

Table des matières

[Table des matières 2](#_Toc417830486)

[1. Introduction 3](#_Toc417830487)

[2. Présentation de l’équipe 4](#_Toc417830488)

[3. Définition des classes 5](#_Toc417830489)

[Les classes du module «Recherche d’informations» 5](#_Toc417830490)

[4. Fonctions du projet 5](#_Toc417830491)

[5. Maquette de l’interface graphique 6](#_Toc417830492)

[6. Organigramme 12](#_Toc417830493)

[Module reporting 12](#_Toc417830494)

[Module connexion 13](#_Toc417830495)

[Module mise à jour 13](#_Toc417830496)

[Module récupération de données 14](#_Toc417830497)

[Module d’affichage 16](#_Toc417830498)

[7. Conclusion 20](#_Toc417830499)

# Introduction

Pour ce projet de 2ème semestre d’ING3, nous devons faire une application Java qui va nous permettre de gérer une base de données hospitalière fournie dans le sujet. L'objectif est donc la création d'une interface graphique claire et intuitive à partir de laquelle l'utilisateur peut effectuer des requêtes, en voir le résultat et, si il le souhaite, réaliser des graphes à partir de ces résultats.  
  
 Pour mener à bien ce projet nous disposons de plusieurs documents et fichiers faisant office de bases à partir desquels nous devons construire notre application.

# Présentation de l’équipe

**Clément Raibaud(chef de projet) :** Ex-Ing2, il a été désigné comme chef par ses équipiers car il a fait du Java lors de son semestre en Corée et est donc le plus expérimentée. Son but est de gérer l'équipe, donner les directives et vérifier l'avancée du projet.

**Lisa Cot :** Ex-Ing2,elle a séjourné à Laval au 1er semestre où elle a étudié le C++. C’est une personne motivée et dynamique, qui a cœur de réussir ce projet et de perfectionner sa connaissance du java.

**Jérôme Ducrocq (débutant en Java):** Ex-Ing2, il a passé le 1er semestre en Suède, où il n'a pas étudié de langage objets comme le C++ ou le Java. Il est le moins expérimenté du groupe mais très curieux et motivé d'apprendre ce langage qui est indispensable pour la majeure SI qu'il souhaite suivre.

**Florian Verberke :** ex-Ing2, il était au premier semestre en Corée où il a fait ses premiers projets en C++ POO.

Nous sommes une équipe qui a déjà travaillé ensemble sur d'autres projets et savons que chacun s'impliquera pour mener ce projet jusqu'au bout.

# Définition des classes

### Les classes du module «Recherche d’informations»

Pour stocker les informations recueillies par ce module, nous avons décidé d’utiliser un tableau de classe**(s)** **générique(s).** Nous aurons donc une classe générique et, par héritage, plusieurs classes filles:

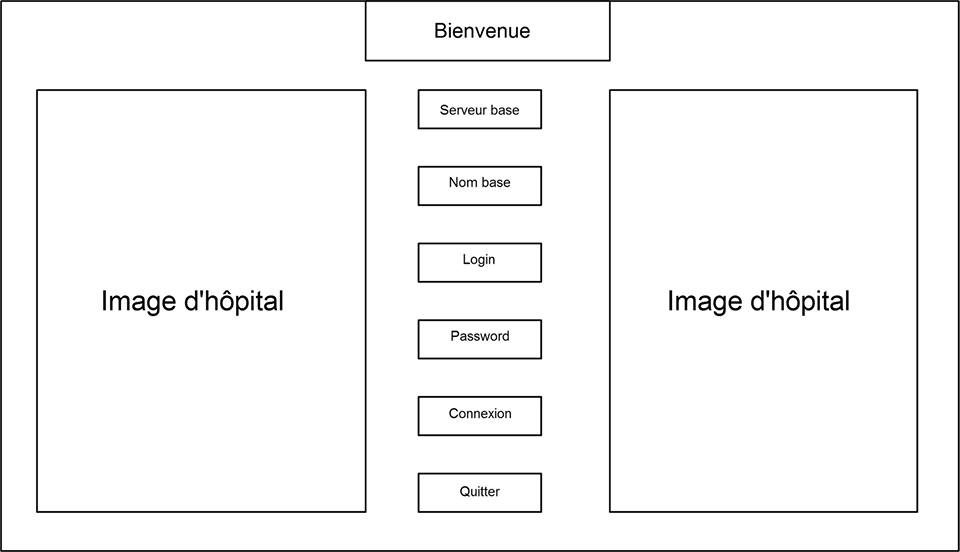
* La classe 'générique': la classe mère permettant de faire un tableau de classe**(s)** pour stocker les résultats de chaque requête.
* La classe 'personne': une classe fille de la classe générique mais qui sera aussi la classe mère de plusieurs autres classes. Elle contient des attributs correspondant aux champs communs à toutes les personnes stockées dans la base de données.
* La classe 'malade': une classe fille de personne, contient les attributs correspondant aux champs de l’entité malade.
* La classe 'docteur': une classe fille de personne, contient les attributs correspondant aux champs de l’entité docteur.
* La classe 'infirmier': une classe fille de personne, contient les attributs correspondant aux champs de l’entité infirmier.
* La classe 'autre': une classe fille de personne, contient les attributs correspondant aux champs de l’entité autre.
* La classe 'service': une classe fille de générique, contient les attributs correspondant aux champs de l’entité service.
* La classe 'chambre': une classe fille de générique, contient les attributs correspondant aux champs de l’entité chambre.

