

Cahier des Charges - Projet Mnemosyne

Projet	Mnemosyne : Outil de Suivi des Cohortes Étudiantes
Professeur Référent	Monsieur Hébert
Responsable Projet	Monsieur Hébert
Parties Prenantes	IUT de Villetaneuse, Direction, Équipes Pédagogiques, Rectorat, Ministère
Objectif Principal	Visualiser et analyser les parcours des cohortes d'étudiants (BUT) de manière simple, rapide et interactive.
Version du Document	7.1
Date	13 novembre 2025

SOMMAIRE

1. Définitions et Périmètre du Projet

- 1.1. Scénario / Manifeste
- 1.2. Contexte et Problématique
- 1.3. Objectif Principal (Le "Cœur de Métier")
- 1.4. Périmètre du MVP (Inclus vs. Exclus)
- 1.5. Glossaire Métier (Terminologie)

2. Modèle de Valeur et Stratégie

- 2.1. Proposition de Valeur
- 2.2. Stratégie d'Adoption
- 2.3. Coûts et Ressources

3. Utilisateurs et "User Stories"

- 3.1. Profils Utilisateurs
- 3.2. User Stories (Fonctionnalités MVP)

4. Approfondissement Métier et Technique

- 4.1. Logique de Synchronisation
- 4.2. Scénario de Parcours Complexes (Exemples)
- 4.3. Structure de Données (Conceptuelle)
- 4.4. Architecture Applicative (Conceptuelle)

5. Impératifs et Contraintes

- 5.1. Contraintes Techniques (Stack)
- 5.2. Contraintes de Données
- 5.3. Contraintes Opérationnelles et Risques
- 5.4. Exigences Non-Fonctionnelles (Performance)
- 5.5. Exigences Non-Fonctionnelles (Sécurité)

6. Éléments Graphiques et Identité

- 6.1. Identité
- 6.2. Charte Graphique et Inspiration

7. Critères d'Acceptation (Definition of Done)

8. Stratégie de Test

9. Planning Prévisionnel et Livrables

- 9.1. Phases du Projet (Exemple de découpage)
- 9.2. Jalons Clés (Milestones)
- 9.3. Livrables Finaux

10. Aspects Opérationnels (Déploiement & Maintenance)

- 10.1. Procédure de Déploiement Cible
- 10.2. Documentation Requise
- 10.3. Maintenance Post-Projet

1. Définitions et Périmètre du Projet

1.1. Scénario / Manifeste

Hélène, directrice des études à l'IUT, prépare une réunion pour le rectorat. On lui demande : « Quel est le taux de réussite de vos étudiants en BUT INFO depuis la réforme ? Combien d'étudiants de BUT1 SD se réorientent ? ».

Avant, elle devait demander à la scolarité de compiler manuellement des fichiers Excel, ce qui prenait 3 jours. Aujourd'hui, elle ouvre **Mnemosyne**. Elle sélectionne "BUT INFO" et l'année "2022/2023". En un coup d'œil, elle voit le diagramme Sankey : 104 entrées, 100 passages en BUT2, 11 abandons. Elle clique sur "Réorientation" (4 étudiants) pour voir d'où ils venaient. Elle a sa réponse en 5 minutes.

1.2. Contexte et Problématique

- **Contexte :** La réforme des BUT (Bachelor Universitaire de Technologie), initiée en septembre 2021, a complexifié le suivi des parcours (compétences, SAE, ressources).
- **Problématique :** Un besoin institutionnel (IUT, rectorat, ministère) est apparu pour un suivi structuré, fiable et rapide des cohortes d'étudiants.

1.3. Objectif Principal (Le "Cœur de Métier")

Le projet **Mnemosyne** vise à développer une plateforme web qui :

- **Centraliser** les données de scolarité depuis l'API ScoDoc.
- **Visualiser** les flux d'étudiants (parcours) à l'aide de diagrammes dynamiques.
- **Permettre** un suivi facile et rapide des cohortes.

1.4. Périmètre du MVP (Inclus vs. Exclus)

Fonctionnalités INCLUSES dans le MVP (v1) :

- **Consultation :**
 - Filtres par Formation (les 6 BUT + "Tout l'IUT").
 - Filtre par Année de promotion.
 - Visualisation via Diagramme Sankey.
 - Visualisation des statuts (ADM, RED, etc.).
 - Interactivité (clic sur un flux pour suivre une sous-cohorte).
- **Administration :**
 - Authentification des administrateurs.
 - Bouton de synchronisation manuelle (important uniquement les *nouvelles* données).
 - Interface de "mapping" (Code ScoDoc -> Libellé).
 - Interface de définition des "scénarios" (règles de calcul).

