



RAPPORT DE STAGE

Filière : Ingénierie Logicielle

Dématérialisation du système de management de la qualité (ISO 17025)

Présenté par :

EL ABODUI Mohamed Reda

Soutenue le 01/12/2025 à l'Heure 14h40

Devant le jury :

Pr. Mohamed Saad AZIZI

Président

Pr. Basma SAAD

Examineur

Pr. Sara EL-ATEIF

Examineur

Pr.KARBIL Loubna

Encadrant académique

Pr.BOUDHANE Alaaeddine

Encadrant professionnel

Année Universitaire 2024-2025

Dédicaces

Je dédie ce travail...

*À mes parents, pour leur soutien inconditionnel,
À mes enseignants, pour leurs précieux conseils,
À tous ceux qui m'ont accompagné durant ce parcours.*

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de fin d'études.

Mes remerciements s'adressent en premier lieu à mon encadrant [Nom de l'encadrant], pour ses conseils avisés, sa disponibilité et son accompagnement tout au long de ce travail.

Je remercie également l'ensemble du corps professoral de l'École Nationale Supérieure d'Intelligence Artificielle et Sciences des Données pour la qualité de la formation dispensée durant mon cursus.

Mes sincères remerciements vont également à l'équipe du laboratoire qui m'a accueilli et m'a permis de mener à bien ce projet dans les meilleures conditions.

Enfin, je remercie ma famille et mes amis pour leur soutien moral et leurs encouragements constants.

Résumé

Ce projet de fin d'études porte sur la conception et la mise en œuvre d'un système complet de gestion de la qualité et des formations au sein d'un laboratoire. L'objectif principal est de digitaliser et d'optimiser les processus de gestion des employés, de planification des formations, de suivi des présences et d'évaluation des sessions de formation.

Le système développé se compose de trois composants principaux : une application web destinée aux responsables hiérarchiques pour la gestion quotidienne, une application de bureau pour la validation des formations par les directeurs, et une base de données relationnelle MySQL assurant la cohérence et l'intégrité des données.

L'architecture adoptée repose sur des technologies modernes : React pour le frontend, Spring Framework pour le backend, et Tkinter pour l'application de validation. Cette solution modulaire garantit la maintenabilité, la sécurité et l'évolutivité du système.

Les résultats obtenus démontrent l'efficacité de la solution dans l'automatisation des processus, le suivi rigoureux des formations et l'amélioration de la qualité des données. Le système offre également des outils d'évaluation et de génération de statistiques permettant une amélioration continue.

Mots-clés : Gestion de la qualité, Formations, React, Spring Framework, MySQL, Système d'information

Table des matières

Introduction générale	1
1 Étude des besoins et Conception	2
1.1 Introduction	2
1.2 Besoins fonctionnels	2
1.3 Besoins techniques	3
1.4 Acteurs et descriptions	3
1.5 Diagramme de cas d'utilisation	3
1.5.1 Cas d'utilisation du Responsable hiérarchique	4
1.5.2 Cas d'utilisation du Directeur	4
1.5.3 Cas d'utilisation de l'Employé	5
1.6 Diagramme de classes	5
1.7 Diagrammes de séquence	6
1.7.1 Diagramme de séquence : Authentification	6
1.7.2 Diagramme de séquence : Ajout d'une formation	7
1.7.3 Diagramme de séquence : Réponse à un formulaire d'évaluation avec CIN et présence	8
1.8 Conclusion	8
2 Mise en œuvre de la solution	9
2.1 Introduction	9
2.2 Architecture globale du système	9
2.2.1 Frontend	9
2.2.2 Backend	9
2.2.3 Application dédiée au Directeur général	10
2.2.4 Base de données	10
2.2.5 Avantages de l'architecture	10
2.3 Choix des technologies adoptées	11
2.3.1 IDE et logiciels	11
2.3.2 Langages et Frameworks	12
2.3.3 SGBDR et serveurs	14
2.3.4 Autres outils	14

2.4	Implémentation de l'application Web	15
2.4.1	Connexion à l'application	15
2.4.2	Page d'accueil	15
2.4.3	Page de gestion des employés	16
2.4.4	Page de gestion des formations	18
2.4.5	Application de confirmation des formations	22
2.5	Conclusion	23
3	Déploiement de l'application	24
3.1	Architecture de déploiement	24
3.2	Déploiement de la base de données	24
3.3	Déploiement du backend Spring Boot	25
3.4	Déploiement du frontend React	25
3.5	Mise en place de l'intégration et du déploiement continu (CI/CD)	25
3.6	Sécurité et configuration CORS	25
3.7	Résultat final	26
	Conclusion générale et perspectives	27
	Bibliographie	29
A	Annexe A : Modèle conceptuel de données	30
B	Annexe B : Diagrammes de séquence	31
C	Annexe C : Captures d'écran supplémentaires	32
D	Annexe D : Guide d'installation et de déploiement	33

Table des figures

1.1	Cas d'utilisation du Responsable hiérarchique	4
1.2	Cas d'utilisation du Directeur	4
1.3	Cas d'utilisation de l'Employé	5
1.4	Diagramme de classes du système de gestion des RH et des formations	5
1.5	Diagramme de séquence : Authentification	6
1.6	Diagramme de séquence : Ajout d'une formation avec présence automatique . .	7
1.7	Diagramme de séquence : Réponse à un formulaire d'évaluation avec CIN et présence	8
2.1	Exemple de flux des composants du système	11
2.2	Logo Visual Studio Code	11
2.3	Logo IntelliJ IDEA	12
2.4	Logo Postman	12
2.5	Logo Spring Framework	13
2.6	Logo React	13
2.7	Logo Tailwind CSS	13
2.8	Logo MySQL	14
2.9	Logo MySQL Workbench	14
2.10	Logo Jira	15
2.11	Page de connexion de l'application Web	15
2.12	Page d'accueil avec statistiques	16
2.13	Page de gestion des employés	16
2.14	Formulaire d'ajout d'un nouvel employé	17
2.15	Informations personnelles de l'employé	17
2.16	Formations initiales de l'employé	17
2.17	Expériences professionnelles	17
2.18	Page de gestion des formations	18
2.19	Interface d'ajout d'une formation	18
2.20	Détails de la formation	19
2.21	Lien de formation pour l'évaluation	19
2.22	Vérification de l'accès via CNI	20
2.23	QCM d'évaluation	20

2.24	Confirmation de l'enregistrement	20
2.25	Statistiques de la formation	21
2.26	Fiche de présence	21
2.27	Liste récapitulative des présences	22
2.28	Application de confirmation avec Tkinter	23

Liste des tableaux

1.1	Besoins fonctionnels avec priorité	2
1.2	Besoins techniques avec priorité	3
1.3	Acteurs, description et priorité	3
2.1	Actions disponibles selon le statut	19

Liste des abréviations

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CNI	Carte Nationale d'Identité
CSS	Cascading Style Sheets
ENSIASD	École Nationale Supérieure d'Intelligence Artificielle et Sciences des Données
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JWT	JSON Web Token
MySQL	My Structured Query Language
OAuth2	Open Authorization 2.0
PFE	Projet de Fin d'Études
QCM	Questionnaire à Choix Multiples
REST	Representational State Transfer
SGBDR	Système de Gestion de Base de Données Relationnel
UML	Unified Modeling Language
3NF	Troisième Forme Normale

Introduction générale

Dans un contexte professionnel en constante évolution, la gestion de la qualité et le développement des compétences constituent des enjeux stratégiques majeurs pour toute organisation. Les formations professionnelles représentent un investissement essentiel permettant aux employés d'actualiser leurs connaissances et d'acquérir de nouvelles compétences adaptées aux besoins du marché.

Cependant, la gestion manuelle de ces processus engendre souvent des difficultés : planification complexe, suivi approximatif des présences, évaluations peu exploitables, et absence de traçabilité fiable. Ces limites nuisent à l'efficacité des formations et rendent difficile l'évaluation de leur impact réel sur les performances de l'organisation.

Face à ces constats, la digitalisation des processus de gestion des formations s'impose comme une solution incontournable. Elle permet non seulement d'optimiser l'organisation et le suivi des sessions, mais également de garantir une meilleure exploitation des données collectées pour une amélioration continue.

