**Cahier des charges pour le développement d'une application de gestion pour un garage automobile**

**1. Introduction**

Le présent cahier des charges a pour objectif de définir les besoins et les fonctionnalités requises pour le développement d'une application de gestion destinée à un garage automobile. L'application doit permettre une gestion efficace des clients, des réparations, du stock des pièces de rechange, et de la facturation, tout en incluant la gestion détaillée des véhicules de chaque client. Trois types de profils seront pris en compte : Client, Mécanicien, et Administrateur. De plus, une nouvelle fonctionnalité sera ajoutée pour permettre aux clients de demander un rendez-vous et de marquer l'état extérieur de leur voiture avant l'admission.

**2. Objectifs de l'application**

L'application doit répondre aux besoins spécifiques du garage automobile, en fournissant les fonctionnalités suivantes :

2.1. Gestion des Clients

* Enregistrement des informations clients (nom, prénom, adresse, numéro de téléphone, adresse e-mail).
* Consultation facile de l'historique des services et des réparations effectuées pour chaque client.
* Ajout, modification et suppression des informations liées aux véhicules.

2.2. Gestion des Véhicules par Client

* Possibilité pour le client d'ajouter plusieurs véhicules avec les informations suivantes :
  + Marque du véhicule.
  + Modèle du véhicule.
  + Type de carburant.
  + Numéro d'immatriculation.
  + Photos du véhicule.

2.3. Gestion des Réparations

* Création de fiches de réparation pour chaque véhicule.
* Suivi du statut des réparations (en attente, en cours, terminée).
* Attribution de mécaniciens aux réparations.

2.4. Gestion du Stock de Pièces de Rechange

* Enregistrement des pièces de rechange disponibles avec leurs caractéristiques (nom, référence, fournisseur, prix).
* Suivi des niveaux de stock en temps réel.
* Notification de réapprovisionnement lorsque le stock atteint un seuil critique.

2.5. Facturation

* Création de factures automatiques à la fin de chaque réparation.
* Possibilité d'ajouter des pièces de rechange et des services supplémentaires à la facture.
* Historique des factures pour chaque client.

2.6. Nouvelle Fonctionnalité : Demande de Rendez-vous

* Les clients peuvent demander un rendez-vous pour le diagnostic et la réparation de leur véhicule.
* Possibilité de sélectionner la date et l'heure souhaitées.
* Ajout d'une fonction pour marquer l'état extérieur de la voiture avant l'admission (par exemple, via des photos ou des annotations).

**3. Gestion des Profils**

3.1. Profil Client

* Accès limité aux informations liées à ses propres véhicules et réparations.
* Consultation des factures et de l'historique des réparations.
* Possibilité de demander un rendez-vous et de marquer l'état extérieur de la voiture avant l'admission.

3.2. Profil Mécanicien

* Attribution des réparations en cours.
* Mise à jour du statut des réparations.
* Accès aux informations techniques des réparations.

3.3. Profil Administrateur

* Accès complet à toutes les fonctionnalités de l'application.
* Gestion des profils utilisateurs.
* Suivi global des activités du garage.

**4. Contraintes Techniques**

4.1. Plateforme de développement

* Utilisation du framework de développement Laravel pour la réalisation de l'application.

4.2. Base de données

* Utilisation d'une base de données MySQL pour le stockage des informations.

4.3. Langues prises en charge

* Mise en place d'une application multilingue prenant en charge les langues suivantes : arabe, français, anglais.

4.4. Interface Utilisateur

* Développement d'une interface responsive pour une utilisation optimale sur différentes plateformes (ordinateurs, tablettes, smartphones).

**5. Intégration de Fonctionnalités Avancées**

5.1. Rapports

* Génération de rapports sur l'activité du garage, les réparations en cours, les pièces de rechange les plus utilisées, etc.

5.2. Notification

* Système de notification pour informer les clients de la fin des réparations ou pour les rappeler des rendez-vous.

**6. Livrables attendus**

6.1. Application fonctionnelle répondant aux spécifications du cahier des charges.

6.2. Documentation complète sur l'utilisation de l'application et la maintenance.

**7. Calendrier et Suivi du Projet**

7.1. Élaboration d'un plan de projet détaillé avec des étapes clés et des dates de livraison prévues.

**8. Budget**

8.1. Estimation des coûts de développement, incluant les frais de conception, de programmation, de tests, et de maintenance.

