

Documentation Technique

JACLET Hôpital : Gestion des séjours

Sommaire :

<Elliot>

1) <u>Projet en général</u>	2-7
A - Utilité	2
B - Users Story	2-4
C - Comptes utilisateurs de l'application	5
D - Compte utilisateur du SGBD	5
E - Serveurs	6
F - Schéma relationnel (base de données)	7
G - Diagramme de classe	7
 2) <u>Projet point de vue technique</u>	 8-25
A - Fonctionnalités & Tests	8-23
B - Repositories	23-25

</Elliot>

Documentation Technique

JACLET Hôpital : Gestion des séjours

I) Projet en général

Utilité :

<Elliot>

Application WEB de gestion des séjours des patients.

Elle permettra aux personnels infirmiers de prendre en compte l'arrivée et la sortie d'un patient dans un service ainsi que de consulter les séjours à une date donnée et à venir.

Pour le personnel administratif, elle permettra de pouvoir gérer les patients et les séjours (ajout / suppression / modification).

L'application est codée via un Framework Symfony, le SGBD utilisé est MySQL sur le serveur local du lycée, l'OS utilisé est Windows 10 et / ou Windows 7.

</Elliot>

Users Story :

<Jordan>

User Story n°1 :

En tant que **personnel infirmier** je dois pouvoir **m'authentifier et voir « mon » menu**.

Si le login et le mot de passe sont corrects lorsque je clique sur « me connecter » j'ai accès à toutes mes fonctions.

Si le login et/ou le mot de passe sont incorrects alors lorsque je clique sur « me connecter » un message d'erreur s'affiche.

Les informations de connections doivent être également mémorisées sous forme d'historique de connections.

User Story n°2 :

En tant que **personnel infirmier** je peux **consulter les séjours à une date donnée**.

Si le personnel infirmier est bien connecté alors il peut en cliquant sur « Consulter les séjours à une date donnée » accéder à la liste des séjours effectifs à la date du jour et également choisir une autre date, valider et voir les séjours effectifs à cette nouvelle date. Il peut également afficher les informations du séjour en question.

Documentation Technique

JACLET Hôpital : Gestion des séjours

User Story n°3 :

En tant que **personnel infirmier** je peux **consulter les séjours à venir**.

Si le personnel infirmier est bien connecté alors il peut en cliquant sur « Consulter les séjours à venir » accéder à la liste des séjours à venir, il peut également en sélectionner un pour afficher les informations de celui-ci.

User Story n°4:

En tant que **Admin** je peux consulter l'historique de connections des utilisateurs

User Story n°5:

En tant qu'utilisateur disposant d'un compte, je peux en cliquant sur mot de passe oublié, renseigner mon adresse mail pour pouvoir obtenir un lien par mail qui me permettra de modifier mon mot de passe.

</Jordan>

<Elliot>

User Story n°1 :

En tant que **personnel infirmier**, je peux **m'authentifier** afin **d'avoir accès aux différentes fonctionnalités du site auxquelles j'ai accès**.

User Story n°2 :

En tant que **personnel infirmier**, je souhaite **visualiser les séjours des patients du jour affectés au service du personnel infirmier**.

User Story n°3 :

En tant que **personnel infirmier**, je souhaite **pouvoir valider les arrivées et sorties effectives de chaque patient du jour ainsi qu'un éventuel commentaire**.

User Story n°4 :

En tant que **personnel infirmier**, je souhaite **visualiser les séjours des patients en cours affectés au service du personnel infirmier**.

User Story n°5 :

En tant que **personnel infirmier**, je souhaite **pouvoir valider les arrivées et sorties effectives de chaque patient pour lequel le séjour est en cours ainsi qu'un éventuel commentaire**.

</Elliot>

Documentation Technique

JACLET Hôpital : Gestion des séjours

<Arthur>

User Story n°1 :

En tant que **personnel administratif**, je peux **m'authentifier** afin **d'avoir accès aux différentes fonctions auxquelles j'ai accès.**

User Story n°2 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les patients »** afin de **voir la liste des patients et d'avoir la possibilité d'en ajouter un.**

User Story n°3 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les patients »** afin de **voir la liste des patients et d'avoir la possibilité d'en modifier un existant.**

User Story n°4:

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les patients »** afin de **voir la liste des patients et d'avoir la possibilité d'en supprimer un existant.**

User Story n°5 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les patients »** afin de **voir la liste des patients et d'avoir la possibilité d'ajouter un séjour pour l'un d'entre eux.**

