



## Planning Hospitalier

### Rapport du Jalon 3

Antivackis Vanessa

Forest Jules

Bennama Meriem

Haton Tom

Cailleau Dylan

Mohammed Arwa

1.	Backlog .....	3
a.	Résumé .....	3
b.	Ce qui a été réalisés .....	6
c.	Ce qui reste à faire .....	7
d.	Commentaire .....	8
2.	Tests et résultats .....	8
3.	Taux de couverture .....	9
4.	Bilan des sprints .....	9
a.	Bilan Générale .....	9
b.	Temps passé par chacun sur les différentes parties .....	10
5.	Diagramme de classe .....	11
a.	Niveau métier .....	11
b.	Niveau conception .....	12
6.	Modèle Conceptuel de Données .....	12
7.	Codes sources et Dump de la base .....	12

## 1. Backlog

### a. Résumé

Lors du précédent jalon, nous vous avons présentés ceci :

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

#### 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)*

Regardons maintenant ce qui a été réalisé et ce qui reste à faire.

#### b. Ce qui a été réalisés

*La légende des couleurs reste la même.*

- 1. En tant que cadre je veux pouvoir créer un EDT avec différents créneaux
- 2. En tant que cadre je veux pouvoir consulter l'EDT de façon globale et mensuelle
  - 2.1. Gestion des créneaux problématiques avec des couleurs  
(partiellement réalisée)
- 3. En tant que cadre je veux pouvoir consulter les EDT par personnel
- 4. En tant que cadre je veux pouvoir consulter les contraintes légales
  - 4.1. Récupération des contraintes légales dans la BDD
  - 4.2. Fenêtre de récap des contraintes légales
  - 4.3. Ajout d'une contrainte
    - 4.3.1. Pop-up de saisie

- 4.3.2. Pop-up de confirmation ajout
  - 4.4. Modification d'une contrainte
    - 4.4.1. Pop-up de modification
    - 4.4.2. Pop-up de confirmation modification
  - 4.5. Suppression d'une contrainte
    - 4.5.1. Pop-up demande de validation suppression
    - 4.5.2. Pop-up confirmation de suppression
- 5. En tant que cadre je veux pouvoir consulter les contraintes de spécialité
  - 5.1. Récupération des contraintes de spécialités dans la BDD
  - 5.2. Fenêtre de récap des contraintes de spécialité
  - 5.3. Ajout d'une contrainte de spécialité
    - 5.3.1. Pop-up de saisie
    - 5.3.2. Pop-up de confirmation ajout
  - 5.4. Modification d'une contrainte de spécialité
    - 5.4.1. Pop-up de saisie
    - 5.4.2. Pop-up de confirmation modification
  - 5.5. Suppression d'une contrainte de spécialité
    - 5.5.1. Pop-up demande de validation suppression
    - 5.5.2. Pop-up confirmation de suppression

### c. Ce qui reste à faire

*La légende des couleurs reste la même.*

- 1. En tant que cadre je veux pouvoir consulter l'EDT de façon globale et mensuelle
  - 1.1. Imprimer l'EDT (BONUS)
  - 1.2. Gestion des créneaux problématiques avec des couleurs  
(partiellement réalisée)
- 2. En tant que cadre je veux pouvoir consulter les EDT par personnel
  - 2.1. Imprimer l'EDT
- 3. Je veux pouvoir dupliquer un créneau (BONUS)
- 4. Je veux pouvoir dupliquer une semaine type (BONUS)
- 5. Je veux pouvoir intégrer une prise de congés (BONUS)

## d. Commentaire

L'objectif principale de ce projet était de réaliser un emploi du temps à l'aide d'une connexion à une base de données, de java et de javafx pour la représentation.

L'idée initiale, la vision initiale que nous avons eu pour ce projet (représentation via [Figma](#)), le rendu final n'était pas ce dont nous souhaitions initialement.

## 2. Tests et résultats

Etudions maintenant les tests et leur résultat. Pour cela, nous nous baserons sur la classe `TestContrainteDeSpecialite.java` ainsi que `TestContrainte.java`.

Ces classes possèdent un bon nombre de tests qui sont, selon leur logique, communs à chacune des classes métiers créées.

