

Gestion des données d’une entreprise d’électronique

Bloc Programmation Orientée Objet

Projet

Livrable

SOMMAIRE

[Cadrage du projet](#_Toc57930280)

[Contexte](#_Toc57930281)

[Objectif](#_Toc57930282)

[Contraintes](#_Toc57930283)

[Acteurs](#_Toc57930284)

[Planning prévisionnel](#_Toc57930285)

[Modélisation](#_Toc57930286)

[Modélisation de l’application : Diagrammes UML](#_Toc57930287)

[Diagramme cas d’utilisation](#_Toc57930288)

[Diagramme d’activité](#_Toc57930289)

[Diagramme de classes](#_Toc57930290)

[Diagrammes de séquence](#_Toc57930291)

[Modélisation des données : Analyse SI](#_Toc57930292)

[Dictionnaire de définition des données](#_Toc57930293)

[Matrice de dépendance](#_Toc57930294)

[MCD](#_Toc57930295)

[MLD](#_Toc57930296)

[Réalisation du projet](#_Toc57930297)

[Définition de l’architecture](#_Toc57930298)

[Documentation utilisateur](#_Toc57930299)

[Connexion](#_Toc57930300)

[Menu principal](#_Toc57930301)

[Gestion du personnel](#_Toc57930302)

[Gestion des clients](#_Toc57930303)

[Gestion des commandes](#_Toc57930304)

[Gestion du stock](#_Toc57930305)

[Gestion des statistiques](#_Toc57930306)

[Clôture du projet](#_Toc57930307)

[Planning réel](#_Toc57930308)

[Conclusion](#_Toc57930309)

[Bilan](#_Toc57930310)

[Annexes](#_Toc57930311)

[Outils](#_Toc57930312)

[Organisation : Daily Meeting](#_Toc57930313)

Cadrage du projet

Contexte

Une entreprise souhaite un logiciel permettant de vendre en ligne des composants électroniques. Il doit permettre la gestion du personnel, la gestion des clients, la gestion des commandes, la gestion du stock, la gestion des statistiques.

Objectif

**Informatique de gestion** : L’objectif est de réaliser une architecture de type client-serveur composée d’une application et d’une base de données.

Contraintes

* **Contraintes concernant l’environnement de travail**

→ Utilisation de l’IDE « Visual Studio »

→Atelier de Génie Logiciel : Star UML utilisé

→ Analyse SI : Looping utilisé

→ Langage C++

→Système de gestion de Base De Données : SQL Server

→ Intégration continue : GitHub (<https://github.com/Juggernes/ProjetPOO.git>)

- **Utilisation des formes normales** : 3ème forme normale

- **Architecture logicielle** : Services, Composants, IHM

- **Modélisation des données**

→ Dictionnaires des données

→ MCD

→ MLD

- **Modélisation de l’application**

→ Diagramme de classes

→ Diagramme de cas d’utilisation

→ Diagramme de séquences

* + - *Créer un personnel*
    - *Modifier un personnel*
    - *Supprimer un personnel*
    - *Afficher un personnel*

→ Diagramme d’activité (Gestion d’un personnel)

- **Utilisation des pointeurs de fonctions pour la gestion des simulations**

- **Gestion de la mémoire pour les objets déréférencés**

- **Utilisation de classes de mappages pour fournir les requêtes SQL**

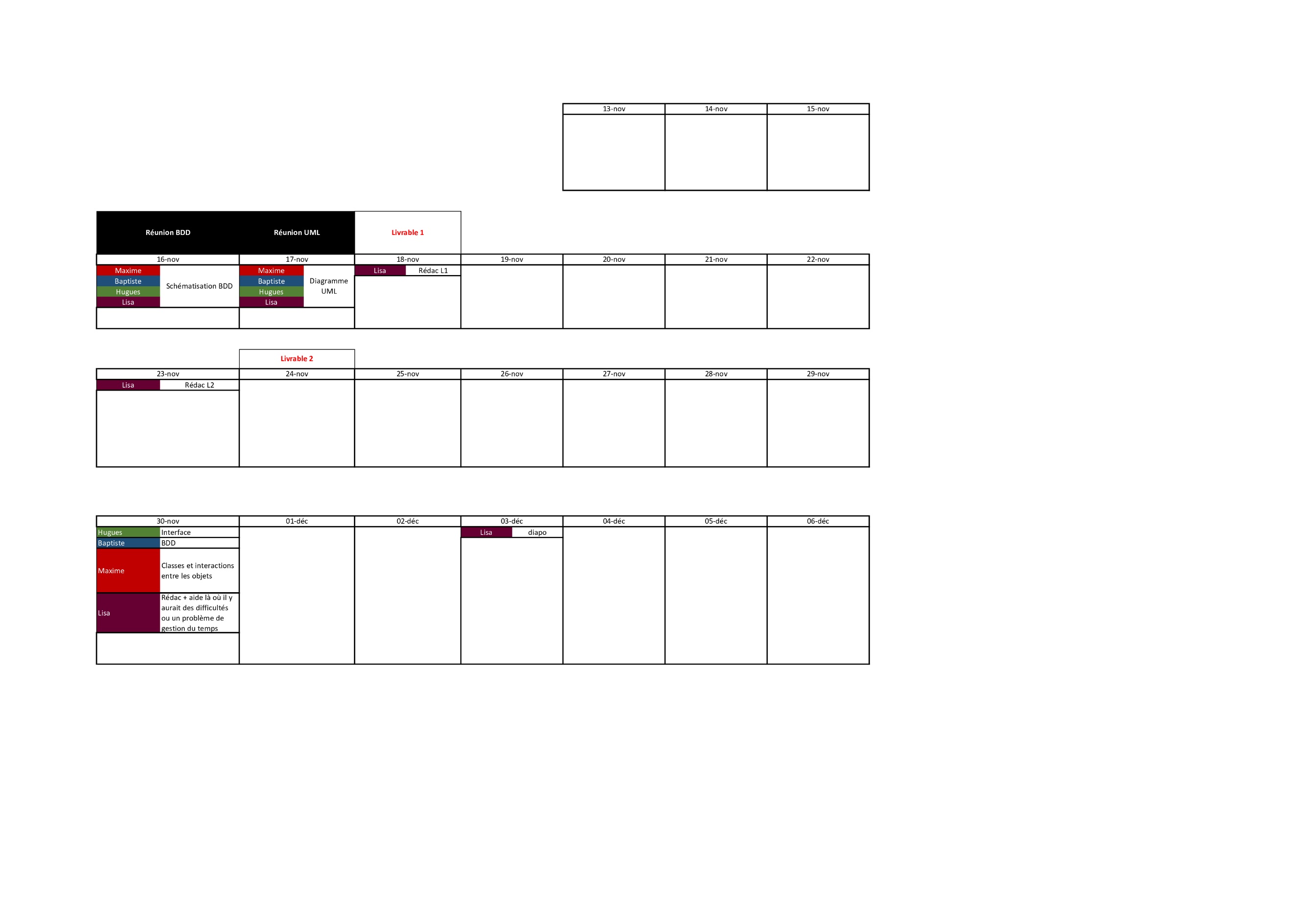
- **Tests Unitaires** : sur la séquence (Créer un personnel)

