**BOUDET Alexandre**
**MARTEL Benoît**

**Département INFORMATIQUE**
**1ère année – Groupe A**

**Année 2016-2017**

# M2107 - PROJET DE PROGRAMMATION

## Rendu final

Création d'un éditeur de base de données

**IUT de Vannes**
M. LEFEVRE, professeur référent

# Table des matières

# I- Présentation de l'organisation du projet

**Comment avez-vous géré votre planning d'actions durant le projet ? 1ère période à 4, 2nde période à 2**

Durant la première période à 4, nous nous sommes répartis les tâches de façon à gagner en rapidité et productivité. Notre binôme s'est occupé de la rédaction du diagramme de classes ainsi que de l'établissement du diagramme de Gantt et de MS Project en général. Ça a été une partie assez difficile à réaliser dans le sens où nous avions quelques contraintes à prendre en compte, les partiels, le temps de travail par jours, etc. Ceci étant fait, le diagramme de classe nous a causé par ailleurs quelques souci, étant donné que nous avions du mal à nous projeter, sur les classe de « *View* » et « *Controller* » nous avons en partie respecté ce que nous avons mis dans notre cahier prévisionnel, quant au « *Model* », c'est la package qui a subi la plus grosse transformation, notamment sur l'ajout de nombreuses classes auxquelles nous n'avions pas forcément pensé comme par exemple les classes concernant les tables, et la classe RWFile a été abandonné en cours de codage (nous l'avons utilisé d'une manière différente). L'autre binôme avec qui nous travaillons s'est occupé de commenter les diagrammes de classe et de Gantt, ainsi que la réalisation des diagrammes de séquences.

Ici je n'expose que les tâches qui nous ont pris le plus de temps, les autres parties de la rédaction du cahier prévisionnel ont été faites au fur et à mesure avec un accord mutuel entre les deux binômes.

En ce qui concerne la seconde période, celle du codage, nous avons agi de la même manière. C'est-à-dire une intelligible répartition des tâches, Benoît s'est dévoué pour l'utilisation du JDBC et donc du « Model », quant à Alexandre il s'est occupé de la partie graphique donc du package « View ». Pour le « Controller », c'est une partie que nous avons modifié tous les deux. Ceci étant dit nous avons été pas mal embêter par le partage de code, nous aurions dû utilisé un GitHub qui est bien plus approprié au travail collaboratif. De manière plus globale, nous sommes fiers de ce que nous avons produit, car nous avons essayé de respecter au maximum ce que nous avons prévu de faire.

**Planning prévisionnel vs planning réel : Décrivez, expliquez les écarts constatés entre le prévisionnel et le réel. (Rem : décrivez ce qui s'est réellement passé : il n'y a pas de bonnes ou de mauvaises réponses)**

A propos de la comparaison prévisionnel vs réel, nous avons constaté qu'une grande partie de ce que nous avons prévu de faire a été réalisé. Cependant dans la partie optionnelle nous avons dû faire l'impasse sur deux choses, l'export sous un autre format, qui perdait son utilité et la représentation graphique du schéma relationnel ainsi que du diagramme de classes, qui devenait trop compliqué à réaliser, nous avons été peut-être un peu trop prétentieux. Et nous avons fini dans les temps sans trop de retard.

**<u>Réponse à la question suivante</u> : évaluez de 1 à 10, l'utilité de planifier d'organiser, planifier un projet. Justifiez votre évaluation en expliquant concrètement pourquoi ce planning a été utile ou pas – Précisez les + (intérêts, résultats + constatés...) et les – (difficultés rencontrées, résultats - constatés) - là aussi, il n'y a pas de bonnes ou de mauvaises réponses.**

Pour noter l'utilité de planifier et d'organiser un projet, nous mettrions sur une échelle de 1 à 10 la note de 5. Il est vrai que cette planification nous a beaucoup apporté sur les grandes lignes du projet mais dès que nous rentrons dans les détails nous avons préféré régler le souci sur le champ sans trop regarder le Gantt. Ce n'est peut-être pas la bonne technique, mais le Gantt nous a semblé trop subjectif pour pouvoir s'appuyer dessus. En fait, la vision des choses avant de commencer le projet est totalement différente que pendant sa réalisation. Par exemple si nous avons un souci pendant la réalisation d'une tâche, nous pouvons rester sur cette dernière bien plus longtemps que prévu. Cependant le diagramme de classe a vraiment été utile sur les classes dont on était sûre de leur existence.

