


Gestion de prêt de matériel du laboratoire d'électronique



Gruber, Adam
Coteaux des ifs 43
1400, Cheseaux-Noréaz
adam.gruber@cpnv.ch
SI-C4r
07.02.2023

Table des matières

1	Analyse préliminaire	4
1.1	Introduction	4
1.2	Organisation	4
1.3	Objectifs.....	4
1.4	Planification initiale	5
1.5	Structure du dossier.....	6
2	Analyse.....	6
2.1	Cahier des charges détaillé	8
2.1.1	Définition du contenu et des fonctionnalités	8
2.1.2	Situation actuelle	8
2.1.3	Utilisateurs cibles.....	8
2.1.4	Détermination des intervenants	8
2.1.5	Présentation du budget disponible.....	8
2.2	Analyse comparative des solutions envisageables.....	9
2.2.1	Contraintes liées à la situation actuelle.....	9
2.2.2	Présentation et comparaison des solutions matérielles	9
2.2.3	Présentation et comparaison des solutions logicielles.....	9
2.2.4	Aspects humains des différentes solutions	9
2.2.5	Justification de la solution retenue.....	10
2.3	Etude de faisabilité	10
2.4	Planification	10
2.5	Stratégie de test.....	10
2.6	Historique	11
3	Conception	11
3.1	Plans topologiques	11
3.1.1	Topologie physique.....	11
3.1.2	Topologie logique	11
3.2	Architecture logique.....	12
3.2.1	Conventions de dénomination et d'adressage	12
3.2.2	Structures logiques et arborescences.....	12
3.3	Implémentation matérielle.....	12
3.3.1	Caractéristiques techniques détaillées	12
3.3.2	Configurations spécifiques.....	12
3.4	Implémentation logicielle	12
3.4.1	Types de logiciels et de licences	13
3.4.2	Mise en place et options spéciales	13
3.5	Mise en place de la Sécurité.....	13
3.5.1	Sécurité appliquée aux utilisateurs	13
3.5.2	Sécurité dans la gestion de serveurs et d'un parc de machines	14
3.5.3	Sécurisation de la liaison au monde extérieur	14
3.6	Historique	14

4	Réalisation et mise en service	15
4.1	Description des tâches effectuées	19
4.2	Modifications apportées par rapport à la conception	20
4.3	Description des tests effectués	20
4.4	Erreurs restantes	20
4.5	Rapport de mise en service	20
4.6	Liste des documents fournis et dossier d'archivage	21
5	Conclusions	21
6	Annexes.....	22
6.1	Sources – Bibliographie	22
6.2	Glossaire	22
6.3	Journal de bord de chaque participant.....	22
6.4	Manuel d'Installation	22
6.5	Manuel d'Utilisation.....	22
6.6	Présentation de fin de projet (6 diapos par page) - pas obligatoire si la présentation est à venir	22
6.7	Supports d'archivage du projet	22

NOTE à L'INTENTION DES UTILISATEURS DE CE CANEVAS:

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

N'oubliez pas d'adapter les entête et pieds de page.

1 Analyse préliminaire

1.1 Introduction

Le but de ce projet est de réaliser une application web qui permettra de gérer les prêts matériels du laboratoire d'électronique de la filière informatique du CPNV à Ste-Croix

Ce chapitre décrit brièvement le projet, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

1.2 Organisation

Ce chapitre décrit l'organisation du projet :

Eleve 1 : Nom, prénom, e-mail et téléphone

Eleve 2 : Nom, prénom, e-mail et téléphone

Responsable de projet (enseignant : Nom, prénom, e-mail et téléphone

Expert 1 : Nom, prénom, e-mail et téléphone

Expert 2 : Nom, prénom, e-mail et téléphone

Ce chapitre peut également montrer la répartition des tâches dans le projet.

Exemple :

	<i>Eleve 1</i>	<i>Eleve 2</i>
<i>Tâche 1</i>	X	
<i>Tâche 2</i>		X
<i>...</i>		
<i>Tâche n</i>		X

1.3 Objectifs

Le but de ce projet est de réaliser une application web qui permettra de gérer les prêts matériels du laboratoire d'électronique de la filière informatique du CPNV à Ste-Croix

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

1.4 Planification initiale

La planification initiale de mon projet, faite avec une Template Excel.
Il est important de préciser que ma planification initiale est que sur 7 semaines, tout simplement car j'ai reçu mon cahier des charges une semaine plus tard.

