

كلية العلوم  
ⵜⴰⵎⴻⵔⴰⵏⵜ ⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ  
FACULTÉ DES SCIENCES



## UNIVERSITÉ IBN ZOHR FACULTÉ DES SCIENCES

Département Informatique  
Filière Sciences Mathématiques et Informatique

Projet de Fin d'Études  
Réalisé par : TRKZI Omar

En vue d'obtenir la  
Licence en Sciences Mathématiques et Informatique

---

### **Conception et développement d'une application de gestion de stage avec le framework Symfony**

---

Organisme d'accueil:



كلية الطب والصيدلة - أكادير  
ⵜⴰⵎⴻⵔⴰⵏⵜ ⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ ⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ - ⴰⵎⴻⵔⴰⵏⵜ  
FACULTÉ DE MÉDECINE ET DE PHARMACIE D'AGADIR

Encadré par :

Pr. KOUTTI Lahcen

Responsable dans  
l'organisme d'accueil :

M. OUCHRAA Adil

Soutenu le 16 Juin 2022

Page laissée intentionnellement vide

# Dédicaces

A Dieu tout puissant notre créateur

A mes très chers parents, les personnes les plus dignes de mon respect, pour leur support et sacrifices indénombrables.

A M. KOUTTI, mon encadrant et Mme LEMKHENTE, vice-doyenne de la FMPA, qui sans leur collaboration, ce travail n'aura jamais vu le jour, et à M. OUCHRAA, qui m'a permis d'explorer le monde professionnel tout au long de la réalisation de ce projet.

A tous mes proches, qui n'ont jamais arrêté de m'encourager pour avancer.

A tous mes amis, qui m'ont soutenu et cru en moi.

A tout professeur, qui a laissé sa marque dans moi et qui m'a tant inspiré pour ajouter de la valeur à notre société.

Je leur dédie ce modeste travail en leur souhaitant un immense bonheur

dans la vie et l'au-delà.

TRKZI Omar

# Remerciements

En tout premier lieu, nous devons adresser nos remerciements à nos encadrants M. KOUTTI et M. OUCHRAA, leurs encouragements et leur bonne volonté, dont ils ont fait preuve pour la durée de notre stage, ainsi que leurs conseils et encadrement.

Nous tenons encore à remercier le personnel de l'administration de la Faculté de Médecine et Pharmacie d'Agadir, qui nous a chaleureusement accueilli et donné l'opportunité de travailler dans un contexte informatique professionnel.

Nos remerciements sont aussi pour tous nos enseignants à la Faculté de Sciences d'Agadir, nous ayant offert une formation alignée avec les demandes internationales du monde professionnel.

Et nous remercions de tout cœur les respectueux membres du jury, qui nous ont offert l'honneur de juger notre travail.

Enfin, nous remercions toute personne ayant contribué de loin ou de proche à la réalisation et aboutissement de ce travail.

A toutes et à tous, nous vous remercions infiniment.

# Résumé

Notre projet de fin d'études au sein de la Faculté de Médecine et Pharmacie d'Agadir s'inscrit dans le cadre de la numérisation de quelques tâches administratives annuelles. Cela consiste à offrir aux responsables de l'administration une application web exécutant ces mêmes tâches avec plus de facilité et de précision.

Afin de remédier à cela, nous développons une application web, basée sur le framework Symfony 5.4, permettant d'une part, aux administratifs de la FMIPA, la gestion numérique des données relatives aux calendriers des stages et la génération automatique de ces derniers, et d'autre part aux professeurs de noter les stages passés sous leur supervision.

**Mots clés :** PHP, Symfony, CRUD, MVC, DevOps, MariaDB, Twig, Docker, GitHub.

# Table des matières

Dédicaces.....	ii
Remerciements.....	iii
Résumé.....	iv
Table des matières.....	v
Liste des figures.....	vii
Liste des abréviations.....	ix
Introduction générale.....	1
Chapitre 1 : Contexte général du projet.....	2
1. Organisme d'accueil.....	3
a. Présentation générale.....	3
b. Partenaires.....	3
2. Présentation du projet.....	4
a. Contexte du projet.....	4
b. Problématique abordée.....	6
c. Solution envisagée.....	6
3. Conduite du projet.....	7
a. Choix méthodologique.....	7
b. Planification du projet.....	2
Chapitre 2 : Analyse et conception.....	13
1. Spécification des besoins.....	14
2. Cas d'utilisations.....	15
3. Diagrammes de séquence.....	18
4. Architecture du système.....	21
5. Modèle de données.....	21
Chapitre 3 : Architecture et environnement.....	23
1. Architecture adoptée.....	24
2. Choix du langage.....	24
3. Framework et container.....	25
Chapitre 4 : Réalisation, interfaces, tests.....	28
1. Environnement et outils.....	29
1.1 Environnement logiciel.....	29
1.2 Outils.....	29
2. Interfaces et tests.....	33

Conclusion Générale et perspectives.....	42
Webographie et ressources.....	43
Annexe.....	44

# Liste des figures

Figure 1. Logo de la Faculté de Médecine et de Pharmacie d'Agadir.....	3
Figure 2. Logo du SFD.....	3
Figure 3. Canevas des répartitions manuels des stages.....	5
Figure 4. Carnet de stage manuel.....	5
Figure 5. Implémentation de la méthode Kanban avec Trello.....	9
Figure 6. Logo de Gantt Project.....	10
Figure 7. Table du diagramme de Gantt.....	10
Figure 8. Diagramme de Gantt.....	11
Figure 9. Diagramme de cas d'utilisation de l'administratif.....	16
Figure 10. Diagramme d e cas d'utilisation d'un professeur.....	17
Figure 11. Diagramme d e cas d'utilisation de l'application de scolarité.....	17
Figure 12. Diagramme d e séquence de l'administratif.....	18
Figure 13. Diagramme d e séquence d'un professeur.....	19
Figure 14. Architecture de la solution.....	20
Figure 15. Modèle conceptuel des données.....	21
Figure 16. Architecture MVC.....	24
Figure 17. Logo de PHP.....	25
Figure 18. Logo Symfony.....	25
Figure 19. Logo de Docker.....	26
Figure 20. Logo de PhpStorm.....	29
Figure 21. Logo de DDEV.....	30
Figure 22. Logo de Portainer.....	30
Figure 23. Logo de GitHub.....	31
Figure 24. Logo de git.....	31
Figure 25. Logo de Lucidchart.....	32
Figure 26. Logo de Trello.....	32
Figure 27. Paramètres du calendrier à afficher.....	33
Figure 28. Le calendrier n'est pas encore généré.....	33
Figure 29. Affichage du calendrier.....	34
Figure 30. Paramètres du calendrier à générer.....	34
Figure 31. Confirmer les périodes, les stages et les services.....	35
Figure 32. Le nombre des périodes doit être égal à celui des stages.....	35



Figure 33. Confirmer les étudiants inscrits.....	36
Figure 34. Au moins un étudiant a déjà un groupe.....	36
Figure 35. Avertissement des groupes.....	37
Figure 36. Confirmer le calendrier.....	37
Figure 37. Choisir l'AU.....	38
Figure 38. Liste des groupes.....	38
Figure 39. Lise des étudiants de ce groupe.....	39
Figure 40. Afficher le carnet de stage.....	39
Figure 41. Vérification des champs du carnet de stage.....	40

# Liste des abréviations

**MVC** : Modèle-Vue-Contrôleur

**CRUD** : Create, Read, Update, Delete

**MCD** : Modèle Conceptuel des Données

**API** : Application Programming Interface

**JAT** : Juste À Temps

**AU** : Année Universitaire

**FMPA** : Faculté de Médecine et Pharmacie Agadir

**UML** : Unified Modeling Language

# Introduction générale

Depuis le début des années 2000, la révolution numérique bouleverse progressivement le quotidien des citoyens. Tout est désormais plus simple et plus rapide dans la plupart des secteurs. Par conséquent, les établissements publics comme privés, visent de plus en plus à intégrer des technologies numériques. C'est désormais une nécessité pour améliorer leur productivité en diminuant la probabilité de l'erreur humaine, ainsi qu'en permettant à leurs personnels de se concentrer sur d'autres tâches.

Dans ce cadre, l'administration de la FMPA souhaite numériser le processus de la génération des calendriers des stages ainsi que la notation de ses étudiants. Ce processus se déroule de façon annuelle, pour chaque niveau, et répartit ainsi ses étudiants en groupes en fonction de certains paramètres.

Afin d'automatiser cette tâche, nous développons une application web basée sur le framework Symfony 5.4, permettant au personnel concerné de la FMPA d'une part la gestion numérique (CRUD) des données relatives aux calendriers ainsi que la génération automatique de ces derniers en fonction des paramètres nécessaires, et d'autre part l'attribution des notes de stage pour chaque étudiant par le professeur qui l'a encadré.

Le présent rapport de notre projet de fin d'études, s'articule donc autour de quatre chapitres :

- **Le premier chapitre** : explique le contexte du projet, ainsi que la problématique qu'il résout.
- **Le deuxième chapitre** : présente une analyse de besoins, ainsi que la conception de notre application.
- **Le troisième chapitre** : traite les technologies utilisées pour le développement de la solution.
- **Le quatrième chapitre** : est dédié à la partie réalisation de notre projet.

# Chapitre 1 : Contexte général du projet

*Dans ce chapitre, nous abordons le contexte général du projet. Nous présentons, tout d'abord, l'organisme d'accueil. Puis, pour mettre en exergue l'objectif de ce travail, nous présentons les problématiques ainsi que les objectifs visés. Et nous clôturons par le plan d'action présentant la méthodologie adoptée.*

## 1. Organisme d'accueil :

### a. Présentation générale :



Figure 1: Logo de le FMFA

Créée en 2016, la Faculté de Médecine et Pharmacie Agadir (FMFA), est l'une des 21 institutions de l'Université Ibn Zohr. Elle est située au quartier Tilila en face de Universiapolis. Sa capacité d'accueil atteint 4500 étudiants, et comporte 58 enseignants-chercheurs. Son centre hospitalier universitaire (CHU) est en cours de construction.

La FMFA forme ses étudiants aux métiers de soignants et enseignant-chercheurs en santé, et les accompagne tout au long de leur parcours professionnel. L'offre de formation de la Faculté comprend les études médicales, du premier au troisième cycle.

### b. Partenaires :

Sa construction a été en partie financée par le Fonds Saoudien de Développement (SFD) et les fonds propres de l'université Ibn Zohr. Le SFD est une agence gouvernementale saoudienne qui fournit une aide aux pays en développement, en finançant des projets sociaux et d'infrastructure dans le but d'améliorer les vies et les communautés.



Figure 2: Logo du SFD

Grâce à ses partenariats pédagogiques, notamment avec la Faculté des Sciences de la Santé (Casablanca), l'Université de Bretagne Occidentale (France), l'Université de Porto (Portugal), l'Université de Las Palmas Gran Canaria (Espagne), l'Université Claude Bernard Lyon I (France) et la Faculté de Médecine Paris-Saclay, la FMPA offre un important potentiel de soins et d'enseignements qui bénéficie d'un environnement scientifique permettant de contribuer au renforcement du système de santé et à l'amélioration de l'offre de soins au Maroc.

## 2. Présentation du projet:

### a. Contexte du projet:

Le savoir pratique a une importance capitale en médecine. Dans ce cadre, la formation en médecine exige aux étudiants la passation de certains stages tout au long de leur cursus.

Les études de médecine suivent un parcours précis, composé de trois cycles :

- **Le premier cycle** (la première et la deuxième année) se veut plus général et théorique (1 stage d'immersion par année),
- **Le deuxième cycle** (de la troisième à la cinquième année), dont le volume est constitué de 50 % de stages hospitaliers passés au CHU (vu qu'il est en cours de construction, les stages se passent actuellement dans les trois centres hospitaliers : Hôpital Hassan II, Premier Centre Médico-Chirurgical, Centre Hospitalier Provincial Inezgane).
- **Le troisième cycle** (à partir de la sixième année) dont 100 % du volume de la formation est sous forme de stages.

