

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement 2016-2017

Outil de gestion de parcours patients dans un hôpital de jour

Tuteur académique
Yannick KERGOSIEN

Étudiant
Guillaume POCHE (DI5)

3 avril 2017



Liste des intervenants

| Nom | Email | Qualité |
|-------------------|--|--|
| Guillaume Pochet | guillaume.pochet@etu.univ-tours.fr | Étudiant DI5 |
| Yannick Kergosien | yannick.kergosien@univ-tours.fr | Tuteur académique, Département Informatique |



Avertissement

Ce document a été rédigé par Guillaume Pochet susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Yannick Kergosien susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Guillaume Pochet, *Outil de gestion de parcours patients dans un hôpital de jour*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2016-2017.

```
@mastersthesis{
  author={Pochet, Guillaume},
  title={Outil de gestion de parcours patients dans un hôpital de jour},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2016-2017}
}
```

Table des matières

| | |
|---|----------|
| Liste des intervenants | a |
| Avertissement | b |
| Pour citer ce document | c |
| Table des matières | i |
| Table des figures | iv |
| Liste des tableaux | vi |
| Introduction | 1 |
| I Recherche | 2 |
| 1 Présentation du problème | 3 |
| 1 Contexte et définition..... | 3 |
| 2 Identification du besoin | 3 |
| 3 Définition des objectifs..... | 3 |
| 4 Description générale..... | 4 |
| 4.1 Environnement du projet | 4 |
| 4.2 Caractéristiques des utilisateurs | 4 |
| 4.3 Fonctionnalités du système | 5 |
| 4.4 Structure générale du système | 5 |
| 5 Contraintes liées au projet..... | 5 |

| | | |
|-----------|--|-----------|
| 2 | Etat de l'art | 6 |
| 1 | Veille technologique | 6 |
| 1.1 | FullCalendar | 6 |
| 1.2 | WdCalendar | 7 |
| 1.3 | AngularBootstrapCalendar | 7 |
| 1.4 | Conclusion | 8 |
| 2 | Logiciels existants..... | 9 |
| 3 | Positionnement du projet | 9 |
| 3 | Analyse et conception | 10 |
| 1 | Conception des fonctionnalités..... | 10 |
| 1.1 | Visualiser son planning (patient)..... | 10 |
| 1.2 | Affecter patient à un parcours | 10 |
| 1.3 | Suivi de l'avancement du patient dans son parcours..... | 11 |
| 1.4 | Visualiser et modifier le planning de toutes les ressources..... | 11 |
| 4 | Plan de développement | 12 |
| 1 | Découpage en tâches | 12 |
| 2 | Sprint 1..... | 14 |
| 3 | Sprint 2 | 14 |
| 4 | Sprint 3 | 14 |
| 5 | Sprint 4 | 14 |
| 6 | Sprint 5 | 14 |
| 7 | Matrice des risques | 15 |
| 8 | Cartographie des risques..... | 15 |
| 5 | Bilan et conclusion | 16 |
| 1 | Avancement du projet | 16 |
| 1.1 | Tâches réalisées..... | 16 |
| 2 | Planning du semestre 10 | 16 |
| II | Développement | 17 |
| 6 | Méthodologie de suivi et gestion de projet | 18 |
| 1 | Suivi par l'encadrant | 18 |
| 2 | Versionning et gestion de projet..... | 18 |
| 7 | Mise en oeuvre | 19 |
| 1 | Fonctionnement général..... | 19 |
| 2 | Conception de la planification | 19 |

| | | |
|-------------------------------------|--|-----------|
| 2.1 | Principe | 20 |
| 2.2 | Choix d'implémentation..... | 20 |
| 2.3 | Limites | 20 |
| 3 | Sauvegarder et restaurer le planning | 21 |
| 3.1 | Principe | 21 |
| 3.1.1 | Sauvegarder | 21 |
| 3.1.2 | Restaurer | 21 |
| 3.2 | Limites | 22 |
| 8 | Validation et tests | 23 |
| 1 | Tests unitaires | 23 |
| 2 | Tests fonctionnels | 23 |
| 9 | Avancement du projet | 24 |
| 1 | Réalisé | 24 |
| 2 | Reste à faire..... | 24 |
| 10 | Reproductivité | 26 |
| 1 | Prérequis et installation | 26 |
| 2 | Déploiement | 26 |
| 3 | Documentation | 26 |
| 11 | Conclusion | 28 |
| Annexe | | 29 |
| Maquettes détaillées | | 30 |
| Spécifications | | 33 |
| Modèle conceptuel de données | | 35 |
| Diagramme de Gantt | | 38 |
| Guide développeur | | 40 |
| Rapport de tests | | 43 |
| Analyse de code (Sonar Cube) | | 44 |
| Librairies utilisées | | 46 |
| Bibliographie | | 47 |
| Comptes rendus hebdomadaires | | 48 |

Table des figures

1 Présentation du problème

| | | |
|---|------------------------------------|---|
| 1 | Diagramme de déploiement..... | 4 |
| 2 | Cas d'utilisation du système | 5 |

2 Etat de l'art

| | | |
|---|--|---|
| 1 | Exemple d'utilisation de wdCalendar..... | 7 |
| 2 | Exemple d'utilisation de angularCalendar | 8 |
| 3 | Exemple d'utilisation de fullCalendar 1 | 8 |
| 4 | Exemple d'utilisation de fullCalendar 2..... | 9 |

4 Plan de développement

| | | |
|---|---------------------------------------|----|
| 1 | Liste des tâches de notre projet..... | 12 |
| 2 | Cartographie des risques..... | 15 |

6 Méthodologie de suivi et gestion de projet

| | | |
|---|----------------------------|----|
| 1 | Versionning du projet..... | 18 |
|---|----------------------------|----|

7 Mise en oeuvre

| | | |
|---|-------------------------------|----|
| 1 | Exemple de planification..... | 20 |
| 2 | Sauvegarder le planning..... | 21 |
| 3 | Restaurer le planning | 21 |

10 Reproductivité

| | | |
|---|--|----|
| 1 | Modèle de documentation - classe | 27 |
|---|--|----|

| | | |
|-------------------------------------|---|----|
| 2 | Modèle de documentation - méthode | 27 |
| Maquettes détaillées | | |
| 1 | Valider activité..... | 30 |
| 2 | Affecter patient..... | 31 |
| 3 | Affecter patient 2..... | 31 |
| 4 | Suivre patient dans son parcours | 32 |
| 5 | Visualiser palnning vue patient | 32 |
| Spécifications | | |
| 6 | Phase de de planification | 34 |
| Modèle conceptuel de données | | |
| 7 | Modèle conceptuel de données | 35 |
| Diagramme de Gantt | | |
| 8 | Diagramme de Gantt prévisionnel..... | 39 |
| Guide développeur | | |
| 9 | Structure de notre application | 40 |
| 10 | Contenu du répertoire app..... | 41 |
| 11 | Exemple de contrôleur | 41 |
| 12 | Exemple de modèles..... | 42 |
| 13 | Vue générale | 42 |
| 14 | Code changement de vue | 42 |
| Rapport de tests | | |
| 15 | Résultats des tests..... | 43 |
| Analyse de code (Sonar Cube) | | |
| 16 | Analyse de code - Sonar - partie 1..... | 44 |
| 17 | Analyse de code - Sonar - partie 2..... | 45 |
| 18 | Analyse de code - Sonar - partie 3..... | 45 |



Liste des tableaux

4 Plan de développement

| | | |
|---|---------------------------|----|
| 1 | Analyse des risques | 15 |
|---|---------------------------|----|



Introduction

Dans le cadre de mon projet de recherche et développement (PRD) au sein de l'école polytechnique de Tours, je suis amené à développer une application permettant la gestion des parcours patients dans un hôpital de jour.

Le projet s'inscrit en deux grands thèmes, d'un côté, la gestion des utilisateurs (système d'informations) et d'un autre côté, l'optimisation des parcours patients. Pour mon projet de recherche et développement, j'ai décidé de me consacrer en majeure partie au système d'informations.

Le système d'informations a été en partie réalisé par six étudiants lors de leur 5ème année, plusieurs fonctionnalités ont déjà été implémentées (gestion des patients, des ressources matérielles ou humaines, etc.). La partie optimisation des parcours patients a été réalisée par un étudiant dans le cadre de son PRD.

Mon objectif dans ce projet est de reprendre l'application et d'intégrer un module permettant la gestion des plannings (planning des patients, planning des ressources) et également de pouvoir suivre en temps réel l'avancement du patient dans son parcours.

Ce projet a été proposé par Lucie ROUSSEL, ingénieur en ordonnancement à l'assistance Publique – Hôpitaux de Paris (AH-HP), et est encadré par Yannick KERGOSIEN, enseignant chercheur à l'école Polytechnique de Tours.

Première partie

Recherche

1

Présentation du problème

1 Contexte et définition

L'Assistance Publique – Hôpitaux de Paris (AP-HP) envisage un regroupement de différents services à l'hôpital Antoine Bécère (Clamart, 92) dans le cadre de son plan stratégique 2015 - 2019. Le regroupement de ces différents services permettra une optimisation des ressources et de l'espace sur le site.

Ce regroupement nécessite le développement d'un système d'informations permettant la gestion des parcours patients (succession d'activités médicales) mais également le développement d'algorithmes permettant la planification des différents parcours.

2 Identification du besoin

Le but de ce projet est de concevoir un outil de gestion de parcours patients en temps réel. L'outil doit permettre d'ajouter des données sur les patients, les ressources mais également de pouvoir visualiser les différents plannings de ces dernières. De plus le patient devra pouvoir consulter son planning en temps réel (connaissance de son prochain rendez vous, horaire, retard, etc.).

Dans un premier temps, la gestion des différents plannings se fera de manière manuelle (sous forme de gantt interactif) mais il ne faudra pas oublier que la planification pourra se faire de manière automatique.

3 Définition des objectifs

Une première version du système d'informations a déjà été développée par six étudiants de 5ème année au cours de l'année 2015-2016. Les algorithmes de planification ont également été développés par un étudiant dans le cadre de son projet de recherche et développement.

Mon travail consiste à reprendre ce système d'informations et à développer un module permettant la visualisation et la modification des parcours patients sous forme de Gantt interactif et également de pouvoir suivre en temps réel l'évolution d'un patient dans son parcours.

4 Description générale

4.1 Environnement du projet

Ce projet est la reprise de deux projets existants, un projet commencé par un groupe de 6 étudiants lors de leur 5ème année à l'école polytechnique de Tours et un projet réalisé par un étudiant dans le cadre de son projet de recherche et développement.

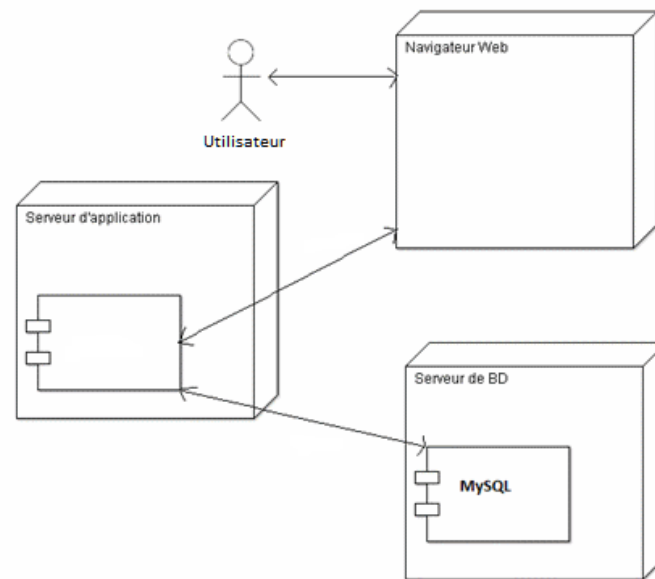


Figure 1 – Diagramme de déploiement

L'application existante est exécutée sur un serveur en local. Le serveur de base de données (MySQL) est également en local.

