

RAPPORT

PROJET DEVELOPPEMENT :

GESTION

DEMATÉRIALISÉE DES

INSCRIPTIONS

JUIN 2025

LICENCE 3 MIAGE

Enseignants encadrants : Gilbert RINALDO
Wilfried SEGRETIER

Nom du groupe : 4CODERS

Membres du groupe : CHARINI Jordan, KITTAVINY Wilma,
MCLEAN Tyrece, SAVONNIER Océane

Année scolaire : 2024/2025

Table des matières

1. Spécifications et conception du besoin (Phase de définition)	
1.1 Contexte et objectifs	3
1.2 Expression des besoins fonctionnels	3
1.3 Interfaces utilisateurs clés	4
2. Conception technique (Phase de spécification)	
2.1 Architecture retenue	6
2.2 Définition des modules principaux	6
3. Réalisation et intégration (Phase de développement et validation)	
3.1 Développement	7
3.2 Recette fonctionnelle	8
3.3 Mise en production	9
4. Exploitation et retours (Phase de déploiement)	
4.1 Suivi des objectifs	9
4.2 Évaluation des retours utilisateurs	9
5. Environnement technique et outils utilisés	
5.1 Communication et coordination d'équipe	10
5.2 Tableau des choix technologiques	10
6. Diagrammes de planification	
.....	12
7. Cas d'utilisation UML	
.....	13
8. Plan de développement initial	
.....	14
9. Diagramme d'activité UML	
.....	15
10. Diagramme de classe UML	
.....	16
11. Modélisation MERISE	
.....	17
12. Charte graphique du projet	
.....	17

1. SPÉCIFICATIONS ET CONCEPTION DU BESOIN (PHASE DE DÉFINITION)

Durant ce projet, nous avons utilisé un cycle en V, un modèle de développement structuré qui offre une grande clarté dans l'enchaînement des phases, ainsi qu'une traçabilité efficace entre les étapes de conception et de validation. Ce choix s'est imposé naturellement, car les besoins fonctionnels étaient clairement définis dès le départ. Le cycle en V garantit un bon alignement entre la spécification des besoins et les phases de test, assurant ainsi la qualité et la conformité du produit final.

1.1 CONTEXTE ET OBJECTIFS

Le projet INSCRIPTIONS a été lancé dans le but de moderniser et simplifier le processus de candidature à la Licence 3 MAGE, gérée en partenariat avec le GRETA.

L'objectif principal était de remplacer le système manuel d'envoi de dossiers par une plateforme numérique centralisée, plus fiable, plus rapide et plus ergonomique. Cette solution devait respecter les exigences suivantes :

- Conformité au RGPD (protection des données personnelles)
- Accessibilité et ergonomie (notamment sur mobile)
- Réduction des erreurs humaines et des retards de traitement
- Intégration des contraintes pédagogiques et administratives

1.2 EXPRESSION DES BESOINS FONCTIONNELS

Les besoins fonctionnels identifiés pour la plateforme sont les suivants :

- Portail candidat accessible depuis un navigateur web ou mobile
- Formulaires dynamiques pour la saisie des informations personnelles, du parcours scolaire et professionnel
- Téléversement sécurisé des pièces justificatives (CV, lettre de motivation, etc.)
- Suivi en temps réel de l'état d'avancement de la candidature
- Interface de gestion pour les utilisateurs du GRETA et du bureau MAGE
- Moteur de statuts configurable permettant de personnaliser les étapes du processus
- Notifications automatiques (accusés de réception, changements de statut, relances)
- Exports de données au format PDF et Excel pour les besoins de reporting et d'archivage

1.3 INTERFACES UTILISATEURS CLÉS

Interfaces Candidat

Titre de l'Interface	Description	Composants UI
Tableau de bord Candidat	Vue récapitulative personnalisée pour le candidat : statut actuel du dossier, étapes restantes, alertes/notifications, accès rapide aux formulaires, historique des actions.	CarteStatutDossier, BarreProgressionÉtapesTableauHistoriqueActions, BadgeNotification, BoutonAccèsRapide, CarteStatutGlobal,
Formulaire de préinscription	Interface de saisie initiale permettant au candidat de renseigner ses informations de base : coordonnées, informations générales.	FormulaireCoordonnées, ChampTexte, ChampTéléphone, ChampEmail, BoutonSoumettre, NotificationValidation
Formulaire de scolarité	Saisie détaillée du parcours académique : établissements fréquentés, diplômes obtenus, notes, spécialisations.	FormulaireScolarité, ChampTexte, TableauDiplômes, ChampDate, BoutonAjouterDiplôme, BoutonSoumettre, NotificationErreurValidation
Formulaire parcours professionnel	Interface dédiée à la saisie de l'expérience professionnelle : emplois précédents, stages, compétences acquises, projets réalisés.	FormulaireParcoursPro, TableauExpériences, ChampCompétences, BoutonAjouterExpérience, BoutonSupprimerExpérience, BoutonSoumettre

