COMP 3350 Project Iteration 1

Group 2 (Handy Application)

Planning and process (5.75/6)

- GIT (1.5/1.5)
  - is accessible (0.5/0.5)
  - 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: GIT is accessible, your team should utilize more of GIT features like issues and so on. One member alone contributed more than twice of entire group.

- Architecture sketch (1.5/1.5)
  - Should be 3-layer (0.5/0.5)
  - Should have some high-level classes in each layer (not include very low level details) (0.5/0.5)
  - Should show the relationships between classes (0.5/0.5)

Comments: Architectural sketch with basic 3-layers is provided.

- Updated plan (2/2)
  - Plan should be up-to-date (if there is any change to the previous plan for Iteration 1 it should be explicit and justified) (0.5/0.5)
  - Big user stories for iteration 2, if it was not already in plan (0.5/0.5)
  - Development tasks assigned in iteration 1 (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: The log.txt is provided with necessary contents.

- Wiki (0.75/1)
  - Should include description of the content of the submission. Can include other things as well. (0.75/1)

Comments: Your wiki should provide a link to your architecture documents and any other document you have in the codebase. (-0.25)

Functionality (6/6)

- 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) (1/1)
- Database stub and its interface (1/1)
- At least one completely functional GUI, which performs end-to-end processing for at least one big story (1/1)
- No easy bug (No crashes or unexpected behavior while trying normal scenarios) (1/1)

Comments: Your application works as expected on both emulator and device.

Implementation (4/4)

- Appropriate package structure for code and the test base (1/1)
- Good standard coding style (2/2)
    - Informative naming
    - Comments explain "why" and not "What"
    - No to-do
    - Too much code duplication (copy-paste)
- No obvious design smells (1/1)
    - 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

Comments: No major issue.

Unit tests (3/4)

Automated JUnit test cases and test suites are available (1/1)

Passes all unit tests for domain objects and business logic (1/1)

Reasonable test coverage of normal and corner cases (1/2)

Comments: Limited number of unit tests methods are provided in TestList.java. Unit tests are not running and throwing Exception in thread "main" java.lang.NoClassDefFoundError: junit/textui/ResultPrinter plus you have limited test cases which doesn't cover all of your methods. (-1)

Penalties ()

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

Comments: No issue.

Total (18.75/20)