4.2 Caractéristiques des utilisateurs

Dans notre projet, on peut différencier 3 types d'utilisateurs de l'application web :

- Les patients : Accès limité à l'application, ils pourront uniquement visualiser leur planning et suivre l'avancement dans leur parcours (sans indications horaires excepté sur la prochaine activité).
- Le personnel de soins : Ils auront accès à l'ajout / la recherche de patients, l'accès à leur planning et à la visualisation d'un planning d'un patient.
- Administrateurs et infirmières de coordination : C'est l'utilisateur qui a tous les droits sur l'application. Il peut visualiser/modifier les plannings de toutes les ressources, ajouter/-modifier une activité, gérer les ressources (humaines ou matérielles), ajouter/modifier un parcours patient.

Les différents utilisateurs de l'application devront s'authentifier pour pouvoir accéder aux différentes fonctionnalités.

4.3 Fonctionnalités du système

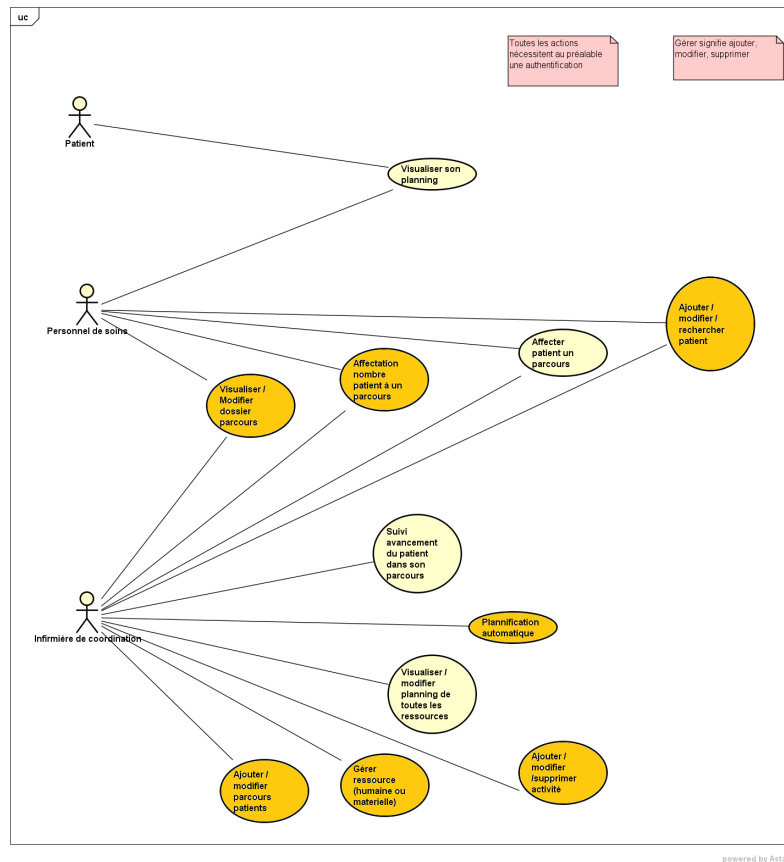


Figure 2 – Cas d'utilisation du système

Les fonctionnalités en orange sont déjà présentes dans l'application.

4.4 Structure générale du système

La structure du logiciel est basée sur l'architecture MVC (modèle vue contrôleur). Le modèle représente toutes les données de l'application (patient, ressources humaines, ressources matérielles, parcours, activités, etc.). Le contrôleur est utilisé pour faire le lien entre les données et les différentes vues. Ce dernier permet également de gérer la cohérence et l'intégrité des données. Les vues représentent les pages web de l'application. La base de données est identique aux projets réalisés précédemment (voir annexe).

5 Contraintes liées au projet

Pour la mise en place de l'application web, je vais reprendre les différents langages utilisés pour le développement du système d'informations (HTML, CSS, JavaScript, PHP...). Je vais également suivre la même architecture de l'application (MVC) à l'aide du Framework PHP (codeIgniter).

La réalisation des différents plannings sera réalisée à l'aide de la librairie fullCalendar/timeline.

2

Etat de l'art

1 Veille technologique

Pour l'affichage et la modification des plannings, j'ai besoin de trouver une librairie qui nous permet de faire cela. Dans notre cas nous avons besoin d'une librairie qui puisse afficher le calendrier sous forme de semaine, qu'on puisse effectuer des drags and drop d'événements.

Après différentes recherches, voici les principales librairies qui existent pour répondre à notre besoin :

- FullCalendar
- WdCalendar
- AngularBootstrapCalendar

1.1 FullCalendar

FullCalendar est une librairie JavaScript permettant la gestion des calendriers. Elle permet de pouvoir drag and drop des événements, de pouvoir modifier la durée des événements et surtout de pouvoir afficher le calendrier de manière simple.

Points forts :

- Design
- Beaucoup de documentations
- Simple d'utilisation
- Possibilité d'afficher les ressources à coté du calendrier
- Beaucoup de fonctionnalités

Points faibles :

- Impossible d'ajouter ou modifier des événements (de manière simple)

1.2 WdCalendar

WdCalendar est un plugin JQuery permettant la gestion des calendriers de manière interactif. Il est possible de faire des clics droits sur le calendrier ce que fullCalendar ne permet pas. De plus le drag and drop est également présent.

Points forts :

- Ajout et modification d'événements sur le calendrier
- Simple d'utilisation

Points faibles :

- Peu de documentations
- Design
- Pas de visualisation des ressources à côté du calendrier

Exemple d'utilisation :

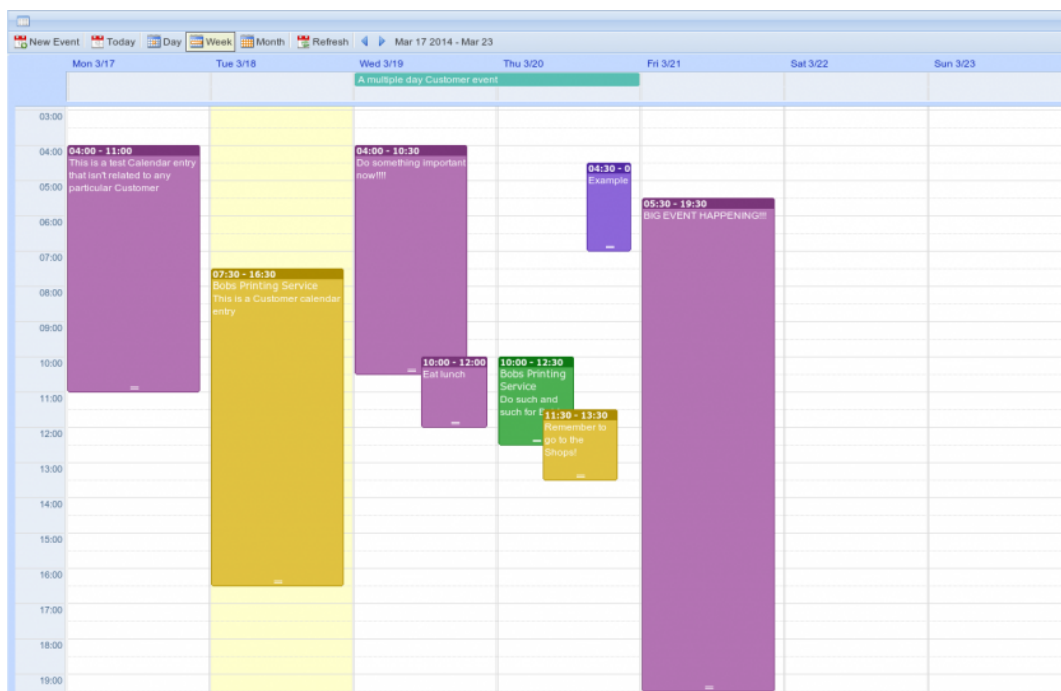


Figure 1 – Exemple d'utilisation de wdCalendar

1.3 AngularBootstrapCalendar

AngularBootstrapCalendar est une librairie JavaScript développée sous le Framework AngularJS. Cette dernière possède toutes les fonctionnalités requises pour notre projet.

Points forts :

- Design
- Ajout et modification d'événements sur le calendrier

Points faibles :

- Connaissances en AngularJS
- Difficile à utiliser
- Peu de documentations

Exemple d'utilisation :

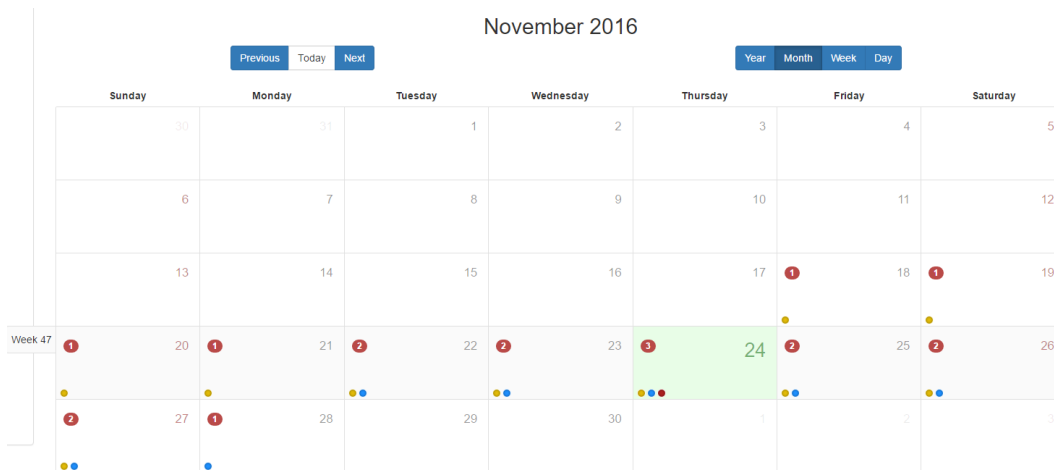


Figure 2 – Exemple d'utilisation de angularCalendar

1.4 Conclusion

Les trois librairies ci-dessus sont assez semblables, elles proposent à peu près les fonctionnalités dont j'ai besoin. Mon choix se porte sur la librairie fullCalendar car cette dernière est plus simple d'utilisation (pour l'utilisateur). De plus la librairie possède une documentation complète.

Voici quelques exemples que j'ai pu réaliser avec cette librairie :

Dans l'exemple, l'affichage du planning s'effectue à la semaine. En haut à gauche se trouve la liste des différentes activités associées à leurs parcours, ces dernières se déplacent dans le calendrier. Elles peuvent être réduites ou agrandies.

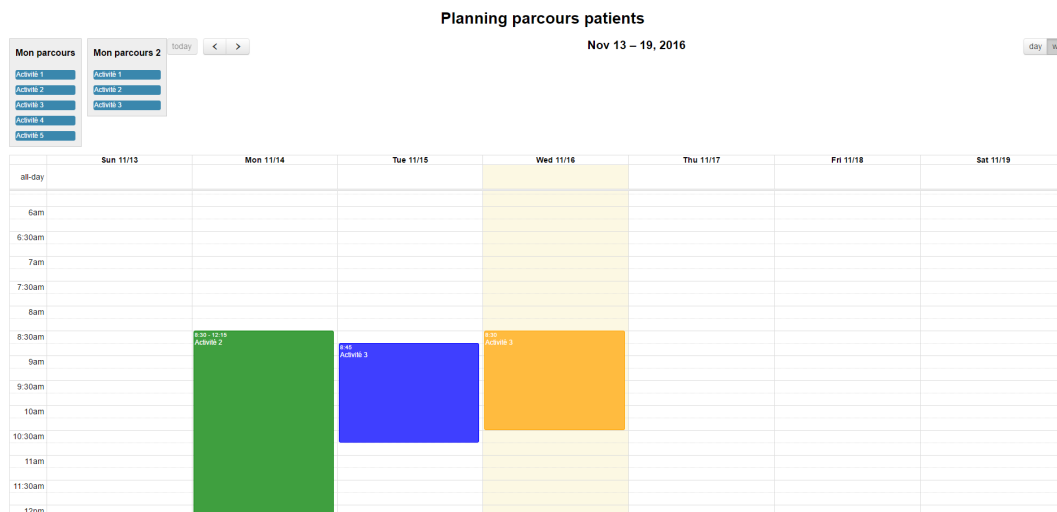


Figure 3 – Exemple d'utilisation de fullCalendar 1

Dans le second exemple, ce trouve le planning pour un jour donné. Une liste de ressources à gauche du tableau est ajoutée, ainsi elle permet la visualisation des activités pour une ressource pour un jour donné.

