Lemarquis Sulivan

Marrucho Hèlen 2ème année de BTS SIO

Silva Thibaut

**RAPPORT DE PPPE**

Maison des ligues : projet HelpDesk

Lycée Raymond Poincaré

Table des matières

[Introduction : 3](#_Toc384366878)

[Présentation du contexte : 3](#_Toc384366879)

[Qu’est-ce que la maison des ligues ? 3](#_Toc384366880)

[Quelle est la demande de la maison des ligues ? 3](#_Toc384366881)

[Étude du Projet / Présentation et objectifs généraux du projet 4](#_Toc384366882)

[Outils utilisés dans le cadre du projet Maison des ligues : 4](#_Toc384366883)

[Elaboration du cahier des charges : 4](#_Toc384366884)

[Les technologies et moyens utilisées 5](#_Toc384366885)

[Qu’est-ce que Doctrine ? 5](#_Toc384366886)

[Qu’est-ce qu’un orm ? 5](#_Toc384366887)

[Qu’est-ce que Select2 ? 5](#_Toc384366888)

[Qu’est-ce que github ? 5](#_Toc384366889)

[Élaboration du projet 6](#_Toc384366890)

[Les différentes phases et aspects du travail 6](#_Toc384366891)

[Etude de projet 7](#_Toc384366892)

[Conclusion : 13](#_Toc384366893)

# Introduction :

## Présentation du contexte :

### Qu’est-ce que la maison des ligues ?

La Maison des Ligues de Lorraine (M2L) a pour mission de fournir des espaces et des services aux différentes ligues sportives régionales et à d’autres structures hébergées. La M2L est une structure financée par le Conseil Régional de Lorraine dont l’administration est déléguée au Comité Régional Olympique et Sportif de Lorraine (CROSL).

### Quelle est la demande de la maison des ligues ?

Le projet BugLigue a pour but de fournir à l'ESN (Entreprise de Services du Numérique) DevPlus un outil d'aide à la résolution des problèmes qui surviennent sur ses applications fournies à la maison des ligues. En effet, DevPlus a installé plusieurs applications différentes au sein de la M2L et la gestion de la maintenance nécessitait un nouvel outil pour rester efficace.

# Étude du Projet / Présentation et objectifs généraux du projet

## Outils utilisés dans le cadre du projet Maison des ligues :

* PhpStorm
* GitHub
* Wamp
* PhpMyAdmin
* Doctrine 2

## Elaboration du cahier des charges :

* Etude des fonctionnalités proposées par l’application
* Mise en place d’un diagramme UML
* Elaboration d’une application fonctionnelle comportant :
* Un espace responsable qui pourra :

- Filtrer et affecter des bugs.

- Attribution d’une priorité sur un bug.

* Un espace technicien qui pourra :

- Avoir la liste des bugs à résoudre.

- Clôturer un bug avec un compte rendu.

* Un espace club qui pourra :

- Déclarer un bug.

- Voir l’historique de leurs bugs avec un statut.

## Les technologies et moyens utilisées

Nous avons utilisé du php, javascript, jquery mais particulièrement Doctrine 2 qui est une nouvelle notion pour nous. Nous avons aussi utilisé SELECT2 une librairie jquery. Notre projet a été créé sur un repository Github.

### Qu’est-ce que Doctrine ?

Doctrine est un ORM (couche d'abstraction à la base de données) pour PHP.

### Qu’est-ce qu’un orm ?

Un mapping objet-relationnel est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé.

### Qu’est-ce que Select2 ?

Select2 est une librairie permettant un meilleur rendu design sur les champs select. Elle est très simple à utiliser grâce à sa documentation correctement fournie.

<http://ivaynberg.github.io/select2/>

Nous avons aussi utilisé github et crée un repository pour le projet afin de le partager et de l’utiliser sur phpstorm.

<https://github.com/ThibautSilva/bug>

### Qu’est-ce que github ?

Github est un outil utilisant git, qui permet le versionning et le travail en collaboration.

Couplé à Phpstorm, github permet de directement partagé les différents changements entre différent membres de l’équipe.