C'est dans ce contexte que l'administration de la FMPA génère, chaque début d'année universitaire, des calendriers de stages pour chaque niveau (première à septième année). L'année universitaire est ainsi divisée en périodes (e.g : Du 25/04/2022 Au 15/06/2022) selon le nombre de stages que les étudiants d'un certain niveau sont sensés passés (Voir la figure 3).

← Canevas des répartitions des stages d'externat 2021-2022 3 année.xlsx

3ème année

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													

Répartition des groupes de stages d'externat 3ème Année de Médecine 2021/2022

STAGE	Médecine	Médecine	Radiologie	Médecine	Chirurgie	Chirurgie	Pneumologie	Chirurgie	Gastrologie
	Hôpital Hassan II	Hôpital Hassan II	CMC	CMC	Hôpital Hassan II	Hôpital Hassan II	Hôpital Hassan II	CMC	Hôpital Hassan II
Périodes	Date	Médecine Interne	Maladies infectieuses	Hôpital Hassan II	CMC	Médecine Interne	Chirurgie générale	chirurgie thoracique et vasculaire	Hôpital Hassan II
1	Du 06/09/2021 Au 05/10/2021	G1	G2	G3 A	G3 B	G4	G5	G6	G7
2	du 06/10/2021 Au 09/11/2021	G9	G1	G2 A	G2 B	G3	G4	G5	G6
3	Du 10/11/2021 Au 10/12/2021	G8	G9	G1 A	G1 B	G2	G3	G4	G5
4	Du 13/12/2021 Au 28/01/2022	G7	G8	G9 A	G9 B	G1	G2	G3	G4
5	Du 31/01/2022 Au 02/03/2022	G6	G7	G8 A	G8 B	G9	G1	G2	G3
6	Du 03/03/2022 Au 01/04/2022	G5	G6	G7 A	G7 B	G8	G9	G1	G2
7	Du 04/04/2022 Au 06/05/2022	G4	G5	G6 A	G6 B	G7	G8	G9	G1
8		G3	G4	G5 A	G5 B	G6	G7	G8	G9
9		G2	G3	G4 A	G4 B	G5	G6	G7	G8

Figure 3: Canevas des répartitions manuels des stages

À la fin de chaque période, le professeur (médecin) responsable de chaque service d'hôpital concerné, note chaque étudiant qui a passé son stage sous sa supervision selon certains critères (voir Figure 4), décidant ainsi si l'étudiant l'a validé ou pas.

Niveau d'étude, Intitulé du stage	Période	Service	Note /20 (Obligatoire)	Session de Stage Validation/Revalidation	Date, Signature du chef de service
Niveau :		Service	1-Absence		
Intitulé du stage :			2-Grille de comportement	Validation <input type="checkbox"/>	Date :
			Ponctualité	Revalidation <input type="checkbox"/>	--/--/----
			assiduité		Signature :
			Intérêt pour le malade		
			Degré de motivation	Stage Validé <input type="checkbox"/>	
			Sens des responsabilités	Stage non validé <input type="checkbox"/>	
			Capacité de relations		
			Total		
			/20		
			3. Validation des objectifs : /40		
			4. Examen de fin de stage : /40		
			Note globale :..... /20		

Figure 4: Carnet de stage manuel

## **b. Problématique abordée:**

Dans ce contexte, un nombre de problématiques rendent ces tâches plus difficiles, et qui peuvent être présentées comme suit :

- Les calendriers des stages prennent beaucoup de temps pour qu'ils soient générés :

Ceci est dû au fait que l'administratif chargé de générer ces calendriers doit le faire pour 7 niveaux (première à septième année), en prenant à chaque fois en considération que les hôpitaux peuvent ou non, avoir des services dont la catégorie (e.g : pneumologie) est celle du stage requis dans chaque période. Prendre tous ces paramètres en considération implique un retard relatif pour générer des calendriers avec précision, et affecter les étudiants concernés aux différents groupes.

- Le remplissage des carnets de stages manque de précision :

Ceci est dû au fait que le carnet de stage ci-dessus est rempli manuellement par les professeurs (médecins), par conséquent ceci peut entraîner l'oubli de certains champs.

## **c. Solution envisagée:**

La solution proposée pour résoudre ces problèmes concerne trois composantes différentes :

### **1 Génération automatique des calendriers en fonctions des paramètres entrés :**

Nous avons travaillé sur le développement d'un algorithme qui génère les calendriers de stages, en fonction des paramètres entrés par l'administratif, i.e :

- L'année universitaire, e.g : 2022/2023.
- Le niveau concerné, e.g : Troisième année.
- Les périodes de cette année universitaire, e.g : Du 05/09/2022 Au 10/10/2022.
- Les stages requis dans ce niveau, e.g : Gynécologie.
- Les hôpitaux ayant des services de mêmes catégories que les stages, e.g : Hôpital Hassan II.

### **2 Attribution des groupes de stages aux étudiants :**

Dès qu'une proposition de calendrier de stage d'un certain niveau d'une année universitaire est validée, et que les étudiants inscrits dans ce même niveau sont confirmés par l'administratif, un algorithme affecte ces étudiants à des groupes et sous groupes en prenant en considération des paramètres tels que le nombre de stages à effectuer, et le nombre de services pouvant les accueillir.

### **3 La transformation du carnet de stage en formulaire numérique :**

Après avoir constaté les difficultés rencontrées par les professeurs pour remplir les carnets de stage, nous avons pensé à ajouter une interface pour permettre à chaque professeur de saisir le nombre d'absences pour chaque étudiant ayant passé un stage chez lui, et de le noter selon les mêmes critères précisés dans la figure 4, i.e:

- Ponctualité, assiduité (note allant de 0 à 4).
- Intérêt pour le malade (note allant de 0 à 4).
- Degré de motivation (note allant de 0 à 4).
- Sens de responsabilité (note allant de 0 à 4).



- Capacité de relations (note allant de 0 à 4).
- Validation des objectifs (note allant de 0 à 40).
- Examen de fin de stage (note allant de 0 à 40).

Par la suite l'application additionne les notes de chaque étudiant, pour lui attribuer une note globale de stage sur 100. D'où est déduit une note sur 20 (en divisant la note sur 100 par 5). Décidant, ainsi s'il a validé (note totale  $\geq 10$ ) ou non (note totale  $< 10$ ) son stage. Cela facilite la tâche de notation pour les professeurs d'une part, et diminue la probabilité d'erreur d'une autre part.

### **3. Conduite du projet :**

#### **a. Choix méthodologique :**

##### **1.1.1 L'approche agile :**

L'approche agile est une approche du développement logiciel qui vise la livraison continue de logiciels fonctionnels créés par itérations rapides.

« Les méthodes de développement de logiciels agile sont axées sur la distribution rapide de petites parties de logiciels opérationnels pour améliorer la satisfaction client. Elles se basent sur une approche qui encourage l'adaptation et le travail d'équipe pour favoriser l'amélioration continue. En général, le développement logiciel agile implique la réunion régulière et en personne de petites équipes autonomes de développeurs logiciels et de représentants métier tout au long du cycle de vie. La méthode agile privilégie une approche simplifiée plutôt que la documentation logicielle et favorise les changements à toutes les étapes du cycle de vie, au lieu d'y faire obstacle. »[1]

Les méthodes agiles sont apparues parce que les méthodes rigides et séquentielles de développement de logiciels ne pouvaient pas suivre l'évolution rapide des exigences et des priorités. En 2001, des leaders du développement de logiciels se sont réunis pour discuter d'idées communes et de diverses approches du développement de logiciels. À l'issue de cette réunion, ils ont rédigé le Manifeste Agile (*The Agile Manifesto*).

Le Manifeste Agile [2] précise douze principes communs à toutes les méthodes agiles :

- 1 Prioriser la satisfaction du client.
- 2 Accueillir positivement les changements de besoins, même tard dans le projet.
- 3 Livrer en permanence des versions opérationnelles de l'application.
- 4 Assurer une coopération entre les utilisateurs et les développeurs.
- 5 Réaliser les projets avec des personnes motivées.
- 6 Favoriser le dialogue direct.
- 7 Mesurer l'avancement en fonction de l'opérationnalité du logiciel.
- 8 Adopter un rythme constant et soutenable par tous les intervenants du projet.
- 9 Prêter attention à l'excellence technique et à une bonne conception.
- 10 Privilégier la simplicité en évitant le travail inutile.
- 11 Auto-organiser et responsabiliser les équipes.
- 12 Réfléchir régulièrement à améliorer l'efficacité de l'équipe.

### 1.1.2 La méthodologie adoptée :

Concernant la démarche du projet, nous avons choisi une démarche de gestion de projet appelée Kanban, dont le nom provient du japonais, qui peut être traduit comme 'carte de signalisation'.

Cette méthode a été initialement développée par Toyota dans les années 1940 et s'inscrivait dans une approche JAT (juste à temps) qui permettait aux usines de ne créer que les pièces nécessaires à un moment donné, et de ne pas gaspiller de ressources en fabriquant d'autres.

Les développeurs de logiciels se sont inspirés de ces idées pour créer leur propre version du système Kanban dans le cadre du mouvement de la méthodologie Agile.

Une carte Kanban représente une tâche unique, qui est déplacée dans les colonnes du tableau Kanban jusqu'à son achèvement.

Un tableau Kanban est une visualisation réelle de l'avancement d'un projet, qui reflète la réalité du flux de travail de l'équipe, et non la façon dont on souhaite que les choses se passent, ou la façon dont les parties prenantes pensent que les choses se font. Le tableau Kanban le plus simple ne comporte que trois colonnes: "À faire", "En cours" et "Terminé".

La méthode Kanban s'appuie sur **six pratiques** de base pour créer un ensemble de comportements positifs et obtenir une meilleure agilité. Ces pratiques doivent être appliquées, revues et améliorées sans relâche :

#### 1 Visualiser :

Nous devons visualiser tout ce qui nous aide à prendre des décisions, ainsi que les signaux et contrôles visuels qui nous indiquent quand nous devons agir ou quand il y a un problème.

#### 2 Limiter le travail en cours :

Observer, limiter puis optimiser la quantité de travail en cours est essentiel pour le succès de Kanban, car cela se traduit par un meilleur délai de livraison des services, une meilleure qualité et un taux de livraison plus élevé.

#### 3 Gérer le flux :

Le flux de travail dans un système Kanban doit maximiser la fourniture de valeur, minimiser les délais de livraison et être aussi prédictible que possible. Nous voulons que le travail soit acheminé le plus rapidement possible vers le client, avec une qualité et une sécurité maximales.

#### 4 Rendre les politiques explicites :

Nous devons veiller à ce que toutes les décisions, tous les processus, tous les critères et toutes les données soient explicites et visibles. Les limites du travail en cours (*Work In Progress*) sont un type de politique.

