

Mémoire de projet de fin d’étude

Présenté par

***SIABI Walid***

***SOUMIR Chaimae***

***HAJJAOUI Mohamed***

En vue de l’obtention du diplôme

**Licence Universitaire Professionnelle**

***Administration De Base De Données Et Technologie Web***

Thème

**« *7irassa* »**

Encadré par :

**SAOU Rachid**

**Année universitaire : 2017/2018**



07/07/2018

Projet Fin d’étude

Sous thème : 7irassa

###### Centre d’excellence en ingénierie et management industrielle FST DE SETTAT

**RAPPORT DE PROJET FIN D’ETUDE**

### « Administration De Base De Données Et Technologie Web »

# PROJET HIRASSA

****

**Encadrant** : SAOU Rachid **réalisé par :** SIABI Walid

SOUMIR Chaimae

HAJJAOUI Mohamed

### NOMENCLATURE

|  |  |
| --- | --- |
| **ABRÉVIATION** | **SIGNIFICATION** |
| **php** | Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP |
| **VsCode** | Editeur de code extensible développé par Microsoft |
| **MySql SGBD** | Système de Gestion de Bases de données |
| **UML** | Unified Modeling Language |
| **Scrum** | méthode agile de gestion de projet, utilisée notamment en développement logiciel |
| **SQL** | Structured Query Language |
| **CodeIgniter** | CodeIgniter est un framework PHP puissant avec un très faible encombrement, conçu pour les développeurs qui ont besoin d'une boîte à outils simple et élégante pour créer des applications Web complètes. |
| **Ajax** | (acronyme d'asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005. |
| **Git** | Est un système de contrôle de version distribué gratuit et open source conçu pour gérer tout, des petits aux très grands projets avec rapidité et efficacité. |
| **Github** | (exploité sous le nom de GitHub) est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git |

**REMERCIEMENTS**

Mes remerciements s’adressent dans un premier temps, à toute l’équipe pédagogique du centre d’excellence en ingénierie et management industriel et les intervenants professionnels responsables de la formation, pour avoir assuré la partie théorique et pratique de celle-ci.

Un grand merci va à tous les enseignants qui nous ont accompagnés durant notre cursus à CIGMA. Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à la réussite de ce rapport.

A cette occasion, nous exprimons notre profonde gratitude à Mr Rachid SAOU, pour ses conseils, son soutien et ses précieuses directives.

Nos chers parents, pour leurs immenses supports moraux et financiers, leurs confiances et leurs soutiens.

Je remercie bien entendu mes ami(e)s qui m’ont admirablement apporté soutien et réconfort depuis toujours.

Un remerciement particulier aux membres du jury pour avoir accepté de juger ce travail et de nous faire profiter de leurs remarques et conseils.

Merci à vous tous.

Table des matières

[PROJET HIRASSA 1](#_bookmark0)

CHAPITRE 1 : 2

CONTEXTE GENERAL DU PROJETS 2

Introduction GENERAL. 5

1. [Présentation du projet 5](#_TOC_250019)
2. Objectif de l'application. 5
   1. **- Objectif principaux. 5**
   2. **- Objectif spécifiques 6**

ANALYSE ET CONCEPTION 2

[PRESENTATION DU LANGAGE D'ANALYSE 11](#_TOC_250011)

[LE FORMALISME UML 11](#_TOC_250010)

[MISE EN ŒUVRE D'UNE DEMARCHE A L'AIDE D'UML 12](#_TOC_250009)

[LES DIAGRAMMES. 13](#_TOC_250008)

[Diagramme de Cas d'Utilisation. 16](#_TOC_250007)

Modele Conceptuel de Données. 18

Modele Logique de Données 19

CHAPITRE 3 : 2

TECHNOLOGIES ET OUTILS UTILISEES 2

[OUTILS DE MISE ON ŒUVRE UTILISES 21](#_TOC_250005)

[**Le langage PHP. 21**](#_TOC_250004)

[**VsCode 22**](#_TOC_250003)

[**Github… 22**](#_TOC_250002)

**MySql… 23**

[**Le MVC (Codeigneter) 23**](#_TOC_250001)

CHAPITRE 4 : 2

REALISATION ET MISE EN OEUVRE. 2

[DEVELOPPEMENT DE L'APPLICATION 25](#_TOC_250000)

**CHAPITRE 1 :**

**CONTEXTE GENERAL DU PROJETS.**

**INTRODUCTION GENERALE**

Ce chapitre a pour but de situer le projet dans son environnement. Il décrit dans un premier temps le projet généralement, et deuxièmement présente les objectifs des projets.

## Présentation du projet :

HIRASSA va mettre en place une solution pour gérer le planning des examens ainsi que la distribution des profs  et la gestion des sales pour les examens.

## Objectifs de l’application

##### Objectifs principaux :

**a-1 Le progiciels HIRASSA permettent-la :**

\_ Gestion des profs

\_ Gestion des sales

\_ Gestion des examens

\_ Gestion des Horaire des examens

\_ Gestion des convocations des profs

\_ Gestion des convocations pour des examens

\_ exporter des fichiers Excel et Word pour les différentes informations

\_ Ou cas de crache de système HIRASSA doit exporter des fichiers  pour passer ou traitements manuelle

\_Le mot de passe pour accéder à l’application

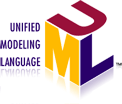
## **a-2 besoin non fonctionnel :**

La gestion des contrant la performance et la compatibilité se sent des     aspecte que le logiciel Doit les respecter

**CHAPITRE 2 :**

**ANALYSE ET CONCEPTION.**

#### PRESENTATION DU LANGAGE D’ANALYSE

Pour les phases de modélisation, j’ai fait usage de la toute dernière norme UML 2.0 publiée en 2006 par l’Object Management Group (OMG). (La version 2.2 est toujours en cours de spécification par l’OMG).

UML (en anglais Unified Modeling Language, « langage de modélisation unifié ») est un langage graphique semi-formel de modélisation des données et des traitements. C'est une formalisation très aboutie et non- propriétaire de la modélisation objet utilisée en génie logiciel.

UML est l'accomplissement de la fusion des précédents langages de modélisation objet Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'OMG.

## LE FORMALISME UML :

UML 2 propose 13 types de diagrammes (9 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML ; des méthodologies, telles que l'UnifiedProcess, axent elles l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (Use Case).

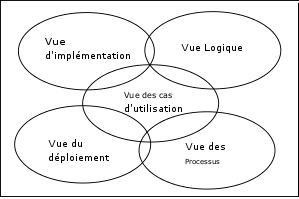
De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

UML se décompose en plusieurs sous-ensembles

* Les vues : Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues il est possible de définir (ou retrouver) le système complet.
* Les diagrammes : Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
* Les modèles d'élément : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagramme. Exemple d'élément : cas d'utilisation (CU), classe, association, etc.

## MISE EN ŒUVRE D’UNE DEMARCHE A L’AIDE D’UML :

Une façon de mettre en œuvre UML est de considérer différentes vues qui peuvent se superposer pour collaborer à la définition du système.



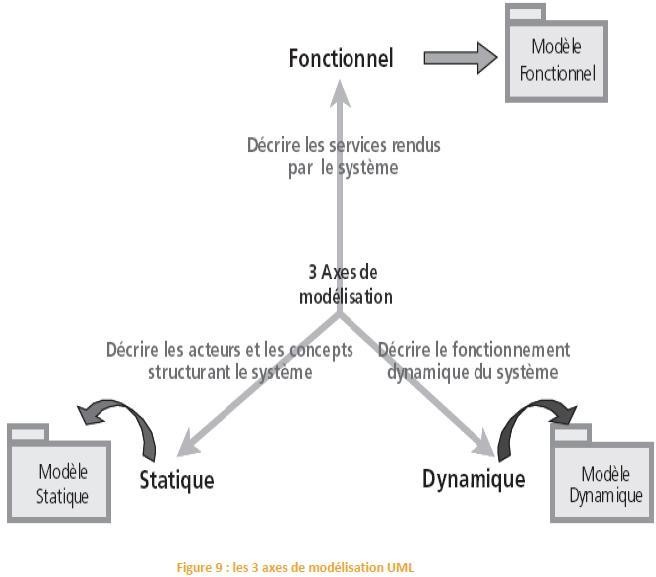
**Figure 1 : Complémentarité des différentes vues du système en modélisation UML**

Vue des cas d'utilisation : c'est la description du modèle "vue" par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI).

* Vue logique : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).
* Vue d'implémentation : cette vue définit les dépendances entre les modules.
* Vue des processus : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
* Vue de déploiement : cette vue décrit la position géographique et l'architecture physique de
* chaque élément du système (c'est le OÙ).

UML s’articule autour de plusieurs types de diagrammes, chacun d’eux étant dédié à la représentation des concepts particuliers d’un système logiciel. UML modélise le système suivant trois modes de représentation : le premier décrit les services fonctionnels rendus par le système, le second concerne la structure statique du système, le troisième concerne sa dynamique de fonctionnement.

Les trois représentations sont nécessaires et complémentaires pour schématiser la façon dont est composé le système et comment ses composantes fonctionnent entre elles :

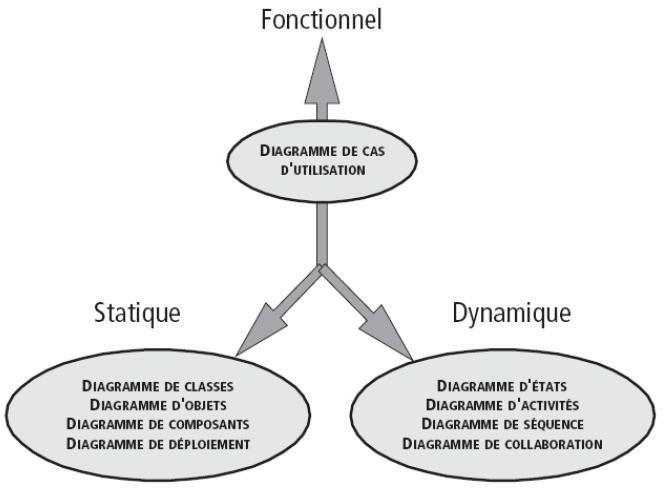


**Figure 3 : les 3 axes de modélisation UML**

## LES DIAGRAMMES

La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes.

