



Cahier des Charges Projet JojoApps

Validation	Visa et date
X	X

Suivi des modifications

Date	N° de version	Chapitres concernés	Nature des modifications
11/10/2023	V1.0	Tous	Initialisation du document
19/10/2023	V.1.1	Tous	Post revue d'équipe
24/10/2023	V1.2	1 et 2	Revue après présentation client

Table des matières

1	Introduction	12
1.1	Présentation du projet	12
1.2	Objectifs	12
1.3	Public cible	13
1.3.1	Clients :	13
1.3.2	Joggeurs (Transporteurs et Manutentionnaires) :	13
1.4	Contexte du projet	14
2	Périmètre du Projet	14
2.1	Partie Client	14
2.1.1	Authentification et Gestion de Compte	14
2.1.2	Paieement et Finance	15
2.1.3	Interactions et Communications	15
2.1.4	Paramétrage et Personnalisation	15
2.1.5	Fonctionnalités de Service	15
2.2	Partie Joggeur	16
2.2.1	Authentification et Gestion de Compte	16
2.2.2	Paieement et Finance	16
2.2.3	Interactions et Communications	16
2.2.4	Fonctionnalités de Service	17
2.3	Fonctionnalités du Backoffice	18
2.3.1	Authentification et Gestion de Compte	18
2.3.2	Interactions et Communications	18
2.3.3	Paramétrage et Personnalisation	18
2.3.4	Fonctionnalités de Service	18
2.4	Solution en Marque Blanche	19
2.4.1	Flexibilité et Personnalisation	19
2.4.2	Accès au Code Source	19
2.4.3	Évolutivité	19
2.4.4	Optimisation des Coût	19
2.4.5	Liberté de Choix de Prestataire et Futurs Développements	19
2.5	Fonctionnalités Écartées	20
2.5.1	Groupage de Colis	20
2.5.2	Négociation en Temps Réel	20
2.5.3	Gestion Professionnelle de la Flotte de Véhicules	20
2.5.4	Fonctionnalité Déménagement	21
2.5.5	Paieement en espèces	21
2.5.6	Périmètre des Services fournis par 2CConseil	22
3	Acteurs du Projet	22
3.1	Équipe Interne	22
3.2	Partenaires Externes	23
3.3	Utilisateurs	23

3.4	Parties Prenantes.....	23
4	Exigences Fonctionnelles	23
4.1	Interface Utilisateur (Frontend).....	23
4.1.1	Authentification et Inscription	24
4.1.2	Recherche de Missions	24
4.1.3	Commande de Service	24
4.1.4	Suivi de Mission	24
4.1.5	Historique des Missions	25
4.1.6	Profil Utilisateur	25
4.1.7	Notifications.....	25
4.1.8	Appel Anonymisé	25
4.1.9	Paieement.....	25
4.1.10	Evaluation et Commentaires	25
4.1.11	FAQ et Assistance	26
4.1.12	Gestion des Favoris.....	26
4.1.13	Paramètres de Confidentialité.....	26
4.1.14	Langues.....	26
4.2	Interface d'Administration (Backoffice)	26
4.2.1	Authentification et Gestion de Compte	26
4.2.2	Suivi en Temps Réel des Joggers	27
4.2.3	Gestion des Missions	27
4.2.4	Statistiques et Analyse des Données	27
4.2.5	Gestion des Clients et des Joggers	27
4.2.6	Paramétrage Tarifaire	27
4.2.7	Notifications et Alertes	28
4.2.8	Sécurité et Conformité.....	28
4.2.9	Gestion des Réclamations et du Support Client	28
4.2.10	Gestion des Paiements et des Facturations	28
4.2.11	Rapports et Historiques	28
4.2.12	Intégrations Externes.....	28
4.2.13	Gestion des Contenus.....	29
4.2.14	Planification et Scheduling	29
4.2.15	Mises à jour et Maintenance	29
4.2.16	Tests et Feedbacks.....	29
4.2.17	Gestion Multilingue	29
4.3	Service de Paiement.....	29
4.3.1	Sélection du Mode de Paiement.....	29
4.3.2	Confirmation de Paiement.....	30
4.3.3	Gestion des Erreurs de Paiement	30
4.3.4	Historique des Transactions.....	30
4.3.5	Remboursements et Annulations	30
4.3.6	Sécurité des Paiements	30
4.4	API et Services Tiers	30
4.4.1	Google Maps API.....	31
4.4.2	Service de SMS.....	31
4.4.3	Passerelles de Paiement	31
4.4.4	Système de Gestion des Avis (Review)	31
4.4.5	Service de Support Client.....	31
4.4.6	Analytics.....	32
4.4.7	Appels Anonymisés	32
4.5	Notifications	32
4.5.1	Notifications Push	32
4.5.2	Notifications par Email.....	32

4.5.3	Notifications In-App	33
4.5.4	SMS	33
4.5.5	Notifications de Système	33
4.6	Sécurité	33
4.6.1	Authentification à Deux Facteurs (2FA)	33
4.6.2	Chiffrement des Données	34
4.6.3	Contrôle d'Accès	34
4.6.4	Protection contre les Attaques Communes	34
4.6.5	Surveillance et Alertes de Sécurité	34
4.6.6	Conformité Réglementaire	34
4.7	Communication	35
4.7.1	Appel Anonymisé	35
4.7.2	Notifications Push	35
4.7.3	Email & SMS	35
4.7.4	Intégration avec des Outils de Support Client	35
4.7.5	API de Communication Externe	36
5	Exigences Non-Fonctionnelles.....	36
5.1	Performance.....	36
5.1.1	Temps de Chargement.....	36
5.1.2	Scalabilité	36
5.1.3	Réactivité	36
5.1.4	Latence de l'API.....	37
5.1.5	Optimisation des Requêtes.....	37
5.1.6	Gestion du Cache	37
5.1.7	Gestion des Fichiers Médias	37
5.1.8	Traitement en Arrière-Plan	37
5.2	Sécurité.....	37
5.2.1	Authentification	37
5.2.2	Chiffrement.....	37
5.2.3	Audit et Suivi.....	37
5.2.4	Contrôle d'Accès	38
5.2.5	Protection contre les Attaques	38
5.2.6	Mise à jour de Sécurité	38
5.2.7	Test de Pénétration.....	38
5.2.8	Sauvegarde et Restauration	38
5.2.9	Formation et Sensibilisation	38
5.2.10	Conformité.....	38
5.3	Expérience Utilisateur	38
5.3.1	Temps de Chargement des Pages	38
5.3.2	Intuitivité de l'Interface	39
5.3.3	Accessibilité.....	39
5.3.4	Support Multilingue	39
5.3.5	Feedback Utilisateur	39
5.3.6	Aide en ligne et Documentation	39
5.3.7	Récupération de Compte	39
5.3.8	Continuité de Service	39
5.3.9	Conformité RGPD	39
5.3.10	Interactions Tactiles	39
5.3.11	Personnalisation	40
5.4	Fiabilité	40
5.4.1	Disponibilité du Service.....	40
5.4.2	Gestion des Erreurs.....	40
5.4.3	Redondance des Serveurs.....	40

5.4.4	Sauvegardes Régulières	40
5.4.5	Monitoring en Temps Réel	40
5.4.6	Tests de Charge	40
5.4.7	Mises à Jour Planifiées	41
5.4.8	Gestion de la Capacité	41
5.4.9	Mécanismes de Reprise après Panne	41
5.4.10	Journalisation et Audit	41
5.5	Maintenabilité	41
5.5.1	Documentation du Code	41
5.5.2	Modularité	41
5.5.3	Versionnage	41
5.5.4	Tests Unitaires et d'Intégration	42
5.5.5	Plan de Maintenance	42
5.5.6	Accès à la Maintenance	42
5.5.7	Révisions de Code	42
5.5.8	Formation du Personnel	42
5.5.9	Outils de Diagnostic	42
5.5.10	Gestion des Dépendances	42
5.5.11	Gestion et Mise à jour du Contenu	42
5.6	Contraintes Matérielles et Logicielles	42
5.6.1	Système d'Exploitation	43
5.6.2	Navigateurs Web	43
5.6.3	Serveurs	43
5.6.4	Base de Données	43
5.6.5	Stockage	43
5.6.6	Bande Passante	43
5.6.7	Terminaux Mobiles	43
5.6.8	APIs et Bibliothèques Externes	43
5.6.9	Limitations du Réseau	43
5.6.10	Environnement de Développement	44
5.7	Législation et Conformité	44
5.7.1	Protection des Données Personnelles	44
5.7.2	Propriété Intellectuelle	44
5.7.3	Paiements en Ligne	44
5.7.4	Accessibilité	44
5.7.5	Contrats avec des Tiers	44
5.7.6	Impôts et Taxes	44
5.7.7	Cybersécurité	44
5.7.8	Conditions Générales d'Utilisation (CGU)	45
5.7.9	Code de Conduite pour les Chauffeurs et Utilisateurs	45
5.7.10	Licences et Permis	45
6	Spécifications Techniques	45
6.1	Architecture Logicielle	45
6.1.1	Backend	45
6.1.2	Frontend	45
6.1.3	Bases de Données	45
6.1.4	APIs et Services Tiers	46
6.1.5	Authentification et Sécurité	46
6.2	Matériel et Infrastructure	46
6.2.1	Serveurs	46
6.2.2	Stockage	46
6.2.3	Réseau	46
6.2.4	Sécurité	46
6.2.5	Environnements	46

6.2.6	Automatisation CI/CD	46
6.2.7	Coût.....	46
6.2.8	Conformité et Audits.....	47
6.2.9	Gestion des Catastrophes	47
6.2.10	Documentation.....	47
6.3	Environnements de Développement.....	47
6.3.1	IDE (Integrated Development Environment)	47
6.3.2	Gestion de Version	47
6.3.3	Gestion des Dépendances.....	47
6.3.4	Outils de Linting et de Formatage.....	47
6.3.5	Automatisation des Tests.....	47
6.3.6	CI/CD	48
6.3.7	Déploiement	48
6.3.8	Monitoring et Logging.....	48
6.3.9	Revue de Code	48
6.3.10	Documentation.....	48
6.3.11	Collaboration et Communication	48
6.4	Procédures de Déploiement	48
6.4.1	Construction du Code	48
6.4.2	Tests Automatisés.....	48
6.4.3	Environnements de Déploiement	48
6.4.4	Migration de la Base de Données	49
6.4.5	Mise à Jour des Services Tiers.....	49
6.4.6	Rollback.....	49
6.4.7	Monitoring Post-déploiement	49
6.4.8	Sauvegarde et Restauration.....	49
6.4.9	Maintenance	49
6.4.10	Documentation de Déploiement.....	49
6.4.11	Validation.....	49
6.5	Normes et Protocoles.....	50
6.5.1	Normes de Codage.....	50
6.5.2	Protocoles de Communication.....	50
6.5.3	Normes API	50
6.5.4	Protocoles de Sécurité	50
6.5.5	Gestion des Données	50
6.5.6	Accessibilité.....	50
6.5.7	Localisation et Internationalisation	50
6.5.8	Versioning	50
6.5.9	Documentation	51
6.5.10	Tests.....	51
6.5.11	Revue de Code.....	51
6.5.12	Conformité Légale	51
6.6	Test et Qualité	51
6.6.1	Stratégie de Test	51
6.6.2	Environnements de Test	51
6.6.3	Méthodologie de Test	52
6.6.4	Outils de Test	52
6.6.5	Critères de Qualité	52
6.6.6	Performances.....	52
6.6.7	Sécurité	52
6.6.8	Accessibilité.....	52
6.6.9	Monitoring et Rapports	52
6.6.10	Revue de Code.....	52
6.6.11	Automatisation	52

6.7	Documentation Technique	53
6.7.1	Introduction et Vue d'Ensemble	53
6.7.2	Architecture Logicielle	53
6.7.3	Codes Sources	53
6.7.4	API et Services Tiers	53
6.7.5	Bases de Données	53
6.7.6	Procédures de Déploiement	53
6.7.7	Tests	53
6.7.8	Maintenance et Support	54
6.7.9	Sécurité	54
6.7.10	Licences et Conformités	54
6.7.11	Historique des Versions	54
6.7.12	Manuels Utilisateur	54
6.8	Maintenance et Support	54
6.8.1	Types de Maintenance	54
6.8.2	Calendrier de Maintenance	55
6.8.3	Support Client	55
6.8.4	SLA (Service Level Agreement)	55
6.8.5	Monitoring	55
6.8.6	Sauvegardes	55
6.8.7	Documentation	55
6.8.8	Formation	55
6.8.9	Gestion des Versions	55
6.8.10	Révisions d'Exigence	56
6.9	Conformité et Législation	56
6.9.1	Protection des Données	56
6.9.2	Transactions Financières	56
6.9.3	Accessibilité	56
6.9.4	Droit d'Auteur	56
6.9.5	Commerce Électronique	56
6.9.6	Travail Indépendant et Contrats	56
6.9.7	Communication et Marketing	57
6.9.8	Sécurité	57
6.9.9	Conformité Locale	57
6.9.10	Audits et Contrôles	57
7	Contraintes du Projet	57
7.1	Contraintes Temporelles	57
7.1.1	Date de Lancement Prévue	57
7.1.2	Phases de Développement	58
7.1.3	Jalons Importants	58
7.1.4	Contraintes de Sprints	58
7.1.5	Révisions et Approbations	58
7.1.6	Maintenance et Mises à Jour	58
7.1.7	Restrictions Externes	58
7.1.8	Revue Post-Projet	58
7.2	Contraintes Budgétaires	59
7.2.1	Estimation Initiale du Budget	59
7.2.2	Allocation des Ressources	59
7.2.3	Gestion des Risques Financiers	59
7.2.4	Suivi et Contrôle	59
7.2.5	Marges de Manœuvre	59
7.2.6	Revue Post-Projet	59
7.3	Contraintes Technologiques	59

7.3.1	Matériel.....	59
7.3.2	Logiciel	60
7.3.3	Dépendances Externes.....	60
7.3.4	Sécurité	60
7.3.5	Réseau.....	60
7.3.6	Scalabilité	60
7.3.7	Accessibilité.....	60
7.3.8	Localisation et Internationalisation	60
7.3.9	Maintenance	61
7.3.10	Conformité et Standards	61
7.4	Contraintes de Ressources Humaines	61
7.4.1	Compétences Techniques	61
7.4.2	Disponibilité de l'Équipe	61
7.4.3	Répartition Géographique	61
7.4.4	Organisation et Culture d'Entreprise	61
7.4.5	Gestion de Projet	61
7.4.6	Charge de Travail	62
7.4.7	Risques Associés aux Ressources Humaines	62
7.5	Contraintes Légales et de Conformité	62
7.5.1	Protection des Données.....	62
7.5.2	Propriété Intellectuelle	62
7.5.3	Commerce Électronique	62
7.5.4	Accessibilité.....	62
7.5.5	Sécurité	63
7.5.6	Transfert de Données Internationales	63
7.5.7	Audit et Conformité	63
7.5.8	Responsabilités des Fournisseurs	63
7.5.9	Engagements Contractuels	63
7.5.10	Considérations Locales ou Spécifiques au Secteur	63
7.6	Contraintes de Marché.....	63
7.6.1	Concurrence	63
7.6.2	Tendance du Marché	64
7.6.3	Exigences du Client	64
7.6.4	Canaux de Distribution	64
7.6.5	Réglementations de Prix	64
7.6.6	Barrières Culturelles et Linguistiques	64
7.6.7	Partenariats et Intégrations	64
7.6.8	Cycle de Vie du Produit	64
7.6.9	Gestion des Risques de Marché	65
7.6.10	Évaluation et Surveillance des KPIs	65
7.7	Contraintes Géographiques.....	65
7.7.1	Zones de Service	65
7.7.2	Infrastructure Locale.....	65
7.7.3	Lois et Réglementations Locales	65
7.7.4	Langue et Culture.....	65
7.7.5	Partenariats Locaux.....	65
7.7.6	Saisonnalité et Conditions Climatiques.....	66
7.7.7	Gestion des Catastrophes	66
7.7.8	Taxes et Tarifications Locales.....	66
7.7.9	Plans d'Expansion.....	66
7.8	Contraintes de Qualité	66
7.8.1	Normes de Qualité	66
7.8.2	Assurance Qualité	66
7.8.3	Gestion des Défauts	66

7.8.4	Performance et Temps de Réponse	67
7.8.5	Disponibilité et Fiabilité	67
7.8.6	Gestion des Risques Qualitatifs	67
7.8.7	Expérience Utilisateur	67
7.8.8	Tests de Qualité	67
7.8.9	Revue de Code	67
7.8.10	Documentation de Qualité	67
7.9	Contraintes de Communication	67
7.9.1	Canaux de Communication	68
7.9.2	Fréquence de Communication	68
7.9.3	Confidentialité et Sécurité	68
7.9.4	Outils de Suivi et de Documentation	68
7.9.5	Communication avec les Parties Prenantes	68
7.9.6	Gestion des Attentes	68
7.9.7	Langue et Culture	68
7.9.8	Rapports et Mise à Jour du Projet	68
7.9.9	Communication en Cas d'Urgence	69
8	Méthodologies de Développement	69
8.1	Choix de la Méthodologie	69
8.2	Planification et Gestion de Projet	70
8.3	Gestion des Risques	71
8.4	Contrôle de Version	72
8.5	Gestion de la Qualité	73
8.5.1	Objectifs de Qualité	73
8.5.2	Normes de Qualité Applicables	73
8.5.3	Mesure de la Qualité	74
8.5.4	Outils de Gestion de la Qualité	74
8.5.5	Revue de Qualité	74
8.5.6	Formation et Sensibilisation	74
8.5.7	Documentation de la Qualité	74
8.5.8	Responsabilités	74
8.5.9	Gestion des Anomalies	74
8.6	Intégration et Déploiement Continu (CI/CD)	74
8.6.1	Objectifs du CI/CD	75
8.6.2	8.6.2 Outils de CI/CD	75
8.6.3	8.6.3 Pipeline CI/CD	75
8.6.4	8.6.4 Gestion des Branches	75
8.6.5	Surveillance et Monitoring	75
8.6.6	Notification et Alertes	75
8.6.7	Documentation du CI/CD	75
8.7	Documentation	76
8.7.1	Objectifs de la Documentation	76
8.7.2	Types de Documentation	76
8.7.3	Outils pour la Documentation	76
8.7.4	Gestion des Versions de la Documentation	76
8.7.5	Normes et Conventions	76
8.7.6	Accessibilité et Sécurité	77
8.7.7	Responsabilités	77
8.8	Maintenance et Support	77
8.8.1	Objectifs de la Maintenance et du Support	77
8.8.2	Types de Maintenance	77

8.8.3	Procédures de Support	77
8.8.4	Niveaux de Service (SLA)	77
8.8.5	Ressources Humaines et Techniques	78
8.8.6	Planification	78
8.8.7	Budget et Coût	78
8.8.8	Évaluation et Retours	78
8.8.9	Mise à Jour du Plan	78
8.9	Plan de Test	78
8.9.1	Stratégie de Test	78
8.9.2	Niveaux de Test	78
8.9.3	Environnements de Test	79
8.9.4	Planification des Tests	79
8.9.5	Cas de Test	79
8.9.6	Critères de Succession et d'Échec	79
8.9.7	Automatisation des Tests	79
8.9.8	Gestion des Incidents et des Bugs	79
8.9.9	Rapports et Métriques de Test	79
8.9.10	Revue de Test et Clôture	79
8.9.11	Responsabilités	79
9	Plan de Mise en Œuvre	80
9.1	Stratégie de Déploiement	80
9.1.1	Objectifs du Déploiement	80
9.1.2	Phases de Déploiement	80
9.1.3	Environnements Cibles	80
9.1.4	Outils de Déploiement	80
9.1.5	Calendrier de Déploiement	80
9.1.6	Équipe de Déploiement	80
9.1.7	Risques et Contingences	81
9.1.8	Revue et Approbation	81
9.2	Planification des Ressources	81
9.2.1	Objectifs de la Planification des Ressources	81
9.2.2	Types de Ressources	81
9.2.3	Allocation des Ressources	81
9.2.4	Outils de Gestion des Ressources	81
9.2.5	Calendrier de ressources	82
9.2.6	Gestion des Risques liés aux Ressources	82
9.2.7	Revue et Ajustements	82
9.3	Plan de Communication	82
9.3.1	Objectifs de Communication	82
9.3.2	Parties Prenantes	82
9.3.3	Canaux de Communication	82
9.3.4	Types de Communication	83
9.3.5	Fréquence de Communication	83
9.3.6	Méthodes de Documentation	83
9.3.7	Protocole en cas de Crise	83
9.3.8	Responsabilités	83
9.4	Stratégie de Backup et de Restauration	83
9.4.1	Objectifs	83
9.4.2	Types de Données à Sauvegarder	84
9.4.3	Méthodologie de Sauvegarde	84
9.4.4	Stockage des Sauvegardes	84
9.4.5	Procédures de Restauration	84
9.4.6	Test des Procédures de Restauration	84
9.4.7	Rôles et Responsabilités	84

