**Cahier des charges**

**Gestion des échanges d’interventions médicales au sein d’un hôpital**

Carole Belliardo, Hussam Nachabe, Marine Poullet, Marin Truchi

Master 2 sciences de la vie et de la santé

Parcours Biologie, Informatique et mathématiques

U.E. Bases de données / Génie logiciel et UML

Liste de diffusion :

* Collaboration :

Hussam Nachabe <hussamnachabe@gmail.com>;

Marine Poullet <poullet.m@hotmail.fr>;

Marin Truchi <[marint06@laposte.net](mailto:marint06@laposte.net)>;

Carole Belliardo <carole.belliardo@etu.unice.fr>

* Validation :

Jean-Paul Comet <comet@unice.fr>; Gilles Bernot <bernot@unice.fr> ; Ingrid Grenet <grenet@unice.fr>

[I. Présentation du site et spécifications fonctionnelles des besoins 3](#_Toc500831185)

[1. Liste des utilisateurs 3](#_Toc500831186)

[2. Les actions possibles de chaque utilisateur de la plateforme 3](#_Toc500831187)

[3. Spécifications fonctionnelles 3](#_Toc500831188)

[A) Accès niveau 1 4](#_Toc500831189)

[B) Accès niveau 2 4](#_Toc500831190)

[C) Accès administrateur 5](#_Toc500831191)

[II. Conception du site 6](#_Toc500831192)

[1. Les fonctionnalités proposées aux utilisateurs et les interactions utilisateur - système 6](#_Toc500831193)

[A) Diagramme des cas d’utilisations 7](#_Toc500831194)

[B) Description textuelle de chaque cas d’utilisation 7](#_Toc500831195)

[2. Les éléments de base du système : diagramme de classe 26](#_Toc500831196)

[3. Spécifications détaillés 27](#_Toc500831197)

[4. Stockage des informations nécessaires au système 34](#_Toc500831198)

[III. Contrôle qualité 39](#_Toc500831199)

[1. Tests fonctionnels 39](#_Toc500831200)

[2. Tests unitaires 43](#_Toc500831201)

[IV. Retours d’expérience 44](#_Toc500831202)

[V. ANNEXES 45](#_Toc500831203)

# Présentation du site et spécifications fonctionnelles des besoins

## Liste des utilisateurs

|  |  |  |
| --- | --- | --- |
| Utilisateur | Type d’accès | Nombre |
| Administrateur | Accès administrateur | 1 pour l’ensemble de l’hôpital |
| Chefs de service | Restreint niveau 2 | 1 par service |
| Médecins | Restreint niveau 1 | 1 à plusieurs par service |

## Les actions possibles de chaque utilisateur de la plateforme

La plateforme gère les échanges de demandes d’interventions au sein d’un hôpital. Pour réaliser ces échanges, l’utilisateur dispose d’un ensemble d’onglets selon le type d’accès auquel son compte est associé. Chaque type d’accès dispose de droits particuliers caractérisés par un ensemble d’actions spécifiques, ainsi que l’ensemble des droits d’accès relatif au niveau inférieur.

**Figure 1** : Diagramme hiérarchisé des actions possibles selon les différents niveaux d’accès utilisateur

## Spécifications fonctionnelles

Pour accéder à la plateforme, chaque utilisateur doit s’identifier sur une page d’accueil en entrant son identifiant et son mot de passe. Une fois l’utilisateur connecté, il est dirigé vers sa page principale sur laquelle ses Nom, Prénom, Service et Fonctions sont affichés, ainsi qu’un ensemble d’onglets relatifs aux actions proposées par la plateforme en accord avec le niveau d’accès.

### **Accès niveau 1**

Les actions de niveau 1 sont proposées sur la page d’accueil de tous les utilisateurs.

#### Les demandes d’interventions

Une intervention ne peut être demandée que sur un patient pour lequel une fiche « patient » a été enregistrée dans l’hôpital. Il est donc demandé à l’utilisateur de soit :

* Créer une fiche pour un nouveau patient via l’onglet « Patients ». Elle renseigne sur les informations relatives au patient (Nom, Prénom, Sexe, date et lieu de naissance, n° de sécurité sociale, adresse). Dans cet onglet « Patients », une fiche patient peut aussi être modifiée.
* Sélectionner la fiche d’un patient ayant déjà subi une intervention à l’hôpital

Une fois la fiche patient créée/sélectionnée, l'ajout d’intervention devient alors possible. Le médecin doit renseigner le patient qui doit subir l’intervention, sa pathologie diagnostiquée, la nature de l’acte médical, parmi celles réalisées dans l’hôpital, son niveau d’urgence (niveau 0 à 3) et peut ajouter un commentaire. La plateforme attribue un créneau horaire à l'intervention dans le planning du service de l'intervention demandée.

#### Consultation du planning

A partir de sa page principale dans l'onglet « planning », le médecin peut consulter le planning quotidien de son service, en sélectionnant la date du jour qui l’intéresse.

### **Accès niveau 2**

Les actions de niveau 2 sont accessibles à partir de la page principale de tous les utilisateurs sauf les utilisateurs de niveau 1.

#### Modification du planning

##### Gestion des urgences

Lorsqu’une demande d’intervention urgente est soumise (niveau 1, 2 ou 3), celle-ci doit être traitée dans les heures suivant la demande, pour cela 4 possibilités sont alors envisageables.

* Soit, il y a une disponibilité immédiate et le rendez-vous est ajouté au planning selon la procédure classique.
* Soit il y a un créneau disponible dans le délai horaire jugé acceptable pour traiter l’intervention (relatif au niveau d’urgence), et le rendez-vous est alors inséré dans ce créneau.
* Soit, il n’y a pas de créneau disponible dans le délai horaire, et le système procède automatiquement à un réarrangement du planning afin d’insérer l’urgence dans le délai, en décalant les rendez-vous initialement prévus.
* Si ce décalage des créneaux abouti sur un débordement du planning après l’horaire de fermeture du service, une notification apparait sur la session du responsable de service qui doit alors procéder à un réarrangement manuel du planning. Il peut alors réaliser un surbooking, en rajoutant des créneaux de travail supplémentaires au planning de son service.

Lorsque des interventions sont décalées dans le planning, alors leur niveau de priorité augmente, elles doivent ainsi être réalisées dans les heures suivant le rendez-vous initial.

##### Suppression de rendez-vous

En cliquant directement sur le bouton supprimer dans un créneau d’intervention du planning de leur service, les médecins et responsables de service peuvent supprimer un rendez-vous prévu.

#### Facturation des interventions

La facturation des interventions réalisées au sein d’un service sur un patient peut être réalisée par le chef de ce service.

### **Accès administrateur**

L’administrateur peut réaliser l’ensemble des actions associées aux accès restreints précités. Il est aussi garant de la maintenance et du bon fonctionnement de la plateforme. Pour cela, il est le seul à avoir accès au serveur hébergeant la plateforme et peut modifier les scriptes HTML/PHP/MySQL. Il dispose aussi de plusieurs onglets spécifiques relatifs aux tâches d’administration du système.

#### Mise à jour du système

##### Ajout/suppression/modification de services

L’administrateur dispose d’un onglet « Services » qui lui permet de gérer la liste des différents services proposés par l’hôpital. Il peut invalider un service sur la plateforme si celui est fermé au sein de l’hôpital. En cas de fermeture définitive d’un service, l’administrateur peut archiver l’ensemble des informations attachées à ce service, cependant pour des raisons de sécurité et de suivi des dossiers médicaux, les informations ne pourront pas être supprimées.

Si de nouvelles structures d’accueil sont mises en place, l’administrateur peut créer un nouveau service dans la base de données. Lors de l'enregistrement d'un service seules les informations générales sur le service (nom du service et localisation) sont renseignées. On peut supposer que chaque service peut dispenser un nombre variable d'interventions et qu'il est plus pertinent de gérer la liste des actes médicaux de manière indépendante.

L’administrateur peut aussi modifier un service déjà existant pour mettre à jour les informations relatives à celui-ci.

##### Ajout/suppression/modification de compte utilisateur

L’administrateur a la possibilité de suspendre les comptes de tous les utilisateurs. Après suspension, l’administrateur aura aussi la possibilité d’archiver l’ensemble des informations relative au compte. Tous les services devant disposer d’un chef de service, l’administrateur devra d’abord déclarer un nouveau chef de service pour suspendre un compte de niveau 2. L’ancien chef de service devra passer en niveau 1 pour être invalidé.

Lors de l’arrivée d’un nouveau membre dans l’équipe hospitalière, l’administrateur peut créer de nouveaux comptes utilisateurs. Pour cela, il doit saisir un ensemble minimum d’information tel que le nom, le prénom, l’adresse, le service et créer un mot de passe pour ce nouveau compte. A noter que le nouvel utilisateur enregistré dans la plateforme ne peut pas choisir son identifiant. L’identifiant est généré par le système. Un identifiant est défini par la première lettre du nom de famille de l’utilisateur, puis de la première lettre de prénom et suivi de 5 chiffres aléatoires.

Ces informations pourront être modifiées par l’administrateur, ce qui permettra par exemple à un utilisateur de changer son mot de passe, ou de définir un médecin comme nouveau chef de service.

##### Ajout et suppression d’actes médicaux

L’administrateur a la possibilité de tenir à jour la liste des interventions dispensées par un service en ajoutant ou supprimant des actes médicaux.

#### Contrôle de la bonne utilisation du système

##### Notifications incompatibilité urgence-pathologie – contrôle automatique

L’administrateur dispose d’une page « notifications » qui affiche une liste des événements particulier détectés par la plateforme. Cette page facilite par exemple, le contrôle de la cohérence entre le niveau de priorité et la pathologie des demandes d’interventions. La base de données dispose d’une liste des pathologies pouvant être associée à un niveau de priorité élevé. Cette liste peut être modifiée par l’administrateur.

Lorsqu’une demande d’intervention est saisie, si le niveau de priorité attribué est élevé et que la pathologie ne figure pas dans la liste des pathologies susceptibles d’être prioritaires une notification apparait. Si un patient à plusieurs rendez-vous urgents demandés par le même médecin à plusieurs reprises, alors une notification apparaitra avec les noms du médecin et du patient.

##### Statistiques demandes interventions – contrôle manuel

Pour affiner le contrôle de la cohérence et la légitimité des demandes d’interventions prioritaires, l’administrateur dispose d’une page synthétisant l’ensemble des informations concernant ces demandes. Cette page affichera les informations de tous les services, comme le nombre d’interventions total demandées les 12 derniers mois, le nombre d’interventions prioritaires, la proportion d’interventions annulées, ou le nombre d’incompatibilités détectées. Ces informations pourront permettre sur le long terme de gérer le surbooking en réservant des rendez-vous à la même heure pour 2 patients, en pariant sur l’annulation de l’un des deux.

