**D I P T R A C K**

**Rapport de conception du projet**

Corentin MAPELLA & Glodie TANDU

1ère Année Master MIAGE Apprentissage

2016 - 2017



Ce document comporte le cahier des charges du projet DIPTRACK. Il inscrit les objectifs et les besoins attendus de l’application ainsi qu’une présentation des fonctionnalités, les spécifications et le détail des développements de celui-ci.

# **Information du projet :**

|  |  |
| --- | --- |
| **Nom du projet :** | DIPTRACK |
| **Groupe :** | Corentin MAPELLA – Glodie TANDU |
| **Version du document :** | 2.1.0 |
| **Statut :** | Amélioration du compte rendu |

# **Information groupe :**

|  |  |
| --- | --- |
| **NOM** | **RÔLE** |
| Corentin MAPELLA | Etude des technologies  Rédaction et mise en forme du compte rendu  Développement du logiciel |
| Glodie TANDU | Etude des technologies  Rédaction et mise en forme du compte rendu  Développement du logiciel |

# **Révision du document :**

|  |  |  |
| --- | --- | --- |
| **VERSION** | **NOM** | **DESCRIPTION** |
| 1.0 | gtandu – cmapella | Création du document |
| 1.1 | gtandu – cmapella | Rédaction de la *Présentation du Sujet* |
| 1.2 | gtandu – cmapella | Ajout des maquettes et des diagrammes de cas d’utilisations |
| 1.3 | gtandu – cmapella | Ajout des diagrammes détaillés |
| 1.4 | gtandu – cmapella | Ajout d’un diagramme des classes |
| 1.5 | gtandu – cmapella | Ajout des diagrammes de séquences |
| 1.6 | gtandu – cmapella | Ajout des diagrammes d’états et de paquetage |
| 1.7 | gtandu – cmapella | Correction des fautes d’orthographe  Modification du diagramme de classe  Ajout des premiers jets de l’application |
| 2.0 | gtandu – cmapella | Mise à jour du compte rendu |
| 2.1 | gtandu – cmapella | Modification du diagramme de classe  Amélioration du contenu |
| 2.2 | gtandu – cmapella | Ajout de l’implémentation |

**Sommaire**

[I- Présentation du Sujet 5](#_Toc485253147)

[1. Contexte 5](#_Toc485253148)

[2. Description de l’existant 5](#_Toc485253149)

[3. Objectifs 5](#_Toc485253150)

[4. Besoins fonctionnels 6](#_Toc485253151)

[5. Contraintes fonctionnelles 7](#_Toc485253152)

[6. Environnement de développement et langages 7](#_Toc485253153)

[a. Framework J2EE 7](#_Toc485253154)

[b. Bases de données 7](#_Toc485253155)

[c. L’environnement de développement 7](#_Toc485253156)

[d. Outils collaboratifs 8](#_Toc485253157)

[7. Maquettes 8](#_Toc485253158)

[a. Le portail d’authentification 9](#_Toc485253159)

[b. La page d’accueil d’un élève 9](#_Toc485253160)

[c. La page d’accueil d’un professeur 10](#_Toc485253161)

[d. La page contenant les notes de l’élève 10](#_Toc485253162)

[e. Une page des pages de matière vu par un professeur 11](#_Toc485253163)

[f. Formulaire de création de filière 11](#_Toc485253164)

[g. Formulaire de création d’une matière 12](#_Toc485253165)

[h. Formulaire de création d’une classe 12](#_Toc485253166)

[8. Spécifications Techniques et Fonctionnelles 13](#_Toc485253167)

[a. Diagramme de cas d’utilisations 13](#_Toc485253168)

[b. Présentation du périmètre fonctionnel de l’application 13](#_Toc485253169)

[c. Détails des cas d’utilisations généraux 14](#_Toc485253170)

[d. Détails des cas d’utilisations pour l’administrateur 15](#_Toc485253171)

[e. Détails des cas d’utilisations pour le responsable de filière 16](#_Toc485253172)

[f. Détails des cas d’utilisations pour l’enseignant 19](#_Toc485253173)

[g. Détails des cas d’utilisations pour l’étudiant 20](#_Toc485253174)

[9. Diagramme de classe 21](#_Toc485253175)

[10. Diagramme de Séquence 22](#_Toc485253176)

[h. Connexion 22](#_Toc485253177)

[i. Création d’une filière 23](#_Toc485253178)

[j. Création d’une matière 24](#_Toc485253179)

[k. Création d’une classe 24](#_Toc485253180)

[l. Publier les notes 25](#_Toc485253181)

[11. Diagramme d’État 25](#_Toc485253182)

[a. Connexion à compte 25](#_Toc485253183)

[b. Validation d’une matière 26](#_Toc485253184)

[12. Diagramme de paquetage 26](#_Toc485253185)

[II- Implémentation 27](#_Toc485253186)

[1. Interface Homme-Machine 27](#_Toc485253187)

[a) Page sans authentification 27](#_Toc485253188)

[c. Création et/ou réinitialisation d’un mot de passe 28](#_Toc485253189)

[b) Interface Administrateur 28](#_Toc485253190)

[c) Interface Enseignant 31](#_Toc485253191)

[d) Interface Responsable 32](#_Toc485253192)

[e) Autres pages 33](#_Toc485253193)

[2. Modèle Conceptuel de Données 35](#_Toc485253194)

[3. Fonctionnalités développées et à développer 36](#_Toc485253195)

[4. Difficultés rencontrés / Solutions trouvées 37](#_Toc485253196)

[5. Changements apportés par rapport à la conception 37](#_Toc485253197)

[6. Répartition du travail 38](#_Toc485253198)

# Présentation du Sujet

## Contexte

Dans le cadre de notre première année MIAGE Apprentissage, nous avons à développer une application. Il nous a été confié la création d’une **plateforme de gestion du parcours universitaire** des étudiants au sein de la MIAGE.

La mise en place du projet nommé « DIPTRACK » s’effectuera en trois phases :

* Une présentation du sujet, qui comporte une description du sujet pour confirmer la bonne compréhension du sujet (à rendre le **24 Novembre**)
* Un rapport intermédiaire, qui décrit la phase de conception du projet (à rendre le **6 Mars**)
* Un rapport final, qui débouche sur le développement du système spécifié (à rendre le **15 juin**)

## Description de l’existant

A l’heure actuelle, l’administration utilise un progiciel nommé « APOGEE » destiné à la gestion des inscriptions et des dossiers des étudiants dans les universités françaises.

Cette application est composée de plusieurs modules tels que :

* **Dossier Étudiant** : Les dossiers étudiants qui réunit les données administratives des étudiants : cursus, situation comptable vis-à-vis de l’université, etc.
* **Contrôle des connaissances** : Le contrôle des connaissances inclue la saisie des barèmes, des coefficients, des règles de calcul, des notes et de résultat pour les UE ou la matière générale.
* **Résultat** : Les résultats qui se composent de la saisie des notes, calcul automatique des notes, classements des étudiants, le calcul des ECTS, etc.

Ce progiciel, comporte plusieurs contraintes :

* L’administration est le seul service qui a la capacité d’insérer les notes des étudiants
* Lorsque les notes de tous les étudiants pour une matière donnée sont introduites dans Apogée, l’enseignant responsable de la matière n’est pas notifié.
* Apogée n’est pas multiplateforme.
* En cas de panne du système, les utilisateurs n’ont plus accès aux informations.
* Les enseignants n’ont pas accès à l’application.

## Objectifs

DIPTRACK est une application web et mobile, qui a pour vocation de **permettre la simulation d’obtention du diplôme**.

Il résulte d’une problématique soulevée par le corps enseignant qui n’a pas accès au logiciel Apogée. Ils sont donc dans l’incapacité de visualiser toutes les notes d’un étudiant, d’une classe ou même de les insérer eux-mêmes dans le système.

Cette application a donc pour but de permettre au corps enseignant d’avoir un support en ligne, regroupant les notes des étudiants de leurs classes et d’y insérer les notes de leurs matières.

Elle permettra au corps enseignant d’avoir une visibilité sur les notes d’une classe ou d’un étudiant en particulier.

De plus, les étudiants pourront aussi accéder à l’application pour visualiser leurs notes en cours.

## Besoins fonctionnels

Pour définir les besoins ce projet, nous avons dans un premier temps, recueilli les principales requêtes de Madame Benzakki, représentant l’administrateur et l’enseignant.

Dans un second temps, nous avons sollicité nos collègues étudiants pour prendre en compte leurs attentes.

Il en découle quatre principaux acteurs de l’application : l’administrateur, le responsable de filière, l’enseignant et l’étudiant.

Chacun de ses acteurs possèdent des capacités qui lui sont propres.

Dans le cadre d’une démarche agile, ces capacités ont été déclinées en user-story.

**L’administrateur**

* + Gère les comptes utilisateurs.

**Le responsable de filière**

* + Définit le contenu d’une filière.
  + Créé les classes des filières.
  + Définit les règles de compensation pour l’obtention du diplôme.

(UE, Semestre, Session)

* + Modifie le procès-verbal.
  + Définit le MCC (Mode de contrôle des connaissances).

**L’enseignant**

* + Insère les notes d’une classe pour sa ou ses matières.
  + Consulte les notes de chaque étudiant de sa classe.
  + Consulte la moyenne d’une classe dans sa matière.
  + Est notifié lorsqu’un étudiant à une note éliminatoire.
  + Peut valider sa saisie de note afin de la terminer plus tard.
  + Publie les notes une fois sa saisie terminée et notifie les étudiants et les enseignants concernés.
  + Visualise le procès-verbal.
  + Extrait le procès-verbal au format PDF et/ou Excel.
  + Importe un fichier Excel correspondant aux notes des étudiants d’une classe.

**L’étudiant**

* + Peut consulter sa fiche étudiante.
  + Peut visualiser son relevé de note.
  + Peut simuler l’obtention du diplôme.

## Contraintes fonctionnelles

Il faut aussi prendre en compte les contraintes liées aux règles d’attribution d’un diplôme : MCC (Modalités du Contrôle de Connaissance).

En effet, certaines matières ne sont pas rattrapables. Nous pouvons prendre l’exemple du projet qui est éliminatoire si la note obtenu est inférieure au seuil de compensation. De plus, les enseignants ne peuvent modifier que leurs propres matières. Enfin, la version mobile ne permettra que la consultation des notes.

## Environnement de développement et langages

### Framework J2EE



Le projet est développé en J2EE à l’aide du Framework **Spring**.

Étant un Framework est très populaire et répandu dans le monde de l’entreprise, il nous a semblé judicieux d’utiliser cette technologie afin d’accroître nos compétences sur le sujet. Son principal avantage est de permettre un développement rapide d’une application web grâce à une multitude de module tels que **Spring Web** ou **Spring MVC**.

Spring nous permettra de rapidement mettre en place une application web notamment grâce à **Spring Boot**[[1]](#footnote-1).

**Spring Boot** a pour but de faciliter la configuration d’un projet Spring. Le site web <https://projects.spring.io/spring-boot/> permet générer la structure d’un projet en y incluant toutes les dépendances Maven que l’on souhaite.

### Bases de données



Concernant la base de données, nous avons opté pour un système de gestion de base de données MySQL, étant libre d’accès et simple à mettre en place.

Nous utilisons une plateforme de développement de type **WAMP** [[2]](#footnote-2)sous Windows et **MAMP** [[3]](#footnote-3)sous Mac OS.

### L’environnement de développement



L’environnement de développement sera **Eclipse**, qui est la référence en termes de développement Java.

Le développement d’une application robuste, nécessite d’être testé unitairement. Nous avons décidé d’utiliser **JUnit** associé au Framework **Mockito** qui permet de simuler le comportement des objets réels tels que la base de données, un web service etc.

Afin de faciliter le passage entre le code principal et le code de test, nous utilisons le plugin **MoreUnit**.

Il permet d’ajouter de nombreux raccourcis fluidifiant la navigation entre les méthodes testées et réelles.

Pour assurer une couverture optimale, nous avons opté pour le Plugin **ElcEmma** qui se combine parfaitement à **JUnit**. Il permet d’obtenir des métriques quant au pourcentage des classes, méthodes qui sont testés. Il donne donc une vision d’ensemble de la couverture de code testé du projet.

