

Cahier des charges fonctionnel et technique

Système intelligent d'automatisation et d'optimisation des emplois du temps

Équipe projet B1 2025

Juin 2025

Table des matières

1	Contexte du projet	2
2	Objectifs du projet	2
2.1	Objectif principal	2
2.2	Objectifs spécifiques	2
3	Périmètre fonctionnel	3
3.1	Fonctionnalités principales	3
4	Contraintes à respecter	4
4.1	Contraintes fonctionnelles	4
4.2	Contraintes techniques	4
5	Architecture prévue	4
5.1	Architecture générale	4
5.2	Technologies proposées	5
6	Livrables attendus	5
7	Organisation et planification	5
8	Critères de réussite	6
9	Analyse critique et perspectives	6
9.1	Limites potentielles	6
9.2	Perspectives dévolution	6
10	Conclusion	6

1 Contexte du projet

La gestion des emplois du temps dans les institutions éducatives est une tâche complexe en raison des nombreuses contraintes à respecter : disponibilités des enseignants, capacités des salles, répartition des groupes d'étudiants, préférences pédagogiques et absence de chevauchements. Actuellement, cette gestion est souvent manuelle, ce qui entraîne des erreurs, des conflits horaires et une surcharge administrative. Ce projet tuteuré, réalisé dans le cadre de la formation B1 2025, vise à développer une plateforme intelligente pour automatiser et optimiser la création des emplois du temps, en offrant une interface conviviale pour la saisie, la visualisation et l'exportation des résultats.

2 Objectifs du projet

2.1 Objectif principal

Concevoir et implémenter un système intelligent capable de générer automatiquement des emplois du temps optimisés, respectant les contraintes académiques et logistiques d'une institution éducative, tout en minimisant les conflits et en facilitant la gestion.

2.2 Objectifs spécifiques

- Gérer efficacement les ressources de l'établissement (enseignants, salles, groupes, matières).
- Intégrer un moteur d'optimisation basé sur des algorithmes intelligents (ex. : satisfaction de contraintes ou algorithmes génétiques).
- Fournir une interface utilisateur intuitive pour la saisie des données, la visualisation des emplois du temps et la gestion des conflits.
- Permettre l'exportation des emplois du temps dans des formats standards (PDF, CSV, ICS pour calendriers numériques).
- Assurer une journalisation des actions pour faciliter l'administration et le débogage.

3 Périmètre fonctionnel

3.1 Fonctionnalités principales

Module	Fonctionnalités
Gestion des utilisateurs	<ul style="list-style-type: none">— Création, modification, suppression des comptes (administrateurs, enseignants).— Authentification sécurisée avec rôles (administrateur, utilisateur).
Gestion des ressources	<ul style="list-style-type: none">— Ajout, modification, suppression des enseignants (nom, matières, disponibilités, préférences).— Gestion des salles (nom, capacité, équipements).— Gestion des groupes d'étudiants (effectif, matières suivies).— Gestion des matières (nom, durée, prérequis).
Saisie des contraintes	<ul style="list-style-type: none">— Définition des contraintes globales (ex. : horaires de fonctionnement de l'établissement).— Définition des contraintes individuelles (ex. : disponibilités des enseignants, capacités des salles, maximum d'heures par jour).— Support des contraintes strictes (hard) et préférentielles (soft).
Moteur d'optimisation	<ul style="list-style-type: none">— Génération automatique d'emplois du temps respectant les contraintes.— Détection et résolution des conflits avec suggestions d'ajustements.
Visualisation	<ul style="list-style-type: none">— Affichage des emplois du temps sous forme de grille ou calendrier.— Filtres par enseignant, groupe, salle ou période.
Exportation / Impression	<ul style="list-style-type: none">— Exportation en PDF, CSV ou ICS (calendrier numérique).— Impression directe depuis l'interface.
Administration / Logs	<ul style="list-style-type: none">— Journalisation des générations, erreurs et ajustements manuels.— Tableau de bord pour les administrateurs.

4 Contraintes à respecter

4.1 Contraintes fonctionnelles

- Respect strict des disponibilités des enseignants et des salles.
- Aucun chevauchement de cours pour un même groupe, enseignant ou salle.
- Répartition équilibrée de la charge horaire hebdomadaire pour les groupes et enseignants.
- Affectation des cours à des salles avec une capacité suffisante.
- Prise en compte des préférences pédagogiques (ex. : éviter les cours après 18h).