#### Diagramme de classeC:\Users\Clément\AppData\Local\Microsoft\Windows\INetCache\Content.Word\DiagrammeDeClasses.png

# Maquette de l’interface graphique

Pour la partie graphique, nous avons décidés de faire une classe par page à afficher qui vont tous hériter d’une classe plus globale ( la classe fenêtre ) qui rassemblera toutes les fonctions communes de la partie graphique.

Classe FenetreConnexion hérité de fenêtre



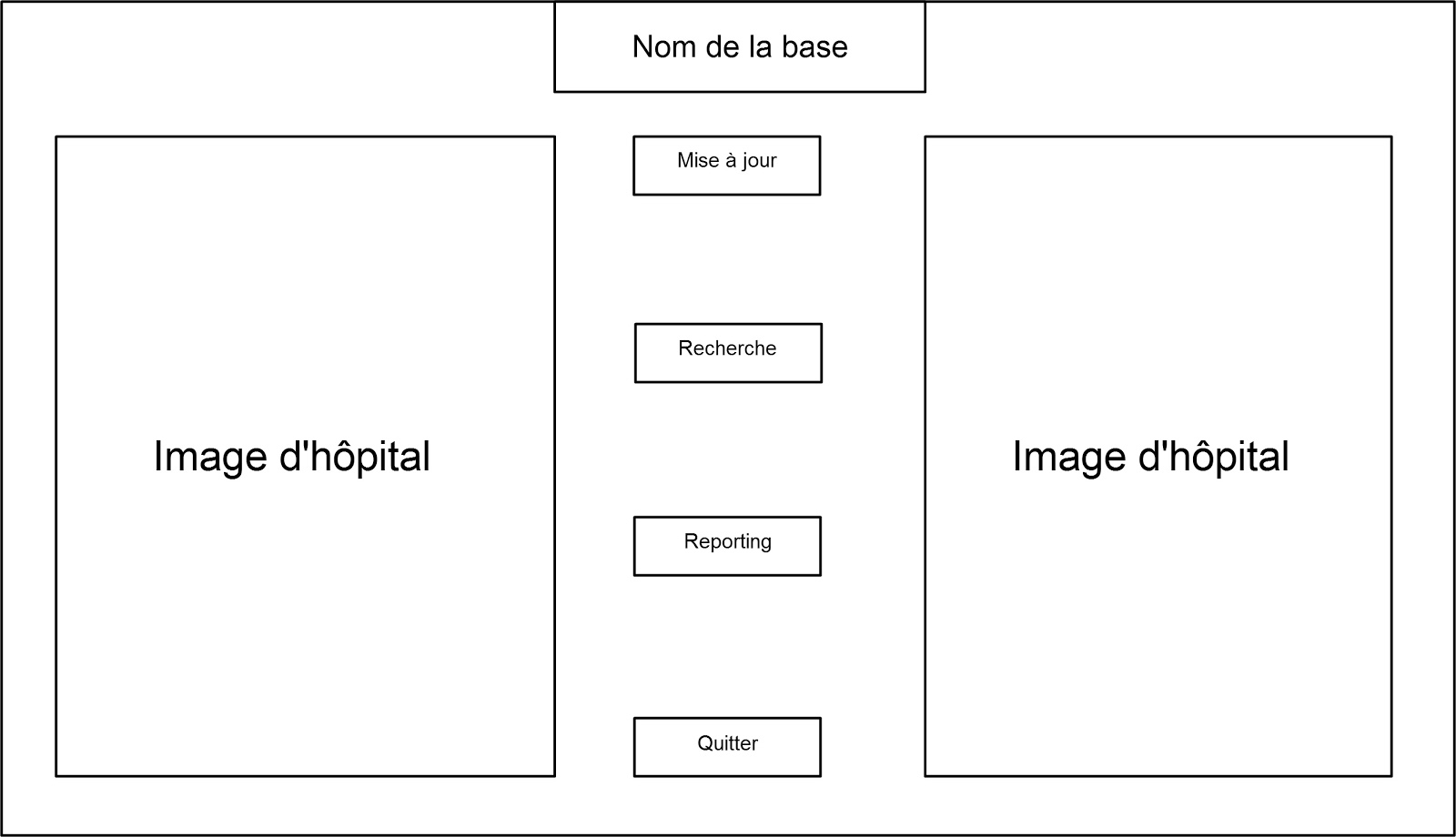
Ceci est la maquette du menu de connexion pour notre projet. On devra remplir les 4 informations ( à savoir l’adresse du serveur de la base, le nom de la base, le login et enfin le mot de passe ) dans les cases prévues à cet effet et appuyer sur le bouton de connexion.