Récapitulatif et validation finale	Page de synthèse permettant au candidat de vérifier toutes ses informations avant soumission définitive.	CarteRécapitulatifDossier, TableauSynthèse, BoutonValidationFinaleNotificationSoumission
---	--	--

Interfaces Bureau MIAGE

Titre de l'Interface	Description	Composants UI
Espace Bureau MIAGE - Dashboard	Vue d'ensemble pour le personnel administratif : liste des dossiers à traiter, statistiques des inscriptions, alertes sur les dossiers en attente.	TableauDossiersÀTraiter, CarteStatistiquesInscriptions, CarteAlertesDossiers, BarreRechercheCandidat, MenuFiltres
Fiche candidat complète	Interface détaillée d'un dossier candidat : consultation de toutes les informations (coordonnées, parcours), pièces justificatives, historique des modifications.	CarteCoordonnées, TableauParcours, GestionnairePiècesJustificatives, HistoriqueModifications, OutilAnnotation
Interface de validation	Écran permettant de valider/rejeter un dossier, d'ajouter des commentaires, de demander des pièces complémentaires.	TableauDossiers, BoutonValiderDossier, BoutonRejeterDossier, ChampCommentaire, NotificationDemandePièce
Gestionnaire de pièces justificatives	Interface de téléchargement et gestion des documents : diplômes, relevés de notes, CV, lettre de motivation, pièces d'identité.	TableauPièces, ChampUploadDocument, BarreStatutDocument, BoutonTélécharger, NotificationDocumentValidé

Interfaces Client (Greta)

Titre de l'Interface	Description	Composants UI
Interface Client Greta - Consultation	Accès en lecture seule pour les Clients Greta : consultation des dossiers d'étudiants, export de	TableauDossiersGreta, BoutonExporterDonnées, CarteSuiviInscriptions,

	données, suivi des inscriptions.	BarreRecherche, BadgePermissions
--	----------------------------------	----------------------------------

Interfaces Communes

Titre de l'Interface	Description	Composants UI
Gestionnaire de statuts et workflow	Interface pour suivre et modifier les statuts des dossiers : visualisation du workflow complet, gestion des cas particuliers.	DiagrammeWorkflow, TableauStatuts, BoutonModifierStatut, NotificationActionStatut
Tableau de bord statistiques	Interface de reporting accessible aux administrateurs : statistiques d'inscription, taux de validation, export de données.	GraphiquesStatistiques, TableauStatistiques, BoutonExporterRapport, BarreFiltresStatistiques
Gestion des coordonnées et profils	Interface pour modifier les informations personnelles et gérer les comptes utilisateurs.	FormulaireModificationCoordonnées, TableauUtilisateurs, BoutonRéinitialiserMotDePasse, BoutonModifierProfil, NotificationGestionProfil

2. CONCEPTION TECHNIQUE (PHASE DE SPÉCIFICATION)

2.1 ARCHITECTURE RETENUE

L'architecture technique du projet a été définie selon des critères de performance, de compatibilité multiplateforme, de sécurité et de maintenabilité. Chaque choix technologique vise à garantir la robustesse de la plateforme tout en facilitant son évolution.

- **React Native** : utilisé pour le développement de l'application mobile (iOS et Android) avec une base de code unique. Il offre des performances proches du natif et bénéficie d'un écosystème riche et actif, favorisant l'évolutivité.
- **Node.js** : retenu pour le backend en raison de sa légèreté et de sa capacité à gérer un grand nombre de connexions simultanées. Il est associé à Express.js pour sa modularité.
- **API REST** : architecture utilisée pour assurer l'interopérabilité entre le frontend et le backend. Elle est standardisée, facile à documenter, sécuriser et maintenir.
- **Authentification JWT (JSON Web Tokens)** : permet une gestion sécurisée des rôles utilisateurs (candidat, gestionnaire, GRETA), adaptée aux environnements multi-accès.
- **Hébergement sécurisé** : sur serveur Linux avec HTTPS, garantissant la confidentialité des données personnelles conformément au RGPD.

Résumé technique :

- **Frontend mobile** : React Native
- **Backend** : Node.js avec Express.js
- **Communication** : API REST
- **Authentification** : JWT avec gestion des rôles
- **Hébergement** : Serveur sécurisé (HTTPS, Linux) – conformité RGPD

2.2 DÉFINITION DES MODULES PRINCIPAUX

La plateforme a été conçue autour de modules fonctionnels bien définis :

- **Module candidat** : formulaire d'inscription + suivi du dossier en temps réel
- **Module gestionnaire** : tableau de bord pour le traitement des candidatures
- **Moteur de statuts** : système configurable pour gérer les transitions d'un dossier
- **Module de notifications** : envois automatiques d'e-mails aux différentes étapes
- **Outils d'export et de reporting** : génération de fichiers PDF et Excel
- **Calendrier dynamique** : gestion des campagnes d'inscription avec périodes paramétrables

3. RÉALISATION ET INTÉGRATION (PHASE DE DÉVELOPPEMENT ET VALIDATION)

3.1 Développement

La phase de développement s'est déroulée en mode **collaboratif**, avec une répartition claire des rôles et une communication continue entre les membres de l'équipe.