Les 13 diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation des 3 axes précités d'un projet tout au long de son cycle de vie :



**Figure 3 : Les diagrammes UML représentés par axes de modélisation**

## DIAGRAMMES STRUCTURELS OU STATIQUES :

Les diagrammes structurels ou statiques (Structure Diagram) rassemblent :

* Diagramme de classes (cf. Class diagram) : il représente les classes intervenant dans le système.
* Diagramme d'objets (cf. Object diagram) : il sert à représenter les instances de classes (objets) utilisées dans le système.
* Diagramme de composants (cf. Component diagram) : il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)
* Diagramme de déploiement (cf. Deployment diagram) : il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
* Diagramme des paquetages (cf. Package Diagram) : un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, le Diagramme de paquetage sert à représenter les dépendances entre paquetages, c’est-à-dire les dépendances entre ensembles de définitions.
* Diagramme de structure composite (depuis UML 2.x, cf. Composite Structure Diagram) : permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

## DIAGRAMMES COMPORTEMENTAUX :

Les diagrammes comportementaux (Behavior Diagram) rassemblent :

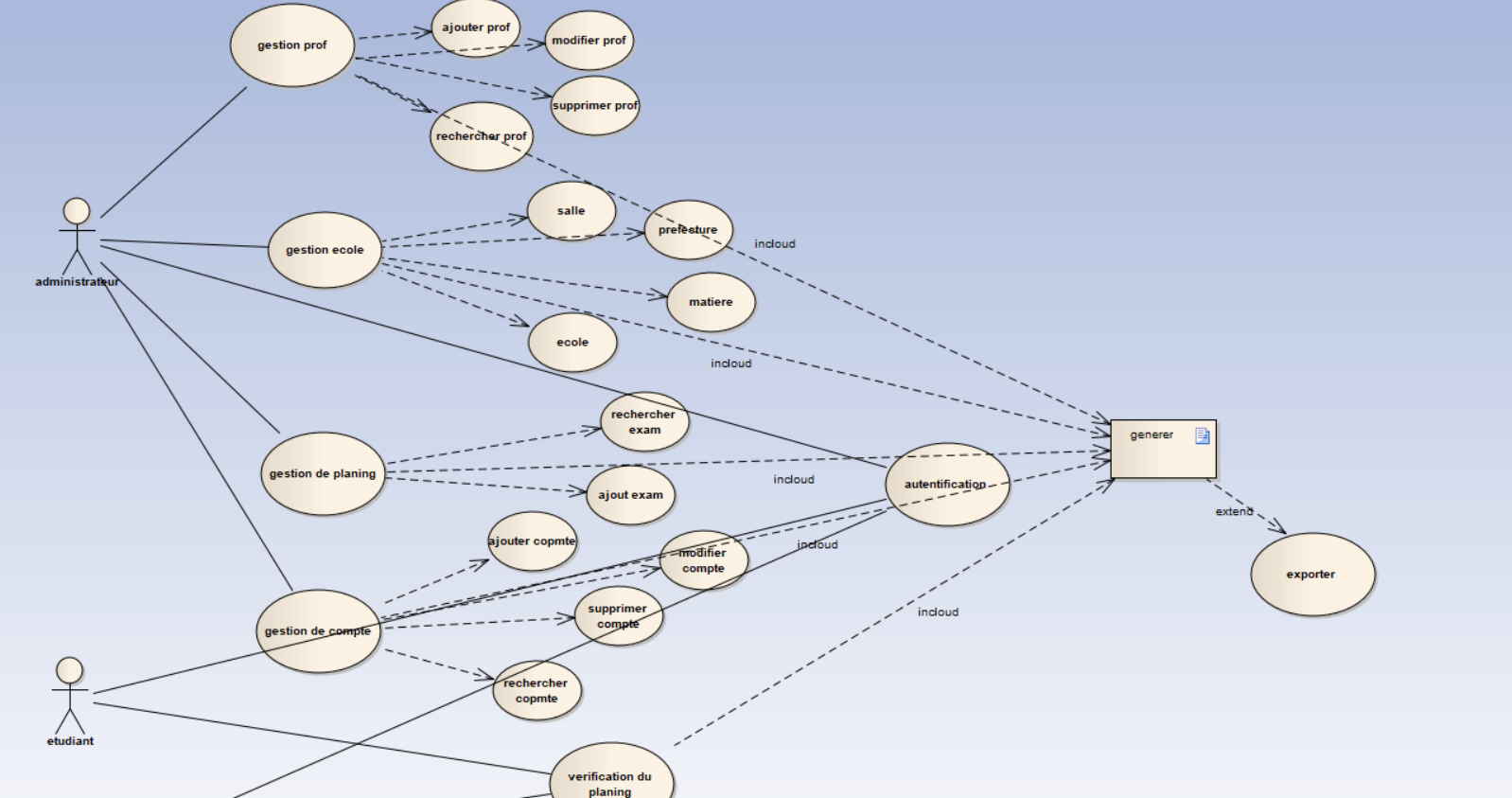
* Diagramme des cas d'utilisation (use-cases) (cf. Use Case Diagram) : il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
* Diagramme états-transitions (cf. State Machine Diagram) : permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.
* Diagramme d'activité (cf. Activity Diagram) : permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants*.*

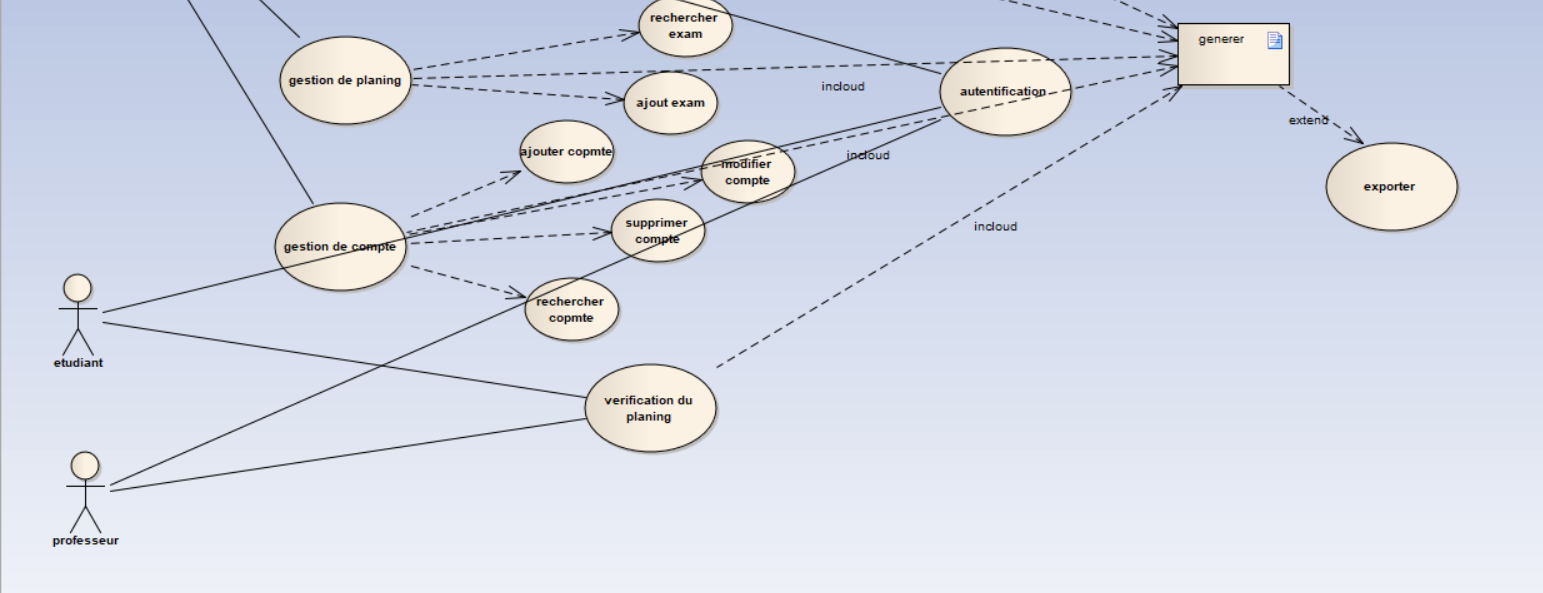
## DIAGRAMMES D’INTERACTION OU DYNAMIQUE :

Les diagrammes d'interaction ou dynamiques (Interaction Diagram) rassemblent :

* Diagramme de séquence (cf. Sequence Diagram) : représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
* Diagramme de communication (depuis UML 2.x, cf. Communication Diagram) : représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
* Diagramme global d'interaction (depuis UML 2.x, cf. Interaction Overview Diagram) : permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
* Diagramme de temps (depuis UML 2.x, cf. Timing Diagram) : permet de décrire les variations d'une donnée au cours du temps.

