

PROJET JEE

Matière : Développement des composants distribués :

WEB (JSP,Servlet,JSTL)

Intitulé :

GESTION DES AUTORISATIONS

Préparé par

Farouk Kacem

Zouari Yesmine

GLID 4

Table des matières

CHAPITRE1 : spécification des besoins	3
1. Introduction	4
2. Objectif de l'application	4
3. Spécification des besoins fonctionnels	4
4. Spécification des besoins non fonctionnels	4
5. Conclusion	5
CHAPITRE 2 : Conception	6
1. Introduction	7
2. Diagramme de cas d'utilisation	7
2.1. Diagramme de cas d'utilisation système	7
2.2. Description des cas d'utilisation	8
3. Diagramme de classes	9
4. Architecture de l'application	10
5. Conclusion	11
CHAPITRE 3 : Réalisation	12
1. Introduction	13
2. Environnement du travail	13
2.1. Environnement logiciel	13
3. Exposition du travail réalisée	14
4. Conclusion	20

Table des figures

FIGURE 1: DIAGRAMME DE CAS D'UTILISATION SYSTÈME	8
FIGURE 2: DIAGRAMME DE CLASSES	10
FIGURE 3: ARCHITECTURE MVC	10
FIGURE 4: PROTOTYPE DU PAGE LOGIN	15
FIGURE 5: INTERFACE DU PAGE LOGIN	16
FIGURE 6: PROTOTYPE DU PAGE LISTE DES ENSEIGNANTS	16
FIGURE 7: INTERFACE DU PAGE LISTE DES ENSEIGNANTS	17
FIGURE 8: INTERFACE PAGE AJOUTER/MODIFIER ENSEIGNANT	17
FIGURE 9: INTERFACE DU PAGE AJOUTER/MODIFIER ENSEIGNANT	18
FIGURE 10: INTERFACE DU PAGE LISTE DES AUTORISATIONS	18
FIGURE 11: PROTOTYPE DU PAGE AUTORISATION	19
FIGURE 12: AUTORISATION EN PDF	19

CHAPITRE1 : spécification des besoins

1. Introduction

La phase d'analyse et de spécification est la première étape du processus qui permet de dégager l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système en termes de métier. Nous allons définir l'objectif de l'application puis les besoins fonctionnels et non fonctionnels.

2. Objectif de l'application

L'objectif du projet, c'est de développer une application de gestion des autorisations des enseignants.

3. Spécification des besoins fonctionnels

Notre application doit répondre aux exigences suivantes :

- L'utilisateur doit avoir un identifiant et un mot de passe à saisir pour se connecter.
- L'utilisateur peut visualiser la liste des enseignants.
- L'utilisateur peut ajouter un enseignant.
- L'utilisateur peut modifier un enseignant.
- L'utilisateur peut supprimer un enseignant.
- L'utilisateur peut consulter la liste des autorisations de chaque enseignant.
- L'utilisateur peut ajouter une autorisation.
- L'utilisateur peut modifier une autorisation.
- L'utilisateur peut imprimer en état PDF une autorisation pour un enseignant choisi.

4. Spécification des besoins non fonctionnels

En plus des besoins fonctionnels, notre système doit répondre aux critères suivants :

- **La performance** : L'application à développer doit être principalement performante c'est-à-dire, à travers ses fonctionnalités, elle doit répondre à toutes les exigences des usagers d'une manière optimale.
- **La convivialité** : L'application doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales (simples et ergonomiques) et adaptées aux besoins de l'utilisateur.

-
- **La sécurité :** L'application devra être hautement sécurisée, les informations ne devront pas être accessibles à tout le monde. L'utilisateur accède à travers un login et un mot de passe attribués.
 - **L'extensibilité :** Dans le cadre de ce travail, l'application devra être extensible, qui permet de lui rajouter de nouvelles fonctionnalités ou d'en modifier des fonctionnalités déjà développées.

5. Conclusion

Ce chapitre présente une phase indispensable pour l'étude et l'analyse de notre projet. Nous avons défini les différents besoins fonctionnels et non fonctionnels. Nous entamerons dans le chapitre suivant la conception de cette application qui comporte les diagrammes de cas d'utilisation de classes.

CHAPITRE 2 : Conception

1. Introduction

Après avoir fixé les besoins et les objectifs dans le chapitre précédent, nous nous focalisons sur l'aspect architectural de l'application. Cette phase a pour le but de concevoir les schémas généraux qui permettent la modélisation et la description d'une manière non ambiguë du fonctionnement désiré de l'application. Dans ce chapitre une vue détaillée en se basant sur le langage UML (Unified Modeling Language).