Deux méthodes sont à prévoir pour ce menu en particulier :

GererErreurConnexion : Cette méthode devra renvoyer un message d’erreur à l’utilisateur si on arrive pas à se connecter ( soit parce que les informations sont erronées ou si une liaison internet n’est pas assurée )

EtoileMdp : Lors de la saisie du mot de passe, il serait commode que l’on affiche pas celui-ci en clair dans la case “Password” mais qu’on remplace tous les caractères affichés par ‘ \* ’

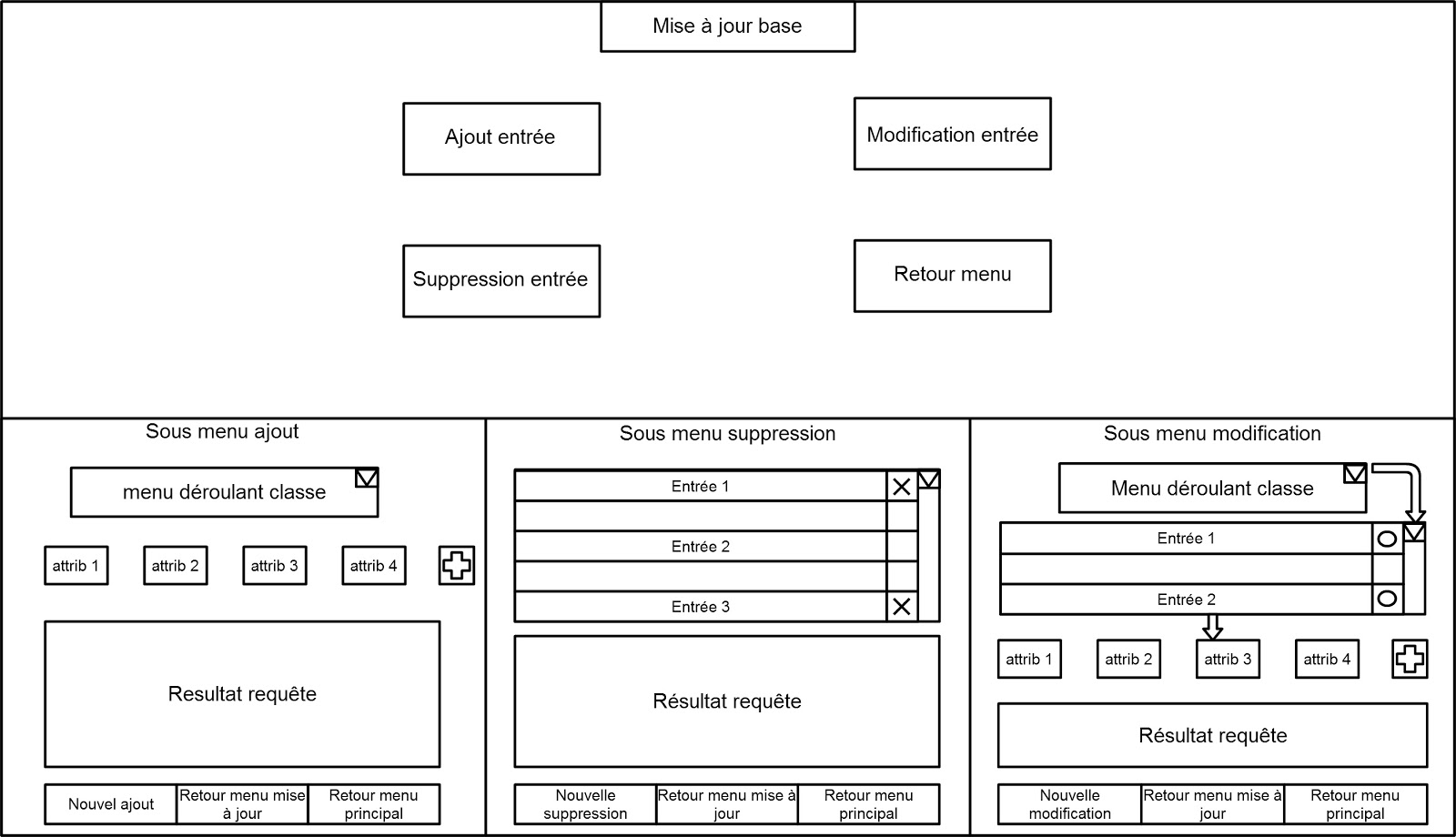
Classe MenuPrincipal hérité de fenêtre



Un menu principal assez classique qui permettra d'accéder soit à la mise à jour de la base ( c’est à dire la suppression, l’ajout ou bien la modification d’un élément dans la base de donnée ) ou bien la recherche d’informations via des requêtes prédéfinies ou bien encore la visualisation de diagrammes plus parlant pour le personnel médical.

Il est aussi possible que l’on rajoute un bouton pour revenir à la fenêtre de connexion au lieu de quitter l’application si on veut changer de base.

Classe MajBase



Comme on peut s’y attendre, la page de mise à jour de la base contient 3 parties distincts:

Ajout entrée