User Story n°6 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les séjours »** afin de **voir la liste des séjours et d'avoir la possibilité d'en ajouter un.**

User Story n°7 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les séjours »** afin de **voir la liste des séjours et d'avoir la possibilité d'en modifier un existant.**

User Story n°8 :

En tant que **personnel administratif**, étant donné que je suis connecté, je peux **choisir la fonction « gérer les séjours »** afin de **voir la liste des séjours et d'avoir la possibilité d'en supprimer un existant.**

</Arthur>

Documentation Technique

JACLET Hôpital : Gestion des séjours

Comptes utilisateurs de l'application

<Elliot>

Radiologie: **login** : Radio1 **mdp** : @Radio59

mail : radiologiejaclet@gmail.com

Cardiologie: **login** : Cardio1 **mdp** : @Cardio59

mail : cardiologiejaclet@gmail.com

Ambulatoire: **login** : Ambu1 **mdp** : @Ambu59

mail : ambulatoirejaclet@gmail.com

Urgence: **login** : Urg1 **mdp** : @Urg59

mail : urgencejaclet@gmail.com

Soins intensif : **login** : Soins1 **mdp** : @Soins59

mail : soinsjaclet@gmail.com

Administratif : **login**: Admin1 **mdp** : @Admin59 **mail** : Ø

</Elliot>

<Jordan>

Au niveau de la sécurité on définit que les routes sont accessibles qu'à certains rôles qui sont attribués dans la bdd à certains utilisateurs (ADMIN ou INFIRMIER).

Il y a également une contrainte sur les mots de passe (au moins une lettre, un chiffre et un caractère spécial) depuis le controller registration.

</Jordan>

Compte utilisateur du SGBD

<Elliot>

Utilisateur : **login** : JACLET-Sejour **mdp** : Jaclet-Sejour

Droits sur la base de donnée jaclet-sejour :

Données : SELECT, INSERT, UPDATE, DELETE

Structure : CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, CREATE VIEW, EVENT, TRIGGER

Administration : Ø

</Elliot>

Documentation Technique

JACLET Hôpital : Gestion des séjours

Serveurs

<Elliot>

- Serveur de production :

login: vizimmo **mdp :** vizimmo

réseau : accès par pont

URL à saisir pour aller sur le site jaclet séjour :

<http://192.168.1.188/JACLET-SEJOUR/JACLET-Sejour/public/index.php/index>

- Serveur RedMine :

login administrateur : admin **mdp administrateur :** ErtY1234

URL à saisir : <http://192.168.1.33>



Le serveur redmine est située sur le poste 27 de la salle de cours sur l'utilisateur sio2

- Serveur GLPI:

login administrateur : GLPI **mdp administrateur :** GLPIjaclet!6

URL à saisir : <http://192.168.1.179>



Le serveur GLPI est située sur le poste 27 de la salle de cours sur l'utilisateur sio2

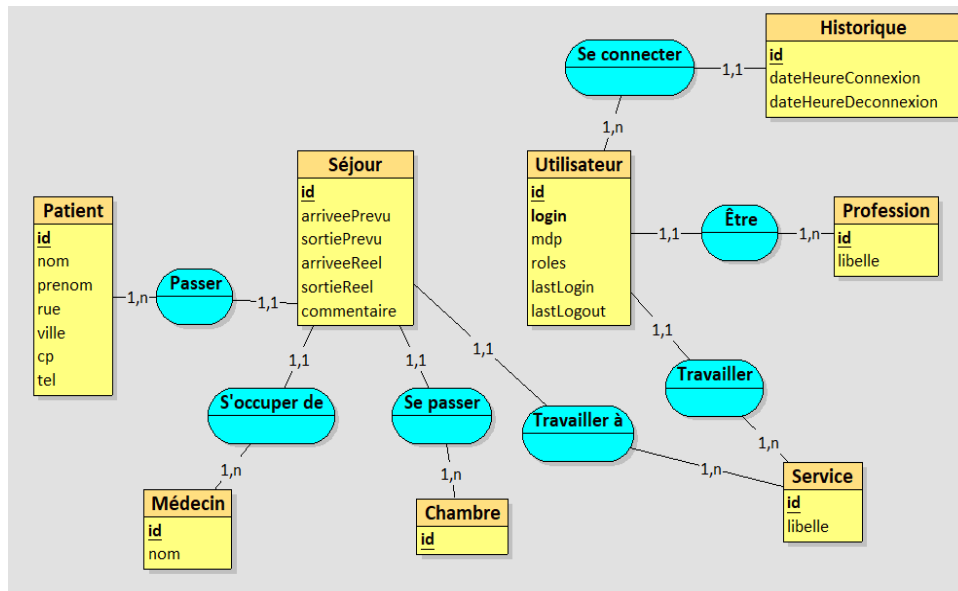
</Elliot>

Documentation Technique

JACLET Hôpital : Gestion des séjours

Schéma relationnel (Base de données)

