

Meeting 3: Scrum #2

Time: December 11, 4:00 PM to 4:20 PM

Place: Discord call

Description:

Roles:

- Tanush Nadimpalli: Scrum Master **[present]**
- Rohan Shiva: Developer **[present]**
- Ramki Pitchala: Product Owner **[present]**

Ramki

What did you do?

- In the midst of finishing up implementation for `FillInTheBlankProblemGenerator`.

What's not working?

- Tight coupling between `ProblemSetGenerator` and the `ProblemGenerator` classes. Tight coupling between `PermutationGenerator` and `ReorderProblemGenerator`. I need to spend some time decoupling these classes in order to ensure adding generation for future problem types isn't strenuous.

What are you planning on working on?

- Planning to finish **[Task 1]**, **[Task 2]** and **[Task 3]** by the next upcoming sprint.

Task Updates

- No updates since last time

Tanush

What did you do?

- I worked on adding the new fill in the blank option to the generation form and updating the necessary interfaces. I also worked on modifying the generation form tests to include the new fill in the blank checkbox. I also had a new test to make sure that all the checkboxes were being triggered as expected when the user clicks on them.

What's not working?

- We realized that the `GenerationViewResult` should be more modular and the problems should be separated clearly from each other based on their types (related to **[Task 8]**).
- Also blocked on **[Task 8]** by **[Task 1 & 3]**

What are you planning on working on?

- I'm planning on modularizing the `GenerationViewResult` so that the problems are separated and are easier to distinguish.
- Planning on finishing up Task 8 and Working on Task 9.

Task Updates

- **[Task 4 & 5]** moved from IN PROGRESS to DONE
- Updated description and scope for **[Task 8]**
- **[Task 8]** moved from TODO to IN PROGRESS

Rohan

What did you do?

- Modularized the `GenerationForm` state to include problem type config such as number of problems and selected status for each problem type
- Started working on tests for `GenerationForm`'s state updates

What's not working?

- One of the previous tests failed after the updates, due to an issue with Jest's clean up function after running each test.

What are you planning on working on?

- Planning on looking into why some tests are failing.
- Add functionality to export a generated problem set to a JSON file and modify `GenerationResult` tests to ensure proper functionality.

Task Updates

- **[Task 6]** moved to DONE
- **[Task 9]** created and assigned status IN PROGRESS
- **[Task 10]** created and assigned status IN PROGRESS

Overall

Some things to consider:

- [Ramki]: `ProblemSetGeneration` is tightly coupled with `ProblemGeneration` which makes onboarding the generation of new problems more difficult. I will need to spend time decoupling this region in addition to adding the `FillInTheBlankProblemGenerator` class.
 - [Rohan]: +1
 - [Tanush]: +1
- [Rohan]: Although the state in `GenerationForm` is modular with regards to different problem types, adding a new problem type still requires changes in a lot of places.
 - It might be better to look into this to make things less coupled, as it will allow us to easily support more problem types in the future.
 - [Tanush]: +1

Updated Sprint Backlog

[Task 1] Create a FillInTheBlankProblemGenerator class

- Assignee: **Ramki**
- Status: IN PROGRESS
- User Story: **[Epic 1]**

[Task 2] Write unit tests for FillInTheBlankProblemGenerator

- Assignee: **Ramki**
- Status: TODO
- User Story: **[Epic 1]**

[Task 3] Modify ProblemSetGenerator to use FillInTheBlankProblemGenerator to generate “fill in the blank” problems

- Assignee: **Ramki**
- Status: TODO
- User Story: **[Epic 1]**

[Task 4] Modify GenerationForm to allow instructors to select “fill in the blank” problem type

- Assignee: **Tanush**
- Status: DONE
- User Story: **[Epic 1]**

[Task 5] Modify GenerationForm tests to ensure “fill in the blank” problem type can be selected in GenerationForm

- Assignee: **Tanush**
- Status: DONE
- User Story: **[Epic 1]**

[Task 6] : Modify GenerationForm to allow the # of problems per problem type to be set by the user

- Assignee: **Rohan**
- Status: DONE
- User Story: **[Epic 3]**

[Task 7] Modify GenerationForm tests to ensure # of problems per problem type implementation is respected

- Assignee: **Rohan**
- Status: IN PROGRESS
- User Story: **[Epic 3]**

[Task 8] Modularize GenerationResult component and display the “fill in the blank” problem type. Update tests to make sure they reflect the new changes.

- Assignee: **Tanush**
- Status: IN PROGRESS
- User Story: **[Epic 3]**

[Task 9] Add functionality in GenerationResult to export a generated problem set as a JSON file.

- Assignee: **Rohan**
- Status: IN PROGRESS

- User Story: **Epic 4**

Task 10 Modify `GenerationResult` tests to ensure export button to export generated problem set as a JSON file functions properly.

- Assignee: Rohan
- Status: IN PROGRESS
- User Story: **Epic 4**