Outils utilisés

- **Figma** : utilisé par Océane et Wilma pour la création des maquettes et prototypes des interfaces.
- **VSCode** : environnement principal de développement pour le front-end et le back-end, utilisé par Jordan et Tyrece, avec intégration Git.
- **Cursor** : outil de collaboration en temps réel, utilisé pour partager et corriger du code ensemble, notamment lors de phases de debugging.
- **GitHub** : gestion du versioning. Chaque membre effectuait des commits réguliers pour assurer la traçabilité du projet.
- **Microsoft Teams** : principal canal de communication. L'équipe y échangeait du code, des solutions, et s'entraidait en direct.

Organisation du travail

- **Front-end** : Océane et Wilma se sont chargées de l'intégration des maquettes Figma sous **React Native**, avec des allers-retours fréquents pour affiner les écrans.
- **Back-end** : Jordan et Tyrece ont développé l'API sous **Node.js/Express** et structuré la base de données. Chaque fonctionnalité (authentification, statuts, exports, etc.) a été développée en parallèle puis intégrée.
- **Partage de code** : Cursor et Teams ont permis de résoudre les blocages rapidement via du debug collaboratif ou des revues de code.
- **Versioning** : L'ensemble du code était synchronisé sur GitHub, avec des push réguliers pour maintenir une cohérence d'équipe.

Workflow typique

1. Les tâches étaient réparties sur **Trello**.
2. Après validation du design sur Figma, l'intégration commençait côté front.
3. Les fonctionnalités back-end étaient développées et testées localement.
4. Le code était versionné sur **GitHub**.
5. Les éventuels problèmes étaient discutés et résolus en équipe via **Teams** ou **Cursor**.

Gestion des difficultés

Des ajustements ont été nécessaires pour assurer la compatibilité entre le front et le back (notamment au niveau des formats de données).

Grâce à GitHub, les conflits de version ont pu être corrigés rapidement, permettant un travail fluide malgré la répartition à distance.

3.2 RECETTE FONCTIONNELLE

Une phase de recette a été réalisée par l'équipe projet afin de valider la conformité de la plateforme aux besoins exprimés. Les vérifications ont porté sur :

- La validation du parcours candidat, du dépôt initial jusqu'à la décision finale
- Le traitement automatisé des statuts, y compris les cas particuliers
- La gestion des exceptions, via la simulation de cas réels (dossiers incomplets, erreurs de saisie, etc.)

3.3 MISE EN PRODUCTION

La mise en production est prévue pour le 4 juillet 2025, date de fin du projet. La plateforme est aujourd'hui terminée et fonctionnelle.

Des tests complets ont été réalisés pour vérifier que tout fonctionne correctement :

- Dépôt d'une candidature
- Suivi du dossier
- Changement de statuts
- Envoi automatique d'e-mails
- Gestion des accès selon les rôles

La plateforme est prête à être utilisée dès sa mise en ligne.

4. EXPLOITATION ET RETOURS (PHASE DE DEPLOIEMENT)

4.1 Suivi des objectifs

Module	Intitulé du livrable	Date	Statut	Diffusion
1	Portail de candidature (web et mobile)	4 Juillet 2025	Terminé	Interne
2	Espace gestionnaire	4 Juillet 2025	Terminé	Interne
3	Moteur de gestion des statuts	4 Juillet 2025	Terminé	Interne
4	Notification et relance automatique	4 Juillet 2025	Terminé	Interne
5	Export de données (Excel / PDF)	4 Juillet 2025	Terminé	Interne
6	Gestion du calendrier	4 Juillet 2025	Terminé	Interne
7	Documentation utilisateur	4 Juillet 2025	Terminé	Interne

Tous les livrables ont été réalisés dans les délais et sont prêts pour utilisation.

