

# Projet Transversal : Application de prise de rendez-vous

## Table des matières

1. Dossier d'étude.....	2
a. Spécification technique .....	2
b. Détermination des tests .....	7
c. Niveau d'habilitation .....	7
d. Trello à jour .....	8
2. Réalisation de l'application Symfony .....	8
3. Mise en production .....	8

# 1. Dossier d'étude

## a. Spécification technique

### **Bundle :**

```
{
  "type": "project",
  "license": "proprietary",
  "require": {
    "php": ">=7.2.5",
    "ext-ctype": "*",
    "ext-iconv": "*",
    "composer/package-versions-deprecated": "1.11.99.1",
    "doctrine/annotations": "^1.0",
    "doctrine/doctrine-bundle": "^2.2",
    "doctrine/doctrine-migrations-bundle": "^3.0",
    "doctrine/orm": "^2.7",
    "phpdocumentor/reflection-docblock": "^5.2",
    "sensio/framework-extra-bundle": "^5.1",
    "symfony/asset": "5.1.*",
    "symfony/console": "5.1.*",
    "symfony/dotenv": "5.1.*",
    "symfony/expression-language": "5.1.*",
    "symfony/flex": "^1.3.1",
    "symfony/form": "5.1.*",
    "symfony/framework-bundle": "5.1.*",
    "symfony/http-client": "5.1.*",
    "symfony/intl": "5.1.*",
    "symfony/mailer": "5.1.*",
    "symfony/mime": "5.1.*",
    "symfony/monolog-bundle": "^3.1",
    "symfony/notifier": "5.1.*",
    "symfony/process": "5.1.*",
    "symfony/property-access": "5.1.*",
    "symfony/property-info": "5.1.*",
    "symfony/security-bundle": "5.1.*",
    "symfony/serializer": "5.1.*",
    "symfony/string": "5.1.*",
    "symfony/translation": "5.1.*",
    "symfony/twig-bundle": "^5.1",
    "symfony/validator": "5.1.*",
    "symfony/web-link": "5.1.*",
    "symfony/yaml": "5.1.*",
    "twig/extra-bundle": "^2.12|^3.0",
    "twig/twig": "^2.12|^3.0"
  },
  "require-dev": {
    "symfony/browser-kit": "^5.1",
    "symfony/css-selector": "^5.1",
    "symfony/debug-bundle": "^5.1",
    "symfony/maker-bundle": "^1.0",
    "symfony/phpunit-bridge": "^5.1",
    "symfony/stopwatch": "^5.1",
    "symfony/var-dumper": "^5.1",
    "symfony/web-profiler-bundle": "^5.1"
  },
  "config": {
```

```

        "optimize-autoloader": true,
        "preferred-install": {
            "*": "dist"
        },
        "sort-packages": true
    },
    "autoload": {
        "psr-4": {
            "App\\": "src/"
        }
    },
    "autoload-dev": {
        "psr-4": {
            "App\\Tests\\": "tests/"
        }
    },
    "replace": {
        "paragonie/random_compat": "2.*",
        "symfony/polyfill-ctype": "*",
        "symfony/polyfill-iconv": "*",
        "symfony/polyfill-php72": "*",
        "symfony/polyfill-php71": "*",
        "symfony/polyfill-php70": "*",
        "symfony/polyfill-php56": "*"
    },
    "scripts": {
        "auto-scripts": {
            "cache:clear": "symfony-cmd",
            "assets:install %PUBLIC_DIR%": "symfony-cmd"
        },
        "post-install-cmd": [
            "@auto-scripts"
        ],
        "post-update-cmd": [
            "@auto-scripts"
        ]
    },
    "conflict": {
        "symfony/symfony": "*"
    },
    "extra": {
        "symfony": {
            "allow-contrib": false,
            "require": "5.1.*"
        }
    }
}

```

Les Bundle lors de la création du projet symfony web site skeleton nous suffisait. Nous n'avons donc rien installé de plus.