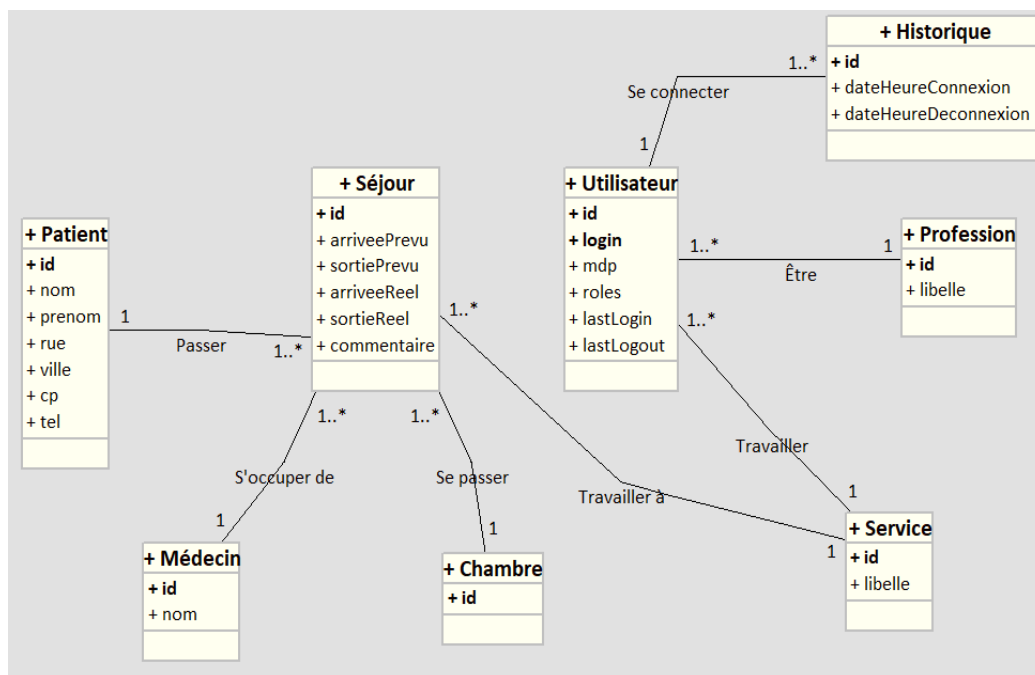
<Elliot/arthur>



</Elliot/arthur>

Diagramme de classe

<Elliot>



</Elliot>

Documentation Technique

JACLET Hôpital : Gestion des séjours

II) Projet point de vue technique

Fonctionnalités & Tests

<Arthur>

GererPatientsController :

index()

Permet de récupérer toute les données de la table Patients pour les passer et les afficher dans gerer_patients/index.html.twig

ajoutPatient(Request \$request)

affiche un formulaire qui permet de renseigner les informations pour un nouveau patient à ajouter dans la BDD

modifPatient(\$id, Request \$request)

affiche un formulaire qui permet de modifier les informations pour le patient que nous avons choisi grâce à son ID donné en attribut

removePatient(\$id)

supprime le patient que l'on a choisi grâce à son ID donné en attribut

affecterPatient(\$patient_id, Request \$request)

permet d'ajouter un séjour pour le patient choisi grâce à son ID donné en attribut

SejourController :

index()

Permet de récupérer toute les données de la table Sejour pour l'afficher dans sejour/index.html.twig

ajoutSejour(Request \$request)

affiche un formulaire qui permet de renseigner les informations pour un nouveau séjour à ajouter dans la BDD

modifSejour(\$id, Request \$request)

affiche un formulaire qui permet de modifier les informations pour le séjour que nous avons choisi grâce à son ID donné en attribut


removePatient(\$id)

supprime le séjour que l'on a choisi grâce à son ID donné en attribut

Documentation Technique

JACLET Hôpital : Gestion des séjours

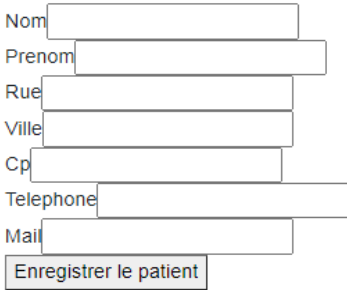
Test :

