# Blanc Guillaume Mercredi 15 mai Givelet Elise

# Rapport projet auto-école

# Synthèse gestion de projet

## Répartition du travail

La répartition du travail c’est faite naturellement. Ceci en fonction des compétences, disponibilités et motivations de chacun.   
  
Dans un premier temps, pendant qu’Elise s’occupait de la base de donnée, Guillaume à mis en place la structure du projet. L’architecture choisie est qualifiée de MVC pour : model, view, controller.   
Guillaume a commencé par développer la couche modèle de donnée. Cette couche gère les données du site. Son rôle est de récupérer les informations dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve notamment toutes les requêtes SQL.   
Dans un second temps Elise à pris en charge la réalisation de l’interface graphique de l’application pendant que Guillaume terminait le modèle.

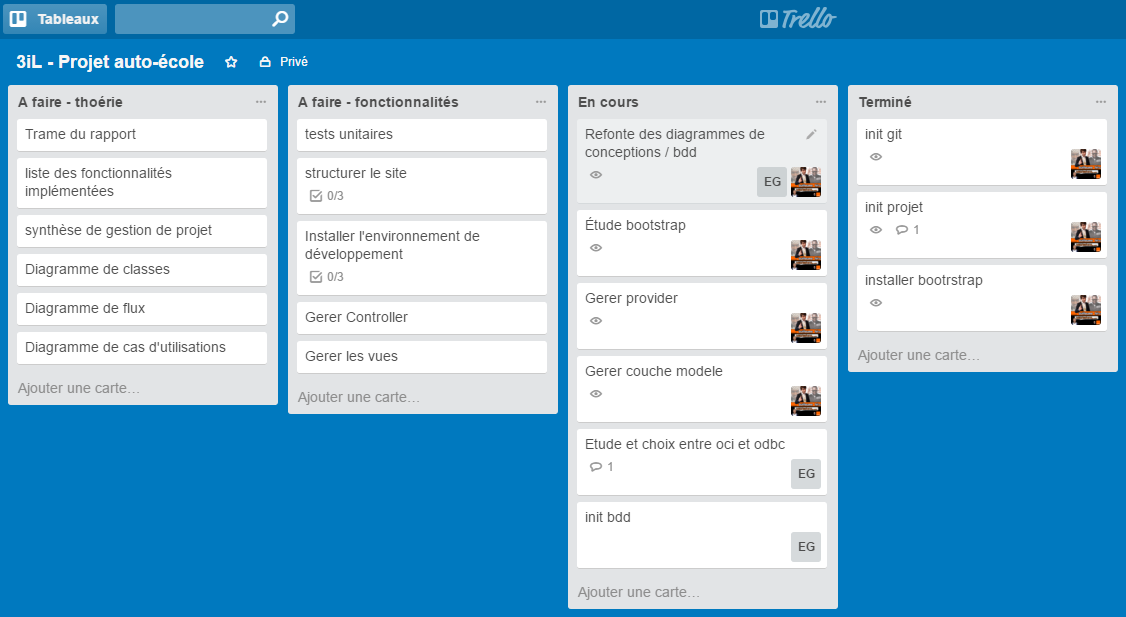
Une fois le site fonctionnel nous avons essayé de l’améliorer en s’assurant du bon fonctionnement des pages déjà présentes et en implémentant de nouvelles fonctionnalités. Ici la répartition du travail était plus souple et nous avons parfois travaillé sur une couche en même temps.

Au cours de ce projet, nous avons été amenés à développer en « pair programming » c'est-à-dire à deux devant la même machine. Ceci afin de réaliser certaines portions du code essentielles à l’application ou bien pour corriger certains bugs.

## 1.2) Planification des taches

Pour être honnête, nous n’avons réalisés ni PERT ni GANT pour ce projet. Tout d’abord parce que nous sommes septiques sur la pertinence de ces deux outils, notamment quand la date de fin du projet est fixée dès le début.  
Scepticisme renforcé par le fait que nous effectuons notre alternance dans deux des plus grandes entreprises Françaises (Orange et Airbus) et que ces procédés ne sont pas utilisés.   
De plus, au début du projet nous manquions de visibilité et nos estimations n’auraient en aucun cas été pertinentes. L’exemple le plus concret concerne l’initialisation de la base de données qui a duré plusieurs semaines dues aux problèmes liés à OCI et Oracle …  
  
Toutefois afin de mesurer notre avancée nous avons utilisé le logiciel Trello. Trello est un outil de gestion de projet en ligne inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.  
Cet outil permet d’avoir à un instant T une bonne représentation du travail qui a été fait et qui reste à faire.   
Nous avons priorisé les fonctionnalités, dans le but de réaliser les fonctions fondamentales du site en premier.