### **Cas d'utilisation :**

Cette application sera donc utilisée dans le cadre de la gestion des séjours des patients et de leur rendez-vous avec les médecins. C'est-à-dire que l'administration pourra prendre en compte l'arrivée d'un patient d'en un service, la sortie d'un service. Ils pourront aussi connaître le lit et la chambre du patient dans lequel il y était affecté. Les patients peuvent grâce à leur compte réserver un rendez-vous avec le médecin de leur choix et il sera confirmé ou non par les secrétaires médicale. Grâce a cela le client possède également un endroit ou il pourra retrouver ses rendez-vous. Quand un rendez-vous est annulé ou confirmé, le patient reçoit un mail.

Cette application sera donc pratique pour retrouver l'emplacement où se trouvait un patient à une heure précise. Cela permettra un gain de temps pour le service hospitalier pour retrouver les personnes.

### **IHM et enchainement des écrans :**

Connexion – Nous aurons donc à l'ouverture de l'application une page de connexion avec un identifiant et un mot de passe à renseigner pour se connecter. Une fois connecté à l'application, nous nous retrouverons sur l'accueil. Selon le rôle de l'utilisateur, il possèdera certaines fonctionnalités.

accueil – C'est ici que nous aurons la liste des différents patients avec la possibilité de rechercher par nom, chambre, lit... Nous pourrons aussi cliquer sur un bouton « Ajouter un patient » qui ouvrira un formulaire en popup pour ajouter un nouveau patient. Une fois le patient trouvé nous aurons quelques informations essentielles et nous pourrons cliquer sur sa fiche pour y accéder. Si l'utilisateur est un patient, il pourra réserver un rendez-vous en cliquant sur « réserver un rendez-vous » puis en sélectionnent le médecin et en choisissent la date et le jour. enfin les secrétaires peuvent changer le statut du rendez-vous en cliquant sur liste rendez-vous et sur éditer.

Fiche patient – Dans la fiche patiente nous retrouverons ses différentes informations (Nom, prénom, âge...) qui seront aussi modifiables directement dans cette page.