Dans mon cahier des charges il est précisé que mon Pré-TPI commence le 7 février à 13h30 jusqu'au 29 Mars à 16h05

Projet		Planification									
gestion de prêt de matériel du laboratoire d'électronique			07.02.23	14.02.23	21.02.23	28.02.23	07.03.23	14.03.23	21.03.23	28.03.23	04.04.23
Total											
Prévu	61 h 30		9 h 00	9 h 00	8 h 00	9 h 00	7 h 00	9 h 00	10 h 30		
NI	5 h 45		5 h 45								
SEM			1	2	3	4	5	6	7	8	9
1 Analyse											12 h 00
	11 Planification	Prévu	3 h 00								5 h 45
		NI	2 h 15								3 h 00
	12 Reflexion	Prévu		1 h 00	1 h 00	1 h 00	1 h 00	2 h 00			6 h 00
		NI	2 h 00								2 h 00
	13 Prise en main de Yii	Prévu	3 h 00								3 h 00
		NI	1 h 30								1 h 30
	14 -	Prévu									
		NI									
	15 -	Prévu									
		NI									
	16 -	Prévu									
		NI									
2 Implémentation Front end											9 h 00
	21 Mise en place de la template	Prévu	3 h 00								
		NI									
	22 Adaption de la template	Prévu		1 h 00	1 h 00	2 h 00	1 h 00	2 h 00	2 h 00		9 h 00
		NI									
	23 -	Prévu									
		NI									
	24 -	Prévu									
		NI									
	25 -	Prévu									
		NI									
	26 -	Prévu									
		NI									
3 Implémentation Back end											16 h 00
	31 Creation de la base de donnée	Prévu		3 h 00							3 h 00
		NI									
	32 Consultation du matériel	Prévu			4 h 00						4 h 00
		NI									
	33 Inscription / connexion d'un utilisateur	Prévu				3 h 00					3 h 00
		NI									
	34 La recherche de matériel	Prévu					3 h 00				3 h 00
		NI									
	35 Gestion des erreurs	Prévu						1 h 00	2 h 00		3 h 00
		NI									
	36 -	Prévu									
		NI									
4 Tests											8 h 00
	41 Creation des tests	Prévu		2 h 00		2 h 00	1 h 00	3 h 00			8 h 00
		NI									
	42 -	Prévu									
		NI									
	43 -	Prévu									
		NI									
	44 -	Prévu									
		NI									
	45 -	Prévu									
		NI									
	46 -	Prévu									
		NI									
5 Documentation											13 h 30
	51 Documentation	Prévu		2 h 00	2 h 00	1 h 00	1 h 00	1 h 00	2 h 00		9 h 00
		NI									
	52 Préparation présentation	Prévu							4 h 30		4 h 30
		NI									
	53 -	Prévu									
		NI									
	54 -	Prévu									
		NI									
	55 -	Prévu									
		NI									
	56 -	Prévu									
		NI									

Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées et inclure au besoin une rubrique "si le temps le permet". Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme

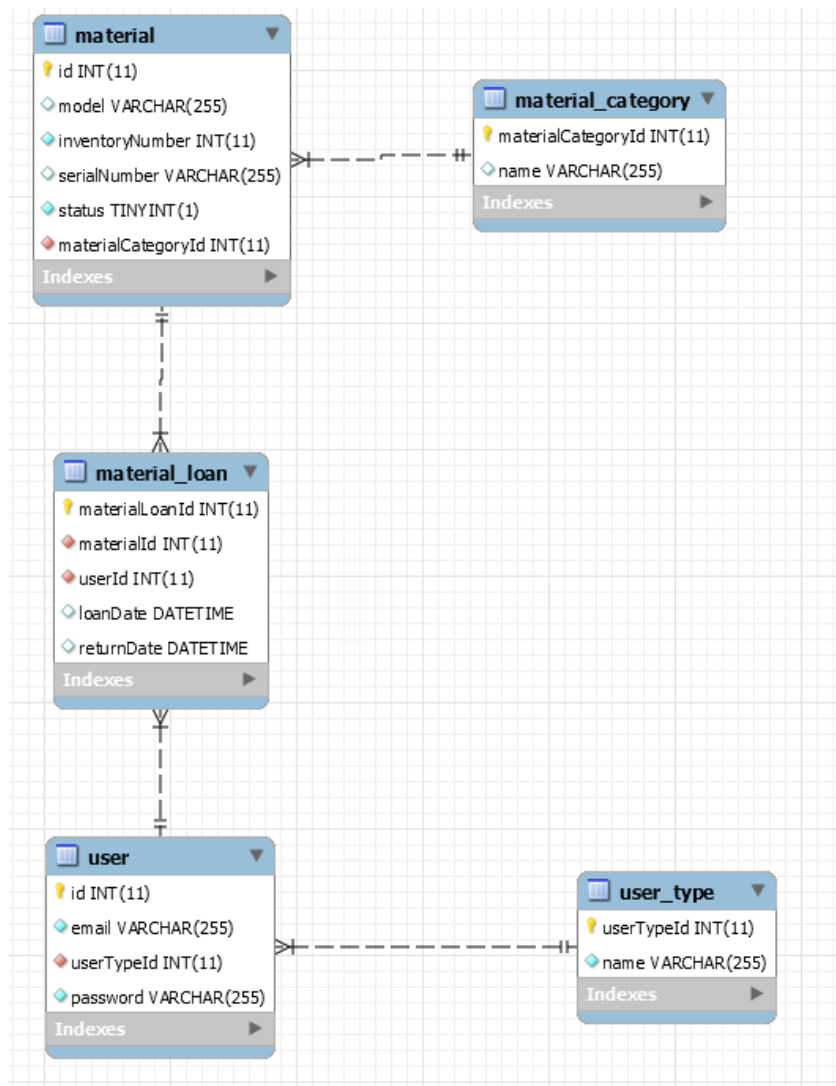
de Gantt et/ou de PERT (l'utilisation d'un logiciel de planification est recommandée).