Six (6) tests furent effectuées :

Test	Description	Résultat
TestIsActivated()	Cette méthode vérifie le renvoie de l'état d'activation d'une contrainte de spécialité.	1 ou 0, selon son activation ou non.
TestSwitchActivation()	Cette méthode vérifie le changement de l'état d'activation en son opposé d'une contrainte de spécialité.	1, si activation précédente égale à 0. 0, si activation précédente égale à 1.
TestIdContrainteSpe()	Cette méthode vérifie le renvoie de l'identifiant de la contrainte de spécialité.	Integer
TestSetIdContrainteSpe()	Cette méthode vérifie que la modification de l'identifiant de la contrainte de spécialité ait bien été effectuée.	void
TestGetSpecialite()	Cette méthode vérifie le renvoie d'une spécialité associée à la contrainte de spécialité.	Spécialité(int id, String nom)
TestSetSpecialite()	Cette méthode vérifie que la modification de l'identifiant et du nom de la spécialité ait bien été effectuée.	void
TestGetNom()	Cette méthode vérifie le renvoie du nom de la contrainte.	String nom



TestSetNom()	Cette méthode vérifie que la modification du nom de la contrainte ait bien été effectuée.	void
TestGetDuree()	Cette méthode vérifie le renvoie de la durée associée à une contrainte.	Double duree
TestSetDuree()	Cette méthode vérifie que la modification de la durée associée à une contrainte ait bien été effectuée.	void
TestGetTemporalite()	Cette méthode vérifie le renvoie d'une temporalité associée à une contrainte.	String temporalite
TestSetTemporalite()	Cette méthode vérifie que la modification de la temporalité ait bien été effectuée.	void

### 3. Taux de couverture

Pour les classes `TestPersonnel`, `TestContainte`, `TestContrainteLegale`, `TestContrainteDeSpecialite`, les taux de couverture sont de 100%.

Cela signifie que pour toutes les méthodes testées, les résultats furent ceux attendus.

La vérification de ces derniers fut effectuée à l'aide JUnit 5.

### 4. Bilan des sprints

#### a. Bilan Générale

Les sprints furent pour la plupart, chronophages et compliqués.

Au début du projet, au moment où nous avons tous individuellement commencé à coder, nous n'avions pas de soucis majeurs pour tester ou bien mettre à jour nos versions.

Or, plus les jours passaient, plus le git commençait à s'élargir, plus les remises à niveau/version furent difficile et causèrent de nombreux problèmes.

L'étape d'apprentissage concernant javaFX a été une difficulté pour tous.

Vous trouverez joint au rapport, un document intitulé "Résumés des réunions", entretenus par Vanessa Antivackis, qui résumé à la fois l'avancée mais aussi les tâches que chacun avait à effectuer.

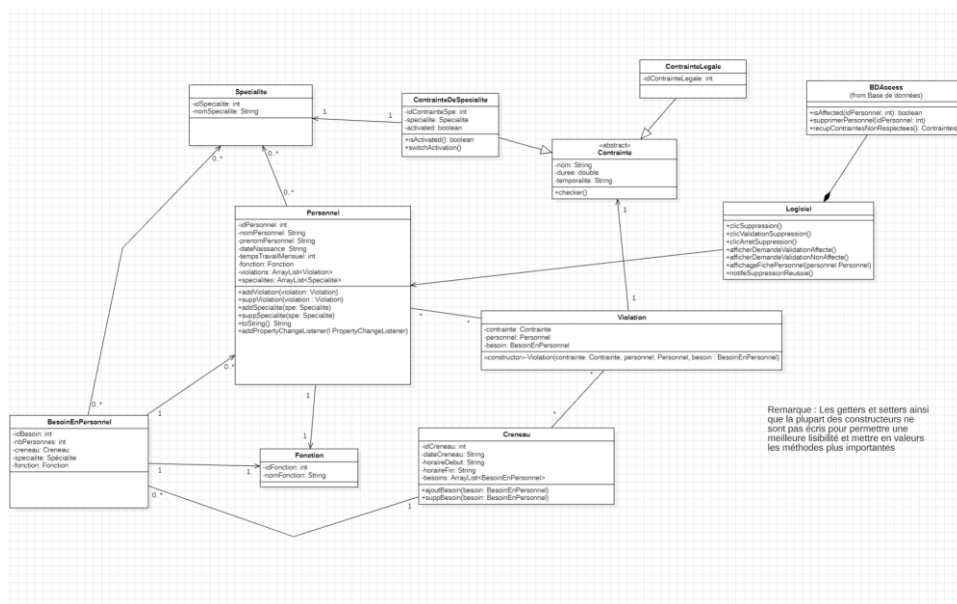
b. Temps passé par chacun sur les différentes parties