9.4.8	Durée de Conservation des Données.....	84
9.4.9	Conformité.....	84
9.4.10	Surveillance et Alerte.....	85
9.4.11	Budget	85
9.5	Plan de Suivi et de Maintenance	85
9.5.1	Objectifs.....	85
9.5.2	Types de Maintenance.....	85
9.5.3	Calendrier de Maintenance	85
9.5.4	Indicateurs de Performance (KPIs).....	85
9.5.5	Suivi des Incidents.....	86
9.5.6	Documentation	86
9.5.7	Budget et Coûts.....	86
9.5.8	Formation et Transfert de Compétences.....	86
9.5.9	Audit et Conformité	86
9.6	Documentation et Manuels d'Utilisateur	86
9.6.1	Objectifs.....	86
9.6.2	Types de Documentation	86
9.6.3	Format et Accessibilité.....	87
9.6.4	Mise à Jour de la Documentation	87
9.6.5	Rôles et Responsabilités	87
9.6.6	Versionnage	87
9.6.7	Localisation	87
9.6.8	Révisions et Approbations	87
9.6.9	Archivage	87
9.6.10	Formation et Sensibilisation	87
10	Annexes.....	88
10.1	Glossaire de Termes.....	88
10.2	Références Documentaires.....	88
10.3	Cahier des Clauses Techniques Particulières (CCTP).....	88
10.4	Modèles de Wireframes et maquettes.....	88
10.5	Diagrammes	88
10.6	Spécifications d'API.....	88
10.7	Plan de test détaillé	88
10.8	Spécifications de Matériel et Logiciel.....	89
10.9	Contacts et Responsabilités.....	89
10.10	Calendrier de Projet	89
10.11	Accords de Niveau de Services (SLA)	89
10.12	Licences et Droits d'Auteur	89
10.13	Audits et Conformités	89
10.14	Code Source	89
10.15	Modifications et Mises à Jour	89
10.16	Clarification des Responsabilités Hors du Périmètre de 2CConseil.....	89
10.16.1	Mise à jour du contenu.....	89
10.16.2	Gestion de la sécurité	90
10.16.3	Marketing et Distribution	90
10.16.4	Formation des utilisateurs.....	90

10.16.5	Conformité réglementaire et licences	90
10.16.6	Surveillance et suivi	90
10.16.7	Financement et budgétisation.....	90
10.16.8	Sauvegardes et récupérations	90
10.16.9	Gestion de la qualité.....	91
10.16.10	Service Client et Support Technique	91
10.16.11	Utilisation des Données par l'Administrateur Final.....	91
11	Validation du Cahier des charges	91
11.1	Procédure de Validation.....	91
11.1.1	Objectifs.....	91
11.1.2	Parties Prenantes Impliquées	91
11.1.3	Étapes de la Validation	91
11.1.4	Calendrier de Validation	92
11.1.5	Méthodes de Communication	92
11.1.6	Documentation de Validation	92
11.1.7	Procédure de Révision Post-Validation	92
11.2	Critères de Validation	92
11.2.1	Objectifs.....	93
11.2.2	Critères de Contenu.....	93
11.2.3	Critères Techniques	93
11.2.4	Critères de Conformité Réglementaire.....	93
11.2.5	Critères de Revue des Parties Prenantes.....	93
11.2.6	Critères Temporels	93
11.2.7	Critères Financiers	93
11.2.8	Conclusion	94
11.3	Responsabilités.....	94
11.3.1	Rôles et Responsabilités	94
11.3.2	Responsabilités des Clients.....	94
11.3.3	Processus de Sign-Off	94
11.3.4	Suivi et Mise à Jour	94
11.3.5	Calendrier	94
11.3.6	Conclusion	94
11.4	Calendrier de Validation.....	95
11.5	Processus de Révision	96
11.6	Documentation de Validation.....	97
11.7	Signature des Parties Prenantes	98
11.8	Date de Validation	99
11.9	Modifications Post-Validation	99
11.10	Accord et Validation du Cahier des Charges	100

1 Introduction

1.1 Présentation du projet

Le projet JojoApps a pour but de créer une plateforme mobile composée de deux applications distinctes, une pour les clients et une autre pour les joggeurs. Les deux applications fonctionneront sur Android et iOS. Une troisième composante de ce projet est une interface de backoffice destinée à la gestion et l'administration de la plateforme.

Application pour les clients : Cette application offre une interface utilisateur permettant aux clients de solliciter les services de joggeurs dans leur région. Les utilisateurs peuvent créer des profils, définir leurs besoins spécifiques, et effectuer des transactions financières sécurisées.

Application pour les joggeurs : Cette application est destinée aux joggeurs qui souhaitent proposer leurs services. Elle permet de créer un profil professionnel, de suivre les demandes en temps réel et de gérer les transactions et les évaluations.

Interface de backoffice : Cette interface web est conçue pour permettre aux gestionnaires et aux administrateurs de surveiller et de gérer l'ensemble du système. Elle offre des fonctionnalités comme le suivi des transactions, la gestion des utilisateurs, l'analyse des données et la génération de rapports.

Solution en Marque Blanche : L'ensemble de l'architecture est conçu pour être modulaire et réutilisable, permettant des adaptations mineures d'interface utilisateur et d'expérience utilisateur pour servir divers autres marchés ou besoins spécifiques.

1.2 Objectifs

Fournir une plateforme fiable et sécurisée : Assurer l'intégrité des données et la protection de la vie privée des utilisateurs.

Expérience utilisateur de haute qualité : Les applications doivent être intuitives, faciles à utiliser, et esthétiquement plaisantes.

Interopérabilité entre Android et iOS : Assurer une compatibilité fluide entre les deux systèmes d'exploitation mobiles pour toucher un public plus large.

Système de Matching efficace : Développer un algorithme qui met en relation les clients et les joggeurs de manière efficace et pertinente.

Notifications en temps réel : Mettre en place un système de notifications pour informer les utilisateurs d'événements importants ou d'opportunités de jogging.

Système de paiement intégré : Offrir une méthode de paiement sécurisée et facile à utiliser au sein de l'application.

Gestion du backoffice : Développer une interface administrateur qui permettra une gestion efficace des utilisateurs, des transactions et des contenus.

Scalabilité de la plateforme : Construire l'architecture de manière qu'elle soit facilement évolutive en termes de fonctionnalités et de nombre d'utilisateurs.

Solution en marque blanche : L'architecture doit être conçue de manière à être facilement réutilisable pour d'autres clients, nécessitant uniquement des ajustements d'interface utilisateur et d'expérience utilisateur mineurs.

Conformité réglementaire : Veiller à ce que la plateforme soit en conformité avec les réglementations en vigueur, notamment en ce qui concerne la protection des données personnelles.

Analytique et suivi : Intégrer des outils d'analyse pour suivre les KPIs et mesurer le succès de la plateforme.

Support et Maintenance : Mettre en place un système de support client et un plan de maintenance post-lancement.

1.3 Public cible

La plateforme "JojoApps" a pour objectif de servir différents groupes d'utilisateurs ayant des besoins spécifiques en matière de transport et/ou de manutention d'objets et de colis.

1.3.1 Clients :

-Particuliers : Personnes ayant des besoins occasionnels de transport de biens, que ce soit pour un déménagement, l'achat de gros objets, ou d'autres nécessités.

-Professionnels : Petites et moyennes entreprises ayant des besoins réguliers ou ponctuels de transport de marchandises ou d'équipements. ~~Afin de réduire le temps de développement, la gestion/inscription ne sera pas différencié entre pro et particulier.~~

Ajout post présentation du 23 Octobre 2023 : La précision du statut de l'utilisateur sera indiqué à côté du profil sans que cela n'affecte le paramétrage et l'interface utilisateur.

-Commerces en ligne : Boutiques qui nécessitent un service de livraison fiable et flexible. Afin de réduire le temps de développement, la gestion/inscription ne sera pas différencié entre pro et particulier.

1.3.2 Joggeurs (Transporteurs et Manutentionnaires) :

-Personnes à la recherche d'une source de revenu complémentaire et qui disposent d'un moyen de transport approprié (véhicule, vélo, etc.).

-Individus cherchant à monétiser leur temps libre en fournissant des services de transport ou de manutention.

- Ajout post présentation du 23 Octobre 2023 : Selon la faisabilité d'un point de vue technique et fonctionnel, les entités professionnelles pourront avoir un statut différencier des particuliers, avec une tarification différentes.

1.4 Contexte du projet

L'économie collaborative a pris un essor considérable ces dernières années, modifiant ainsi la manière dont les services sont fournis et consommés. Des entreprises comme *Cocolis* ont déjà mis en place des plateformes de mise en relation pour le transport d'objets encombrants, mais elles se concentrent principalement sur des particuliers disposant de véhicules adaptés pour de telles tâches.

Le projet "JojoApps" s'inscrit dans cette dynamique mais souhaite se démarquer en créant une plateforme multi-services axée autour du concept de "joggeurs", des individus pouvant offrir des services de transport et de manutention à pied ou avec des moyens de transport légers. Le service proposé est donc plus flexible et adapté à une gamme plus large de besoins, des petits colis aux objets plus volumineux.

L'initiative se positionne également dans une optique de développement durable, en maximisant l'utilisation des ressources disponibles (humaines et matérielles) et en minimisant l'empreinte carbone grâce à des trajets optimisés.

La possibilité de fournir une solution en marque blanche ajoute une dimension commerciale supplémentaire, permettant à 2CConseil de réutiliser l'architecture pour répondre aux besoins spécifiques d'autres clients, avec des ajustements mineurs et des personnalisations sur devis, tout en assurant une qualité de développement optimal, robuste et fiable.

2 Périmètre du Projet

2.1 Partie Client

2.1.1 Authentification et Gestion de Compte

Connexion de Compte Utilisateur : Le client pourra se connecter à l'application en utilisant son login et mot de passe.

Connexion OAuth : Le système supportera la connexion via OAuth, permettant aux utilisateurs de se connecter via leur compte Google, Facebook, ou Apple.

Inscription Manuelle : Les nouveaux utilisateurs auront la possibilité de s'inscrire manuellement en fournissant des informations essentielles.

Réinitialisation du Mot de Passe : Fonctionnalité permettant aux utilisateurs de réinitialiser leur mot de passe en cas d'oubli.

Vérification et Association du Numéro de Téléphone : Après vérification via OTP, l'utilisateur pourra associer son numéro de téléphone à son compte.

Vérification et Association de l'Adresse Mail : Après vérification via OTP, l'utilisateur pourra associer son adresse mail à son compte.

2.1.2 Paiement et Finance

Gestion des Cartes Bancaires : Les utilisateurs pourront ajouter, modifier ou supprimer leurs cartes bancaires via l'API STRIPE. Les formalités de transactions seront gérées par l'API

2.1.3 Interactions et Communications

Système d'Aide : Les utilisateurs auront la possibilité de soumettre un formulaire pour demander de l'aide sur un sujet spécifique.

Système de Redirection d'Appel : Pour la confidentialité, le numéro de téléphone du jogger et du client sera masqué et redirigé via un service tiers.

2.1.4 Paramétrage et Personnalisation

Paramètres du Compte : Les utilisateurs pourront personnaliser les réglages liés aux notifications, à la localisation et à la langue.

Modification des Informations de Profil : Les utilisateurs auront la possibilité de modifier l'Email, le mot de passe, la photo de profil et les informations liées à une entreprise s'ils sont professionnels.

2.1.5 Fonctionnalités de Service

Historique d'Activité : Les clients pourront consulter l'historique de leurs missions, qu'elles soient passées, en cours ou à venir.

Recherche de Lieu via GoogleMaps : Les utilisateurs pourront chercher des lieux via l'API GoogleMaps et visualiser les trajets, informations, etc.

Upload d'Images : Pendant la configuration d'une mission, il sera possible d'uploader des images pour clarifier les besoin, ou en fin de mission pour attester de la conformité de la prestation.

Recherche de Jogger : Les utilisateurs pourront rechercher des joggers disponibles par leur nom.

Suggestion d'Objet : Si un objet n'est pas présent dans la bibliothèque d'objets de l'application, les utilisateurs pourront le suggérer. Certains objets spécifiques (ex piano, table en marbre) seront exclus et feront l'objet d'une redirection vers un formulaire de demande à destination des professionnels de l'application.

Ajout de Temps Supplémentaire : Une option permettra d'ajouter du temps supplémentaire à une mission en cours.

Réservation de Mission : Les clients pourront réserver une mission pour une date ultérieure.

Mission Instantanée : Les clients auront la possibilité de lancer une mission à prendre en charge immédiatement. Le système recherchera le jogger le plus proche disponible et enverra des notifications à tous les joggers de la zone si nécessaire.

Notation de la Prestation : À la fin de chaque mission, le client devra obligatoirement noter la prestation du jogger pour pouvoir la clôturer.

2.2 Partie Joggeur

2.2.1 Authentification et Gestion de Compte

Connexion de Compte Utilisateur : Permet aux joggeurs de se connecter à l'application en utilisant leur identifiant et mot de passe.

Connexion OAuth : Offre une alternative de connexion via des plateformes tiers comme Google, Facebook, ou Apple pour faciliter l'accès.

Inscription de Compte Utilisateur : Permet aux nouveaux joggeurs de créer un compte en remplissant un formulaire.

Inscription OAuth : Possibilité de créer un nouveau compte via des plateformes comme Google, Facebook, ou Apple.

Réinitialisation du Mot de Passe : Fonction permettant aux joggeurs de récupérer leur compte en cas d'oubli du mot de passe.

Associer/vérifier le Numéro de Téléphone : Après une vérification par OTP, le joggeur peut associer son numéro de téléphone à son compte.

Associer/vérifier l'Adresse Mail : Fonction similaire à la vérification du numéro de téléphone mais pour l'adresse email.

Ajouter une Image de Profil : Les joggeurs peuvent personnaliser leur profil en y ajoutant une photo.

2.2.2 Paiement et Finance

Ajouter/modifier/supprimer une Carte Bancaire :
Les joggeurs peuvent gérer leurs options de paiement via l'intégration de l'API STRIPE.

Modifier les Informations du Profil :
Les joggeurs peuvent mettre à jour les informations de leur compte telles que l'email, le mot de passe, la photo de profil, et les informations de l'entreprise.

Consulter l'Historique d'Activité :
Un accès à l'historique des missions, qu'elles soient terminées, en cours, ou planifiées est disponible.

2.2.3 Interactions et Communications

Demander de l'Aide : En cas de problème ou de question, un formulaire est disponible pour contacter le support.

Téléphoner : Un système de redirection d'appel est mis en place pour éviter le partage de numéros personnels entre les clients et les joggeurs.

Uploader des Documents : Pendant le processus d'inscription, des documents peuvent être requis pour vérifier l'identité ou des compétences spécifiques.

Mode Connecté / Hors Réseau : Les joggeurs peuvent indiquer leur disponibilité en passant en mode connecté ou hors réseau.

2.2.4 Fonctionnalités de Service

Afficher les Missions Disponibles : Les joggeurs peuvent consulter une liste de missions disponibles pour y postuler. *Ajout post présentation du 23 Octobre 2023 : En attendant la validation de ses documents, et le passage en statut « Profil Vérifié » par l'admin, le joggeur a la possibilité de répondre à des propositions de missions, après avoir signer une attestation sur l'honneur de conformité réglementaire.*

Afficher les Détails d'une Mission : Informations complètes sur une mission spécifique (lieu, heure, type de service, etc.).

Valider la liste d'objet : Le joggeur vérifie et valide que la liste d'objet envoyé par le client est conforme.

Renvoyer la Position du Jogger en Temps Réel : Un suivi en temps réel via une carte dynamique est disponible pendant la mission.

Gestion des Véhicules : Les joggeurs ont la possibilité d'ajouter, de modifier ou de supprimer des informations relatives à leurs véhicules.

Noter la Prestation : À la fin de chaque mission, il est obligatoire pour le joggeur de noter la prestation du client.

Signature Client : Un système de signature numérique est mis en place pour confirmer la fin d'une mission.

Signaler un QUOTA Dépassé : En cas de dépassement d'un quota (temps, distance, etc.), le joggeur peut le signaler. *Ajout post présentation du 23 Octobre 2023 : Possibilité de modifier et valider une extension de temps avec modification du prix en cours de prestation.*

Émettre une Demande de Virement : Permet aux joggeurs de récupérer leurs gains via un virement bancaire.

Enregistrer un IBAN : Pour permettre les transactions financières, l'enregistrement d'un IBAN est requis.

Gestion des Missions : Une vue calendrier est disponible pour la gestion des missions futures et passées.

Gestion des Périodes de Disponibilité : Les joggeurs peuvent définir leurs plages horaires de disponibilité.

2.3 Fonctionnalités du Backoffice

2.3.1 Authentification et Gestion de Compte

Afficher les positions des joggers en temps réel :

Le back office permettra à l'administrateur de suivre en temps réel la position de tous les joggers actifs grâce à une intégration avec l'API de géolocalisation. Cela aidera à monitorer le déploiement effectif des ressources sur le terrain.

Effectuer des tris sur les missions :

Une interface de recherche avancée permettra aux administrateurs de filtrer les missions en cours ou terminées selon divers critères (date, statut, lieu, etc.).

Afficher le nombre de joggers connectés :

Une section du tableau de bord indiquera en temps réel le nombre de joggers actuellement connectés à l'application, aidant ainsi à évaluer la disponibilité du service.

Recherche des emplacements sur la carte :

L'API GoogleMaps sera intégrée pour permettre aux administrateurs de rechercher des lieux spécifiques sur la carte, utile pour les besoins d'analyse géographique ou d'assignation de missions.

Afficher un menu de données statistiques :

Un tableau de bord complet permettra aux administrateurs de suivre divers indicateurs de performance clés, tels que le nombre de missions effectuées, les revenus générés, et les évaluations des utilisateurs.

2.3.2 Interactions et Communications

Validation des documents : Le système permettra aux administrateurs de valider les documents officiels soumis par les joggers lors de leur inscription. Cette fonction est essentielle pour garantir le respect des normes et réglementations.

Gestion des formules : Les administrateurs pourront configurer et enregistrer les paramètres qui constituent les formules de tarification pour les différents types de services proposés.

2.3.3 Paramétrage et Personnalisation

Gestion des clients : Un registre d'utilisateurs sera disponible pour les administrateurs, qui pourront ajouter, supprimer ou modifier les profils des utilisateurs selon les besoins.

Gestion des joggers : De même, un registre pour les joggers sera disponible pour permettre leur gestion efficace, y compris la validation de leurs compétences, équipements et disponibilités.

2.3.4 Fonctionnalités de Service

Gestion de la bibliothèque d'objets : Le back office permettra d'ajouter, de supprimer ou de modifier les objets qui peuvent être sélectionnés par les utilisateurs lors de la création de missions.

Réglage des bases tarifaires : Les administrateurs auront la possibilité de définir et de mettre à jour les bases tarifaires pour les différents services offerts, en fonction des coûts opérationnels et des stratégies de tarification.

Modification du temps d'attente : Le temps d'attente avant de pouvoir faire appel au prochain joueur le plus proche est modifiable.

Toutes ces fonctionnalités ont été retenues pour le MVP afin de fournir un produit complet et fonctionnel dès la première version.

2.4 Solution en Marque Blanche

L'approche de développement en marque blanche présente plusieurs avantages stratégiques et opérationnels :

2.4.1 Flexibilité et Personnalisation

L'architecture en marque blanche est conçue pour permettre des adaptations et des personnalisations selon les besoins spécifiques du client.

2.4.2 Accès au Code Source

Bien que la propriété intellectuelle de l'architecture de base demeure la propriété de 2CConseil, un engagement est pris pour fournir le code source de la solution au client. Cela permet au client de ne pas être verrouillé avec un fournisseur unique pour d'éventuelles évolutions futures.

2.4.3 Évolutivité

La plateforme est conçue pour être évolutivement robuste, offrant la possibilité d'ajouter de nouvelles fonctionnalités ou de s'intégrer à d'autres systèmes, en accord avec les termes contractuels.

2.4.4 Optimisation des Coût

Le choix d'une architecture en marque blanche peut permettre un développement initial plus rapide et potentiellement moins coûteux, tout en bénéficiant d'un produit de qualité éprouvée.

2.4.5 Liberté de Choix de Prestataire et Futurs Développements

L'accès au code source confère au client la liberté de choisir un prestataire différent pour les besoins futurs en matière d'évolution. Néanmoins, il est à noter que 2CConseil restera disponible pour toute évolution future. Chaque fonctionnalité supplémentaire fera l'objet d'un devis en adéquation avec les prix du marché.

2.5 Fonctionnalités Écartées

Dans le cadre du développement du MVP, certaines fonctionnalités souhaitées par le client ont été écartées en raison de diverses contraintes, notamment en termes de temps et de ressources.

2.5.1 Groupage de Colis

Justifications pour l'écartement :

Augmentation significative du temps de développement : La mise en œuvre de cette fonctionnalité exigerait une révision majeure de la logique métier et du flux de l'application. Elle nécessiterait également le développement de nouveaux algorithmes de tri, de routage et de suivi. Cette complexité supplémentaire aurait pour conséquence un allongement considérable du temps nécessaire pour le développement du MVP.

Complexité logistique : La gestion et le suivi de plusieurs colis dans une seule mission ajoutent une complexité opérationnelle importante.

Expérience utilisateur : L'introduction de fenêtres de livraison plus larges et d'un système de suivi plus complexe pourrait rendre l'expérience moins fluide pour l'utilisateur.

