



UNIVERSITÉ
DE LORRAINE



nancy

Charlemagne
Informatique

OUTIL D'AIDE À LA GESTION POUR LA DIRECTION DES ÉTUDES

2024/2025

TUTEURS : Philip DOSCH et Slim OUNI

HAROUNA Laeticia – HOLCVART Lothaire – MARI Farid –
RAGOT Yogan
AILI

Introduction	4
Objet du document	4
Notre démarche vise à fournir un document qui soit utile autant pour un lecteur technique souhaitant reprendre ou maintenir l'application, que pour un lecteur non technique souhaitant comprendre l'organisation générale du projet et sa mise en œuvre concrète.	4
Contexte du projet	5
Présentation du projet	6
Interface enseignant	6
Interface gestionnaire	6
Gestion des utilisateurs	7
Présentation de l'équipe et rôle de chacun	8
Planning de déroulement du projet	9
Déploiement :	12
Analyse :	13
Fonctionnalités :	13
Gestion des utilisateurs :	13
Gestion des enseignants et des préférences :	13
Gestion des groupes et des cours :	13
Planification et affectations :	13
Validation et contrôle :	14
Interfaces et expérience utilisateur :	14
Historisation des données :	14
Diagrammes	14
Diagrammes de séquence	15
Connexion :	15
Fiche de vœux :	16
Fiche de service :	17
Fiche répartition :	18
Gestion validation fiches :	19
Planning détaillé :	20
Configuration planning détaillé :	21
Evolutions par rapport à l'étude préalable de décembre	22
Réalisation	23
Partie enseignants	24
Partie gestionnaire	24
Mapping fonctionnalité souhaité / réalisé	25
Conclusion	27
Annexes	29
Handsontable et son Importance dans le Projet	29
Fonctionnement du planning détaillé	29
1. Configuration du planning	30
2. Construction des En-têtes du Tableau	30
3. Construction du Tableau de Données	30
Structure des Données	30

4. Construction des totaux	31
5. Gestion des modifications	32
6. La Fonction saveAllData	32
7. Optimisation avec mergeRepartitions	33
8. Gestion du scroll et de la modification de certaines lignes	33
Conclusion	34
Fiche Répartition	35
Validation de fiche	37

Introduction

Objet du document

Ce document a pour objectif de détailler l'ensemble du travail réalisé dans le cadre de notre projet tutoré, effectué durant la troisième année du BUT Informatique, parcours Réalisation d'Applications, option Ingénierie Logicielle, à l'IUT Nancy Charlemagne.

À travers ce rapport, nous présentons de manière complète le site que nous avons conçu et développé. Il retrace toutes les étapes du projet, de l'analyse du besoin initial à la mise en œuvre technique finale, en passant par la conception, le développement et les phases de tests. Ce document constitue également une trace écrite de l'ensemble des décisions prises tout au long de la réalisation, qu'il s'agisse de choix techniques, méthodologiques ou organisationnels.

Notre démarche vise à fournir un document qui soit utile autant pour un lecteur technique souhaitant reprendre ou maintenir l'application, que pour un lecteur non technique souhaitant comprendre l'organisation générale du projet et sa mise en œuvre concrète.

Contexte du projet

Le projet que nous avons réalisé s'inscrit dans un besoin concret exprimé au sein du département Informatique de l'IUT Nancy Charlemagne. Chaque année, le département doit planifier les emplois du temps des enseignants et des étudiants, ce qui demande beaucoup de travail. Pour réussir à construire ces emplois du temps, il faut prendre en compte plusieurs informations, comme les contraintes des enseignants, leurs préférences ou encore les besoins spécifiques de chaque module.

Jusqu'à présent, la gestion de toutes ces données se faisait principalement avec des fichiers Excel, des documents papier et des échanges de mails. Même si ces méthodes fonctionnaient, elles étaient longues et compliquées à gérer. Le fait d'avoir des informations dispersées sur plusieurs supports augmentait le risque d'erreurs ou d'oublis, et il devenait difficile pour le gestionnaire de retrouver facilement toutes les données dont ils avaient besoin.

En plus de cela, cette manière de faire prenait beaucoup de temps, notamment quand il fallait faire des modifications ou vérifier que toutes les contraintes étaient bien respectées. Pour toutes ces raisons, le département a souhaité changer sa façon de fonctionner en mettant en place une solution plus simple, plus rapide et plus fiable.

C'est dans ce cadre que notre projet a vu le jour. L'objectif était de créer un **outil** qui centralise toutes les informations nécessaires pour préparer l'année universitaire. Grâce à cette application, les enseignants peuvent remplir eux-mêmes leurs contraintes et leurs vœux, et les gestionnaires peuvent consulter, modifier, valider et utiliser ces données plus facilement.

Notre projet a donc pour but d'aider à gagner du temps, d'éviter les erreurs et de rendre l'organisation des emplois du temps plus simple pour tout le monde.

Présentation du projet

Le projet que nous avons réalisé a pour but de faciliter la gestion des contraintes et des vœux des enseignants du département Informatique. L'idée était de remplacer les fichiers Excel et les documents papiers par un site web où tout serait centralisé et organisé.

Le site que nous avons développé est divisé en deux parties, selon le type d'utilisateur : **l'interface enseignant** et **l'interface gestionnaire**.

Interface enseignant

L'**interface enseignant** permet aux enseignants de se connecter à leur espace personnel et de remplir trois types de fiches :

- **La fiche de vœux**, où ils peuvent indiquer leurs impossibilités, leurs préférences horaires, et préciser s'ils sont prêts à donner cours le samedi.
- **La fiche ressource**, réservée aux enseignants responsables de modules, pour préciser leurs besoins spécifiques (type de salle, matériel nécessaire, organisation de DS etc...).
- **La fiche prévisionnelle**, qui leur permet d'exprimer leurs souhaits pour l'année suivante en choisissant les modules sur lesquels ils aimeraient intervenir.

En plus de remplir leurs fiches, les enseignants peuvent :

- **Consulter le nombre d'heures** qu'ils réalisent à partir des vœux effectués dans la fiche prévisionnelle
- **Voir directement sur leur page d'accueil** leurs vœux ainsi que leurs affectations pour l'année universitaire.

L'interface enseignant est conçue pour être simple et intuitive, afin que chacun puisse saisir ses informations rapidement et de manière autonome.

Interface gestionnaire

L'**interface gestionnaire** est différente. Elle permet de voir l'ensemble des fiches remplies par tous les enseignants. Les gestionnaires peuvent :

- **Consulter** les contraintes et les vœux de chaque enseignant.
- **Valider, remplir et modifier** des fiches.
- **S'assurer** que toutes les informations utiles à l'organisation de l'année universitaire sont bien saisies

- **Remplir le tableau des affectations** en associant les enseignants aux différents modules, groupes
- **Préparer le planning détaillé** des cours pour l'année à venir.

Le rôle des gestionnaires est donc plus global. Contrairement aux enseignants, qui ne voient que leurs propres fiches, les gestionnaires ont accès à l'ensemble des données. Cela leur permet d'avoir une vue d'ensemble et de construire les emplois du temps de manière cohérente et équilibrée, en tenant compte des contraintes et des souhaits de chacun.

Gestion des utilisateurs

En plus de la gestion des fiches, l'interface gestionnaire permet aussi d'administrer les utilisateurs. Il peut :

- **Créer de nouveaux utilisateurs**, qu'il s'agisse d'enseignants ou de gestionnaires, en leur attribuant les bons rôles dès la création du compte
- **Modifier** les informations d'un utilisateur existant si besoin
- **Supprimer** des comptes lorsque cela est nécessaire.

Lors de la création d'un enseignant, le gestionnaire peut également :

- **Limiter le nombre de cases** que l'enseignant pourra cocher dans sa fiche de contraintes
- **Désigner les responsables de module**, c'est-à-dire indiquer quels enseignants seront chargés d'une ressource pour l'année.

Toutes ces fonctionnalités ont été ajoutées pour faciliter le travail des gestionnaires et rendre l'application plus complète et plus adaptée aux besoins du département.

Pour conclure, notre application a été pensée pour être simple et intuitive, aussi bien pour les enseignants que pour les gestionnaires. Elle permet de centraliser toutes les informations importantes, de gagner du temps, d'éviter les erreurs et de rendre la préparation des emplois du temps beaucoup plus fluide.

Présentation de l'équipe et rôle de chacun

Notre équipe était composée de quatre étudiants, chacun ayant un rôle bien défini dans le projet. Nous avons pris soin de répartir les tâches en fonction des compétences et des préférences de chacun, ce qui nous a permis d'avancer efficacement tout au long du semestre et d'assurer une bonne cohésion de groupe.

Laeticia HAROUNA a travaillé sur la mise en place de mot de passe sécurisé, elle a aussi participé au développement de la **fiche de vœux**, en créant l'interface. Elle s'est aussi occupée de la partie **gestion des fiches** pour l'interface gestionnaire avec les fonctionnalités de modifier, valider et remplir une fiche. En plus de ses missions principales, elle a contribué à la **création de la base de données**, en aidant à la structuration des différentes tables nécessaires au projet.