#### DIAGRAMME DE CAS D’UTILISATION

****

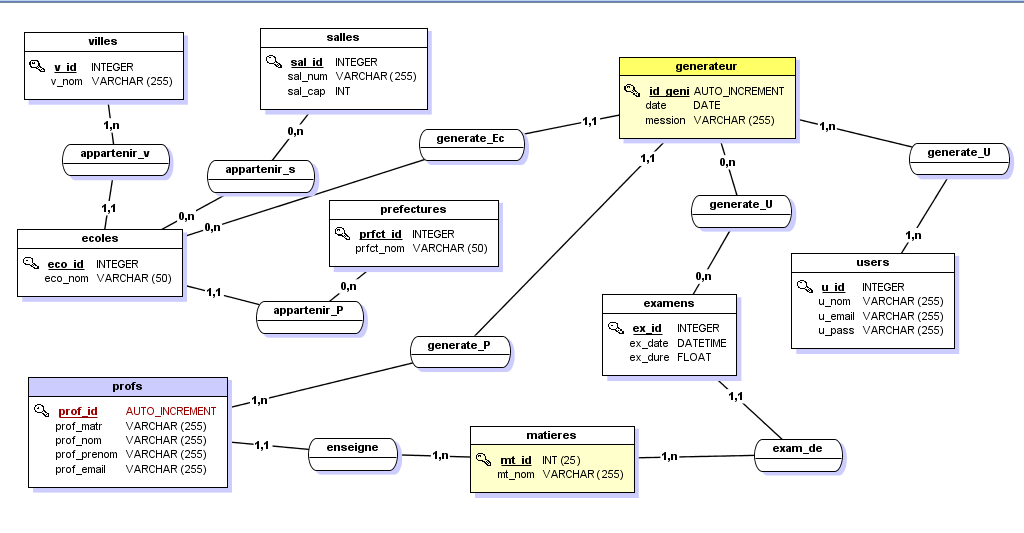
****

**Figure 4 : Diagramme de cas d’utilisation**

#### MODELE CONCEPTUEL DE DONNEES

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc d'une représentation des données permettant de décrire le système d'information à l'aide d'entités.

Le schéma suivant représente le Modèle Conceptuel des Données finales de l’application de gestion de formation.

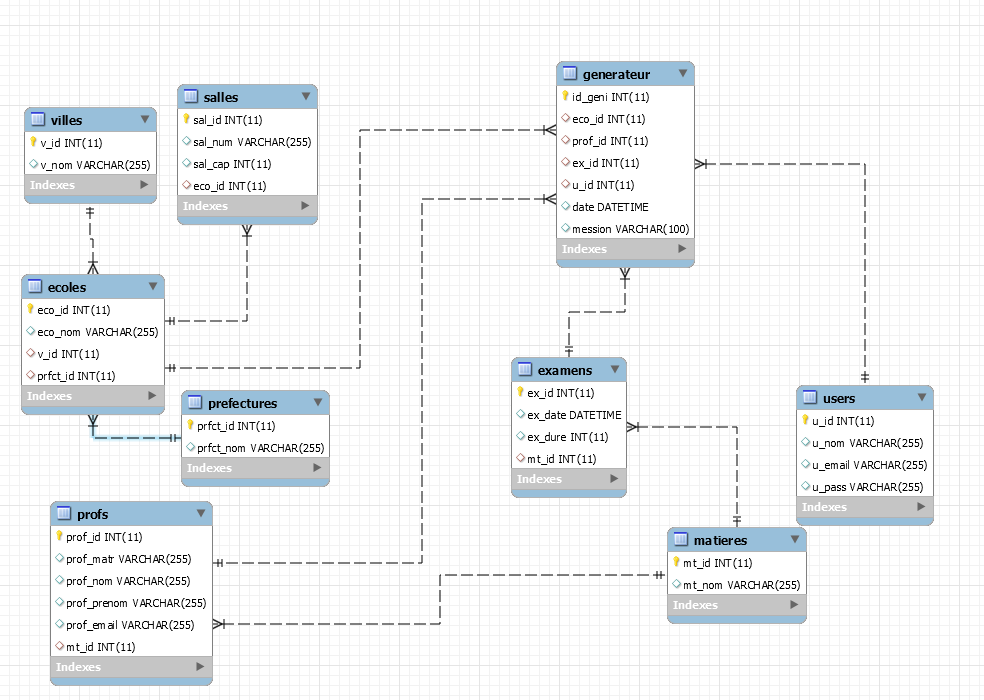


**Figure 6 : Modèle conceptuel de données**

#### MODELE LOGIQUE DE DONNEES

Le modèle logique est une image du modèle conceptuel intégrant les choix d’organisation des données.

Le schéma suivant représente le MLD final de l’application de gestion de formation.



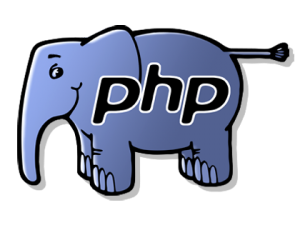
**Figure 7 : Modèle logique de données**

**CHAPITRE 3 :**

**TECHNOLOGIES ET OUTILS UTILISEES.**

#### OUTILS DE MISE ON ŒUVRE UTILISES

###### php

PHP est un langage de script utilisé le plus souvent côté serveur : dans cette architecture, le serveur interprète le code PHP des pages web demandées et génère du code (HTML, XHTML, CSS par exemple) et des données (JPEG, GIF, PNG par exemple) pouvant être interprétés et rendus par un navigateur. PHP peut également générer d'autres formats comme le WML, le SVG, le PDF.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent développées pour le Web. PHP est le plus souvent couplé à un serveur Apache bien qu'il puisse être installé sur la plupart des serveurs HTTP tels que IIS. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondants effectuée sur le serveur. La force la plus évidente de ce langage est qu'il a permis au fil du temps la résolution aisée de problèmes autrefois compliqués et est devenu par conséquent un composant incontournable des offres d'hébergements.

Il est multi-plateforme : autant sur Linux qu'avec Windows il permet aisément de reconduire le même code sur un environnement à peu près semblable (prendre en compte les règles d'arborescences de répertoires qui peuvent changer).

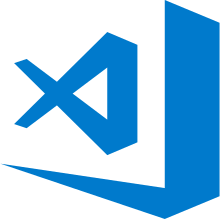
Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de programmation une bonne compréhension des principales fonctions usuelles ainsi qu'une connaissance aiguë des problèmes de sécurité liés à ce langage.

La version 5.3 a introduit de nombreuses fonctions nouvelles : les espaces de noms — un élément fondamental de l'élaboration d'extensions, de bibliothèques et de frameworks structurés, les fonctions anonymes, les fermetures, etc.

Aujourd'hui, près de 80 % des sites web utilisent le langage PHP sous ses différentes versions[Quand ?]. Plusieurs développeurs PHP responsables de ces sites utilisent en majorité la version 5.4 (38 %) dans leurs missions quotidiennes45.

Le langage PHP fait l'objet depuis plusieurs années maintenant d'un forum organisé par l'AFUP (l'Association Française des Utilisateurs de PHP), où experts de la programmation et du milieu se retrouvent pour échanger autour du PHP et de ses développeurs. Ce forum est par ailleurs un moment important pour ces acteurs car il est synonyme du lancement d'une étude consacrée aux développeurs PHP46 réalisée par l'AFUP et le cabinet de recrutement Agence-E.

###### Visual studio code

Visual Studio Code est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code [cross-platform](https://fr.wikipedia.org/wiki/Cross-platform), [open source](https://fr.wikipedia.org/wiki/Open_source) et gratuit, supportant une dizaine de [langages](https://fr.wikipedia.org/wiki/Langage_informatique).

Le [code source](https://fr.wikipedia.org/wiki/Code_source) est fourni sous la [licence libre](https://fr.wikipedia.org/wiki/Licence_de_logiciel_libre) MIT (plus précisément la [licence Expat](https://fr.wikipedia.org/wiki/Licence_Expat)) sur le site du projet sur [Github](https://fr.wikipedia.org/wiki/Github" \o "Github). En revanche, l'exécutable est proposé sur le site officiel de Microsoft sous une [licence privatrice](https://fr.wikipedia.org/wiki/Logiciel_privateur).

###### MySql

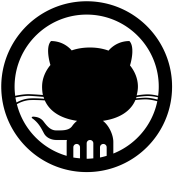


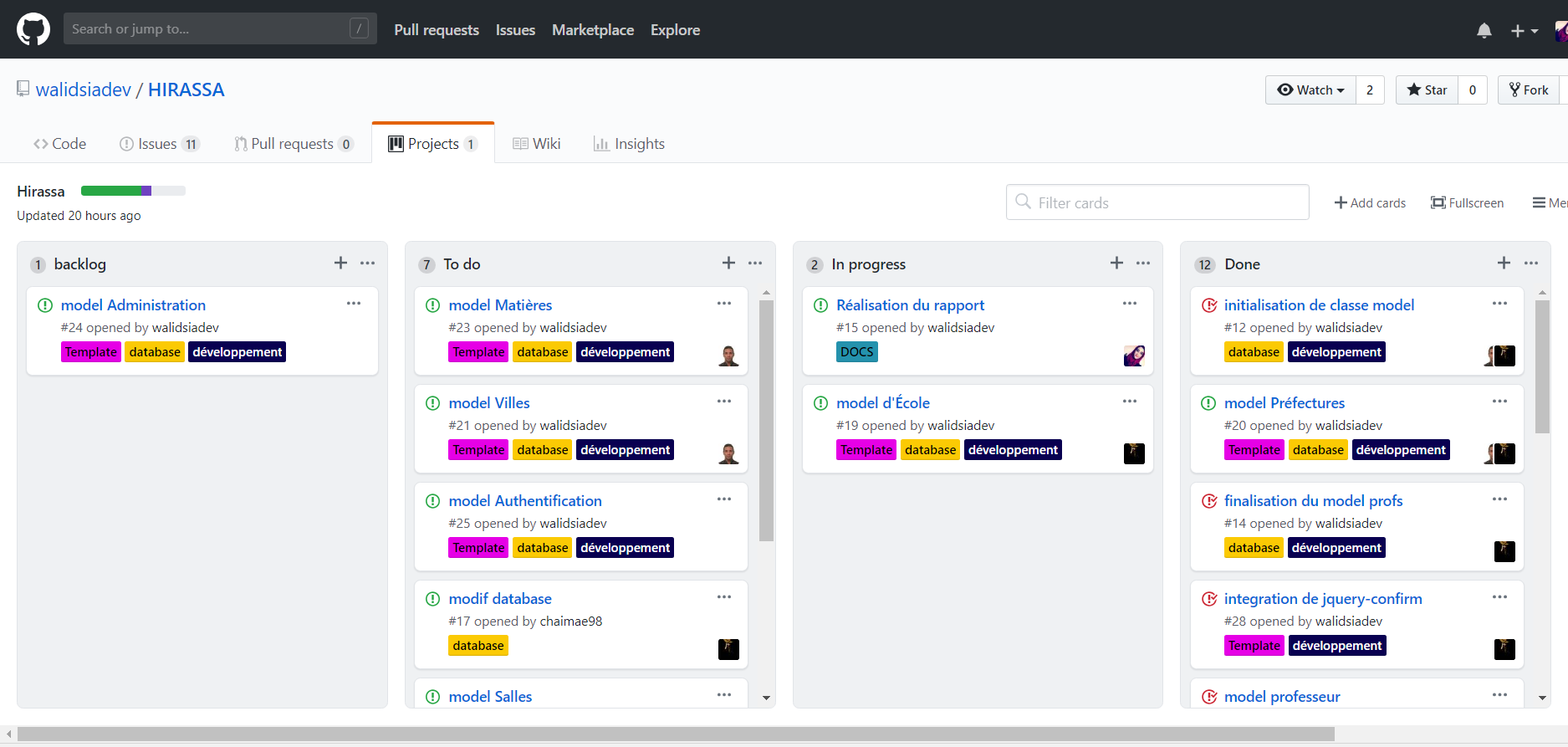
Le MySql est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de