Acteurs

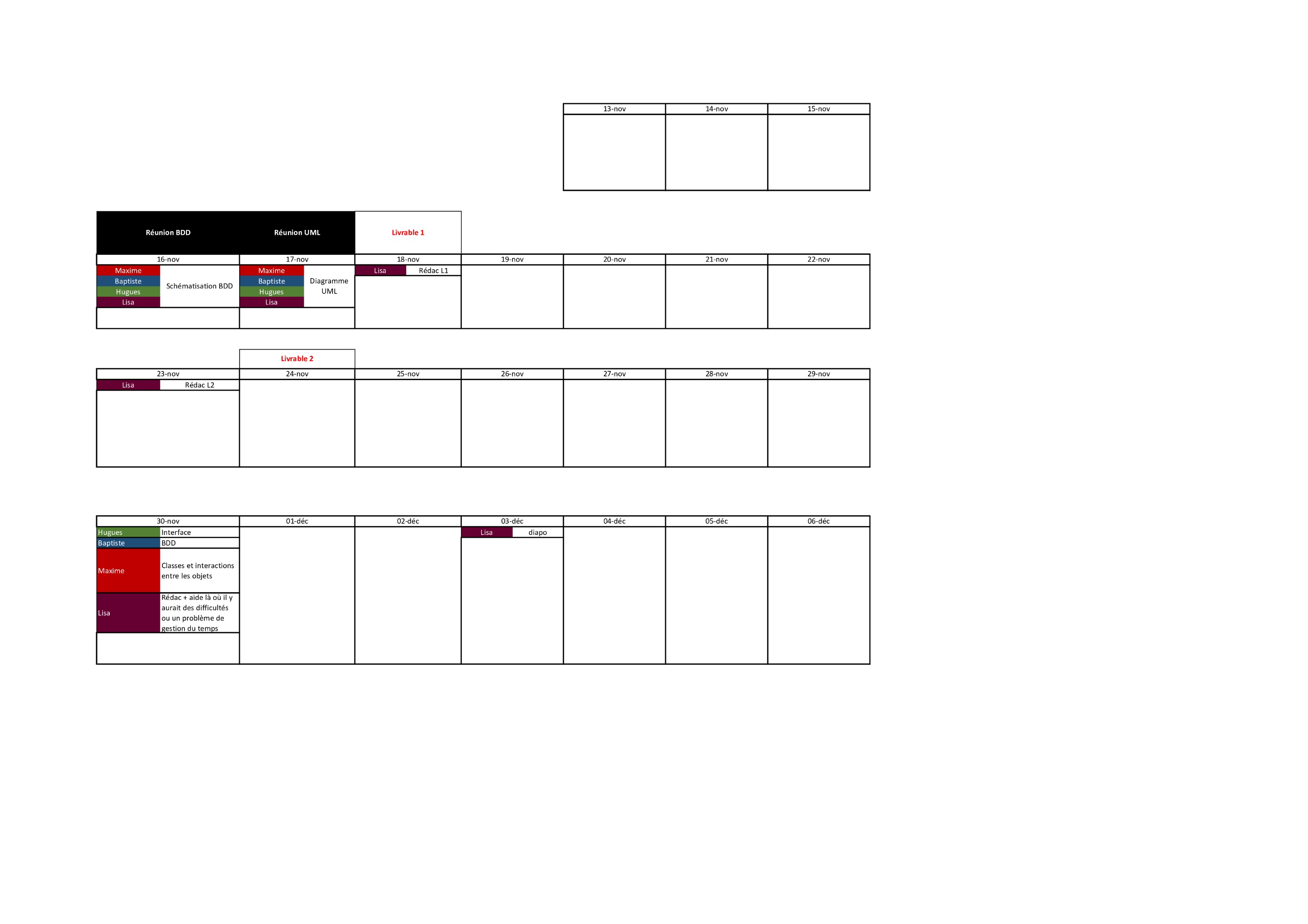
Ce projet a été réalisé par :

* CROS Hugues
* DEREMY Baptiste
* MOIROUD Maxime
* PAPY Lisa

Planning prévisionnel



1. Planning prévisionnel partie ½



1. Planning prévisionnel 2/2

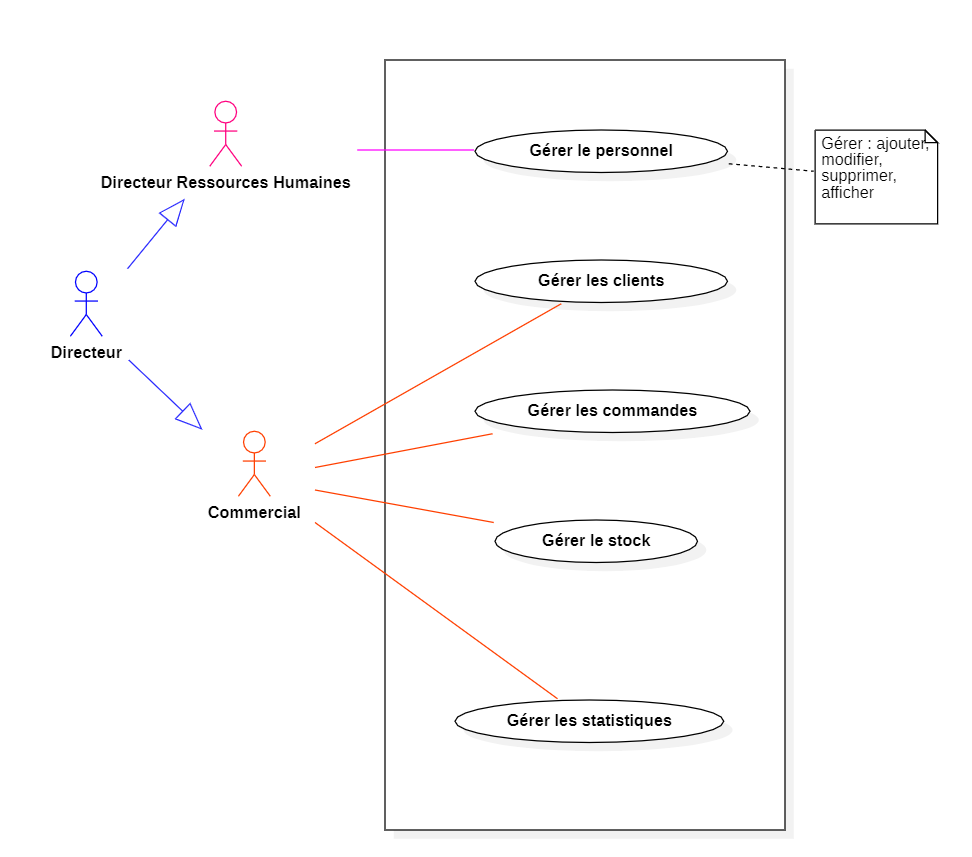
Nous avons choisi de nous répartir le travail en fonction des différentes parties de l’architecture.