# Conception du site

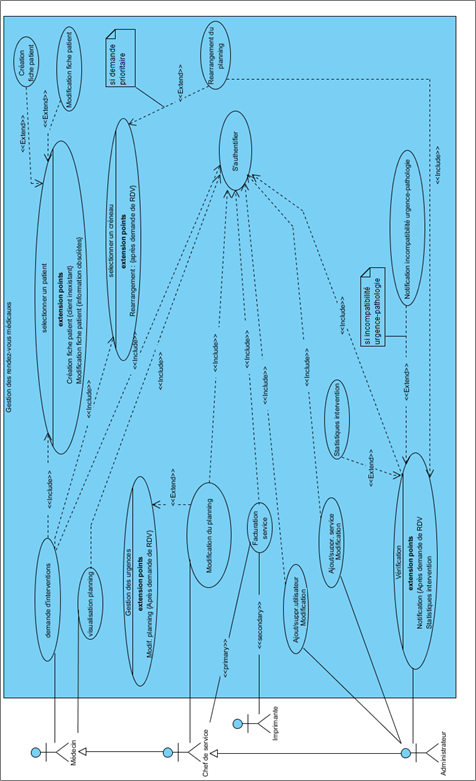
## Les fonctionnalités proposées aux utilisateurs et les interactions utilisateur - système

Les étapes précédentes nous ont permis d’identifier les principaux packages nécessaires :

* Maintenance (Gestion des Comptes Utilisateur, Gestion des Services)
* Vérification (cohérence Pathologie-Urgence, statistique demande rendez-vous)
* Gestion du planning (Edition, modification, visualisation, gestion des urgences)

On va maintenant essayer de définir de manière plus détaillée les besoins de chaque utilisateur en déterminant qui devra pouvoir faire quoi et comment le système répondra à ces attentes. Dans un premier temps, on va illustrer comment les différents acteurs interagissent avec ces packages en réalisant un diagramme des cas d’utilisation.

### Diagramme des cas d’utilisations



**Figure 2**: Diagramme cas d’utilisation

### Description textuelle de chaque cas d’utilisation

### Cas s’authentifier

**Nom** : S'authentifier

**Acteur(s)** : Admin, Chef de service, Médecin

**Description** : L'authentification est le processus permettant au système de s'assurer de la légitimité d’un utilisateur à demander l'accès aux ressources du système

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit posséder un identifiant et un mot de passe valide.

**Démarrage** : L’utilisateur est sur la page « accueil »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur renseigne les champs du formulaire d’authentification et valide.
2. Le système appel la fonction « Authentification ».
3. Le système fait appel à la fonction « Afficher le la page principale ».
4. Le système fait appel au cas d’utilisation interne « Planning ».
5. Le système fait appel au cas d’utilisation interne « Demande d’intervention ».
6. Le système fait appel au cas d’utilisation interne « Patient ».
7. Le système fait appel au cas d’utilisation interne « Facturation ».
8. Le système fait appel au cas d’utilisation interne « Service ».
9. Le système fait appel au cas d’utilisation interne « Compte Utilisateur ».
10. Le système fait appel au cas d’utilisation interne « Vérification ».
11. Le système fait appel au cas d’utilisation interne « Déconnexion ».

**Les scénarios alternatifs**

7 - 10.a Le système ne reconnaît pas l’utilisateur comme un chef de service ou admin. L’utilisateur est un médecin, il n’a pas accès à ces cas d’utilisation interne.

8 - 10.a Le système reconnaît l’utilisateur comme un chef de service. L’utilisateur n’est pas reconnu comme administrateur, il ne peut donc pas avoir accès aux cas « Service », « Compte Utilisateur » et « Vérification », mais a accès au cas d’utilisation interne « Notifications ».

**Scénario d’exception** :

3.a Le système ne reconnaît pas l’identifiant et/ou le mot de passe pour l’utilisateur. Il affiche « Identifiant ou Mot de passe incorrect »

**Fin** : Scénario nominal : réalisation des étapes 4, 5, 6, 7, 8, 9, 10 ou 11 selon les privilèges de l'utilisateur.

**Post-condition(s)** : La page principale qui offre l'ensemble des fonctionnalités selon les privilèges de l'utilisateur.

Cas Créer Fiche Patient

**Nom** : Créer Fiche Patient

**Acteur(s)** : Admin, Chef de service, Médecin

**Description** : La fiche patient permet d’enregistrer et consulter des informations relatives aux patients. Un utilisateur souhaite créer une fiche pour un nouveau patient.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié.

**Démarrage** : L’utilisateur est sur la page « principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur se place sur l’onglet « Patient ».
2. L’utilisateur clique sur le bouton « création ».
3. Le système affiche un formulaire de création fiche patient vide.
4. L’utilisateur renseigne les informations relatives au patient.
5. L’utilisateur valide la saisie en cliquant sur le bouton « valider ».
6. Le système enregistre les informations saisies dans la base de données.
7. Le système affiche une confirmation de la création de la fiche.

**Les scénarios alternatifs**

2.a L'utilisateur décide d'interrompre la procédure

3.a L'utilisateur décide d'interrompre la procédure

5.a L'utilisateur décide d'interrompre la procédure

8. L'utilisateur souhaite fixer un rendez-vous pour le patient sélectionné. Il clique sur le bouton « demande de rendez-vous ». Le système réalise les étapes 7 et 8 puis fait appel au cas d’utilisation « demande de rendez-vous ».

**Scénario d’exception** :

6.a Un champ est vide, alors l’utilisateur doit compléter ce champ avant de pouvoir valider.

6.b Le patient est déjà existant dans la base de données, une notification s'affiche

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-condition(s)** :  La fiche patient a été enregistrée dans la base de données.

Cas Modifier Fiche Patient

**Nom** : Modifier Fiche Patient

**Acteur(s)** : Admin, Chef de service, Médecin

**Description** : La fiche patient permet d’enregistrer et consulter des informations relatives aux patients. Un utilisateur souhaite modifier les informations relatives au patient.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié.

**Démarrage** : L’utilisateur est sur la page « principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur sélectionne l’onglet « Patient ».
2. L’utilisateur clique sur le bouton « Modification ».
3. Le système affiche la page « Modification fiche patient ».
4. L’utilisateur sélectionne un patient et valide.
5. Le système affiche un formulaire comportant comme valeurs par défauts les informations existantes relatives au patient sélectionné.
6. L’utilisateur modifie un ou plusieurs champs du formulaire
7. L’utilisateur valide la saisie en cliquant sur le bouton « valider ».
8. Le système enregistre les informations saisies dans la base de données.
9. Le système affiche une confirmation de la modification de la fiche.

**Les scénarios alternatifs**

4.a L'utilisateur décide d'interrompre la procédure.

5.a L'utilisateur décide d'interrompre la procédure.

7.a L'utilisateur décide d'interrompre la procédure.

10.a L'utilisateur souhaite fixer un rendez-vous pour le patient sélectionné. Il clique sur le bouton « demande de rendez-vous ». Le système réalise les étapes 9 et 10 puis fait appel au cas d’utilisation « demande de rendez-vous ».

**Scénario d’exception** :

5.b L’utilisateur a rentré un mauvais numéro de patient, le système le lui signale.

8.b L’utilisateur a rentré de mauvaises informations ou a laissé des champs obligatoires vides. Le système signale les erreurs commises.

**Fin** : Scénario nominal : validation sur décision de l’utilisateur

**Post-condition(s)** :  Les modifications de la fiche ont été enregistrées dans la base de données

Cas demande d’intervention

**Nom** : demande d’intervention

**Acteur(s)** : Admin, Chef de service, Médecin

**Description** : La demande d’intervention permet à l’utilisateur de demander une intervention pour un patient.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur sélectionne l’onglet « demande de rendez-vous »
2. L’utilisateur sélectionne un patient et valide.
3. Le système affiche un formulaire
4. L’utilisateur informe les champs relatifs au patient et à l’acte médical.
5. L’utilisateur valide la demande d’intervention en cliquant sur le bouton « valider »
6. Le système recherche le créneau disponible le plus proche et enregistre l’intervention dans le planning
7. Le système affiche une confirmation de rendez-vous et le créneau attribué

**Les scénarios alternatifs**

2.a L'utilisateur décide d'interrompre la procédure.

4.a L'utilisateur décide d'interrompre la procédure.

5.a L'utilisateur décide d'interrompre la procédure.

6.a L’utilisateur saisie un niveau de priorité élevé, le système fait appel au cas d’utilisation interne « gestion des priorités »

**Scénario d’exception** :

1.a L’utilisateur a initié la demande depuis une fiche patient, le formulaire va alors associer dans les champs correspondant les informations relatives au patient

2.b L'utilisateur a rentré un mauvais identifiant de patient, le système lui signale

6.a L’utilisateur a oublié de remplir certains champs ou a entré des valeurs interdites, il est informé de ses erreurs par le système.

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-condition(s)** : une intervention est planifiée pour le patient, l’acte et le créneau horaire spécifiés

Cas gestion des priorités

**Nom** : gestion des priorités

**Acteur(s)** : Admin, Chef de service, Médecin

**Description** : Sélection d'un créneau adapté pour fixer un rendez-vous d'intervention prioritaire.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié. Une demande de rendez-vous est soumise au système et celui-ci identifié comme urgent

**Démarrage** : L’utilisateur a soumis une demande de rendez-vous avec un niveau de priorité > 0.

**DESCRIPTION**

Le scénario nominal

1. Le système confirme la compatibilité entre la pathologie et son niveau d'urgence.

2. Le système identifie la présence de créneau disponible dans un délais de temps spécifique au niveau d’urgence (niveau 1 : 24 heures / niveau 2 : 6 heures / niveau 3 : 3 heures).

3. Le système enregistre le rendez-vous dans le planning

4. Le système signale à l’utilisateur que la demande de d’intervention a été validée et fournit les informations sur le créneau attribué.

**Les scénarios alternatifs**

1.a Le niveau d'urgence n'est pas compatible avec la pathologie. Une notification est envoyée à l'administrateur

1 .b La pathologie n’est pas connu du système, elle est donc enregistrée comme nouvelle pathologie sans a priori de niveau d’urgence maximale.

2.a Il n'y a pas de créneaux horaires disponibles dans le délai imparti. Le système fait appel au cas d’utilisation « notifications », afin de suggérer un surbooking au chef de service qui devra valider la demande de surbooking. Le rendez-vous est placé dans le planning avant la fin du délai. Si l’urgence est placée en même temps qu’un autre rendez-vous celui-ci et les suivants sont décalés sur le créneau suivant. Quand un rendez-vous est décalé, la valeur de son niveau d’urgence est incrémentée systématiquement.

2.b Le Chef de service modifie le planning manuellement, le système fait appel au cas d’utilisation interne « Modification du planning »

**Scénario d’exception** :

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-condition(s)** : une intervention prioritaire est planifiée pour le patient selon un créneaux horaire approprié. Si nécessaire, le planning de travail du service est modifié pour permettre le surbooking.

Cas Facturation

**Nom** : Facturation

**Acteur(s)** : Admin et Chef de service

**Description** : Il est possible d’imprimer une facture pour l’ensemble des interventions réalisées pour un patient.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-conditions** : L’utilisateur doit être identifié. L’utilisateur est le chef du service dans lequel l’intervention a été effectuée.