No test	Type scénario	Description
A1	Normal	S'identifier sur le site avec un nom d'utilisateur de type d'accès "personnel administratif" et son mot de passe associé.
Résultat attendus		
-l'utilisateur est redirigé vers l'écran d'accueil -l'écran d'accueil indique qu'il est connecté en tant que Admin du service d'administration -la barre n'affiche maintenant plus l'option "Se connecter" -la barre affiche maintenant aussi une option "Gérer les patients", "Gérer les séjours" et "Se déconnecter"		
Résultat		
<div><div>Hôpital JACLET</div><div></div></div> <div>Connecté en tant que Admin du service d'Administration</div> <div>Accueil Créer un compte Gérer les patients Gérer les séjours Se déconnecter</div>		

No test	Type scénario	Description
A2	Normal	En temps qu'admin j'appuie sur le bouton "Gérer les patients"
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de gestion des patients -la page a un bouton "ajouter un patient" -à côté de chaque patient se trouvent les boutons "modifier" "supprimer" et "ajouter un séjour"		
Résultat		
<div><div>Gérer les patients</div><div>Nb de patients : 20 ajouter un patient</div><div><ul style="list-style-type: none">Lécuyer Amélie - 0599067501 - gilles.bailly@live.com<div>modifier supprimer ajouter un séjour</div></div></div>		


Documentation Technique

JACLET Hôpital : Gestion des séjours

No test	Type scénario	Description
A3	Normal	En temps qu'admin, dans le menu gérer les patients, j'appuie sur le bouton "ajouter un patient"
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de création de patients -la page contient un formulaire qui sert à l'ajout des patients avec un bouton pour valider		
Résultat		
		

Documentation Technique

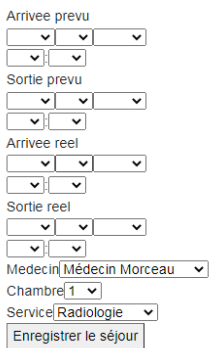
JACLET Hôpital : Gestion des séjours


No test	Type scénario	Description
A4	Normal	En temps qu'admin, dans le menu gérer les patients, j'appuie sur le bouton "modifier" à côté d'un patient
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de création de patients -la page contient un formulaire qui sert à l'ajout des patients déjà rempli avec les infos actuelle du patient avec un bouton pour valider la modification		
Résultat		
		

No test	Type scénario	Description
A5	Normal	En temps qu'admin, dans le menu gérer les patients, j'appuie sur le bouton "supprimer" à côté d'un patient qui n'a pas de séjours
Résultat attendus		
-la page s'actualise et le patient à bien été supprimé		
Résultat		
le patient est bien supprimé		

Documentation Technique

JACLET Hôpital : Gestion des séjours

No test	Type scénario	Description
A6	Normal	En temps qu'admin, dans le menu gérer les patients, j'appuie sur le bouton "ajouter un séjour"
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de création de séjours -la page contient un formulaire qui sert à l'ajout des séjours (sans le champ du nom de patient) avec un bouton pour valider		
Résultat		
		

No test	Type scénario	Description
A7	Normal	En temps qu'admin j'appuie sur le bouton "Gérer les séjours"
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de gestion des séjours -la page a un bouton "ajouter un séjour" -à côté de chaque séjour se trouvent les boutons "modifier" et "supprimer"		
Résultat		
		

Documentation Technique

JACLET Hôpital : Gestion des séjours

No test	Type scénario	Description
A8	Normal	En temps qu'admin, dans le menu gérer les séjours, j'appuie sur le bouton "ajouter un séjour"
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de création de séjours -la page contient un formulaire qui sert à l'ajout des séjours avec un bouton pour valider		
Résultat		
<div>Arrivée prévu [v] [v] [v] [v] [v] Sortie prévu [v] [v] [v] [v] [v] Arrivée reel [v] [v] [v] [v] [v] Sortie reel [v] [v] [v] [v] [v] Patient [Lécuyer Amélie] Medecin [Médecin Morceau] Chambre [1] Service [Radiologie] Enregistrer le séjour</div>		

