

DOSSIER D'ANALYSE PROJET TUTEORE

✓ ETAPE D 'ANALYSSE

Voici quelques questions pour commencer l'analyse:

1. Quels sont les types d'utilisateurs de l'application et quels sont leurs rôles ?
2. Quelles sont les informations que nous souhaitons stocker dans l'application ?
3. Comment souhaitons-nous visualiser les emplois du temps pour l'école ?
4. Comment souhaitons-nous planifier automatiquement les emplois du temps pour les enseignants et les étudiants ?
5. Avons-nous des exigences particulières en matière de sécurité et de confidentialité des données ?

Une fois que nous aurons répondu à ces questions, nous pourrons commencer à définir les fonctionnalités de l'application et à les diviser en tâches plus petites pour les inclure dans les sprints.

Voici une liste de fonctionnalités que nous pouvons inclure dans l'application de gestion des emplois du temps académique :

1. Système de gestion des cours : permettant de stocker les informations sur les cours, les enseignants, les étudiants et les salles de classe pour chaque école.
2. Système de gestion des horaires : permettant de stocker les horaires pour chaque cours et pour chaque enseignant.
3. Fonctionnalité de planification automatique : permettant de générer des emplois du temps pour les enseignants et les étudiants en évitant les conflits.
4. Interface utilisateur pour la visualisation des emplois du temps : permettant aux utilisateurs de visualiser facilement les emplois du temps pour chaque école.
5. Fonctionnalité de recherche : permettant aux utilisateurs de rechercher rapidement des informations sur les cours, les enseignants, les étudiants et les salles de classe.
6. Fonctionnalité d'importation/exportation : permettant aux écoles de transférer facilement des données entre l'application et d'autres systèmes.
7. Système d'authentification et de gestion des utilisateurs : permettant de contrôler l'accès à l'application et d'attribuer des rôles à chaque utilisateur.
8. Fonctionnalité de notifications : permettant d'envoyer des notifications aux utilisateurs pour les informer des changements ou des mises à jour importantes.

Cette liste n'est pas exhaustive et il est possible que nous ajoutions ou supprimions des fonctionnalités au fil du temps en fonction de nos besoins.

➔ LISTE DES SPRINTS

liste de sprints pour le développement de l'application de gestion des emplois du temps académiques :

Sprint 1

- Configuration de l'environnement de développement
- Mise en place du système d'authentification et de gestion des utilisateurs
- Mise en place de la base de données pour stocker les informations sur les écoles, les cours, les enseignants, les étudiants et les salles de classe
- Développement de l'interface utilisateur pour la visualisation des emplois du temps

Sprint 2

- Développement de la fonctionnalité de recherche
- Mise en place de la fonctionnalité d'importation/exportation
- Ajout de la fonctionnalité de notifications
- Tests et révision du code développé lors du sprint 1

Sprint 3

- Développement de la fonctionnalité de planification automatique
- Tests et révision du code développé lors du sprint 2

Sprint 4

- Finalisation du développement de l'application
- Tests et révision de l'ensemble de l'application
- Préparation pour le déploiement

Nous pouvons ajuster cette liste en fonction de nos besoins. Chaque sprint durera environ une semaine, ce qui nous permettra d'avancer rapidement tout en conservant une grande souplesse.

✓ ETAPE DE CONCEPTION

partie conception, nous allons utiliser les informations collectées lors de l'étape d'analyse pour concevoir l'architecture de l'application, définir les technologies à utiliser et planifier la mise en œuvre. Voici les éléments clés de la conception de notre application :

1. **Architecture** : Nous allons concevoir une architecture qui permettra de stocker les informations sur les écoles, les cours, les enseignants, les étudiants et les salles de classe dans une base de données centralisée. Nous utiliserons un modèle relationnel pour stocker ces informations.
2. **Technologies** : Nous allons utiliser des technologies modernes pour le développement de l'application, y compris des frameworks web tels que SprigBoot. Nous utiliserons également des technologies frontales telles que Bootstrap pour la partie client de l'application.
3. **Planification automatique** : Nous allons concevoir un algorithme de planification automatique qui permettra de générer des emplois du temps pour les enseignants et les étudiants en évitant les conflits. Nous utiliserons des techniques d'optimisation pour garantir que les emplois du temps générés sont optimaux.
4. **Interface utilisateur** : Nous allons concevoir une interface utilisateur intuitive et conviviale qui permettra aux utilisateurs de visualiser facilement les emplois du temps pour chaque école. Nous utiliserons des techniques de conception centrée sur l'utilisateur pour garantir que l'interface est facile à utiliser.