1.5 Structure du dossier

Cette partie peut présenter la façon dont est organisé le document. En quelques lignes, on décrit le contenu des différents chapitres, ce qui permet d'avoir une vue d'ensemble.

2 Analyse

MLD



On peut directement voir que les consommables ne sont pas dans la base de donnée, c'est tout simplement parce que moi et Mr.Saison avons décidé que ce ne sera pas une priorité.

On gardera donc 3 tables principales en compte : Les utilisateur, le matériel et les emprunts.

Petite précision au niveau du type « TINYINT » qui se trouve à coté champ « status » de la table « material ». « TINYINT » représente tout simplement un boolean qui peut donc être vrai ou faux. Dans notre situation, le status sert à savoir si le matériel est disponible ou pas.

Méthode de gestion de projet utilisé

a

En partant du principe que mon chef de projet est mon client (Mr. Yvan Saison), j'ai utilisé la méthode agile.

La méthode agile est basée sur plusieurs points : rapidité, communication, retours.

1.

Le product owner (client) fait une liste de ce qu'il souhaite avec des priorités inscrites

Le cahier des charges représente cette partie, il donne une vue d'ensemble de ce que souhaite le client avec les priorités et les points importants à respecter.

2.

Normalement l'équipe informatique prends les envies du client et en fait des sprints. Les sprints durent généralement 1 à 2 semaines.

Ce point est censé être fait grâce à ce qu'on appelle un logiciel de gestion de projet, comme par exemple iceScrum. Malheureusement cela n'a pas été exécuté donc les sprints n'ont pas réellement été créés.

3.

Lorsqu'un sprint est terminé, le client fait un review et donne son avis sur les améliorations possibles. Et c'est à ce moment que la boucle recommence avec un autre sprint, ou l'amélioration du sprint actuel.

Pour ma part, Mr Yvan Saison passait tout les Mercredi voir l'avancement de mon travail, il me disait ce qui allait ou pas. Grâce à ce contact direct avec mon client, il a pu voir la direction que je prends et donc il a pu me recadrer sur un projet qui atteint ses besoins.

Et c'est pour cela qu'on dit que la méthode agile est très bonne. La satisfaction du client est toujours haute car ils ne doivent pas attendre des mois ou même années pour avoir un programme qui finalement n'attendaient pas. Ce système permet d'éviter ce genre de situation et s'adapte selon les besoins du client.

Outil de Versionning utilisé

J'ai utilisé Git durant mon projet, ça m'a permis de garder une trace de chaque modification du code effectuée. De plus si une erreur a été effectuée, j'ai le privilège de revenir en arrière pour résoudre le problème. Vous pouvez trouver mon projet GitHub avec ce lien : <https://github.com/GruberAdam/gestion-materiel-laboratoire-electronique>

Pour manager Git j'ai utilisé un outil connu sous le nom de Git Bash. Git Bash permet d'exécuter des lignes de commandes sur système d'exploitation. Git Bash est bien évidemment optionnel, il n'est pas nécessaire de l'utiliser afin de travailler avec Git. J'ai décidé de l'utiliser parce que il apporte des couleurs dans le terminal et donc est plus explicite à mon goût.

