

# CAHIER DES CHARGES

T u b o P a r k  
2 0 2 4



Préparé par:

TuboQuest

15 avenue du Rhin - Strasbourg

# Sommaire



1

Introduction

2

Membres de  
l'équipe

3

Le projet

4

Maquette

5

Aspect financier

6

Aspect  
technique

7

Planning et  
développement

8

Conclusion

# 1 – Introduction

Vous étudiez dans une école incapable d'avoir un parking ? **Nous** avons la solution. **Tubopark** est un disque de stationnement électronique, qui peut être géré à distance, via votre smartphone ! En un clic vous pouvez modifier l'heure du disque, planifier le changement ou alors simplement nous laisser gérer cela automatiquement.

Dans un monde de plus en plus connecté et digitalisé, la gestion du stationnement reste une tâche souvent fastidieuse et source de stress pour de nombreux usagers. Les disques de stationnement traditionnels, bien qu'ayant simplifié la régulation du temps de stationnement dans les zones urbaines, présentent des limitations en termes de praticité et de contrôle. C'est dans ce contexte que s'inscrit notre projet de développement d'une application mobile innovante dédiée à la gestion automatisée et à distance des disques de stationnement.



## 2 – Membres de l'équipe

### Colette OSWALD – Gestion de projet & Développeuse Web

#### Objectif Principaux :

- Mise en place du cahier des charges.
- Création des maquettes
- Création du site web



### Philippe LOCATELLI – Développeur IA

#### Objectif Principaux :

- Création du module physique du disque
- Mise en place du modèle IA



### Alexis HENRY – Développeur Backend

#### Objectif Principaux :

- Mise en place de l'API



**Florian Wu – Développeur Web Mobile****Objectif Principaux :**

- Création de l'interface Mobile – Front

**Valentin LORQUIN – Développeur Web****Objectif Principaux :**

- Création de la plateforme de jeu

**Benjamin FAECHTIG – Développeur Web Mobile****Objectif Principaux :**

- Mise en place de la base de donnée
- Création de l'interface Mobile – Front





## 3 – Le projet

### 3.1 – Objectifs

Ce projet vise à développer une application mobile innovante pour la gestion automatisée et à distance des disques de stationnement. L'objectif principal est de simplifier et d'optimiser l'expérience des usagers en leur permettant de gérer leur stationnement de manière intuitive et pratique, tout en offrant aux autorités compétentes un outil de contrôle et de gestion efficace.

L'application permettra aux utilisateurs de configurer leur disque de stationnement, de le démarrer et de le stopper à distance, de recevoir des notifications et des rappels, ainsi que de suivre l'historique de leur utilisation. Cela contribuera à réduire les contraintes liées au stationnement.

### 3.2 – Cible

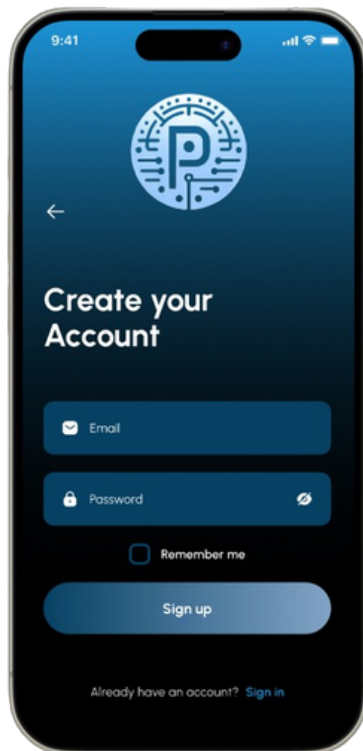
Principalement les conducteurs urbains qui utilisent régulièrement des zones de stationnement régulées par des disques. Ils bénéficieront d'une solution pratique et moderne pour gérer leur stationnement.

