



Planning Hospitalier

Rapport du Jalon 2

Antivackis Vanessa

Forest Jules

Bennama Meriem

Haton Tom

Cailleau Dylan

Mohammed Arwa

1. Backlog : Mise à jour.....	1
1.1. Nouvelle version.....	1
1.2. Ce qui a été réalisés.....	2
1.3. Priorités à venir.....	2
2. Planification.....	2
2.1. Tableau descriptif des tâches.....	2
2.2. Diagramme de PERT.....	2
3. Diagramme de classe au niveau conception.....	2
4. Cas jusqu'au niveau conception d'un ajout d'un créneau.....	2
4.1. Alternatif 1 : Acceptation de la demande.....	2
4.1.1. Niveau système.....	2
4.1.1.1. Scénario Système.....	2
4.1.1.2. Diagramme de séquence système.....	2
4.1.2. Niveau métier.....	2
4.1.2.1. Scénario métier.....	2
4.1.2.2. Diagramme de séquence métier.....	2
4.1.2.3. Diagramme de classe métier.....	2
4.1.2.4. Diagramme de package métier.....	2
4.1.2.5. Diagramme de déploiement métier.....	2
4.1.3. Niveau conception.....	2
4.1.3.1. Diagramme de séquence conception.....	2
4.1.3.2. Diagramme de classe conception.....	2
4.1.3.3. Diagramme de package conception.....	2
4.1.3.4. Diagramme de déploiement conception.....	2
4.2. Alternatif 2 : Un créneau déjà existant.....	2
4.2.1. Niveau système.....	2
4.2.1.1. Scénario système.....	2
4.2.1.2. Diagramme de séquence système.....	2
4.2.2. Au niveau métier.....	2
4.2.2.1. Scénario métier.....	2
4.2.2.2. Diagramme de séquence métier.....	2
4.2.2.3. Diagramme de classe métier.....	2
4.2.2.4. Diagramme de package métier.....	2
4.2.2.5. Diagramme de déploiement métier.....	2
4.2.3. Niveau conception.....	2
4.2.3.1. Diagramme de séquence conception.....	2

4.2.3.2.	Diagramme de classe conception.....	2
4.2.3.3.	Diagramme de package conception.....	2
4.2.3.4.	Diagramme de déploiement conception.....	2
4.3.	Alternatif 3 : Contraintes non respectés.....	2
4.3.1.	Niveau système.....	2
4.3.1.1.	Scénario système.....	2
4.3.1.2.	Diagramme de séquence système.....	2
4.3.2.	Niveau métier.....	2
4.3.2.1.	Scénario métier.....	2
4.3.2.2.	Diagramme de séquence métier.....	2
4.3.2.3.	Diagramme de classe métier.....	2
4.3.2.4.	Diagramme de package métier.....	2
4.3.2.5.	Diagramme de déploiement métier.....	2
4.3.3.	Niveau conception.....	2
4.3.3.1.	Diagramme de séquence conception.....	2
4.3.3.2.	Diagramme de classe conception.....	2
4.3.3.3.	Diagramme de package conception.....	2
4.3.3.4.	Diagramme de déploiement conception.....	2
4.4.	Alternatif 4 : Surcharge/Souscharge.....	2
4.4.1.	Niveau système.....	2
4.4.1.1.	Scénario système.....	2
4.4.1.2.	Diagramme séquence système.....	2
4.4.2.	Niveau métier.....	2
4.4.2.1.	Scénario métier.....	2
4.4.2.2.	Diagramme de séquence métier.....	2
4.4.2.3.	Diagramme de classe métier.....	2
4.4.2.4.	Diagramme de package métier.....	2
4.4.2.5.	Diagramme de déploiement métier.....	2
4.4.3.	Niveau conception.....	2
4.4.3.1.	Diagramme de séquence conception.....	2
4.4.3.2.	Diagramme de classe conception.....	2
4.4.3.3.	Diagramme de package conception.....	2
4.4.3.4.	Diagramme de déploiement conception.....	2
5.	Conventions de codage.....	2
5.1.	Conventions de codage.....	2
5.1.1.	Formatage du code.....	2

5.1.1.1.	Indentation.....	2
5.1.1.2.	Taille des lignes.....	2
5.1.1.3.	Lignes blanches.....	2
5.1.1.4.	Espaces.....	2
5.1.1.5.	Déclarations de Classe avec Extends et Implements.....	2
5.1.1.6.	Résumé.....	2
5.1.2.	Nommage.....	2
5.1.2.1.	Général.....	2
5.1.2.2.	Package.....	2
5.1.2.3.	Classes et Interfaces.....	2
5.1.3.	Méthodes.....	2
5.1.3.1.	Nommage.....	2
5.1.3.2.	Déclaration.....	2
5.1.3.3.	Instructions Return.....	2
5.1.4.	Variables.....	2
5.1.4.1.	Nommage.....	2
5.1.4.2.	Déclaration.....	2
5.1.4.3.	Constantes.....	2
5.1.4.4.	Types de variables.....	2
5.1.5.	Blocs de code.....	2
5.1.5.1.	Structures conditionnelles.....	2
5.1.5.2.	Boucles.....	2
5.1.6.	Commentaires.....	2
5.1.6.1.	Documentation.....	2
5.1.6.2.	JavaDoc.....	2
5.1.7.	Autres conventions spécifiques.....	2
5.1.7.1.	Saut de ligne.....	2
5.1.7.2.	Caractères spéciaux.....	2
5.1.7.3.	Accents.....	2
5.2.	Comment sont-elles vérifiées.....	2
6.	Documentation.....	2
7.	Principales requêtes de la Base de Données.....	2
7.1.	Requêtes en SQL.....	2
7.2.	Requêtes en algèbre relationnel.....	3
7.3.	Requêtes en calcul relationnel.....	4

1. Backlog : Mise à jour

1.1. Nouvelle version

<u>Couleurs</u>	<u>Niveaux D'Importance & Explication</u>
Rouge	Le plus important
Orange	Moyennement important pour le moment
Bleu	Le moins important (fonctionnalité 2)

1. Je veux pouvoir en tant que cadre, ajouter du personnel, le modifier, le supprimer, le consulter

1.1. Création de l'IHM sous forme d'un pop-up pour afficher tous les personnels

1.2. Pop-up de consultation de la fiche d'un personnel choisi

1.3. Ajout d'un personnel :

1.3.1. Pop-up pour saisir les informations et ajouter

1.3.2. Mise en forme pour la base

1.3.3. Insertion dans la BDD

1.3.4. Pop-up confirmation ajout

1.4. Modification d'un personnel :

1.4.1. Pop-up de pour modifier les informations

1.4.2. Check et mise en forme des données saisies pour la BDD

1.4.3. Modification dans la BDD

1.4.4. Pop-up confirmation de la modification

1.5. Suppression d'un personnel

1.5.1. Checking si personnel non-affecté

1.5.2. Pop-up de demande de validation de suppression

1.5.3. Pop-up de confirmation de suppression

2. En tant que cadre je veux pouvoir créer un EDT avec différents créneaux

2.1. Créer un créneau dans l'EDT

2.1.1. Pop-up saisie de données (donner une date et un créneau horaire logique (pas de création dans le passé))

2.1.2. Premier checking des données saisies

2.1.3. Pop-up pour définir le personnel requis et les spécialités requises correspondantes (lors de la saisie le cadre pourra saisir une spécialité et y lier un nombre de personnes demandées)

2.1.4. Deuxième checking des données saisies

2.1.5. Checking des contraintes

2.1.6. Insertion dans la BDD

3. En tant que cadre je veux pouvoir consulter l'EDT d'une façon globale et hebdomadaire
 - 3.1. Récupération de l'EDT général dans la BDD
 - 3.2. Création de l'IHM d'affichage qui sera la page principale de l'app
 - 3.3. Pouvoir ajouter un personnel sur un créneau depuis cet EDT
 - 3.3.1. Pop-up d'affichage de tous les personnels disponibles
 - 3.4. Pouvoir supprimer un personnel d'un créneau
 - 3.5. Pouvoir modifier une affectation de personnel, demande d'affectation
 - 3.5.1. IHM de modification de demande
 - 3.6. Affichage des non-respects de contraintes
 - 3.6.1. IHM d'affichage des contraintes non-respectées
 - 3.6.2. Checking des contraintes non-respectées
 - 3.7. Imprimer l'EDT (BONUS)
 - 3.8. Gestion des créneaux problématiques avec des couleurs
4. En tant que cadre je veux pouvoir consulter les EDT par personnel
 - 4.1. Choix du personnel ciblé (encart pour choisir dans l'EDT général), Pop-up de la liste des personnels (réutilisation du Pop-up en 1.2 mais revisité pour renvoyer sur le planning du personnel au lieu de sa fiche)
 - 4.2. Récupération de l'EDT dans la BDD
 - 4.3. Affichage de l'EDT dans la page principale
 - 4.4. Pouvoir ajouter un personnel sur un créneau depuis cet EDT
 - 4.4.1. Pop-up ajout de personnel sur un créneau en 3.3
 - 4.5. Imprimer l'EDT
5. En tant que cadre je veux pouvoir consulter les contraintes non-respectées
 - 5.1. Récupération des contraintes dans la base
 - 5.2. IHM d'affichage des contraintes non-respectées avec mini-descriptif
 - 5.3. Possibilité en cliquant sur la contrainte d'y être emmené directement dans le planning (hyperlien)
6. En tant que cadre je veux pouvoir consulter les contraintes légales
 - 6.1. Récupération des contraintes légales dans la BDD
 - 6.2. Pop-up de récap des contraintes légales
 - 6.3. Ajout d'une contrainte
 - 6.3.1. Pop-up de saisie
 - 6.3.2. Pop-up de confirmation ajout
 - 6.4. Modification d'une contrainte
 - 6.4.1. Pop-up de modification

- 6.4.2. Pop-up de confirmation modification
- 6.5. Suppression d'une contrainte
 - 6.5.1. Pop-up demande de validation suppression
 - 6.5.2. Pop-up confirmation de suppression

7. En tant que cadre je veux pouvoir consulter les contraintes de spécialité

- 7.1. Récupération des contraintes de spécialités dans la BDD
- 7.2. Pop-up de récap des contraintes de spécialité
- 7.3. Ajout d'une contrainte de spécialité
 - 7.3.1. Pop-up de saisie
 - 7.3.2. Pop-up de confirmation ajout
- 7.4. Modification d'une contrainte de spécialité
 - 7.4.1. Pop-up de saisie
 - 7.4.2. Pop-up de confirmation modification
- 7.5. Suppression d'une contrainte de spécialité
 - 7.5.1. Pop-up demande de validation suppression
 - 7.5.2. Pop-up confirmation de suppression

8. En tant que cadre je veux pouvoir consulter les fonctions

- 8.1. Récupération des fonctions dans la BDD
- 8.2. Pop-up de récap des fonctions existantes
- 8.3. Ajout d'une fonction
 - 8.3.1. Pop-up ajout d'une fonction
 - 8.3.2. Checking des données saisies
 - 8.3.3. Insertion dans la base
 - 8.3.4. Pop-up confirmation ajout
- 8.4. Modification d'une fonction
 - 8.4.1. Pop-up modification d'une fonction
 - 8.4.2. Checking des données saisies
 - 8.4.3. Insertion dans la base
 - 8.4.4. Pop-up confirmation modification
- 8.5. Suppression d'une fonction
 - 8.5.1. Pop-up demande de validation suppression
 - 8.5.2. Pop-up confirmation de suppression

9. En tant que cadre je veux pouvoir consulter les spécialités

- 9.1. Récupération des contraintes de spécialités dans la BDD
- 9.2. Pop-up de récap des spécialités existantes
- 9.3. Ajout d'une spécialité
 - 9.3.1. Pop-up ajout d'une spécialité
 - 9.3.2. Checking des données saisies

- 9.3.3. Insertion dans la base
 - 9.3.4. Pop-up confirmation ajout
- 9.4. Modification d'une spécialité
 - 9.4.1. Pop-up modification d'une spécialité
 - 9.4.2. Checking des données saisies
 - 9.4.3. Insertion dans la base
 - 9.4.4. Pop-up confirmation modification
- 9.5. Suppression d'une spécialité
 - 9.5.1. Pop-up demande de validation suppression
 - 9.5.2. Pop-up confirmation de suppression
- 9.6. Proposition d'ajouter une contrainte associée

10. Je veux pouvoir dupliquer un créneau (BONUS)

11. Je veux pouvoir dupliquer une semaine type (BONUS)

12. Je veux pouvoir intégrer une prise de congés (BONUS)

1.2. Ce qui a été réalisés

Nous avons depuis fait 2 nouvelles réunions avec le groupe.