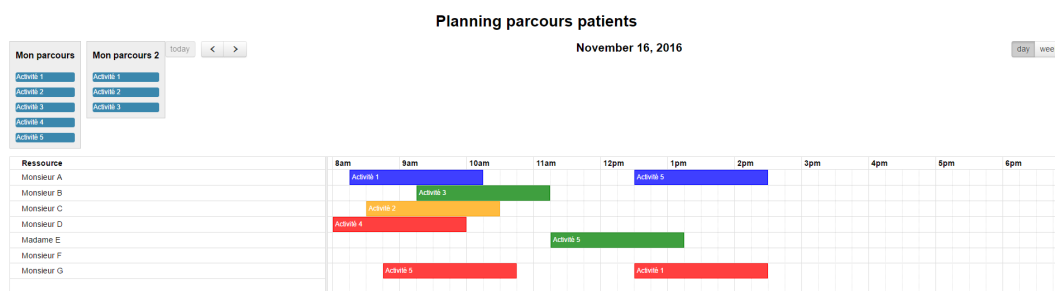


Figure 4 – Exemple d'utilisation de fullCalendar 2

2 Logiciels existants

La société Effigen développe des solutions pour l'optimisation des hôpitaux de jours. Cette optimisation permet de réduire les délais de prise en charge et d'augmenter la productivité.

Leur solution se découpe en plusieurs logiciels, un qui permet de visualiser la planification des parcours patients mais également une solution permettant de reprogrammer les plannings en fonction des flux observés (retard, absence de patient, etc.)

Leur solution permettrait d'augmenter de 40 à 65% le taux de prise en charge des patients et de lisser la charge de travail au cours de la journée et de la semaine.

Gesplic est un logiciel de gestion et de planification des chimiothérapies. Ce dernier permet de suivre et de planifier les différentes séances de chimiothérapies d'un patient. Le logiciel calcule en temps réel les prochaines séances de chimiothérapie en fonction de l'analyse des précédentes.

IC-Plan est un logiciel permettant de gérer les patients dans un hôpital (gestion des lits d'hôpitaux suivie des séjours, etc.). Il existe également plusieurs logiciels de gestion de données sur les patients dans les hôpitaux et de gestion des personnels.

3 Positionnement du projet

Pour le moment, il n'existe pas de logiciels qui répondant au besoin. Les services hospitaliers gèrent de manières différentes leurs prises de rendez-vous. Il existe cependant des logiciels de gestion de suivi de patients mais aucun de ces logiciels permettent de prendre des rendez-vous et d'ainsi pouvoir optimiser le temps d'attente des patients.

3

Analyse et conception

1 Conception des fonctionnalités

Dans ce chapitre, je vais analyser en détail les fonctionnalités que je vais implémenter dans ce projet. La liste des fonctionnalités à faire est :

- Visualiser le planning (patient)
- Affecter un patient à un parcours
- Suivi de l'avancement du patient dans son parcours
- Visualiser et modifier le planning de toutes les ressources

1.1 Visualiser son planning (patient)

Une fonctionnalité « afficher planning du patient » doit être disponible et doit permettre de visualiser l'heure de début et de fin de la prochaine activité du patient dans son parcours. Cette visualisation doit être visible par l'administrateur, le personnel de soins et également par le patient. (Voir annexe)

1.2 Affecter patient à un parcours

Le patient doit pouvoir être affecté à un parcours en précisant les disponibilités de ce dernier. Les disponibilités du patient seront demandées sous formes d'horaires (exemple : 8h-17h). Cette fonctionnalité doit être disponible à l'administrateur et aux personnels de soins.

L'administrateur ou infirmière de coordination doit pouvoir affecter un patient à un parcours pour une journée donnée et pouvoir visualiser également le nombre de patients déjà affecté sur cette dernière. (Voir annexe)

Lors d'un clic sur le bouton affecter, l'administrateur sera redirigé vers une page où il pourra choisir le patient à affecter pour la date choisie précédemment. (Voir annexe)

1.3 Suivi de l'avancement du patient dans son parcours

La fonctionnalité « suivre le patient dans son parcours » doit permettre à l'administrateur et/ou personnel de soins de visualiser l'avancement du patient dans son parcours (si il a bien effectué tel ou tel activité). De plus la ressource qui a consulté le patient doit pouvoir valider que le patient a bien effectué son activité. (Voir annexe)

L'administrateur aura accès à l'avancement de chaque patient dans leur parcours respectifs. Si un patient ne s'est pas présenté à ces rendez-vous, l'administrateur pourra replanifier automatiquement ou manuellement la journée. (Voir annexe)

1.4 Visualiser et modifier le planning de toutes les ressources

L'administrateur doit pouvoir visualiser et modifier en temps réel le planning de toutes les ressources et également d'une ressource en particulier. La modification doit pouvoir s'effectuer de façon manuelle (en déplaçant les activités) mais également de façon automatique. L'administrateur doit pouvoir planifier automatiquement le planning de ses ressources pour une journée donnée. La planification automatique ne sera pas incluse dans le cadre de mon projet de recherche et développement. Ces fonctionnalités ne sont disponibles qu'à l'administrateur.

Les différentes interfaces ont été étudié dans la partie veille technologique sur fullCalendar.

(Voir [Figure 4](#) (Chapitre 2) et [3](#) (Chapitre 2))

4

Plan de développement

| Nom | Date de déb... | Date de fin |
|--|----------------|-------------|
| ◦ Gestion de projets | 21/09/16 | 10/04/17 |
| ◦ Cahier de spécifications | 21/09/16 | 03/11/16 |
| ◦ État de l'art | 04/11/16 | 25/11/16 |
| ◦ Design | 03/12/16 | 10/04/17 |
| ◦ Mise en place des outils et récupération des sources | 26/11/16 | 02/12/16 |
| ◦ Authentification tests | 03/12/16 | 03/12/16 |
| ◦ Gestion des ressources tests | 04/12/16 | 04/12/16 |
| ◦ Gestion des activités tests | 05/12/16 | 05/12/16 |
| ◦ Gestions des patients tests | 06/12/16 | 06/12/16 |
| ◦ Affichage planning vue patient | 10/03/17 | 16/03/17 |
| ◦ Affichage planning vue ressource | 03/03/17 | 09/03/17 |
| ◦ Affichage et gestion vue patient | 13/01/17 | 09/02/17 |
| ◦ Affichage et gestion vue ressource | 10/02/17 | 02/03/17 |
| ◦ Ergonomie des parcours patients | 07/12/16 | 20/12/16 |
| ◦ Affichage en temps réel | 17/03/17 | 10/04/17 |
| ◦ Gestion des rendez vous | 21/12/16 | 12/01/17 |

Figure 1 – Liste des tâches de notre projet

Le projet est à terminer pour début avril. J'ai choisi de travailler avec une méthode agile. Plusieurs rendez-vous réguliers avec mon tuteur académique et le client auront lieu afin d'avoir des retours réguliers sur le projet et sur les attentes du client. Chaque sprint amènera à une validation de ce qu'il a été fait précédemment.

1 Découpage en tâches

Tâche 0 : Mise en place des outils et récupération des sources

Priorité : Haute

Durée estimé : 2 jours

Tâche 1 : Gestion de projet (réunion, rédaction des rapports)

Priorité : Haute

Durée estimé : Toute la durée du projet

Tâche 2 : Authentification tests

Priorité : Haute

Durée estimé : 0.5 jour

Tâche 3 : Gestion des ressources tests

Priorité : Haute

Durée estimé : 0.5 jour

Tâche 4 : Gestion des activités tests

Priorité : Haute

Durée estimé : 0.5 jour

Tâche 5 : Gestion des patients tests

Priorité : Haute

Durée estimé : 0.5 jour

Tâche 6 : Affichage planning vue patient

Priorité : Haute

Durée estimé : 2 jours

Tâche 7 : Affichage planning vue ressource

Priorité : Haute

Durée estimé : 2 jours

Tâche 8 : Affichage et gestion vue général patient

Priorité : Haute

Durée estimé : 8 jours

Tâche 9 : Affichage et gestion vue général ressource

Priorité : Haute

Durée estimé : 6 jours

Tâche 10 : Gestion des rendez-vous

Priorité : Haute

Durée estimé : 6 jours

Tâche 11 : Design

Priorité : Basse

Durée estimé : Toute la durée du projet

Tâche 12 : Ergonomie de la construction des parcours patients

Priorité : Haute

Durée estimé : 4 jours

Tâche 13 : Affichage en temps réel

Priorité : Basse

Durée estimé : 8-10 jours

Tâche 14 : Cahier de spécifications – compréhension du besoin

Priorité : Haute

Durée estimé : 11 jours

Tâche 15 : État de l'art

Priorité : Haute

Durée estimé : 6 jours

2 Sprint 1

Le Sprint 1 comprendra toutes les tâches de tests de l'existant, c'est-à-dire les tâches numérotées précédemment de 2 à 5. Ce sprint contiendra également la gestion des parcours patients (tâche 12) et la mise en place des outils (dépôts SVN, récupération de l'existant). Le livrable de ce sprint sera un projet où toutes les fonctionnalités de gestions seront présentes.

3 Sprint 2

Le Sprint 2 contiendra la tâche 10 qui est la gestion des rendez-vous. La gestion des rendez-vous est une étape primordiale de l'application c'est ce qu'il permettra de planifier ces derniers. Ce Sprint amènera à la livraison d'un produit fonctionnel contenant toutes les fonctionnalités de gestions de l'application.

4 Sprint 3

Le Sprint 3 sera composé des tâches 8 et 9 (visualisation et modification des plannings patients et ressources). Le livrable de ce sprint sera une application permettant de planifier les différents parcours patients pour une journée donnée de manière manuelle.

5 Sprint 4

Le Sprint 4 est la suite du Sprint 3, il permettra aux patients de pouvoir visualiser leur planning (de même pour les ressources). Le livrable sera une application fonctionnelle où chaque utilisateur pourra visualiser leurs différents plannings.

6 Sprint 5

Le Sprint 5 comprendra la tâche 13, l'affichage et la gestion des plannings en temps réels. Les tâches 1 et 11 (gestion de projet et design de l'application) sont communes à tous les sprints.

7 Matrice des risques

Les tâches 2, 3, 4 et 5 sont des tâches de tests de l'existant. Cependant il existe un risque de retard dans la suite du développement si une ou plusieurs de ces tâches ne passent pas les tests.

