

NG summer uni 2016 task comments

Ants Kristjan Rooma

GitHub link for the assignment: <https://github.com/akrooma/ng-summer-uni-assignment>

I decided to add Domain, DAL and BLL(didn't have this at first) subprojects for this assignment. At first it may seem like unnecessary complexity and layering, but I reckon the extra 30-60 minutes spent on setting everything up is worth it. It's easier for me to keep track of things and implementation. + having a good foundation is crucial. Rather spend a little extra time on that than to waste time later on trying to navigate some nonsensical pasgetti.

Domain holds the data model entities and their relationships.

DAL has the context file + necessary repos, interfaces, uow file to later access the database.

BLL doesn't have much, a few "optimized" domain entity models and their factories. Ended up removing them from DAL since logically they don't belong there.

The program reads the data from the csv file and adds them to the database accordingly. Given some old csv data exists, the data tables are all cleared before starting work on the new csv file.

The program first tries to access the file specified by the command line parameter. If something goes wrong, the user can re-enter the information from the console application.

Task 1 is rather straight forward, nothing to comment on it really.

Same can be said about task 2. The repo method that solves the second task has a collapsed region that has commented code; an alternative solution. The active solution goes over the original Disease entities, going through a "complex" object tree. The other solution first simplifies those disease objects and then checks a simple list of symptom name strings.

My algorithm for task 3 chops up the disease list until there's just one of them left. Since every disease has an equal chance of appearing and that a minimum amount of questions must be asked, I figured I should order the initial disease list accordingly – diseases are ordered by their least popular symptom (by disease count). For example: say there is a symptom that has just 1 disease. That disease would be the first one in the initial list. The idea is that, if the patient has that specific disease, then the least amount of questions have been asked.