Documentation technique CSUNVB

Documentation pour les éventuels nouveaux membres de l'équipe de développement.

A quoi sert le site du CSU ? Qui l'utilise et pourquoi ?

Le site internet actuellement en développement sera utilisé par les ambulanciers du CSU Nord Vaudois et Broye.
Il sera utile aux ambulanciers afin de faciliter leurs tâches administratives quotidiennes qui jusqu'à aujourd'hui s'effectuent sur le papier.

Ce site fonctionnera en interne c'est à dire qu'uniquement les membres agréés auront la possibilité d'utiliser le site internet CSUNB. Par membre agréés,s'entend les secouristes.

Il y aura deux grands types d'utilisations du site. D'une part il y aura les utilisateurs simples qui ne peuvent que remplir les rapports et les signer et d'autres part il y aura les administrateurs qui sont les responsables qui créent, préparent et gèrent les rapports et la plateforme en général.

Un utilisateur peut se connecter à toutes les bases mais certaines actions ne sont possibles que si l'utilisateur est connecté à la base auquel il veut modifier des données.

Dans quel contexte (technique) fonctionne ce site ?

Le site sera hébérgé chez swisscenter pour la durée du TPI, par la suite il sera migré vers les serveurs du CSU. Une connexion internet sera donc nécessaire pour accèder au site.

La procédure de déploiement est disponible est disponible dans le dossier doc.

Celui-ci sera accessible et utilisable avec un pc ou une tablette, car le site, a terme, devra être entierement responsive.

Qu'est-ce que je dois faire pour pouvoir essayer ce site ?

Pour l'instant, une version en développement, régulièrement mise à jour, est disponible à l'adresse [csunvb.mycpnv.ch]. Cependant, il est nécessaire de posséder un identifiant pour s'y connecter. Pour le récupérer, il faut s'adresser au chef de projet, M. Carrel.

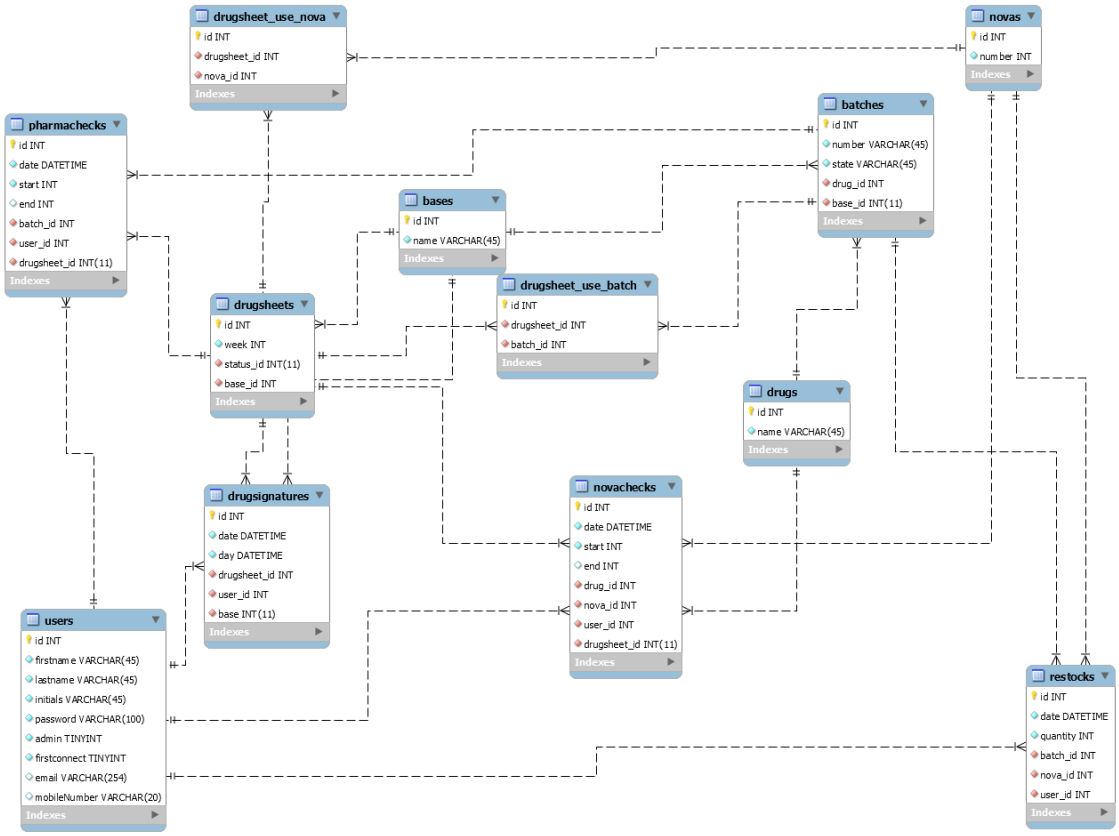
Quelles sont les données / informations que ce site manipule ?