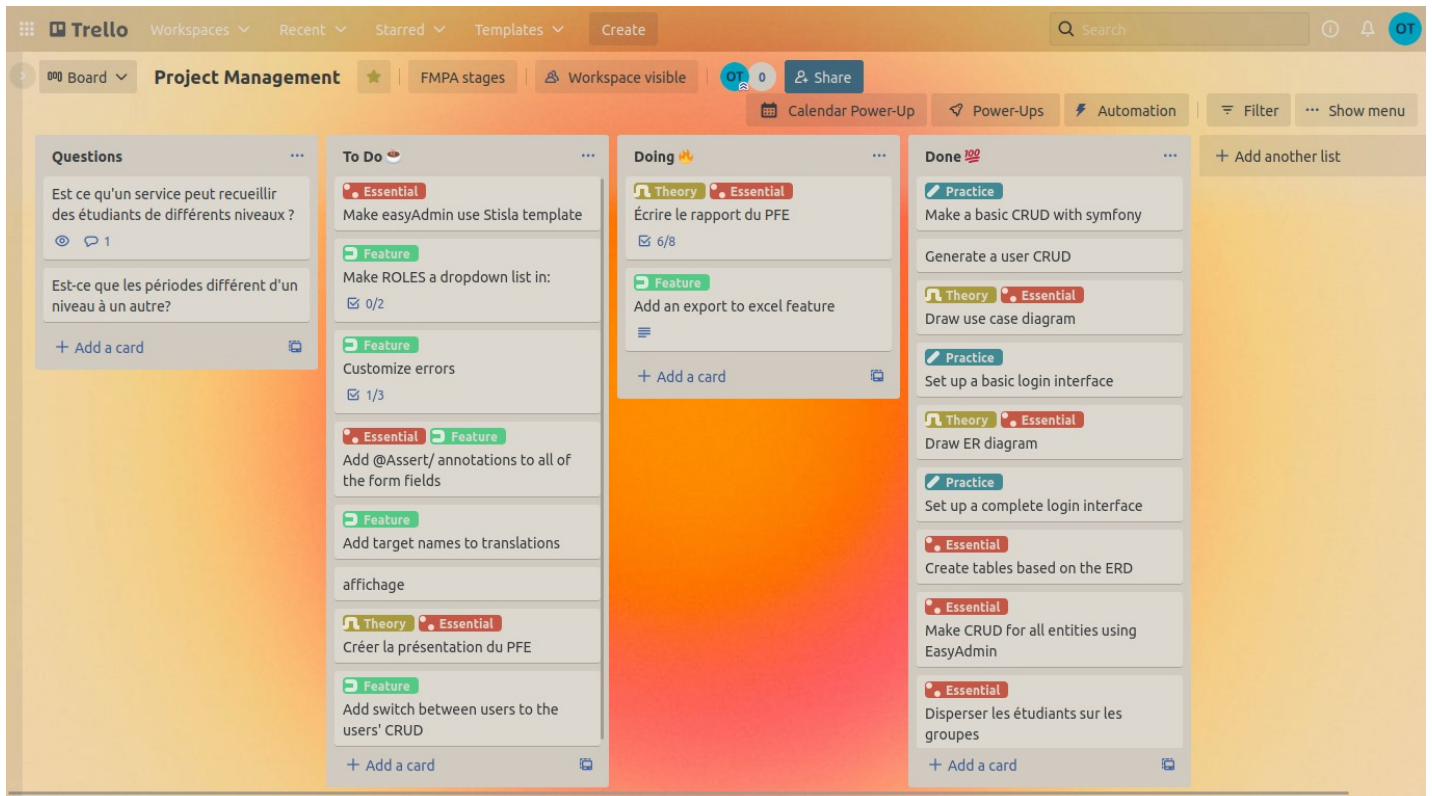
#### 5 Mettre en place des boucles de suivi (*feed-back*) :

Les boucles de suivi permettent de relier périodiquement les différents niveaux de décision en échangeant des informations pour une amélioration continue.

#### 6 Améliorer de manière collaborative, évoluer de manière expérimentale :

Toute amélioration doit être apportée en collaboration par cette équipe. Les systèmes Kanban doivent évoluer, c'est-à-dire commencer avec ce que nous avons et construire à partir de cela. L'évolution vient de l'expérimentation.

Pour implémenter la méthodologie Kanban nous avons remédié à l'application Trello (voir Figure 5).



**Figure 5 : Implémentation de la méthode Kanban avec Trello**

La motivation derrière le choix de cette méthode est que la méthode Kanban est très agile, et que le flux de travail est visualisé, ce qui a amélioré notre productivité.

## **b. Planification du projet :**

La planification fait partie des étapes les plus importantes pour la réalisation efficace d'un projet, elle permet de s'assurer que ce dernier va atteindre ses objectifs dans des délais bien définis et contrôlés.

Le diagramme GANTT permet d'évaluer et de suivre la réalisation du projet tout en spécifiant les phases de chaque partie et les délais pour lesquels elles doivent être réalisées. Le diagramme de GANTT, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités et tâches qui constituent les phases d'un projet. Le tableau d'un diagramme GANTT énumère toutes les tâches à effectuer et la ligne d'en-tête du diagramme représente les unités de temps les plus adaptés au projet (jours, semaines, mois, etc.), nous avons fait cette représentation sur le logiciel Gantt Project.



**Figure 6 : Logo de Gantt Project**

Gantt Project est un logiciel libre permettant de modifier la planification des différentes tâches qui sont nécessaires à un projet.

Nom	Date de début	Date de fin
Lancement du projet	04/04/2022	04/04/2022
Compréhension de la problématique	05/04/2022	05/04/2022
Étude et état d'art	06/04/2022	07/04/2022
Formation Symfony	07/04/2022	30/04/2022
Formation Docker	12/04/2022	15/04/2022
Élaboration du modèle conceptuel	16/04/2022	20/04/2022
Création des CRUD des entités	18/04/2022	03/05/2022
Rédaction du rapport	30/04/2022	05/06/2022
Génération des calendriers (coté vue: Twig)	12/05/2022	15/05/2022
Génération des calendriers (coté contrôleur: PHP)	16/05/2022	22/05/2022
Affectation des groupes aux étudiants	20/05/2022	23/05/2022
Validation des résultats	24/05/2022	24/05/2022
Ajout de l'interface professeur	01/06/2022	07/06/2022
Validation des résultats	10/06/2022	10/06/2022
Amélioration de l'UI/UX	10/06/2022	12/06/2022
Préparation de la présentation	11/06/2022	15/06/2022

**Figure 7 : Table du diagramme de Gantt**

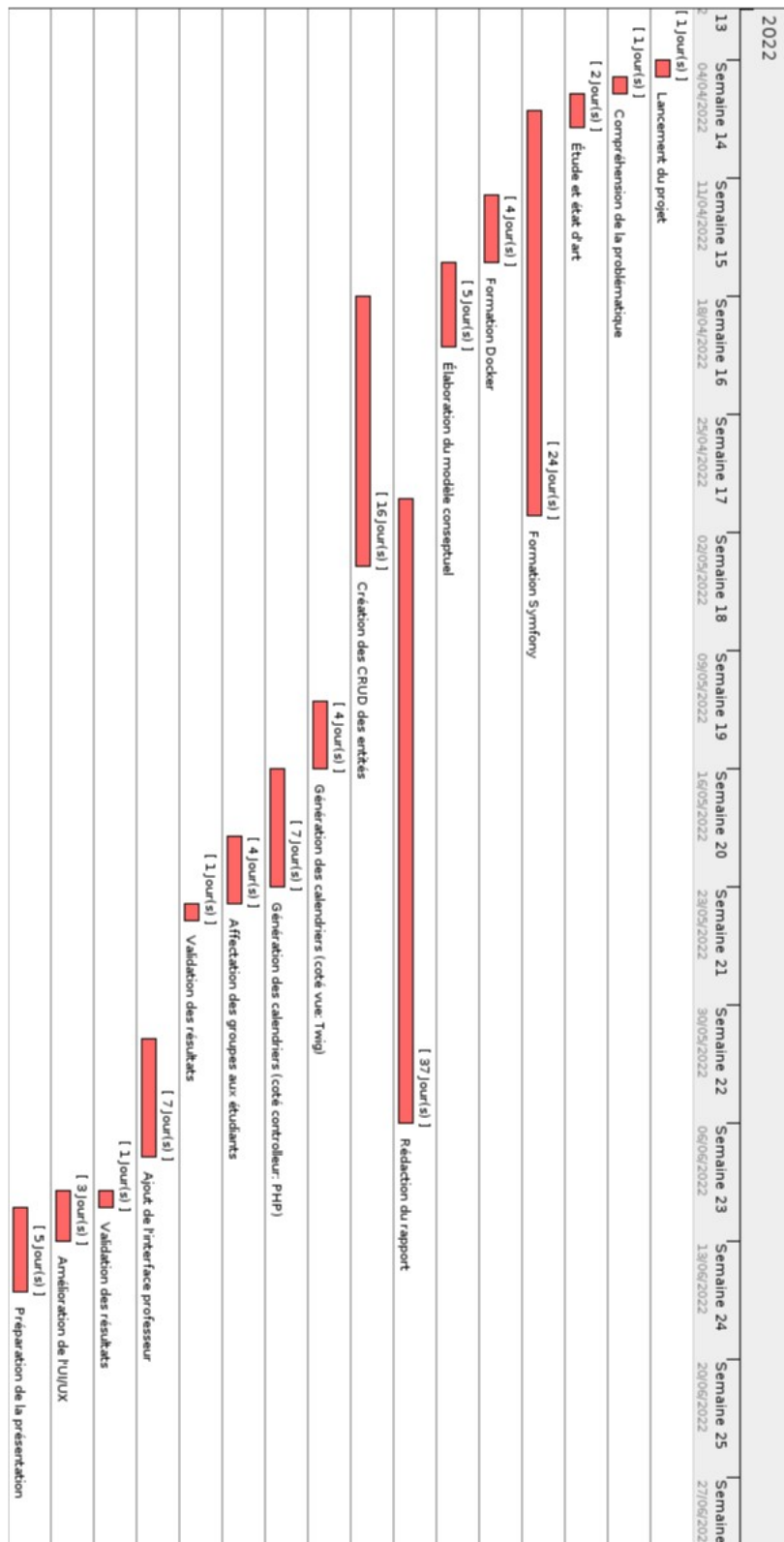


Figure 8 : Diagramme de Gantt

## Conclusion :

Ce chapitre a été consacré à la présentation de l'organisme d'accueil, ainsi qu'à l'explication du contexte de notre projet et de la manière avec laquelle nous avons planifié la solution pour la déployer à temps, et pour permettre aux usagers d'exécuter leurs tâches relatives aux stages avec plus de facilité et précision. Le chapitre suivant discutera notre analyse des besoins et des spécifications, afin d'obtenir une architecture adéquate de notre solution.

## Chapitre 2 : Analyse et conception

*Dans ce chapitre, nous entamerons l'analyse et la spécification des besoins. Nous commençons par définir les besoins fonctionnels et non fonctionnels de notre application. On établit ensuite le diagramme de cas d'utilisation. Ce dernier donne une vision globale du comportement fonctionnel de la solution.*

## 1 Spécification des besoins :

Afin de réaliser notre projet, il est impératif d'explicitier les différents besoins fonctionnels et non fonctionnels, comme suit :

### ○ **Besoins fonctionnels :**

L'application à réaliser doit offrir un ensemble de fonctionnalités qui doivent être mises en relation avec un ensemble de besoins des administratifs ainsi que des professeurs. Ces derniers définissent les services que les utilisateurs s'attendent à voir fournis par cette application.

La présente application web doit satisfaire différents besoins fonctionnels en fonction de l'utilisateur :

- **L'administration :**
  - Gérer les AU.
  - Gérer les catégories.
  - Gérer les groupes.
  - Gérer les hôpitaux.
  - Gérer les niveaux.
  - Gérer les passages (calendriers).
  - Gérer les périodes.
  - Gérer les professeurs.
  - Gérer les services.
  - Gérer les stages.
  - Gérer les étudiants.
  - Gérer les inscriptions.
  - Gérer les notes.
  - Gérer les utilisateurs de l'application.
- **Les professeurs :**
  - Consulter la liste des étudiants qui passent des stages dans leurs services.
  - Noter ces étudiants.
- **L'application de gestion de scolarité :**
  - Consulter les données des étudiants.
  - Insérer les données des étudiants.
  - Consulter les calendriers.