Lothaire HOLCVART a pris en charge la réalisation du **planning des affectations** et de la **fiche prévisionnelle**. Son travail a permis aux enseignants de choisir les modules qu'ils souhaitent enseigner et en précisant leurs préférences en termes de volume horaire. Il a également travaillé sur l'affichage des heures par semaine. Ce travail a été particulièrement important pour le projet, car il facilite à la fois la planification côté enseignant et l'organisation générale du planning par les gestionnaires.

Farid MARI s'est concentré sur la création de la **fiche de ressources**, destinée aux enseignants responsables de modules. Il a travaillé sur le style des interfaces et c'est lui qui a fait la page d'accueil des enseignants.

Yogan RAGOT s'est chargé de l'**interface gestionnaire**, et plus particulièrement du **planning détaillé**, qui constitue le point de départ de toute l'organisation du projet. Grâce à cette fonctionnalité, les gestionnaires peuvent préparer de manière précise l'affectation des enseignants sur les modules et structurer l'année universitaire de façon cohérente. Il a également développé les fonctionnalités permettant aux gestionnaires de **créer, modifier ou supprimer** des comptes. Il a également participé à la **conception de la base de données**. Enfin, il a travaillé sur certaines fonctionnalités de l'**interface enseignant**, notamment sur la fiche de vœux en ajoutant la limitation des contraintes.

Même si chacun avait des missions précises, nous avons régulièrement travaillé ensemble pour nous entraider. Nous avons échangé nos idées et testé les différentes parties du site au fur et à mesure pour vérifier que tout fonctionnait correctement. Cette collaboration nous a permis de corriger les éventuels problèmes rapidement et d'assurer une **cohérence générale** entre les différentes interfaces.

Planning de déroulement du projet

Le projet a été réalisé en **sept itérations** successives. Chaque itération a permis de faire évoluer l'application progressivement, en suivant une méthode de travail agile et collaborative. Le tableau suivant détaille pour chaque itération les fonctionnalités développées ainsi que la répartition du travail au sein de l'équipe.

Itération	Travail réalisé	Membres concernés
Itération 1 – Conception	<ul style="list-style-type: none"> - Définition des besoins. - Conception du modèle relationnel de la base de données. - Réalisation des premières maquettes. - Développement du squelette des interfaces enseignant et gestionnaire. - Première version des fiches de vœux et de ressources (visuel uniquement). 	<p>Laeticia HAROUNA : connexion, début de l'interface enseignant.</p> <p>Lothaire HOLCVART : maquettes, début fiche de vœux.</p> <p>Farid MARI : maquettes, début fiche de ressources.</p> <p>Yogan RAGOT : début interface gestionnaire, base de données.</p>
Itération 2 – Mise en place des fonctionnalités principales	<ul style="list-style-type: none"> - Connexion sécurisée et création de comptes utilisateurs (avec envoi de lien de création de mot de passe). - Développement de la fiche de vœux et de la fiche prévisionnelle. - Premier visuel du planning détaillé gestionnaire. 	<p>Laeticia HAROUNA : gestion des comptes utilisateurs et fiche de vœux</p> <p>Lothaire HOLCVART : fiche prévisionnelle</p> <p>Farid MARI : fiche de ressources.</p> <p>Yogan RAGOT : début du planning détaillé.</p>

Itération 3 – Stabilisation exportations	<ul style="list-style-type: none"> - Correction de bugs sur les fiches. - Mise en place de l'exportation en PDF de la fiche de vœux. - Développement de la fiche de répartition des enseignants et de l'affectation aux groupes. - Optimisation du planning détaillé. 	<p>Laeticia HAROUNA : export PDF fiche de vœux.</p> <p>Lothaire HOLCVART : fiche de répartition.</p> <p>Farid MARI : corrections fiche ressource.</p> <p>Yogan RAGOT : optimisation planning détaillé.</p>
Itération 4 – Finalisation des fonctionnalités principales	<ul style="list-style-type: none"> - Correction et stabilisation des fiches. - Finalisation du planning détaillé (ajout de la gestion dynamique des heures, affichage des vacances, dépassement d'heures signalé en rouge). - Limitation du nombre de contraintes saisies par les enseignants. - Export PDF amélioration. 	<p>Laeticia HAROUNA : limitation sur fiche de contraintes, export PDF</p> <p>Lothaire HOLCVART : corrections fiche prévisionnelle.</p> <p>Farid MARI : refonte graphique du site.</p> <p>Yogan RAGOT : finalisation planning détaillé, corrections générales.</p>
Itération 5 – Corrections et ajout de fonctionnalités complémentaires	<ul style="list-style-type: none"> - Ajout de l'affichage du nombre d'heures réalisées par chaque enseignant. - Ajout d'une page de consultation des vœux et affectations sur l'accueil enseignant. - Validation des fiches par le gestionnaire. - Amélioration de l'ergonomie générale. 	<p>Laeticia HAROUNA : Système de validation des fiches</p> <p>Lothaire HOLCVART : affichage horaire par semaine.</p> <p>Farid MARI : accueil enseignant (avec les calculs des heures) + profil enseignant</p> <p>Yogan RAGOT : Planning détaillé + page</p>

		cours (modification des d'une ressource ou ajout d'une nouvelle ressource)
Itération 6 – Correctifs	<ul style="list-style-type: none"> - Corrections de l'ensemble des fonctionnalités de base 	<p>Tous les membres : tests et corrections.</p> <p>Laeticia HAROUNA : ajout zone de texte fiche de voeux + correction pdf</p> <p>Lothaire HOLCVART : style planning de répartition + vue heure par semaine</p> <p>Farid MARI : blackboard des ressources + correction bug</p> <p>Yogan RAGOT : planning détaillé + calcul des heures totales onglet Gestion Cours</p>
Itération 7 – Livraison finale	<ul style="list-style-type: none"> - Validation, Modification, Remplissage des fiches - Refonte du site - Derniers améliorations des fiches - Historisation - Présentation finale et soutenance. - Déploiement 	<p>Tous les membres : finalisation.</p> <p>Laeticia HAROUNA : implémentation de la gestion des fiches (modification, remplissage, validation)</p> <p>Lothaire HOLCVART : finalisation fiche prévisionnelle et répartition</p> <p>Farid MARI : Refonte site</p>

		Yogan RAGOT : Historisation et dernier correctif planning détaillé + déploiement + implémentation changement de mot de passe
--	--	---

Déploiement :

<https://webetu.iutnc.univ-lorraine.fr/www/ragot31u/ProjetTutore/>

Gestionnaire compte :

gestionnairedosch@mail.com

mdp : azertyuiop

Vous pourrez par la suite créer un compte enseignant.

Analyse

Dans cette partie nous revenons sur le découpage fonctionnel du projet, nous présenterons également quelques modèles UML afin de mieux comprendre certaines interactions au sein de l'outil. De plus, nous identifierons les écarts ou ajustements qui ont été réalisés par rapport à l'étude préalable de décembre, en expliquant les raisons de ces évolutions et leur impact sur la solution finale.

Fonctionnalités :

Gestion des utilisateurs :

- Création de comptes utilisateurs (enseignants et gestionnaires) avec enregistrement en base de données.
- Envoi automatique d'un lien pour définir le mot de passe.
- Modification et suppression des comptes utilisateurs.

Gestion des enseignants et des préférences :

- Fiche de vœux permettant aux enseignants de renseigner préférences horaires et contraintes.
- Limitation des choix des préférences de la fiche contrainte à un nombre spécifique, personnalisable par le gestionnaire.
- Exportation de fiches au format PDF.
- Fiche prévisionnelle permettant aux enseignants de renseigner les modules qu'ils souhaitent enseigner, avec calcul automatique du total des heures.
- Fiche de ressources permettant de spécifier les besoins matériels ou autres (types de salles, équipements spécifiques).

Gestion des groupes et des cours :

- Fiche de répartition à affectation de plusieurs enseignants à un même cours avec un affichage ergonomique vertical.
- Création, modification et suppression des cours par le gestionnaire.
- Calcul automatique des heures totales affectées par cours et par enseignant.

Planification et affectations :

- Planning détaillé ergonomique de type tableur permettant d'organiser les horaires des cours et les affectations d'enseignants.
- Totaux horaires dans le planning détaillé.
- Configuration avancée des dates spéciales (ateliers, projets, stages, jury, vacances...) avec personnalisation des couleurs.
- Visualisation claire des affectations et modifications dynamiques possibles (scroll souris par sélection).

Validation et contrôle :

- Validation définitive des fiches (vœux, ressources, répartition) par le gestionnaire avec verrouillage immédiat empêchant toute modification par l'enseignant.
- Affichage automatique des heures totales affectées sur l'accueil enseignant après validation.

Interfaces et expérience utilisateur :

- Interface gestionnaire intuitive avec menu clair (gestion des utilisateurs, des fiches, des ressources, du planning détaillé).
- Interface enseignant dédiée avec affichage personnalisé des affectations, heures totales et fiches.
- Modernisation globale des interfaces pour une navigation intuitive, fluide et ergonomique.