L’utilisateur devra d’abord choisir via un menu déroulant la classe de la base où il souhaite faire un ajout puis il doit remplir les attributs proposés par l’application (comme la nom et prénom d’un patient par exemple ). Il valide enfin son choix et l’application affiche en résultat l’ensemble de la classe pour montrer que l’on a bien rajouter un élément.

Suppression entrée

Comme pour l’entrée, l’utilisateur choisi qu’elle est la classe qui va être affectée, puis choisir via un tableau “déroulant” les entrées qu’il souhaite supprimer en cochant la case associée. Ensuite, comme pour l’ajout, on montre que la demande du client a été pris en compte en ré affichant tous les éléments de la classe.

Modification entrée

Le début est similaire aux deux autres options : on choisit la classe à modifier. Ensuite on choisit une entrée et seulement une ( symbolisée par le rond au lieu de la croix ! ). Ensuite on fait la même démarche que pour l’ajout d’une entrée.

Classe RechercheBase

La page de recherche permettra de trouver des renseignement que le personnel médical d’un hôpital pourrait requérir sur leurs patients, les services et les docteurs associés pour un traitement plus efficace et plus rapide.

On propose une première approche qui sera implémentée ainsi qu’une deuxième qui restera en option et sera implémentée si le commanditaire du projet jugera de la pertinence de cette fonction.

