



Rapport du projet d'ERP

Programme Hôpital Numérique

Table des matières

1	<i>Présentation du projet</i>	3
2	<i>Module gestion de l'hôpital</i>	3
2.1	Technologies employées	3
2.2	Interconnexions	3
2.2.1	Création de rendez-vous	3
2.2.2	Confirmation de rendez-vous	4
2.2.3	Paiement de l'acte	4
2.3	Ajout de pièces-jointes	4
3	<i>Module gestion du Dossier Médical Informatisé (DMI)</i>	5
3.1	Technologies employées	5
3.2	Documentation technique	5
3.2.1	Structure de la base de données	5
3.2.2	Endpoints du backend	6
3.2.3	Structure du frontend	7
3.2.4	Interconnexion avec le module Hôpital	7
4	<i>Module gestion de la mutuelle</i>	7
4.1	Technologies employées	8
4.2	Interconnexions	8
4.2.1	Traitement d'un acte médical	8
5	<i>Conclusion sur les modules interconnectés</i>	8

1 Présentation du projet

Le projet Stratégie Nationale de Santé, proposé par Mesdames Marisol TOURAINE, ministre des Affaires Sociales et de la Santé et Geneviève FIORASO, ministre de l'Enseignement Supérieur et de la Recherche, propose la réalisation d'un Programme Hôpital Numérique, visant à mieux partager les données entre les différents acteurs de santé. Ces derniers sont au nombre de deux, à savoir la mutuelle et la centralisation du Dossier Médical Informatisé (DMI).

Nous avons donc répondu à leur projet en implémentation 3 modules distincts, qui ont chacun un fonctionnement distinct et local, et qui propose dans le même temps une interconnexion entre chaque, comme nous allons le préciser dans chacune des sections ci-dessous.

2 Module gestion de l'hôpital

Ce module est l'épine dorsale du projet. Il s'axe sur la gestion de l'hôpital, et plus précisément la création des rendez-vous, avec différentes informations sur le patient, que ce dernier fournira lors de la création de fiches de l'acte à venir, à savoir :

- **ID Graulandais**, le numérique unique de chaque graulandais,
- **Date et heure** convenues de l'examen,
- **Examen**, le type d'examen (scanner, IRM, consultation, ...),
- **Lieu** où se déroulera le rendez-vous,
- **Patient**, le nom du patient,
- **Metadata 1**, des notes sur le rendez-vous (quelle jambe est à scanner, par exemple),
- **Metadata 2**, des notes sur le rendez-vous (quelles observations ont déjà été faites),
- **Montant**, rempli par l'hôpital, qui sera payé par le patient au travers de la DMI et la mutuelle.

Ces informations permettent donc de créer le rendez-vous, qui sera par la suite à confirmer. Cette action va permettre au patient de voir que son rendez-vous se déroulera bien aux horaires indiqués, sur son DMI.

Il est également possible d'ajouter à ces rendez-vous des pièces jointe permettant alors de pouvoir déposer des documents médicaux lié au rendez-vous, type prescriptions ou ordonnances.

Une fois le rendez-vous confirmé passé, le paiement sera attendu par l'hôpital. Le patient, au travers de la DMI paiera ses 50% du tarif et la part restante que la mutuelle se chargera de régler.

2.1 Technologies employées

Afin d'offrir une interface compréhensible, « *mobile first* », nous avons utilisé de l'HTML, CSS et JavaScript pour le frontend, sans *framework*.

Pour toute la gestion dite *backend*, nous avons utilisé PHP. Cela nous a permis de réaliser les interconnexions, qui sont décrite ci-dessous (2.2), mais également la gestion de la base de données (SQLite), aussi bien pour la création si elle n'existait pas, que pour les ajouts, suppressions et obtentions d'informations.

2.2 Interconnexions

2.2.1 Création de rendez-vous

La première interconnexion s'effectue avec la DMI et la mutuelle lors de la création d'un rendez-vous. Nous leur fournissons toutes les informations que nous avons récoltées, ainsi que le numéro

identifiant le rendez-vous dans la base de données de l'hôpital (en l'occurrence, l'index dans la base de données), sous forme de requêtes GET, grâce à *curl*.

Cette requête, lorsqu'elle est forgée dans le backend de l'hôpital, va viser un fichier propre à chaque module : `backend/src/createact.php` (pour la DMI) et `api/add_intervention.php` (pour la mutuelle).

Cela permet donc aux deux modules de connaître les informations sur l'examen à venir et plus particulièrement pour la DMI, le fait de savoir si le rendez-vous est bien confirmé.