Historisation des données :

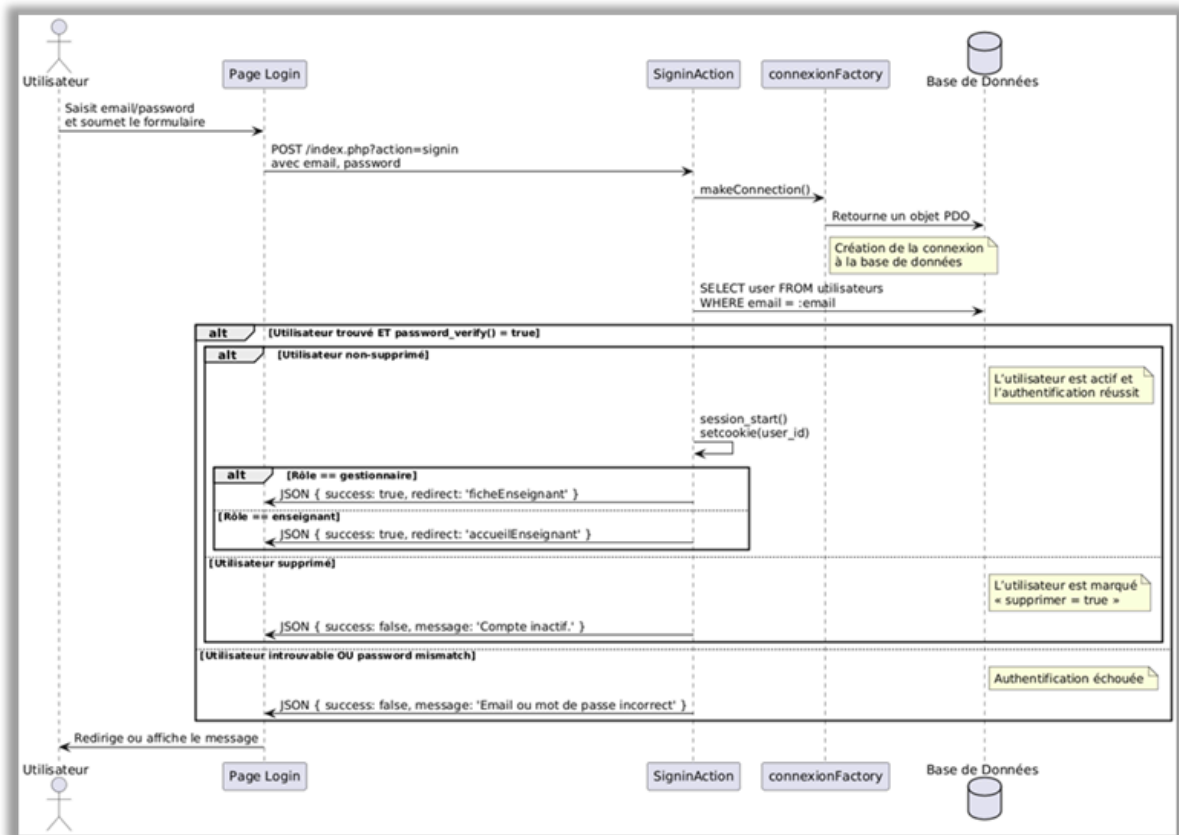
- Possibilité d'historiser une année universitaire complète, incluant les fiches de vœux, fiches de répartition, fiches prévisionnelles, et affectations.
- Permet de garder une trace des affectations passées et de consulter les données d'années antérieures.
- Fonctionnalité essentielle pour la traçabilité, l'analyse des répartitions précédentes, ou encore la préparation d'une nouvelle année à partir de l'existant.

Diagrammes

Passons maintenant à l'analyse UML, à travers une sélection de diagrammes représentatifs du fonctionnement global de l'application.

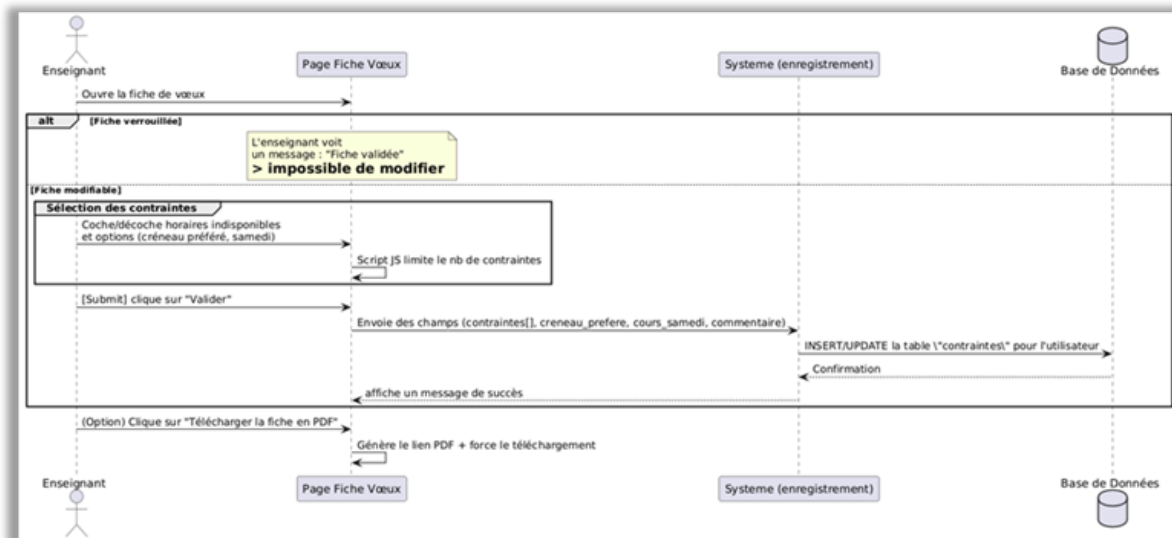
Diagrammes de séquence

Connexion :



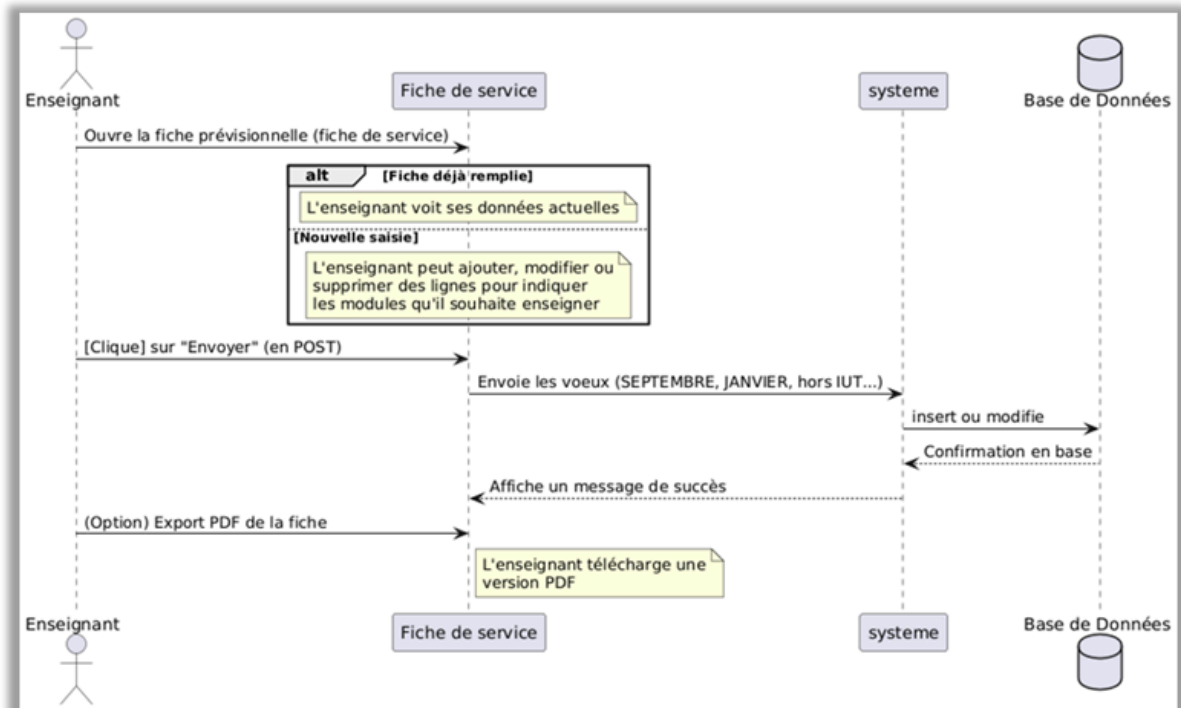
- L'utilisateur saisit ses identifiants sur la page de login.
- SigninAction demande une connexion en passant par connexionFactory qui lui fournit un objet PDO pointant vers la BDD.
- SigninAction exécute une requête et :
 - o s'il trouve un utilisateur correspondant et que password_verify() est valide, la session est créée et on détermine la redirection en fonction du rôle.
 - o si le compte est marqué supprimé on renvoie un message d'erreur.
 - o et sinon on renvoie aussi un JSON d'erreur.
- La page reçoit ce JSON et redirige l'utilisateur vers le bon espace ou affiche un message d'erreur