**Démarrage** : L’utilisateur est sur la page « Facturation »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur sélectionne un patient et valide.
2. Le système affiche la facture correspondant au patient sélectionné, avec l’ensemble de ces interventions réalisées au sein du service.
3. Le système lance la fonction « édition de facture » et affiche la facture au format pdf.
4. L’utilisateur clique sur « imprimer ».
5. Le système lance la fonction « impression ».

**Les scénarios alternatifs** : 1.a L'utilisateur décide d'interrompre la procédure

**Scénario d’exception** :

1.b L’utilisateur entre un patient non présent dans la base de données

4.a L'utilisateur décide d'interrompre la procédure

5.a Le patient n’existe pas dans la base de données, une notification apparait pour informer l’utilisateur

6.a L'utilisateur décide d'interrompre la procédure

7.a Le patient n’a subi aucune intervention, la page est vide

8.a L'utilisateur décide d'interrompre la procédure

9.a L’intervention n’a pas encore été effectué, son statut est « en cours » ou « annulé », le système affiche une notification

10.a L'utilisateur décide d'interrompre la procédure

11.a Aucune imprimante n’est détectée, le système affiche une notification

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :  La facture est imprimée par le périphérique

Cas Modification du planning

**Nom** : Modification du planning

**Acteur(s)** : Admin et Chef de service

**Description** : Il est possible de modifier manuellement le créneau horaire d’un rendez-vous par l’utilisateur.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié. L’utilisateur est un chef de service ou l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Planning »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur sélectionne l’une des interventions affichées.
2. Le système affiche un formulaire comportant comme valeur par défaut les informations portant sur le rendez-vous.
3. L’utilisateur modifie la valeur de l’un des champs du formulaire.
4. Le système affiche une confirmation de modification des informations.

**Les scénarios alternatifs**

**Scénario d’exception** :

1.a L'utilisateur décide d'interrompre la procédure

2.a Le créneau est libre

3.a L'utilisateur décide d'interrompre la procédure

4.a L’un des champs obligatoires est vide

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-condition(s)** :

Cas Ajout d’utilisateur

**Nom** : Ajout d’utilisateur

**Acteur(s)** : Admin

**Description** : Il est possible d’ajouter des utilisateurs

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur se place sur l’onglet « Compte Utilisateur »
2. L’utilisateur clique sur le bouton « Création »
3. Le système affiche un formulaire vide comportant l’ensemble des champs relatif aux attributs des utilisateurs
4. L’utilisateur saisie les informations et valide
5. Le système affiche une confirmation que l’action a été exécutée.

**Les scénarios alternatifs**

3.a L'utilisateur décide d'interrompre la procédure

6.a Le « compte utilisateur » existe déjà, une notification apparait

**Scénario d’exception** : 5.a L’utilisateur a oublié de remplir certains champs ou a rempli de mauvaises informations, un message d’erreur est affiché.

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :  Cas suppression des utilisateurs

**Nom** : Suppression des utilisateurs

**Acteur(s)** : Admin

**Description** : Il est possible de supprimer des utilisateurs par exemple en cas de licenciement d’un employé

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur se place sur l’onglet « Compte Utilisateur »
2. L’utilisateur clique sur le bouton « Supprimer »
3. Le système affiche une liste des utilisateurs existant
4. L’utilisateur sélectionne l’utilisateur souhaité et valide
5. Le système affiche une confirmation que l’action a été exécutée.

**Les scénarios alternatifs**

3.a L'utilisateur décide d'interrompre la procédure

4.a L'utilisateur décide d'interrompre la procédure

**Scénario d’exception** :  4.b L'utilisateur entre un identifiant non valide, un message d’erreur est affiché.

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas modification des services

**Nom** : Modification des services

**Acteur(s)** : Admin

**Description** : Un service de l’hôpital peut être modifié.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur place le pointeur de la souris sur l’onglet « Services »
2. L’utilisateur clique sur le bouton « Modification »
3. Le système propose une liste des services existant
4. L’utilisateur sélectionne le service à modifier et valide
5. Le système affiche un formulaire comportant comme valeurs par défaut les informations existantes relatives au service sélectionné.
6. L’utilisateur modifie un ou plusieurs champs du formulaire
7. L’utilisateur valide la saisie en cliquant sur le bouton « valider ».
8. Le système enregistre les informations saisies dans la base de données.
9. Le système affiche une confirmation de la modification de la fiche.

**Les scénarios alternatifs**

3.a L'utilisateur décide d'interrompre la procédure

4.a L'utilisateur décide d'interrompre la procédure

6.a L'utilisateur décide d'interrompre la procédure

7.a L'utilisateur décide d'interrompre la procédure

8.a Si des champs ne sont pas remplis, le système demande de remplir les champs manquants

**Scénario d’exception** :

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas modification des utilisateurs

**Nom** : Modification des utilisateurs

**Acteur(s)** : Admin

**Description** : Les informations d’un employé de l’hôpital peuvent être modifiées

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur place le pointeur de la souris sur l’onglet « Compte Utilisateur »
2. L’utilisateur clique sur le bouton « Modification »
3. Le système propose une liste des d’identifiants correspondant aux employés de l’hôpital
4. L’utilisateur sélectionne le compte utilisateur à modifier et valide
5. Le système affiche un formulaire comportant comme valeurs par défaut les informations existantes relatives au compte sélectionné.
6. L’utilisateur modifie un ou plusieurs champs du formulaire
7. L’utilisateur valide la saisie en cliquant sur le bouton « valider ».
8. Le système enregistre les informations saisies dans la base de données.
9. Le système affiche une confirmation de la modification de la fiche.

**Les scénarios alternatifs**

3.a L'utilisateur décide d'interrompre la procédure

4.a L'utilisateur décide d'interrompre la procédure

6.a L'utilisateur décide d'interrompre la procédure

7.a L'utilisateur décide d'interrompre la procédure

8.a Si des champs ne sont pas remplis, le système demande de remplir les champs manquants

**Scénario d’exception** :

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas Ajout Service

**Nom** : Ajout de service

**Acteur(s)** : Admin

**Description** : Il est possible d’ajouter des services lorsqu’une nouvelle infrastructure est mise en place dans l’hôpital

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur clique sur l’onglet « Maintenance »
2. Le système affiche la page de maintenance
3. L’utilisateur clique sur le bouton « ajouter un service »
4. Le système affiche un formulaire vide comportant l’ensemble des champs relatif aux attributs des services
5. L’utilisateur saisie les informations et valide
6. Le système affiche une confirmation que l’action a été exécutée.

**Les scénarios alternatifs**

1.a L'utilisateur décide d'interrompre la procédure

3.a L'utilisateur décide d'interrompre la procédure

5.a L'utilisateur décide d'interrompre la procédure

6.a Le « service » existe déjà, une notification apparait

**Scénario d’exception** :

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas ajout acte

**Nom** : Ajout d’un acte médical

**Acteur(s)** : Admin

**Description** : Il est possible d’ajouter de nouveaux actes dispensés par les services de l’hôpital

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 29/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur se place sur l’onglet « Services »
2. L’utilisateur clique sur le bouton « ajouter un acte »
3. Le système propose la liste des services de l’hôpital
4. L’administrateur sélectionne le service pour lequel un nouvel acte doit être ajouté et valide
5. Le système affiche un formulaire de 2 champs, « nom de l’acte » et « tarif »
6. L’utilisateur rempli ces 2 champs et valide
7. Le système enregistre ce nouvel acte pour le service sélectionné et le signale à l’utilisateur.

**Les scénarios alternatifs**

4.a L'utilisateur décide d'interrompre la procédure

6.a L'utilisateur décide d'interrompre la procédure

**Scénario d’exception** :  4.b L’utilisateur entre un service inexistant, un message signale l’erreur

6.b L’utilisateur entre des valeurs invalides ou laisse des champs vides, le système lui demande de corriger son erreur

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas Archivage d’un acte

**Nom** : Archivage d’un acte médical

**Acteur(s)** : Admin

**Description** : Il est possible d’archiver les actes médicaux qui ne sont plus dispensés dans l’hôpital

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 29/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur se place sur l’onglet « Services »
2. L’utilisateur clique sur le bouton « Archiver un acte »
3. Le système affiche une liste des actes existant
4. L’utilisateur sélectionne l’acte à supprimer et valide
5. Le système affiche une confirmation que l’action a été exécutée.

**Les scénarios alternatifs**

4a. L'utilisateur décide d'interrompre la procédure

**Scénario d’exception** :

5a. L’acte que l’utilisateur a écrit n’existe pas, ou le champ est toujours vide, un message d’erreur le signalant apparaît.

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Archivage des services

**Nom** : Archivage des services

**Acteur(s)** : Admin

**Description** : Il est possible d’archiver des services par exemple lorsqu’un service est fermé

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 28/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié, L’utilisateur doit être l’administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur place le pointeur de la souris sur l’onglet « Services »
2. L’utilisateur clique sur « Archivage »
3. Le système propose une liste des services existant
4. L’utilisateur sélectionne le service à archiver et valide
5. Le système affiche une confirmation que l’action a été exécutée.

**Les scénarios alternatifs**

4.a L'utilisateur décide d'interrompre la procédure

**Scénario d’exception** :  4.b L’utilisateur ne sélectionne pas de services ou entre un service inexistant, cette erreur lui est signalée par le système

**Fin** : Scénario nominal : validation sur décision de l’utilisateur.

**Post-conditions** :

Cas statistiques des demandes

**Nom** : statistiques des demandes

**Acteur(s)** : Admin

**Description** : Il est possible de vérifier de manière synthétique des demandes effectuer pour mettre en évidence des anomalies

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié comme administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal

1. L’utilisateur clique sur « Synthèse des demandes » dans l’onglet « Vérification ».
2. Le système affiche le tableau récapitulant les statistiques des interventions pour chacun des services

**Les scénarios alternatifs** :

**Scénario d’exception** :

**Fin** :

**Post-condition(s)** :

Cas Vérification des compatibilités d’urgences

**Nom** : Vérification des compatibilités demandes d’urgences

**Acteur(s)** : Admin

**Description** : Il est possible de vérifier la légitimité des demandes d’urgences par l’administrateur.

**Auteur** : Belliardo, Poullet, Truchi, Nachabe

**Date(s)** : 07/11/2017 (première rédaction)

**Pré-condition(s)** : L’utilisateur doit être identifié comme administrateur.

**Démarrage** : L’utilisateur est sur la page « Principale »

**DESCRIPTION**

Le scénario nominal :

1. L’utilisateur clique sur « Notifications » dans l’onglet « Vérification ».
2. Le système affiche la page « Notifications » qui listent les interventions demandées pour lesquels il y a une incompatibilité entre la pathologie et le niveau d’urgence, ou si la pathologie entrée n’est pas connue par le système
3. L’utilisateur peut accepter la notification ou la refuser.
4. La liste des notifications se met à jour.

**Les scénarios alternatifs**

2.a Aucune notification apparait si aucune incompatibilité d'urgence pathologique n'est à signaler.

3.a L’utilisateur peut ignorer les notifications.