C'est dans ce contexte que s'inscrit notre projet de fin d'études, qui vise à concevoir et développer un système complet de gestion de la qualité et des formations. Ce système intègre la gestion des employés, la planification des formations, le suivi rigoureux des présences, l'évaluation des sessions et la génération de statistiques exploitables.

Ce rapport est structuré en deux chapitres principaux. Le premier chapitre présente l'étude des besoins fonctionnels et techniques du système, ainsi que la conception générale de la solution. Le second chapitre détaille la mise en œuvre pratique, incluant le choix des technologies, l'architecture du système et l'implémentation des différents modules.

Étude des besoins et Conception

I. Introduction

Dans ce chapitre, nous présentons l'analyse des besoins du système de gestion des formations et de suivi des absences, ainsi que les premières étapes de conception.

L'objectif est de définir clairement les fonctionnalités attendues, les exigences techniques, ainsi que les acteurs principaux et leurs interactions avec le système.

Cette analyse servira de base pour la conception et le développement de l'application.

II. Besoins fonctionnels

Les besoins fonctionnels représentent les services et fonctionnalités que le système doit offrir aux utilisateurs. Ils sont listés dans le tableau 1.1 avec leur niveau de priorité.

TABLE 1.1 – Besoins fonctionnels avec priorité

Description du besoin	Détail	Priorité
Gestion des employés	Ajouter, modifier, supprimer et consulter les informations personnelles des employés	Haute
Gestion des services	Créer des services, affecter des employés et consulter l'organisation interne	Moyenne
Gestion des formations	Créer, modifier, supprimer et planifier les formations, affecter les formateurs et les participants	Haute
Suivi de présence	Marquer les employés comme présents ou absents, générer des feuilles de présence	Haute
Évaluation des formations	Créer et gérer les questionnaires, collecter les réponses et générer des statistiques	Moyenne
Validation hiérarchique	Permettre aux directeurs de valider ou rejeter les formations	Haute
Notifications	Envoyer des invitations et rappels automatiques pour les formations	Moyenne
Consultation des statistiques	Générer des graphiques et rapports sur les formations et présences	Moyenne

III. Besoins techniques

TABLE 1.2 – Besoins techniques avec priorité

Description du besoin technique	Détail	Priorité
Sécurité	Authentification des utilisateurs, gestion des rôles et protection des données	Haute
Performance	Temps de réponse rapide pour les consultations et traitements	Haute
Fiabilité	Maintien de la cohérence et intégrité des données lors des opérations CRUD	Haute
Scalabilité	Possibilité d'ajouter des modules et de gérer un nombre croissant d'employés	Moyenne
Maintenabilité	Code modulaire et bien documenté pour faciliter les évolutions futures	Moyenne
Ergonomie	Interface utilisateur intuitive et responsive	Haute
Compatibilité	Compatible avec les navigateurs modernes et systèmes d'exploitation	Moyenne
Stockage	Base de données relationnelle MySQL pour un stockage fiable et structuré	Haute

IV. Acteurs et descriptions

Dans cette section, nous présentons les acteurs du système de gestion de la qualité, leur rôle principal ou secondaire, ainsi que la description de leurs interactions avec le système.

TABLE 1.3 – Acteurs, description et priorité

Acteur	Description	Priorité
Responsable hiérarchique	Gérer les employés, services, formations, présences et consulter les statistiques du système	Haute
Directeur	Valider ou rejeter les formations proposées et consulter les statistiques globales	Haute
Employé	Participer aux formations et répondre aux évaluations	Moyenne

V. Diagramme de cas d'utilisation

Le diagramme ci-dessous illustre les interactions entre les acteurs et le système de gestion de la qualité.

V.1 Cas d'utilisation du Responsable hiérarchique

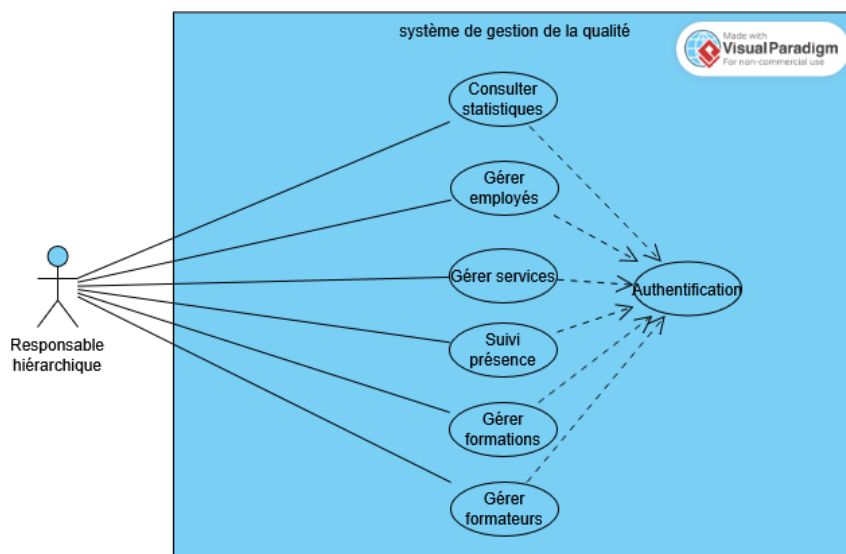


FIGURE 1.1 – Cas d'utilisation du Responsable hiérarchique

V.2 Cas d'utilisation du Directeur

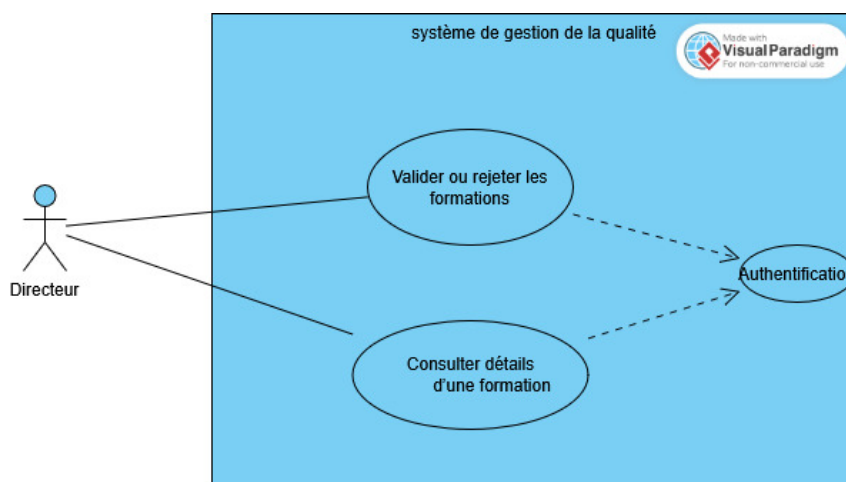


FIGURE 1.2 – Cas d'utilisation du Directeur

V.3 Cas d'utilisation de l'Employé

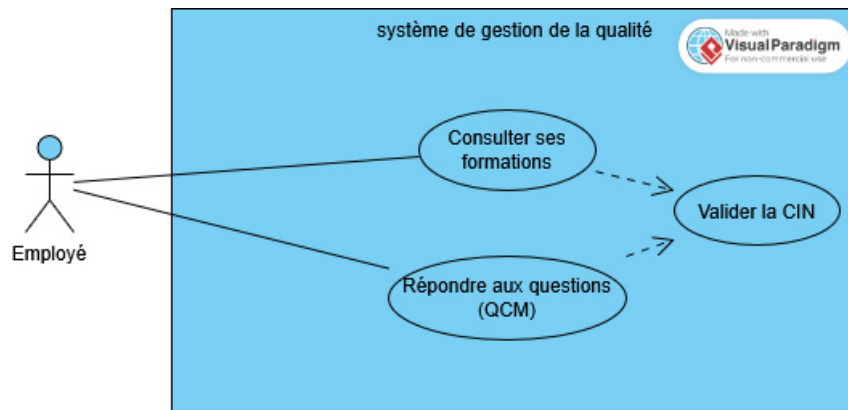


FIGURE 1.3 – Cas d'utilisation de l'Employé

VI. Diagramme de classes

Le diagramme de classes, présenté ci-dessous, illustre la structure statique du système de gestion des ressources humaines et des formations. Il met en évidence les principales entités du domaine, telles que **Employé**, **Service**, **Formation** et **Formateur**, ainsi que leurs relations. Ce modèle conceptuel sert de base à la conception de la base de données relationnelle et à la définition de la couche métier de l'application.

