**CS3203 Iteration 2 Evaluation (Report)**

**Team: 05 Evaluator: Zaw Lin**

|  |  |
| --- | --- |
| **Item** | **Remarks** |
| **General Quality**   * Readability / Organization | * Good that you have a report documentation standard. * The report suffers from the same issue as It. 1. Try to avoid having a lot of text. * Try to use tables and figures appropriately. |
| **Scope** | * Should have a summary of It. 1 requirements. |
| **Development Plan and  Work Division** | * Should organize into mini iterations instead of having a task list. A mini iteration is full dev cycle including its own test suite and specific deliverable. Right now, it’s just a list of tasks. |
| **Parsing**   * Parsing and Validation | * Parsing section should be condensed. It is too verbose now. * The explanation with the example is not quite clear. For example, what are the tokens? And what happens after that? Try to condense key points and expand on that with an example instead of giving a lot of info in one go. * Should have a small section to describe how syntax and semantic validation is done referring the relevant sections. |
| **Frontend 🡪 PKB**   * How and when is PKB populated? (Abstract APIs can be used for explanation.) | * Information is presented but it is unclear. * Should describe more details with an example on how Uses/Modifies are populated instead of giving the results. Use a smaller example enough to cover all cases. * Should use more compact tables, e.g., Table 4.2.2e. |
| **PKB**   * Data Structures | * Data structures are described with contents, but the presentation is disorganized. You should describe first what is the general structure of each table before giving examples. |
| **Query Processor**   * Parsing and Validation * Query Object * Basic Query Evaluation * Representation for Intermediate / Final Results | * Description is too verbose. Should use diagrams and condense the text descriptions. * Should describe clearly how query parsing results in query object. * Should describe more details on how validation works. * BQE is quite detailed but try to condense the descriptions and use tables/figures to reduce the amount of description. * Should give a general description of results before giving the example. |
| **PKB 🡪 Query Processor**   * How does the Query Processor call / use PKB? (Abstract APIs can be used for explanation.) | * Described. |
| **Design Decisions**   * Well-spreaded and Well-documented? | * Should have clear criteria (design principles, performance, ease of implementations etc.) and pros and cons of each approach. * Should clearly explain what approach is chosen. * Should clearly explain the rationale. * Should have consistent formatting for all DD. |
| **Coding Standards and API Development Experiences** | * Should describe more on how concrete apis correspond to abstract apis with examples and what steps are taken to enhance correspondence. |
| **Testing**   * Test Plan * Unit / Integration Testing * System Testing (are the test cases well-designed?) | * Test plan should describe activities in more details (e.g. design test for Next\*) * Should give examples for source code design. * Should give examples for query design. |
| **Abstract APIs for PKB** | * If API always returns LIST\_OF\_ENTITIES, how can you evaluate queries that are Boolean in nature such as Follows (1,2)? Provide a description on how this is done. |
| **Others** |  |