**PLAN:**

1. **Project implementation and group reflection**
2. How to design this project and with which tools
3. Setting up the project on our computer
4. **Distribution of different tasks / start of project management**
5. Trello
6. Gantt
7. Pert
8. **DB**
9. Stored procedures
10. Maps creation
11. Diagrams
12. Class Diagram
13. Package Diagram
14. Component Diagram
15. REX
16. Hugo
17. Mathieu
18. Thomas
19. Peyton

**LORANN PROJECT**

**Type Dev : Java**

**TEAM PRESENTATION:**

**Constraints :**

The use of Java, Maven, Git and Junit is mandatory.

No graphical framework other than Swing is allowed.

Start of the project: Monday, May 28th.

End of the project: Wednesday, June 6th.

The development team will consist of up to 4 members.

A project manager:

It will be responsible for:

- the proper distribution of the load between the team members

- the deliverables (time and content)

- the making of optional appointments (but recommended) with your tutor

No SQL request must be present in the Java code. All calls must be via stored procedures.

**Deliverables :**

- Full JavaDoc of your project (tests included)

- Full JXR of your project (tests included)

- SureFire report of your project

- Component diagram

- Package Diagram

- Class diagram (one per package)

- Sequence diagram (as much as you will find it useful to understand and explain the

operation of your program)

- A GIT report to identify the production of each member of the team.

- Any other documents you deem necessary (CDM, stored procedures, other

diagrams, comments, ...)

Deliverables generated by our Java code

Different diagrams to better understand the structure of the code

Project Management Tools / Group Work

1. **Project implementation and group reflection**
2. **How to design this project and with which tools**

At the very beginning of the project, we took some time to get started. The reading and the comprehension of the subject was long enough. Once the project was correctly read, we shared our different point of view / idea of ​​project design. We have drawn ideas and technical points not to miss. We made a list of what we needed for a successful project:

- Common IDE

- Functional git for members of the group

- Functional Maven

- UML software

- GANTT software

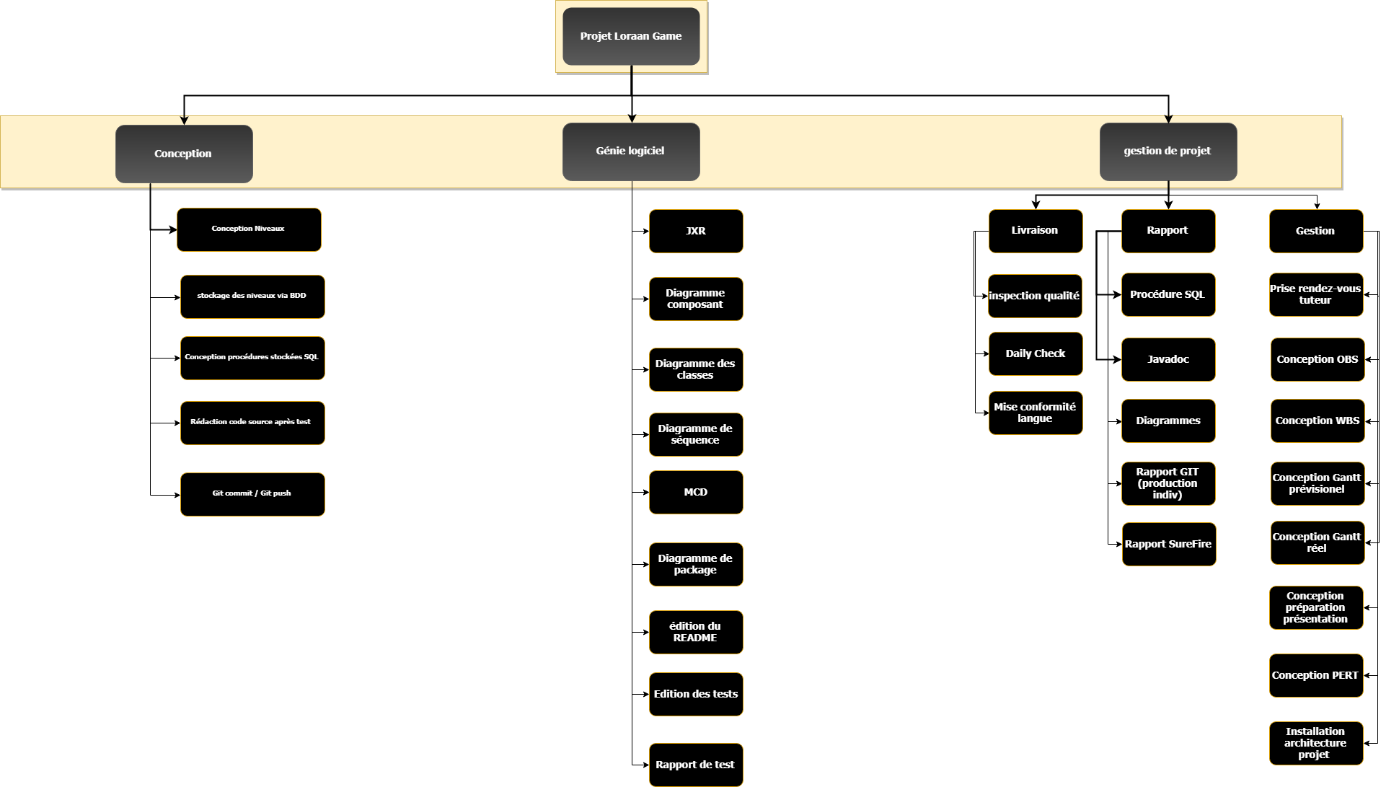
- PERT software

1. **Setting up the project on our computer**

Once we started in our project, we decided to install the project on our IDEs to start from a functional project, however we realized that we could not do a common GIT, so we had to create separate GIT repositories, and put them in common at the end.

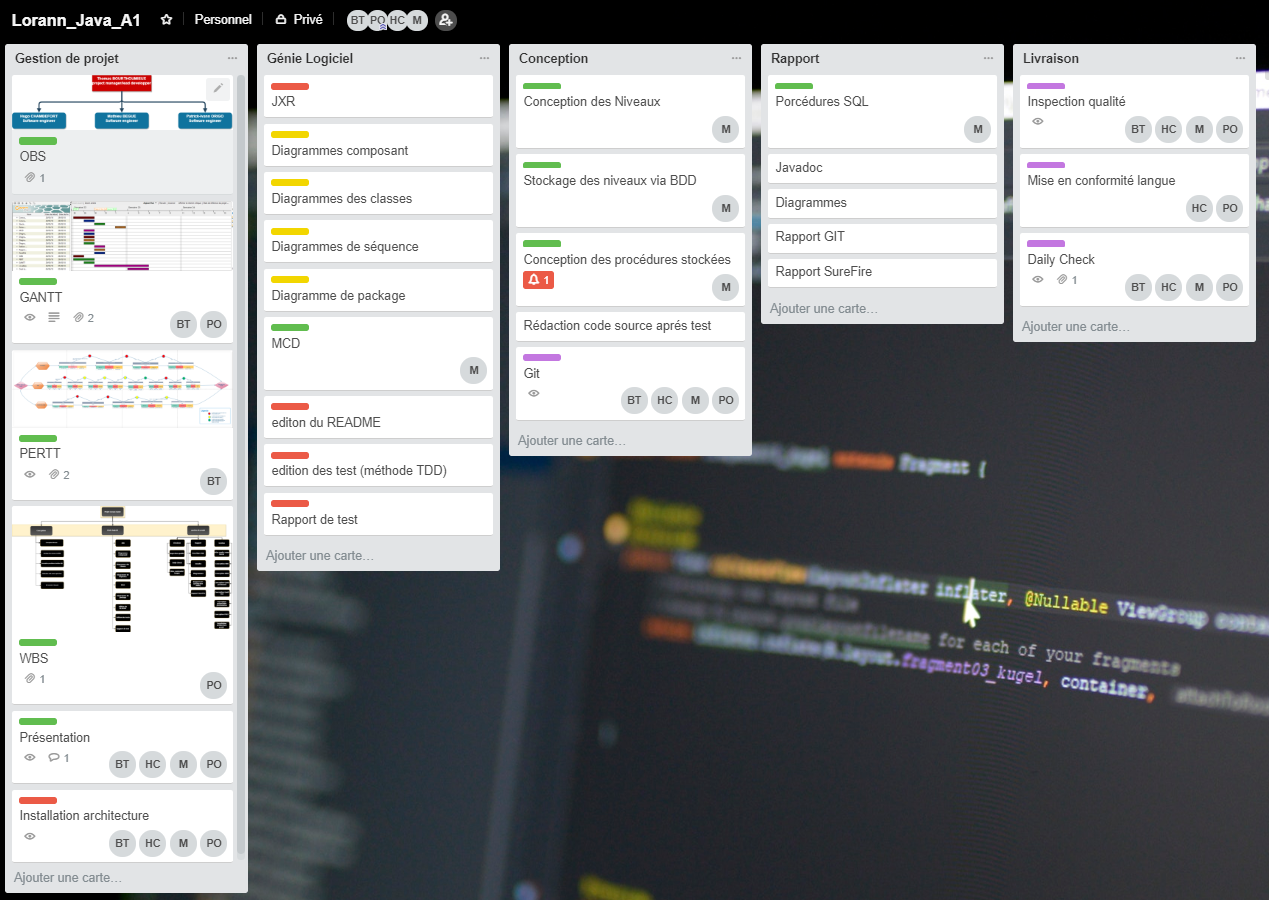
1. **Distribution of different tasks / start of project management**
2. **WBS**