Fiche de vœux :



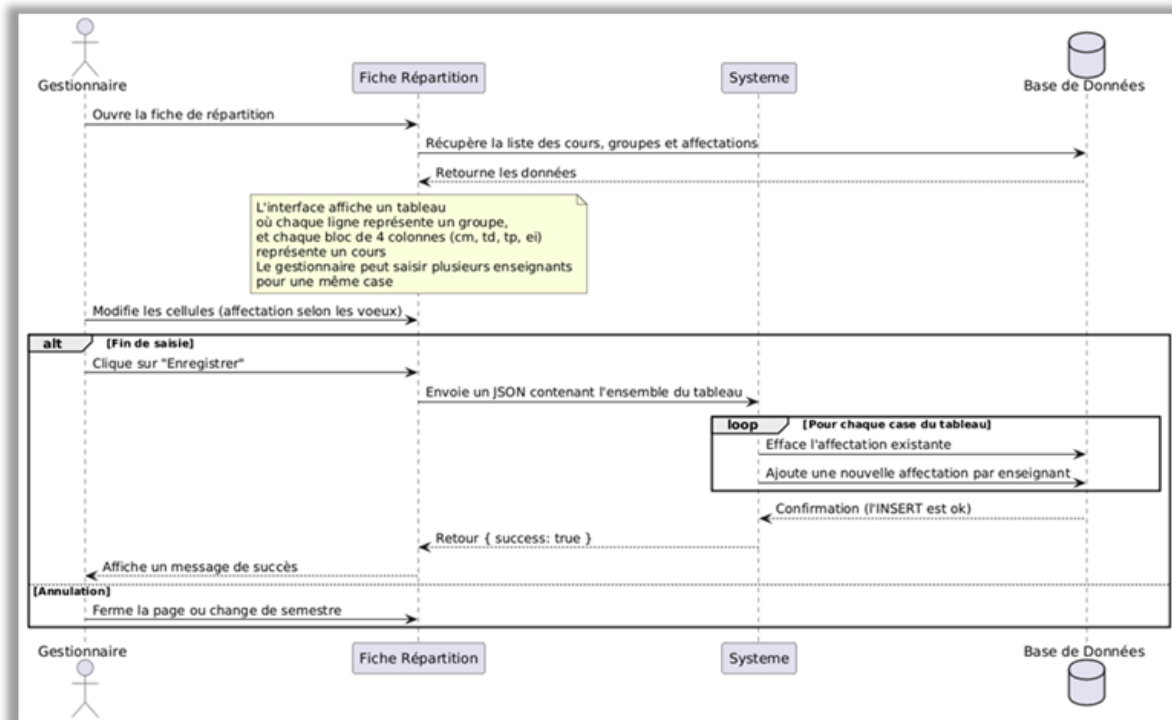
- Enseignant ouvre la page fiche vœux .
- Si la fiche est déjà verrouillée, l'enseignant voit un message d'alerte comme quoi aucune modification n'est possible.
- Sinon, il coche/décoche les cases pour indiquer ses indisponibilités et il remplit les options. (Le code JavaScript limite automatiquement le nombre de cases cochées.)
- Quand l'enseignant clique sur Valider, la page appelle le système en lui envoyant les données.
- Le système met à jour la base de données.
- Une fois l'opération réussie, la page confirme la prise en compte des informations.
- L'enseignant peut enfin télécharger la fiche en PDF

Fiche de service :



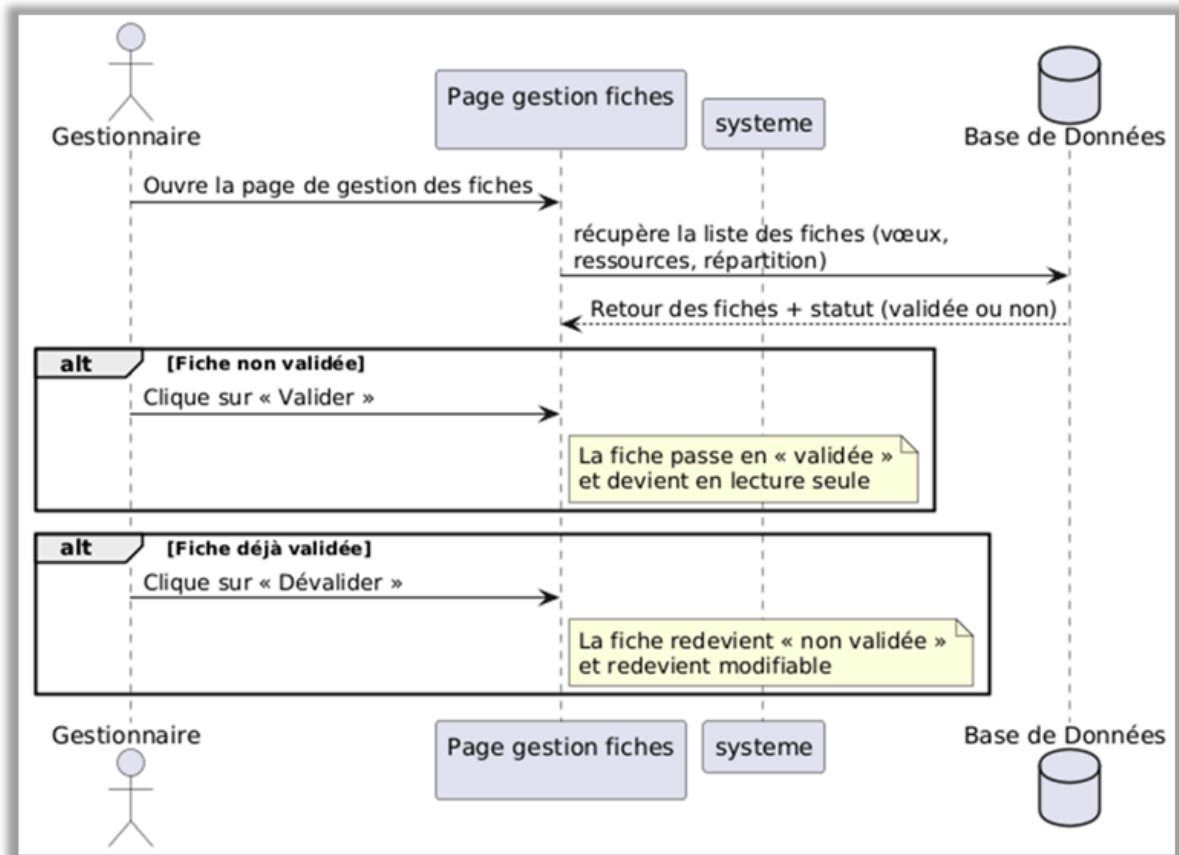
- Enseignant ouvre la fiche prévisionnelle
- S'il existe déjà des vœux enregistrés, l'enseignant voit directement ses données actuelles. Sinon, il peut ajouter/éditer des lignes pour préciser certains éléments
- Quand l'utilisateur clique sur Envoyer, la page envoie les données en POST vers le système qui réalise l'insertion ou la mise à jour en bd.
- Une fois terminé le serveur renvoie une confirmation.
- L'enseignant peut alors exporter la fiche au format PDF (choix optionnel).

Fiche répartition :