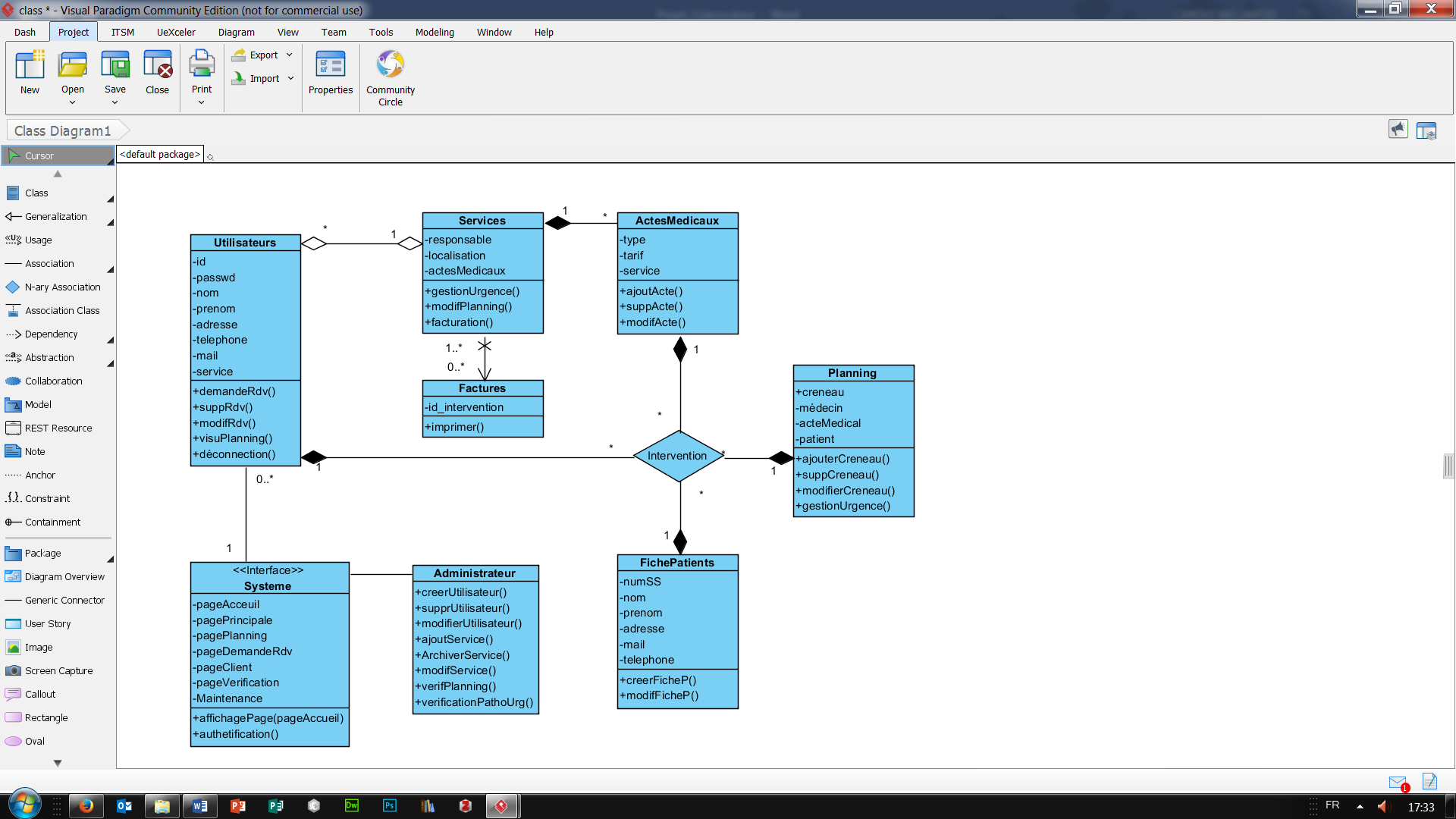
**Scénario d’exception** :

**Fin** :

**Post-condition(s)** :

## Les éléments de base du système : diagramme de classe

Après avoir vu le système du point de vue des auteurs, on va maintenant s’intéresser au système d’un point de vu interne. On va voir une représentation abstraite des objets qui le compose et vont interagir pour la réalisation des cas d’utilisation.



**Figure 3** : Diagramme de classes

Les classes sont constituées par un ensemble d’objets qui partagent des caractéristiques générales, les attributs. Les attributs permettent de distinguer précisément les objets qui sont les différentes instances de la classe. Par exemple, l’attribut « acte » de la classe « Planning » fait référence à un service. Ainsi, cet attribut permettra entre les différents rendez-vous de distinguer ceux qui appartiennent au planning de l’un des services.

Ces classes possèdent aussi des méthodes qui sont des fonctions permettant de modifier les attributs. Les méthodes peuvent modifier les attributs de la classe auxquels elles appartiennent mais aussi les attributs des classes auquel elles sont associées, la classe « utilisateur » peut modifier les attributs de la classe « planning »

Cette étape nous permet de mettre en évidence l’ensemble minimum de fonction à implémenter pour structurer le système et garantir qu’il dispose des fonctionnalités attendues par le client.

## Spécifications détaillés

1. Hypothèses :

Chaque service propose un ensemble d'actes médicaux, chaque acte médical dispose d'un créneau toutes les 15 minutes entre l'heure d'ouverture et l'heure de fermeture.

Chaque acte n'est pas rattaché à un médecin en particulier. Le personnel médical du service se répartit donc les interventions pour s'assurer que ces tâches soient réalisées en continue et en parallèle. Ainsi, pour chaque service un planning est affiché avec l'ensemble des actes médicaux qu’il propose. Une liste des créneaux successives est donc affiché pour chaque acte de ce service.

Un rendez-vous n'est pas pris avec un médecin mais avec un service et plus précisément pour une intervention.

Tous les services travaillent 7/7 jours, il n'y a pas de week-end

Il ne peut pas y avoir deux services portant le même nom

- Chaque acte a un tarif unique. Aucun prix spécifique est attribué aux urgences.

- L'administrateur est un informaticien qui ne prends pas parti aux transactions de rendez-vous au sein de l'hôpital. Cependant, il a accès aux fonctionnalités des autres utilisateurs pour faciliter l'évaluation de la plateforme et le débuggage. Les services de l'hôpital peuvent disposer de differentes localisations (batiment, étage, aile).

- Il n'y a qu'un administrateur, du moins qu'un compte utilisateur d'administration de la plateforme.

- Chaque intervention a des niveaux d'urgences (minimum et maximum) de référence relatifs à la pathologie associée.

- Les niveaux d'urgence minimums sont préétablis dans la base de données.

- Pour tous services, les rendez-vous durent 15 minutes.

2. L'organisation du code :

Le script étant assez long, il a été décomposé en plusieurs fichiers. Ces nombreux fichiers ont été répartis dans plusieurs répertoires organisés selon leurs types et fonctions. De plus, cette architecture du code, nous a permis de factoriser certaines parties du code en individualisant les blocs d'instructions réutilisés sous forme de fonctions.

Dans “htdocs” se trouve :

Un fichier “404.php” et un fichier “.htaccess”. Ils permettent la gestion d’erreur de type 404 renvoyé par le serveur HTTP pour indiquer que la ressource demandée ou une page web n’existe pas. Lorsqu’un utilisateur tente d'accéder à une ressource inexistante, il est automatique renvoyer vers la page d’accueil du site.

Dans chaque sous dossier de “htdocs” se trouve un fichier “index.php”. Celui ci permet la redirection de l’utilisateur vers l’accueil dans le cas où l’utilisateur souhaiterait voir le contenu d’un dossier.

Ainsi, à la racine se trouve :

- le fichier “Accueil.php” correspondant à la page d'accueil du site qui permet de se connecter et d'accéder à l’ensemble des services proposés par notre site web.

- le fichier “dbconfig.php” qui contient la classe “Basededonnee”. Cette classe permet de récupérer de façon privé l’ensemble des informations permettant l'établissement de la connection avec la base de donnée. La localisation, le nom, le nom d’utilisateur et le mot de passe sont déclaré dans cette classe. La fonction “bddConnection()” permet de réaliser la connection à la base de donnée. De plus, les erreurs de connection sont récupérées et peuvent être affichées.

- Le fichier “Classe.Systeme.php” qui contient la classe Systeme. Cette classe qui compose notre site web possède 8 fonctions essentiels à la bonne exécution de l’ensemble des services proposés par le site. Elle permet d'ouvrir une connexion au serveur MySQL et possède plusieurs fonctions.

Une fonction “\_\_construct() ” de construction de la classe Systeme est appelé à chaque création d’une nouvelle instance de l’objet. Cette fonction est nécessaire pour toutes les initialisations dont l'objet a besoin avant d'être utilisé. De cet façon, à chaque utilisation des fonctions associés à cette classe, la fonction “\_\_construct() ” va établir la bonne connection à la base de donnée en appelant la classe “Basededonnee”.

La fonction “runQuery()” qui permet de préparer le requetes SQL tout en s’assurant la bonne connection à la base de donnée.

La fonction “creerUtilisateur()” permet de récupérer en entrée le nom d’utilisateur et son mot de passe. Le mot de passe sera alors crypté grâce à une clé de qui utilise un hachage fort et irréversible. Seul le résultat est stocké dans la base de données.

La fonction “authentification()” permet de s’authentifier au site. Elle vérifie que le mot de passe correspond à l’utilisateur qui souhaite s’identifier.

La fonction “estConnecté()” permet de tester si l’utilisateur est bien connecté.

La fonction “redirect” permet de rediriger vers une autre page qui peut être donnée en argument.

La fonction “viderSession()” qui permet de supprimer des variables associé à la session en cours. Principalement utilisé lorsque l’utilisateur souhaite abandonner une action entreprise.

La fonction “seDeconnecter()” qui permet à l’utilisateur de se déconnecter en toute sécurité

-Le fichier “session.php” permet d’initialiser la session pour l’utilisateur qui vient de se connecter.

- Le fichier “deconnection.php” qui permet de déconnecter un utilisateur de sa session et de le rediriger vers la page d’accueil.

- PagePrincipale est la page sur laquelle l'utilisateur est redirigé une fois qu’il est connecté ou à chaque fois qu’il clique sur le bouton abandonner affiché en fin de formulaire. Elle propose l'accès aux différentes fonctionnalités selon les droits d'accès de l'utilisateur.

On trouve aussi 6 répertoires :

- BDD : Le fichier de structuration de la base de donnée, ainsi qu'un fichier comportant quelques insertions pour permettre les tests

- Images : l’ensemble des images libres de droits et logo réalisé pour les aspects visuels

- Config :

> Style.css qui structure l'ensemble des pages du site,

> MenuPage.php qui correspond au menu de navigation de l'ensemble des pages. Après validation de l'authentification, cette page permet aussi de gérer les droits d'accès des utilisateurs aux pages en disposant d'un affichage différentielle en fonction de la qualité de l'utilisateur (pour gérer le cas ou l'utilisateur saisit l'adresse dans la barre pour une page à laquelle il n'a pas de droits d'accès, la restriction d'accès est aussi renforcée sur chaque page par un test conditionnel évaluant le niveau de restriction de l'utilisateur lors de l'arrivée sur la page, et une redirection si nécessaire ),

> Footer : le pied de page permettant d'accéder aux contacts des développeurs et aux conditions d'utilisation.