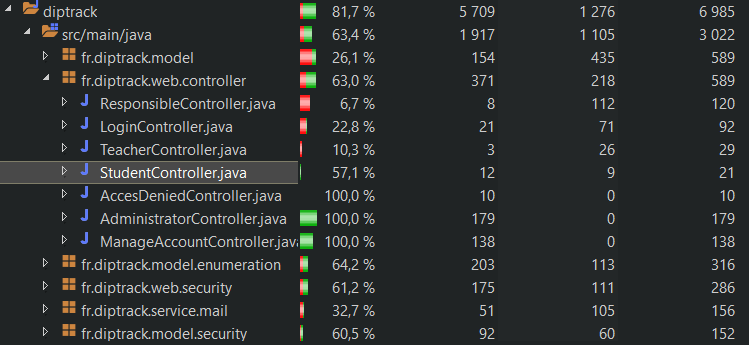
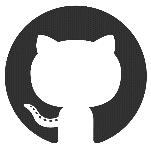
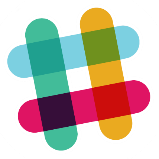


Figure 1 - Coverage du Projet

### Outils collaboratifs



Afin d’améliorer la communication et de faciliter les échanges, nous utilisons **GitHub** qui va nous permettre de travailler à plusieurs sur le projet et de gérer les versions du projet.



De plus, Madame Benzakki nous a suggéré d’utiliser **Slack** qui permet de communiquer plus facilement entre l’enseignant et les étudiants ou encore de partager des documents. Il permet aussi d’intégrer des plugins tel que GitHub.

## Maquettes

Afin d’illustrer l’agencement de l’interface web, nous avons créé des maquettes à l’aide de l’application **Balsamiq Mockups 3**, un logiciel de conception de croquis.

### Le portail d’authentification

Le professeur ou l’élève va s’authentifier et va être rediriger vers sa page d’accueil.

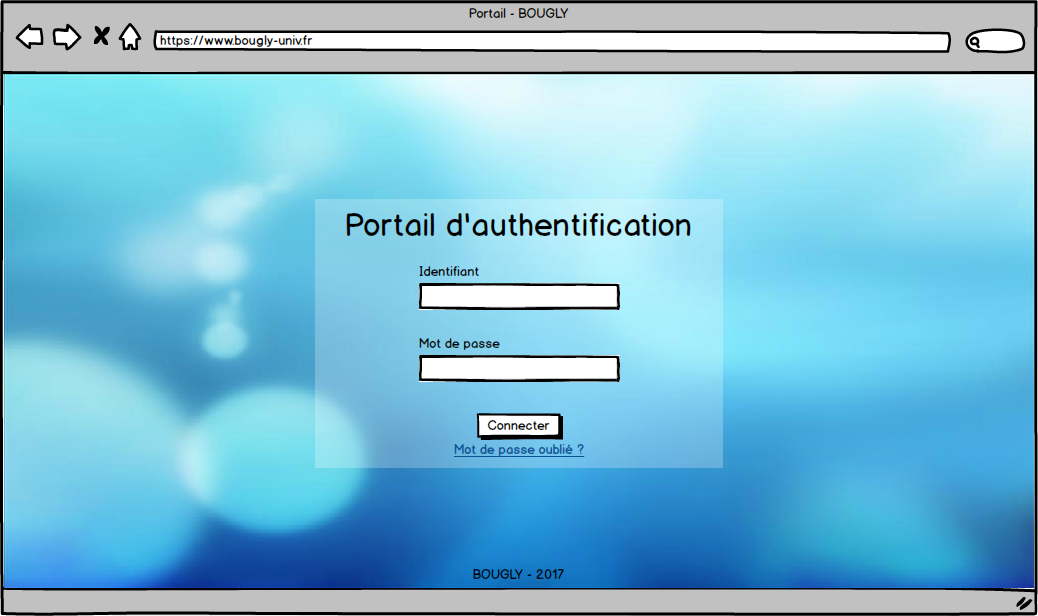


Figure 2 - Portail d'authentification

### La page d’accueil d’un élève

**Pour l’élève**, sa page d’accueil sera composée du libellé de sa classe, d’un graphique contenant les notes de toutes les matières et d’autres informations.

Il pourra accéder à ses messages, au paramétrage de son compte et à ses notes.

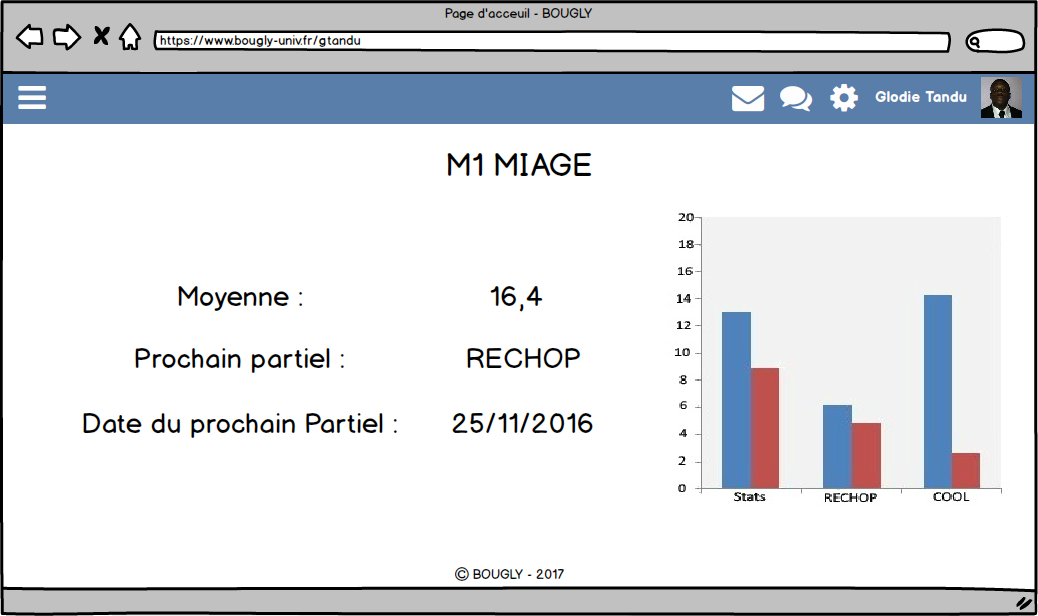


Figure 3 - Page d’accueil d’un élève

### La page d’accueil d’un professeur

**Pour les professeurs**, sa page d’accueil sera composée de toutes ses classes et des mêmes raccourcis que l’élèves.

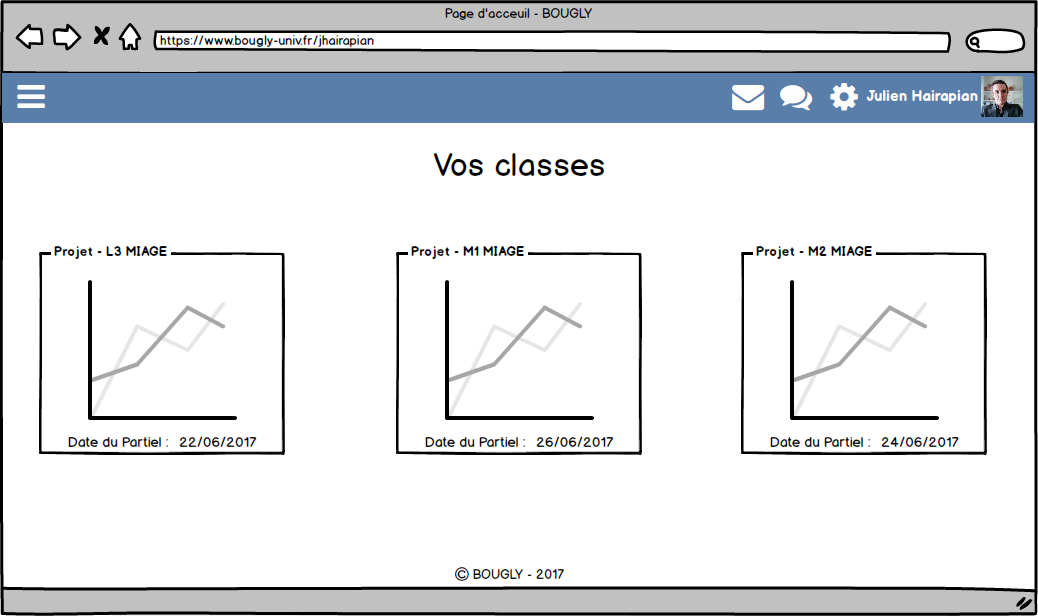


Figure 4 - Page d’accueil d’un enseignant

### La page contenant les notes de l’élève

**Pour l’élève**, il aura à disposition une page réunissant ses notes où il pourra simuler sa moyenne et savoir s’il obtient son diplôme. De plus, il pourra en extraire le tableau en PDF.

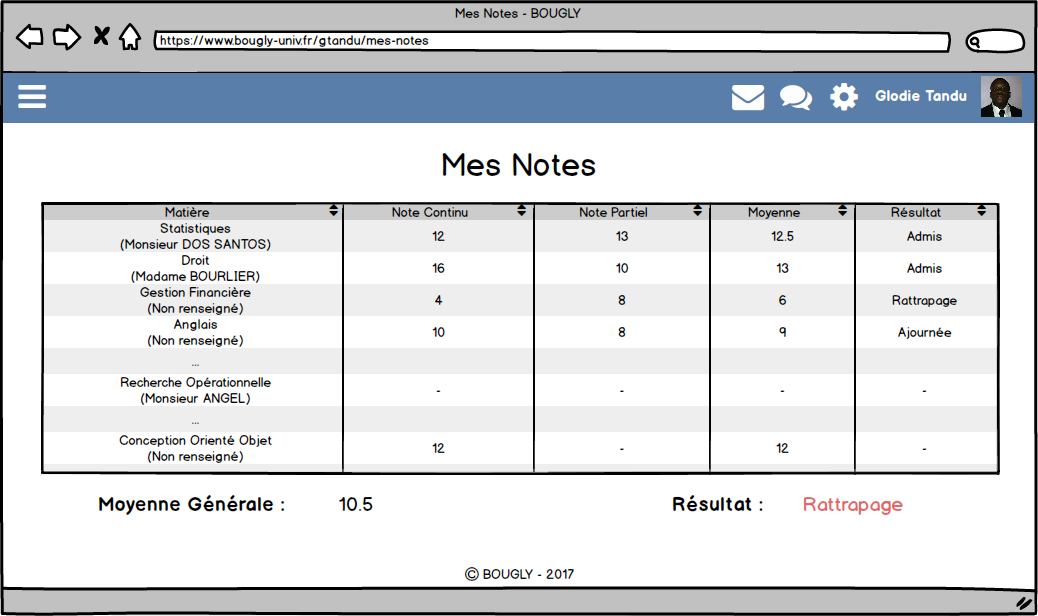


Figure 5 - Page d'accueil d'un étudiant

### Une page des pages de matière vu par un professeur

**Pour le professeur**, il pourra rentrer les notes de ses élèves et extraire le tableau en PDF.

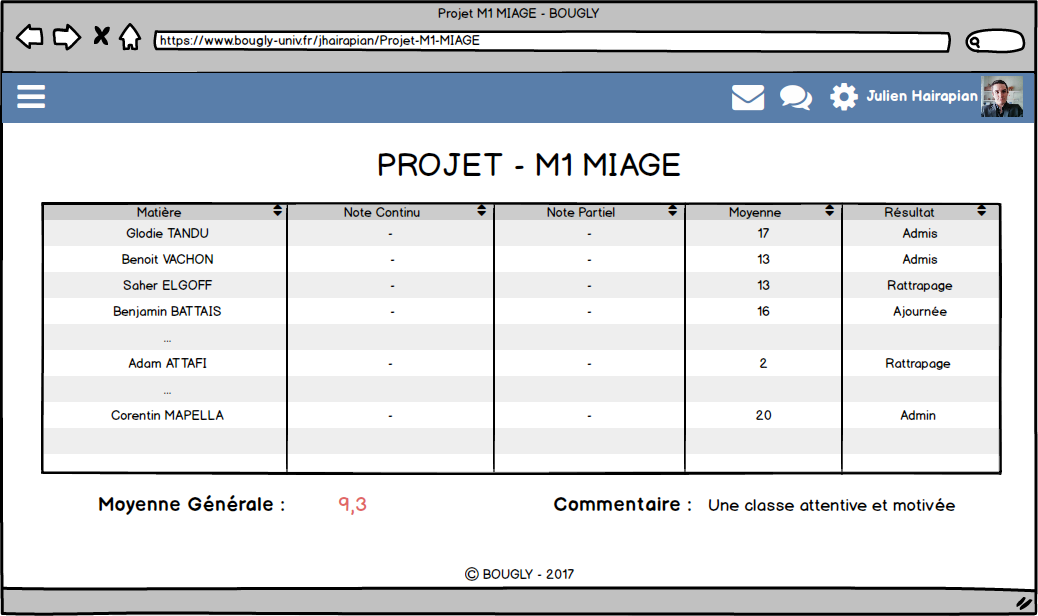


Figure 6 - Page d’une classe d’un enseignant

### Formulaire de création de filière

L’administrateur peut utiliser le formulaire suivant afin de créer une nouvelle filière.