4.2 Contraintes techniques

- Accessibilité via un navigateur web (Chrome, Firefox, Safari).
- Persistance des données dans une base de données relationnelle.
- Temps de génération de l'emploi du temps inférieur à 1 minute pour une institution moyenne (100 enseignants, 50 salles, 20 groupes).
- Interface responsive, utilisable sur mobile et tablette.
- Sécurité des données avec authentification JWT et gestion des rôles.

5 Architecture prévue

5.1 Architecture générale

Le système suivra une architecture client/serveur :

- **Frontend** : Interface utilisateur web pour la saisie, la visualisation et l'exportation.
- **Backend** : API REST pour la logique métier et le moteur d'optimisation.
- **Base de données** : Stockage persistant des ressources et contraintes.

Schéma : [Frontend (React)] <-> [Backend (Django/Flask + OR-Tools)] <-> [Base de données (PostgreSQL)]

5.2 Technologies proposées

Composant	Technologie
Frontend	React.js avec Tailwind CSS pour une interface responsive et moderne.
Backend	Python avec Django (gestion des API et ORM) ou Flask (légèreté).
Moteur doptimisation	Google OR-Tools (CSP) ou PyGAD (algorithmes génétiques).
Base de données	PostgreSQL (robuste) ou SQLite (léger, pour prototypes).
Exportation	ReportLab (PDF), pandas (CSV), iCalendar (calendriers numériques).
Authentification	JWT via Django REST Framework.
Hébergement	Docker pour conteneurisation, déploiement sur VPS (Render/Heroku).

6 Livrables attendus

- Base de données relationnelle opérationnelle avec MCD/MLD.
- Interface web fonctionnelle pour la saisie, la visualisation et l'exportation.
- Moteur doptimisation respectant les contraintes avec gestion des conflits.
- Fonctionnalités d'exportation en PDF, CSV et ICS.
- Documentation technique (architecture, code) et utilisateur (guide d'utilisation).
- Rapport de tests (unitaires, fonctionnels, utilisateurs).

7 Organisation et planification

Phase	Durée	Livrables
Étude et modélisation	2 semaines	MCD, MLD, diagrammes UML (cas d'utilisation, classes, séquences).
Développement backend + BDD	3 semaines	API REST, ORM, base de données fonctionnelle.
Développement du moteur	3 semaines	Moteur doptimisation avec gestion des contraintes.
Interface utilisateur	2 semaines	Frontend fonctionnel (saisie, visualisation, exportation).
Tests et corrections	1 semaine	Rapport de tests (unitaires, fonctionnels, utilisateurs).
Documentation / livraison	1 semaine	Documentation technique, manuel utilisateur, rapport final.

8 Critères de réussite

- Taux de satisfaction des contraintes : $> 95\%$.
- Temps de génération : < 1 minute pour un emploi du temps complet.
- Interface fluide, intuitive, sans bugs majeurs (note utilisateur $> 4/5$).
- Exportations lisibles et conformes (PDF, CSV, ICS).
- Documentation claire et complète.

9 Analyse critique et perspectives

9.1 Limites potentielles

- Complexité des contraintes : Les préférences multiples ou contradictoires peuvent augmenter le temps de calcul.
- Évolutivité : Les grandes institutions (> 200 enseignants) pourraient nécessiter des optimisations supplémentaires.
- Intégration : La synchronisation avec des outils tiers (ex. : Google Calendar) peut poser des problèmes d'API.

9.2 Perspectives dévolution

- Ajout de contraintes avancées (ex. : pauses obligatoires, priorités pédagogiques).
- Intégration avec des environnements numériques de travail (ENT).
- Adaptation pour d'autres contextes (écoles primaires, entreprises).
- Optimisation des performances via des algorithmes hybrides ou parallélisation.

10 Conclusion

Ce projet tuteuré permettra de développer un système intelligent pour l'automatisation des emplois du temps, réduisant les erreurs et la charge administrative. Il mobilise des compétences en programmation, gestion de bases de données, optimisation et conception d'interfaces. Les perspectives de dévolution offrent un potentiel d'extension à divers contextes éducatifs et professionnels.