First, we had a WBS in a group to simply establish all the tasks to be done for the realization of the project. We have divided this project into 3 main branches design software engineering and project management.



1. **Trello**

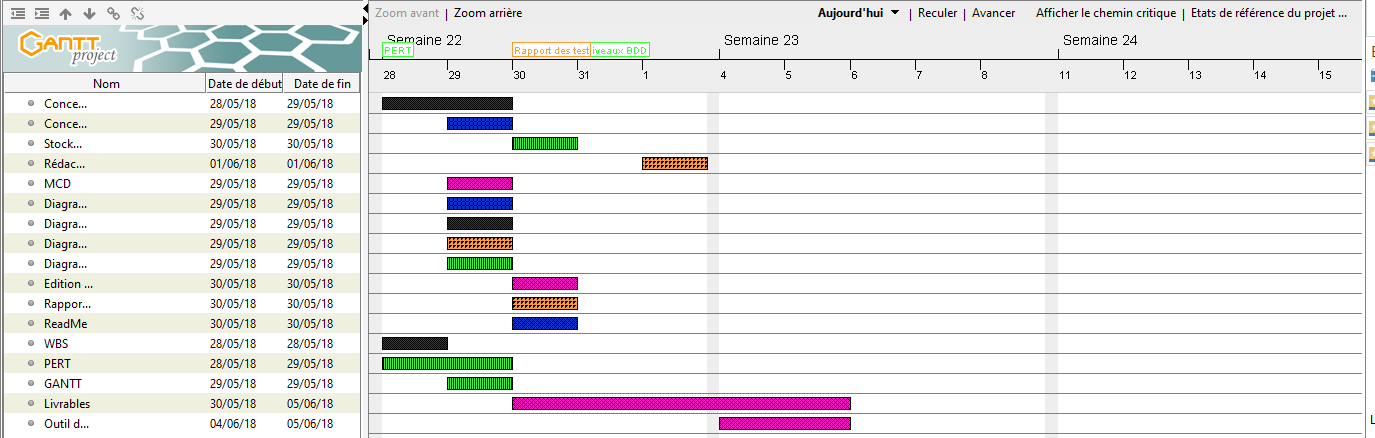
So, we used the Trello tool to first establish the different tasks to perform. We also plan and assign the different tasks. Trello was our first project management software



Trello - LorannJavaProject 1

1. **Gantt**

Then to be able to see if the distribution of the tasks between the members of the group and if they were of consistent size and if they could be realized during this week of project.

