


T7 Sample Answer (for discussion)

Question: for reference

 Question adapted from a past exam paper.

Pull Request Tracker (PRT) is a desktop application meant to help tutors of a Software Engineering module deal with GitHub PRs more efficiently (compared to the GitHub Web interface). For example, it will help tutors find and review PRs from their *mentees* easily. It will help the *managers* of the module (e.g., professor, head TA) to easily keep track of how tutors are dealing with mentee PRs. PRT will communicate with GitHub using the GitHub API.

1. Write 1 must-have and 1 nice-to-have user stories, covering user types *tutor* and *manager*.
2. Write at least 1 NFRs, related to performance/scalability and/or usability.
3. Give at least 1 terms worth recording in the glossary.
4. Complete the following use case. Give at least one extension. Note that the tutor should give comments in the order of PR size (i.e., give comments to smaller PRs first). Assume the following use cases exists already: *U1. Sort PRs by a criterion*, *U2. Add comments to a PR*

System: PRT

Use Case: U3. Add comments to mentee PRs

Actor: ...

Precondition: ...

...

User stories

1. As a tutor, I can see a list of PRs from my mentees, so that the tutor is not distracted by PRs from other students.
2. As a tutor, I can comment on PRs, so that I can give feedback to my mentees.
3. As a manager, I can specify which students are allocated to which tutors, so that I can direct student PRs to the matching tutor.
4. As a manager, I can close PRs, so that I can remove stray PRs.
5. As a tutor, I can see which PRs I haven't reviewed, so that I can sort PRs based on review status.

Non-Functional Requirements

1. Should not take a lot of memory while in operation.
2. Should be able to view a PR diff of up to 1500 lines without a noticeable lag.
3. Should lose no more than 1-minute worth of work in case of connectivity loss.
4. All user operations should complete within 100 milliseconds.
5. Should support navigating to the corresponding PR on GitHub Web interface, if needed.
6. Will not perform any automated tasks e.g., send periodic reminders. And hence, will not have a server component.
7. Should allow use only by authorized users.

Glossary

- **Pull Request (PR)**: a GitHub feature for proposing code changes
- **Mentee**: a student assigned to a tutor.
- **Manager**: a person with admin privileges to the corresponding GitHub org/repo.

U3: add comments to mentee PRs | **Actor:** tutor | **Precondition:** logged in

MSS:

1. Tutor enters the 'list' command
2. System shows PRs
3. Tutor performs U1. Sort PRs by criterion to sort by comments count
4. Tutor performs U2. Add comments to a PR

Repeat step 4 until satisfied

Use case ends.

Variations:

2a. No PRs allocated to the tutor

Use case ends

*a. Connectivity lost

*a1. System informs the user

Use case ends