2.1 Cahier des charges détaillé

2.1.1 Définition du contenu et des fonctionnalités

- *Détails complets des objectifs et des besoins du client.*

2.1.2 Situation actuelle

- *Représentation physique du réseau et des principaux éléments actifs*
- *Représentation logique du rôle des différents serveurs*
- *Inventaire du parc matériel et logiciel*
- *Services offerts par les différents serveurs en fonction*

2.1.3 Utilisateurs cibles

Liste des personnes qui seront concernées par le travail :

- *Types d'utilisateurs ou fonctions dans l'entreprise (utilisateurs, administrateurs)*
- *Formation ou niveau de compétences informatiques*
- *Définition des besoins suivant le type d'utilisateurs*

2.1.4 Détermination des intervenants

Listes des personnes qui peuvent intervenir dans le projet :

- *Fournisseurs de matériels et logiciels*
- *Service informatique interne, entreprise de services externe*
- *Coordonnées en cas d'urgence*

2.1.5 Présentation du budget disponible

Si le client a déjà formulé des contraintes ou limites budgétaires, on les présente.

Analyse comparative des solutions envisageables

Pour un même problème (création ou migration d'un réseau local, connexion d'une entreprise à Internet, implémentation de services serveurs mail, ftp ou autres) plusieurs solutions techniques (matérielles et logicielles) et humaines sont envisageables. Ce chapitre expose ces différentes possibilités, leurs avantages, leurs inconvénients en fonction des besoins du client. Ce chapitre permettrait par exemple de répondre à un appel d'offres.

2.1.6 Contraintes liées à la situation actuelle

- *Etat du parc de machines et mise à jour matérielles indispensables*
- *Versions logicielles et problèmes de compatibilité (migration, etc...)*
- *Disponibilités en ressources humaines*

2.1.7 Présentation et comparaison des solutions matérielles

Cette partie concerne l'achat de serveurs, d'un parc de machines, de matériel réseau (routeurs, switchs, câblage), d'équipements de bureaux, etc ...

- *Résumé des caractéristiques techniques (fiabilité, performances)*
- *Fournisseurs, liens internet*
- *Prix disponibles*

2.1.8 Présentation et comparaison des solutions logicielles

- *Possibilités techniques offertes par les logiciels*
- *Version, compatibilité et évolutivité*
- *Fournisseurs, liens internet*
- *Prix disponibles*

2.1.9 Aspects humains des différentes solutions

- *Aspects ergonomiques (confort de travail, facilité d'administration, etc ...)*
- *Besoins en formation des administrateurs et des utilisateurs*
- *Coûts humains liés à la mise en place, la maintenance du système (prix à l'heure, forfait, etc ...), à la formation des utilisateurs et des administrateurs (formation en entreprise privée, coûts des certifications, etc ...)*

2.1.10 Justification de la solution retenue

On présente surtout en quoi cette solution est la plus avantageuse parmi celles présentées pour les critères matériels, logiciels et humains. On peut inclure ici un tableau récapitulatif des différents points de comparaison (performances, prix, SAV, fournisseurs, références internet). Suivant l'étude, un bilan chiffré synthétisant les différents coûts pourra être intégré. Les détails techniques de la solution retenue pourront ensuite être abordés dans la Conception.

2.2 Etude de faisabilité

Détailler les 4 aspects de l'étude de faisabilité:

- *risques techniques (complexité, manque de compétences, ...).*
- *risques concernant le planning (délai de livraison du matériel et des licences)*
- *risques concernant les ressources humaines (disponibilité des administrateurs actuels, du personnel, des intervenants divers)*
- *risques concernant le budget.*

2.3 Planification

Révision de la planification initiale (Gantt et/ou PERT) du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*
- *rubrique "si le temps le permet"*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

2.4 Stratégie de test

Utilisateur qui souhaite consulter le matériel :

Lorsque l'utilisateur clique sur matériel, une fenêtre s'ouvre avec différentes informations,

Décrire la stratégie globale de test :

- *types de tests et ordre dans lequel ils seront effectués.*

- *matériels ou licences d'évaluation à acquérir pour les tests uniquement (machine portable, licence de test ...)*
- *contraintes diverses (utilisateurs connectés, serveurs en production, etc...)*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?)*
- *les testeurs extérieurs éventuels.*

2.5 Historique

- *Pour le cahier des charges: lister toutes les modifications demandées par le client, et aussi celles décidées pour d'autres raisons*
- *Pour le budget : comparaison entre le budget initial et le budget actuel*

Conception

Après une discussion avec monsieur saison, on a décidé d'abandonner l'idée de travailler avec les consommable. Tout simplement parce que le travail avec les consommables et le matériel est identique, donc je peux aller plus loin en me concentrant sur une seule table. Si le temps le permet j'ajouterai les consommables après coup.