Modélisation

Modélisation de l’application : Diagrammes UML

Diagramme cas d’utilisation

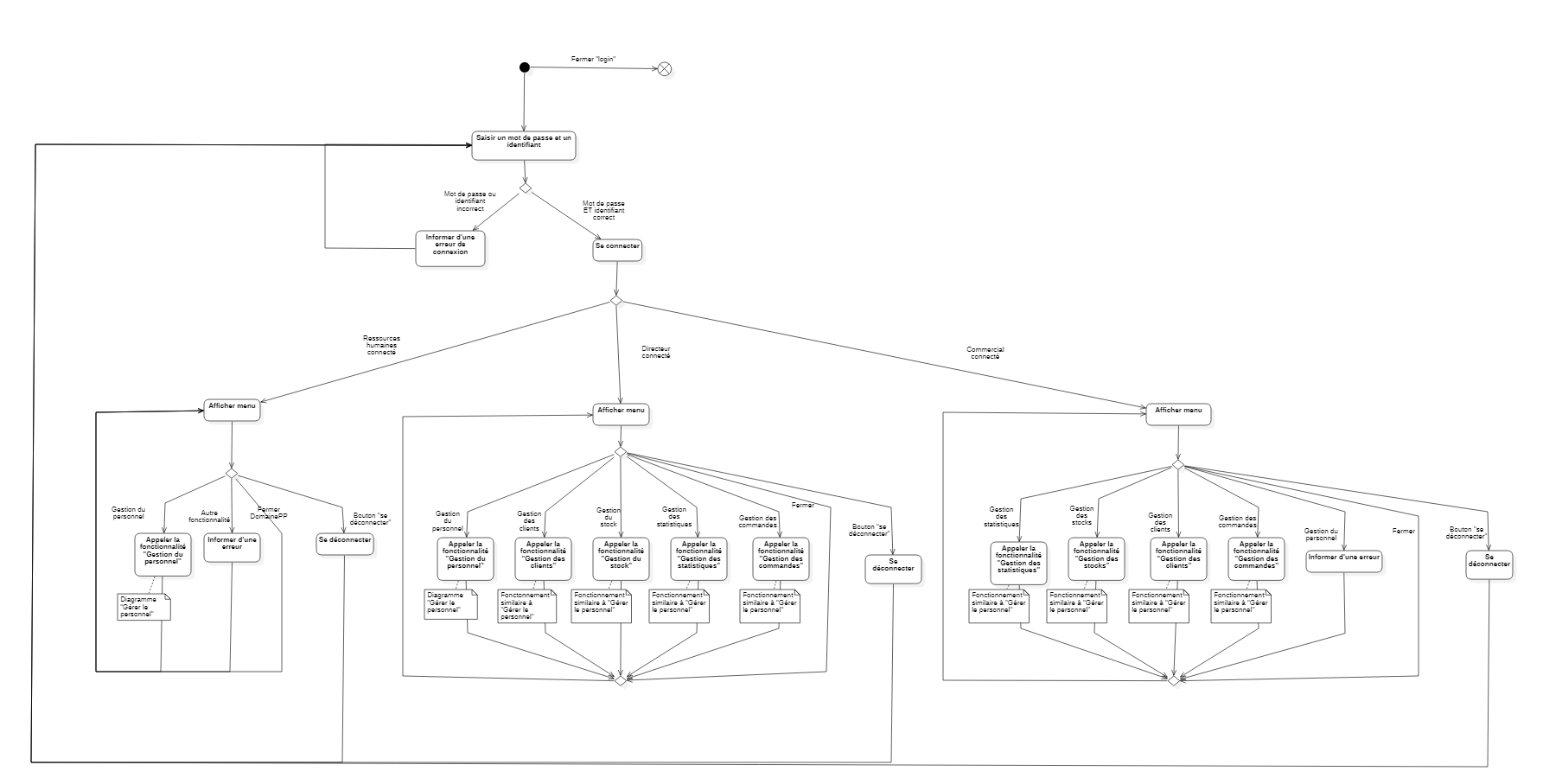
Le diagramme des cas d’utilisation (Use Case Diagram) permet de représenter les services offerts par le système. Chaque cas d’utilisation regroupe un ensemble de fonctionnalités. Il faut identifier un nombre restreint de cas d'utilisation directement accessibles par les acteurs.



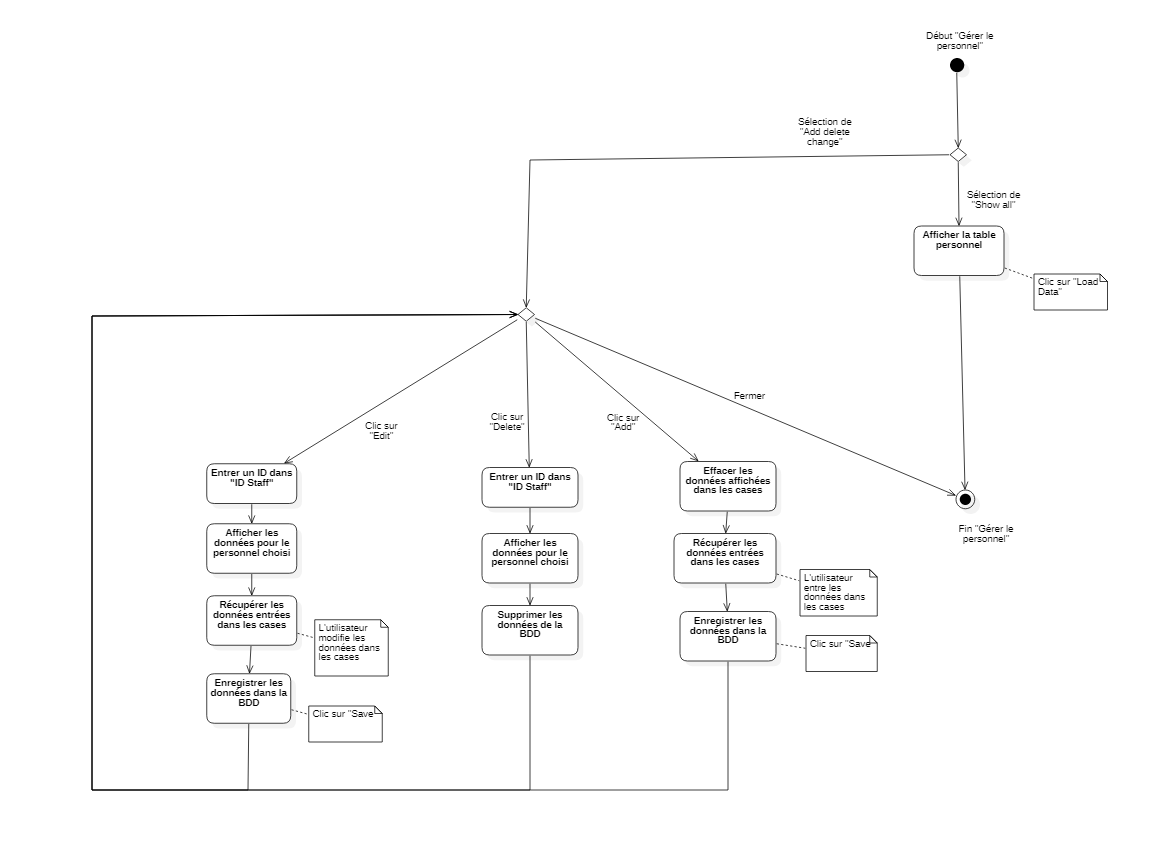
1. Diagramme cas d’utilisation

Diagramme d’activité

Le diagramme d’activité raffine les cas d’utilisation en précisant les activités/actions pour le réaliser. Il modélise les flux d’informations et d’activité.