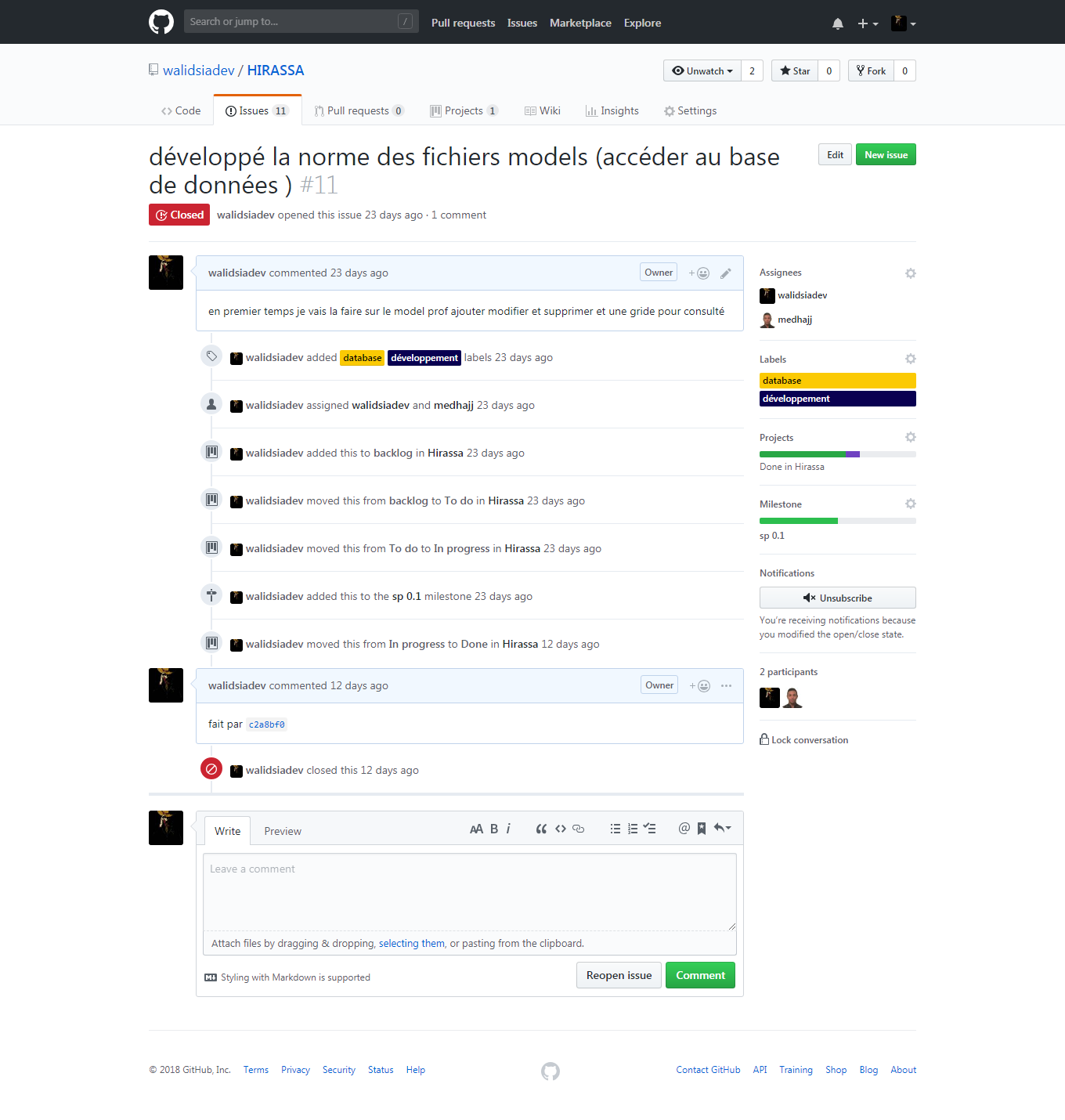
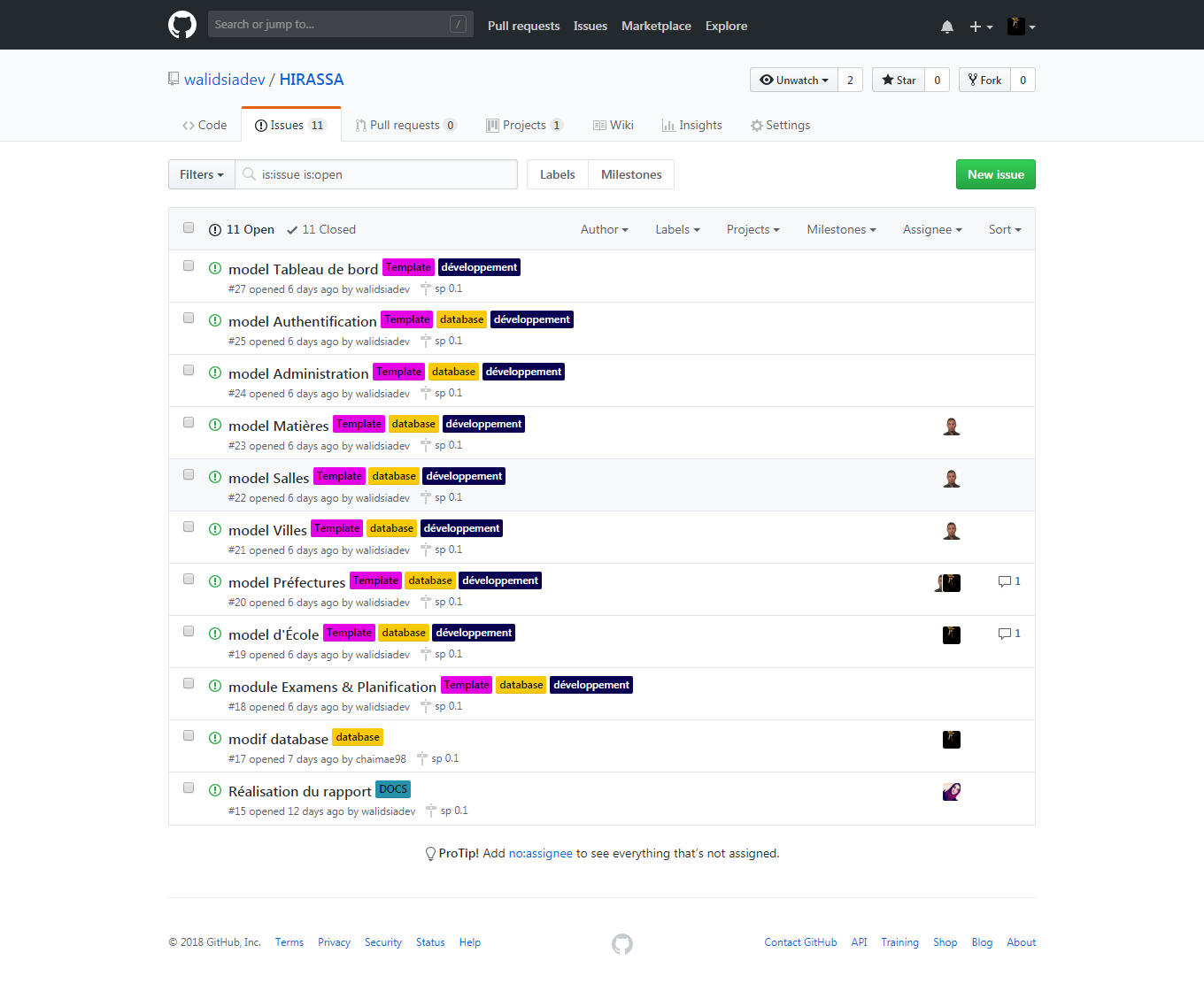
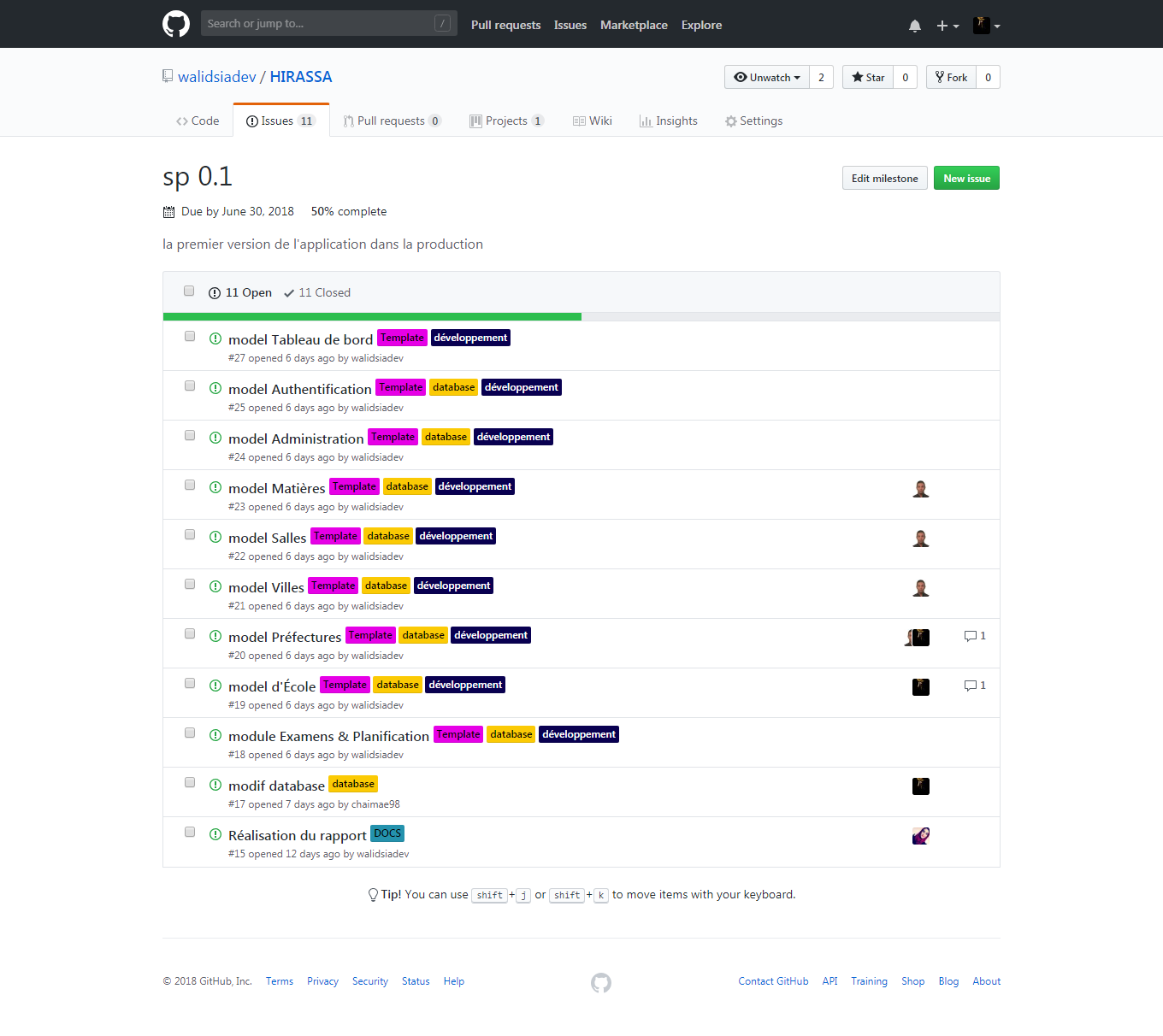
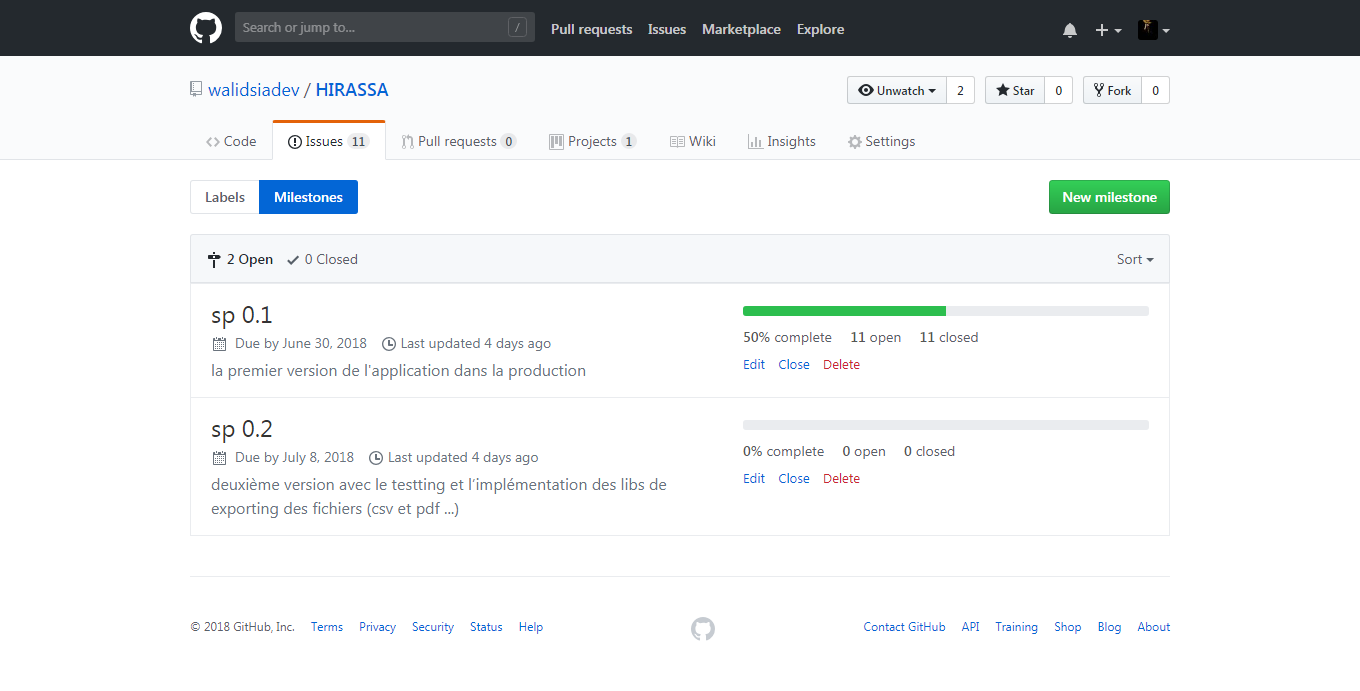