**Quelles recommandations feriez-vous aux futurs étudiants de 1ère année qui, l'an prochain découvriront le projet de programmation ?**

Tout d'abord je conseillerais d'utiliser un Git pour pouvoir partager le code en toute simplicité, car cela peut porter préjudice si on ne trouve pas de bons compromis.

Ne pas hésiter à faire des pauses pendant le codage, cela permet de faire un point sur le projet et de se rafraichir les idées.

Ne pas hésiter à travailler le week-end sur le projet car c'est toujours mieux de finir en avance le projet pour ensuite prendre du recul et corriger les éventuelles erreurs.

Ne vraiment pas négliger les cours concernant le build XML, la javadoc, JUnit, etc.

Eviter de trop se focaliser sur une fonctionnalité optionnelle au détriment d'une fonctionnalité de base.

# II- Mise à jour du cahier des charges et du diagramme

## A) Cahier des charges de base

- Accéder à une base de données depuis Java
- Posséder une interface texte (console où sera affiché le résultat)
- Posséder une interface graphique (menu, boutons …)
- Ouvrir, enregistrer, enregistrer sous, nouveau fichier (stocker dans des fichiers où sont enregistré les requêtes et permettre leur lecture etc.)
- Visualiser des informations (BD, métadonnées, données, table, champs, fichier d'enregistrement)
- Édition, insertion, suppression d'une table, d'un t-uple et enregistrement automatique
- Sécurité pour la connexion à la base de données (attribuer droits aux utilisateurs)
- Posséder un champ de saisie de texte (pour l'écriture des requêtes SQL)