Lors de ces réunions, nous avons répartis les différentes tâches qui nous attendaient soient :

- L'ajout d'un cas d'utilisation jusqu'au niveau conception, réalisé par Arwa Mohammed,
- Le développement des cas d'utilisation et du diagramme de classe au niveau conception, réalisé par Jules Forest,
- L'écriture des conventions de codage, réalisé par Meriem Bennama,
- La planification des tâches (temporalité), réalisé par Dylan Cailleau
- La réalisation d'un premier jet de l'emploi du temps en java, réalisé par Dylan Cailleau,
- L'écriture des requêtes qui seront les plus utilisées en SQL, algèbre relationnel et calcul relationnel, réalisé par Tom Hatton et Vanessa Antivackis,
- La documentation technique au travers de l'IDE Eclipse (Javadoc), réalisé par Vanessa Antivackis.

Une fois que toutes ces tâches fussent terminées, nous avons convenus de quelles portions de code chacun effectuerait.

1.3. Priorités à venir

Nous passons donc maintenant à la partie la plus conséquente et emblématique de notre projet : le code.

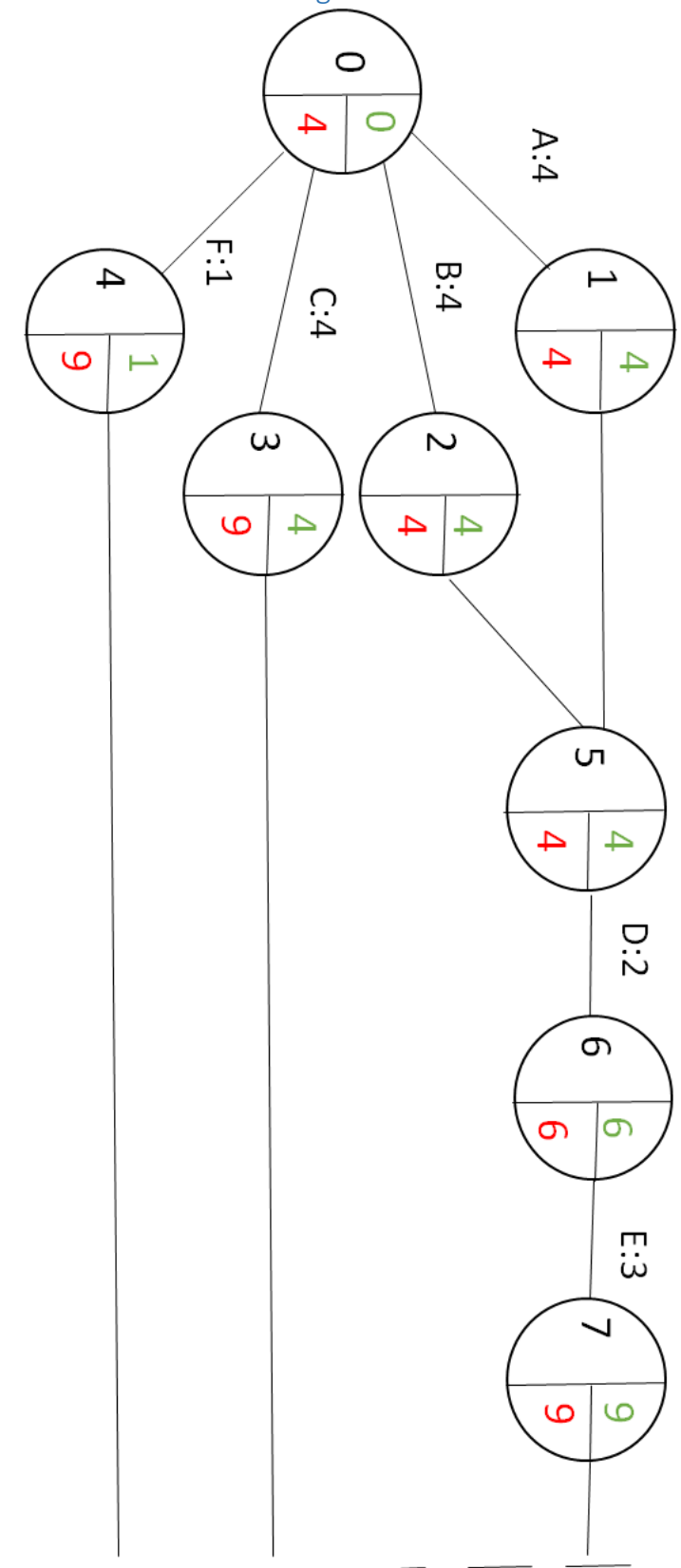
Il nous faudra aussi réaliser un diaporama afin d'avoir un support visuel lors de la présentation de notre projet.

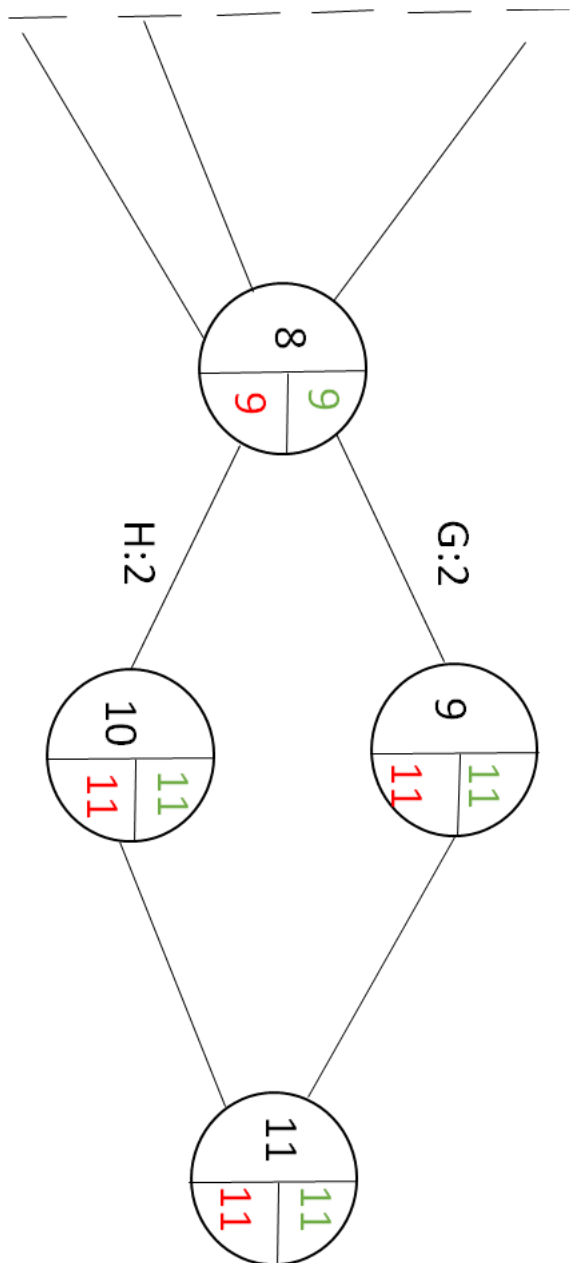
2. Planification

2.1. Tableau descriptif des tâches

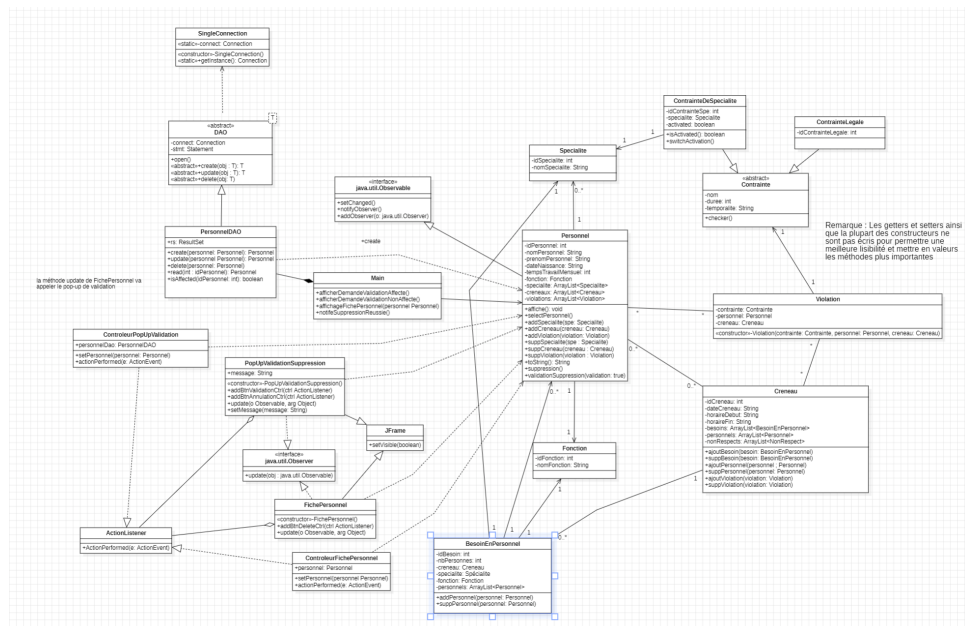
Études	Objectif	Ressource humaine	Temps de conception	Date de fin	Dépendances
A	Nouveau Cas : afficher le planning global	1 personne	4 jours	15/12	Aucune
B	Nouveau Cas : afficher le planning personnel	1 personne	4 jours	15/12	Aucune
C	Faire les requêtes principales de la BDD	2 personnes	4 jours	15/12	Aucune
D	Réviser le diagramme de classe	3 personnes	2 jours	17/12	A, B
E	Ecrire la Documentation du Diagramme de Classe	1 personne	3 jours	20/12	D
F	Convention de codage	Tout le monde	1 jour	04/12	Aucune
G	Préciser le Backlog	3 personnes	2 jours	21/12	E, F, C
H	Faire le compte rendu	1 personne	2 jours	21/12	E, F, C