4.2 Évaluation des retours utilisateurs

Même si la plateforme n'a pas encore été utilisée en situation réelle, les premiers **tests fonctionnels** et retours internes montrent des résultats positifs :

- **Taux de complétude des dossiers plus élevé** (moins d'oublis dans les formulaires)
- **Réduction des erreurs de saisie**, grâce aux champs dynamiques et validations automatiques
- **Traitement plus rapide des candidatures**, grâce au moteur de statuts et aux exports automatisés

5. ENVIRONNEMENT TECHNIQUE ET OUTILS UTILISES

5.1 Communication et coordination d'équipe

L'équipe projet était composée de **4 membres** avec des rôles bien définis :

Nom / Rôle	Missions principales
Tyrece – Chef de projet & Back-end	Planification, coordination, suivi des deadlines, intégration front/back, déploiement serveur
Océane – Dév. Front-end	Intégration UI mobile, tests d'ergonomie, affichage des présences
Wilma – Dév. Front-end	Interfaces utilisateur (React Native), parcours étudiant, intégration graphique
Jordan – Dév. Back-end	API Node.js, sécurité, base de données, authentification
4Coders – Référent fonctionnel	Cahier de tests, validation fonctionnelle, rapports de recette

Outils de coordination utilisés :

- **Microsoft Teams** : pour les réunions régulières et les échanges quotidiens.
- **Trello** : pour le suivi des tâches en mode Kanban (organisation des livrables, affectation, avancement, blocages).

Ces outils ont permis une **communication fluide**, une **meilleure organisation**, et un **suivi structuré** dans un projet court et intensif.

5.2 Tableau des choix technologiques

Le tableau ci-dessous résume les principales technologies utilisées, leur rôle et les raisons de leur choix :

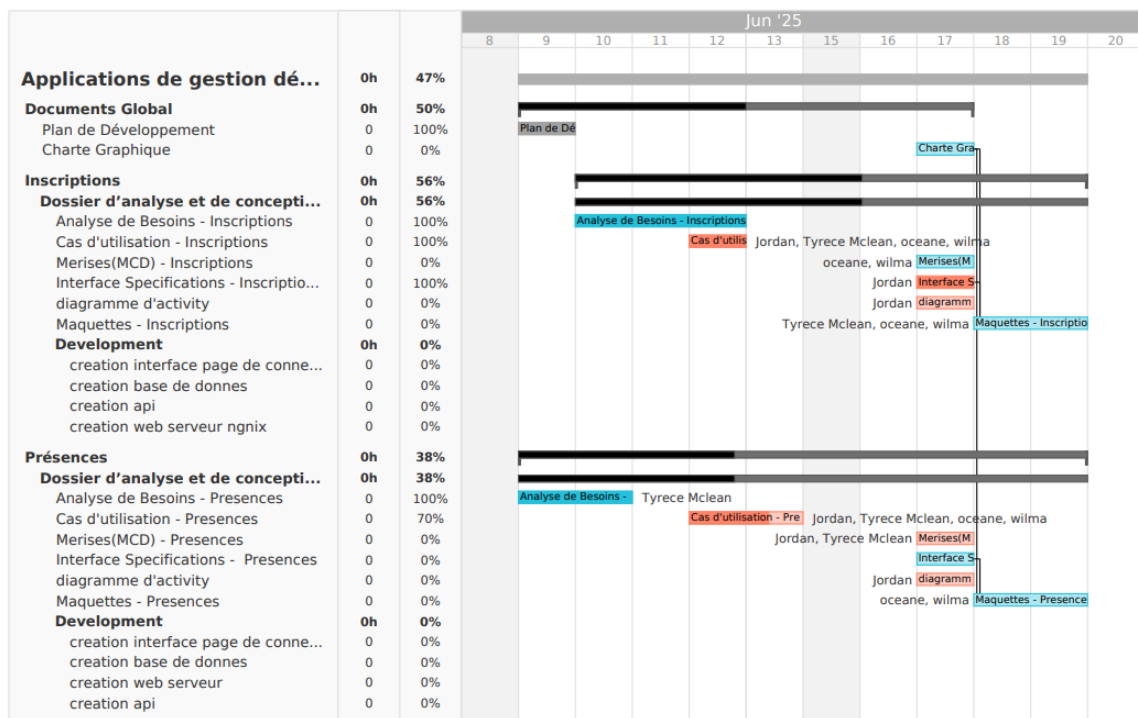
Technologie	Rôle dans le projet	Justification
React Native	Frontend mobile	Code unique pour iOS/Android, rapide à déployer
Expo Go	Frontend mobile	Simplifie le développement et le test
Android Studio	Développement Android	IDE officiel Android
Xcode	Développement iOS	IDE nécessaire pour iOS
Node.js	Backend / API	Léger, rapide, adapté aux applis temps réel
Express.js	Framework serveur	Simple, efficace pour les API REST
JWT	Sécurité / rôles	Authentification sécurisée par jetons
bcrypt	Sécurité	Hachage sécurisé des mots de passe
CORS	Sécurité	Autorise l'accès entre domaines
Redis	Cache / backend	Performant, utile pour les sessions
Microsoft SQL Server	Base de données principale	Fiable pour les données relationnelles
mssql	Connecteur base de données	Connexion entre Node.js et SQL Server
Docker	Conteneurisation	Déploiement stable et reproductible
Docker Compose	Orchestration	Coordination des services Docker
Nginx	Serveur web / proxy	Répartition des requêtes, reverse proxy
uuid	Génération d'ID	Création d'identifiants uniques
dotenv	Environnement	Gestion des variables dans des fichiers dédiés
body-parser	Middleware HTTP	Lecture des données envoyées au serveur
axios	Client HTTP	Requêtes vers API et services externes
RESTful API	Architecture API	Standard de communication claire et maintenable
teamGantt	Gestion de projet	Suivi des tâches et des délais
TypeScript	Langage frontend	Meilleure robustesse et lisibilité du code
SQLite	Base de données locale	Légère, idéale pour un usage hors-ligne