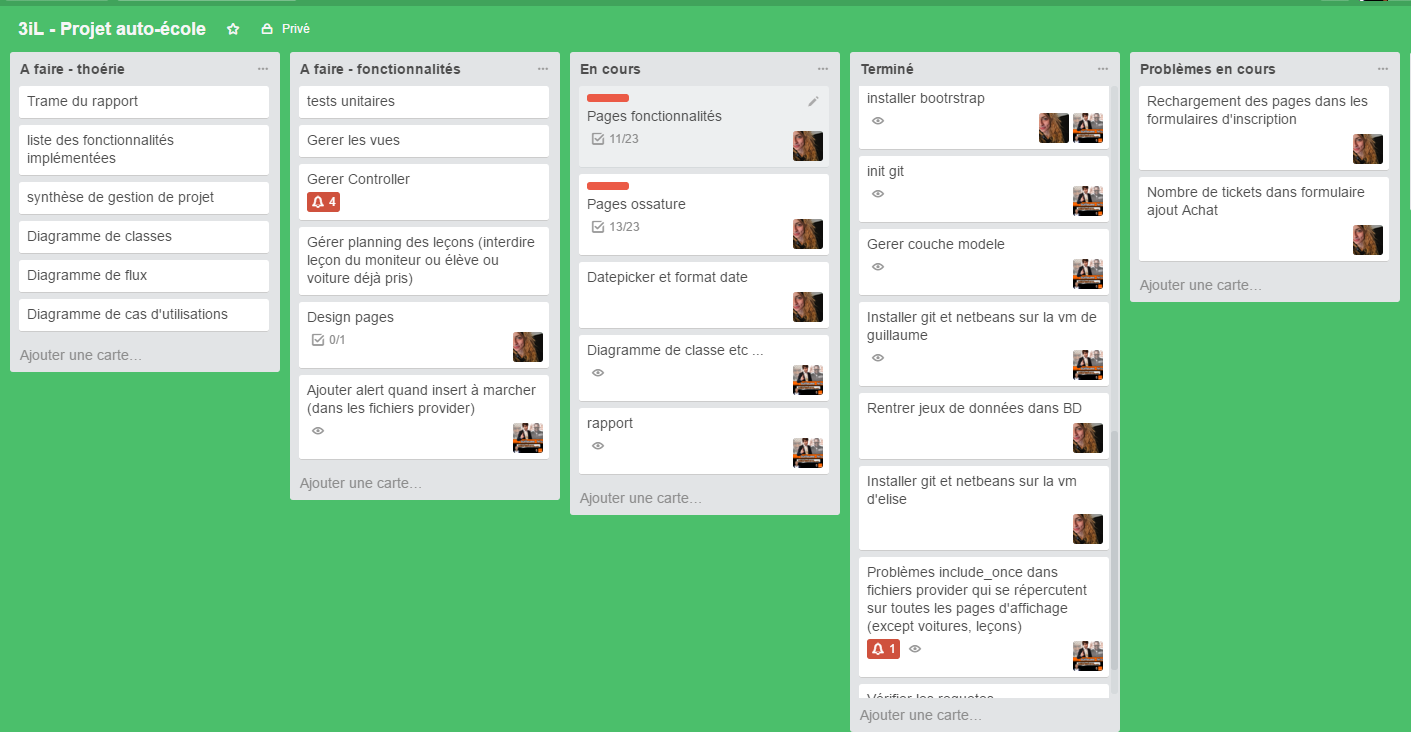
Ci-dessous notre projet Trello à différents stades du projet :

Le projet au 2 mai 2016  


Le projet au 10 mai 2016



Le projet au 15 mai 2016



## Méthodologie utilisée

Nous avons travaillé avec des principes de l’agilité que nous avons adaptée à nos besoins réels, puisque nous ne sommes que deux sur un projet de quelques semaines. A l’image de la réunion qualifiée de « Daily Scrum » nous organisions à la fin de chaque journée de projet une petite réunion dans laquelle nous évoquions le travail réalisé, les difficultés rencontrés et ce que nous comptions faire la prochaine séance.