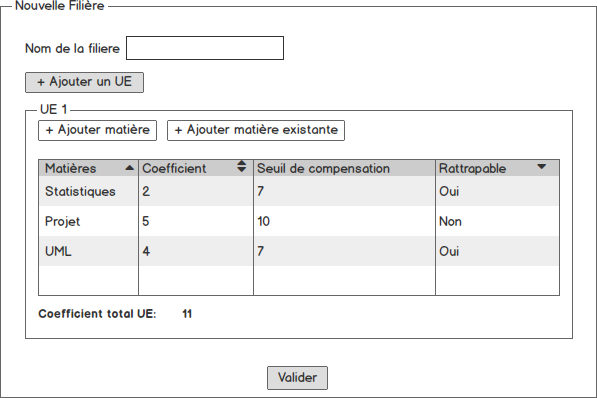


Figure 7 - Formulaire de création

### Formulaire de création d’une matière

L’administrateur peut utiliser le formulaire suivant afin de créer une nouvelle filière.

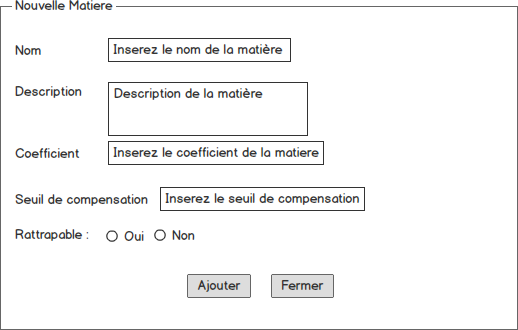


Figure 8 - Formulaire d’une matière

### Formulaire de création d’une classe

L’administrateur peut utiliser le formulaire suivant afin de créer une nouvelle classe.

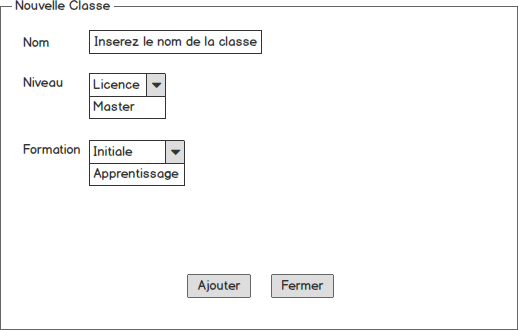


Figure 9 - Formulaire de création d’une classe

## Spécifications Techniques et Fonctionnelles

### Diagramme de cas d’utilisations

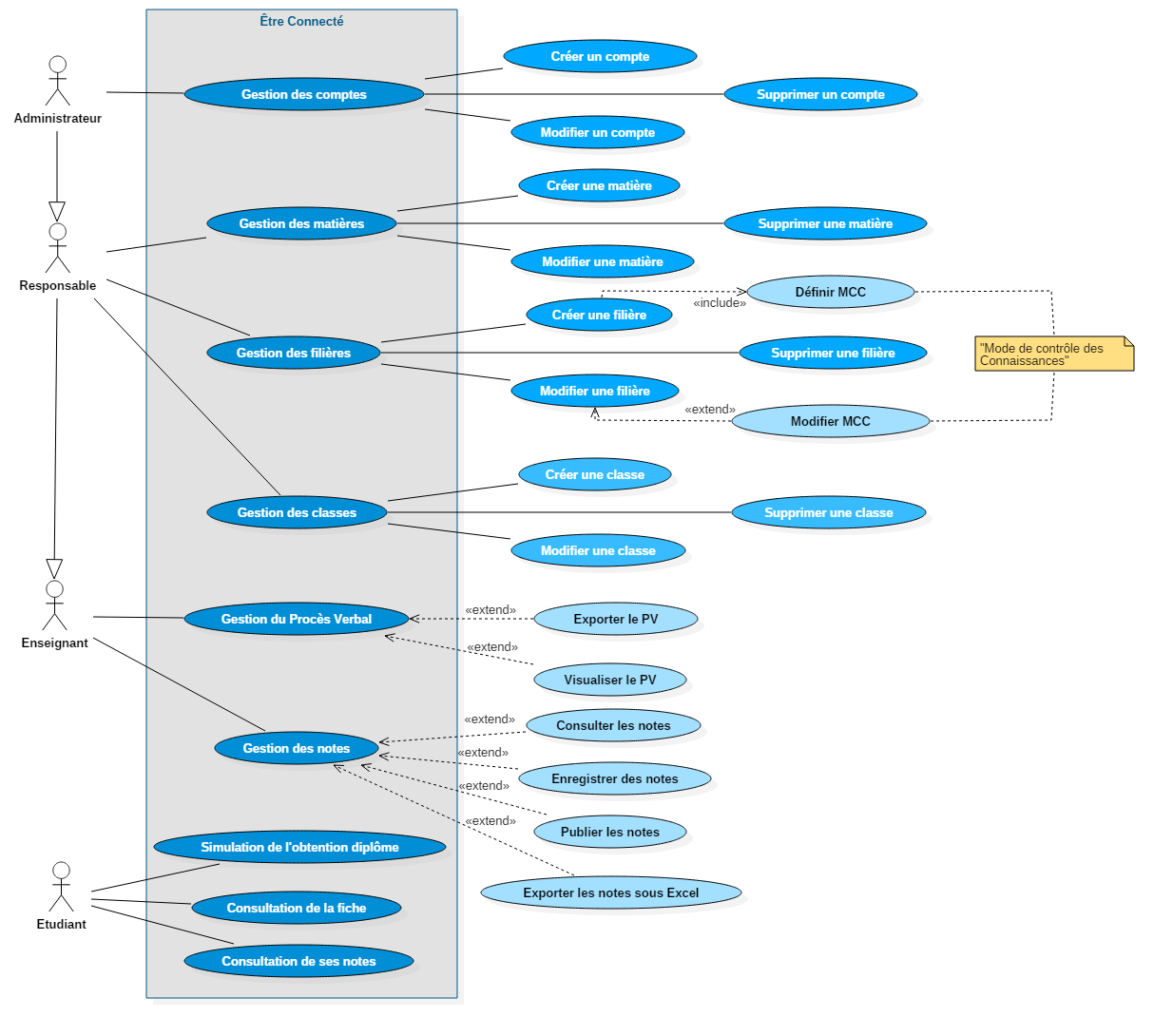


Figure 10 - Diagramme de cas d’utilisations général

### Présentation du périmètre fonctionnel de l’application

Nous comptons 4 cas d’utilisations au sein de l’application :

* ***Cas Administrateur***
  + Gestion des comptes : Il pourra créer, modifier, supprimer les comptes de chaque acteur. Ce processus est primordial sachant qu’il structure l’ensemble des comptes.
* ***Cas Responsable***
  + Gestion des filières : Il s’occupe de créer, modifier, supprimer les filières. Cette étape permet de définir le mode de contrôle des connaissances pour chaque filière.
  + Gestion des classes : Il s’occupe de créer, modifier, supprimer les classes. Dès lors, les étudiants et les enseignants pourront être affiliés à des classes.
* ***Cas Enseignant***
  + *Gestion du Procès-Verbal :* Le procès-verbal est un récapitulatif complet.
  + *Gestion des notes :* L’enseignant pourra ajouter, supprimer et modifier des notes. Il lui sera possible d’enregistrer la saisie et d’y revenir plus tard dessus. Une fois terminé, il pourra publier les notes (Un mail sera diffusé aux acteurs concernés).
* ***Cas Etudiant***
  + *Simulation de l’obtention diplôme : L’élève a la possibilité de compléter la grille des notes et d’y ajouter des notes fictives pour simuler l’obtention de son année.*
  + *Consultation de la fiche :* Il y constatera son profil et pourra faire les modifications si besoin.
  + *Consultation de ses notes :* Un récapitulatif des notes sera disponible avec l’ensemble des matières de l’étudiant.

### Détails des cas d’utilisations généraux



Figure 11- Connexion à un compte

#### Intitulé

* Se connecter à son compte

#### Acteur

* Administrateur
* Responsable
* Enseignant
* Etudiant

#### Données en entrée

* L’utilisateur doit posséder une adresse mail et un mot de passe.

#### Scénario

1. Le système affiche la fenêtre d’authentification.
2. L’utilisateur saisit son adresse mail et son mot de passe.
3. L’utilisateur s’authentifie et est redirigé vers sa page d’accueil.

### Détails des cas d’utilisations pour l’administrateur

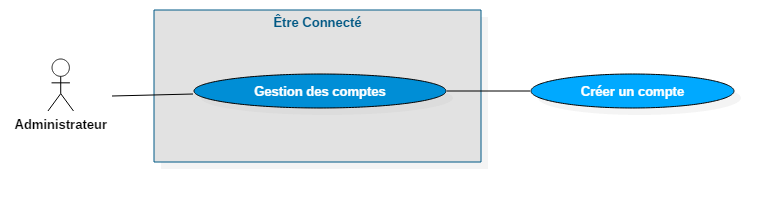


Figure 12- Création d’un compte

#### Intitulé

* Créer un compte utilisateur

#### Acteur

* Administrateur

#### Données en entrée

* L’administrateur est connecté à l’application.
* L’administrateur a à disposition un fichier Excel contenant les informations des étudiants (numéro étudiant, nom, prénom, classe, adresse mail).

#### Scénario Principal

1. Le responsable de filière crée une classe.
2. Le système affiche un écran avec deux boutons (importer une liste d’étudiants et ajouter un étudiant).
3. L’administrateur clique sur « importer une liste d’étudiants ».
4. L’administrateur sélectionne son fichier Excel au format xls.
5. Le système crée les différents comptes étudiants grâce aux données présentes dans le Excel.
6. Le système affiche les informations des étudiants ajoutés dans la classe.

#### Scenario Alternatif

1. Le responsable de la filière sélectionne la classe déjà existante à laquelle il va affecter le compte étudiant.
2. L’administrateur clique sur « ajouter un étudiant ».

### Détails des cas d’utilisations pour le responsable de filière

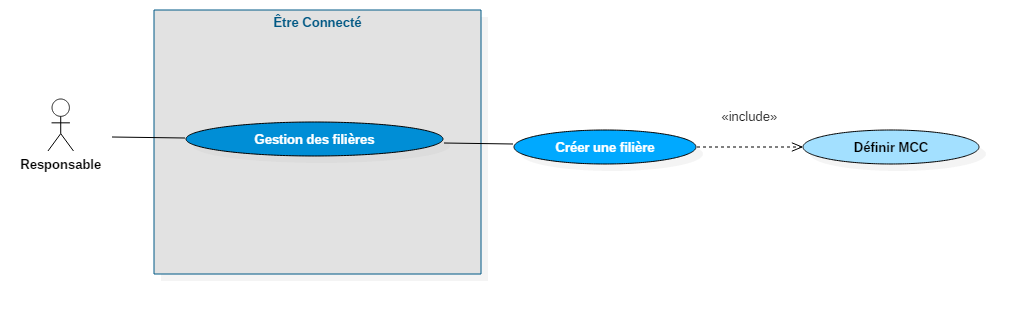


Figure 13- Création d’une filière

#### Intitulé

* Créer une filière

#### Acteur

* Responsable de filière

#### Données en entrée

* Le responsable de filière est connecté à l’application.

#### Scénario principal

1. Le système affiche la fenêtre de création d’une filière.
2. Le système grise le bouton « Valider » tant que le champ « nom de la filière » n’est pas correctement renseigné.
3. Le responsable de filière remplit le champ du nom de la filière.
4. Une fois le champ « nom de la filière » valorisé, le système rend disponible le bouton de validation.
5. Le responsable de filière crée un UE grâce au bouton prévu à cette effet.
6. Le système affiche un tableau contenant les matières de l’UE.
7. Le responsable crée une ou plusieurs matières.
8. Le système alimente le tableau des matières de l’UE à chaque création.
9. Le responsable de filière défini le MCC pour chaque matière.
10. Le responsable de filière valide la création de la filière.

#### Scénario alternatif

1. Le responsable ajoute une ou plusieurs matières déjà existantes.

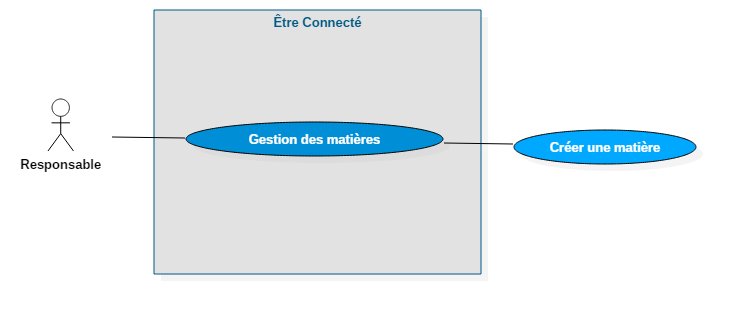


Figure 14 - Création d’une matière

#### Intitulé

* Créer une matière

#### Acteur

* Responsable de filière

#### Données en entrée

* Le responsable de filière est connecté à l’application.