Parties	Antivackis Vanessa	Bename Meriem	Cailleau Dylan	Forest Jules	Haton Tom	Mohammed Arwa
Jalon 1 : Tâches effectués   Temps passé	Backlog  MCD  Rapports  Scénario UML  Correction  Liste de contraintes  Temps : 30h	MockUp (Figma)  Temps : 30h	Diagrammes UML  Apprentissage JavaFX  Apprentissage SceneBuilder  Temps : 15h	Backlog  Diagrammes UML  Correction  Temps : 30h	Backlog  Cas d'utilisation  Diagramme UML  Temps : 20h	Cas d'utilisation  Diagramme UML  Temps : 20h
Jalon 2 : Tâches effectuées   Temps passé	Convention de codage Backlog  Documentation Java  Traduction requêtes  Temps : 25h	Convention de codage  Rédaction Convention de codage  Temps : 15h	Convention de codage  Planification : Diagramme de GANTT et PERT, Planning du code  Temps : 20h	Convention de codage  Backlog  Diagramme de conception  UML final Temps 20h	Convention de codage  Diagramme de conception  Traduction requêtes  Temps 20h	Convention de codage Diagramme de conception (grosse difficultés)  Temps : 50h
Jalon 3 : Tâches effectuées   Temps passé	Classes : Contrainte,  ContrainteDe Spécialité,  ContrainteLégale,  ContrainteDe SpécialitéDAO,  ContrainteLégaleDAO, TestContrainte, TestContrainteDeSpécialité,	Base de données  Classes : Fonction ViolationDAO FonctionDAO TestFonction  Fichier FXML correspondant et leur contrôleur  Temps : 90h	Classes : Creneau CreneauDAO TestCreneau x DAO SingleConnexion  Fichier FXML correspondant et leur contrôleur  Temps : 115h	UML  Git (Maintenance)  Classes : Personnel TestPersonnel TestPersonnelDAO PersonnelDAO  DAO SingleConnexion Fichier FXML correspondant	Classes : Violation(Méti) ViolationDAO  Temps : 40h	Classes : Spécialité SpécialitéDAO  Fichier FXML correspondant et leur contrôleur  Temps : 80h

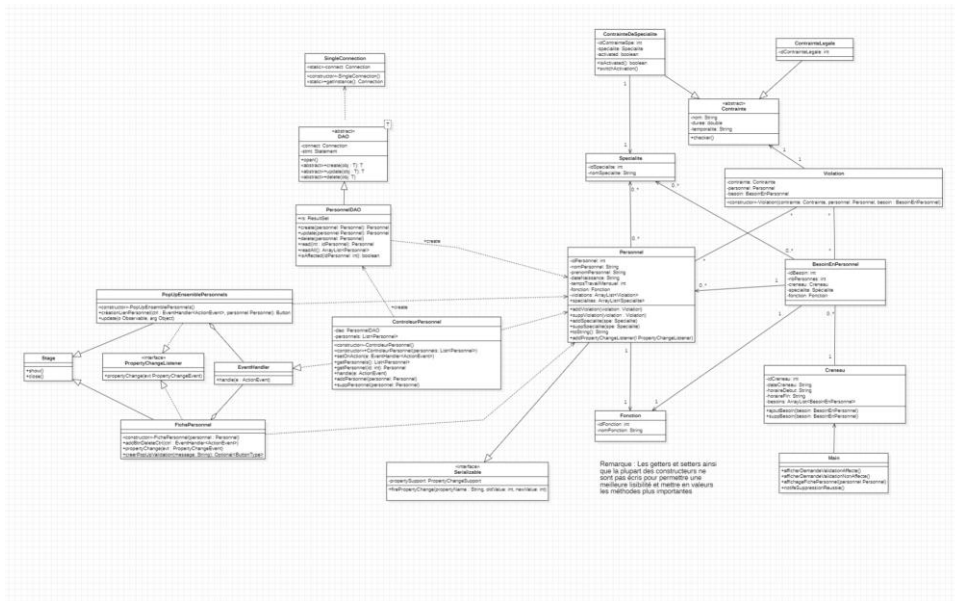
	TestContrain teLegale, TesContraint eDeSpecialit eDAO, TestContrain teLegaleDAO DAO SingleConnec tion  Fichier FXML corresponda nt et leur controleur  Temps : 100h			nt et leur controleur  Temps : 90h		
--	--	--	--	---	--	--