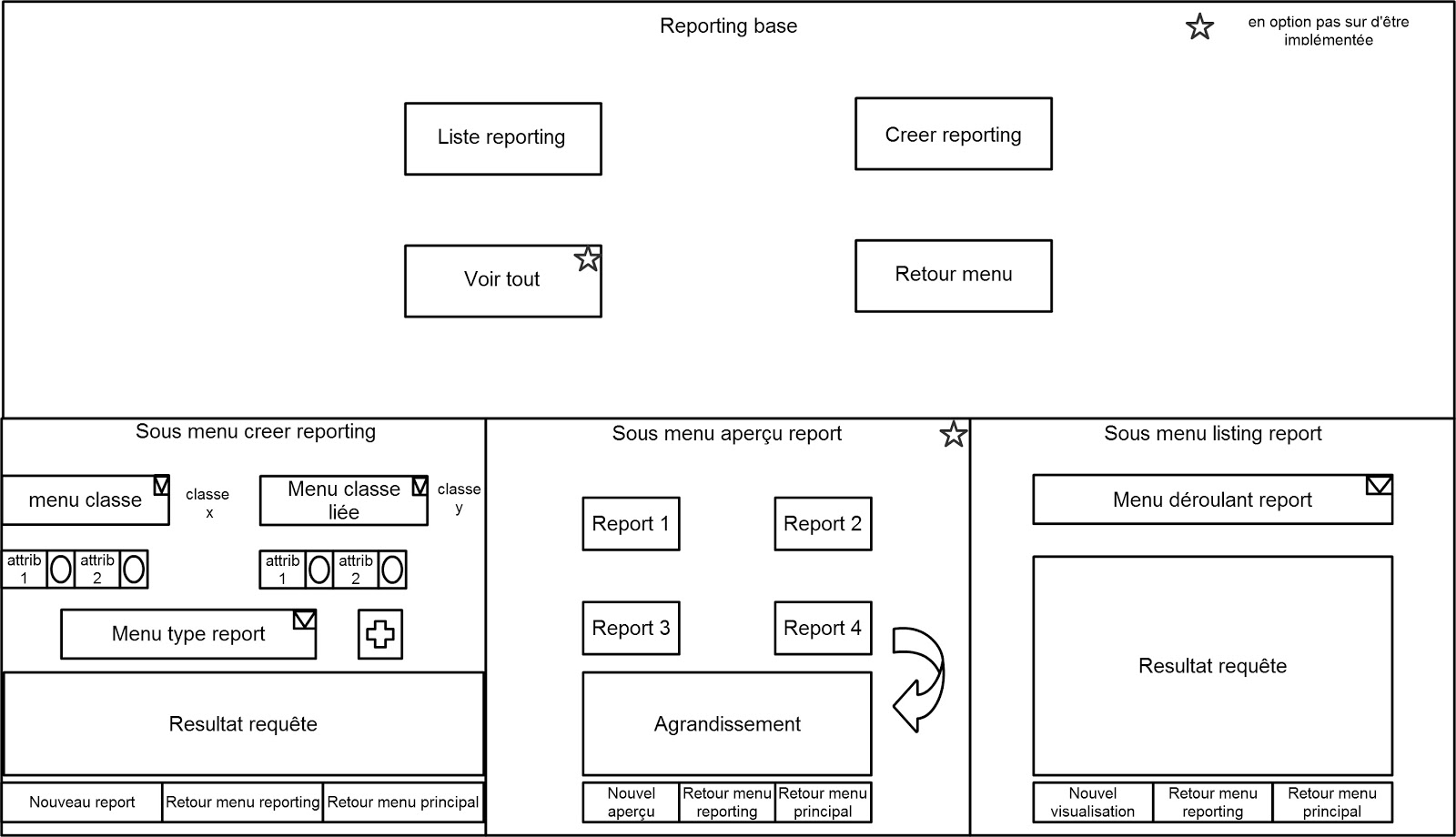
Recherche par requête prédéfinie

Fourni avec le logiciel, plusieurs requêtes seront à votre disposition dans un menu déroulant ( tel que  le prénom et le nom des docteurs ayant au moins un malade hospitalisé. ). Il vous suffira de valider la requête choisie et le résultat s’affichera.

Recherche par ordre croissant/décroissant et ordre alphabétique

Parfois il est important de connaître en un clin d’oeil quel service accueille le plus de malade et inversement afin de prendre des mesures augmentant l’efficacité de tous les services. La recherche de tri vous propose d’abord de choisir la classe puis l’attribut qui vous intéresse dans la base de donnée ainsi que le type de tri souhaité. Il vous suffira de valider votre demande et de voir le résultat s’afficher sous vos yeux.

Classe BaseReporting



Il arrive que la fonction de recherche classique ne soit pas assez percutante visuellement pour se rendre compte d’un phénomène. C’est ici qu’intervient le menu de “reporting”.

Créer reporting

L’utilisateur choisit tout d’abord un attribut d’une première classe ( x ) qui servira d’abscisse ou de total ( comme par exemple le nombre de patients ) puis on désigne une deuxième classe ( y  ) avec attribut ( comme par exemple type de service ) et on valide le type de “report” comme par exemple un camembert ou un graphique. On valide le tout et l’application nous ressort le résultat.