2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit le comportement du système du point de vue utilisateur sous la forme d'action et de réaction. Un cas d'utilisation indique une fonctionnalité du système déclenchée par un acteur externe au système. Dans le diagramme interviennent trois éléments : les acteurs, le système et les cas d'utilisation. L'acteur représente un rôle joué par une personne ou un autre système.

2.1. Diagramme de cas d'utilisation système

Le diagramme de la figure suivante représente les cas d'utilisation décrivant le système d'une façon générale .

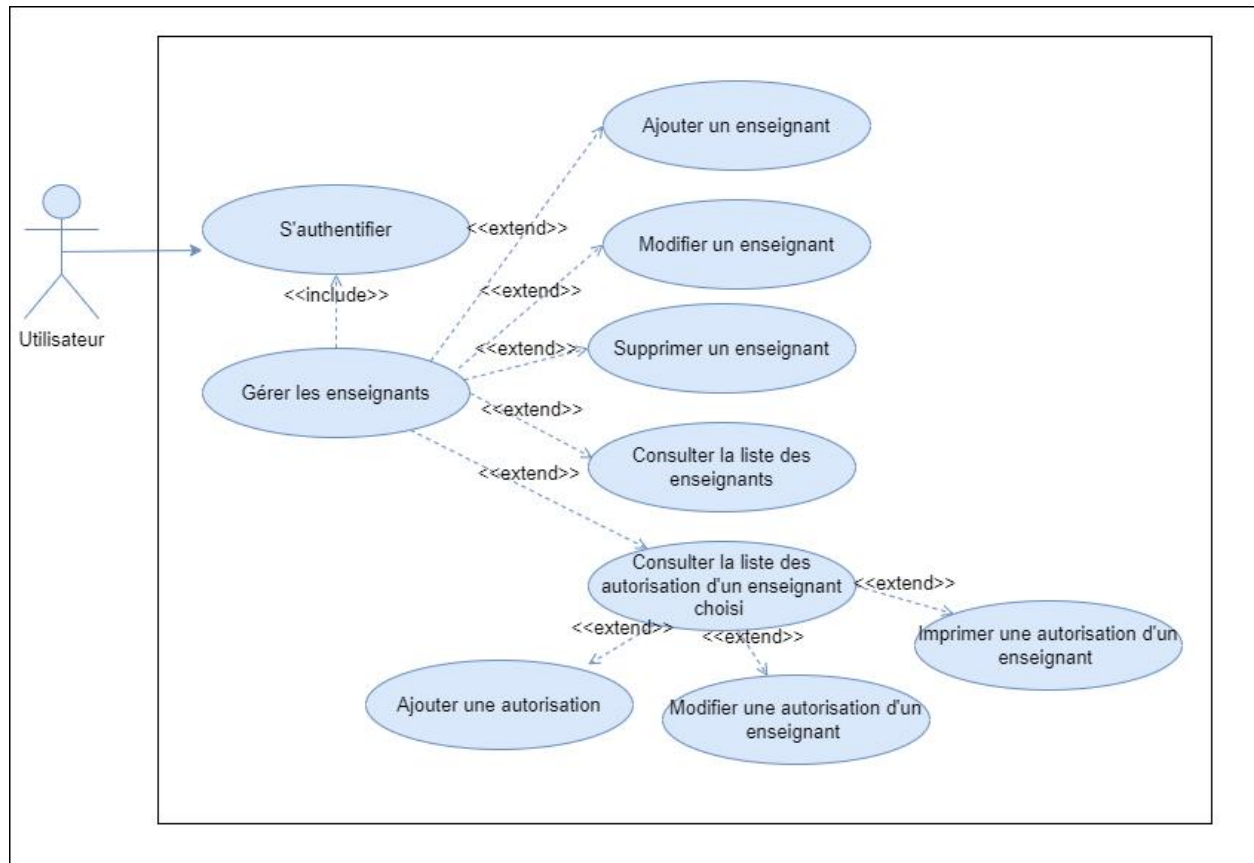


Figure 1: Diagramme de cas d'utilisation système

2.2. Description des cas d'utilisation

Dans le but de mieux comprendre le système et les interactions avec les utilisateurs, nous allons détailler dans cette partie les scénarios des quelques cas d'utilisation :

- *Description textuelle des cas d'utilisations « S'authentifier »*

Cas d'utilisation	S'authentifier
Acteurs	Utilisateur
Précondition	L'acteur n'est pas encore identifié
Postcondition	L'acteur est connecté
Scenario nominal	<ol style="list-style-type: none"> 1. L'acteur saisit son login et son mot de passe 2. L'acteur appui sur le bouton « Log in » 3. Le système affiche l'interface principale de l'application
Scenario alternatif	Si Le login ou le mot de passe de l'acteur est incorrect, le system affiche le message d'erreur « Login incorrect » ou « mot de passe incorrecte ».