- Formulaires : comporte l'ensemble des formulaires nécessaires au fonctionnement du site dont certains sont appelés par plusieurs pages telles que la recherche de patients, de services ou d'actes médicaux.

- Page : comporte 1 fichier pour chaque cas d'utilisation. Chacun de ces fichiers gèrent les redirections nécéssaires, renforcements de sécurité selon droits d'utilisation ou encore redirections vers la sélection d'un patient pour pouvoir demander un rendez-vous ou sélectionner un service pour une suppression.

- Fonction : comporte 10 modules différents regroupant chacun un ensemble de fonctions organisé au sein de ses fichiers de manière thématique.

3. Détails implémentation des cas d'utilisation

a. Généralités sur les formulaires

Les formulaires permettent à l'utilisateur de dialoguer avec la base de données, et ainsi d'insérer ou d'identifier et d'afficher des informations. Pour l'ensemble des formulaires établis, un bouton de validation permet de récupérer les valeurs saisies dans la variable '$\_POST'. Différents traitements sont imposés à chaque éléments de ce tableau pour faciliter la gestion d'erreur et limiter la redondance des données. Par exemple, la suppression des espaces ou encore le formatage des chaînes de caractères en minuscule avec la première lettre en majuscule. Si le format accepté par la base de données ou si l'un des champs qui doit être renseigné est vide (champ 'not null' dans la base de données), la gestion d'erreurs prévoit l'interruption de la procédure et l'envoi d'un message à l'utilisateur pour l'informer du type d'erreurs détectées. Pour les entrées dans des tables qui comportent des clés étrangères, soit la clé est récupérée dans la table parent, si il y a un tuple correspondant, soit, à défaut, elle est créée.

b. Insertion d'entrées dans la base de données

b1. Création de patient

Cette page affiche un unique formulaire structuré selon la quantité et la qualité des informations nécessaires pour réaliser une nouvelle entrée sur la table 'Patients'. En effet pour cela, il faut au préalable avoir renseigné la table 'Villes' et l'ensemble des attributs qu'elle contient; ou bien récupérer la clé primaire d'un tuple dont l'ensemble des champs correspond aux informations de l'utilisateur ( ex: même ville, même code postal, même département, même pays). Ensuite, il faut renseigner la table 'Adresse' où la clé étrangère pointant sur la ville est alors fournie par l'étape précédente. De même que pour l'étape précédente, le site vérifie si un tuple comporte un ensemble de valeurs d'attributs associé correspondant à l'entrée et récupère la clé primaire de celui-ci, ou effectue une nouvelle entrée, si ce n'est pas le cas en récupérant la clef étrangère de celle-ci. Enfin, les données relatives à la table patient sont insérées, ici encore, en utilisant entre autre les données du formulaire d'une part et la clé primaire de la table parent récupérée à l'étape précédente. Le numéro de sécurité sociale étant unique, il garantit l'intégrité référentielle de cette table.

b2. Création de compte utilisateur

Si l'utilisateur est l'administrateur, alors il a accès à l'onglet permettant d'accéder directement à cette page et le test d'identité est validé. Le formulaire est alors généré et traité en cas de validation de manière similaire au cas de création de patient. Ici, la spécificité est que la clé primaire de la table Compte Utilisateur est généré automatiquement, en fonction des initiales de l'utilisateur et de 5 chiffres 'aléatoires' (basés sur la date courante).

b3. Création de services

Si l'utilisateur est l'administrateur, alors il a accès à l'onglet permettant d'accéder directement à cette page et le test d'identité est validé. Comme pour les formulaires précédents, la structure est basée sur les informations à collecter pour réaliser l'entrée. La gestion d'erreurs prévoit de tester l'existence ou, le cas échéant, l'ajout de la localisation du service qui représente une clé étrangère sur la table 'Services'.

b4. Création d'une intervention

De manière similaire aux cas précédents, la page fait appel au formulaire 'recherche de service', si aucun service n'a été sélectionné au préalable, ou affiche après sélection un formulaire.

b5.Demande de rendez-vous

Lorsqu'un utilisateur clique sur demande de rendez-vous, si aucun patient n'est sélectionné alors le formulaire de sélection de patients est appelé. Après validation, l'identifiant est stocké dans la variable '$\_SESSION' et l'utilisateur est redirigé sur la même page. Lorsque cette variable est renseignée, alors la page appelle un formulaire comportant les informations nécessaires pour réaliser l'entrée. Après validation, tous les cas de gestion d'erreurs liés au formulaire sont vérifiés. Si tous ces cas sont négatifs, alors la fonction ProchainCréneaux(), dont l'algorithme est basé sur l'utilistation d'un mélange de langage php est mySQL, est appelée.

La fonction de recherche de créneaux lance une requête sql composée de 4 parties :

Min(Max(Max(rdv prévu auj.) + Max(rdv prévu>auj)) + Min(rdv annulé auj.) + Min(rdv annulé>auj.)) + une contrainte relative aux horaires du service

Cette requête retourne ainsi le dernier créneau occupé ou le prochain créneau disponible. Ainsi, selon le statut de ce créneau, 15 min si prévu (durée standard d'un rendez-vous).

Si aucun retour n'est renvoyé par cette requête, alors, si cette heure est inférieure à l'heure d'ouverture, on écrase l'heure du créneau par l'heure d'ouverture du service. En revanche, si l' heure est supérieure ou égale à l'heure d'ouverture alors on écrase l'heure par l'heure d'ouverture et on écrase la date en la remplaçant par celle du lendemain. Sinon on attribue au créneau, l'heure et la date actuelle. Si le niveau d'urgence est <, alors on appel la fonction de gestionUrgence(). Enfin, on teste la compatibilité du niveau d'urgence et on insère le rendez-vous dans la table 'CreneauxIntervention'.

b6. Gestion des urgences

La première étape de cette fonction vise à établir l'heure et la date de fin du délai à compter de maintenant. Ensuite, si le créneau retourné par la fonction ProchainCréneaux() est compatible (<) avec le niveau d'urgence demandé alors la procédure se termine et le rendez-vous s'insère de manière classique. Si ce n'est pas le cas, une nouvelle requête est lancé pour rechercher

-le dernier créneau du planning dont le niveau d'urgence est supérieur ou égale au niveau demandé

-le premier créneau annulé

Min(Max(Max(rdv prévu auj.) + Max(rdv prévu>auj)) + Min(rdv annulé auj.) + Min(rdv annulé>auj.)) + pas de contraintes relatives aux niveaux d'urgence.

S'il n'y a pas de créneaux de même niveau d'urgence déjà présent dans le planning est que le retour de la requête est un tableau vide, on attribue alors le rendez-vous immédiatement ( avec comme créneaux le jour d'aujourd'hui et l'heure arrondie à la prochaine quinzaine). Tous les rendez-vous suivants sont décalés de 15 minutes jusqu'à ce que la boucle rencontre un rendez-vous dont le statut est annulé.

b6. Enregistrement des notifications

Lorsqu'une intervention est demandée, la fonction de gestion d'attribution des créneaux appelle deux fonctions relatives aux envois de notifications.

1. Elle teste systématiquement ( car il y a un niveau minimum aussi ), s'il le niveau d'urgence demandé est compatible avec les niveaux de références. Si une incompatibilité est détectée, une notification est ajoutée sur la Table notification.

2. En cas de niveau d'urgence >0 , elle appelle la fonction de vérification de surbooking. Si le dernier rendez vous de la journée du créneau attribué est >= à l'heure de fermeture alors une notification est ajouté sur la Table de notification.

Chaque notification comporte l'identifiant du créneaux concerné ainsi que le nom du service auquel elle est attribuée ( ex. informatique adressée à l'administrateur dans le cas des notif 1., et cardiologie pour le chef de cardiologie dans le cas des notif 2)

C. Généralités sur l'affichage des pages

Plusieurs pages proposent un affichage dynamique pour permettre, entre autre, la sélection de pré-requis pour accéder aux étapes suivantes. Cette fonctionnalité est basée sur la pérennité des informations contenues dans le '$\_SESSION' tout au long de la session d'utilisation. Pour assurer la modularité de cette dynamique, les clés générées pour l'affichage des pages sont systématiquement supprimées lors du retour sur la page d'accueil ( redirection assurée par bouton 'abandonner'). D'autres associations clé-valeur sont stockées dans cette variable tels que les noms, prénoms et identifiants de l'utilisateur, ou encore ses droits d'utilisateur ( bool '$\_SESSION['chef']'). D'autres valeurs de la base de données sont récupérées ponctuellement selon les besoins, comme lors des cas de modification.

D. Mises à jour des entrées de la base de données

d1. Modification des fiches patient

Cette page permet de mettre à jour les informations d'une ou des tables associées à l'enregistrement d'un patient ( tables : 'Patients', 'Villes', 'Adresses' ), L'affichage se fait de manière différentielle selon le contenu de '$\_SESSION'. Si ce tableau ne comporte pas de clé nommée 'patient' alors la page appelle un formulaire ( du répertoire formulaire ), qui permet d'identifier un patient présent dans la base de donnée selon ses noms et prénoms. Lors de la validation de la sélection, si aucune erreur n'est identifiée, alors la clé primaire identifiant le patient est affectée à '$\_SESSION['patient']' et l'utilisateur est alors redirigé vers cette même page.

Quand cette variable existe alors la page affiche un formulaire de même structure que pour l'ajout d'un nouveau patient. Cependant, ici le formulaire affiche comme valeur par défaut, dans chaque champ, l'ensemble des informations associées au patient. Lors de la validation, les instructions de validation sont semblables concernant la gestion d'erreur au cas d'utilisation 'Ajout de patient', en revanche la requête ne correspondant plus à une insertion mais à une mise à jour.

d2. Modification du compte utilisateur

Si l'utilisateur est l'administrateur, alors il a accès à l'onglet permettant d'accéder directement à cette page et le test d'identité est validé. Avec un affichage différentiel, le comportement de la page est identique à celui du cas précédent. De même, comme pour le cas de modification patient, le formulaire est généré selon la structure de création de compte utilisateur additionné de valeurs par défaut correspondant aux valeurs pré-existantes dans la base de données pour le compte sélectionné. Il est à noter que l'utilisateur ne peut modifier son identifiant.

d3. Modification Service

Si l'utilisateur est l'administrateur, alors il a accès à l'onglet permettant d'accéder directement à cette page et le test d'identité est validé. La logique de conception et d'implémentation de ce cas d'utilisation est identique aux deux précédents cas de modification.

d4. mises à jour des niveaux d'urgence de référence et gestion des notifications

Les notifications stockées lors de la prise de l'attribution de créneaux, peuvent être consulté par les chefs de service. Ainsi lors de l'ouverture de la page, la fonction d'affichage recherche si l'utilisateur est chef pour restreindre l'accès, mais recherche aussi le service concerné pour afficher les notifications spécifiques.