# Élaboration du projet

## Les différentes phases et aspects du travail

Avant toute chose, notre projet a été découpé en différentes phases de travail, afin de planifier et se répartir le travail au mieux. Notre axe de travail est donc découpé en quatre grandes phases qui sont les suivantes :

Phase 1 : Etude du projet.

Phase 2 : Réalisation de documentation

Phase 3 : Création des vues de l'application

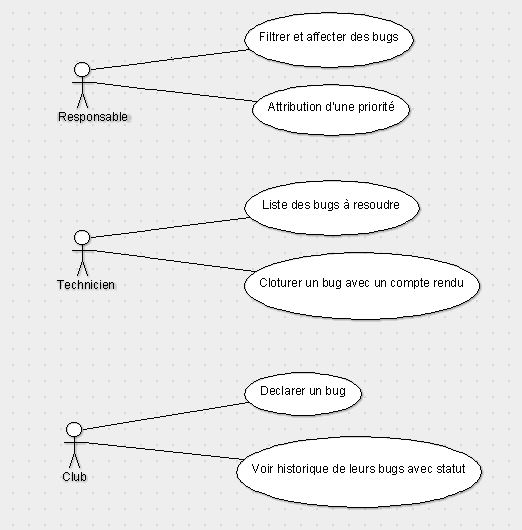
Phase 4 : Coder l'application

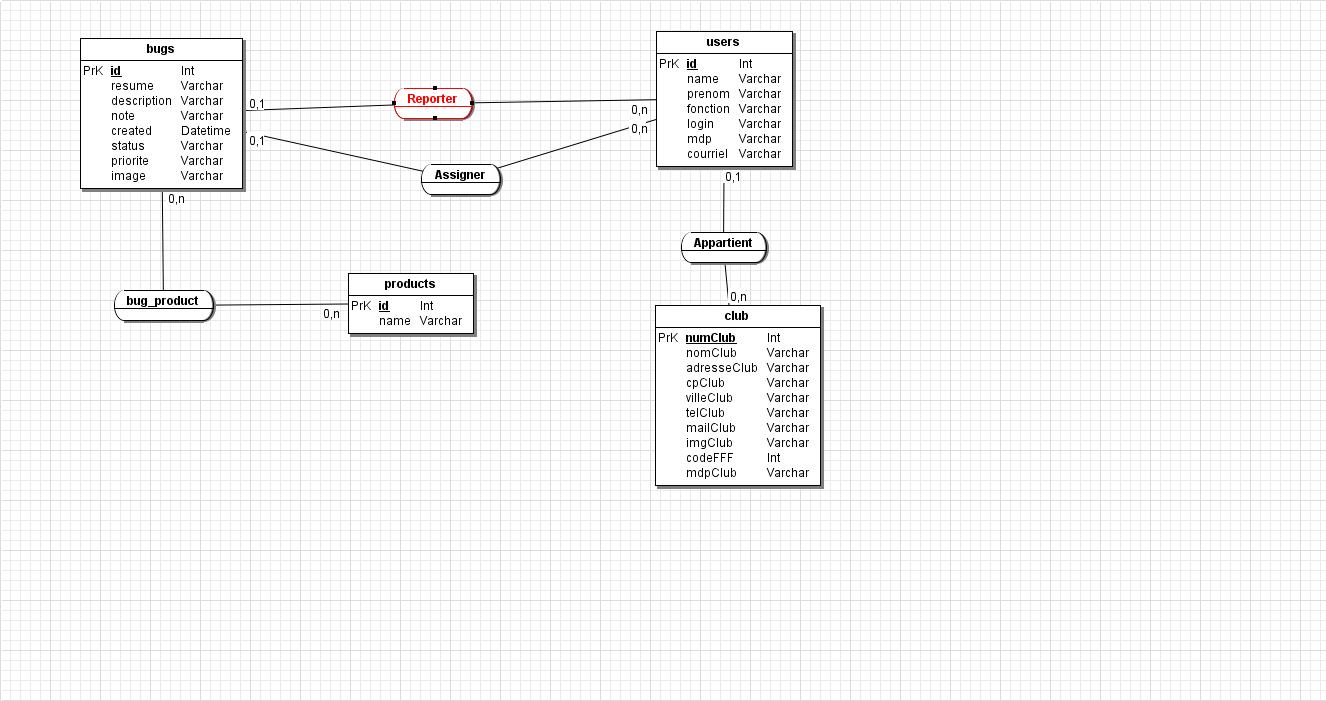
