

COMP 3350 Project Iteration 2

Group 8 (Course Manager)

Planning and process (3.25/4)

- 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: No comments

- Updated plan (1.25/2)
 - 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 (0/0.5)
 - 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 (0.25/0.5)

Comments:

1. Iteration 3 user stories were not found. -0.5
2. Actual time spent was not found in the logs. -0.25

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

Comments: No comments

Functionality (6.5/8)

- 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)
- No easy bug (No crashes or unexpected behavior while trying normal scenarios) (0/1)

Comments:

1. Seems like user stories are not completed, or are not functioning correctly. (Create Profile, Login etc). -0.5

2. BUGS: a) Dates and times not validated. b) Adding new course: Not showing new courses after pressing Save button. c) Create Profile, Log in works without any input -1.

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: No comments

Unit tests (4.5/7)

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

Passes all unit tests for domain objects and business logic (1.5/2)

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

Integration tests (actual DB in the loop) (0/2)

Comments:

1. Test “TestSQL” failed. -0.5
2. Application specific tests are not run using DB. -2

Penalties ()

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

Comments:

Total (20.25/25)