### ○ **Besoins non fonctionnels :**

Pour notre application, nous avons pris en considération les critères ergonomiques de Scapin et Bastien, cités dans leur rapport [3], et qui sont comme suit:

- **Guidage :** L'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur;
- **Groupement/Distinction entre Items:** Groupement des différents éléments visuels de façon cohérente et ordonnée;
- **Feed-back immédiat:** Dans tous les cas, l'application doit répondre à l'utilisateur en fonction des actions et des requêtes de ce dernier;
- **Brièveté:** La brièveté correspond à l'objectif de limiter la charge de travail de lecture et d'entrée ainsi que le nombre d'étapes d'action.
- **Adaptabilité:** Capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs;
- **Prise en compte de l'expérience de l'utilisateur:** Le système doit respecter le niveau d'expérience de l'utilisateur; donc il doit être simple, compréhensible et facile à utiliser;



- **Correction des erreurs:** Mettre à la disposition des utilisateurs des moyens pour corriger leurs erreurs. Par exemple, fournir la possibilité de modifier les commandes lors de leur saisie;
- **Compatibilité:** Il faut qu'il y ait accord entre les caractéristiques des utilisateurs et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. Par exemple, les termes employés doivent être familiers aux utilisateurs, et relatifs à la tâche à réaliser.

## 2 Cas d'utilisations

### ○ Identification des acteurs :

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une ou plusieurs fonctions concernant les cas d'utilisations.

Il existe 3 types d'acteurs dans notre application:

- **L'administratif:** C'est un utilisateur avec pouvoir, il est le responsable de l'administration de la plateforme, allant de remplir les informations nécessaires pour générer les calendriers de stage jusqu'à l'attribution des étudiants aux différents groupes.
- **Le professeur:** C'est un utilisateur avec pouvoir restreint, il note les étudiants qui ont passé un stage dans son service.
- **L'application de gestion de scolarité via une API dédiée:** C'est un usager qui peut lire ou modifier les informations des étudiants.

### ○ Diagrammes des cas d'utilisations :

Après l'étude et l'analyse des besoins, les cas d'utilisation de notre solution commencent à apparaître plus clairs, et peuvent se modéliser dans les diagrammes des cas d'utilisations suivants :

■ Diagramme de cas d'utilisation de l'administratif :

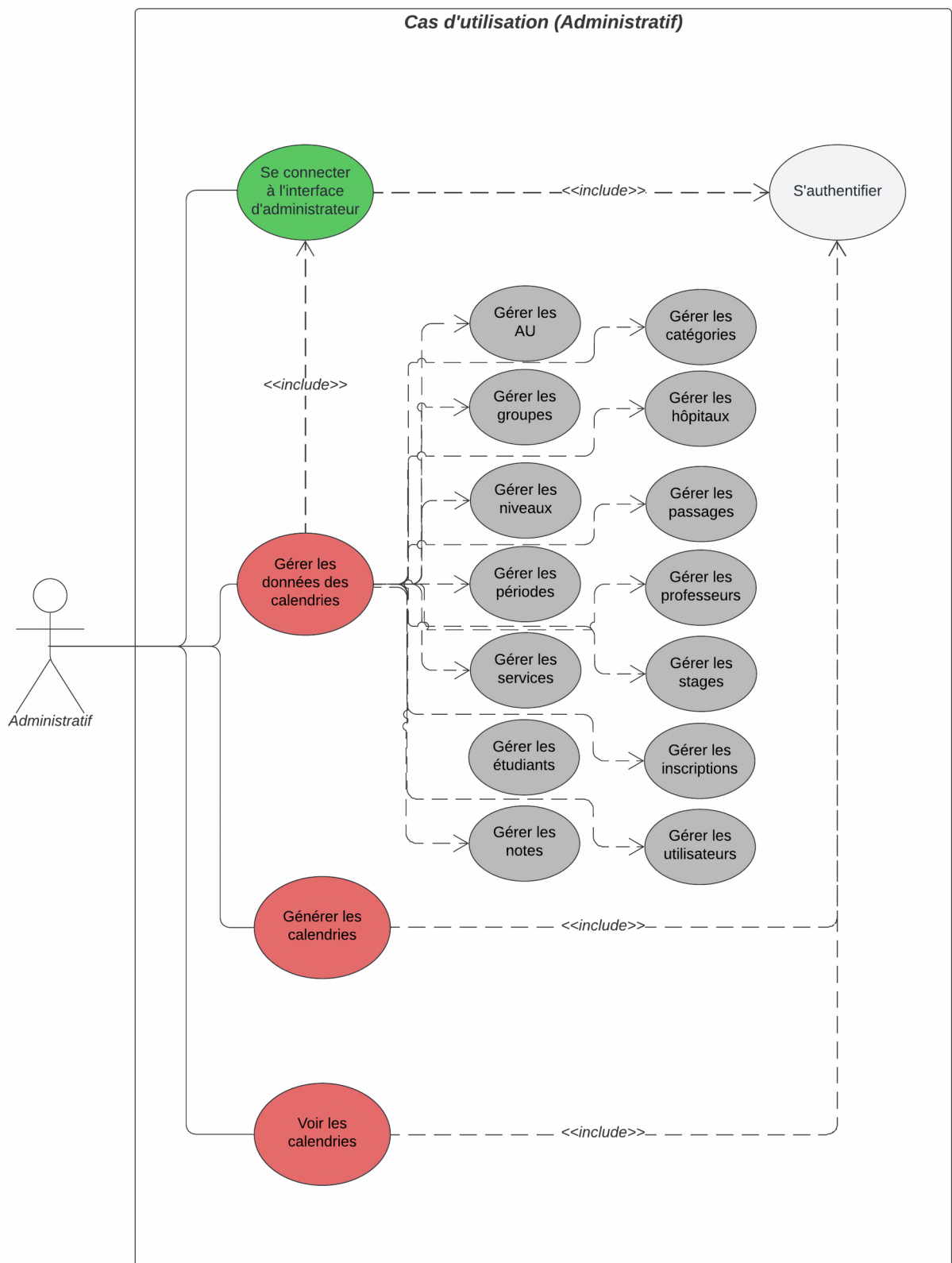
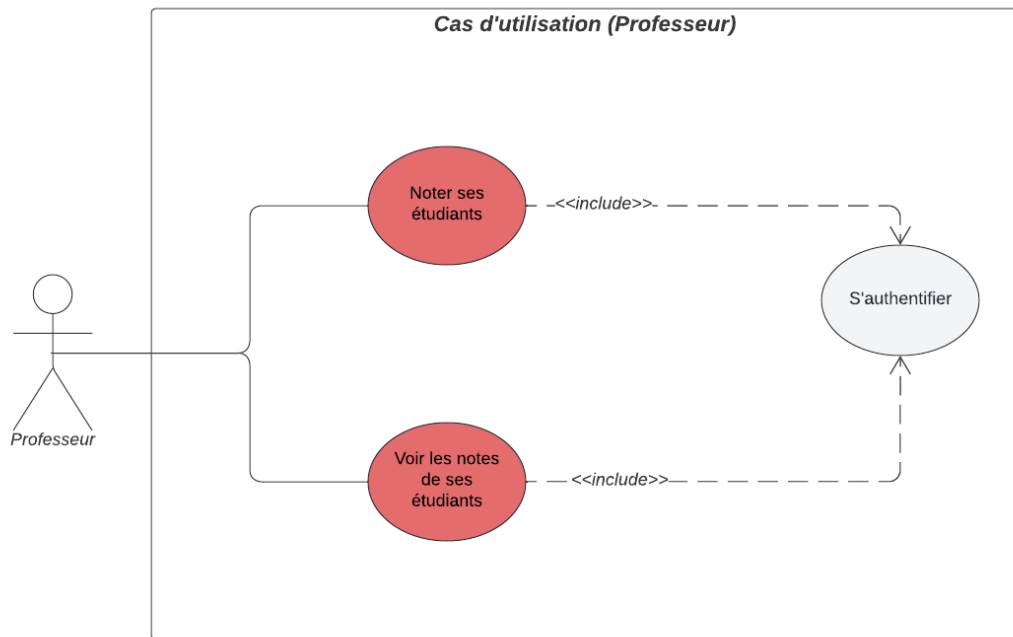


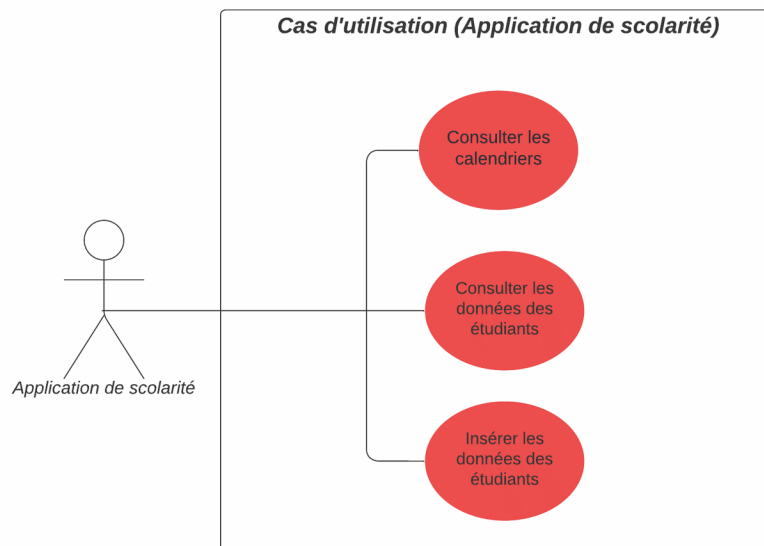
Figure 9 : Diagramme de cas d'utilisation de l'administratif

- **Diagramme de cas d'utilisation d'un professeur:**



**Figure 10 : Diagramme de cas d'utilisation d'un professeur**

- **Diagramme de cas d'utilisation de l'application de scolarité:**



**Figure 11 : Diagramme de cas d'utilisation de l'application de scolarité**

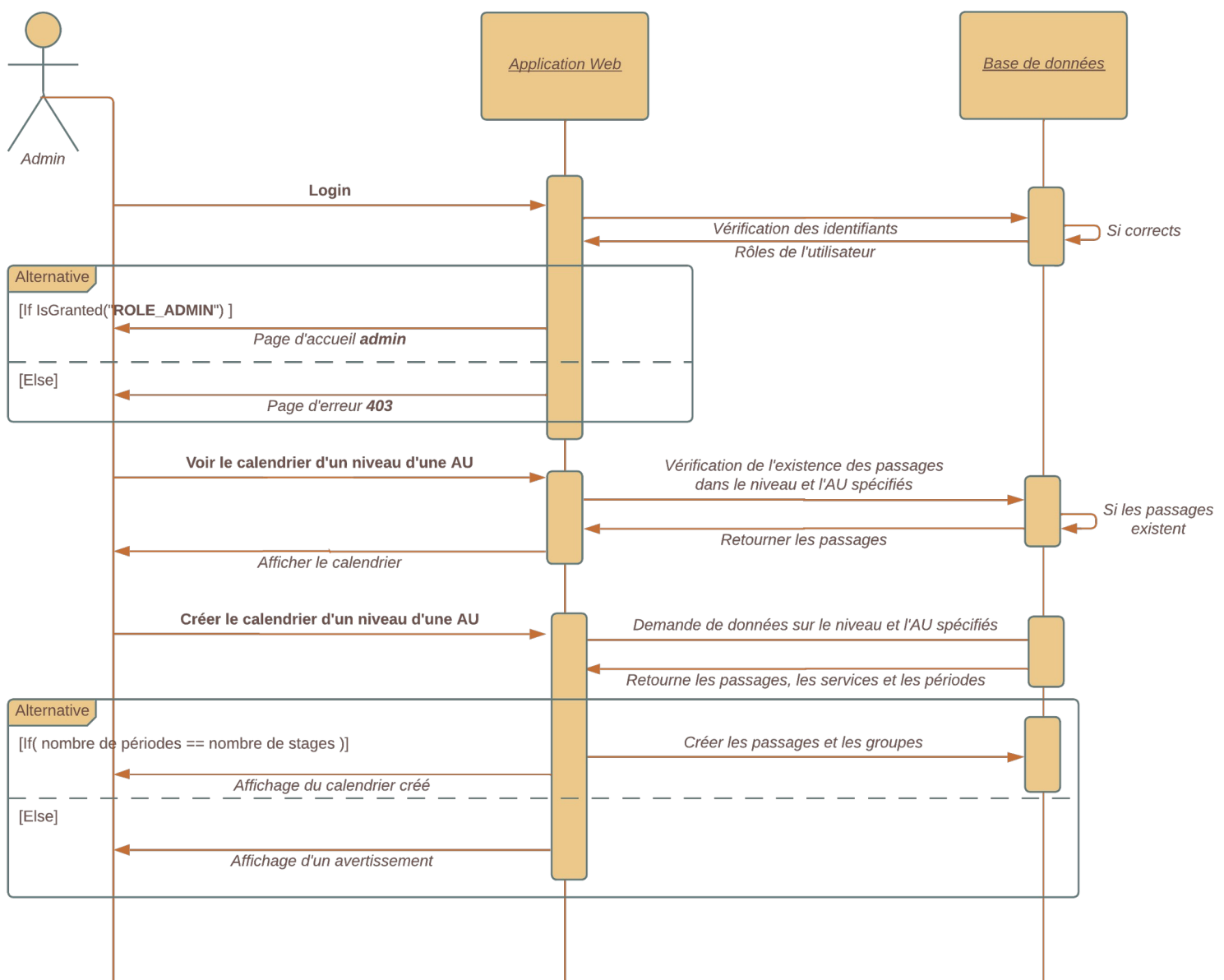
### 3 Diagrammes de séquence :