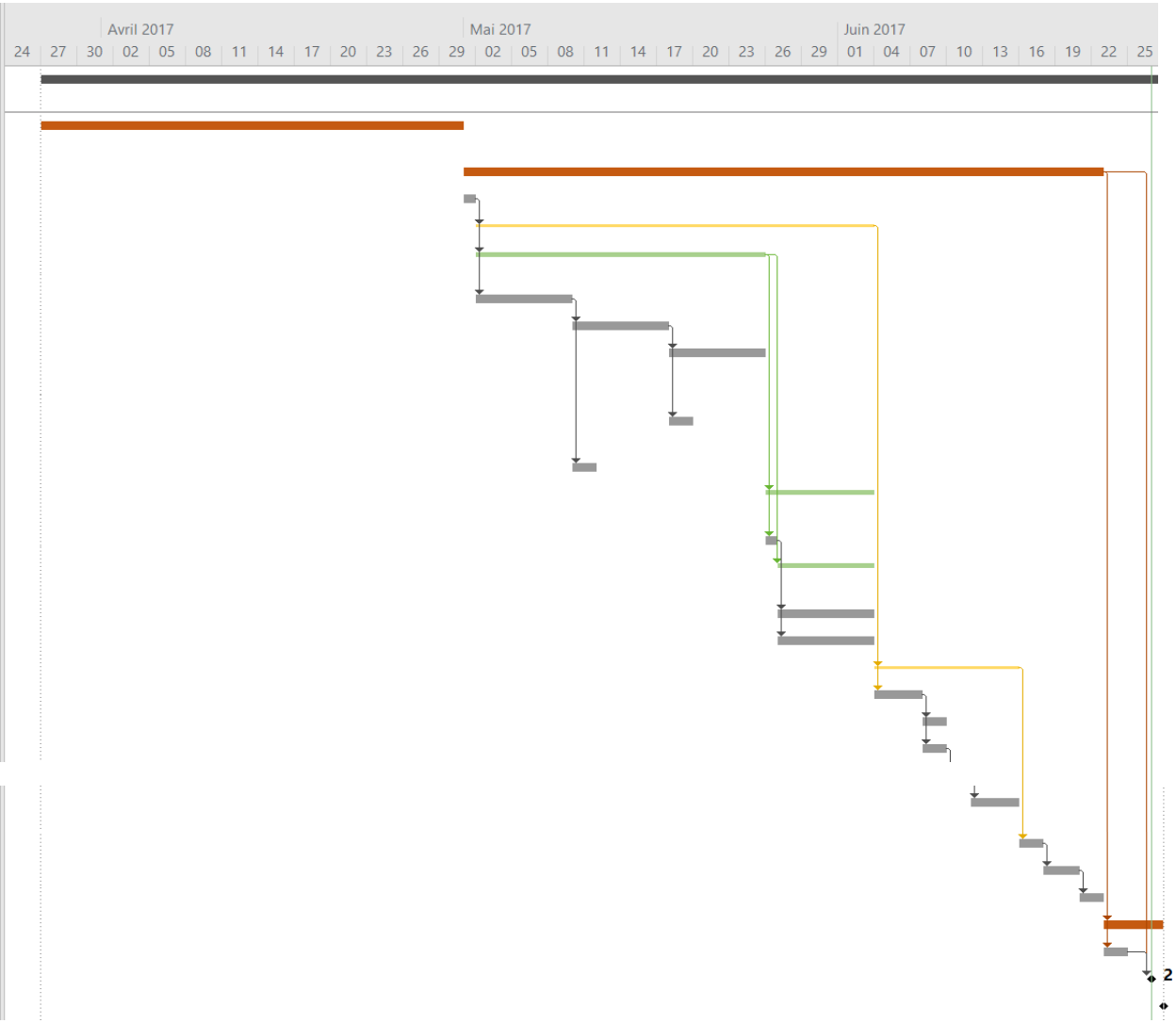
## B) Cahier des charges optionnelles

- Mini quiz pour apprendre le SQL
- Pouvoir créer une table juste avec un enchaînement de clic sur des boutons et de saisie de nom d'attribut
- Onglet aide (nouvelle fenêtre qui s'ouvre avec des commandes de bases, des conseils et autres)
- Représentation graphique (schéma relationnel, diagramme de classes)
- Changer le fond/les couleurs (esthétisme)
- Exporter sous un autre format
- Imprimer le fichier SQL qui a été chargé

Apparait en vert les tâches réalisées et en rouge celles encore à développer.