Fonctionnalités EXCLUES (pour V2 ou V3) :

- Pas d'export PDF/CSV des graphiques.
- Pas de tableau de bord personnalisé pour les enseignants.
- Pas de comparaison directe entre les formations (ex: "BUT INFO vs BUT GEA").
- Pas de synchronisation automatique (cron job).
- Pas de gestion fine des droits (un seul rôle "Admin").

1.5. Glossaire Métier (Terminologie)

- **Cohorte** : Ensemble d'étudiants commençant une formation donnée la même année (ex: "Cohorte BUT1 INFO 2022/2023").
- **ScoDoc** : API de gestion de la scolarité, source de toutes les données.
- **Diagramme Sankey** : Visualisation des flux privilégiée pour montrer les parcours (réussite, redoublement, abandon...). C'est la vue principale de l'outil.
- **Mapping** : Action de lier un code technique (ex: B1-INFO-FI) à un libellé clair (ex: BUT1 INFO FI).
- **Scénario** : Règle métier définie par l'admin pour gérer un cas de parcours complexe.
- **Statuts ScoDoc (exemples) :**
 - ADM : Admis.
 - PASD : Non validé.
 - RED : Redoublant.
 - NAR : Abandon.
 - ADJ : Validation par décision de jury.

2. Modèle de Valeur et Stratégie

2.1. Proposition de Valeur

Ce projet n'a pas de but lucratif. Sa valeur est opérationnelle :

- **Gain de temps** : Réduit le temps d'analyse de plusieurs jours à quelques minutes.
- **Aide à la décision** : Fournit des données fiables pour le pilotage pédagogique et stratégique.
- **Transparence** : Permet de répondre rapidement aux demandes du rectorat et du ministère.

2.2. Stratégie d'Adoption

Pour que l'outil soit utilisé, il doit être :

- **Fiable** : La synchronisation des données doit être parfaite. Les chiffres affichés doivent être 100% justes.
- **Simple** : L'interface de consultation doit être intuitive et ne nécessiter aucune formation.
- **Rapide** : (Voir 5.4 - Indicateurs de Performance).

2.3. Coûts et Ressources

- **Budget (Financier)** : 0€ (Projet étudiant).
- **Ressources (Temps)** : Le budget principal est le temps de développement.
 - **Estimation** : [X] jours-personne (à définir par l'équipe, ex: 30 jours-personne si 2 personnes travaillent 3 semaines à plein temps).
 - Ce temps couvre les phases de la conception au déploiement (voir Section 9).
- **Hébergement** : Pris en charge par l'IUT (voir 5.3).
- **Maintenance** : La maintenance post-projet n'est pas couverte par ce budget et sera transférée à l'IUT (voir 10.3).

3. Utilisateurs et "User Stories"

3.1. Profils Utilisateurs

Profil	 Administrateur	 Consultant
Rôle	Personnel de la scolarité, direction IUT (côté technique).	Équipes pédagogiques, direction IUT (côté analyse), rectorat, ministère.
Besoin	Configurer l'application, garantir la fiabilité des données.	Consulter, filtrer et analyser les données de suivi.
Accès	Interface d'administration sécurisée.	Interface de consultation.

3.2. User Stories (Fonctionnalités MVP)

En tant qu'utilisateur "Consultant" :

- Je veux pouvoir sélectionner une **formation** (parmi les 6 BUT) et une **année de promotion**, **afin d'isoler mon analyse à un département spécifique**.
- Je veux pouvoir sélectionner "**Tout l'IUT**", **afin d'avoir une vue d'ensemble de la performance de l'établissement**.
- Je veux voir un **diagramme Sankey** affichant les flux de la cohorte sélectionnée, **afin de comprendre visuellement les parcours, les succès et les points de sortie**.
- Je veux voir le **détail des statuts** (ADM, PASD, RED...) pour une année donnée, **afin de quantifier les décisions de jury**.
- Je veux pouvoir cliquer sur un flux (ex: "Redoublants BUT1"), **afin de suivre le parcours**

de cette sous-cohorte spécifique l'année suivante.