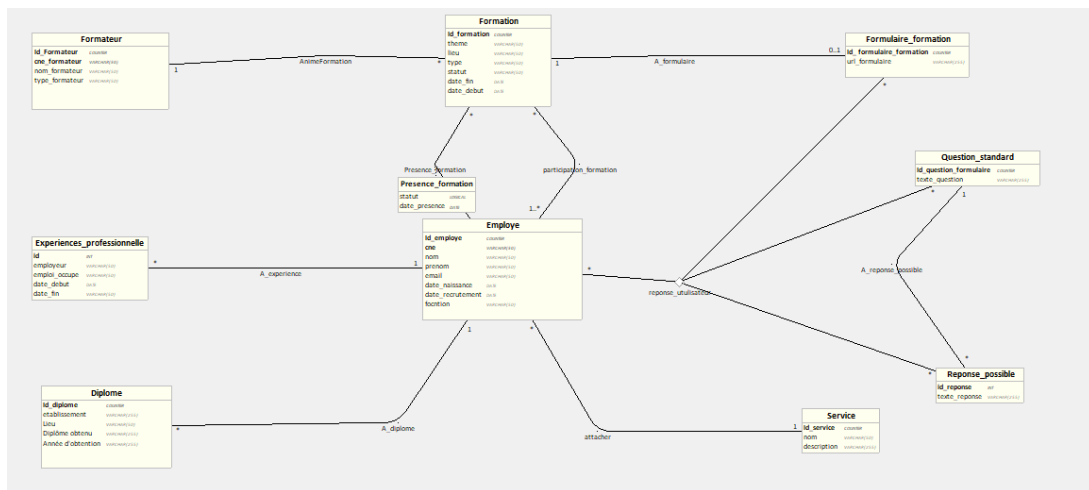


FIGURE 1.4 – Diagramme de classes du système de gestion des RH et des formations

Le diagramme met en évidence les classes principales suivantes :

- **Employé** : regroupe les informations personnelles et professionnelles du personnel de l'organisation.
- **Service** : correspond au département auquel appartient chaque employé.
- **Formation** : décrit les sessions de formation (thème, type, statut, dates, etc.).
- **Formateur** : représente les intervenants internes ou externes chargés d'animer les formations.

- **Diplôme et Expérience professionnelle** : précisent le parcours académique et professionnel des employés.

Le système intègre également des classes associatives permettant de gérer les relations complexes :

- **Présence_formation** : assure le suivi de la participation des employés aux formations.
- **AnimerFormation** : relie les formateurs aux formations qu'ils dispensent.
- **Formulaire_formation**, **Question_standard**, **Réponse_possible** et **Réponse_utilisateur** : modélisent le processus d'évaluation et de collecte des retours après formation.

Les principales relations identifiées sont les suivantes :

- Un **Service** regroupe plusieurs **Employés**.
- Un **Employé** peut suivre plusieurs **Formations**.
- Chaque **Formation** est animée par un **Formateur** et peut être associée à un **Formulaire_formation**.
- Un **Employé** peut posséder plusieurs **Diplômes** et **Expériences professionnelles**.

Ce diagramme offre une vue d'ensemble cohérente de la structure logique du système et facilite la transition vers la phase de conception physique de la base de données.

VII. Diagrammes de séquence

VII.1 Diagramme de séquence : Authentification

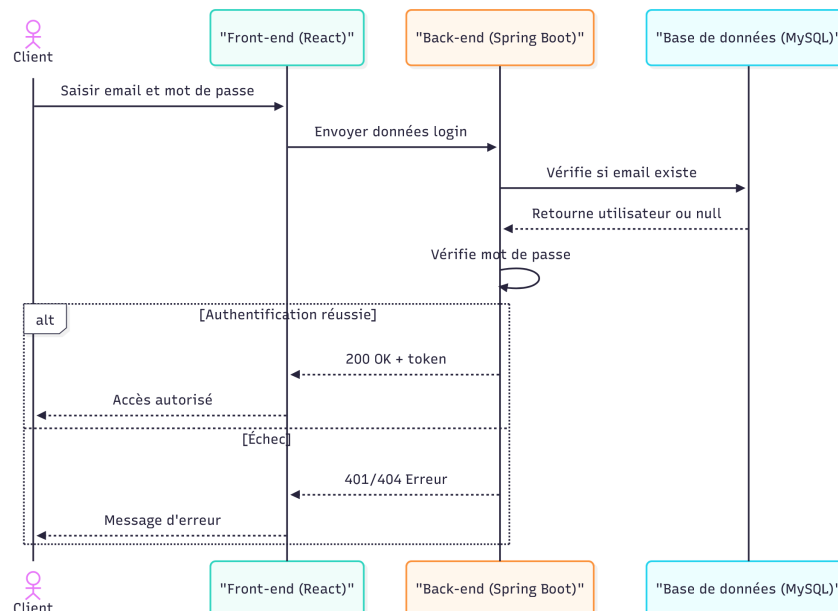


FIGURE 1.5 – Diagramme de séquence : Authentification

Ce diagramme illustre le processus d'authentification des utilisateurs dans le système. L'utilisateur saisit ses identifiants sur le front-end, qui sont transmis au back-end pour vérification. La

base de données valide les informations et le système retourne soit un accès autorisé, soit un message d'erreur en cas d'échec.

VII.2 Diagramme de séquence : Ajout d'une formation

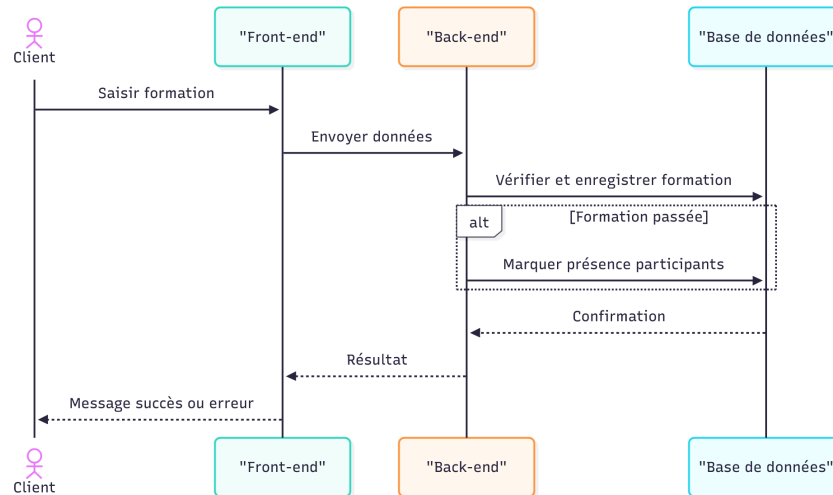


FIGURE 1.6 – Diagramme de séquence : Ajout d'une formation avec présence automatique

Ce diagramme montre le processus d'ajout d'une formation par le Responsable. Le système vérifie la validité des données avant de les enregistrer dans la base de données. Si la formation est déjà passée, la présence des participants est automatiquement mise à jour. Le Responsable reçoit une confirmation de succès ou un message d'erreur en cas d'échec.