# C) Diagramme de GANTT

| | Nom de la tâche | Durée | Début | Fin |
|---|---|---|---|---|
| 1 | **Projet de programmation M2107** | **88 jrs** | **Lun 27/03/17** | **Mar 27/06/17** |
| 2 | **Etudes Préalables** | **35 jrs** | **Lun 27/03/17** | **Dim 30/04/17** |
| 8 | **Réalisation (codage)** | **48 jrs** | **Lun 01/05/17** | **Jeu 22/06/17** |
| 9 | Recherche JDBC | 1 jr | Lun 01/05/17 | Lun 01/05/17 |
| 10 | **Programmation de base** | **28 jrs** | **Mar 02/05/17** | **Sam 03/06/17** |
| 11 | **Programmation back** | **24 jrs** | **Mar 02/05/17** | **Jeu 25/05/17** |
| 12 | Accès à une base | 8 jrs | Mar 02/05/17 | Mar 09/05/17 |
| 13 | Extraction des données | 8 jrs | Mer 10/05/17 | Mer 17/05/17 |
| 14 | Implementation des fonctions pour editer, inserer, supprimer une table, un | 8 jrs | Jeu 18/05/17 | Jeu 25/05/17 |
| 15 | Création de statistiques sur la base | 2 jrs | Jeu 18/05/17 | Ven 19/05/17 |
| 16 | Sécurisation pour la connexio | 2 jrs | Mer 10/05/17 | Jeu 11/05/17 |
| 17 | **Programmation front** | **4 jrs** | **Ven 26/05/17** | **Sam 03/06/17** |
| 18 | Interface texte | 1 jr | Ven 26/05/17 | Ven 26/05/17 |
| 19 | **Interface graphique** | **3 jrs** | **Sam 27/05/17** | **Sam 03/06/17** |
| 20 | Menu de l'interface | 3 jrs | Sam 27/05/17 | Sam 03/06/17 |
| 21 | Bouton de l'interface | 3 jrs | Sam 27/05/17 | Sam 03/06/17 |
| 22 | **Programmation optionnelle** | **12 jrs** | **Dim 04/06/17** | **Jeu 15/06/17** |
| 23 | Mini Quiz | 4 jrs | Dim 04/06/17 | Mer 07/06/17 |
| 24 | Onglet aide | 2 jrs | Jeu 08/06/17 | Ven 09/06/17 |
| 25 | Imprimer le fichier sql | 2 jrs | Jeu 08/06/17 | Ven 09/06/17 |
| 26 | | | | |
| 27 | Représentation graphique de la base de données | 4 jrs | Lun 12/06/17 | Jeu 15/06/17 |
| 28 | Test code | 2 jrs | Ven 16/06/17 | Sam 17/06/17 |
| 29 | Debug | 3 jrs | Dim 18/06/17 | Mar 20/06/17 |
| 30 | Test après debug | 2 jrs | Mer 21/06/17 | Jeu 22/06/17 |
| 31 | **Déploiement** | **5 jrs** | **Ven 23/06/17** | **Mar 27/06/17** |
| 32 | Manuel utilisateur | 2 jrs | Ven 23/06/17 | Sam 24/06/17 |
| 33 | Rendu de l'application | 0 jr | Lun 26/06/17 | Lun 26/06/17 |
| 34 | Oral et fin de projet | 0 jr | Mar 27/06/17 | Mar 27/06/17 |



Sur le diagramme de GANTT ci-dessus, de légère modification ont eu lieu. Les plus notables sont les rétrécissements de la durée de certaines tâches, car en effet nous avions sur évaluée le temps qu'il fallait pour les réaliser. Par exemple la sécurisation de la connexion à pris beaucoup moins de temps que prévu tout comme la réalisation du manuel utilisateur qui est passé de 5 jours à seulement 3 jours.

Ce diagramme de GANTT nous a permis de suivre les principales phases du projet et de coordonnées le binôme sur les tâches qu'il devait réaliser.

# III- Description des choix techniques et algorithmiques

Nous avons choisi de développer l'application sous le modèle MVC afin de séparer la vue, le contrôleur et le modèle pour des choix de clarté dans le code mais aussi de développement. En effet pendant que l'un réalisait le modèle de l'application l'autre était chargé de réaliser la partie graphique sans importuner le travail de l'autre puisque c'est deux parties ne communiquent pas entre elles.