Questions de responsabilité et d'assurance : Le transport de plusieurs colis pour différents clients dans une seule tournée poserait des défis en termes de responsabilité et d'exigences d'assurance.

2.5.2 Négociation en Temps Réel

La fonctionnalité de négociation en temps réel permettrait aux clients et aux joggers de négocier les coûts d'une mission avant de l'accepter.

Justifications pour l'écartement :

Complexité du développement : La mise en œuvre de cette fonctionnalité nécessiterait le développement d'un système de messagerie en temps réel sécurisé et fiable, intégrant également des mécanismes de mise à jour des tarifs et de confirmation.

Expérience utilisateur : La négociation en temps réel pourrait ralentir le processus d'attribution des missions, ce qui va à l'encontre de la notion de service "à la demande".

2.5.3 Gestion Professionnelle de la Flotte de Véhicules

La fonctionnalité de "Gestion Professionnelle de la Flotte de Véhicules" avait pour but de fournir aux entreprises un accès professionnel à la plateforme. Cet accès leur permettrait de gérer une flotte de véhicules et de chauffeurs, d'attribuer des horaires, des pauses et des coupures, et de suivre en temps réel la performance et la disponibilité des ressources.

Sous-fonctionnalités nécessaires :

-Attribution des horaires, pauses et coupures : Les gestionnaires pourraient définir les horaires de travail, les pauses et les périodes de coupure pour chaque chauffeur.

-Suivi en temps réel : Un tableau de bord serait disponible pour suivre en temps réel la localisation des véhicules, le statut des missions en cours, etc.

-Maintenance et conformité : Les entreprises pourraient suivre l'entretien des véhicules, les inspections nécessaires, et toute autre exigence de conformité.

Justifications pour l'écartement :

Complexité du développement : Mettre en place une telle fonctionnalité nécessiterait une structure de données très complexe, capable de gérer des informations sur l'entretien des véhicules, la certification des chauffeurs, les horaires, etc.

Sécurité et conformité : Les exigences en matière de sécurité des données seraient plus élevées, notamment pour la protection des informations sensibles liées à la flotte et aux chauffeurs.

Éloignement du concept de base : Le modèle actuel est conçu pour offrir un service de livraison simple et rapide aux particuliers. Intégrer une gestion professionnelle de la flotte élargirait considérablement la portée du projet et s'éloignerait de l'objectif initial.

Complexité non nécessaire pour un MVP : Le but du MVP (Minimum Viable Product) est de valider l'adéquation produit-marché le plus rapidement possible. L'ajout de cette fonctionnalité ajouterait une couche de complexité qui n'est pas nécessaire pour atteindre cet objectif.

En résumé, bien que cette fonctionnalité puisse avoir une grande valeur ajoutée pour certains clients, elle ne sera pas incluse dans le MVP en raison de sa complexité et de son éloignement du concept de base. Des développements ultérieurs pourraient envisager son inclusion, et chaque fonctionnalité supplémentaire fera l'objet d'un devis en adéquation avec les prix du marché

2.5.4 Fonctionnalité Déménagement

Cette fonctionnalité serait axée sur des services de déménagement classique.

Justifications pour l'écartement :

Éloignement du concept de base : La fonction s'écarte du modèle initial du service rapide et à la demande.

Complexité accrue : Le développement de cette fonction nécessiterait de nouvelles interfaces utilisateur, des calculs de coût différents, et des critères de sélection de mission spécifiques.

2.5.5 Paiement en espèces

Un système de paiement en espèces a été envisagé.

Justifications pour l'écartement :

Complexité de gestion : Le paiement en espèces poserait des problèmes de sécurité, de traçabilité et de

conformité fiscale qui rendraient la gestion plus complexe.

Expérience utilisateur : Le paiement en espèces ne s'aligne pas avec l'expérience utilisateur fluide et digitale que le service vise à offrir.

Il est à noter que ces fonctionnalités pourraient être réévaluées pour inclusion dans des versions ultérieures du produit, en fonction des besoins du marché et des évaluations de faisabilité. Chaque nouvelle fonctionnalité pourra faire l'objet d'un devis en adéquation avec les prix du marché.

2.5.6 Périmètre des Services fournis par 2CConseil

Dans le cadre de ce projet JojoApps, il est crucial de souligner que les responsabilités de 2CConseil se limitent exclusivement au développement de l'application. Les aspects suivants, bien qu'importants pour le succès à long terme du projet, sont hors du périmètre de 2CConseil et restent à définir :

- **Mise à jour du contenu** : La gestion et la mise à jour du contenu après le développement initial.
- **Gestion de la sécurité** : Toute gestion continue de la sécurité de l'application.
- **Marketing et Distribution** : La promotion et la distribution de l'application.
- **Formation des utilisateurs** : Toute formation nécessaire pour les utilisateurs finaux.
- **Conformité réglementaire et licences** : La gestion de la conformité réglementaire et des licences.
- **Surveillance et suivi** : Le suivi et la surveillance de l'application après son déploiement initial.
- **Financement et budgétisation** : Toute budgétisation et financement en dehors des coûts de développement.
- **Sauvegardes et récupérations** : La gestion continue des sauvegardes et des récupérations.
- **Gestion de la qualité** : Tout audit de qualité continu ou gestion de la qualité post-développement.

3 Acteurs du Projet

3.1 Équipe Interne

Chef de Projet : Responsable de la gestion du projet, de la coordination entre les différentes équipes et des relations avec les clients et les parties prenantes.

Développeurs Backend : Chargés de la mise en place de la logique métier, des APIs et de la gestion de la base de données.

Développeurs Frontend : Responsables de la création de l'interface utilisateur en fonction des besoins spécifiés.

Designer UX/UI : Chargé de l'expérience utilisateur et de la conception visuelle de l'application.

Testeurs : Assurent que l'application est exempte de bugs et respecte les critères de qualité établis.

DevOps : Gère l'intégration et le déploiement continu ainsi que la maintenance de l'infrastructure technique.

Spécialistes en Sécurité : Assurent que l'application respecte toutes les normes et réglementations en matière de sécurité des données.

Équipe Support : Fournit un soutien en cas de problèmes techniques ou questions de la part des utilisateurs.

3.2 Partenaires Externes

STRIPE : Partenaire pour la gestion des paiements.

Fournisseurs de Services OAuth : Google, Facebook, Apple, etc. pour l'authentification.

Fournisseurs d'API de Géolocalisation : Partenaires pour les fonctionnalités de suivi en temps réel à définir.

3.3 Utilisateurs

Clients : Ceux qui vont commander les services.

Joggeurs : Les prestataires de service qui vont répondre aux demandes des clients.

3.4 Parties Prenantes

Investisseurs/Clients : Personnes ou entités qui ont investi dans le projet et souhaitent un retour sur investissement.

4 Exigences Fonctionnelles

4.1 Interface Utilisateur (Frontend)

Cette section détaille les exigences fonctionnelles liées à l'interface utilisateur, souvent désignée par le terme "Frontend". Le but est de fournir une expérience utilisateur intuitive, sécurisée, et efficace. L'interface couvre plusieurs aspects, de l'authentification à la commande de services, en passant par la

recherche et le suivi des missions. Chaque sous-section décrit les fonctionnalités essentielles requises pour une implémentation réussie.

4.1.1 Authentification et Inscription

Inscription : Les nouveaux utilisateurs doivent pouvoir s'inscrire en fournissant un identifiant, un mot de passe, et d'autres informations optionnelles comme le numéro de téléphone et l'adresse email.

Connexion : Les utilisateurs inscrits doivent pouvoir se connecter en utilisant leur identifiant et leur mot de passe.

Mot de Passe Oublié : Les utilisateurs doivent avoir la possibilité de réinitialiser leur mot de passe via leur adresse email.

Déconnexion : Les utilisateurs doivent pouvoir se déconnecter de leur session active.

4.1.2 Recherche de Missions

Barre de Recherche : Les utilisateurs doivent pouvoir effectuer une recherche de missions en utilisant une barre de recherche interactive.

Filtres de Recherche : Des filtres supplémentaires comme le type de mission, la localisation, le coût, etc., doivent être disponibles pour affiner la recherche.

Vue Carte/Liste : Les utilisateurs doivent pouvoir basculer entre une vue en liste et une vue sur carte pour les résultats de recherche.

4.1.3 Commande de Service

Sélection de Type de Mission : Avant de commander, les utilisateurs doivent pouvoir sélectionner le type de mission qu'ils souhaitent (ex : livraison, manutention, etc.).

Saisie des Détails : Les utilisateurs doivent remplir des champs obligatoires et optionnels pour décrire la mission : type d'objet, dimensions, adresse de collecte, adresse de livraison, instructions spéciales, etc.

Estimation du Coût : Avant de confirmer, une estimation du coût doit être affichée.

Confirmation de Commande : Les utilisateurs doivent pouvoir confirmer leur commande et procéder au paiement.

4.1.4 Suivi de Mission

Statut en Temps Réel : Les utilisateurs doivent pouvoir suivre le statut de leur mission en temps réel (en attente, en cours, terminé).

Traçabilité GPS : Une carte doit être disponible pour suivre le trajet du jogger en temps réel.

Notifications d'Étape : Les utilisateurs doivent recevoir des notifications à chaque étape clé : prise en charge, arrivée à destination, etc.

4.1.5 Historique des Missions

Liste des Missions Passées : Les utilisateurs doivent pouvoir consulter l'historique de toutes leurs missions précédentes.

Détails des Missions Passées : Possibilité de voir les détails de chaque mission terminée : coût, durée, jogger, etc

4.1.6 Profil Utilisateur

- **Description** : Les utilisateurs peuvent créer et modifier leur profil.
- **Exigences** :
 - Ajout/modification de la photo de profil, nom, email, numéro de téléphone.
 - Visualisation et modification des préférences de compte.

4.1.7 Notifications

- **Description** : Les utilisateurs reçoivent des notifications en temps réel.
- **Exigences** :
 - Notifications push pour les nouveaux événements liés aux missions (démarrage, fin, annulation, etc.).

4.1.8 Appel Anonymisé

- **Description** : Fonction permettant un appel anonymisé entre le client et le prestataire de service.
- **Exigences** :
 - Option d'appel à partir de l'interface de suivi de mission.

4.1.9 Paiement

- **Description** : Les utilisateurs peuvent effectuer des paiements en ligne.
- **Exigences** :
 - Intégration avec des plateformes de paiement sécurisées.
 - Historique des transactions.

4.1.10 Evaluation et Commentaires

- **Description** : Les utilisateurs peuvent évaluer les missions complétées et laisser des commentaires.
- **Exigences** :
 - Système de notation à étoiles.

- Option pour laisser des commentaires textuels.

4.1.11 FAQ et Assistance

- **Description** : Section dédiée aux questions fréquemment posées et à l'assistance client.
- **Exigences** :
 - FAQ dynamique, modifiable à partir du Backoffice.
 - Lien vers les options de support client.

4.1.12 Gestion des Favoris

- **Description** : Les utilisateurs peuvent ajouter des missions ou des services à une liste de favoris.
- **Exigences** :
 - Option pour marquer une mission ou un service comme favori.
 - Liste des favoris accessible depuis le profil utilisateur.

4.1.13 Paramètres de Confidentialité

- **Description** : Les utilisateurs peuvent gérer les paramètres liés à leur confidentialité.
- **Exigences** :
 - Options pour gérer les préférences de partage de données et les permissions de l'application.

4.1.14 Langues

- **Description** : Les utilisateurs peuvent changer la langue de l'application
- **Exigences** :
 - Sélection d'une liste prédéfinie de langues.

(Anglais/Français)

4.2 Interface d'Administration (Backoffice)

Cette section met en lumière les exigences fonctionnelles associées à l'interface d'administration du système, souvent appelée "Backoffice". Cette interface est destinée aux administrateurs du système et permet de gérer de manière exhaustive tous les aspects opérationnels, analytiques et administratifs de la plateforme. Le Backoffice est essentiel pour assurer un suivi efficace, une analyse détaillée et une gestion optimale des ressources. Les sous-sections suivantes décrivent en détail les fonctionnalités principales qui devront être mises en place.

4.2.1 Authentification et Gestion de Compte

- **Description** : L'interface doit permettre une authentification sécurisée pour les administrateurs via des identifiants et mots de passe uniques.
- **Exigences** :

- Implémentation d'un système de vérification à deux facteurs.
- Possibilité de réinitialiser les mots de passe oubliés via un lien envoyé par email.

4.2.2 Suivi en Temps Réel des Joggers

- **Description** : Le Backoffice doit fournir une carte interactive montrant les positions en temps réel de tous les joggers actifs.
- **Exigences** :
 - Utilisation de l'API Google Maps ou d'une alternative pour la visualisation cartographique.
 - Option pour filtrer par statut (actif, inactif).

4.2.3 Gestion des Missions

- **Description** : Les administrateurs doivent être capables de voir, filtrer, et trier les missions en cours et terminées.
- **Exigences** :
 - Possibilité de trier les missions par date, statut, jogger ou client.
 - Accès à un historique complet des missions pour des analyses détaillées.

4.2.4 Statistiques et Analyse des Données

- **Description** : Un tableau de bord présentant des indicateurs clés de performance (KPI) doit être accessible.
- **Exigences** :
 - Graphiques dynamiques pour le suivi des revenus, des utilisations, et d'autres métriques pertinentes.
 - Export des données en formats CSV ou Excel pour des analyses externes.

4.2.5 Gestion des Clients et des Joggers

- **Description** : Les administrateurs doivent être en mesure de gérer les informations des clients et des joggers.
- **Exigences** :
 - Possibilité d'ajouter, de modifier ou de supprimer des comptes.
 - Outils pour vérifier et valider les documents soumis par les joggers.

4.2.6 Paramétrage Tarifaire

- **Description** : L'interface doit permettre une personnalisation des grilles tarifaires pour les différents services offerts.
- **Exigences** :
 - Flexibilité pour créer des formules tarifaires variables.
 - Option pour appliquer des remises ou des promotions de manière ciblée.

4.2.7 Notifications et Alertes

- **Description** : Système de notifications pour alerter les administrateurs en cas d'événements importants.
- **Exigences** :
 - Notifications par email ou via l'interface pour des événements tels que nouvelles inscriptions, missions terminées, etc.

4.2.8 Sécurité et Conformité

- **Description** : Le Backoffice doit être sécurisé et conforme aux réglementations en vigueur.
- **Exigences** :
 - Respect des normes GDPR pour la protection des données.
 - Journalisation des actions administratives pour un audit éventuel.

4.2.9 Gestion des Réclamations et du Support Client

- **Description** : Le Backoffice doit fournir une section dédiée à la gestion des plaintes, des réclamations et du support aux clients.
- **Exigences** :
 - Ticketing système pour suivre les demandes des clients.
 - Options pour assigner des tickets à des membres spécifiques de l'équipe support.

4.2.10 Gestion des Paiements et des Facturations

- **Description** : Cette fonctionnalité permet la surveillance et le contrôle des transactions financières.
- **Exigences** :
 - Historique des transactions, triable par client, date ou montant.
 - Outil pour générer des factures et des récapitulatifs financiers.

4.2.11 Rapports et Historiques

- **Description** : Le Backoffice devrait permettre la création de rapports périodiques pour une analyse plus profonde.
- **Exigences** :
 - Capacité à créer des rapports personnalisés en fonction des critères choisis.
 - Exportation de ces rapports en formats PDF, CSV ou Excel.

4.2.12 Intégrations Externes

- **Description** : Le système doit permettre des intégrations avec d'autres plateformes ou services.
- **Exigences** :
 - API pour permettre l'intégration avec d'autres systèmes comme les CRM, ERP, etc.

4.2.13 Gestion des Contenus

- **Description** : Permettre l'édition de contenus statiques du site ou de l'application.
- **Exigences** :
 - Éditeur de texte pour modifier les contenus des pages, les FAQ, etc.

4.2.14 Planification et Scheduling

- **Description** : Cette fonctionnalité permet de planifier des événements, des maintenances ou des missions spéciales.
- **Exigences** :
 - Calendrier intégré avec option de planification et rappels.

4.2.15 Mises à jour et Maintenance

- **Description** : Le Backoffice doit permettre de gérer les mises à jour de l'application (formule de prix).
- **Exigences** :
 - Outil pour annoncer les mises à jour ou des périodes de maintenance aux utilisateurs.

4.2.16 Tests et Feedbacks

- **Description** : Collecter et gérer les retours des utilisateurs pour amélioration continue.
- **Exigences** :
 - Outils pour suivre les retours des clients, avec option pour gérer des enquêtes de satisfaction.

4.2.17 Gestion Multilingue

- **Description** : Le système doit supporter plusieurs langues pour une utilisation globale.
- **Exigences** :
 - Options pour ajouter et traduire des langues dans l'interface d'administration.

4.3 Service de Paiement

Cette section décrit les fonctionnalités relatives au paiement dans l'application. L'objectif est de fournir un processus de paiement sécurisé, transparent et pratique pour les utilisateurs.

4.3.1 Sélection du Mode de Paiement

- **Description** : Les utilisateurs peuvent choisir parmi plusieurs modes de paiement.
- **Exigences** :
 - Intégration avec des plateformes de paiement sécurisées.
 - Option pour sauvegarder des informations de paiement pour des transactions futures.

4.3.2 Confirmation de Paiement

- **Description** : Les utilisateurs reçoivent une confirmation du paiement effectué.
- **Exigences** :
 - Émission d'un reçu ou d'une facture par e-mail.
 - Affichage d'un récapitulatif de la transaction dans l'application.

4.3.3 Gestion des Erreurs de Paiement

- **Description** : En cas d'erreur de paiement, l'utilisateur est informé et guidé pour résoudre le problème.
- **Exigences** :
 - Messages d'erreur explicites.
 - Options pour retenter le paiement ou contacter le support.

4.3.4 Historique des Transactions

- **Description** : Les utilisateurs peuvent consulter l'historique de leurs transactions.
- **Exigences** :
 - Liste des transactions effectuées avec détails (date, montant, service, etc.).

4.3.5 Remboursements et Annulations

- **Description** : Les utilisateurs peuvent demander des remboursements ou annuler des transactions.
- **Exigences** :
 - Option pour annuler une mission avant son démarrage. (Condition de temps par défaut : **20 Min**)
 - Politique de remboursement clairement énoncée dans l'application (si la prestation n'a pas été réalisée).

4.3.6 Sécurité des Paiements

- **Description** : Les mesures de sécurité liées au processus de paiement.
- **Exigences** :
 - Chiffrement des données de paiement.
 - Conformité avec les normes de sécurité de l'industrie (comme PCI DSS).

4.4 API et Services Tiers

Cette section détaille les exigences et les fonctionnalités relatives à l'intégration des API et des services tiers utilisés pour enrichir les fonctionnalités et améliorer l'efficacité de l'application JojoApp.

4.4.1 Google Maps API

- **Description** : Intégration de Google Maps pour le suivi en temps réel, la navigation et la géolocalisation.
- **Exigences** :
 - Affichage de la carte avec les points de ramassage et de livraison.
 - Estimation du temps de trajet et des coûts.

Sous-points :

- Affichage des conditions de trafic en temps réel
- Suggestions d'itinéraires alternatifs
-

4.4.2 Service de SMS

- 4.4.2 Service de SMS
- **Description** : Service pour envoyer des messages texte pour la vérification, les notifications, etc.
- **Exigences** :
 - Envoi de SMS pour l'authentification à deux facteurs.
 - Notifications sur l'état de la mission.

4.4.3 Passerelles de Paiement

- **Description** : Intégration de passerelles de paiement pour la gestion des transactions.
- **Exigences** :
 - Support de plusieurs méthodes de paiement.
 - Sécurité des transactions financières.

Sous-points :

- Gestion des remboursements
- Suivi des transactions échouées

4.4.4 Système de Gestion des Avis (Review)

- **Description** : Intégration de passerelles de paiement pour la gestion des transactions.
- **Exigences** :
 - Support de plusieurs méthodes de paiement.
 - Sécurité des transactions financières.

Sous-points :

- Gestion des remboursements
- Suivi des transactions échouées

4.4.5 Service de Support Client

- **Description** : Intégration d'un service de support client tiers, tel qu'un chat en direct ou un système de ticket.
- **Exigences** :
 - Accessibilité du service de support depuis l'application.
 - Stockage des interactions pour analyse future.

Sous-points :

- FAQ intégrée
- Base de données des interactions pour des améliorations futures

4.4.6 Analytics

- **Description** : Intégration d'outils d'analyse pour suivre les comportements des utilisateurs, les conversions, etc.
- **Exigences** :
 - Suivi des KPIs clés.
 - Analyse des taux de conversion.

4.4.7 Appels Anonymisés

- **Description** : Intégration d'un service d'appels anonymisés pour les communications entre clients et prestataires.
- **Exigences** :
 - Masquage des numéros de téléphone.

Sous-points :

- Journaux d'appels pour audit

4.5 Notifications

Cette section couvre les exigences et les fonctionnalités nécessaires pour informer les utilisateurs des mises à jour, des changements d'état de mission, et d'autres informations pertinentes via des notifications.

4.5.1 Notifications Push

- **Description** : Envoi de notifications push pour les événements critiques comme la nouvelle mission, l'achèvement de la mission, etc.
- **Exigences** :
 - Alertes immédiates pour les nouveaux événements.
 - Priorisation des notifications selon l'urgence.