Les diagrammes de séquence sont une solution populaire de modélisation dynamique en langage UML, car ils se concentrent plus précisément sur les *lignes de vie*, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.

Dans cette partie, nous allons présenter les interactions des objets du système par des diagrammes de séquence pour les scénarios des cas d'utilisations les plus importants.

- **Diagramme de séquence de l'administratif :**

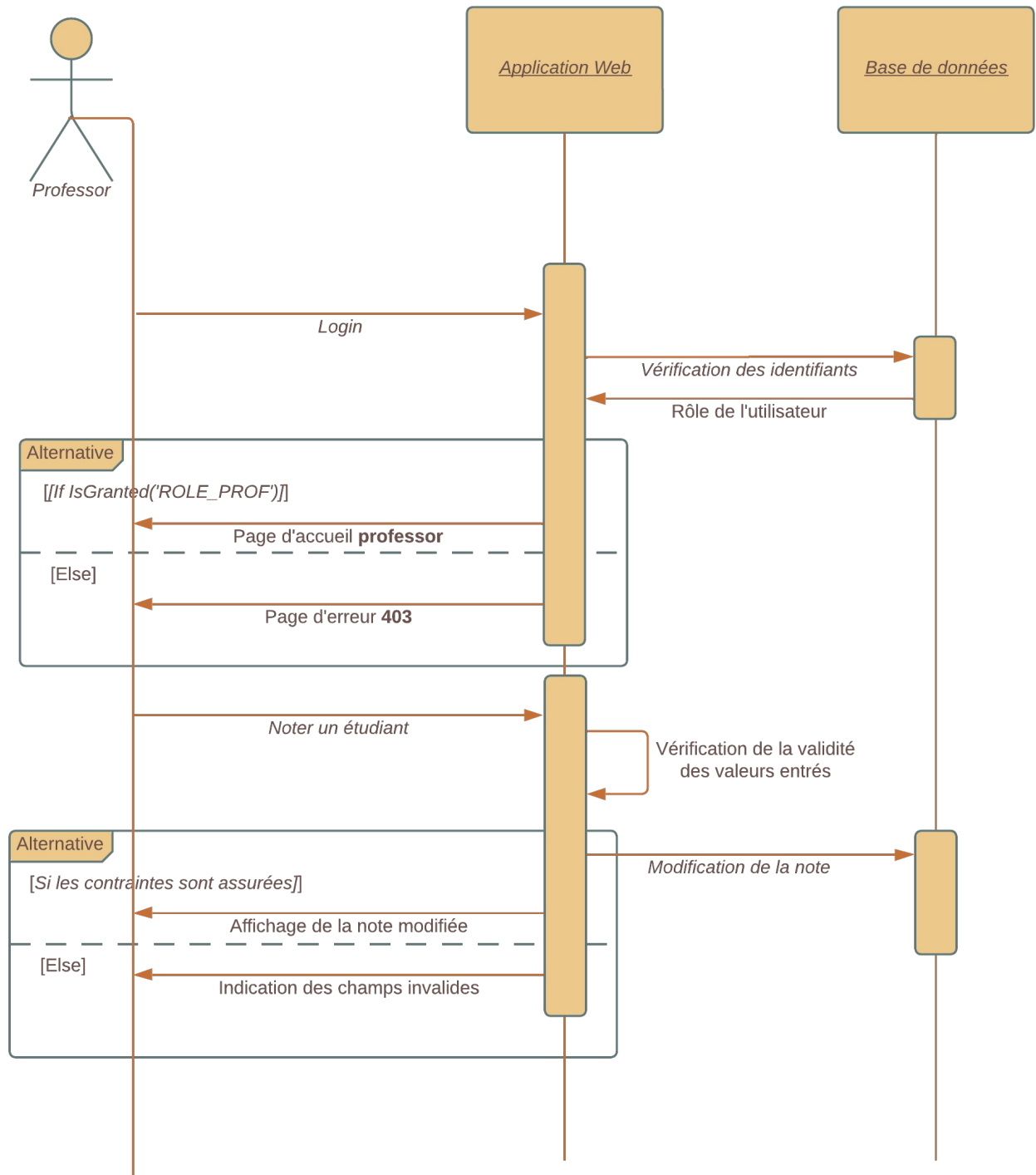
Diagramme de séquence administratif



**Figure 12 : Diagramme de séquence de l'administratif**

- **Diagramme de séquence professeur :**

### Diagramme de séquence professeur



**Figure 13 : Diagramme de séquence d'un professeur**

## 4 Architecture du système :

Selon les précédents diagrammes d'utilisations (Voir figures 9, 10 et 11), nous avons été capables de proposer une architecture à notre solution. (Voir figure 14)

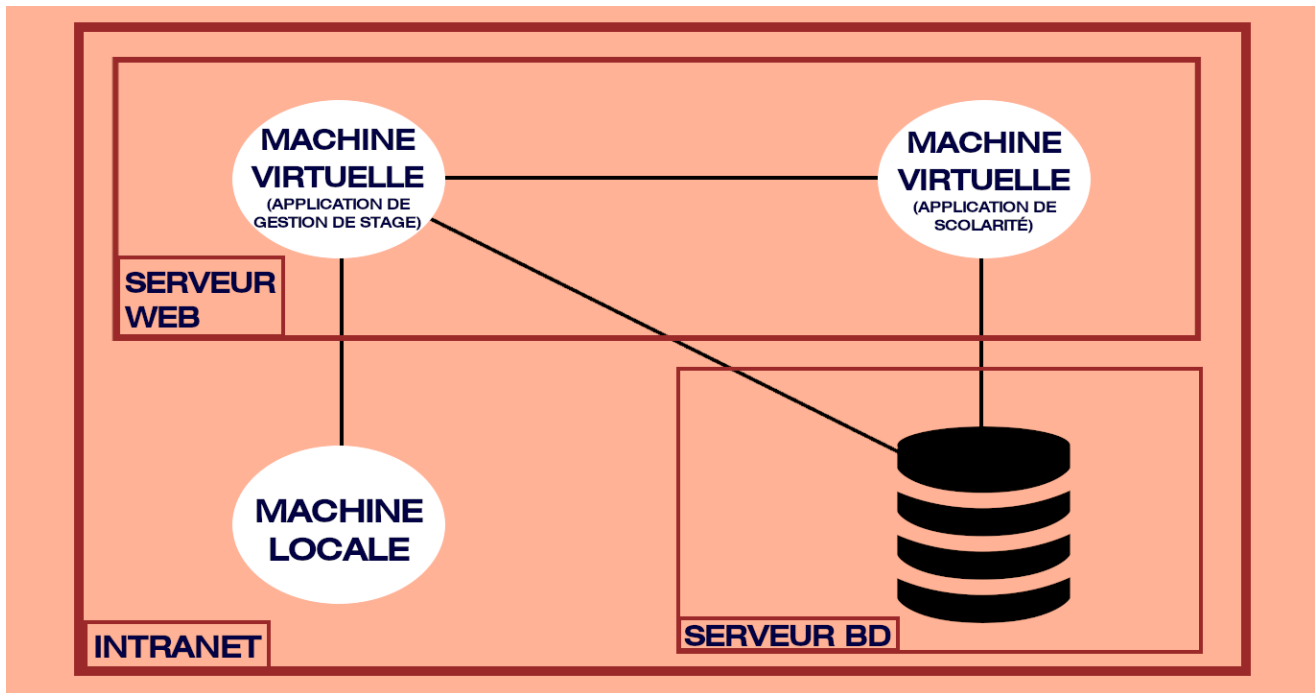


Figure 14 : Architecture de la solution

## 5 Modèle de données :

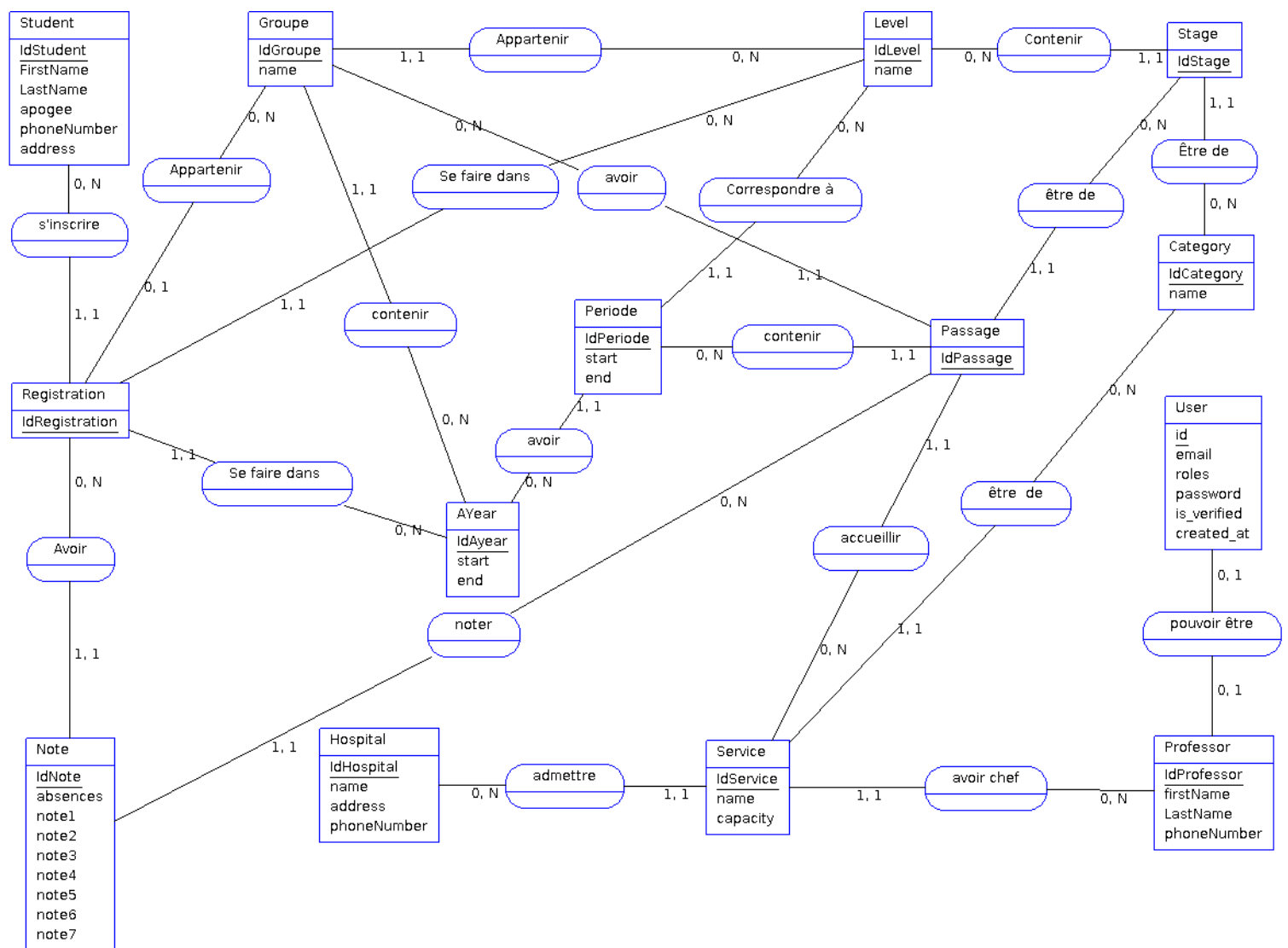
### ◦ MCD :

MCD (Modèle Conceptuel des Données) est un diagramme de haut niveau permettant de donner une représentation schématique de tout ou partie d'une base de données relationnelle. Les MCD font partie plus généralement d'une méthode de conception d'origine française appelée la méthode MERISE, laquelle définit un certain nombre de schémas et diagrammes.