- *Description textuelle des cas d'utilisations « Consulter la liste des enseignants »*

Cas d'utilisation	Consulter la liste des enseignants
Acteurs	Utilisateur
Précondition	L'acteur peut être identifié
Postcondition	L'acteur est connecté
Scenario nominal	<ol style="list-style-type: none"> 1. Accéder à la page « Accueil » 2. Le système affiche la liste des enseignants
Scenario alternatif	Le système reste sur la liste principale

- *Description textuelle des cas d'utilisations « Supprimer un enseignant »*

Cas d'utilisation	Supprimer un enseignant
Acteurs	Utilisateur
Précondition	L'acteur n'est pas encore identifié
Postcondition	L'acteur est connecté
Scenario nominal	<ol style="list-style-type: none"> 1. Accéder à la page « Accueil » 2. Choisir l'enseignant à supprimer 3. Cliquer sur le bouton de suppression 4. Le système va afficher un prompt de confirmation 5. Cliquer sur « Ok » 6. Le système supprime l'enseignant
Scenario alternatif	Le système reste sur la page d'accueil

3. Diagramme de classes

Le diagramme de classes est l'un des diagrammes de la vue statique du système. Il est modélisé sous formes de classes (entités du système d'information) et de relation (liens) entre ses classes.

La figure suivante présente le diagramme de classes :

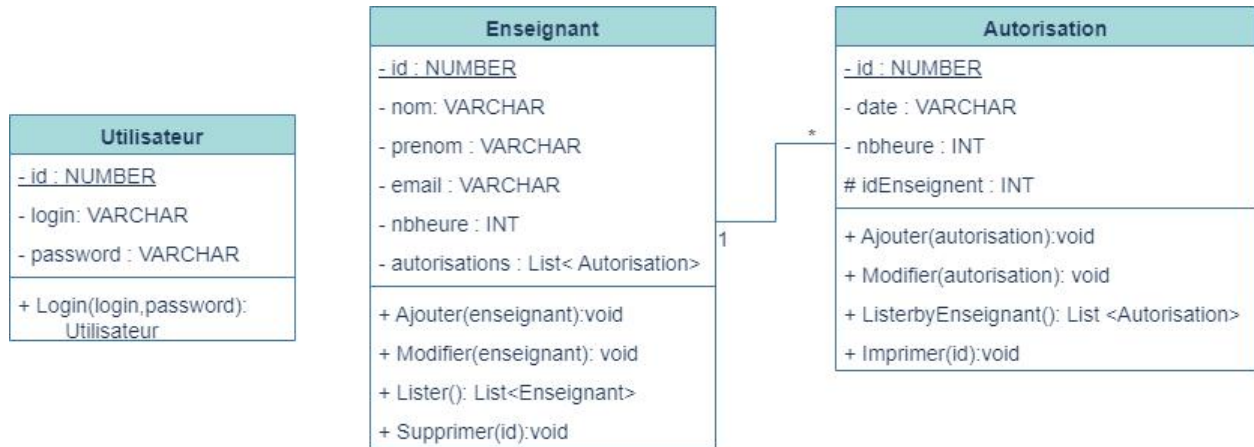


Figure 2: Diagramme de classes

4. Architecture de l'application

L'application est développée avec l'architecture MVC (Modèle, Vue, Contrôleur) représenté dans la figure ci- dessous :