Table 1 – *Analyse des risques*

| Risques | Nature |
|---------|--|
| R1 | Retard dans le développement à cause de la monter en compétences sur la technologie fullCalendar. |
| R2 | Retard dans le développement à cause des tests effectués au début du projet (si un des tests n'est pas satisfaisant, cela engendrera des développements supplémentaires) |

8 Cartographie des risques

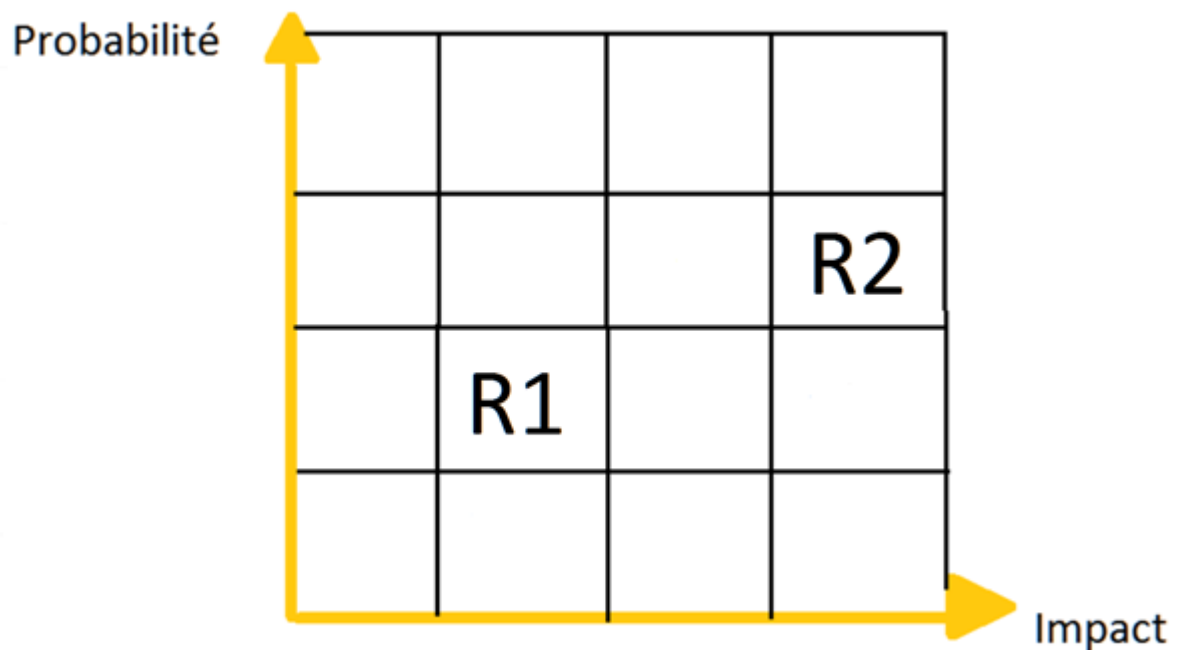


Figure 2 – *Cartographie des risques*

5

Bilan et conclusion

1 Avancement du projet

1.1 Tâches réalisées

Les différentes tâches que j'ai déjà réalisées sont :

- les spécifications
- l'état de l'art et la veille technologique
- la mise en place des outils de gestion de projet (Git, trello)
- la mise en place du projet existant
- tests sur les fonctionnalités existantes

2 Planning du semestre 10

Le planning du semestre 10 contient toutes la phase de développement. Le premier sprint à livrer contiendra tout les tests des différentes fonctionnalités existantes. Ce sprint aboutira à la continuation des tâches initialement prévu si l'existant est correct sinon nous devons refaire un planning en intégrant des nouvelles tâches de développement. (Voir chapitre 4 - plan de développement)

Deuxième partie

Développement

6

Méthodologie de suivi et gestion de projet

1 Suivi par l'encadrant

Concernant la méthodologie de suivi de gestion de projet, nous avons mis en place des rendez vous réguliers avec mon encadrant (environ toutes les 2-3 semaines) afin de vérifier l'avancement du projet. Chaque rendez vous a amené à une petite démonstration de ce qui avait été fait.

A la suite de chaque rendez vous, nous redéfinissions les principaux objectifs à atteindre pour la suite du projet.

2 Versionning et gestion de projet

Pour le versionning du projet, j'ai mis en place un repository sous git où à chaque fin de journée, les sources du projet étaient versionnées. Ce versionning permet de revenir à une ancienne version si nécessaire et également il permet de ne pas perdre les sources du projet en cas de problèmes.

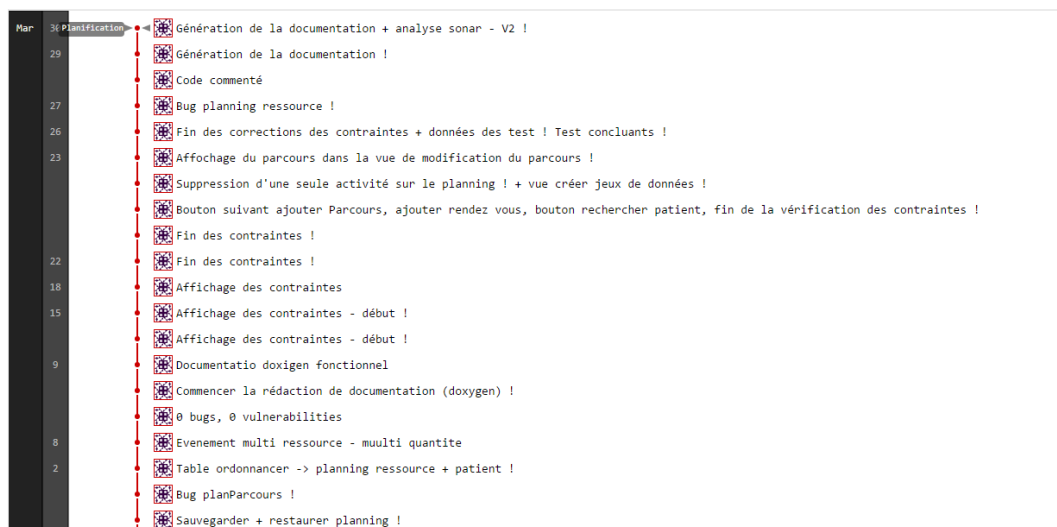


Figure 1 – Versionning du projet

7

Mise en oeuvre

Dans ce chapitre, je vais mettre en avant uniquement la partie importante de mon projet de recherche et développement, c'est à dire la conception d'une planification de parcours patients de manière manuelle et interactive.

1 Fonctionnement général

Considérant pour une journée donnée, un ensemble de parcours à planifier, le principe est de proposer à l'utilisateur l'affectation des différentes activités des parcours patients sur les différentes ressources présentes dans notre système d'informations.

L'utilisateur doit pouvoir déplacer les différentes activités sur le temps tout en ayant un visuel sur le respect des différentes contraintes.

Voici la liste des différentes contraintes que nous devons respecter :

- Vérifier fenêtre de temps du patient
- Délai min et max entre chaque activité d'un même parcours patients
- Précédences dans un parcours patients
- Affecter activité à une ressource
- Toutes les activités sont planifiées
- Ressources synchronisés
- 1 ressource = 1 activité à la fois
- 1 patient = 1 activité à la fois

Néanmoins, l'utilisateur doit rester libre sur ses choix et aucune de ses contraintes ne doit bloquer le système.

2 Conception de la planification

Concernant la conception de la planification, j'ai décidé d'utiliser la librairie fullCalendar. Cette librairie permet d'avoir un affichage dynamique d'un planning.

De plus les différentes requêtes permettant d'ajouter, de supprimer ou de déplacer des événements sur le planning sont exécutées en AJAX permettant ainsi le non rechargement de la page à chaque actions de l'utilisateur.

2.1 Principe

Le principe de la planification est assez simple. Nous disposons d'une liste de parcours à planifier. L'utilisateur peut alors déplacer ces différentes activités sur le planning.

L'application se charge d'afficher le calendrier en récupérant toutes la liste des ressources et toute la liste des activités.

2.2 Choix d'implémentation

Concernant cette planification, j'ai effectué plusieurs choix importants limitant ainsi l'utilisateur :

- Les activités qui sont déplacées dans le calendrier sont directement placées sur les bonnes ressources (humaines et matérielles)
- De plus ces activités sont affectées à la première ressource disponible

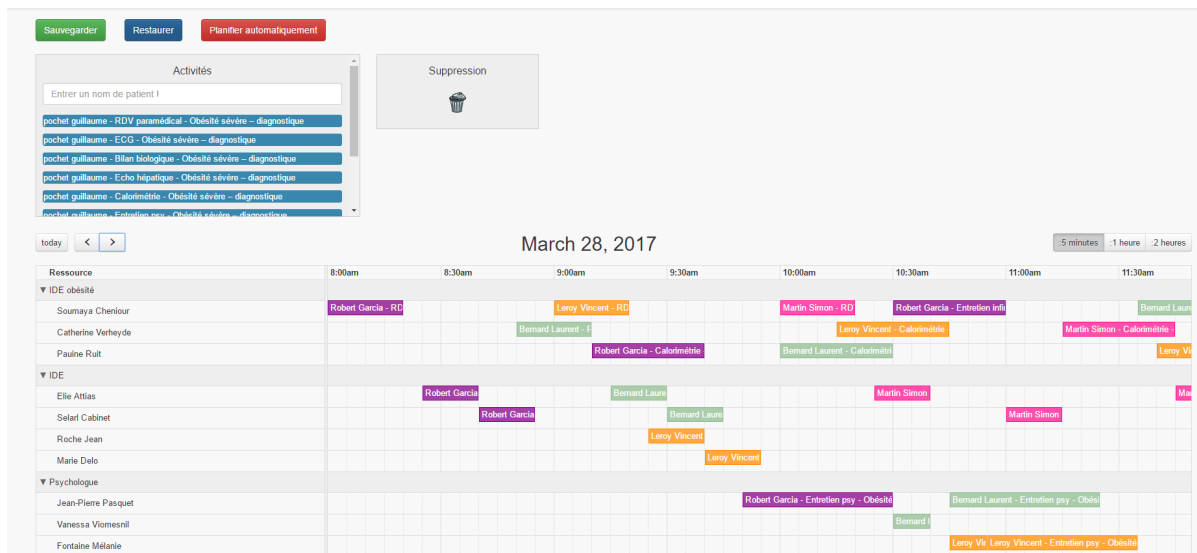


Figure 1 – Exemple de planification

Un panneau d'affichage des contraintes est également présent en dessous du planning.

2.3 Limites

Plusieurs limites peuvent être aperçues :

- Le panneau de la liste des activités peut être trop petit si il y a beaucoup d'activités à planifier
- Impossible de visualiser réellement la planification de chaque patient

Pour remédier à ces limites, nous avons pensé à l'ajout d'une ressource Patient qui contiendrait la liste des patients à planifier pour ce jour. Ainsi chaque activité qui serait déplacée dans le calendrier serait également déplacée sur le patient ce qui permettrait d'avoir une vision rapide du planning d'un patient.

3 Sauvegarder et restaurer le planning

Une partie importante du projet est la sauvegarde et la restauration de la planification des parcours. Voici les principes et les limites de la solution retenue.

3.1 Principe

Dans notre modèle conceptuel de données, nous disposons de deux tables permettant la gestion de la sauvegarde des plannings : la table ordonnancer et la table événements (voir [Figure 7ERROR](#)).

La table ordonnancer correspond à la solution de la planification des ressources et des patients, la table événements est une table temporaire permettant uniquement de récupérer l'état actuel de la planification. C'est la table événements qui est utilisé pour l'affichage de l'onglet planification dans notre système d'informations.

L'avantage d'utiliser ce principe c'est que l'utilisateur peut modifier l'état de son planning, si il ne sauvegarde pas, cela n'aura aucun impact sur la planification des ressources et des patients.

3.1.1 Sauvegarder

La sauvegarde d'un planning s'effectue donc de cette façon :



Figure 2 – Sauvegarder le planning

La fonctionnalité sauvegarder écrase le contenu de la table ordonnancer.

3.1.2 Restaurer

Voici le principe de la fonctionnalité restaurer un planning :

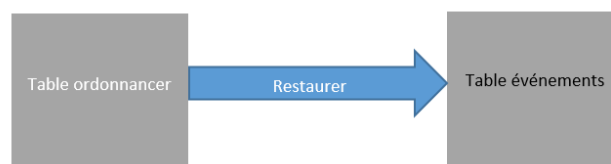


Figure 3 – Restaurer le planning

La fonctionnalité restaurer écrase le contenu de la table événements.

3.2 Limites

Les limites de ce principe peuvent être multiples :

- Duplication des données
- Peut prendre du temps si il faut sauvegarder ou restaurer plusieurs années de planification

Néanmoins pour remédier à ces différents problèmes, nous avons pensé mettre en place un fichier de paramètres qui contiendra l'intervalle de temps des données de planification que l'on souhaite conservé en base de données. (Par exemple : deux jours avant la date courante jusqu'à 30 jours après la date courante).

