

07/06/2018

Gestionnaire de Junior Entreprise

Module Application Synthèse – M. Bonneton



Tayfun Ozturk–Julien Frillici–Bilge Ekinici

Gestionnaire de Junior Entreprise

Module Application Synthèse – M. Bonneton

Table des matières

.....	2
I. Présentation du projet	3
A) Fonctionnalités et spécifications	3
B) Maquette.....	5
C) Conception et modèle de données.....	6
II. Application web et implémentation.....	7
A) Base de données	7
B) Implémentation, MVC Pattern et langage PHP	8
C) Gestion de projet	10
III. Bilan	10

I. Présentation du projet

A) Fonctionnalités et spécifications

Notre projet d'application synthèse est un logiciel de gestion d'une junior entreprise. Nous avons choisi de proposer deux principales fonctionnalités : la gestion de convention et la gestion de factures.

Il y a deux types d'utilisateurs distingués : l'administrateur et les membres de la junior entreprise. L'administrateur a le droit d'effectuer toutes les actions sans restrictions, et surtout gère les comptes utilisateurs et a accès à la base donnée. L'utilisateur lambda, peut gérer ce qu'il veut faire par le biais de menu qui permet de naviguer à travers l'application. Il aura donc la possibilité de créer, modifier et finalement éditer des conventions qui ont été remplies avec un client et un projet spécifique. Il y aura une convention par projet interne de cette junior entreprise. La convention n'est plus modifiable après validation par l'administrateur. Puis, il y a également une création de facture associée à la création d'une convention qui reprend donc tous les critères relatifs à la facturation dans le cadre d'un projet. Celle-ci ne peut pas être modifiée.


En ce qui concerne l'édition de la convention, lorsque l'utilisateur décide d'éditer un document, que ce soit la convention ou la facture, il y a un export PDF réalisé et une impression des documents est possible.

En cas de volonté de modification d'une convention déjà éditée, nous n'avons pas prévu la création d'un avenant. En revanche nous avons convenu avec le client que dans ce cas de figure il conviendra de supprimer la convention existante et d'en recréer une nouvelle qui correspondra aux attentes du client.

On trouvera sur la convention toutes les informations relatives à la junior entreprise, au client, et au projet qui les lie.

L'application est codée en PHP, ce qui nous permet de proposer une application web portable et légère.

Voici ci-dessous deux captures d'écran d'un morceau de l'application :




Gestionnaire de Junior Entreprise.

N° Nom

N°	Nom du projet	Id Client	Date début	Date fin	Montant HT	Montant TTC	Acompte	TVA	Signature	Commentaire
1	Projet1	1	2018-06-04	2018-06-12	4500	5400	2200	20	0	Ceci est un test
2	Projet2	2	2018-06-12	2018-06-27	2000	2400	1000	20	1	Ceci est un nouveau test
3	Projet3	1	2016-05-01	2018-06-01	1300	1560	600	20	1	ceci est encore un test
4	TestBilge	3	2018-06-08	2018-09-14	2000	2460	120	23	0	Bien joué
5	TestBilge	3	2018-06-08	2018-09-14	2000	2460	120	23	0	Bien joué

Figure 1 : capture d'écran de consultation des convention



Gestionnaire de Junior Entreprise.

Nom Projet

Nom client Mail Tél N° Siret

N° Rue Code postal

Nom Collaborateur Prénom Collaborateur

Intitule tâche Qté Prix HT

Total HT TVA Total TTC

Acompte Date début : Date fin :

