

INF5153 - Génie logiciel : Conception

Projet de session - HIVER 2024

Ce projet doit être réalisé en équipe de 3 ou 4 personnes. Idéalement, les équipes devraient rester les mêmes durant toute la session.

Dossier médical centralisé

Vous devez concevoir un logiciel permettant aux utilisateurs du système de santé québécois d'avoir un dossier médical centralisé, peu importe le médecin ou l'établissement qu'ils fréquentent.

Contexte

Actuellement, chaque bureau de médecin et chaque hôpital possède son propre dossier sur un patient. Dans la majorité des cas, ces dossiers ne sont jamais mis en commun, ne permettant pas à un médecin d'avoir une idée globale de l'état de santé d'un patient.

Le système que vous devez concevoir est une solution à ce problème. Le dossier médical d'un patient sera centralisé. À chaque fois qu'un patient visite un médecin, le médecin aura accès à l'ensemble du dossier du patient, incluant son historique de visites, les autres médecins traitant le patient, les diagnostics connus et ainsi de suite.



Le dossier médical

Le dossier médical d'un patient contient les informations de base d'identification et les coordonnées du patient :

- nom et prénom;
- date de naissance;
- genre;
- identification des parents connus;
- ville de naissance;
- coordonnées (adresse résidentielle, téléphones, courriel);
- numéro d'assurance maladie.

Le dossier médical peut contenir une liste d'antécédents médicaux. Chaque antécédent contient :

- le diagnostic;
- le traitement;
- le médecin traitant;
- le début et la fin de la maladie, si applicable.

Le dossier médical contient une liste des visites médicales d'un patient. Chaque visite contient :

- l'établissement visité;
- le médecin vu;
- la date de la visite;
- le diagnostic établi, si applicable;
- le traitement, si applicable;
- un résumé de la visite par le médecin;
- des notes pour les autres médecins.

Interfaces

Les médecins et les professionnels de la santé auront une installation d'un logiciel de gestion de dossiers médicaux sur un poste de travail (laptop ou ordinateur de bureau). L'ordinateur doit être connecté à un réseau d'un établissement médical pour avoir accès aux données sur les patients.

Les patients, au travers d'un site web, peuvent voir les données de leur dossier médical.

Une application mobile (android et iOS) est également disponible pour permettre aux patients de consulter leur dossier.

Fonctionnalités

En plus de la gestion du dossier médical, voici quelques fonctionnalités et exigences à considérer :

- Le patient ne peut rien modifier dans son dossier, sauf ses coordonnées.
- Le dossier d'un patient est créé automatiquement pour toutes les personnes inscrites à la RAMQ.
- Chaque modification faite par un médecin est automatiquement sauvegardée au dossier, sans que le médecin n'ait explicitement à sauvegarder les modifications.
- Un médecin peut, en tout temps, annuler les modifications qu'il a apporté à un dossier.
- Plusieurs médecins peuvent apporter des modifications à un même dossier en même temps sans conflit de concurrence.
- Chaque modification faite au dossier doit être archivée. En tout temps, la RAMQ peut reconstruire un dossier complet, tel qu'il était à cette époque, à partir d'une modification précise faite dans le passé. La RAMQ peut également reconstruire un dossier à une date donnée.
- La carte d'assurance-maladie du patient deviendra la façon d'accéder au dossier du patient. La carte sera munie d'une puce électronique qui émettra un code lorsque présentée devant l'ordinateur d'un médecin. Avec ce code, le dossier du patient pourra être téléchargé afin d'être consulté par le médecin. La carte d'assurance-maladie peut aussi être présentée à un professionnel de la santé (non médecin) pour un accès en lecture seule du dossier.

Partie 1 : présentation les 13 et 14 mars

Vous devez concevoir et présenter votre conception en partant de la conception architecturale à la conception détaillée . Vous devez préparer un rapport détaillant la conception projetée du système. Le rapport doit contenir :

- Le diagramme de votre conception architecturale illustrant les différents composants du système et leurs interactions.
- le(s) diagramme(s) de cas d'utilisation modélisant les fonctionnalités du système;
- le(s) diagramme(s) de classes illustrant les concepts du domaine;
- les diagrammes de séquence montrant les interactions entre les classes pour chaque cas d'utilisation;
- les diagrammes de packages du système (en version détaillée, c'est-à-dire en indiquant les classes contenues dans les packages);
- le diagramme de composants;
- le diagramme de déploiement du système.
- chaque responsabilité des classes doit être justifiée par un pattern GRASP; cette justification doit être documentée (une justification par méthode publique; une justification pour la création des instances d'une classe).

Le rapport doit contenir une page de présentation et une table des matières. Il doit également contenir le texte nécessaire pour introduire la modélisation et ce qui est présenté dans chaque section du document.

Le rapport doit être un fichier markdown nommé **partie1.md**. Le rapport et le code (PlantUml) des diagrammes doivent être disponible à la racine de votre projet dans la branche principale avant le **17 mars 2024 à 23h55**.

Partie 2 : présentation les 17 et 18 avril - remise le 21 avril 2024

Vous devez modifier votre conception détaillée afin d'y introduire des patrons de conception. Le nombre de patrons à introduire correspond au nombre de membres dans l'équipe (p-ex. 3 patrons pour une équipe de 3 personnes).