Cette solution permettrait de diminuer le nombre de données et ainsi améliorer les performances de ces deux fonctions.

8

Validation et tests

1 Tests unitaires

Mon projet consistait principalement à développer un Gantt interactif afin de pouvoir planifier les différents parcours dans un hôpital de jour.

Le projet étant principalement graphique, je n'ai pas mis en place de tests unitaires. Néanmoins, j'ai effectué des données de tests afin de pouvoir tester l'affichage des plannings et surtout le respect des différentes contraintes.

2 Tests fonctionnels

Pour tester l'affichage des données sur le calendrier ainsi que l'affichage des contraintes de planification, j'ai réalisé un jeu de données tests comprenant 5 parcours patients pour une date donnée. La planification de ces parcours représente une solution réalisable (respect des différentes contraintes définies avec mon encadrant).

Ces données de tests m'ont permis de tester toutes les contraintes suivantes :

- Vérifier fenêtre de temps du patient
- Délai min et max entre chaque activité d'un même parcours patients
- Précédences dans un parcours patients
- Affecter activité à une ressource
- Toutes les activités sont planifiées
- Ressources synchronisés
- 1 ressource = 1 activité à la fois
- 1 patient = 1 activité à la fois

Les premiers tests effectués n'étaient pas tous concluants mais des corrections ont été effectuées (Voir rapport de tests).

9

Avancement du projet

Dans ce chapitre, nous allons faire un point sur l'avancement du projet. Lors des différents rendez vous avec mon encadrant, nous avons définis la liste des fonctionnalités restantes.

De plus, un rendez vous à Paris a eu lieu afin de montrer l'avancement du projet au client. Lors de cette réunion, nous avons fait un point sur l'état du projet.

1 Réalisé

Voici une liste des fonctionnalités qui ont été réalisées :

- Gestion des ressources (humaines ou matérielles)
- Saisie des parcours patients
- Saisie des activités ainsi que leurs besoins en ressources
- Visualisation des parcours patients sous formes de graphes
- Prise de rendez vous d'un patient
- Planification manuelle des différents parcours patients
- Sauvegarder et restaurer la planification
- Affichage du planning d'une ressource
- Affichage des rendez vous d'un patient (à revoir)

2 Reste à faire

Voici la liste complète des fonctionnalités restantes qui ont été définis au travers de plusieurs rendez vous :

- Planification en temps réel
- La gestion des dossiers parcours
- Revoir l'affichage du planning d'un patient
- Affichage d'un planning d'une ressource (visualisation des ressources matérielles liées à chaque activité)
- Page à propos (logo de polytech Tours et de la liste des personnes ayant travaillé sur le projet)
- Ajouter des indisponibilités matérielles

- Pouvoir ajouter des indisponibilités fréquentes pour les ressources (humaines et matérielles)
- Gestion des comptes utilisateur (modification de mot de passe, ajout de compte)
- Gérer les archives des précédentes planification
- Revoir la sécurité concernant la suppression d'une activité, d'un parcours, d'une ressource
- Revoir la sauvegarde et la restauration du planning (Voir limites de la solution retenue)
- Planification automatique

10

Reproductivité

1 Prérequis et installation

Le projet a été développé à l'aide de l'IDE netbeans, compatible avec Windows, Mac OS et Linux. La base de données utilisé est une base de données MySQL.

L'IDE est disponible en téléchargement sur le site officiel de netbeans.(<https://netbeans.org/>).

Pour pouvoir utiliser l'application, un serveur web et de base de données sont nécessaires. Pour ma part, j'ai utilisé WAMP (<http://www.wampserver.com/>).

2 Déploiement

Pour déployer l'application, il suffit de réaliser ces différentes étapes :

- Copier toutes les sources dans le dossier www de votre serveur WAMP
- Exécuter le script "subway.sql" dans PhpMyAdmin afin de créer et d'ajouter des données
- Aller sur l'adresse suivante : <http://localhost/subway/>.

3 Documentation

Toute la documentation du projet a été généré avec doxygen. (Voir documentation). La documentation est disponible dans le répertoire doc de l'application.

Chaque classe a été documenté suivant ce modèle : (Voir [Figure 1](#))

Chaque méthode a été documenté suivant ce modèle : (Voir [Figure 2](#))

De plus un guide développeur est disponible en annexe afin d'aider la personne qui sera en charge de la poursuite du projet.

```

/**
 * \file      Activites.php
 * \author    Guillaume Pochet
 * \version   1.0
 * \date      09 Mars 2017
 * \brief     Définit les méthodes liées aux activités
 *
 * \details   Ce fichier permet de définir les méthodes d'ajout,
 *            de suppression et de modification lié aux activités
 */
class Activites extends CI_Controller {

```

Figure 1 – Modèle de documentation - classe

```

/**
 * \brief     Affichage la liste des différentes activités
 * \details   La méthode récupérer la liste des activités
 *            et envoie ces données sur la page V_activite.
 *
 *            Cette page se charge d'afficher les données
 * \param     Aucun
 */
public function index() {

```

Figure 2 – Modèle de documentation - méthode

11

Conclusion

A la suite de ce projet, l'application web permet la saisie de toutes les données métiers (patients, ressources, activités, parcours). De plus, elle permet d'utiliser ces données afin de réaliser une planification de manière interactive.

Néanmoins, il reste des fonctionnalités importantes à implémenter comme la planification en temps réel.

D'un point de vue personnel, le projet a été une réussite, il m'a permis d'améliorer mes compétences dans la programmation web. De plus, le fait d'avoir eu l'occasion de présenter mon travail lors d'un rendez vous à Paris a vraiment été enrichissant.

Annexe

Maquettes détaillées

The screenshot shows a web browser window titled "Interface ressource". The address bar contains "http://www.climab.fr/ressource/1". The browser's navigation buttons (back, forward, stop, home) are visible on the left. Below the address bar is a navigation menu with links: "Accueil", "Patient", and "Planning". On the right side of the menu is a link "Se deconnecter". The main content area has the heading "Suivi patient - Nom Prénom - Activité". Below this heading is a form with the label "Activité réalisé :" followed by two radio buttons: "Oui" and "Non". Below the radio buttons is a button labeled "Valider". The browser window has a standard grey border with a small double-slash icon in the bottom right corner.

Interface ressource

http://www.climab.fr/ressource/1

Accueil Patient Planning Se deconnecter

Suivi patient - Nom Prénom - Activité

Activité réalisé : ☐ Oui ☐ Non

Valider

Figure 1 – Valider activité

The screenshot shows a web browser window titled 'Interface administrateur' with the URL 'http://www.climab.fr/administrateur/'. The navigation bar includes 'Accueil', 'Ressources', 'Activités', 'Parcours', 'Patient', 'Planning', and 'Se deconnecter'. The main content area is titled 'Affecter patient'. On the left, there is a calendar for February 2008. The main content area displays the date '1 Février 2008' and two rows of information: 'Parcours 1 : 10 patients' with 'Places restantes : 1' and 'Parcours 2 : 3 patients' with 'Places restantes : 5'. Each row has an 'Affecter' button next to it.

Figure 2 – Affecter patient

The screenshot shows a web browser window titled 'Interface administrateur' with the URL 'http://www.climab.fr/administrateur/'. The navigation bar is the same as in Figure 2. The main content area is titled 'Affecter patient - Parcours 1'. It contains a form with two fields: 'Patient :' with a 'ComboBox' and a dropdown arrow, and 'Date :' with a text input containing '/' and a calendar icon. Below these fields is a 'Valider' button.

Figure 3 – Affecter patient 2

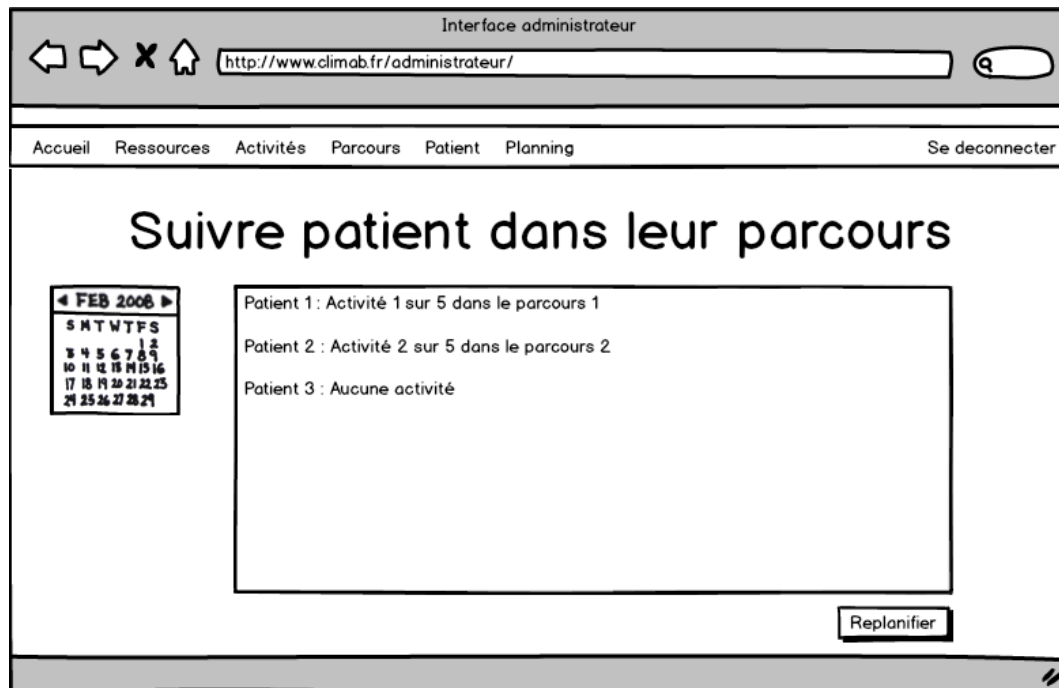


Figure 4 – Suivre patient dans son parcours

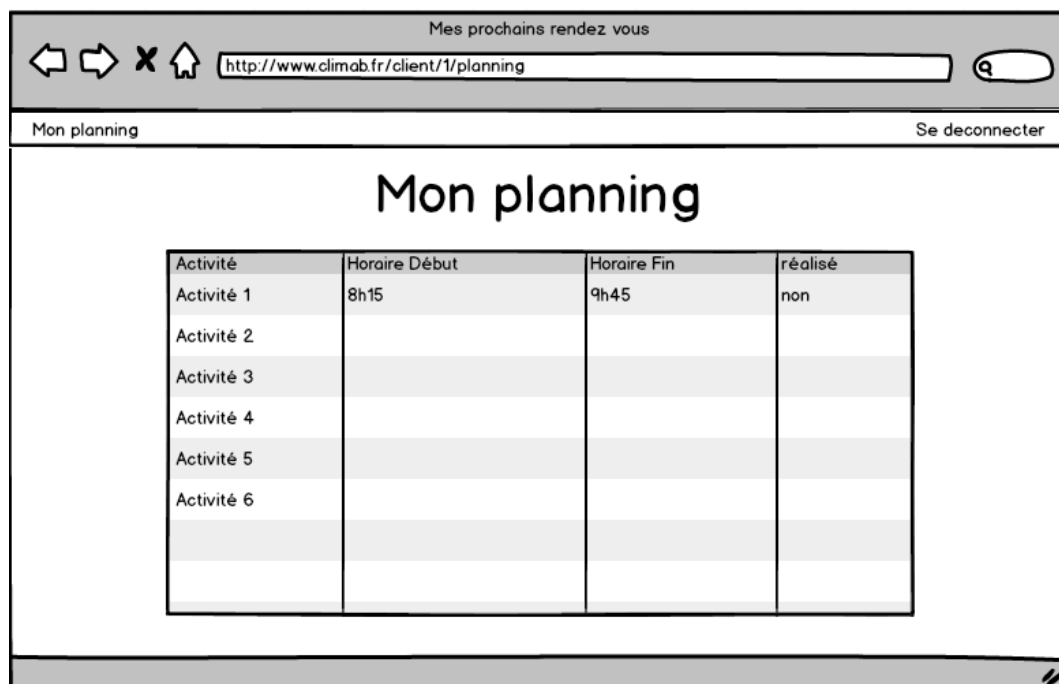


Figure 5 – Visualiser palnning vue patient



Spécifications

Description des interfaces externes du logiciel

Interfaces matériel/logiciel

Dans ce projet, je vais poursuivre le développement d'une interface web. Pour cela, les utilisateurs de l'application devront être munis d'un ordinateur ou smartphone (pour les patients) et d'un navigateur (avec JavaScript activé).