Si il existe des notifications, alors, un tableau dynamique est généré en fonction du nombre de notifications mais aussi du nombre d'attributs. Des boutons sont aussi générés dynamiquement pour chaque intervention de manière à permettre de garantir la possibilité d'une interaction indépendante de l'utilisateur avec chaque notification.

Comme vu précédemment, l'administrateur reçoit des notifications en cas d'incompatibilité. Celui-ci peut choisir de mettre à niveau de la valeur de référence selon la demande de l'utilisateur, et ainsi supprimer la notification. Cependant, il peut aussi refuser la mise à jour et simplement supprimer la notification.

Les chefs de services ont accès aux notifications de surbooking prévu dans leur service d'affectation, ils peuvent consulter cette notification et la déclarer comme vue pour l'effacer.

E. Généralités sur l'archivage

Certaines tables ont été dédoublées avec l'appellation 'TableArchive' pour permettre de basculer des informations dans ces nouvelles tables. Cette manipulation permet de restreindre l'accès des utilisateurs à l'information mais de conserver une trace des informations relatives aux échanges d'intervention entre les services. En effet, si, par exemple, un service est définitivement fermé ou arrête de dispenser un acte alors, les utilisateurs ne pourront pas demander de rendez-vous pour celui-ci.

F. Suppression et archivage des données

f1. Suppression de compte utilisateur

Seul l'administrateur a accès à cette fonctionnalité. Il peut ainsi supprimer un utilisateur de la table 'Compte Utilisateur', cependant les données de la table 'employé' sont conservées.

f2. Archivage d'actes médicaux

L'identité de l'utilisateur est testé, si celle ci correpond à l'administrateur alors la page archivage d'actes médicaux s'affiche de manière dynamique. Si aucun service n'est identifié dans '$\_SESSION', alors il y a une redirection vers le formulaire de recherche d'actes.

Si un acte est identifié dans cette variable, alors l'acte en question est basculé vers une table d'archive et l'utilisateur est redirigé vers la même page comportant un message de validation. Plusieurs gestions d'erreurs garantissent encore ici le bon déroulement de la procédure ( variables non vides, idIntervention existant ...ect).

Il est à noter, que cette table est la table parent de 'Tarification'. Ainsi, une table d'archivage a été créée. Les informations relatives aux tarifs sont stockées dans une variable temporaire, l'intervention est déplacée dans la table d'archivage et le nouvel identifiant attribué par auto-incrémentation est alors récupéré. Ainsi, l'étape final correspond à l'entrée du tarif sur la table TarificationArchivage avec comme valeurs de clé étrangère, la nouvelle clé primaire de l'acte.

D. Le traitement des requêtes

Après exécution d'une requête mySQL, il est parfois nécessaire d'appliquer un traitement particulier pour accéder et manipuler les données retournées. La structure de ces données peut varier de la valeur d'un unique attribut à plusieurs de plusieurs attributs. Pour cela, le module 'Fonctions\_reqTraitement.php' comporte l'implémentation de plusieurs fonctions de récupération (boucle while) et de réorganisation des retours de requêtes. Chaque fonction, parcours la requête et stocke les informations dans des tableaux de 1, 2 ou 3 dimensions selon les nécessités.

E. Planning

Lorsque l'utilisateur arrive sur le planning, la page appelle la fonction planning() pour récupérer l'ensemble des informations nécessaires. Cette fonction récupère les heures de début et de fin de service théoriques et réelles, ainsi, l'affichage permet un nombre de lignes minimums en accord avec les horaires théoriques du service, ou un ajustement à l'heure réelle en cas de surbooking.

Le premier affichage est celui du planning du jour du service auquel l'utilisateur est rattaché. Cependant, l'utilisateur peut sélectionner un autre jour et une autre date pour naviguer dans le planning. De plus, l'affichage du planning peut générer un ensemble de boutons en fonction des caractéristiques du rendez-vous, de l'heure de la date actuelle, et de l'utilisateur. En effet, pour un rendez-vous dont le statut est 'prévu' et dont la date/heure sont passés. Un bouton 'R' apparaît, en cas de clique, le statut du créneau correspondant est mis à jour et devient 'réalisé' permettant ainsi l'édition d'une facture.

Si l'utilisateur est le chef du service en question ou l'utilisateur ayant demandé le rendez-vous et que le créneau n'est pas encore passé, alors des boutons annulé et modifié apparaissent.

Le bouton supprimé 'X' , après confirmation, lance une requête de mise à jour du créneau correspondant en modifiant le statut en 'annulé'. En revanche, le bouton mettre a jour individuellement les autres champs de la base de données relatif à ce rendez-vous.

F. Visualisation des statistiques de demande de rendez-vous

Les données à afficher sont évaluées grâce à l'utilisation de la fonction SQL d'agrégation. De plus, des structures de contrôle de type boucle ont permis d'itérer les requêtes, par exemple, pour compter le nombre de demandes pour chaque niveau d'urgence. L'ensemble des données collectées sont stockées dans un tableau à 3 dimensions. Les informations ainsi stockées sont ensuite retournées dans un tableau généré de manière dynamique grâce à des boucles en language php, qui produit les balises php nécessaires.

E Facturation

Si l'utilisateur est un chef, et qu'aucun patient n'est associé à la variable '$\_SESSION' alors, un formulaire de sélection de patients est affiché. Après validation du choix du patient, si toutes les étapes de gestion d'erreurs se déroulent bien alors une facture comportant l'ensemble des actes réalisées s'affiche. Une requête recherche alors tous les rendez-vous dont le statut est 'réalisé' associés au patient sélectionné et au service de rattachement de l'utilisateur. Si des créneaux remplissant ces conditions sont sélectionnés, alors une facture apparaît. Sinon, un message informe l'utilisateur qu'aucune intervention ne peut être facturé pour ce patient. L'utilisation de requête avec opérateurs arithmétiques permet d'afficher les tarifs TTC et HT.

## Stockage des informations nécessaires au système

Les multiples informations indispensables au fonctionnement du site de gestion des échanges d’interventions médicales de l’hôpital sont enregistrées, protégées et organisées dans une base de données.

Cette base de données permet tout d’abord d’enregistrer les **comptes d’utilisateurs** et les mots de passe (cryptés) de tous les utilisateurs autorisés à s’authentifier sur le site.

Les utilisateurs du site sont l’administrateur, les responsables de services et les médecins **employés** de l’hôpital. Leur nom, prénom, service, ainsi que leur téléphone, email et adresse sont enregistrés. Les **chefs de services** sont répertoriés pour chaque service.

Les différents **services** de l’hôpital sont classés selon leur nom. Leur téléphone et email sont renseignés, ainsi que leur localisation dans l’hôpital et leurs horaires d’ouverture et de fermeture.

Cette **localisation** est déterminée par l’aile et l’étage du bâtiment dans lesquels se situe le service.

Chaque service propose différentes **interventions,** pour lesquelles est spécifié l’acte médical réalisé. A chaque acte médical est appliquée une **tarification** spécifique.

Lorsqu’un **patient** se rend à l’hôpital, il est pris en charge par le service d’accueil, qui va demander une intervention après diagnostic. Pour ce faire, le médecin de prise en charge doit créer ou éditer une fiche patiente. Cette fiche est constituée du numéro de sécurité sociale, des nom, prénom, date de naissance, sexe du patient ; de son téléphone et mail ; de sa taille et de son poids enregistrés lors de la consultation, ainsi que de commentaires que le médecin trouve important de mentionner.

Les **adresses** des employés et des patients sont stockées afin de notamment faciliter l’envoi des factures. Y sont enregistrés les noms et numéro de rue, la **ville**, le code postal, le département et le pays de résidence.

Une fois la fiche patiente remplie, le médecin peut demander au système de trouver un **créneau** disponible dans le planning du service de l’intervention nécessitée. Pour cela, il doit préciser au système le niveau d’urgence de l’intervention, afin d’éventuellement demander au responsable du service associé de prioriser l’intervention. Une fois le créneau de l’intervention attribué dans le planning avec l’approbation du patient, une fiche de rendez-vous est créée. Cette fiche est constituée de la date et de l’heure prévue de l’intervention, de la **pathologie** à traiter, du niveau d’urgence stipulé, du numéro de sécurité sociale du patient sur laquelle l’intervention est réalisée, de l’acte médical prescrit, du numéro du médecin qui a diagnostiqué le patient ainsi que des commentaires ajoutés par ce dernier. Le statut de l’intervention est aussi affiché. « Prévue » lorsque le créneau a été attribué dans le planning, « annulé » quand le patient a annulé son rendez-vous et « réalisée » lorsque l’intervention a bien eu lieu.

Une **facture** est éditée pour chaque intervention réalisée.

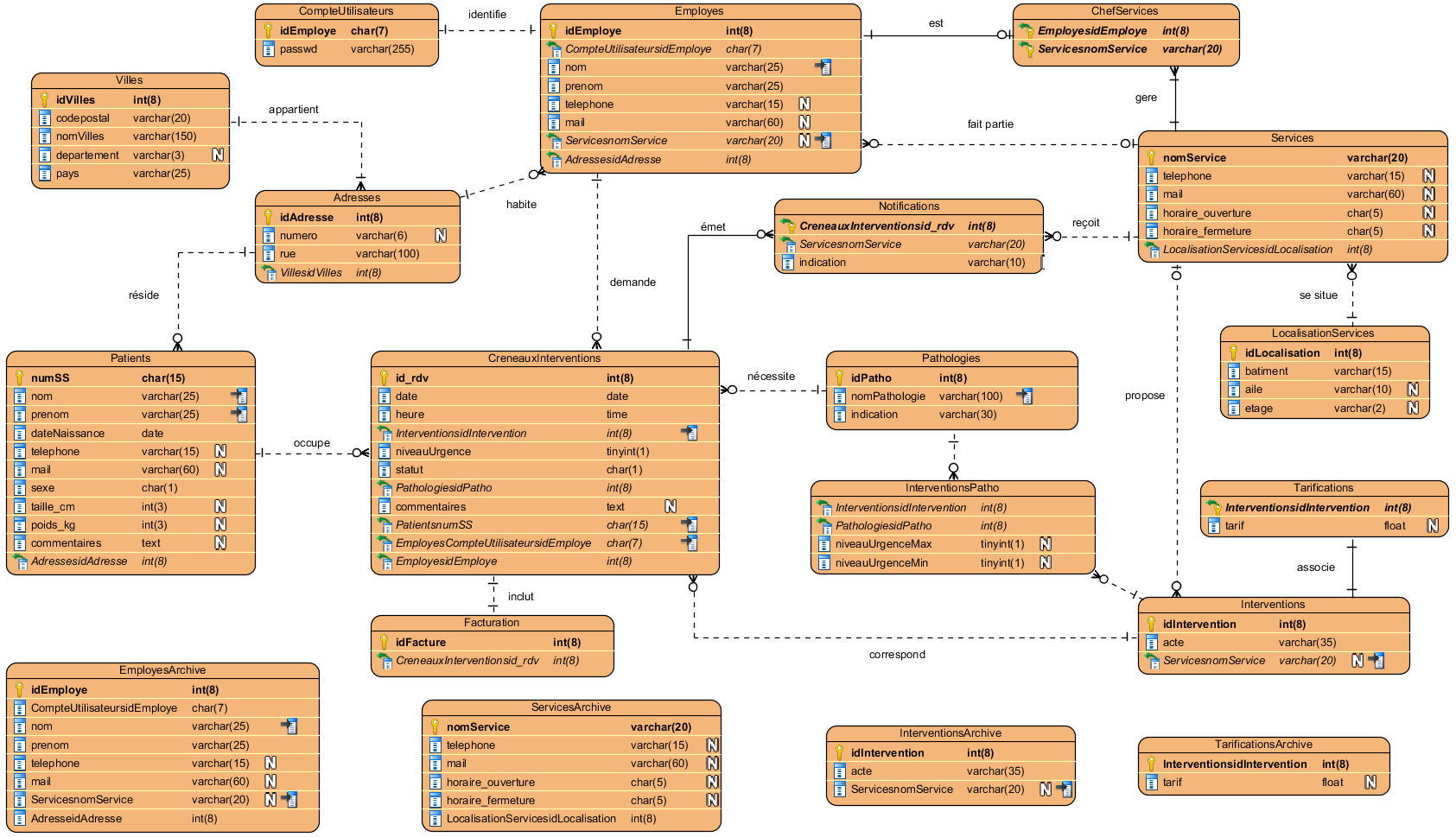
Enfin, des **notifications** sont envoyées aux services responsables des interventions devant être priorisées.