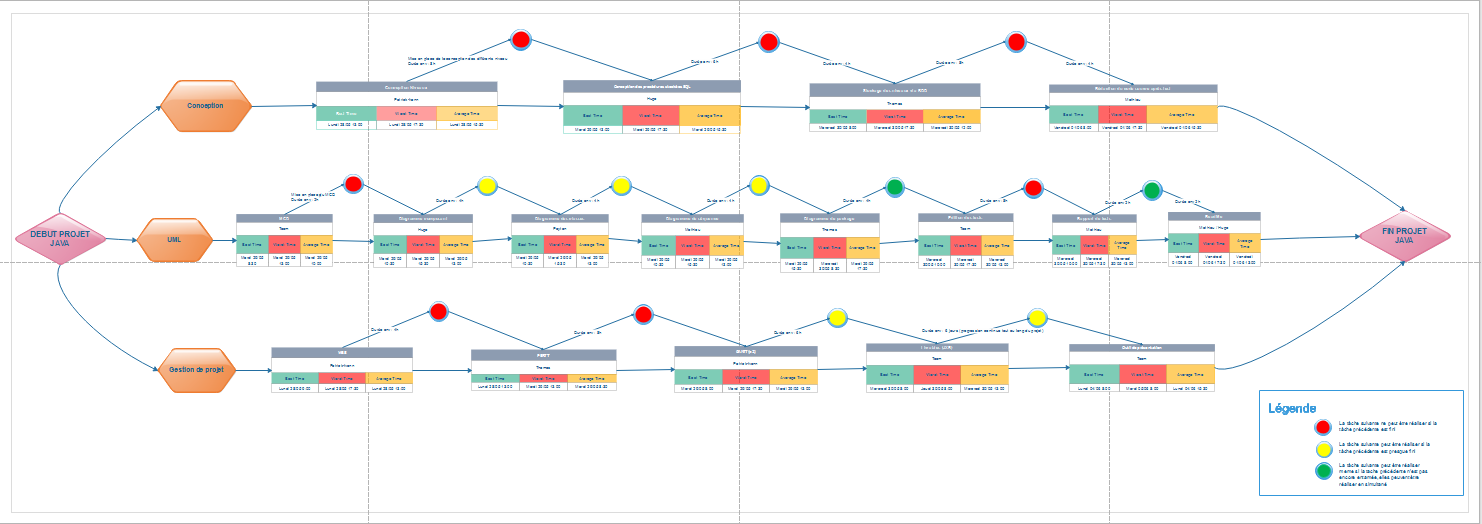


GANTT - LorannJavaProject 1

*To better see our Gantt click* [*here*](https://github.com/BOURTHOUMIEUXThomas/PROJET5/blob/master/gantt.png)

1. **Pert**

We then established a PERT diagram to be able to see the dependencies between the different tasks as well as the potentially adoptable critical paths if a task is harder or longer than expected.



PERT - LoraanJavaProject 1

*To better see our PERT diagram click* [*here*](https://github.com/BOURTHOUMIEUXThomas/PROJET5/blob/master/PERT.png)

1. **DB**
2. **Stored procedures**

A stored procedure is an SQL query that will perform a specific task, but it will only execute when given permission. To realize the different stored procedures we first defined in which database we wanted to store them. Then we declared them and we wrote the queries that will only execute during the call of the procedure.

To see our stored procedures check: <https://code.empreintesduweb.com/12853.html>

1. **Database/Map creation**

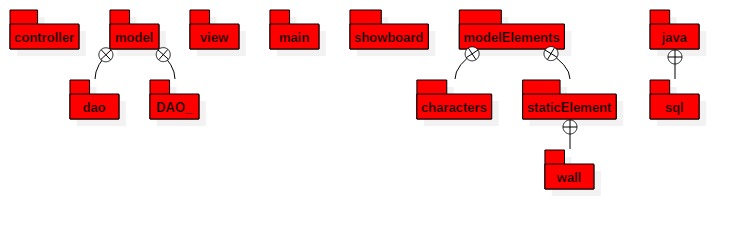
To create the database, we used the Workbench software. In the database we have created a table that will allow us to store our data in the form of a table. In the table we have created 3 columns: an ID column, a name column and a column that contains the level (the level is in the form of a long string of characters that in the java code will display images).