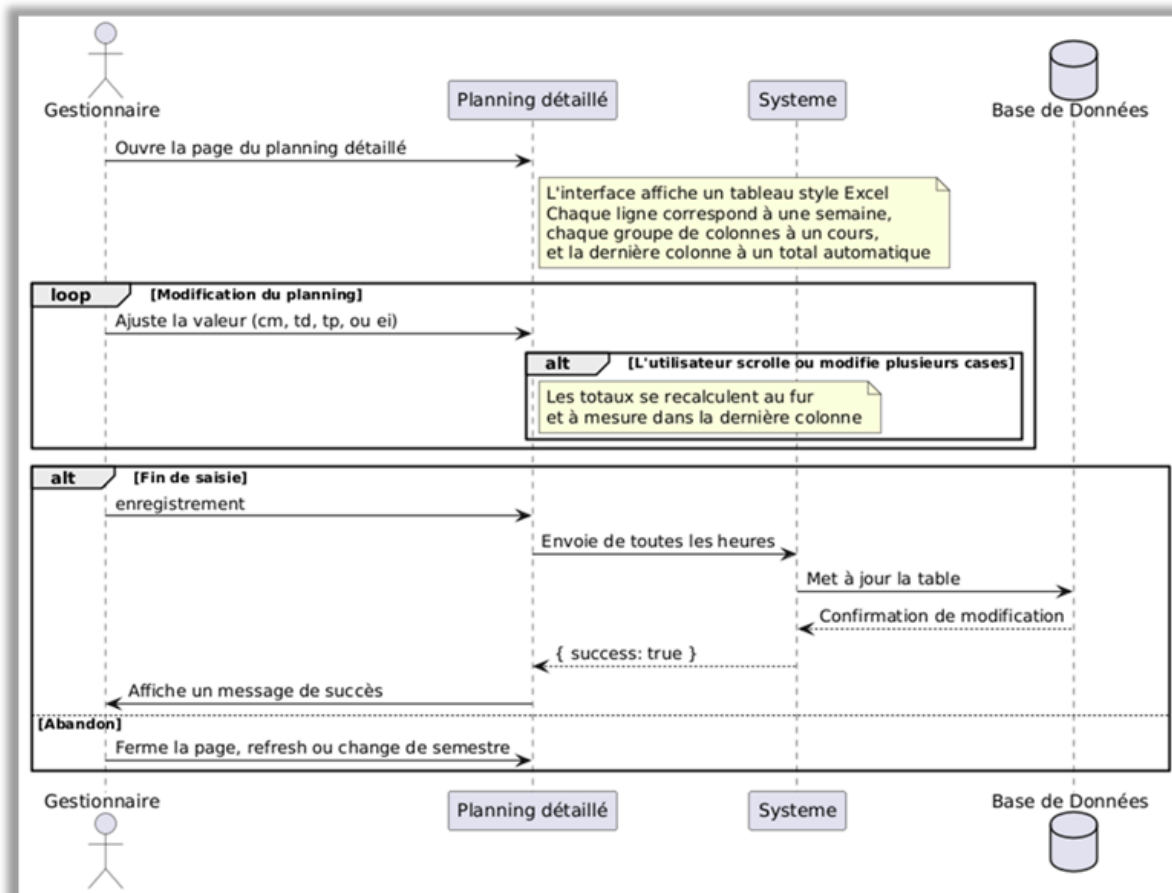
- Le gestionnaire accède à la fiche répartition qui récupère depuis la base de données la liste des cours, groupes et affectations déjà existantes.
- Le tableau interactif s'affiche avec la possibilité de saisir plusieurs enseignants dans la même case.
- Quand le gestionnaire clique sur Enregistrer, l'interface envoie le JSON global au système.
- Le système supprime d'abord les anciennes affectations puis insère une nouvelle affectation pour chacun des enseignants saisis dans la cellule.
- Une fois la BD mise à jour, un renvoie success: true est fait.
- L'interface affiche un message de réussite à l'utilisateur.

Gestion validation fiches :



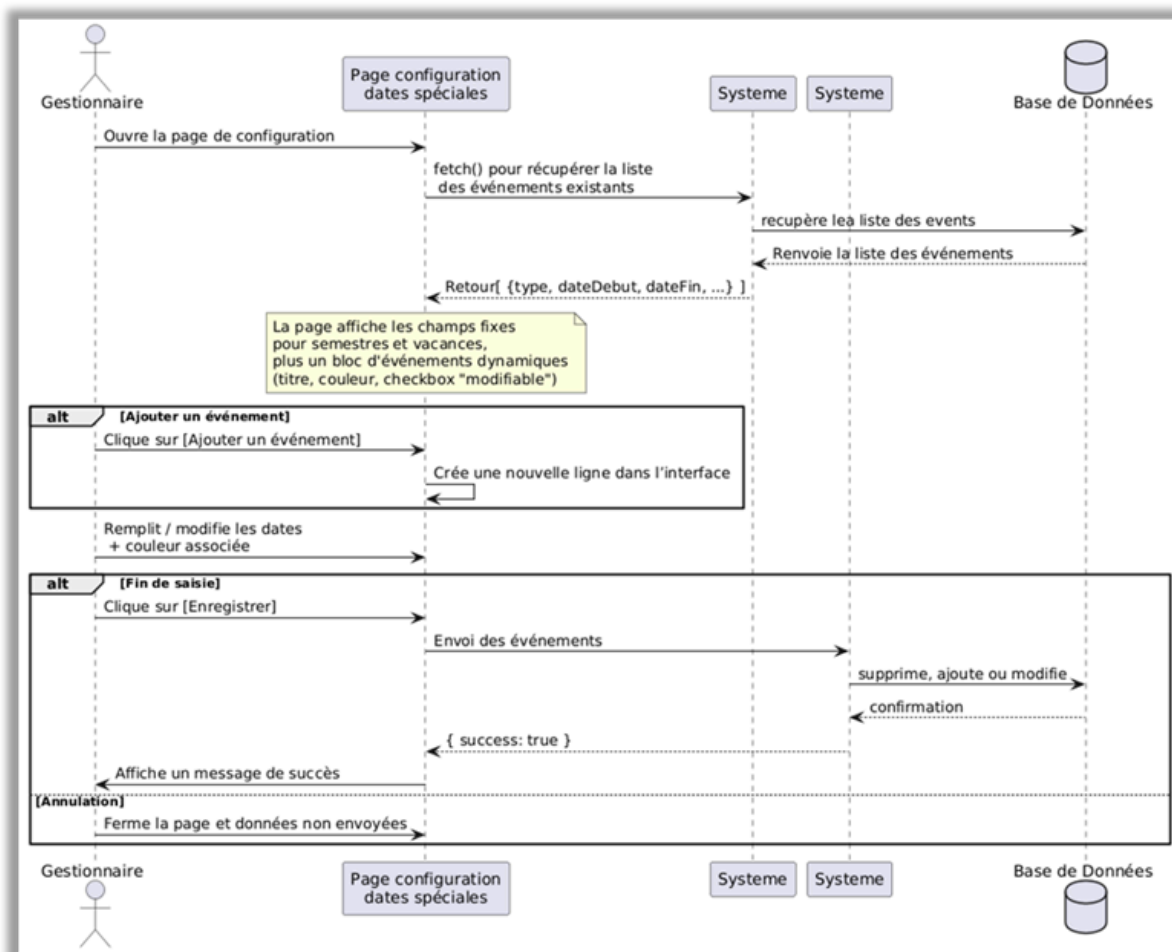
- Le gestionnaire ouvre la page de gestion des fiches.
- Cette page récupère d'abord la liste des différentes fiches ainsi que leur statut (validée ou non) depuis la base de données.
 - o Si la fiche n'était pas encore validée, le gestionnaire clique sur Valider, ce qui envoie une requête au système pour mettre à jour le statut à validée.
 - o Si la fiche était déjà validée, le bouton Dévalider réalise l'opération inverse en rendant la fiche à nouveau modifiable par l'enseignant.
- Chaque action renvoie un JSON de confirmation, et la page affiche un message de succès au gestionnaire.

Planning détaillé :



- Le gestionnaire ouvre le planning détaillé qui se présente sous forme d'un tableau interactif type excel.
- Chaque semaine est représentée en ligne, et chaque cours s'étend sur plusieurs colonnes. La dernière colonne calcule automatiquement le total pour la semaine.
- Le gestionnaire modifie plusieurs valeurs. Le recalcul des totaux se fait instantanément côté interface.
- Quand le gestionnaire clique sur Enregistrer, l'interface envoie un JSON qui système qui lui met à jour la base de données.
- Une confirmation de succès est renvoyée à l'interface, qui l'affiche à l'utilisateur

Configuration planning détaillé :



- Le gestionnaire ouvre la page de configuration où sont présentés divers champs (dates de semestres, vacances, etc.) et un bouton « Ajouter un événement » qui permet de définir des plages de dates personnalisées (ateliers, jury, stages...).
- La page fait appel au système pour récupérer la configuration existante depuis la base et l'affiche en formulaires et champs de couleur.
- Le gestionnaire peut ajouter ou supprimer des lignes, entrer un titre, choisir une couleur, et cocher ou non le champ « modifiable ».
- En cliquant sur « Enregistrer », la page envoie un JSON au serveur qui met à jour la table « configurationplanningdetaillé » en supprimant et recréant ou modifiant les événements.
- Après la réussite de l'opération un message de succès apparaît sur la page.

Evolutions par rapport à l'étude préalable de décembre

- **Planification enrichie :**
Une interface plutôt avancée a été intégrée pour configurer des événements spécifiques (vacances, stages, ateliers, jury, etc.), avec une **personnalisation des couleurs**. Cela permet d'avoir un **planning visuel et clair**, ce qui facilite la planification annuelle.
- **Gestion fine de la validation des fiches :**
Le **gestionnaire peut valider ou dévalider** une fiche enseignant (vœux, contraintes, ressources).
Une **fiche validée est verrouillée** et devient **non modifiable** par l'enseignant, garantissant la stabilité des données.
Une interface claire permet de **filtrer les fiches remplies ou non**, ce qui facilite le suivi global.
- **Ajout d'un profil enseignant :**
Chaque enseignant peut accéder à son **profil personnel**, où il est possible de **changer son mot de passe** directement depuis l'application.
- **Création différenciée des utilisateurs :**
La création de comptes distingue clairement les **enseignants** (avec statut, heures...) et les **gestionnaires** (plus simple avec moins de champs). Cela permet une personnalisation adaptée dès l'ajout.
- **Déploiement sur Webetu :**
Le projet a été **déployé sur les serveurs Webetu**, une solution d'hébergement universitaire non anticipée dans l'étude préalable, mais qui s'est avéré être une bonne solution.
- **Gestion des cours :**
Une **interface dédiée** permet désormais d'**ajouter, modifier ou supprimer** des cours. Ils sont liés à un semestre et comportent un **volume horaire précis (CM, TD, TP, EI)**, comme d'habitude.
- **Améliorations ergonomiques globales :**
Plusieurs interfaces ont été repensées pour être plus intuitives (scroll vertical, champs dynamiques, messages d'erreur, filtres...). Tous ces éléments renforcent l'expérience utilisateur, d'autant plus que c'était un des critères de réussite.