- Je veux pouvoir cliquer sur un lien pour voir le détail d'un étudiant dans ScoDoc (si faisable), **afin d'accéder à la source de l'information en cas de besoin.**

En tant qu'utilisateur "Administrateur" :

- Je veux un **accès sécurisé** (login/mot de passe) à la partie administration, **afin d'empêcher toute modification non autorisée des règles.**
- Je veux un **bouton "Synchroniser"** qui importe les nouvelles données depuis ScoDoc, **afin de garantir que les données de consultation sont à jour.**
- Je veux que la synchronisation **ne crée pas de doublons**, **afin de maintenir l'intégrité de la base de données.**
- Je veux une interface simple pour **définir les "scénarios"** (règles de calcul), **afin d'adapter la logique de l'outil à la réalité du terrain et aux cas particuliers.**
- Je veux une interface simple pour **mapper les codes** ScoDoc (ex: B1-GEA-FI), **afin de rendre les graphiques lisibles pour des utilisateurs non techniques.**

4. Approfondissement Métier et Technique

4.1. Logique de Synchronisation

- La synchronisation doit être **idempotente** (cliquer 10 fois sur le bouton a le même effet que de cliquer 1 fois).
- Le système doit stocker un "marqueur" (timestamp de dernière synchro ou ID de la dernière entrée ScoDoc) pour ne récupérer que les données *strictement nouvelles*.
- Les données importées ne sont **jamais modifiées** (immuabilité) pour garantir l'historique.

4.2. Scénario de Parcours Complex (Exemples)

L'interface d'administration doit permettre de définir ces règles :

- **Règle 1 (Passerelle Interne) :**
 - Si un étudiant passe de BUT1 SD à BUT2 Passerelle.
 - Alors le flux doit être compté dans le Sankey comme "Passerelle".
- **Règle 2 (Changement de Statut) :**
 - Si un étudiant passe de BUT2 GEA FI (Formation Initiale) à BUT3 GEA FA (Apprentissage).
 - Alors l'administrateur doit pouvoir choisir si cela compte comme "Flux Standard" ou "Passerelle FI->FA".

4.3. Structure de Données (Conceptuelle)

Une réflexion préliminaire sur la structure de la base de données (SQL) identifie les entités suivantes :

- **Promotions** (ex: id, annee_debut, libelle -> "2022-2023")
- **Formations** (ex: id, code_scodoc, libelle_court -> "INFO", libelle_long -> "BUT Informatique")
- **Etudiants** (ex: id_scodoc, nom_anonymise)
- **Parcours_Annuel** (ex: id, etudiant_id, formation_id, promotion_id, statut_scodoc, annee_but -> 1, 2, ou 3)
- **Mapping_Codes** (ex: code_scodoc, libelle_graphique)
- **Regles_Scenario** (ex: code_source, code_cible, type_flux -> "Passerelle")
- **Utilisateurs_Admin** (ex: login, hash_mot_de_passe)

4.4. Architecture Applicative (Conceptuelle)

L'application suivra une architecture web classique à 3 niveaux, avec une connexion à une API externe :

1. **Client (Navigateur)** : HTML/CSS/JS - Interface utilisateur pour la consultation et l'administration.
2. **Serveur (Backend)** : PHP - Reçoit les requêtes du client, gère la logique métier (calculs, scénarios) et dialogue avec la BDD.
3. **Base de Données (BDD)** : SQL - Stocke les données de parcours, les mappings et les utilisateurs.
4. **API Externe** : ScoDoc - Le serveur PHP contacte cette API (via le bouton "Synchroniser") pour importer les données brutes.

5. Impératifs et Contraintes

5.1. Contraintes Techniques (Stack)

- **Frontend (Client)** : HTML5, CSS3, JavaScript (JS) (Aucun framework type React/Vue n'est demandé).
- **Backend (Serveur)** : PHP (Préférence M. Hébert, à débattre).
- **Base de Données** : SQL (ex: MySQL/MariaDB). La plateforme aura sa propre BDD.

5.2. Contraintes de Données

- **Source de Données :** L'application DOIT se connecter à l'API **ScoDoc** (<https://scodoc.org/ScoDoc9API/>).
- **Données de Test :** Le développement se fera **UNIQUEMENT** avec des fichiers **JSON** ("fausses données") fournis par M. Hébert.
- **Confidentialité :** L'accès à l'API ScoDoc de production ne sera possible que lors des tests finaux avec M. Hébert.

5.3. Contraintes Opérationnelles et Risques

- **Hébergement :** L'application devra être hébergée sur un serveur géré par l'IUT (probablement infrastructure PHP/MySQL existante).
- **Langue :** L'interface doit être en **Français** uniquement pour le MVP.