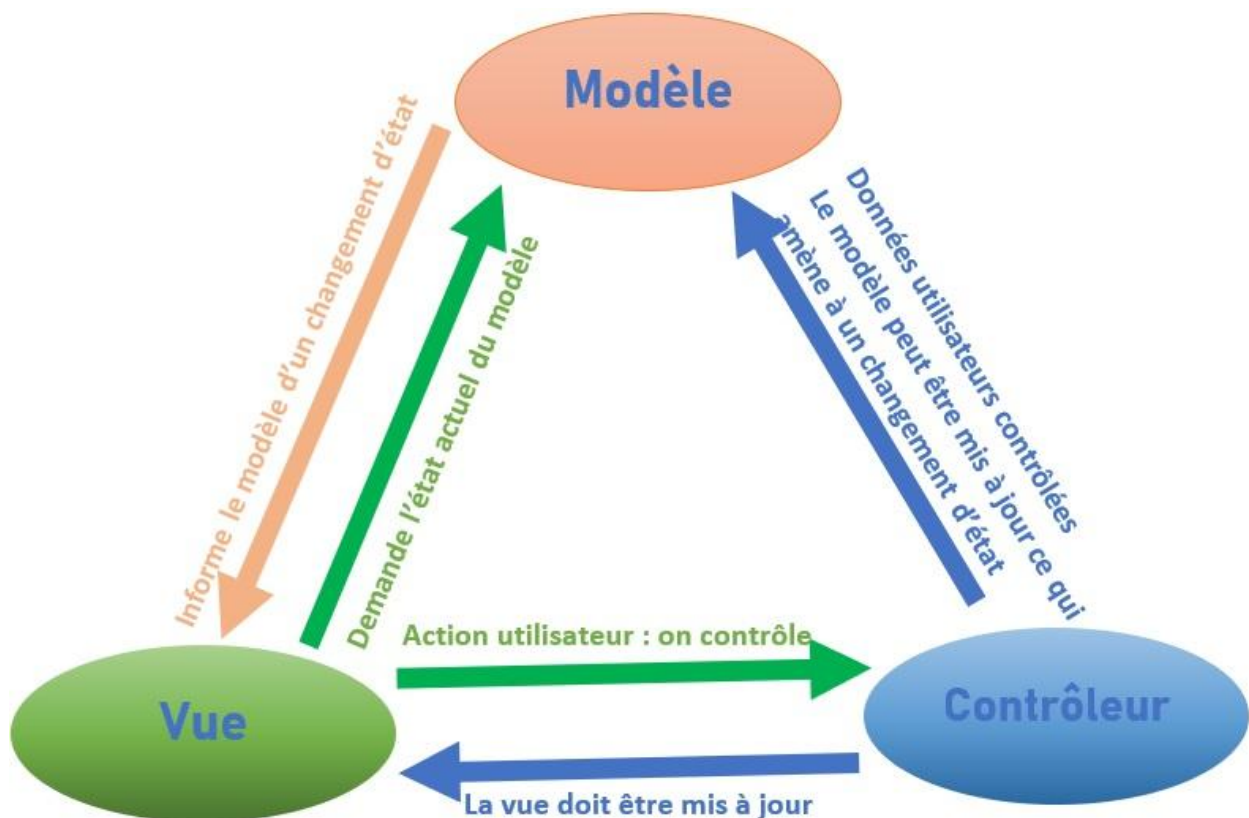


Figure 3: Architecture MVC

5. Conclusion

À travers ce chapitre, nous avons présente notre conception proposée pour l'application. Nous avons fourni, dans le premier lieu, une conception globale. Par la suite, nous avons Detaillé la conception à travers les diagrammes UML qui nous permirent de dégager la vue statique ainsi que la vue dynamique de l'application ainsi que la vue dynamique de l'application. Afin d'illustrer la concrétisation de cette conception, nous présentons dans le chapitre suivant une description de l'état de la réalisation de notre projet.

CHAPITRE 3 : Réalisation

1. Introduction

Dans ce chapitre nous allons mettre l'accent sur l'environnement de développement en présentant les différents outils utilisés. Enfin, nous illustrons la réalisation de notre travail par des imprimes écran des interfaces réalisées.

2. Environnement du travail

Nous présentons dans cette section l'environnement logiciel utilisé pour le développement et la mise en place de notre application.

2.1. Environnement logiciel

- Eclipse IDE for Enterprise Java Developers



: Outils pour les développeurs créant des applications Java Enterprise et Web, y compris un IDE Java, des outils pour les services Web, JPA et Data Tools, JSF, Mylyn, Maven et Gradle, Git, etc..

- MYSQL



: MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.

- phpMyAdmin



: phpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL et MariaDB, réalisée principalement en PHP et distribuée sous licence GNU GPL.

- HTML



: Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web.

➤ CSS



: Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium.

➤ Bootstrap



: Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

➤ Hibernate



: Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa.

➤ Apache PDFBox



: est une bibliothèque open source purement Java qui peut être utilisée pour créer, rendre, imprimer, diviser, fusionner, modifier, vérifier et extraire du texte et des métadonnées de fichiers PDF.

3. Exposition du travail réalisée

Dans cette partie, nous présentons nos maquettes d'interface IHM souhaitées après la réalisation finale qu'on a réalisée :

- *Prototype du page login :*

The image shows a wireframe of a login page within a browser window. The browser's address bar displays 'https://LoginPage'. The page content is centered and includes a placeholder for a user profile picture, the text 'Sign In', a 'User Name:' label with a text input field containing 'johndoe', a 'Password:' label with a password input field showing eight asterisks, a blue 'SIGN IN' button, and a blue underlined link for 'Forgot Password?'.

