COMP 3350 Project Iteration 1

Group 8 (Course Manager)

Planning and process (2.75/6)

- 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: {Seems like only one of the user has most commits (smithi35), all the other users have 2 commits each. }

- Architecture sketch (0/1.5)
  - Should be 3-layer (0/0.5)
  - Should have some high-level classes in each layer (not include very low level details) (0/0.5)
  - Should show the relationships between classes (0/0.5)

Comments: {Architecture is not layered as per required (Presentation, Business Logic & Persistent) -1.5}

- Updated plan (0.25/2)
  - 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) (0.25/0.5)
  - The time planned for the development tasks and detailed user stories and the actual time it took, in iteration 1 (0/0.5)

  Comments:

  1. The files submitted for the log should be renamed to log.txt.
  2. {Planned development time cannot be found in the log. -0.5}
  3. {Detailed user stories were not provided for big stories. -0.5}
  4. {The detailed user stories were either not divided among the group members or were not specified. -0.25} There was more than one person working on each user story of this iteration, as mentioned in the log.

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

  Comments: {Description wasn't found for plan and log. -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: No issue.

Implementation (2/4)

- Appropriate package structure for code and the test base (0/1)
- 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 (0.5/1)
  - 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:

1. {Currently there is no segregation maintained in the code with respect to layers. -1}
2. {Presentation, logic and db code all seem to be contained in the same class -0.5}

Unit tests (2/4)

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 (0/2)

Comments: {Test cases don't cover all the cases mentioned in the plan. Only 1 test case testing if DB is working fine. -2}

Penalties ()

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

Comments:

Total (12.75/20)