5. Sécurité et confidentialité : Nous allons concevoir un système d'authentification et de gestion des utilisateurs qui permettra de contrôler l'accès à l'application et d'attribuer des rôles à chaque utilisateur.

Une fois que la conception sera terminée, nous pourrons passer à l'étape de développement de l'application.

- ✗ **pour l'architecture de l'application**, nous allons concevoir une architecture qui permettra de stocker les informations sur les écoles, les cours, les enseignants, les étudiants et les salles de classe dans une base de données centralisée. Nous utiliserons un modèle relationnel pour stocker ces informations.

Voici le schéma de base de données pour l'application :

- Table "Schools" : Cette table contiendra les informations sur chaque école, telles que le nom de l'école, l'adresse, le numéro de téléphone, etc.
- Table "Courses" : Cette table contiendra les informations sur chaque cours, telles que le nom du cours, la description, le nombre de crédits, etc.
- Table "Teachers" : Cette table contiendra les informations sur chaque enseignant, telles que le nom, l'adresse e-mail, le numéro de téléphone, etc.
- Table "Students" : Cette table contiendra les informations sur chaque étudiant, telles que le nom, l'adresse e-mail, le numéro de téléphone, etc.
- Table "Classrooms" : Cette table contiendra les informations sur chaque salle de classe, telles que le nom de la salle de classe, la capacité, les équipements disponibles, etc.
- Table "Schedules" : Cette table contiendra les informations sur les emplois du temps pour chaque école. Elle sera liée aux tables "Schools", "Courses", "Teachers", "Students" et "Classrooms".

Nous allons utiliser un système de clés étrangères pour lier ces tables entre elles et assurer l'intégrité des données. Nous allons également utiliser des index pour améliorer les performances des requêtes.

En utilisant cette architecture, nous pourrons stocker toutes les informations nécessaires pour générer les emplois du temps pour chaque école.

Pour la conception, Nous allons utiliser Bootstrap 5 pour la conception de l'interface utilisateur de l'application.

Pour la partie back-end de l'application, nous allons utiliser Spring Boot. Nous allons également utiliser Spring Data JPA pour la gestion des données de l'application.

Nous allons utiliser des outils de développement modernes tels que Maven pour la gestion des dépendances et la construction de l'application. Nous allons utiliser Git pour la gestion de version.

En utilisant Bootstrap 5 pour la partie frontale de l'application, nous pourrons développer une interface utilisateur moderne et responsive qui offrira une expérience utilisateur améliorée.

pour la planification automatique des emplois du temps, nous allons concevoir un algorithme qui permettra de générer des emplois du temps pour les enseignants et les étudiants en évitant les

conflits. Nous utiliserons des techniques d'optimisation pour garantir que les emplois du temps générés sont optimaux.

Voici les étapes clés de l'algorithme de planification automatique :

1. Collecte des données : Nous allons collecter toutes les informations nécessaires sur les écoles, les cours, les enseignants, les étudiants et les salles de classe à partir de la base de données.
2. Définition des contraintes : Nous allons définir toutes les contraintes liées à la planification des emplois du temps, telles que les horaires des cours, les disponibilités des enseignants et des salles de classe, les préférences des étudiants, etc.
3. Génération d'un emploi du temps initial : Nous allons générer un emploi du temps initial en utilisant un algorithme simple tel que l'algorithme de tri par bulles. Cet emploi du temps initial sera utilisé comme point de départ pour l'optimisation.
4. Optimisation de l'emploi du temps : Nous allons utiliser des techniques d'optimisation telles que l'algorithme génétique ou l'algorithme du recuit simulé pour optimiser l'emploi du temps initial. Ces algorithmes permettent de trouver rapidement une solution optimale en explorant l'espace de recherche des solutions possibles.
5. Évaluation de l'emploi du temps : Nous allons évaluer l'emploi du temps généré en fonction des critères définis par les utilisateurs. Nous utiliserons des métriques telles que le nombre de conflits d'horaire, la satisfaction des enseignants et des étudiants, etc.
6. Affichage de l'emploi du temps : Nous allons afficher l'emploi du temps généré dans une interface utilisateur conviviale qui permettra aux utilisateurs de visualiser facilement les horaires des cours pour chaque école.

En utilisant cet algorithme de planification automatique, nous pourrions générer rapidement des emplois du temps optimaux pour les enseignants et les étudiants en évitant les conflits d'horaire.