De plus les consommables demandaient d'avoir un model ce qui ne fait pas de sens. On a donc décidé de changer le nom du champ avec « description ».

On explique en détails les choix présentés dans la phase d'analyse, concernant notamment les aspects matériels et logiciels. On présente la façon dont on va structurer le réseau de manière physique et logique sans toutefois rentrer dans la mise en place qui sera abordée dans le chapitre Réalisation.

2.6 Plans topologiques

2.6.1 Topologie physique

- *Plans des locaux (salle machines, local serveur ...) si possible à l'échelle*
- *Emplacement des éléments actifs (armoires de brassage, switches...) et représentation du câblage*

2.6.2 Topologie logique

- *Représentation d'un réseau local*
- *Schématisation des serveurs (service mail, web ...) et du parc de machines*
- *Représentation de la connectique et de la liaison vers le monde extérieur*
- *Description succincte des adresses réseau, protocoles et ports*
- *Plans d'adressage IP*

2.7 Architecture logique

On détaille tout ce qui concerne l'implémentation logique du réseau.

2.7.1 Conventions de dénomination et d'adressage

- *Types d'utilisateurs et groupes de sécurité*
- *Adresses et listes de mails*
- *Adressae et dénomination des serveurs, machines et éléments de connectique*

2.7.2 Structures logiques et arborescences

- *Arborescence de l'annuaire (LDAP, Active Directory, X500...)*
- *Structures et imbrication des groupes de sécurité*
- *Structure et imbrication des listes de mails*
- *Structure des partages et des queues d'impression*

Implémentation matérielle

2.7.3 Caractéristiques techniques détaillées

Le matériel a été rapidement présenté dans l'Analyse pour justifier le choix de la solution retenue. Si nécessaire, on détaille ici les performances, capacités du matériel choisi.

2.7.4 Configurations spécifiques

Si notre projet nécessite des configurations ou mises en place spécifiques pour le matériel (configuration des armoires, mise en place du câblage, installation des postes).

2.8 Implémentation logicielle

Les logiciels qui ont été utilisés :

Visual Studio Code (Version: 1.74.3 user setup) :

Environnement de travail utilisé pour programmer l'application Sploks.

PHP

Langage de programmation utilisé pour mon projet

Yii :

Un Framework PHP qui permet d'avoir une Template d'un MVC. De plus, ce Framework m'évite d'écrire un CRUD complet et donc me fait gagner du temps.

Git Bash :

Permet d'effectuer le versioning du projet et aussi de mettre à jour le repository sur GitHub (LIEN DU REPO).

Suite Office :

Utilisé pour cette documentation, et pour le journal de travail.

MySQL Workbench :

Utilisé pour pouvoir consulter la base données et exécuter des requêtes.

Présentation des logiciels utilisés. On peut inclure certains dumps d'écrans.

2.8.1 Types de logiciels et de licences

- *Description et caractéristiques du logiciel (administration, serveur, client...)*
- *Licence d'évaluation, licence de site, logiciel libre, mise à jour, etc ...*
- *Prix et fournisseurs*

2.8.2 Mise en place et options spéciales

- *On précise le type de serveur sur lequel le logiciel sera installé*
- *On expose les configurations spéciales et les possibilités non standards que l'on va utiliser.*
- *On liste les patchs et les mises à jour à faire*

2.9 Mise en place de la Sécurité

On détaille tout ce qui concerne la sécurité de l'installation :

2.9.1 Sécurité appliquée aux utilisateurs

- *Définition et utilisation des mots de passe*
- *Définition des droits d'accès (services, dossiers partagés, imprimantes ...)*
- *Stratégies de sécurité*

2.9.2 Sécurité dans la gestion de serveurs et d'un parc de machines

- *Mise en place d'un antivirus, scanner de mail, etc ...*
- *Configuration d'un système de sauvegarde / restauration / scripts backup*
- *Ecriture de scripts de login utilisateur / montage automatisé des disques*
- *Mise en place de tâches automatisées diverses pour l'administration*

2.9.3 Sécurisation de la liaison au monde extérieur

- *Mise en place d'un routeur / firewall / proxy*
- *Règles de filtrage*
- *Installation de NIDS, HIDS*

2.10 Historique

Si la conception du projet a dû être modifiée plusieurs fois, ou de manière significative, expliquez ces changements et leurs causes.

3 Réalisation et mise en service

Mise en place de git

Pour mettre en place git, ce n'est pas très compliqué. Il suffit d'installer git sur votre machine (<https://gitforwindows.org/>).

Je recommande d'installer git bash afin de travailler avec des couleurs.