Ce cahier des charges est soumis pour approbation avant le démarrage du développement de l'application. Des révisions pourront être apportées en cours de projet suite aux besoins émergents.

**Épique 1: Gestion des Clients**

**User Stories :** 1.1 En tant qu'Administrateur, je veux pouvoir ajouter un nouveau client avec toutes ses informations nécessaires (nom, prénom, adresse, numéro de téléphone, adresse e-mail).

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la saisie des informations du client.

1.2 En tant qu'Administrateur, je veux pouvoir consulter la liste complète des clients enregistrés dans le système.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 1.1 terminée.

1.3 En tant que Client, je veux pouvoir mettre à jour mes informations personnelles dans le système.

* **Valeur :** ★★★☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 1.1 terminée.

1.4 En tant qu'Administrateur, je veux pouvoir supprimer un client du système.

* **Valeur :** ★★★☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 1.2 terminée.

**Épique 2: Gestion des Véhicules par Client**

**User Stories :** 2.1 En tant que Client, je veux pouvoir ajouter un nouveau véhicule avec toutes ses informations (marque, modèle, carburant, numéro d'immatriculation, photos).

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la saisie des informations du véhicule.

2.2 En tant que Client, je veux pouvoir consulter la liste de tous mes véhicules enregistrés dans le système.

* **Valeur :** ★★★☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 1.1 terminée.

2.3 En tant qu'Administrateur, je veux pouvoir visualiser et modifier les informations relatives aux véhicules de chaque client.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 1.2 et User Story 2.2 terminées.

2.4 En tant qu'Administrateur, je veux pouvoir supprimer un véhicule d'un client du système.

* **Valeur :** ★★☆☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 2.3 terminée.

**Épique 3: Gestion des Réparations**

**User Stories :** 3.1 En tant que Mécanicien, je veux pouvoir créer une fiche de réparation pour un véhicule.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la création d'une fiche de réparation.

3.2 En tant que Mécanicien, je veux pouvoir attribuer des réparations à d'autres mécaniciens.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 3.1 terminée.

3.3 En tant que Mécanicien, je veux pouvoir mettre à jour le statut des réparations (en attente, en cours, terminée).

* **Valeur :** ★★★★☆
* **Taille :** S
* **Prérequis :**
  + User Story 3.1 terminée.

3.4 En tant que Client, je veux pouvoir consulter l'état en temps réel des réparations en cours sur mes véhicules.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 3.1 et User Story 3.3 terminées.

3.5 En tant qu'Administrateur, je veux pouvoir accéder à une vue globale de toutes les réparations en cours.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 3.3 terminée.

**Épique 4: Gestion du Stock de Pièces de Rechange**

**User Stories :** 4.1 En tant qu'Administrateur, je veux pouvoir ajouter une nouvelle pièce de rechange avec toutes ses informations nécessaires (nom, référence, fournisseur, prix).

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la saisie des informations de la pièce.

4.2 En tant que Mécanicien, je veux pouvoir consulter le stock de pièces de rechange disponible.

* **Valeur :** ★★★☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 4.1 terminée.

4.3 En tant que Mécanicien, je veux être notifié lorsque le stock d'une pièce de rechange atteint un seuil critique.

* **Valeur :** ★★★★☆
* **Taille :** S
* **Prérequis :**
  + User Story 4.2 terminée.

4.4 En tant qu'Administrateur, je veux pouvoir mettre à jour les informations relatives aux pièces de rechange.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 4.1 terminée.

4.5 En tant que Mécanicien, je veux pouvoir sélectionner les pièces de rechange nécessaires lors de la création d'une fiche de réparation.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 4.1 et User Story 4.4 terminées.

**Épique 5: Facturation**

**User Stories :** 5.1 En tant que Mécanicien, je veux pouvoir ajouter des pièces de rechange et des services supplémentaires à une facture.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 4.5 terminée.

5.2 En tant que Client, je veux recevoir automatiquement une facture détaillée à la fin de chaque réparation.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 3.3 et User Story 5.1 terminées.

5.3 En tant que Client, je veux pouvoir consulter l'historique de toutes mes factures dans le système.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 5.2 terminée.

5.4 En tant qu'Administrateur, je veux pouvoir accéder à une vue globale de toutes les factures générées par le garage.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 5.2 terminée.