1. Diagramme d’activité général

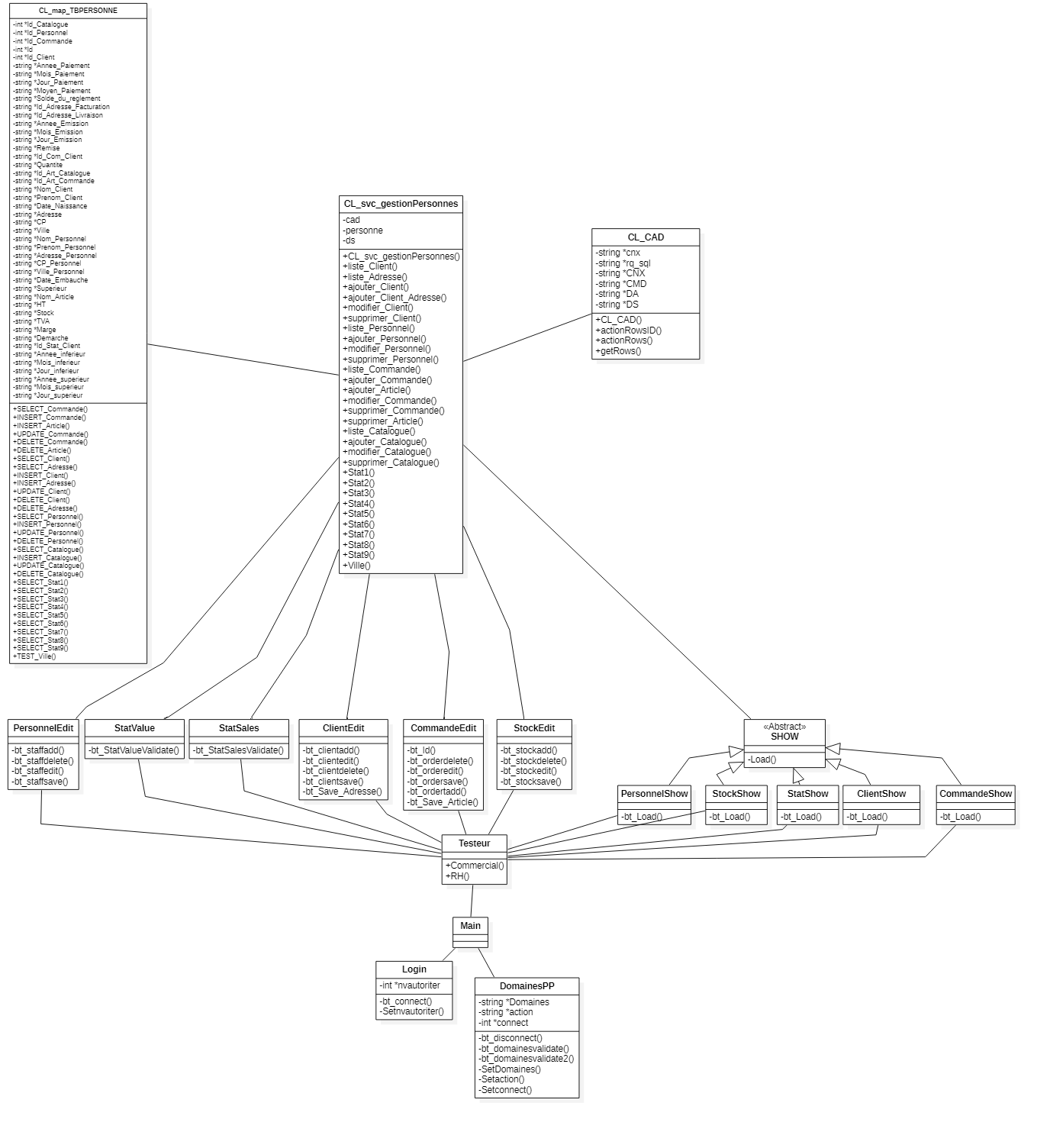


1. Diagramme d’activité « Gérer le personnel »

Diagramme de classes

Le diagramme de classes (Class Diagram) est considéré comme le plus important de la modélisation orientée objet. Il montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système. Le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Nous avons décidé de ne pas représenter l’initialisation des éléments de l’interface tels que les « boutons » ou les « text box » par exemple. Les méthodes inscrites correspondent alors aux clics des boutons malgré que cela n’a pas été spécifié sur le diagramme de classe représenté ci-deesous.



1. Diagramme de classe

Diagrammes de séquence

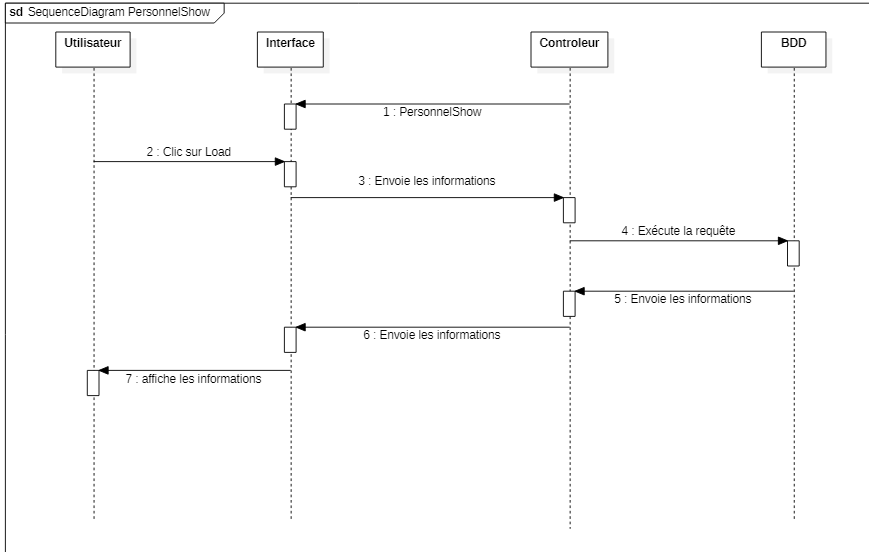
Les diagrammes de séquence (Sequence Diagram) permettent de modéliser dans un premier temps les scénarios opérationnels, puis de décrire le comportement (interactions) entre les objets à tous les niveaux de décomposition du modèle, suivant le besoin. Plus simplement, il modélise la chronologie des interactions entre le système et l’utilisateur.

Nous avons décidé de représenter quatre diagrammes de séquence concernant la gestion du personnel. Ces diagrammes représentent le fonctionnement général de toutes les fonctionnalités du logiciel. En effet, toutes les fonctionnalités se ressemblent car elles permettent toutes de réaliser quatre actions différentes : afficher, ajouter, modifier, supprimer. Seule la fonctionnalité « Gestion des statistiques » ne suit pas ce principe. Ainsi, nous avons représenté quatre diagrammes de séquence différents pour les quatre actions possibles.

