

Phase 1 Progress Report

Changes from phase 0:

- We removed the “label” feature and the commands associated with it.
- We allowed user login/register, added the “team” feature, and allowed users to assign tasks to teammates (along with a bunch of new commands).
- We changed our program structure significantly, implemented design patterns, and added more comments and tests.

To-do list for phase 2:

- Improve program structure based on feedback from phase 1.
- Change outer layers (database, interface).
- Create more constants:
 - Use more Enums for use case results.
 - Create specific Exceptions, but also remove controller warnings and let them return more Strings instead of throwing Exceptions (maybe).

Group member tasks:

- Phase 1 tasks:
 - Jiayang: Wrote some test cases
 - Jingyang: Proposed future changes to database.
 - Krystal: Wrote some comments
 - Richard:
 - Brainstormed, decided on, and implemented new features
 - Tried to improve code structure and refactored code
 - Implemented design patterns
 - Fixed teammates’ errors
 - Wrote some test cases and comments
 - Wrote design document and progress report
 - Yixin:
 - Brainstormed, decided on, and implemented new features
 - Wrote majority of the test cases
 - Improved some part of code structure and all the code clean up
 - Wrote design document and progress report
 - Zixiu: Wrote some entity classes
- Future tasks:
 - Jiayang:
 - Testing
 - GUI
 - Jingyang:
 - Improve database and interface (using SQLite)
 - GUI
 - Improving code structure
 - Krystal:

- Testing
- GUI
- Richard:
 - GUI
 - Debugging
 - Improving code structure
- Yixin:
 - GUI
 - Debugging
 - Improving code structure
- Zixiu:
 - Testing and debugging
 - GUI

What has worked well so far:

- Functionality wise our program seems to be performing well. Tests are passing and no obvious bugs are present.
- Design patterns work well from our perspective.

Questions:

- When testing one class, other classes used by that class also get tested automatically. For example, testing a use case class would automatically increase code coverage for entities. Does this mean we do not need to test the entity classes separately?