# Etude de projet

La première phase a été d'étudier le projet, de prendre en compte la demande, d'élaborer un cahier des charges et nous nous sommes données certaines consigne à respecter pour le travail en groupe. Nous avons aussi réparti les tâches mais beaucoup de parties ont été travaillé à plusieurs du fait du double usage de certaines parties comme les fonctions.

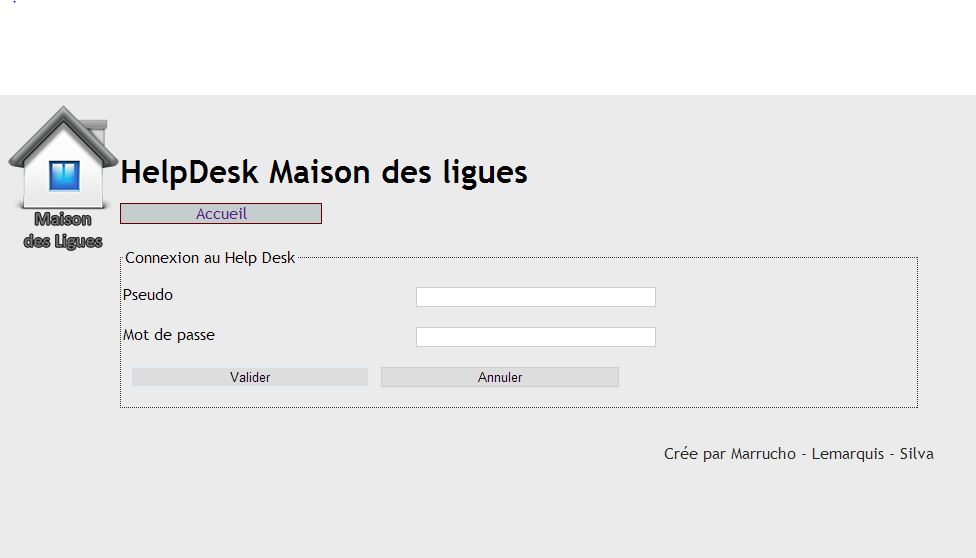
La deuxième a été de réaliser un diagramme UML avec toutes les fonctionnalités que l'application devra comporter, un mcd de la base et un diagramme des classes. Ces documents se trouvent ci-dessous.

