



Projet SE :
Prototype/Conception



SIMPLIFY

Par Thomas Fournier, Salim Belkhir, Ayoub Hakemi et Emma Mathieu

Sommaire

<i>1 - Contexte</i>	<i>2</i>
<i>2 - Organisation</i>	<i>3</i>
<i>3 - Problèmes rencontrés</i>	<i>3</i>
<i>4 - Solutions apportées</i>	<i>4</i>
<i>5 - Remarques</i>	<i>4</i>
<i>6 - Livrable</i>	<i>5</i>

1 - Contexte

La troisième étape de ce projet consistait à réaliser la création du login de notre application. Pour cela il a été demandé, de réaliser le diagramme de classes de cette partie ainsi que le diagramme de séquence. De plus, nous devions réaliser le diagramme de package de notre application afin d'avoir une arborescence de fichiers définis. Evidemment, pour faire suite à tout cela, nous avons codé entièrement le login, que ce soit la user interface, la business logic et la persistance layer, en réalisant évidemment des tests.

Cette étape a alors duré une semaine et comme l'étape précédente nous avons beaucoup collaboré ensemble pour réaliser le travail demandé. D'ailleurs, le responsable de cette étape était Thomas Fournier.

2 - Organisation

Pour cette phase, nous avons utilisé lucidchart pour réaliser les différents diagrammes, Notion pour assurer un suivi du projet, Discord pour discuter du projet et Github pour collaborer sur le code.

Concernant l'organisation, nous avons commencé le 30 novembre par débuté le class diagram lors d'une séance prévue dans notre emploi du temps. Nous avons ensuite organisé une réunion le 2 décembre pour terminer à la fois une première itération du class diagram et du package diagram. Pendant cette réunion, nous avons également organisé la suite du projet en répartissant les tâches, c'est à dire Ayoub et Thomas sur l'implémentation de la user interface, Salim sur l'implémentation de la business layer et Emma sur l'implémentation du persistance layer. Ensuite, nous avons organisé une nouvelle réunion le 5 décembre pour réunir les différentes parties et identifier le travail restant et comment le réaliser. Enfin, le 7 décembre, nous avons terminé pendant la séance prévue une nouvelle itération des différents diagrammes et du code avec en plus de cela la réalisation du diagramme de séquence au propre.

3 - Problèmes rencontrés

Lors de cette phase, nous avons rencontré deux difficultés en particulier. D'abord, réaliser le class diagram du login n'a pas été évident. En effet, il a fallu faire appel à de nombreuses connaissances vu en cours mais surtout les mettre en pratique pour la première fois.

Ensuite, la collaboration sur le code a posé quelques problèmes. Effectivement, lors du développement de chaque partie, il a été compliqué de faire le lien avec les autres parties. De plus, lorsque nous avons rassemblé les trois parties du code nous avons fait face à des conflits entre les différentes classes qu'il a fallu régler. Après, lors de ce même événement, nous avons eu un problème d'importation de JavaFx dans le projet. Enfin, l'utilisation de bibliothèques non partagées avec les autres membres a également posé des problèmes.

4 - Solutions apportées

Il a alors fallu apporter des solutions aux problèmes qui viennent d'être évoqués.

Ainsi, pour la réalisation du class diagram, nous avons d'abord revu le cours avant de le réaliser intégralement en collaborant tous dessus. En effet, être nombreux à réfléchir ensemble dessus nous a permis de confronter nos idées et d'apporter chacun les nôtres.

Au sujet de la collaboration sur le code, nous avons pour assurer un lien entre les différentes parties posé des questions à la personne de la partie avec laquelle nous communiquer. Ensuite, lors du merge, nous nous sommes tous posés devant un même ordinateur et nous avons regardé en détail les différents avant de longuement discuter sur chacun d'eux. Après, concernant JavaFx, nous avons été contraints de créer un nouveau projet sur lequel nous avons intégré les différentes parties réalisées précédemment. Enfin, au sujet de l'utilisation de certaines bibliothèques, nous avons utilisé des maven repository pour régler cela avec l'exception d'une seule, non disponible par l'intermédiaire de cette méthode et que nous avons donc décidé de retirer du fait de sa faible valeur ajoutée.