## 5. Diagramme de classe

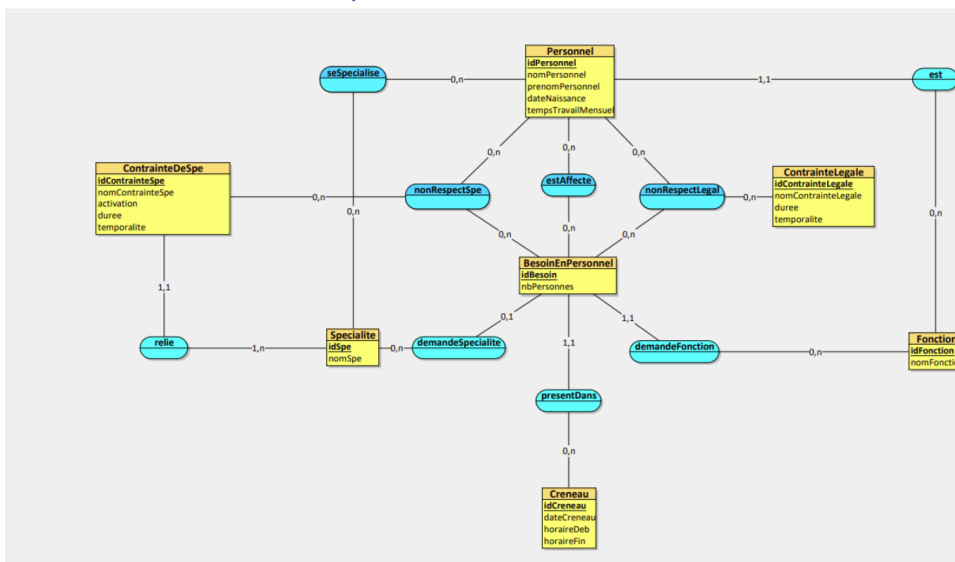
### a. Niveau métier



## b. Niveau conception



## 6. Modèle Conceptuel de Données



## 7. Codes sources et Dump de la base

### a. Codes sources et Dump de la base

Vous les retrouverez joint au zip.

### b. Configuration du projet

Vous trouverez joint au rapport 2 vidéos démontrant l'étendue de notre projet mais aussi leur configuration, réalisé par Dylan Cailleau.

Voici la configuration par écrit :

- Installer l'extension e(fx)clipse sur Eclipse IDE en allant dans Help > eclipse marketplace
- Installer les fichiers JavaFX depuis le site officiel
- Télécharger via le fichier .lib du projet les fichiers suivants : controlsfx, fontawesomefx, jfoenix et mysql-connector
- Configurer le module path de tel manière à ce qu'il est tous les fichiers de javaFX, ajouter la librairie JvaFX SDK (donné par l'extension), rajouter les fichiers téléchargés via le .lib grâce à l'option Add External JARS et dans Classpath mettre JUnit 5
- Configurezrvotre run en allant dans run configuration..., allez dans Arguments et dans la section VM arguments mettez --module-path " " --add-modules javafx.controls,javafx.graphics,javafx.fxml, en faisant attention de bien mettre entre les guillemets, le chemin du dossier qui contient vos fichiers JavaFXX
- Récupérer le fichier planninghopital.sql dans le package database du projet
- Ouvrir votre Uwamp/Xamp et lancez votre serveur
- Aller dans phpmyadmin, et créé une nouvelle database du nom planninghospitalier, pour ensuite aller dans importe et importer le fichier planninghospitalier.sql
- Après cela allez dans le package dao > singleConection.java et dans la méthodegetInstance() changez les arguments afin de mettre votre utilisateur et votre mot de passe phpmyadmin :  
new SingleConnection("127.0.0.1", "planninghospitalier", "VOTRE\_UTILISATEUR", "VOTRE\_MOT\_DE\_PASSE");