

Project PM: OO programming project, TDA367/DIT212, LP4 2013

General

- The project group's first task is to decide on one of the project proposals (or better if you have an idea of your own). This isn't critically related to the grade, almost any application can be "complexified" by adding more requirements.
- Second task is to name the project. The group number (assigned) and the name must be used in all communication with assistants, course responsible, etc.!
- The third task is to setup the project site. The project must use the Git versioning system.
- Final task is to send a mail to course responsible with the following content; Group number, Group/project name, URL to Git repository and lastly info about all members. The format for members should be (phone optional);

```
Lastname, Firstname, email, pnumber, phone  
Lastname, Firstname, email, pnumber, phone  
Lastname, Firstname, email, pnumber, phone  
Lastname, Firstname, email, pnumber, phone
```

If using any alias you must inform us!

Requirements

This is what you are supposed to produce. All required documents and code should be down-loadable from the Git repository. NO handing in of any papers!

The application

- At least a standalone desktop Java application with a graphical user interface using some kind of MVC design. It's assumed that the Git-branch "master" is the final delivery.
- The size of the application must be 2400 source lines of code (SLOC, comments excluded) or more (i.e. at least 600 SLOC/group member).
- The application must realize at least 8 use cases.
- Application must have a sound design and implementation.

- The application must be possible to run on Win/Mac/Linux. There should be a script to launch the application. If additional information is required to run the project there should be a README-file explaining what to do.

The documentation

All documentations should be in pdf format. Standard templates should be used (see course page). The group must handle in:

- The RAD
 - There should be at least 6 use cases as full texts.
 - There should be a class diagram for the analysis model.
- SDD
 - There should be at least one package diagram with dependencies (or separate dependency diagram).
 - There should be class diagrams for the more interesting packages, in particular the design model (not GUI, utils, ...)
 - For the dynamic model it suffices with 2 sequence-diagram.
 - NO auto-generated UML (other UML welcome).
 - NO Javadoc.
- The group meeting agendas.

Project grading

It's very hard to formulate exact criterion's for the project. If in doubt consult your assistant.

Examples

Below are some typical examples (non exhaustive).

U (U)

Not possible to run or project too small, too few use cases, design too strange/bad (hard to understand). Documentation missing or poor.

Grade 3 (G)

A project fulfilling the base requirements above. Which means: Functionality; 8-10 use cases implemented and working. Simple but functional GUI. In-house MVC. Clean implementation of at least one subsystems, application uses interfaces to reduce dependencies. There are some usable tests. Documentation is short but correct (and in sync. with the application). It's possible to trace requirements and follow the process from the agendas. The presentation is ok.

Grade 4 (G)

(NOTE: This is ****not**** the union of ..., it's a list of possibilities) A somewhat larger application with a more sophisticated design, more flexible. Solid code and packaging, everything is easy to locate. Clean subsystems. More functionality (use cases) implemented, possible attention to non-functional requirements. Good control over dependencies, clean interfaces. Possible use of external libraries. A more advanced GUI. External configuration and data. Test suites cover a lot of application code. It's easy to trace the process. The process is well controlled. Documentation is short and correct and obviously useful for others. The presentation gives a good view of the strength and weakness of the application.

Grade 5 (VG)

Like grade 4 but even more and/or possible higher design/technical level.