6.DIAGRAMME DE PLANIFICATION (GANTT)

Cette section présente un outil de planification utilisés dans le projet :

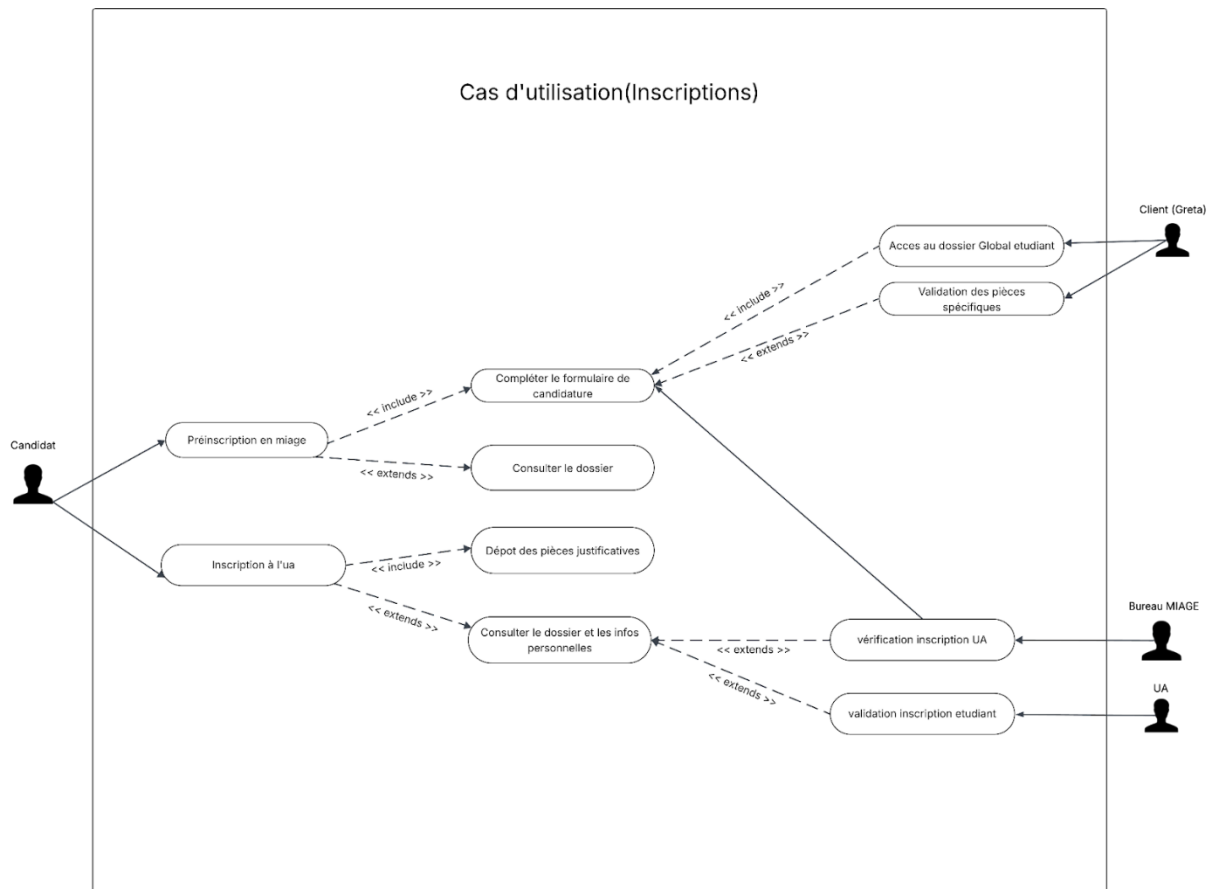
- Diagramme de Gantt : pour visualiser les tâches dans le temps.

Diagramme de GANTT



7.CAS D'UTILISATION UML

Cette section décrit les cas d'utilisation principaux du système sous forme de diagrammes UML.