#### Scénario

1. Le système affiche la fenêtre de création d’une matière.
2. Le système grise le bouton valider tant que tous les champs ne sont pas valorisés correctement.
3. Le responsable de filière remplit les champs nom, description, le coefficient de la matière, le seuil de compensation et coche un bouton radio pour savoir si la matière est rattrapable ou non.
4. Une fois tous les champs valorisés, le système rend disponible le bouton de validation.
5. Après avoir soumis la création de la matière, le système redirige le responsable de filière vers l’écran présentant les informations de la filière.

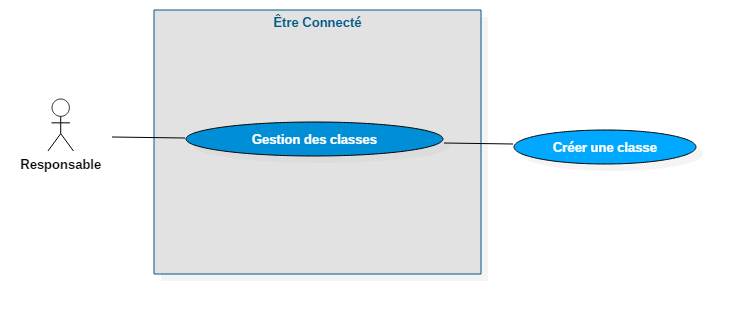


Figure 15 - Création d’une classe

#### Intitulé

* Créer une classe

#### Acteur

* Responsable de filière

#### Données en entrée

* L’administrateur est connecté à l’application.

#### Scénario Principal

1. Le système affiche la fenêtre de création d’une classe.
2. Le système grise le bouton « Valider » tant que le champ « nom de la classe » n’est pas correctement renseigné.
3. Le responsable de filière remplit le champ nom de la classe, choisit le niveau de la classe (Licence, Master) et le type de formation (Initiale, Apprentissage).
4. Une fois le champ « nom de la filière » valorisé, le système rend disponible le bouton de validation.
5. Le responsable de filière valide la création de la classe.

### Détails des cas d’utilisations pour l’enseignant

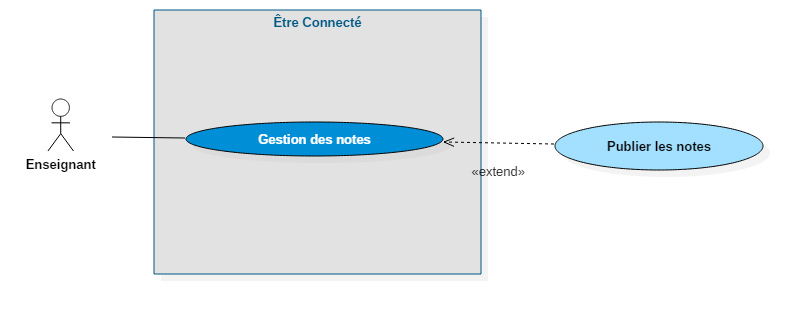


Figure 16 - Publication des notes

#### Intitulé

* Publier les notes d’une matière

#### Acteur

* Enseignant

#### Données en entrée

* L’enseignant doit avoir terminé la saisie des notes.

#### Scénario

1. Le système affiche la fenêtre de saisie des notes.
2. Le système grise le bouton « Publier » tant que toutes les notes ne sont pas renseignées.
3. Le responsable de filière insère les notes correspondant aux étudiants.
4. Une fois la totalité des notes saisies, le système rend disponible le bouton de publication.
5. L’enseignant peut alors publier les notes et un mail sera diffusé à l’administration et aux corps enseignants.

### Détails des cas d’utilisations pour l’étudiant

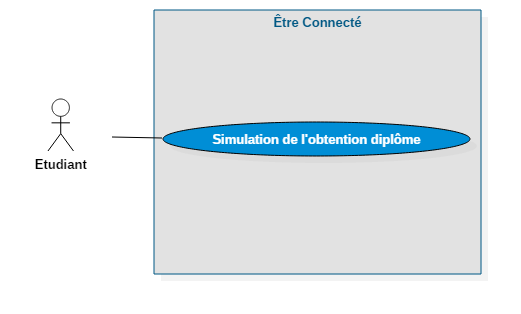


Figure 17 - Simulation de l’obtention du diplôme

#### Intitulé

* Simuler l’obtention du diplôme

#### Acteur

* Etudiant

#### Données en entrée

* L’étudiant doit être connecté à l’application.

#### Scénario

1. Le système affiche la fenêtre des notes de l’étudiant.
2. En plus de voir les notes obtenues, il saisit les notes dans les champs « Note ».
3. L’étudiant sélectionne le bouton simulation.

## Diagramme de classe

La structure du logiciel est composée de **15 classes** dont **3 énumérations** et **1 abstraite** :

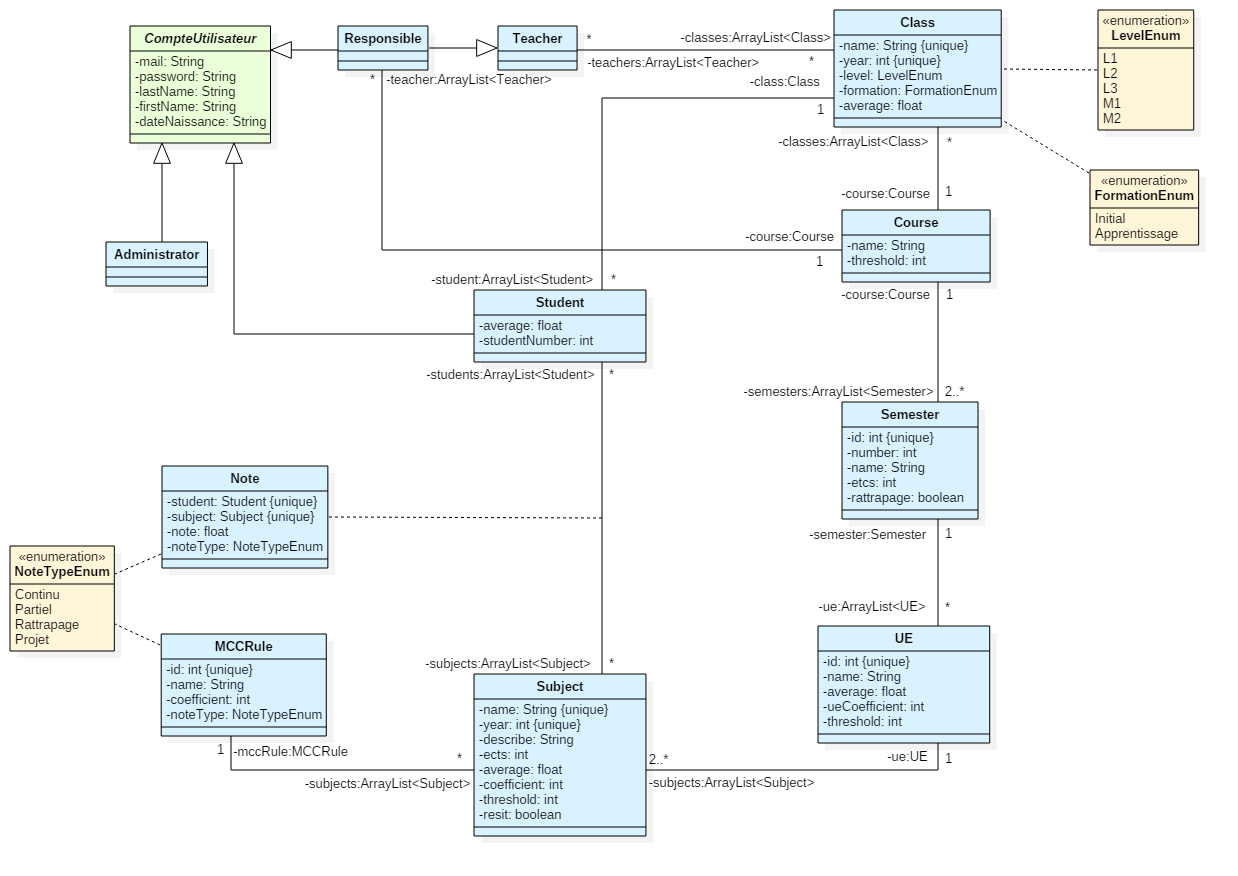


Figure 18 - Diagramme de classe

La classe « CompteUtilisateur » est une classe abstraite qui hérite de l’administrateur, de l’enseignant, du responsable et de l’étudient.

L’épine dorsale du projet est la classe « Class ». En effet, il est la jonction entre l’étudiant et la filière qui permet d’articuler notre modèle et ainsi imbriquer le semestre avec l’UE, l’UE avec les matières, etc.  
  
Nous disposons aussi de 3 classes d’énumération :

* + - * « LevelEnum » contenant les niveaux de formation disponibles
      * « FormationEnum » contenant les types de formation existants
      * « NoteTypeEnum » contenant les types de note afin d’associer correctement les règles MCC

## Diagramme de Séquence

### Connexion

Ce diagramme de séquence représente les échanges d’informations permettant à un utilisateur de se connecter à son compte.

À l’aide d’un portail de connexion classique, l’utilisateur entre son adresse mail et son mot de passe.

En cas d’erreur, un message avertit l’utilisateur.

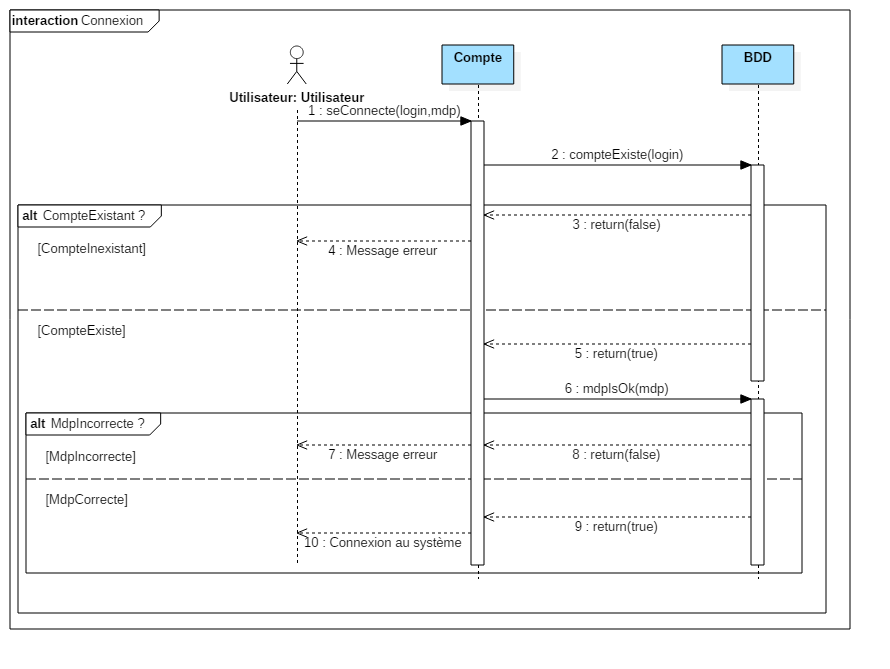


Figure 19 - Diagramme de séquence – Connexion

### Création d’une filière

Ce diagramme de séquence représente la création d’une filière.

À l’aide d’un formulaire de création d’une filière, l’administrateur a la possibilité de créer une nouvelle filière. Cependant, la filière étant la clé de voute de l’implémentation des données, il est obligatoire d’ajouter une matière, un UE, un semestre, etc.