- ✗ **pour la collecte des données nécessaires à la planification automatique des emplois du temps**, nous allons utiliser une base de données centralisée qui stockera toutes les informations sur les écoles, les cours, les enseignants, les étudiants et les salles de classe.

Nous allons utiliser des formulaires web pour collecter les données auprès des utilisateurs. Les utilisateurs pourront saisir les informations sur les écoles, les cours, les enseignants, les étudiants et les salles de classe dans ces formulaires. Les données saisies seront ensuite stockées dans la base de données centrale.

En utilisant ces méthodes de collecte de données, nous pourrions collecter toutes les informations nécessaires à la planification automatique des emplois du temps de manière efficace et fiable.

- ✗ **pour la définition des contraintes liées à la planification automatique des emplois du temps**, nous allons prendre en compte les contraintes suivantes :

1. Contraintes d'horaire : Nous allons définir les horaires des cours pour chaque école ainsi que les horaires de disponibilité des enseignants et des salles de classe. Nous veillerons à ce que

les horaires des cours soient compatibles avec les horaires de disponibilité des enseignants et des salles de classe.

2. Contraintes de capacité : Nous allons définir les capacités des salles de classe ainsi que les capacités maximales des cours. Nous veillerons à ce que les salles de classe soient suffisamment grandes pour accueillir tous les étudiants inscrits aux cours.
3. Contraintes de préférence : Nous allons prendre en compte les préférences des étudiants en matière d'horaire et de cours. Nous veillerons à ce que les étudiants soient inscrits aux cours qui correspondent le mieux à leurs préférences.
4. Contraintes de conflit : Nous allons éviter les conflits d'horaire entre les cours, les enseignants et les salles de classe. Nous veillerons à ce qu'un enseignant ne soit pas programmé pour enseigner deux cours en même temps ou qu'une salle de classe ne soit pas programmée pour deux cours en même temps.

En définissant ces contraintes, nous pourrions garantir que les emplois du temps générés sont optimaux et répondent aux besoins des enseignants et des étudiants.

pour la génération d'un emploi du temps initial, nous allons utiliser un algorithme simple tel que l'algorithme de tri par bulles. Cet algorithme nous permettra de générer rapidement un emploi du temps initial qui sera utilisé comme point de départ pour l'optimisation.

L'algorithme de tri par bulles fonctionne en comparant les éléments adjacents et en les échangeant s'ils ne sont pas dans le bon ordre. Nous allons utiliser cet algorithme pour trier les cours en fonction de leur horaire et pour les attribuer aux enseignants et aux salles de classe disponibles.

Voici les étapes clés de l'algorithme de génération de l'emploi du temps initial :

1. Collecte des données : Nous allons collecter toutes les informations nécessaires sur les écoles, les cours, les enseignants, les étudiants et les salles de classe à partir de la base de données.
2. Définition des contraintes : Nous allons définir toutes les contraintes liées à la planification des emplois du temps, telles que les horaires des cours, les disponibilités des enseignants et des salles de classe, les préférences des étudiants, etc.
3. Tri des cours : Nous allons trier les cours en fonction de leur horaire en utilisant l'algorithme de tri par bulles.
4. Attribution des cours : Nous allons attribuer les cours triés aux enseignants et aux salles de classe disponibles en respectant les contraintes définies.
5. Génération de l'emploi du temps initial : Nous allons générer l'emploi du temps initial en utilisant les informations sur les cours, les enseignants et les salles de classe attribués.

En utilisant cet algorithme de génération d'un emploi du temps initial, nous pourrions générer rapidement un emploi du temps qui sera utilisé comme point de départ pour l'optimisation.

- ✗ **pour l'affichage de l'emploi du temps**, nous allons utiliser une interface utilisateur conviviale qui permettra aux utilisateurs de visualiser facilement l'emploi du temps. Cette interface utilisateur sera développée sous forme d'application web.

Voici les étapes clés de l'affichage de l'emploi du temps :

1. Collecte des données : Nous allons collecter toutes les informations nécessaires sur les écoles, les cours, les enseignants, les étudiants et les salles de classe à partir de la base de données.
2. Création d'une interface utilisateur : Nous allons créer une interface utilisateur conviviale qui permettra aux utilisateurs de visualiser facilement l'emploi du temps. Cette interface utilisateur peut inclure des fonctionnalités telles que la visualisation des cours par jour, par semaine ou par mois, la recherche de cours, la modification des préférences, etc.
3. Intégration de l'emploi du temps : Nous allons intégrer l'emploi du temps généré dans l'interface utilisateur. L'emploi du temps peut être affiché sous forme de tableau ou de calendrier.
4. Personnalisation de l'affichage : Nous allons permettre aux utilisateurs de personnaliser l'affichage de l'emploi du temps en fonction de leurs préférences. Les utilisateurs peuvent choisir d'afficher uniquement certains cours ou de masquer certains cours.
5. Notification : Nous allons permettre aux utilisateurs de recevoir des notifications pour les changements apportés à l'emploi du temps. Les notifications peuvent être envoyées par e-mail ou par message texte.