\*

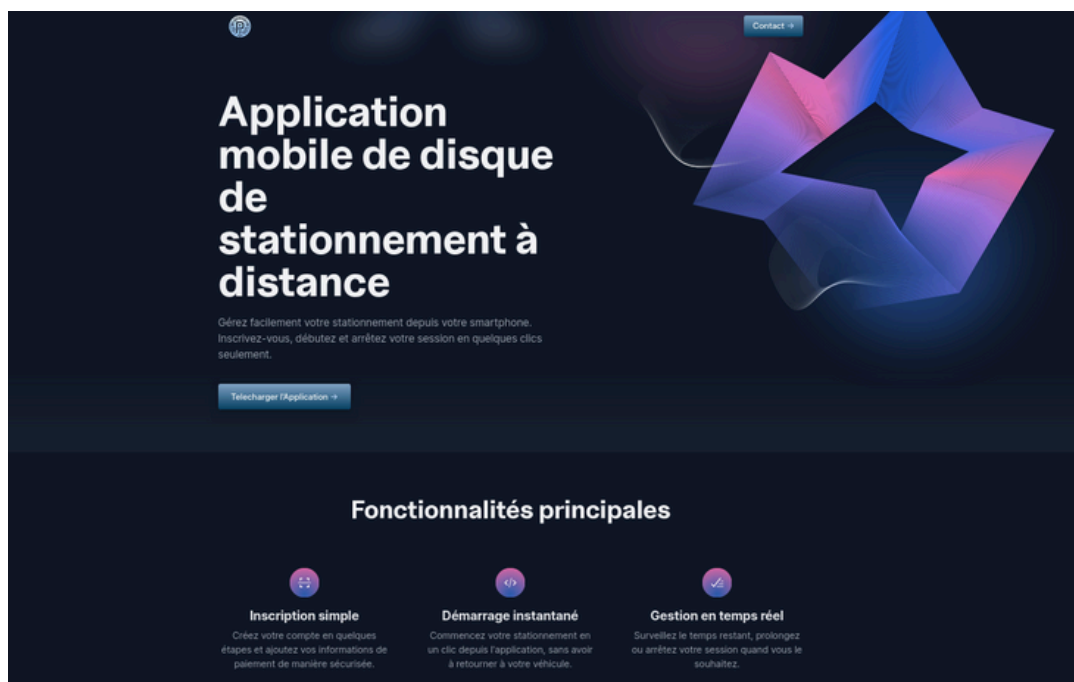
**Il est à préciser que le cahier des charges aura une évolution constante en vue de l'avancement du projet**

## 4 – Maquettes

### Interface Application Mobile



### Interface Web



## 5 – Aspect Financier

Matériel	Quantité
Raspberry pi 5	1
AI kit hailo pour raspberry pi 5	1
Stepper motor (moteur pas à pas)	1
Module moteur	1
Camera module 3	1
Nappe camera pour raspi 5	1
Battery	

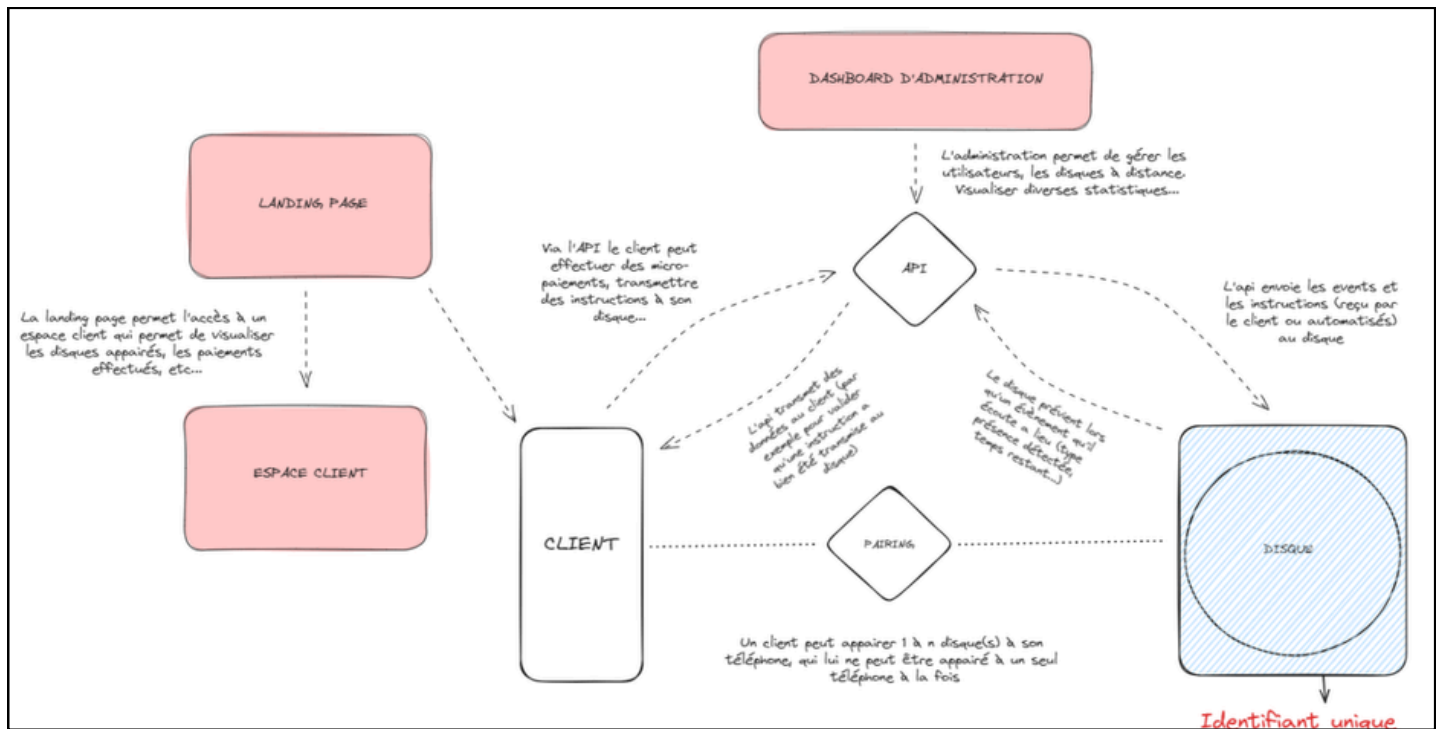
\* Le module wifi ne permet pas une communication par internet, ce qui fait que dans notre cas le disque ne sera atteignable qu'au sein du réseau sur lequel il est connecté. Nous avons fait ce choix pour des raisons de coûts, un module permettant une connexion 4G/5G est plus cher et nécessite également une carte SIM, donc un abonnement auprès d'un opérateur.