L'activation de JavaScript est importante pour la visualisation des différents plannings. L'interface communiquera avec une base de données pour pouvoir récupérer les différentes informations de l'application (liste des ressources, parcours patients, planning, etc.).

Interface homme/machine

Les utilisateurs de l'application auront tous un accès à l'application. L'authentification à l'aide d'un nom d'utilisateur et d'un mot de passe définira le type d'utilisateur.

Par conséquent, le type d'utilisateur définira les différents onglets ou pas à afficher.

Il existe 3 types d'utilisateurs (patient, ressource et administrateur ou infirmière de coordination).

Interface logiciel/logiciel

Concernant les interfaces logiciel / logiciel, notre application web sera connectée à une base de données pour pouvoir sauvegarder et récupérer les données nécessaires au fonctionnement de l'application.

De plus, notre application utilisera des fichiers pour communiquer avec une application externe pour pouvoir effectuer la planification automatique.

Planification automatique

Pour la partie planification automatique, notre système devra prévoir 3 phases de planifications.

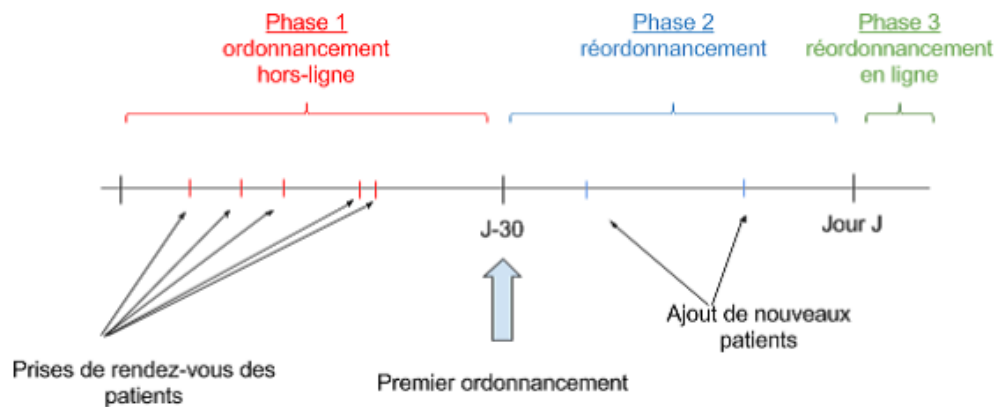


Figure 6 – Phase de de plannification

Modèle conceptuel de données

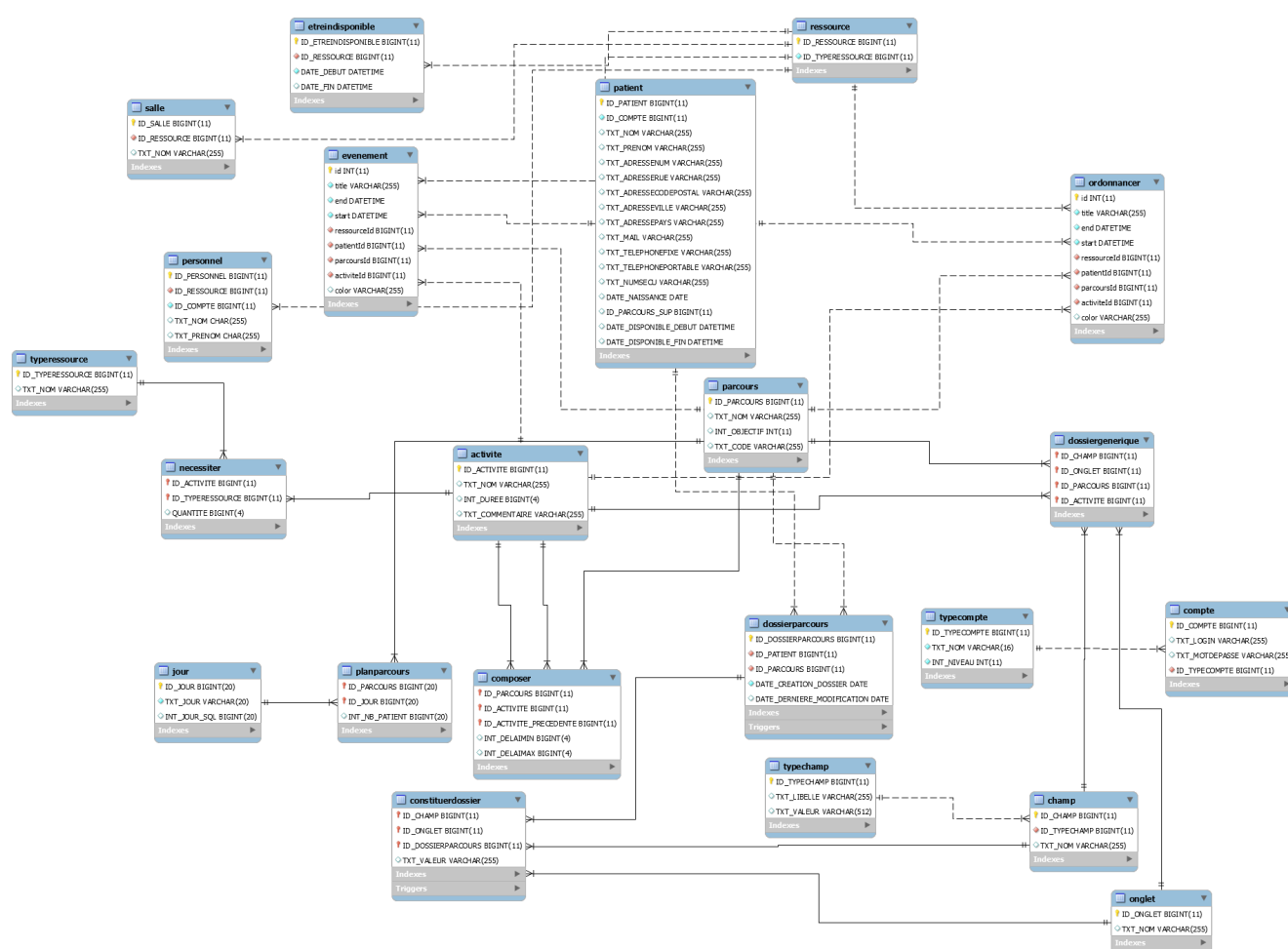


Figure 7 – Modèle conceptuel de données

Notre Système d'Information est basé sur 22 tables. Nous allons maintenant faire un petit détail de l'ensemble des tables afin de connaître leur utilité.

Explication des tables de notre modèle

Activité : Table regroupant l'ensemble des informations concernant une activité. C'est cet élément qui constitue les parcours. Une activité peut obliger la réalisation de certaines autres activités avant d'être réalisable. Cette notion de dépendance dépend du parcours en cours. Cette notion de précédence est sera vue plus loin via la table composer. En plus de cela, une activité a besoin de ressources. Ce lien se fait par la table nécessiter.

Champ : Table contenant l'ensemble des champs qu'il est possible d'ajouter dans un onglet d'un dossier Parcours. A chaque champ est lié un type de champ, que nous verrons plus loin, avec la table typechamp.

Composer : Table permettant de faire le lien entre un parcours et une activité. Par l'intermédiaire de cette table nous pouvons dire en fonction d'un parcours et d'une activité s'il y a des besoins en termes de précédence. Chaque ligne, dans cette table, a pour signification : « Pour l'activité A dans le parcours P, il faut avoir réalisé l'activité B avant, et ce dans un délai compris entre delaiMin et delaiMax. ». Il est important de noter qu'il est possible de mettre à « null », l'id de l'activité précédente.

Compte : Table regroupant l'ensemble des comptes qu'ils soient des comptes patient ou des comptes de type ressource médicale.

Constituerdossier : Table permettant de faire le lien entre un dossier parcours et les informations qui le constituent. En effet, nous retrouvons pour chaque dossier parcours et pour chaque onglet dans ce dernier, la valeur des champs le composant.

Dossiergenerique : Table définissant, pour un parcours, les onglets et les champs que tous les dossiers parcours doivent avoir impérativement.

Dossierparcours : Table renseignant les informations génériques d'un dossier parcours. Soit le patient associé, le parcours, dates de création et de dernière modification.

Etreindisponible : Table regroupant l'ensemble des indisponibilités pour une ressource. Cette indisponibilité est caractérisée par une date de début et de fin, acceptant toutes les deux le renseignement de l'heure.

Jour : Table qui contient les jours de semaine, ainsi que leurs index sous MySQL. Cette table a un enjeu au niveau des prévisions du nombre de patient par jour pour un parcours.

Nécessiter : Table permettant de renseigner les types de ressources requises pour une activité, ainsi que la quantité nécessaire.

Onglet : Table des onglets disponibles pour constituer un dossier parcours.

Parcours : Table décrivant un parcours de façon générale.

Personnel : Table contenant l'ensemble de personnel médical de l'établissement. Chaque personnel a un compte, et est considéré comme une ressource.

Planparcours : Table regroupant l'ensemble des objectifs concernant le nombre de patients pour un parcours pour un jour donné.

Ressource : Table faisant le lien entre la table typeressource et personnel ou materiel. Ce lien sera expliqué plus en détail avec la table typeressource.

Typechamp : Table regroupant les différents type de champ qu'il est possible d'ajouter dans un dossier parcours. Elle contient également le code HTML des composants, permettant ainsi une mise en page en accord avec les autres éléments des pages.

Typecompte : Table utilisée pour la gestion des droits.

Typersource : Table contenant les types de ressource d'un point de vue activité. En effet, une activité peut avoir besoin d'un type de ressource bien caractéristique (ex : IDE obésité).

C'est pourquoi nous avons un double niveau de type de ressource. Un concernant les activités (typeressource) et un second plus d'un point de vue logique générale (personnel, materiel).

Ordonnancer : Table de fait de notre système. C'est la table la plus importante. Chaque ligne veut dire : « Pour le patient P qui fait le parcours Pa à la date D, il a besoin de la ressource R pour faire l'activité A de start à end. Cette table contient la planification réalisée de manière manuelle ou automatique (pas implémenté pour le moment).

Evenement : Table identique à la table ordonnancer. Elle a le même but que la table ordonnancer mais cette table contient la planification en cours. Cette table permet de donc de pouvoir sauvegarder ou restaurer la planification en fonction des modifications que l'utilisateur a effectué.



Diagramme de Gantt

Le diagramme de gantt réel est quasiment identique à celui réalisé au début du projet. Cependant, la dernière activité de ce diagramme a été supprimé pour plusieurs raisons :

- Nous préférons fournir un code fonctionnel plutôt que d'en faire un maximum
- Nous pensions au début du projet que les vues de gestion du temps réel avait été développé auparavant ce qui n'était pas le cas.