En utilisant ces étapes d'affichage, nous pourrions fournir aux utilisateurs une interface utilisateur conviviale qui leur permettra de visualiser facilement l'emploi du temps généré.

- ✗ **pour l'interface utilisateur**, nous allons créer une interface conviviale qui permettra aux utilisateurs de facilement interagir avec le système de planification d'emploi du temps. Cette interface utilisateur sera développée sous forme d'application web.

Voici les étapes clés de la création de l'interface utilisateur :

1. Collecte des données : Nous allons collecter toutes les informations nécessaires sur les écoles, les cours, les enseignants, les étudiants et les salles de classe à partir de la base de données.
2. Analyse des besoins des utilisateurs : Nous allons analyser les besoins des utilisateurs en termes d'interface utilisateur. Nous allons prendre en compte les préférences des utilisateurs en matière de conception, d'ergonomie, de navigation, etc.
3. Conception de l'interface utilisateur : Nous allons concevoir l'interface utilisateur en utilisant des outils tels que Figma. Nous allons créer des maquettes pour chaque écran de l'interface utilisateur, en prenant en compte les besoins des utilisateurs.
4. Développement de l'interface utilisateur : Nous allons développer l'interface utilisateur en utilisant des technologies telles que HTML, CSS . Nous allons créer des pages web qui permettent aux utilisateurs d'interagir avec le système de planification d'emploi du temps.
5. Test de l'interface utilisateur : Nous allons tester l'interface utilisateur en utilisant des techniques telles que les tests d'utilisabilité et les tests A/B. Nous allons collecter les commentaires des utilisateurs et apporter les modifications nécessaires en fonction de leurs commentaires.

6. Déploiement de l'interface utilisateur : Nous allons déployer l'interface utilisateur sur un serveur web.

En utilisant ces étapes, nous pourrions créer une interface utilisateur conviviale qui permettra aux utilisateurs d'interagir facilement avec le système de planification d'emploi du temps.

- ✗ **pour la sécurité et la confidentialité**, nous allons prendre les mesures nécessaires pour garantir que les données des utilisateurs sont sécurisées et confidentielles.

Voici les étapes clés pour assurer la sécurité et la confidentialité des données :

1. Protection des données : Nous allons protéger les données des utilisateurs en utilisant des techniques telles que les pare-feux, les antivirus, etc. Nous allons également mettre en place des protocoles de sécurité pour protéger les données contre les attaques de pirates informatiques.
2. Gestion des accès : Nous allons mettre en place un système de gestion des accès pour garantir que seuls les utilisateurs autorisés ont accès aux données. Nous allons également mettre en place des niveaux d'accès pour garantir que chaque utilisateur a accès uniquement aux données qui lui sont nécessaires.
3. Confidentialité : Nous allons garantir la confidentialité des données en utilisant des techniques telles que l'anonymisation des données, la pseudonymisation des données, etc. Nous ne divulguons pas les données des utilisateurs à des tiers sans leur consentement.
4. Respect de la réglementation : Nous allons respecter les réglementations en vigueur en matière de protection des données, telles que le RGPD (Règlement général sur la protection des données) en Europe ou le CCPA (California Consumer Privacy Act) aux États-Unis.
5. Formation du personnel : Nous allons former notre personnel sur la sécurité et la confidentialité des données pour garantir qu'ils comprennent les risques liés à la sécurité et à la confidentialité des données et qu'ils sont en mesure de prendre les mesures nécessaires pour protéger les données des utilisateurs.

En utilisant ces étapes, nous pourrions garantir que les données des utilisateurs sont sécurisées et confidentielles.

Voici comment nous pourrions utiliser UML pour concevoir le système de planification d'emploi du temps :

1. Diagramme de classes : Nous pouvons utiliser un diagramme de classes pour identifier les classes nécessaires pour le système de planification d'emploi du temps, ainsi que leurs attributs et leurs relations. Nous pouvons identifier les classes suivantes : École, Cours, Enseignant, Étudiant, Salle de classe, etc.
2. Diagramme de séquence : Nous pouvons utiliser un diagramme de séquence pour modéliser les interactions entre les différents objets du système. Nous pouvons modéliser la séquence d'actions qui se produit lorsque l'utilisateur ajoute un nouveau cours à l'emploi du temps.

