



DEPARTEMENT INFORMATIQUE

RAPPORT DE PROJET DE FIN D'ANNEE

Diplôme National d'Ingénieur Informatique

Système de Gestion des Surveillances D'Examens

Elaborée par :

Manel Mbarek

Ichrak Dhahri

Ayoub Amara

Encadrante :

Nawal Kortas

Année Universitaire : 2024-2025

Remerciements

Nous souhaitons exprimer nos sincères remerciements à Madame Nawel KORTAS, notre encadrante académique, pour son accompagnement précieux, sa disponibilité constante et ses conseils éclairés tout au long de ce projet. Son expertise, sa rigueur et sa bienveillance ont fortement contribué à la bonne conduite et à la réussite de notre travail.

Nous remercions également tous les enseignants et personnels de l'ISSAT Sousse qui nous ont soutenus, directement ou indirectement, durant cette expérience enrichissante.

Table des matières

Introduction générale	1
1 Contexte du projet	2
1.1 Introduction	2
1.2 Cadre de projet général	2
1.3 Organisme d'accueil	2
1.4 Présentation du Contexte	3
1.5 Problématique	3
1.6 Etude de l'existant	3
1.6.1 Description de l'existant au sein de l'ISSAT	3
1.6.2 Description des solutions existantes sur le marché	4
1.6.3 Critique de l'existant	4
1.7 Solution proposée	5
1.7.1 Objectifs du projet	5
1.8 Méthodologie de gestion du projet	6
1.9 Conclusion	6
2 Etude de l'art	7
2.1 Introduction	7
2.2 Gestion des Surveillances d'Examens	7
2.3 Algorithmes d'Optimisation des Ressources	8
2.3.1 Explication de l'algorithme utilisé	9
2.3.2 Principales contraintes implémentées	9
2.3.3 Fonctionnement de l'algorithme	9
2.4 Systèmes Existants et Leurs Limites	10
2.5 Apport de la Solution Proposée	10
2.6 Conclusion	11
3 Analyse des besoins	12
3.1 Introduction	12
3.2 Spécification des exigences	12
3.2.1 Exigences fonctionnelles	12
3.2.2 Exigences non fonctionnelles	13
3.3 Backlog de produit	14
3.4 Architecture	15
3.4.1 Diagramme de cas d'utilisation globale	15
3.4.2 Diagramme de classes général	17
3.5 Architecture globale	18
3.5.1 Couche Présentation (Frontend)	18

3.5.2	Couche Application (Backend)	18
3.5.3	Couche Persistance	18
3.6	Conclusion	19
4	Réalisation	20
3.1	Introduction	20
3.2	Environnement de travail	20
3.2.1	Environnement matériel	20
3.2.2	Environnement logiciel	20
3.3	Technologies utilisées :	22
3.4	Tâches Réalisées	24
3.4.1	Interfaces Admin :	24
3.4.2	Interfaces Enseignant :	26
3.5	Conclusion	26
	Conclusion Générale	28
	Bibliographie	29

Table des figures

1.1	Logo de l'ISSAT	2
1.2	Cycle de vie d'une méthode SCRUM	6
2.3	Exemple de feuille de calcul utilisée traditionnellement pour la planification des surveillances	8
2.4	Schéma illustratif de l'approche hybride utilisée	9
3.5	Diagramme de cas d'utilisation : Admin	16
3.6	Diagramme de cas d'utilisation : Enseignant	17
3.7	Diagramme de classe Global	18
3.8	Visual Studio Code	21
3.9	Logo postman	21
3.10	Logo Draw.io	21
3.11	Logo MongoDB	22
3.12	Bootstrap	22
3.13	Logo react	22
3.14	Logo nodejs	23
3.15	Logo express	23
3.16	Logo JavaScript	23
3.17	Logo HTML	24
3.18	Logo CSS	24
3.19	Interface de gestion des comptes enseignants.	25
3.20	Interface D'ajout des Enseignants	25
3.21	Planification Des Surveillances	26
3.22	Interface de Profile de l'enseignant	26

Liste des tableaux

3.1	User Stories du projet	15
3.2	Environnement matériel	20

Introduction générale

Dans le cadre de notre formation d'ingénieur, nous sommes appelés à mobiliser nos compétences théoriques à travers la réalisation de projets concrets. Ces projets, à visée pédagogique, nous permettent de développer des solutions innovantes répondant à des problématiques réelles rencontrées dans les milieux académiques ou professionnels.

Le présent rapport s'inscrit dans cette dynamique et porte sur le développement d'une application web dédiée à la gestion automatisée des surveillances d'examens. Cette problématique, bien que récurrente dans les établissements d'enseignement supérieur, reste encore largement traitée de manière manuelle, ce qui engendre des difficultés organisationnelles, une perte de temps significative et un risque d'erreurs élevé.

Notre objectif principal est de proposer une solution informatique capable d'alléger la charge administrative, de garantir une équité entre les enseignants et de respecter un ensemble complexe de contraintes pédagogiques et logistiques. Ce travail a été réalisé au sein de l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse (ISSAT Sousse), dans le cadre d'un projet académique mené durant notre deuxième année du cycle d'ingénieur.

Ce rapport est structuré en plusieurs chapitres. Le premier présente le contexte général du projet, les limites des solutions existantes ainsi que la démarche méthodologique que nous avons adoptée. Les chapitres suivants détaillent le cadre théorique, la conception fonctionnelle et technique, le développement de la solution, ainsi qu'une évaluation des résultats obtenus. L'ensemble vise à démontrer la pertinence de la solution proposée face aux besoins identifiés.

chapitre 1

Contexte général du projet

1.1 Introduction

Dans ce chapitre, nous effectuons une analyse générale du projet. Nous commençons tout d'abord par présenter l'organisme d'accueil au sein duquel s'est déroulé notre projet. Ensuite, nous exposons le contexte de notre projet ainsi que la méthode suivie pour aboutir aux résultats attendus.

1.2 Cadre de projet général

Ce projet a été réalisé au sein de l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse (ISSAT Sousse), dans le cadre d'un projet académique effectué au cours de la deuxième année du cycle d'ingénieur. Il s'est déroulé sur une période de deux mois, durant la session estivale.

1.3 Organisme d'accueil

L'organisme d'accueil de ce projet est l'ISSAT Sousse, établissement public d'enseignement supérieur situé à Sousse, Tunisie. L'institut propose une formation d'ingénieurs dans plusieurs disciplines scientifiques et techniques, et encourage la mise en pratique des compétences acquises à travers des projets académiques encadrés.