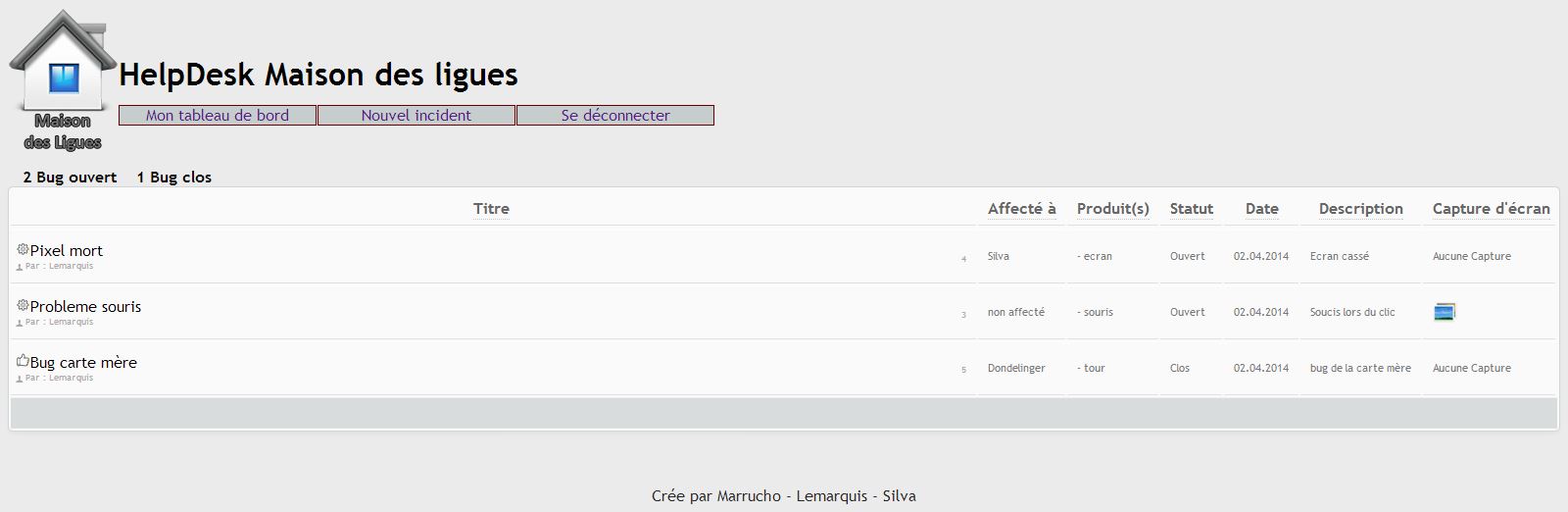
Le diagramme des cas d’utilisation tout d’abord :



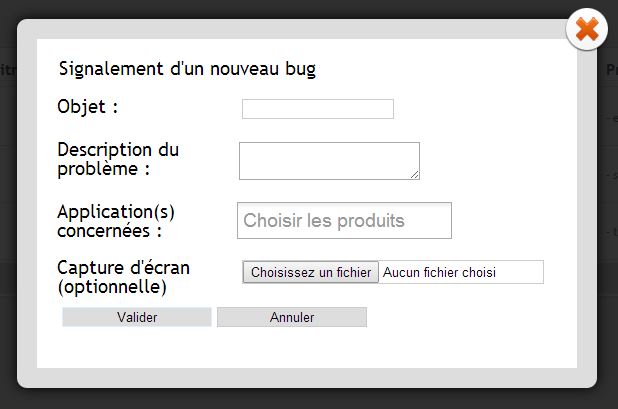
Le MCD a aussi été réalisé :

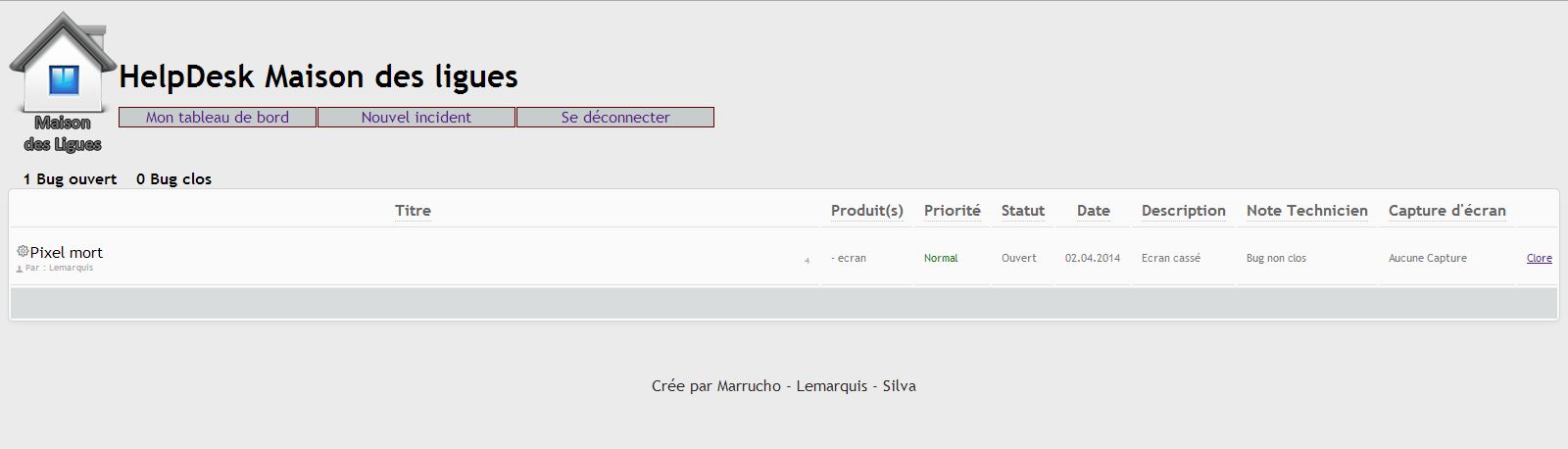
Suite à ça, la création des vues pour avoir un aspect global du rendu que devra donner l’application.