2.2. Diagramme de PERT





3. Diagramme de classe au niveau conception



4. Cas jusqu'au niveau conception d'un ajout d'un créneau

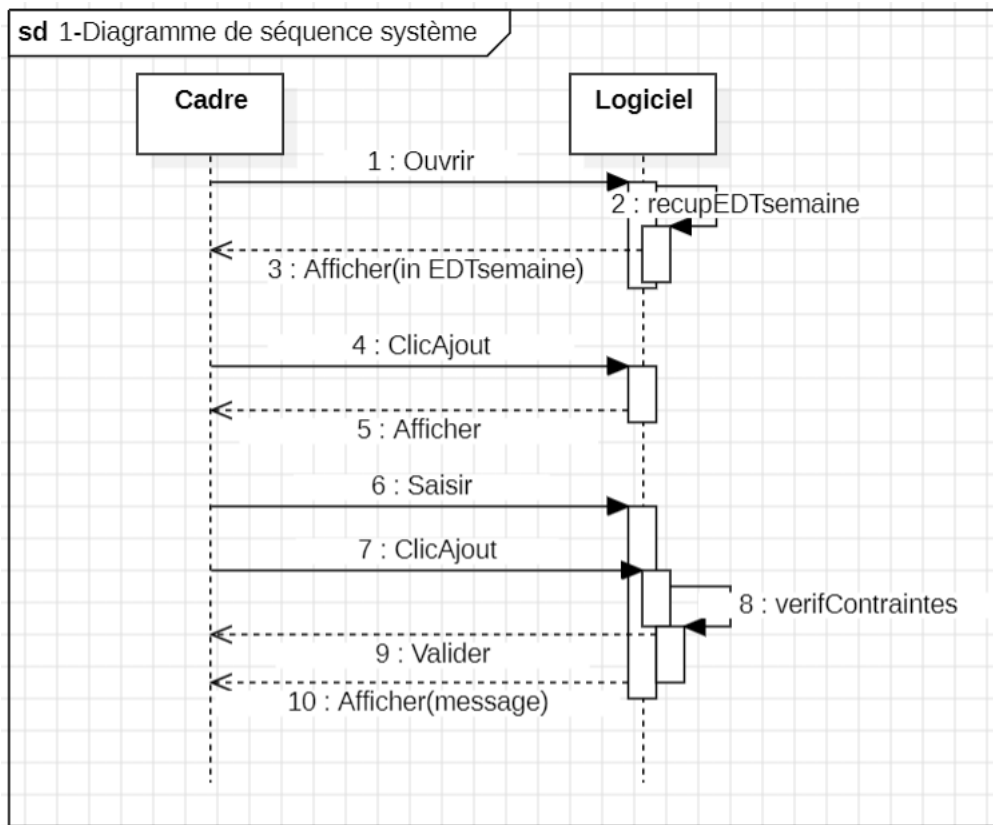
4.1. Alternatif 1 : Acceptation de la demande

4.1.1. Niveau système

4.1.1.1. Scénario Système

1. Le cadre ouvre le logiciel
2. Affichage de l'EDT par semaine
3. Le cadre clique sur l'icône « +créneau »
4. Il entre les informations du créneau souhaité
5. Appuie sur Ajouter
6. Vérification des contraintes
7. Message d'acceptation de la vérification du créneau.

4.1.1.2. Diagramme de séquence système

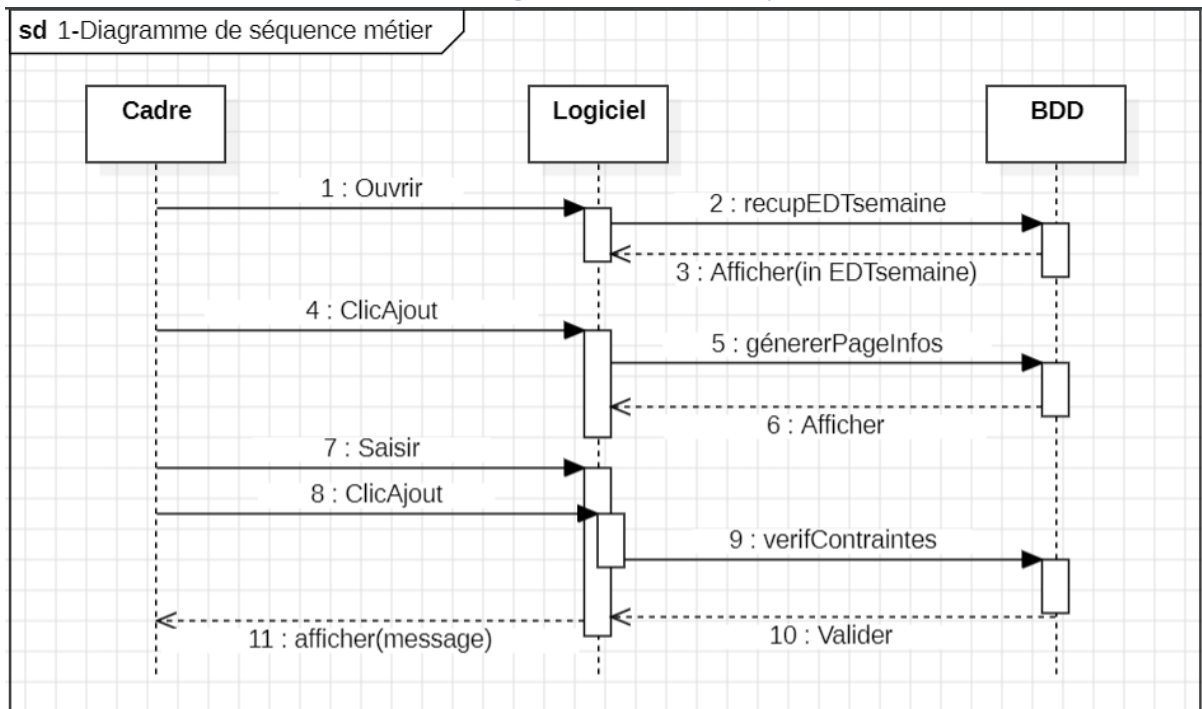


4.1.2. Niveau métier

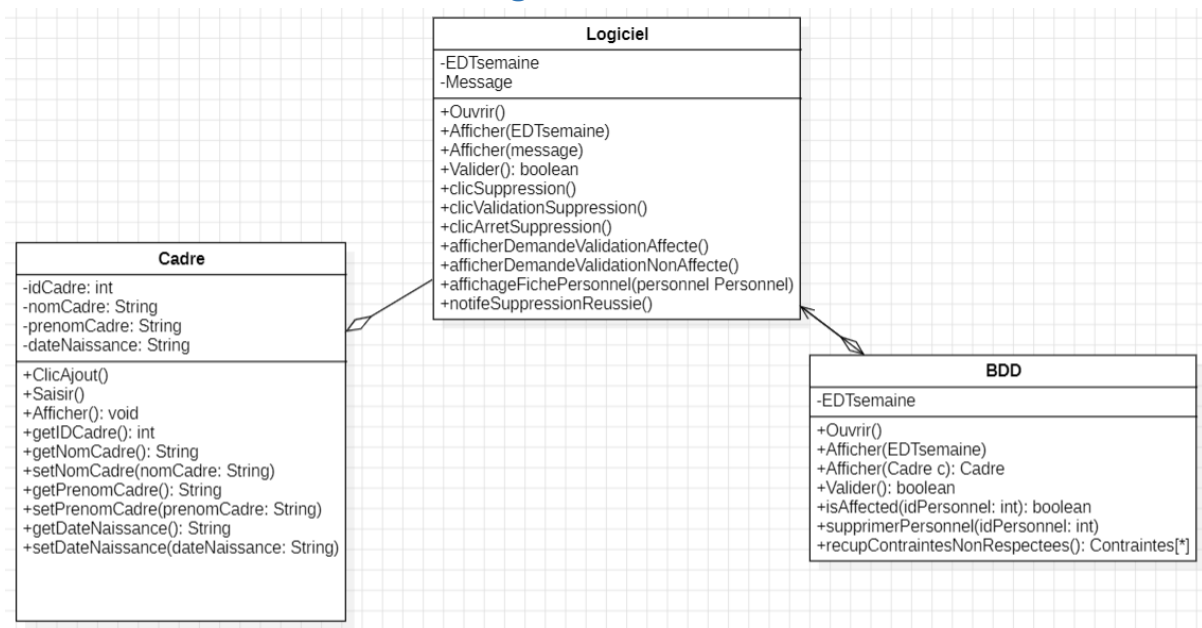
4.1.2.1. Scénario métier

1. Le cadre ouvre le logiciel
2. Récupération de l'EDT/semaine depuis la BDD
3. Affichage EDT/semaine
4. Cliquer sur « +créneau »
5. Générer page informations
6. Affichage page d'informations
7. Saisir informations
8. Vérifier les contraintes
9. Contraintes validées
10. Message d'ajout de créneau réussi

