Rapport de stage

Développeur Web et Web Mobile

2021 – 2022



PEROCHE Julien 7/03/2022 – 13/05/2022

# Sommaire

Table des matières

[Sommaire 2](#_Toc103813668)

[Remerciements 3](#_Toc103813669)

[Présentation du Centre Hospitalier 4](#_Toc103813670)

[Universitaire de Lille 4](#_Toc103813671)

[Les chiffres du CHU en 2020 : 5](#_Toc103813672)

[L’activité du Département des Ressources Numériques : 5](#_Toc103813673)

[Langages et technologies utilisées 6](#_Toc103813674)

[L’éditeur de code Visual Studio Code : 6](#_Toc103813675)

[HTML5 et CSS3 : 6](#_Toc103813676)

[JavaScript : 7](#_Toc103813677)

[PHP : 7](#_Toc103813678)

[GitHub et Git Desktop : 7](#_Toc103813679)

[Le projet 8](#_Toc103813680)

[Méthodes de travail 8](#_Toc103813681)

[Introduction du projet 11](#_Toc103813682)

[Le Transport Automatique Léger ou TAL 14](#_Toc103813683)

[Expression des besoins et cahier des charges 15](#_Toc103813684)

[L’application 16](#_Toc103813685)

[Compétences couvertes 22](#_Toc103813686)

[Développement 23](#_Toc103813687)

[Difficulté rencontrée 38](#_Toc103813688)

[Conclusion 41](#_Toc103813689)

[Annexes 42](#_Toc103813690)

[Arborescence 42](#_Toc103813691)

[Le code HTML : l’index 43](#_Toc103813692)

[Le CSS 44](#_Toc103813693)

[Le JavaScript 50](#_Toc103813694)

[Le PHP : header et tableau d’analyse 56](#_Toc103813695)

[Le PHP : modal d’impression 60](#_Toc103813696)

[Le PHP : modal de renseignement clinique 62](#_Toc103813697)

[Le HTML : impression 63](#_Toc103813698)

[Le CSS : 64](#_Toc103813699)

[Le JavaScript : impression 65](#_Toc103813700)

# Remerciements

Je tiens à remercier monsieur Dylan HUART, maitre de stage, qui de par sa bonne humeur, son encadrement et sa pédagogie, m’a permis de passer une période très enrichissante et me confortant pour l’avenir.

Je remercie également l’équipe du Département des Ressources Numériques et Système d’Informations pour leur accueil et leur sympathie tout au long de mon stage.

Et enfin un grand merci à monsieur Ryad CHAOURAR et madame Nora ABDERRAHMANI qui m’ont permis de suivre ce stage et pour leur soutien.

Une image contenant texte, ciel

Description générée automatiquement

# Présentation du Centre Hospitalier

# Universitaire de Lille

Le Centre Hospitalier Régional Universitaire de Lille l’un des plus grands centres hospitaliers de France au dimension Européenne avec pas moins de 3 197 lits. Le centre est composé de 14 hôpitaux spécialisé (10 hôpitaux, 1 Centre de Lutte Contre le Cancer, l’Institut Cœur Poumons et deux secteurs dédiés), 16 pôles cliniques et médico-techniques et est affilié à l’université de Lille, ce qui fait de lui, avec le Parc Eurasanté, le plus grand campus hospitalo-universitaire d’Europe.

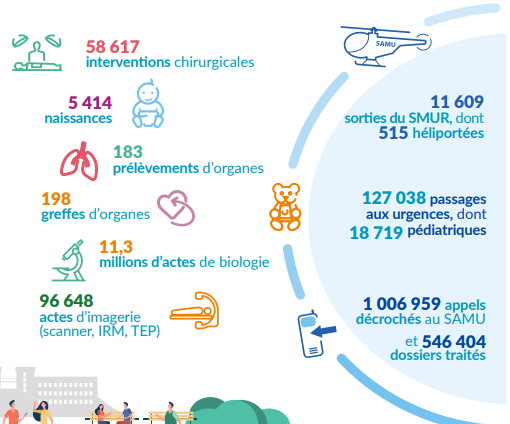
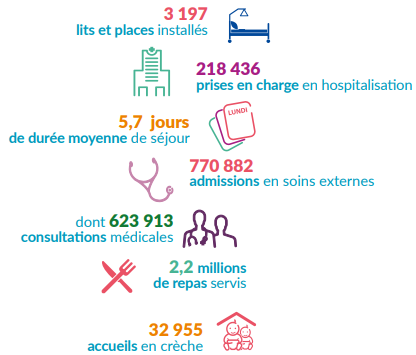
Une image contenant nature, route, ravin

Description générée automatiquementOuvert depuis plus de 80 ans (inaugurer en 1953) ces hôpitaux sont voués à l’accueil des malades et personnes en détresses 7 jours sur 7 et 24 heures sur 24.