Ce site internet est lié à une base de données qui contient toutes les données nécessaires pour la gestion administrative.

Le site est composé de 4 grandes sections :

- Les taches hebdomadaires :
  -- Permet la gestion des tâches à effectuer au cours de la semaine
- Les remises de garde :
  -- Permet la gestion des gardes à bord des ambulances (matériel, équipage, tâches, remarques pour la garde ...)
- Les stups
  -- Permet la gestion des stocks de médicaments dans les ambulances et à la base

MLD:



- L’administration
  -- Permet l’administration du site (utilisateurs, stocks, ...)

Données

Les différentes tables et données peuvent être retrouvées dans les MCD des différentes parties (DOC\SQL)

De quels composants le site est-il fait ?

Le site est basé sur la méthode MVC (Model, vue, controleur), pour chaque partie du site (garde, tâches hebdomadaire, ect) un fichier de chaque type est créé dans le dossier associé.

- Il y as un dossier Doc contenant la documentation
- Il y as un dossier public qui contient l'index, le js, le CSS et les assets, ce dossier sera utilisé comme root du serveur web afin que le reste du site ne soit pas accessible en tappant l'url
- A la racine on trouve les fichiers globalhelpers.php (les fonctions communes a toutes les parties), path.php (les chemins), policies.php (les politiques d'accès), pageList (les différentes paramètre $_GET['action'] possible pour chaque page)
- Dans le dossier Vue on retrouve le gabarit et le fichier helpers.php (fonction générale d'affichage)
- Le dossier PHPMailer packet utilisé pour envoyer de mail

Quelles technologies est-ce que je dois connaître pour pouvoir développer ce site ?

Les languages PHP et javascripts (utilisant parfois JQuery) sont indispensables pour travailler sur ce projet.

Ajax est aussi utilisé afin de réaliser des requêtes asyncrones au serveur.

Il faut aussi être à l'aise avec le html et le css pour tout ce qui est de la mise en forme

Il est aussi nécessaire de connaître mysql car il est utilisé pour interroger la base de donnée.

Le choix de ces languages paraissent évidents pour le développement d'un site internet.

Résumé:
-PHP, javascript
-MySQL
-html, css , bootstrap

Qu'est-ce que je dois installer sur mon poste de travail pour pouvoir commencer à bosser sur ce site ?

Les logiciels suivant sont ceux que nous avaons utiliser pour travailler. Des alternatives sont possible mais attention à la compatibilité.

Il faut posséder :

- Un environnement de développement : PhpStorm 2019.3.x https://www.jetbrains.com/fr-fr/phpstorm/
- PHP version 7.4.x
- Serveur de base de donnée : MySQL Community Server 8.0.23 https://dev.mysql.com/downloads/mysql/
- Client de base de donnée : MySQL Workbench (distribué avec MySQL serveur), Heidi SQL v11.2 https://www.heidisql.com/
- une adresse gmail servant pour l'envoi de mail
- drawio pour consulter ou éditer certains fichiers comme les MCD de base de donnée
- balsamiq pour créer ou éditer les maquettes

créer le fichier .const.php dans le dossier model, et y ajouter les bons paramètres en se référrant à .const.example.php dans le même dossier

créer le fichier .mailconf.php dans le dossier PHPMAILER, et y ajouter les bons paramètres en se référrant à .mailconfexemple.php dans le même dossier

désactiver la double authentification sur le compte gmail utilisé pour l'envois de mail et activer l'accès mois sécursié des applications (dans sécurité)

Exécuter les script mysql (du dossier DOC\SQL) dans l'odre suivant structure > data init puis finalement data test pour le développement ou data_prod pour la version sur swisscenter.

Utiliser le dossier public comme dossier root du serveur

Démarrer le serveur

Est-ce qu'on a des conventions de codage ?

La majorité de ce qui est de nature technique est rédigé en anglais: code, commentaires, noms de fonction, de fichiers, de variables, de base de données, de champs, ...