VII.3 Diagramme de séquence : Réponse à un formulaire d'évaluation avec CIN et présence

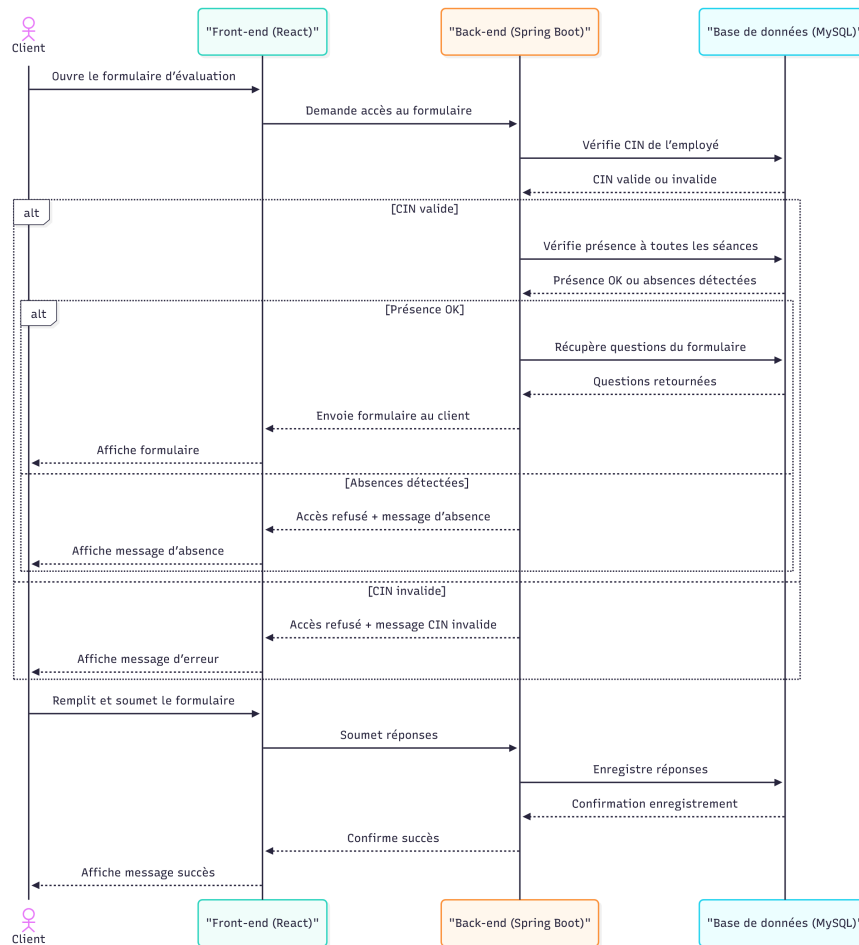


FIGURE 1.7 – Diagramme de séquence : Réponse à un formulaire d'évaluation avec CIN et présence

Ce diagramme décrit comment un employé répond à un formulaire d'évaluation. Avant l'accès, le système vérifie le CIN de l'employé et sa présence à toutes les séances de formation. Si ces conditions sont respectées, le formulaire est affiché, rempli et soumis, puis les réponses sont enregistrées dans la base de données avec confirmation de succès.

VIII. Conclusion

Cette analyse des besoins a permis de clarifier les fonctionnalités, les contraintes techniques et les acteurs du système.

Elle constitue une base solide pour la conception détaillée et le développement de l'application de gestion des formations et des absences, garantissant ainsi un système fonctionnel, sécurisé et adapté aux utilisateurs.

Mise en œuvre de la solution

I. Introduction

Dans ce chapitre, nous détaillons la mise en œuvre pratique de la solution de gestion de la qualité et des formations. Après avoir défini les besoins fonctionnels et techniques dans le chapitre précédent, il est essentiel de traduire ces exigences en un système opérationnel.

Nous présentons ici le choix des technologies, l'architecture globale du système, les composants du frontend et du backend, ainsi que la conception de la base de données. Ce chapitre décrit également les interfaces principales, les fonctionnalités de l'application web et l'application complémentaire développée pour le responsable hiérarchique.

L'objectif est de montrer comment les choix technologiques et la conception assurent la performance, la sécurité, l'ergonomie et la maintenabilité du système.

II. Architecture globale du système

Le système de gestion de la qualité est conçu selon une architecture modulaire, assurant une gestion centralisée et efficace des activités du laboratoire.

II.1 Frontend

Le frontend est développé avec **React**, utilisant **React Router** pour la navigation et **Axios** pour la communication avec le backend via des requêtes HTTP.

L'interface permet au responsable hiérarchique de :

- Gérer les employés, les formations et les services
- Planifier des événements et consulter le calendrier intégré
- Envoyer des invitations et suivre la présence des participants
- Valider ou modifier des informations durant les sessions de formation
- Collecter les retours des participants via un formulaire post-formation

Cette structure frontend modulable assure une interface réactive, intuitive et facilement maintenable.

II.2 Backend

Le backend, développé avec **Spring**, gère l'ensemble des fonctionnalités du frontend. Il prend en charge :

- La logique métier et le traitement des données liées aux employés, formations, services et événements

- L'authentification et la sécurité des utilisateurs via **JWT** et **OAuth2**
- La communication avec la base de données **MySQL**
- La documentation et le test des API avec **Swagger**

Cette architecture backend modulaire garantit sécurité, maintenabilité et évolutivité du système.

II.3 Application dédiée au Directeur général

Une application séparée, développée avec **Tkinter** en **Python**, est destinée au Directeur général. Elle permet de valider ou refuser les formations proposées.

L'application offre une interface simple et efficace pour faciliter la prise de décision tout en restant connectée à la base de données centrale.

II.4 Base de données

La base de données est conçue selon un modèle relationnel respectant les **trois formes normales (3NF)**, garantissant l'intégrité et la cohérence des données.

Les principales tables comprennent :

- **Employés** : informations personnelles, rôle et compétences
- **Formations** : détails des sessions, dates et responsables
- **Présences** : suivi de la participation des employés
- **Services** : regroupement des employés par service et fonctions
- **Événements** : calendrier des formations et autres activités

II.5 Avantages de l'architecture

Cette architecture modulaire assure :

- Une séparation claire des responsabilités
- Une meilleure maintenabilité du système
- La sécurisation des données
- Une évolutivité adaptée aux besoins futurs
- Une scalabilité verticale et horizontale

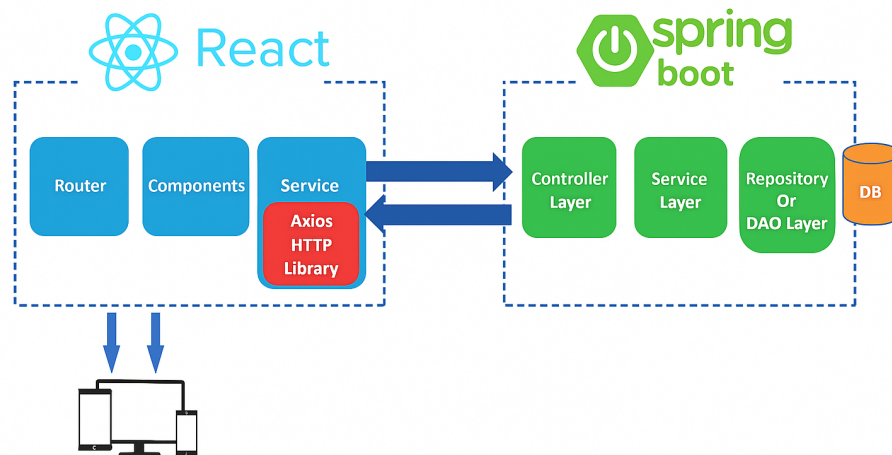


FIGURE 2.1 – Exemple de flux des composants du système

III. Choix des technologies adoptées

Pour assurer l’efficacité, la maintenabilité et la performance du système, plusieurs technologies et outils ont été adoptés.

III.1 IDE et logiciels

Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft. Ses fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la refactorisation du code et l’intégration de Git.



FIGURE 2.2 – Logo Visual Studio Code

IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré avancé pour Java, développé par JetBrains. Il permet l'autocomplétion, la refactorisation avancée et la gestion des tests unitaires.



FIGURE 2.3 – *Logo IntelliJ IDEA*

Postman

Postman permet de tester des API et de gérer les collections de requêtes HTTP de manière centralisée.



FIGURE 2.4 – *Logo Postman*

III.2 Langages et Frameworks

Spring Framework

Le **Spring Framework** est un framework Java complet pour les applications d'entreprise.



FIGURE 2.5 – *Logo Spring Framework*

React

React est une bibliothèque JavaScript pour construire des interfaces utilisateur dynamiques et interactives.

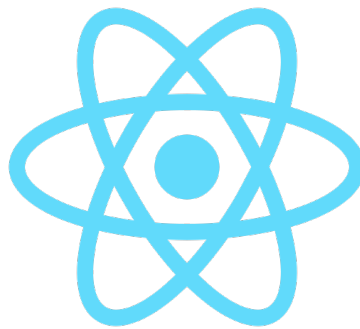


FIGURE 2.6 – *Logo React*

Tailwind CSS

Tailwind CSS est un framework CSS basé sur des classes utilitaires.



FIGURE 2.7 – *Logo Tailwind CSS*

III.3 SGBDR et serveurs

MySQL

MySQL est un système de gestion de bases de données relationnelles open source, robuste et rapide.