**Épique 6: Demande de Rendez-vous et État Extérieur de la Voiture**

**User Stories :** 6.1 En tant que Client, je veux pouvoir demander un rendez-vous en spécifiant la date et l'heure souhaitées.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la demande de rendez-vous.

6.2 En tant que Client, je veux pouvoir marquer l'état extérieur de ma voiture avant l'admission en utilisant des photos ou des annotations.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface permettant de télécharger des photos et d'ajouter des annotations.

6.3 En tant que Mécanicien, je veux pouvoir accéder à l'état extérieur enregistré avant l'admission pour chaque véhicule.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 6.2 terminée.

6.4 En tant que Mécanicien, je veux pouvoir consulter les demandes de rendez-vous en attente et les attribuer à d'autres mécaniciens.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 6.1 terminée.

**Épique 7: Fonctionnalités d'Administrateur**

**User Stories :** 7.1 En tant qu'Administrateur, je veux pouvoir ajouter un nouveau client avec toutes ses informations nécessaires (nom, prénom, adresse, numéro de téléphone, adresse e-mail).

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la saisie des informations du client.

7.2 En tant qu'Administrateur, je veux pouvoir consulter la liste complète des clients enregistrés dans le système.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 7.1 terminée.

7.3 En tant qu'Administrateur, je veux pouvoir visualiser et modifier les informations relatives aux véhicules de chaque client.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 7.2 terminée.

7.4 En tant qu'Administrateur, je veux pouvoir supprimer un client du système.

* **Valeur :** ★★★☆☆
* **Taille :** S
* **Prérequis :**
  + User Story 7.3 terminée.

7.5 En tant qu'Administrateur, je veux pouvoir accéder à une vue globale de toutes les réparations en cours.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 3.3 terminée.

7.6 En tant qu'Administrateur, je veux pouvoir ajouter une nouvelle pièce de rechange avec toutes ses informations nécessaires (nom, référence, fournisseur, prix).

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + Interface de formulaire pour la saisie des informations de la pièce.

7.7 En tant qu'Administrateur, je veux pouvoir mettre à jour les informations relatives aux pièces de rechange.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 7.6 terminée.

7.8 En tant qu'Administrateur, je veux pouvoir ajouter des pièces de rechange et des services supplémentaires à une facture.

* **Valeur :** ★★★☆☆
* **Taille :** M
* **Prérequis :**
  + User Story 7.7 terminée.

7.9 En tant qu'Administrateur, je veux pouvoir accéder à une vue globale de toutes les factures générées par le garage.

* **Valeur :** ★★★★☆
* **Taille :** M
* **Prérequis :**
  + User Story 7.8 terminée.

7.10 En tant qu'Administrateur, je veux pouvoir accéder à l'état extérieur enregistré avant l'admission pour chaque véhicule. - **Valeur :** ★★★☆☆ - **Taille :** M - **Prérequis :** - User Story 6.3 terminée.

**MLD**Vous pouvez vous basez sur ce MLD préliminaire

**Utilisateur (User)**

* id (clé primaire)
* username
* password
* email
* role

**Client**

* id (clé primaire)
* firstName
* lastName
* address
* phoneNumber
* userID (clé étrangère vers User)

**Véhicule (Vehicle)**

* id (clé primaire)
* make
* model
* fuelType
* registration
* photos
* clientID (clé étrangère vers Client)

**Réparation (Repair)**

* id (clé primaire)
* description
* status
* startDate
* endDate
* mechanicNotes
* clientNotes
* mechanicID (clé étrangère vers User)
* vehicleID (clé étrangère vers Vehicle)

**Pièce de rechange (SparePart)**

* id (clé primaire)
* partName
* partReference
* supplier
* price

**Facture (Invoice)**

* id (clé primaire)
* repairID (clé étrangère vers Repair)
* additionalCharges
* totalAmount

**Sprint 0 (Semaine 0 - Initialisation):**

1. **Réunion de Lancement:**
   * Rassemblement de l'équipe pour discuter des objectifs, des rôles et des attentes.
2. **Planification du Projet:**
   * Définir les grandes lignes du projet, les échéances et les jalons.
3. **Préparation de la Charte Graphique:**
   * Travail avec les concepteurs pour élaborer la charte graphique de l'application.
4. **Conception de la Base de Données:**
   * Modélisation détaillée de la base de données en fonction des exigences.
5. **Configuration de l'Environnement:**
   * Installation et configuration des outils de développement, du framework Laravel, de la base de données, etc.
6. **Planification des Sprints Suivants:**
   * Identifier les épicentres, décomposer en user stories, et planifier les sprints suivants.
7. **Documentation:**
   * Création d'une documentation initiale pour la référence ultérieure.
8. **Tests Initiaux:**
   * Mise en place de tests initiaux pour s'assurer que l'environnement de développement fonctionne correctement.