5 - Remarques

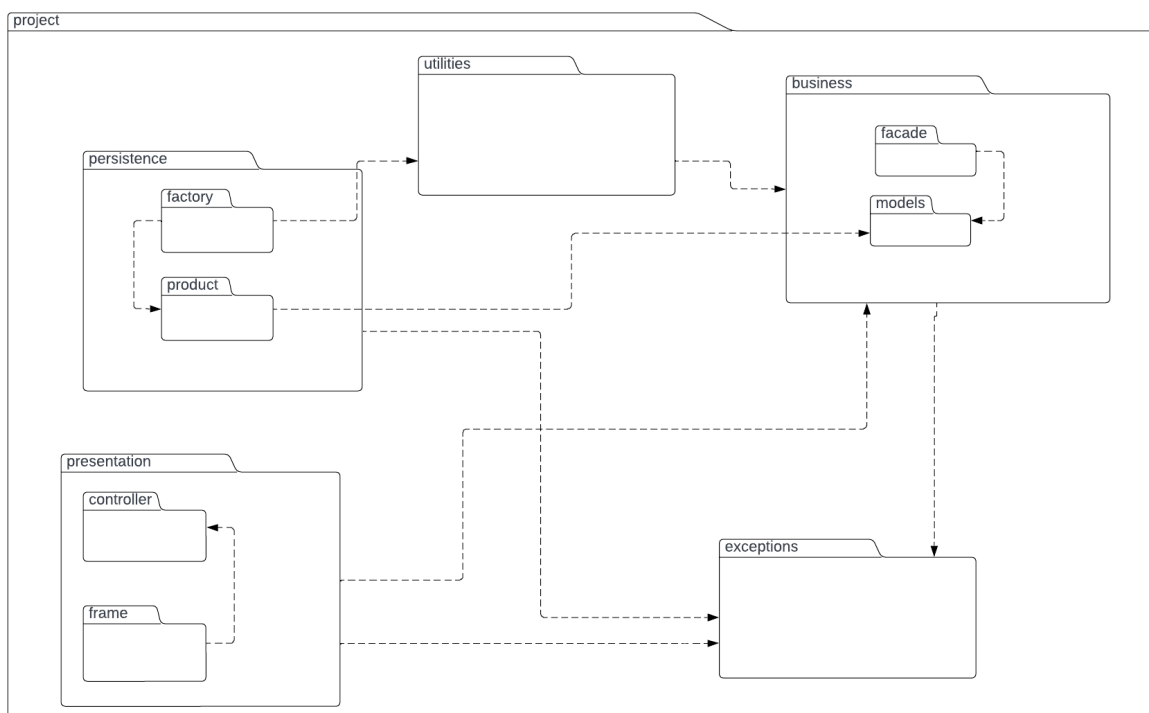
Au cours de cette étape, nous avons identifié différentes remarques à évoquer.

D'abord, abordons des remarques positives. Ainsi, nous avons pu identifier une bonne répartition des tâches entre les membres du groupe. De plus, chacun a effectué dans les temps les tâches qui lui était demandé. Dans le même temps, les nombreux échanges avec les autres membres ont été très enrichissants et donc d'apprendre de nombreux éléments. Enfin, nous allons au cours de cette phase apprendre des compétences et mettre en place des choses qui vont nous être indispensables lors de la prochaine phase où il faudra coder l'intégralité de l'application.