Une fois git installé, aller sur GitHub et créer un repository.

Cela fait, il suffit de cloner le repository GitHub sur git bash avec la commande ci-dessous :

git clone {Le lien de votre repository GitHub}

```
Adam.GRUBER@SC-C236-PC12 MINGW64 /c/Program Files
$ git clone https://github.com/GruberAdam/gestion-materiel-laboratoire-electronique
Cloning into 'gestion-materiel-laboratoire-electronique'...
remote: Enumerating objects: 192, done.
remote: Counting objects: 100% (192/192), done.
remote: Compressing objects: 100% (129/129), done.
remote: Total 192 (delta 52), reused 181 (delta 41), pack-reused 0
Receiving objects: 100% (192/192), 2.87 MiB | 7.80 MiB/s, done.
Resolving deltas: 100% (52/52), done.

Adam.GRUBER@SC-C236-PC12 MINGW64 /c/Program Files
$ |
```

Git est à présent mis en place.

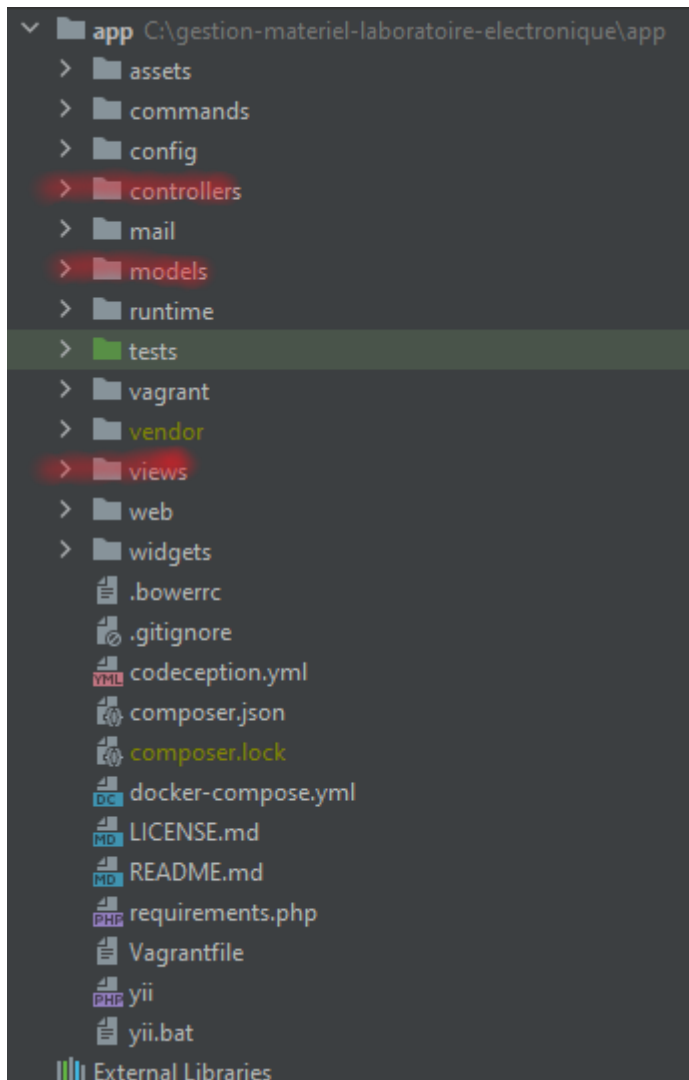
Mise en place du framework yii2

Pour mettre en place yii2, vous devez premièrement disposer de composer.

Composer est tout simplement un outil qui vous permet de manager les dépendances de votre projet.

Une fois que composer est installé, je vous invite à suivre les instructions du lien Ci-dessous afin de finir l'installation de yii2 :
<https://www.yiiframework.com/doc/guide/2.0/fr/start-installation>

Une fois yii2 mis en place, l'architecture de votre site devrait ressembler à ça :