Le CHU de Lille ne cesse de s’améliorer grâce aux progrès techniques et aux connaissances qui évoluent, mais également aux fortes contraintes d’organisation que le centre met en place pour s’adapter en toutes circonstances.

Document 1 : Vue du ciel de plusieurs bâtiment du CHU de Lille

## Les chiffres du CHU en 2020 :



Également :

* 22 projets de recherche collaboratifs européens, dont quatre coordonnés par le CHU de Lille
* 1 IRM Peropératoire
* 1 Gamma Knife
* 1 robot chirurgical
* 2 salles hybrides
* 3 chambres hyperbares

## L’activité du Département des Ressources Numériques :

* Plus de 120 employés
* 5 domaines différents :

Domaine Applicatif (gestion du run et maintiens des applications)

Domaine Opérationnel (organisation du Système d’Information)

Plateau technique (imagerie et logistique)

Le pilotage (gestion de projet)

L’infrastructure (gestion des serveurs)

* Plus de 300 applications gérées

# Langages et technologies utilisées

Durant mon stage j’ai été amené à développer le projet sur les langages HTML5, CSS3, PHP, JavaScript et l’IDE Microsoft Visual Studio. Pour la gestion de la base de données nous avons utilisé Oracle SQL Developer.

## L’éditeur de code Visual Studio Code :

Une image contenant texte, clipart

Description générée automatiquementVisual Studio Code est un éditeur de code open source et gratuit développé par Microsoft. Il possède un accès au terminal, prend en charge le débogage, la mise en évidence de la syntaxe, la complétion intelligente et Git intégrer. Enfin, il prend en charge presque tous les principaux langages de programmation.

HTML5 et CSS3 :

**HyperText Markup Language 5** est la dernière révision majeure du HTML. Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d’un modèle abstrait défini en termes de DOM (Document Objet Model) : HTML5 et XHTML5.

Le langage comprend également :

* Une couche application avec de nombreuses API (Application Programming Interface)
* Un algorithme afin de pouvoir traiter les documents à la syntaxes non conforme.

Cette dernière version présente plusieurs nouveautés qui doivent faciliter le travail des développeurs d’applications Web.Une image contenant texte, clipart

Description générée automatiquement

**Cascading Style Sheets 3** est un langage de règles de style utilisé pour mettre en forme le contenu HTML. Par exemple en modifiant la couleur d’arrière-plan ou les polices.

## JavaScript :

JavaScript est un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d’animer des images.

## PHP :

PHP est un langage de programmation orienté objet influencé par du C  
et du Java. Ce langage est utilisé pour développer des sites web  
dynamiques. C’est également un langage multiplateforme.

## GitHub et Git Desktop :

Git est un outil de contrôle de version qui représente un  
moyen de sauvegarder son code sur des « repository »  
distants (l’inverse de locale qui représente  
 l’enregistrement de son code sur son ordinateur). Il est  
aussi un moyen de pouvoir travailler à plusieurs sur un même projet. Il permet  
notamment la récupération et l’ajout de code provenant  
du même répertoire. Il est majoritairement utilisé pendant  
le développement de logiciel et permet un suivi de ce  
même développement. GitHub est un service web  
d'hébergement et de gestion de développement de  
logiciels utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on  
Rails. Il propose des comptes professionnels payants ainsi que des comptes gratuits  
pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des  
fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de  
fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

# Le projet

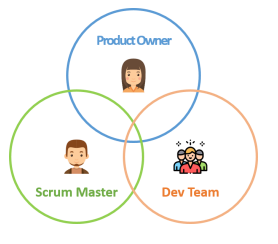
## Méthodes de travail

Pour mener à bien le projet nous avons utilisé l’approche Agile couplé à la méthode SCRUM.

L’approche Agile un état d’esprit dans la conduite d’un projet. Elle répond à douze grands principes, les trois plus importants sont :

* **La Simplicité :** Transformer un problème complexe en une somme de sous problèmes simple.
* **L’Efficacité :** Gommer les silos, avoir plus de transversalité.
* **La Transparence :** Toute l’équipe doit avoir le même niveau d’information sur l’avancement de projet.