Aperçu reporting ( en option )

L’idée ici est de proposer des miniatures des camemberts et autres graphique sous forme d’aperçu et l’utilisateur peut en sélectionner un pour le voir en taille réelle.

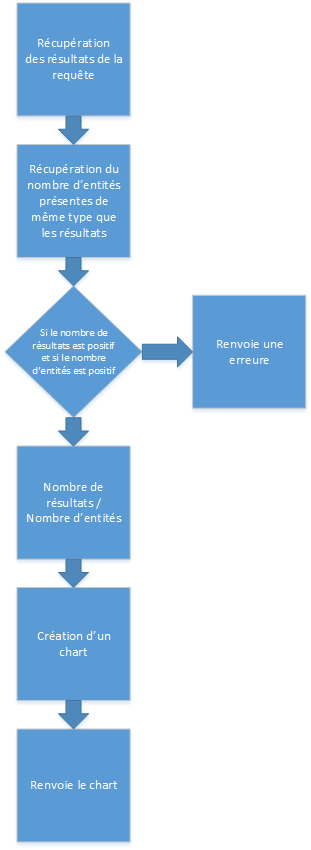
Listing reporting

Si l’utilisateur sait déjà quel graphique l'intéresse et qu’il est déjà prévu via les requêtes de la base de données, il peut choisir ce menu et accéder à un menu déroulant pour sélectionner le bon graphique à afficher.

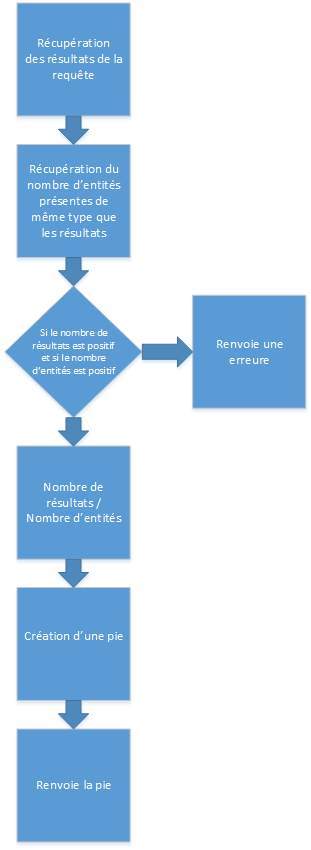
# Organigramme

### Module reporting

Graph chart



Graph pie



### Module connexion

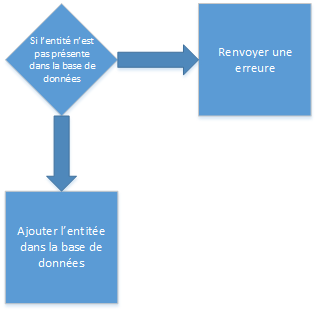
Connexion bdd ece

Connexion bdd local

Ces fonctions seront inspirées du TP3.