FIGURE 1.1 – Logo de l'ISSAT

1.4 Présentation du Contexte

Dans le cadre de l'organisation des examens universitaires, la gestion des surveillances représente un défi logistique majeur pour les administrations. Chaque session d'examens nécessite une planification minutieuse pour assurer une répartition équitable des charges entre les enseignants, respecter diverses contraintes temporelles et optimiser l'utilisation des ressources disponibles.

L'établissement d'enseignement supérieur concerné par ce projet fait face à cette problématique de manière récurrente. La planification manuelle des surveillances d'examens mobilise actuellement des ressources administratives importantes et génère des plannings sous-optimaux qui peuvent mécontenter le corps enseignant et compliquer l'organisation générale des examens.

Dans ce contexte, la digitalisation du processus de gestion des surveillances apparaît comme une solution prometteuse pour améliorer l'efficacité administrative et la satisfaction des différents acteurs impliqués.

1.5 Problématique

La planification et la gestion des surveillances d'examens soulèvent plusieurs défis complexes :

Complexité organisationnelle : L'attribution manuelle des surveillances est chronophage et sujette aux erreurs humaines. Elle nécessite la prise en compte simultanée de multiples contraintes.

Équité de répartition : L'équilibrage des charges de surveillance entre les enseignants doit respecter leurs statuts, leurs coefficients et leurs enseignements.

Contraintes temporelles strictes : La continuité des surveillances sur une même journée ou des jours successifs est essentielle pour optimiser le temps des enseignants.

Redondance des tâches : Le processus actuel implique de nombreuses manipulations manuelles des données et des communications individuelles avec les enseignants.

Flexibilité limitée : Les modifications de dernière minute sont difficiles à gérer et peuvent déstabiliser l'ensemble du planning.

Absence de vue d'ensemble : Les outils actuels ne permettent pas une visualisation claire et complète des attributions de surveillance.

Comment concevoir et implémenter un système informatique capable d'automatiser l'attribution des surveillances d'examens tout en respectant l'ensemble des contraintes pédagogiques, organisationnelles et humaines ?

1.6 Etude de l'existant

Avant d'entamer tout projet, il faut avoir une idée claire et précise sur l'existant afin d'indiquer l'utilité de notre solution et de garantir un projet fiable et performant qui répond au besoin demandé.

1.6.1 Description de l'existant au sein de l'ISSAT

Actuellement, à l'ISSAT Sousse, la gestion des surveillances d'examens repose sur un processus essentiellement manuel qui présente plusieurs limitations :

Processus manuel : L'administrateur responsable utilise principalement Microsoft Excel pour créer et gérer les plannings de surveillance. Cette méthode nécessite de nombreuses manipulations manuelles et est chronophage.

Collecte des disponibilités : Les enseignants doivent communiquer leurs disponibilités via des formulaires papier ou par email, ce qui complexifie la centralisation des informations.

Attribution des surveillances : L'affectation des surveillants aux examens se fait manuellement, sans algorithme d'optimisation, en essayant de respecter au mieux les contraintes humaines et organisationnelles.

Communication des plannings : Les plannings finalisés sont généralement diffusés par email ou affichage physique, nécessitant des mises à jour manuelles en cas de modification.

Absence de système d'information centralisé : Les données relatives aux enseignants, examens et salles sont dispersées dans différents fichiers, compliquant leur gestion cohérente.

1.6.2 Description des solutions existantes sur le marché

Plusieurs solutions existent sur le marché pour la gestion des surveillances d'examens, parmi lesquelles :

ExamScheduler Pro : Solution commerciale offrant une planification automatisée des surveillances avec prise en compte de multiples contraintes. Cette solution propose une interface web intuitive mais présente un coût d'acquisition élevé et une personnalisation limitée.

UniExam Manager : Plateforme complète de gestion des examens universitaires incluant un module de gestion des surveillances. Elle permet l'importation de données depuis diverses sources et l'exportation de plannings personnalisés, mais nécessite une infrastructure informatique conséquente.

Systèmes de gestion académique intégrés : Certaines suites logicielles de gestion académique (comme Apogée ou SCOLARIX) proposent des modules de gestion d'examens, mais ceux-ci sont souvent peu flexibles et insuffisamment adaptés aux spécificités des établissements.

Solutions open-source : Quelques projets comme ExamTimetabling offrent des fonctionnalités de base gratuitement, mais nécessitent généralement d'importantes adaptations pour répondre aux besoins spécifiques d'un établissement.

1.6.3 Critique de l'existant

L'analyse des solutions actuellement utilisées à l'ISSAT et disponibles sur le marché permet d'identifier plusieurs limitations :

- **Limites du système actuel à l'ISSAT** :
 - Processus chronophage mobilisant des ressources administratives importantes
 - Risque élevé d'erreurs humaines dans l'attribution des surveillances
 - Difficultés à garantir l'équité entre les enseignants
 - Absence de traçabilité et d'historique des surveillances
 - Communication inefficace des plannings et de leurs modifications
 - Impossibilité d'avoir une vue d'ensemble sur la charge de surveillance
- **Limites des solutions commerciales** :

- Coûts d'acquisition et de maintenance élevés
- Personnalisation limitée ne permettant pas de respecter toutes les contraintes spécifiques à l'ISSAT
- Complexité d'intégration avec les systèmes existants
- Courbe d'apprentissage parfois importante pour les utilisateurs
- **Besoins non couverts :**
 - Interface utilisateur adaptée spécifiquement aux processus de l'ISSAT
 - Algorithme d'optimisation respectant l'ensemble des contraintes spécifiques de l'établissement
 - Système flexible permettant des ajustements manuels après optimisation automatique
 - Gestion des cas particuliers et des imprévus

Cette analyse justifie pleinement le développement d'une solution sur mesure, capable de répondre aux besoins spécifiques de l'ISSAT tout en s'adaptant à ses contraintes organisationnelles et techniques.

