

A thick dark blue vertical bar runs down the left side of the page. A teal arrow points to the right from this bar, containing the text 'MIF01'. At the bottom left, there are several thin, curved lines in dark blue and light grey.

MIF01

Rapport Projet : Mon espace santé

Gestion de projet et génie logiciel
2022

Yann VINCENT p1906701 Iliesse LARIBI p1911241

Introduction :

Le but du projet est de réaliser une petite application en Java capable de gérer plusieurs patients et professionnels de santé qui peuvent interagir entre eux par le biais de prescriptions. C'est une représentation simpliste du site internet monespacesante.fr. Les réels objectifs de ce TP sont divisés en trois catégories :

- Bien organiser le code de son application. On utilise les principes MVC et 3 designs différents.
- Prendre du recul sur ce que l'on code et analyser avec des enjeux humains en tête afin de réaliser une application éthique.
- Organiser la conception de son projet de la bonne manière en réalisant des tests pertinents pour vérifier le bon fonctionnement de l'application. Chacun de ces objectifs va être détaillé dans la suite du rapport.

Table des matières

Introduction :	1
Design Pattern MVC :	2
MVC :	2
Design-Pattern :	2
Observer :	2
Strategy :	3
Builder :	4
Ethique :	5
Tests :	7
Reference :	9

Design Pattern MVC :

Dans cette partie, il est question de développer la façon dont nous avons décidé de coder ce projet ainsi que nos différents choix d'implémentation.

MVC :

Notre application est codée sous la forme MVC : Modèle-Vue-Contrôleur. Cette façon de faire permet de diviser les projets en ces trois grandes catégories. Le modèle s'occupe de regrouper toutes les fonctions comme la création du patient ou les interactions comme l'ajout de prescription. C'est dans cette partie qu'on initialisera grâce à un fichier yml les conditions initiales de l'application (création des patients, des professionnels de santé, des prescriptions). La vue qui consiste à gérer l'affichage de l'application grâce à la bibliothèque Java Fx. Elle se charge d'afficher les données mais aussi de récupérer les actions de l'utilisateur. L'ajout d'un patient pas l'interface ou l'ajout de prescription d'un médecin après avoir recherché l'utilisateur etc. Le contrôleur est comme le chef d'orchestre ou le cerveau de l'application, il récupère les données du modèle et les envoie à l'interface pour l'afficher, il récupère les données et les actions de l'interface pour les faire s'exécuter par le modèle. Cette façon d'organiser l'architecture de son code permet de diviser les tâches et de faire une distinction entre chaque domaine.

Design-Pattern :

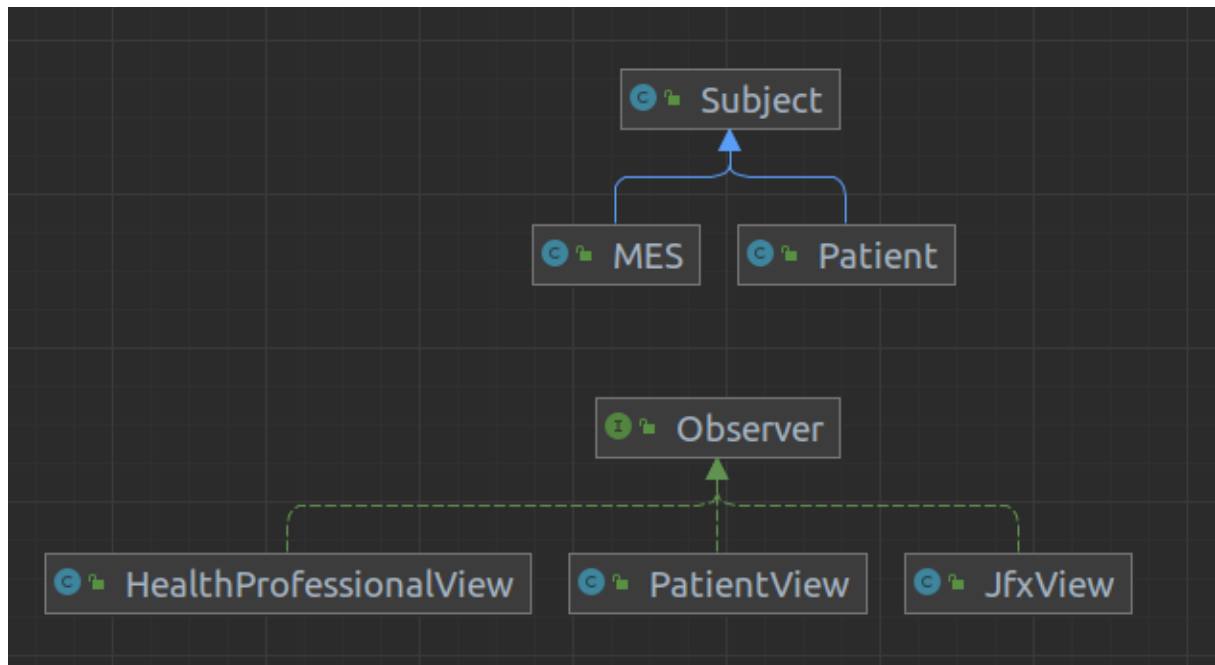
Afin d'améliorer encore l'architecture de notre code nous avons choisi 3 designs pattern qui fonctionnaient avec la façon dont on imaginait notre application.