La méthode SCRUM (venant de la mêlée au rugby) se » caractérise par plusieurs rôles et plusieurs étapes. Pour les rôles :

* **Le Product Owner** qui collabore avec le client pour définir, grâce aux feedbacks clients, les spécificités fonctionnelles du produit, les prioriser ensuite avec l’équipe, valider les fonctionnalités développées et endosser le rôle du client auprès de l’équipe.
* **Le Scum Master** qui est le guide de l’avancement du projet. Il s’assure que les principes et les valeurs du Scrum soient respectés. C’est le coordinateur des équipes.
* **La Delivery Team**, des développeurs sans hiérarchie entre eux, quand bien même leur savoir-faire et compétence seraient différents

Pour les grandes étapes :

1. **Le Product Backlog :** le Product Owner rencontre le client et analyse son besoin. Il identifie toutes les fonctionnalités dont le produit devra être composé (les users stories) dans ce qui s’appelle le Product Backlog. Ce sera notre « cahier des charges » qui pourra évoluer en fonction des besoins du client et l’avancement du projet. L’équipe décide de ce qu’elle peut faire et dans quel ordre le faire.
2. **Le Sprint :** L’équipe trie les fonctionnalités et tâches qu’elle répartit dans des Sprint. Et pendant un cycle (environ deux semaines), l’équipe s’occupera par exemple uniquement de coder une fonctionnalité du produit qu’elle devra livrer à la fin de cette phase.

On organise avant chaque sprint une réunion de planification, le sprint planning meeting, pour sélectionner dans le product backlog les exigences les plus prioritaires pour le client.

Chaque matin, on réalise le daily scrum : une réunion de 5-10 minutes où l’on parle de trois choses :

* + - Ce qu’on a fait hier ?
    - Quels problèmes on a rencontrés ?
    - Une image contenant herbe, personne, champ, extérieur

      Description générée automatiquementQue va-t-on faire aujourd’hui ?

1. **Le Sprint Review**, vient à la fin du sprint, est l’occasion de tester les fonctionnalités avec le Product Owner et le client confirme ou non si la fonctionnalité marche comme il le souhaitait.



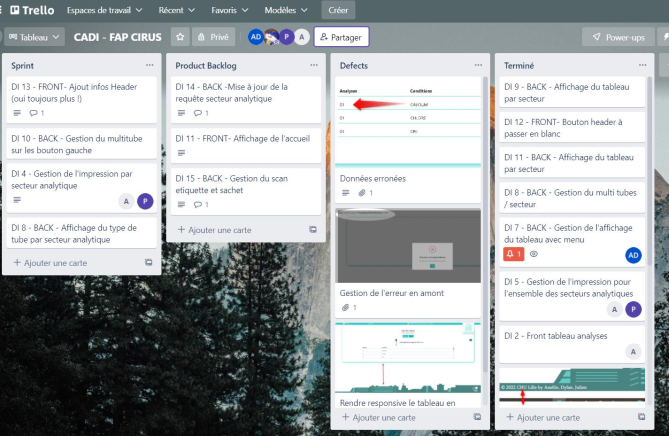
**L’organisation et la gestion des tâches :**

Une image contenant texte, clipart

Description générée automatiquementPour l’organisation et la gestion des tâches, sprint et product backlog, nous avons utilisé Trello, un outil de gestion de projet en ligne, inspiré par la méthode Kanban de Toyota. Il est basé sur un e organisation des projets en planches listant des cartes, chacune représentant une tâche, assignables à des utilisateurs.

Trello nous a été utile pour voir clairement les objectifs et l’avancement du projet

**Le Trello du projet CADI après un mois**



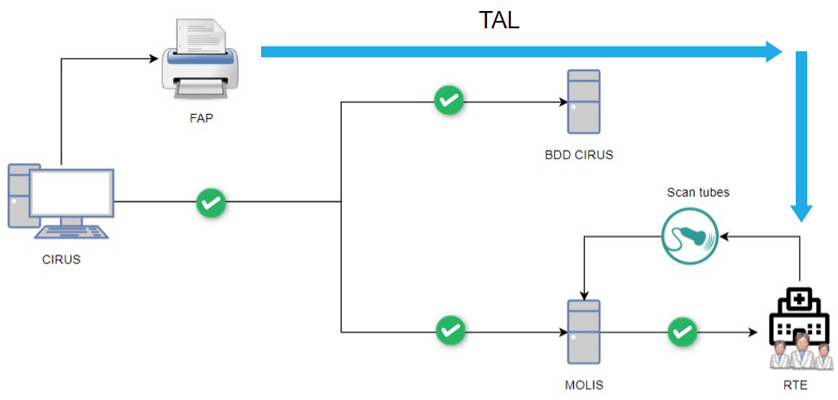
## Introduction du projet

Le CHU de Lille réalise en moyenne **8000 demandes** d’examens de biologie par jour, aujourd’hui la seule procédure dégradée, permettant aux laboratoires de réaliser les analyses, se repose sur une impression papier et cela génère pas loin de **3 000 000 de feuilles par an** !