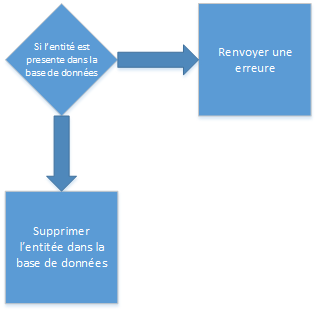
### Module mise à jour

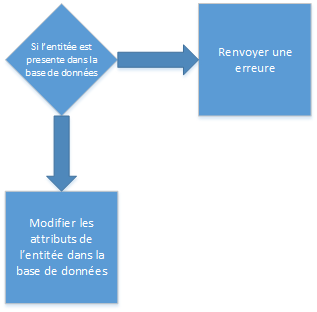
Ajouter



Supprimer

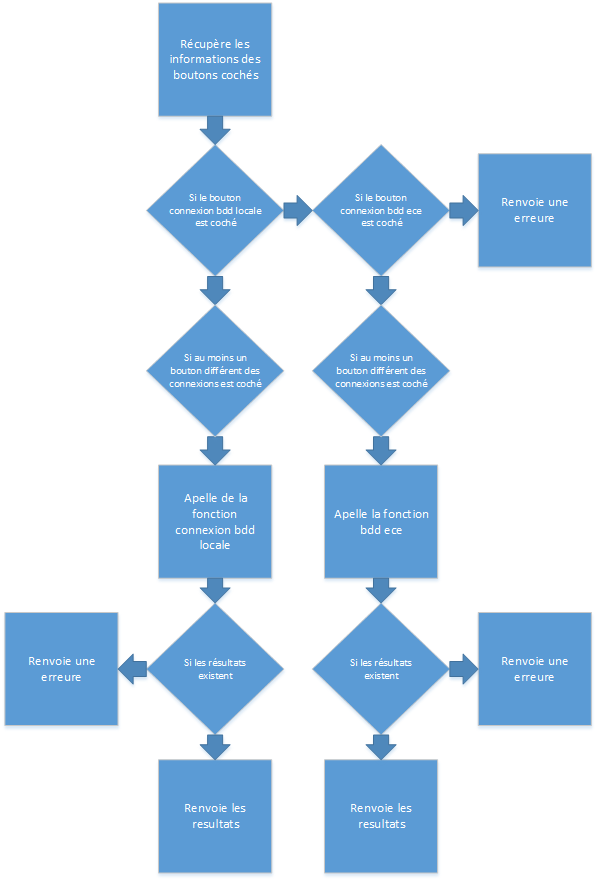
Modifier



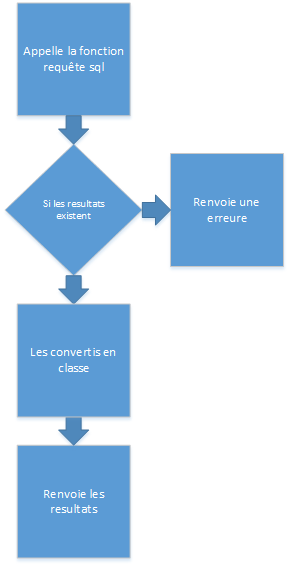


### Module récupération de données

Requête sql

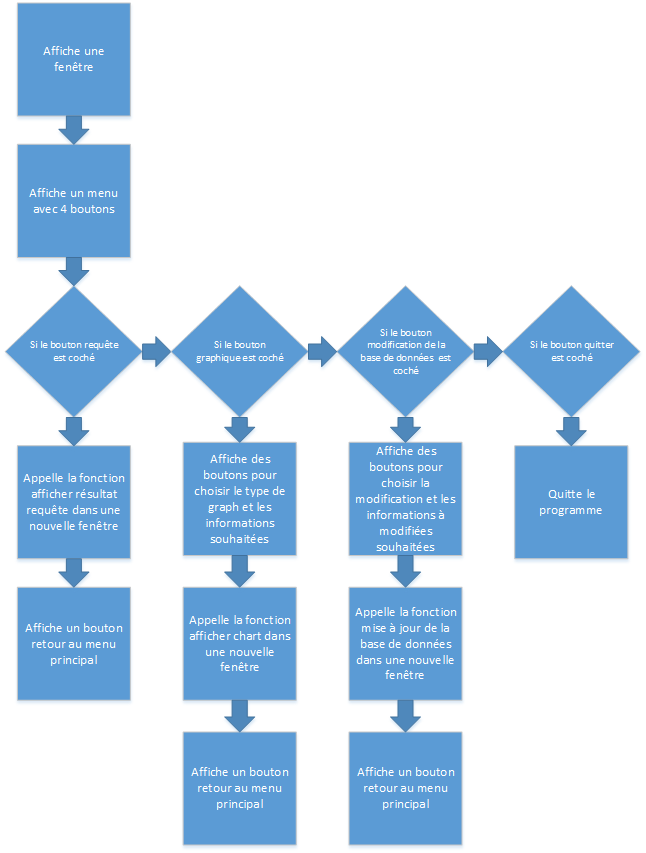


Transformation DAO

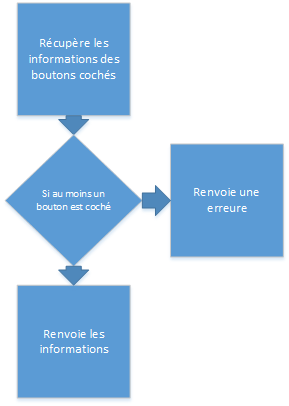


### Module d’affichage

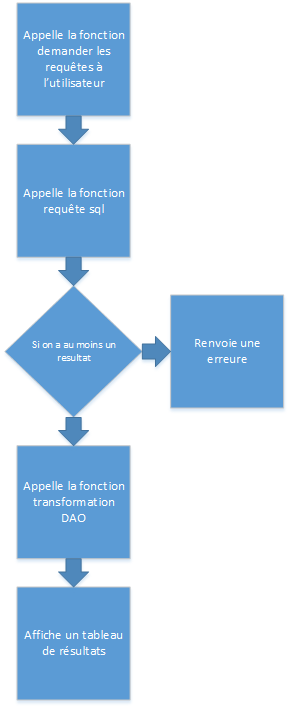
Afficher interface



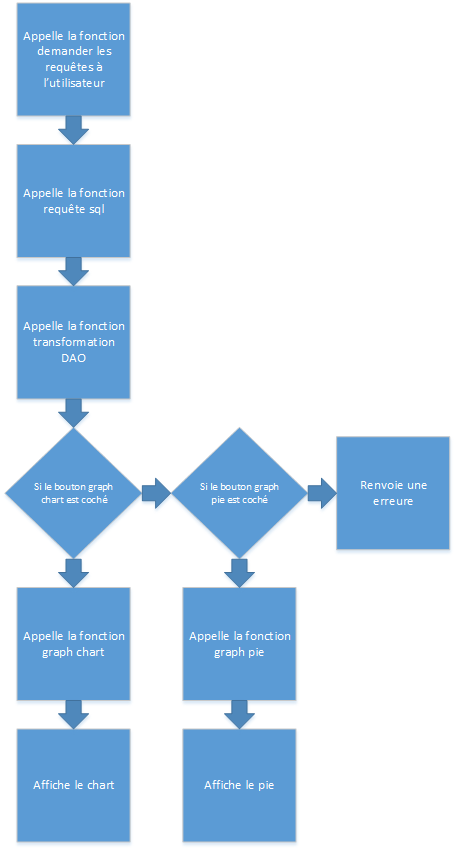
Demander les requêtes à l’utilisateur



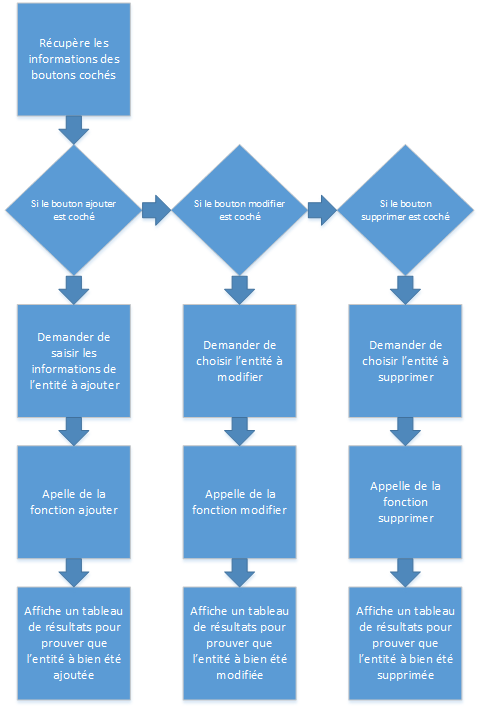
Afficher le résultat des requêtes



Afficher les charts



Mise à jour des données



# Conclusion

Ce projet est réellement intéressant pour nous pour plusieurs raisons.

La première, c'est que cela nous permet d'appréhender la conception et la réalisation d'une application en Java. Ensuite, nous souhaitons tous faire en ING4 une majeure où le Java sera fortement utilisé (SI,SE et Transports) et donc cela nous apporte des connaissances que nous devrons réutiliser.  
  
 De plus, ce projet fait interagir plusieurs modules ensemble: une base de données, une interface graphique et les connexions les reliant. Nous travaillons donc sur plusieurs aspects, ce qui permet d'éviter une certaine monotonie dans la phase de conception et de développement.  
  
 Nous avons beaucoup échangé nos points de vue respectifs quant à la façon dont nous devions aborder ce projet, la vision que chacun avait du problème et de sa réalisation. Nous avons raisonné logiquement, en posant des questions à notre chargé de TP pour nous assurer que nous partions sur les bonnes bases.  
  
 Nous sommes donc arrivés à une vision précise de ce que nous souhaiterions réaliser. Cependant, il faut prendre en compte d'éventuels problèmes qui, comme dans tout projet, risquent de survenir tôt ou tard. Nous serions alors amener à emprunter un chemin différent de celui initialement imaginé afin de continuer.