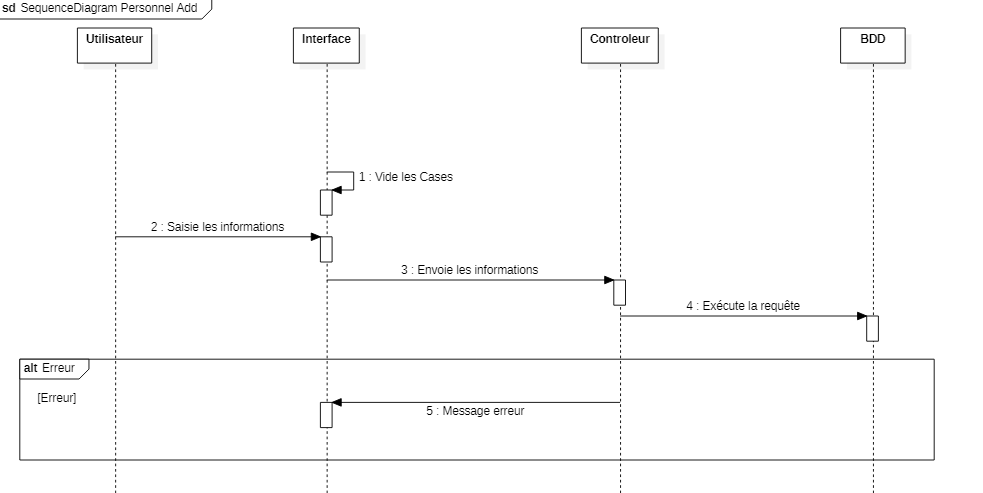
Une image contenant table

Description générée automatiquement

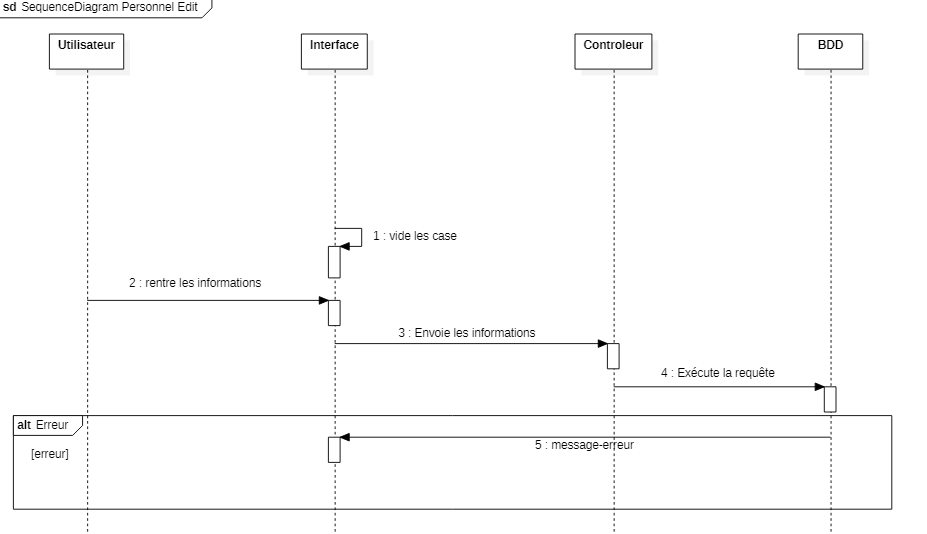
1. Diagramme de séquence : entrée dans le mode personnel

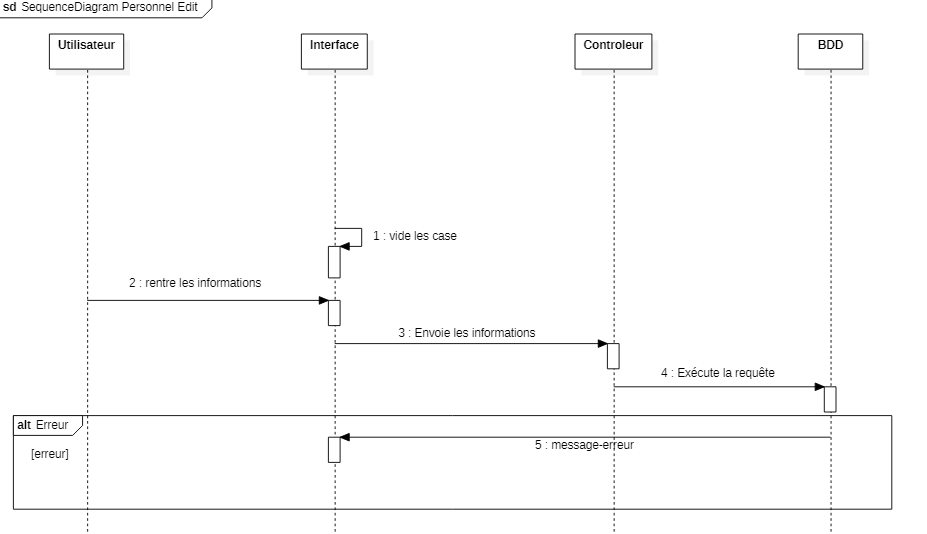


1. Diagramme de séquence : Gestion du personnel *(affichage d’un personnel)*

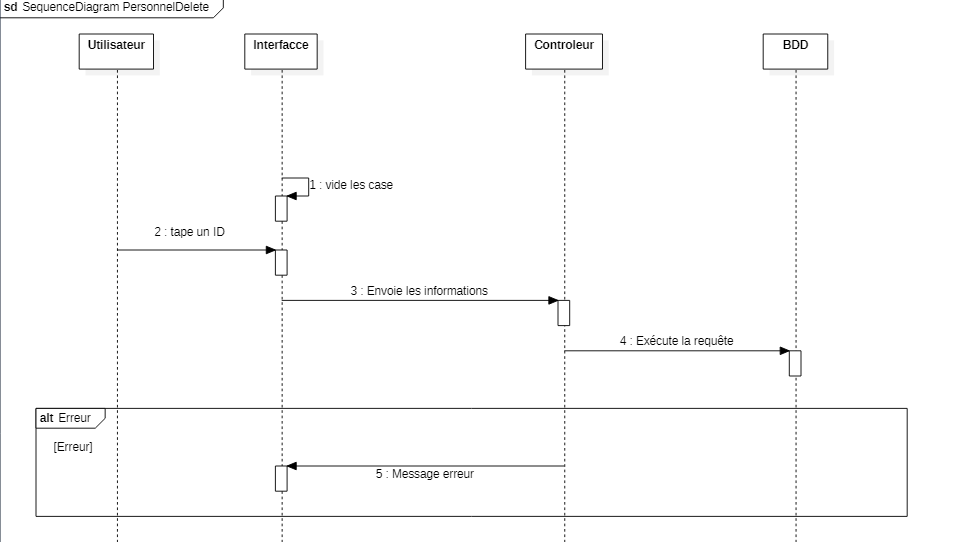


1. Diagramme de séquence : Gestion du personnel *(ajout d’un personnel)*





1. Diagramme de séquence : Gestion du personnel *(modification d’un personnel)*



1. Diagramme de séquence : Gestion du personnel *(suppression d’un personnel)*

Modélisation des données : Analyse SI

Dictionnaire de définition des données

Le dictionnaire nous permet de garantir l’unicité de la compréhension de la base de données. En effet, toutes les données doivent y être regroupées. Les caractéristiques de celles-ci y sont aussi renseignées. Nous y retrouvons alors les types de données (alphabétique, date, numérique) mais aussi la description de cette donnée par exemple.

Une image contenant table

Description générée automatiquement

1. Dictionnaire de définition des données partie 2/2

Matrice de dépendance

La matrice est utilisée afin de mettre en évidence les différents liens entre les données. Un “1” spécifie une dépendance. Cette matrice permet, par la suite, de réaliser le MCD.

