**Rapport de Projet.**

Réalisé par **Rachid Oussama**

**Intitulé**

Conception et développment d'une application Web d'une communité interne ' School Community '.

# Liste des tableaux

Liste des tableaux ................................................................................................................ 2 Liste des figures ...................................................................................................................... 3

Introduction générale ......................................................................................................... 4

Chapitre 1: Présentation du projet .............................................................................. 5

1.1 Introduction ..................................................................................................................... 6

1.2 Problèmatiques ........................................................................................................... 6

1.3 Objectifs du projet ..................................................................................................... 6

1.4 Solution proposée ...................................................................................................... 6

1.5 Conclusion ....................................................................................................................... 7

Chapitre 2: Spécification du projet ............................................................................ 8

2.1 Introduction .................................................................................................................... 9

2.2 Besoins fonctionnels ............................................................................................... 6

2.3 Besoins non-fonctionnels .................................................................................... 6

2.4 Traduction des besoins en digramme de cas d'utilisation .......... 6

2.4.1 Description des acteurs ................................................................................ 6

2.4.2 Diagramme de cas d'utilisation global ............................................. 6

2.4.2 Diagramme de cas d'utilisation raffiné ............................................. 6

2.5 Conclusion ...................................................................................................................... 6

Chapitre 3: Architecture et conception ................................................................... 6

3.1 Introduction .................................................................................................................... 6

3.2 Conception générale de L'application Web ............................................ 6

3.3 Conception détaillée ............................................................................................... 6

3.3.1 Diagramme de classes ................................................................................... 6

3.3.2 Dictionnaire de données ............................................................................. 6 3.4 Conclusion ...................................................................................................................... 6

Conclusion ................................................................................................................................... 5

# Liste des figures

Figure 1 : Diagramme de cas d'utilisation globale ........................................... 12 Figure 2 : Diagramme de cas d'utilisation raffiné - cas Moderator ...... 13 Figure 3 : Diagramme de cas d'utilisation raffiné - cas Admin .............. 13

Figure 4 : Diagramme de cas d'utilisation raffiné - cas Formateur ..... 14

Figure 5 : MVC Pattern process ..................................................................................... 16

Figure 6 : Diagramme de classe ................................................................................... 18

# Introduction

De nos jours, chaque organisation cherche le meilleur moyen pour développer la communication entre ses utilisateurs. Dans ce cas, les réseaux sociaux sont devenues les environnements les plus favorables pour chaque organisation pour genre d'activité. En effet, la plupart de ces réseaux garantissent la gratuité, et plusieurs fonctionnalités. Ils leurs offrent un espace libre permettant une interraction directe entre les membres. Cela se fait à travers les publications, les commentaires, le partage des avis.

Pour cela, nous cherchons à créer une application web qui

offre les outils et les interfaces nécessaires pour pouvoir communiquer, échanger et interagir facilement entre eux..

Le présent rapport synthétise le travail que nous effectué. Il est organisé en chapitres comme suit :

* Le premier chapitre donne une présentation générale du projet .
* Le deuxième chapitre intitulé ' Spécification des besoins ' , présente les différents besoins fonctionnels et non fonctionnels auxquels doit satisfaire l'application .
* La conception de notre solution sera présentée dans le chapitre trois .
* Et nous allons finir par une conclusion et des perspectives .

# Chapitre 01

**Presentation du projet**

## 1.1 Introduction

Ce Chapitre a pour objet de situer le projet dans son contexte général, à savoir le problématique qui a inspiré la création de notre application, la description du projet et les objectifs à atteindre.

## 1.2 Problématique

Après la pandémie de Corona, toutes les institutions ont eu recours aux sites de réseaux sociaux comme moyen de communication entre leurs membres, méthode facile à adopter étant donné que la majorité des gens utilisaient ces plateformes. Mais il y avait plusieurs défauts qui n'étaient pas perceptibles, le plus marquant de ces défauts est que les plateformes ne sont pas dédiées à ce type de communication et ne seront donc pas le moyen approprié, par contre ces plateformes travaillent pour garder leur les utilisateurs le plus de temps possible à naviguer sur leurs sites, il est dangereux de placer un membre dans ce environnement.

## 1.3 Obectifs du projet

Les problèmes mentionnés précédemment peuvent être résolus par un moyen qui garantit:

* La bonne gestion des membres du commnunité.
* Facilité la communication entre toutes les membres de l'institut.
* Le partage des informations.

## 1.4 Solution proposée

La solution est d'offrir une communité interne qui offre à ses membres (Admin - Moderator - Teacher - Student ) les outils et les

**6**

interfaces nécessaires pour pouvoir communiquer, échanger et interagir facilement entre eux.