Exemple de requête forgée pour la mutuelle :

```
http://mutuelle/api/add_intervention.php?nuig=1001&lieu=16+Rue+des+pins+67100+Strasbourg&intervention=Scanner&date=2023-12-24+13%3A34&total=460&id_acte=4
```

2.2.2 Confirmation de rendez-vous

Pour cela, une seconde requête est réalisée vers la DMI lorsque l'hôpital valide le rendez-vous. Il indique le numéro de l'examen, permettant ainsi à la DMI d'actualiser sa base de données.

Exemple de requête forgée pour la DMI :

```
http://dmi/api/confirm_rdv.php?id=5
```

2.2.3 Paiement de l'acte

Lorsque l'acte est passé, l'hôpital attend les paiements de la mutuelle et du patient (grâce au DMI). Il y a donc un booléen pour chacun des deux modules dans la base de données de l'hôpital, permettant de savoir lequel a déjà payé sa part.

Lorsque la DMI ou la mutuelle veut indiquer qu'elle a payé, elle envoie une requête en GET sur le fichier `confirmPayment.php`, avec l'entité qui émet la requête (« dmi » ou « hopital »), ainsi que le numéro de l'examen. Ces informations permettent à l'hôpital de prendre en compte le paiement et de passer le booléen à 1.

Exemple de requête attendue :

```
http://hopital/backend/confirmPayment.php?idActe=1&entite=dmi
```

2.3 Ajout de pièces-jointes

Une fois le rendez-vous passé, il est possible via le site de l'hôpital de pouvoir ajouter des documents liés au rendez-vous, et donc d'avoir un suivi avec ordonnances ou autres pour le rendez-vous en question.

Pour implémenter cela, nous utilisons déjà les technologies utilisées précédemment pour créer un rendez-vous notamment, à savoir la base de données. Cependant au lieu d'utiliser la même table que pour les rendez-vous, nous allons utiliser une autre table prévue uniquement pour les dépôts de fichier. Lorsque le fichier est déposé sur le site, le script vérifie qu'il n'y a qu'un seul fichier déposé et un fichier valide (pas un dossier ou une référence d'un fichier).

Une fois cette vérification faite, le contenu intégral du fichier (tous les octets) sont encodés en base64 avant d'être envoyé en requête vers le backend du site. Le backend reçoit trois informations pour ajouter le fichier dans la base de données : l'ID du rendez-vous, le nom du fichier et son contenu en base64.

Encoder le contenu en base64 est pratique car il évite d'insérer dans la requête les caractères spéciaux et notamment ceux d'échappement qui pourraient être présents dans le contenu du fichier

(par exemple : « \n »). Le backend insère donc dans le fichier dans la table liès au fichier dans la base de données, voici le format de la table et de ses données :

ID du fichier	Nom du fichier	ID du rendez-vous	Contenu encodé
---------------	----------------	-------------------	----------------

Ainsi chaque fichier possède un ID unique dans la base de données (peu importe à quel rendez-vous il est lié).

Le frontend du site permet de lister les fichiers relatifs à un certain rendez-vous en cliquant sur une icône liée à cette action. Pour y arriver, il suffit d'envoyer une requête en sélectionnant tous les fichiers avec comme ID du rendez-vous celui sur lequel on vient de cliquer. Dans cette liste il est possible de faire deux actions sur les fichiers, les supprimer de la base de données ou les télécharger. Pour les deux actions, à chaque fois une requête différente est envoyée : pour la suppression, une requête de suppression dans la base de données avec l'ID du fichier est envoyée, ainsi que pour le téléchargement, on envoie une requête avec l'ID du fichier et on reçoit le contenu du fichier en base64 qui sera décodé par le script et va déclencher le téléchargement.

3 Module gestion du Dossier Médical Informatisé (DMI)

Le Module Gestion du Dossier Médical Informatisé (DMI) permet la consultation par le patient de ses rendez-vous futur et passé. Cela lui permet d'avoir des informations importantes sur les opérations comme le type d'opération, le lieu, les coûts et leurs répartitions (proportion payée par la mutuelle, et reste à charge pour le patient). Cette plateforme est connectée au système de l'hôpital. Elle permet à l'utilisateur de créer un compte et de se connecter avec son identifiant grolandais.

3.1 Technologies employées

Nous sommes partis sur une structure classique en trois parties : base de données, backend et frontend. La base de données était du MySQL, le backend du PHP et le Frontend du Vue.js.