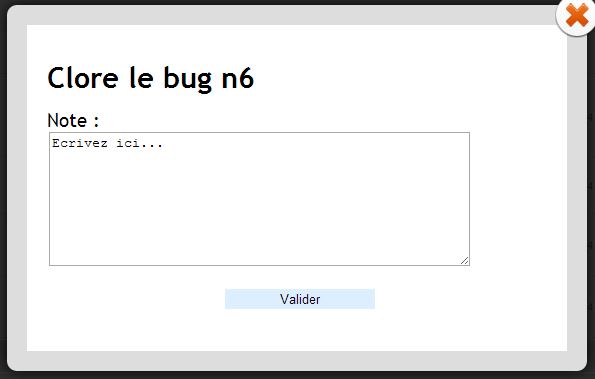
La page de connexion :

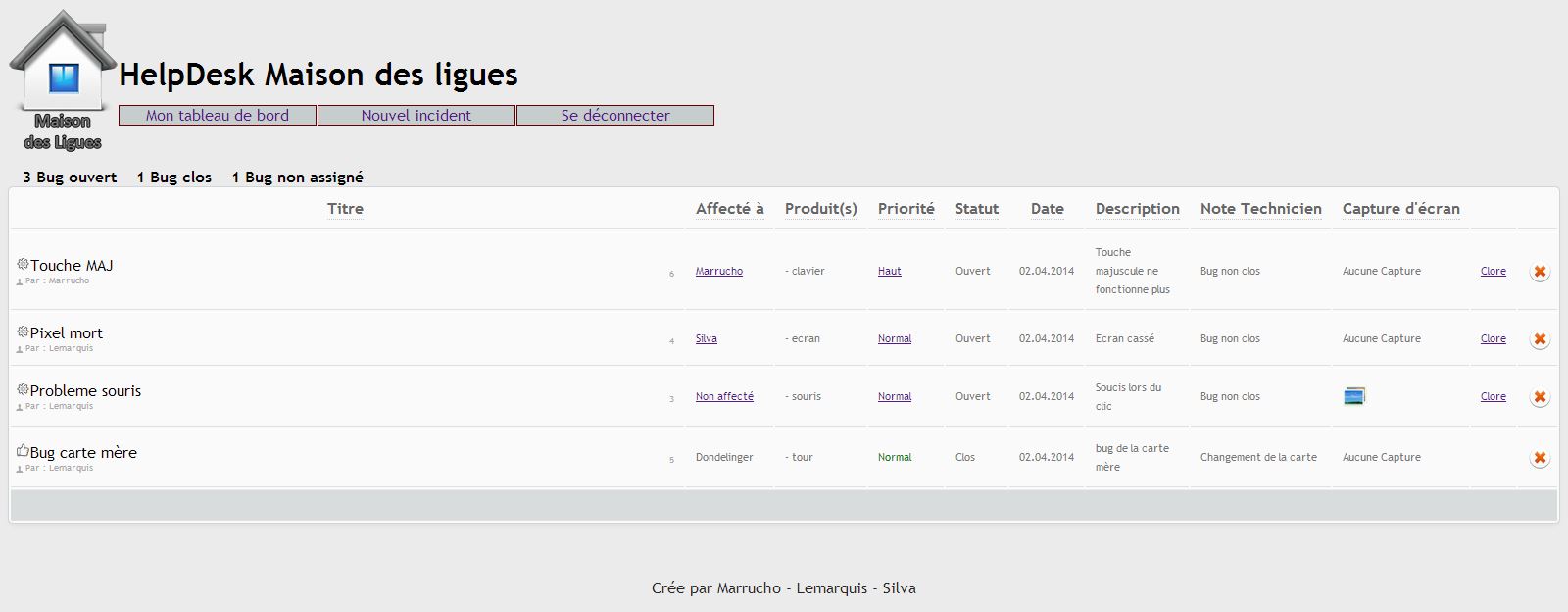
Le tableau de bord côté club tout d’abord, avec la liste des bugs qu’il a créé. Son nom, sa capture d’écran, ses produits, son statut, sa date de création et son affectation. Il a juste la possibilité de crée un nouveau Bug.