No test	Type	Description
---------	------	-------------

Documentation Technique

JACLET Hôpital : Gestion des séjours

	scénario	
A9	Normal	En temps qu'admin, dans le menu gérer les séjours, j'appuie sur le bouton "modifier" à côté d'un séjour
Résultat attendus		
-L'utilisateur est redirigé vers l'écran de création de séjours -la page contient un formulaire qui sert à l'ajout des séjours déjà rempli avec les infos actuelle du séjour avec un bouton pour valider la modification		
Résultat		
<div>Arrivée prévu Dec ▼ 9 ▼ 2021 ▼ 00 ▼ 00 ▼ Sortie prévu ▼ ▼ ▼ ▼ Arrivée réel Dec ▼ 5 ▼ 2021 ▼ 00 ▼ 00 ▼ Sortie réel Dec ▼ 20 ▼ 2021 ▼ 00 ▼ 00 ▼ Patient Dagenais Yolande ▼ Medecin Médecin Benoit ▼ Chambre 5 ▼ Service Radiologie ▼ Modification du séjour</div>		

No test	Type scénario	Description
A10	Normal	En temps qu'admin, dans le menu gérer les séjours, j'appuie sur le bouton "supprimer" à côté d'un séjour
Résultat attendus		
-la page s'actualise et le séjour à bien été supprimé		
Résultat		
le séjour est bien supprimé		

</Arthur>

<Jordan>

Documentation Technique

JACLET Hôpital : Gestion des séjours

User Story n°1 :

En tant que **personnel infirmier** je dois pouvoir **m'authentifier et voir « mon » menu**.

La fonction **index(AuthenticationUtils \$authenticationUtils)** dans LoginController.php permet de vérifier les informations de connexion et de renvoyer une erreur en cas de problèmes et renvoie l'utilisateur qui permettra la connexion

La page de connexion :

[Accueil](#)[Créer un compte](#)[Se connecter](#)

Login : Mot de Passe :

Test numéro	Type scénario	Description	Résultat si login correct	Résultat si login incorrect
J1	Connexion	<p>En tant que personnel infirmier je dois pouvoir m'authentifier et voir « mon » menu.</p> <p><i>Si le login et le mot de passe sont corrects lorsque je clique sur « me connecter » j'ai accès à toutes mes fonctions.</i></p> <p><i>Si le login et/ou le mot de passe sont incorrects alors lorsque je clique sur « me connecter » un message d'erreur s'affiche.</i></p> <p><i>Les informations de connexions doivent être également mémorisées sous forme d'historique de connexions.</i></p>	<p>Accueil Séjour</p> <p>Connecté en tant que Verriest Jordan</p>	<p>Invalid credentials.</p> <p>Login : <input type="text" value="ertzererzer"/></p>

Pour l'utilisateur n°2 :

En tant que **personnel infirmier** je peux **consulter les séjours à une date donnée**.

Documentation Technique

JACLET Hôpital : Gestion des séjours

La fonction **afficherSejourEffectif()** permet d'afficher les séjours effectifs à ce jour et elle renvoie également un formulaire où l'on peut sélectionner une date qui servira à rechercher les séjours d'une autre date.

Test numéro	Type scénario	Description	Résultat
J2	Normal	<i>Si le personnel infirmier est bien connecté alors il peut en cliquant sur « Consulter les séjours à une date donnée » accéder à la liste des séjours effectifs à la date du jour et également choisir une autre date, valider et voir les séjours effectifs à cette nouvelle date. Pour le test on clique sur consulter les séjours effectifs et ça nous affiche bien la liste des séjours ci-dessus</i>	<p><u>Séjour(s) à la date du 2021-12-05</u></p> <p>Date: j/mm/aaaa <input type="text"/></p> <p><input type="button" value="Valider la date"/></p> <ul style="list-style-type: none">• Séjour n°1 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°3 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°4 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°5 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°6 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°7 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00

La fonction **afficherSejourEffectif(Request \$request)** permet de renvoyer les séjours à une date donnée si l'on choisit une autre date dans cette fonction on utilise également une fonction repository **sejourFutur(\$user)** (voir la partie sur les fonctions repository)

Jan ▼ 1 ▼ 2016 ▼

_On choisit la date avec ce formulaire

Test numéro	Type scénario	Description	Résultat
J3	Normal	Suite de l'user story n°2 mais avec le choix d'une autre date	<p><u>Séjour(s) à la date du 2021-12-05</u></p> <p>Date: j/mm/aaaa <input type="text"/></p> <p><input type="button" value="Valider la date"/></p> <ul style="list-style-type: none">• Séjour n°1 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°3 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°4 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°5 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°6 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00 Informations du séjour• Séjour n°7 : Arrivée réelle - 03/12/2021 00:00:00 - Sortie réelle - 23/12/2021 00:00:00