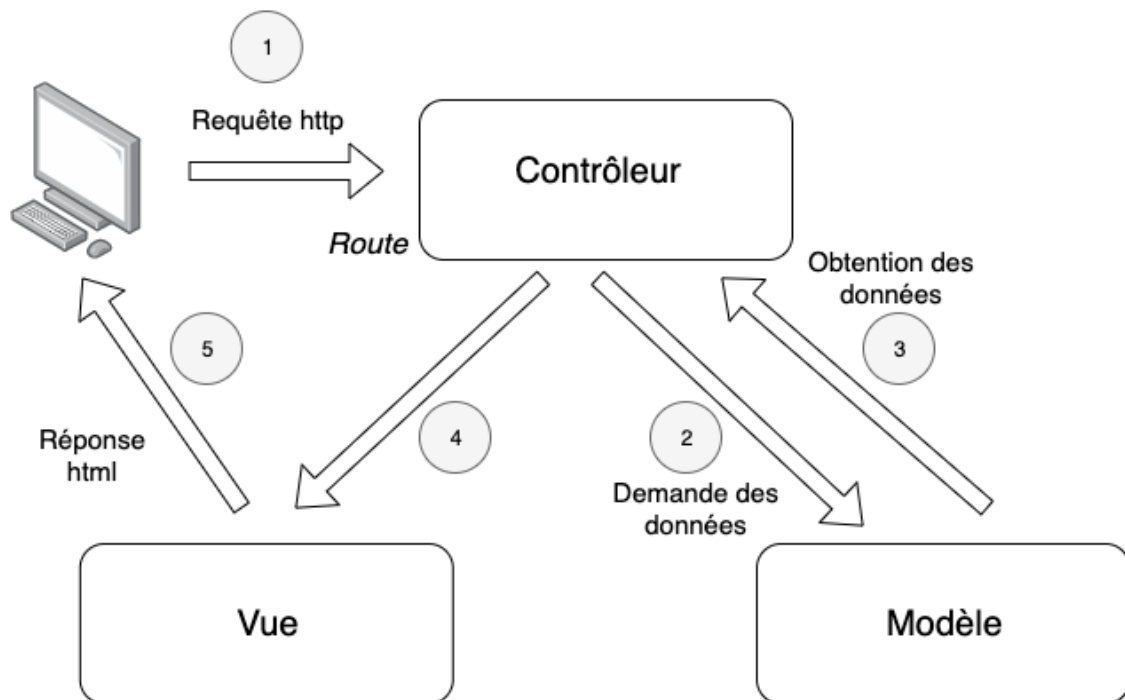


On peut directement voir que l'architecture MVC est utilisée automatiquement avec ce framework.

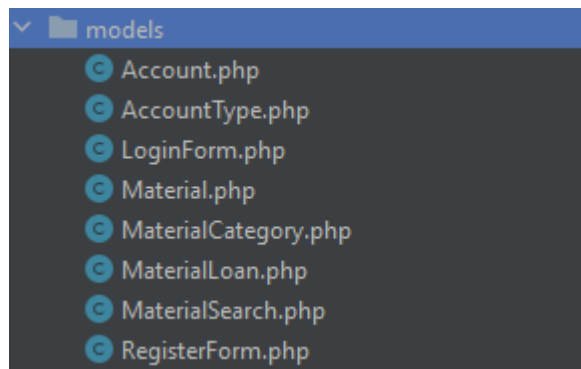
Architecture du site

Comme dit précédemment, yii2 crée une architecture MVC lors de la mise en place du framework.

Une structure MVC permet de séparer les données et le business logic (model) de l'application de la présentation (vue). Le contrôleur sert d'intermédiaire entre les modèles et les vues.



Pour être plus précis du contenu des dossier Model, Vue et Contrôler, en voilà le contenu :



Il est important de comprendre que yii2 comporte une extension qui s'appelle gii. Gii permet de générer des models, des CRUD et bien d'autres choses. Ici dans le dossier « models », gii a créé un modèle pour chaque table de la base de donnée sélectionnée.

Grâce à ça les chaque modèle a les bonnes relations avec les autres tables, ça implique que les requêtes SQL sont très simples à exécuter.

Sans comprendre les modèles automatiquement créés, sont à clarifier :

LoginForm.php : C'est le model derrière le formulaire du login, il va donc par exemple vérifier la validité des identifiants entrés

RegisterForm.php : Comme le LoginForm.php, sauf qu'il travail derrière le formulaire du register.

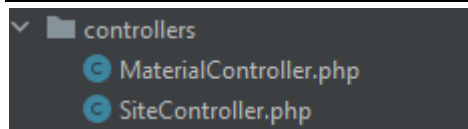
MaterialSearch.php : Ce fichier a été généré grâce à gii. Il a pour but de s'occuper du filtrage lorsqu'un utilisateur filtrera sur un ou des champs des matériaux.
(Les champs rouges ci-dessous)

Materials

Create Material

Showing 1-1 of 1 item.

#	Material Name	Model	Inventory Number	Serial Number	Status	
		Rev3				
1	Arduino	Rev3	24	139872dfx	Unavailable	  



FINIR APPLICATION AVANT DE COMMENTER CETTE PARTIE

Authentification sécurisée (cryptage de mot de passe)

Afin de sécuriser les mots de passe des utilisateurs, je les ai hashés les mots de passe grâce à une fonction de yii2.