8.PLAN DE DEVELOPPEMENT INITIAL

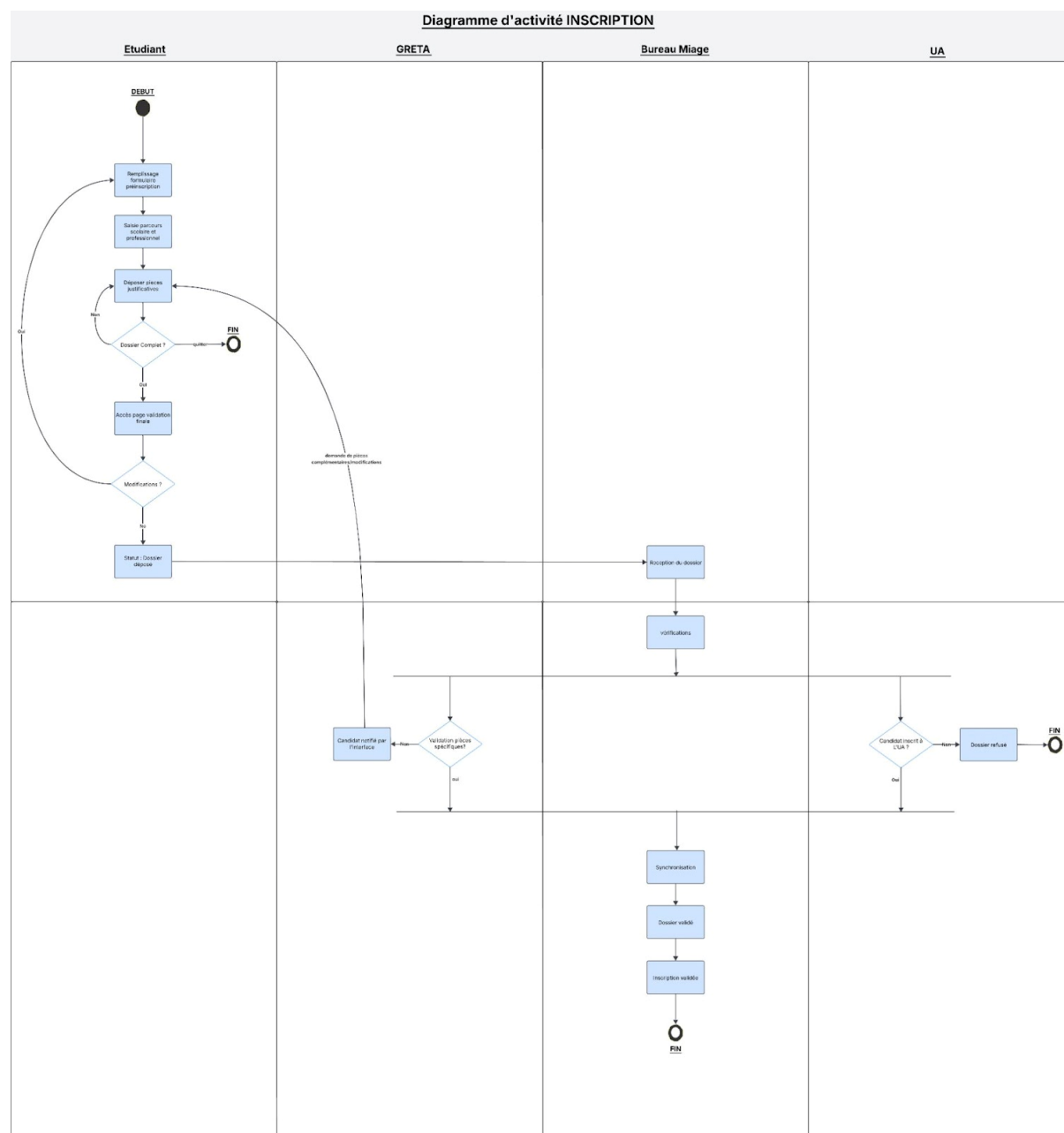
Le plan de développement initial a été défini avant le lancement du projet, afin d'assurer une organisation claire et un avancement fluide.

Le découpage a été réalisé selon les grandes phases classiques du cycle en V : analyse, conception, développement, tests/recette, déploiement.