Le formatage du code php suit ce PhP Style Guide

Les fonctions sont précédées d'un bloc de commentaire qui a la forme suivante:

```
/**
* <nomFonction> : à quoi ça sert
* <paramètre1> : qu'est-ce qu'est + type
* <paramètre2> : qu'est-ce qu'est + type
*…
* return : ce que ça renvoie
**/
```

Les nom de fonction appelée depuis le javascript en ajax se termine par _ajax() pour les différencier

Stratégie de test

Les testes se déroulent en 3 phases différentes :

Dans un premier temps, le développeur crée les différents tests pour chaque story sur Icescrum, ceux-ci seront validés lors de la reviews de sprint, s'ils sont refusés, ils seront vérifiés à nouveau lors de la prochaine review

Dans un deuxième temps, le développeur ira sur place faire tester les fonctionnalités aux membres de Csunvb afin de savoir si elles conviennent bien à leurs attentes, dans le cadre du TPI, si des problèmes sont mis en évidence mais que ces points ne font pas partie du cahier des charges, ils seront traités à plus tard

Dans un troisième, une remise au client est faite ou celui-ci atteste de ce qui est présent sur le site un signant, des tests de régression sont aussi fait à cette occasion afin de vérifier que tout fonctionne.

Points techniques détaillés

Comment marche le système de routage entre les différentes page du site ?

Le site utilise un modèle MVC et le fichier index.php et toujours appelé et il sert pour le routage, (et aussi à définir les différents paramètres PHP et à appeler les autres fichiers utiles).
Le paramètre servant au routage est un GET "action" est utilisé dans toutes les url du site, il définit l'action que l'on cherche à effectuer, par exemple : http://localhost:8080/index.php?action=home
si l'utilisateur n'est pas connecté il aura un choix d'action possible restreint par un switch une fonction de cette action,
sinon s'il est connecté et qu'une fonction d'un des contrôleur possède le même nom que le paramètre GET "action", celle-ci sera appelée,
sinon la fonction home() sera appelée par défault

Comment marche le système de policies ?

Avec le système de routage ci-dessus, si un utilisateur connait le nom de l'action, il pourra entrer l'url et effectuer l'action même si son rôle ne devrait pas le lui permettre,
C'est pourquoi, il faut vérifier que l'utilisateur a bien les droits d'effectuer cette action :

Le fichier policies.php liste les actions possible et leur niveau de droit nécéssaire

Les actions possibles par tout le monde (niveau 0) n'ont pas besoin d'être renseignées

Lorsqu'on tente d'accèder un une page, la fonction ican("nom de l'action") est appelée (dans index.php) afin de savoir si l'utilisateur possède un niveau de privillège suffisant, si ce n'est pas le cas il est redirigé vers la page d'accueil.

Qu'est-ce que c'est que ce champ 'slug' dans la table 'status' ?

Un slug est un identifiant sous contrôle du code de l'application. Il se situe entre l'id de base de donnée dont on ne peut jamais présumer de la valeur dans le code et la valeur affichée. Exemple: status 'Ouvert', qui a un slug 'open' et un id 2. Si je veux sélectionner les rapports ouverts, je fait un select « where slug='open' » . Si l'id de l'état 'open' est différent dans une autre db => ça marche, si un jour je veux changer le terme visible de « Ouvert » en « Actif » par exemple, je peux le faire en ne changeant que des données.

Voir cette référence (parmi tant d'autres)

Qu'est-ce que c'est que le 'flashmessage' ?

Le flash message est un message en haut de la page utilisé pour indiquer à l'utilisateur si une action s'est effectuée correctement ou si il y a une erreur.

Celui-ci peut être affiché de deux manières différentes :

- En php, avec la fonction : setFlashMessage("mon message") lors du chargement d'une nouvelle page
- En javascript, avec la fonction : flashMessage("mon message") si la page n'est pas rechargée

Comment marchent les commentaires épinglés ?

Le système d'épinglage permet un commentaire d'un rapport de garde, d'être affiché sur les prochains rapports.
Par exemple : "Les piles de la radio sont plates" pour l'indiquer au prochain groupes jusqu'à ce que le problème soit réglé.

Au survol d'un commentaire, un bouton pour épingler gris est visible, lors que le commentaire est épinglé il sera visible sur les prochains rapports et aura champs champ carryOn à 1 dans la base de donnée.
Pour désactiver l'épinglage cliquer à nouveau sur l'icone "épinglé" cela aura pour conséquence de remplir le champ "endOfCarryOn" avec la date du rapport sur lequel la personne se trouve, à partir de ce moment, le message ne sera plus affiché sur les prochains rapports.