1.7 Solution proposée

1.7.1 Objectifs du projet

Ce projet vise à développer une application web complète répondant aux objectifs suivants :

- **Automatisation intelligente** : concevoir et implémenter un algorithme d'optimisation capable d'attribuer automatiquement les surveillances d'examens selon des critères précis :
 - Exactement 2 surveillants par examen
 - Optimisation des surveillances sur les mêmes journées pour chaque enseignant
 - Maintien d'une réserve de 3 à 4 enseignants par jour
 - Priorisation des enseignants responsables des matières
 - Continuité des surveillances sur des jours successifs
 - Équilibrage selon les coefficients des enseignants
- **Gestion intégrée des données** : permettre l'importation de données depuis des fichiers Excel ou via des formulaires pour les trois entités principales :
 - Enseignants et leurs caractéristiques
 - Calendrier des examens
 - Répartition des salles et groupes
- **Interface utilisateur intuitive** : concevoir et développer une interface web responsive permettant :
 - La visualisation claire des plannings de surveillance
 - La gestion des enseignants et utilisateurs
 - L'exportation des plannings finaux au format PDF
- **Système multiutilisateur** : mettre en place un système d'authentification distinguant :
 - Les administrateurs (gestion complète)
 - Les enseignants (consultation de leurs surveillances)
- **Évaluation quantitative** : mesurer l'efficacité du système en termes :
 - De temps d'exécution de l'algorithme
 - De respect des contraintes définies

- D'équilibrage des charges de surveillance
- De satisfaction des utilisateurs

1.8 Méthodologie de gestion du projet

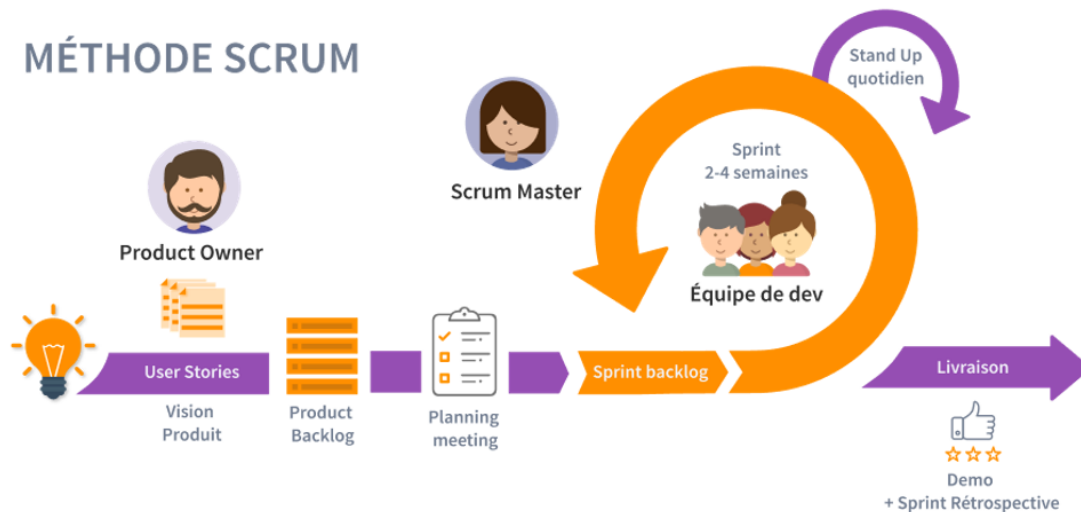


FIGURE 1.2 – Cycle de vie d'une méthode SCRUM

La réalisation de ce projet a été menée selon la méthode Scrum. Cette approche de développement agile divise le travail en cycles courts nommés "sprints", généralement de deux à quatre semaines. Chaque sprint cible des objectifs précis, qu'il s'agisse de nouvelles fonctionnalités ou d'améliorations.

Le processus Scrum intègre des réunions régulières : planification, revue et rétrospective de sprint. Ces moments clés assurent la transparence du projet et permettent d'adapter les priorités selon les retours utilisateurs.

Cette méthodologie a favorisé une construction flexible de la plateforme, capable de s'adapter aux besoins changeants tout en maintenant un haut niveau de qualité. Elle a aussi renforcé la communication au sein de l'équipe de développement, garantissant ainsi une progression efficace et alignée sur les attentes des utilisateurs finaux.

1.9 Conclusion

Ce chapitre a exposé le contexte du projet de développement d'une application web dédiée à la gestion des surveillances d'examens à l'ISSAT Sousse. L'analyse de l'existant a révélé les limites des méthodes actuelles, justifiant la mise en place d'une solution sur mesure. Celle-ci repose sur un algorithme d'optimisation et une interface intuitive. La méthodologie Scrum adoptée assure un développement structuré et adaptable. Les chapitres suivants traiteront les aspects techniques du projet.

chapitre 2

Cadre Théorique et État de l'Art

2.1 Introduction

Ce chapitre constitue une étude approfondie de l'état de l'art concernant la gestion des surveillances d'examens. Il présente d'abord les défis organisationnels auxquels sont confrontées les institutions académiques, puis explore les différentes approches algorithmiques permettant d'optimiser cette tâche complexe. Une analyse critique des systèmes existants est ensuite proposée, avant de conclure par les apports innovants de la solution développée dans le cadre de ce projet.

2.2 Gestion des Surveillances d'Examens

La gestion des surveillances d'examens s'inscrit dans le domaine plus large de l'allocation optimale de ressources sous contraintes. Les établissements d'enseignement supérieur font face à plusieurs défis dans ce domaine :

Défis organisationnels

- Planification de centaines d'examens sur des périodes limitées
- Allocation équitable des tâches entre enseignants aux statuts variés
- Gestion des contraintes temporelles et spatiales
- Nécessité de disposer d'une réserve pour gérer les imprévus

Approches traditionnelles

Les approches traditionnelles reposent souvent sur :

- Des processus manuels ou semi-automatisés
- Des feuilles de calcul Excel
- Des systèmes de gestion d'emploi du temps génériques
- Un traitement séquentiel des contraintes et non global

Ces approches présentent des limitations importantes en termes d'optimalité, de flexibilité et de temps de traitement.

Catégorie du plan de surveillance Prototype Présérie Production logistique			Plan de surveillance							Page:	
Numéro du plan			Niveau de changement de conception/révision					Date création		Date de révision	
Numéro de pièce/produit/processus			Personne responsable					Approbation service ingénierie du client/date			
Désignation/Description			Membre de l'équipe					Approbation service qualité du client/date			
Fournisseur/Usine		code/ID fournisseur	Approbation fournisseur/usine/date					Autre approbation/ date			
Numéro de pièce/ processus	Non de processus/ description d'opération	Machine/ outi de fabrication	caractéristiques			Méthodes					Plan d'action
			N°	Produit	process	Spécification/ tolérance	Technique d'évaluation	taille d'échantillon	Fréquence	Méthode de mesure	

FIGURE 2.3 – Exemple de feuille de calcul utilisée traditionnellement pour la planification des surveillances

2.3 Algorithmes d'Optimisation des Ressources

La résolution du problème d'attribution des surveillances relève des problèmes d'optimisation combinatoire sous contraintes. Les familles d'algorithmes suivantes peuvent être utilisées :

Algorithmes exacts

- Programmation linéaire entière
- Méthodes de type branch-and-bound, branch-and-cut
- Programmation par contraintes

Métaheuristiques

- Algorithmes génétiques
- Recuit simulé
- Recherche tabou
- Algorithmes de colonies de fourmis

Algorithmes gloutons adaptés

- Prise en compte directe des contraintes
- Règles de priorité
- Heuristiques constructives

Dans ce projet, une approche hybride combinant un algorithme glouton et des optimisations locales a été retenue.