Commentaire

Figure 2: capture d'écran de création de convention

B) Maquette

Etudiant

LOGO

Identifiant

Mot de passe

OK

Paramètre serveur

Mot de passe oublié

Etudiant

LOGO

Gestion des conventions

Gestion des factures

Déconnexion

Etudiant

Nom client

Siret

N°

Rue

Code postal

Collaborateur

Intitulé tâche

Qte

Prix HT

+

Total HT

TVA

Total TTC

Acompte

Date début

Date fin

Commentaire

Retour

Valider

Déconnexion

Etudiant

N°

Nom

Recherche

N°	Nom	Date début	Date fin	Status
----	-----	------------	----------	--------

Modifier

Editer

Etudiant

LOGO

Création convention

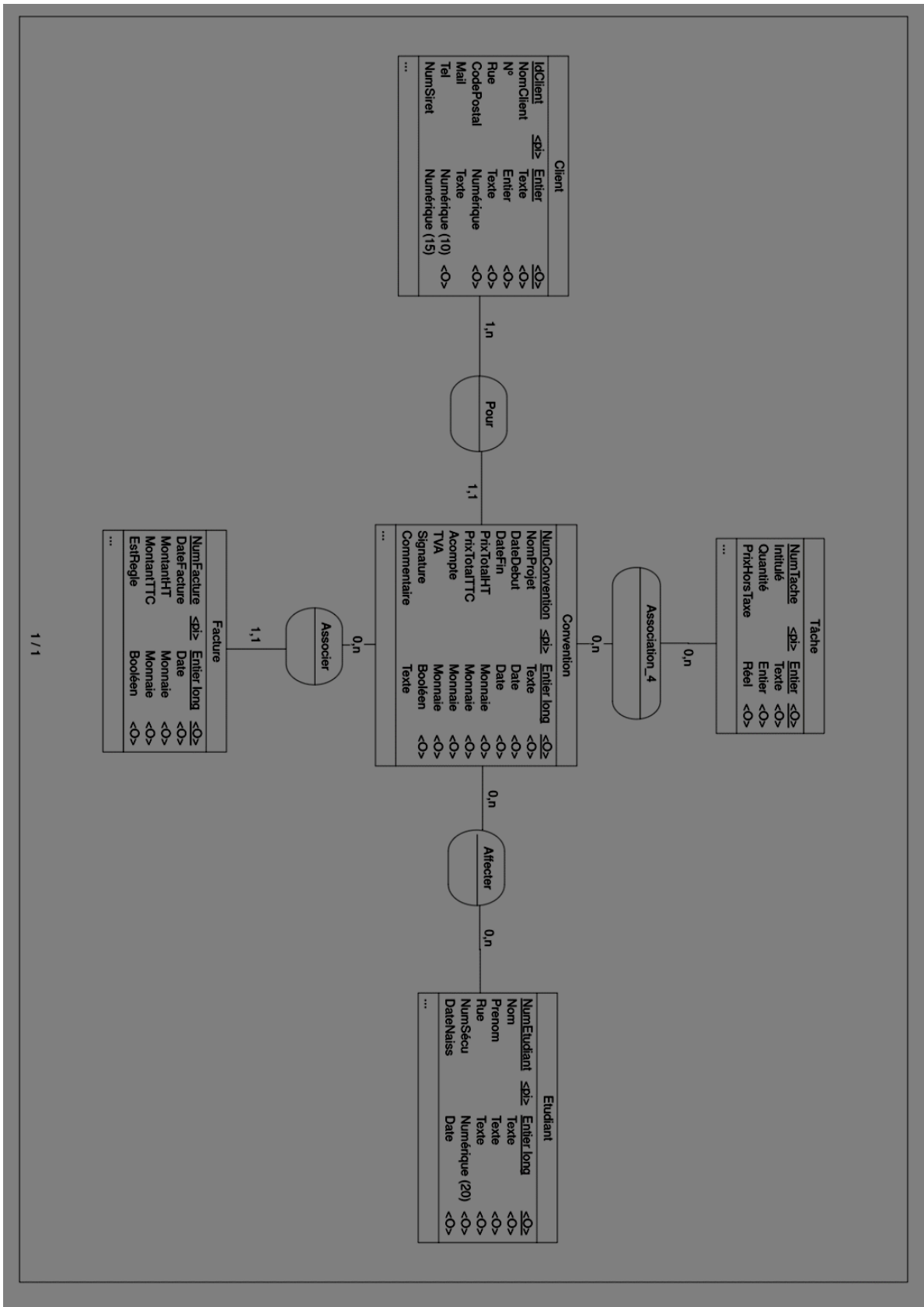
Modifier/Editer convention

Consultation convention

Retour

Déconnexion

C) Conception et modèle de données



II. Application web et implémentation

A) Base de données

La base de données regroupe les différentes tables qui composent les différents éléments de la junior entreprise. Elle se réfère au modèle conceptuel de données (cf. partie I.C)) .Ci-dessous un rapide aperçu de la table de convention, facture, et client :