## 1.5 Conclusion

Tout au long de ce chapitre , nous avons mis notre projet dans son cadre général, à savoir la présentation du projet et ses objectifs, et nous allons spécifier dans le prochain chapitre les différents besoins auxquels doit répondre notre application .

# Chapitre 02

**Spécification des besoins**

## 1.1 Introduction

La réussite de tout projet dépend de la qualité de son départ . De ce fait , l'étape de spécification des besoins constitue la base de départ de notre travail , elle doit décrire sans ambiguïté le logiciel à développer . Pour assurer les objectifs attendus , il est essentiel que nous parvenions à une vue claire des différents besoins escomptés de notre projet . Au cours de ce chapitre, nous allons consacrer ce chapitre pouridentifier les acteurs de notre plateforme , recenser les besoins fonctionnels et les besoins non fonctionnels pour enfin terminer avec la présentation du diagramme des cas d'utilisation générale ainsi que le raffinement de chaque cas d'utilisation de notre application.

## 1.2 Besoins fonctionnels

Les besoins fonctionnels représentent les fontionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée/sortie du système, ces besoins sont classés par acteurs:

* Un ***admin*** peut publier des postes, voir autres postes, Gérer les postes(Accepter/Refuser), Gérer les emplois du temps, Gérer touts les utilisateurs.
* Un ***moderator*** peut publier des postes, voir autres postes, Voir les emplois du temps, Gérer les postes(Accepter/Refuser).
* Un ***teacher*** peut publier des postes, voir autres postes, Voir les emplois du temps, Gérer le contenu des Modules(Cours, TPs, Exams)
* Un ***student*** peut publier des postes, voir autres postes, Voir les

emplois du temps, Consulter le conles TPs, les Anciens Exams

des modules...)

## 1.3 Besoins non-fonctionnels

* L'application doit ذثtre facile à utiliser.
* Le temps de réponse de l'application doit être le plus rapide possible.
* Possibilité d'optimiser l'application.
* L'application doit sécuriser les informations de ses utilisateurs.

## 1.4 Traduction des besoins en diagrammes de cas d'utilisation

**1.4.1 Description des acteurs** Un acteur repésente une abstraction d'un rôle joué par des entités externes au système qui interagissent directement avec lui.

Dans cette application, nous identifions quatre types d'acteurs : le Admin, le Moderator, le Teacher et le Student.

***Admin :*** C'est l'acteur qui gère les utilisateurs, les posts et les emplois du temps.

L'application lui offre plusieurs fonctionnalités telles que consulter et publier et gérer les posts, Ajouter des nouveaux membres au communité, Gérer tous les membres, Ajouter les emplois du temps de tous les groups.

***Moderator :*** C'est un acteur qui est presque comme l***'Admin*** mais il n'a pas le droit de gérer les membres.

***Teacher :*** Un acteur qui peut publier des posts et et qui peut gérer ses modules par ajouter des fichiers (Cours-TP-Exams) à un certain module.

***Student :*** Il s'agit d'un acteur qui peut publier des posts, consulter les emplois du temps, et les fichiers des modules.

**1.4.2 Diagramme de cas d'utilisation global** Un acteur repésente une abstraction d'un rôle joué par des entités externes au système qui interagissent directement avec lui.

Dans cette application, nous identifions quatre types d'acteurs : le Admin, le Moderator, le Teacher et le Student.

Moderator

Admin

Student

Teacher

Gérer les posts

S>authentifier

Gérer les emplois

du temps

Publier et consulter

les postes

Commenter aux postes

Consulter les membres

Gérer les membres

Gérer les modules

Consulter les modues

Consulter les emplois

du temps

Figure 1 : Diagramme de cas d'utilisation globale

**1.4.3**

**Diagramme de cas d'utilisation raffiné**

Moderator

Gérer les posts

S'authentifier

Publier des posts

Supprimer des posts

Accepter/Refuser

des posts

Figure 2 : Diagramme de cas d'utilisation raffiné - cas de Moderator

Admin

Gérer les emplois

S'authentifier

Ajouter les emplois

mettre à jour

les emplois

Supprimer

les emplois

Figure 3 : Diagramme de cas d'utilisation raffiné - cas de Admin

Formateur

Gérer les modules

S'authentifier

Ajouter des fichiers

(

Cours - Exam - TP

)

Supprimer

mettre à jour

Figure 4 : Diagramme de cas d'utilisation raffiné - cas de Formateur

## 1.5 Conclusion

Ce chapitre nous a permis de couvrir les différents besoins fonctionnels et non fonctionnels auxquels doit satisfaire notre application. Nous avons aussi détaillé ces besoins à travers des diagrammes de cas d'utilisation pour passer par la suite à la conception de notre application qui sera présentée dans le chapitre suivant.