La selection des commentaires de la base de donnée à affiché sur le rapport se fait deux manières :

- Si le commentaire est lié au rapport, il est affiché ainsi que l'heure et la personne qui l'a écrit
- Si le commentaire n'est pas lié au rapport mais qu'il est sur la même base et que :

  (date du commentaire < date du rapport sélectionné) et que (date du rapport sélectionné < endOfCarryOn du commentaire ou endOfCarryOn est null)
  Alors celui-ci est affiché en tant que commentaire épinglé et la date de celui-ci est aussi affichée

Comment marchent les pop-up de confirmation ?

En bas du gabari est présent un squelette de pop-up de confirmation,
ensuite il peut être affiché est adapté comme on le souhaite dans modal.js grâce aux functions : showModal, setTitleModal, setBodyModal, addBodyModal, setSubmitModal

Comment est clôturé un rapport ?

Actuellement le principe n'est pas encore appliqué rapports de stupéfiants

Pour les parties tâches hebdomadaire et garde :

Lors du clic sur le boutons fermer, une requête AJAX est envoyée au serveur qui retourne le nombre de tâches manquante

- Confirmation simple si le rapport est complet
- Confirmation avec avertissement si le rapport n'est pas complet

Pour les rapports clôturés

- Les tâches non-validées sont affichée en rouge
- Une icône /!\ indique l'erreur dans la liste des rapports, au survol de l'icone est affiché le nombre de tâches non effectuées

De plus les initiales de la personnes ayant clôturé un rapport seront affichées en haut de celui-ci

Navbar

La Navbar est présente sur toutes les pages tant que l'utilisateur est connecté

Pour faire correspondre l'action (paramètre GET) à l'onglet de la Navbar, le fichier pageList.php est utilisé
Il contient une liste des actions possible pour chaque onglet de la navbar comme par exemple :
 $shiftPages = array("shiftList","shiftShow","shiftLog");

Pour chaque onglet de la navbar si le paramètre $_Get['action'] est présent dans la liste, celui sera mis en évidence

Edition Rapport Todo

Pour éditer un rapport une icone crayon est présent, et pour sortir de ce mode il faut fermer le formulaire d'édition

Les listes déroulantes "jour" et "créneau" possèdent un champ vide avec comme paramètre "selected disabled hidden" afin d'avoir une option par défaut non-sélectionnable

Si les champs jour et créneau sont renseignés, deux champs permettent d'ajouter des actions existante ou d'en crée une nouvelle.

Envois de formulaire en javascript

La fonction javascript post() permet de créer et d'envoyer directement un formulaire en post sans avoir à le créer dans le code html

Il faut lui fournir les paramètres path et un tableau des paramètres comme ceci :

```
const param = { param1: 1, param2: 2 };
post("?action=exemple", param )
```

Comment fonctionne le formulaire de modification des données d'un rapport de stupéfiant ?

Les données sont envoyées par méthode POST comme nous le ferions en général avec des formulaires HTML.

La subtilité de ce formulaire est que tous les champs qui ont des données à envoyer ont un attribut "name" qui est un index de tableau. Cela permet d'avoir du côté du Controller PHP une variable qui est un array que l'on peut parser efficacement avec un foreach.

Comment fonctionnent les boutons qui sont déjà à l'intérieur d'un autre formulaire dans la partie des rapports de stupéfiants ?

La problématique est qu'il ne faut pas faire de formulaire dans un autre formulaire. Donc la solution suivante a été mise en place :
Les boutons appellent une fonction javascript en lui donnant les données en paramètres et la fonction créent un formulaire caché qui est ajouté à la fin de la page et envoyés.

Comment marchent la validation des équipes d'un rapport de garde avant son activation

Pour qu'un rapport puisse être activé :

- aucun autre rapport doit être ouvert sur cette base
- Tous les champs des équipes suivent être renseignés

Afin d'enregister les données des équipes de garde sans ajouter de latence, chaque fois qu'un champ est modifié, l'information est envoyée au serveur par une requête Ajax.

Ensuite si la modification s'est effectuée normalement, une seconde requête demande si tous les champs sont indiqué, si c'est le cas, la page est finalement rechargé et le boutons activer devient actif s'il le doit

Comment marchent les calendriers pour le planning des secouristes ou des novas ?

Importation du planning des secouristes

Comment marchent la gestion de droits sur un rapport de garde

Les droits sur un rapport de garde se divise en 4 :

- $enableDataUpdate : définit si la personne a le droit de remplir les équipes pour le rapport de garde
- $enableFilling : définit si l'utilisateur peut remplir le rapport, valider des actions, etc
- $enableStateChange : définit si l'utilisateur a le droit de changer l'état du rapport (ex. passer de ouvert à fermé)
- $enableStructureChange : définit si l'utilisateur a le droit de modifier la structure (ex. ajouter une action à effectuer sur le rapport)