Workshop 2 by Frida Holmström

Peer-Review by Johan Landbris & Jonas Mattsson

We had to add a file ”registry.txt” to be able to run the exe. Otherwise it said class not found.

Is the Architecture ok?

The architecture looks good. Model-view separation is ok and follow the MVC pattern [1].

Is the requirement of a unique member id correctly done?

Yes. We had problems whit correct unique id's when deleting a member and then adding a new one, but you managed that well!

What is the quality of the implementation/source code?

Overall, the code is good and easy to follow. The naming is self-explanatory. Not that many comments but we don't see any need for that when the code is well written.

As a developer would the diagrams help you and why/why not?

As you pointed out in the readme file, there is room for improvement. One sequence diagram is missing, but the one you have done is good. The class diagram could have a few attributes and operations.

What are the strong points of the design/implementation, what do you think is really good and why?

Very easy to understand and navigate through the program.

What are the weaknesses of the design/implementation, what do you think should be changed and why?

Some inputs needs to be checked. Personal numbers for example consist of letters.

If we close the program without using the "0" choice in the menu, the system is not persistent and nothing is stored to the registry.

Do you think the design/implementation has passed the grade 2 criteria?

Yes. Overall you have done a really good implementation. If you fix your diagrams you will probably pass.

[1] Larman C., Applying UML and Patterns 3rd Ed, 13.7 Guideline: The ModelView Separation Principle