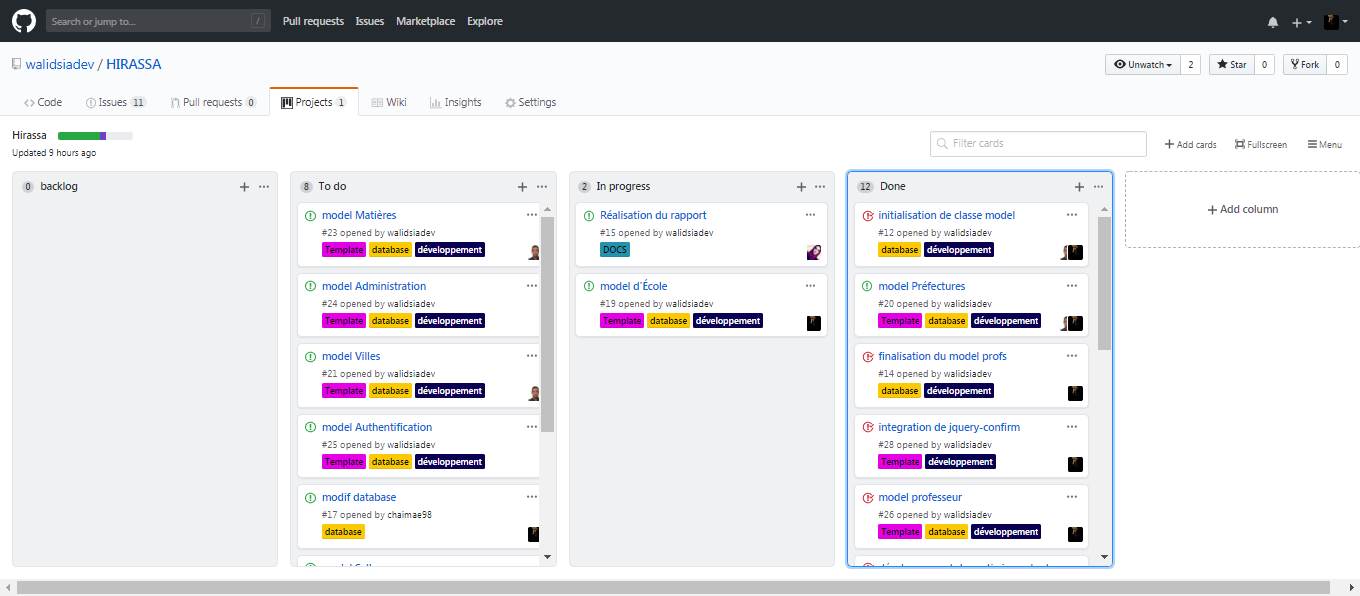
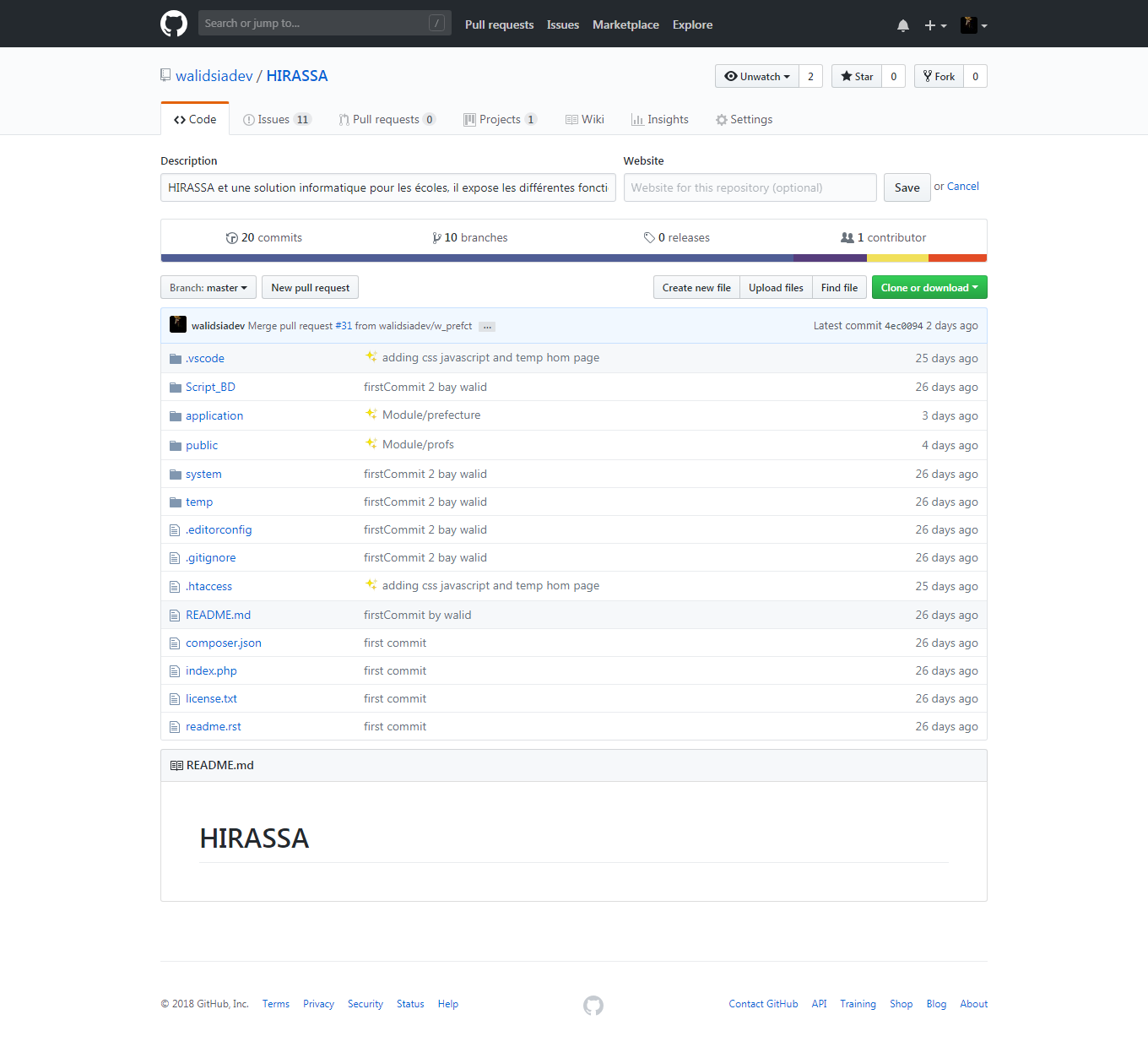
Gestion de base de données les plus utilisés au monde3, autant par le grand public