Sous-points :

- Personnalisation des sons de notification
- Gestion des préférences de notification dans les paramètres utilisateur

4.5.2 Notifications par Email

- **Description** : Envoi de notifications par email pour les événements moins urgents comme les récapitulatifs, les factures, etc.
- **Exigences** :
 - Formatage des emails pour lisibilité et cohérence.
 - Contenu informatif et pertinent.

Sous-points :

- Option de désabonnement
- Suivi des taux d'ouverture et de clics

4.5.3 Notifications In-App

- **Description** : Affichage des notifications à l'intérieur de l'application pour informer les utilisateurs pendant qu'ils sont actifs dans l'application.
- **Exigences** :
 - Design non intrusif.
 - Tri des notifications par type et par importance.

Sous-points :

- Historique des notifications in-app
- Options de marquage "lu/non-lu"

4.5.4 SMS

- **Description** : Envoi de notifications SMS pour des informations cruciales comme l'OTP, les changements de statut de la mission, etc.
- **Exigences** :
 - Envoi rapide et fiable.
 - Messages courts et clairs.

Sous-points :

- Option pour activer/désactiver les notifications SMS
- Conservation des logs pour suivi et audits

4.5.5 Notifications de Système

- **Description** : Notifications à l'échelle du système pour informer les utilisateurs de mises à jour de l'application, de maintenance prévue, etc.
- **Exigences** :
 - Notification préalable pour toute maintenance ou mise à jour.
 - Messages informatifs détaillant la portée de l'événement.

Sous-points :

- Canal de communication d'urgence pour les incidents critiques
- Options pour l'utilisateur de reporter des bugs ou des problèmes directement via la notification

4.6 Sécurité

Cette section détaille les exigences et fonctionnalités relatives à la sécurité de l'application, garantissant l'intégrité, la confidentialité et la disponibilité des données utilisateur et système.

4.6.1 Authentification à Deux Facteurs (2FA)

- **Description** : Mécanisme supplémentaire d'authentification pour garantir que seul l'utilisateur autorisé accède au compte.

- **Exigences :**
 - Support de diverses méthodes de 2FA, y compris OTP via SMS et applications d'authentification tierces.

Sous-points :

- Fallback pour récupération du compte
- Verrouillage du compte après un certain nombre de tentatives échouées

4.6.2 Chiffrement des Données

- **Description :** Chiffrement des données sensibles lors du stockage et de la transmission.
- **Exigences :**
 - Utilisation de protocoles de chiffrement reconnus et sécurisés, tels que TLS pour le transfert de données.

Sous-points :

- Chiffrement des données au repos
- Gestion sécurisée des clés de chiffrement

4.6.3 Contrôle d'Accès

- **Description :** Définition et gestion des rôles et permissions au sein de l'application.
- **Exigences :**
 - Possibilité d'établir différents niveaux de rôles (utilisateur, administrateur, super-administrateur, etc.)

4.6.4 Protection contre les Attaques Communes

- **Description :** Mesures pour prévenir les attaques telles que l'injection SQL, le Cross-Site Scripting (XSS), etc.
- **Exigences :**
 - Filtrage et validation des entrées utilisateur.

Sous-points :

- Tests de pénétration réguliers
- Mises à jour de sécurité fréquentes

4.6.5 Surveillance et Alertes de Sécurité

- **Description :** Surveillance en temps réel des activités suspectes et des violations de la politique de sécurité.
- **Exigences :**
 - Système d'alerte en temps réel pour les activités suspectes.

Sous-points :

- Notifications aux administrateurs en cas d'activités suspectes
- Capacité à bloquer des adresses IP ou des utilisateurs spécifiques

4.6.6 Conformité Réglementaire

- **Description** : Assurer que l'application est en conformité avec les réglementations locales et internationales en matière de protection des données.
- **Exigences** :
 - Conformité aux réglementations telles que le RGPD en Europe.

Sous-points :

- Processus de consentement explicite
- Portabilité et suppression des données utilisateur

4.7 Communication

Cette section couvre les exigences et les fonctionnalités associées à la communication entre les utilisateurs, les administrateurs et les systèmes externes via l'application JojoApp.

4.7.1 Appel Anonymisé

- **Description** : Fonctionnalité permettant aux utilisateurs de contacter les chauffeurs sans révéler leur numéro de téléphone personnel.
- **Exigences** :
 - Intégration avec des services tiers pour la gestion des appels anonymes.

Sous-points :

- Journalisation des appels pour des raisons de sécurité

4.7.2 Notifications Push

- **Description** : Notifications en temps réel pour informer les utilisateurs de l'état des missions, des mises à jour de l'application, etc.
- **Exigences** :
 - Divers niveaux de priorité pour les notifications.

Sous-points :

- Notifications pour les administrateurs
- Possibilité pour les utilisateurs de personnaliser les notifications

4.7.3 Email & SMS

- **Description** : Communication automatisée par email et SMS pour les confirmations, rappels, et autres interactions importantes.
- **Exigences** :
 - Utilisation de services de messagerie tiers fiables pour l'envoi d'emails et de SMS.

Sous-points :

- Template d'email personnalisables
- Historique et suivi des communications

4.7.4 Intégration avec des Outils de Support Client

- **Description** : Intégration support client

- **Exigences** : Numéro de téléphone du support client.

4.7.5 API de Communication Externe

- **Description** : API pour permettre aux partenaires et services tiers d'interagir avec l'application.
- **Exigences** :
 - Documentation complète de l'API.

Sous-points :

- Limitation des taux d'appel
- Sécurité et authentification pour l'accès à l'API

5 Exigences Non-Fonctionnelles

Ce point du cahier des charges s'intéresse aux aspects non fonctionnels du projet JojoApp, y compris la performance, la sécurité, la conformité, l'expérience utilisateur et autres aspects qui affectent la qualité du système sans être directement liés à des fonctionnalités spécifiques.

5.1 Performance

Ce point aborde les exigences liées à la performance de l'application JojoApp, notamment en ce qui concerne le temps de réponse, la gestion des ressources, et l'évolutivité du système. L'objectif est de garantir une expérience utilisateur fluide et réactive.

5.1.1 Temps de Chargement

- Temps de démarrage de l'application : Doit être inférieur à 3 secondes.
- Temps de chargement des pages : Ne doit pas excéder 2 secondes.
- Temps de réaction aux actions utilisateur : Maximum de 500 ms.

5.1.2 Scalabilité

- Gestion des pics de trafic : Le système doit être capable de supporter jusqu'à 10 000 utilisateurs simultanés.
- Augmentation linéaire des ressources : Le système doit être conçu pour permettre une augmentation linéaire des ressources en fonction du nombre d'utilisateurs.

5.1.3 Réactivité

- Délai de réponse aux actions utilisateur : Doit être immédiat ou quasi-immédiat (moins de 500 ms).
- Délai de mise à jour des informations en temps réel (comme le suivi des missions) : Moins de 2 secondes.

5.1.4 Latence de l'API

- Temps de réponse de l'API : Moins de 200 ms pour 95% des requêtes

5.1.5 Optimisation des Requêtes

- Toutes les requêtes vers la base de données doivent être optimisées pour minimiser le temps de chargement.

5.1.6 Gestion du Cache

- Utilisation du cache pour les données fréquemment consultées afin de réduire le temps de réponse

5.1.7 Gestion des Fichiers Médias

- Compression des images et vidéos pour réduire le temps de chargement.

5.1.8 Traitement en Arrière-Plan

- Toutes les opérations lourdes (comme le calcul des itinéraires) doivent être effectuées en arrière-plan pour ne pas affecter la réactivité de l'interface utilisateur.

5.2 Sécurité

5.2.1 Authentification

Le système doit utiliser une méthode d'authentification robuste, comme l'authentification à deux facteurs, pour assurer que seuls les utilisateurs autorisés ont accès à l'application.

5.2.2 Chiffrement

Toutes les données sensibles, y compris les informations de paiement et les données personnelles, doivent être chiffrées en utilisant des algorithmes de chiffrement modernes et fiables.

5.2.3 Audit et Suivi

Des logs d'audit seront conservés pour toutes les opérations critiques afin de permettre un suivi en cas de problèmes de sécurité ou pour des besoins de conformité.

5.2.4 Contrôle d'Accès

Des niveaux de rôles et permissions seront définis pour limiter l'accès aux différentes parties de l'application selon les besoins fonctionnels.

5.2.5 Protection contre les Attaques

Des mesures de sécurité comme des pare-feu, des systèmes de détection d'intrusion, et la protection contre les attaques par déni de service seront mises en place.

5.2.6 Mise à jour de Sécurité

Les mises à jour de sécurité seront appliquées régulièrement pour protéger contre les vulnérabilités connues.

5.2.7 Test de Pénétration

Des tests de pénétration seront effectués périodiquement pour évaluer la robustesse du système contre les menaces externes.

5.2.8 Sauvegarde et Restauration

Des plans de sauvegarde et de restauration seront en place pour récupérer rapidement en cas d'incident de sécurité.

5.2.9 Formation et Sensibilisation

Le personnel sera formé sur les meilleures pratiques de sécurité et sera sensibilisé aux menaces courantes, comme le phishing.

5.2.10 Conformité

Le système doit être en conformité avec les réglementations locales et internationales sur la sécurité des données, telles que le RGPD en Europe.

5.3 Expérience Utilisateur

5.3.1 Temps de Chargement des Pages

Le temps de chargement des pages ne doit pas excéder 2 secondes pour maintenir une expérience utilisateur fluide.

5.3.2 Intuitivité de l'Interface

L'interface doit être conçue de manière qu'un nouvel utilisateur puisse comprendre les fonctionnalités de base de l'application en moins de 5 minutes.

5.3.3 Accessibilité

L'application doit être accessible et utilisable sur smartphones.

5.3.4 Support Multilingue

L'application doit au minimum supporter le français et l'anglais pour s'adresser à une audience plus large.

5.3.5 Feedback Utilisateur

Des messages de confirmation ou d'erreur clairs doivent être affichés à l'utilisateur pour indiquer le résultat d'une action.

5.3.6 Aide en ligne et Documentation

Une section FAQ détaillée et des guides d'utilisation doivent être disponibles pour aider les utilisateurs.

5.3.7 Récupération de Compte

Les options de récupération de compte doivent être simples et sécurisées, permettant à l'utilisateur de récupérer son compte via son adresse e-mail ou son numéro de téléphone.

5.3.8 Continuité de Service

En cas de perte de connexion, l'application doit sauvegarder l'état actuel de l'utilisateur pour permettre une reprise rapide de l'activité.

5.3.9 Conformité RGPD

L'application doit être conforme au Règlement Général sur la Protection des Données, notamment en ce qui concerne la collecte et le stockage des données utilisateur.

5.3.10 Interactions Tactiles

Les éléments d'interface doivent être facilement cliquables ou tapables, avec des zones d'interaction

suffisamment grandes pour éviter les erreurs.

5.3.11 Personnalisation

L'application doit permettre une certaine personnalisation de l'interface, comme le choix des couleurs ou l'affichage des informations, pour améliorer l'engagement utilisateur.

5.4 Fiabilité

Ce point couvre tous les aspects de la fiabilité du projet JojoApp, assurant ainsi un service constant et de haute qualité pour les utilisateurs.

5.4.1 Disponibilité du Service

Le service doit être disponible 24/7, avec un taux de disponibilité d'au moins 99.9%.

5.4.2 Gestion des Erreurs

Le système doit pouvoir identifier et récupérer les erreurs de manière à minimiser l'impact sur l'utilisateur final.

5.4.3 Redondance des Serveurs

Une redondance doit être mise en place pour assurer la continuité du service en cas de défaillance d'un ou plusieurs composants.

5.4.4 Sauvegardes Régulières

Des sauvegardes complètes du système et des données utilisateur doivent être effectuées régulièrement pour permettre une récupération rapide en cas de perte de données.

5.4.5 Monitoring en Temps Réel

Le système doit être surveillé en temps réel pour détecter toute anomalie pouvant affecter la fiabilité.

5.4.6 Tests de Charge

Des tests de charge réguliers doivent être effectués pour s'assurer que le système peut gérer un grand nombre de requêtes simultanées.

5.4.7 Mises à Jour Planifiées

Toutes les mises à jour du système doivent être planifiées et communiquées aux utilisateurs pour minimiser les interruptions de service.

5.4.8 Gestion de la Capacité

Le système doit être conçu de manière à être évolutif, afin de gérer une augmentation du nombre d'utilisateurs et de transactions.

5.4.9 Mécanismes de Reprise après Panne

Des mécanismes de reprise après panne doivent être en place pour minimiser le temps d'arrêt et la perte de données en cas de défaillance du système.

5.4.10 Journalisation et Audit

Les activités du système doivent être enregistrées de manière sécurisée pour faciliter les audits et le dépannage.

5.5 Maintenabilité

Ce point détaille tous les éléments nécessaires pour assurer une maintenabilité efficace et durable du projet JojoApp, ce qui est crucial pour sa longévité et sa fiabilité.

5.5.1 Documentation du Code

Le code source doit être entièrement documenté pour faciliter la compréhension et la maintenance ultérieure.

5.5.2 Modularité

Le code doit être écrit de manière modulaire pour permettre des mises à jour et des améliorations sans affecter le reste du système.

5.5.3 Versionnage

Un système de versionnage doit être mis en place pour suivre les différentes versions du logiciel et faciliter les rollbacks en cas de problèmes.

5.5.4 Tests Unitaires et d'Intégration

Des tests unitaires et d'intégration doivent être réalisés pour chaque composant du système afin de garantir que les mises à jour ou les ajouts de nouvelles fonctionnalités n'introduisent pas de bugs.

5.5.5 Plan de Maintenance

Seul un personnel qualifié doit avoir accès aux environnements de maintenance et de production.

5.5.6 Accès à la Maintenance

Seul un personnel qualifié doit avoir accès aux environnements de maintenance et de production.

5.5.7 Révisions de Code

Un processus de revue de code doit être mis en place pour s'assurer que le code correspond aux standards de qualité et de performance.

5.5.8 Formation du Personnel

Le personnel doit être formé pour comprendre le système en profondeur, ce qui facilite la résolution rapide des problèmes.

5.5.9 Outils de Diagnostic

Des outils de diagnostic doivent être disponibles pour identifier et résoudre les problèmes rapidement.

5.5.10 Gestion des Dépendances

La gestion des bibliothèques et des dépendances externes doit être automatisée pour assurer une maintenance plus facile et plus fiable.

5.5.11 Gestion et Mise à jour du Contenu

Note : La responsabilité de 2CConseil dans ce projet se limite au développement initial de l'application et ne couvre pas la gestion et la mise à jour du contenu après cette phase. La gestion continue du contenu, y compris les mises à jour, les corrections, et les ajouts de nouvelles fonctionnalités, devra être assurée par un tiers ou être soumise à un nouvel accord.

5.6 Contraintes Matérielles et Logicielles

5.6.1 Système d'Exploitation

L'application doit être compatible avec les systèmes d'exploitation mobiles les plus répandus, tels qu'Android et iOS.

5.6.2 Navigateurs Web

Pour la version web de l'application, la compatibilité doit être assurée avec les navigateurs les plus couramment utilisés comme Chrome, Firefox, Safari, et Edge.

5.6.3 Serveurs

Le serveur back-end doit être capable de gérer un trafic important et d'évoluer en fonction des besoins (scalabilité).

5.6.4 Base de Données

Les contraintes liées à la base de données doivent être prises en compte, notamment en termes de taille, de performances et de sécurité.

5.6.5 Stockage

Des solutions de stockage suffisamment vastes et fiables doivent être mises en place pour les données utilisateur et les logs.

5.6.6 Bande Passante

La bande passante nécessaire pour les opérations de l'application doit être évaluée et prévue.

5.6.7 Terminaux Mobiles

Les exigences en matière de matériel pour les terminaux mobiles doivent être spécifiées, comme la version minimale du système d'exploitation, la résolution d'écran, etc.

5.6.8 APIs et Bibliothèques Externes

Toute dépendance à des services tiers doit être documentée, incluant les versions compatibles et les licences.

5.6.9 Limitations du Réseau

Les contraintes de réseau, notamment en termes de latence et de disponibilité, doivent être identifiées.

5.6.10 Environnement de Développement

Les outils et environnements de développement compatibles doivent être spécifiés (IDEs, langages de programmation, etc.)

5.7 Législation et Conformité

5.7.1 Protection des Données Personnelles

Le projet doit se conformer au Règlement général sur la protection des données (RGPD) pour assurer la protection des données personnelles des utilisateurs.

5.7.2 Propriété Intellectuelle

Tous les aspects liés à la propriété intellectuelle, tels que l'utilisation de bibliothèques sous licence, doivent être clairement définis et respectés.

5.7.3 Paiements en Ligne

La solution doit se conformer aux normes de sécurité PCI DSS pour les paiements en ligne.

5.7.4 Accessibilité

Le projet doit être conforme aux directives sur l'accessibilité des contenus web (WCAG) pour garantir son accessibilité à tous les utilisateurs.

5.7.5 Contrats avec des Tiers

Toute collaboration avec des services ou fournisseurs tiers doit être conforme aux lois et réglementations applicables, et cela doit être stipulé dans les contrats.

5.7.6 Impôts et Taxes

Le système doit être capable de gérer la TVA et autres taxes applicables en fonction de la localisation des utilisateurs.

5.7.7 Cybersécurité

La législation relative à la cybersécurité, y compris les obligations de notification en cas de violation de données, doit être respectée.

5.7.8 Conditions Générales d'Utilisation (CGU)

Les CGU doivent être rédigées en conformité avec la législation applicable, et elles doivent être facilement accessibles aux utilisateurs de l'application.

5.7.9 Code de Conduite pour les Chauffeurs et Utilisateurs

Un code de conduite doit être établi et conforme aux lois locales et nationales pour régir le comportement des chauffeurs et des utilisateurs.

5.7.10 Licences et Permis

Les licences et permis nécessaires pour opérer le service, notamment pour le transport de personnes ou de biens, doivent être obtenus et maintenus à jour.

6 Spécifications Techniques

6.1 Architecture Logicielle

6.1.1 Backend

- **Langage de Programmation** : JavaScript / SQL
- **Architecture** : BaaS / Monolithique
- **Services de Backend** : Appwrite
- **Communication** : API RESTful / WebSockets
- **Sécurité** : JWT / OAuth 2.0 / TLS / SSL / CORS / Rate Limiting
- **Stockage** : MariaDB / MongoDB
- **Caching** : Redis
- **File d'attente des messages** : RabbitMQ

6.1.2 Frontend

- **Technologies** : React Native / React Native Libs
- **Gestion de l'état** : Zustand/Redux
- **Communication avec Backend** : HTTPS / API RESTful / WebSockets
- **Sécurité** : OAuth2 / JSON Web Tokens
- **Responsivité** : React Native Paper

6.1.3 Bases de Données

- **Base de Données Principale** : MariaDB / MongoDB

- **Base de Données de Session** : MariaDB / MongoDB
- **Object-Relational Mapping (ORM)** : Appwrite Web SDK

6.1.4 APIs et Services Tiers

- **Géolocalisation** : Leaflet / react-native-geocoding / react-native-maps
- **Paiement** : Stripe API
- **Vérification d'Identité** : OAuth2 / JSON Web Tokens
- **Appels anonymisés** : Twilio / Sinch / Plivo

6.1.5 Authentification et Sécurité

- **Authentification** : OAuth2 / JSON Web Tokens
- **Cryptage** : TLS / bcrypt / AES / RSA
- **Stockage sécurisé des mots de passe** : Appwrite
- **Firewall et IDS (Système de Détection d'Intrusion)** : ufw

6.2 Matériel et Infrastructure

6.2.1 Serveurs

- **Provider** : AWS
- **Type** : Instances EC2
- **Region** : Paris – France (eu-west-3)

6.2.2 Stockage

- **Region** : Paris – France (eu-west-3)

6.2.3 Réseau

- **Service DNS** : Route 53

6.2.4 Sécurité

- **Firewalls** : ufw / Security Groups
- **Roles** : IAM
- **Reverse Proxy** : Caddy

6.2.5 Environnements

6.2.6 Automatisation CI/CD

6.2.7 Coût

- **Serveurs** : 2 serveurs (backend / back-office avec deux environnements minimum (dev-préprod

/ prod))

- **Estimation minimale** : 80 euros/mois

6.2.8 Conformité et Audits

- **Audits** : Réalisation d'audits régulier pour identifier et corriger les vulnérabilités / mettre à jour les serveurs et les différentes technos.
- **Protection des données** : Respect de la RGPD (accès aux données, stockage sécurisé, politique claire et accessible)

6.2.9 Gestion des Catastrophes

- **Backups réguliers**
- **Stockage des backups sur serveurs externes**
- **Process de rétablissement des backups**

6.2.10 Documentation

- **Architecture** : Architecture des serveurs et de l'application accessible

6.3 Environnements de Développement

6.3.1 IDE (Integrated Development Environment)

- **VSCode**

6.3.2 Gestion de Version

- **Git**
- **Jira**

6.3.3 Gestion des Dépendances

- **Node Package Manager (npm)**

6.3.4 Outils de Linting et de Formatage

- **ESLint**
- **Prettier**

6.3.5 Automatisation des Tests

- **Tests E2E : Cypress**

- **Tests Unitaires : Jest**
- **Tests de composants : Jest**

6.3.6 CI/CD

- **GitHub Actions**

6.3.7 Déploiement

- **GitHub Actions**

6.3.8 Monitoring et Logging

- **AWS CloudWatch**
- **AWS CloudTrail**

6.3.9 Revue de Code

- **Revue de code à chaque pull request de l'équipe de développement**

6.3.10 Documentation

- **Documentation générale de l'exécution du projet dans des Readme GitHub**
- **Documentation de code constante**

6.3.11 Collaboration et Communication

- **Collaboration sur les différents projets sur GitHub**

6.4 Procédures de Déploiement

6.4.1 Construction du Code

- **Architecture claire des dossiers de code, conventions de nommages, centralisation des différents services et création de fonctions généralistes**

6.4.2 Tests Automatisés

- **Tests lancés avant et après chaque déploiement en production**

6.4.3 Environnements de Déploiement

- **GitHub Actions**

6.4.4 Migration de la Base de Données

- **Tests des bases de données dans un environnement de préproduction avant migration en production**
- **Vérification des logs constante des BDD**

6.4.5 Mise à Jour des Services Tiers

- **Mise à jour régulière des dépendances, librairies, et outils de stockage**

6.4.6 Rollback

- **Procédure de rollback permettant de revenir rapidement à une version stable antérieure.**

6.4.7 Monitoring Post-déploiement

Post-déploiement, nous activons un niveau de surveillance renforcé pour nos services. À l'aide de AWS CloudWatch, nous surveillons les métriques essentielles pour identifier et traiter rapidement toute anomalie.

