

1. Contexte et Objectifs du Projet

1.1. Contexte général

Ce projet constitue l'évaluation principale du module « Processus de Développement Logiciel » (50% de la note finale). Il vise à mettre en application l'ensemble des compétences acquises : choix et justification d'un processus de développement, application des pratiques agiles, production d'artefacts, qualité du code et intégration continue.

Le domaine métier retenu est celui d'une agence de location de voitures disposant de plusieurs points de service. L'agence souhaite se doter d'un système ERP intégré couvrant l'ensemble de ses opérations : gestion du parc automobile, suivi des locations, gestion financière, et relation client.

1.2. Objectifs pédagogiques

- Appliquer un processus de développement complet (de l'inception à la livraison) sur un projet réel
- Justifier le choix du cycle de vie et des pratiques méthodologiques adoptées
- Produire les artefacts attendus : Product Backlog, User Stories, Sprint Backlog, Burndown Chart, documentation technique
- Développer un logiciel fonctionnel respectant les principes de Clean Code et d'architecture logicielle
- Mettre en place un pipeline CI/CD avec gestion de versions (Git workflow)
- Travailler en équipe selon les cérémonies SCRUM (ou le processus choisi et justifié)

1.3. Organisation des équipes

- Chaque équipe doit développer les deux plateformes décrites ci-dessous
- Attribution des rôles obligatoire : Scrum Master, Product Owner, Développeurs (avec rotation possible)
- Un dépôt Git unique par équipe (GitHub, GitLab ou équivalent) avec historique des contributions

2. Description des Deux Plateformes

Le système ERP est décomposé en deux plateformes complémentaires. Chaque équipe doit développer les deux plateformes, qui partagent la même base de données et communiquent entre elles.

PLATEFORME A — Back-Office (Administration & Gestion Interne)

Destinée aux employés et gestionnaires de l'agence. Cette plateforme constitue le cœur opérationnel du système ERP.

2.1. Plateforme A — Back-Office (Administration)

Module A1 : Gestion du Parc Automobile

- Fiche technique complète de chaque véhicule : immatriculation, marque, modèle, date de mise en circulation, kilométrage, description de l'entretien, date du dernier entretien, prochain vidange
- Suivi des dates d'expiration : assurance, contrôle technique, amortissement
- Gestion des statuts : disponible, en location, en maintenance, transféré, hors service
- Gestion des disponibilités en temps réel avec calendrier visuel

Module A2 : Gestion des Locations

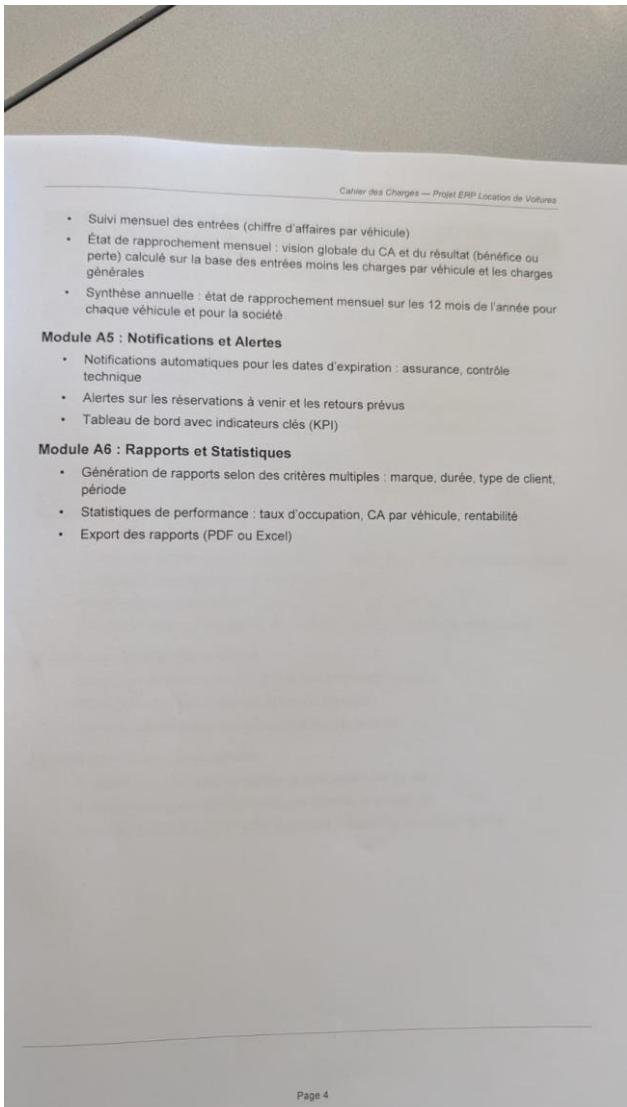
- Enregistrement complet d'une location : client, véhicule, dates de début/fin, tarif appliqué, caution
- Historique complet des locations par véhicule et par client
- Calcul automatique des coûts de location en fonction de la durée, la catégorie du véhicule et le type de client
- Gestion des prolongations et retours anticipés
- Gestion des contrats de location (génération PDF)