Risque Identifié	Niveau	Stratégie de Mitigation
Risque 1 : Les "fausses données JSON" de test ne sont pas complètes ou représentatives.	Fort	<ol style="list-style-type: none">1. Définir un "schéma JSON" (contrat de données) que M. Hébert doit valider.2. Le développement se basera sur ce schéma.
Risque 2 : La complexité de l'API ScoDoc (authentification, pagination) est sous-estimée.	Moyen	<ol style="list-style-type: none">1. Planifier un "Spike" (tâche de découverte) de 1-2 jours en début de projet, dédié à l'authentification et à un premier appel à l'API.
Risque 3 : Un "scénario" de parcours étudiant est oublié, faussant les calculs.	Faible	<ol style="list-style-type: none">1. L'interface d'admin (User Story) est la mitigation principale.2. Planifier un atelier avec M. Hébert pour lister les 5 scénarios les plus courants avant de coder l'interface.

5.4. Exigences Non-Fonctionnelles (Performance)

- **Temps de réponse (Consultation)** : L'affichage du diagramme Sankey après sélection des filtres doit être **inférieur à 3 secondes**.
- **Temps de réponse (Admin)** : La synchronisation manuelle (avec données JSON de test) doit fournir un retour utilisateur en **moins de 10 secondes**.
- **Scalabilité (MVP)** : L'application doit supporter **5 utilisateurs concurrents** en consultation sans dégradation notable des performances.

5.5. Exigences Non-Fonctionnelles (Sécurité)

- **Authentification** : Les mots de passe des Utilisateurs_Admin doivent être hachés (ex: password_hash() en PHP).
- **Protection (Backend)** : Le backend doit valider et nettoyer ("sanitize") toutes les entrées utilisateur pour prévenir les injections SQL (via des requêtes préparées) et les failles XSS (Cross-Site Scripting).
- **Communication** : L'accès à l'API ScoDoc (serveur à serveur) doit utiliser les jetons d'authentification requis par l'API.

6. Éléments Graphiques et Identité

6.1. Identité

- **Nom : Mnemosyne.**
 - *Origine* : Déesse grecque de la mémoire, gardant la trace du passé pour nourrir le savoir.
- **Slogan (à débattre)** : « Garder la mémoire, éclairer les parcours ».
- **Logo (à débattre)** : Proposition de logo fournie par M. Hébert.

6.2. Charte Graphique et Inspiration

- **Inspiration Principale** : Le diagramme Sankey de l'exemple fourni par M. Hébert.
- **Ambiance** : Professionnelle, sobre, "data-driven". Doit ressembler à un outil d'analyse institutionnel.
- **Couleurs Proposées :**
 - **Bleu IUT (Principal)** : #004A99 (Bleu institutionnel)
 - **Gris Fond (Background)** : #F4F7FA (Gris très clair)
 - **Texte (Principal)** : #222222 (Gris foncé)
 - **Succès (ADM)** : #28A745 (Vert)
 - **Alerte (RED)** : #DC3545 (Rouge)
 - **Info (Passerelle)** : #17A2B8 (Cyan)
 - **Neutre (NAR)** : #6C757D (Gris moyen)

7. Critères d'Acceptation (Definition of Done)

L'application (MVP v1) sera considérée comme "terminée" et prête pour la validation lorsque tous les critères suivants sont remplis :

1. **Validation Données** : Les données issues du fichier JSON de test sont affichées correctement dans le diagramme Sankey et le graphique des statuts.
2. **Validation Fonctionnelle (Consultant)** : Toutes les User Stories de la Section 3.2 pour le "Consultant" sont fonctionnelles.
3. **Validation Fonctionnelle (Admin)** : Un administrateur peut se connecter, créer un Mapping et une Regle_Scenario (correspondant à Regles_Scenario), et voir le changement (simulé) sur le graphique.
4. **Validation Scénario** : Le scénario 1.1 (Manifeste) peut être réalisé par M. Hébert sur l'application en moins de 5 minutes.
5. **Validation Performance** : Les indicateurs de la Section 5.4 sont respectés lors des tests.
6. **Validation Technique** : L'application est déployée sur l'infrastructure d'hébergement cible (Section 5.3) et la synchronisation (Section 4.1) fonctionne avec les données de test.

8. Stratégie de Test

Pour garantir la qualité et la fiabilité (critère 2.2), le plan de test suivant sera appliqué.

