

Cahier des charges

Demande de Mr. Stationneur – 20 Octobre 2025

DÉVELOPPEMENT D'UN PROGRAMME NUMÉRIQUE

DESTINÉ À LA GESTION D'UN PARKING

Concaténatation

Table des matières

1.	Introduction	3
1.1.	Objet du document	3
1.2.	Contexte	3
1.3.	Parties prenantes au projet	3
2.	Description générale du système	4
2.1.	Objectif du projet	4
2.2.	Fonctions du produit	4
2.3.	Profils et utilisateurs concernés par le besoin	4
2.4.	Contraintes générales du système	4
3.	Exigences fonctionnelles	5
3.1.	Données et sauvegarde	5
3.2.	Gestion des paiements	5
3.3.	Gestion des places de parking	6
3.4.	Tarifs et abonnements	6
3.5.	Options de confort (facultatives mais appréciées)	6
4.	Exigences non fonctionnelles	6
4.1.	Ergonomie et simplicité d'usage	6
4.2.	Sécurité et gestion des erreurs	7
5.	Interfaces externes	7
5.1.	Interfaces utilisateur	7
5.2.	Interfaces matérielles et réseau	7
5.3.	Interfaces logicielles	7
6.	Gestion du projet	8
6.1.	Livrables attendus	8
6.2.	Planning et jalons	8
6.3.	Organisation	8
7.	Critères d'acceptation et validation	8
8.	Annexes	9

1. Introduction

1.1. Objet du document

Le présent cahier des charges a pour objectif de définir les besoins et les fonctionnalités attendues pour le développement d'un programme numérique destiné à la gestion d'un parking.

1.2. Contexte

Le client, Antoine Stationneur, est gérant d'un parking situé au centre de la ville de StationVille.

Actuellement, Mr. Stationneur assure la gestion de son parking à l'aide de carnets papier et de relevés manuels. Cette méthode, longue et fastidieuse, lui demande une grande concentration et entraîne régulièrement des erreurs.

Afin d'améliorer son organisation et d'optimiser son temps de travail, M. Stationneur souhaite disposer d'une interface informatique, installée sur l'ordinateur de sa cabine, qui lui permettrait d'assurer la gestion du parking de manière plus simple, plus rapide et plus fiable.

1.3. Parties prenantes au projet

Le client est Mr. Antoine Stationneur, gérant de parking.

Les prestataires sont les membres de l'équipe de développement du nom de Concaténatation et est composée de :

- Gaspard Munguia Coca
- Cyril Schweicher
- Thomas Charlier
- Sofiane Amqrane

2. Description générale du système

2.1. Objectif du projet

Comme mentionné précédemment, Mr. Stationneur éprouve des difficultés à maintenir une organisation claire dans son travail. L'objectif principal de cette application est donc de l'aider à gérer son activité de manière plus ordonnée et plus efficace depuis son ordinateur.

2.2. Fonctions du produit

Le rôle général du système consiste à mettre en place une solution permettant de structurer, centraliser et sauvegarder l'ensemble des informations liées à la gestion du parking.

2.3. Profils et utilisateurs concernés par le besoin

Les profils ciblés par la mise en place du programme sont :

- Antoine Stationneur, le client et le gérant du parking (supervision et gestion globale)
- Les employés de cabine de Mr. Stationneur (utilisation quotidienne)
- Les clients du parking de Mr. Stationneur (bénéficieront d'un service plus fluide).

2.4. Contraintes générales du système

Les contraintes du client sont majoritairement graphiques et techniques :

❖ **Philosophie générale :**

- Le programme doit être simple, clair et intuitif, conformément à la demande du client.
- L'interface doit permettre à l'utilisateur de comprendre l'état du parking rapidement et sans effort.

❖ **Disposition de l'interface (GUI) :**

- **Écran principal** : tableau de bord visuel représentant l'état du parking.
- **Menu latéral gauche** : accès rapide aux sections *Entrée*, *Sortie*, *Paiement*, *Abonnés* et *Rapports*.
- **Zone centrale** : vue des places de parking, avec un code couleur indiquant leur état d'occupation.
- **Bandeau supérieur** : barre d'état affichant la date, l'heure, le taux d'occupation et les messages système.
- **Organisation par onglets ou boutons fixes** (aucun menu déroulant caché).

- o Absence de pop-ups non essentiels.
- ❖ Palette de couleurs :
 - o Palette sobre et contrastée demandée par le client.
 - o Interface utilisable dans un environnement très lumineux comme sombre.
- ❖ Typographie :
 - o Police lisible à environ deux mètres de distance.
 - o Éviter les textes entièrement en majuscules.
- ❖ Ergonomie :
 - o Toute fonctionnalité doit être accessible en environ trois clics.
 - o Raccourcis clavier personnalisables.
 - o Boutons de grande taille, accompagnés d'icônes et de texte.
 - o Retour visuel immédiat après chaque action.
 - o Limiter les doubles confirmations, sauf pour les actions critiques (ex. suppression).
- ❖ Rapports et statistiques :
 - o Interface de consultation sobre : fond blanc, texte noir, graphiques colorés.
 - o Graphiques simples et facilement interprétables.
 - o Affichage d'un résumé clair des statistiques, sous forme textuelle et graphique.