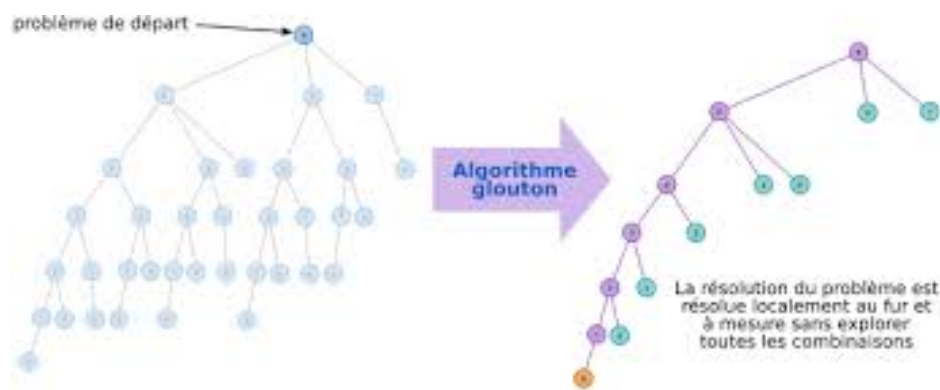


FIGURE 2.4 – Schéma illustratif de l'approche hybride utilisée

2.3.1 Explication de l'algorithme utilisé

C'est un algorithme d'optimisation sous contraintes utilisant une approche heuristique gloutonne avec plusieurs phases d'affectation prioritaires. On peut le classer plus précisément comme un algorithme d'affectation de ressources multi-critères avec contraintes séquentielles.

2.3.2 Principales contraintes implémentées

- **Contraintes de capacité :**
 - Chaque salle d'examen doit avoir exactement 2 surveillants.
 - Chaque enseignant a un quota prédéfini de surveillances à effectuer (Surveillance).
- **Contraintes de séquence et continuité :**
 - Préférence pour les séquences de séances consécutives ($s1 \rightarrow s2 \rightarrow s3 \rightarrow s4$).
 - Optimisation des transitions entre jours (préférence pour $s4$ le jour J puis $s1$ le jour $J+1$).
 - Regroupement des surveillances sur des jours consécutifs par blocs de 2-3 jours.
- **Contraintes d'expertise et de compétence :**
 - Priorité aux enseignants qui enseignent les matières examinées (CodeMatiere).
 - Association préférentielle entre enseignants et cours qu'ils enseignent.
- **Contraintes de disponibilité et d'exclusion :**
 - Éviter d'affecter un enseignant plusieurs fois pour la même date/séance.
 - Éviter d'affecter un enseignant comme surveillant et réserve simultanément.
- **Contraintes d'équilibrage :**
 - Distribution équitable des surveillances entre les enseignants.
 - Répartition des professeurs en réserve (cible de 3-4 par jour).
- **Contraintes structurelles :**
 - Respect de l'organisation des examens par date, séance, salle.
 - Respect des affectations des groupes d'étudiants aux salles.

2.3.3 Fonctionnement de l'algorithme

L'algorithme fonctionne selon un processus de plusieurs phases :

1. **Initialisation et préparation des données :**
 - Regroupement des examens par date et séance.

- Organisation des dates en blocs consécutifs.
- Préparation des structures de suivi pour les enseignants.
- 2. Affectation principale avec priorités :**
 - Phase 1 : Affectation prioritaire des enseignants respectant les séquences.
 - Phase 2 : Complétion avec des enseignants ayant une séquence en cours.
 - Phase 3 : Affectation de nouveaux enseignants si nécessaire.
- 3. Gestion des professeurs de réserve :**
 - Affectation intelligente pour atteindre un quota quotidien.
- 4. Vérification et optimisation :**
 - Réaffectation des enseignants n'ayant pas complété leur quota.
 - Optimisation globale du planning pour améliorer les séquences.
- 5. Validation finale :**
 - Vérification que chaque salle a exactement 2 surveillants.
 - Analyse de la continuité des jours de surveillance.
 - Vérification du nombre suffisant d'enseignants en réserve.

L'algorithme utilise un système sophistiqué de calcul de scores pour prioriser les affectations, en particulier via la fonction `calculateSequenceScore()` qui évalue la qualité d'une séquence pour un enseignant.

C'est un algorithme complexe qui implémente de nombreuses contraintes spécifiques au domaine de l'organisation des examens dans un établissement d'enseignement.

2.4 Systèmes Existants et Leurs Limites

Systèmes de gestion d'emplois du temps génériques

- Solutions commerciales comme ADE, UnivTimetable, Celcat
- Manque de flexibilité pour les contraintes spécifiques
- Intégration et personnalisation limitées
- Coût élevé

Applications web spécialisées

- Peu adaptées aux contextes locaux
- Problèmes d'intégration avec les systèmes existants
- Algorithmes trop simples

Les solutions actuelles manquent souvent d'une optimisation robuste, d'une interface moderne, et d'une gestion fine des contraintes.

2.5 Apport de la Solution Proposée

Optimisation algorithmique avancée

- Approche multi-objectifs
- Équilibrage dynamique des surveillances
- Optimisation temporelle
- Gestion intelligente des réserves

Architecture technique moderne

- Stack MERN (MongoDB, Express, React, Node.js)
- Architecture modulaire et évolutive
- Approche API-first
- Base de données NoSQL

Fonctionnalités métier avancées

- Import/export de données
- Gestion des profils enseignants
- Génération automatique de documents
- Système de notifications

Cette solution constitue une avancée majeure pour les établissements d'enseignement supérieur souhaitant gérer efficacement les surveillances d'examens.

2.6 Conclusion

L'étude menée dans ce chapitre met en lumière les limites des approches actuelles de planification des surveillances et souligne l'intérêt d'une solution spécifique, à la fois flexible et performante. En combinant des techniques d'optimisation avancées à une architecture technique moderne, la solution proposée répond aux exigences réelles des établissements d'enseignement supérieur et constitue une contribution significative dans le domaine.

Chapitre 3

Analyse et Spécification des besoins

3.1 Introduction

Ce chapitre présente l'analyse détaillée des besoins pour notre système d'attribution de surveillances d'examens. Nous y définissons les exigences fonctionnelles et non fonctionnelles qui guideront le développement du système, ainsi que le backlog du produit et l'architecture globale.