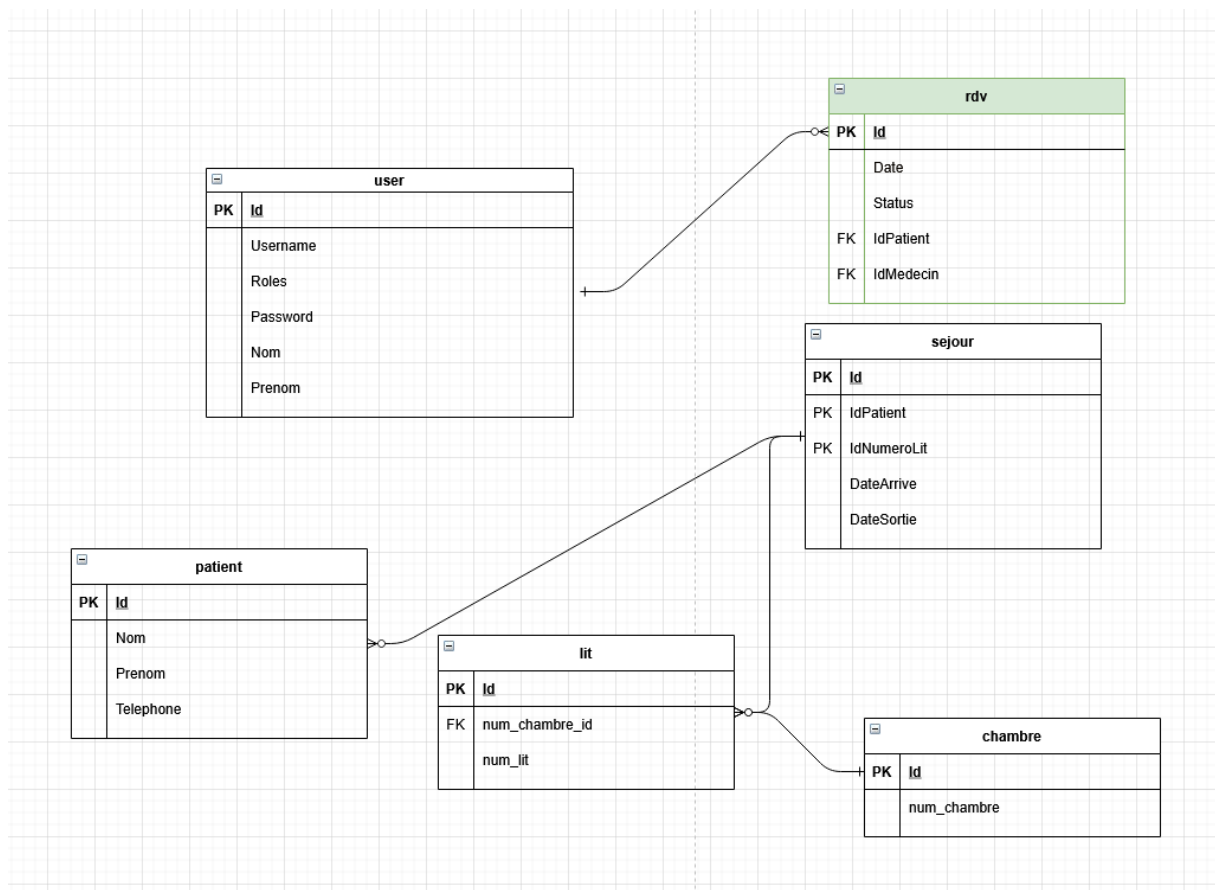
### **Les contrôleurs :**

- SecurityController
- RegistrationController
- PatientController
- LitController
- ChambreController
- SejourController
- RdvController

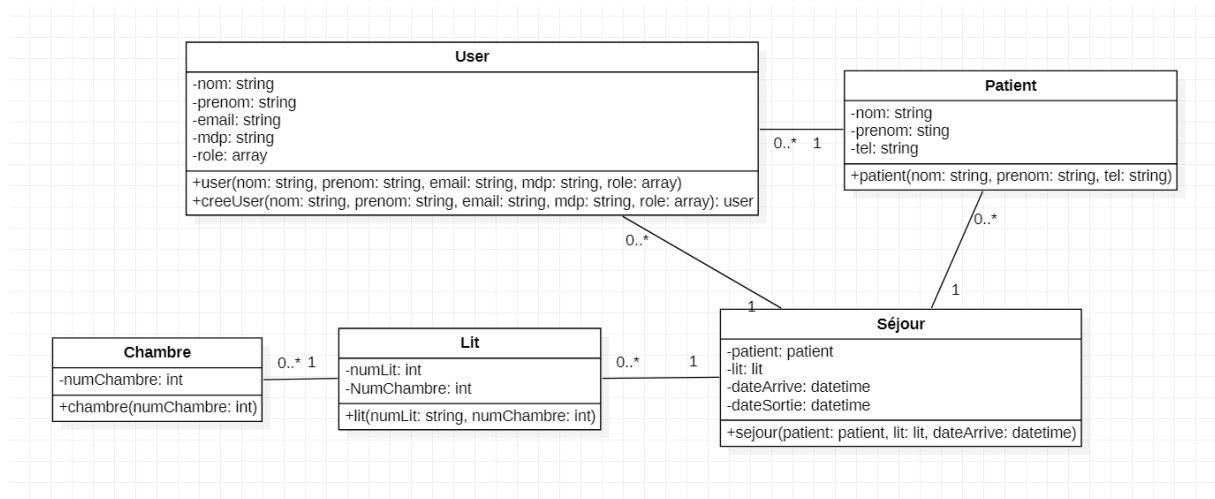
### **Les routings :**

- Login (page de connexion)
- Register (page d'inscription)
- Index (liste des patients)
- FichePatient (fiche du patient avec toutes ses informations)
- FichePatient/new (créer une nouvelle fiche)
- FichePatient/edit (éditer les informations du patient)
- Même principe pour Lit, Chambre, Sejour.
- /rdv/
- /rdv/listmodif
- /rdv/selectmedecin
- /rdv/new/{id}
- /rdv/{id}
- /modif/{id}