3. Exigences fonctionnelles

3.1. Données et sauvegarde

- Possibilité d'effectuer une sauvegarde manuelle simple des données.
- Mise en place d'un journal de sauvegarde permettant de tracer l'historique des enregistrements.

3.2. Gestion des paiements

- Option pour annuler ou corriger un paiement en cas d'erreur.
- Indication du moyen de paiement utilisé (espèces, carte, abonnement, etc.).
- Génération d'un résumé clair des paiements en fin de journée.
- Présence d'un bouton permettant de clôturer officiellement la journée.

3.3. Gestion des places de parking

- Affichage d'un plan global du parking avec un code couleur indiquant l'état des places.
- Recherche rapide par numéro de plaque ou par nom d'abonné.
- Visualisation en temps réel de la libération ou de l'occupation d'une place.
- Possibilité de marquer une place comme indisponible (maintenance, réservation, etc.).
- Affichage d'un message d'avertissement lorsque 90 % de la capacité du parking est atteinte.

3.4. Tarifs et abonnements

- Mise à jour des différents types d'abonnements disponibles.
- Modification des tarifs et des paramètres liés aux abonnements.
- Création ou reconnaissance automatique d'un nouveau client dans le système.

3.5. Options de confort (facultatives mais appréciées)

- Présence d'un résumé visuel regroupant les informations essentielles.
- Affichage permanent de l'horloge et de la date.
- Section dédiée à la prise de notes ou à des observations diverses.

4. Exigences non fonctionnelles

4.1. Ergonomie et simplicité d'usage

- Le système doit permettre d'effectuer toutes les actions sans nécessiter l'usage du clavier ou de la souris, afin de garantir une utilisation optimale en situation réelle (cabine exiguë, gants, etc.).
- Les raccourcis clavier doivent être personnalisables en fonction des besoins de l'utilisateur.
- L'interface doit rester parfaitement lisible en toute circonstance.
- Aucun écran ne doit bloquer l'utilisateur : pas d'action qui force l'arrêt du flux de travail inutilement.

4.2. Sécurité et gestion des erreurs

- Toute suppression d'un élément doit être précédée d'une demande de confirmation.
- Un journal d'actions (logs) doit être conservé afin d'assurer la traçabilité des opérations.
- Le système doit fonctionner hors ligne et ne pas nécessiter de connexion Internet.

5. Interfaces externes

5.1. Interfaces utilisateur

Le système proposera plusieurs modes d'interaction afin de s'adapter aux besoins et préférences de l'utilisateur :

- Interface en ligne de commande (CLI)** : doit permettre d'effectuer les principales opérations via des commandes textuelles. Cette interface sera utile pour les tâches techniques, le débogage ou l'automatisation.
- Interface graphique (GUI)** : offrira une expérience utilisateur visuelle, intuitive et ergonomique, conformément aux exigences définies dans les sections précédentes.

5.2. Interfaces matérielles et réseau

- Le système doit fonctionner localement sur un ordinateur standard utilisé dans la cabine du parking.
- Aucune connexion Internet ne doit être requise, conformément à la demande du client, garantissant l'indépendance et la sécurité du système.
- Le logiciel communiquera uniquement avec les périphériques locaux nécessaires (ex. clavier, souris, éventuellement imprimante de tickets si ajouté ultérieurement).

5.3. Interfaces logicielles

- Le système doit utiliser une solution de stockage basée sur des fichiers structurés.
- Les interactions avec les fichiers se feront via des bibliothèques standards du langage utilisé, garantissant stabilité et maintenabilité.

6. Gestion du projet

6.1. Livrables attendus

Le projet produira les livrables suivants :

- Code source complet
- Documentation utilisateur
- Plan de test et rapport de validation, comprenant les tests unitaires, fonctionnels et d'intégration.

6.2. Planning et jalons

Le projet sera organisé en plusieurs phases :

1. **Phase d'analyse et conception** : définition des besoins, rédaction du cahier des charges et création d'un diagramme UML.
2. **Phase de développement initial** : implémentation de l'architecture, des interfaces et du stockage des données.
3. **Phase de test intermédiaire** : tests unitaires et validation des fonctionnalités principales.
4. **Phase d'optimisation et d'ergonomie** : amélioration de l'interface utilisateur et des performances.
5. **Phase finale** : tests complets, correction des bugs, rédaction de la documentation et livraison du système.

6.3. Organisation

Les membres du groupe Concaténatation s'organiseront entre eux et s'attribueront des tâches à faire quotidiennement pour livrer le produit dans le temps imparti.

Des échanges réguliers via réunions de suivi, comptes-rendus et outils de gestion de projet seront mis en place pour garantir la transparence et la traçabilité des décisions

7. Critères d'acceptation et validation

Des tests unitaires seront mis en place pour chaque fonction du programme pour s'assurer de son bon fonctionnement.

Le programme sera testé par l'entièreté du groupe pour s'assurer de son bon fonctionnement et trouver les éventuelles failles restantes. Elles seront ensuite corrigées puis vérifiées.

8. Annexes

Diagramme UML :