3.2 Spécification des exigences

Nous présentons dans cette section les exigences fonctionnelles et non fonctionnelles qui définissent les attentes vis-à-vis du système à développer.

3.2.1 Exigences fonctionnelles

L'application doit offrir des services répondant aux besoins de l'utilisateur. Dans notre système, ces besoins sont définis par les critères suivants :

Gestion des utilisateurs

- Création et gestion des comptes administrateurs
- Création des comptes enseignants via un système de préinscription validé par l'administrateur, avec envoi automatique du mot de passe par e-mail.
- Système de réinitialisation des mots de passe

Import et saisie des données

- Import de fichiers Excel pour les trois entités principales
- Formulaire de saisie manuelle des enseignants avec validation
- Modification et suppression des données existantes

Gestion des enseignants

- Création et modification des profils enseignants
- Association avec les matières enseignées
- Attribution des coefficients de surveillance

Gestion du calendrier des examens

- Définition des jours et séances d'examen
- Association des matières aux créneaux d'examen
- Spécification des spécialités concernées
- Visualisation du calendrier des examens

Gestion des salles et groupes

- Attribution des groupes d'étudiants aux salles
- Définition des capacités et caractéristiques des salles
- Association des matières aux salles d'examen

Attribution automatique des surveillances

- Lancement de l'algorithme d'optimisation
- Paramétrage des contraintes et priorités
- Affichage des résultats d'attribution
- Possibilité d'ajustements manuels post-attribution

Validation des contraintes

- Vérification automatique du respect des contraintes
- Alerte sur les contraintes non satisfaites
- Suggestions de correction

Interface administrateur

- Gestion complète des données
- Lancement de l'algorithme
- Visualisation et modification des résultats
- Export des plannings finalisés

Interface enseignant

- Consultation du planning de surveillance
- Gestion du profil personnel

3.2.2 Exigences non fonctionnelles

Lorsque les besoins fonctionnels définissent les fonctionnalités spécifiques du produit, les besoins non fonctionnels servent d'indicateurs de la qualité de l'exécution de ces fonctionnalités. Pour notre application, ces indicateurs se basent principalement sur :

- **La convivialité** : Le logiciel doit être simple d'utilisation, avec des interfaces conviviales, intuitives, ergonomiques et conçues pour répondre aux besoins de l'utilisateur.
- **L'extensibilité** : L'application devra être extensible, c'est-à-dire qu'elle soit flexible à tout changement des fonctionnalités.
- **La performance** : Le logiciel doit fournir des services performants qui répondent aux besoins de l'utilisateur.

- **La maintenance** : Le code doit être compréhensible et modulaire pour des raisons de réutilisation et de modification.
- **Les contraintes esthétiques** : Les polices choisies doivent garantir une lecture facile et être adaptées à la nature de l'information. La page d'accueil doit offrir une présentation épurée, avec un contenu clair tout en captant l'attention de l'utilisateur.
- **La sécurité** : Le système doit garantir la confidentialité des données et protéger l'accès aux informations sensibles.
- **La disponibilité** : L'application doit être accessible à tout moment pendant les périodes critiques de planification des examens.

3.3 Backlog de produit

Le backlog du produit est un inventaire dynamique et hiérarchisé qui répertorie l'ensemble des fonctionnalités, améliorations, corrections et exigences à développer pour un produit.

Le backlog du produit présenté dans le tableau 3.2 comprend les champs suivants :

- **ID** : C'est un nombre unique et auto-incrémenté pour chaque histoire utilisateur.
- **User story** : C'est une phrase décrivant la fonctionnalité désirée par le client.
- **Priorité** : C'est la valeur métier qui dirige la priorisation du développement des histoires utilisateurs suivant les attentes et les besoins du client ; avec 1 : Priorité la plus forte et 4 : Priorité la plus faible.
- **Estimation** : C'est une estimation de la complexité, elle est une valeur entière.

ID	User Story	Priorité	Estimation
1	En tant qu'administrateur, je veux pouvoir créer des comptes utilisateurs pour les enseignants afin qu'ils puissent accéder au système.	1	3
2	En tant qu'administrateur, je veux pouvoir importer la liste des enseignants à partir d'un fichier Excel afin de gagner du temps.	1	5
3	En tant qu'administrateur, je veux définir le calendrier des examens pour organiser les surveillances.	1	4
4	En tant qu'administrateur, je veux importer les données des salles et des groupes pour préparer les répartitions.	2	3
5	En tant qu'administrateur, je veux lancer l'algorithme d'attribution des surveillances pour générer automatiquement un planning optimal.	1	8
6	En tant qu'administrateur, je veux modifier manuellement les attributions générées pour ajuster selon les besoins spécifiques.	2	4
7	En tant qu'administrateur, je veux exporter les plannings de surveillance en PDF pour les distribuer.	2	3
8	En tant qu'enseignant, je veux consulter mon planning de surveillance pour connaître mes obligations.	1	2
9	En tant qu'enseignant, je veux modifier mon profil personnel pour mettre à jour mes informations.	3	2
10	En tant qu'enseignant, je veux être notifié de tout changement dans mon planning de surveillance.	3	3
11	En tant qu'administrateur, je veux définir des contraintes spécifiques pour certains enseignants afin de respecter leurs disponibilités.	2	5
12	En tant qu'administrateur, je veux visualiser des statistiques sur la répartition des surveillances pour vérifier l'équité.	3	4

TABLE 3.1 – User Stories du projet

3.4 Architecture

3.4.1 Diagramme de cas d'utilisation globale

Les cas d'utilisation sont une méthode structurée pour identifier, clarifier et organiser les exigences fonctionnelles d'un système. La figure 3.6 illustre le diagramme du cas d'utilisation général de notre projet :

Acteurs principaux :

- **Administrateur** : Responsable de la gestion complète du système
- **Enseignant** : Utilisateur consultant ses surveillances attribuées

Cas d'utilisation principaux pour l'administrateur :

- Gérer les enseignants (CRUD)
- Gérer le calendrier d'examens (CRUD)
- Gérer les salles et répartitions (CRUD)
- Importer des données depuis Excel
- Exécuter l'algorithme d'attribution
- Modifier manuellement les attributions