Page 1

https://LoginPage

Sign In

User Name:

johndoe

Password:

SIGN IN [Forgot Password?](#)

Figure 4: Prototype du page Login

- *Interface du login :*

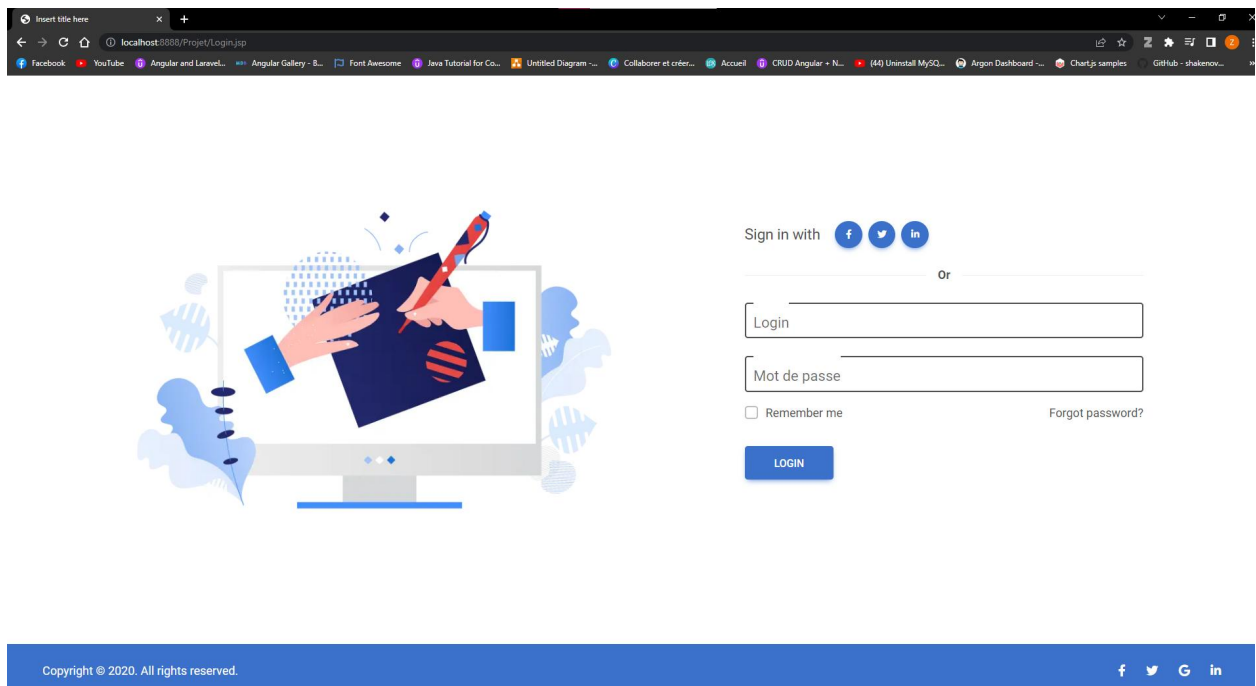


Figure 5: Interface du page login

- *Prototype du page Liste des enseignants*

Page 2				
ListPage				
#	nom	prenom	email	Action
1	Nom1	Prenom1	Email 1	delete update
2	Nom 2	Prenom2	Email 2	delete update
3				
4				

Figure 6: Prototype du page Liste des enseignants

- Interface du page Liste des enseignants :