Note : La création d’un nouveau bug est disponible pour tous les utilisateurs, nous avons choisis cela par facilité pour les responsables et techniciens qui veulent déclarer un bug sans devoir crée un compte « Club ».

Ci-dessous la vue popup du formulaire nouveau bug, il comprend un objet, une description, les applications concerné (avec select2) et une capture d’écran : 

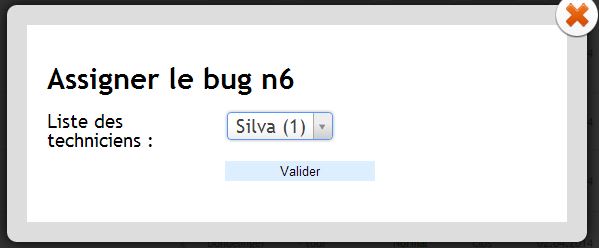
Ensuite la partie Technicien, il peut créer lui aussi des bugs, voir les bugs qui lui sont assigné et peut les clore. Sa liste des bugs contient le nom du bug, ses produits, sa priorité, son statut, sa date de création, sa description, la note du technicien si le bug est clos et la capture d’écran.Voici son dashboard :

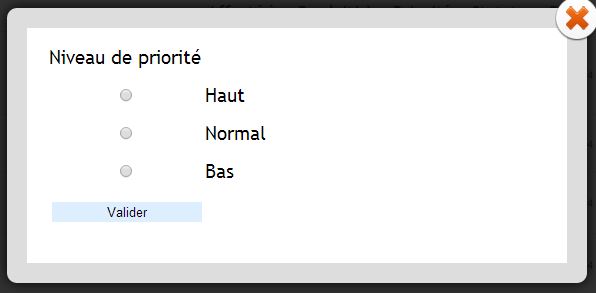
Il peut clore un bug avec un compte rendu : 

Côté responsable, nous avons un dahsboard qui contient toutes les fonctionnalités des deux autres, avec en plus le choix de l’assignation d’un bug, supprimer un bug et changer la priorité.

Pour affecter un bug à quelqu’un le formulaire est très simple : un simple select :

Note : Le nombre à côté et le nombre de bug qui lui sont déjà attribués.



La gestion de la priorité comprend trois niveaux ; haut, normal et bas. Le formulaire comprend juste des radios boutons : 

Suivi du codage des fonctionnalités liées à ces vues.

# Conclusion :

Dans le cadre de notre projet, nous devions réaliser une application permettant de gérer une plateforme de gestion d’incident. Tout s’est bien déroulé et nous avons pu terminer à temps.  
Le projet nous a amené à mieux comprendre le fonctionnement d’une architecture MVC. Plusieurs langages ont été approfondis tel que PHP, JavaScript. Nous avons aussi appris de nouvelles notions avec Doctrine2. L’utilisation de Composer et aussi nouveau pour nous. Nous avons appris à travailler à plusieurs en utilisant github ainsi nous avons appris à gérer notre projet au mieux.