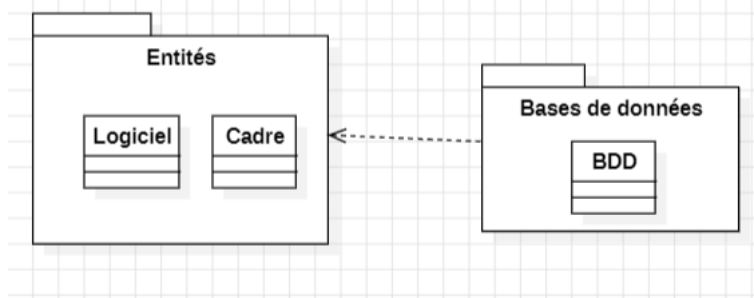
4.1.2.2. Diagramme de séquence métier



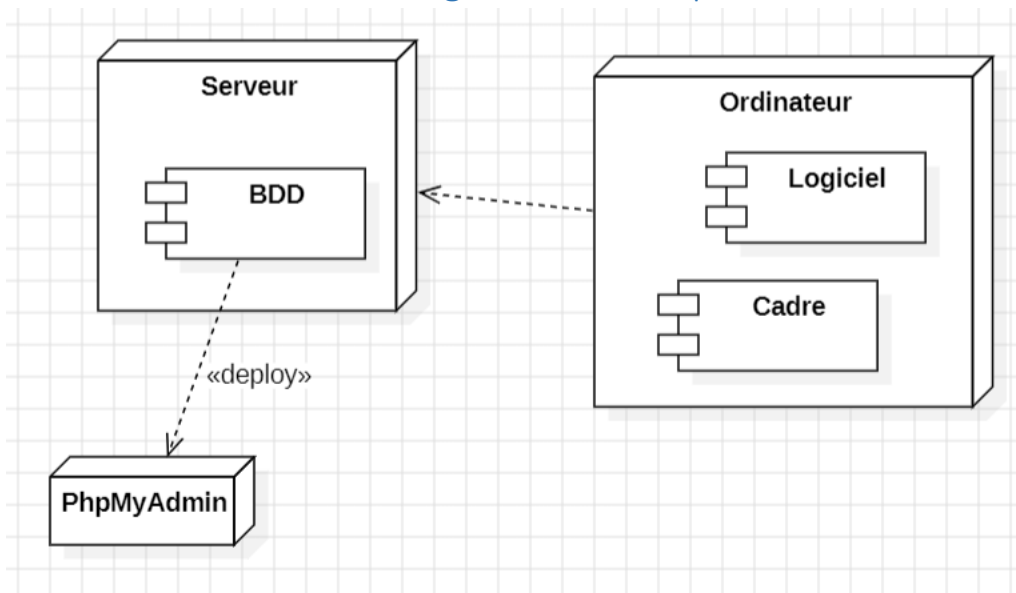
4.1.2.3. Diagramme de classe métier



4.1.2.4. Diagramme de package métier

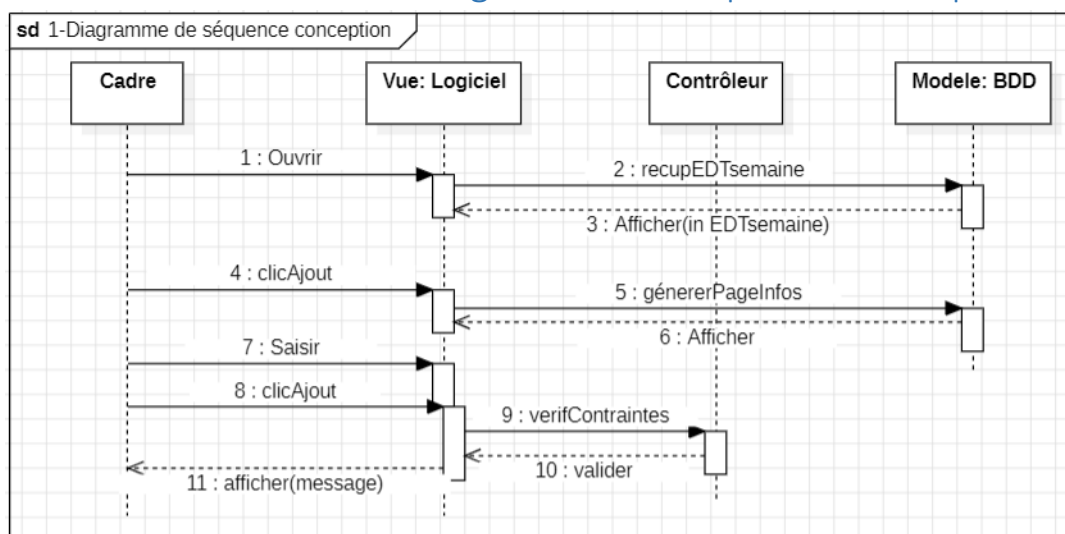


4.1.2.5. Diagramme de déploiement métier

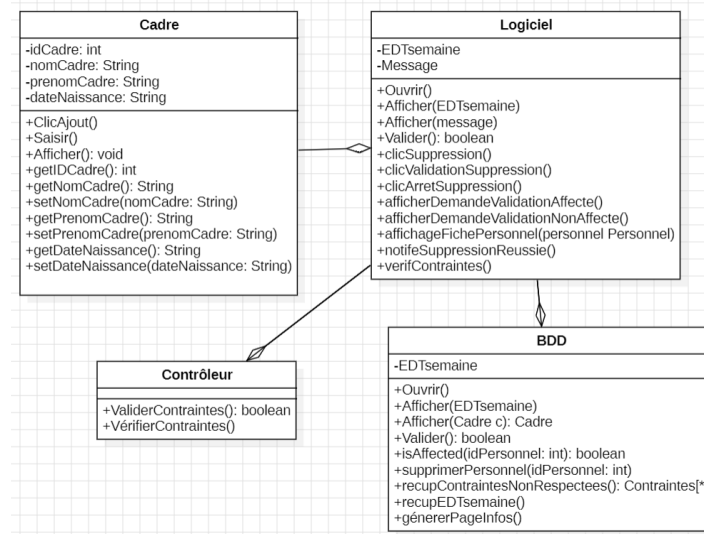


4.1.3. Niveau conception

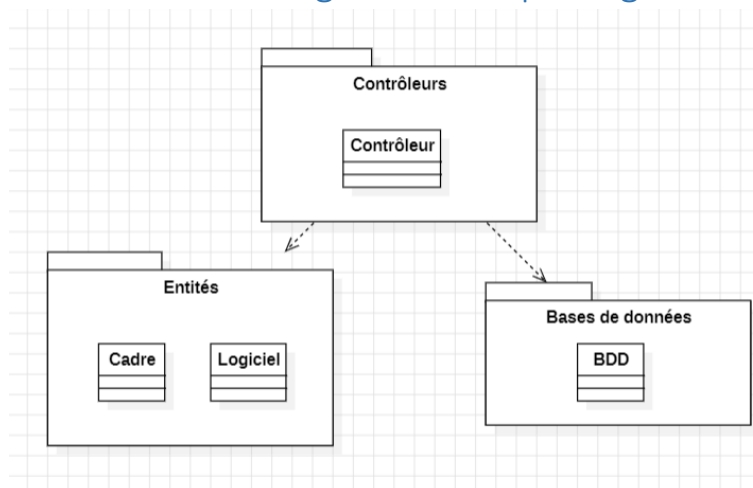
4.1.3.1. Diagramme de séquence conception



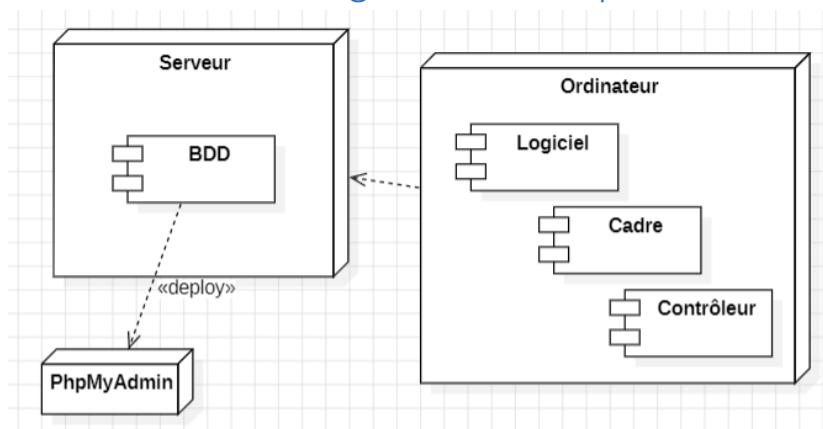
4.1.3.2. Diagramme de classe conception



4.1.3.3. Diagramme de package conception



4.1.3.4. Diagramme de déploiement conception



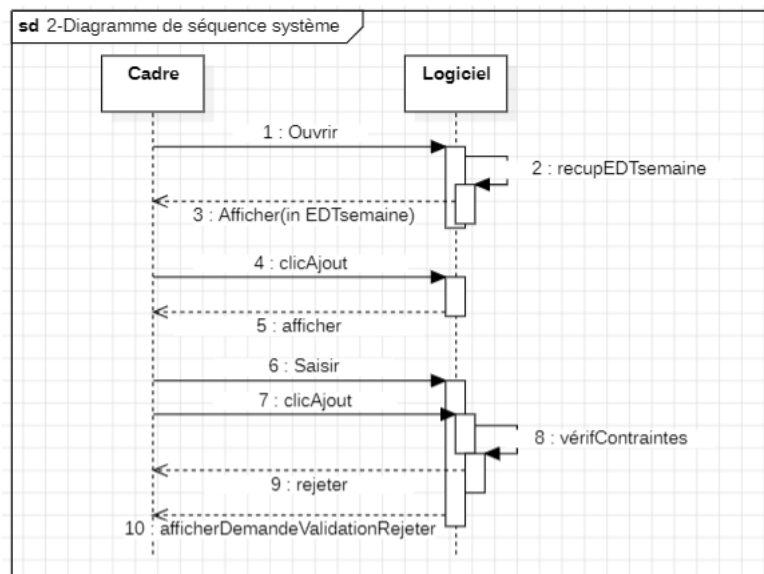
4.2. Alternatif 2 : Un créneau déjà existant

4.2.1. Niveau système

4.2.1.1. Scénario système

1. Le cadre ouvre le logiciel
2. Affichage de l'EDT par semaine
3. Le cadre clique sur l'icône « +créneau »
4. Entrer les informations du créneau souhaité
5. Appuyer sur ajouter
6. Vérification des contraintes
7. Vérification rejetée
8. Message erreur : il y a déjà un créneau

4.2.1.2. Diagramme de séquence système



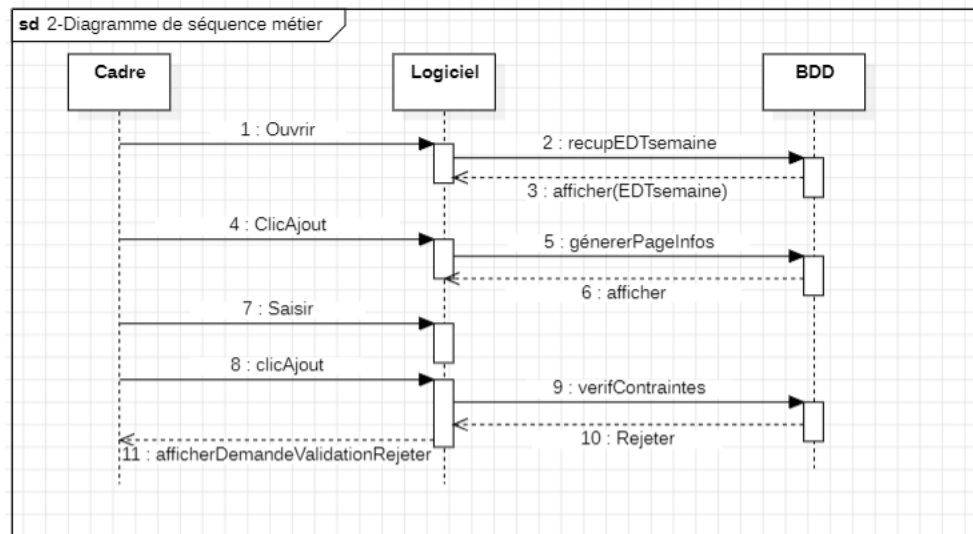
4.2.2. Au niveau métier

4.2.2.1. Scénario métier

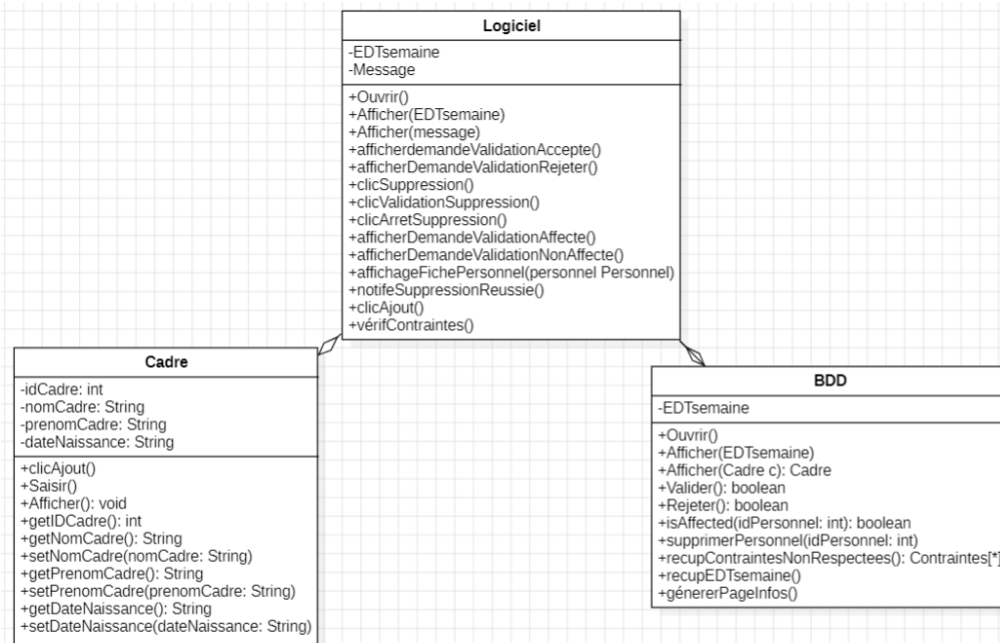
1. Le cadre ouvre le logiciel
2. Récupération de l'EDT/semaine depuis la BDD
3. Affichage EDT/semaine
4. Cliquer sur « +créneau »
5. Générer page informations
6. Affichage page d'informations
7. Saisir informations
8. Vérifier les contraintes
9. Vérification rejetée

