COMP 3350 Project Iteration 2

Group 2 (Handy Application)

Planning and process (4/4)

- GIT (1/1)
    - Version control is being used properly for example, has more than one committer and commits are reasonable size and frequency. They are not only big commits at the end. (1/1)

Comments: All members committed to GitHub, commits are mostly done around the deadline and they are within reasonable size and frequency.

- Updated plan (2/2)
    - Plan should be up-to-date (if there is any change to the previous plan for Iteration 2 it should be explicit and justified) (0.5/0.5)
    - Big user stories for iteration 3, if it was not already in plan (0.5/0.5)
    - Development tasks assigned in iteration 2 (what exactly has been done by developers) (0.5/0.5)
    - The time planned for the development tasks and detailed user stories and the actual time it took, in iteration 1 (0.5/0.5)

Comments: In general, your documents are poorly structured. You need to specify the actual time/expected time for each development tasks which is assigned to particular person.

- Wiki (1/1)
    - Should include description of the content of the submission. Can include other things as well. (1/1)

Comments: Your wiki should provide a link to Documentation folder. This comment was part of your previous iteration marking sheet as well.

Functionality (8/8)

- DB support including the actual DB and stubbed version (2/2)
- System performs end-to-end (including GUI) processing for all stories (2/2)
- Works on both emulator and tablet device. (2/2)
- The developed program conforms the updated plan (the stories that are claimed to be implemented, are indeed there) (0.5/1)
    - Modify PDFs, it is mentioned that it will be implemented in the iteration 3.

- No easy bug (No crashes or unexpected behavior while trying normal scenarios) (1/1)

Comments: Overall, the application is built properly but some features are not working as per plan.

Implementation (6/6)

- No obvious code/design smells (2/2)
    - Classes are in the wrong package (e.g., logic is developed in the UI layer)
    - Big classes: Classes are taking too much responsibility (SRP)
    - Very long methods (over 20 lines)
    - Wrong usage of inheritance
- Proper dependency injection for DB (2/2)
- Good standard coding style (2/2)
    - Informative naming
    - Comments explain "why" and not "What"
    - No to-do
    - Too much code duplication (copy-paste)

Comments: There is proper packaging for each layer (business, persistence …), your code could have been written more neat, plus you have limited number of comments.

Unit tests (5.5/7)

- Automated JUnit test cases for both new features and old ones (1/2)

You have provided limited number of unit tests. For instance there is no tests for methods in TextNoteWrapper.java (-1)

- Passes all unit tests for domain objects and business logic (2/2)
- Reasonable test coverage of normal and corner cases (0.5/1)

Coverage is low and there is no unit test for testing special cases. (-0.5)

- Integration tests (actual DB in the loop) (2/2)

Comments: Two simple test methods are provided for integration test, in your next iteration, you need to provide more unit tests.

Penalties (-0.5)

- Log file (up to -2 if missing or incomplete)
- Missing libraries. Unspecified dependencies. (up to -2)
- Previous unresolved issues (up to -5)

Comments: You still have very limited number of test methods under TestList.java and TestIT.java (-0.5).

Total (23/25)