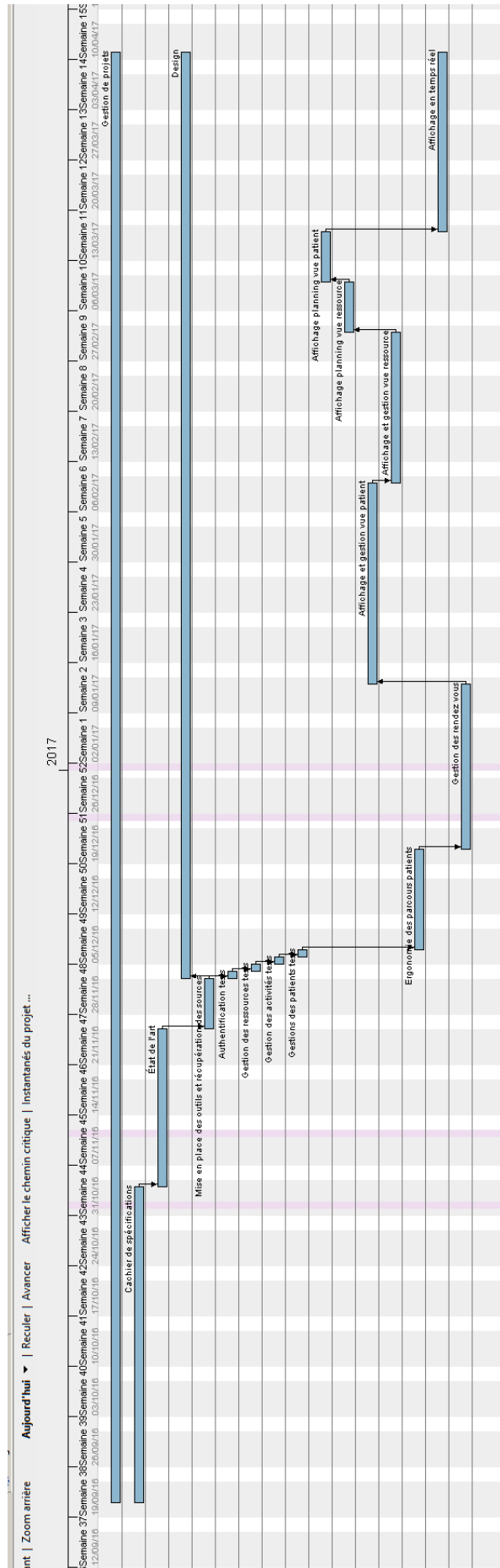


Figure 8 – Diagramme de Gantt prévisionnel

Guide développeur

Structure des sources de l'application

L'application web a été développée en utilisant les technologies suivantes :

- PHP
- HTML, CSS
- JS

Pour le code PHP, nous avons utilisé un Framework PHP (codeIgniter) basé sur le modèle MVC (modèle vue contrôleur). L'utilité de ce Framework est de pouvoir séparer les données des différentes vues permettant l'affichage de ces dernières.

Pour le CSS, nous avons également utilisé bootstrap afin d'avoir un rendu agréable sans avoir des capacités de graphistes.

Voici l'architecture générale de notre application web :

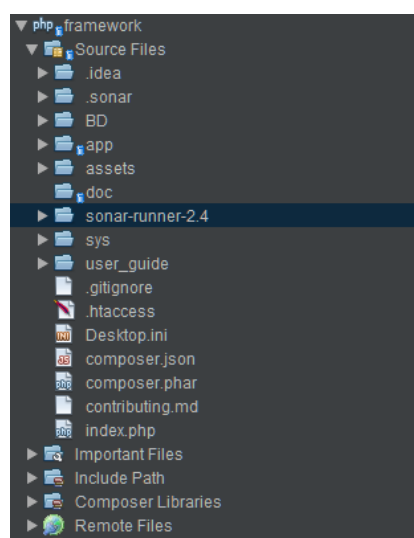


Figure 9 – Structure de notre application

Nous nous plaçons dans le répertoire source files qui est le répertoire qui nous intéresse le plus. Dans ce répertoire, nous trouvons un dossier BD qui contient le script à exécuter permettant

l'ajouter de données tests afin d'utiliser l'application.

Le répertoire « assets » contient tout les fichiers images, javascript, css nécessaire à notre application.

Le répertoire doc contient toute la documentation nécessaire à la compréhension du code.

Les répertoires « sys » et « user-guide » ne nous intéressent pas forcément. Le premier contient les fichiers nécessaires à l'utilisation du Framework codeIgniter.

Le répertoire qui nous intéresse le plus est le répertoire « app » qui contient le code de notre application.

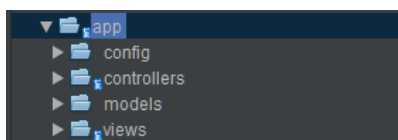


Figure 10 – Contenu du répertoire app

Dans ce répertoire nous trouvons 4 sous répertoires :

- Config (répertoire de configuration du projet comme la base de données etc.)
- Controllers (les contrôleurs de notre application)
- Models (toutes les données de notre application : voir [Figure 7](#) **ERROR**)
- Viws (les vues permettant l'affichage des données)

Contrôleurs

Tous les contrôleurs héritent d'une classe CI Controller (contrôleur de base du Framework codeIgniter). Les contrôleurs permettant de faire le lien entre les données et les vues. Les contrôleurs permettent dans la majeure partie des cas de faire appel à un modèle qui lui-même exécute une requête à la base de données et le contrôleur envoie ces données à la vue qui se charge de les afficher.

Nous allons prendre un exemple pour voir l'utilisation d'un contrôleur.

```

/** \file      Activites ...10 lines */
class Activites extends CI_Controller {

    function __construct() {
        parent::__construct();
        if ($this->session->userdata("username") === null)
            redirect('/Auth', 'refresh');
        if ($this->session->userdata("level") === "1")
            redirect('/AffichageSejour', 'refresh');
    }

    /** \brief      Affichage la liste des différentes activités ...7 lines */
    public function index() {
        $this->load->model('M_Activite');
        $data = array();
        $data['activite'] = $this->M_Activite->getAllActivites();
        $data['chemin'] = '/activite/V_activite';
        $this->load->view('/V_generale', $data);
    }
}

```

Figure 11 – Exemple de contrôleur

Dans l'exemple ci-dessus, nous prenons le cas du contrôleur « Activites » et la méthode « index » permettant d'afficher toutes les activités sur une page.

La méthode charge le modèle nécessaire à la récupération des différentes activités, nous faisons appel à la méthode « getAllActivites » du modèle « M-Activite » permettant de récupérer toute la liste des activités. Ensuite, nous créons un tableau data permettant de récupérer le résultat de la méthode « getAllActivites » ainsi que le chemin de la vue correspondante.

Ensuite, notre contrôleur charge la vue principale (voir dans la partie explication des vues) en lui donnant également le chemin de la vue « V-activite » ainsi que les données provenant du modèle.

Chaque contrôleur et chaque méthode sont basés sur cette structure.

Modèles

Dans cette partie, nous allons voir la structure des différents modèles de notre application. Les modèles sont présents dans notre dossier « models ».

```
public function supprActivite($id) {
    $txt_sql = "DELETE FROM activite
               WHERE id_activite = " . $id;
    $query = $this->db->query($txt_sql);
}
```

Figure 12 – Exemple de modèles

Les modèles exécutent uniquement des requêtes à la base de données.

Vues

Les vues de notre application sont dans le dossier « views ». Chaque vue de notre application dépend d'une vue qu'on appelle vue générale (V-generale).

C'est dans la vue générale que l'on ajoute le menu, le pied de page ainsi que tous les fichiers js et css nécessaires au fonctionnement de notre application.

Voici la structure de notre V-generale :

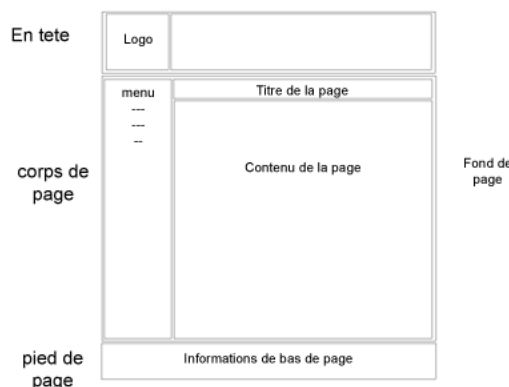


Figure 13 – Vue générale

Dans notre application, chaque vue contient la vue générale, c'est uniquement la partie contenue de la page qui est modifiée.

Voici le code expliquant ce mécanisme :

```
<body>
  <?php
    $this->view('/menu/V_menu');
    $this->view($chemin);
  ?>
</body>
```

Figure 14 – Code changement de vue

La variable « \$chemin » contient le chemin de la vue permettant d'afficher les données en fonction des actions de l'utilisateur. C'est pour cela que nous devons définir une variable chemin dans nos différentes méthodes de nos contrôleurs.

Rapport de tests

Voici la liste des tests effectués sur une journée avec 5 patients et 5 parcours patients (les parcours étant bien planifiés au départ) :

| Tests | Résultats attendus | Résultats obtenus | Etat du test |
|--|--|--|--------------|
| Affichage des données sur le calendrier | Les événements sont affichés au bon endroit | Aucune erreur d'affichage | OK |
| Tests de la liste des contraintes sans modification des événements | Aucune erreur de planification | Aucune erreur de planification dans le panneau d'affichage des contraintes | OK |
| Suppression d'une activité | Affichage d'une erreur sur les contraintes (toutes les activités ne sont pas planifiées) | Toutes les activités ne sont pas planifiées | OK |
| Placer 2 activités sur une même ressource | Erreur (une ressource ne peut pas faire 2 activités à la fois) | Une ressource ne peut pas faire 2 activités à la fois | OK |
| Placer 2 activités d'un même patient en même temps | Erreur (un patient ne peut pas faire 2 activités à la fois) | Un patient ne peut pas faire 2 activités à la fois | OK |
| Placer une activité en dehors de la fenêtre de temps du patient | Erreur (le patient n'est pas disponible) | Le patient n'est pas disponible | OK |
| Ne pas respecter les délais min ou max entre 2 activités | Non respects des délais min et max | L'activité 1 ne respecte pas les délais min ou max | OK |
| Non-respect des précédences dans un parcours | (Erreur) L'activité 1 ne peut pas être avant l'activité 2 | L'activité 1 ne peut pas être avant l'activité 2 | OK |

Figure 15 – Résultats des tests

Analyse de code (Sonar Cube)

De plus, j'ai réalisé une analyse de mon code à l'aide de Sonar Cube. Cette analyse a été défini avec les règles de bases de sonar Cube (variable initialisé mais non utilisé, fonction non utilisé, paramètres dans une fonction inutilisé, portion de code morte, etc.) .Voici le résultat de cette analyse :

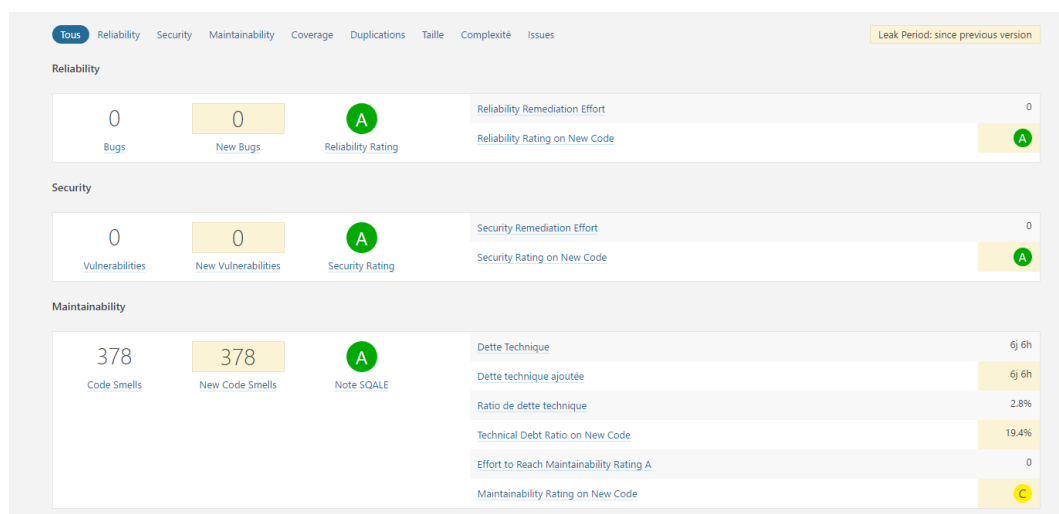


Figure 16 – Analyse de code - Sonar - partie 1

A travers cette analyse, nous pouvons voir que notre code contient 0 bugs (d'écriture), 0 bugs de sécurité. Les 3,6% de duplications sont essentiellement du aux différentes requêtes de l'application (rechercher un patient par son nom, puis après par son prénom, etc). Le point négatif ce projet est la note de C dans la maintenabilité du code et dans le faite qu'il y ai 0% de tests unitaires. Cette note pourrait être amélioré en mettant en constante les valeurs utilisées plusieurs fois dans un même fichier.