Réalisation

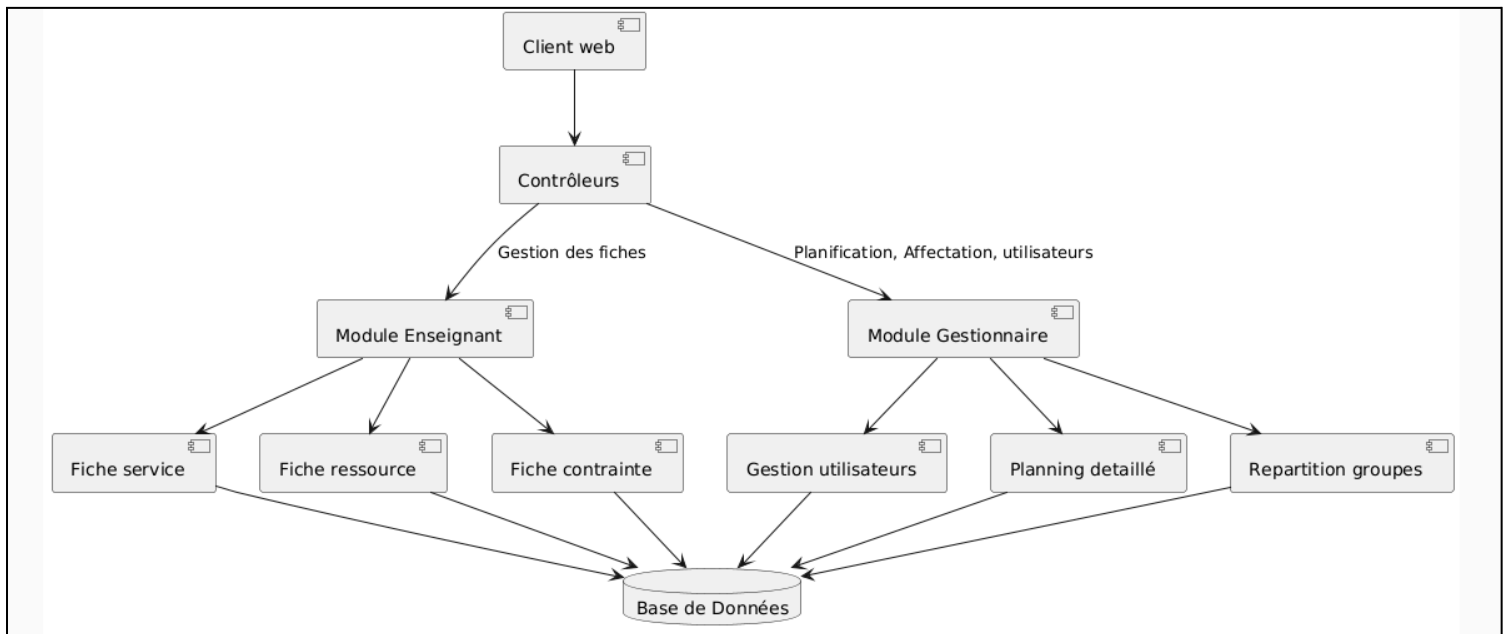
Le développement de notre projet s'est appuyé sur une approche itérative et agile, nous permettant de progresser étape par étape tout en ajustant notre travail en fonction des retours réguliers de nos tuteurs. Cela nous a permis d'intégrer progressivement les fonctionnalités, tout en affinant nos choix visuels et fonctionnels à partir des remarques des intervenants.

La première phase du projet a été consacrée à la création de maquettes et à l'élaboration d'un diagramme de classes détaillé, dans le cadre de notre étude préalable. Cette étape préparatoire a joué un rôle clé dans la définition de l'architecture générale du projet, dans l'identification des contraintes techniques, et dans l'anticipation des principaux défis. Elle nous a permis de mettre en place une feuille de route claire, garantissant une cohérence tout au long du développement et le respect des objectifs initiaux.

Le projet s'articule autour de deux volets complémentaires, chacun répondant à des besoins spécifiques : une interface destinée aux enseignants et une autre au gestionnaire. L'interface enseignant a été conçue pour leur permettre de remplir aisément les fiches nécessaires à l'élaboration des emplois du temps de l'année universitaire. Pensée pour être intuitive et ergonomique, elle facilite la saisie et la gestion des informations tout en assurant une présentation cohérente des données.

La partie dédiée au gestionnaire constitue le cœur du projet, en proposant des fonctionnalités avancées. Elle permet notamment de générer un planning précis pour chaque semestre en prenant en compte les différentes composantes du BUT Informatique. Elle facilite également l'affectation des groupes aux enseignants en fonction de leurs préférences, et centralise la gestion des comptes utilisateurs. En intégrant ces éléments clés, cette interface vise à optimiser la création des emplois du temps, apportant un gain de temps significatif ainsi qu'une meilleure organisation administrative.

On peut représenter ce projet sous cette architecture logicielle :



Nous allons maintenant voir les ajouts réalisés au cours des trois dernières itérations. Ces étapes ont permis de finaliser le projet en ajoutant les dernières fonctionnalités, en corrigeant les retours reçus et en améliorant l'interface et la performance de l'application.

Partie enseignants

Pour la partie dédiée aux enseignants, plusieurs améliorations ont été apportées au cours des dernières itérations. Sur la page d'accueil, nous avons intégré l'affichage des vœux d'enseignement ainsi que celui des affectations aux cours, afin de fournir une vue d'ensemble claire et actualisée. Dans la fiche prévisionnelle, une nouvelle vue permet désormais de consulter la répartition des heures de cours par semaine, en lien avec les vœux formulés. Un mécanisme de verrouillage a également été mis en place : une fois les trois fiches validées par le gestionnaire, elles deviennent non modifiables pour garantir la cohérence des données. Enfin, une fonctionnalité d'export en PDF des trois fiches a été ajoutée, permettant aux enseignants de conserver une version téléchargeable et imprimable de leurs documents.

Partie gestionnaire

Du côté gestionnaire, plusieurs fonctionnalités clés ont été développées pour faciliter la supervision et la gestion des emplois du temps. Une interface permet désormais de visualiser, modifier, valider ou compléter les fiches des enseignants. La configuration du planning détaillé a également été modifiée : il est désormais possible d'ajouter des

événements personnalisés, avec un nom, une couleur, et la possibilité de bloquer ou non la modification du planning pendant la période concernée.

Nous avons aussi intégré une interface de suivi permettant de vérifier en un coup d'œil si les fiches ressources ont été validées ou non. La fiche de répartition a été améliorée pour permettre l'affectation de plusieurs enseignants à un même groupe, offrant ainsi plus de flexibilité. Par ailleurs, des options d'export en CSV et XLSX ont été ajoutées, aussi bien pour le planning détaillé que pour la fiche de répartition.

Enfin, un système d'historisation a été mis en place pour permettre la création d'une nouvelle année universitaire tout en archivant les données de l'année précédente, assurant ainsi une continuité et une traçabilité dans la gestion des informations.

Mapping fonctionnalité souhaité / réalisé

Fonctionnalité demandé	Fonctionnalité réalisé	Etat
Saisie des préférences et contraintes horaires des enseignants	Création de la fiche contrainte	Réalisée
Saisie des besoins en ressources pédagogiques	Création de la fiche ressource	Réalisée
Saisie des vœux des enseignants	Création de la fiche prévisionnelle	Réalisée
Visualisations des heures de cours lors du choix des vœux	Création d'un onglet montrant les heures de cours par semaine des vœux choisies	Réalisée
Gestion des utilisateurs	Création, suppression et modification des utilisateurs	Réalisée
Gestion des fiches pour le gestionnaire	Validation, visualisation, modification et remplissage des fiches côté gestionnaire	Réalisée
Implémentation du planning détaillé	Création du planning détaillé et d'une page de configuration	Réalisée
Répartition des enseignants dans les groupes selon leur souhait	Création de la fiche répartition,	Réalisée
Création d'une nouvelle année	Historisation des données : création d'une nouvelle ou import d'une année précédente	Réalisée

Au cours du projet, nous avons été confrontés à certaines difficultés, notamment lors de l'ajout de nouvelles fonctionnalités. En effet, ces ajouts ont souvent nécessité de revenir sur du code déjà existant et d'apporter des modifications à la base de données. Ces ajustements ont parfois été complexes à gérer, car la structure initiale du projet n'avait pas toujours été pensée en anticipant l'intégration future de certaines fonctionnalités. Cette mauvaise anticipation nous a contraints à effectuer plusieurs refactorings, à adapter les modèles de données, et à revoir certains choix techniques pour garantir la cohérence et la stabilité de l'application.