6.4.8 Sauvegarde et Restauration

Nous automatiserons des sauvegardes régulières de nos base de données et rétabliront ces backups en cas de problèmes techniques grâce à des scripts automatisés et fiables.

6.4.9 Maintenance

Calendrier des fenêtres de maintenance pour des mises à jour non urgentes ou des améliorations.

6.4.10 Documentation de Déploiement

Chaque déploiement de versions majeures sera documenté pour apporter la liste des éléments et features se trouvant dans le ou les déploiements en question.

6.4.11 Validation

Une fois le déploiement effectué, l'équipe de QA (Quality Assurance) entreprend une série de tests pour valider les nouvelles fonctionnalités et s'assurer qu'il n'y a pas de régressions. Les critères d'acceptation doivent être satisfaits avant de marquer le déploiement comme réussi.

6.5 Normes et Protocoles

6.5.1 Normes de Codage

- Suivre les conventions de codage JavaScript/React Native pour une cohérence du code.
- Utiliser ESLint et Prettier pour maintenir la qualité du code.

6.5.2 Protocoles de Communication

- Utiliser HTTPS pour les communications sécurisées avec les serveurs AWS et Appwrite.
- Mettre en place un système de gestion des erreurs pour les appels d'API.

6.5.3 Normes API

- Créer une API RESTful avec Appwrite en utilisant des endpoints cohérents.
- Utiliser des noms de route compréhensibles et des paramètres d'URL explicites.

6.5.4 Protocoles de Sécurité

- Implémenter l'authentification et l'autorisation sécurisées avec Appwrite.
- Chiffrer les données sensibles en transit et au repos.

6.5.5 Gestion des Données

- Utiliser AWS pour le stockage des données et des médias.
- Mettre en place des sauvegardes régulières des données.

6.5.6 Accessibilité

- Concevoir l'interface utilisateur pour être la plus accessible possible.
- Utiliser une navigation claire et intuitive.

6.5.7 Localisation et Internationalisation

- Prendre en charge plusieurs langues et cultures grâce à la localisation.
- Utiliser des fichiers de ressources pour les textes traduits.

6.5.8 Versioning

- Gestion des versions de l'application à l'aide d'un système de contrôle de version à savoir Git.
- Suivre les principes de gestion de version sémantique.

6.5.9 Documentation

- Fournir une documentation complète pour l'installation, la configuration et l'utilisation de l'application.
- Documenter les API et les composants clés.

6.5.10 Tests

- Mettre en place des tests E2E et des tests d'intégration sur les fonctionnalités clés pour garantir la stabilité de l'application.
- Automatiser les tests à l'aide de frameworks tels que Jest et Appium en priorisant Appium.

6.5.11 Revue de Code

- Effectuer des revues de code régulières pour garantir la qualité du code.
- Utiliser des outils de revue de code comme l'interface GitHub.

6.5.12 Conformité Légale

- Respecter les lois sur la confidentialité des données et la protection des consommateurs.

6.6 Test et Qualité

Cette section couvre l'ensemble des activités de test et des métriques de qualité qui seront utilisées tout au long du cycle de vie du développement du projet JojoApps. L'objectif est d'assurer que l'application est fiable, performante et facile à maintenir.

6.6.1 Stratégie de Test

- **Tests Unitaires** : Tests des composants individuels pour valider leur fonctionnalité.
- **Tests d'Intégration** : Tests visant à valider les interactions entre les différents composants.
- **Tests de Bout-en-Bout (End-to-End)** : Tests pour valider les flux de travail dans l'application en conditions réelles.

6.6.2 Environnements de Test

- **Développement** : Environnement où les développeurs testeront en temps réel pendant le développement.
- **Staging** : Environnement répliquant le serveur de production pour des tests préliminaires avant le déploiement.
- **Production** : Environnement réel où le produit final est déployé.

6.6.3 Méthodologie de Test

- **BDD (Behavior-Driven Development)** : Développement guidé par le comportement attendu du logiciel.
- Adoption du BDD sur les fonctionnalités clés notamment avec le framework Appium.

6.6.4 Outils de Test

- Appium ainsi que Jest

6.6.5 Critères de Qualité

- **Couverture de Code** : Les fonctionnalités clés et à risques côté utilisateur doivent être couvertes par les tests.
- **Taux de Défauts** : Le taux de bugs découverts pendant les tests doit être minimisé.

6.6.6 Performances

- **Temps de Chargement** : Les pages doivent se charger en moins de 3 secondes.
- **Réactivité** : L'application doit répondre aux actions de l'utilisateur en moins de 200 millisecondes.

6.6.7 Sécurité

Effectuer des **tests de pénétration** pour identifier et corriger les vulnérabilités de sécurité.

6.6.8 Accessibilité

Utilisation d'outils comme **axe (jest-axe)** pour tester l'accessibilité de l'application.

6.6.9 Monitoring et Rapports

Utilisation d'outils comme **Sentry** pour le suivi des erreurs et **New Relic** pour le suivi des performances.

6.6.10 Revue de Code

Chaque pull request doit passer une revue de code avant d'être fusionnée.

6.6.11 Automatisation

Utilisation de **CI/CD** pour automatiser le processus de test et de déploiement.

6.7 Documentation Technique

Cette section du cahier des charges sert à expliquer en détail le niveau et le type de documentation qui sera produit tout au long du cycle de vie du projet JojoApps. La documentation technique est cruciale pour comprendre, maintenir, et étendre l'application. Ce plan vise à créer une documentation complète et détaillée, facilitant ainsi la maintenance future et le transfert de connaissances au sein de l'équipe de développement et aux parties prenantes du projet JojoApps.

6.7.1 Introduction et Vue d'Ensemble

- **Présentation du Projet** : Description sommaire du projet, de ses objectifs et de sa portée.
- **Glossaire** : Définition des termes techniques et des acronymes utilisés.

6.7.2 Architecture Logicielle

- **Diagrammes d'Architecture** : UML, diagrammes de flux de données, etc.
- **Description des Modules** : Fonction et responsabilité de chaque module.

6.7.3 Codes Sources

- **Commentaires de Code** : Lignes de commentaires expliquant le fonctionnement du code.
- **Conventions de Codage** : Standards pour le nommage de variables, l'indentation, etc.

6.7.4 API et Services Tiers

- **Documentation API** : Détails sur les endpoints, les méthodes, les paramètres et les réponses.
- **Intégrations Tiers** : Documentation sur comment intégrer des services tiers comme les systèmes de paiement.

6.7.5 Bases de Données

- **Schéma de la Base de Données** : Diagrammes expliquant la structure de la base de données.
- **Dictionnaire de Données** : Description des tables, des champs et des relations.

6.7.6 Procédures de Déploiement

Guides de Déploiement : Étapes à suivre pour déployer l'application dans différents environnements.

6.7.7 Tests

- **Plan de Test** : Stratégies et cas de tests.
- **Rapports de Tests** : Résultats et analyses des tests.

6.7.8 Maintenance et Support

- **Guides de Maintenance** : Procédures de dépannage et de résolution des problèmes courants.

6.7.9 Sécurité

- **Politiques de Sécurité** : Protocoles et pratiques pour maintenir la sécurité.
- **Registre des Incidents** : Un fichier où tous les incidents de sécurité sont notés et analysés.

6.7.10 Licences et Conformités

- **Liste des Licences** : Détails des licences pour chaque bibliothèque ou service tiers utilisé

6.7.11 Historique des Versions

- **Journal des Changements** : Une trace écrite de toutes les modifications apportées au projet, y compris les mises à jour et les corrections de bugs.

6.7.12 Manuels Utilisateur

- **Guides d'Utilisation** : Instructions détaillées sur comment utiliser l'application.

6.8 Maintenance et Support ¹

Cette section couvre les plans et les stratégies détaillées pour la maintenance continue et le support de l'application JojoApps après son déploiement. L'objectif est de garantir une expérience utilisateur optimale et de répondre rapidement aux besoins changeants du marché et aux exigences légales.

6.8.1 Types de Maintenance

- **Maintenance Corrective** : Corrections de bugs et autres problèmes qui se manifestent en production.
- **Maintenance Évolutive** : Ajout de nouvelles fonctionnalités ou améliorations des fonctionnalités existantes.
- **Maintenance Adaptative** : Adaptations aux changements dans l'environnement, tels que les mises à jour de système d'exploitation.
- **Maintenance Préventive** : Mises à jour de sécurité et autres modifications pour prévenir les problèmes futurs.

¹ Voir Annexe Section 11.16 p 89 - 90

6.8.2 Calendrier de Maintenance

- **Mises à Jour Mineures** : Programmées chaque mois.
- **Mises à Jour Majeures** : Semestrielles, ou selon les besoins du marché.

6.8.3 Support Client

- **FAQ** : Foire aux questions pour résoudre les problèmes courants.
- **Billetterie** : Un système de gestion de tickets pour les requêtes plus complexes.

6.8.4 SLA (Service Level Agreement)

- **Temps de Réponse** : Moins de 24 heures pour tous les types de requêtes client.
- **Résolution de Bug** : Moins de 48 heures pour les bugs critiques.

6.8.5 Monitoring

- **Outils de Surveillance** : Utilisation d'outils comme New Relic ou Datadog pour surveiller la performance et l'état de l'application en temps réel.

6.8.6 Sauvegardes

- **Sauvegardes Quotidiennes** : Des copies complètes de la base de données sont sauvegardées chaque jour.
- **Stockage Redondant** : Utilisation de plusieurs disques durs et services de cloud pour le stockage des sauvegardes.

6.8.7 Documentation

- **Manuels d'Utilisation** : Des guides complets pour aider les utilisateurs à naviguer dans l'application.
- **Documentation Technique** : Pour aider l'équipe de maintenance à comprendre le système en détail.

6.8.8 Formation

Formation de l'Équipe de Support : Formation continue sur les fonctionnalités de l'application et les meilleures pratiques du service client. (1)

6.8.9 Gestion des Versions

Utilisation de systèmes de gestion de version comme Git pour suivre les modifications et faciliter la

maintenance.

6.8.10 Révisions d'Exigence

Examen régulier des exigences pour s'assurer qu'elles sont toujours alignées avec les objectifs du projet et les besoins des utilisateurs.

6.9 Conformité et Législation

Dans cette section du cahier des charges, nous allons aborder les différentes réglementations et normes légales que le projet JojoApps doit respecter. Le but est de garantir que l'application est en conformité avec toutes les lois et réglementations applicables.

6.9.1 Protection des Données

- **RGPD (Règlement Général sur la Protection des Données)** : Mesures prises pour le respect de la protection des données personnelles en accord avec les réglementations européennes.
- **Cryptage des Données** : Méthodes et techniques utilisées pour sécuriser les données sensibles.

6.9.2 Transactions Financières

- **PCI DSS (Payment Card Industry Data Security Standard)** : Conformité aux standards pour les transactions par carte bancaire.

6.9.3 Accessibilité

- **WCAG (Web Content Accessibility Guidelines)** : Niveau de conformité pour rendre l'application accessible aux personnes ayant des handicaps.

6.9.4 Droit d'Auteur

- **Licences de Logiciel** : Gestion et vérification des licences pour tous les logiciels et librairies tiers.
- **Propriété Intellectuelle** : Protection des éléments propriétaires tels que le logo, le nom de l'application, etc.

6.9.5 Commerce Électronique

- **Loi sur le Commerce Électronique** : Conformité aux régulations sur le commerce en ligne, y compris les contrats, les déclarations, et les remboursements.

6.9.6 Travail Indépendant et Contrats

- **Régulations du Travail Indépendant** : Assurer que le statut des freelancers ou des travailleurs indépendants est clairement défini et conforme à la loi.

6.9.7 Communication et Marketing

- **CAN-SPAM Act et Loi LCEN** : Conformité aux lois relatives à la communication électronique, notamment les emails et notifications.

6.9.8 Sécurité

- **Normes ISO** : Conformité aux normes ISO en matière de sécurité des systèmes d'information.

6.9.9 Conformité Locale

- **Lois et Règlements Locaux** : Assurer la conformité aux lois spécifiques à la France, ou d'autres juridictions où l'application est déployée.

6.9.10 Audits et Contrôles

- **Audits Internes et Externes** : Plans et méthodologies pour les audits de conformité.
- **Rapports d'Audit** : Présentation régulière de rapports d'audit pour vérifier la conformité continue. (1)

7 Contraintes du Projet

Dans cette section, nous discuterons des différentes contraintes qui pourraient affecter le projet JojoApps. Ces contraintes peuvent être liées au temps, aux ressources, à la technologie ou à d'autres facteurs.

7.1 Contraintes Temporelles

Dans cette section, nous décrivons les diverses contraintes de temps auxquelles le projet JojoApps est soumis. Il est crucial de prendre ces éléments en compte pour une planification et une exécution réussie du projet.

7.1.1 Date de Lancement Prévue

- **Lancement de la Version Bêta** : Le projet vise à lancer une version bêta pour un nombre limité d'utilisateurs dans un délai de 7 mois à compter de la date de début du projet.
- **Lancement Officiel** : Le déploiement complet de l'application est prévu pour 9 mois après le début du projet.

7.1.2 Phases de Développement

- **Phase de Conception** : Durée estimée de 2 semaines.
- **Phase de Développement** : Durée estimée de 5 mois.
- **Phase de Test** : Durée estimée de 2 semaines.
- **Phase de Recette** : Durée estimée de 2 semaines.

7.1.3 Jalons Importants

- **Fin de la Phase de Conception** : deux semaines après le début du projet.
- **Premier Prototype Fonctionnel** : Premier prototype sur au moins un service, environ 2 mois après le début du projet. (Hors back office)
- **Version Alpha** : Une version Alpha sera développée pour se concentrer davantage sur l'application côté client et jogger. Cette version exclura le back-office et visera à tester les fonctionnalités clés. (Portée fonctionnelle à définir)

7.1.4 Contraintes de Sprints

- Chaque sprint durera 2 semaines.
- Revue et planification de sprint toutes les deux semaines.

7.1.5 Révisions et Approbations

- Temps alloué pour la revue par les parties prenantes : 2 semaines avant chaque phase importante.
- Temps alloué pour les approbations légales et de conformité : 1 mois avant le lancement.

7.1.6 Maintenance et Mises à Jour¹

- Des fenêtres de maintenance seront planifiées et communiquées aux utilisateurs au moins 2 semaines à l'avance.
- Les mises à jour mineures sont prévues tous les 3 mois et les mises à jour majeures tous les 6 mois.

7.1.7 Restrictions Externes

- Délais de réponse des services tiers (par exemple, validation de paiement).
- Délais de traitement des demandes d'utilisateur pour le support client.

7.1.8 Revue Post-Projet

- Une période de revue post-projet est prévue 1 mois après le lancement pour évaluer les résultats et planifier les prochaines étapes.

¹ Voir Annexe Section 11.16 p 89 - 90

7.2 Contraintes Budgétaires

7.2.1 Estimation Initiale du Budget

- Coûts des ressources humaines (développeurs, chefs de projet, designers, testeurs)
- Coûts des licences de logiciels et outils de développement
- Coûts des services tiers (bases de données, serveurs, services d'API)
- Coût des tests (matériel, utilisateurs de tests rémunérés, etc.)

7.2.2 Allocation des Ressources

- Répartition du budget entre les différentes phases du projet (conception, développement, tests, déploiement)
- Priorisation des fonctionnalités et modules en fonction du budget

7.2.3 Gestion des Risques Financiers

- Méthodes pour identifier et mitiger les risques financiers, tels que les dépassements de coûts
- Plans d'urgence pour le financement en cas de besoin de plus de ressources

7.2.4 Suivi et Contrôle

- Mécanismes de suivi des dépenses en temps réel
- Approbations nécessaires pour les dépenses supplémentaires
- Fréquence et méthode de révision du budget

7.2.5 Marges de Manœuvre

- Pourcentage du budget réservé pour les imprévus et les urgences
- Stratégies pour réaffecter le budget en cas de modifications du périmètre du projet ou de retards

7.2.6 Revue Post-Projet

- Analyse des performances financières du projet après sa réalisation
- Comparaison du budget estimé et du budget réel, et explication des écarts

7.3 Contraintes Technologiques

7.3.1 Matériel

- Types de dispositifs cibles (smartphones, tablettes, desktop)

- Exigences minimales du système pour les utilisateurs finaux
- Infrastructure serveur nécessaire

7.3.2 Logiciel

- Compatibilité avec des systèmes d'exploitation spécifiques (iOS, Android, Windows)
- Navigateurs pris en charge pour les parties web de l'application
- Versions des librairies et frameworks utilisés

7.3.3 Dépendances Externes

- APIs, services tiers, ou bibliothèques dont l'application dépend
- Contraintes de licence pour ces dépendances
- Risques associés à la disponibilité et la fiabilité de ces services

7.3.4 Sécurité

- Cryptage des données, stockage sécurisé et autres mesures de sécurité
- Normes et protocoles de sécurité à respecter
- Matériel de sécurité nécessaire (pare-feu, VPN)

7.3.5 Réseau

- Exigences de bande passante et de latence pour un fonctionnement optimal
- Support pour les connexions à faible bande passante ou instables
- Exigences en matière de connectivité pour les services externes

7.3.6 Scalabilité

- Plans pour gérer une augmentation du nombre d'utilisateurs ou du volume de données
- Limites actuelles des technologies choisies
- Options pour la migration ou l'évolution technologique

7.3.7 Accessibilité

- Adaptations nécessaires pour les utilisateurs ayant des besoins spécifiques (malvoyants, malentendants, etc.)
- Normes d'accessibilité à respecter

7.3.8 Localisation et Internationalisation

- Contraintes liées à la prise en charge de plusieurs langues et régions
- Gestion des fuseaux horaires, des formats de date et de devises

7.3.9 Maintenance¹

- Facilité de mise à jour des technologies utilisées
- Complexité et coûts associés à la mise à jour ou à la modification de technologies existantes

7.3.10 Conformité et Standards

- Normes et certifications que les technologies doivent respecter
- Audits et vérifications techniques nécessaires (1)

7.4 Contraintes de Ressources Humaines

7.4.1 Compétences Techniques

- Types de compétences techniques requises (développement mobile, backend, etc.)
- Disponibilité de ces compétences sur le marché ou au sein de l'équipe
- Coût associé à l'embauche ou la formation de personnel qualifié

7.4.2 Disponibilité de l'Équipe

- Calendriers de chaque membre de l'équipe, notamment pour ceux qui sont partagés entre plusieurs projets
- Périodes de congés, de vacances ou d'indisponibilité prévues
- Temps nécessaire pour le recrutement et l'intégration de nouveaux membres

7.4.3 Répartition Géographique

- Emplacement des membres de l'équipe et impact sur la communication et la collaboration
- Besoin de voyages ou de déplacements pour les réunions, les formations, etc.
- Considérations sur le fuseau horaire pour la coordination

7.4.4 Organisation et Culture d'Entreprise

- Mode de travail (Agile, Scrum, Waterfall, etc.) et adaptabilité de l'équipe à ces méthodes
- Impact des valeurs et de la culture d'entreprise sur la dynamique de l'équipe et la réalisation du projet
- Processus de décision internes et niveaux d'autorité

7.4.5 Gestion de Projet

¹ Voir Annexe Section 11.16 p 89 - 90

- Outils de gestion de projet utilisés et leur adéquation avec les besoins du projet JojoApps
- Expérience et compétences du chef de projet
- Méthodes de suivi et de reporting

7.4.6 Charge de Travail

- Evaluation de la charge de travail totale par rapport à la taille et à la disponibilité de l'équipe
- Plan pour gérer les périodes de haute charge ou de stress pour l'équipe
- Possibilités de sous-traitance ou d'embauche temporaire