(applications web principalement) que par des professionnels, en concurrence avec Oracle, Informixet Microsoft SQL Server.  
Son nom vient du prénom de la fille du cocréateur Michael Widenius, My. SQL fait référence au Structured query Langage, le langage de requête utilisé.  
Mysql ab a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et My Sql. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.  
Depuis mai 2009, son créateur Michael Widenius a créé Maria Db (Maria est le prénom de sa deuxième fille) pour continuer son développement en tant que projet Open Source.

**GitHub**

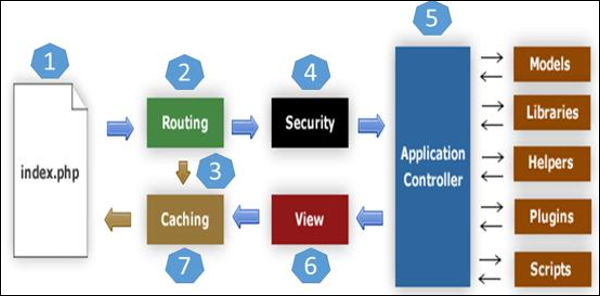
 GitHub est centré vers l'aspect social du développement. En plus d'offrir l'hébergement de projets avec Git, le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les dépôts (en anglais repository). GitHub offre aussi la possibilité de créer un wiki et une page web pour chaque dépôt. Le site offre aussi un logiciel de suivi de problèmes (de l'anglais issue tracking system). GitHub propose aussi l'intégration d'un grand nombre de services externes, tels que l'intégration continue, la gestion de versions, badges, chat basés sur les projets, etc.



****

**Codeigniter**

CodeIgniter est un framework libre écrit en PHP. Il suit le motif de conception MVC et s'inspire du fonctionnement de Ruby on Rails. Les versions inférieures à la 2.0.0 sont compatibles avec PHP 4 et 5, tandis que celles supérieures à la 2.0.0 ne sont compatibles qu'avec PHP 5.1.6 ou plus. La version 3.0 requiert PHP 5.2.4 et la version 3.1 requiert PHP 5.3.7.