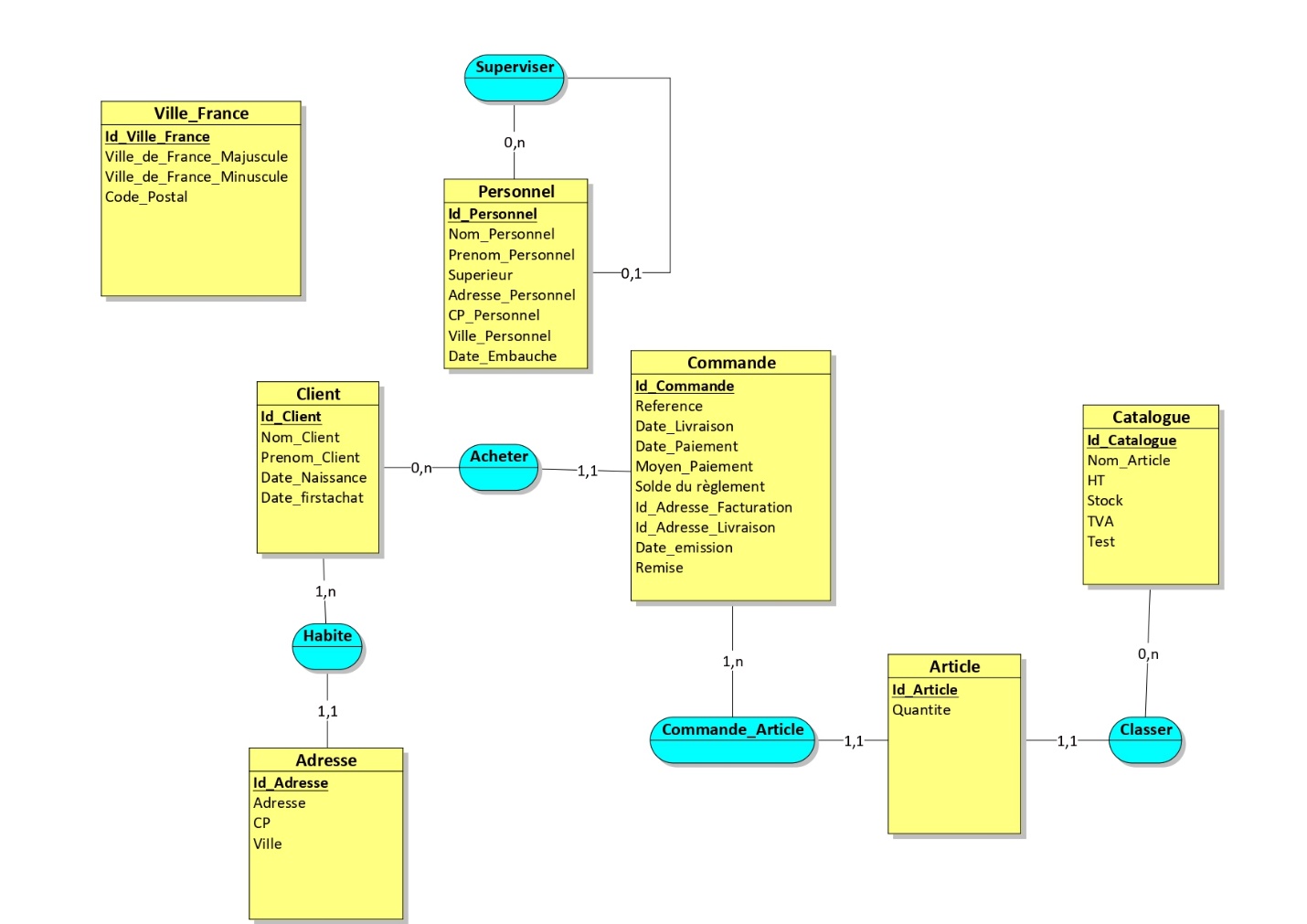
Une image contenant table

Description générée automatiquement

1. Matrice de dépendance des données

MCD

Le modèle conceptuel de données va être une sorte de premier jet schématique de notre base de données. Il explicite au client notre vision de sa future base de données.



1. Modèle Conceptuel des Données

La première forme normale est définie par l’atomicité. Tous les attributs dépendent de la clé primaire. Sinon, il faut enlever l’attribut de cette table. La deuxième forme normale est une extension de la première. La première forme normale est validée. De plus, il n’y a pas de dépendances entre un attribut dans la clé primaire et un autre élément de la table. Enfin, la troisième forme normale regroupe la première et la deuxième forme normale. De plus, il n’y a pas de dépendances entre deux attributs de la même table. Finalement, nous avons donc respecté la troisième forme normale.

MLD

Le MCD permet d’avoir un premier aperçu sur les liens entre les données cependant il ne permet pas de créer les différentes tables de notre base de données. Il est nécessaire de réaliser le MLD pour pouvoir, par la suite, réaliser l’implémentation de la base de données. Grâce au MLD nous trouvons des clés primaires et des clés étrangères. Les clés primaires, où Primary key, sont des attributs ou identifiants uniques d’une relation c’est-à-dire d’une entité ou association. Les clés étrangères, aussi appelées Foreign key, sont des attributs qui font référence à la clé primaire.

**Personnel** = (**Id\_Personnel** INT IDENTITY, **Nom\_Personnel** VARCHAR(250), **Prenom\_Personnel** VARCHAR(250), **Superieur** INT, **Adresse\_Personnel** VARCHAR(250), **CP\_Personnel** INT, **Ville\_Personnel** VARCHAR(250), **Date\_Embauche** DATE, #Id\_Personnel\_1\*);

Client = (**Id\_Client** INT IDENTITY, **Nom\_Client** VARCHAR(250), **Prenom\_Client** VARCHAR(250), **Date\_Naissance** DATE, **Date\_firstachat** DATE);

Adresse = (**Id\_Adresse** INT IDENTITY, **Adresse** VARCHAR(250), **CP** INT, **ViIle** VARCHAR(250), #Id\_Client);

Commande = (**Id\_Commande** INT IDENTITY, **Reference** VARCHAR(250), **Date\_Livraison** DATE, **Date\_Paiement** DATE, **Moyen\_Paiement** VARCHAR(250), **Solde\_du\_règlement** FLOAT, **Id\_Adresse\_Facturation** INT, **Id\_Adresse\_Livraison** INT, **Date\_emission** DATE, Remise INT, #Id\_Client);

Catalogue = (**Id\_Catalogue** INT IDENTITY, **Nom\_Article** VARCHAR(250), **HT** FLOAT, **Stock** INT, **TVA** FLOAT, **Test** INT);

Ville\_France = (**Id\_Ville\_France** INT IDENTITY, **Ville\_de\_France\_Majuscule** VARCHAR(250), **Ville\_de\_France\_Minuscule** VARCHAR(250), **Code\_Postal** INT);

Article= (**Id\_Article** INT IDENTITY, **Quantite** INT, #Id\_Catalogue, #Id\_Commande);

1. Modèle Logique des Données

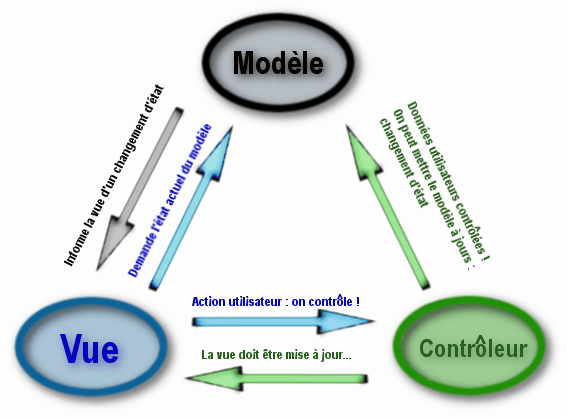
Réalisation du projet

Définition de l’architecture

L’architecture utilisée est de type **Services, Composants, IHM**. C’est une architecture de type MVC. Cela signifie que le modèle, la vue et le contrôleur sont séparés en trois parties. Cette organisation du code source sépare le code en trois fichiers dont les rôles sont distincts.

La partie **modèle** gère les donnés. Son rôle est de récupérer les informations dans la base de données. Elle les organise et les assemble ensuite afin d’être traitée par le contrôleur. C’est dans cette partie que l’on retrouve la connexion à la base de données ainsi que les requêtes SQL.

La partie **vue** se concentre sur l’affichage. Elle correspond à l’interface homme machine (IHM). Cette partie ne fait qu’afficher les informations souhaitées. C’est avec cette interface que notre client va interagir.

 La partie **contrôleur** est celle qui prend les bonnes décisions en fonction de la demande du client. C’est l’intermédiaire entre le modèle et la vue. Le contrôleur demande au modèle les données et les envoie à la vue afin de les afficher. Il gère les appels des différentes fonctions et méthodes afin de correspondre à la demande de l’utilisateur.