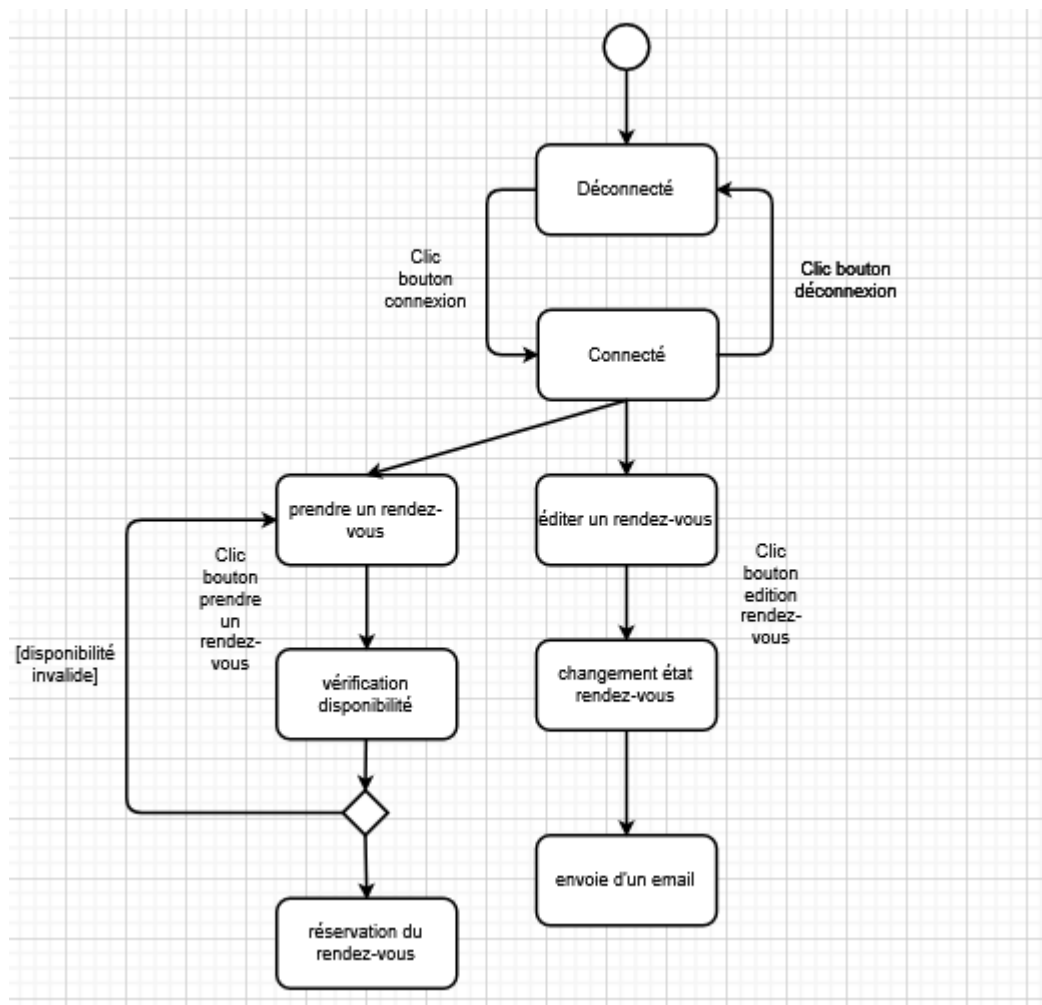
### La BDD :



### Le diagramme des classes :



### Le diagramme état-transition :



## b. Détermination des tests

### **Test d'acceptation par processus :**

Il va falloir vérifier que l'inscription se déroule correctement et qu'elle soit bien enregistré en BDD. Ensuite, nous testerons la connexion et si celle-ci se déroule correctement nous vérifierons que redirige bien vers la bonne page avec le bon contenu et les bonnes permissions que l'utilisateur a dessus.

Après nous vérifierons que les différentes pages affichent bien leurs contenus et les bonnes informations en BDD. Nous testerons notamment l'édition d'une fiche patiente, l'ajout d'une fiche patient en BDD avec des tests dans les formulaires pour être sûr que les utilisateurs ne puissent pas ajouter d'erreur (ex : un lit qui serait déjà occupé).

Une fois ces différents tests réalisés, nous regarderons que toutes les pages soient bien sécurisées est que les personnes non connectées ne puissent pas y accéder.