La fonction **infosSejours(\$id)** permet de renvoyer les informations d'un séjour dont on connaît l'id, cet id est récupéré lorsque l'on clique sur informations du séjour :

Documentation Technique

JACLET Hôpital : Gestion des séjours

- Séjour n° 1 : Arrivée réelle - 01/01/2022 00:00:00 - Sortie réelle - 03/01/2022 00:00:00

Informations du séjour

Test numéro	Type scénario	Description	Résultat
J4	normal	Lorsque que l'on clique sur informations du séjours cela ouvre une page avec les informations du séjours	<u>Séjour n° 1</u> Patient : Lécuyer Amélie Medecin : Dr.Morceau Arrivée prévue : 14/12/2021 Arrivée réelle : 01/01/2022 Sortie prévue : 30/12/2021 12:40:00 Sortie réelle : 03/01/2022

User Story n°3 :

En tant que **personnel infirmier** je peux **consulter les séjours à venir**.

On utilise la fonction **sejourAVenir()** qui permet de renvoyer les séjours à venir et dans cette fonction on a également une fonction repository **sejourFutur(\$user)** (voir la partie sur les fonctions repository)

Test numéro	Type scénario	Description	Résultat
J5	normal	<i>Si le personnel infirmier est bien connecté alors il peut en cliquant sur « Consulter les séjours à venir » accéder à la liste des séjours à venir, il peut également en sélectionner un pour afficher les informations de celui-ci.</i>	<u>Séjour(s) à venir</u> <ul style="list-style-type: none">• <u>Séjour n°8</u> : Arrivée prévue - 04/01/2022 09:15:00 <u>Informations du séjour</u>• <u>Séjour n°9</u> : Arrivée prévue - 05/01/2022 10:15:00 <u>Informations du séjour</u>• <u>Séjour n°10</u> : Arrivée prévue - 06/01/2022 15:30:00 <u>Informations du séjour</u>

User Story n°4:

En tant que **Admin** je peux consulter l'historique de connections des utilisateurs

Documentation Technique

JACLET Hôpital : Gestion des séjours

Pour cette user story on utilise un event Listener c'est à dire une fonction qui se déclenche lorsqu'un certain événement se passe, ici l'événement est la connexion et la déconnexion, le nom de la fonction est **onSecurityAuthenticationSuccess(AuthenticationEvent \$event)** celle-ci permet d'ajouter dans la table historique, une ligne à chaque fois qu'un utilisateur se connecte.

Pour la déconnexion on utilise la fonction **onSymfonyComponentSecurityHttpEventLogoutEvent(LogoutEvent \$event)** celle-ci permet de modifier une ligne dans la table historique pour y ajouter la date et l'heure de la déconnexion au moment où l'utilisateur se déconnecte.

Test numéro	Type scénario	Description	Résultat
J6	normal	<i>Un admin doit pouvoir accéder à l'historique de connexion des utilisateurs. Il doit y avoir le login de l'utilisateur et la date et l'heure de connexion et de déconnexion.</i>	<ul style="list-style-type: none">• Connection de Cardio1 Il s'est connecté le 17/12/2021 à 17:48 et il s'est deconnecté le 17/12/2021 à 19:13• Connection de Radio1 Il s'est connecté le 30/12/2021 à 17:49 et il s'est deconnecté le 30/12/2021 à 19:27• Connection de Radio1 Il s'est connecté le 30/12/2021 à 19:28 et il s'est deconnecté le 30/12/2021 à 20:31

User Story n°5:


En tant qu'utilisateur disposant d'un compte, je peux en cliquant sur mot de passe oublié, renseigner mon adresse mail pour pouvoir obtenir un lien par mail qui me permettra de modifier mon mot de passe.

Pour cette user story on utilise les fonctions **oublPass(Request \$request , Mailer \$mailer)** celle-ci permet d'envoyer un mail avec un lien de réinitialisation de mot de passe, ce lien est généré grâce à un token avec des caractères aléatoires qui est ensuite ajouté dans la bdd. Ce lien est envoyé à l'adresse mail qui est renseignée par l'utilisateur.