**Sprint 1 (Semaine 1): Gestion des Clients et des Véhicules**

**Objectif Principal:** Mettre en place les fonctionnalités de base pour la gestion des clients et de leurs véhicules.

1. **User Stories:**
   * 1.1 En tant qu'Administrateur, je veux pouvoir ajouter un nouveau client avec toutes ses informations nécessaires.
   * 1.2 En tant qu'Administrateur, je veux pouvoir consulter la liste complète des clients enregistrés dans le système.
   * 2.1 En tant que Client, je veux pouvoir ajouter un nouveau véhicule avec toutes ses informations.
2. **Tâches Techniques:**
   * Configuration de la base de données pour les entités "User" et "Client".
   * Mise en place de la création d'un nouveau client via une interface de formulaire.
   * Mise en place de la consultation de la liste des clients pour l'administrateur.
   * Mise en place de la création d'un nouveau véhicule via une interface de formulaire.
3. **Tests:**
   * Tester la création d'un nouveau client.
   * Tester la consultation de la liste des clients.
   * Tester la création d'un nouveau véhicule.
4. **Conception:**
   * S'assurer que les pages d'interface utilisateur sont conçues de manière conviviale.
   * Concevoir les formulaires nécessaires pour la création de clients et de véhicules.
5. **Documentation:**
   * Documenter le modèle de données mis à jour pour les entités "User" et "Client".
6. **Révision de Sprint:**
   * Tenir une réunion de révision du sprint pour évaluer les accomplissements et planifier les étapes suivantes.
7. **Formation (le cas échéant):**
   * Si nécessaire, fournir une formation à l'équipe sur les nouvelles fonctionnalités mises en place.
8. **Rétroaction et Amélioration Continue:**
   * Collecter des retours d'utilisateurs ou d'autres membres de l'équipe.
   * Identifier des améliorations potentielles pour les sprints suivants.
9. **Tests Unitaires pour la Gestion des Clients (Exemples) :**
   * **Test de Création de Client :**
     1. Vérifier que l'ajout d'un nouveau client crée une entrée valide dans la base de données.
     2. S'assurer que les champs obligatoires sont correctement validés.
     3. Tester la gestion des cas où la création de client échoue.
   * **Test de Consultation de la Liste des Clients :**
     1. Vérifier que la liste des clients récupérée correspond aux données attendues.
     2. Tester les cas où aucun client n'est enregistré dans le système.
     3. S'assurer que seuls les clients autorisés ont accès à la liste complète.
10. **Tests Unitaires pour la Gestion des Véhicules (Exemples) :**
    * **Test de Création de Véhicule :**
      1. Vérifier que l'ajout d'un nouveau véhicule crée une entrée valide dans la base de données.
      2. S'assurer que les champs obligatoires sont correctement validés.
      3. Tester la gestion des cas où la création de véhicule échoue.
    * **Test de Consultation de la Liste des Véhicules :**
      1. Vérifier que la liste des véhicules récupérée correspond aux données attendues.
      2. Tester les cas où aucun véhicule n'est enregistré pour un client donné.
      3. S'assurer que seuls les utilisateurs autorisés ont accès à la liste des véhicules.
11. **Intégration des Tests dans le Processus de Développement:**
    * Intégrez les tests unitaires dans le processus de développement, en les exécutant automatiquement à chaque modification du code.
    * Utilisez des outils de tests unitaires tels que PHPUnit pour Laravel.
12. **Mesures de Couverture de Code :**
    * Suivez la couverture de code obtenue par les tests unitaires pour vous assurer qu'une grande partie du code est testée.
13. **Révision des Tests et Correction des Bogues :**
    * En cas d'échec des tests unitaires, révisez le code pour identifier et corriger les erreurs.
    * Révisez régulièrement les tests pour vous assurer qu'ils restent pertinents avec l'évolution du code.