7.4.7 Risques Associés aux Ressources Humaines

- Turn-over, risque de départ de membres clés de l'équipe
- Conflits potentiels au sein de l'équipe ou avec d'autres parties prenantes
- Mesures pour la résolution des conflits et la gestion des risques

7.5 Contraintes Légales et de Conformité

7.5.1 Protection des Données

- Obligations légales relatives à la collecte, au stockage et au traitement des données, comme le RGPD en Europe
- Consentement de l'utilisateur et politiques de confidentialité

7.5.2 Propriété Intellectuelle

- Licences de logiciels tiers et de composants
- Droits d'auteur sur le contenu et le code source
- Brevets et marques déposées associées à l'application JojoApps

7.5.3 Commerce Électronique

- Conformité avec les lois sur le commerce électronique, y compris les taxes et les tarifs
- Obligations de fournir des informations claires sur les prix, les remboursements et les conditions d'utilisation

7.5.4 Accessibilité

- Conformité avec les lois et les normes sur l'accessibilité, telles que la directive européenne sur l'accessibilité des sites web et des applications mobiles
- Solutions pour rendre l'application accessible aux utilisateurs ayant des besoins spéciaux

7.5.5 Sécurité

- Conformité avec les lois et réglementations sur la cybersécurité
- Obligations de rapport en cas de violation de données ou de failles de sécurité

7.5.6 Transfert de Données Internationales

- Restrictions et exigences pour le transfert de données en dehors des frontières, en particulier pour les données personnelles

7.5.7 Audit et Conformité¹

- Besoin d'audits réguliers pour vérifier la conformité avec les lois et réglementations applicables
- Mécanismes de suivi et de documentation pour faciliter ces audits

7.5.8 Responsabilités des Fournisseurs

- Obligations légales associées aux fournisseurs tiers, y compris les sous-traitants et les fournisseurs de services cloud

7.5.9 Engagements Contractuels

- Limitations et obligations découlant des contrats avec les clients, les partenaires ou les fournisseurs

7.5.10 Considérations Locales ou Spécifiques au Secteur

- Lois ou réglementations spécifiques à certains marchés ou industries cibles, comme les licences de transport pour une application de covoiturage

7.6 Contraintes de Marché

7.6.1 Concurrence ²

- Identification des concurrents majeurs sur le marché cible
- Barrières à l'entrée dues à la présence de concurrents bien établis

¹ Voir Annexe Section 11.16 p 89 - 90

² Voir Annexe Section 11.16 p 89 - 90

7.6.2 Tendances du Marché ¹

- Adaptabilité aux changements rapides dans les préférences des consommateurs ou les technologies émergentes
- Veille concurrentielle et technologique pour rester compétitif

7.6.3 Exigences du Client ²

- Besoins spécifiques des clients cibles qui doivent être pris en compte dans le développement
- Feedback des utilisateurs et intégration dans les itérations de développement

7.6.4 Canaux de Distribution ³

- Limitations liées aux plateformes de distribution (Apple App Store, Google Play Store)
- Exigences en matière de marketing et de promotion pour atteindre une large audience

7.6.5 Réglementations de Prix ⁴

- Politique de prix, remises et promotions et leur impact sur la compétitivité
- Éventuelles réglementations gouvernementales affectant la tarification des services

7.6.6 Barrières Culturelles et Linguistiques

- Localisation et adaptation culturelle de l'application pour différents marchés
- Traductions et support multilingue

7.6.7 Partenariats et Intégrations ⁵

- Exigences ou limitations liées aux partenariats avec d'autres entreprises ou intégrations avec d'autres services
- Négociations contractuelles et accords de niveau de service

7.6.8 Cycle de Vie du Produit ⁶

- Durée du cycle de vie prévue pour l'application JojoApps et plans pour les mises à jour futures
- Planification de l'obsolescence, si applicable

¹ Voir Annexe Section 11.16 p 89 - 90

² Voir Annexe Section 11.16 p 89 - 90

³ Voir Annexe Section 11.16 p 89 - 90

⁴ Voir Annexe Section 11.16 p 89 - 90

⁵ Voir Annexe Section 11.16 p 89 - 90

⁶ Voir Annexe Section 11.16 p 89 - 90

7.6.9 Gestion des Risques de Marché ¹

- Évaluation des risques financiers, opérationnels et de réputation
- Plans de contingence pour faire face à des scénarios imprévus sur le marché

7.6.10 Évaluation et Surveillance des KPIs ²

- Indicateurs clés de performance à suivre pour mesurer le succès sur le marché
- Méthodes d'évaluation et outils d'analyse pour la surveillance continue

7.7 Contraintes Géographiques

7.7.1 Zones de Service

- Limitations relatives aux zones géographiques où le service peut être offert
- Coût de l'expansion vers de nouvelles zones géographiques

7.7.2 Infrastructure Locale

- Disponibilité et qualité des réseaux de télécommunication dans les zones de service
- Accessibilité aux serveurs locaux pour un fonctionnement optimal de l'application

7.7.3 Lois et Réglementations Locales

- Conformité aux lois locales concernant les services à la personne, les transactions financières, et les données utilisateur
- Restrictions éventuelles à l'import/export de données entre différentes juridictions

7.7.4 Langue et Culture

- Nécessité de localiser l'application en fonction des langues parlées dans la zone de service
- Adaptation culturelle des services offerts

7.7.5 Partenariats Locaux³

- Potentiel de collaboration avec des entreprises locales pour étendre le service
- Contraintes contractuelles avec des partenaires locaux

¹ Voir Annexe Section 11.16 p 89 - 90

² Voir Annexe Section 11.16 p 89 - 90

³ Voir Annexe Section 11.16 p 89 - 90

7.7.6 Saisonnalité et Conditions Climatiques¹

- Impact des conditions climatiques sur la disponibilité ou la demande pour certains types de missions
- Plans pour gérer les pics ou les creux de demande dus à la saisonnalité

7.7.7 Gestion des Catastrophes²

- Plans d'urgence en cas de catastrophes naturelles ou autres événements perturbateurs
- Protocoles pour la continuité des opérations

7.7.8 Taxes et Tarifications Locales

- Compréhension des structures fiscales locales et de leur impact sur la tarification
- Obligations en matière de TVA, de taxes à l'importation ou autres charges locales

7.7.9 Plans d'Expansion³

- Stratégies pour étendre le service à de nouvelles zones géographiques
- Évaluation des opportunités et des risques liés à l'expansion géographique

7.8 Contraintes de Qualité

7.8.1 Normes de Qualité

- Description des normes de qualité que l'application doit atteindre, y compris les certifications éventuelles (ISO, normes sectorielles, etc.)
- Méthodes pour mesurer la conformité aux normes

7.8.2 Assurance Qualité⁴

- Processus et méthodologies utilisées pour assurer la qualité du code et des services offerts
- Plans pour les audits de qualité internes et externes

7.8.3 Gestion des Défauts⁵

- Processus pour la détection, le suivi et la résolution des bugs et des défauts dans l'application
- Méthodes pour informer les utilisateurs des défauts connus et des mises à jour pour les résoudre

¹ Voir Annexe Section 11.16 p 89 - 90

² Voir Annexe Section 11.16 p 89 - 90

³ Voir Annexe Section 11.16 p 89 - 90

⁴ Voir Annexe Section 11.16 p 89 - 90

⁵ Voir Annexe Section 11.16 p 89 - 90

7.8.4 Performance et Temps de Réponse

- Objectifs de performance pour différents aspects de l'application (temps de chargement, réactivité, etc.)
- Plans pour le monitoring de la performance et les ajustements nécessaires

7.8.5 Disponibilité et Fiabilité

- Pourcentage de temps de disponibilité attendu pour l'application (par exemple, 99.9%)
- Plans pour la redondance et la récupération en cas de défaillance du système

7.8.6 Gestion des Risques Qualitatifs

- Identification des risques qui pourraient affecter la qualité du projet
- Stratégies pour la mitigation de ces risques

7.8.7 Expérience Utilisateur

- Critères pour évaluer la qualité de l'expérience utilisateur (UX), y compris la facilité d'utilisation, l'esthétique, etc.
- Méthodes pour recueillir les commentaires des utilisateurs sur l'expérience utilisateur

7.8.8 Tests de Qualité

- Types de tests à réaliser pour assurer la qualité (tests unitaires, tests d'intégration, tests d'acceptation utilisateur, etc.)
- Calendrier des cycles de test

7.8.9 Revue de Code

- Politiques et procédures pour la revue de code
- Outils utilisés pour automatiser certaines parties de la revue de code

7.8.10 Documentation de Qualité

- Exigences pour la documentation liée à la qualité, y compris les manuels de l'utilisateur, la documentation technique, et les journaux de changement

7.9 Contraintes de Communication

7.9.1 Canaux de Communication

- Description des canaux de communication officiels utilisés au sein de l'équipe (e-mail, Slack, Microsoft Teams, etc.)
- Limitations concernant l'usage de canaux de communication non-officiels

7.9.2 Fréquence de Communication

- Calendrier des réunions régulières, y compris les stand-ups quotidiens, les revues de sprint et les réunions de planification
- Méthodes pour assurer une communication continue entre les membres de l'équipe

7.9.3 Confidentialité et Sécurité

- Exigences concernant le chiffrement et la sécurisation des communications internes et externes
- Protocoles pour la communication de données sensibles ou confidentielles

7.9.4 Outils de Suivi et de Documentation

- Outils utilisés pour le suivi des communications, comme des systèmes de gestion de projet ou des tableaux Kanban
- Exigences pour la documentation des communications importantes

7.9.5 Communication avec les Parties Prenantes

- Méthodes et fréquences de communication avec les parties prenantes, y compris les clients, les fournisseurs et les investisseurs
- Limitations ou exigences légales concernant la communication avec les parties prenantes

7.9.6 Gestion des Attentes

- Stratégies pour gérer les attentes en matière de communication, tant au sein de l'équipe que vis-à-vis des parties prenantes
- Méthodes pour gérer les conflits ou les malentendus liés à la communication

7.9.7 Langue et Culture

- Considérations linguistiques si l'équipe est multilingue ou multiculturelle
- Stratégies pour surmonter les barrières linguistiques et culturelles dans la communication

7.9.8 Rapports et Mise à Jour du Projet

- Types et fréquences des rapports sur l'état du projet

- Méthodologie pour la mise à jour régulière des parties prenantes sur le progrès du projet

7.9.9 Communication en Cas d'Urgence

- Protocoles pour la communication en cas de crise ou d'urgence, y compris les canaux à utiliser et les personnes à contacter
- Plans pour la formation de l'équipe en matière de communication en cas d'urgence

8 Méthodologies de Développement

Dans cette section, nous détaillerons la méthodologie de développement et les pratiques qui seront employées pour la réalisation du projet JojoApps. Il s'agit d'un élément crucial pour assurer le bon déroulement du projet, depuis la phase de conception jusqu'à son déploiement et sa maintenance.

8.1 Choix de la Méthodologie

Pour le développement de JojoApps, la méthodologie choisie est le modèle en cascade (Waterfall) avec une incorporation de sprints pour la réalisation de tâches spécifiques. Ce choix est motivé par le fait que toutes les fonctionnalités de l'application ont été définies à l'avance, et aucun changement majeur n'est prévu pendant le cycle de développement de 9 mois.

Rationalisation du choix

- **Fixité des Exigences** : Toutes les exigences et fonctionnalités ont été clairement définies dès le début.
- **Durée du Projet** : Le projet a une durée déterminée de 9 mois, ce qui permet une planification rigoureuse.

Structure de la méthodologie

- **Phase de Conception** : Celle-ci comprend l'élaboration du cahier des charges, des wireframes et des maquettes.
- **Phase de Développement** : Chaque fonctionnalité ou module est développé indépendamment. Cette phase peut être divisée en sprints pour permettre une meilleure gestion des tâches.
- **Phase de Test** : Les tests sont effectués à la fin de chaque module développé.
- **Phase de Déploiement** : Une fois tous les modules développés et testés, l'application est déployée.
- **Phase de Maintenance** : Surveillance continue et mises à jour selon les retours des utilisateurs.

Utilisation des Sprints

Bien que le modèle en cascade soit principalement linéaire, l'introduction de sprints permet une certaine flexibilité dans la gestion des tâches. Les sprints peuvent être utilisés pour :

- Accélérer le développement de modules spécifiques.
- Réaffecter les ressources en cas de besoins non prévus mais limités.
- Faciliter la communication entre les équipes en focalisant les efforts sur des tâches spécifiques pendant une durée déterminée.

Cette approche hybride permet de combiner la rigueur et la prévisibilité du modèle en cascade avec la flexibilité des méthodes agiles, tout en respectant les contraintes du projet.

8.2 Planification et Gestion de Projet

Objectifs du Plan de Gestion de Projet

- Établir un calendrier réaliste et adapté aux ressources disponibles
- Identifier les responsabilités et rôles des membres de l'équipe
- Définir les indicateurs de performance clés (KPI) pour évaluer le progrès
- Gérer les risques et prévoir des plans de contingence

Outils et Techniques

- Utilisation d'un logiciel de gestion de projet pour le suivi des tâches, des délais et des ressources (comme Jira ou Trello)
- Mise en place de sprints réguliers pour le développement
- Utilisation de diagrammes de Gantt pour la planification à long terme

Phases du Projet

1. Phase de Démarrage : Finalisation du cahier des charges, constitution de l'équipe de projet
2. Phase de Planification : Élaboration du plan de projet, définition des sprints et des jalons
3. Phase de Développement : Développement du produit en suivant les sprints planifiés
4. Phase de Test : Tests unitaires, d'intégration et utilisateur
5. Phase de Déploiement : Mise en production et monitoring
6. Phase de Clôture : Finalisation du projet, documentation et passation aux équipes de maintenance

Rôles et Responsabilités

- Chef de Projet : Supervision globale, planification, et gestion des risques
- Développeurs : Développement des fonctionnalités selon les spécifications
- Testeurs : Élaboration et exécution des plans de test
- Designers : Conception des interfaces utilisateur
- DevOps : Gestion de l'infrastructure et du déploiement

Calendrier

- Sprint 1-2 : Mise en place de l'architecture de base et des wireframes (mois 1)
- Sprint 3-14 : Développement des fonctionnalités clés (mois 2-8)
- Sprint 14-16 : Tests et corrections de dernière minute (mois 9)

Indicateurs de Performance Clés (KPI)

- Vitesse du sprint : Nombre de points d'histoire complétés par sprint
- Taux d'achèvement : Pourcentage de tâches achevées par rapport au total prévu
- Temps de réponse des services : Mesuré en millisecondes, pour évaluer les performances du backend

Gestion des Risques

-
- Identifier les risques (comme des retards dans le calendrier, des dépassements de budget, etc.)
- Élaborer des plans de contingence pour chaque risque identifié

Révisions et Ajustements

- Revues de sprint régulières pour évaluer les progrès et ajuster la planification en conséquence
- Utilisation de métriques et de rétroactions pour apporter des améliorations continues

En utilisant cette approche structurée, l'équipe de projet JojoApps peut aborder efficacement la planification et la gestion du projet, tout en s'assurant que les livrables sont de haute qualité et livrés à temps.

8.3 Gestion des Risques

La gestion des risques est un élément crucial pour le succès du projet JojoApps. Elle implique l'identification, l'évaluation et la priorisation des risques pour les réduire ou les éliminer. Voici comment la gestion des risques sera abordée dans ce projet :

Identification des Risques

- **Retard dans le Développement** : Possibilité que les délais ne soient pas respectés en raison de problèmes techniques ou de ressources.
- **Dépassement du Budget** : Risque que les coûts du projet dépassent les prévisions.
- **Qualité Insuffisante du Code** : Risque de bugs ou de failles de sécurité.
- **Changements dans les Exigences du Client** : Risque que les exigences du client changent en cours de projet.
- **Indisponibilité des Ressources Clés** : Risque de perte d'un membre de l'équipe ou d'une technologie essentielle.
- **Problèmes de Conformité** : Non-respect des lois ou des réglementations en vigueur.

Évaluation et Priorisation

- Utilisation de la matrice de risques pour évaluer chaque risque en fonction de sa probabilité d'occurrence et de son impact sur le projet.

Plans de Contingence et d'Atténuation

- **Retard dans le Développement** : Allocation de temps supplémentaire dans le calendrier pour des imprévus. Éventuellement, prévoir des ressources supplémentaires.
- **Dépassement du Budget** : Surveillance stricte du budget et mise en place de mécanismes d'alerte en cas de dépassement.
- **Qualité Insuffisante du Code** : Revues de code et tests automatisés pour garantir la qualité du code.
- **Changements dans les Exigences du Client** : Communication régulière avec le client pour s'assurer que les exigences sont clairement comprises et fixées.
- **Indisponibilité des Ressources Clés** : Cross-training des membres de l'équipe pour s'assurer qu'au moins deux personnes sont compétentes sur chaque aspect clé du projet.

- **Problèmes de Conformité** : Consultation régulière avec des experts juridiques pour s'assurer que le projet respecte toutes les lois et réglementations en vigueur.

Surveillance et Revue

- Réunions régulières de l'équipe de gestion des risques pour évaluer l'efficacité des plans de contingence et d'atténuation.
- Suivi continu des indicateurs de risque.

Documentation

- Tous les risques identifiés et les mesures prises seront documentés et mis à jour dans un registre des risques.

8.4 Contrôle de Version

Le contrôle de version est une pratique essentielle dans tout projet de développement logiciel, et il est particulièrement crucial pour le projet JojoApps. Il permet de suivre les changements dans le code source, de collaborer plus efficacement et de revenir à des versions précédentes en cas de besoin. Voici comment le contrôle de version sera géré pour ce projet :

Outil de Contrôle de Version

- **Git** : L'outil de contrôle de version choisi pour ce projet est Git, en raison de sa flexibilité, de sa robustesse et de sa large adoption dans la communauté de développement.

Hébergement du Répertoire

- **GitHub / GitLab / Bitbucket** : Le code sera hébergé sur une de ces plateformes, qui offrent des fonctionnalités supplémentaires comme les pull requests, la gestion des problèmes et l'intégration continue.
- Branching Strategy
- **Modèle Gitflow** : Ce modèle de branching sera utilisé pour séparer les fonctionnalités, les correctifs et les versions. Il comprend des branches principales comme **master** et **develop**, ainsi que des branches de fonctionnalités, de correctifs et de versions.
- Règles de Commit
- Des messages de commit clairs et descriptifs seront exigés pour maintenir une histoire de code compréhensible.
- Utilisation de la librairie **Husky** pour assurer que les règles sont respectées avant chaque commit.

Revue de Code

- Aucun code ne sera fusionné dans les branches principales sans avoir été revu par au moins un

autre développeur.

Accès et Sécurité

- Des niveaux d'accès différenciés seront mis en place pour les membres de l'équipe, en fonction de leur rôle dans le projet.
- Activation de l'authentification à deux facteurs pour accéder au répertoire.

Sauvegardes et Restaurations

- Des sauvegardes régulières du dépôt seront effectuées pour prévenir toute perte de données.
- Un plan de restauration sera en place pour récupérer rapidement le code en cas de perte de données ou de corruption.

Documentation

- Toutes les pratiques et procédures relatives au contrôle de version seront documentées et mises à jour régulièrement.

En suivant ces pratiques de contrôle de version, le projet JojoApps sera mieux structuré et moins susceptible de rencontrer des problèmes liés à la gestion du code.

8.5 Gestion de la Qualité

La qualité de l'application JojoApps est une priorité, diverses mesures et méthodes seront mises en place pour assurer un niveau élevé de qualité tout au long du cycle de développement du projet.

8.5.1 Objectifs de Qualité

- **Performance** : L'application doit répondre en temps réel, avec des temps de chargement optimisés pour tous les écrans.
- **Fiabilité** : Assurer une disponibilité de 99,9% avec des mécanismes de récupération en cas de panne.
- **Sécurité** : Authentification forte, chiffrement des données, et mesures de prévention contre les vulnérabilités communes.
- **Extensibilité** : L'architecture doit permettre une extension facile des fonctionnalités.
- **Expérience utilisateur** : Fluidité de navigation, intuitivité, et répondant aux normes d'accessibilité.

8.5.2 Normes de Qualité Applicables

- Nous suivrons des normes de qualité telles que ISO 9001 pour le management de la qualité et ISO 27001 pour la sécurité de l'information.

8.5.3 Mesure de la Qualité

- **Tests Unitaires** : Chaque composant logiciel sera testé indépendamment pour assurer son bon fonctionnement.
- **Tests d'Intégration** : Les modules une fois réunis seront testés dans leur interaction.
- **Tests de Performance** : Des tests sous charge seront effectués pour vérifier le comportement de l'application sous différentes conditions.
- **Tests de Sécurité** : Des audits et tests de pénétration seront menés pour identifier et corriger les vulnérabilités.

8.5.4 Outils de Gestion de la Qualité

- **JIRA** : Pour le suivi des bugs et des tâches.
- **Sélénium/Appium** : Pour les tests automatisés.
- **SonarQube** : Pour l'analyse de la qualité du code.

8.5.5 Revues de Qualité

Des revues de qualité seront organisées à la fin de chaque sprint pour évaluer le respect des critères de qualité. Elles seront suivies d'un rapport détaillé.

8.5.6 Formation et Sensibilisation

Tous les membres de l'équipe seront formés et sensibilisés aux standards de qualité et aux meilleures pratiques à suivre.