## 6 – Aspect Technique

### 6.1 – Structure

Ce projet nécessite la mise en place de plusieurs applications qui vont pour la plupart interagir entre elles.



## 6.2 – Applications et technologies utilisées

### 6.2.1 – Page de destination

**Technologie** : Next.js, TypeScript, TailwindCSS

**Description** : Met en avant l'application et ses fonctionnalités.

### 6.2.2 – API

**Technologie** : Laravel

**Description** : Gère la liaison entre les clients et les disques de stationnement, l'authentification, et les micro-paiements.

### 6.2.3 – Application Mobile

**Technologie** : React Native, TypeScript, TailwindCSS

**Description** : Produit principal permettant de gérer le disque de stationnement.

### 6.2.4 – Disque de stationnement automatisé

**Technologie** : Raspberry Pi 5, C++/Arduino

**Description** : Disque de stationnement avec caméra, wifi, servomoteur, batterie, etc.

### 6.2.3 – Modèle d'IA pour la détection vidéo

**Technologie** : Python

**Description** : Analyse du flux vidéo pour détecter des événements spécifiques

### 6.2.4 – Jeu interactif

**Technologie** : Next.js

**Description** : Éduque les utilisateurs aux bonnes pratiques de stationnement.

## 6.3 – Fonctionnalité Principales

- Partager les détails du stationnement (lieu, durée, etc.) avec ses amis.
- Historique complet des stationnements passés, incluant les lieux, les durées et les frais associés.
- Sauvegarde des paramètres et des préférences de l'utilisateur dans le cloud pour une récupération facile en cas de changement de téléphone.
- Demander l'adresse à l'utilisateur lors du démarrage du disque et possibilité d'entrer des lieux de stationnement régulier
- Recevoir des notifications en fonction du temps de stationnement restant ou autres événements.
- Option pour désactiver les notifications de stationnement et autres alertes pendant des périodes définies pour éviter les interruptions.
- Possibilité d'ajouter des widgets sur son smartphone pour un accès rapide aux fonctions principales.
- Support client intégré pour une assistance en temps réel en cas de problème avec le disque.

## 6.4 – Fonctionnalité Prémium

- Automatisation du disque (rotation automatisée pour suivre le temps actuel, tout en vérifiant à l'aide de la caméra la présence si il n'y a pas de risques)
- Accès au flux vidéo enregistré par la caméra embarquée sur le disque de stationnement.
- Activation d'un mode sentinelle qui alerte des présences autour du véhicule.
- Mise à disposition de statistiques d'usage (temps de stationnement...).
- Visualiser la batterie restant ainsi qu'une estimation du temps de fonctionnement restant.
- Possibilité de changer le thème du disque de stationnement présent sur l'application.

# 7 – Planning et développement

Toutes les tâches vont être créés et suivies à l'aide de Github Project.  
Chaque projet aura son Github Project et les tâches associées y seront ajoutées.