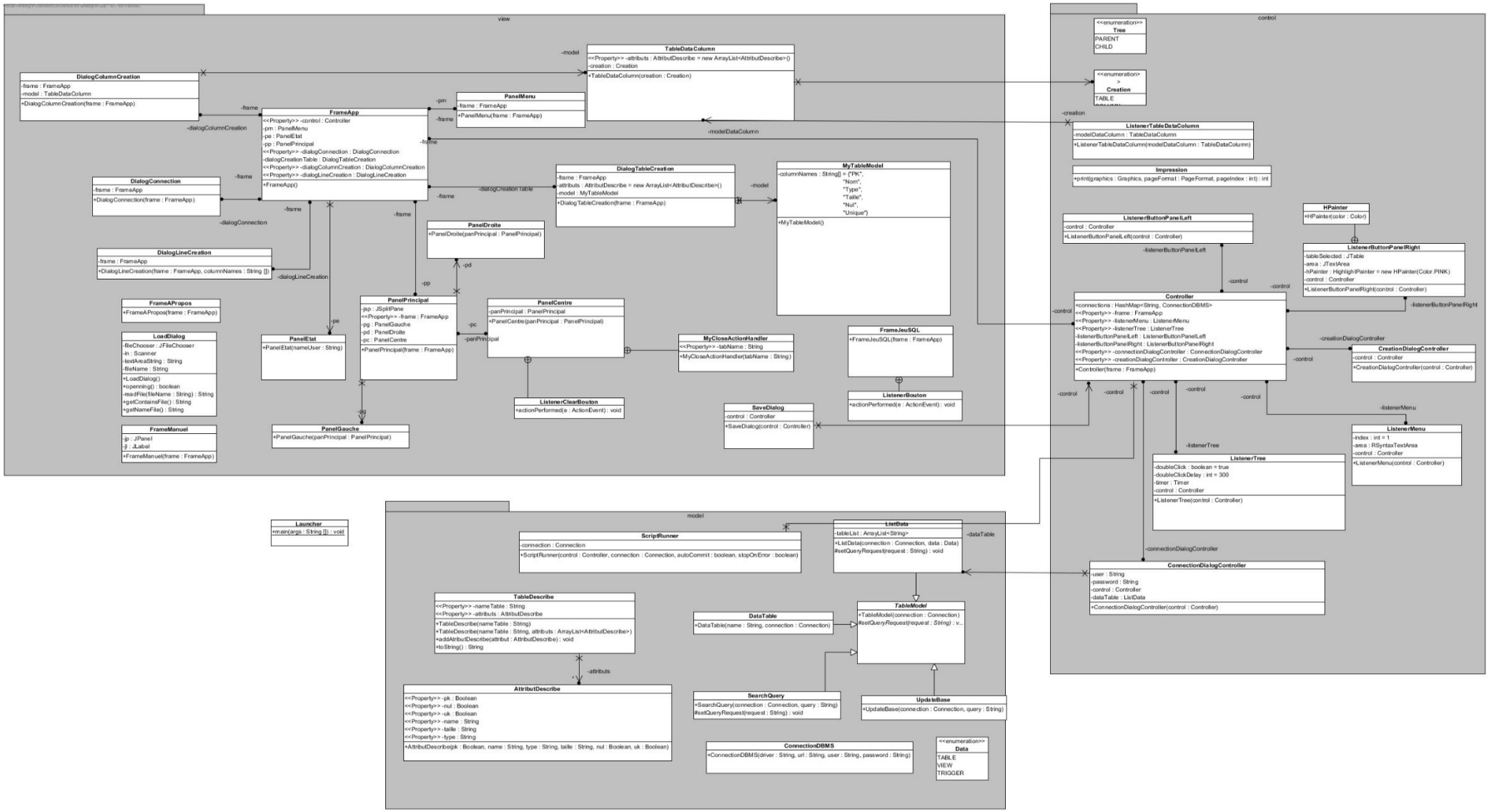
Dans le modèle nous avons choisis de créer une classe qui permet de gérer les propriétés d'une table, c'est-à-dire, son nom, le nombre de colonnes qu'elles possèdent avec leurs noms et leurs types, le référencement des clés primaires et des contraintes de non nullité et d'unicité.

Une autre classe du modèle est chargé de se connecter à une base de données en donnant les paramètres de connexion. Elle vérifie donc la bonne connexion, la bonne déconnexion, etc.