1. Architecture Modèle Vue Contrôleur

Documentation utilisateur

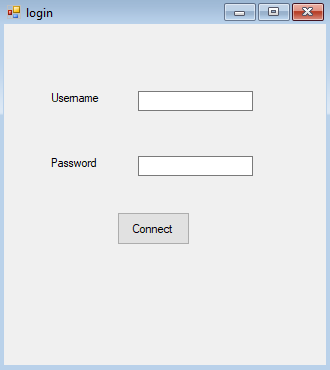
Connexion

Tout d’abord, nous avons souhaité protéger l’accès à certaines fonctionnalités. Ainsi, comme vu précédemment, il existe différents utilisateurs.

Ainsi, les Ressources Humaines (RH) peuvent accéder uniquement à la gestion du personnel. Ils peuvent se connecter l’aide de l’identifiant « RH » et du mot de passe « HR ». Cependant, les autres fonctionnalités s’affichent tout de même dans le menu principal. Dans le cas d’une sélection inaccessible, le logiciel renvoie au menu principal afin de sélectionner une autre fonctionnalité.

De plus, les commerciaux de l’entreprise peuvent accéder à toutes les fonctionnalités hormis à la gestion du personnel. Ils peuvent se connecter l’aide de l’identifiant « commercial » et du mot de passe « laicremmoc ». Comme précédemment, les fonctionnalités inaccessibles s’affichent tout de même.

Finalement, le compte administrateur permet d’accéder à toutes les fonctionnalités. C’est aussi le compte qui pourra être utilisé par le directeur. Ils peuvent se connecter l’aide de l’identifiant « admin » et du mot de passe « nimda ».



1. Page de connexion

Menu principal

Après la connexion, le menu principal s’affiche. Il s’agit d’un menu déroulant permettant de choisir les différentes fonctionnalités : gestion du personnel, des clients, des commandes, du stock ou des statistiques. Une fois le choix réalisé, il suffit de cliquer sur le bouton « Validate ».

Un second menu déroulant apparaît alors présentant les différentes actions réalisables : l’affichage ou la modification des données. Seule la gestion des statistiques présente un menu différent.

C’est aussi sur ce menu que se trouve le bouton « Disconnect ». Il permet de retourner sur la page de connexion afin de quitter ou de changer d’utilisateur.

Une image contenant texte

Description générée automatiquement

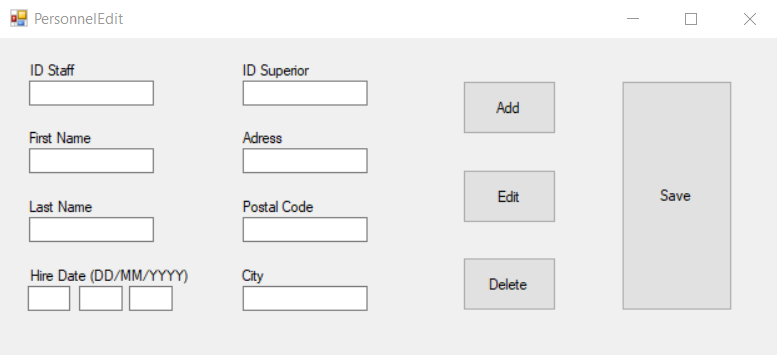
1. Page du menu principal

Gestion du personnel

Lors de la sélection de la fonctionnalité « Staff », nous pouvons choisir entre deux actions puis cliquer sur valider :

1. Si l’action choisie est « show all », toutes les données des personnels sont affichées.
2. Si l’action choisie est « add, delete, change », il est alors possible d’ajouter un nouveau membre du personnel, de le supprimer ou de le modifier :

* Pour modifier un membre du personnel, il faut cliquer sur le bouton « edit » puis saisir l’ID du personnel que l’on souhaite modifier dans « ID Staff ». Enfin, il suffit de modifier toutes les cases présentes sur la page puis cliquée sur save.
* De même, pour supprimer, il faut cliquer sur « delete » entrer l’ID de la personne que l’on souhaite supprimer. Puis sélectionner « save » afin de réaliser la suppression des données.
* Pour finir, pour ajouter un nouveau membre du personnel, il faut tout d’abord cliquer sur « Add » puis entrer les données dans toutes les cases excepté l’ID staff. Comme précédemment, il suffit de sélectionner « Save » pour enregistrer le nouveau membre.



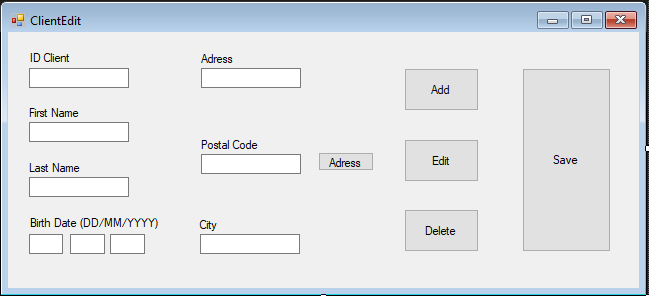
1. Page pour modifier un membre du personnel

Gestion des clients

Lors de la sélection de la fonctionnalité « Client », nous pouvons choisir entre deux actions puis cliquer sur valider :

1. Si l’action choisie est « show all », toutes les données des clients sont affichées avec leurs adresses.
2. Si l’action choisie est « add, delete, change », il est alors possible d’ajouter un nouveau client, de le supprimer ou de le modifier mais aussi de modifier les adresses :

* Pour supprimer, il faut cliquer sur « delete » entrer l’ID de la personne que l’on souhaite supprimer. Puis sélectionner « save » afin de réaliser la suppression des données. Cette commande supprime à la fois le client et ses adresses.
* Pour finir, pour ajouter un nouveau client, il faut tout d’abord cliquer sur « Add » puis entrer les données dans toutes les cases excepté l’ID Client. Comme précédemment, il suffit de sélectionner « Save » pour enregistrer le nouveau client. Il n’est pas possible d’ajouter une adresse avant d’avoir enregistré la création d’un nouveau client. Pour ajouter une adresse, entrer l’adresse, le code postal et la ville puis cliquer sur « Adress » pour ajouter l’adresse. Une vérification est réalisée afin de ne pas entrer de ville erronée.
* Pour modifier un client, il faut cliquer sur le bouton « edit » puis saisir l’ID du client que l’on souhaite modifier dans « ID Client ». Enfin, il suffit de modifier toutes les cases présentes sur la page puis cliquer sur save. Cependant, il n’est pas possible de modifier une adresse. Nous pouvons uniquement en ajouter une nouvelle. Cela signifie qu’en cas de déménagement, les anciennes adresses resteront sauvegardées en plus de la nouvelle.