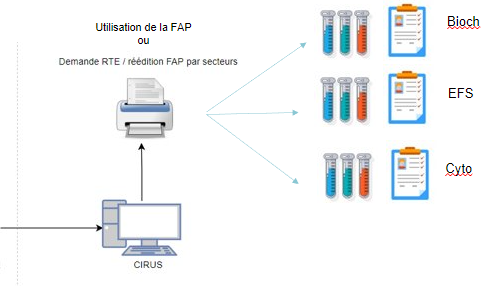
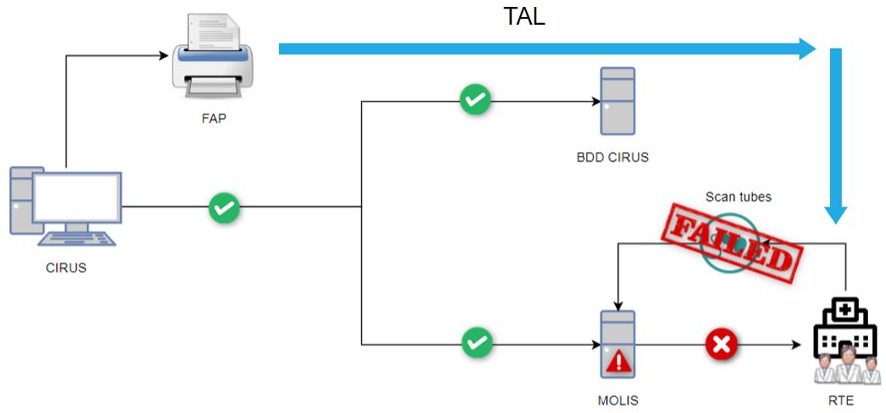
Dans le contexte actuel, les prescriptions sont envoyées par **CIRUS,** (logiciel permettant la prescription et la réception des analyses de biologie), de là est imprimé la Feuille d’Aides aux Prélèvements (FAP) automatiquement.

Les tubes patients sont envoyés au **RTE** (Réception Tri Enregistrement) avec la FAP dans un sachet transporté par le TAL (Transport Automatique Léger que nous expliquerons par la suite), dans un même temps, la prescription est alors enregistrée dans une base de données et envoyée sur **MOLIS** (application transmettant les résultats à CIRUS). Le laboratoire réceptionne le sachet, le scan et procède au tri et à l’enregistrement des prélèvements, les automates s’occupent du reste.

En temps normal la FAP est jetée à la poubelle, car elle est utilisée uniquement en cas de panne du système d’information (SI).

L’équipe projet a remarqué que seul 30% des FAP étaient envoyées par les services, ce qui fait donc que **70% des FAP** finissent à la poubelle et ne pourront être utilisées en procédure dégradée.

En cas de panne MOLIS, les laborantins doivent se connecter à CIRUS pour récupérer la FAP par secteur, si elle n'est pas transmise, pour envoyer les tubes et la FAP par secteur aux services concernés.



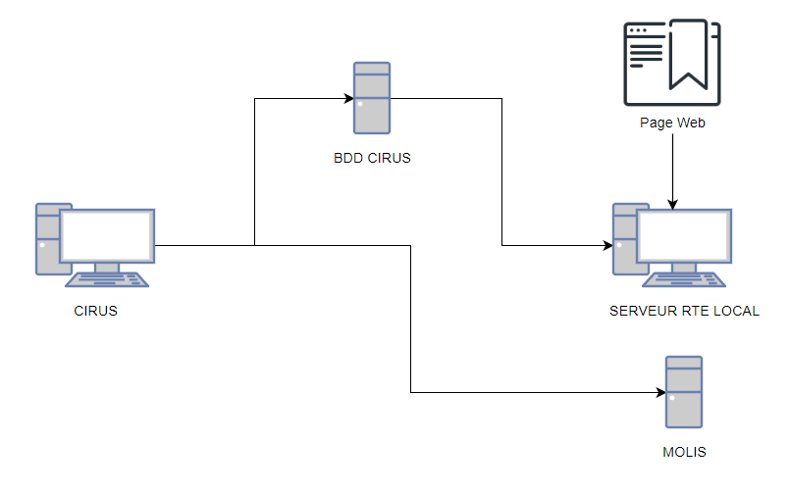
Panne du logiciel Molis

En cas de panne général, ce qui est déjà arrivé, les laborantins n'ont plus accès aux prescriptions alors que les sachets, pour la plupart sans FAP, s'accumulent au RTE. Ils perdent du temps à contacter les hôpitaux pour avoir les renseignements, ce qui entraine des retards, des tubes peuvent être jetés à la poubelle et peut entrainer un risque potentiellement vital pour le patient.