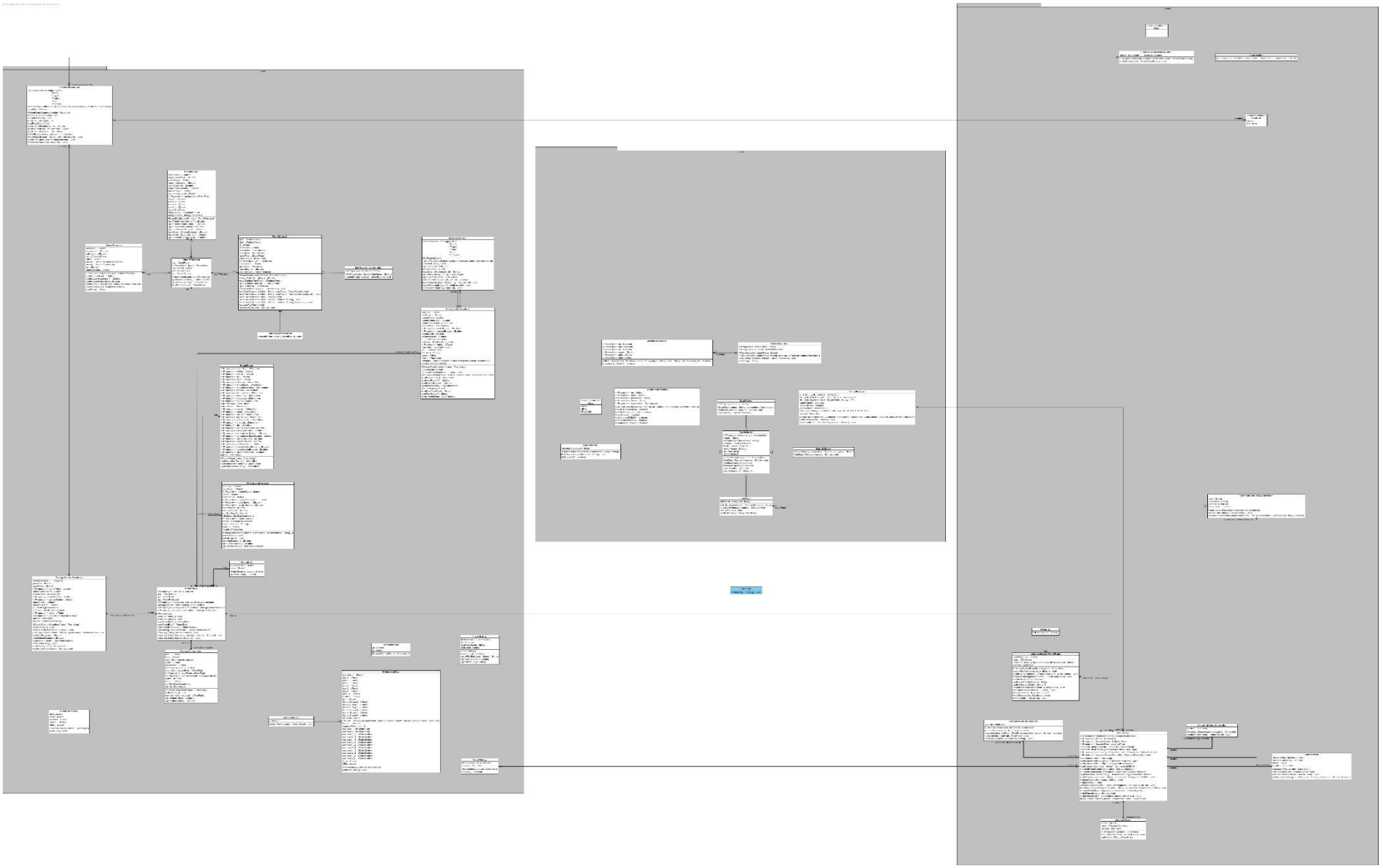
Une autre classe était chargé de récupérer le texte rentré dans la zone de texte pour le découper en plusieurs requêtes.
Nous avons choisi de créer une classe abstraite qui gère les envois de requêtes à la base de données. Cette classe possède différente classe filles qui permettent de soit mettre à jour la base de données soit aller chercher des informations dans la base de données.