Observer :

Après avoir restructuré notre code avec MVC nous avons décidé de dynamiser l'application. En l'état après avoir ajouté une prescription il fallait appuyer sur un bouton « RELOAD » afin de voir apparaître les nouvelles données. De même si plusieurs fenêtres de l'application étaient ouvertes elle ne s'actualisait pas d'elle-même. Pour remédier à ceci nous avons décidé d'utiliser le pattern observer. Le principe d'observer est simple, dès qu'une action qui nécessite une mise à jour graphique est effectuée on notifie l'observer et il met à jour l'interface. On a créé 2 classes. La classe Subject qui regroupe toutes les fonctions dont aura besoin : `notifyObserver()` qui demande à tous les observer de faire un `update()` donc de mettre à jour leur interface graphique, `register()` qui ajoute un observer à la liste des observer pour gérer plusieurs fenêtres par exemple (un observer par fenêtre). Et l'interface observer qui permet d'avoir une `update()` différente pour chaque classe qui l'utilise (par exemple une `update()` pour afficher les

prescriptions côté professionnel de santé après avoir recherché un patient ou un update pour afficher le patient ajouté sur toutes les vues.).

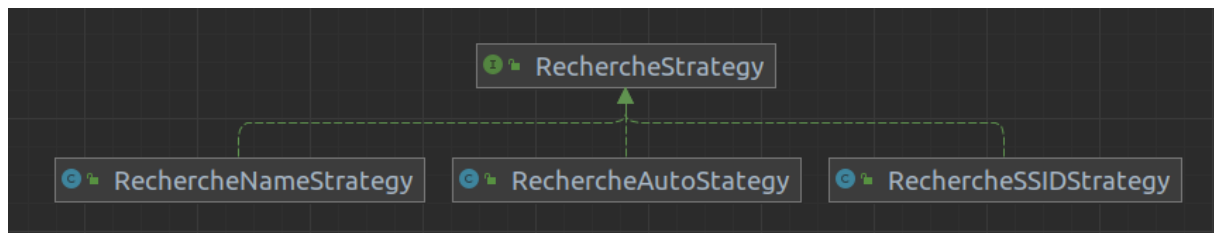
En résumé le pattern Observer nous permet de mettre à jour facilement l’affichage sans passer par un bouton. Il nous permet aussi d’avoir plusieurs instances de l’application ouverte qui s’actualise entre elles.