3.2 Documentation technique

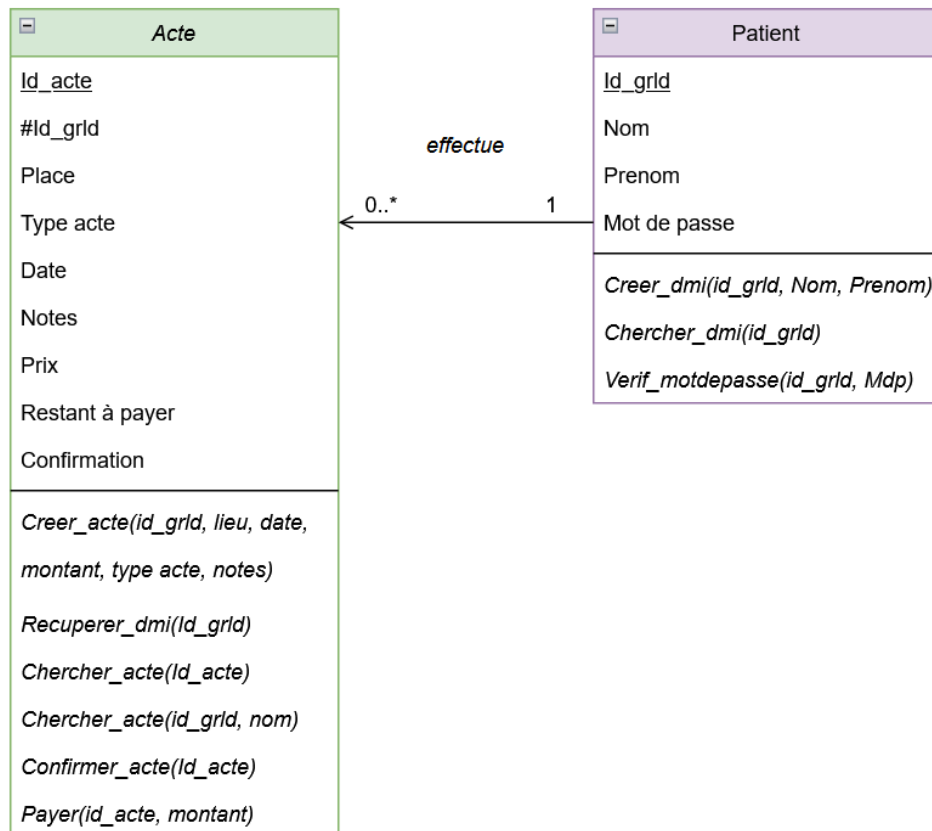
3.2.1 Structure de la base de données

La structure de la base de données est plutôt simple.

Nous avons ci-dessous la représentation de la base de données, ainsi que les différentes fonctions qui permettront d'interagir avec cette dernière.

La classe Patient correspond aux informations de bases du DMI, et qui permettent l'identification d'un dossier médical par l'id grolandais qui est unique.

La classe Acte représente les différents actes médicaux que le patient a effectué à l'Hopital. Le lien entre les actes et le patient est fait grâce à l'id grolandais, et est utilisé pour pouvoir récupérer toutes les informations du patient.



3.2.2 Endpoints du backend

7 endpoints ont été créés pour que le frontend communique avec le backend.

Création d'un compte :

POST url/createdmi.php

Paramètres : id grolandais (idGroland), nom (nom), prénom (prenom) et mot de passe (mdp).
L'opération de création d'un DMI est supposément faite par une autorité administrative (mairie, mutuelle...). Suite à cette opération, le patient peut consulter le DMI sur le web.

Connection :

POST url/loginuser.php

Paramètres : id grolandais (id), et mot de passe (password).
Cette fonction permet de vérifier si l'identifiant et le mot de passe correspondent.
Elle est utilisée pour vérifier si l'utilisateur peut consulter le DMI

Récupérer les informations d'un patient

POST url/ getpatientinfo.php

Paramètres : id grolandais (idGroland).
Cette fonction permet de récupérer les informations de base du patient.

Récupérer les actes d'un patient

POST url/ getdmi.php

Paramètres : id grolandais (idGroland).

Cette fonction permet de récupérer l'entièreté du dossier médical du patient en utilisant son Id grolandais.

Payer un acte médical

GET url/ payact.php

Paramètres : id acte, montant.

Cette fonction permet de payer un acte médical. Lorsque la somme restante à payer est de 0, on envoie une requête cUrl à l'Hôpital pour leur confirmer le paiement.

Créer un acte médical

GET url/ createact.php

Paramètres : id grolandais (idGroland), id acte, lieu, type acte, date, montant.