Conclusion

Le projet que nous avons mené dans le cadre de notre troisième année de BUT Informatique à l'IUT Nancy-Charlemagne nous a permis de concevoir et de développer un outil centralisant les données essentielles à la planification des emplois du temps au sein du département informatique.

En partant des fichiers Excel et des documents papiers, nous avons proposé une solution numérique centralisée, fiable et ergonomique. Notre site web répond aux besoins concrets des enseignants et des gestionnaires en leur offrant une interface intuitive, des fonctionnalités avancées pour la saisie et la validation des fiches, ainsi qu'un planning détaillé configurable.

Tout au long du projet, nous avons mis en œuvre une approche itérative qui nous a permis d'améliorer continuellement l'application en fonction des retours et des besoins exprimés. Chaque membre de l'équipe a contribué activement à la réussite du projet, en développant des fonctionnalités clés tout en assurant la cohérence globale de l'ensemble.

Le projet est aujourd'hui pleinement fonctionnel et répond aux attentes initiales. Il dispose de toutes les fonctionnalités principales prévues dans l'étude préalable : saisie des fiches contraintes, ressources et prévisionnelles, validation des fiches, génération de plannings détaillés, historisation des données et gestion des utilisateurs.

Cependant, plusieurs pistes d'amélioration peuvent être envisagées pour de futurs développements :

- **Amélioration de l'ergonomie :**
Retravailler certaines interfaces pour rendre la navigation plus intuitive.
- **Mise en place d'un système de notifications :**
Ajouter l'envoi automatique de messages pour prévenir les enseignants et gestionnaires lors des validations, modifications ou rappels importants.
- **Développement de tableaux de bord :**
Proposer une vision synthétique des affectations réalisées et des charges horaires pour faciliter le suivi par les gestionnaires.
- **Optimisation des performances :**
Améliorer le fonctionnement du planning détaillé, notamment lorsque le nombre de cours ou d'événements devient important.
- **Renforcement de la sécurité :**
Imposer des mots de passe d'au moins huit caractères avec des caractères spéciaux pour les utilisateurs.
- **Gestion conditionnelle des exports :**
Autoriser l'exportation des fiches uniquement après validation par le gestionnaire.

- **Amélioration des alertes :**
Modifier certains messages d'alerte pour les rendre plus explicites et clairs.
- **Notifications côté enseignant :**
Afficher une alerte aux enseignants lorsqu'un gestionnaire modifie leur fiche ou remplit leur fiche.
- **Corrections de petites erreurs :**
Finaliser le fonctionnement du remplissage de la fiche prévisionnelle, qui présente encore quelques dysfonctionnements dans certains cas.

Grâce à l'architecture mise en place, **le projet peut être repris** par d'autres étudiants afin de poursuivre son évolution et de renforcer encore davantage la qualité de l'application.

Annexes

Handsontable et son Importance dans le Projet

Handsontable est une bibliothèque JavaScript spécialisée dans la création de tableaux interactifs et éditables, offrant des fonctionnalités avancées telles que la gestion des formules, la fusion de cellules et l'intégration avec d'autres outils comme SheetJS pour l'exportation de données.

Dans le cadre de notre projet, Handsontable s'est révélé être un choix stratégique pour la gestion du planning détaillé et des affectations. Son principal avantage réside dans sa légèreté, ce qui réduit l'empreinte mémoire et optimise la réactivité de l'application. De plus, la gestion des formules intégrée permet de calculer automatiquement certains totaux, limitant ainsi la surcharge de traitement. Une autre fonctionnalité clé est la gestion des modifications en attente, qui permet d'optimiser la sauvegarde des données en regroupant les modifications avant de les enregistrer.

Cependant, malgré ses nombreux atouts, l'intégration de Handsontable dans notre projet a soulevé plusieurs défis. La complexité de la structure du tableau a rendu la mise en place difficile, notamment en raison de la nécessité de gérer des entêtes fusionnées et des types de données variés (SAÉ, cours, etc.). En outre, la personnalisation du style s'est avérée limitée, rendant l'affichage moins esthétique que souhaité.

Enfin, bien que Handsontable propose une option d'exportation en Excel, celle-ci s'est avérée insuffisamment flexible pour répondre à nos besoins. C'est pourquoi nous avons opté pour l'utilisation de la bibliothèque SheetJS, qui nous a permis de proposer un export plus complet et structuré.

Fonctionnement du planning détaillé

Le planning détaillé est créé en plusieurs en plusieurs étapes qui sont les suivantes :

1. Récupération des différents éléments comme les configurations de planning
2. La création des en têtes
3. La création du tableau de données
4. La création des totaux
5. Les modifications lors des changements
6. Les vérifications des cellules pour les différents style
7. La fonction `saveAllData` qui récupérer chacun des éléments pour les mettre dans un tableau
8. La fonction `mergedRepartition` qui permet de fusionner les repartition d'heure pour pas insérer une case par une case dans la base de données
9. La gestion du scroll si on peut modifier ou non un élément du tableau

1. Configuration du planning

Au départ nous récupérons les différentes données qui nous seront utiles

- Dates de début et de fin pour chaque semestre
- Nombre de semaines par semestre
- Périodes de vacances (Toussaint, Noël, Hiver, Printemps)
- Liste de toutes les semaines de vacances
- Événement spéciaux ajouté

La fonction `getWeek()` est définie pour calculer le numéro de semaine ISO à partir d'une date donnée, ce qui est essentiel pour le suivi chronologique du planning.

```
for (let item of configData) {  
  switch (item.type) {  
    case 'Semestre1':  
      // Affectation des valeurs pour le semestre 1  
      const semestre1 = item;  
      dateDebutSemestre1 = new Date(semestre1.dateDebut);  
      dateFinSemestre1 = new Date(semestre1.dateFin);  
      nbSemaine1 = semestre1.nbSemaines;  
      break;  
  
    case 'Semestre2':
```

2. Construction des En-têtes du Tableau

La création des en-têtes est un processus qui génère un tableau à structure complexe avec plusieurs niveaux d'informations. Dont des parties fusionnées et qui doivent être variable selon le semestre choisi et/ou les modules qui sont associés à ce semestre.

3. Construction du Tableau de Données

Structure des Données

Pour chaque semaine du semestre, on génère une ligne de données avec :

- Le numéro de la semaine
- La date de début de semaine
- Une description (vide, "Vacances", ou autre description d'événement)
- Des cellules pour chaque type d'heure (CM, TD, TP) pour chaque cours standard
- Des cellules pour chaque SAE

```
// Construire la liste des colonnes pour les cours normaux (non SAE)
for (let cours of colCours) {
  let valueCM = "";
  let valueTD = "";
  let valueTP = "";

  for (let rep of repartitionSansProf) {
    if (cours === rep.nom_cours && rep.semaine_debut ≤ semaineActuelle && rep.semaine_fin ≥ semaineActuelle) {
      if (!estVacances) {
        if (rep.type_heure === 'CM') {
          valueCM = rep.nb_heures_par_semaine;
        } else if (rep.type_heure === 'TD') {
          valueTD = rep.nb_heures_par_semaine;
        } else if (rep.type_heure === 'TP') {
          valueTP = rep.nb_heures_par_semaine;
        }
      }
    }
  }
}
```

4. Construction des totaux

Une ligne supplémentaire est ajoutée en bas du tableau et à la fin de chaque ligne du tableau pour calculer les totaux par colonne :

```
// Ajouter les formules des totaux en une seule opération
const totalRowIndex = dataT.length - 1;
const totalColIndex = dataT[0].length - 1;

planning.batch(() => {
  // Pour chaque colonne de données (à partir de la colonne 4)
  for (let i = 4; i < totalColIndex; i++) {
    let totalFormula = `=SUM(${Handsontable.helper.spreadsheetColumnLabel(i)}1:$
    planning.setDataAtCell(totalRowIndex, i, totalFormula);
  }
});
```

5. Gestion des modifications

On utilise l'événement `afterChange` de `Handsontable` pour détecter les modifications apportées au tableau. Les modifications sont d'abord stockées dans un tableau `pendingChanges`, puis un minuteur de temporisation (`debounce`) est utilisé pour éviter

d'envoyer trop de requêtes au serveur. La sauvegarde est déclenchée 3 secondes après la dernière modification..

```
afterChange: (changes, source) => {
  if (source === 'loadData' || !changes) return;

  changes.forEach(([row, col, oldValue, newValue]) => {
    if (oldValue !== newValue) {
      pendingChanges.push({ row, col, newValue });
    }
  });

  if (!debounceTimer) {
    debounceTimer = setTimeout(() => {
      saveAllData(pendingChanges);
      pendingChanges = [];
      debounceTimer = null;
    }, 3000);
  }
},
```

6. La Fonction saveAllData

Cette fonction cruciale parcourt toutes les données du tableau pour préparer deux structures :

repartitions : Les heures planifiées pour chaque cours et type d'heure
descriptions : Les descriptions associées à certaines semaines

Pour se faire la fonction réalise les actions suivantes :

Parcourir chaque ligne (semaine) du tableau
Pour chaque cours standard, extraire les heures CM, TD et TP planifiées
Pour chaque SAE, extraire les heures planifiées

Pour chaque description, extraire le texte et les dates

Chaque information est organisée avec sa semaine de début, sa semaine de fin, le code du cours concerné et le type d'heure.