Test numéro	Type scénario	Description	Résultat
-------------	---------------	-------------	----------

Documentation Technique

JACLET Hôpital : Gestion des séjours

J7	normal	Lorsque l'utilisateur clique sur mot de passe oublié il arrive sur une page ou il doit renseigner son adresse mail, puis cela envoie un mail pour pouvoir changer le mot de passe.	
----	--------	--	--

Et on utilise également la fonction **resetPassword(Request \$request, string \$token, UserPasswordHasherInterface \$userPasswordHasherInterface)**, celle-ci va de paire avec la fonction précédente car la route qui est lié à cette fonction comprend le token qui est généré et le lien qui est envoyé par mail pour la réinitialisation, cette fonction permet de récupérer le nouveau mot de passe saisi par l'utilisateur et de modifier dans la bdd l'ancien mot de passe.

Test numéro	Type scénario	Description	Résultat
J8	normal	Lorsque l'utilisateur clique sur le lien reçu par mail pour réinitialiser son mot de passe, il arrive sur une page ou il doit rentrer son nouveau mot de passe et valider, ce qui changera son mot de passe	

</Jordan>

<Elliot>

- Logout ()

Documentation Technique

JACLET Hôpital : Gestion des séjours

Permet à l'utilisateur de pouvoir se déconnecter.

Se déconnecter

Modification du security.yaml :

```
main:
  lazy: true
  provider: app_user_provider
  form_login:
    login_path: login
    check_path: login
  logout:
    path: logout
    target: /login
```

Test :

No test	Type scénario	Description
E1	Normal	En tant qu'utilisateur (administrateur ou infirmier) je peux me déconnecter de ma session
Résultat attendus		
Quand l'on appuie sur le bouton, nous sommes déconnectés et nous sommes dirigés par l'accueil principal		
Résultat		
Nous sommes bien redirigé vers l'accueil tout en n'ayant plus notre session utilisateur active		

- **afficherSejourJ ()**

Permet au personnel infirmier de pouvoir visualiser les séjours du jour affectés à leur service hospitalier.

Documentation Technique

JACLET Hôpital : Gestion des séjours

Test :

No test	Type scénario	Description							
E2	Normal	En tant que personnel infirmier, je souhaite visualiser les séjours du jours affectés à leur service.							
Résultat attendus									
Quand l'on appuie sur l'onglet Séjours du jour, le personnel infirmier peut visualiser les séjours du jour assignés à leur service médical									
Séjours du Jour									
Résultat									
<u>Séjour(s) à la date du 27/12/2021</u>									
N° Séjour	Arrivée prévue	Sortie prévue	Arrivée effective	Sortie effective	Patient	Médecin	N° chambre	Service	
<u>Séjour n° 41</u>	27/12/2021 14:20:00	19/01/2022 12:40:00	A compléter	A compléter	Picard Latimer	Morceau	Chambre n°41	Soins intensif	Modifier

- **afficherSejourEnCours ()**

Permet au personnel infirmier de pouvoir visualiser les séjours en cours affectés à leur service hospitalier.

Test :

No test	Type scénario	Description
E3	Normal	En tant que personnel infirmier, je souhaite visualiser les séjours en cours affectés à leur service.

Documentation Technique

JACLET Hôpital : Gestion des séjours

Résultat attendus

Quand l'on appuie sur l'onglet Séjours en Cours, le personnel infirmier peut visualiser les séjours en cours assignés à leur service médical

Séjours en Cours

Résultat

Séjour(s) en cours

N° Séjour	Arrivée prévue	Sortie prévue	Arrivée effective	Sortie effective	Patient	Médecin	N° chambre	Service	
<u>Séjour n° 46</u>	17/12/2021 12:40:00	26/12/2021 09:40:00	17/12/2021 21:09:05	A compléter	Bouvier Maurice	Morceau	Chambre n°46	Soins intensif	Modifier

- **modificationSejourEnCours(\$id,Request \$request) & modificationSejourJ(\$id,Request \$request)**

Permet au personnel infirmier de pouvoir modifier les dates de Séjours du jour, tant bien la date d'arrivée effective, la date de sortie effective ainsi qu'un éventuelle commentaire.

La fonction **modificationSejourJ(\$id,Request \$request)** est mise en place pour la fonction **afficherSejourJ()**.

La fonction **modificationSejourEnCours(\$id,Request \$request)** est mise en place pour la fonction **afficherSejourEnCours()**.