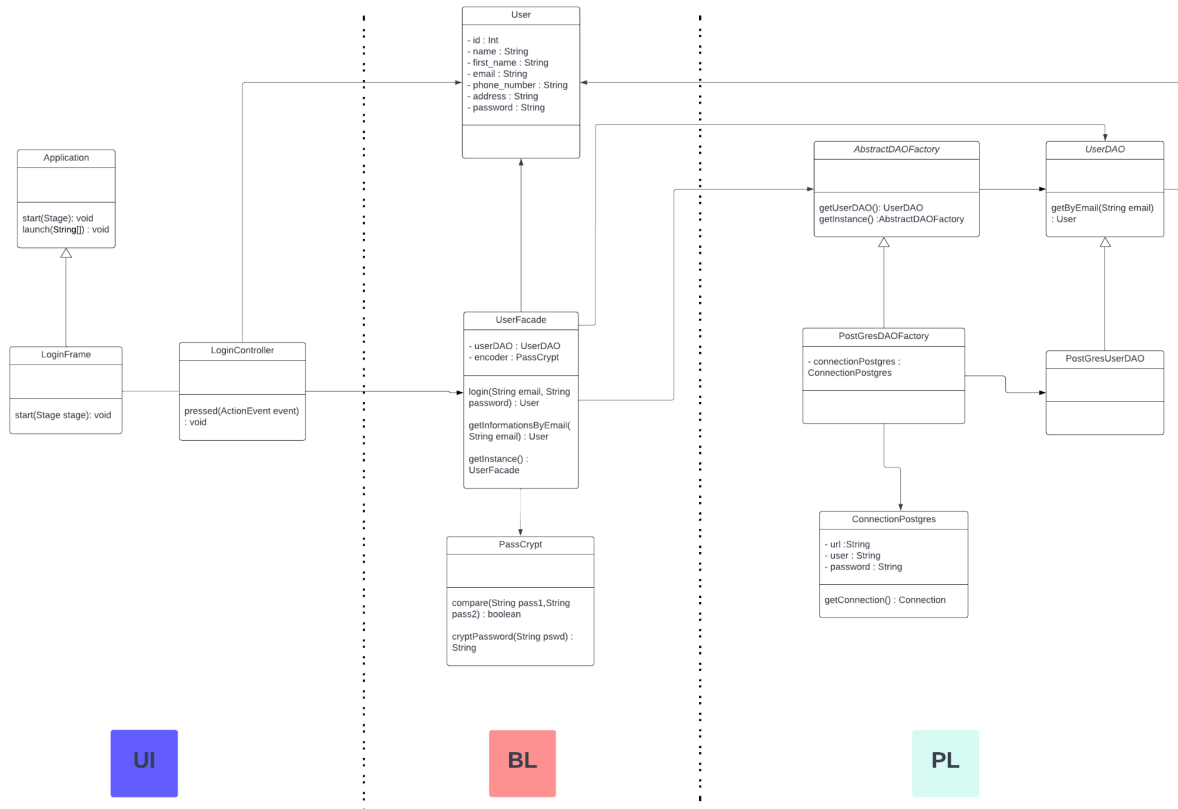
Ensuite, concernant les remarques négatives, elles sont toutes liées aux consignes de cette phase. En effet, il a d'abord été difficile de comprendre ce qui était exactement attendu. De plus, le fait que des choix techniques comme celui de l'utilisation de JavaFx ont été imposés nous déplait fortement puisque cela nous a obligé à revoir certains objectifs que nous avons.

6 - Livrable

D'abord, voici le package diagram global que nous avons réalisé :



Ensuite, nous avons construit le class diagram du login suivant :



Le point important à noter est l'utilisation du pattern factory. En effet, nous voulions utiliser un DAO mais nous ne pouvions pas mettre à l'intérieur une multitude de fonctions (`getByName`, `getByNumber`, `getSalaire`, `getTable`, etc). Ainsi, il fallait créer plusieurs DAO et le pattern factory était utile pour cela sans avoir accès au constructeur. Nous voulions également que cela s'adapte en fonction des bases de données (oracle, postgres, etc) donc il fallait utiliser la factory pour créer un objet lié à une base de données et celui-ci crée le DAO qui lui correspond.

Enfin, nous avons réalisé le diagramme de séquence du login ci-dessous :