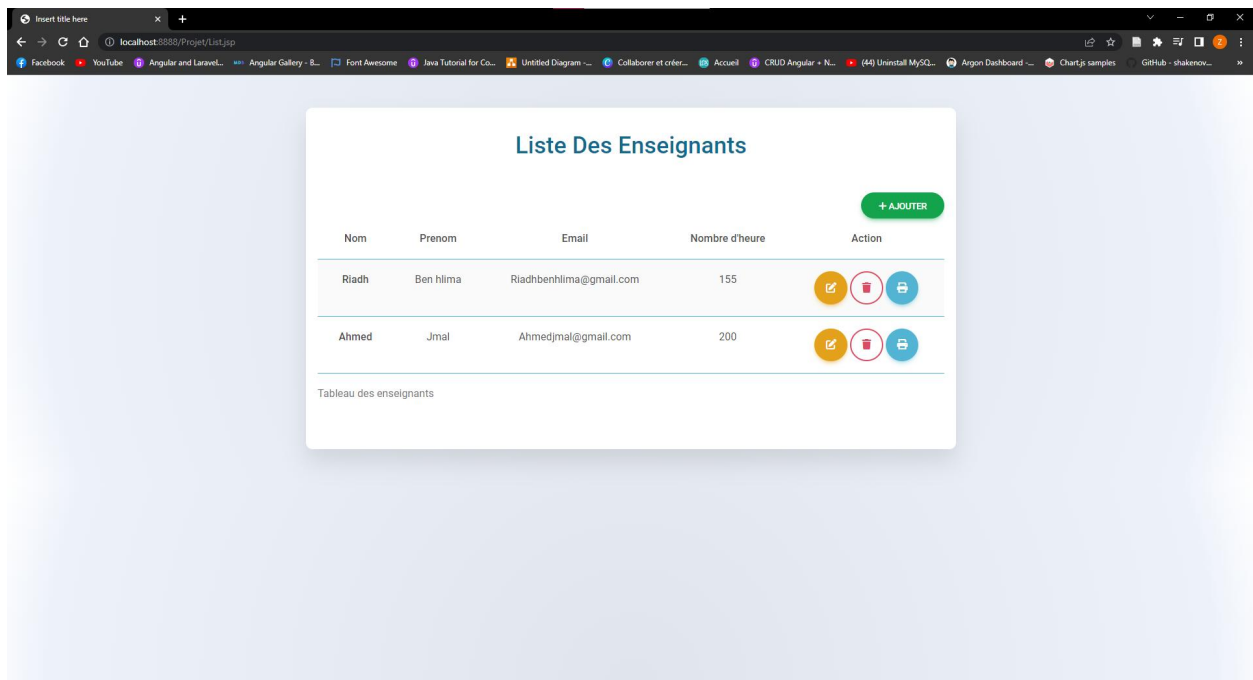


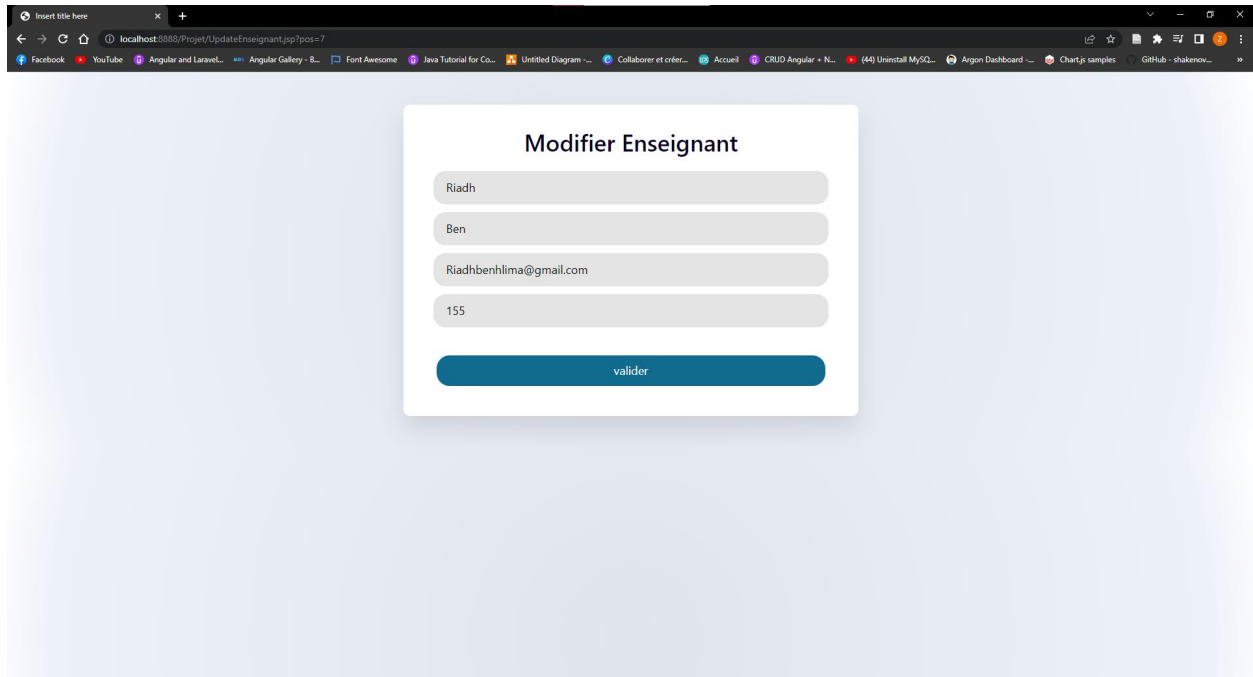
Figure 7: Interface du page Liste des enseignants

- Prototype du page Ajouter/Modifier Enseignant :



