CSCI 3060U Course Project - Winter 2022

Phase #2 - Front End Rapid Prototyping

Due Sunday, Mar. 6, 2022 (11:59pm)

In this phase, you will design and rapidly program the first version of your team's Front End. Since the next assignment will involve running your Requirements Tests from Phase #1, do not do full testing of your Front End at this time **(no marks are for correctness yet!)**

You should hand in:

1. A design document for your Front End, giving the overall structure of your solution, showing the classes and methods in a UML class diagram and a brief (one sentence) description of the intention of each method and class in a table.
2. The first version of the source code to implement your design. This version should run on at least some inputs, but should not yet be completely tested (since that will be the next assignment, and it's better to leave some failures until then).

Your solution to this assignment will be judged on the clarity and readability of the design and the code, so work at using your best programming practices, including naming variables, classes and methods meaningfully and commenting appropriately to make it easy for another programmer to understand.

Your solution to this assignment will not be judged on its correctness – this is a rapid first version, not final product. Therefore design and program it to work correctly, but don't worry about getting it fully debugged or tested yet since that's what we'll do next.

Submissions must be completed electronically through Canvas.

Marking Scheme for Phase #2

Marking for Phase #2 will be out of ten, according to the following criteria. In each category, marks will be assigned between zero and the number of marks shown, to a resolution of 1/2 mark.

Design

Architecture — 2 marks
- *clearly documents structure of solution*
- *explicitly describes intention of each class and method*
- *accurately reflects solution structure*
- *clearly shows where inputs and outputs fit in*

Completeness — 2 marks
- *evidently addresses all required functionality*
  (solution has parts to address all required operations and results)
- *has specified inputs, outputs and files (only!)*
  (Front End takes in transactions on std input, produces responses on std output, has two input files (current user accounts file, available rental units file) and has one output file (daily transaction file))

Source Code

Structure and format — 2 marks
- *code is structured and formatted such that architecture is clearly visible in code*
- *naming of classes and methods clearly reflect their role in the solution*
- *as little cloning or redundancy as possible*

Maintainability
- *avoidance of coding tricks and hacks*
- *simplest solution possible, no frills and extras*
- *clearest solution possible, no gratuitous optimizations* 2 marks

Internal documentation
- *clear naming of all variables and constants to reflect their role in the solution*
- *comments at beginning of every class and method clearly documenting their interface and intention* 2 marks
- *comment at beginning of main program documenting overall program intention, input and output files, and how the program is intended to be run*

TOTAL **10 marks**