Test :

No test	Type scénario	Description
---------	---------------	-------------

Documentation Technique

JACLET Hôpital : Gestion des séjours

E4	Normal	En tant que personnel infirmier, je souhaite visualiser les séjours des patients afin de pouvoir valider leur arrivée et de compléter éventuellement le commentaire.
Résultat attendus		
Quand l'on appuie sur le lien Modifier, un formulaire de saisie s'affiche permettant au personnel infirmier de renseigner l'arrivée et sortie effective ainsi qu'un potentiel commentaire <u>Modifier</u>		
Résultat		
<div><p><i>Modification du séjour</i></p><p>Arrivée prévue : 27/12/2021, 14:20</p><p>Sortie prévue : 19/01/2022, 12:40</p><p>Arrivée effective : jj/mm/aaaa, --:--</p><p>Sortie effective : jj/mm/aaaa, --:--</p><p>Commentaire : <input type="text"/></p><p>Modification du séjour</p></div>		
Message d'erreurs		
Si l'on saisit une date de sortie effective inférieure à la date d'arrivée effective, une erreur se produit et ne valide pas la saisie du personnel infirmier		
Résultat		
<div><p><i>Modification du séjour</i></p><p>Arrivée prévue : 27/12/2021, 14:20</p><p>Sortie prévue : 19/01/2022, 12:40</p><p>Arrivée effective : 14/12/2021, 15:30</p><p>Sortie effective : 07/12/2021, 15:30</p><p>Commentaire : <input type="text"/></p><p>Modification du séjour</p><p><u>Vous ne pouvez pas saisir une date de sortie effective inférieure à la date d'arrivée effective. Veuillez la corriger</u></p></div>		

</Elliot>

Les repositories

Documentation Technique

JACLET Hôpital : Gestion des séjours

</Jordan>

Pour certaines fonctions on a besoin de requêtes qui ne sont pas implémentées de base dans Symfony, nous allons donc devoir les faire nous mêmes grâce au Query Builder, celui-ci nous permet de créer nos requêtes :

```
public function sejourFutur($user){
    return $this->createQueryBuilder('s')
        ->andWhere('s.arriveePrevu > :date and s.service= :user')
        ->setParameter('date',date('Y-m-d H:i:s'))
        ->setParameter('user',$user->getService()->getId())
        ->orderBy('s.arriveePrevu','ASC')
        ->getQuery()
        ->getResult()
    ;
}

public function sejourEffectif($date,$user){
    return $this->createQueryBuilder('s')
        ->andWhere('s.arriveeReel <= :date and s.sortieReel > :date and s.service= :user')
        ->setParameter('date',$date)
        ->setParameter('user',$user->getService()->getId())
        ->orderBy('s.arriveeReel','ASC')
        ->getQuery()
        ->getResult()
    ;
}
```

On met ces requêtes dans des fonctions dans lesquels on passe les attributs dont on a besoin puis ces fonctions sont exécutées dans le controller:

La fonction **sejourFutur(\$user)** permet de récupérer les séjours à venir donc ceux dont l'arrivée prévue est supérieur à la date et l'heure actuelle.

Celle-ci est utilisée dans la fonction **sejourAVenir** de sejourController.

La fonction **sejourEffectif(\$user)** permet de récupérer les séjours effectifs à une date passé en paramètre. Celle-ci est utilisée dans la fonction **afficherSejourEffectif()** de sejourController.

</Jordan>

<Elliot>

- sejourJ(\$user)

Permet de récupérer les séjours du jour actuel associé au service du personnel infirmier entré en paramètre. Celle-ci est utilisée dans la fonction **afficherSejourJ()** de sejourController.

Documentation Technique

JACLET Hôpital : Gestion des séjours

```
public function sejourJ($user){  
    return $this->createQueryBuilder('s')  
        ->andWhere('s.arriveePrevu <= :date and s.arriveePrevu >= :date2 and s.service = :user')  
        ->setParameter('date',date('Y-m-d 23:59:59'))  
        ->setParameter('date2',date('Y-m-d 00:00:00'))  
        ->setParameter('user',$user->getService()->getId())  
        ->orderBy('s.arriveePrevu','ASC')  
        ->getQuery()  
        ->getResult()  
    ;  
}
```

- sejourEnCours(\$user)

Permet de récupérer les séjours en cours, c'est-à-dire les séjours où la date de sortie effective n'est pas renseignée alors que la date d'arrivée effective l'est, associé au service du personnel infirmier entré en paramètre. Celle-ci est utilisée dans la fonction afficherSejourEnCours() de sejourController.

```
public function sejourEnCours($user){  
    return $this->createQueryBuilder('s')  
        ->andWhere('s.sortieReel IS NULL and s.arriveeReel IS NOT NULL and s.service = :user')  
        ->setParameter('user',$user->getService()->getId())  
        ->orderBy('s.arriveeReel','ASC')  
        ->getQuery()  
        ->getResult()  
    ;  
}
```

</Elliot>