Strategy :

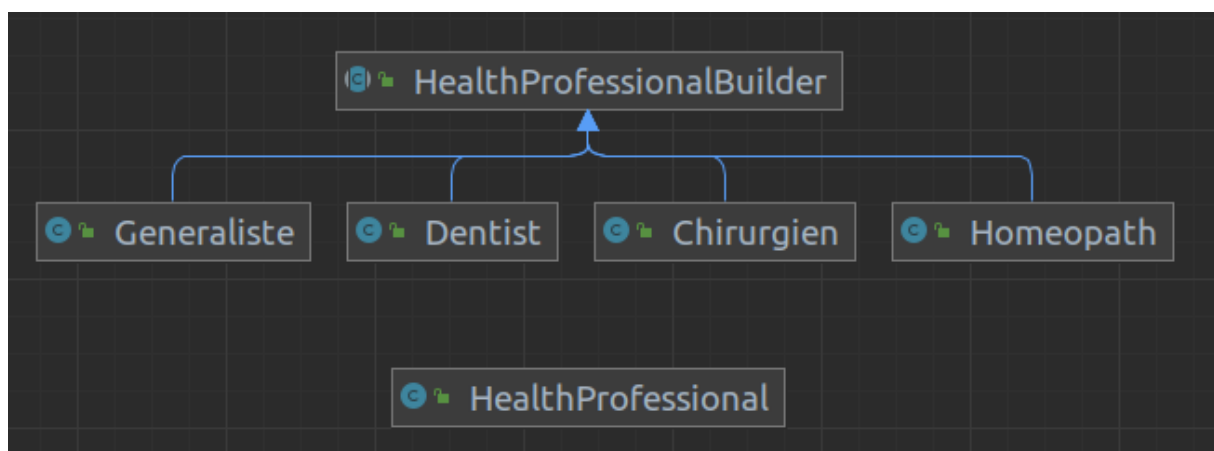
Afin d’améliorer notre application nous avons ajouté une extension de recherche pour l’ajout de prescription. Lorsqu’un professionnel de santé souhaite ajouter une prescription il doit d’abord choisir dans un menu s’il préfère renseigner le nom ou le ssId (identifiant) du patient. Après avoir récupéré le patient il peut lui ajouter une prescription. Nous avons décidé d’utiliser le pattern strategy pour gagner en flexibilité sur l’option de recherche. En effet il existe plusieurs méthodes de recherche autre que ces deux-là. Ce pattern permet d’ajouter n’importe quelle méthode facilement en créant la classe Java associé et la possibilité de la choisir dans la liste de choix de recherche. Nous avons pu rajouter une façon de rechercher dite automatique. Elle nous donne le patient qui a le nom le plus proche de ce que l’on a tapé dans la barre de recherche s’il n’y a pas d’ambiguïté entre 2 personnes (sinon on doit affiner la recherche) par exemple « Al » pour « Alice ». Pour ce pattern on crée une interface RechercheStrategy avec la fonction `getPatient()`. Cette fonction sera redéfini dans chaque classe qui implémentera l’interface (par exemple RechercheNameStrategy qui recherchera par nom). Dans le code on appellera la fonction `getPatient(String info)` avec `info` qui pourra être n’importe quelle information (nom, identifiant ou morceau de nom) sur une strategy définie par un choix dans le menu déroulant (choix du type de recherche).

En résumé le pattern Strategy nous permet de rajouter facilement n’importe quel type de façon de rechercher de patient tout en gardant une cohérence dans l’interface.



Builder :

Toujours dans une recherche de flexibilité de notre projet nous avons décidé d'implémenter le pattern Builder à notre projet. Lors de la création des différents professionnels de santé on spécifie leurs domaines de profession (dentiste, homéopathe, etc...). Ce pattern nous permet de gérer les différents types de profession et d'ajouter facilement des informations supplémentaires chaque professionnelle de santé (année d'obtention du diplôme). On crée une classe Builder pour chaque profession qui hérite d'une classe général Builder des professionnelles de santé. Dans cette classe générale on peut ajouter facilement autant d'information que l'on veut sans être obligé de les renseigner à la création de professionnelle (on peut ajouter un dentiste sans spécifier son année d'obtention de diplôme). Cette façon de structurer notre code nous permet aussi de forcer des attributs de chaque professionnelle (à la création d'un dentiste on force sa spécialité à dentiste ce qui nous permettra de lui ajouter des possibilités uniques à sa profession). Lors de la création d'un professionnel de santé (état initiale de l'application) on récupère ses informations dans un yml. Ici on spécifie les informations que l'on souhaite lui attribuer en plus de son nom et de sa spécialité (un médecin sans spécialité sera un généraliste). Si on ne précise pas l'année d'obtention du diplôme d'un médecin alors elle ne sera pas spécifiée dans le programme. En résumé le pattern Builder nous permet de rajouter facilement des informations pour les professionnels mais en gardant toujours la possibilité de créer un professionnel en ne donnant que son nom. Il permet en plus de distinguer les différents professionnels en fonction de leurs spécialités.