FIGURE 2.8 – *Logo MySQL*

MySQL Workbench

MySQL Workbench est un outil graphique pour concevoir et administrer les bases de données MySQL.



FIGURE 2.9 – *Logo MySQL Workbench*

III.4 Autres outils

Jira

Jira est un outil de gestion de projet agile pour suivre les tâches et les bugs.

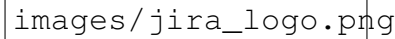


FIGURE 2.10 – *Logo Jira*

LaTeX

LaTeX est utilisé pour la rédaction du rapport, offrant qualité typographique et structuration professionnelle.

IV. Implémentation de l'application Web

L'implémentation de l'application web consiste à traduire la conception en une solution fonctionnelle accessible aux utilisateurs via un navigateur.

IV.1 Connexion à l'application

Cette page permet à l'administrateur de s'authentifier afin d'accéder aux différents services proposés par le système.

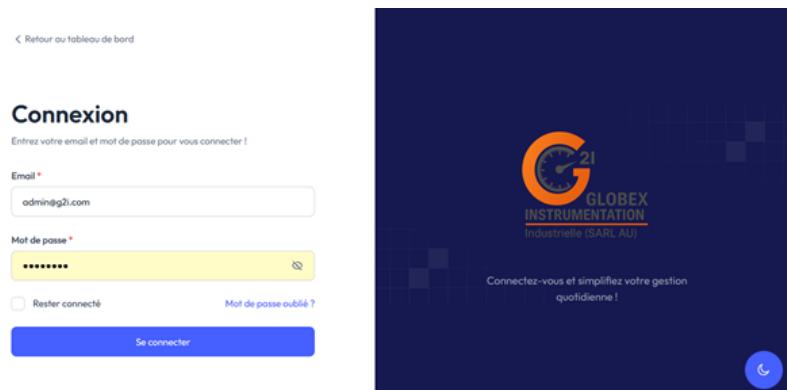


FIGURE 2.11 – *Page de connexion de l'application Web*

IV.2 Page d'accueil

Cette page contient des éléments graphiques afin de fournir différentes statistiques sur l'utilisation de l'application.

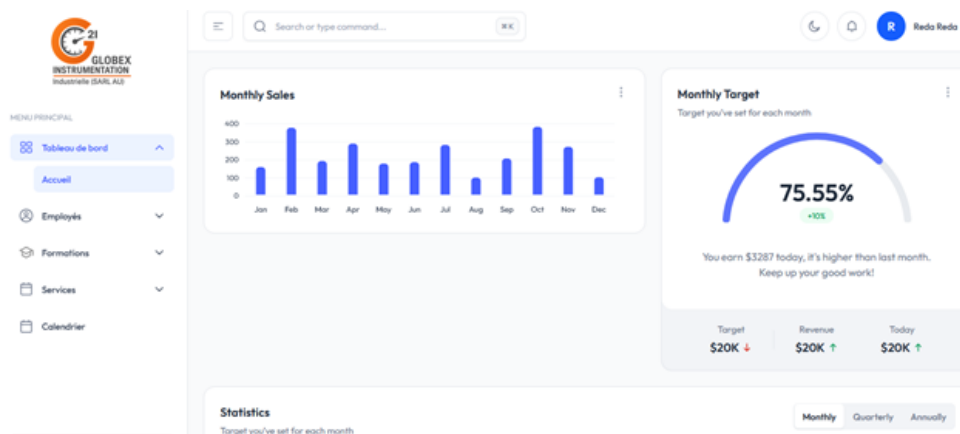


FIGURE 2.12 – Page d'accueil avec statistiques

IV.3 Page de gestion des employés

Cette page permet de consulter la liste des employés de l'entreprise, présentée sous forme de tableau.

The 'Gestionnaire des Employés' page displays a table of employees with the following columns: NOM COMPLET, CIN, EMAIL, FONCTION, SERVICE, DATE RECRUTEMENT, and ACTIONS. The table contains 5 entries.

NOM COMPLET	CIN	EMAIL	FONCTION	SERVICE	DATE RECRUTEMENT	ACTIONS
EL ABOUDI Mohamed Reda	X123	mohamedreda@gmail.com	Architect	Ressources Humaines	2025-08-08	[Red] [Checkmark]
EL ABOUDI Mohamed Reda	X12344	RED44@gmail.com	Architect	Finance	2025-08-18	[Red] [Checkmark]
Mohamed REDA	3727	bahamport7@gmail.com	Architect	Ventes	2025-08-14	[Red] [Checkmark]
EL ABOUDI Mohamed Reda	L485155	mohamedreda@laboudi@gmail.com	Architect	Ressources Humaines	2025-06-16	[Red] [Checkmark]
Alaeddine BOUDHANE	X562443	Alaeddine@boudhane.com	Manager	Finance	2017-08-08	[Red] [Checkmark]

FIGURE 2.13 – Page de gestion des employés

Ajout d'un employé

Un bouton **Ajouter Employé** permet d'ouvrir un formulaire pour saisir les informations du nouvel employé.

Informations de l'employé

CNI de l'employé x123456	Nom de l'employé Nom
Prénom de l'employé Prénom	Email info@gmail.com
Date de naissance Sélectionner une date	Date de recrutement Sélectionner une date
Fonction Développeur	Service Choisir un service

[Ajouter l'employé](#)

FIGURE 2.14 – Formulaire d'ajout d'un nouvel employé

Détails et modification d'un employé

Le bouton **Modifier** affiche une page détaillée avec les informations de l'employé.

Informations Personnelles

CNE K789456	Nom Bennani
Prénom Salma	Email salma.bennani@entreprise.ma
Date de naissance 1995-03-22	Date de recrutement 2021-02-15
Fonction Chargée RH	Service Ressources Humaines

[Modifier](#)

FIGURE 2.15 – Informations personnelles de l'employé

Formations initiales

Master en Gestion Université Hassan II 2018 Casablanca	Modifier Supprimer
Licence en gestion Université Ibn Zohr 2016 Agadir	Modifier Supprimer

[Ajouter formation initiale](#)

FIGURE 2.16 – Formations initiales de l'employé

Expériences professionnelles

RH Stagiaire Maroc Telecom 2017-02-01 - 2017-08-01	Modifier Supprimer
Stagiaire Gestion Société DEF (Service Gestion) 2022-03-01 - 2024-08-04	Modifier Supprimer

[Ajouter expérience](#)

FIGURE 2.17 – Expériences professionnelles

IV.4 Page de gestion des formations

Cette page permet de visualiser et de gérer l'ensemble des formations disponibles.

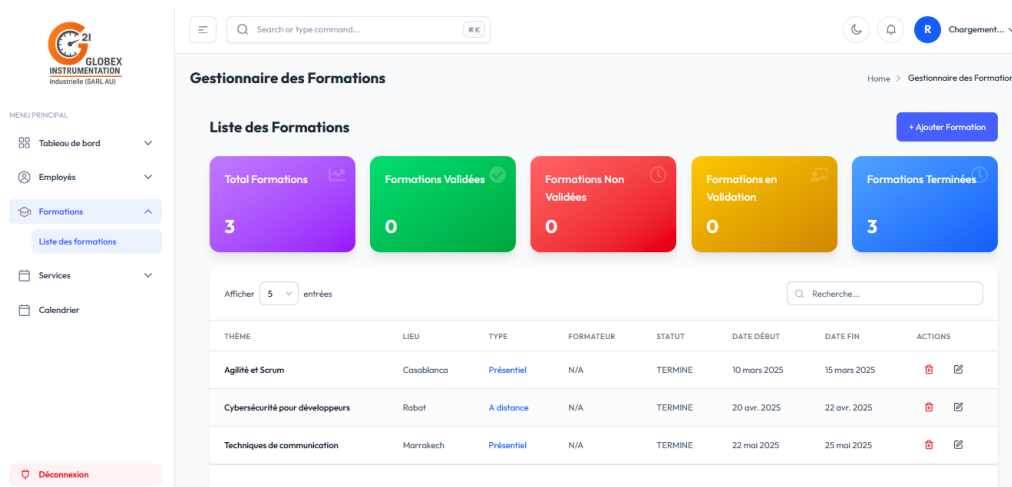


FIGURE 2.18 – Page de gestion des formations

Ajout d'une formation

Un formulaire permet d'ajouter une nouvelle formation avec toutes les informations nécessaires.

