EECS 2311: Software Development Project: Iteration 1, Starting the Implementation

Deadline: Friday Feb 14, 2025 by 11.59 pm

In iteration 1, you will begin implementing the project you planned in the previous iteration. This will include the business objects, a stub database (that is no real database but fake data), unit tests, and the GUI for at least one big story.

You need to start using GIT for version control and set up a public GIT repository for your team (all the team members may have their own account and fork this official repo). If for any reason you want to keep the repo private, you need to invite all TAs and the instructor to the project. By the ITR1 deadline, your project on GitHub should include all the recent files for the iteration 1.

Each group will also keep a log, which describes their progress and meeting minutes and summaries. The log may be updated by the individual team members. You should keep the log in a text file log.txt alongside your code (the log file also goes through a proper versioning).

Process requirements:

Allocate each user story of this iteration to one student. If there are extra user stories that need to be done before you can start working on your stories, do them as a team, up front. Do not (gratuitously) exceed the time available for this iteration. Task assignments, estimates, and the actual time spent should be detailed in your log. Also in the log, include information about team meetings and the rationale behind any major design decisions. Use the log to detail who worked on each task, and approximately how long was spent on each. Make sure you include all user stories and development tasks planned for this iteration, regardless of whether you completed them all.

Revise your plan from iteration 0, accordingly. Include both old and the new version of your plan. You will be discussing these changes during the project presentations.

You need to commit frequently to the repository. There may be penalties for big commits and only commits close to the deadline. Each team member is supposed to contribute directly into the team's GIT repository. The contributions will be visible by commits and also in the issue tracking feature of GitHub.

Implementation requirements:

For this iteration, you must implement all classes that represent the domain specific components of your system; for example, the student records system requires objects for students, courses, and registrations. It must specify an interface to the database, plus a stub implementation that uses non-persistent storage (say, an ArrayList) with a set of

initial contents. It will also need additional classes to connect components in the system and for calculations or logic.

You must implement automated unit tests for your domain-specific and business logic classes using a unit testing framework (e.g., junit). Construct a very thorough set of unit test cases as described in class (introduction to testing week). Ensure that your code passes all tests.

You must also implement a basic GUI for all user stories in ITR1. You can (and should) improve the GUI in the next iterations.

Use appropriate coding standards and simple design style (e.g., no unnecessary generalization). Ensure everyone on the project uses the same coding style. There should be no dead code or TODOs. Divide the project into packages that match your architecture.

Documentation requirements:

Create a Wiki for your Github repository to describe the content of your submission. Identify the packages and major source code files. Sketch out the overall architecture of your system (internal and external components with their dependencies). Include a copy of the sketch with your submission.

Evaluation:

Your work for this iteration will be marked together with deliverable 1. The rubric will be posted together with the deliverable 1's rubric.
In general, marks will be allocated roughly equally to four areas: a) functionality, b) implementation quality, c) the quality and thoroughness of your unit tests, and d) process/release/documentation details.

There will also be additional substantial penalties for missing elements (log files, developer task lists, etc.), broken releases (that do not compile), or compile/run-time errors or meaningful compiler warnings.

Hand-in:

For now, you only need to keep your GitHub project up-to-date and your project history accessible (publicly or privately). **That is no eClass submission, yet!**
You should also tag the version you consider as iteration 1 as "ITR1", so we can check it out easily. The tag should be on a version before the deadline!

Your repository should contain all the files necessary to run/test your program and the corresponding documentation by the deadline. These include:

- Source code
    - all source files in a folder called src (each layer's classes in separate package)
    - all test files in a folder called test with the same organization as source

files
    - a lib folder that contains any necessary jar files // or equivalent for extra libraries in other languages
- Documents
    - Wiki
    - Architecture sketch
    - Log file (including meetings minutes, rationale behind changes on plan and big design decisions, any concern with the project or group members, task assignments, the development tasks per user story and the times originally allocated per development task and the actual time spent on each task)
    - Old planning document (from ITR0) and the modified version of it (containing any change to the plan that happened during ITR1).

You may develop your code on any platform. The project must contain all the files it needs to execute and run on a "fresh" development computer (Windows and Mac), it should take nothing more than "Compile" and "Run As..." to execute it on the TA's machine.