Ethique :

Dans cette partie nous discuterons peu du code mais nous apporterons une analyse avec un regard humain sur l'application. Nous allons aborder les enjeux du numérique et de la création d'application dans ce domaine en prenant exemple sur le site <https://www.monespacesante.fr/>.

Dans le domaine de la santé il est important de respecter la vie privée des patients. C'est d'ailleurs un précepte du serment d'Hippocrate « je tairai les secrets qui me seront confiés ». À l'époque de son écriture il suffisait au médecin de ne rien dire et le secret professionnel ne risquait rien. Mais à notre époque, l'ère du numérique où toutes les informations sont stockées sur des ordinateurs, serveur, cloud etc. Le secret professionnel n'engage plus que des personnes mais aussi des machines. Comme dans tous les domaines cela facilite la vie des professionnels de santé et permet de meilleur soin (le médecin ne se repose pas que sur sa mémoire pour se souvenir des différentes pathologies du patient). Ce système permet aussi d'instaurer plus facilement des dispositifs comme la sécurité sociale etc. Mais comme toutes choses ces innovations ont un coût :

- Financier : La mise en place de ce système demande des technologies de pointe et des travailleurs pour la maintenance du système et du matériel (serveur cloud, etc.)
- Déshumanisation : Dans ce système les patients peuvent vite devenir des numéros avec des listes de pathologie. Pour les professionnels de santé chaque patient est associé à un formulaire et cela peut entraîner une déshumanisation du métier (le patient n'est plus qu'un numéro). On pourrait aussi imaginer une personne qui a eu des pathologies avant que les données soient écrits dans leurs feuilles de patient numérique, celle-ci ne sera pas prise en compte dans le diagnostic du médecin (trop habitué à se reposer uniquement sur le numérique).
- Confidentialité : La création de base de données répertoriant tous les dossiers sont sujets au piratage. Comme n'importe quel type de données de masse, elles valent de l'argent et donc incitent au vol. Et sans parler de vol de masse, on pourrait aussi penser que n'importe qui de mal intentionné pourrait avoir accès aux données privées d'une personne en particulier s'il dispose des accréditations d'un personnel de santé.

Dans le cadre de notre projet nous pouvons avoir un impact sur la confidentialité. Divisons, ce problème en 2 catégories : fuite de masse, accès aux données au cas par cas.

Commençons par une rapide présentation de monespacesante.fr. Le but de ce projet est de créer une base de données globale regroupant toutes les informations de santé d'un Français. Chaque Français est inscrit d'office et peut se désinscrire s'il le souhaite. Le but est de centraliser toutes les données pour faciliter le travail des professionnelles de santé. Cela permet aussi de donner toutes les informations d'un patient lors d'une urgence (accident de la route, etc...). À l'avenir ce système compte aussi partager les données des utilisateurs à des tiers approuvés qui pourront utiliser nos données pour d'autre projets.