Le MCD, en particulier, vise en outre à modéliser des tables et les relations existantes entre elles selon un ensemble de règles et de diagrammes codifiés :

- Les entités (représentées par des rectangles)
- Les propriétés (la liste des données de l'entité)
- Les relations qui expliquent et précisent comment les entités sont reliées entre elles (représentées par des ovales)
- Les cardinalités (0,1, 1,1, 0,N ou 1,N)

Le diagramme MCD de notre application est représenté dans la figure 15.



**Figure 15 : Modèle conceptuel des données**

## Conclusion :

Tout au long de ce chapitre, nous avons analysé les différents besoins, puis nous avons détaillé la conception de notre application à travers les diagrammes des cas d'utilisations, les diagrammes de séquence ainsi que le modèle de conception de données afin que la phase réalisation et la mise en place de l'application soit plus souple et plus aisée. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application proposée.



## Chapitre 3 : Architecture et environnement

*Dans ce chapitre, nous allons aborder les outils et technologies utilisées pour la mise en place de notre solution. Pour assurer la pérennité de notre projet, le choix des outils a requis une réflexion importante ainsi que des projections futures de l'utilisation de ces technologies dans notre projet.*

## 1 Architecture adoptée :

Modèle-vue-contrôleur (MVC), est un motif d'architecture logicielle destiné aux interfaces graphiques et qui est très populaire pour les applications web.

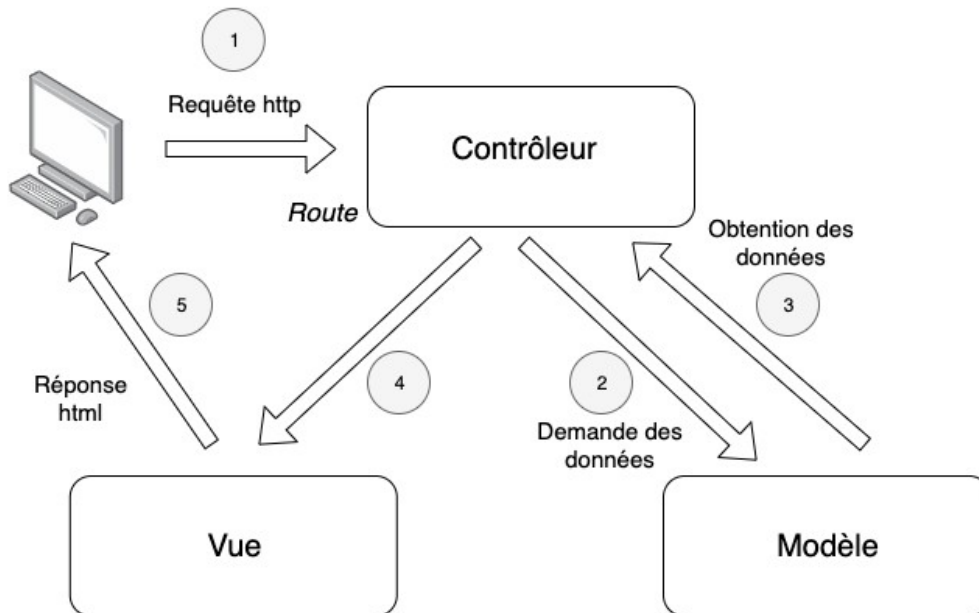


Figure 16 : Architecture MVC

Le modèle est composé de trois types de modules ayant trois responsabilités différentes: les modèles, les vues et les contrôleurs.

- Un modèle (*Model*) contient les données à traiter (insérer, lire, modifier ou supprimer).
- Une vue (*View*) contient la présentation de l'interface graphique (*UI*), elle se sert du modèle. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle.
- Un contrôleur (*Controller*) contient la logique concernant les actions effectuées par l'utilisateur.

L'objectif principal de l'architecture MVC est de séparer la logique commerciale et les données de l'application de l'interface utilisateur, et par conséquent améliorer la réutilisabilité, la sécurité et la performance de l'application.

## 2 Choix du langage :

### PHP :

PHP (PHP: Hypertext Preprocessor), est un langage de script libre utilisé le plus souvent côté serveur, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage orienté objet.



**Figure 17 : Logo de PHP**

Dans une architecture MVC, 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 web.

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** ou **nginx**. 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.

Nous avons utilisé PHP 8.1, puisque c'est la dernière version maintenue disponible lors du développement de notre application.

### **3 Framework et container :**

#### **3.1 Symfony :**



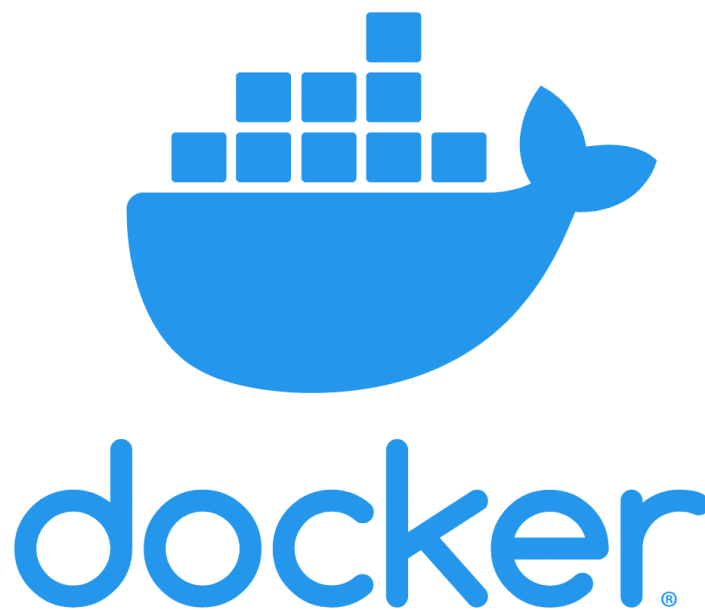
**Figure 18 : Logo Symfony**

Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

L'agence web française SensioLabs est à l'origine du framework Sensio Framework. À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion ORM, etc., elle a développé ce framework pour ses propres besoins. Comme ces problématiques étaient souvent les mêmes pour d'autres développeurs, le code a été par la suite partagé avec la communauté des développeurs PHP. Le projet est alors devenu Symfony (conformément à la volonté du créateur de conserver les initiales S et F de Sensio Framework).

Nous avons choisi le framework Symfony principalement car c'est le framework utilisé dans la plupart des applications web maintenues par le service informatique de la FMPPA. Les responsables du service peuvent ainsi aisément ajouter des fonctionnalités et assurer la maintenance de notre application.

### **3.2 Docker :**



**Figure 19 : Logo de Docker**

Docker est un logiciel libre permettant de lancer des applications dans des conteneurs logiciels. C'est une plateforme ouverte pour le développement, l'expédition et l'exécution d'applications. Docker permet de séparer les applications de l'infrastructure afin que la livraison des logiciels soit rapide. En tirant parti des méthodologies de Docker pour expédier, tester et déployer rapidement le code, le délai entre l'écriture du code et son exécution en production peut se réduire considérablement.[4]

Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement. Cette approche permet d'accroître la flexibilité et la portabilité d'exécution d'une application, laquelle va pouvoir, avec l'aide d'autres outils DevOps, faciliter l'intégration continue (CI) et la livraison continue (CD).

## Conclusion :

Au cours de ce chapitre, nous avons conçu les différentes composantes de notre système. Après ce choix de technologies, nous nous sommes lancés dans la réalisation de notre solution. Le prochain chapitre permettra de présenter l'environnement logiciel et les outils utilisés dans la réalisation de cette solution, ainsi que les résultats obtenus.

## Chapitre 4 : Réalisation, interfaces, tests

*La phase de réalisation consiste à concrétiser notre projet par la réalisation des fonctionnalités et interfaces du système. Dans cette dernière partie nous commencerons par la description des environnements, matériel et logiciel, utilisés pour développer notre application, ensuite nous présenterons le résultat final.*

## 1 Environnements et outils :

### 1.1 Environnement logiciel:

- **PhpStorm :**



**Figure 20 : Logo de PhpStorm**

PhpStorm est un IDE (environnement de développement intégré) multiplateforme pour PHP, HTML, CSS et JavaScript, conçu par la société JetBrains. Il est écrit en Java.

L'éditeur nous a offert :

- Une coloration syntaxique dynamique ;
- Des analyseurs d'erreurs syntaxiques et orthographiques (en anglais) ;
- Une auto-complétion intelligente du code en PHP ;
- Des fonctionnalités de réusinage automatisé de code en PHP.

Enfin, il permet l'intégration d'outils d'opérations serveur comme Docker, une console **SSH** et d'autres outils.

### 1.2 Outils :

- **DDEV :**

DDEV est un outil open source qui simplifie à l'extrême la mise en place et le fonctionnement d'environnements de développement PHP locaux en quelques minutes. Il est puissant et flexible grâce à ses configurations d'environnement par projet, qui peuvent être étendues, contrôlées par version et partagées.



Figure 21 : Logo de DDEV

DDEV nous a permis d'utiliser Docker dans notre flux de travail sans les complexités d'une configuration sur mesure (création des fichiers docker-compose et intégration avec le DNS local pour la création des sous-domaines du développement), surtout avec son intégration de **SSH** ce qui nous a facilité l'utilisation des clés à l'intérieur du conteneur pour accéder aux hôtes ou aux dépôts privés du **composer**.

- **Portainer:**



Figure 22 : Logo de Portainer

Portainer est un organisateur de conteneurs. Il transforme le code sur les lignes de commande en une interface web qui peut créer et surveiller de nouveaux conteneurs dans le socket du docker.[5]

Portainer est une application simple et légère, mais puissante. Elle nous a permis de gérer les différents containers et images de notre application, à partir d'une interface de gestion Web sur notre hôte Docker.



- Github :



**Figure 23 : Logo de GitHub**

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant Git, qui est un logiciel libre de gestion de versions décentralisé, créé par Linus Torvalds, auteur du noyau Linux. La gestion de versions (*version control*) consiste à gérer l'ensemble des versions d'un ou plusieurs fichiers.



**Figure 24 : Logo de git**

GitHub nous a offert un moyen simple et agréable de disposer en ligne le code de notre application et, plus important encore, l'historique des versions, indépendamment de ce qui arrive à notre machine locale.