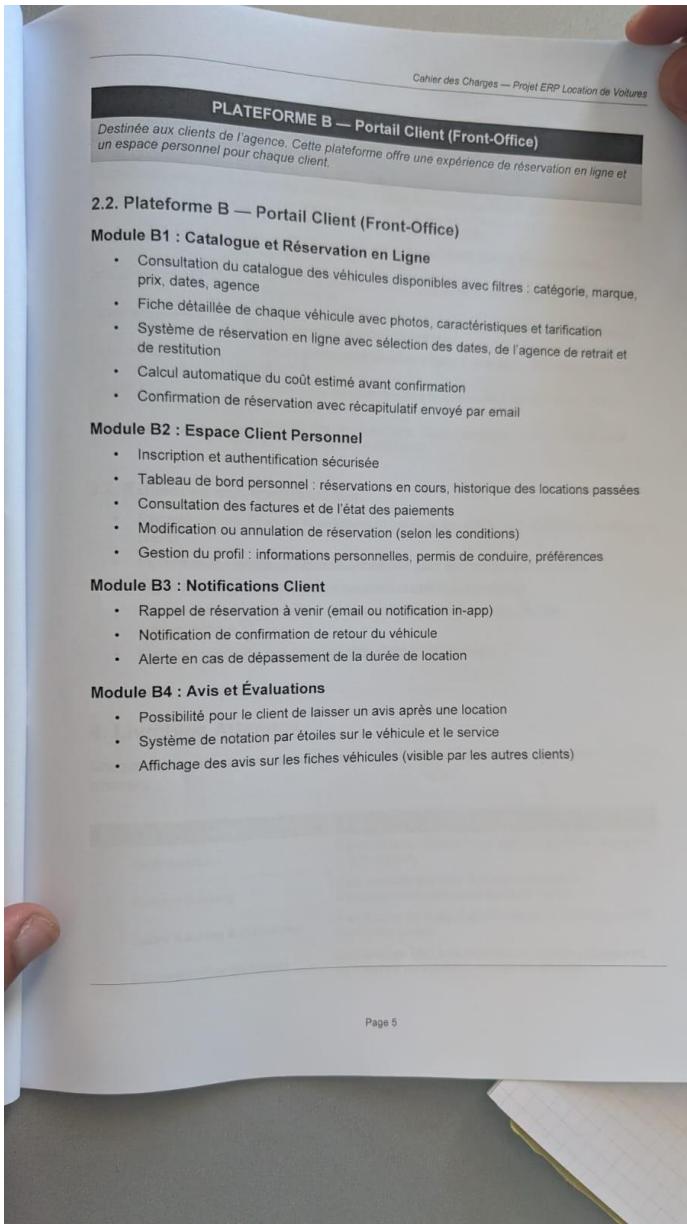
Module A3 : Transfert Inter-Agences (Offshoring)

- Suivi des mouvements de véhicules entre les différentes agences
- Enregistrement des informations de transfert : agence d'origine, agence de destination, dates, coûts associés
- Traçabilité complète de l'historique des transferts