Comme on le remarque ce système va donc disposer d'une énorme partie de nos informations de santé (presque toutes). Si sa base de données venait à être piratée ce sont des millions de données qui pourrait être vendu à notre insu. Or il est difficile de se protéger efficacement face au piratage de données mêmes pour les plus grandes instances. En mars 2022 (il y a donc 8 mois) l'assurance-maladie a subi une fuite de données de 510 000 assurées « les conséquences peuvent être importantes puisque les données compromises comprennent le numéro de sécurité sociale en plus de l'identité et des coordonnées des patients. ». Cela pourrait amener à l'usurpation d'identité. Même si la faille a vite été repérée, que les assurés ont été avertie et qu'ils peuvent prendre des dispositions de sécurité auprès de « <https://www.cybermalveillance.gouv.fr> ». Les données sont tout de même entre les mains de personne potentiellement malveillante. L'assurance-maladie (créateur de monespacesante.fr) est donc susceptible de se faire pirater malgré les mesures prises pour l'éviter. Cependant il est difficile de savoir quelle sont ces mesures exactes. « En outre, la CNAM et le Ministère chargé de la Santé exigent de leurs sous-traitants qu'ils mettent en place des mesures de sécurité techniques et organisationnelles appropriées afin de garantir la confidentialité des données personnelles et un niveau de sécurité adapté au risque. ». On peut espérer dans l'avenir avec des données chiffrées de bout en bout comme le fait le site Doctolib depuis juin 2020 « Doctolib annonce aujourd'hui la mise en œuvre du chiffrement de bout en bout pour les données personnelles de santé de ses utilisateurs. ». Un système très sécurisé qui limite grandement les possibilités de piratage de masse.

Concernant la deuxième catégorie de problème, l'accès aux données au cas par cas. Même en ayant une base de données protégée à la perfection on ne peut pas empêcher une personne malveillante de se faire passer pour un professionnel de santé afin de récupérer les données de ses patients. Et même si ce problème existait déjà avant la création de monespacesante.fr il est maintenant décuplé car « n'importe quelle professionnelle » peut accéder au données de tous les patients. C'est en réalité un peu plus compliqué que ça car le patient doit autoriser un personnel de santé à accéder à ses informations. Mais dans la pratique ce n'est pas aussi sécurisé qu'il n'y paraît. Comme le précise Mediapart dans son article daté de mars 2022 « le recueil du consentement est problématique, pour ne pas dire quasiment factice : une simple case à cocher par le ou la soignante servira de « preuve » que l'on a donné son accord pour qu'il ou elle y accède. » Pour limiter ces problèmes on pourrait imaginer qu'un personnel de santé puisse uniquement voir ce qui le concerne (ses prescriptions etc.) et il ne pourrait voir le reste que si le patient le décide sur son propre espace sur internet. Cela permettrait de gagner plus de confidentialité sans perdre le principe du projet. Actuellement un usager peut masquer des éléments de son dossier mais s'il décide de les cacher personne ne pourra les voir (il ne choisit pas au cas par cas) « Il lui est possible de masquer un document à tout moment depuis le site Mon espace santé. ». Dans notre projet un professionnel de santé ne peut voir que les prescriptions qu'il a lui-même faites à un patient. On aurait pu implémenter la possibilité de donner l'accès à un personnel de santé de voir d'autre prescription des professionnels si le patient le décidait.

Le sujet est vaste, on pourrait parler de la façon de considérer une « urgence » qui permettrait à n'importe quel personnel de santé d'avoir accès au dossier complet sans autorisation préalable « par exemple lorsqu'un patient est pris en charge par un médecin qui n'est pas son

médecin habituel, à l'occasion d'un déplacement ou en cas d'urgence, le dossier médical contenu dans mon espace santé permet d'être soigné plus efficacement. ». Ce sujet a d'ailleurs été légèrement traité dans l'article de Mediapart cité plus haut « Mais quels contrôles techniques permettent de définir ce qui est une situation d'urgence et débloquent l'accès des documents aux professionnel·les ? ». Il est dangereux de faire rentrer en cause des décisions subjectives quand cela touche des millions de données sensibles.

Pour conclure cette partie, nous pensons qu'il faudrait bien plus d'options dans l'espace de chacun pour choisir précisément qui a accès à ses données (accepter ou non de les donner en cas d'urgence, choisir entre différents stades ce qu'on considère être de l'urgence). Le patient pourrait choisir de cacher telle donnée à tel professionnel. Et à l'avenir le patient pourrait aussi choisir à quel organisme tiers il décide de fournir ses données (comme précisées plus haut monespacesante.fr veut par la suite fournir ses données à des organismes approuvés pour des études par exemple). Toutes ses options ne régleront pas tous les problèmes mais cela augmenterait l'éthique du projet. Concernant la protection des données pour les vols de masse, un projet aussi ambitieux se doit d'utiliser des moyens de protection à la pointe de la technologie.