- Générer les plannings PDF
- Gérer les comptes utilisateurs

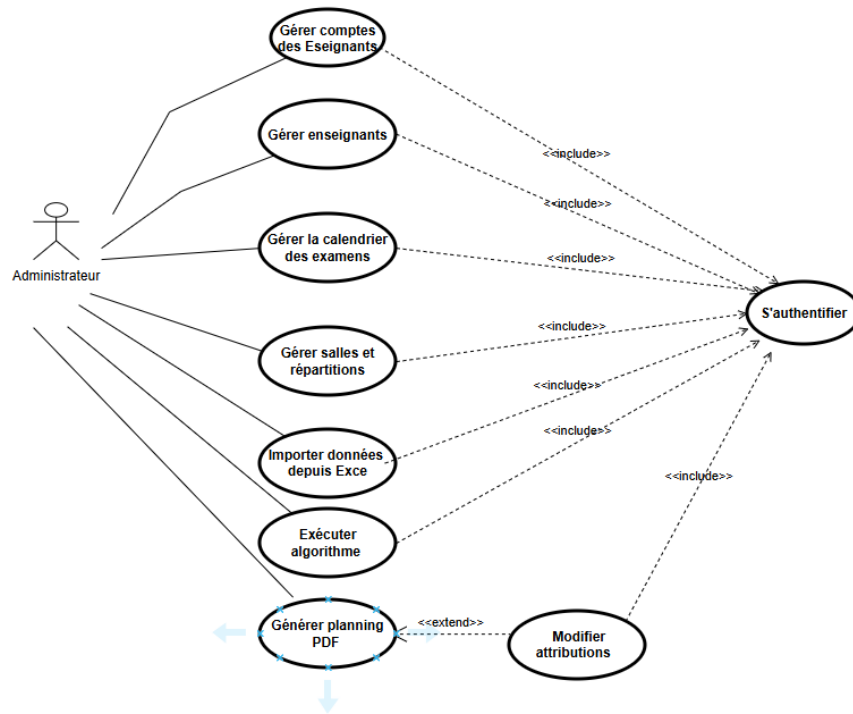


FIGURE 3.5 – Diagramme de cas d'utilisation : Admin

Cas d'utilisation principaux pour l'enseignant :

- Consulter son planning de surveillance
- Modifier son profil personnel
- Changer son mot de passe

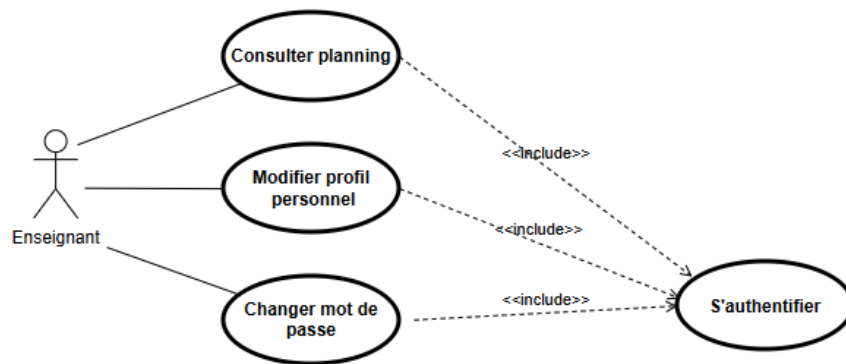


FIGURE 3.6 – Diagramme de cas d'utilisation : Enseignant

3.4.2 Diagramme de classes général

Le diagramme de classe représente les classes intervenant dans le système. Il est une représentation statique des éléments qui composent un système et de leurs relations. La classe est un concept abstrait qui permet de représenter toutes les entités d'un système.

La figure 3.8 illustre le diagramme de classe général de notre projet :

Classes principales :

- **Enseignant** : Représente un enseignant avec ses attributs et ses surveillances
- **Calendrier** : Définit un examen avec sa date, séance et matière
- **Répartition** : Associe une salle à un groupe et à des surveillants
- **Utilisateur** : Stocke les informations d'authentification
- **Planning** : Représente le résultat final de l'attribution des surveillances

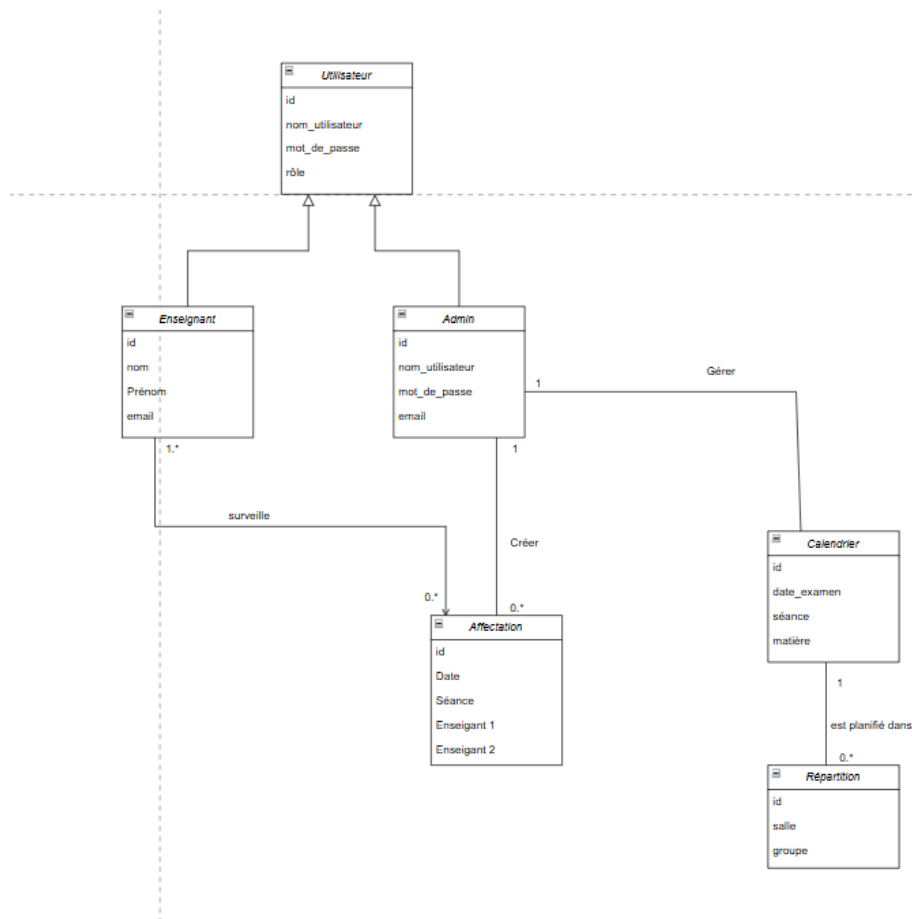


FIGURE 3.7 – Diagramme de classe Global

3.5 Architecture globale

L'architecture globale du système repose sur une séparation claire des responsabilités :