FIGURE 2.19 – Interface d'ajout d'une formation

Détails de la formation

Le bouton **Modifier** affiche un formulaire avec l’ensemble des informations détaillées.

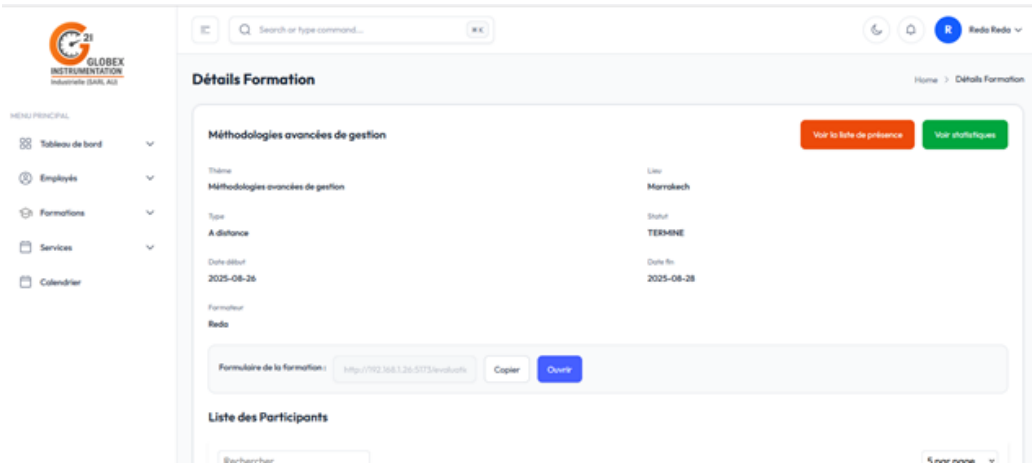


FIGURE 2.20 – Détails de la formation

TABLE 2.1 – Actions disponibles selon le statut

Statut	Actions disponibles
Terminé	Liste de présence, Statistiques
Validé	Envoyer invitations, Modifier
Non validé	Modifier
En validation	Modifier
Avant début	Liste de présence

Gestion de l’évaluation des formations

L’évaluation à chaud permet de mesurer la satisfaction des participants et d’identifier les axes d’amélioration.

Lien de formation Un lien d’évaluation est automatiquement généré une fois la formation terminée.

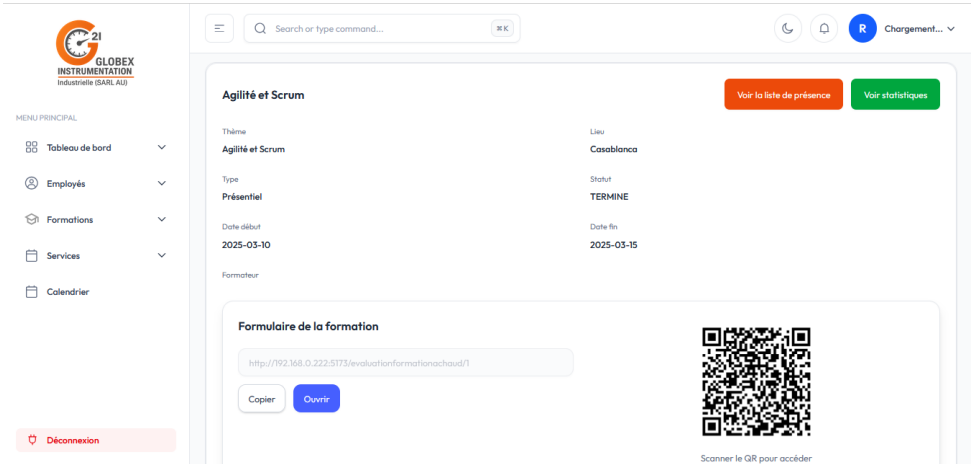
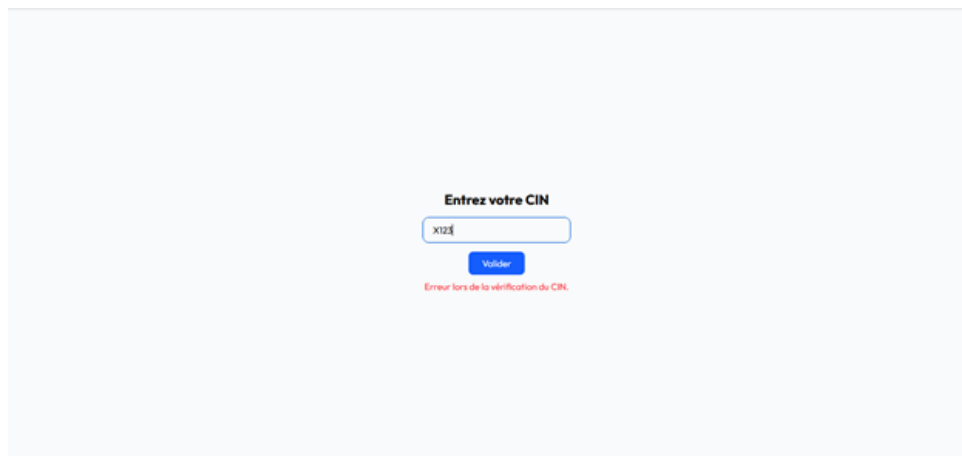


FIGURE 2.21 – Lien de formation pour l’évaluation

Vérification via CNI Le système vérifie que la CNI correspond bien à celle d'un employé inscrit.



The screenshot shows a web interface for CNI verification. At the top, it says "Entrez votre CIN". Below this is a text input field containing "X123". To the right of the input field is a blue button labeled "Valider". Below the button, there is a red error message: "Erreur lors de la vérification du CIN."

FIGURE 2.22 – Vérification de l'accès via CNI

Déroulement du QCM L'employé accède au questionnaire d'évaluation après validation des vérifications.



The screenshot shows a QCM evaluation interface. At the top, it displays "Formation : Méthodologies avancées de gestion" and "EL ABOUDI Mohamed Reda" with "CIN : X123". Below this, it says "Question 1 sur 10". The main question is "Comment évaluez-vous la qualité du contenu de la formation ?". There are four radio button options: "Excellent", "Bon", "Moyen", and "Mauvais". At the bottom, there is a blue button labeled "Suivant →".

FIGURE 2.23 – QCM d'évaluation



The screenshot shows a confirmation screen for the QCM. At the top, it displays "Formation : Méthodologies avancées de gestion" and "EL ABOUDI Mohamed Reda" with "CIN : X123". In the center, there is a green checkmark icon followed by the text "QCM Terminé". Below this, it says "Merci Mohamed Reda d'avoir complété le QCM." and "Formation suivie : Méthodologies avancées de gestion". At the bottom, it says "Nombre de réponses enregistrées : 10 / 10".

FIGURE 2.24 – Confirmation de l'enregistrement

Consultation des statistiques

Le bouton **Voir statistiques** permet d'accéder aux résultats sous forme graphique.

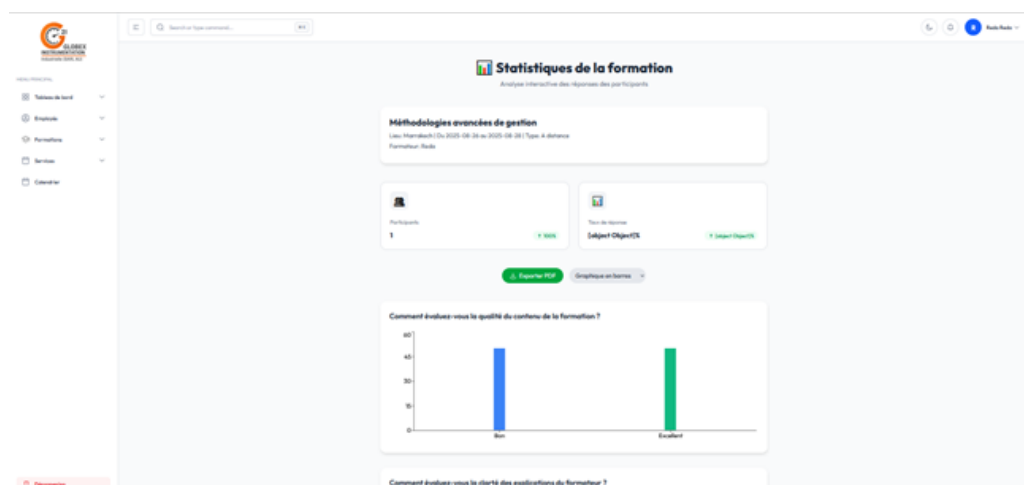


FIGURE 2.25 – Statistiques de la formation

Gestion de la présence

Le responsable peut marquer chaque employé comme présent ou absent.