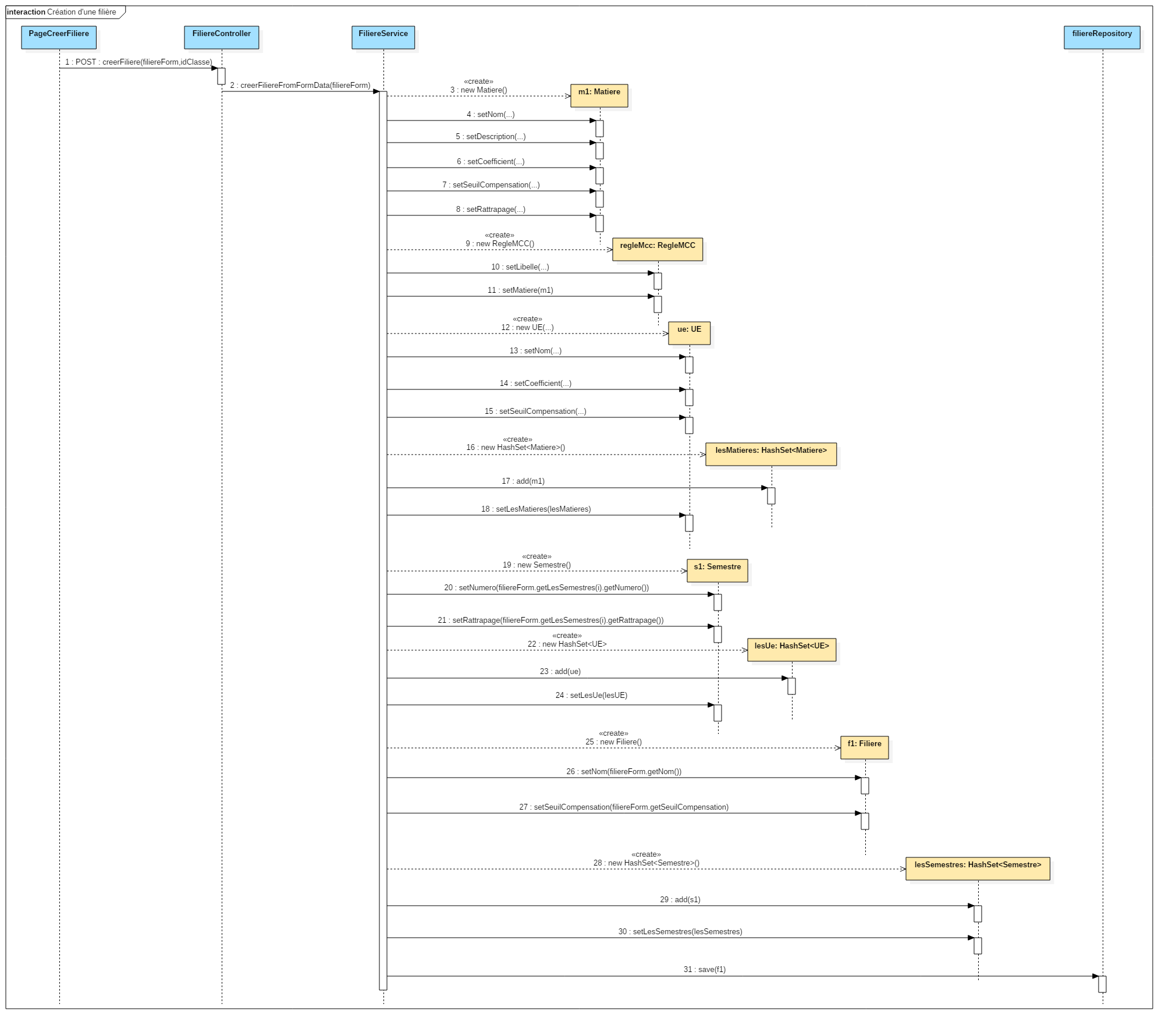


Figure 20 - Diagramme de séquence – Création d’une filière

### Création d’une matière

Ce diagramme de séquence représente la création d’une matière.

À l’aide d’un formulaire de création d’une matière, l’administrateur créé une matière.

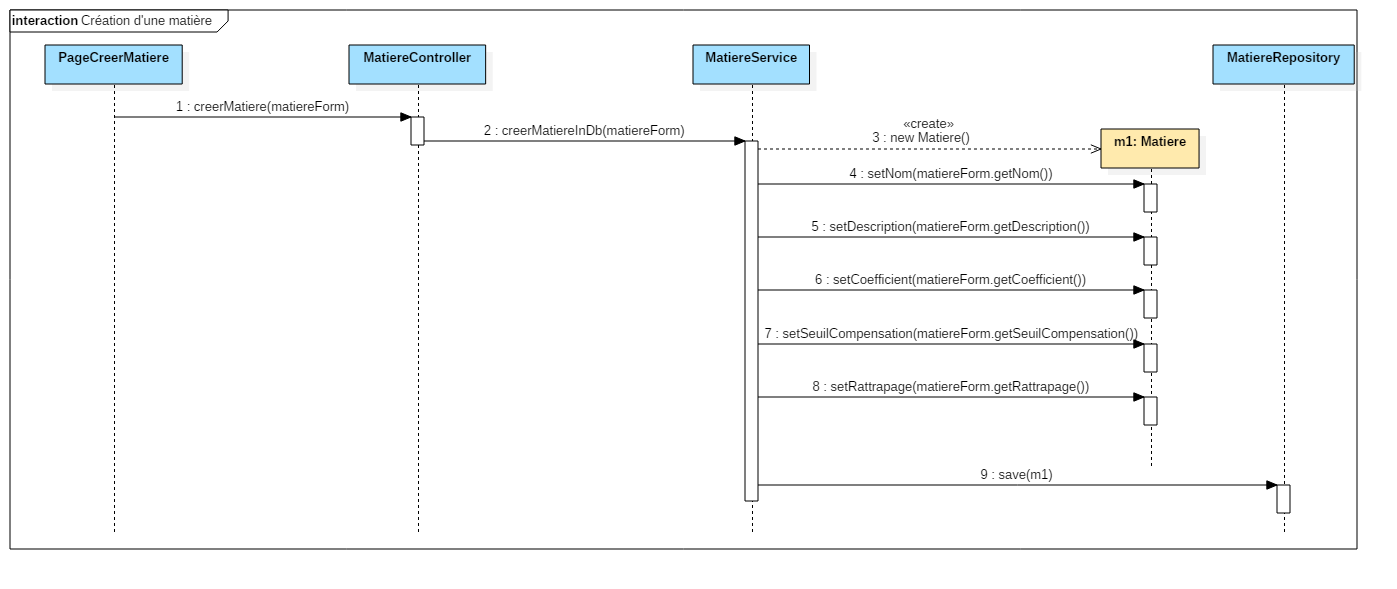


Figure 21 - Diagramme de séquence – Création d’une matière

### Création d’une classe

Ce diagramme de séquence représente la création d’une classe.

À l’aide d’un formulaire de création d’une classe, l’administrateur créé une classe et peut lui attribuer des étudiants.

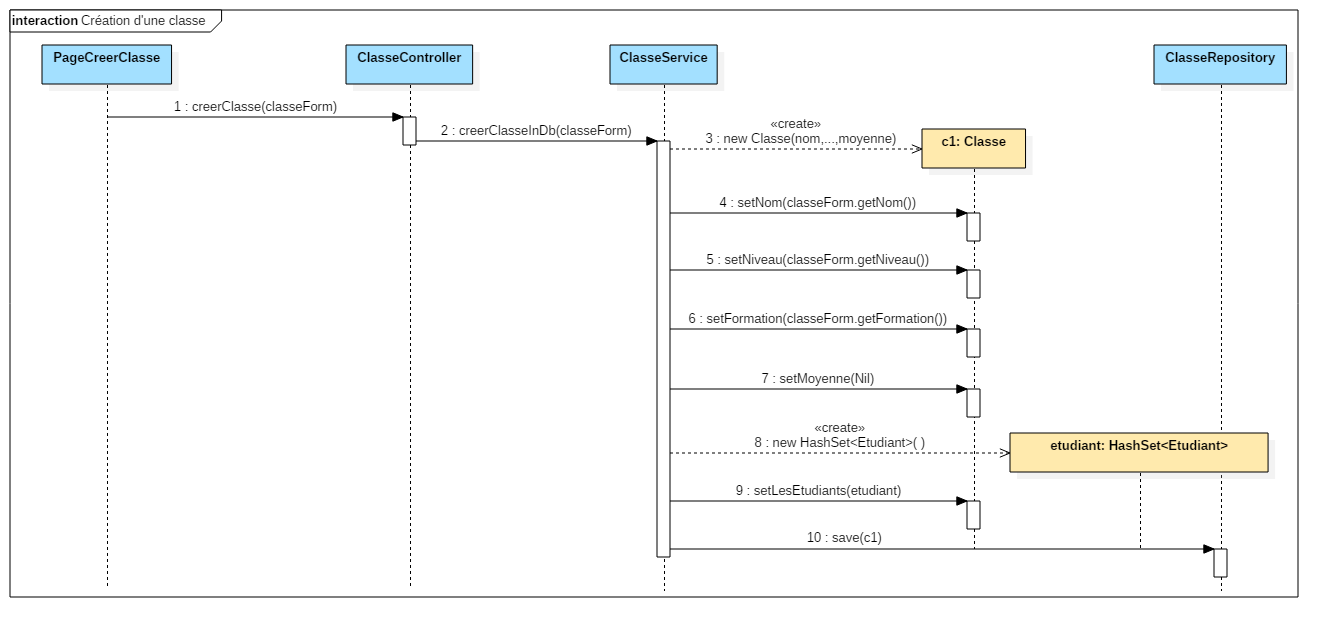


Figure 22 - Diagramme de séquence – Création d’une classe

### Publier les notes

Ce diagramme de séquence représente l’enchainement d’actions du bouton « publier une note ».

Un enseignant peut publier une note une fois saisie. Elle permet d’actualiser les notes Matière et Etudiante.

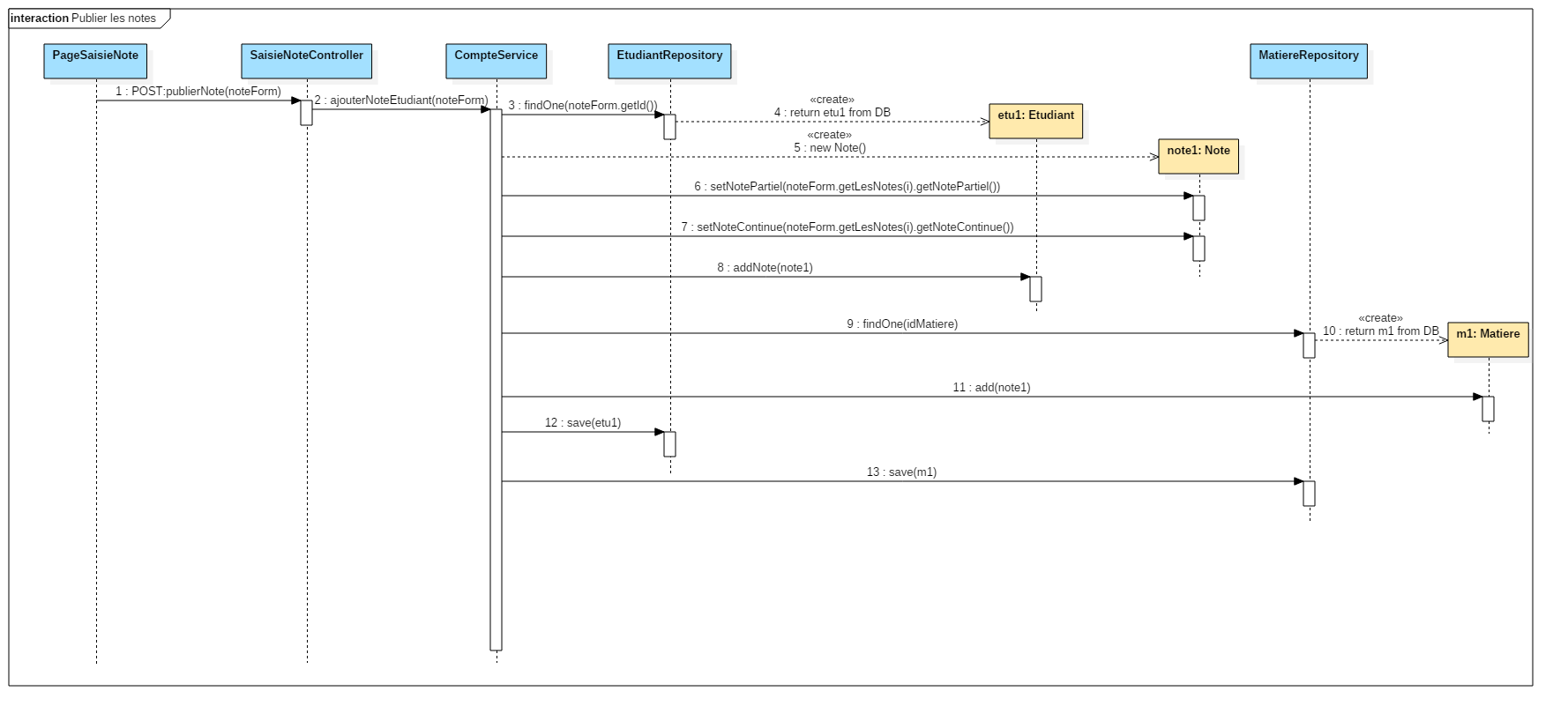


Figure 23 - Diagramme de séquence – Publier des notes

## Diagramme d’État

### Connexion à compte

Ce diagramme d’état représente les échanges d’informations permettant à un utilisateur de se connecter à son compte.