8.5.7 Documentation de la Qualité

Toute la documentation relative à la qualité, y compris les rapports de test et les revues, sera conservée dans un référentiel centralisé et sécurisé.

8.5.8 Responsabilités

- **Chef de Projet** : Supervision globale et garant de la qualité du projet.
- **Développeurs** : Responsables de la qualité du code.

8.5.9 Gestion des Anomalies

Un processus formel sera mis en place pour la gestion des anomalies, de leur identification à leur résolution, en passant par leur classification et leur priorisation.

8.6 Intégration et Déploiement Continu (CI/CD)

8.6.1 Objectifs du CI/CD

- **Rapidité de Mise en Production** : Réduire le temps entre l'écriture du code et sa mise en production.
- **Qualité** : Automatiser les tests pour assurer que chaque modification est fiable.
- **Transparence** : Avoir un suivi clair et ouvert du statut de chaque modification apportée au code.

8.6.2 Outils de CI/CD

- Jenkins : Pour automatiser l'intégration et le déploiement.
- GitLab CI/CD : Alternative à Jenkins pour le pipeline CI/CD.
- Docker : Pour containeriser l'application et ses dépendances.
- Kubernetes : Pour orchestrer les conteneurs Docker.

8.6.3 Pipeline CI/CD

- **Source Control** : Tous les changements sont versionnés avec Git.
- **Build** : Compilation et construction du code.
- **Tests Automatisés** : Exécution de tests unitaires, d'intégration, et de performance.
- **Analyse de Code** : Utilisation de SonarQube pour vérifier la qualité du code.
- **Déploiement en Environnement de Staging** : Pour des tests additionnels en conditions réelles.
- **Validation Manuelle** : Revue du code et validation par les responsables du projet.
- **Déploiement en Production** : L'application est déployée dans l'environnement de production.

8.6.4 Gestion des Branches

- **Branches de fonctionnalités** : Pour chaque nouvelle fonctionnalité ou correctif.
- **Branches de release** : Pour préparer une nouvelle version de l'application.
- **Branches hotfix** : Pour des correctifs d'urgence.

8.6.5 Surveillance et Monitoring

Des outils comme New Relic seront utilisés pour surveiller la performance de l'application ainsi que le bon déroulement des pipelines CI/CD.

8.6.6 Notification et Alertes

Des notifications seront envoyées aux membres de l'équipe à chaque étape clé du pipeline, et en cas d'échec d'une des étapes.

8.6.7 Documentation du CI/CD

Une documentation détaillée du processus CI/CD sera maintenue, comprenant les étapes du pipeline,

les scripts utilisés, et les procédures de dépannage.

8.7 Documentation

La documentation est un aspect crucial du développement logiciel, car elle fournit des directives claires pour tous les membres de l'équipe du projet, ainsi que pour les futurs développeurs et administrateurs système qui interagiront avec le projet. Voici comment nous allons aborder la documentation dans le projet JojoApps.

8.7.1 Objectifs de la Documentation

- **Clarté** : Offrir des informations claires et compréhensibles pour tous les membres de l'équipe.
- **Actualité** : Maintenir la documentation à jour avec les dernières modifications.
- **Accessibilité** : Rendre la documentation facilement accessible à toute l'équipe et aux parties prenantes.

8.7.2 Types de Documentation

- **Documentation Technique** : Pour les développeurs, elle inclura des informations sur l'architecture, les bibliothèques, et les algorithmes.
- **Documentation Utilisateur** : Instructions pour les utilisateurs finaux sur la manière d'utiliser l'application.
- **Documentation API** : Spécifications et exemples d'utilisation de l'API de l'application.
- **Documentation de Code** : Commentaires dans le code expliquant le fonctionnement de sections spécifiques.

8.7.3 Outils pour la Documentation

- **Confluence/Notion** : Pour des documents plus larges et des collaborations d'équipe.
- **Javadoc/Swagger** : Pour la documentation d'API.

8.7.4 Gestion des Versions de la Documentation

La documentation sera versionnée en parallèle avec le code, en utilisant Git pour le suivi des modifications.

8.7.5 Normes et Conventions

- **Style d'écriture** : Utilisation d'un guide de style pour assurer la cohérence dans la documentation.
- **Métadonnées** : Chaque document comprendra des métadonnées comme l'auteur, la date de création et de modification.

8.7.6 Accessibilité et Sécurité

Des mesures seront prises pour assurer que la documentation soit accessible uniquement aux personnes autorisées.

8.7.7 Responsabilités

- **Rédacteur Technique** : Responsable principal de la documentation.
- **Développeurs** : Fourniront des commentaires de code et de la documentation technique.

8.8 Maintenance et Support¹

La maintenance et le support sont des aspects essentiels pour garantir la fiabilité et la durabilité de l'application JojoApps. Cette section détaille la stratégie, les procédures et les responsabilités associées à ces deux éléments.

8.8.1 Objectifs de la Maintenance et du Support

- **Fiabilité** : Assurer un fonctionnement sans faille de l'application.
- **Sécurité** : Appliquer les mises à jour de sécurité pour protéger les données et les fonctionnalités.
- **Évolutivité** : Adapter l'application aux besoins changeants des utilisateurs et aux nouvelles technologies.

8.8.2 Types de Maintenance

- **Maintenance Corrective** : Réparation des bugs et des défauts.
- **Maintenance Évolutive** : Ajout de nouvelles fonctionnalités ou améliorations.
- **Maintenance Préventive** : Mises à jour régulières pour éviter des problèmes futurs.

8.8.3 Procédures de Support

- **Service d'assistance** : Un service d'assistance sera disponible pour répondre aux questions et résoudre les problèmes.
- **Tickets de Support** : Système de gestion de tickets pour suivre les demandes et les résolutions.
- **FAQ et Base de Connaissances** : Ressources en ligne pour l'auto-assistance des utilisateurs.

8.8.4 Niveaux de Service (SLA)

- **Temps de Réponse** : Engagement sur le délai de réponse initial.
- **Temps de Résolution** : Engagement sur le délai de résolution des problèmes.

¹ Voir Annexe Section 11.16 p 89 - 90

8.8.5 Ressources Humaines et Techniques

- **Équipe de Maintenance** : Personnels dédiés pour la maintenance.
- **Outils de Suivi** : Logiciels et outils pour le suivi des problèmes et des mises à jour.

8.8.6 Planification

- **Plan de Maintenance** : Calendrier des maintenances prévues.
- **Plan d'Urgence** : Protocoles en cas de panne majeure ou de faille de sécurité.

8.8.7 Budget et Coût

- **Budget de Maintenance** : Estimation des coûts liés à la maintenance régulière.
- **Budget de Support** : Estimation des coûts pour la mise en place et l'opération du support.

8.8.8 Évaluation et Retours

- **KPIs** : Indicateurs clés de performance pour évaluer l'efficacité de la maintenance et du support.
- **Enquêtes de Satisfaction** : Collecte de retours des utilisateurs pour améliorer continuellement le service.

8.8.9 Mise à Jour du Plan

Révisions périodiques : Le plan de maintenance et de support sera revu et mis à jour régulièrement pour l'aligner avec les besoins changeants de l'entreprise et des utilisateurs.

8.9 Plan de Test

Le Plan de Test vise à définir une stratégie de test solide pour garantir la qualité et la fiabilité du projet JoJoApps. Cette section décrit les différents niveaux de tests, les méthodologies utilisées, et les ressources nécessaires pour mener à bien ces tests.

8.9.1 Stratégie de Test

Nous suivrons une stratégie de tests à plusieurs niveaux, combinant des tests unitaires, d'intégration, de système, et d'acceptation utilisateur.

8.9.2 Niveaux de Test

- Tests unitaires
- Tests d'intégration
- Tests de système
- Tests d'acceptation utilisateur

8.9.3 Environnements de Test

Nous disposerons d'environnements de test dédiés qui répliquent le plus fidèlement possible l'environnement de production.

8.9.4 Planification des Tests

Un calendrier de tests sera établi, coïncidant avec les sprints de développement.

8.9.5 Cas de Test

Des scénarios de test spécifiques seront écrits pour chaque fonctionnalité et composant de l'application.

8.9.6 Critères de Succession et d'Échec

Les critères de succès et d'échec pour chaque cas de test seront clairement définis.

8.9.7 Automatisation des Tests

Les tests répétitifs et à grande échelle seront automatisés pour augmenter l'efficacité.

8.9.8 Gestion des Incidents et des Bugs

Un système de suivi des bugs sera utilisé pour enregistrer, suivre et résoudre tous les problèmes détectés pendant les phases de test.

8.9.9 Rapports et Métriques de Test

Des rapports réguliers seront générés pour suivre les progrès, avec des métriques telles que le taux de réussite des tests, le nombre de bugs ouverts/fermés, etc.

8.9.10 Revue de Test et Clôture

À la fin de chaque cycle de test, une revue sera effectuée pour évaluer les résultats et prendre les mesures correctives nécessaires.

8.9.11 Responsabilités

L'équipe de QA sera principalement responsable de la mise en œuvre du plan de tests, en collaboration avec les développeurs et les parties prenantes.

9 Plan de Mise en Œuvre

9.1 Stratégie de Déploiement

9.1.1 Objectifs du Déploiement

- Assurer une transition en douceur de l'environnement de développement à l'environnement de production.
- Minimiser les risques liés à la mise en œuvre.
- Assurer la stabilité et la disponibilité de l'application dès le premier jour de lancement.

9.1.2 Phases de Déploiement

- **Phase de préparation** : Collecte et validation des exigences, installation des environnements et outils nécessaires.
- **Phase de développement** : Création des packages de déploiement et des scripts associés.
- **Phase de test** : Vérification dans un environnement de pré-production.
- **Phase de lancement** : Déploiement proprement dit dans l'environnement de production.
- **Phase de surveillance et d'ajustement** : Suivi des métriques, résolution des problèmes.

9.1.3 Environnements Cibles

- **Préproduction (Staging)** : Un environnement similaire à la production, mais utilisé pour les tests finaux.
- **Production** : L'environnement où l'application sera finalement déployée et accessible aux utilisateurs finaux.

9.1.4 Outils de Déploiement

- Utilisation de Jenkins pour l'automatisation du déploiement continu.
- Utilisation de Docker pour la conteneurisation des services.
- Utilisation de Kubernetes pour l'orchestration des conteneurs.

9.1.5 Calendrier de Déploiement

A définir

9.1.6 Équipe de Déploiement

- **DevOps** : Responsable de l'automatisation et de la mise en œuvre technique.
- **QA Engineers** : Validation que l'application est prête pour la production.
- **Administrateurs système** : Préparation et maintenance des environnements.
- **Chef de projet** : Coordination et communication entre toutes les parties prenantes.

9.1.7 Risques et Contingences

- **Retards dans le calendrier** : Avoir un plan B pour les tâches critiques.
- **Échec des tests en pré-production** : Procédure de rollback prête à être déclenchée.
- **Problèmes de performance post-lancement** : Surveillance accrue et plan d'ajustement rapide.

9.1.8 Revue et Approbation

Avant le déploiement, une revue finale sera effectuée avec toutes les parties prenantes pour s'assurer que tous les éléments sont en place et fonctionnent comme prévu.

9.2 Planification des Ressources

La planification des ressources est cruciale pour s'assurer que tous les aspects du projet sont correctement alimentés en ressources nécessaires — qu'il s'agisse de personnel, de matériel, de logiciels ou de temps. Cette section détaille comment ces ressources seront allouées tout au long du projet.

9.2.1 Objectifs de la Planification des Ressources

- Identifier les besoins en ressources à chaque phase du projet.
- Assurer une allocation efficace et efficiente des ressources disponibles.
- Prévenir les goulets d'étranglement et les retards dus à des ressources insuffisantes.

9.2.2 Types de Ressources

- **Ressources Humaines** : DevOps, Développeurs, Testeurs, Gestionnaires de Projet, UX/UI Designers.
- **Ressources Matérielles** : Serveurs, espaces de stockage, ordinateurs, appareils de test.
- **Ressources Logicielles** : Licences de logiciel, outils de développement, plateformes de test.
- **Ressources Temporelles** : Délais, jalons, sprints.

9.2.3 Allocation des Ressources

- **Phase de Conception** : Accent sur les ressources humaines (Architectes, Designers).
- **Phase de Développement** : Besoin accru de développeurs et d'environnements de test.
- **Phase de Test** : Ressources concentrées sur l'équipe de test et les environnements de test.
- **Phase de Déploiement** : DevOps et administrateurs système en première ligne.

9.2.4 Outils de Gestion des Ressources

- Utilisation de logiciels comme Jira pour le suivi des tâches et la gestion des ressources.

- Utilisation de Microsoft Project pour la planification globale.
- Tableaux Kanban pour la gestion agile des tâches et des ressources.

9.2.5 Calendrier de ressources

- "Le calendrier de ressources sera établi en consultation étroite avec les parties prenantes du projet, notamment les chefs d'équipe techniques, les gestionnaires de projet et le client. Ce calendrier sera aligné avec le calendrier général du projet."
- "Le calendrier de ressources sera un document évolutif, sujet à des révisions périodiques pour s'adapter à tout changement dans le périmètre du projet, les disponibilités des ressources, et autres contraintes."
- "Des jalons temporels préliminaires seront définis pour les principales étapes du projet. Ces jalons serviront de guide pour l'allocation des ressources et seront ajustés selon les besoins du projet."

9.2.6 Gestion des Risques liés aux Ressources

- **Turnover de l'équipe** : Plan de remplacement et formation rapide.
- **Indisponibilité des ressources matérielles** : Avoir un fournisseur de secours.
- **Dépassement du budget** : Réévaluation régulière et ajustements budgétaires.

9.2.7 Revue et Ajustements

Des revues périodiques seront menées pour s'assurer que les ressources sont utilisées de manière optimale. Les ajustements nécessaires seront faits en temps réel pour prévenir tout impact négatif sur le projet.

9.3 Plan de Communication

9.3.1 Objectifs de Communication

L'objectif est d'assurer une communication claire, efficace et opportune entre toutes les parties prenantes du projet, pour garantir une compréhension mutuelle des objectifs, des progrès et des problèmes éventuels.

9.3.2 Parties Prenantes

- Équipe de projet
- Client

9.3.3 Canaux de Communication

- E-mails
- Appels téléphoniques

- Réunions (physiques et virtuelles)
- Plateforme de gestion de projet
- Documentation partagée

9.3.4 Types de Communication

- Informations de statut et mises à jour du projet
- Alertes et notifications urgentes
- Comptes-rendus de réunions
- Rapports d'avancement

9.3.5 Fréquence de Communication

- Réunions hebdomadaires de l'équipe de projet
- Mises à jour mensuelles du client
- Communications d'urgence selon les besoins

9.3.6 Méthodes de Documentation

Toutes les communications importantes seront documentées et archivées dans un emplacement centralisé accessible à toutes les parties prenantes.

9.3.7 Protocole en cas de Crise

Un protocole spécifique sera établi pour la gestion des communications en cas de crise, incluant qui doit être informé, comment, et dans quel délai.

9.3.8 Responsabilités

- Le chef de projet sera responsable de la mise en œuvre du plan de communication.
- Les chefs d'équipe seront responsables de la communication au sein de leurs équipes respectives.
- Le client sera consulté pour valider les méthodes de communication le concernant.

9.4 Stratégie de Backup et de Restauration¹

9.4.1 Objectifs

L'objectif est de garantir la continuité de l'application en cas de panne, d'erreur humaine, ou de catastrophe naturelle, en mettant en place un plan robuste de sauvegarde et de restauration des données.

¹ Voir Annexe Section 11.16 p 89 - 90

9.4.2 Types de Données à Sauvegarder

- Base de données
- Fichiers de configuration
- Code source
- Documents et autres actifs numériques

9.4.3 Méthodologie de Sauvegarde

- Sauvegarde complète quotidienne
- Sauvegardes différentielles toutes les 6 heures
- Sauvegardes incrémentielles toutes les heures

9.4.4 Stockage des Sauvegardes

- Stockage local pour récupération rapide
- Stockage à distance pour scénarios de catastrophe

9.4.5 Procédures de Restauration

- Plan de restauration en une étape pour les scénarios les moins critiques
- Plan de restauration en plusieurs étapes pour les scénarios plus complexes

9.4.6 Test des Procédures de Restauration

- Tests mensuels de restauration à partir des sauvegardes
- Documentation des résultats et des temps de restauration

9.4.7 Rôles et Responsabilités

- Les administrateurs système sont responsables de la mise en œuvre des sauvegardes
- Le chef de projet est responsable de la vérification régulière de l'efficacité du plan

9.4.8 Durée de Conservation des Données

- Sauvegardes quotidiennes : conservées pendant 7 jours
- Sauvegardes mensuelles : conservées pendant 1 an

9.4.9 Conformité

Respect des réglementations locales et internationales sur la protection des données

9.4.10 Surveillance et Alerte

- Utilisation de logiciels pour surveiller l'état des sauvegardes
- Alerte en temps réel en cas d'échec de sauvegarde

9.4.11 Budget

- Coût des solutions de sauvegarde
- Coût des tests et de la maintenance

9.5 Plan de Suivi et de Maintenance ¹

Il est à noter que la maintenance à long terme ainsi que le support client de l'application JojoApps ne seront pas assurés par l'équipe de développement initiale. Ces aspects sont dévolus à une entité tierce, qui prendra en charge ces responsabilités une fois l'application livrée et mise en service.

Cette entité tierce devra être compétente dans la gestion de la maintenance et du support client et devra s'accorder avec les termes et les exigences définis dans le présent cahier des charges. Le choix et l'engagement de cette entité tierce devront être formalisés dans un accord distinct.

9.5.1 Objectifs

Le but est d'assurer le bon fonctionnement continu de l'application, en définissant des procédures de suivi et de maintenance claires et en assignant les responsabilités.

9.5.2 Types de Maintenance

- Maintenance préventive : Actions planifiées pour éviter des problèmes futurs
- Maintenance corrective : Actions pour résoudre les problèmes qui sont déjà apparus
- Maintenance évolutive : Mise à jour de l'application pour ajouter de nouvelles fonctionnalités ou améliorations

9.5.3 Calendrier de Maintenance

- Maintenance préventive mensuelle
- Mises à jour de sécurité trimestrielles
- Révisions majeures semestrielles

9.5.4 Indicateurs de Performance (KPIs)

- Temps de disponibilité

¹ Voir Annexe Section 11.16 p 89 - 90

- Temps de réponse
- Taux d'erreur

9.5.5 Suivi des Incidents

- Utilisation d'un outil de suivi des incidents pour enregistrer, suivre et résoudre tous les problèmes rencontrés
- Revue hebdomadaire des incidents enregistrés

9.5.6 Documentation

- Tous les changements et mises à jour doivent être documentés
- Les rapports de suivi de performance et d'incidents doivent être archivés

9.5.7 Budget et Coûts

- Budget alloué pour les activités de maintenance
- Estimation des coûts des outils de suivi et de gestion des incidents

9.5.8 Formation et Transfert de Compétences

- Formation initiale et continue de l'équipe sur les procédures de maintenance
- Mise en place d'un plan de transfert de compétences pour assurer la continuité

9.5.9 Audit et Conformité

- Audits internes et externes pour vérifier la conformité avec les standards et réglementations
- Plan d'action pour corriger les écarts identifiés lors des audits

9.6 Documentation et Manuels d'Utilisateur

9.6.1 Objectifs

L'objectif est de fournir une documentation complète et des manuels d'utilisateur qui permettent une utilisation optimale de l'application, ainsi qu'une maintenance efficace.

9.6.2 Types de Documentation

- Documentation technique : Destinée à l'équipe de développement et aux administrateurs système
- Manuels d'utilisateur : Destinés aux utilisateurs finaux
- Documentation d'API : Si l'application comporte des API externes ou internes

9.6.3 Format et Accessibilité

- Formats PDF et/ou HTML pour une consultation facile
- Accès via une plateforme en ligne sécurisée, ou intégrée directement dans l'application

9.6.4 Mise à Jour de la Documentation

- À chaque nouvelle version de l'application
- À chaque mise à jour significative de fonctionnalités

9.6.5 Rôles et Responsabilités

- Les développeurs sont responsables de la documentation technique
- Le chef de projet est responsable de s'assurer que toute la documentation est à jour et accessible
- Les experts en UX/UI peuvent contribuer aux manuels d'utilisateur pour optimiser l'expérience utilisateur

9.6.6 Versionnage

- Suivi des versions pour chaque type de documentation
- Historique des changements

9.6.7 Localisation

Traduction des manuels d'utilisateur dans les langues cibles, si nécessaire

9.6.8 Révisions et Approbations

- Processus de révision interne avant la publication de toute documentation
- Approbation par les parties prenantes concernées

9.6.9 Archivage

- Conservation des versions précédentes pour des raisons de conformité et d'audit
- Accès restreint aux archives

9.6.10 Formation et Sensibilisation

- Formation des utilisateurs et du personnel technique sur l'utilisation des documents et manuels
- Sessions de sensibilisation à l'importance de la documentation dans le cycle de vie de l'application.

10 Annexes

Note préliminaire :

Les documents et supports listés dans cette section "Annexes" sont envisagés comme des éléments clés pour le bon déroulement et la compréhension globale du projet. Néanmoins, il est important de noter que certains de ces documents ou supports ne sont pas encore disponibles au moment de la rédaction initiale de ce cahier des charges.