# Liste des Fonctionnalités

## 2.1) Implémentées A l’heure actuelle le site permet de créer : - des formules - des clients - des élèves - des voitures D’effectuer : - des achats - des leçons de conduite D’obtenir la liste de : - tout les employés - toutes les voitures - toutes les formules - tous les élèves - tous les clients - toutes les leçons de conduites - tous les achats D’obtenir le détail - d’un élève : ses informations, ses lecons, le nombre de lecons disponible - d’un client : ses informations, ses élèves, ses achats De mettre à jour le kilométrage d’une voiture

## 2.2) Qui restent à faire

Possibilitié de modifier ou de supprimer les informations d’un client, d’un eleve etc …

# Diagrammes

## 3.1) Diagramme de classes

3.2) MCD

3.3) MLD

ELEVE(idEleve,nomEleve,prenomEleve,dateNaissanceEleve,telDomicileEleve,telPortableEleve,telTravailEleve,#idClient,#idAdresse,#idSalarie)

CLIENT(idClient, nomClient, prenomClient ,telDomicileClient,telPortableClient,#idAdresse)

SALARIE(idSalarie,nomSalarie,prenomSalarie,telSalarie,dateRecrutementSalarié, catgeorie, #idAdresse,#idVoiture)

ADRESSE(idAdresse,adresse,codePostal,ville)

FORMULE(idFormule,forfaitInscription,accesLeconsCode,nombreLeconsConduite,prixLeconConduite)

LECONDECONDUITE(idLeconConduite,dateDebutLeconConduite,#idVoiture, #idSalarie, #idEleve)

EXAMENDECODE(idSessionExamenCode,dateExamenCode,resultatExamenCode)

EXAMENPERMIS(idSessionExamenPermis,dateExamenPermis,resultatExamenPermis,#idEleve)

ACHAT(idAchat,nombreHeuresAchetees,montant,#idClients)

VOITURE(idVoiture,immatriculationVoiture,dateAchatVoiture,prixAchatVoiture,kilometrageVoiture,etatVoiture,#idModele)

3.4) Diagramme de flux

# Cas d’utilisation, tests unitaires, tests, scénarios de validation

Par manque de temps nous n’avons pas pu réaliser de tests unitaires. Nous pensions utiliser PHPUnit mais une ressource supplémentaire dédiée au test aurait été nécessaire.  
  
Toutefois chaque requête a été testée et le code de ce test a permis de réaliser les vues rapidement puisqu’il suffisait de reprendre le code correspondant.

L’exemple ci-dessous est le test des méthodes qui permettent de récupérer toutes les formules de l’auto-école ou d’une formule en fonction de son identifiant.



# Bonnes pratiques

Nous avons mis en place de nombreuses bonnes pratiques afin de tendre au mieux vers une application de qualité professionnelle.   
  
Nous avons notamment utilisé GIT qui est un système de contrôle de version. Notre dépôt est accessible à l’adresse suivante : <https://github.com/BlancGuillaume/auto-ecole>.   
La aussi, nous avons appliqué cet outil avec rigueur. A la fin de chaque nouvelle fonctionnalité un commit, commenté était réalisé.

Le code a été continuellement commenté, notamment les méthodes vitales au fonctionnement de l’application. Nous avons développé avec les conventions de code PHP officielles : <http://www.php-fig.org/psr/psr-2/>.   
  
Comme évoqué précédemment notre projet a été découpé en différentes couches ou modules. Cette structure permet une meilleure lisibilité du code. Il sera ainsi plus facile par la suite d’effectuer des évolutions.   
  
Nous avons travaillé avec le paradigme « objet ». La aussi, ce type de programmation permet une organisation plus cohérente du projet, une maintenance facilitée et une distribution du code plus aisée.

Pour conclure sur ce point, notre philosophie était de faire les choses avec rigueur et professionnalisme. Nous préférons n’avoir fait que 70 % de ce qui était demandé, mais l’avoir fait correctement : code commenté, fonctionnel, organisé, évolutif.