Tableau 1: Table SQL convention

Options		NumConvention	NomProjet	IdClient	DateDebut	DateFin	MontantHT	MontantTTC	Acompte	TVA	Signature	Commentaire
<input type="checkbox"/>	Modifier Copier Effacer	1	Projet1	1	2018-06-04	2018-06-12	4500	5400	2200	20	0	Ceci est un test
<input type="checkbox"/>	Modifier Copier Effacer	2	Projet2	2	2018-06-12	2018-06-27	2000	2400	1000	20	1	Ceci est un nouveau test
<input type="checkbox"/>	Modifier Copier Effacer	3	Projet3	1	2016-05-01	2018-06-01	1300	1560	600	20	1	ceci est encore un test

Tableau 2: Table SQL facture

		NumFacture	DateFacture	MontantHT	MontantTTC	Payee
<input type="checkbox"/>	Modifier Copier Effacer	1	2018-06-06	2000	2200	0
<input type="checkbox"/>	Modifier Copier Effacer	2	2018-04-09	4500	5400	1
<input type="checkbox"/>	Modifier Copier Effacer	3	2018-06-02	1300	1560	1

Tableau 3 : Table SQL client

Options		IdClient	NomClient	NumRue	NomRue	CP	Mail	Tel	Siret
<input type="checkbox"/>	Modifier Copier Effacer	1	TechInfo SA	62	Ru de Vizay, DARDILLY	69230	test1@gamil.com	0400252525	19691774400019
<input type="checkbox"/>	Modifier Copier Effacer	2	IUT Lyon 1	90	Rue de la Technologie	69100, VILLEURBANNE	test2@gmail.com	0478000000	19691774400019

B) Implémentation, MVC Pattern et langage PHP

Afin d'implémenter ce projet, notre choix s'était à la base porté sur le langage java. Avec du recul, notre groupe a décidé de migrer sur le langage PHP afin de profiter de tous les avantages d'une application web : logiciel léger et portatif. Travaillant en trinôme, nous avons profité du design pattern MVC, composé du model, de la vue et du contrôleur pour séparer les tâches de chacun des membres du groupe.

Nous avons commencé par définir les besoins en général pour ce projet. Il y a eu beaucoup d'aller et venues pour demander des spécifications au client.

De là, nous sommes partis sur les vues dont nous allions avoir besoin pour l'application. On a décidé de partir d'un menu d'accueil, sur lequel nous pourrions décider de la navigation avec les deux fonctionnalités proposées. La conception des maquettes nous ont beaucoup aidé, nous nous en sommes servis de base, pour pouvoir broder dessus.

Nous avons utilisé un layout, sur lequel le bandeau de navigation apparaît, puis pour chaque vue, ajouté les différents formulaires ou boutons.

En parallèle, nous nous sommes concentrés sur les données et les bases de données. C'est à ce moment là que nous avons conçus les bases de données, et commencés une base pour ce qui allait nous servir dans l'implémentation de notre modèle MVC.

Puis, pour lier le tout, à chaque page est associée un contrôleur. Le modèle du pattern MVC comprend les classes DAO et DTO qui représentent les classes objets qui nous permettent d'être liés à la base de données, que nous avons implémenté progressivement. La base de données est sous MySQL, et pour des raisons pratiques, nous utilisons une base de données locale du serveur wamp.

Au niveau des vues, c'est donc principalement du CSS et de l'HTML. Afin de centraliser les fichiers CSS dans le dossier des Vues, nous avons utilisé un fichier htaccess qui nous permet de relier ces fichiers avec les fichiers contrôleurs.

```
Options +FollowSymlinks
RewriteEngine On
RewriteRule ^foo.html$ test.html [L]
RewriteRule ^Controller/css/convention.css View/css/convention.css [L]
RewriteRule ^Controller/css/header.css View/css/header.css [L]
RewriteRule ^Controller/src/logo.png View/src/logo.png [L]
```


Comme dans tout projet, nous avons eu quelques soucis au niveau de quelques données : il manquait certaines informations dans les formulaires des vues qui n'ont pas été prises en compte par les fichiers contrôleurs lors des vérifications de saisie utilisateur. Cela nous a fait perdre un peu de temps, sans compter que certaines autres informations étaient manquantes dans les classes du modèle : des attributs figurent dans la base de données mais ne sont jamais renseignés par l'utilisateur lors de la création d'une convention. Il a fallu faire de longues modifications dans chacune des implémentations des méthodes dans le modèle, ainsi que dans le contrôleur.