Phase du projet	Période	Livrables associés
Analyse & cadrage	Semaine 1	Analyse de besoins, Expression fonctionnelle, Cahier des charges
Conception fonctionnelle	Semaine 1-2	Cas d'utilisation, MCD/MLD, Maquettes UI
Conception technique	Semaine 2	Architecture technique, Spécifications API
Développement Back-end	Semaine 2-3	Création base de données, Développement API Node.js, Authentification
Développement Front-end	Semaine 2-3	Développement interfaces React Native, Intégration UI, Notifications
Tests & Recette fonctionnelle	Semaine 4	Cahier de tests, Recette, Correction de bugs
Déploiement & Livraison	Fin Semaine 4	Mise en production, Documentation utilisateur

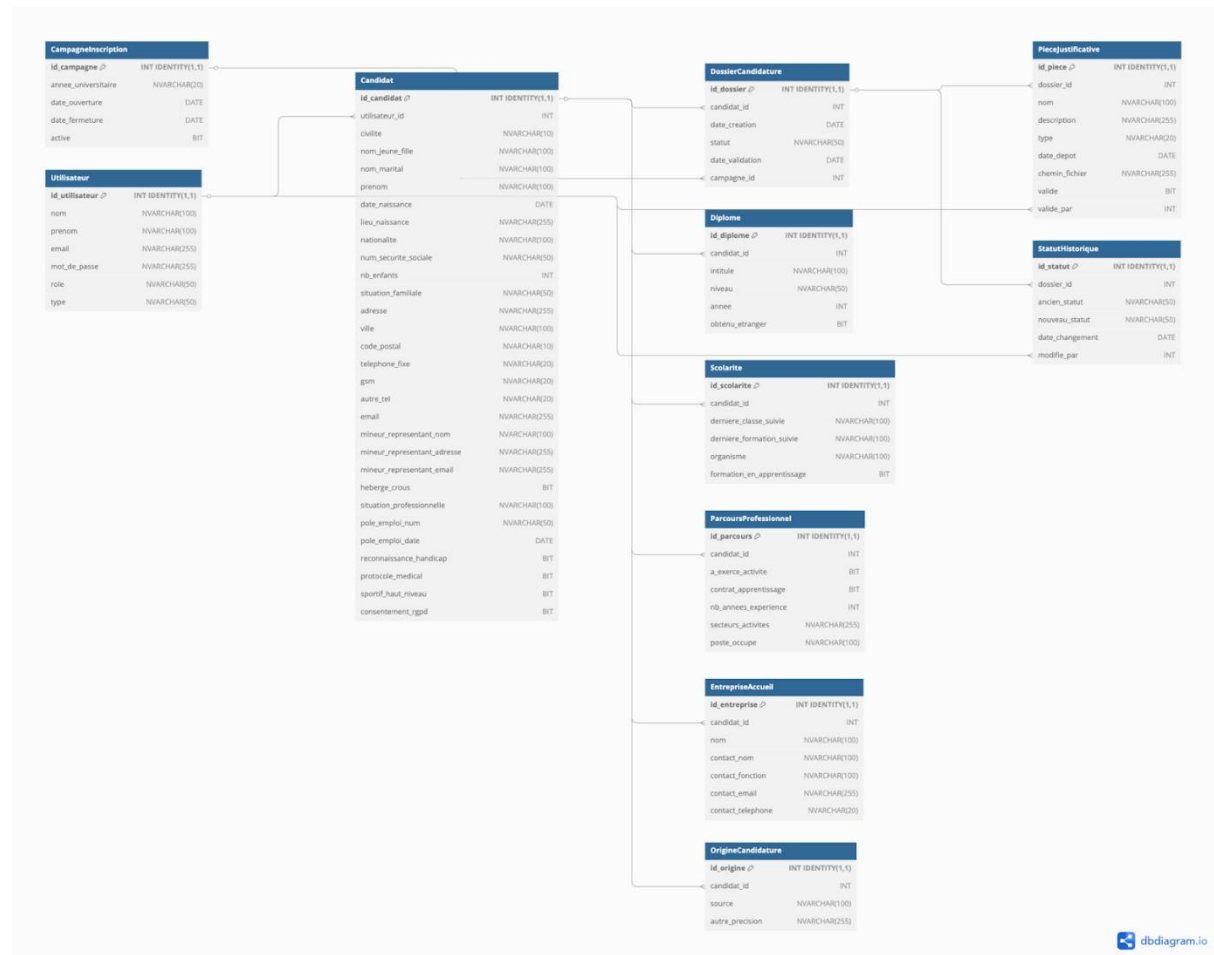
9.DIAGRAMME D'ACTIVITE UML

Cette section présente le diagramme d'activité décrivant les enchaînements d'actions et de décisions clés du processus de candidature.