To see our Database code check : <https://code.empreintesduweb.com/12852.html>

**Table of distribution of tasks:**

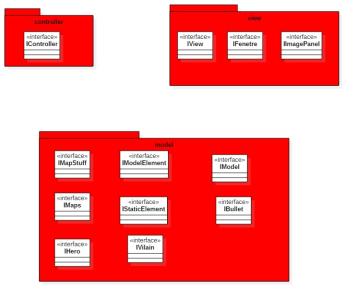
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Monday** | **Tuesday** |
| **Mathieu** | Implementation of the project  Group reflection | MCD  Level design  Setup env | Storing levels in the database | Stored procedures and queries | View design | View design | Code comment |
| **Hugo** | Implementation of the project  Group reflection | Setup env | Diagrams | View design | View design | View design | Code comment  Presentation tools |
| **Peyton** | Implementation of the project  Group reflection  WBS | Setup env | Diagrams / software architecture | Model design | Model design | Pooling and debugging | Pooling and debugging |
| **Thomas** | Implementation of the project  Group reflection | Setup env  Gantt  Pert | Pert  Diagrams | Controller design | Controller design | Shoot  Debugging | Deliverables |

**Diagram Package:**

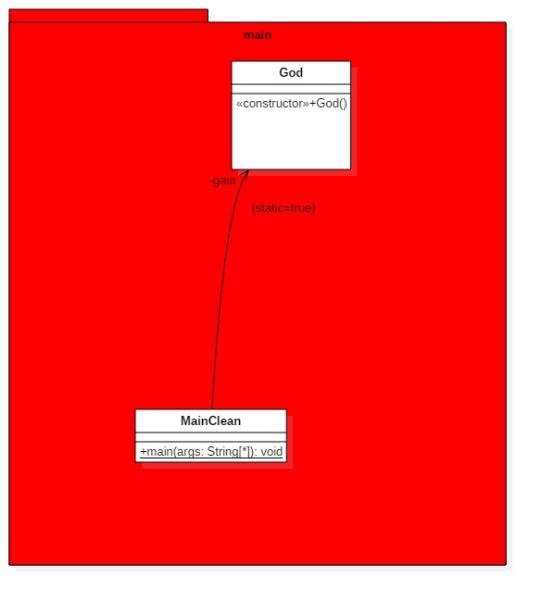


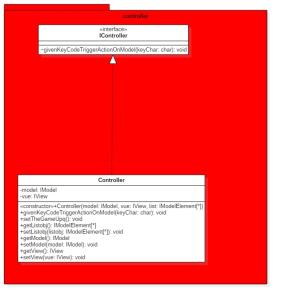
Package diagram 1

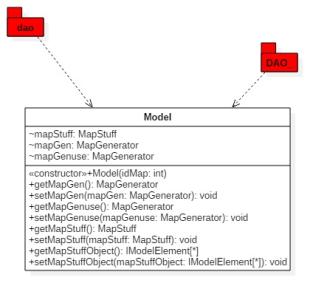
**Component diagram :**



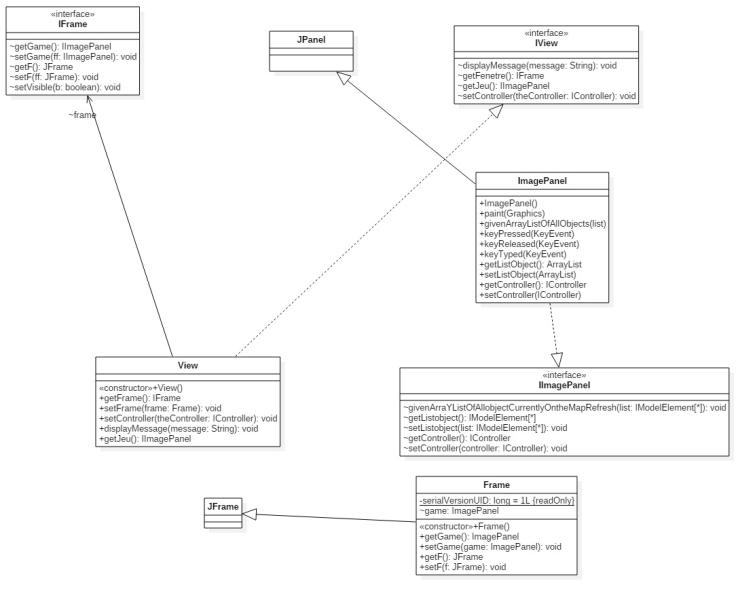
Component Diagram 1

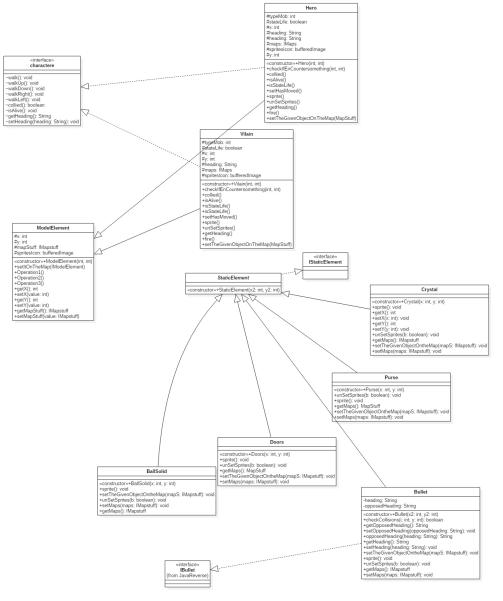




