

Scrum Report

Sprint Duration: 3/31 - 4/11

Sprint Goal: Write a working program that features general UI framework

Meeting Logs:

3/31: We started most of the day off by learning about the project and its requirements. After understanding the assignment, we discussed what kind of add-on we would like to work on. Some of the key things we wanted was an add-on that would be challenging enough for us to code and be something we could be proud of. After coming to a conclusion on that, we discussed some possible ideas and decided on everyone coming back to the next meeting with a couple ideas.

4/2: This day we focused on leaving class with a solid idea that we all agreed on. To begin this meeting, we learned a little more about how we could apply different libraries to our add-on. Then, we started discussing our ideas. After diligent discussion, we came to the decision of a google add-on that would help understand data structures together. We thought the possibilities of animating different data structures along with their changes would be challenging but doable. For the next meeting, we decided on splitting our tasks.

4/4: On this day, we discussed what specifically everyone was going to do for this sprint. Adharsh took the lead with UI framework and Ethan took the lead with the dialog box. Charles, Yosh, and Sean decided they would work on the scrum report, the user studies, and help Adharsh and Ethan when needed. We discussed the time everyone needed and when we might need to help someone with a specific task. We also decided that by the next meeting we should have a bulk of our code done.

4/7: We came together and worked on the UI framework with Adharsh taking the lead. He communicated with us his general idea of the sidebar and created a working framework. Ethan also did a bulk of his work on the dialog box and confirmed that it would be working by the deadline. Charles, Yosh, and Sean discussed with each other what other work was needed and what parts we were going to work on. At the end, we agreed on checking on everyone's progress next meeting and finalizing what needs to be done.

4/9: This meeting, we came together and completed the majority of the UI code. Ethan finished the dialog box and Adharsh completed the sidebar as well. Charles, Yash, and Sean helped when needed and they also worked on the different documents we needed to submit. The

Scrum Report and User Stories were still incomplete but in progress. We decided to finish that as soon as possible so we would not have to worry about the deadline.

4/11: We came together briefly to discuss that everything was working and we would not need to change anything. Everyone looked over everyone else's work and was satisfied with what was done. We tested the code at the end and made sure it was ready before submission. We took the majority of this time to discuss what we were going to do for Sprint 3, which is the main algorithm. We discussed that this would be the most challenging Sprint and that we would need to start early and fast.

Initial sprint backlog:

Purpose: to write the UI code for our google add-on, essentially a framework for our core algorithm.

Build UI framework for each Data Structure Holder:

Create stackSidebar.html with:

- Input for push
- Push button
- Pop button
- Delete button
- Preview button

Apply consistent styles (fonts, padding, button states)

Add placeholder functions: pushValueStack(), popValueStack(), etc.

Create queueSidebar.html with:

- Input for enqueue
- Push and Pop buttons
- Delete button

- Preview button

Match style with Stack sidebar

Add placeholder functions: `pushValueQueue()`, `popValueQueue()`

Create `linkedListSidebar.html` with:

- Two inputs (value and "insert before")
- Add and Remove buttons
- Delete button
- Preview button

Apply same styling scheme

Stub functions: `addValueLinkedList()`, `removeValueLinkedList()`

Create `bstSidebar.html` with:

- Input fields for Add and Remove
- Sort By Max / Min buttons
- Delete and Preview buttons

Apply same visual style

Add JS hooks: `addValueBST()`, `removeValueBST()`, etc.

Create dialog.html as an empty template:

- Add comment for future structure preview
- Ensure showDialog() is wired to google.script.run.showDialog() in all sidebars

Final Sprint backlog:

Purpose: to write the UI code for our google add-on, essentially a framework for our core algorithm.

Build UI framework for each Data Structure Holder:

Create stackSidebar.html with:

- Input for push
- Push button
- Pop button
- Delete button
- Preview button

Apply consistent styles (fonts, padding, button states)

Add placeholder functions: pushValueStack(), popValueStack(), etc.

Time: 2 days

Create queueSidebar.html with:

- Input for enqueue
- Push and Pop buttons

- Delete button

- Preview button

Match style with Stack sidebar

Add placeholder functions: `pushValueQueue()`, `popValueQueue()`

Time: 2 days

Create `linkedListSidebar.html` with:

- Two inputs (value and "insert before")
- Add and Remove buttons
- Delete button
- Preview button

Apply same styling scheme

Stub functions: `addValueLinkedList()`, `removeValueLinkedList()`

Time: 2 days

Create `bstSidebar.html` with:

- Input fields for Add and Remove
- Sort By Max / Min buttons

- Delete and Preview buttons

Apply same visual style

Add JS hooks: addValueBST(), removeValueBST(), etc.

Time: 2 days

Create dialog.html as an empty template:

- Add comment for future structure preview
- Ensure openDialog() is wired to google.script.run.showDialog() in all sidebars

Time: 1 day

Updated Product Backlog:

ID	User Story	Estimation (time)	Priority
1	As a student, I want to visualize data structures after writing them in the document	8	4
2	As a student, I want to test different data structures	2	5
3	As a student, I want to see the step-by-step process of different data structures changing	2	3
4	As a professor, I want to create visualizations for students to learn from my lecture notes	8	4

5	As a professor, I want it to detect various data structures, not just one	5	2
6	As a professor, I want a sidebar that appears with different options once the structure is created	4	1 (completed)
7	As a professor, I want to see a step by step breakdown of how data structures change	2	3

*Green means completed