À l’aide d’un portail de connexion classique, l’utilisateur entre son adresse mail et son mot de passe.

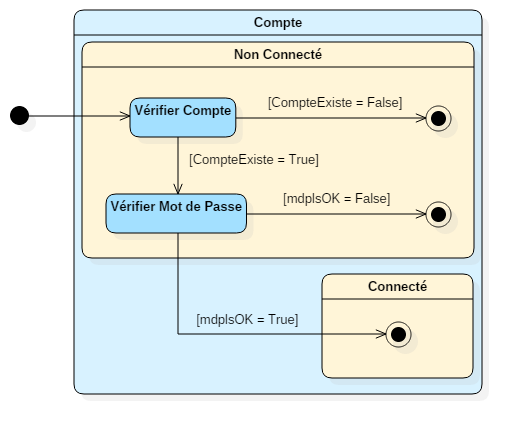


Figure 24 - Diagramme d’état– Connexion à un compte

### Validation d’une matière

Ce diagramme d’état représente l’état d’une matière lorsqu’une matière est renseignée.

Elle peut être « Non validé », « Ajourné » ou « Validé ».

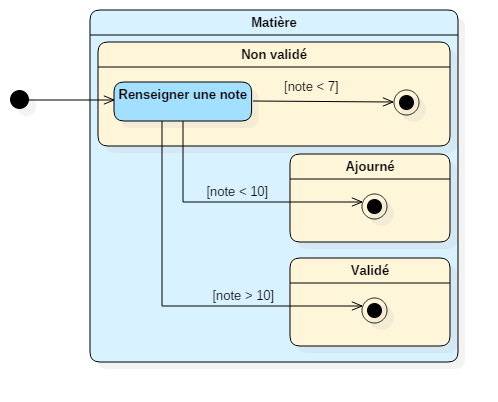


Figure 25 - Diagramme d’état – Validation d’une matière

## Diagramme de paquetage

Ce diagramme de paquetage représente les relations générales entre les objets de l’application.

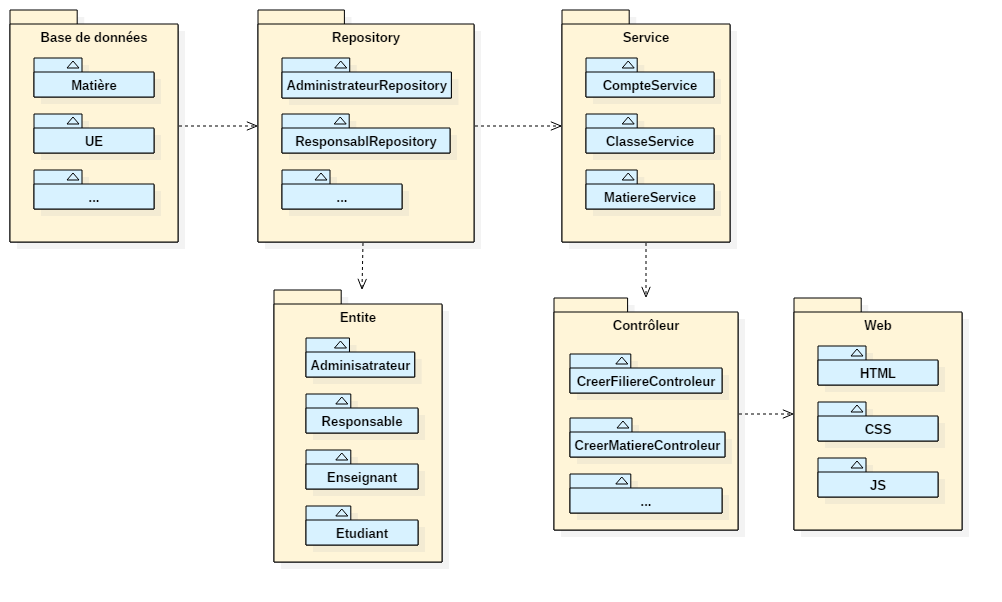


Figure 26 - Diagramme de paquetage

# Implémentation

## Interface Homme-Machine

### **Page sans authentification**

#### Portail de connexion

L’utilisateur entre son adresse mail et son mot de passe.

En cas d’erreur de format, le champs e-mail devient rouge et prévient d’une erreur.

En cas d’erreur dans l’authentification, un message d’erreur s’affiche.

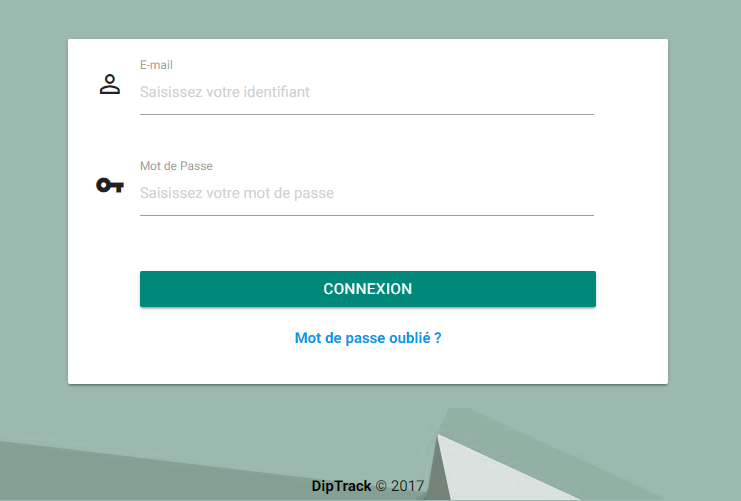


Figure 27 - Portail de connexion

#### Passerelle d’activation d’un compte

Lorsqu’un utilisateur dispose d’un nouveau compte, il **reçoit automatiquement** un mail afin d’activer son compte. Dès lors, il sera redirigé vers une page d’attente avant d’atterrir sur la création de mot de passe.

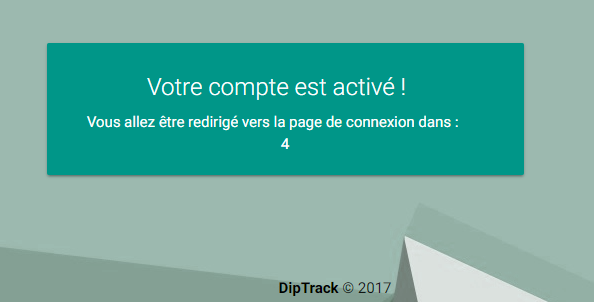


Figure 28 - Page de redirection

### Création et/ou réinitialisation d’un mot de passe

Après la page de redirection, l’utilisateur apparaît sur une page lui permettant de saisir son mot de passe. Celui-ci sera haché[[4]](#footnote-4) et insérer dans la base de données.

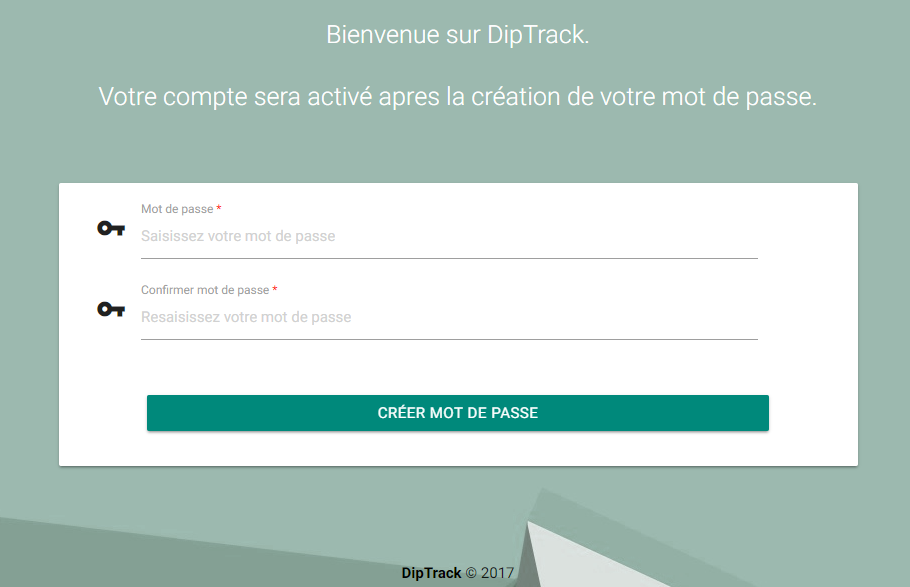


Figure 29 - Création et/ou réinitialisation du mot de passe

### **Interface Administrateur**

#### Gestion des comptes

L’administrateur est en charge de gérer les comptes de chaque utilisateur.

Il peut supprimer des comptes existants (autre que le sien), en ajouter ou bien les modifier.

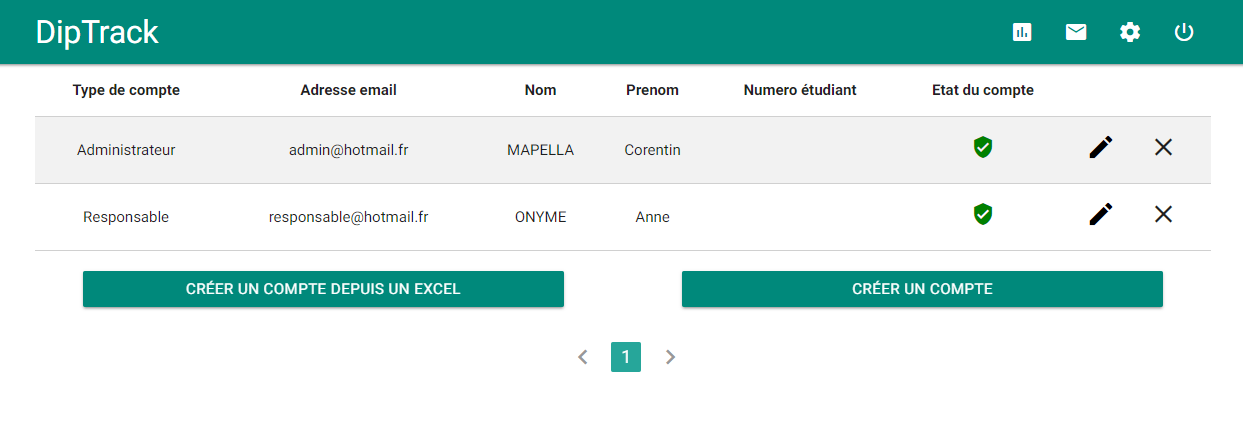


Figure 30 - Gestion des comptes

### 

#### Créer un compte

L’administrateur est en charge de gérer les comptes de chaque utilisateur.

Il peut supprimer des comptes existants (autre que le sien), en ajouter ou bien les modifier.

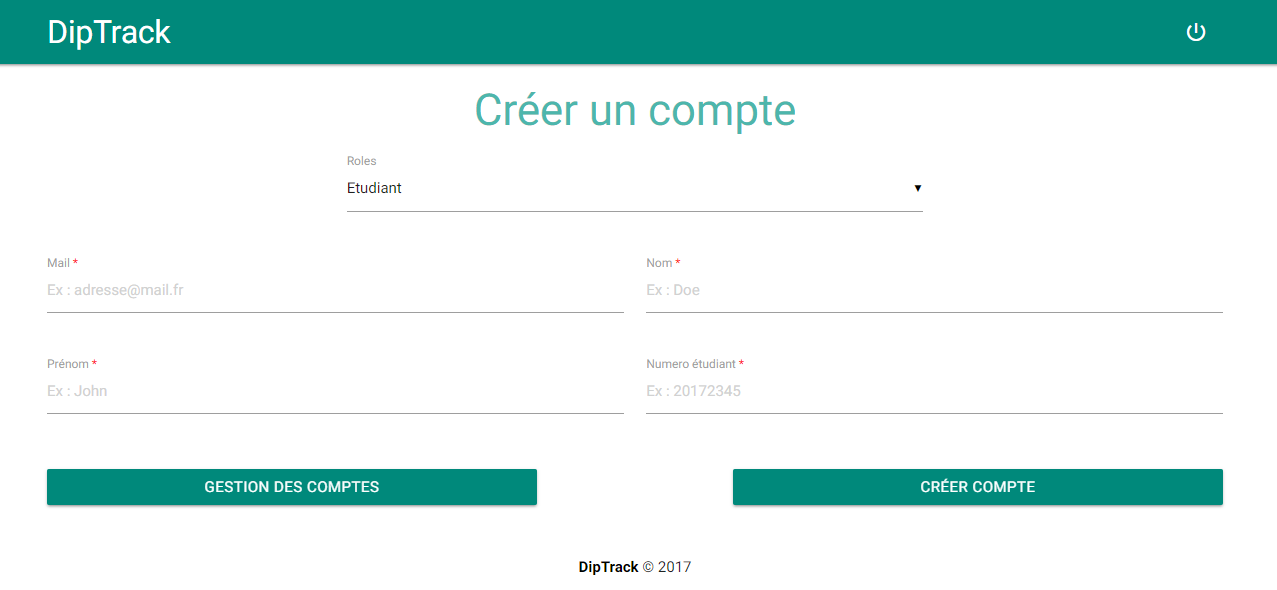


Figure 31 - Création d'un compte manuellement

### 

#### Éditer un compte

Un pop-up s’affiche lorsque l’on veut modifier un compte.