Pour chaque patron, vous devez fournir un diagramme de classe et un (ou plusieurs, si nécessaire) diagramme de séquence illustrant les relations entre les classes du patron et les autres classes du projet.

Vous devez également développer l'application dans le but de valider votre conception. Vous devez développer l'application des médecins. Toute différence entre l'implémentation et la conception doit être documentée dans le rapport que vous allez me remettre. Pour chaque problème de conception que vous allez corriger, vous devrez décrire comment vous avez détecté le problème de conception et comment vous l'avez corrigé.

La base de données utilisée dans le projet doit être une base de données SQLite3. La modélisation de la base de données est à votre discrétion. Un fichier SQL doit être remis contenant le script pour créer les tables avec des données de test. La documentation de votre projet (README) doit indiquer comment créer la base de données, dans quel répertoire du projet elle doit être placée, comment compiler et exécuter votre logiciel.

Le code source du projet, le script de création de la base de données et le rapport nommé *partie2.md*, doivent être remis au plus tard le **21 avril 2024 à 23h55**. Seule la branche principale du projet sera évaluée.

Contraintes technologiques

Voici les contraintes que vous devez respecter :

- Le logiciel doit être développé avec le langage de programmation Java (JDK17 ou plus).
- Vous devez utiliser Maven pour automatiser la construction du logiciel et la gestion des dépendances.
- Il est conseillé d'utiliser l'environnement de développement intégré IntelliJ.
- JUnit5 sera utilisé comme cadre de test.
- Il est impératif d'utiliser SQLite3 pour la gestion de votre base de données.
- Les sources doivent être entreposées dans un dépôt GIT sous le GitLab du département d'informatique.
- Les fichiers utilisés doivent être en UTF-8.
- Il est impératif d'utiliser PlantUML pour la création des différents diagrammes.
- Le choix du framework pour vos interfaces clientes (frontend) est laissé à votre appréciation. La qualité des interfaces sera évaluée.

Autres exigences

- Toutes les dates doivent être spécifiées en format ISO 8601 (AAAA-MM-JJ).
- Vous devez rédiger suffisamment de «bons» tests unitaires afin d'acquiescer une couverture de tests d'au moins 80% sur tout votre projet.
- Chaque commit doit être atomique et documenté à l'aide d'un message de commit décrivant la tâche qui y est accomplie. Ne pas utiliser de sociolecte comme le langage SMS
- Vous devez avoir des branches par fonctionnalité et initier des *merge request* pour envoyer le contenu à évaluer vers votre branche principale (qui sera la seule à être évaluée).
- Votre logiciel doit disposer d'un système d'intégration continue (GitLab-CI) pour garantir sa non-régression.

Composition de l'équipe

Vous devez identifier chaque membre de l'équipe dans un fichier *equipe.md*. Pour chaque membre, vous devez fournir l'information suivante :

- nom et prénom;
- code permanent;
- adresse courriel;
- le meilleur moyen de communication pour rejoindre cette personne (ex. courriel, texto, téléphone) et fournir l'information pour joindre la personne;
- les disponibilités (à quels moments cette personne peut être jointe);
- une description des forces, faiblesses et compétences techniques de la personne;
- Le fichier *equipe.md* doit préciser le chef de l'équipe (votre représentant avec l'équipe enseignante) et être disponible à la racine de votre projet.

Charte de l'équipe

La charte de l'équipe est une entente entre les membres de l'équipe sur la façon dont vous allez travailler durant le projet de session. La charte doit décrire :

- comment vous allez répartir les tâches dans l'équipe;
- la méthode de travail que vous prévoyez utiliser;
- la méthode de résolution de conflit que vous prévoyez utiliser;
- les sanctions prévues pour un membre de l'équipe qui ne remplit pas ses engagements;
- votre définition de "*Terminé*".

Remise :

- Vous devez créer un dépôt **privé** nommer exactement *equipe#-inf5153-h2024-projet* (où # représente le numéro de l'équipe) sur le GitLab de l'UQAM.
- Les usagers *@correcteur* et *@dogny_g* doivent avoir accès à votre projet comme *Developer*.
- Le non respect d'une des exigences du projet entrainera automatiquement une pénalité de 50%

Barème partie 1 :

Critères	Points
Qualité de la présentation (diapos, durée, réponses, ...)	/15
Justesse & Pertinence de la conception	/20
Respect des principes de conception	/20
Justification des choix de conception	/20
Qualité des diagrammes PlantUML et de la documentation	/25
Total	/100

Barème partie 2 :

Critères	Points
Qualité de la présentation (diapos, durée, réponses, ...)	/15
Justesse & Pertinence de la conception ou justification des ajustements	/10
Justification des choix de patrons de conception	/15
Qualité du code Java, du dépôt Git, des tests et de la documentation	/10
Intégration continue et qualité des tests	/10
Fonctionnalité	/40
Total	/100

Note : Toute équipe qui ne présentera pas son travail lors des séances de présentation se verra attribuer la note de zéro pour la partie du projet concernée.