

COMP 3350 Project Iteration 1

Group 1 (HealthTrack: The Applet for your Diet)

Planning and process (5.5/6)

- 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, there is more than one committer but some member's contribution in coding part is less. You should utilize GIT features like issues more often.**

- Architecture sketch (1.25/1.5)
 - 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 (0.25/0.5)

Comments: **The content of Project_Architecture.jpg and Project_Architecture2.jpg is poorly presented! And detailed classes are provided whereas should have stated high level architecture only and the relations are ambiguous! (-0.25) For the next iteration you are required to have proper architecture diagram.**

- Updated plan (1.75/2)
 - 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 (0.25/0.5)
 - 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: **You should clearly specify the iteration name, which the big stories are belong to! Plus detailed stories are breakdown of a given big story, so you should mention that your detailed stories are breakdown of which of your big stories. Since it is not clear which stories are belong to your 2nd iteration, therefore you will lose mark for that. (-0.25). You are required to group your stories as per iteration. Iteration 1 (UC1, UC2 ...), it is really important to do so!! Also your detailed stories must be breakdown of your big story and it must be clear which big story given detailed story is belong to!**

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

Comments: Wiki is provided with necessary links to files plus some description of architectural layers is stated in the wiki as well.

Functionality (5/6)

- 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 (0/1)
- 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: You need to have stub implementation of your database interface (-1). Referring to Project.Submission_updated.pdf you have limited stories therefore you have to make sure all of them get implemented properly.

Bugs: Under goal submission, your form placeholders should be clear once user click on the field. Goals submission doesn't save in the database, the only working feature you have is display of stats. Food eaten menu is empty, thought you were supposed to have menu to enter eaten meals.

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: You should have proper naming convention, some of your class names have "_" and some don't.

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: **There are limited number of test methods under StatsTest.java, you need to 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 (17.5/20)