3. Diagramme d'activité : Nous pouvons utiliser un diagramme d'activité pour modéliser le flux de travail du système. Nous pouvons modéliser le processus de génération d'un emploi du temps en utilisant un diagramme d'activité.

En utilisant ces diagrammes UML, nous pourrions concevoir le système de planification d'emploi du temps de manière structurée et organisée, en identifiant les classes nécessaires, les interactions entre les objets et le flux de travail du système.

nous pourrions utiliser un diagramme de classe pour concevoir le système de planification d'emploi du temps :

1. Identifier les classes : Nous allons identifier les classes nécessaires pour le système de planification d'emploi du temps, telles que École, Cours, Enseignant, Étudiant, Salle de classe, etc.
2. Ajouter les attributs : Nous allons ajouter les attributs à chaque classe, tels que le nom de l'école, le nom du cours, l'adresse e-mail de l'enseignant, etc.
3. Ajouter les méthodes : Nous allons ajouter les méthodes à chaque classe, telles que la méthode pour ajouter un nouveau cours à l'emploi du temps.
4. Ajouter les relations : Nous allons ajouter les relations entre les classes, telles que la relation entre École et Cours (une école peut avoir plusieurs cours) ou la relation entre Cours et Enseignant (un cours peut être enseigné par un seul enseignant).
5. Ajouter les interfaces : Si nécessaire, nous allons ajouter des interfaces pour représenter les fonctionnalités que les classes doivent implémenter.

En utilisant ce diagramme de classe, nous pourrions concevoir le système de planification d'emploi du temps de manière structurée et organisée, en identifiant les classes nécessaires, leurs attributs et leurs relations.

nous pourrions utiliser un diagramme d'activité pour modéliser le processus de génération d'un emploi du temps :

1. Définir les activités : Nous allons définir les activités nécessaires pour générer un emploi du temps, telles que la sélection des cours, l'attribution des enseignants aux cours, la planification des salles de classe, etc.
2. Ajouter les décisions : Nous allons ajouter des décisions pour représenter les choix qui doivent être faits pendant le processus de génération de l'emploi du temps, par exemple, décider si un enseignant est disponible pour enseigner un cours à un moment donné.
3. Ajouter les transitions : Nous allons ajouter les transitions entre les activités pour représenter le flux de travail, par exemple, la transition entre la sélection des cours et l'attribution des enseignants.
4. Ajouter les entrées et sorties : Nous allons ajouter les entrées et sorties pour représenter les données nécessaires pour chaque activité, par exemple, la liste des cours disponibles pour la sélection des cours.
5. Ajouter les annotations : Nous allons ajouter des annotations pour fournir des informations supplémentaires sur chaque activité, par exemple, la durée approximative de chaque activité.

En utilisant ce diagramme d'activité, nous pourrions modéliser le processus de génération d'un emploi du temps de manière structurée et organisée, en identifiant les activités nécessaires, les décisions à prendre et les flux de travail. Ce diagramme sera également utile pour les développeurs qui travailleront sur le projet, car il leur donnera une vue d'ensemble du flux de travail du système.

Voici comment nous pourrions utiliser un diagramme de séquence pour modéliser le processus de génération d'un emploi du temps :

1. Identifier les objets : Nous allons identifier les objets impliqués dans le processus de génération d'un emploi du temps, tels que l'utilisateur, le système, les cours, les enseignants, les salles de classe, etc.
2. Ajouter les messages : Nous allons ajouter les messages échangés entre les objets pour représenter les interactions, par exemple, l'utilisateur sélectionne un cours, le système vérifie la disponibilité des enseignants pour ce cours, etc.
3. Ajouter les acteurs : Nous allons ajouter les acteurs pour représenter les utilisateurs ou les systèmes externes avec lesquels le système interagit, par exemple, l'utilisateur qui sélectionne un cours.
4. Ajouter les lifelines : Nous allons ajouter les lifelines pour représenter la durée de vie des objets impliqués dans le processus de génération d'un emploi du temps.
5. Ajouter les contraintes temporelles : Nous allons ajouter des contraintes temporelles pour représenter la durée de chaque interaction, par exemple, la durée approximative de la sélection des cours.

En utilisant ce diagramme de séquence, nous pourrions modéliser les interactions entre les différents objets impliqués dans le processus de génération d'un emploi du temps de manière structurée et organisée.