Ces annexes sont considérées comme des éléments évolutifs qui seront ajoutés ou modifiés au fur et à mesure de la progression du projet. La version initiale du cahier des charges ne comprendra donc pas nécessairement toutes les annexes listées, et de nouvelles versions du cahier des charges seront émises pour incorporer ces éléments manquants en temps voulu.

10.1 Glossaire de Termes

Définitions des termes techniques et acronymes utilisés dans le cahier des charges.

10.2 Références Documentaires

Liste des documents, articles, et autres matériaux consultés pendant la rédaction du cahier des charges.

10.3 Cahier des Clauses Techniques Particulières (CCTP)

Un cahier détaillé contenant les clauses techniques particulières si nécessaire.

10.4 Modèles de Wireframes et maquettes

Des exemples de wireframes et de maquettes pour mieux comprendre l'interface utilisateur.

10.5 Diagrammes

Diagrammes de flux, diagrammes de classes, etc., pour une meilleure compréhension du système.

10.6 Spécifications d'API

Documentation complète sur les API intégrées ou développées pour le projet.

10.7 Plan de test détaillé

Une version plus détaillée du plan de test, avec des cas de test spécifiques.

10.8 Spécifications de Matériel et Logiciel

Détails sur le matériel et les logiciels nécessaires pour le projet.

10.9 Contacts et Responsabilités

Liste des personnes clés impliquées dans le projet et leurs responsabilités.

10.10 Calendrier de Projet

Une copie du calendrier du projet, avec des échéances et des jalons.

10.11 Accords de Niveau de Services (SLA)

Les accords de niveau de service, si applicables.

10.12 Licences et Droits d'Auteur

Détails sur les licences de logiciel et les droits d'auteur associés au projet.

10.13 Audits et Conformités

Preuves et rapports d'audit, attestations de conformité, etc.

10.14 Code Source

Informations sur la manière d'accéder au code source, s'il est fourni.

10.15 Modifications et Mises à Jour

Un historique des modifications apportées au cahier des charges.

10.16 Clarification des Responsabilités Hors du Périmètre de 2C Conseil

10.16.1 Mise à jour du contenu

La responsabilité de 2CConseil se limite au déploiement initial de tout le contenu de l'application. Ceci inclut le contenu texte, les médias (images, vidéos) et tout autre élément constitutif de l'application. Toute mise à jour, modification, ou ajout de contenu après le lancement initial est hors du périmètre de 2CConseil.

10.16.2 Gestion de la sécurité

La sécurisation initiale de l'application est la seule responsabilité de 2CConseil. Cela n'inclut pas les mises à jour de sécurité, les patchs, la gestion des failles de sécurité découvertes post-lancement, ou toute autre activité liée à la cybersécurité après le déploiement initial.

10.16.3 Marketing et Distribution

Toutes les activités relatives à la publicité, au marketing digital, à la distribution sur diverses plateformes et aux campagnes de sensibilisation pour l'application sont hors du périmètre de 2CConseil.

10.16.4 Formation des utilisateurs

La formation nécessaire pour les utilisateurs finaux ne relève pas de la responsabilité de 2CConseil. Cette formation devra être fournie par un tiers ou sera soumise à un nouvel accord.

10.16.5 Conformité réglementaire et licences

La gestion de la conformité réglementaire et des licences n'est pas couverte par le périmètre de 2CConseil dans ce projet. Cette gestion sera assurée par un tiers ou fera l'objet d'un nouvel accord.

10.16.6 Surveillance et suivi

Toutes les activités liées à la surveillance en temps réel, à l'analyse des données utilisateur, à la performance de l'application, et à la résolution des problèmes post-lancement sont hors du périmètre de 2CConseil.

10.16.7 Financement et budgétisation

Tous les aspects financiers, autres que les coûts initiaux de développement, y compris mais sans s'y limiter, à la budgétisation pour des mises à jour, pour le marketing, et pour la maintenance, sont hors du périmètre de 2CConseil.

10.16.8 Sauvegardes et récupérations

Bien que 2CConseil puisse mettre en place une solution initiale de sauvegarde, la gestion continue et l'exécution des sauvegardes et des récupérations de données sont hors de son périmètre.

10.16.9 Gestion de la qualité

Toute activité liée aux audits de qualité, aux revues de code, à la gestion des bugs et aux tests d'assurance qualité après le déploiement initial sont hors du périmètre de 2CConseil.

10.16.10 Service Client et Support Technique

2CConseil n'est pas responsable du service client ou du support technique après le lancement initial de l'application. Cela inclut, mais n'est pas limité à, la gestion des tickets de support, la FAQ, la résolution des problèmes des utilisateurs, et toute autre forme de service après-vente ou de support client. Ces responsabilités devront être prises en charge par un tiers ou feront l'objet d'un nouvel accord.

10.16.11 Utilisation des Données par l'Administrateur Final

La responsabilité de 2CConseil ne s'étend pas à l'utilisation des données récupérées via le back-office de l'application par l'administrateur final. Cela inclut l'extraction, la manipulation, l'analyse, ou toute autre utilisation des données à des fins diverses. La conformité aux lois sur la protection des données, y compris le RGPD, et toute autre utilisation éthique des données, relèvent de la responsabilité de l'administrateur final ou devront faire l'objet d'un nouvel accord.

11 Validation du Cahier des charges

11.1 Procédure de Validation

11.1.1 Objectifs

Énoncer clairement les objectifs visés par cette procédure de validation. Par exemple, garantir que toutes les exigences sont bien comprises, que les parties prenantes sont toutes d'accord, et que le projet peut progresser dans des conditions optimales.

11.1.2 Parties Prenantes Impliquées

Liste exhaustive des parties prenantes qui doivent être impliquées dans la procédure de validation. Cela peut inclure le client, les membres de l'équipe de développement, les gestionnaires de projet, les utilisateurs finaux, et les entités régulatrices.

11.1.3 Étapes de la Validation

Description détaillée des différentes étapes de la validation :

Pré-validation :

- Revue interne du document pour s'assurer qu'il est prêt pour la validation.

Distribution aux Parties Prenantes :

- Le cahier des charges est envoyé à toutes les parties prenantes pour examen.

Période de Commentaires :

- Les parties prenantes ont un certain délai pour fournir des commentaires ou des questions.

Révision :

- L'équipe de projet se réunit pour discuter des commentaires et apporter les modifications nécessaires.

Validation Technique :

- Une revue technique est menée pour s'assurer que tous les aspects techniques sont viables et bien documentés.

Validation Finale :

- Une dernière réunion est tenue pour l'approbation finale du document. Si approuvé, le document est signé par toutes les parties prenantes.

Archivage :

- Le document validé est archivé selon les procédures internes de gestion de documents.

11.1.4 Calendrier de Validation

Un calendrier précis est présenté, indiquant les dates clés pour chaque étape de la procédure de validation.

11.1.5 Méthodes de Communication

Les canaux de communication qui seront utilisés pour distribuer le cahier des charges et recevoir les commentaires sont définis (email, plateforme de gestion de projet, réunions en personne, etc.).

11.1.6 Documentation de Validation

Types de documents qui seront produits lors de la validation (compte-rendu de réunions, tableaux de suivi des modifications, etc.) et comment ces documents seront stockés et accessibles.

11.1.7 Procédure de Révision Post-Validation

Étapes à suivre si le cahier des charges doit être modifié après sa validation initiale.

11.2 Critères de Validation

Cette section définit les critères spécifiques qui doivent être remplis pour que le cahier des charges soit considéré comme validé.

11.2.1 Objectifs

Établir des mesures concrètes pour évaluer si le document répond aux exigences fonctionnelles et non fonctionnelles, aux besoins des parties prenantes, et aux standards de qualité.

11.2.2 Critères de Contenu

- **Exhaustivité** : Le cahier doit couvrir toutes les fonctionnalités et exigences du projet.
- **Précision** : Les descriptions et les détails fournis doivent être clairs et sans ambiguïté.
- **Actualité** : Les informations doivent être à jour.

11.2.3 Critères Techniques

- **Faisabilité** : Les solutions techniques proposées doivent être réalisables avec les ressources disponibles.
- **Performance** : Les exigences doivent inclure des critères de performance mesurables.
- **Sécurité** : Les aspects de sécurité, y compris la conformité aux normes pertinentes, doivent être abordés.

11.2.4 Critères de Conformité Réglementaire

Le document doit être conforme aux lois et réglementations applicables, y compris les normes de l'industrie et les réglementations locales ou internationales.

11.2.5 Critères de Revue des Parties Prenantes

- **Engagement** : Toutes les parties prenantes pertinentes doivent avoir la possibilité de revoir et d'approuver le document.
- **Consensus** : Un accord général doit être atteint sur le contenu du document.

11.2.6 Critères Temporels

- **Respect des Échéances** : Le processus de validation doit être achevé selon le calendrier établi.
- **Mise à Jour** : Des révisions périodiques sont nécessaires pour garantir que le document reste actuel.

11.2.7 Critères Financiers

Le budget alloué pour le projet doit être en adéquation avec les exigences et les livrables énoncés dans le cahier des charges.

11.2.8 Conclusion

Un récapitulatif soulignant l'importance de ces critères dans la garantie que le cahier des charges est complet, exact, et prêt à être mis en œuvre.

11.3 Responsabilités

Cette section identifie les différents acteurs responsables du processus de validation du cahier des charges, ainsi que leurs rôles spécifiques.

11.3.1 Rôles et Responsabilités

- **Auteur du Cahier des Charges** : Responsable de la rédaction du document et de son actualisation.
- **Chef de Projet** : Responsable de garantir que le document est en ligne avec les objectifs du projet et le calendrier.
- **Equipe Technique** : Assure que les solutions techniques proposées sont viables et réalisables.
- **Juridique** : Vérifie la conformité avec les règlements et les lois applicables.
- **Equipe de Qualité** : Veille à ce que les standards de qualité sont respectés.

11.3.2 Responsabilités des Clients

Revue et Approbation : Les clients ont la responsabilité de revoir le cahier des charges et de fournir leur approbation ou leurs commentaires.

11.3.3 Processus de Sign-Off

Le cahier des charges sera considéré comme validé une fois que toutes les parties prenantes auront signé le document.

11.3.4 Suivi et Mise à Jour

Le chef de projet est responsable du suivi des activités de validation et de la mise à jour du cahier des charges selon les besoins.

11.3.5 Calendrier

Tous les acteurs doivent respecter le calendrier établi pour le processus de validation.

11.3.6 Conclusion

Cette section sert de guide pour assurer que toutes les parties prenantes comprennent leurs responsabilités dans le processus de validation du cahier des charges.

Chaque acteur identifié a un rôle crucial à jouer pour s'assurer que le cahier des charges est complet,

précis, et conforme à toutes les exigences et standards applicables.

11.4 Calendrier de Validation

Objectif

L'objectif de cette section est de fournir un calendrier détaillé des étapes nécessaires pour valider le cahier des charges, en précisant les délais pour chaque phase et les parties responsables.

Étapes de Validation

1. Révision Interne

- Date prévue : **17 octobre 2023**
- Responsabilité : Équipe de développement, Chef de Projet
- Description : Révision interne du cahier des charges pour s'assurer que tous les points sont couverts et conformes aux attentes initiales.

2. Pré-validation avec le Client

- Date prévue : **23 octobre 2023**
- Responsabilité : Chef de Projet, Client
- Description : Présentation du cahier des charges au client pour une pré-validation. Cette étape permet de recueillir les premiers retours du client et d'effectuer les ajustements nécessaires.

3. Finalisation

- Date prévue : **26 octobre 2023**
- Responsabilité : Équipe de développement, Chef de Projet
- Description : Intégration des retours du client et finalisation du document.

4. Validation Finale et Signature

- Date prévue : **6 novembre 2023**
- Responsabilité : Client (Jocelyn), Chef de Projet
- Description : Réunion pour la validation finale du cahier des charges. Si le document est conforme aux attentes, il sera signé par toutes les parties prenantes.

5. Archivage

- Date prévue : **8 novembre 2023**
- Responsabilité : Chef de Projet, Administratif
- Description : Une fois signé, le cahier des charges sera archivé conformément aux procédures internes et servira de référence tout au long du projet.

6. Suivi de Validation

Un suivi sera effectué à chaque étape de ce calendrier pour s'assurer que les dates prévues sont respectées et que toutes les parties sont informées de l'état d'avancement de la validation.

- **Outils de Suivi** : Tableau de bord du projet, réunions hebdomadaires, emails de mise à jour.
- **Points de Contrôle** : À la fin de chaque étape clé, un point de contrôle sera réalisé pour s'assurer que tout se déroule comme prévu.

Ces dates sont sujettes à modification et toute modification sera communiquée à toutes les parties prenantes dans les plus brefs délais.

11.5 Processus de Révision

Objectif

Le but de cette section est de définir le processus par lequel le cahier des charges sera révisé si nécessaire, en identifiant les conditions sous lesquelles une révision peut être demandée, les parties responsables de la révision et les délais pour effectuer ces changements.

Conditions pour Révision

Une révision du cahier des charges peut être nécessaire en cas de :

- Changements dans les exigences du client.
- Modifications légales ou réglementaires.
- Nouvelles contraintes techniques identifiées.
- Feedback des parties prenantes durant les phases de test ou de déploiement.
- Processus de Demande de Révision

Soumission de Demande :

- La partie demandant la révision doit soumettre une demande formelle, idéalement par écrit, au Chef de Projet.

2. Évaluation de la Demande :

- Le Chef de Projet évalue la demande en consultation avec l'équipe de développement et le client si nécessaire.

3. Décision et Planification :

- Si la demande est approuvée, un plan de révision est établi, y compris les modifications à apporter, les ressources nécessaires et le calendrier de mise en œuvre.

4. Exécution de la Révision :

- Les changements sont apportés conformément au plan de révision. Toutes les modifications sont documentées et une nouvelle version du cahier des charges est produite.

5. Validation :

- La nouvelle version est soumise à un processus de validation similaire à celui décrit dans la section 11.4.

Responsabilités

- **Chef de Projet** : Évaluation de la demande de révision, planification et coordination de la révision.
- **Équipe de Développement** : Mise en œuvre des changements.
- **Client** : Approbation des changements, le cas échéant.

Délais

Le délai pour effectuer une révision dépendra de la nature des changements mais sera communiqué à toutes les parties prenantes dès qu'une décision aura été prise concernant la demande de révision.

11.6 Documentation de Validation

Objectif

Le but de cette section est de décrire tous les documents qui seront utilisés pour valider le cahier des charges. La documentation de validation sert de preuve officielle que tous les critères et processus ont été respectés et accomplis comme prévu.

Types de Documents de Validation

- **Procès-Verbal de Réunion** : Documente les discussions et les décisions prises lors des réunions de validation.
- **Liste de Vérification** : Un outil utilisé pour assurer que tous les éléments du cahier des charges ont été revus et approuvés.
- **Rapports d'Audit** : Fournissent une analyse détaillée de la conformité du cahier des charges aux normes et exigences prédéfinies.
- **Certificats de Conformité** : Attestent que le cahier des charges a été examiné et respecte toutes les normes et réglementations applicables.
- **Emails et Communications Formelles** : Correspondance entre les parties prenantes confirmant l'approbation des différentes sections ou du cahier des charges en entier.

Processus de Documentation

1. **Création** :
 - Les documents de validation sont créés en parallèle avec le processus de validation, et sont mis à jour à chaque étape clé.
2. **Révision** :
 - Avant la validation finale, tous les documents sont revus pour exactitude et complétude.
3. **Approbation** :
 - Les documents sont approuvés par les parties responsables, souvent le Chef de Projet et le client.
4. **Archivage** :
 - Une fois approuvés, les documents de validation sont archivés de manière sécurisée pour référence future et pour des besoins éventuels d'audit ou de conformité.

Responsabilités

- **Chef de Projet** : Supervise la création, la révision et l'approbation de la documentation.

- **Équipe de Développement** : Contribue à la documentation technique.
- **Client** : Examine et approuve les documents de validation pertinents.

Délais

Les délais pour la création et l'approbation de la documentation de validation seront alignés avec le calendrier de validation défini dans la section 11.4.

11.7 Signature des Parties Prenantes

Objectif

L'objectif de cette section est de formaliser l'accord sur le cahier des charges entre toutes les parties prenantes. Les signatures servent de validation officielle que tous les critères et exigences ont été compris, acceptés et approuvés.

Parties Prenantes Concernées

- **Client** : Jocelyn (ou son représentant)
- **Chef de Projet** : Nom du responsable
- **Équipe de Développement** : Noms des développeurs ou titre du groupe
- **Consultants Externes** : Si applicable
- **Fournisseurs** : Si applicable
- **Autres** : Autres parties prenantes impliquées dans le projet

Processus de Signature

- **Préparation du Document Final** : Une fois toutes les révisions terminées et les validations faites, le document final est préparé pour la signature.
- **Examen Final** : Les parties prenantes reçoivent le document final pour un dernier examen.
- **Réunion de Signature** : Une réunion est organisée (physique ou virtuelle) pour que toutes les parties signent le cahier des charges.
- **Archivage** : Les copies signées sont distribuées à toutes les parties prenantes et une copie est archivée pour des besoins futurs.

Responsabilités

- **Chef de Projet** : Responsable de coordonner le processus de signature et de s'assurer que toutes les parties prenantes ont signé.
- **Secrétaire de Projet** : Peut être chargé de la distribution et de l'archivage des documents signés.
- **Client** : Doit signer pour indiquer l'approbation du cahier des charges.

Délais

La date de la réunion de signature sera alignée avec le calendrier de validation défini dans la section 11.4. En l'occurrence, la validation finale et la signature sont prévues pour le lundi 6 novembre.

Documentation Associée

Toutes les signatures seront documentées et une copie de la page de signature sera annexée au cahier

des charges final ainsi qu'à tout autre document de validation pertinente.

11.8 Date de Validation

Objectif

L'objectif de cette section est de spécifier la date à laquelle le cahier des charges est considéré comme validé par toutes les parties prenantes. Cette date marque le point de départ officiel pour la mise en œuvre du projet.

Date de Validation Finale

La date de validation finale est fixée au Lundi 6 novembre. Cette date a été choisie en alignement avec le calendrier de validation décrit dans la section 11.4.

- Conséquences de la Validation
- **Début Officiel du Projet** : Une fois le cahier des charges signé, le projet entre dans sa phase de mise en œuvre.
- **Allocation des Ressources** : Toutes les ressources planifiées dans le cahier des charges sont officiellement allouées au projet à partir de cette date.
- **Engagement Contractuel** : La date de validation signe l'engagement formel entre les parties et peut avoir des implications contractuelles, telles que les pénalités pour retard ou non-conformité.

Responsabilités

- **Chef de Projet** : Veiller à ce que toutes les signatures soient recueillies avant la date de validation et que toutes les parties prenantes soient informées.
- **Client** : Reconnaît que la date de validation est la date à partir de laquelle il s'attend à ce que le projet progresse selon le plan.

Documentation Associée

La date de validation sera documentée et incluse dans les archives du projet, ainsi que dans tout autre document de suivi ou de reporting.

11.9 Modifications Post-Validation

Objectif

L'objectif de cette section est de définir la procédure à suivre pour toute modification apportée au cahier des charges après sa validation officielle.

Procédure de Modification

- **Demande de Modification** : Toute modification doit être formulée par écrit et soumise au chef de projet pour évaluation.
- **Évaluation de l'Impact** : Le chef de projet, en collaboration avec les parties prenantes concernées, évalue l'impact de la modification sur le calendrier, le budget, les ressources et les livrables du projet.
- **Approbation ou Rejet** : Après évaluation, la modification est soit approuvée soit rejetée. En cas d'approbation, elle doit être documentée et toutes les parties prenantes en sont informées.
- **Mise à jour du Cahier des Charges** : Si la modification est approuvée, le cahier des charges est mis à jour en conséquence et une nouvelle version est distribuée.
- **Signature de la Nouvelle Version** : Une nouvelle signature des parties prenantes peut être requise pour valider la modification.

Responsabilités

- **Chef de Projet** : Responsable de la coordination de l'évaluation, de l'approbation et de la mise en œuvre de toutes les modifications.
- **Client** : Doit soumettre toute demande de modification par écrit et participer à l'évaluation de l'impact de la modification.
- **Équipe de Projet** : Assiste le chef de projet dans l'évaluation de l'impact de la modification sur les différentes facettes du projet.

Documentation Associée

Toutes les modifications approuvées et leurs impacts seront documentés et annexés au cahier des charges original. Un registre des modifications sera également maintenu pour suivre toutes les modifications effectuées.

Limitations

Il est important de noter que les modifications post-validation sont exceptionnelles et peuvent entraîner des délais et des coûts supplémentaires. Les parties prenantes doivent être conscientes des implications de toute modification apportée après la validation.

11.10 Accord et Validation du Cahier des Charges

Ce cahier des charges a été lu, compris et approuvé par les parties prenantes suivantes :

- **Pour le Client**

Nom & Prénom :

Date :

Signature :

- **Pour le Prestataire**

Nom & Prénom :

Date :

Signature :

- **Autres Parties Prenantes (Chef de projet)**

Nom & Prénom :

Date :

Signature