Module A4 : Gestion Financière

- Gestion des paiements : enregistrement des factures, modalités de paiement (espèces, virement, chèque), date de paiement, trésorerie (banque ou caisse)
- Gestion des charges des véhicules : maintenance, réparation, prélèvements (crédits), assurance
- Gestion des charges fixes et variables de la société : personnel, loyer, téléphone, bureautique, CNSS, TVA, impôts, dons, autres





Cahier des Charges — Projet ERP Location de Voitures

3. Exigences Techniques

3.1. Architecture et Technologies

Composant	Exigences
Architecture	Architecture en couches (Présentation, Métier, Accès aux données) avec séparation claire des responsabilités
Back-end	Libre choix technologique (Spring Boot, Django, Node.js, Laravel, .NET, ou autre) — à justifier
Front-end	Libre choix (React, Angular, Vue.js, ou autre) — interface responsive et ergonomique
Base de données	Libre choix (MySQL, PostgreSQL, MongoDB, ou autre) — base unique partagée entre les deux plateformes
Gestion de versions	Git obligatoire avec workflow structuré (branching, pull requests, revue de code)
CI/CD	Pipeline d'intégration continue recommandé (GitHub Actions, GitLab CI, ou équivalent)
Qualité du code	Respect des principes Clean Code, nommage cohérent, pas de code mort, commentaires pertinents

3.2. Exigences non fonctionnelles

- Sécurité : authentification et autorisation (gestion des rôles : administrateur, employé, client)
- Performance : temps de réponse acceptable pour les opérations courantes
- Ergonomie : interface intuitive et responsive (desktop et mobile)
- Maintenabilité : code modulaire, respect du principe de séparation des préoccupations
- Fiabilité : validation des données côté client et côté serveur

4. Livrables Attendus

Chaque équipe devra remettre l'ensemble des livrables suivants lors de la semaine des examens :

N°	Livrable	Description
1	Code source	Dépôt Git avec historique complet des commits, branches et pull requests
2	Product Backlog	Liste priorisée des User Stories avec critères d'acceptation et estimation en Story Points
3	Sprint Backlog & Burndown	Planification de chaque sprint avec suivi de l'avancement (Burndown Chart)
4	Documentation technique	Diagrammes UML (cas d'utilisation, architecture du système, modèle)

Page 6

Cahier des Charges – Projet ERP Location de Vélos

5	Justification du processus	Document argumentant le choix du cycle de vie et des pratiques adoptées (2-3 pages)
6	Rapport de sprint	Compte-rendu de chaque sprint : objectifs, réalisations, obstacles, rétrospective
7	Démonstration	Présentation orale avec démo en direct des deux plateformes fonctionnelles

5. Grille d'Évaluation

Le projet représente 50% de la note finale du module. La note sera attribuée selon la grille suivante :

Critère	Pondération	Éléments évalués
Processus et méthodologie	20%	Justification du processus, artefacts produits, cérémonies
Qualité fonctionnelle	25%	Complétude des modules, fonctionnalités, interopérations, cohérence métier
Qualité technique et architecture	20%	Architecture Clean Code, séparation des couches, patterns utilisés
Pratiques DevOps et Git	10%	Historique Git, workflow, CI/CD, tests
Travail d'équipe et collaboration	10%	Répartition équitable, contributions Git, rétrospectives
Présentation et démonstration	15%	Clarté de la présentation, démo fonctionnelle, réponses aux questions
TOTAL		100%

6. Planning et Jalons

Le projet s'étend de la semaine 3 (après la présentation du Processus Unifié et de l'architecture) jusqu'à la semaine des examens. Les jalons suivants sont à respecter :

Semaine	Jalon	Livrable attendu
Semaine 2	Formation des équipes	Fiche projet validée (composition de l'équipe, sujet confirmé)
Semaine 3	Lancement du projet	Justification du processus choisi, Product Backlog initial, architecture préliminaire
Semaine 4	Simulation SCRUM	Sprint 1 planifié et lancé
Semaine 5	Suivi intermédiaire	Démo Sprint 1, planification Sprint 2, Burndown Chart