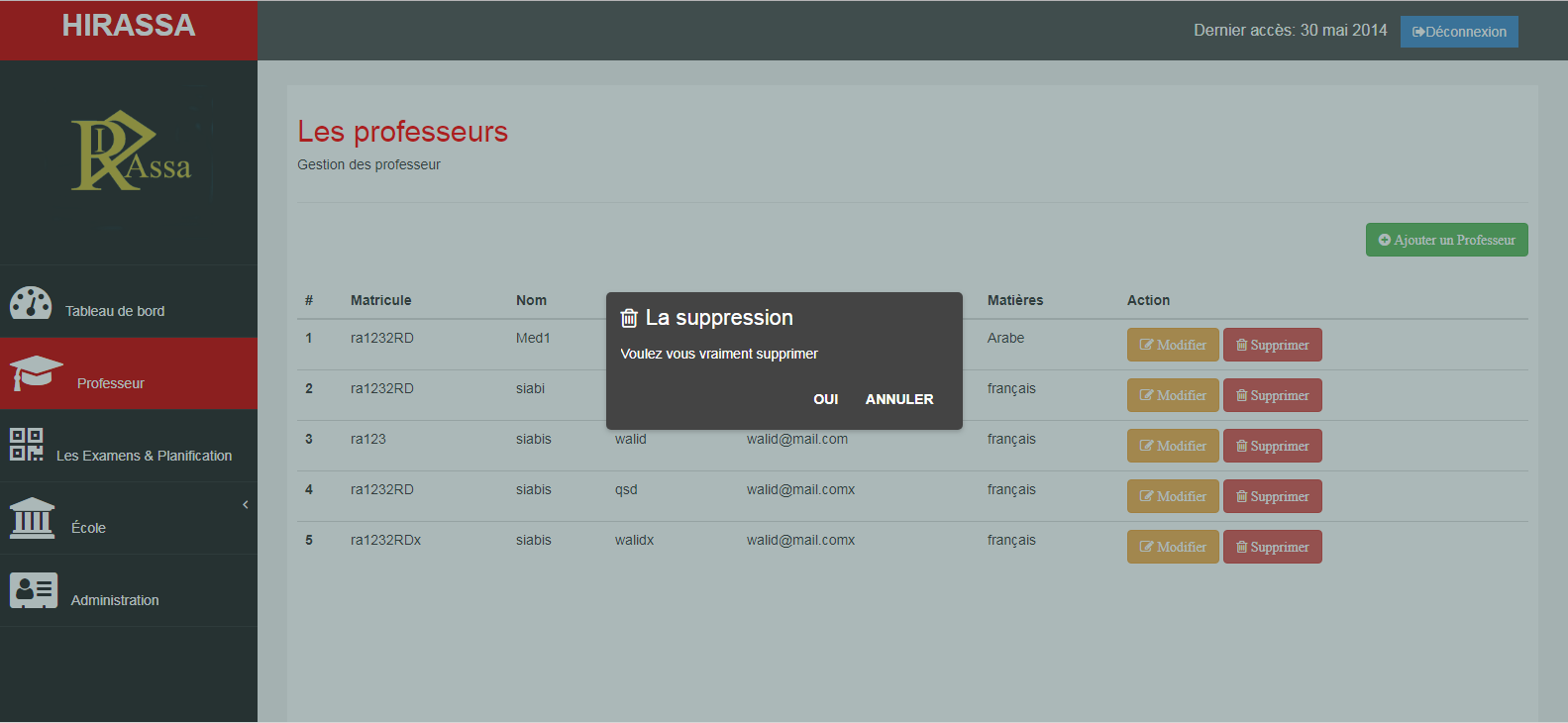
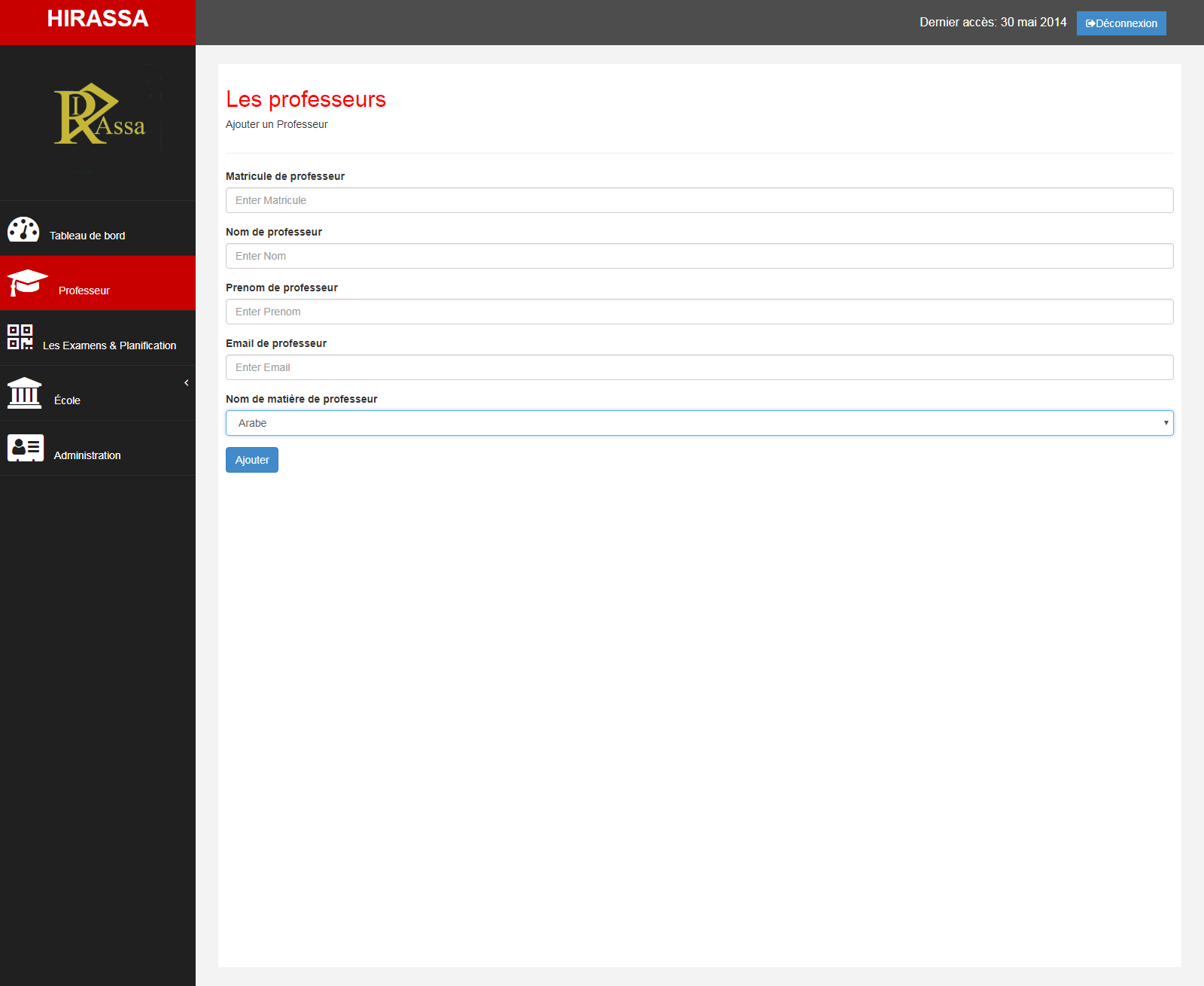
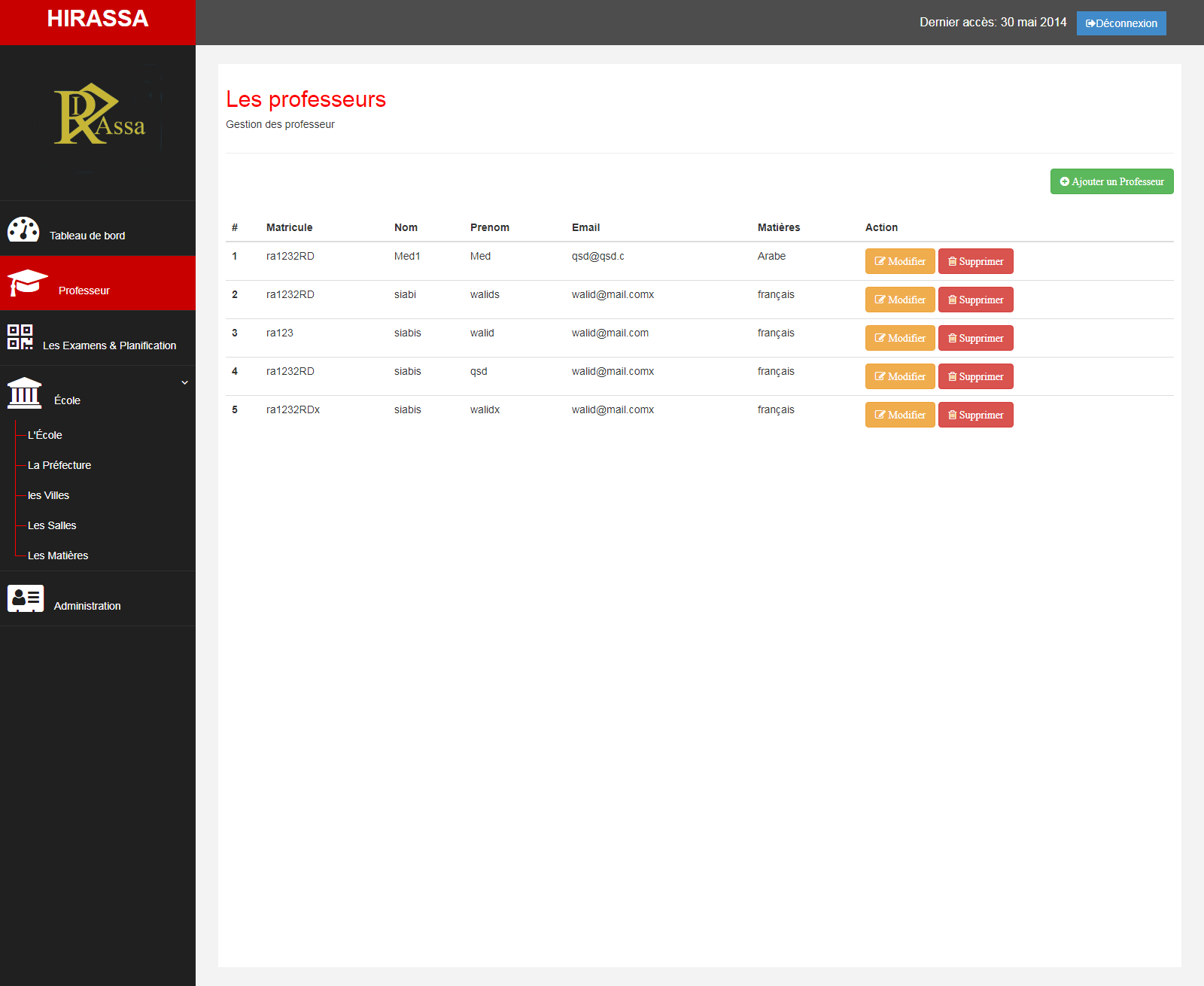