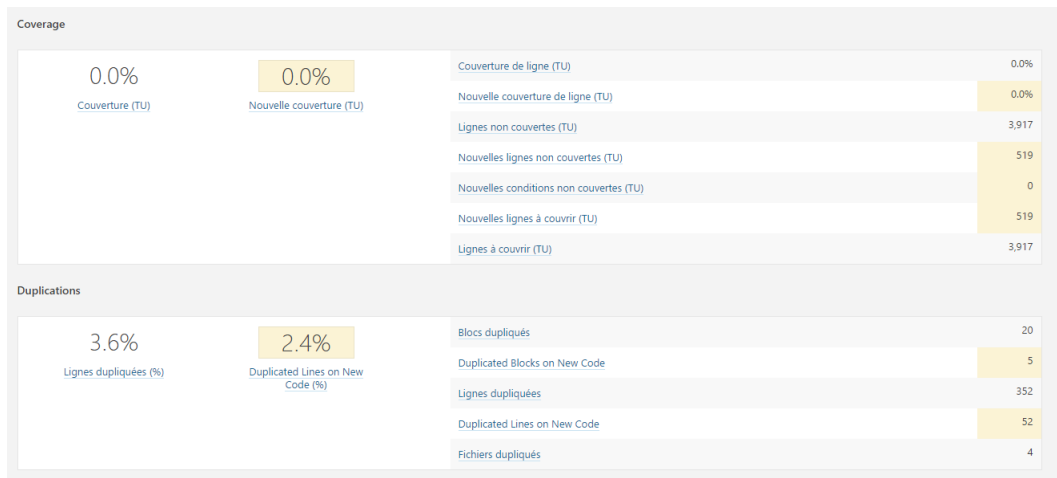


Figure 17 – Analyse de code - Sonar - partie 2

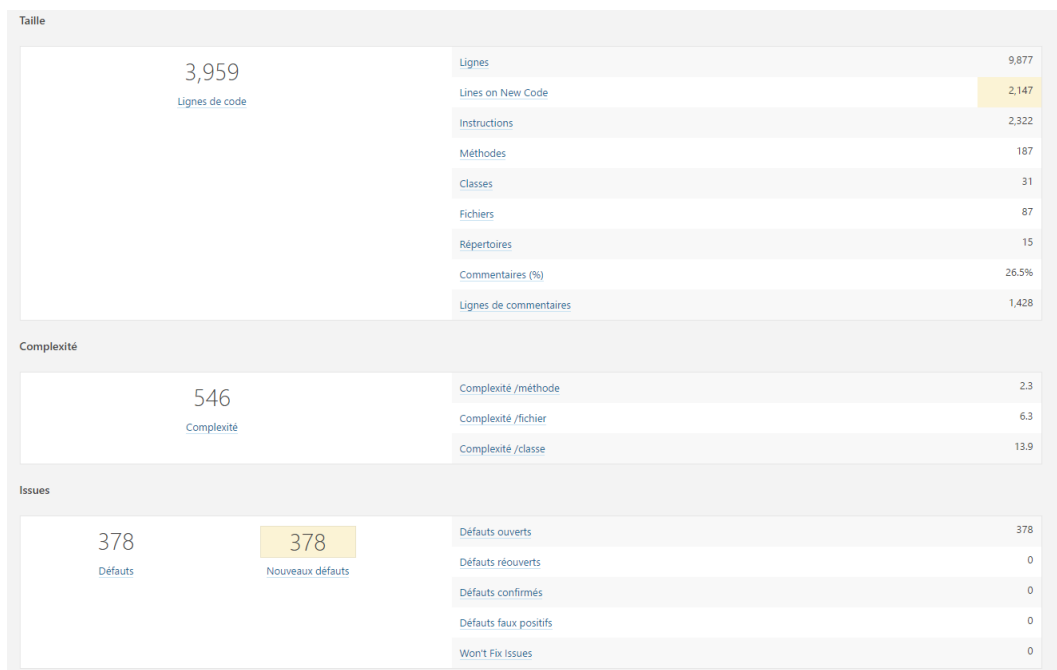


Figure 18 – Analyse de code - Sonar - partie 3



Librairies utilisées

Voici la liste des librairies utilisées ainsi que leur version durant ce projet :

- Php : version 7.0.10
- Bootstrap : version 3.0
- FullCalendar : version 3.0
- MySQL : version 5.7.14



Bibliographie

- [1] <http://www.effigen.com/hopitaux-cliniques/optimisation-des-hopitaux-de-jour/>
- [2] <http://www.vandelaydesign.com/30-best-free-jquery-plugins/>
- [3] <http://www.ticsante.com/lancement-de-Gesplic-premier-logiciel-degestion-et-planification-des-chimiotherapies-NS1394.html>
- [4] <http://www.sih-solutions.fr/gesplic-un-logiciel-de-gestion-et-de-planification-des-chimiotherapies/>
- [5] <http://www.dehosystems.fr/produits-et-solutions/logiciel-gestion-de-planning/>
- [6] <https://fullcalendar.io/>
- [7] <http://www.yriase.fr/1218-fullcalendar-chargement-et-mise-a-jour-des-evenements-dans-une-base-de-donnees.html>
- [8] <https://mattlewis92.github.io/angular-bootstrap-calendar/?example=kitchen-sink>



Comptes rendus hebdomadaires

Compte rendu n°1 du 21/09/2016

- reprise des sources
- lecture du rapport du groupe de projet SI

Compte rendu n°2 du 28/09/2016

- réunion d'explication sur le projet
- objectifs du projet

Compte rendu n°3 du 05/10/2016

- liste des fonctionnalités présentes
- cahier de spécifications
- rapport de PRD
- diagramme de cas d'utilisation et digramme de séquence

Compte rendu n°4 du 12/10/2016

- cahier de spécifications
- état de l'art (fullCalendar)
- diagramme de cas d'utilisation

Compte rendu n°5 du 19/10/2016

- cahier de spécifications
- gestion de projet (ganttt)

Compte rendu n°6 du 10/11/2016

- état de l'art
- cahier de spécifications
- rapport

Compte rendu n°7 du 17/11/2016

- état de l'art
- plan de développement

— rapport

Compte rendu n°8 du 23/11/2016

— état de l'art
— rapport

Compte rendu n°9 du 30/11/2016

— rapport
— tests de l'existant

Compte rendu n°10 du 05/01/2017

— Aucun rapport hebdomadaire n'a été réalisé au cours du second semestre. L'utilisation de trello était plus intéressante.

Outil de gestion de parcours patients dans un hôpital de jour

Guillaume Pochet

Encadrement : Yannick Kergosien

Objectifs et contexte

Le projet est en collaboration avec l'assistance publique - Hôpitaux de Paris.

L'application doit permettre de :

- Visualiser et modifier les parcours patients sous forme de Gantt interactif
- Suivre en temps réel l'avancement du patient dans son parcours

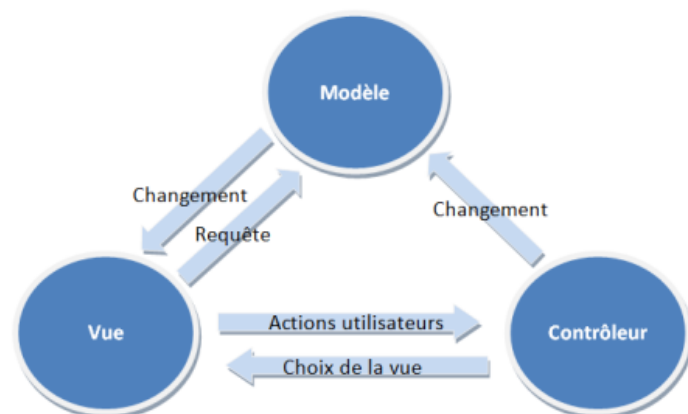


Assistance publique - Hôpitaux de Paris

Mise en œuvre

Pour la réalisation du projet, plusieurs outils sont utilisés :

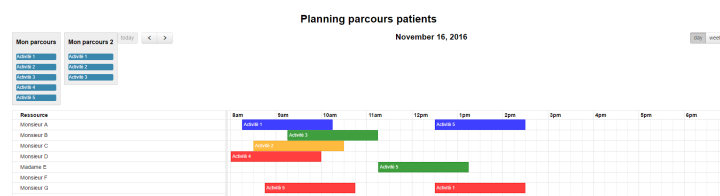
- **FullCalendar**
- Framework php (**codeIgniter**)
- Base de données



Modèle vue contrôleur

Résultats attendus

- Gestion des ressources (matérielles ou humaines)
- Gestion des parcours patients
- **Prise de rendez vous**
- Suivre l'avancement du patient dans son parcours en **temps réel**
- Affichage des différentes plannings



Exemple de planning

Outil de gestion de parcours patients dans un hôpital de jour

Guillaume Pochet

Encadrement : Yannick Kergosien

Objectifs et contexte

Le projet est en collaboration avec l'Assistance publique - Hôpitaux de Paris.

L'application doit permettre de :

- Visualiser et modifier les parcours patients sous forme de Gantt interactif
- Suivre en temps réel l'avancement du patient dans son parcours

Mise en œuvre

Pour la réalisation du projet, plusieurs outils sont utilisés :

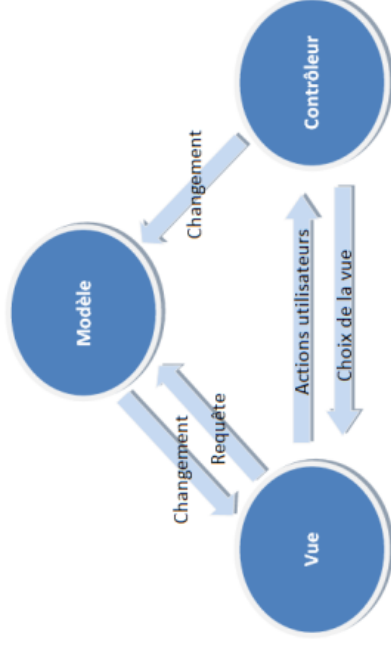
- **FullCalendar**
- Framework php (**codeIgniter**)
- Base de données

Résultats attendus

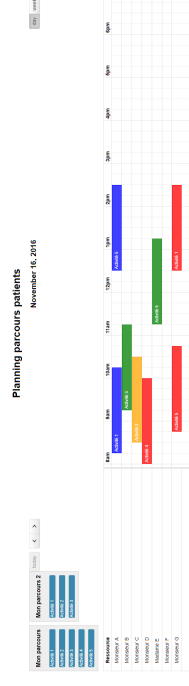
- Gestion des ressources (matérielles ou humaines)
- Gestion des parcours patients
- **Prise de rendez vous**
- Suivre l'avancement du patient dans son parcours en **temps réel**
- Affichage des différents plannings



Assistance publique - Hôpitaux de Paris



Modèle vue contrôleur



Exemple de planning

Outil de gestion de parcours patients dans un hôpital de jour

Résumé

Dans le cadre de mon projet de recherche et développement, je travaille sur le développement d'une application WEB en collaboration avec l'assistance publique - hôpitaux de Paris. Cet outil doit permettre de visualiser et modifier les plannings du personnel de soins dans un hôpital de jour. Ce document a pour but de définir le contexte et les objectifs et aussi de proposer une analyse permettant de répondre aux besoins du client.

Mots-clés

FullCalendar, Web, MVC, planning, Framework PHP, L^AT_EX

Abstract

Keywords

FullCalendar, Web, MVC, Schedule, Framework PHP, L^AT_EX