# Chapitre 03

**Architecture et conception**

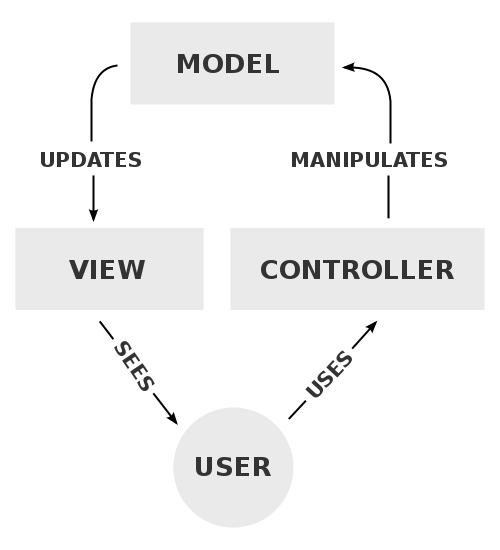
## 3.1 Introduction

Dans le présent chapitre, nous allons entamer une partie importante du développement de l'application qui constitue un point entre la spécification et la réalisation. Nous allons présenter dans un premier temps la conception générale de notre application, puis la conception détaillée comprenant les vues statiques via les diagrammes de classes.

## 3.2 Conception générale de L'application

L'architecture de l'application peut être illustrée par le graphique suivant :

Figure 5 : MVC Pattern Process



Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif 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 afficher.
* Une vue (View) contient la présentation de l'interface graphique.
* Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

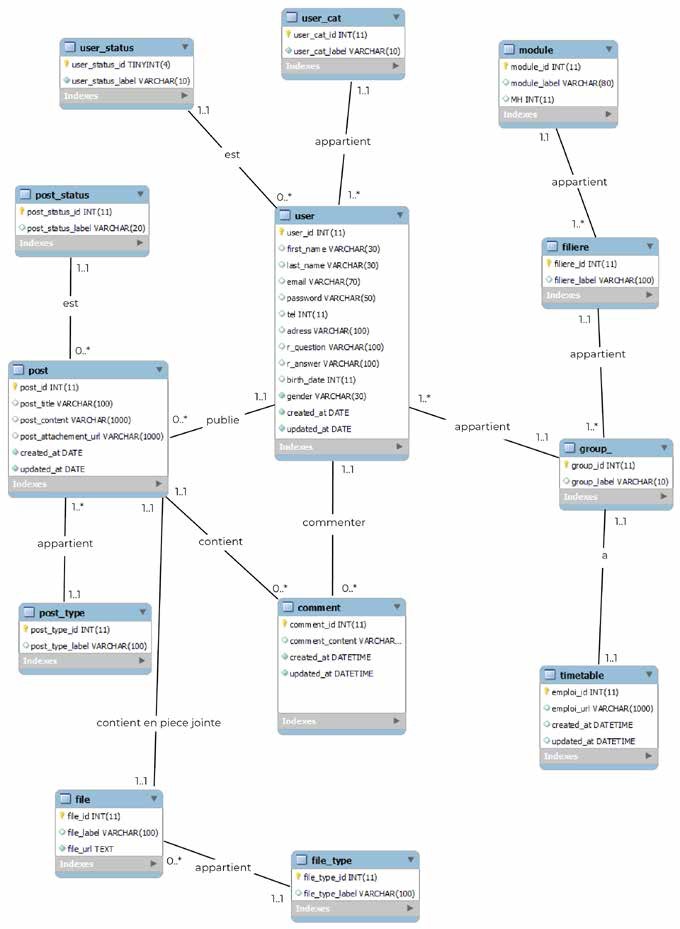
Ce motif est utilisé par de nombreux frameworks pour applications web tels que Ruby on Rails, Grails, ASP.NET MVC, Spring, Struts, Symfony, Apache Tapestry, Laravel, AdonisJS, Django ou AngularJS.