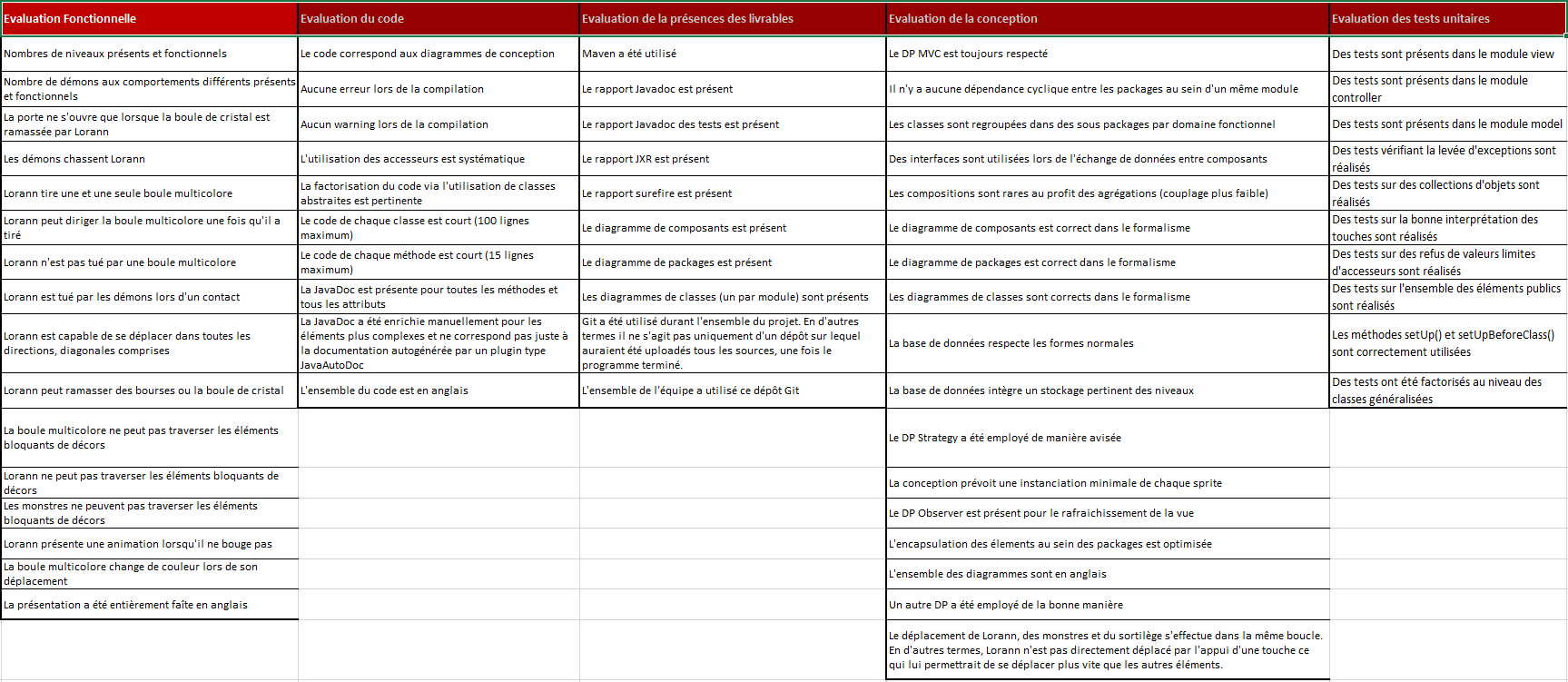


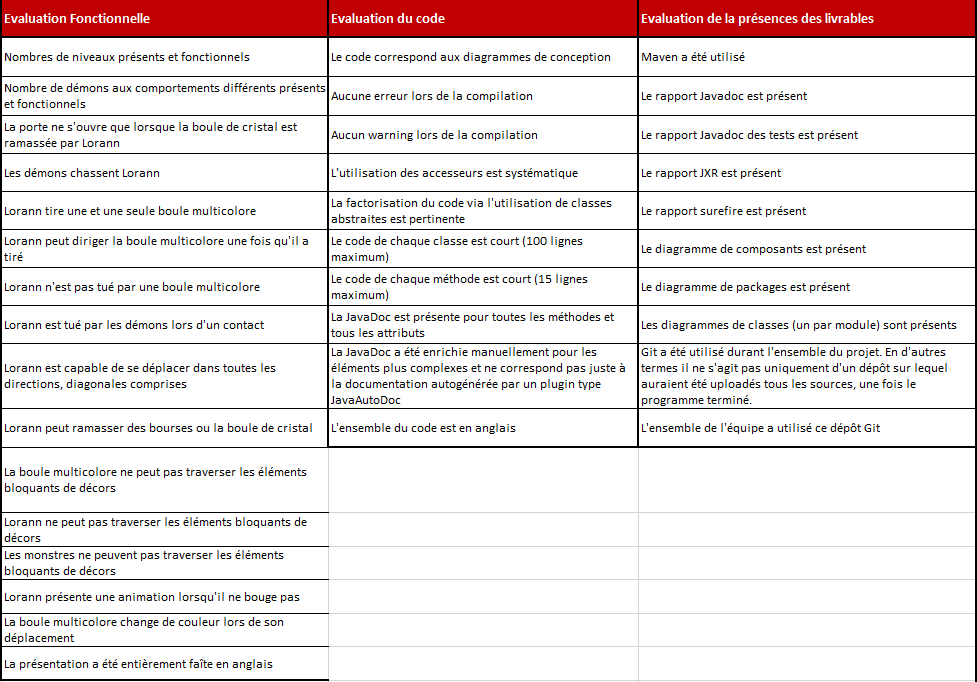
**Class diagram:**





**Evaluation grid:**





1. **REX**

**Hugo:**

During this project I did not have the same skills as the member of my group but with their help I tried to do the task that was entrusted to me in the view. My chief and the others were very helpful even though I could not help them the way I wanted. I found this project quite hard for beginners like us, even if we had all the prosits, I think that 7 prosits are not enough for beginners to do a job like this one even if this project helped me to learn new things

**Mathieu:**

During this project I have worked on the SQL. In this part I needed to create the database, a table and stored procedures. After the realisation of this part I have try to help my team mate on the View but that was very hard, and I don't have all knowledge for this task. I did the most I could to help my teammates. I think this project is difficult for beginners to accomplish a work like that.

**Thomas**:

This project was very complicated especially on the realization of the source code and the debugging. I found the difference between the prosits and the project too big. We have not had enough practice (on the pure code and on Maven). Ivann was a crucial element in the realization of the project. He knew how to help for the sight and the debugging during the pooling. For my part I realized the controller, the deliverables, the pert, the gantt and all the tasks perform in group. In this project the hardest thing was to pool our codes and debug everything. In the end I am quite happy with the rendering of the band.

**Peyton:**

I've encountered many problems during this projects on every Levels.We had understanding and synchronization problem in the groups. I was in charge of the software architecture through UML Diagram then i had to code the Model and at last, to integrate the whole code, so I had the Opportunity to code a lot of things but because of delay on the view, I had to take that part too to optimize it. I did a lot of overtime but the game is still unplayable and nobody on the team was able to help me until Tuesday but, I found help on the controller. But during the las day I had to re-work the whole architecture because the first one Had a lot of dependency with other classes. Now i use interfaces everywhere but I took a lot of time to debug everything under maven. we still have issues with maven and GitHub. It wasn't a good experience on any point.

# END