1. Page pour modifier un client

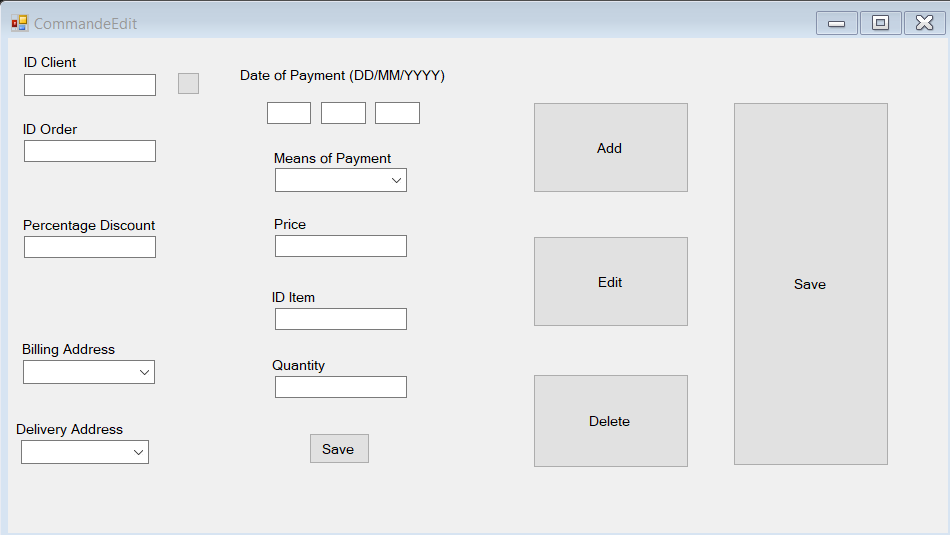
Gestion des commandes

Lors de la sélection de la fonctionnalité « Commande », nous pouvons choisir entre deux actions puis cliquer sur valider :

1. Si l’action choisie est « show all », toutes les données des commandes sont affichées avec leurs articles.
2. Si l’action choisie est « add, delete, change », il est alors possible d’ajouter une nouvelle commande, de la supprimer ou de la modifier mais aussi de modifier les articles commandés :

* Pour supprimer, il faut cliquer sur « delete » entrer l’ID de la commande que l’on souhaite supprimer. Puis sélectionner le bouton « save » à droite afin de réaliser la suppression des données. Cela supprime à la fois la commande et les articles qui la compose.
* Pour finir, pour ajouter d’ajouter une nouvelle commande, il faut tout d’abord cliquer sur « Add » puis entrer les données dans toutes les cases excepté l’ID Commande. Comme précédemment, il suffit de sélectionner le bouton « Save » à droite pour enregistrer la nouvelle commande. Il n’est pas possible d’ajouter un article avant d’avoir enregistré la création d’une nouvelle commande. Pour ajouter un article, entrer l’article et sa quantité puis cliquer sur le bouton « save » en dessous de quantité pour ajouter l’article.
* Pour modifier une commande, il faut cliquer sur le bouton « edit » puis saisir l’ID de la commande que l’on souhaite modifier dans « ID Commande ». Enfin, il suffit de modifier toutes les cases présentes sur la page puis cliquer sur save. Cependant, il n’est pas possible de modifier un article ou de le supprimer. Nous pouvons uniquement en ajouter un nouveau.

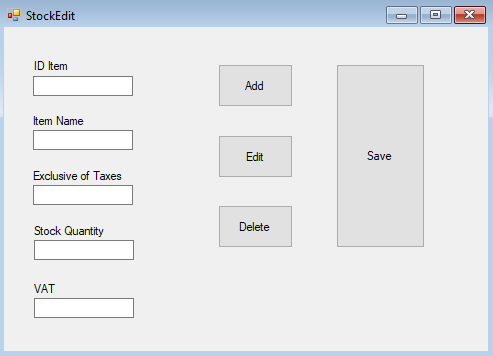
Le bouton situé à droite de « ID Client » devra être utilisé pour remplir les menus déroulants après avoir rentré un ID Client. C’est-à-dire que ce bouton servira lors de l’ajout d’une commande ou de sa modification.



1. Page pour modifier une commande

Gestion du stock

La page de gestion du stock « Stock » reprend le même mode de fonctionnement que la page Gestion du personnel en utilisant l’« ID Item ». C’est pour cela que nous ne le redétaillerons pas ici.



1. Page pour modifier un article

Gestion des statistiques

Lors de la sélection de la fonctionnalité « Statistic ». Un menu déroulant s’affiche montrant les neuf différents calculs de statistiques réalisables.

Une image contenant texte

Description générée automatiquement

1. Page permettant de sélectionner les statistiques souhaitées

Pour certaines statistiques, les données s’affichent simplement sur l’interface sans actions supplémentaires. C’est ce qu’il se produit pour “Calculate the average cart (after discount)”, “Items under the limit replenishment”, “The 10 best - selling items”, “The 10 least sold items”, “Calculate the commercial value of the stock”, “Calculate the purchase value of the stock”.

Pour « Revenue for a specific period », il faudra entrer une date de début et une date de fin. Le bouton « Validate » affichera le chiffre d’affaires sur la période sélectionnée dans la case correspondante.

Pour « Total sales for a selected client » il faudra entrer un Id d’un client. Le bouton « Validate » affichera montant total de toutes les commandes qu’a passé ce client.

Pour « Simulate changes in commercial values to deduce the value of the stock » il faudra sélectionner les différentes données dans les menus déroulants. Le bouton « Validate » affichera alors la valeur du stock en fonction des données sélectionnées.

Clôture du projet

Planning réel

* CROS Hugues pour l’interface de l’application
* DEREMY Baptiste pour l’implémentation et la gestion de la base de données
* MOIROUD Maxime pour la partie contrôleur
* PAPY Lisa pour une partie de l’interface, l’élaboration du cahier des charges et la présentation



1. Planning réel partie 1/2



1. Planning réel partie 2/2

Finalement, nous n’avons pas respecté le planning. En effet, nous avons été contraints de recommencer le projet. La BDD avait été réalisée en MySQL cependant, lors de l’intégration de celle-ci en C++, nous avons eu des problèmes liés à la bibliothèque « mysql.h ». Nous avons aussi dû revoir l’interface car Qt n’était pas compatible avec SQL.

Malgré cela, nous avons réussi à rattraper au mieux le retard que cela nous a engendré.

Conclusion

Finalement, le client pourra utilisé le logiciel afin de réaliser la gestion des différentes fonctionnalités suivantes : personnel, catalogues, client, commande et statistiques. Nous aurions toutefois pu envisage de changer le design de l’interface. Cela reste possible de le modifier afin de correspondre à la volonté de l’entreprise. Nous aurions pu aussi afficher les données dans la même interface que celle qui permet de les modifier au choix de l’entreprise.

Bilan

Nous avons malheureusement perdu beaucoup de temps pour recommencer une grande partie du projet. Sans cela, nous aurions pu aller beaucoup plus loin et se concentrer par la suite sur des points secondaires. Nous aurions par exemple pu générer créer un utilisateur différent pour chaque membre du personnel ou générer des factures. Nous aurions aussi pu entrer le nom de l’article au lieu de l’ID par exemple.

Pour conclure, ce projet c’est bien passé grâce à une excellente communication et entente dans le groupe. Il a été possible de rattraper le retard engendré grâce à beaucoup de travail et une forte volonté de chacun. Il a été possible de demander de l’aide à tout les membres du groupe en cas de problème. L’utilisation de discord a été extrêmement bénéfique afin de collaborer malgré le confinement.

Annexes

Outils

**Github**: <https://github.com/Juggernes/ProjetPOO.git>

Organisation : Daily Meeting

Il n’y a pas eu de réunions planifiées. En effet, nous avons de nous même décider de se connecter tous les jours à 8h30 et à 13h30 afin de pouvoir échanger sur le projet. Lors de ces courtes réunions, nous échangions sur le travail réalisé et sur ce qu’il restait à faire. Lorsque nous travaillions sur le projet, nous nous connections alors sur Discord afin de pouvoir demander de l’aide en cas de problème et prévenir de potentiels changements. Cette méthode a permis de suivre l’avancer de tout le groupe et connaître toutes difficultés rencontrées par nos camarades. Cela permet aussi de maîtriser au mieux les parties réalisées par nos camarades.