- **Tests Unitaires (TU) :**
 - **Quoi** : Fonctions PHP critiques (ex: logique de calcul des scénarios, fonction de synchronisation "idempotente").
 - **Outil** : PHPUnit (ou équivalent).
- **Tests d'Intégration (TI) :**
 - **Quoi** : Tester la chaîne complète "API -> Synchronisation -> BDD -> Affichage".
 - **Scénario** : 1. Démarrer avec une BDD vide. 2. Lancer la synchro (sur JSON). 3. Vérifier que la BDD contient les bonnes données. 4. Vérifier que l'interface de consultation affiche le bon graphique.
- **Tests Fonctionnels (Manuels) :**
 - **Quoi** : Suivre toutes les User Stories (3.2) et les Critères d'Acceptation (7).
 - **Qui** : L'équipe de développement.
- **Tests d'Acceptation Utilisateur (UAT) :**
 - **Quoi** : Validation finale du projet.
 - **Qui** : M. Hébert.
 - **Scénario** : M. Hébert exécute le Scénario 1.1 et valide que le produit correspond aux attentes.
- **Tests de Charge (MVP - Exclus)** : Les tests de charge (simulant 50+ utilisateurs) sont hors périmètre pour le MVP.

9. Planning Prévisionnel et Livrables

9.1. Phases du Projet (Exemple de découpage)

- **Phase 1 : Conception & Spécifications (S+0 - S+1)**
 - Ateliers M. Hébert, validation du schéma JSON, design BDD.
 - *Livrable* : Schéma BDD final, validation des données de test.
- **Phase 2 : Développement Backend & Admin (S+2 - S+4)**
 - Mise en place BDD, authentification, interface admin (mapping, scénarios), logique de synchronisation (sur JSON).
- **Phase 3 : Développement Frontend & Visualisation (S+4 - S+6)**
 - Interface de consultation, filtres, génération des graphiques Sankey et statuts.
- **Phase 4 : Tests & Intégration (S+7)**
 - Tests Unitaires, Tests d'Intégration, connexion à l'API ScoDoc réelle (avec M. Hébert), corrections.
 - *Livrable* : Application testée et validée (fonctionnellement).
- **Phase 5 : Déploiement & Rendu (S+8)**
 - Mise en production sur le serveur IUT, rédaction de la documentation finale, UAT par M. Hébert.
 - *Livrable* : Projet livré.

9.2. Jalons Clés (Milestones)

- **J1 (Fin S+1)** : Données JSON de test et schéma BDD validés par M. Hébert.
- **J2 (Fin S+4)** : L'interface d'administration est fonctionnelle (avec données de test).
- **J3 (Fin S+6)** : L'interface de consultation affiche les graphiques (avec données de test).
- **J4 (Fin S+8)** : L'application est déployée et validée (Critères d'Acceptation Section 7 remplis).

9.3. Livrables Finaux

1. Le code source complet et commenté du projet (dépôt Git).
2. L'application déployée et fonctionnelle sur l'infrastructure de l'IUT.
3. La documentation technique et utilisateur (voir 10.2).

10. Aspects Opérationnels (Déploiement & Maintenance)

10.1. Procédure de Déploiement Cible

1. **Prérequis Serveur** : Serveur web (Apache/Nginx), PHP (version à définir), base de données SQL.
2. **Installation** : git clone du dépôt sur le serveur.
3. **Configuration** : Création d'un fichier config.php (hors Git) pour les accès BDD et les clés API ScoDoc.
4. **Base de Données** : Importation du fichier schema.sql pour créer la structure de la BDD.
5. **Finalisation** : Attribution des droits d'écriture (si nécessaire) et configuration du vhost.

10.2. Documentation Requise

- **Documentation Technique (README.md)** : Destinée aux futurs développeurs. Doit inclure :
 - Instructions d'installation (voir 10.1).
 - Liste des variables d'environnement (fichier config.php).
 - Description de la structure de la BDD (MCD).
- **Documentation Utilisateur (MANUEL_ADMIN.md)** : Destinée à l'IUT (Admin). Doit inclure :
 - Comment se connecter à l'administration.
 - Comment utiliser le "mapping" des codes.
 - Comment définir un nouveau "scénario".

10.3. Maintenance Post-Projet

Comme défini en Section 2.3, ce projet est livré "tel quel" à la date du rendu (Jalon J4). La maintenance corrective (bugs) et évolutive (V2) sera transférée aux équipes techniques de l'IUT. La documentation (10.2) sert de document de handover officiel.