Fiche de présence

- Type : Présentiel
- Titre : Agilité avancé
- Organisme / Animateur : / Reda
- Durée : 2 jour(s)

Nom	Fonction	2025-09-03	2025-09-04
EL ABOUDI Mohamed Reda	Architect	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Enregistrer

FIGURE 2.26 – Fiche de présence

The screenshot shows a web application interface for GLOBEX INSTRUMENTATION. On the left is a sidebar menu with options: Tableau de bord, Employés, Formations, Services, and Calendrier. The main content area displays a 'Fiche de présence (Lecture seule)' for EL ABOUDI Mohamed Reda. It includes details about the training type (A distance), title (Méthodologies avancées de gestion), organizer (Reda), and duration (3 jours). Below this is a table showing attendance for three consecutive dates.

Nom	Fonction	2025-08-26	2025-08-27	2025-08-28
EL ABOUDI Mohamed Reda	Architect	PRESENT	PRESENT	PRESENT

FIGURE 2.27 – Liste récapitulative des présences

IV.5 Application de confirmation des formations

Une application complémentaire a été développée avec **Tkinter** pour permettre aux directeurs de valider ou rejeter les formations proposées.

Fonctionnalités principales :

- Authentification sécurisée
- Consultation des détails d'une formation
- Gestion des participants
- Validation ou rejet

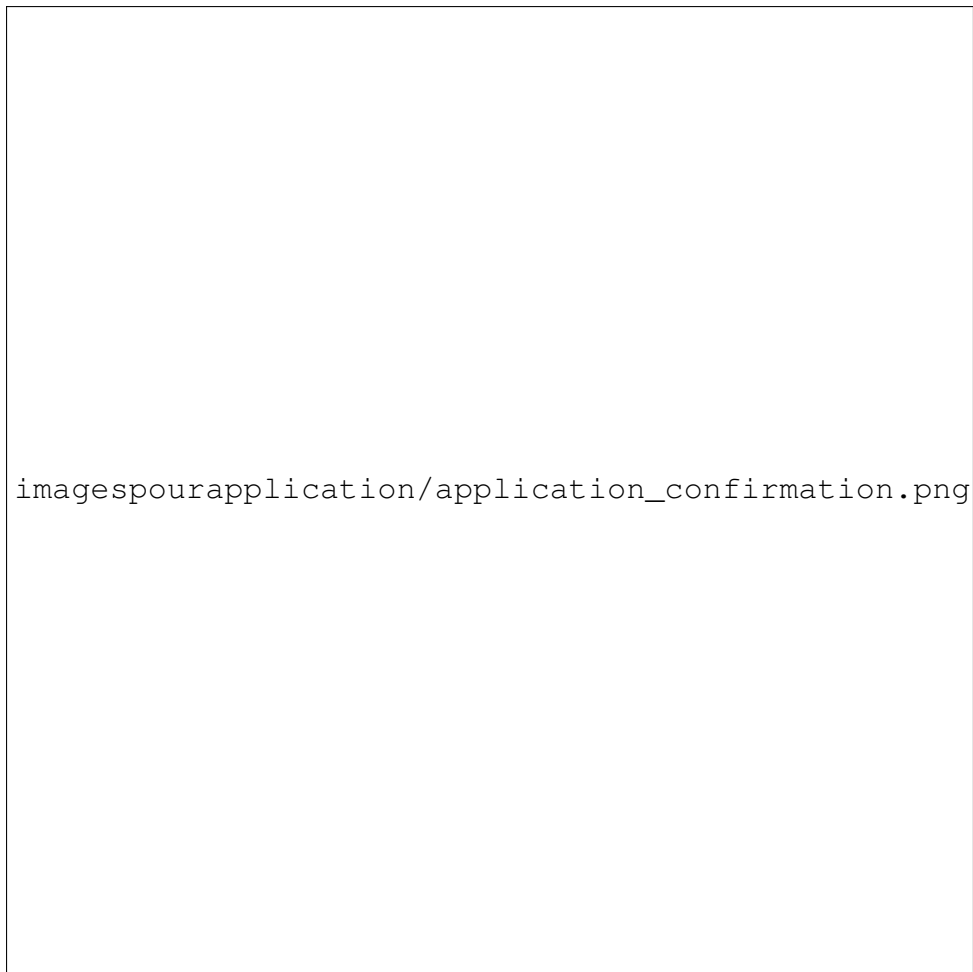


FIGURE 2.28 – *Application de confirmation avec Tkinter*

V. Conclusion

Ce chapitre a présenté la mise en œuvre complète du système de gestion de la qualité et des formations. L'architecture modulaire adoptée, combinant React pour le frontend, Spring pour le backend et MySQL pour la base de données, garantit un système performant, sécurisé et évolutif.

L'implémentation des différents modules démontre la capacité du système à répondre aux besoins identifiés lors de la phase d'analyse. Les interfaces développées offrent une expérience utilisateur intuitive et facilitent la gestion quotidienne des formations.

L'application de validation dédiée aux directeurs assure un contrôle hiérarchique efficace, tandis que le système d'évaluation permet de mesurer la qualité des formations et d'identifier les axes d'amélioration.

Déploiement de l'application

Introduction

Après la phase de développement et de tests, il est nécessaire de rendre l'application accessible aux utilisateurs finaux dans un environnement de production. Cette étape, appelée **déploiement**, consiste à mettre en ligne les différentes composantes de la solution tout en assurant leur bon fonctionnement, leur sécurité et leur performance.

Dans le cadre de ce projet, le déploiement a été réalisé sur la plateforme **Microsoft Azure**, qui offre des services adaptés aux technologies utilisées : **Spring Boot** pour le backend, **React** pour le frontend et **MySQL** pour la base de données. Ce choix a permis de bénéficier d'une infrastructure cloud moderne, fiable et hautement disponible.

I. Architecture de déploiement

L'application repose sur une architecture **trois tiers**, séparant les couches de présentation, de logique métier et de données. Cette approche facilite la maintenance, la mise à jour et l'évolutivité du système.

- La couche **Frontend**, développée avec React, fournit l'interface utilisateur et communique avec l'API backend via des requêtes HTTP sécurisées.
- La couche **Backend**, conçue avec Spring Boot, gère la logique métier et expose des services RESTful.
- La couche **Base de données**, construite avec MySQL, stocke l'ensemble des informations liées aux employés, formations et services.

Chaque couche est hébergée sur un service Azure spécifique afin d'assurer la modularité et la résilience du système.

II. Déploiement de la base de données

La base de données a été déployée sur le service **Azure Database for MySQL Flexible Server**. Ce service managé simplifie la gestion des sauvegardes, la sécurité des accès et la maintenance du serveur.

Après la création du serveur MySQL, le schéma de la base de données a été importé depuis l'environnement local. Des règles de pare-feu ont ensuite été configurées pour autoriser uniquement le backend à se connecter, garantissant ainsi la confidentialité des données.

III. Déploiement du backend Spring Boot

Le backend, développé avec le framework Spring Boot, a été empaqueté et déployé sur le service **Azure App Service**. Ce service permet d'héberger des applications Java sans nécessiter de configuration manuelle du serveur.

Les paramètres de connexion à la base de données et les variables d'environnement nécessaires ont été définis directement sur le portail Azure. Une fois le déploiement terminé, des tests ont été réalisés pour vérifier la disponibilité des endpoints et la stabilité du service.

IV. Déploiement du frontend React

L'interface utilisateur, développée en React et stylisée avec Tailwind CSS, a été déployée sur le service **Azure Static Web Apps**. Ce service est spécialement conçu pour héberger des applications web statiques ou à page unique.

Après la construction du projet, le dossier de production a été mis en ligne sur Azure. Le lien entre le frontend et le backend a été configuré afin que l'interface consomme les données depuis l'API hébergée sur Azure App Service.