7. Optimisation avec mergeRepartitions

Au lieu d'envoyer une entrée de base de données par cellule modifiée, elle fusionne les répartitions consécutives et identiques :

```
repartitions.forEach((repartition, index) => {  
  // Si elles sont consécutives (semaineFin de la précédente égale à semaineDebut de la nouvelle)  
  if (currentRepartition.semaineFin + 1 === repartition.semaineDebut) {  
    // On fusionne en mettant à jour la semaine de fin  
    currentRepartition.semaineFin = repartition.semaineFin;  
  } else {  
    // Si elles ne sont pas consécutives, on les ajoute séparément  
    mergedRepartitions.push(currentRepartition);  
    currentRepartition = { ...repartition }; // Nouvelle répartition  
  }  
} else {  
  // Si la répartition est différente, on ajoute la précédente (si elle existe) et on commence une nouvelle  
  if (currentRepartition) {  
    mergedRepartitions.push(currentRepartition);  
  }  
  currentRepartition = { ...repartition }; // Créer une copie de la répartition actuelle  
}  
});
```

8. Gestion du scroll et de la modification de certaines lignes

Cette fonction permet aux utilisateurs de :

- Sélectionner une ou plusieurs cellules
- Utiliser la molette de la souris pour augmenter ou diminuer les valeurs
- Modifier rapidement plusieurs cellules à la fois

Des vérifications importantes sont effectuées :

- Les cellules des semaines de vacances ne peuvent pas être modifiées
- Les cellules des événements non modifiables (listés dans eventTexts) sont protégées
- Les valeurs ne peuvent pas descendre en dessous de zéro

L'utilisation de `planning.batch()` est cruciale ici car elle regroupe toutes les modifications en une seule opération, ce qui améliore considérablement les performances en réduisant le nombre de rendus.

```

container.addEventListener('wheel', (event) => {

    event.preventDefault(); // Empêche le défilement de la page
    const selectedRanges = planning.getSelected();
    if (!selectedRanges) return;
    //Texts des evenements si checkbox = 0
    console.log(eventTexts);
    // Utiliser batch pour regrouper les modifications
    planning.batch() => {
        selectedRanges.forEach(([startRow, startCol, endRow, endCol]) => {
            for (let row = startRow; row ≤ endRow; row++) {
                for (let col = startCol; col ≤ endCol; col++) {
                    if (dataT[row] && (dataT[row][3] ≡ "Vacances" ) && !eventTexts.includes(dataT[row][3]) && col ≥ 4 && co
                        let currentValue = parseInt(planning.getDataAtCell(row, col) || '0', 10);
                        const delta = event.deltaY > 0 ? -1 : 1; // -1 pour scroll bas, +1 pour scroll haut
                        currentValue = Math.max(0, currentValue + delta); // Ajuster la valeur sans descendre en dessous de 0
                        planning.setDataAtCell(row, col, currentValue); // Mettre à jour la cellule
                }
            }
        })
    }
}

```

Conclusion

Handsontable s'est révélé être un outil clé dans la gestion du planning détaillé de notre projet, offrant des fonctionnalités avancées telles que l'édition interactive, la gestion des formules et l'intégration avec SheetJS nous a permis de faire un export viable de ce planning. Malgré les défis rencontrés, notamment la gestion des entêtes fusionnées et les limitations esthétiques, son implémentation a permis d'optimiser l'organisation et la manipulation des données.

L'ensemble du processus de construction du planning, depuis la récupération des configurations jusqu'à la gestion des modifications et l'optimisation des sauvegardes, a nécessité une approche rigoureuse.

En conclusion, bien que son intégration ait nécessité des ajustements techniques, Handsontable a permis de structurer efficacement le planning détaillé tout en assurant une bonne réactivité et une manipulation intuitive des données.

Fiche Répartition

Pour la fiche répartition nous utilisons la librairie javascript Handsontable pour pouvoir créer les tableaux. Dans le cas de cette page nous créons plusieurs tables de 3 cours.

Cette boucle gère la séparation des données et la création des différentes tables.

```
for (let chunkIndex = 0; chunkIndex < numChunks; chunkIndex++) {  
}
```

La fonction nécessaire à la création d'une Handsontable.

```
const hot = new Handsontable(tableDiv, {  
  data: tableDataChunk,  
  width: '100%',  
  height: 510,  
  stretchH: 'all',  
  nestedHeaders: nestedHeaders,  
  colWidths: 150,  
  rowHeights: 50,  
  wordWrap: true,  
  licenseKey: 'non-commercial-and-evaluation',  
  columns: columnsDefs,  
  afterGetColHeader: function () {  
    const headerRows = tableDiv.querySelectorAll('.ht_clone_top thead tr');  
    headerRows.forEach((row, rowIndex) => {  
      if (rowIndex === 0) {  
        row.querySelectorAll('th').forEach((th) => {  
          th.style.backgroundColor = '#007bff';  
          th.style.color = 'ffffff';  
        });  
      } else if (rowIndex === 1) {  
        row.querySelectorAll('th').forEach((th) => {  
          th.style.backgroundColor = '#ffc107';  
          th.style.color = '000000';  
        });  
      } else if (rowIndex === 2) {  
        row.querySelectorAll('th').forEach((th) => {  
          th.style.backgroundColor = '#28a745';  
          th.style.color = 'ffffff';  
        });  
      }  
    });  
  }  
}
```

afterGetColHeader

Gère le style du header de la table.

tableDataChunk

Il s'agit d'un sous-ensemble des données globales, adapté pour une instance Handsontable spécifique.

Fonctionnement :

Pour chaque chunk (groupe de cours défini par la variable `coursesPerChunk`), on extrait depuis le tableau complet data uniquement les colonnes correspondant aux cours du morceau en cours.

Utilisation :

Cette variable fournit les données pré-remplies qui seront affichées dans le tableau Handsontable. Chaque ligne représente un groupe, avec la première colonne fixe suivie des colonnes spécifiques aux cours sélectionnés pour ce chunk.

columnDefs

Il définit la configuration et les caractéristiques de chaque colonne de l'instance Handsontable.

Fonctionnement :

La première colonne est configurée en lecture seule pour afficher le nom du groupe.

Pour chaque cours présent dans le chunk, on ajoute **4 colonnes** correspondant aux différents types d'heures (CM, TD, TP, EI).

Chaque colonne ajoutée est configurée comme un champ d'autocomplétion (type `autocomplete`) avec une liste de valeurs pour faciliter la saisie.

Utilisation :

Cette variable informe Handsontable sur la manière dont chaque colonne doit se comporter, incluant le type de données, la possibilité d'édition, et l'apparence. Cela permet d'assurer que chaque champ dans le tableau est correctement configuré pour les interactions de l'utilisateur.

Validation de fiche

Dans notre projet, la validation des fiches est un élément essentiel pour garantir la fiabilité des données saisies par les enseignants.

Chaque enseignant doit remplir plusieurs types de fiches : une fiche de contraintes (fiche de vœux), une fiche de ressources (s'il est responsable de module) et une fiche prévisionnelle. Toutes ces fiches sont stockées dans des tables spécifiques de la base de données : **contraintes**, **détails_cours** et **vœux**.

Pour savoir si une fiche est remplie et si elle peut encore être modifiée ou non, nous avons mis en place un **système de verrouillage basé sur un champ statut** dans chaque table.

Ce champ peut prendre deux valeurs :

- **"en attente"** : la fiche est encore modifiable.
- **"validée"** : la fiche est verrouillée, elle devient non modifiable.

Le fonctionnement de la validation est le suivant :

- Lorsqu'un enseignant remplit sa fiche, elle est enregistrée automatiquement en base de données avec un **statut** à **"en attente"**. À ce stade, l'enseignant peut modifier ses réponses librement.
- Lorsqu'un gestionnaire consulte les fiches depuis son interface, il peut cliquer sur un bouton « Valider » pour valider une fiche. Cela envoie une requête qui met à jour la base de données et passe le **statut** de **"en attente"** à **"validée"**.
- Une fois validée, la fiche est verrouillée automatiquement. L'enseignant ne peut plus modifier son contenu : les champs deviennent grisés et un message d'information s'affiche pour l'en avertir.

Si jamais une erreur est constatée après validation, le gestionnaire peut cliquer sur « Dévalider ». Cette action remet le **statut** à **"en attente"**, rendant la fiche à nouveau modifiable.

Le fonctionnement est le même pour la modification sauf que c'est un boolean (colonne **modification_en_cours**)