Étant une clé primaire, l’adresse mail ne peut pas être modifié.

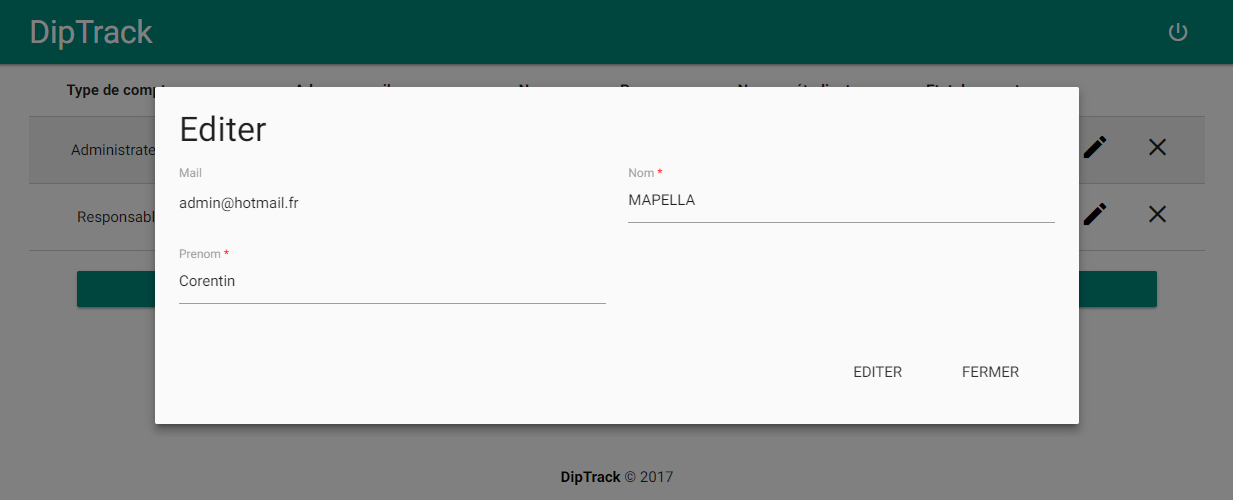


Figure 32- Éditer un compte manuellement

### 

#### Création de comptes depuis un fichier Excel

Un pop-up s’affiche afin d’insérer le fichier Excel édité au préalable.

Une fois renseigné, le fichier Excel va être analysé et intégré à la base de données.

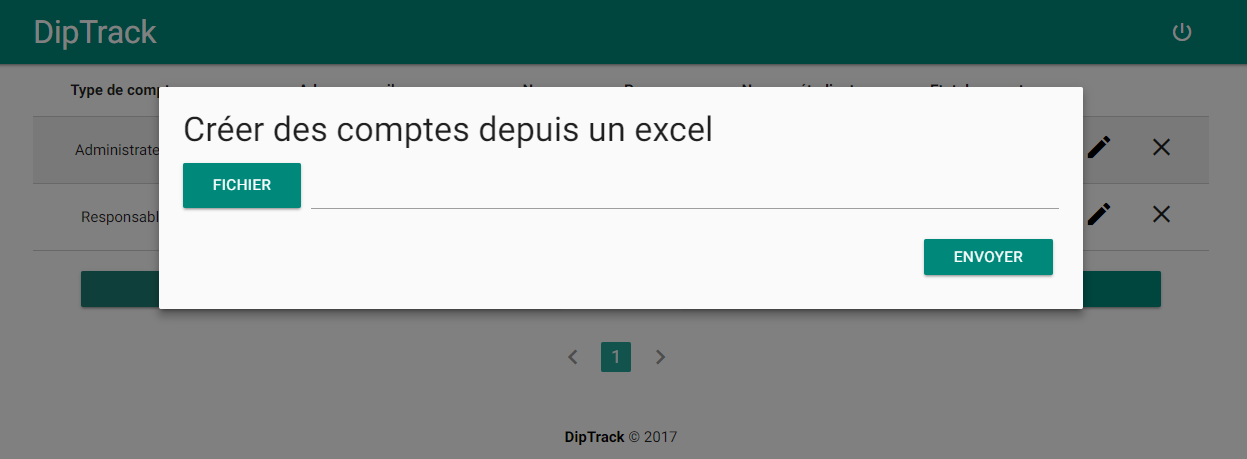


Figure 33 - Création d'un compte via Excel

#### Résultat de la création de compte

Une page s’affiche affichant les lignes qui ont été ajoutées à votre base.

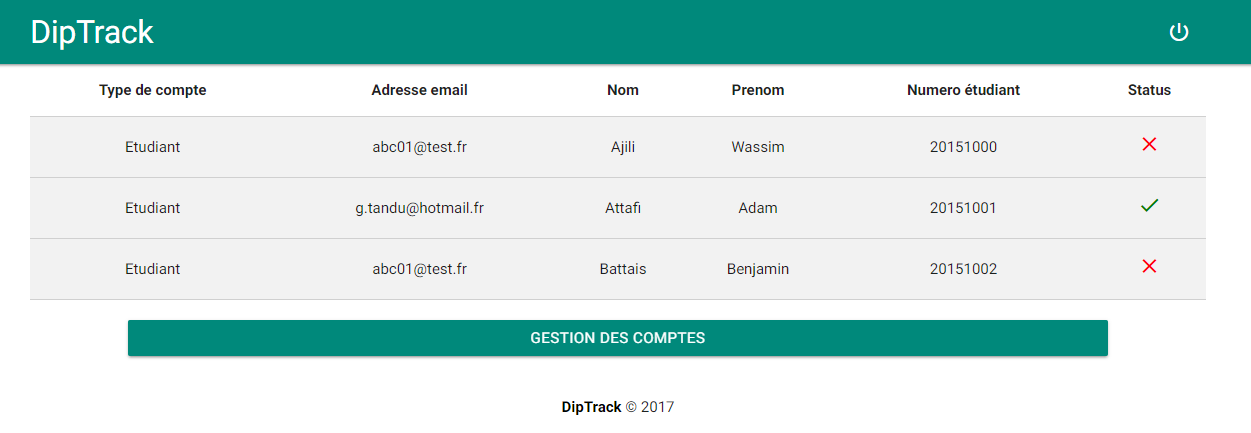


Figure 34 - Résultat de l'intégration des données

### **Interface Enseignant**

#### Page d’accueil d’un enseignant

Lorsqu’un enseignant se connecte à son compte, il est redirigé vers une page d’accueil classique regroupant ses informations et des liens lui donnant accès à ses classes.

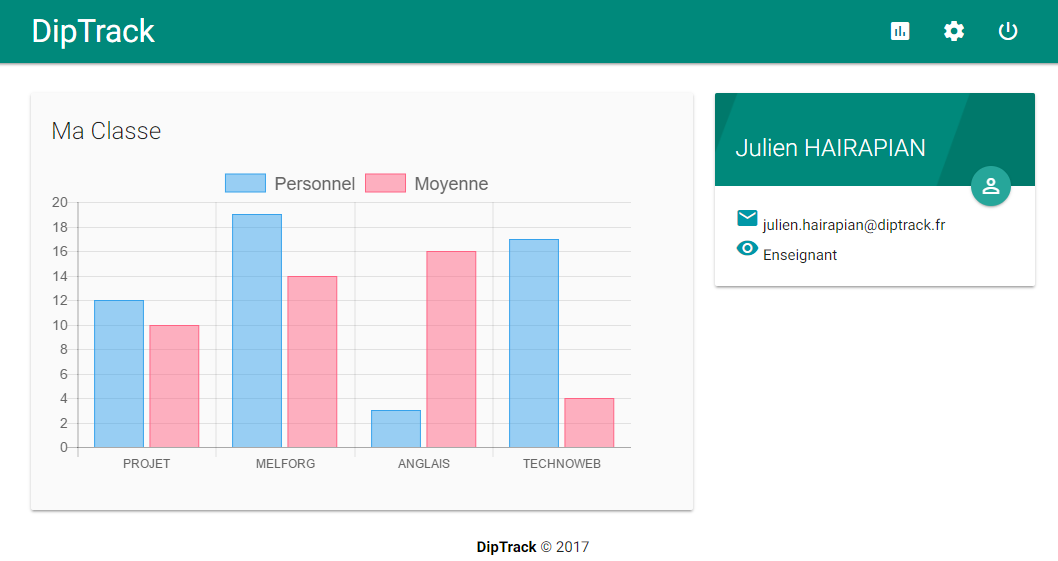


Figure 35 - Page d’accueil d’un enseignant

#### Gestion et publication des notes pour une classe

Lorsque l’enseignant atteint sa page regroupant ses classes, il peut attribuer des notes à ses étudiants, suspendre les opérations pour le reprendre plus tard. Une fois la totalité complété, l’enseignant peut publier ses notes. Dès lors, un mail sera transmis à chaque interlocuteur concerné.

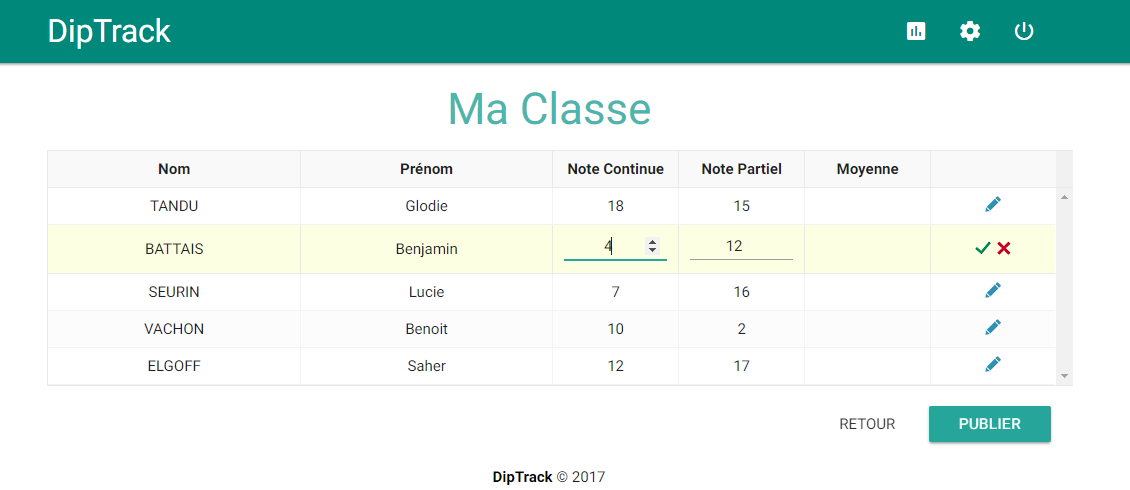


Figure 36 - Gestion et publication des notes pour une classe

### **Interface Responsable**

#### Page d’accueil d’un responsable

La page d’accueil du responsable comporte le même squelette que celle de l’enseignant.

#### Créer une classe manuellement

Le responsable est chargé de créer les classes dont il est chargé et de les attribuer aux enseignants désignés.

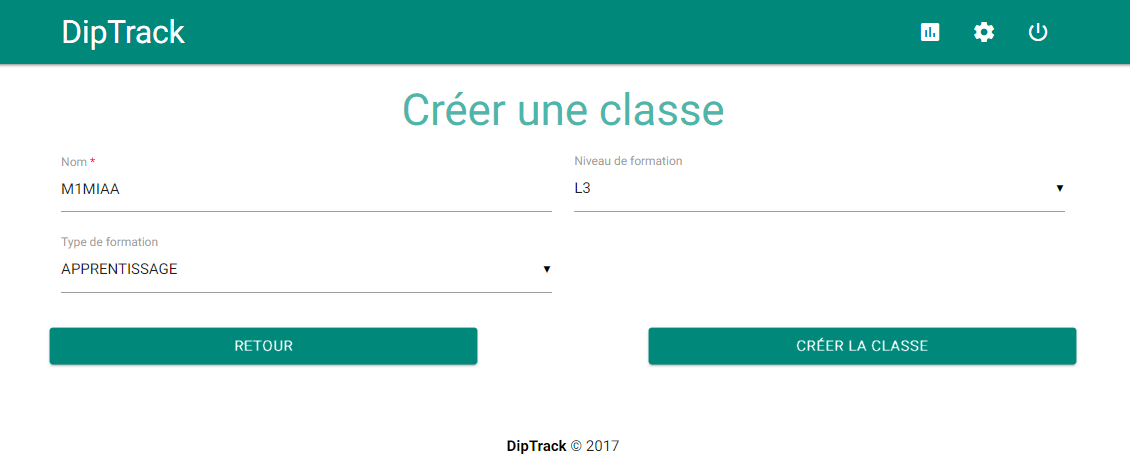


Figure 37 - Créer une classe manuellement

#### Gestion des classes

Cette page permet au responsable de gérer les classes existantes.