- **Lucidchart :**



**Figure 25 : Logo de Lucidchart**

Lucidchart est une plateforme de collaboration en ligne, basée sur le cloud, permettant la création de diagrammes et la visualisation de données, et autres schémas conceptuels.

Lucidchart nous a permis de créer les différents diagrammes utilisés dans notre projet.

- **Trello :**



**Figure 26 : Logo de Trello**

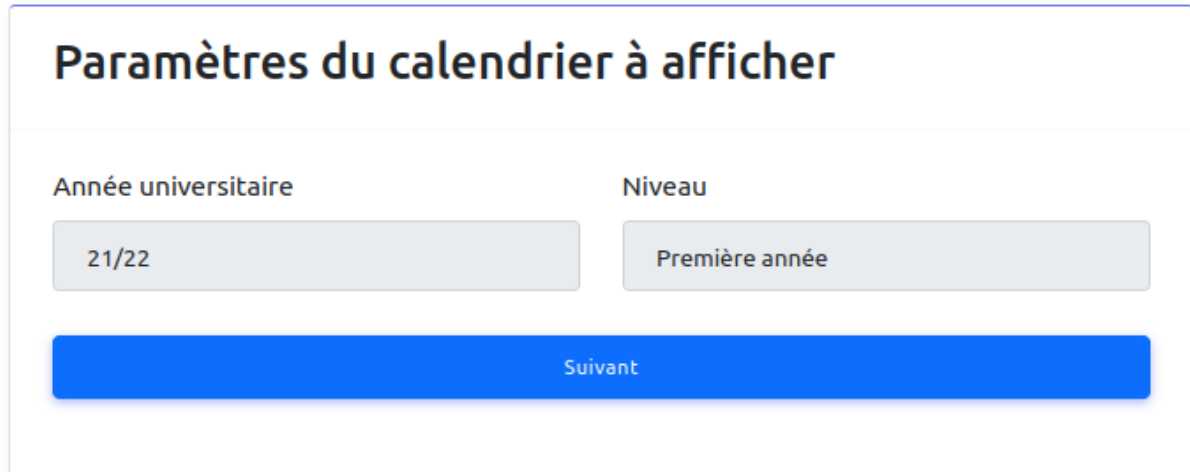
Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant l'avancement.

Trello a simplifié la gestion de notre projet en nous permettant de subdiviser nos tâches en tâches plus petites, ainsi que de bénéficier de toutes les avantages de la méthode Kanban.

## 2 Interfaces et tests :

### 2.1 Interface voir un calendrier :

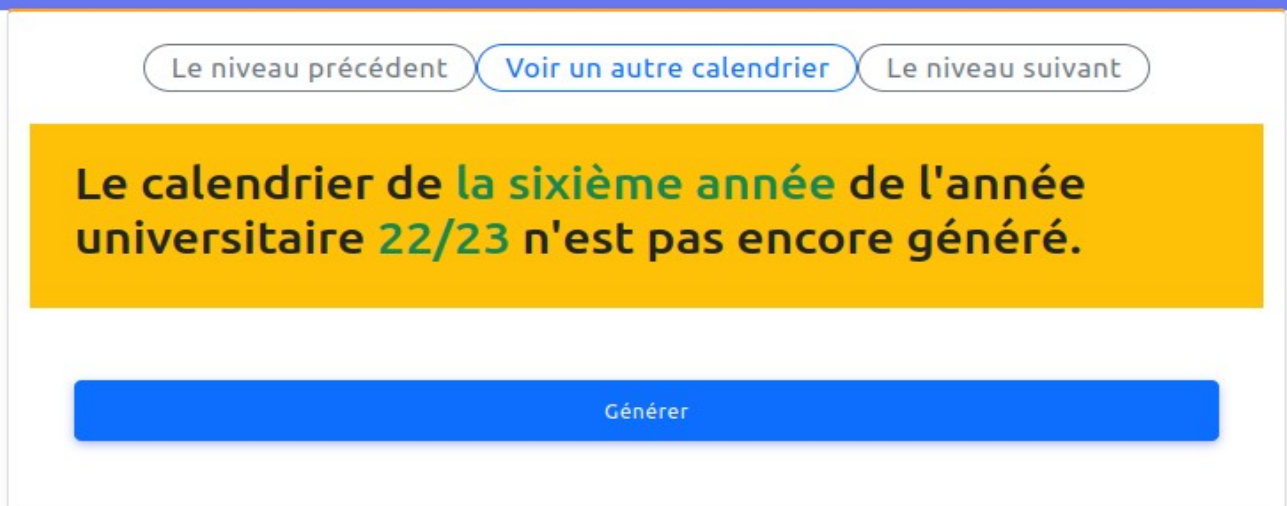
L'administrateur commence par choisir l'année universitaire et le niveau du calendrier à afficher :



The screenshot shows a web form titled "Paramètres du calendrier à afficher". It contains two input fields: "Année universitaire" with the value "21/22" and "Niveau" with the value "Première année". Below these fields is a blue button labeled "Suivant".

**Figure 27 : Paramètres du calendrier à afficher**

Si le calendrier n'est pas encore généré, c'est-à-dire qu'il n'existe pas de passages (Voir annexes 1 et 2) correspondants à ce niveau de cette année universitaire .



The screenshot shows a web interface with three buttons at the top: "Le niveau précédent", "Voir un autre calendrier", and "Le niveau suivant". Below these buttons is a yellow box containing the text: "Le calendrier de la sixième année de l'année universitaire 22/23 n'est pas encore généré." At the bottom of the interface is a blue button labeled "Générer".

**Figure 28 : Le calendrier n'est pas encore généré**

Si le calendrier a déjà été généré (Voir annexes 1 et 3). On l'affiche :

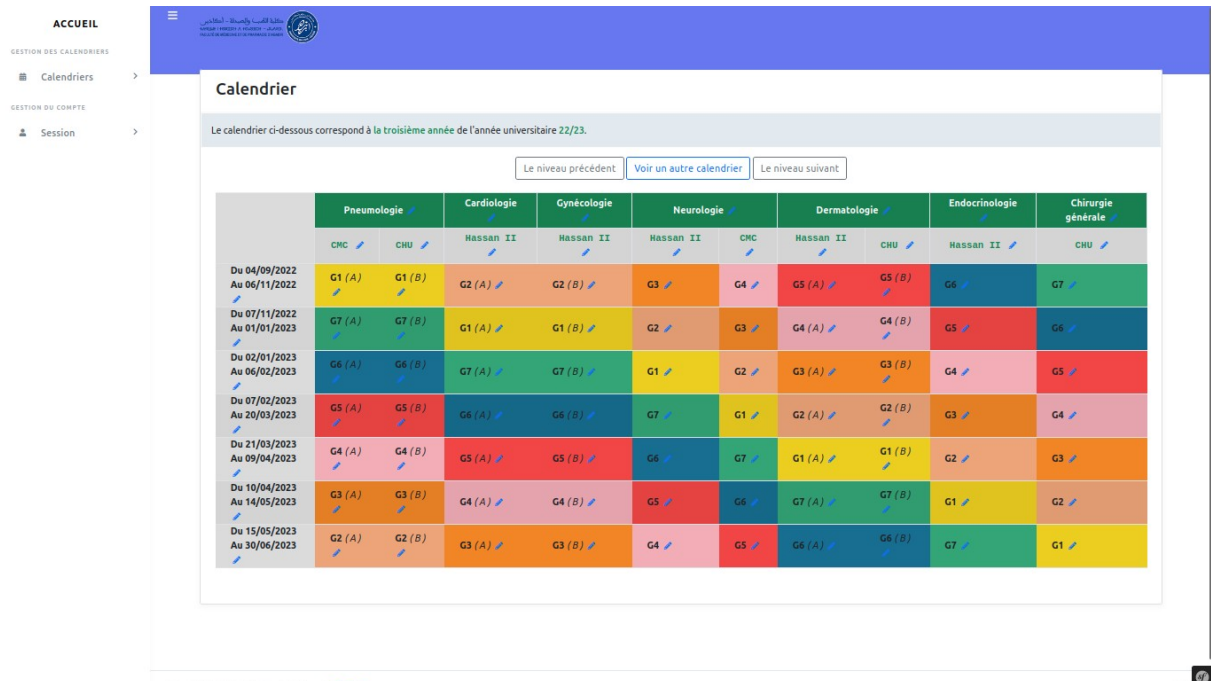


Figure 29 : Affichage du calendrier.

## 2.2 Interface générer un calendrier :

L'administrateur commence par choisir l'année universitaire et le niveau du calendrier à générer :

Figure 30 : Paramètres du calendrier à générer.

L'administrateur poursuit vérifie les périodes, les stages et les services relatifs à l'AU et le niveau qu'il a choisi précédemment.

Si le nombre de périodes est égal à celui des stages à passer on lui permet de poursuivre :




Par la suite, l’administrateur valide les étudiants inscrits dans le niveau et l’AU spécifiés.  
Si les étudiants n’ont pas encore de groupe, on lui permet de poursuivre :

Confirmer les étudiants

Liste des 21 étudiants inscrits dans la troisième année de l'année universitaire 22/23.

Charger le niveau ou l'année universitaire ?

Étudiant	Photo	APOGEE	Modifier un étudiant	Modifier une inscription	Groupe
MOHAMED Mohamed	image_etudiant				Pas encore affecté
IBRAHIM Ibrahim	image_etudiant				Pas encore affecté
AMIR Amir	image_etudiant				Pas encore affecté
NAIM Naim	image_etudiant				Pas encore affecté
AYOUB Ayoub	image_etudiant				Pas encore affecté
YOUSSEF Youssef	image_etudiant				Pas encore affecté
WASSIM Wassim	image_etudiant				Pas encore affecté
HAMZA Hamza	image_etudiant				Pas encore affecté
YASSINE Yassine	image_etudiant				Pas encore affecté
SOFIANE Sofiane	image_etudiant				Pas encore affecté
NORA Nora	image_etudiant				Pas encore affecté
MARIAM Maryam	image_etudiant				Pas encore affecté
ASSIA Assia	image_etudiant				Pas encore affecté
YASMINE Yasmine	image_etudiant				Pas encore affecté
AYA Aya	image_etudiant	777			Pas encore affecté
NOUR Nour	image_etudiant				Pas encore affecté
OUSSAMA Oussama		17021862			Pas encore affecté

Inscrire un étudiant +

Ajouter un nouveau étudiant +

Suivant

Figure 33 : Confirmer les étudiants inscrits.


Sinon, on affiche un message d’avertissement :

NOUR Nour

image\_etudiant

G6A

OUSSAMA Oussama



17021862

G7A

Inscrire un étudiant +

Ajouter un nouveau étudiant +

0

Au moins un étudiant est déjà associé à un groupe!

- Disassocier les groupes des étudiants, en cliquant sur [\(Très recommandé\)](#).
- Si vous voudrez plutôt associer des groupes manuellement:
  - [Affecter](#) des groupes existants aux étudiants [\(Recommandé\)](#).
  - Créer des [nouveaux groupes](#), puis les [affecter](#) aux étudiants. [\(Non recommandé\)](#)

Si vous poursuivez tout groupe affecté à un étudiant sera remplacé par un nouveau groupe.

Suivant

Figure 34 : Au moins un étudiant a déjà un groupe.