Donc voici la fonction ci-dessous, qui permet d'encrypter le mot de passe

```
$newAccount->password = Yii::$app->getSecurity()->generatePasswordHash($this->password);
```

Lorsqu'un mot de passe a été encrypté, il est aussi important de pouvoir le vérifier. Par exemple lorsque l'utilisateur se login, une vérification du mot de passe hashé doit être faite.

```
if (!$user || !Yii::$app->getSecurity()->validatePassword($this->password, $user->password)) {
    $this->addError($attribute, error: 'Incorrect email or password.');
```

On peut voir ici que la fonction validatePassword, prends 2 paramètres. La chaîne de caractère qui n'est pas encrypté et celle qui est hashée. Grâce à cette fonction on peut donc vérifier si les mots de passe se correspondent.

Voilà à quoi ressemblent les mots de passe hashés dans la base de données.

```
password
$2y$13$/O8ngF54/1TT2FZVkr9qHuYIfq3rU9EwioFeoJl2uFJt4qHwvz1EW
$2y$13$m904MthIO7p7c6x9/hJFUOVNWQVBnYoWhVmCt5Jf9E9HfHX4IA...
$2y$13$6UriaI9yMUjx/SBess/x.e4f3U6ChaNZMtQcJ/wWeD95s/69p5Ji
$2y$13$yVre8cXZ0e9.j2WIDupj3eGP3J0gHLZPoXthm4VpTLbluh7pW1nEG
$2y$13$3WlQczRq7UXLM/acsWfOKuCFIMVQh72IEs7OW0BLv50jXk1Horf02
$2y$13$qhMLDFfQAhUWj.7iydB.wOD/pQYnPLIEh9i1nJHVh4fgln6h.GRQq
$2y$13$Y.ffw5FapVNjZXpwTcReGeaeVc6BFsr.B2RSHLdlvP2uvyv4vAOB.
$2y$13$JNbFh/oiw.7EkW/cEUTn4OWOGLb4F1r5xGWnmhbxjCyDwK06G...
$2y$13$2tspIpx4436UJ2GzW/q3eGeVpfbBeb9Oqcrn7NSgjYJ0hSuB4Yss6
$2y$13$wFOsn/PR1VGM84/tWf2q.OqWZdHzqIRHUNYyOSZ3Zmu1RozxLp...
```

3.1 Description des tâches effectuées

Décrire la réalisation "physique" de votre projet :

- les répertoires où les logiciels sont installés, la liste de tous les fichiers et une rapide description de leur contenu
- les versions des systèmes d'exploitation et des outils logiciels
- les principales options de configuration "non standard" choisies
- la description exacte du matériel

3.2 Modifications apportées par rapport à la conception

Si durant la réalisation, on a dû introduire certains changements (problèmes de configuration matérielle ou logicielle, etc...), on les mentionne comme suit :

- *cause détaillée de la modification*
- *conséquences sur l'utilisation du produit*
- *actions envisagées ou possibles*

3.3 Description des tests effectués

Test de la consultation du matériel

Pour chaque partie testée de votre projet, il faut décrire :

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.4 Erreurs restantes

S'il reste encore des erreurs :

- *description détaillée*
- *conséquences sur l'utilisation du produit*
- *actions envisagées ou possibles*

3.5 Rapport de mise en service

Si une phase de mise en service chez le client est nécessaire, on détaille :

- *l'installation du projet chez le client (installation des machines, des logiciels)*
- *les tests officiels effectués chez le client et/ou par le client.*
- *les erreurs répertoriées :*
 - *description détaillée*
 - *conséquences pour le client*
 - *actions envisagées.*
- *Formation des administrateurs et des utilisateurs finaux du produit.*

3.6 Liste des documents fournis et dossier d'archivage

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *les supports d'archivage (2 DVD ou CD)*
- *le manuel d'Installation (en annexe)*
- *les manuels d'utilisation à l'usage des utilisateurs ou des administrateurs*

Attention: les documents de réalisation doivent permettre à une autre personne de maintenir et modifier votre projet sans votre aide !

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Sources – Bibliographie

- *Livres utilisés (Titre, auteur, date)*
- *Articles (Revue, date, titre, auteur)*
- *Sites Internet (URL) consultés*
- *Aides externes (noms)*

5.2 Glossaire

5.3 Journal de bord de chaque participant

Date	Durée	Activité

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Présentation de fin de projet (6 diapos par page) - pas obligatoire si la présentation est à venir

5.7 Supports d'archivage du projet

CD, DVD... dans une fourre en plastique