10. DIAGRAMME DE CLASSES UML

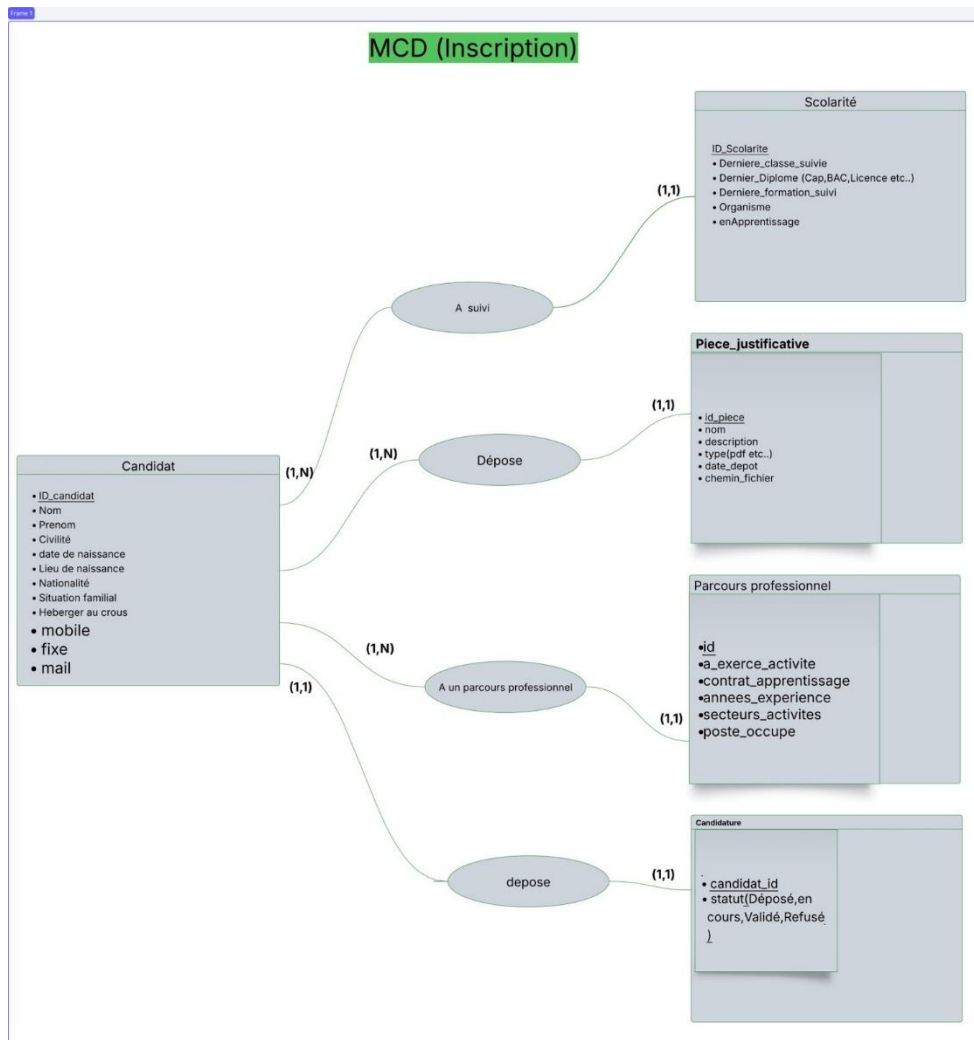
Cette section présente le diagramme de classes décrivant les entités principales du système et leurs relations.



11. MODELISATION MERISE (MCD)

Cette section présente les différentes étapes de modélisation selon Merise :

- MCD (Modèle Conceptuel de Données) : entités, relations, cardinalités



12.CHARTE GRAPHIQUE DU PROJET

Cette section décrit les choix graphiques appliqués à l'application : typographies, couleurs, icônes, hiérarchie visuelle, branding.

LOGO APP



LOGO TYPE



PALETTE DE COULEUR



#0C284F



#128FC9



#F06737



#A3A2A9

COULEURS

	HEX	RVB	CMJN
	#0C284F	12, 40, 79	85, 49, 0, 69
	#128FC9	18, 143, 201	91, 29, 0, 21
	#F06737	240, 103, 55	0, 69, 77, 6
	#A3A2A9	163, 162, 169	4, 4, 0, 34

PICTOGRAMME



POLICE

Aa

HEADER FONT
POPPINS

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

Aa

BODY TYPE
Nunito

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890

ICONOGRAPHIE

STYLE

- LIGNE SIMPLE ET ÉPURÉE, MINIMALISTE.
- ÉPAISSEUR UNIFORME POUR UNE COHÉRENCE VISUELLE (1.5 PX RECOMMANDÉ).

BIBLIOTHÈQUES UTILISÉES

- FLATICON: <https://www.flaticon.com>

TAILLE ET ÉPAISSEUR

- TAILLE STANDARD : 24X24 PX POUR DESKTOP, 20X20 PX POUR MOBILE.
- ÉPAISSEUR : 1.5 PX POUR LES TRAITS.

GRILLE

DESKTOP

- NOMBRE DE COLONNES : 12 COLONNES
- MARGE LATÉRALE : 80PX
- ESPACES ENTRE COLONNES : 24PX
- LARGEUR MAX. DU CONTENU : 1200PX

MOBILE

- NOMBRE DE COLONNES : 4 À 6 COLONNES
- MARGE LATÉRALE : 16 PX
- ESPACES ENTRE COLONNES : 12PX
- CONTENU FULL WIDTH: <480PX