10. Affichage du message d'erreur : « il existe déjà un créneau »

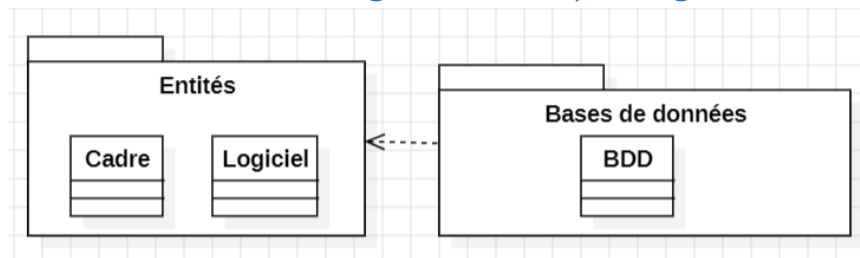
4.2.2.2. Diagramme de séquence métier



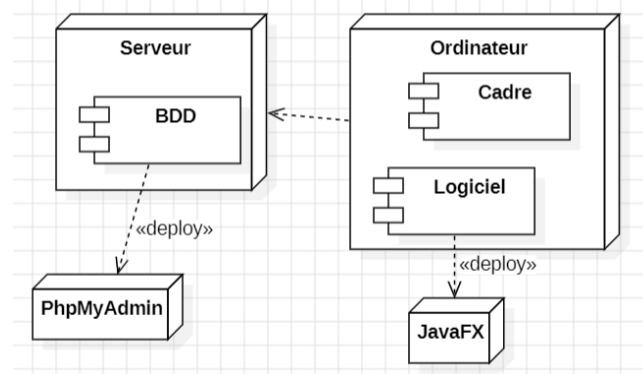
4.2.2.3. Diagramme de classe métier



4.2.2.4. Diagramme de package métier

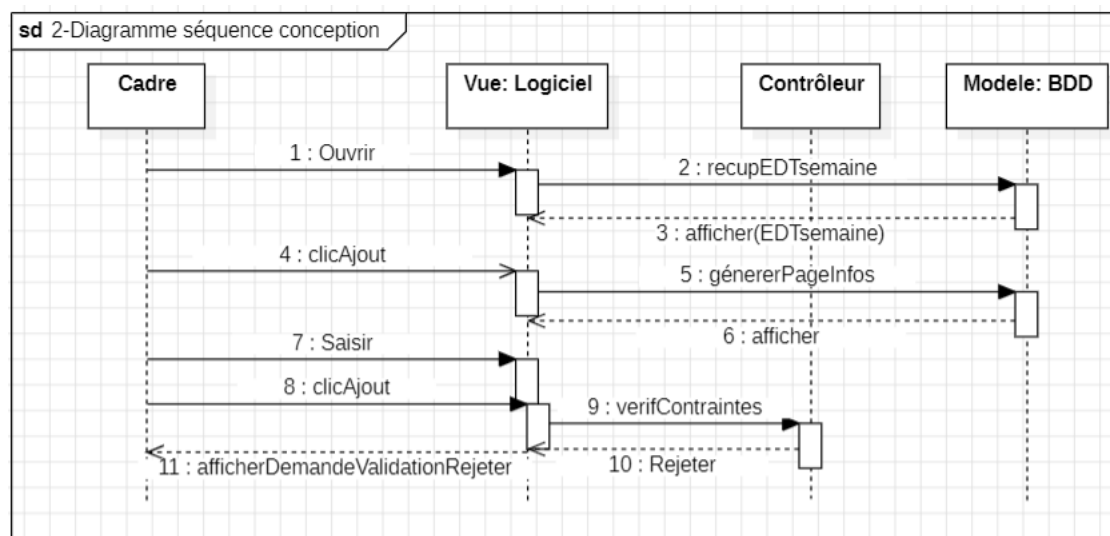


4.2.2.5. Diagramme de déploiement métier

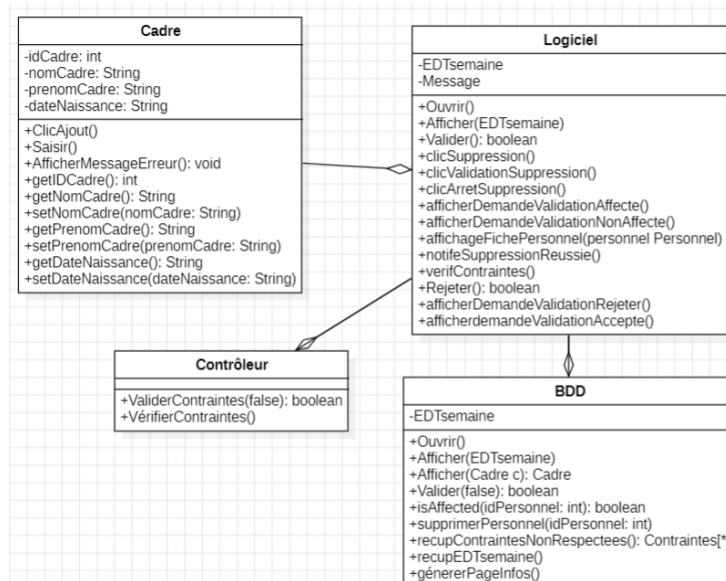


4.2.3. Niveau conception

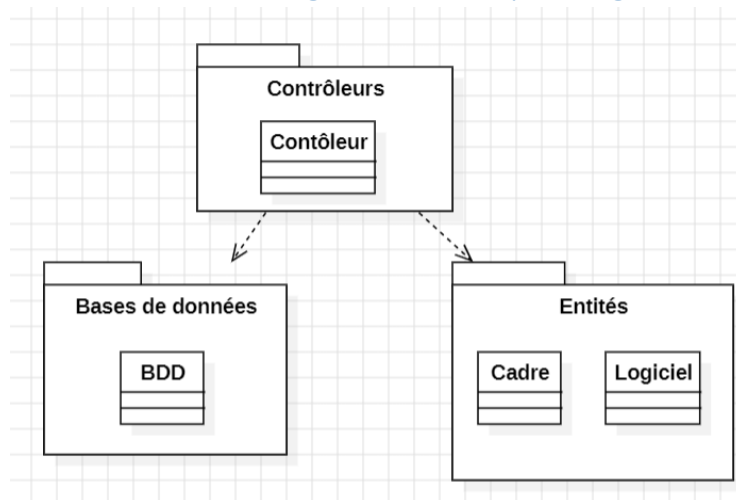
4.2.3.1. Diagramme de séquence conception



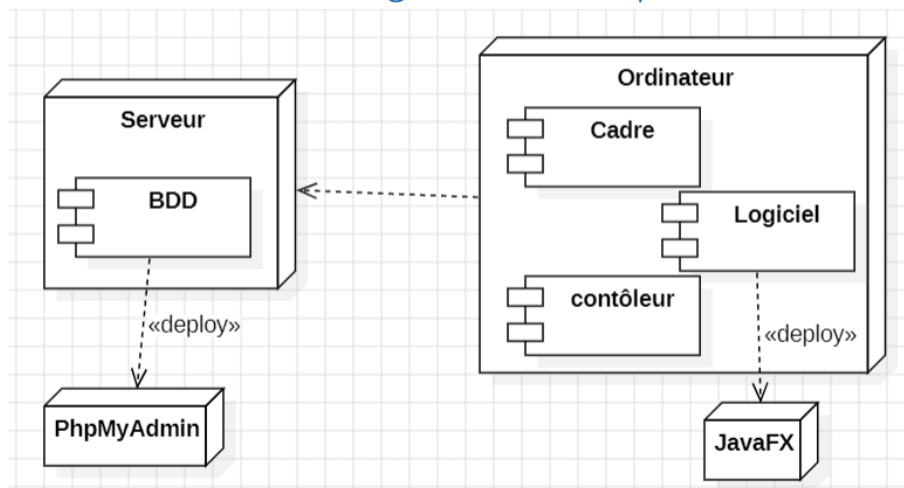
4.2.3.2. Diagramme de classe conception



4.2.3.3. Diagramme de package conception



4.2.3.4. Diagramme de déploiement conception



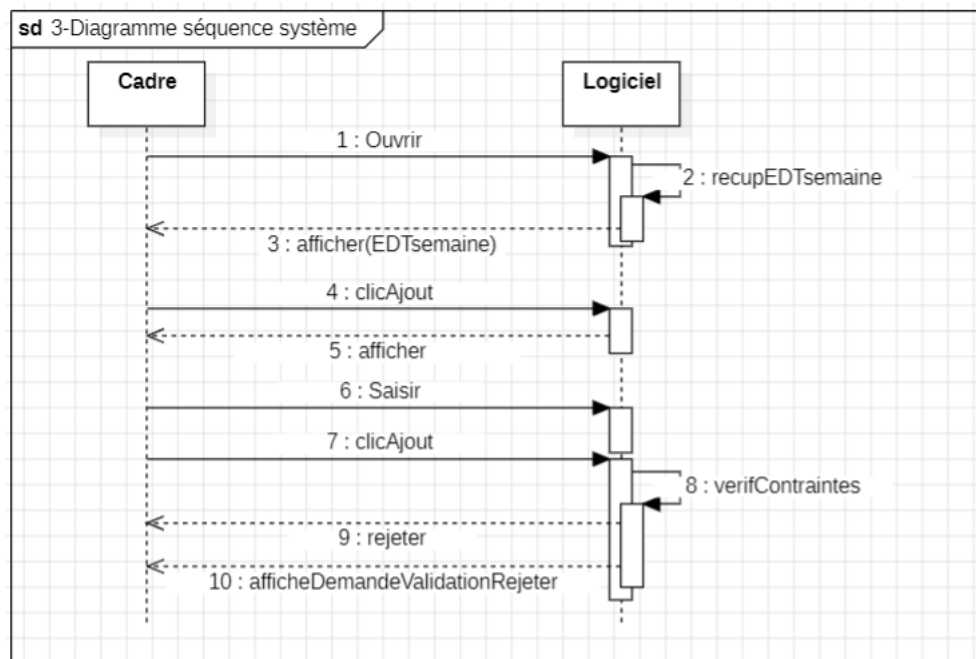
4.3. Alternatif 3 : Contraintes non respectés

4.3.1. Niveau système

4.3.1.1. Scénario système

1. Le cadre ouvre le logiciel
2. Affichage de l'EDT par semaine
3. Le cadre clique sur l'icône « +créneau »
4. Entrer les informations du créneau souhaité
5. Appuyer sur ajouter
6. Vérification des contraintes
7. Vérification rejetée
8. Message erreur : Contraintes non respectés.

4.3.1.2. Diagramme de séquence système

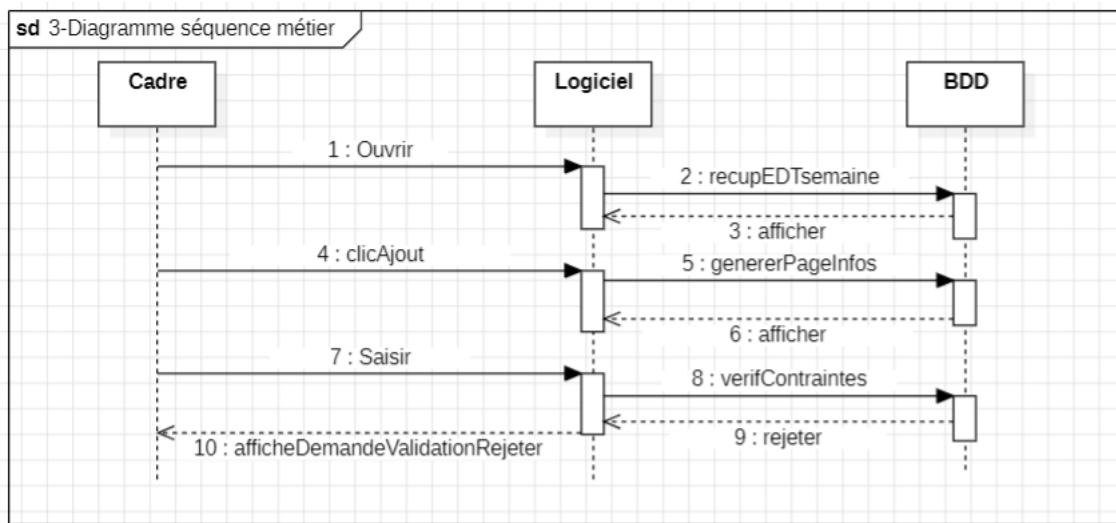


4.3.2. Niveau métier

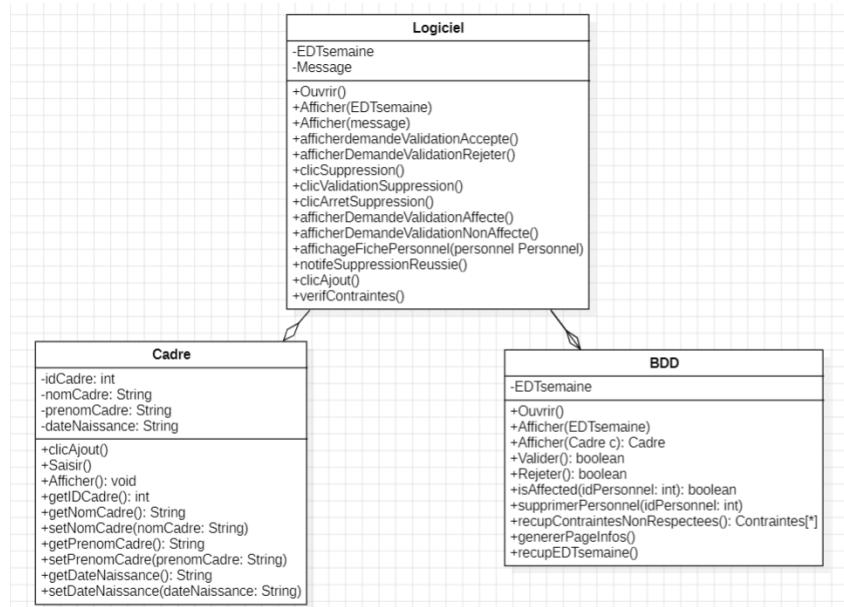
4.3.2.1. Scénario métier

1. Le cadre ouvre le logiciel
2. Récupération de l'EDT/semaine depuis la BDD
3. Affichage EDT/semaine
4. Cliquer sur « +créneau »
5. Générer page informations
6. Affichage page d'informations
7. Saisir informations
8. Vérifier les contraintes
9. Vérification rejetée
10. Affichage du message d'erreur : « Contraintes non respectés »