## 7.1 – Général

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Cahier des charges	2	Colette
Keynote	2	Tuboquest
Tests et validation	6	Tuboquest
Documentation technique	4	Colette
Conception de la base de données	2	Benjamin
Réalisation des supports de présentation	2	Colette
Total	18	

# 7.1 – Page de destination (Next.js)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Création de la maquette	0.5	Colette
Page de mise en avant	1.5	Colette
Intégration de la maquette	1	Colette
Lien de téléchargement de l'application	0.06	Colette
Démo d'utilisation de l'application	0.13	Colette
Gestion du responsive	0.25	Colette
Total	2	

## 7.2 – API (Laravel)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Mise en place de la base de données	0.5	Benjamin
Création des migrations, entités, relations, seeders	6	Benjamin
Authentification	1.5	Alexis
Administration	3.5	Alexis
Appairage	2	Alexis
Communication bidirectionnelle	12	Alexis
Sauvegarde des données	0.5	Alexis
Statistiques d'usage	2	Alexis
Expérience utilisateur	0.13	Alexis
Gestion de l'abonnement premium	2	Alexis
Caméra	5	Colette
Notifications	1	Alexis
Documentation de l'API	1	Alexis
Total	36	



# 7.3 – Disque de stationnement automatisé (C++)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Mechanical assembly & tests	1.5	Philippe
Connect to Main API for receiving rotation angle data	1	Philippe
Connect to AI model API	1	Philippe
Secure connection with token	1	Philippe
Integration with a camera module for video streaming	0.5	Philippe
Real-time video streaming capabilities	1	Philippe
Network connectivity for sending video stream to the model	2	Philippe
Remote device case conception	1	Philippe
Total	9	

## 7.4 – Système d'IA intégré (Python)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Test different model and select one depending on benchmark	2	Philippe
Set up video recognition model	3	Philippe
Set up video stream & pipeline	2	Philippe
Handle real-time video streaming flow & routes (sentinel mode)	3	Philippe
Handle video detection and alert (send an alert on trigger)	2	Philippe
Set up socket for communication	1	Philippe
<b>Total</b>	<b>13</b>	

## 7.5 – Jeu de parking (Next.js)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Initial setup	2	Valentin
Game logic	5	Valentin
Graphisme	4	Valentin
User interface	4	Valentin
Sound FX and soundtrack	3	Valentin
Optimisation & tests	4	Valentin
<b>Total</b>	<b>22</b>	

## 7.6 – Application mobile (React Native)

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Maquette	4	Colette
Authentification et paramètres utilisateur	4	Florian
Gestion du disque de stationnement	4	Florian
Géolocalisation et caméra	5	Florian
Notifications et alertes	1	Florian
Contrôles et interactions	3	Florian
Statistiques et analyses	1	Florian
<b>Total</b>	<b>22</b>	

## 7.7 – Fonctionnalités premium de l'application mobile

Fonctionnalités	Durée (jours)	Personne(s) assignée(s)
Historique et Rapports	1	Benjamin
Sécurité et Surveillance	4	Benjamin
Géolocalisation Avancée	2	Benjamin
Widgets et raccourcis	2	Benjamin
Options de Paiement	2	Benjamin
Total	9	

## 7.8 – Récapitulatif

Personnes	Durée (jours)
Colette	19
Philippe	22
Alexis	26
Florian	20
Benjamin	18
Valentin	22

## 8 – Conclusion

Le projet **TuboPark** vise à révolutionner la gestion du stationnement grâce à un disque électronique innovant, contrôlable à distance via une application mobile. Ce cahier des charges a détaillé les objectifs, les fonctionnalités, et les technologies nécessaires pour mener à bien ce projet ambitieux.

Nous avons défini des fonctionnalités de base et premium pour répondre aux besoins variés des utilisateurs, incluant des options de partage, des notifications intelligentes, et des modes de sécurité avancés. L'infrastructure technique repose sur une intégration cohérente entre une page de destination web, une API robuste, une application mobile intuitive, et un disque de stationnement automatisé avec des capacités de détection vidéo via un modèle d'IA.

L'équipe multidisciplinaire assignée au projet possède les compétences nécessaires pour couvrir tous les aspects, du développement logiciel à la conception matérielle. La répartition des tâches est claire et structurée, assurant une gestion efficace et une réalisation dans les délais impartis.

Le succès de **TuboPark** repose sur une collaboration étroite entre les membres de l'équipe, une utilisation judicieuse des technologies de pointe, et une attention constante aux besoins des utilisateurs finaux. En suivant ce cahier des charges, nous sommes convaincus de pouvoir offrir une solution de stationnement moderne, pratique et sécurisée, qui répondra aux attentes des utilisateurs et se démarquera sur le marché.

Nous sommes impatients de voir **TuboPark** prendre vie et transformer l'expérience de stationnement pour tous ses utilisateurs.