## c. Niveau d'habilitation

### **Liste des utilisateurs avec leur rôle.**

Administrateur – Il aura la possibilité de tout faire et en plus d'éditer les comptes créés sur le site et changer leur permission.

Personnel Infirmier – Ils auront la possibilité de voir les fiches de patients mais ne pourront pas les éditer.

Administratif – Ils pourront voir les fiches des patients et pourront les éditer ou en créer de nouvelles.

Patient – Ils pourront consulter leur rendez-vous et en demander.

Secrétaire – Ils pourront consulter les rendez-vous et modifier leurs statuts

### **Liste des écrans accessible à chaque rôle.**

Administrateur – **Tout les écrans**

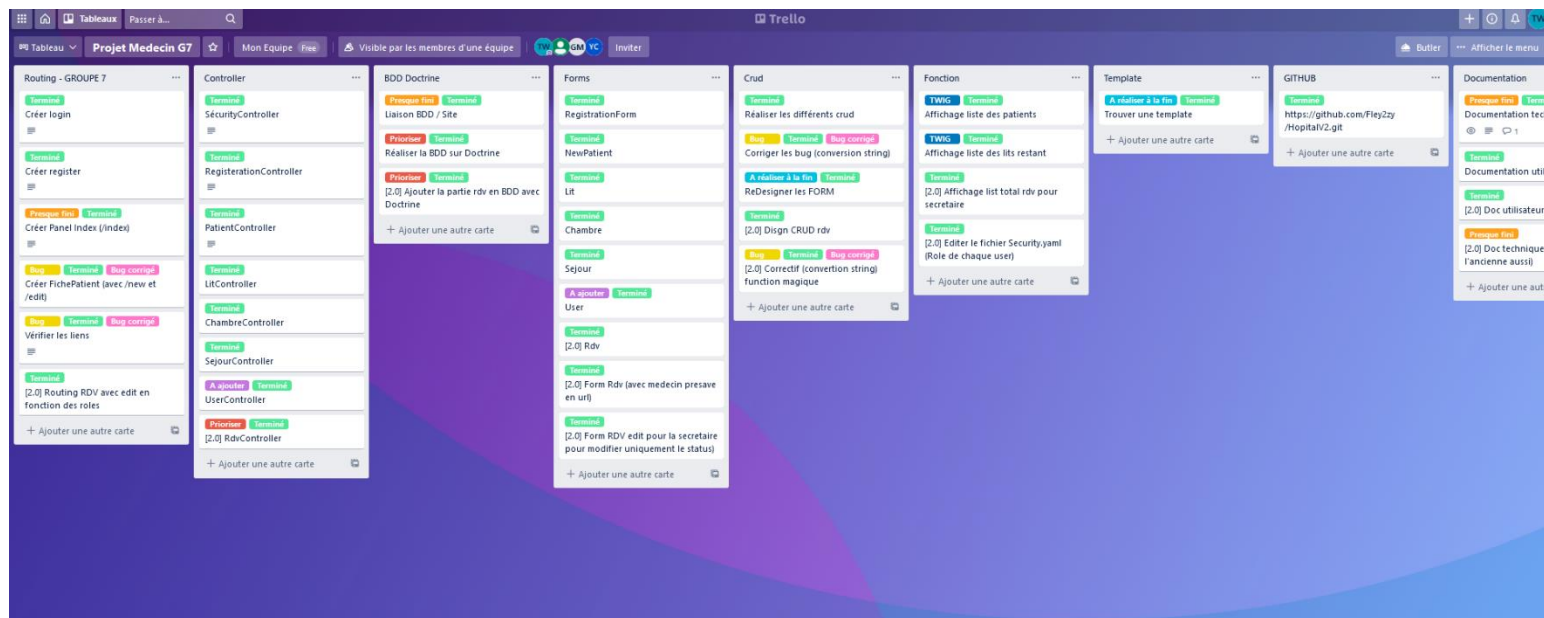
Personnel Infirmier – **accueil**

Administratif – **accueil, Fiche Client**

Patient – **accueil, sélection médecin, création rdv, liste rdv**

Secrétaire – **accueil, liste rdv, modif rdv**

## d. Trello à jour



## 2. Réalisation de l'application Symfony

Voir :

- Application
- Documentation technique
- Documentation utilisateur

## 3. Mise en production

### a. Continuité de services

Que faut-il sauvegarder ?