Ici encore, notre persévérance et notre esprit d'équipe nous a permis de corriger ces points là. Par manque de temps, nous avons priorisé la fonctionnalité de création/consultation de convention et laissé de côté celle de création automatique de facture à partir d'une convention existante.

Pour ce qui est de la création de convention, l'utilisateur se retrouve face à un formulaire qu'il doit remplir. Nous récupérons les données saisies avec le contrôleur, et avec ce lien modèle, accédons à la base de données pour renseigner ces informations directement sur le serveur wamp. S'il manque des données cruciales, la convention n'est pas validée. Pour les calculs de TVA, total HT et total TTC, nous avons rajouté un peu de JavaScript pour dynamiser la page et avoir des calculs instantanés sur le formulaire.

Dans la partie consultation de la convention, on effectue l'opération inverse avec un accès direct à la base de données, que l'on affiche via un tableau dynamique :

Rechercher											
N°	Nom										
	Numéro de convention	Nom du projet	Id Client	Date début	Date fin	Montant HT	Montant TTC	Acompte	TVA	Signature	Commentaire
1		Projet1	1	2018-06-04	2018-06-12	4500	5400	2200	20	0	Ceci est un test
2		Projet2	2	2018-06-12	2018-06-27	2000	2400	1000	20	1	Ceci est un nouveau test
3		Projet3	1	2016-05-01	2018-06-01	1300	1560	600	20	1	ceci est encore un test
4		TestBilge	3	2018-06-08	2018-09-14	2000	2460	120	23	0	Bien joué
5		TestBilge	3	2018-06-08	2018-09-14	2000	2460	120	23	0	Bien joué
Retour											

Pour une recherche directe, il est également possible de réduire les éléments affichés en introduisant un numéro de convention ou bien un nom de projet. Pour cela, on implémente le contrôleur correspondant, en vérifiant les champs entrés.

Dans l'onglet gestion des factures, nous avons pu implémenter les méthodes d'affichage de factures, mais c'était des factures déjà entrées dans la base de données et non générés par l'application. De ce fait nous n'avons pas jugé nécessaire de laisser cette option.

Afin d'exporter les fichiers hors de l'application, nous sommes partis sur un export

PDF. Nous avons importé des classes et des bibliothèques fpdf. Toujours par manque de temps et de connaissance, nous avons changé d'idée : transformer les informations récoltés des formulaires dans un fichier html pour un export. En cours de finition, il manque encore certains points. Nous préférons ne pas présenter ce point.

C) Gestion de projet

Pour la gestion du développement de ce projet, nous avons beaucoup utilisé le système de versionnage décentralisé GIT et la plateforme Github. Sous l'idée du pattern design MVC et des trois couches d'implémentation, Présentation, Métier et Données. On a également suivi une méthode itérative incrémentale. C'est à dire que chaque fonctionnalité a été d'abord dégrossie, puis ajoutée et améliorée au fur et à mesure. Cela nous a permis de présenter au client malgré tout un projet fonctionnel même s'il nous manque certains points, celui de l'édition de facture pour la gestion de cette junior entreprise.

III. Bilan

Pour conclure sur ce projet d'application synthèse nous pouvons distinguer deux points : ce que nous avons prévu en début de module, et ce que nous avons pu réaliser. Comme dit plusieurs fois précédemment, il nous a manqué de temps pour pouvoir finir cette application.

Cependant, livrer ce travail nous permettra de satisfaire le client. Avec une négociation, nous pouvons espérer gagner sa confiance et pouvoir améliorer notre travail avec du temps supplémentaire. A savoir que nous n'avons utilisé aucun framework, par manque de connaissance. Nous aurions peut être du passer un peu plus de temps à nous former sur ce genre d'outil afin d'en gagner davantage sur le long terme.

Néanmoins cela nous a permis de mettre en application nos connaissances de PHP sur un projet concret, et avons eu l'opportunité de choisir les technologies de A à Z , ainsi que les méthodes de développement, organisation et implémentations en groupe et en autonomie.