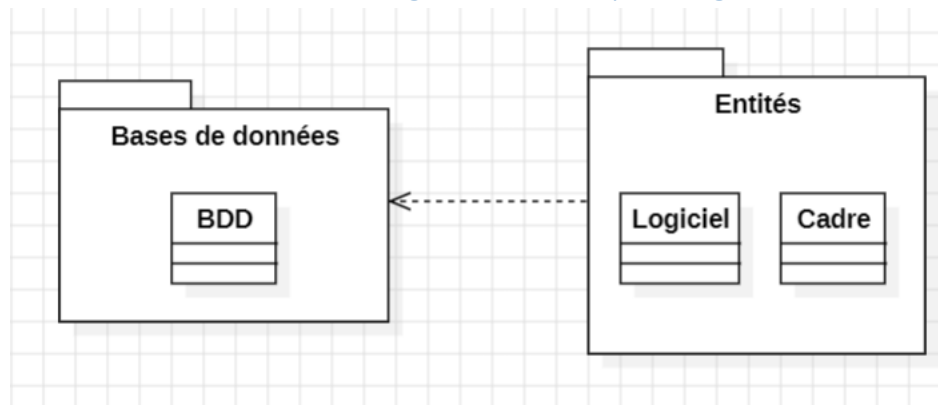
4.3.2.2. Diagramme de séquence métier



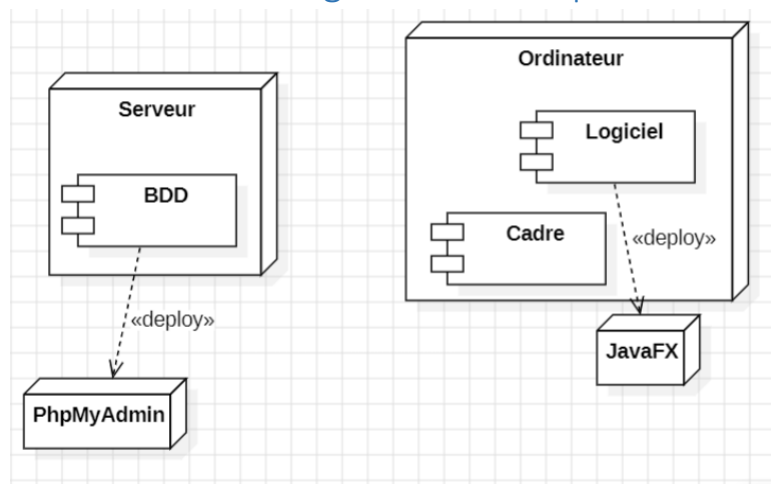
4.3.2.3. Diagramme de classe métier



4.3.2.4. Diagramme de package métier

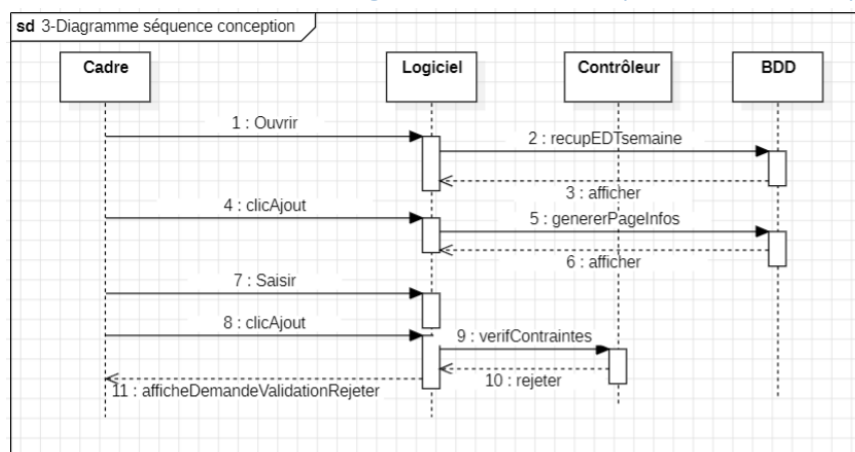


4.3.2.5. Diagramme de déploiement métier

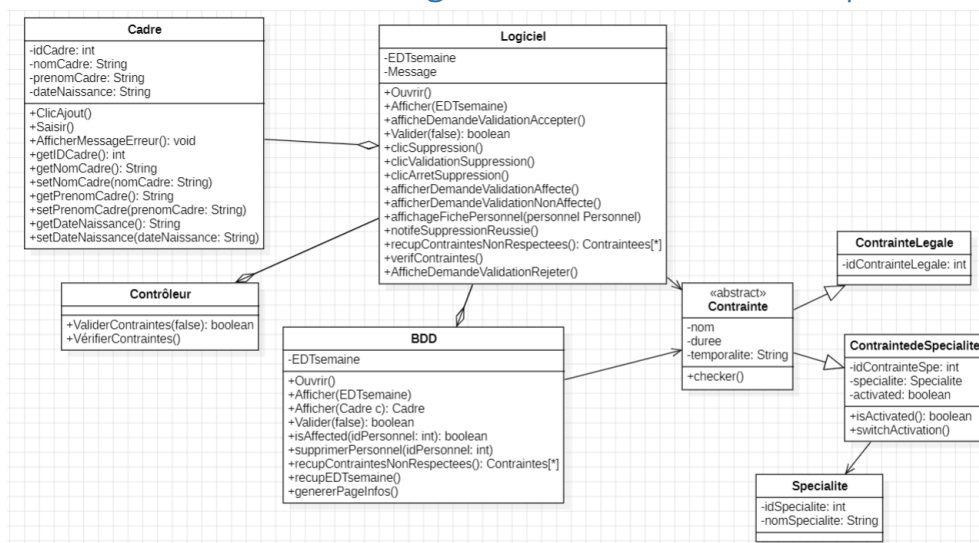


4.3.3. Niveau conception

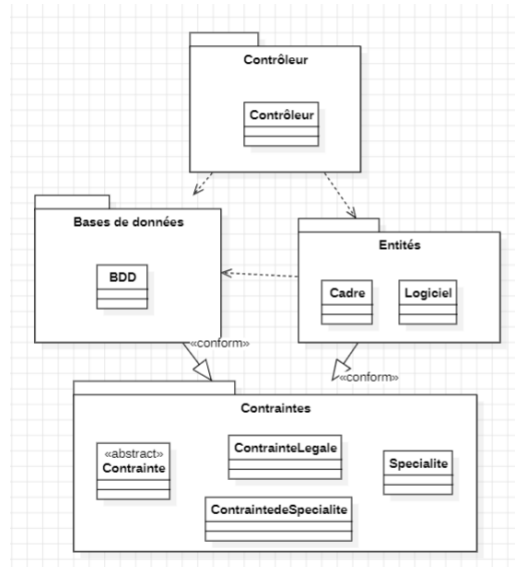
4.3.3.1. Diagramme de séquence conception



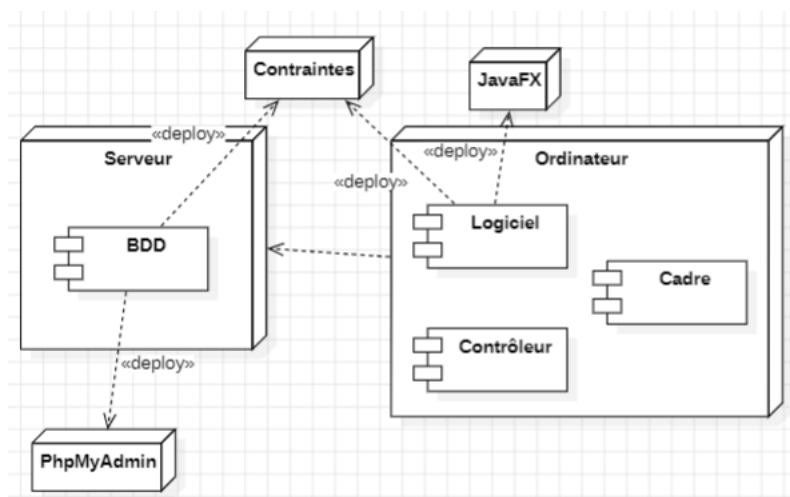
4.3.3.2. Diagramme de classe conception



4.3.3.3. Diagramme de package conception



4.3.3.4. Diagramme de déploiement conception



4.4. Alternatif 4 : Surcharge/Souscharge

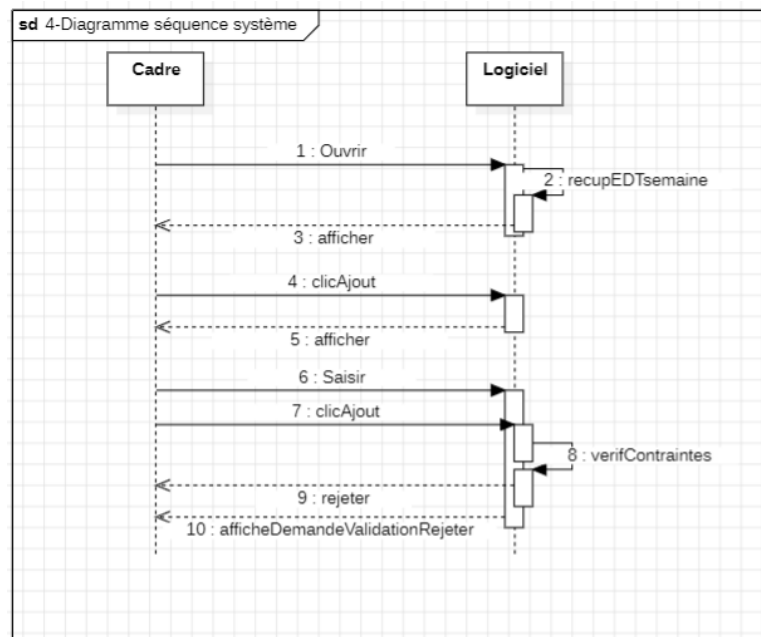
4.4.1. Niveau système

4.4.1.1. Scénario système

1. Le cadre ouvre le logiciel
2. Affichage de l'EDT par semaine
3. Le cadre clique sur l'icône « +créneau »
4. Entrer les informations du créneau souhaité
5. Appuyer sur ajouter
6. Vérification des contraintes