Nous sauvegardons la base de données afin de permettre la conservation et éviter la perte des informations des patients, mais également celle des infirmiers et différents personnels de l'établissement hospitalier.



### Comment fonctionner en mode dégradé ?

Le mode dégradé nous permet de laisser les services indispensables fonctionner lorsque l'accès au réseau n'est pas disponible. Les données sont stockées en local jusqu'à reconnexion aux réseaux. Une fois cette reconnexion effectuée, les données sont envoyées au serveur afin de mettre à jour la base de données serveur.

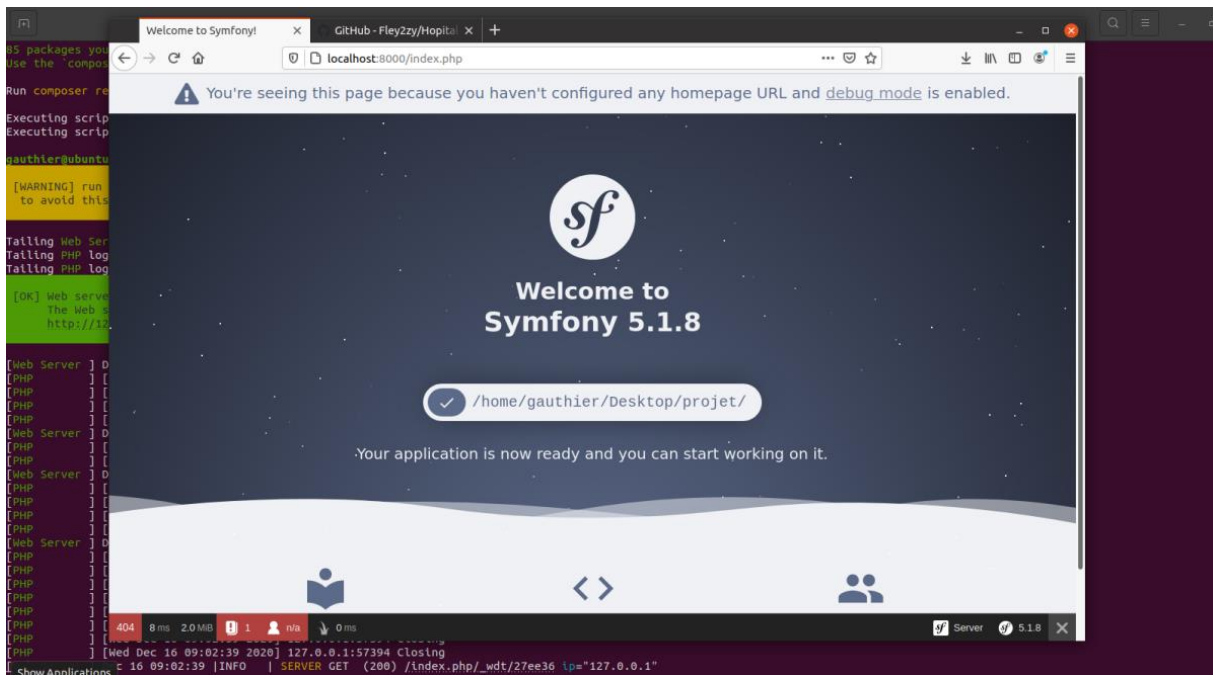
### Procédure de reprise.

En cas de panne de matériel, trouver la faille qui provoque la panne ou l'intrusion avec notamment la lecture des logs. En suite corriger la faille. Puis remettre en route le serveur et effectuer à nouveau la réalisation des tests.

## C. Déploiement de service

### Procédure d'installation à l'automatisation de l'installation.

Pour ce faire nous avons effectué la mise en place d'un serveur Apache sous Ubuntu afin de mettre l'application en place sur le serveur de préproduction.



### Mettre en exploitation de l'application.