**Diagramme Entité-Association**

Le diagramme entité-association de la base de données « gestion de planning des interventions au sein d’un hôpital » est réalisée selon le modèle UML. Notre diagramme comporte ainsi 15 entités.

Ces entités sont reliées par un ensemble de 18 associations, caractérisées par des contraintes structurelles de participation et de cardinalité. Les associations de ce diagramme sont :

* Identifier : Un compte utilisateur identifie de manière unique un employé de l’hôpital et inversement un employé de l’hôpital ne possède qu’un seul compte.
* Etre : Un chef de service est forcément un employé mais un employé n’est pas obligatoirement chef de service.
* Gérer : Un chef de service gère un unique service et un service peut être géré par un ou plusieurs chefs.
* Faire partie : Un employé ne pourra faire partie que d’un seul service. Un service pourra ne comprendre aucun employé le temps de sa création et pourra en accueillir plusieurs une fois en place.
* Se situer : Un service se situe à une unique localisation dans l’hôpital. Une localisation de l’hôpital peut être attribuée à aucun ou plusieurs services.
* Proposer : Une même intervention ne peut être proposée que par un seul service (ou aucun si le service qui réalisé cet acte ferme), et un service peut proposer zéro ou plusieurs interventions.
* Correspondre : Une intervention peut correspondre à zéro ou plusieurs créneaux de rendez-vous tandis qu’un créneau ne correspond qu’à une seule intervention.
* Associer : A chaque intervention est associé un tarif unique. Un tarif s’associe à une seule intervention.
* Demander : Un employé peut demander zéro ou plusieurs créneaux d’interventions médicales. Une intervention n’est réservée que par un seul employé.
* Occuper: Un créneau de rendez-vous pour une intervention n’est occupé que par un patient et un patient peut occuper zéro ou plusieurs créneaux.
* Résider : Un patient ne peut résider qu’à une seule adresse. A une adresse peuvent vivre zéro ou plusieurs patients.
* Appartenir : Une adresse appartient à une ville. Une ou plusieurs adresses peuvent appartenir à une même ville.
* Habiter : Un employé n’habite qu’à une seule adresse mais à une même adresse peuvent habiter zéro ou plusieurs employés.
* Inclure : Chaque créneau d’intervention réalisé inclut une seule facturation. Une facturation correspond à une seule intervention réalisée sur un patient.
* Nécessiter : Une même pathologie nécessite zéro ou plusieurs interventions réalisées dans l’hôpital. Un créneau d’intervention est réservé pour une unique pathologie.
* Emettre : une intervention peut émettre zéros ou plusieurs notifications, une notification est émise par un seul créneau d’intervention.
* Recevoir : Un service reçoit zéro ou plusieurs notifications, mais une notification n’est reçue que par un unique service

****

**Figure 4** : Diagramme entité-association de la base de donnée

Changer la multiplicité Facturation / CreneauInter

**Normalisation**

L’étape de normalisation des relations permet de limiter la redondance des données qui pourrait causer des problèmes lors de mises à jour.

1. Recherche d’attributs composites ou multivalués.

Les adresses étant composées d’un numéro de rue, d’un nom de rue, d’un code postal, d’une ville et d’un pays. Il nous a fallu diviser cet unique attribut complexe « adresse » en plusieurs attributs « Numero, Rue, codePostal, Ville, Pays ».

Cet attribut se retrouvant dans la table Employé et Patient, nous avons préféré créer une nouvelle table adresse regroupant ainsi les adresses de nos deux entités.

La localisation d’un service fonctionne à peu près comme une adresse. Nous avons donc également divisé l’attribut « localisation » en plusieurs attributs « bâtiment, aile, étage, porte » et regroupés ces nouveaux attributs dans une nouvelle table Localisation.

De plus, nous ne prenons en compte que les premiers prénoms des employés ou patients dans notre base de données et également l’adresse de la résidence principale du patient ou de l’employé. Cette simplification nous évite la création de tables supplémentaires comme dans les cas précédents.

Après cette première étape de normalisation, les tables obtenues sont en première forme normale (1NF). C’est à dire qu’elles ne possèdent plus d’attributs composites ou multivalués. On peut alors analyser les relations fonctionnelles entre les attributs.

1. Analyse des dépendances fonctionnelles

Pour chaque table, il faut rechercher quels attributs dépendent fonctionnellement d’autres attributs, sachant qu’un attribut dépend fonctionnellement d’un autre si chaque valeur du premier est associé à exactement une valeur du deuxième.

1. Dépendance partielle

Pour que nos tables soient à la deuxième forme normale (2NF), une fois en 1NF, nous nous sommes assuré que chacun de leurs attributs de clé non primaire étaient fonctionnellement complètement dépendants de leur clé primaire.

Cette étape nous a poussés à définir en tant que clé primaire des identifiants uniques pour certaines de nos tables comme pour la table CompteUtilisateurs (idEmploye).

1. Dépendance transitive

Pour qu’une table soit de troisième forme normale (3NF), il faut qu’elle soit sous forme 2NF et qu’aucun de ses attributs ne dépende transitivement de la clé primaire.

Dans un premier temps nous avons identifié les dépendances transitives présentes dans chacune de nos tables. Les attributs non primaires dépendants d’autres attributs non primaires ont été placés dans une nouvelle table de sorte à effacer cette dépendance transitive tout en gardant un lien avec la clé primaire de leur table d’origine.

Par exemple, le tarif de chaque intervention était intégré à la table Intervention. Cet attribut dépendait de sa clé primaire d’origine (idIntervention) mais aussi de l’acte médical (en effet on part du principe qu’un même acte médical coûte un prix unique). Nous avons donc créé une table tarification comprenant deux attributs : le tarif ainsi que l’intervention associé en tant que clé primaire pointant vers la table Interventions.

Après ces vérifications, les tables considérées comme étant au format UNF (*Un-Normalized-Form*) sont désormais en troisième forme normale (3NF).

# Contrôle qualité

## 1. Tests fonctionnels

Pour présenter l’ensemble des tests fonctionnels effectués, nous avons choisi de les lister pour chaque cas d’utilisation vu précédemment. L’ensemble des affirmations suivantes a donc été vérifié :

* **Cas s’authentifier :**
* Message d’erreur si l’utilisateur entre un mauvais mot de passe/mauvais identifiant
* L’utilisateur peut entrer son identifiant avec les lettres en majuscules ou en minuscules
* **Cas Créer Fiche Patient :**
* Tous les utilisateurs peuvent créer une fiche patient
* Message d’erreur si l’utilisateur entre un numéro de SS déjà existant
* Les noms et prénoms peuvent être écrits en majuscule
* Les noms et prénoms ne peuvent pas contenir des chiffres ou des caractères spéciaux
* Un numéro de téléphone peut contenir des espaces et des caractères spéciaux
* L’adresse email respecte le format xxx@xxx.xx
* Message d’erreur si une date de naissance à un des champs jour/moi/année non renseigné
* Suppression automatique des lettres et des espaces dans la taille et le poids
* Les tailles et les poids sont contraints à maximum 3 valeurs numériques
* Message d’erreur si un numéro de rue est entré avec des caractères autres que des entiers
* Message d’erreur si une adresse de patient est entrée avec des champs invalides
* Quand on rentre un patient et son adresse, l’adresse est liée à ce patient dans la base de données (BdD)
* Le nom de la rue supporte tous les caractères possibles
* Les champs stipulés obligatoires affichent une erreur si laissés vides
* Les champs non obligatoires peuvent être laissés vides
* Le numéro de département accepte les lettres
* Le code postal accepte n’importe quel caractère
* L’ idVilles et l’Adresse.idVilles sont bien liés dans la BdD
* Gestion des espaces
* **Cas modifier une fiche patient :**
* Disponible pour tous les utilisateurs
* On ne peut modifier que les fiches des patients déjà présents dans la BdD
* Message d’erreur si on rentre un mauvais n° de SS
* Le champ date de naissance est au format date
* L’adresse email est au format xxx@xxx.xx sans accents
* Contraindre les tailles et poids à maximum 3 valeurs numériques
* Les modifications sont enregistrées dans la BdD
* **Cas créer un compte utilisateur  :**
* Seul l’admin peut créer un compte utilisateur
* Les informations renseignées (adresse, mot de passe, service) sont enregistrées dans la BdD
* Un identifiant unique est créé automatiquement pour chaque employé
* On peut mettre des espaces et des « -» dans les noms et prénoms, mais pas de valeurs numériques
* Lorsque le médecin est coché en tant que chef, il apparaît comme chef de son service dans la table ChefService de la BdD
* Il peut y avoir plusieurs chefs pour le même service
* On ne peut pas mettre de lettres et de caractères dans le numéro de département
* Le numéro de département peut faire 3 chiffres
* Le code postal accepte n’importe quel caractère
* L’adresse mail est créée automatiquement sous la forme identifiant@hopitalbim.fr
* Le mot de passe de l’utilisateur contient minimum 8 caractères sans espace.
* Les champs stipulés obligatoires affichent une erreur si laissés vides
* Les champs non obligatoires peuvent être laissés vides
* **Cas modifier un compte utilisateur :**
* Seul l’admin peut modifier un compte utilisateur
* Tous les champs d’un utilisateur modifiés sont bien enregistrés dans la BdD.
* Gestion des espaces
* Ajouter la possibilité d’attribuer ou de retirer le rôle de chef de service à un employé
  + Faire en sorte que si on supprime son rôle de chef de service, il n’apparaisse plus dans la table ChefsService
  + Faire en sorte que si on modifie le champ service d’un médecin, et que ce médecin était chef d’un service, il n’apparaisse plus dans la table ChefsService
* **Cas supprimer un compte utilisateur :**
* Seul l’admin peut supprimer un compte utilisateur
* Lorsqu’on supprime un utilisateur, il est supprimé des tables « Employés » et « Comptes utilisateurs »
* Si on supprime un compte de chef de service, il disparaît aussi de la table « ChefService »
* Message d’erreur si on demande de supprimer un compte non existant
* Un compte utilisateur supprimé apparaît dans la table « EmployeArchive »
* **Cas créer un service :**
* Seul l’admin peut créer un service
* Message d’erreur si le nom de service est déjà présent
* Un nom de service peut contenir des espaces et des chiffres
* Un horaire d’ouverture doit être strictement inférieur à l’heure de fermeture
* Gestion des espaces pour le mail automatiquement créé
* L’adresse du service créé est bien enregistré dans la BdD
* Un nom de service écrit en majuscule ou en minuscule est le même
* Les champs stipulés obligatoires affichent une erreur si laissés vides
* Les champs non obligatoires peuvent être laissés vides