Cette fonction permet de créer un nouvel acte médical. Cette opération doit être effectuée par l'Hôpital lors de la création d'un acte médical.

Confirmer un acte médical

POST url/ confirmact.php

Paramètre : id acte

Permet de confirmer un acte médical.

3.2.3 Structure du frontend

La structure de l'application Vue.js suit les guidelines standard du framework. Le dossier **src** contient les sous dossiers api (liste des appels à l'api), components (composants réutilisables : les formulaires ou les tableaux), store (données de l'application) et views (les différentes pages de l'application, le login, signin et page principale).

L'application utilise les modules Vuex, Axios, VueRouter et le catalogue de composants naiveui.

3.2.4 Interconnexion avec le module Hôpital

Le DMI est interconnecté avec l'Hôpital, car c'est l'Hôpital qui crée les différents actes médicaux et qui confirme les rendez-vous. De plus, la DMI doit confirmer le paiement des actes médicaux dû à l'Hôpital.

L'Hôpital peut notifier le DMI de la création d'un acte via le endpoint createact.php, qui permet de créer un acte dans la base de données du DMI. Elle est faite au moment de la création d'un acte sur la page web de l'Hôpital.

Le DMI informe l'Hôpital lorsque l'acte d'un patient a été totalement payé, via l'endpoint payact.php. Cela permet de notifier l'Hôpital du paiement complet de l'acte par le client.

La mutuelle doit également notifier le DMI de la prise en charge du patient de 50% via l'endpoint actmut.php, mais il n'est pas implémenté. On part du principe que la mutuelle prend automatique en charge la moitié du prix des actes médicaux.

4 Module gestion de la mutuelle

Le module s'occupe des traitements associés aux activités de la mutuelle à savoir prendre en charge une partie des coûts des soins des patients ayant souscrit à la mutuelle en question. En résumé la mutuelle reçoit les informations concernant les soins reçus par un patient et effectue un virement à l'hôpital pour en payer une partie (50% en l'occurrence). Cette plateforme permet également à un

client de se connecter et de consulter l'historique de ses actes médicaux et des remboursements effectués par sa mutuelle.

4.1 Technologies employées

Nous avons préféré rester sur des technologies assez simples et connues de notre équipe et nous sommes donc contents de HTML, CSS, JavaScript et PHP pour réaliser cette partie. Nous employons des fichiers texte pour enregistrer les données dont les entrées se présentent comme suit :

- Date Patient Intervention Commentaire Lieu Prix Virement Taux de prise en charge
pour les actes médicaux
- Nom Prénom ID Graulandais Mot de passe
pour les utilisateurs

4.2 Interconnexions

4.2.1 Traitement d'un acte médical

Lorsque la mutuelle reçoit des informations sur des soins reçus par un de ses clients sur l'API correspondante (http://mutuelle/api/add_intervention.php) avec les informations suivantes (requête GET) :

<u>Valeur</u>	<u>Nom de variable</u>
- Date	date
- Identifiant	nuig
- Type d'intervention	intervention
- Commentaire	commentaire
- Lieu	lieu
- Coût total	total
- Identifiant de l'acte	id_acte

Elle enregistre ces informations pour que le client concerné puisse les consulter ultérieurement sur l'interface en ligne puis envoie une réponse à l'hôpital pour confirmer la prise en charge de 50% des coûts ([http://hopital/backend/confirmPayment.php?entite=mutuelle&idActe=\\$id_acte](http://hopital/backend/confirmPayment.php?entite=mutuelle&idActe=$id_acte)). Suite à cela le DMI est également informé du remboursement effectué par la mutuelle via l'API suivante : [http://dmi-php-1/backend/src/actmut.php?id=\\$nuig&intervention=\\$intervention](http://dmi-php-1/backend/src/actmut.php?id=$nuig&intervention=$intervention).

5 Conclusion sur les modules interconnectés

En résumé la communication entre l'hôpital et la mutuelle fonctionne, il y a juste un petit problème lors de la réponse de la mutuelle qui bloque l'hôpital (sans doute une attente d'un certain événement qui ne se produit que si on stoppe le conteneur de la mutuelle). Pour ce qui est de l'interaction de la mutuelle avec le DMI nous n'avons pas eu l'occasion de la tester mais le code des deux parties est en adéquation avec l'API correspondante donc elle devrait fonctionner pour peu qu'il n'y ait pas de soucis sur la connexion. Les échanges entre le DMI et l'hôpital quant à eux sont fonctionnels.