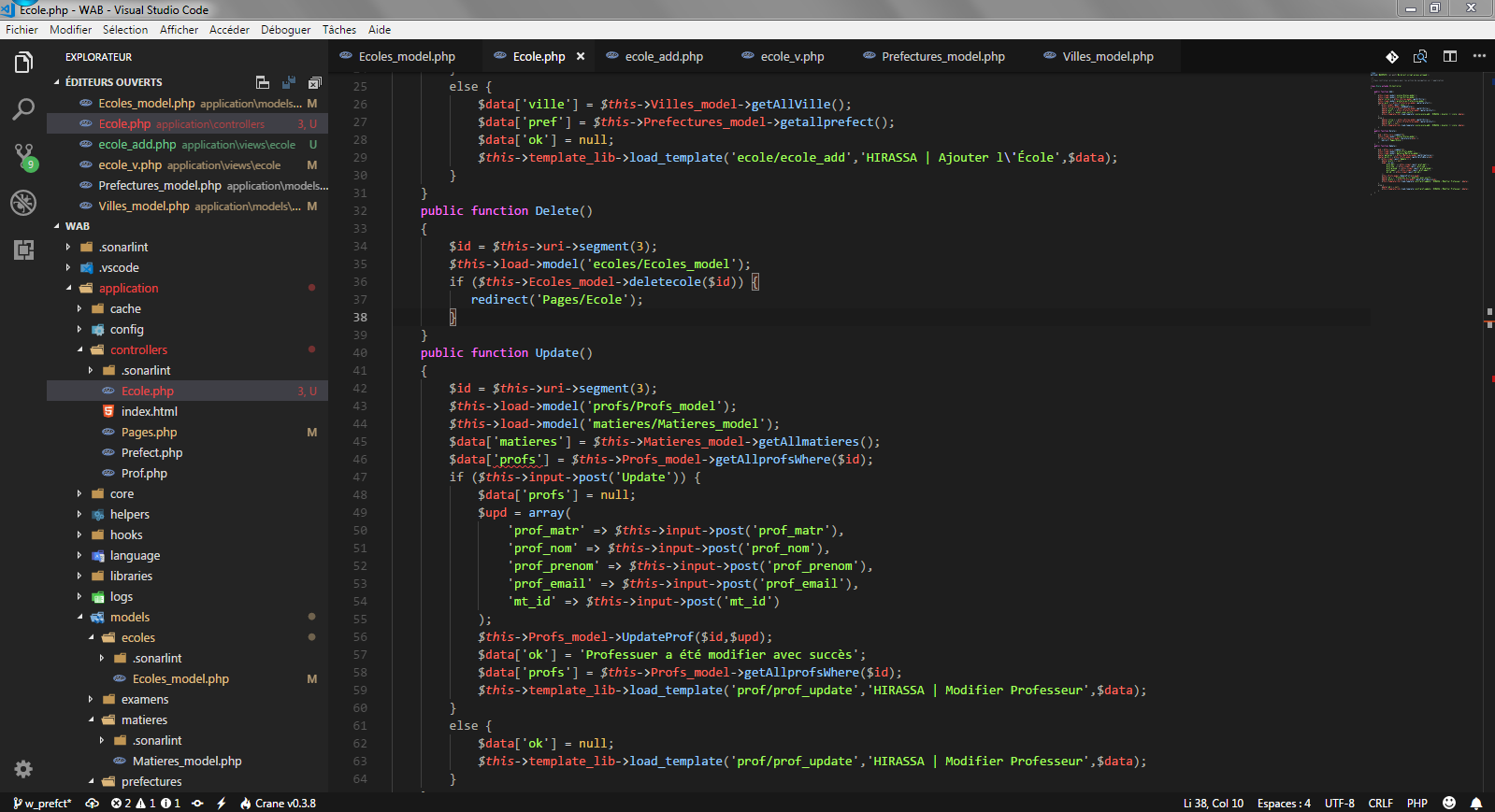
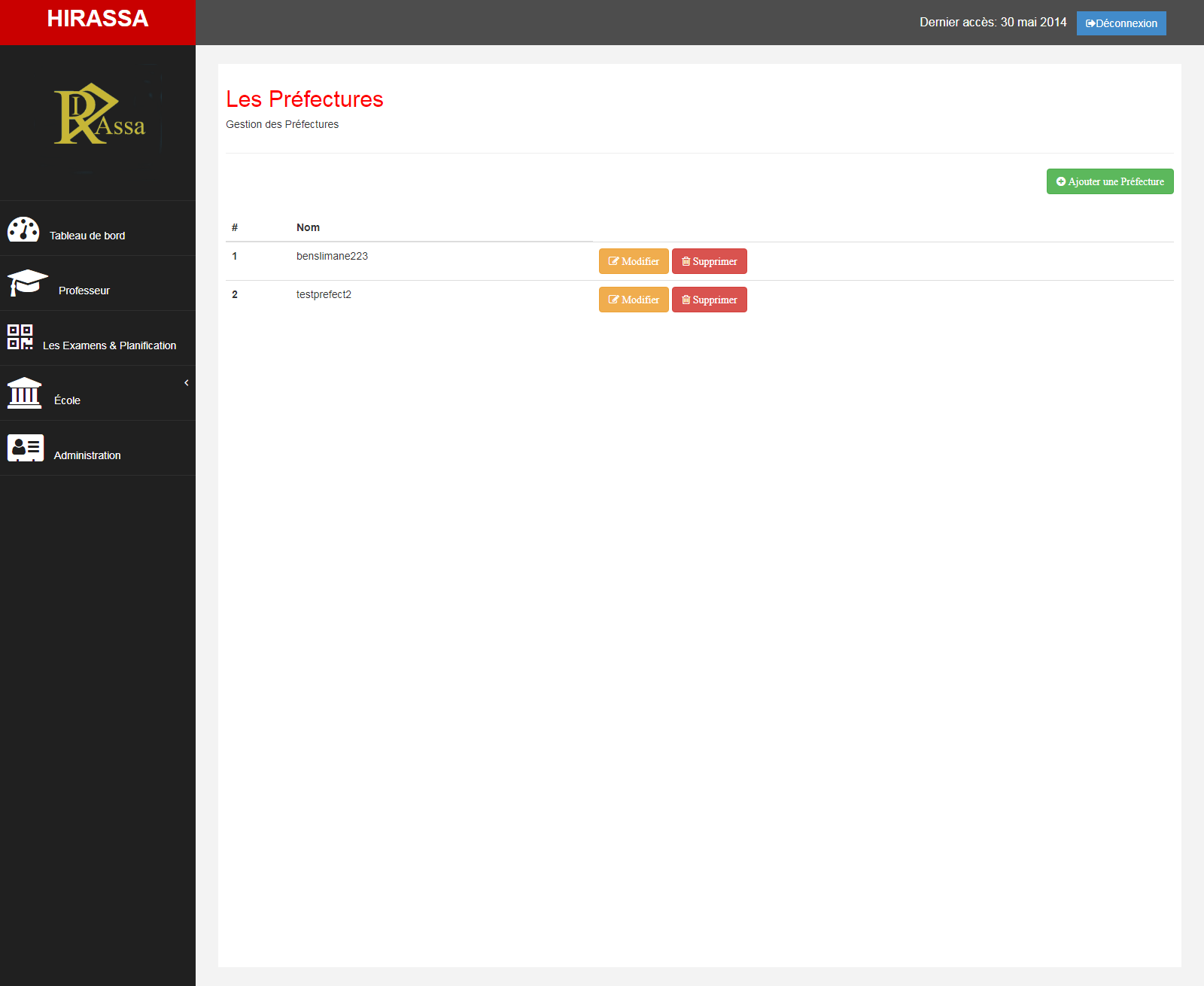
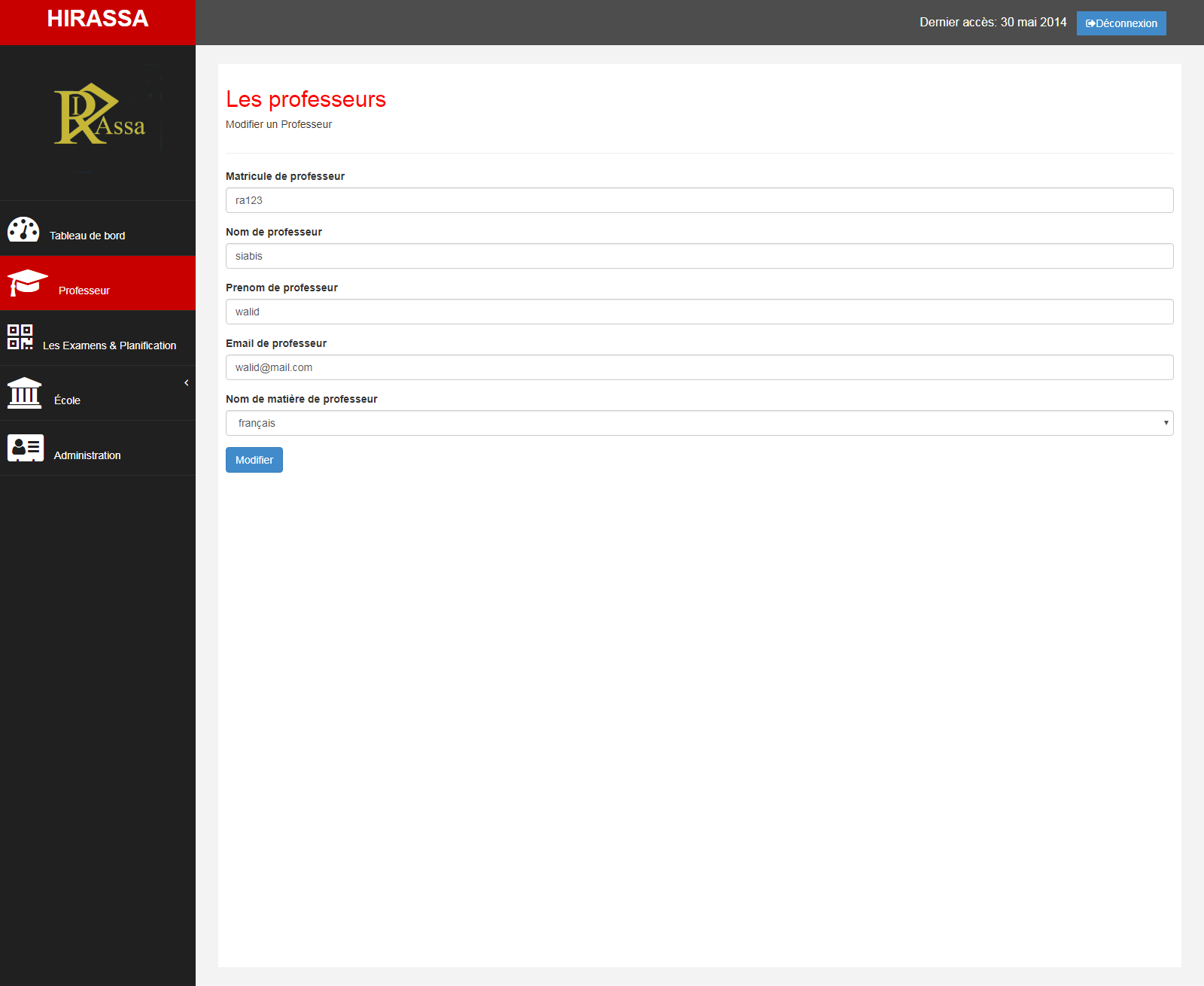
**Figure 8 : Codeigniter Framework**

**CHAPITRE 4 :**

**REALISATION ET MISE EN OEUVRE.**

**DEVELOPPEMENT DE L’APPLICATION**





**CONCLUSION**

Cette formation en sein de **CIGMA** pour obtenir notre licence en **Administration Base De Données Et Technologies Web**, a été pour nous l’occasion de connaître le milieu du développement professionnel,

Ce projet avait pour objectif la réalisation de l'application hirassa qui a mis une solution pour gérer le planning des examens ainsi que la distribution des profs et la gestion des sales pour les examens.

Cette période du PFE nous a permis d’approfondir nos connaissances théoriques acquises le long de notre formation par la pratique des nouvelles technologies.

L’exigence dans la qualité des projets développés et aussi les freins concernant le choix des technologies. Le travail en binôme est très productif puisque le projet est divisé en parties, et chaque intervenant peut se concentrer sur la tâche qui lui est affectée ; en revanche cela exige une très bonne communication entre nous deux pour éviter la répétitivité de certaines parties ou le non cohésion de l’ensemble du projet.

Enfin, je souhaite que ce rapport soit en mesure de satisfaire ses lecteurs et récolter ainsi leurs félicitations.

# FIN

**Références**

# Les définitions utilisé dans OUTILS DE MISE ON ŒUVRE UTILISES :

# Sur Wikipedia <https://www.wikipedia.org/>

# <https://git-scm.com/>

# <https://github.com/about>

# <https://codeigniter.com/docs>

# <https://code.visualstudio.com/>