Une image contenant texte, capture d’écran, lumière

Description générée automatiquement

Panne générale



La solution retenue par l’équipe projet est la suivante :

Une recopie en temps réel des données de la base CIRUS dans une base de données locale situé aux laboratoires, ce qui permet d’avoir la dernière information avant coupure sans dépendre d’une procédure papier ou d’un appel téléphonique, le tout consultable depuis une page web.

L’application **CADI** (Consultation des Analyses dégradées Informatisées) a donc été conçue dans le but de limiter les impressions papier et de fiabiliser les informations, via son interface Web accessible depuis une base de données locales, elle permet de répondre à une éventuelle coupure du SI (système d’information).

Nous avons donc mis en place un système qui permet le scan d’une étiquette sachet (qui contient les tubes) ou directement par une étiquette tube, via un numéro de demande, celui-ci nous renvoi les informations du patient dans un header et les analyses du patient dans un tableau. Ce tableau comporte un menu avec les différents secteurs analytiques des prélèvements à effectuer, dans chaque secteur se trouvent des boutons qui représentent les différents tubes et analyses associées.

Ces tubes sont représentés par une couleur spécifique à l’analyse. Le laboratoire pourra alors imprimer l’ensemble des informations ou par secteur analytique, chose qui était impossible de faire avant. À cet effet nous avons créé un lien vers une page d’impression grâce à un bouton.

## Le Transport Automatique Léger ou TAL

Durant mon stage, j’ai pu observer le **TAL**.

Ce système permet l’acheminement de différent échantillons biologique provenant des divers hôpitaux du CHU jusqu’au **RTE**, Réception Tri et Enregistrement, du Centre de Biologie Pathologie grâce à **20 réseaux** de tuyaux pneumatiques en sous-sol pour une longueur totale de 18km. Près de 200 “gares” d’envoi sont implantées dans toutes les unités de soins qui adressent environ 2000 sachets/24h.

Avec une vitesse d’environ **25km/h**, le transfert le plus long s’effectue en cinq minutes.



Le réseaux TAL

## Expression des besoins et cahier des charges

**Front-end:**

-Création du header avec les informations du patient et du tableau d’analyses avec les différents tubes par secteur analytiques.

-Création de la page d’accueil.

-Création des pages d’impression totale et d’impression sélective.

**Back-end:**

-Récupération des informations du patient depuis la base de données.

-Création du menu tableau avec les différents secteurs analytiques suivant le numéro de demande.

-Récupération des analyses par secteur analytiques.

## L’application

La page d’accueil

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementLes résultats

Une image contenant texte, capture d’écran, moniteur

Description générée automatiquement

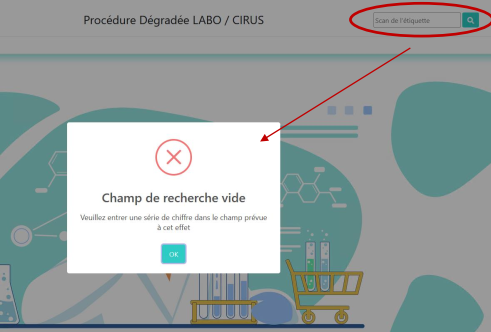
Le fonctionnement de CADI est assez « simple ». Le numéro qui est entré dans l’application permet de récupérer un tableau de données que l’on va parcourir pour afficher les résultats dans un fichier html formaté de différentes <div> possédant un id.