Figure 38 - Gestion des classes

### **Autres pages**

#### Pages d’erreur

Lorsqu’un utilisateur essaye d’aller sur une page dont il n’a pas les droits, un message d’avertissement s’affichera.

En cas de page introuvable, une page d’erreur préviendra l’utilisateur.



Figure 39 - Message « Accès refusé »



Figure 40 - Message « Erreur 404 »

#### Template des mails reçus

L’application nécessite l’envoi de mail pour la confirmation de compte ou même en cas d’oubli de mot de passe. Celui-ci est responsive quel que soit le support (Ordinateurs, Tablettes, mobiles).

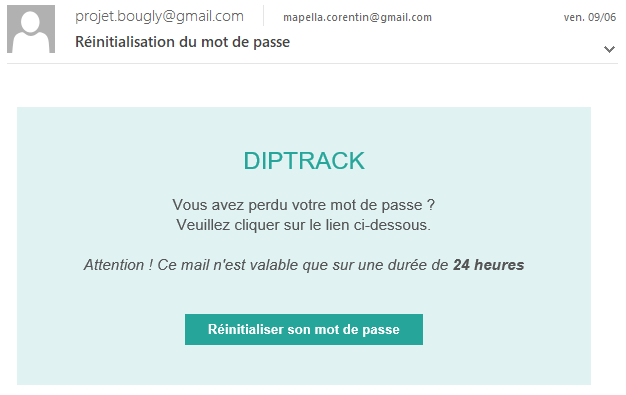


Figure 41 - Email pour réinitialiser son mot de passe

## Modèle Conceptuel de Données

Le modèle conceptuel de données reprend la majorité des objets utilisés dans le modèle java.

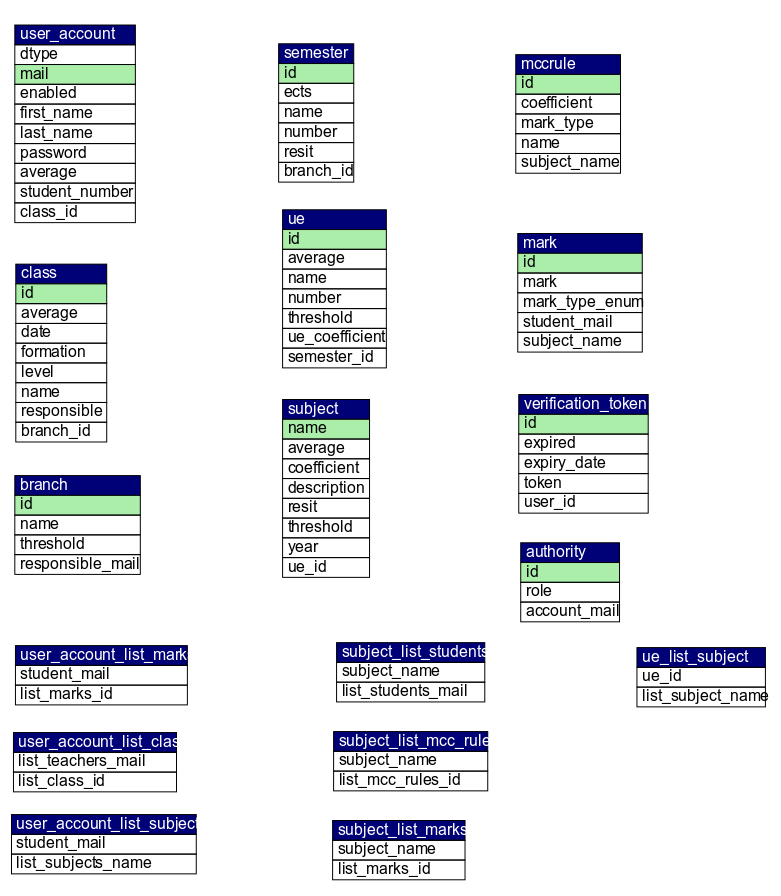


Figure 42 - Modèle Conceptuel de Données

## Fonctionnalités développées et à développer

Depuis le début du développement, de nombreuses fonctionnalités ont été créées, améliorées mais aussi mise de côté. En effet, faute d’un délai courte pour le déploiement d’une application, les fonctionnalités majeurs de DIPTRACK ont été implémentées tels que la gestion des comptes, des classes, l’envoi de mail et la publication de note.

Cependant, la simulation d’obtention du diplôme et même le procès-verbal ont été dépriorisé afin de privilégier le cœur du projet.

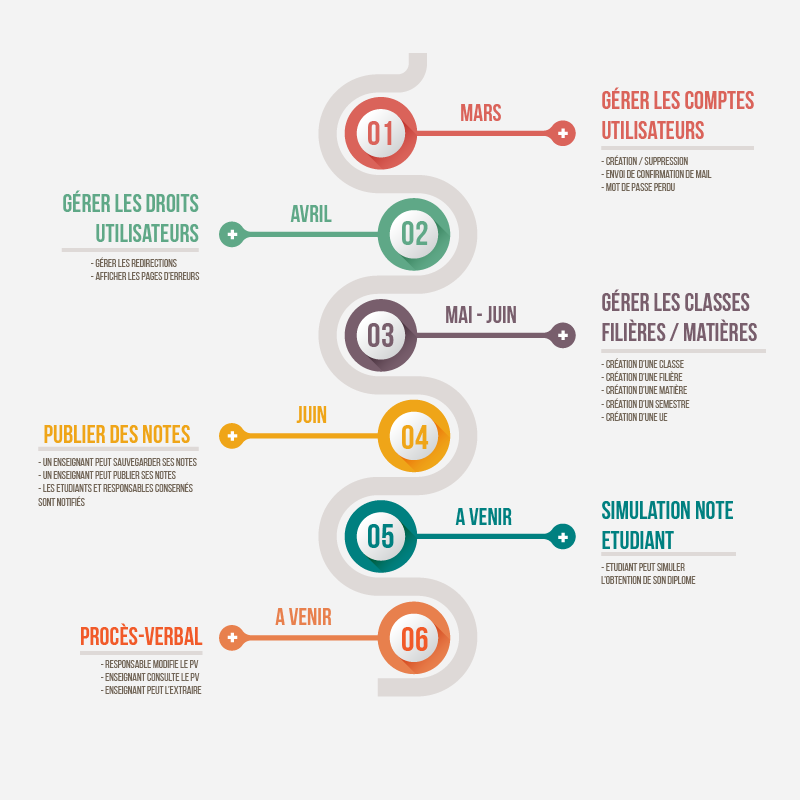


Figure 43 - Timeline du projet

## Difficultés rencontrés / Solutions trouvées

La mise en route du développement sachant que les technologies avaient été choisies rapidement.

Cependant, nous avons rencontré des difficultés sur la gestion des droits utilisateurs et sur la redirection une fois connecté. En effet, ces fonctionnalités sont la base de notre application et il était donc de commencer avec des bases solides et donc optimiser au mieux cet aspect.

Le modèle étant composé de beaucoup de dépendance (Ex : Semestre -> UE, UE -> Matière, etc.), il a été difficile de garder cette cohérence lors de la création du front.

Afin de surmonter ces difficultés, il nous a fallu beaucoup de patience et une multitude de recherches sur internet.

## Changements apportés par rapport à la conception

Le projet s’est déroulé sans changement majeur. Les technologies, les fonctionnalités et même les risques ont été cernés en amont ce qui a facilité le développement. Le contenu n’a donc pas empati de modifications mineures.

Cependant, sur la forme, nous avions commencé notre code en mélangeant français et anglais. Suite à une orientation de Julien, nous avons passé le code entièrement en anglais afin de garder une cohérence et une unicité de langue dans notre projet.

De plus, le nom du projet nommé initialement « Boogly » a été modifié pour donner naissance à « DipTrack » (Contraction des mots « Diploma » et « Tracking ») et a été source de modifications.

Le projet étant déjà avancé, ces modifications, paraissant anodines, ont été chronophage et nous ont fait perdre une semaine de développement.

## Répartition du travail

Ayant utilisé la méthode SCRUM, nous nous répartitions les tâches en fonction des user-stories disponibles.



Figure 44 - Tableau de la répartition des tâches

# Tables des illustrations

[Figure 1 - Coverage du Projet 8](#_Toc485253533)

[Figure 2 - Portail d'authentification 9](#_Toc485253534)

[Figure 3 - Page d’accueil d’un élève 9](#_Toc485253535)

[Figure 4 - Page d’accueil d’un enseignant 10](#_Toc485253536)

[Figure 5 - Page d'accueil d'un étudiant 10](#_Toc485253537)

[Figure 6 - Page d’une classe d’un enseignant 11](#_Toc485253538)

[Figure 7 - Formulaire de création 11](#_Toc485253539)

[Figure 8 - Formulaire d’une matière 12](#_Toc485253540)

[Figure 9 - Formulaire de création d’une classe 12](#_Toc485253541)

[Figure 10 - Diagramme de cas d’utilisations général 13](#_Toc485253542)

[Figure 11- Connexion à un compte 14](#_Toc485253543)

[Figure 12- Création d’un compte 15](#_Toc485253544)

[Figure 13- Création d’une filière 16](#_Toc485253545)

[Figure 14 - Création d’une matière 17](#_Toc485253546)

[Figure 15 - Création d’une classe 18](#_Toc485253547)

[Figure 16 - Publication des notes 19](#_Toc485253548)

[Figure 17 - Simulation de l’obtention du diplôme 20](#_Toc485253549)

[Figure 18 - Diagramme de classe 21](#_Toc485253550)

[Figure 19 - Diagramme de séquence – Connexion 22](#_Toc485253551)

[Figure 20 - Diagramme de séquence – Création d’une filière 23](#_Toc485253552)

[Figure 21 - Diagramme de séquence – Création d’une matière 24](#_Toc485253553)

[Figure 22 - Diagramme de séquence – Création d’une classe 24](#_Toc485253554)

[Figure 23 - Diagramme de séquence – Publier des notes 25](#_Toc485253555)

[Figure 24 - Diagramme d’état– Connexion à un compte 25](#_Toc485253556)

[Figure 25 - Diagramme d’état – Validation d’une matière 26](#_Toc485253557)

[Figure 26 - Diagramme de paquetage 26](#_Toc485253558)

[Figure 27 - Portail de connexion 27](#_Toc485253559)

[Figure 28 - Page de redirection 27](#_Toc485253560)

[Figure 29 - Création et/ou réinitialisation du mot de passe 28](#_Toc485253561)

[Figure 30 - Gestion des comptes 28](#_Toc485253562)

[Figure 31 - Création d'un compte manuellement 29](#_Toc485253563)

[Figure 32- Éditer un compte manuellement 29](#_Toc485253564)

[Figure 33 - Création d'un compte via Excel 30](#_Toc485253565)

[Figure 34 - Résultat de l'intégration des données 30](#_Toc485253566)

[Figure 35 - Page d’accueil d’un enseignant 31](#_Toc485253567)

[Figure 36 - Gestion et publication des notes pour une classe 31](#_Toc485253568)

[Figure 37 - Créer une classe manuellement 32](#_Toc485253569)

[Figure 38 - Gestion des classes 32](#_Toc485253570)

[Figure 39 - Message « Accès refusé » 33](#_Toc485253571)

[Figure 40 - Message « Erreur 404 » 33](#_Toc485253572)

[Figure 41 - Email pour réinitialiser son mot de passe 34](#_Toc485253573)

[Figure 42 - Modèle Conceptuel de Données 35](#_Toc485253574)

[Figure 43 - Timeline du projet 36](#_Toc485253575)

[Figure 44 - Tableau de la répartition des tâches 38](#_Toc485253576)

1. Site officiel Spring Boot : <https://projects.spring.io/spring-boot/> [↑](#footnote-ref-1)
2. Définition officielle de WAMP : <https://fr.wikipedia.org/wiki/WAMP> [↑](#footnote-ref-2)
3. Définition officielle de MAMP : <https://fr.wikipedia.org/wiki/MAMP> [↑](#footnote-ref-3)
4. Hachage : méthode de cryptage de données afin de sécuriser l’information. [↑](#footnote-ref-4)