## 3.3 Conception détailée

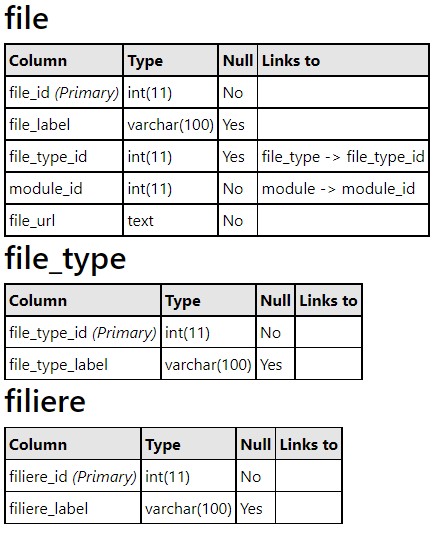
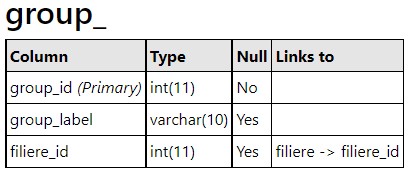
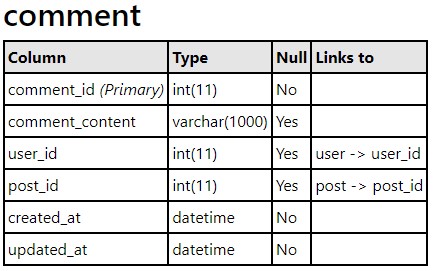
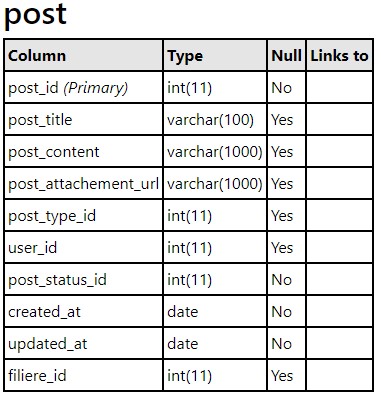
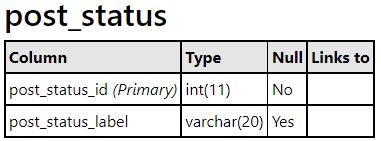
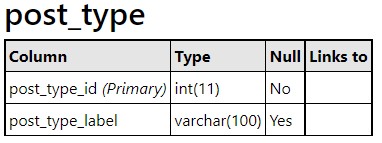
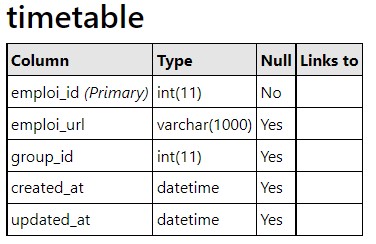
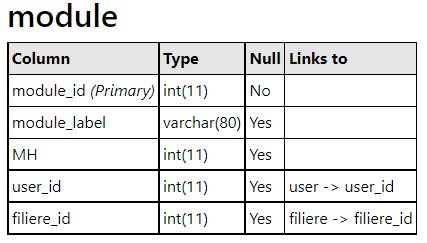
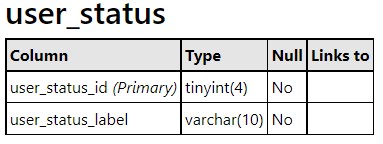
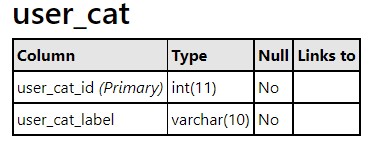
### 3.3.1 Diagram de classes

La **figure 6 représente le diagramme de classes des entités utilisées dans notre système :**

Figure 6 : Diagramme de classe



### 3.3.2 Dictionnaire des données



## 3.3 Conclusion

Dans ce chapitre, nous avons modélisé le fonctionnement de l'application afin d'avoir une vue globale et simplifiée du système . Nous avons aussi détaillé les différents modules de l'application ce qui nous a permis d'organiser le travail et d'avoir une idée claire sur le travail à réalise .

## Conclusion

Dans ce rapport , nous avons exposé les étapes de conception et de développement de notre application qui consiste à créer une application web d'une communité interne d'une institut.

Notre travail s'est déroulé sur trois étapes . Nous avons commencé par une étude de l'existant , suivie de la proposition d'une solution adéquate.

Dans la deuxième phase, nous avons spécifié les besoins fonctionnels et non fonctionnels que doit respecter l'application suivis par des diagrammes des cas d'utilisation mettant en jeu les acteurs qui interagissent avec le système.

La troisième étape concernait la conception de notre application en utilisant des diagrammes de cas d'utilisation et de classes.

Ce projet se situe en effet, dans le cadre du projet de fin de la deuxième année. Ce projet était une véritable expérience de travail en collaboration, qui nous a permis de bien gérer la répartition des tâches et de renforcer l'esprit de partage de connaissances ainsi que la synchronisation de notre travail.