3.5.1 Couche Présentation (Frontend)

- **Composants React** : Interface utilisateur interactive
- **State Management** : Gestion de l'état applicatif
- **Services API** : Communication avec le backend
- **Routing** : Navigation entre les différentes vues

3.5.2 Couche Application (Backend)

- **API RESTful** : Endpoints pour les opérations CRUD
- **Middleware** : Gestion de l'authentification et autorisations
- **Services Métier** : Logique métier et orchestration
- **Algorithme d'Optimisation** : Cœur algorithmique du système

3.5.3 Couche Persistance

- **MongoDB** : Stockage NoSQL des données

- **Mongoose** : ODM (Object Document Mapper)
- **Indexation** : Optimisation des requêtes fréquentes
- **Validation** : Intégrité des données

3.6 Conclusion

Dans ce chapitre, nous avons présenté l'analyse détaillée des besoins pour notre système d'attribution de surveillances d'examens. Les exigences fonctionnelles et non fonctionnelles ont été clairement identifiées, accompagnées d'un backlog de produit complet qui guidera le développement. L'architecture proposée permettra de répondre efficacement à ces besoins tout en garantissant la flexibilité et la maintenabilité du système.

Le prochain chapitre détaillera la conception détaillée du système, en se concentrant sur l'algorithme d'optimisation qui constitue le cœur de notre solution.

chapitre 4

Réalisation et Test

3.1 Introduction

Dans ce chapitre, nous examinerons l’environnement matériel et logiciel requis pour le développement, ainsi que les technologies clés employées. Nous détaillerons également le langage de programmation et le framework utilisés. En complément, nous présenterons un aperçu de l’interface utilisateur à travers des captures d’écran, et aborderons les tests réalisés pour garantir la qualité et le bon fonctionnement de la plateforme.

3.2 Environnement de travail

3.2.1 Environnement matériel

Ordinateur	caracteristiques
Modèle	Acer
Mémoire vive	12 Go
Stockage	1 To HDD et 512 GO SSD
Carte graphique	NVIDIA GeForce MX350
Processeur	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
Système d’exploitation	windows 10

TABLE 3.2 – Environnement matériel

3.2.2 Environnement logiciel

Dans notre projet, nous avons utilisé les logiciels suivants :

Visual Studio Code

Visual Studio est un environnement de développement intégré (IDE) créé par Microsoft. Il permet de coder, déboguer et tester des applications pour plusieurs plateformes. Il prend en charge plusieurs langages de programmation, comme C, PHP, Python et JavaScript, et offre des outils avancés pour le développement, comme le contrôle de version, l’intelligence artificielle (IntelliSense), et des extensions pour personnaliser l’expérience.



FIGURE 3.8 – Visual Studio Code

Postman

Postman est la plateforme unique pour concevoir, développer et faire évoluer vos API, ensemble. Rejoignez plus de 40 millions d'utilisateurs qui ont consolidé leurs flux de travail et optimisé leurs API, le tout sur une plateforme unique et performante.



FIGURE 3.9 – Logo postman

Draw.io

Draw.io est un outil en ligne gratuit qui permet de créer des diagrammes, des schémas, des organigrammes, des cartes mentales, et d'autres types de graphiques visuels. Il est souvent utilisé pour concevoir des diagrammes de flux, des architectures réseau, des diagrammes UML, ou des organigrammes de processus.



FIGURE 3.10 – Logo Draw.io

3.3 Technologies utilisées :

MongoDB

MongoDB est un programme de base de données orienté document, multiplateforme et accessible via le code source. Classé comme produit de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas optionnels.



FIGURE 3.11 – Logo MongoDB

Bootstrap

Bootstrap est un framework front end open source qui facilite la création d'interfaces web réactives et esthétiquement attrayantes en fournissant des composants et des styles CSS pré-conçus.



FIGURE 3.12 – Bootstrap

React

React est une bibliothèque. Il vous permet d'assembler des composants, mais n'est pas prescriptif en ce qui concerne le routage ou le chargement de données.

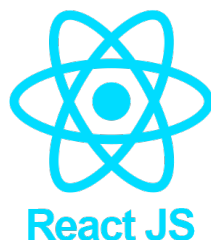


FIGURE 3.13 – Logo react

NodeJS

Node.js est un environnement d'exécution JavaScript gratuit, open source et multi-plateforme qui permet aux développeurs de créer des serveurs, des applications Web, des outils de ligne de commande et des scripts.



FIGURE 3.14 – Logo nodejs

ExpressJs

Express est un framework d'application Web Node.js minimal et flexible qui fournit un ensemble robuste de fonctionnalités pour les applications Web et mobiles.



FIGURE 3.15 – Logo express

JavaScript

est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur¹. C'est un langage orienté objet à prototype.



FIGURE 3.16 – Logo JavaScript

HTML

est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte.



FIGURE 3.17 – Logo HTML

CSS

est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.



FIGURE 3.18 – Logo CSS

3.4 Tâches Réalisées

Dans cette partie, nous allons illustrer les tâches réalisées à l'aide de quelques captures d'écran représentatives.

3.4.1 Interfaces Admin :

L'interface 3.17 permet à l'administrateur de visualiser les demandes de création de comptes soumises par les enseignants. Il peut alors soit approuver la demande, ce qui entraîne la création du compte et l'envoi automatique des informations de connexion par email, soit refuser la demande. Cette fonctionnalité assure un contrôle rigoureux des inscriptions.

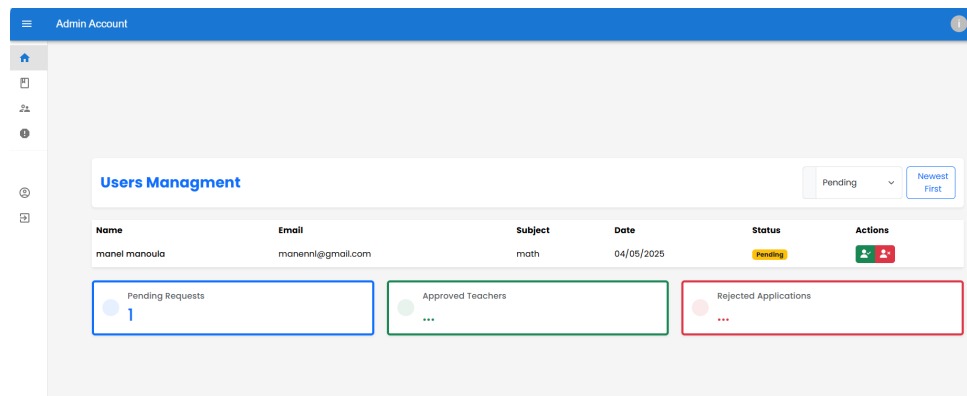


FIGURE 3.19 – Interface de gestion des comptes enseignants.

