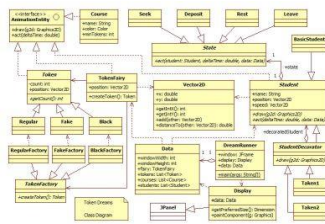


CENG 443 – Object-Oriented Programming Languages and Systems

Spring 2017 - Homework 3

System Design Practice

Selim Temizer



Feedback : Informal (no set date), you may post questions on the newsgroup

Due date : June 15th, 2017 (Submission through COW by 23:55)

In this homework, a verbal description of a software system is provided (similar to a Software Requirements Specification, SRS, document, but lacks all sorts of formalism), and you will create a UML class diagram that is ready (as much as possible) to be directly implemented (you don't need to show getter/setter methods, but the rest of the diagram should be as complete as possible with all major, and some minor fields/methods included).

The description of the software system is as follows: The users of the system will be taking pictures by their smart phones, or by regular cameras, and then they will input these images to our software. The system should allow the users to wirelessly upload the images from their smart phones (for example, through a web interface), or, the users may choose to transfer the images to the host computer manually, by flash disks, etc. Once we have the image(s), the software should timestamp each image and store them in a database. The software should let the user browse and select one of the stored images for processing. After an image is selected, it should be displayed on the screen through a GUI. The GUI should also let the user choose an image processing algorithm from a large set of algorithms, and apply it to the image. The GUI should show both the original image, and a second image with the algorithm applied to it. If the user wishes, the user should be able to choose other algorithms, and apply them to the image, in a sequential order. The system should also let the user undo the actions. For each algorithm, there will be various parameters (specific to each different algorithm) that the user may set before applying the algorithm. The full set of image processing algorithms are already implemented in a library such as OpenCV, therefore, in your UML diagram, you don't need to have classes or methods that correspond to different algorithms. However, you may need class(es) that act as a wrapper and connect the third party library to your software. Also, the system should be able to switch libraries (so that, another third party image processing library might be used instead of the OpenCV library). Once the user applies one or more algorithms on the image, and happy with the result, the user should be able to instruct the system to generate a report (like an MS WORD or PDF document that contains the original image, final image, and a short text that describes which algorithms are applied (with what parameters and in what order). You may also assume that there is a separate library that can create WORD or PDF documents, but you again may need to have class(es) that help your application use those third party libraries for document generation. If the user wishes, the report should also be saved in a database. The database that the system uses should be switchable (i.e., the system should be able to work with different database applications). The user should also be able to search through previously saved reports, select one, and email it to someone else.

Additional (technical) specifications: You need to use at least 2 software design patterns in your design. For bonus points, you may use more software design patterns and/or you may also use architectural design pattern(s) in your diagram.

Bonus: You may prepare mock-up GUIs for bonus points (sketch by hand, or use various tools that help designers quickly create a GUI for just demonstration purposes, without any real functionality).

What to submit? (Use ***only ASCII characters*** when naming all of your files and folders)

1. Your UML class diagram in digital format (in StarUML v1 or v2 format).
2. A very short report (Word or PDF) that contains the UML diagram (as a full-page figure), and textual description of your design decisions, and the design patterns that you used in your diagram. The text should be at most 2 pages long.
3. Optionally, if you prepared mock-up GUIs (one or more), you may add them as images at the end of your report.

Zip the 2 items above together, give the name <ID>_<FullName>.zip to your zip file (tar also works, but I prefer Windows zip format if possible), and submit it through COW. For example:

e1234567_SelimTemizer.zip

IMPORTANT: Late submissions (even for 1 minute) will not be accepted!

We will only grade submissions on COW, and system closes automatically at due time!