.
* Talk about the User Stories you have completed. As you talk about each story, solicit input from Matt and ask clarifying questions to ensure that your shared understanding of the feature and stories is consistent.

Note: there is no Moodle submission for this assignment. Your submission to the WiD repository is the component of this assignment that will be used to determine satisfactory completion.