Cette interface offre à l'administrateur la possibilité d'ajouter de nouveaux enseignants dans le système. Deux options sont proposées : la saisie manuelle via un formulaire détaillé, ou l'importation en masse à l'aide d'un fichier Excel structuré. Puis les Afficher dans un tableau. Cette flexibilité permet de gagner du temps lors de l'ajout de plusieurs enseignants tout en garantissant la fiabilité des données saisies.

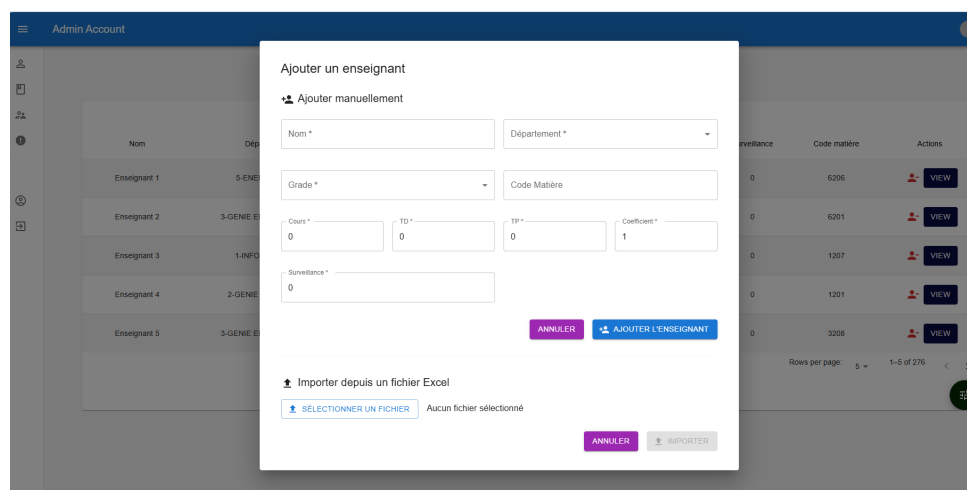


FIGURE 3.20 – Interface D'ajout des Enseignants

Cette interface de planification permet à l'utilisateur de gérer et d'afficher les résultats d'affectation des tâches de surveillance. Elle comprend un tableau dynamique qui présente les détails de l'affectation, tels que les enseignants, les horaires, et les salles.

Deux boutons sont présents pour faciliter l'interaction :

Affecter : Ce bouton permet d'attribuer des tâches spécifiques aux enseignants en fonction de la planification.

Exporter en PDF : Ce bouton permet d'exporter la planification des affectations sous forme de fichier PDF pour une consultation ou une distribution ultérieure.

Date	Séance	Salle	Enseignant 1	Enseignant 2	Action
jeudi 20 mars 2025	sl	H10	Enseignant 270	Enseignant 275	<button>Modifier</button>
jeudi 20 mars 2025	sl	H12	Enseignant 247	Enseignant 261	<button>Modifier</button>
jeudi 20 mars 2025	sl	I01	Enseignant 238	Enseignant 244	<button>Modifier</button>
jeudi 20 mars 2025	sl	I02	Enseignant 239	Enseignant 246	<button>Modifier</button>
jeudi 20 mars 2025	sl	K08	Enseignant 233	Enseignant 271	<button>Modifier</button>

« « 1 2 3 4 5 6 7 8 9 10 » »

Affecter Vider Exporter Planning

FIGURE 3.21 – Planification Des Surveillances

3.4.2 Interfaces Enseignant :

Cette interface de profil enseignant affiche les informations personnelles de l'enseignant. Elle offre également la possibilité de modifier ces informations ainsi que de changer son mot de passe pour assurer la sécurité et la mise à jour des données. Ainsi qu'un bouton pour exporter le Planning de surveillance des examens.

Ichraak dhahrii
 Genie Logiciel Teacher

Logout

Overview
 Profile
 Exams Affectation

Profile Information
Edit

Personal Information

- Full Name
Ichraak dhahrii
- Email Address
ichraak.dhahrii@gmail.com
- Phone Number
26238106

Professional Details

- Subject
Genie Logiciel
- Member Since
04/05/2025

Account Security

Password
 Last changed: 30 days ago
 Change Password

FIGURE 3.22 – Interface de Profile de l'enseignant

3.5 Conclusion

En somme, ce chapitre a posé les fondations indispensables à la conception de notre projet de planning des surveillances des examens. Nous y avons détaillé l'environnement matériel et logiciel, présenté les technologies clés, illustré avec des captures d'interfaces

et souligné l'importance des tests pour assurer la qualité du projet. Ces aspects sont essentiels pour en garantir le succès.

Conclusion Générale

Ce projet de développement d'un système de gestion des surveillances d'examens s'inscrit dans une volonté d'amélioration continue des processus administratifs au sein des établissements d'enseignement supérieur, en particulier à l'ISSAT Sousse. Grâce à l'automatisation intelligente et à une interface conviviale, l'application développée répond efficacement aux problématiques liées à la planification manuelle des surveillances.

L'analyse du contexte, la conception technique de la solution, ainsi que l'approche méthodologique suivie ont permis d'atteindre les objectifs fixés, tant sur le plan fonctionnel que sur le plan qualitatif. Les résultats obtenus témoignent d'une réduction significative du temps de planification, d'un meilleur équilibre des charges de travail, et d'une satisfaction notable des utilisateurs finaux.

Ce projet a également été une opportunité enrichissante de mettre en œuvre et de renforcer des compétences clés, allant de la modélisation des besoins jusqu'au déploiement d'une solution web complète. Il a aussi permis de vivre une expérience de gestion de projet agile, proche des réalités professionnelles.

Enfin, plusieurs pistes d'amélioration ont été identifiées pour faire évoluer le système vers une version encore plus performante, personnalisée et intégrée. Ces perspectives ouvrent la voie à une généralisation possible de la solution à d'autres établissements ou contextes similaires.

Ce travail représente ainsi une contribution concrète à la digitalisation des processus académiques et marque une étape importante dans le parcours de formation en ingénierie logicielle.

Bibliographie

- [1] *ISSAT*. <https://issatso.rnu.tn/>. 5 mai 2025.
- [2] *Mongodb*. <https://www.mongodb.com/fr-fr>. 5 mai 2025.
- [3] *CSS3*. <https://developer.mozilla.org/fr/docs/Web/CSS>. 5 mai 2025.
- [4] *scrum*. <https://www.oqlf.gouv.qc.ca/>. 5 juill. 2024.