Figure 8: Interface Page Ajouter/Modifier Enseignant

- *Interface du Page Ajouter/Modifier Enseignant*



Modifier Enseignant

Riadh

Ben

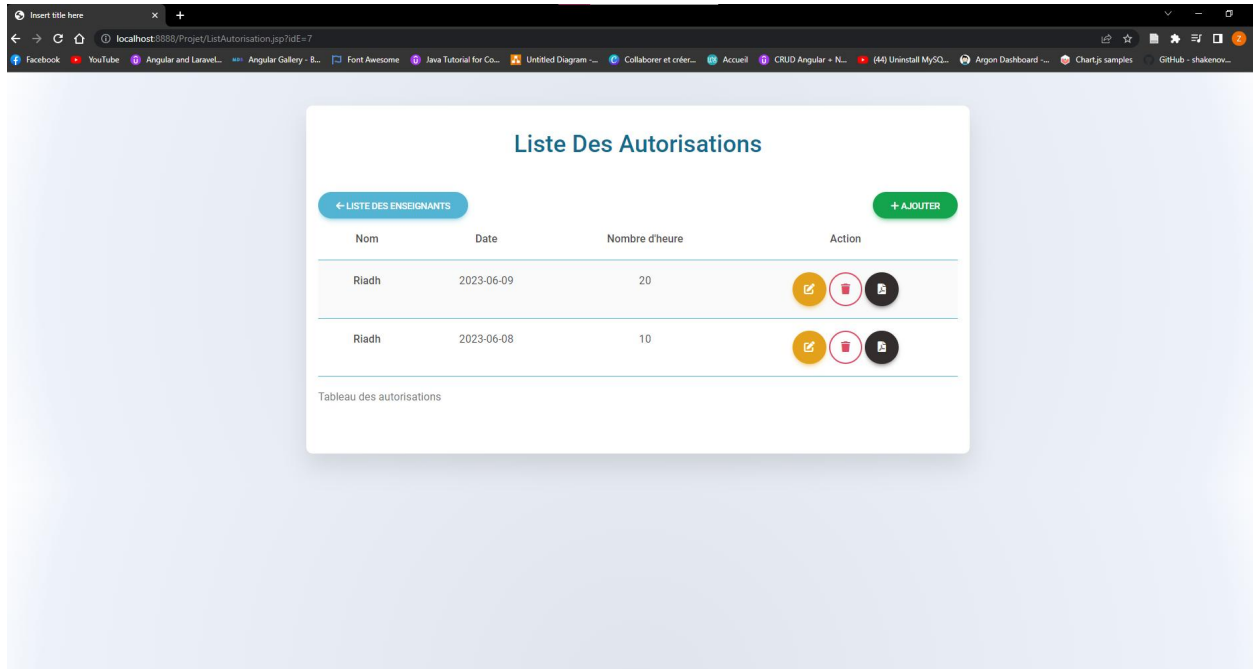
Riadhbenhlila@gmail.com

155

valider

Figure 9: Interface du page Ajouter/Modifier Enseignant

- *Interface du Page Liste des autorisations*



Liste Des Autorisations

← LISTE DES ENSEIGNANTS + AJOUTER







Nom	Date	Nombre d'heure	Action
Riadh	2023-06-09	20	  
Riadh	2023-06-08	10	  

Tableau des autorisations