7. Vérification rejetée
8. Message erreur : Surcharge/Sous-charge sur le personnel.

4.4.1.2. Diagramme séquence système



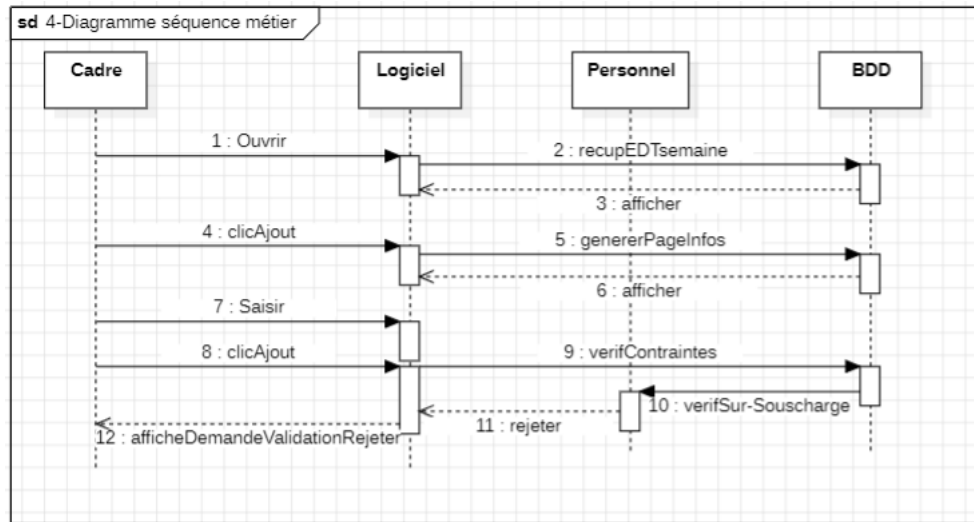
4.4.2. Niveau métier

4.4.2.1. Scénario métier

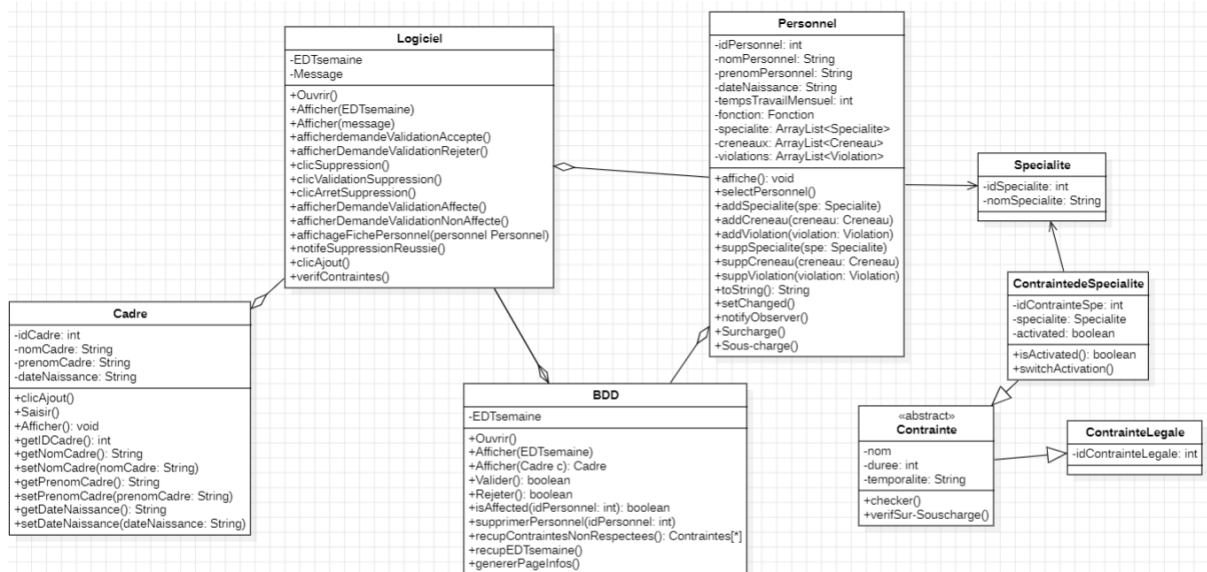
1. Le cadre ouvre le logiciel
2. Récupération de l'EDT/semaine depuis la BDD
3. Affichage EDT/semaine
4. Cliquer sur « +créneau »
5. Générer page informations
6. Affichage page d'informations
7. Saisir informations
8. Vérifier les contraintes
9. Vérification rejetée

10. Affichage du message d'erreur : « Surcharge/sous-charge sur le personnel »

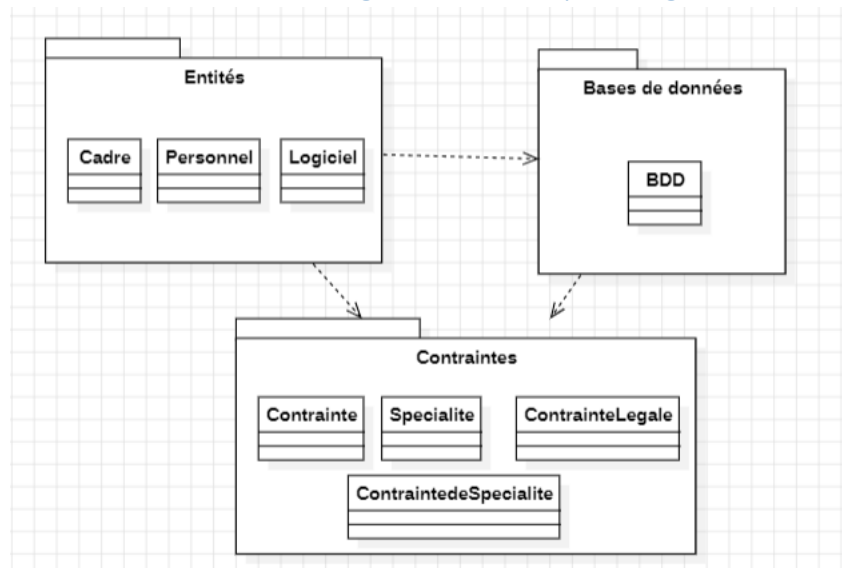
4.4.2.2. Diagramme de séquence métier



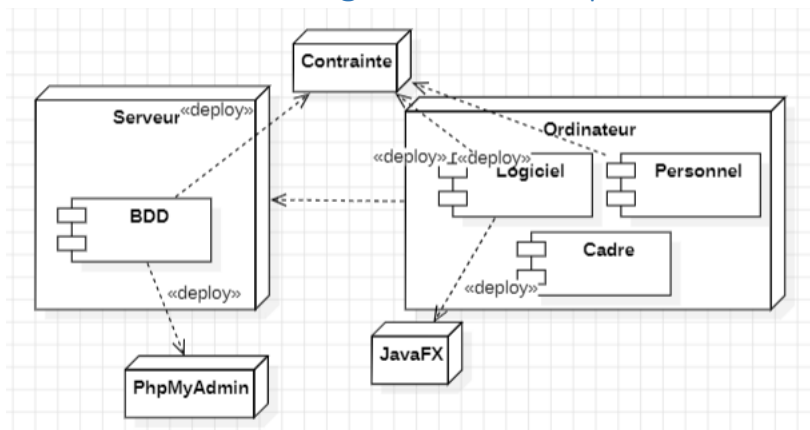
4.4.2.3. Diagramme de classe métier



4.4.2.4. Diagramme de package métier

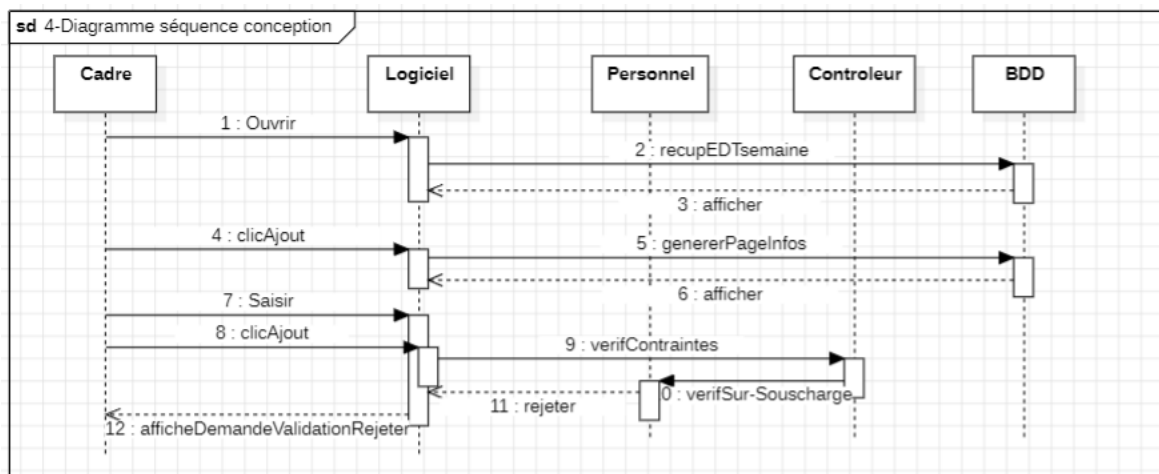


4.4.2.5. Diagramme de déploiement métier

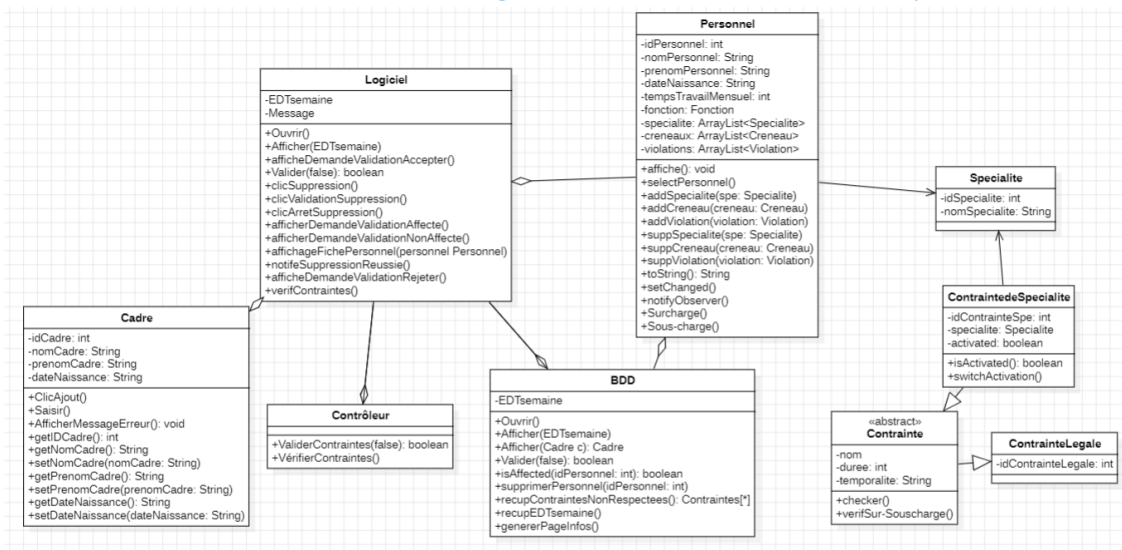


4.4.3. Niveau conception

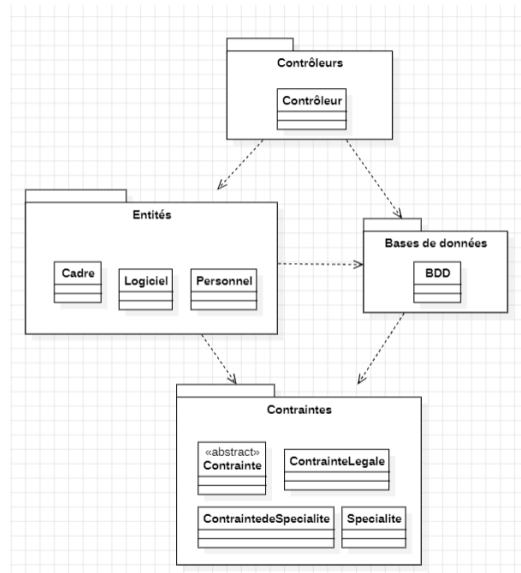
4.4.3.1. Diagramme de séquence conception



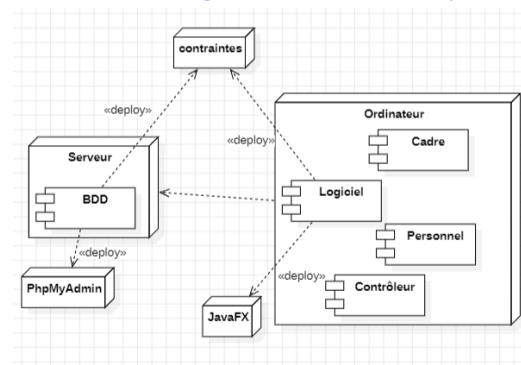
4.4.3.2. Diagramme de classe conception



4.4.3.3. Diagramme de package conception



4.4.3.4. Diagramme de déploiement conception



5. Conventions de codage

5.1. Conventions de codage

5.1.1. Formatage du code

5.1.1.1. Indentation

- Utilisez des espaces plutôt que des tabulations pour l'indentation. Configurez votre éditeur pour remplacer chaque tabulation par 4 espaces.