* **Cas archiver un service :**
* Seul l’admin peut archiver un service
* Lorsqu’on archive un service, il n’y a plus de chef de service associé à ce service dans la table chef de service.
* Lorsqu’on archive un service, tous les employés attachés à ce service ont un « NULL » à la place de leur ancien service archivé
* Message d’erreur si on demande de supprimer un compte non existant
* Un service archivé apparaît dans la table ServicesArchive
* **Cas modifier un service**
* Seul l’admin peut modifier un service
* Lorsqu’on modifie les champs d’un service, ils sont bien enregistrés dans la BdD
* **Cas demander une intervention**
* Disponible pour tous les utilisateurs
* Message d’erreur si un champ sélectionnable (patient, intervention) n’existe pas
* Tous le champs renseignés dans le formulaire sont correctement enregistrés dans la BdD
* Gestion des espaces pour les champs
* Le système n’attribue jamais le même créneau pour 2 demandes d’interventions
* Un même patient ne peut pas subir 2 interventions différentes en même temps
* Les champs stipulés obligatoires affichent une erreur si laissés vides
* Les champs non obligatoires peuvent être laissés vides
* Une intervention s’insère dans le premier créneau vide
* **Cas gestion des priorités**
* Si on demande une intervention urgente de niveau 1, elle doit être insérer dans les 24h
  + Bug sql notifications
* Si on demande une intervention urgente de niveau 2, elle doit être insérer dans les 6h
  + Bug sql notifications
* Si on demande une intervention urgente de niveau 3, elle doit être insérer dans les 3h
  + Bug sql notifications
* Une intervention urgente qui décale d’autres interventions augmente de 1 le niveau d’urgence de ces interventions décalées
* Une intervention urgente de niveau N ne peut pas être insérée avant une urgence de niveau N+1, sauf si il y a un créneau libre entre
* **Cas Notifications**
* Seul les chefs de services concernés par le surbooking de leur service reçoivent ces notifications
* Un décalage du planning qui nécessite un surbooking est notifié au chef du service concerné
* **Cas facturation**

- Seuls les responsables du service dans lequel a eu lieu l’intervention à facturer peuvent facturer l’intervention

-Seules les interventions réalisées peuvent être facturées

- Les factures correspondent bien aux interventions effectuées, avec le bon patient, la bonne date et la bonne tarification.

* **Cas modification du planning**
* Disponible pour tous les utilisateurs
* La suppression d’une intervention via l’interface graphique du planning est bien enregistrée dans la base de données
* Les modifications d’horaires d’un créneau d’intervention sont bien enregistrées
* On ne peut pas modifier/supprimer un rendez-vous déjà réalisé
* Une erreur est indiquée si les champs modifiés sont laissés vides ou s’ils sont dans un format incorrect
* **Cas statistiques des demandes**
* Seul l’admin a accès aux statistiques des demandes
* Le tableau des statistiques des demandes est correct et dynamique
* **Cas vérification des compatibilités d’urgence**
* Seul l’admin reçoit des notifications
* Lorsque le système détecte une incompatibilité entre la pathologie/ le niveau d’intervention et le niveau d’urgence demandé, une notification est envoyée à l’administrateur
* **Cas archivage acte**
* Seul l’admin peut archiver un acte
* Lorsqu’on archive un acte, il est bien enregistré dans la table « InterventionsArchive »
* On ne peut pas Archiver un acte non existant
* Un acte archivé n’est plus proposé dans la liste des actes lors d’une demande d’intervention
* **Cas ajout d’un acte**
* Seul l’admin peut ajouter un acte
* Lorsque l’utilisateur entre un service qui n’existe pas, un message d’erreur s’affiche
* Le nom de l’acte médical supporte tous les caractères
* Le tarif rentré est un float (sinon message d’erreur)
* Les nouvelles informations entrées sont enregistrées dans leurs tables respectives de la base de données, et sont disponibles dans le formulaire « demande d’intervention »
* On ne peut pas rentrer 2 actes avec le même nom
* Les champs stipulés obligatoires affichent une erreur si laissés vides

En revanche, d’autres tests ont révélé certaines faiblesses mineures qui n’ont pas été résolues :

* Lorsqu’on rentre une date avec l’année avec plus de 4 chiffres, la date est enregistrée avec tous ses champs vides (00/00/0000)
* Les accents de sont pas gérés, une entrée de type « entrée » sera différente d’une entrée de type « entrée »

Enfin, le test suivant n’a pu être testé par manque de temps : Si le niveau d’urgence requis pour l’intervention n’est pas compatible avec le niveau d’urgence maximum ou minimum requis pour la pathologie, l’administrateur en est signalé.

## 2. Tests unitaires

Pour garantir la fonctionnalité de notre projet, des tests unitaires ont été réalisés à chaque étape d'implémentation. Lors de l'utilisation de nouvelles fonctions natives, nous avons consulté le résultat retourné par celle-ci pour bien évaluer ce que fait concrètement la fonction avant de multiplier son utilisation. Aussi, durant l'implémentation de nouvelles fonctions ou procédures, nous avons testé pas à pas les effets de chaque instruction utilisée.

Pour visualiser de manière formatée la structure des données de type tableau, nous avons créer la fonction Dumper() basée sur l'instruction 'var\_dump'. En effet, nous avons aussi consulté le contenu des différentes variables manipulées, particulièrement dans le cas de retour de requête SQL. En effet, l'existence de différents types d'instructions et d'options pour le 'Fetch' est pratique et permet d'adapter la structure de la variable de retour en fonction du résultat attendu. Cependant lors des premières utilisations, il est assez difficile d'appréhender la structure des données retournées par la requête et d'évaluer comment le traiter. On peut noter cette difficulté notamment lors de l'élaboration de test conditionnel pour la gestion d'erreurs relatives au retour de requête vide. Par exemple, l'instruction Fetch(PDO::FETCH\_ASSOC) retour un tableau dont les clefs sont les attributs sélectionnés ou un booléen 'FALSE' si il n'y a pas de correspondances dans la table. En revanche, l'instruction fetchColumn() retourne une chaîne de caractères correspondant à la valeur d'attribut ou une chaîne vide s'il n'y a pas de correspondances. Ainsi, les tests conditionnels à mettre en place peuvent être très variables selon le mode de traitement des retours de requête. Le parcours des différents blocs d'instructions lors de l'élaboration de test, nous a permis d'évaluer la pertinence des tests établis.

Aussi, lors de l'emploi de structures de contrôle de type boucle, utilisées pour automatiser certaines procédures telles que la réitération de requêtes ou la génération dynamique de code HTML, nous avons évalué le comportement de celle-ci pour contrôler que le nombre d'itération correspondent à celui attendu.

La structure des formulaires et du CSS a été évalué directement manière visuelle sur les pages associées. On a aussi évalué le bon déroulement de l'ensemble des instructions associées aux boutons et consulté les valeurs des variables retourné par les code HTML de type 'form' .

L'intégrité structurelle de la base de données a été évalué sur l'interface phpMyadmin ( clés, nombre de colonnes, typage des attributs...). Cette interface a aussi été utilisée lors de la conception des requêtes. En effet, cette interface présente trois avantages à nos yeux, elle permet l'exécution individuelle de la requête, le retour est consultable immédiatement et enfin, l'implémentation sur cette interface plus de confort avec la complémentation automatique en accord avec la structure. PhpMyadmin a permis l'observabilité des tests concernant les requêtes SQL et l'opérabilité des tests portant sur les formulaires d'insertion de valeurs notamment lors des étapes préliminaires de développement (par ex. insertion de rendez-vous mais pas encore d'affichage du planning).

# Retours d’expérience

1. Démarche adoptée (lors de l’élaboration du site)
2. Difficultés rencontrés

Si nous devions refaire ce sujet, nous changerions notre approche de certains points du projet, à différents niveaux. Au niveau du développement, nous nous sommes aperçus assez tardivement de la redondance de nos différentes fonctions et de nos différentes gestions d’erreurs. Une application plus systématique de celles-ci aurait permis un gain de temps non négligeable. L’usage de la programmation orientée objet aurait peut être facilité ce traitement, même si elle n’était pas dans les consignes du sujet. De plus, définir des conventions claires pour la structuration des codes aurait facilité la communication entre les différents développeurs.

Au niveau de la réalisation du planning, nous avons conçu une table « CréneauIntervention » dans laquelle nous avons séparé dès le départ la date (au format SQL « date ») et l’horaire (au format SQL « time »). Cette division a été préjudiciable dans la création de nombreuses fonctions en augmentant la complexité, notamment lors de la mise en place des requêtes SQL. Nous nous sommes rendu compte de l’existence d’un format SQL « datetime » trop tardivement pour l’introduire dans notre code, alors que ce format aurait grandement facilité la réalisation de certaines tâches.

Enfin, notre manque d’expérience nous a menés vers une approche parfois exploratoire de certains problèmes, nous écartant probablement de solutions d’implémentation efficaces utilisées dans le milieu professionnel. En effet, nous avons souvent essayé de multiplier les méthodes en explorant les diverses ressources de ces langages de programmation, afin de trouver nos solutions.

1. Améliorations

Un certain nombre d’améliorations du site sont envisageables, dont les plus importantes sont :

* L’amélioration de l’archivage des services/employés/interventions, par la création d’une fonctionnalité « récupérer une archive », qui permettrait à l’administrateur de récupérer les informations relatives à un archivage via le site, et non pas via la base de données comme c’est le cas dans cette version.
* L’ajout de la possibilité pour l’administrateur de pouvoir modifier un acte médical, notamment sa tarification si son coût est sujet à une baisse ou à une augmentation.
* L’ajout de la possibilité pour l’administrateur de définir des niveaux d’urgence maximum requis pour une pathologie donnée
* L’amélioration de la facturation, permettant d’imprimer la facture au format .pdf, et permettant d’enregistrer les interventions déjà facturés afin d’éviter de les refacturer à chaque facturation.
* L’implémentation d’un système de vérification inter champs, qui permettrait par exemple d’empêcher la création d’une fiche patient possédant un numéro de SS compatible avec le sexe et la date de naissance.

# ANNEXES