Nous utilisons la méthode Ajax pour remplir chacune des div avec les informations nécessaires au RTE

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementEn cas de numéro incorrecte ou juste aucun numéro tapé

Une image contenant texte

Description générée automatiquementUn bouton permet d’ouvrir une modal où sont afficher les Renseignement Clinique

Une image contenant texte

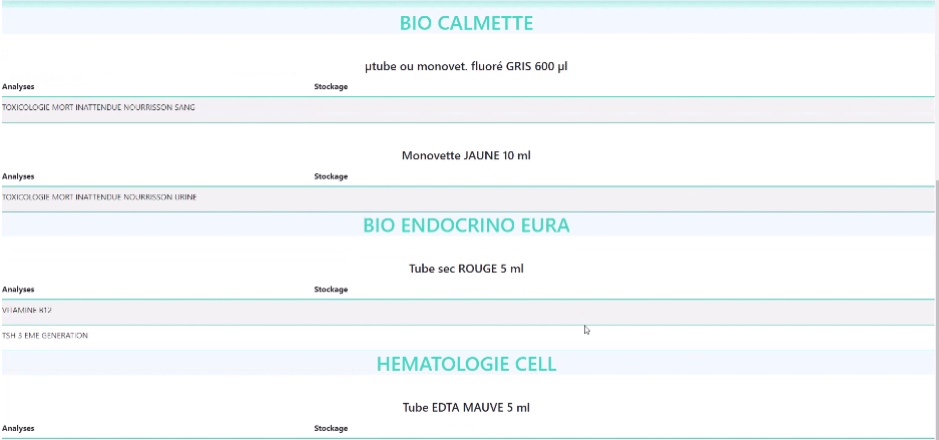
Description générée automatiquementUne image contenant texte

Description générée automatiquementUn bouton permet d’ouvrir une modal où ce trouve la fonction « print »

Une image contenant texte

Description générée automatiquement

Ce qui ouvre une nouvelle page avec les informations dans un header de page et les secteurs sélectionnés.



## Compétences couvertes

Activités :

Développer la partie front-end d’une application web ou web mobile en intégrant les recommandations de sécurité,

Développer la partie back-end d’une application web ou web mobile en intégrant les recommandations de sécurité.

|  |  |  |
| --- | --- | --- |
| Compétences | Activité | Couverte ? |
| Maquetter une application | 1 | x |
| Réaliser une interface utilisateur web statique et adaptable | 1 | x |
| Développer une interface utilisateur web dynamique | 1 | x |
| Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce | 1 | x |
| Créer une base de données | 2 |  |
| Développer les composants d’accès aux données | 2 |  |
| Développer la partie back-end d’une application web ou web mobile | 2 | x |
| Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce | 2 |  |

# Développement

Tout d’abord voici la base de données que nous avons utilisé.

Une image contenant table

Description générée automatiquementLa partie Demandes :

Nous avons surtout utilisé cette partie :

Une image contenant table

Description générée automatiquement

Et la partie Résultats :

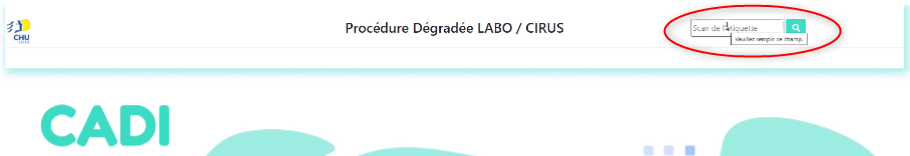
Une image contenant table

Description générée automatiquement

Pour tester les requêtes et vérifier les résultats renvoyer nous avons utilisé Oracle SQL Developer.

Une image contenant table

Description générée automatiquement

Tout d’abord il suffit de scanner une étiquette ou rentrer directement le numéro de demande dans l’input. Puis on presse Enter (1) ou on clique sur la loupe (2) pour lancer la recherche.

Une image contenant texte, intérieur, moniteur

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Ce qui lance une fonction avec des méthodes Ajax pour envoyer le numéro saisi ou scanner vers notre PHP (3), pour effectuer les requêtes vers la base de données. Pour éviter de rentrer une mauvaise valeur ou une injection SQL, la fonction en exécute une autre qui va vérifier que la valeur saisie est au format attendu (4). Une fois exécutée le numéro est envoyé en POST vers nos fichiers PHP en appelant une fonction dans ces fichiers (5).

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Dans notre PHP, on commence par établir une connexion avec la base de données (6) puis on lance nos requêtes (7) pour récupérer, dans différentes variables, les résultats (8).

Une image contenant texte

Description générée automatiquement

Enfin il ne nous reste plus qu’à parcourir nos résultats et les afficher dans notre html grâce à des echo qui renverrons un code html avec les variables PHP (10) et l’Ajax qui retournera ce code dans la <div> prévu à cet effet (9).