Tests :

Pour vérifier le bon fonctionnement de notre application, nous avons réalisé différents tests. Certains de ces tests sont automatiques et peuvent être vérifiés avec la commande "mvn test" qui lance une série de tests qui sont décrits dans le fichier "HealthProTest.java".

Certains tests sont effectués manuellement car ils sont difficiles à automatiser (majoritairement des boutons) sont décrits ci-dessous :

-Add

Pour tester le bouton Add (bouton qui ajoute créer un nouveau patient), il faut :

- remplir la zone de texte de name
- remplir la zone de texte de ssid
- puis appuyer sur le bouton add
- le test est réussi si un nouveau patient s'est ajouté dans la liste des patients.

-copy ssid

Pour tester le bouton copy ssid (bouton qui copie le ssid du patient en question), il faut :

- appuyer sur le bouton, puis coller sur la zone de texte de recherche dans la partie professionnelle de santé
- choisir ssid dans la liste dans "research by" puis cliquer sur le bouton "search"
- le test est réussi si le nom de la personne dont le ssid a été copié est affiché dans la prescription du professionnel de santé

-« x » suppression prescription par un patient

Pour tester le bouton « x » (bouton qui supprime la prescription d'un patient par un patient), il faut :

- cliquez sur le bouton « x » d'une prescription
- le test est réussi si la prescription disparaît

-ComboBox ssid

Pour tester la combobox ssid (ComboBox ssid, choix pour chercher le patient en fonction de ssid), il faut :

- choisir ssid dans la liste déroulante de la ComboBox
- entrer le ssid d'un patient
- et cliquez sur search et vérifier que le ssid entré correspond au patient trouvé

- ComboBox name

Pour tester la ComboBox name (ComboBox name, choix pour chercher le patient en fonction de son nom), il faut :

- choisir name dans la liste déroulante de la ComboBox
- entrer le nom d'un patient
- et cliquez sur search et vérifier que le nom entré correspond au patient trouvé

- ComboBox auto

Pour tester la ComboBox auto (ComboBox auto, choix pour chercher le patient en fonction du texte rentré), il faut :

- choisir auto dans la liste déroulante de la ComboBox
- entrer quelque chose dans la zone de texte
- et cliquez sur search, si le texte entré correspond à un patient, vérifier que le patient s'affiche sinon rien ne devrait s'afficher

-zone de texte + bouton search

Pour tester le bouton search (bouton qui recherche un patient), il faut :

- entrer quelque chose dans la zone de texte selon le choix choisi dans la ComboBox
- cliquez sur "search"
- vérifier que le résultat affiché correspond à la recherche demandée

-Zone de texte + bouton add(prescription)

Pour tester le bouton Add (bouton qui ajoute une prescription personnalisée à un patient), il faut :

- rechercher un patient
- ajouter du texte dans la zone de texte pour la prescription, cliquez sur le bouton add
- vérifier que la prescription a été ajoutée pour le patient en question

-prescription prédéfinie

Pour tester le bouton Add (bouton qui ajoute créer un nouveau patient), il faut :

- rechercher un patient
- cliquez sur le bouton de la prescription prédéfini
- vérifier que la prescription prédéfinie a été ajoutée pour le patient en question

Reference :

<https://www.conseil-national.medecin.fr/medecin/devoirs-droits/serment-dhippocrate>

<https://www.midilibre.fr/2022/03/29/fuite-massive-de-donnees-vous-avez-recu-un-mail-de-lassurance-maladie-voici-ce-que-vous-devez-faire-10201405.php>

<https://www.monespacesante.fr/protection-donnees-personnelles>

<https://info.doctolib.fr/blog/doctolib-adopte-le-chiffrement-de-bout-en-bout-nouvelle-etape-dans-la-securisation-des-donnees-de-sante/>

<https://blogs.mediapart.fr/la-quadrature-du-net/blog/280322/pourquoi-s-opposer-la-creation-de-mon-espace-sante>

<https://www.ameli.fr/rhone/assure/sante/mon-espace-sante>