Si l’administrateur ignore l’avertissement et clique sur ‘Suivant’, l’application le réavertit une deuxième fois (pour diminuer la probabilité d’erreur de clic) :

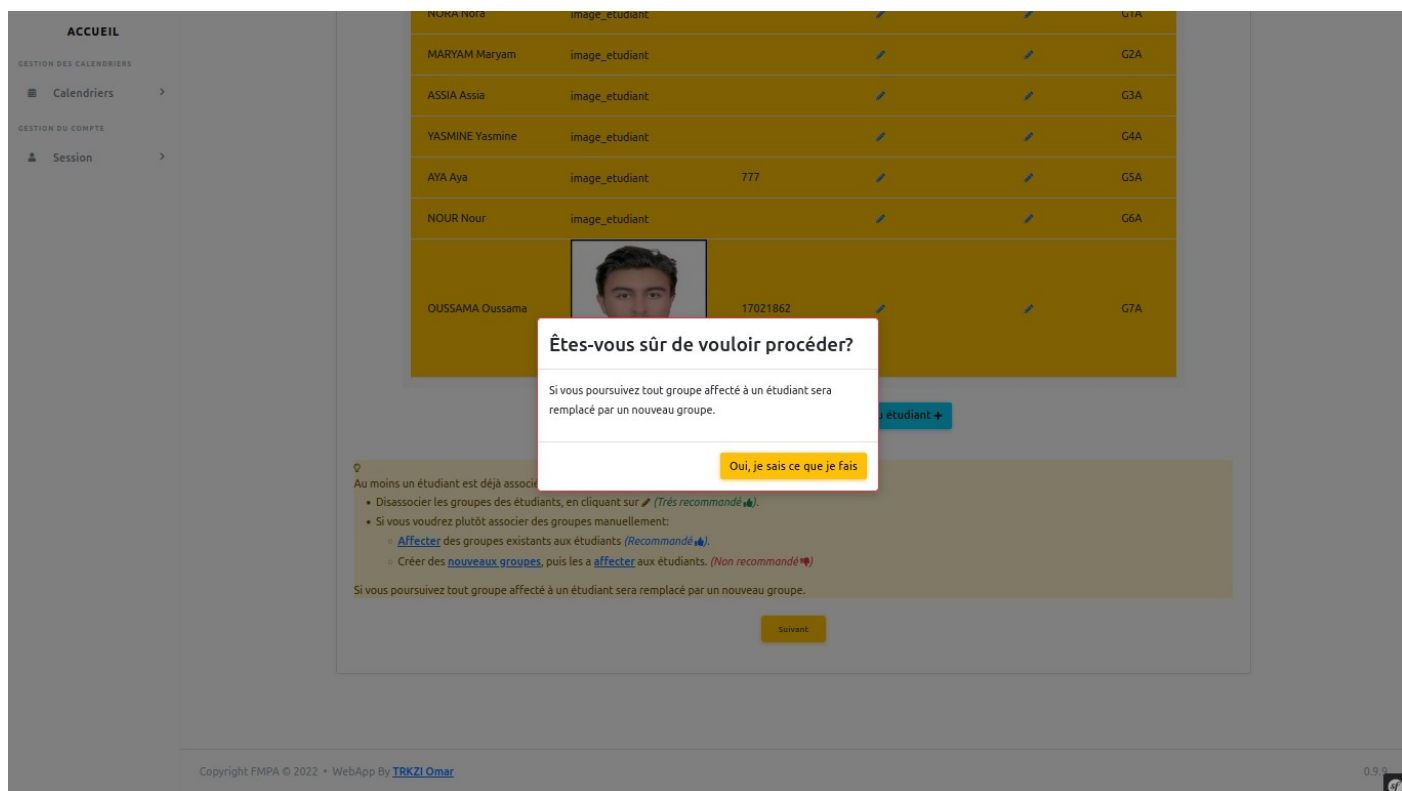


Figure 35 : Avertissement des groupes.

Parsuite, on affiche la proposition du calendrier (généré avec du Twig) :

Confirmer le calendrier

La proposition du calendrier ci-dessous correspond à la troisième année de l'année universitaire 22/23.

Changer le niveau ou l'année universitaire ?

	Pneumologie		Cardiologie	Cynécologie	Neurologie		Dermatologie		Endocrinologie	Chirurgie générale
	CMC	CHU	Hassan II	Hassan II	Hassan II	CMC	Hassan II	CHU	Hassan II	CHU
Du 04/09/2022 Au 06/11/2022	G1 (A)	G1 (B)	G2	G3	G4 (A)	G4 (B)	G5 (A)	G5 (B)	G6	G7
Du 07/11/2022 Au 01/01/2023	G7 (A)	G7 (B)	G1	G2	G3 (A)	G3 (B)	G4 (A)	G4 (B)	G5	G6
Du 02/01/2023 Au 06/02/2023	G6 (A)	G6 (B)	G7	G1	G2 (A)	G2 (B)	G3 (A)	G3 (B)	G4	G5
Du 07/02/2023 Au 20/03/2023	G5 (A)	G5 (B)	G6	G7	G1 (A)	G1 (B)	G2 (A)	G2 (B)	G3	G4
Du 21/03/2023 Au 09/04/2023	G4 (A)	G4 (B)	G5	G6	G7 (A)	G7 (B)	G1 (A)	G1 (B)	G2	G3
Du 10/04/2023 Au 14/05/2023	G3 (A)	G3 (B)	G4	G5	G6 (A)	G6 (B)	G7 (A)	G7 (B)	G1	G2
Du 15/05/2023 Au 30/06/2023	G2 (A)	G2 (B)	G3	G4	G5 (A)	G5 (B)	G6 (A)	G6 (B)	G7	G1

Ceci est une proposition de calendrier

Valider

Figure 36 : Confirmer le calendrier.

Si l'administrateur valide le calendrier, alors l'application :

1.1 Crée les passages.

1.2 Crée les groupes.

1.3 Répartit les étudiants sur les groupes créés.

1.4 Crée les fiches de notations pour chaque étudiant.

1.5 Affiche le calendrier créé qui a été stocké sous forme de passages dans la base de données (Voir figure 29).

## 2.3 Interface noter les étudiants :

Le professeur commence d'abord par choisir l'année universitaire :



Choisir l'année universitaire

Année universitaire

22/23

Suivant

**Figure 37 : Choisir l'AU.**

Puis il choisit un parmi les groupes qu'il a supervisé, c'est-à-dire qui a passé un stage dans son service :



Liste des groupes de l'A.U 22/23

Troisième année:

CMC (Pneumologie):

G1A G7A G6A

G5A G4A G3A

G2A

CHU (Pneumologie):

G1B G7B G6B

G5B G4B G3B

G2B

Hassan II (Dermatologie):

**Figure 38 : Liste des groupes.**

Lorsqu'il clique sur le nom d'un groupe, il voit la liste des étudiants de ce groupe, si l'étudiant n'est pas encore noté on lui affiche le bouton 'Noter', sinon on lui affiche le bouton 'Modifier la note' :





L'application calcule certains champs ('Note totale du comportement', 'Note globale sur 20' et 'Note globale sur 100') avec du JavaScript et vérifie avec du PHP la validité de chaque champ :

ACCUEIL

GESTION DES NOTES

Notes >

GESTION DU COMPTE

Session >

Grille de comportement

Ponctualité assidue	5	/ 4
• Cette note doit être inférieure ou égale à 4		
Intérêt pour le malade	-1	/ 4
• Cette note doit être supérieure ou égale à 0		
Degré de motivation	9	/ 4
• Cette note doit être inférieure ou égale à 4		
Sens de responsabilité	4	/ 4
Capacité de relations	4	/ 4
<hr/>		
Note totale du comportement (calculée)	21	/ 20

Objectifs

Validation des objectifs	59	/ 40
• Cette note doit être inférieure ou égale à 40		

Examen

Examen de fin de stage	38	/ 40
------------------------	----	------

Note globale

Note globale sur 100 (calculée)	118	/ 100
Note globale sur 20 (calculée)	23.6	/ 20

Figure 41 : Vérification des champs du carnet de stage.

## Conclusion :

Dans ce chapitre nous avons présenté en détails le développement de notre application. Nous avons commencé par présenter l'environnement logiciel et le choix des outils de travail utilisés lors du développement. En deuxième lieu nous avons présenté les scénarios de test et de validation de l'application.

## Conclusion générale et perspectives

L'objectif de notre stage de fin d'étude, effectué à la Faculté de Médecine et de Pharmacie d'Agadir, et dont le présent document décrit le déroulement, a consisté à concevoir et développer une application web permettant aux administratifs la gestion des stages, et aux professeurs l'attribution des notes, et aux étudiants de consulter leurs calendriers de stage.

Ce rapport avait commencé par la spécification des besoins, qui nous avaient permis de discerner les cas d'utilisations et les utilisateurs possibles, et d'en conclure une architecture valable pour notre solution.

Après le choix des technologies nécessaires pour le développement de notre solution, qui devaient assurer la qualité, l'extensibilité et l'interopérabilité nous avons développé notre application web qui a été testée et validée par les administratifs au sein de la Faculté de Médecine et de Pharmacie d'Agadir. Par la suite, l'application est déployable sur un serveur virtualisé, pour permettre son interaction avec l'application de gestion de scolarité grâce à une API dédiée.

Notre application a été conçue et développée de façon extensible. Par conséquent, on peut y ajouter plusieurs fonctionnalités. Citons, par exemple, l'importation automatique des informations relatives aux étudiants en utilisant les données de l'application de scolarité. On pourra aussi automatiser la réinscription des étudiants n'ayant pas validé leurs stages et l'inscription dans les stages de l'année suivante pour ceux qui ont validé.

Ce projet n'est que notre premier pas dans une plus grande vision, de créer des outils permettant aux différents acteurs dans les établissements de l'éducation et l'enseignement, de bénéficier de ce que la technologie permet d'offrir, facilement et avec les moindres contraintes. Ceci s'inscrit dans la vision des développeurs, de faire le point de liaison entre les technologies et les usagers, et de produire des solutions plus simples et plus rapides qui poussent à un tel futur.

# Webographie et ressources

- [1] *La méthode agile, qu'est-ce que c'est ?* [Site disponible sur : https://www.redhat.com/fr/devops/what-is-agile-methodology](https://www.redhat.com/fr/devops/what-is-agile-methodology) (Page consultée le 08/06/2022)
- [2] *Manifesto for Agile Software Development*. Site disponible sur : <https://agilemanifesto.org/iso/fr/manifesto.html> (Page consultée le 08/06/2022)
- [3] *Ergonomic criteria for the evaluation of human-computer interface*. Site disponible sur: <https://hal.inria.fr/inria-00070012/fr/> (Page consultée le 03/06/2022)
- [4] *Get started with Docker Compose*. Site disponible sur : <https://docs.docker.com/compose/gettingstarted/> (Page consultée le 12/04/2022)
- [5] : *Symfony 5.4 Documentation*. Site disponible sur : <https://symfony.com/doc/5.4/index.html> (Page consultée le 7/04/2022)
- [6] : *PHP : PHP Manual*. Site disponible sur : <https://www.php.net/manual/en/> (Page consultée le 04/04/2022)
- [7] : *Questions tagged [symfony]*. Site disponible sur : <https://stackoverflow.com/questions/tagged/symfony> (Page consultée le 8/04/2022)
- [8] : *Slack | support | Symfony Devs*. Site disponible sur : <https://symfony.com/slack> (Page consultée le 12/04/2022)
- [9] : *MariaDB Server Documentation*. Site disponible sur : <https://mariadb.com/kb/en/documentation/> (Page consultée le 18/04/2022)
- [10] : *DDEV-Local Documentation*. Site disponible sur : <https://ddev.readthedocs.io/en/stable/> (Page consultée le 13/04/2022)
- [11] : *Portainer Documentation*. Site disponible sur : <https://docs.portainer.io/> (Page consultée le 12/04/2022)
- [12] : *Doctrine Query Language*. Site disponible sur : <https://www.doctrine-project.org/projects/doctrine-orm/en/2.11/reference/dql-doctrine-query-language.html> (Page consultée le 20/04/2022)

## Annexe

```
[1] $passages = $passageRepository  
    ->findByIdAyearIdLevel($idayear, $idlelevel);  
  
[2] {% if passages is empty %}  
  
[3] {% if passages is not empty %}  
  
[4] {% set valid=true %}  
    {% if( stages|length != periodes|length ) %}  
        {% set valid=false %}  
    {% endif %}
```