Une image contenant texte

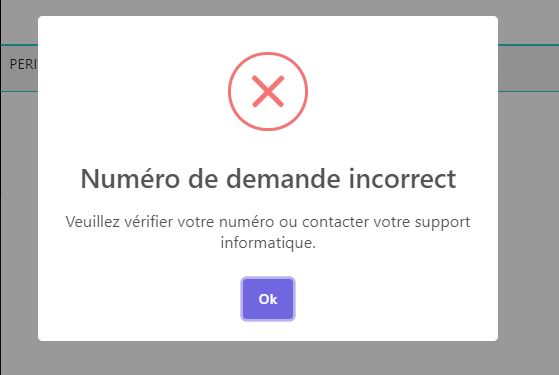
Description générée automatiquementUne image contenant texte

Description générée automatiquement

Si le numéro ne renvoie aucun résultat, alors une Modal va apparaitre invitant l’utilisateur à vérifier le code saisi (11).

Une image contenant texte

Description générée automatiquement



Au préalable nous avons préparé dans notre HTML des <div> avec des id pour y afficher les différents résultats.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementPour les impressions nous passons par un bouton ouvrant une modal. Cette dernière est générée par PHP et gérer en JavaScript. Avec les résultats de la requête on récupère les spécialités pour les placer dans une checkbox (13) qui nous permettra de sélectionner les spécialités que nous souhaitons imprimer en cliquant sur le secteur. Nous avons également mis un bouton pour sélectionner tous les secteurs et un pour les dé-sélectionner rapidement.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

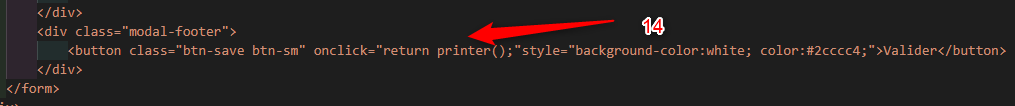
Description générée automatiquement

Une image contenant texte

Description générée automatiquementUne image contenant texte

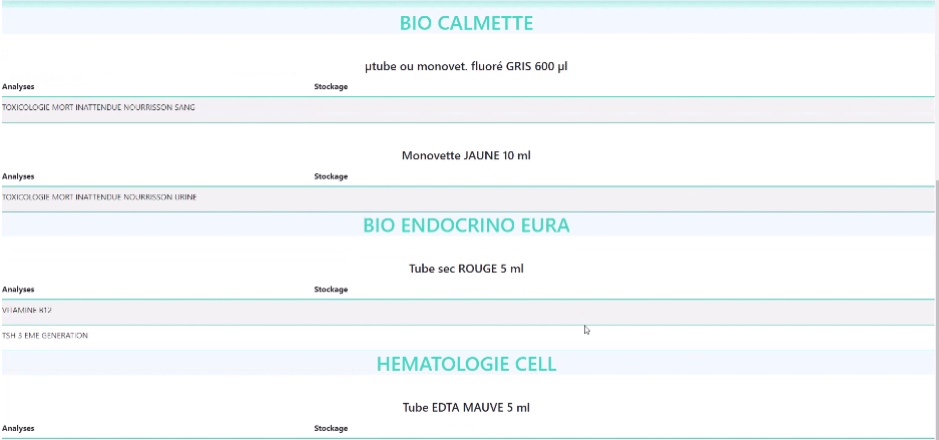
Description générée automatiquement

Une fois le ou les secteurs sélectionnés le bouton « Valider » lance la fonction « printer »(14) qui récupèrera les secteurs sélectionnés, et les stockera en locale pour être récupéré par le PHP de l’impression, une nouvelle page qui mettra en page les informations du patient et les secteurs sélectionnés sera par la suite ouverte (15).



Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Les autres fonctions suivent le même principe, Ajax envoie le numéro de demande vers notre PHP qui se connecte à notre base de données, on effectue une ou plusieurs requêtes, selon dans quelle table se situe les informations rechercher. On stocke dans une ou plusieurs variables les résultats, on les parcourt et via des echo on écrit le code HTML avec les variables PHP. Ce code sera placé, grâce à l’Ajax, dans la <div> correspondante.

## Difficulté rencontrée

Un des problèmes rencontrés était de transférer notre numéro de demande et nos secteurs sélectionnés de notre Javascript vers le la page gérant l’impression. Jusqu’à présent nous passions par Ajax pour que PHP récupère la donnée et affiche le résultat dans notre page d’accueil. Ici on change de page.

Après recherche nous avons retenu la solution du localStorage qui nous a parue être la solution la plus simple et la plus rapide.

Les données stockées dans le localStorage n'ont pas de délai d'expiration, alors que les données stockées dans le sessionStorage sont nettoyées quand la session navigateur prend fin — donc quand on ferme le navigateur.

