



16/08/2022

ECF JAVA

Développer une interface
utilisateur de type Desktop



Boebion Jérôme
AFPA - CDA 21097

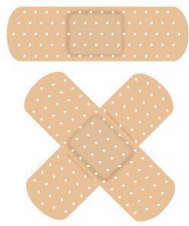
Critères d'évaluation

Les compétences suivantes seront évaluées :

- Le maquettage
- Le diagramme de classes des classes métiers
- L'algorithme
- Les bases du langage JAVA
- Les concepts de la programmation Orientée Objet
- La documentation et les commentaires
- Les collections Java
- Les exceptions
- Les tests sur les classes métiers

Objectifs de qualités recherchés :

- La Javadoc de votre application doit être mise en place.
- Les tests unitaires de votre application doivent être présents.
- Les différents traitements des exceptions doivent être présents.
- La qualité de code demandé pour tous développements doit être présente.
- L'utilisation de Git est demandée.



CAHIER DES CHARGES

La pharmacie **Sparadrap** souhaite mettre en place un outil interne de gestion et suivi des achats de leurs clients. Le client se présente au guichet de la pharmacie pour un achat direct ou au travers d'une ordonnance.

Le pharmacien effectuera donc un enregistrement de cet achat en fonction du client en enregistrant toutes les informations nécessaires au bon suivi. Elle pourra alors afficher tous les achats effectués de la journée. Elle pourra également afficher toutes les ordonnances prescrites par un médecin.

Un client est représenté par son nom, prénom, adresse, téléphone, email et son numéro de sécurité social, sa date de naissance et sa mutuelle. Il a également un médecin traitant et éventuellement un ou plusieurs(s) spécialiste(s).

Un médecin est représenté par son nom, prénom, adresse, téléphone, email et son numéro d'agrément. Il peut avoir un ou plusieurs patients.

Un spécialiste est représenté par son nom, prénom, adresse, téléphone, email et sa spécialité (exemple)

Un médicament est représenté par son nom, sa catégorie (exemple) , son prix et sa date de mise en service et une quantité.

Une mutuelle (exemple) est représentée par son nom, son adresse, son téléphone, son email, son département et le taux de prise en charge du remboursement des médicaments.

Une ordonnance est représenté par une date, le nom du médecin traitant, le nom du patient, la liste des médicaments et éventuellement le nom du spécialiste s'il y a lieu.

Note : Pour simplifier, on ne prendra que les catégories principaux soit Analgésiques, Anti-inflammatoires, Antiviraux etc.... Tout comme les mutuelles et les spécialités, il n'est pas besoin de toutes les prendre.

Note : Ici, on imagine la saisie de l'ordonnance dans l'application comme si le pharmacien l'avait entre ses mains au moment de la saisie.

Note : L'objectif est de simuler une application. On peut au départ partir du principe qu'au lancement de l'application, vous instanciez vos données qui seront utilisées pour la simulation.

1^{er} Priorité / 2nd Priorité / 3^{ème} Priorité

- **Sur la page d'accueil :**
 - L'utilisateur devra choisir entre effectuer un achat et consulter l'historique des achats, consulter l'historique des ordonnances pour un Médecin, consulter le détail d'un client.
 - L'utilisateur pourra quitter l'application ou revenir au choix précédent à tout moment.
 - Si l'utilisateur choisit l'historique des ordonnances alors une liste déroulante s'affichera pour lui permettre de choisir le médecin avant de lui afficher l'historique.
 - Si l'utilisateur choisit de consulter le détails d'un client, une liste déroulante s'affichera pour lui permettre de choisir le client avant de lui afficher son détails.
- **Sur la page achat :**
 - L'utilisateur pourra saisir un achat direct ou un achat via une ordonnance. Un label lui indiquera alors quel type d'achat il effectuera à ce moment précis.
 - L'utilisateur pourra saisir toutes les informations nécessaires à l'achat et à l'historique en fonction du type d'achat. A la validation, l'achat sera alors historisé.
 - L'utilisateur pourra revenir à la page précédente à tout moment ou quitter l'application.
- **Sur la page historique des achats :**
 - L'utilisateur pourra afficher la liste des achats effectués pendant la journée.
 - L'utilisateur pourra consulter les informations pertinentes de ces achats.
 - L'utilisateur pourra revenir à la page précédente à tout moment ou quitter l'application.
- **Sur la page historique des ordonnances :**
 - L'utilisateur pourra consulter la liste des ordonnances écrites par un médecin.
 - L'utilisateur pourra sélectionner une ordonnance et consulter son détail.
 - L'utilisateur pourra revenir à la page précédente à tout moment ou quitter l'application.
- **Sur la page détails des ordonnances :**
 - L'utilisateur pourra consulter tous les détails de cette ordonnance.
 - L'utilisateur pourra revenir à la page précédente à tout moment ou quitter l'application.
- **Sur la page détails d'un client :**
 - L'utilisateur pourra simplement consulter le détails des informations d'un client.
 - L'utilisateur pourra revenir à la page précédente à tout moment ou quitter l'application.

Spécifications techniques

- Le développement se fera en Java avec la bibliothèque graphique Swing
- L'affichage des listes se fera avec une JTable
- Une Javadoc sera fournie
- La gestion des données se fera à l'aide de Collections Java

Spécifications fonctionnelles

- L'analyse UML devrait vous permettre d'identifier les classes à créer dans votre future application.
- Une correction individuelle sera faite après les envois de vos diagrammes de classes.

Pour les plus rapides ou ceux qui veulent encore aller plus loin (ne fera pas partie des critères d'évaluation) :

- *En cas d'ordonnance, la pharmacie souhaite une conservation au format pdf dans l'historique du client.*
- *La pharmacie souhaiterait un module complémentaire de recherche des mutuelles pour afficher tous les patients par mutuelle.*
- *La pharmacie souhaite créer un historique de l'évolution du prix des médicaments dans le temps.*
- *La pharmacie souhaite une gestion minime du stock de ses médicaments.*
- *Comme vous n'avez pas traité la persistance des données. On peut imaginer plusieurs fichiers contenant les données que l'application ira chercher au démarrage pour charger vos objets et de conserver les données après avoir quitté l'application.*

TRAVAIL DEMANDE :

Dans le cadre de cet ECF JAVA, il vous est demandé de réfléchir à la création de cette application soit :

- Une analyse diagramme de classes de votre application.
- Une analyse fonctionnelle de votre application.
- Un maquettage des différentes fenêtres de votre futur application avec l'utilitaire de votre choix.

Ces analyses et ce maquettage devront être réalisés en amont de votre développement et me seront envoyés par mail à jerome.boebion@maformationenligne.bzh avant la fin de votre première semaine d'ECF (date limite du rendu 19 août 2022).

Note :

Si vous avez des questions pendant la période de l'ECF, vous devrez passer soit par email, soit par discord. Si une question dont la réponse peut concerner l'ensemble du groupe, je le ferai au travers du discord (ecf-java). Sinon, je répondrai individuellement.

Conseils :

Ne perdez pas trop de temps sur les détails de vos frames, par exemple, une couleur d'un bouton, ou une image etc... ce n'est pas un ECF basé sur le design !!

Construisez votre application étape par étape. L'analyse fonctionnelle et l'analyse diagramme de classes sont importantes et vous permettent de mettre en place votre futur développement.

Le travail individuel est préconisé mais je n'interdis pas la réflexion en groupe...

Cette période ECF JAVA se déroule : Du 16 août au 2 septembre 2023.

Votre code doit être rendu : Pour le 5 septembre 2023 9h avec ajout à votre projet Git à cette date.