Figure 10: Interface du page Liste des autorisations

- *Prototype du page d'autorisation*

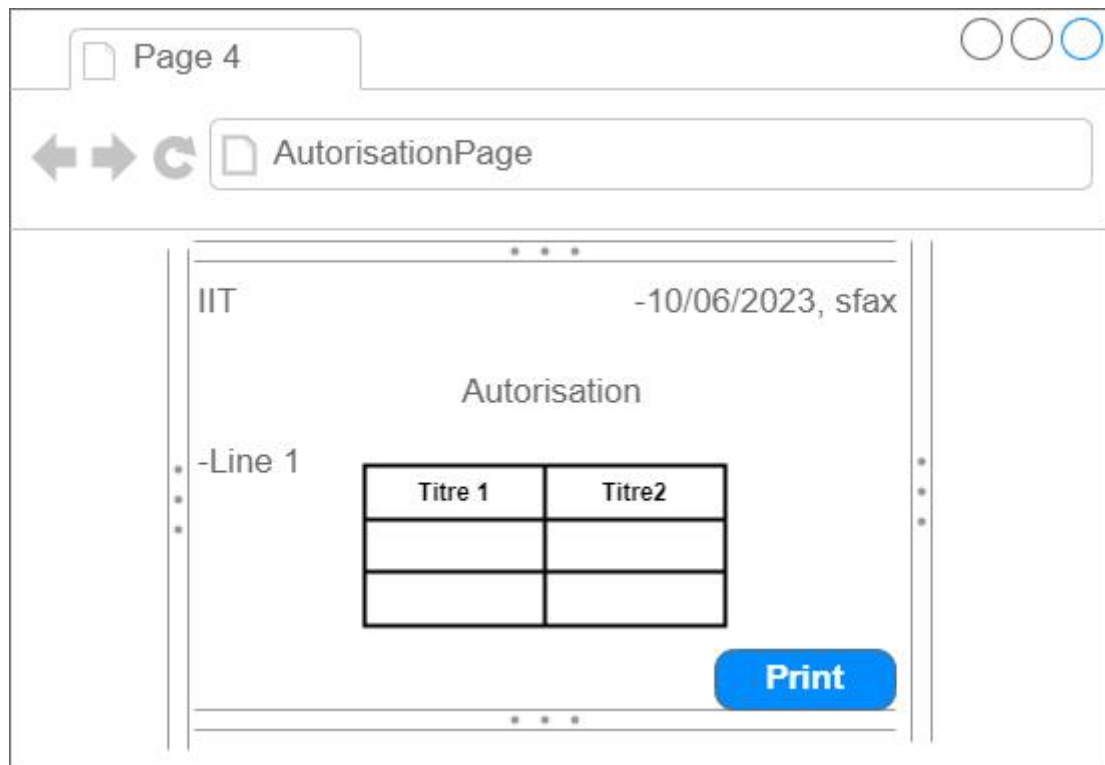


Figure 11: Prototype du page autorisation

- *Interface du Page Autorisation*

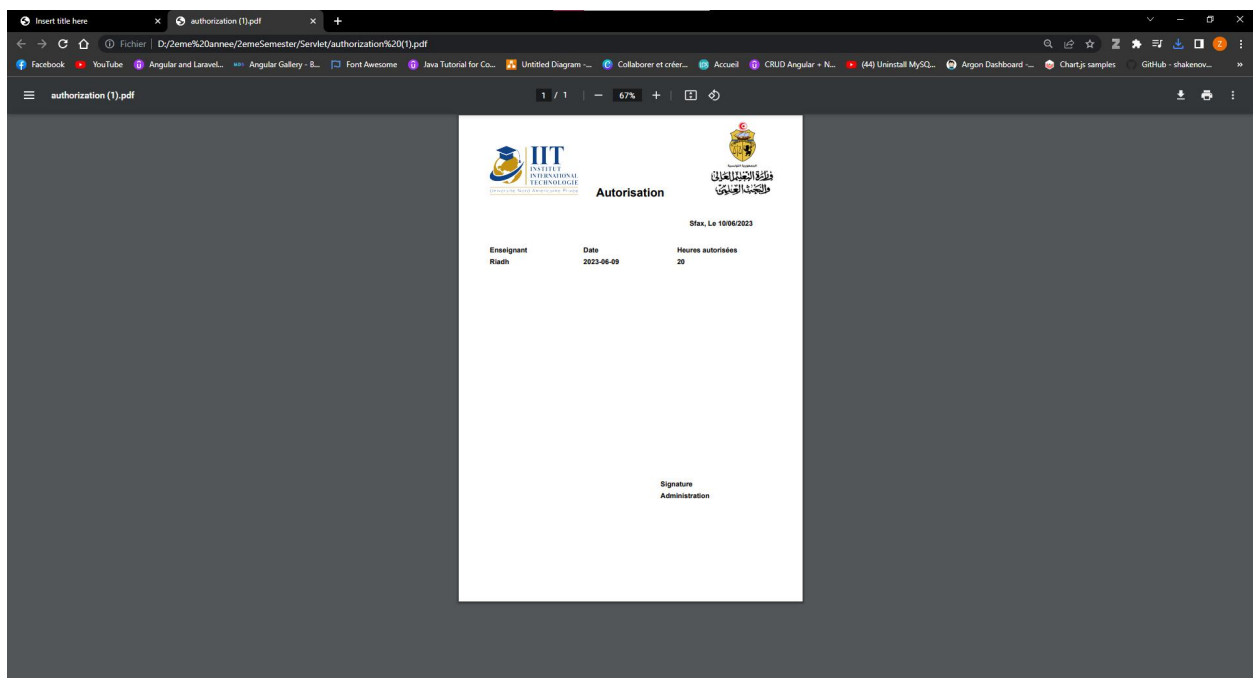


Figure 12: Autorisation en PDF

4. Conclusion

Dans ce chapitre, nous avons pu présenter l'environnement et à définir l'aspect pratique de notre application. Nous avons exposé ainsi le résultat du développement effectué à l'aide de quelques écrans.