Une image contenant texte

Description générée automatiquementOn stocke donc nos variables JS en local avec un setItem (17) que l’on récupère simplement avec un getItem(18)

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

# Conclusion

Ce stage a été très enrichissant et m’a permis de mieux me familiariser avec ce métier, l’organisation autour d’un projet et le travail d’équipe dans ce métier.

Tout au long de cette période, j’ai découvert le monde hospitalier autrement qu’en tant que patient. J’ai pu travailler dans des conditions favorables avec une bonne équipe et un tuteur de stage qui prend le temps de s’occuper de ses stagiaires.

J’ai également découvert certains corps de métiers très intéressants au sein de la DRN. Cela m’a conforté dans mon choix et me donne bon espoir pour l’avenir.

# Annexes

## ArborescenceUne image contenant texte Description générée automatiquement

Une image contenant texte

Description générée automatiquementUne image contenant texte, capture d’écran, écran

Description générée automatiquementUne image contenant texte

Description générée automatiquementLe code HTML : l’index

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementLe CSS

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

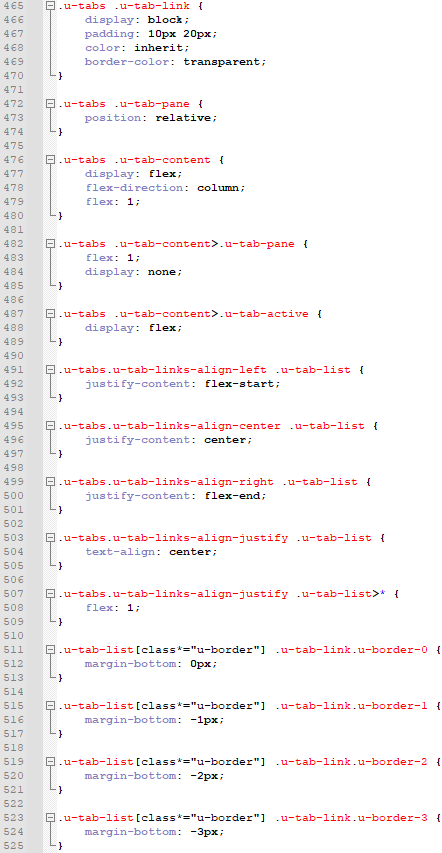
Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

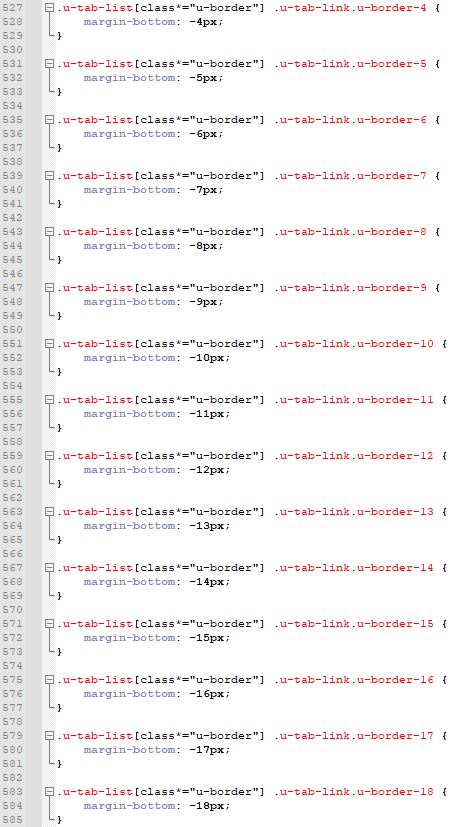
Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

## Le JavaScript

Une image contenant texte

Description générée automatiquement



Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant table

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

## Le PHP : header et tableau d’analyse

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement





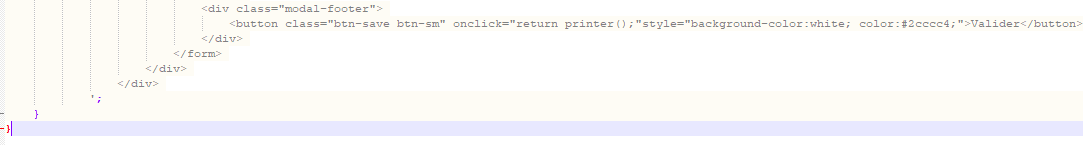
## Le PHP : modal d’impression

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement



## Le PHP : modal de renseignement clinique



## Le HTML : impression

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementLe CSS : impression

## Le JavaScript : impression

Une image contenant texte

Description générée automatiquement