Groupe 9

Project follow-up and conception choices.

For this first iteration we had three main tasks. Define a clean MVC architecture to work with, implement the login story and implement the import/export functionality. Those three tasks represented a larger amount of work-time that we had originally anticipated and a tendency to work over-time, coming from all members of the group, was noticed. This is something that we will, of course, have to improve in the next iteration.

Respecting the Xtreme way of development.

Except for the tendency to work overtime mentioned in the introduction, we are quite satisfied with the dynamic of the group. We managed to find time to work by group of two or three and to rotate those groups each week. As a result the less experienced team members have already started to improve, which gives us hope that we will manage to be all on an equal footing by the end of this project. Everyone was able to choose user stories freely and none of the members had a leader role.
In addition, a drink of team building was organized at the beginning of the first iteration.

A first MVC architecture.

We decided to use a classical approach.
Each layer should only use the interfaces defined in the corresponding layer.
The implementation is associated to it’s interface through the loading of a configuration file (using introspection).
This allows us to use different environments for tests (the use of Mock), production and development.
A singleton was used to load the configuration for the tests only once.

The Business layer is equivalent to the “Controller” layer in MVC. The logic is defined in the Use Case Controllers (UCC).
The View and Persistence layer communicate with the Business layer using the DTO interfaces. The DTO acts as bags of data usable to communicate with the Business layer.

The Persistence layer corresponds to the Model and communicates with the SQLite database.

Finally the View layer contains the JavaFX related class and files.

The login user story

We started the first iteration with this story. JavaFX was new for all of us, it took some time to grab the basic concepts. Moreover, some have not practiced Java programming for some time. Because it was new to us, we encountered some unexpected difficulties that slowed us

down. Working with such a large group means that everyone has his own way of programming and designing its architecture. Hence, it took a bit of time to analyse and understand the structure of the project. User input validation, the singleton design pattern for the only connected user, hashing and salting passwords with BCrypt that required Spring.

The import/export user story

Due to a lack of time, the import/export hasn't been properly implemented, it works, but not in a clean way. There is a problem with the logic for the import function and more precisely, when we insert the pathfile in the database. This has repercussions on the rest of the import/export activity,

Conclusion

As we said above, the structure took more time than expected.
We are also not satisfied with the state of our import/export, who we think it might benefit from some 10 points for refactoring in the next iteration.