Exemple :

```
if (condition) {  
    // Ce bloc est indenté avec 4 espaces  
    executerTache();  
}
```

- Chaque nouveau bloc de code (par exemple, à l'intérieur d'une instruction **if** ou **for**) doit augmenter le niveau d'indentation.

5.1.1.2. Taille des lignes

- Limitez la longueur des lignes de code à 80 colonnes, y compris les espaces d'indentation, pour garantir la lisibilité sur différents écrans.

Exemple :

```
// Cette ligne ne dépasse pas 80 colonnes  
  
String message = "Assurez-vous que la longueur de cette ligne est inférieure à 80  
caractères";
```

5.1.1.3. Lignes blanches

- Utilisez des lignes blanches pour séparer logiquement différentes sections du code, telles que les méthodes ou les blocs de code au sein d'une méthode.

5.1.1.4. Espaces

- Autour des opérateurs,
- Après les virgules,
- Après les mots réservés du langage.
- N'utilisez pas d'espace immédiatement à l'intérieur des parenthèses.
- N'utilisez pas d'espaces après le ' ; ' après les lignes de codes.

Exemple :

```
int sum = a + b; // Espace autour de l'opérateur '+'  
for (int i = 0; i < n; i++) { // Espaces autour des opérateurs '<' et '='
```

5.1.1.5. Déclarations de Classe avec *Extends* et *Implements*

- Lorsqu'une classe étend une autre classe ou implémente des interfaces, placez chaque mot-clé (**extends**, **implements**) sur une nouvelle ligne pour une meilleure lisibilité.

Exemple :

```
public class MyArrayList  
    extends ArrayList  
    implements Serializable {  
    // Corps de la classe  
}
```

5.1.1.6. Résumé

Thème	À Faire	À Ne Pas Faire
Indentation	Utilisez 4 espaces pour l'indentation. <code>`if (condition) {` `executerTache();` `}`</code>	Utilisez des tabulations ou un nombre d'espaces incorrect. <code>`if (condition) {` `executerTache();` `}`</code>
Taille des Lignes	Limitez à 80 colonnes. <code>`String message = "Moins de 80 caractères";`</code>	Dépassez 80 colonnes. <code>`String message = "Cette ligne de code est vraiment trop longue et dépasse les 80 caractères";`</code>
Lignes Blanches	Utilisez des lignes blanches pour séparer des sections de code.	Omettez les lignes blanches, rendant le code confus.
Espaces	Ajoutez des espaces autour des opérateurs et après les virgules. <code>`int sum = a + b;` `for (int i = 0; i < n; i++) {`</code>	Omettez les espaces ou en mettez trop. <code>`int sum=a+b;` `for(int i =0;i<n;i++){`</code>
Déclarations de Classe	Utilisez une nouvelle ligne pour <code>`extends`</code> et <code>`implements`</code> . <code>`public class MyArrayList` `extends ArrayList` `implements Serializable {`</code>	Mettez tout sur une seule ligne. <code>`public class MyArrayList extends ArrayList implements Serializable {`</code>

5.1.2. Nommage

5.1.2.1. Général

- Les identifiants doivent être explicites. Évitez les noms trop longs tout en assurant qu'ils soient descriptifs.

5.1.2.2. Package

- Utilisez des noms en minuscules.
- Évitez les noms qui existent déjà dans le JDK.
- Ne pas utiliser d'accents.
- Le nom doit être descriptif de ce que contient le package.

Exemple :

```
package mypackage;
```

5.1.2.3. Classes et Interfaces

- Utilisez la casse CamelCase avec la première lettre de chaque mot en majuscule

Exemple :

```
public class MaClasse { } // Classe  
public interface MonInterface { } // Interface
```

5.1.3. Méthodes

5.1.3.1. Nommage

- Les noms des méthodes doivent commencer par une minuscule et utiliser la casse CamelCase pour les mots suivants.

Exemple :

```
public void maMethode() { } // Nom de méthode
```

5.1.3.2. Déclaration

- Préférez déclarer chaque paramètre sur une ligne séparée lorsque vous avez de nombreux paramètres, pour une meilleure lisibilité.
- Incluez les exceptions potentielles lancées par la méthode dans la déclaration (*exemple : throws IllegalArgumentException*).

Exemple :

```
public void listerNomPersonnel(  
    // Si trop de paramètres, les mettre sur des lignes séparées  
    int premierParam,  
    int secondParam,  
    int troisièmeParam,  
    int quatrièmeParam  
)  
    throws IllegalArgumentException {  
    // Corps de la méthode  
}
```


5.1.3.3. Instructions *Return*

- Utilisez des parenthèses pour les opérations au sein des instructions return pour clarifier la priorité des opérations.

Exemple :

```
public int additionnerEtMultiplier (int a, int b, int c) {  
    return (a + b) * c; // Utilisation de parenthèses  
}
```

5.1.4. Variables

5.1.4.1. Nommage

- Utilisez des noms courts et descriptifs pour les variables.
- Utilisez la casse CamelCase, en commençant par une lettre minuscule.

5.1.4.2. Déclaration

- Déclarez les variables ligne par ligne, sauf pour les variables temporaires itératives qui peuvent être déclarées sur une même ligne.

Exemple :

```
int x; // Déclaration individuelle  
  
int y;  
  
int i, j, k; // Variables itératives sur la même ligne
```

5.1.4.3. Constantes

- Nommez les constantes en majuscules avec des Under scores pour séparer les mots.

5.1.4.4. Types de variables

- Évitez de déclarer des variables de différents types sur la même ligne.

5.1.5. Blocs de code

5.1.5.1. Structures conditionnelles

- Utilisez des accolades même pour les instructions conditionnelles d'une seule ligne pour maintenir la cohérence et faciliter les modifications ultérieures.

Exemple :

```
if (condition) { // Utilisation des accolades même pour une seule ligne
    executerTache();
}
```

5.1.5.2. Boucles

- Suivez les mêmes règles d'indentation et d'espace que pour les autres blocs de code.

Exemple :

```
for (int i = 0; i < n; i++) { // Boucle avec des règles d'indentation standard
    process(i);
}
```

5.1.6. Commentaires

5.1.6.1. Documentation

- Documentez chaque classe, méthode et section importante de code avec des commentaires clairs et concis.

5.1.6.2. JavaDoc

- Utilisez JavaDoc pour générer automatiquement la documentation du code. Incluez une description, les paramètres, les valeurs de retour, et les exceptions pour chaque méthode.

Exemple :

```
/**
 * Cette méthode effectue une tâche spécifique.
 * @param param Le paramètre pour la méthode
 * @return La valeur calculée
 */
public int maMethode(int param) {
    // Corps de la méthode
}
```

5.1.7. Autres conventions spécifiques

5.1.7.1. Saut de ligne

- Effectuez un saut de ligne si une ligne de code dépasse la limite de 80 colonnes pour assurer la lisibilité sur un écran standard.

5.1.7.2. Caractères spéciaux

- Évitez l'utilisation de caractères spéciaux comme le dollar (\$) et le soulignement (_) dans les noms de variables non constantes ou de méthodes.

5.1.7.3. Accents

- Évitez l'utilisation des accents dans le code pour garantir la compatibilité et la lisibilité sur différents systèmes.

5.2. Comment sont-elles vérifiées

Les conventions de codage sont vérifiées à l'aide de l'extension *Checkstyle* présente sur l'IDE Eclipse, que l'on aura configuré selon notre propre convention de codage.

6. Documentation

Vous retrouverez la documentation technique ci jointe au [dossier](#).

7. Principales requêtes de la Base de Données

7.1. Requêtes en SQL

<u>Requête</u>	<u>SQL</u>
	SELECT * FROM Personnel
	INSERT INTO seSpecialise(idP, idS) VALUES (1, 5)
	DELETE FROM seSpecialise WHERE (idP = 1 AND idS = 5)
	SELECT activation FROM ContrainteDeSpecialite WHERE idContrainteSpe = 1
	UPDATE ContrainteDeSpecialite SET activation = (activation = 1) ? 0 : 1
	INSERT INTO dansEDT(idP, idC) VALUES (1, 7)
	DELET FROM dansEDT WHERE (idP = 1 AND idC = 7)

7.2. Requêtes en algèbre relationnel

<u>Requête</u>	<u>SQL</u>	<u>Algèbre relationnel</u>
	SELECT * FROM Personnel	π * (Personnel)
	INSERT INTO seSpecialise(idP, idS) VALUES (1, 5)	INS (seSpecialise, (1, 5))
	DELETE FROM seSpecialise WHERE (idP = 1 AND idS = 5)	DEL (seSpecialise, (idP = 1 AND idS = 5))
	SELECT activation FROM ContrainteDeSpecialite WHERE idContrainteSpe = 1	π activation (σ idContrainteSpe=1 (ContrainteDeSpecialite))
	UPDATE ContrainteDeSpecialite SET activation = (activation = 1) ? 0 : 1	σ activation = 1(ContrainteDeSpecialite) [activation : = 0]
	INSERT INTO dansEDT(idP, idC) VALUES (1, 7)	INS (dansEDT, (1, 7))
	DELET FROM dansEDT WHERE (idP = 1 AND idC = 7)	DEL (dansEDT, (idP = 1 AND idC = 7))

7.3. Requêtes en calcul relationnel

Requête	SQL	Algèbre relationnel	Calcul relationnel
	SELECT * FROM Personnel	$\pi^* (\text{Personnel})$	$\pi^* (\text{Personnel})$
	INSERT INTO seSpecialise(idP, idS) VALUES (1, 5)	INS (seSpecialise, (1, 5))	$\text{seSpecialise} \cup \{(1, 5)\}$
	DELETE FROM seSpecialise WHERE (idP = 1 AND idS = 5)	DEL (seSpecialise, (idP = 1 AND idS = 5))	$\pi \text{idP, idS} (\text{seSpecialise}) - \pi \text{idP, idS} (\text{seSpecialise} \cap \{(1, 5)\})$
	SELECT activation FROM ContrainteDeSpecialite WHERE idContrainteSpe = 1	$\pi \text{activation} (\sigma \text{idContrainteSpe}=1 (\text{ContrainteDeSpecialite}))$	$\sigma \text{idContrainteSpe} = 1 (\text{ContrainteDeSpecialite}). \text{activation}$
	UPDATE ContrainteDeSpecialite SET activation = (activation = 1) ? 0 : 1	$\sigma \text{activation} = 1 (\text{ContrainteDeSpecialite}) [\text{activation} := 0]$	$\sigma \text{activation} = 1 (\text{ContrainteDeSpecialite}) \bowtie \pi \text{activation} (\text{ContrainteDeSpecialite} \bowtie \{(activation, 0)\})$
	INSERT INTO dansEDT(idP, idC) VALUES (1, 7)	INS (dansEDT, (1, 7))	$\text{dansEDT} \cup \{(1, 7)\}$
	DELET FROM dansEDT WHERE (idP = 1 AND idC = 7)	DEL (dansEDT, (idP = 1 AND idC = 7))	$\pi \text{idP, idC} (\text{dansEDT}) - \pi \text{idP, idC} (\text{dansEDT} \cap \{(1, 7)\})$