Les données récupérées dans la base de données lors d'une requête sont stockés dans un *TableModel* ceux qui permet une exploitation facile des données par la suite dans la partie graphique de l'application.

# IV- Diagramme de classe obtenue par rétroconception (Simple)



# V- Diagramme de classe obtenue par rétroconception (Complet)



Suite à la création du code du projet nous avons réalisé le diagramme de classe obtenue par rétroconception. Nous pouvons remarquer qu'il respecte dans les grandes lignes le premier diagramme de classe que nous avions conçu. Aussi, le modèle MVC est respecté.

Sur ce diagramme on peut voir que certaines classes sont apparues au fil de l'élaboration du code de l'application. Comme par exemples certaines classes internes qui avaient pu nous échapper lors de l'étape de préparation ou certaines classes qui gèrent des boites de dialogues

Le diagramme de classe complet propose l'ensemble des classes réalises au cours du projet avec l'ensemble des méthodes et attributs que comporte chaque classe.

# VI- Campagne de test effectuée

Lors du projet nous effectuons des tests à chaque fin de création de méthode pour analyser si aucun problème apparaissait sur l'application. C'était une manière pour nous de ne pas être pris au dépourvu à la fin du projet. Chaque méthode testée était pourvue de sa javadoc afin de se repérer sur les méthodes fonctionnelles.
Les tests unitaires ont eu lieu seulement sur les classes appartenant au modèle afin de tester s'il n'y avait pas d'erreur sur la récupération et le traitement des données de la base de données.
Les modifications de classes après leur création sont annotées d'un numéro de version différent en fonction de l'importance des mises à jour que la classe a reçu.

# VII- Bilan personnel

## A) Alexandre

Ce projet m'a énormément apporté en termes d'organisation et planification d'un projet. Il m'a aidé à me rendre compte qu'un projet qui n'est pas préparer en amont, peut mener à des complications en cours de réalisation de ce dernier.

Concernant la partie codage je me suis réellement améliorer en IHM et donc en Java. Je me suis occupé du package *« View »* de l'application et ainsi que de certains *« Controllers »*. Mais certes nous avons procédé à une répartition des tâches pour la partie codage mais rien n'empêche que nous nous entraidions selon les besoins de chacun. De plus la découverte de JDBC a été pour moi très intéressante, même si je regrette ne pas avoir ne serait-ce qu'un cours sur ce dernier pour ne pas commencer avec un handicap, mais bien sûr ce jugement n'engage que moi. Je suis très fier du travail accompli avec Benoît, notre application est comme nous l'avions imaginé, et elle est totalement fonctionnelle après nous aurions aimé ajouter d'autres fonctionnalités mais nous manquions de techniques.

## B) Benoît

Ce projet m'as permis d'approfondir mes connaissances en programmation Java et m'a apporté des savoirs sur la gestion et l'organisation d'un projet en somme de ceux que nous avions pu voir en cours.

En réalisant le modèle de l'application j'ai dû développer les tests unitaires de chaque méthode contenue dans les classes de modèle afin de tester si le programme s'exécute correctement sans erreur et sans exception.
J'ai rencontré plusieurs problèmes sur la création du build du projet. Les erreurs rencontrées étaient principalement sur JUnit et la création de l'archive JAR. Cependant avec l'entraide d'Alexandre nous avons pu aboutir à une application fonctionnelle qui prend intégralement en compte les objectifs fixés par le cahier des charges de bases ainsi que quelques éléments optionnels. Nous nous étions accordé sur une répartition des tâches de manière à avancer efficacement. J'étais en charge de la partie « model » et de certaines parties du « control ».

# VIII- Annexe

Ci-joint l’archive complète de notre projet.