V. Mise en place de l'intégration et du déploiement continu (CI/CD)

Pour automatiser le processus de déploiement, une chaîne **CI/CD (Continuous Integration / Continuous Deployment)** a été mise en place à l'aide de **GitHub Actions**. Ce mécanisme permet de :

- Construire et tester automatiquement le code à chaque modification.
- Déployer automatiquement le frontend et le backend sur leurs environnements respectifs.
- Garantir la cohérence et la fiabilité du code déployé.

Grâce à ce pipeline, le projet bénéficie d'un cycle de livraison rapide et d'une maintenance simplifiée.

VI. Sécurité et configuration CORS

La sécurité a occupé une place centrale lors du déploiement. Une configuration CORS (Cross-Origin Resource Sharing) a été mise en place pour permettre uniquement au domaine du frontend d'accéder à l'API. Des certificats SSL ont également été activés afin de garantir la sécurité des communications entre les différents services.

VII. Résultat final

À l'issue du déploiement, l'application est accessible en ligne, opérationnelle et performante. Chaque composant (frontend, backend et base de données) fonctionne harmonieusement sur la plateforme Azure, garantissant ainsi la stabilité, la sécurité et la disponibilité du système.

Conclusion du chapitre

Le déploiement sur la plateforme Microsoft Azure a permis de mettre en production une application fiable, évolutive et sécurisée. L'utilisation de services cloud managés pour chaque couche de l'architecture assure une maintenance facilitée et une haute disponibilité.

Cette étape marque la concrétisation technique du projet et la transition vers la phase d'exploitation. Elle ouvre également la voie à de futures améliorations, telles que la mise à l'échelle automatique, la supervision des performances ou encore l'ajout de nouvelles fonctionnalités basées sur les besoins des utilisateurs.

Conclusion générale et perspectives

Ce projet de fin d'études avait pour objectif de concevoir et de développer un système complet de gestion de la qualité et des formations. Au terme de ce travail, nous avons réussi à mettre en place une solution opérationnelle répondant aux exigences fonctionnelles et techniques identifiées.

Bilan du projet

L'analyse des besoins a permis d'identifier clairement les fonctionnalités essentielles du système, notamment la gestion des employés, la planification des formations, le suivi des présences et l'évaluation des sessions. L'architecture modulaire adoptée garantit la maintenabilité et l'évolutivité de la solution.

Le choix des technologies modernes, telles que React, Spring Framework et MySQL, assure des performances optimales et une sécurité renforcée. L'implémentation des différents modules démontre la viabilité de l'approche retenue et la pertinence des choix technologiques effectués.

L'application web développée offre une interface intuitive permettant aux responsables hiérarchiques de gérer efficacement l'ensemble des processus liés aux formations. L'application complémentaire dédiée aux directeurs facilite le processus de validation et renforce le contrôle hiérarchique.

Apports personnels

Ce projet m'a permis de consolider mes compétences en ingénierie logicielle, notamment dans les domaines suivants :

- La conception et le développement d'applications web modernes
- La maîtrise des frameworks React et Spring
- La gestion de bases de données relationnelles
- L'application des principes d'architecture logicielle
- La gestion de projet et le travail en mode agile

Ce travail m'a également sensibilisé à l'importance de la qualité logicielle, de la documentation et de l'expérience utilisateur dans le développement d'applications professionnelles.

Perspectives et améliorations futures

Plusieurs axes d'amélioration peuvent être envisagés pour enrichir le système :

Améliorations fonctionnelles :

- Intégration d'un module de gestion des compétences et des certifications
- Mise en place d'un système de recommandation intelligent pour suggérer des formations adaptées aux profils
- Développement d'une application mobile pour faciliter l'accès aux informations
- Intégration avec des plateformes d'e-learning pour proposer des formations à distance
- Mise en place d'un système de notification push pour améliorer la communication

Améliorations techniques :

- Déploiement sur une infrastructure cloud pour améliorer la disponibilité
- Mise en place d'un système de cache pour optimiser les performances
- Intégration de mécanismes d'intelligence artificielle pour l'analyse prédictive des besoins en formation
- Développement d'API REST complètes pour favoriser l'intégration avec d'autres systèmes
- Renforcement des mécanismes de sécurité avec l'authentification à deux facteurs

Aspects organisationnels :

- Extension du système à d'autres départements de l'organisation
- Mise en place d'indicateurs de performance avancés
- Intégration avec les systèmes RH existants
- Développement d'outils de reporting avancés pour la direction

En conclusion, ce projet a permis de développer une solution fonctionnelle et évolutive répondant aux besoins de gestion de la qualité et des formations. Les perspectives d'amélioration identifiées ouvrent la voie à des développements futurs enrichissant les fonctionnalités du système et renforçant sa valeur ajoutée pour l'organisation.

Bibliographie

- [1] SPRING FRAMEWORK DOCUMENTATION, *Spring Framework Reference Documentation*, Pivotal Software, 2024. <https://spring.io/projects/spring-framework>
- [2] REACT DOCUMENTATION, *React - A JavaScript library for building user interfaces*, Meta Platforms, 2024. <https://react.dev/>
- [3] MYSQL DOCUMENTATION, *MySQL 8.0 Reference Manual*, Oracle Corporation, 2024. <https://dev.mysql.com/doc/>
- [4] TAILWIND CSS DOCUMENTATION, *Tailwind CSS - Rapidly build modern websites*, Tailwind Labs, 2024. <https://tailwindcss.com/docs>
- [5] JONES, M., BRADLEY, J., SAKIMURA, N., *JSON Web Token (JWT)*, RFC 7519, Internet Engineering Task Force (IETF), 2015.
- [6] HARDT, D., *The OAuth 2.0 Authorization Framework*, RFC 6749, Internet Engineering Task Force (IETF), 2012.
- [7] FIELDING, R. T., *Architectural Styles and the Design of Network-based Software Architectures*, Doctoral dissertation, University of California, Irvine, 2000.
- [8] BECK, K., ET AL., *Manifesto for Agile Software Development*, Agile Alliance, 2001. <https://agilemanifesto.org/>
- [9] SOMMERVILLE, I., *Software Engineering*, 10th Edition, Pearson Education, 2015.
- [10] DATE, C. J., *An Introduction to Database Systems*, 8th Edition, Addison-Wesley, 2003.
- [11] OWASP FOUNDATION, *OWASP Top Ten Web Application Security Risks*, 2021. <https://owasp.org/www-project-top-ten/>
- [12] FOWLER, M., *UML Distilled : A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Addison-Wesley Professional, 2003.

Annexe A : Modèle conceptuel de données

Cette annexe présente le modèle conceptuel de données utilisé pour concevoir la base de données relationnelle du système.

Le modèle respecte les trois formes normales (3NF) et garantit l'intégrité référentielle entre les différentes entités.

Principales entités

- **Employé** : contient les informations personnelles et professionnelles
- **Formation** : détails des sessions de formation
- **Service** : organisation des départements
- **Présence** : suivi de la participation aux formations
- **Évaluation** : réponses aux questionnaires d'évaluation
- **Formateur** : informations sur les intervenants

Annexe B : Diagrammes de séquence

Cette annexe présente les diagrammes de séquence illustrant les interactions entre les différents composants du système pour les cas d'utilisation principaux.

Scénarios couverts

- Processus d'authentification d'un utilisateur
- Création d'une nouvelle formation
- Enregistrement de la présence
- Évaluation d'une formation
- Validation hiérarchique d'une formation

Annexe C : Captures d'écran supplémentaires

Cette annexe regroupe des captures d'écran complémentaires illustrant les différentes fonctionnalités du système.

Annexe D : Guide d'installation et de déploiement

Cette annexe fournit les instructions détaillées pour l'installation et le déploiement du système en environnement de production.

Prérequis système

- Serveur d'application : Apache Tomcat 9.0 ou supérieur
- Base de données : MySQL 8.0 ou supérieur
- Java Development Kit (JDK) 17 ou supérieur
- Node.js 18.0 ou supérieur
- Navigateur web moderne (Chrome, Firefox, Edge)

Étapes d'installation

1. Installation et configuration de la base de données MySQL
2. Création du schéma de base de données
3. Configuration du backend Spring
4. Compilation et déploiement du backend
5. Installation des dépendances frontend
6. Configuration et compilation du frontend React
7. Configuration de l'application Tkinter
8. Tests de validation du déploiement