

Membres

1. TCHAPDA ALAN – 24P761
2. FUL FONKWA AMANDINE – 24P753
3. FOTEDOU BILL JUNIOR – 21P023
4. FOSSO STEPHANE – 24P766
5. NDJOMO JOSEPH – 24P758
6. AGHETSEH NGOH – 22P612
7. MBOUGANG IGOR-FRED – 22P341
8. ATCHUNTCHE DAREEN TAKWI – 24P751
9. ARREYNTOW KERONE ENOW – 22P601
10. MBAH MOFFO JOSUE – 22P214
11. FONKWA HERMANN FUL – 24P752
12. BAKOTCHA KAMTCHE LOÏC — 21p320

Table des matières

1	Introduction	4
1.1	Contexte	4
1.2	Enjeux	4
1.3	Objectifs	4
1.4	Portée du Projet	5
2	Description des fonctionnalités du Système de Gestion de Parking Intelligent	5
2.1	Présentation générale	5
2.2	Fonctionnalités principales	5
2.2.1	Détection et surveillance des places	5
2.2.2	Réservation à distance	6
2.2.3	Entrée du véhicule et identification	6
2.2.4	Gestion automatique du stationnement	6
2.2.5	Facturation et paiement automatisés	7
2.2.6	Gestion des dépassements et amendes	7
2.2.7	Interface d'administration	7
2.2.8	Signalisation et affichage local	7
2.3	Fonctionnalités supplémentaires et perspectives	8
2.4	Avantages du système	8
3	Livrables Attendus	8
3.1	Phase 1 : Étude de Faisabilité et Conception Détailée	8
4	Chronogramme du projet	11
5	Resource nécessaires	11
5.1	Microcontrôleur principal	12
5.2	Module de communication IoT	12
5.3	Capteurs de détection de véhicules	12
5.4	Mécanisme de barrière	12
5.5	Module de paiement automatisé	13
5.6	Alimentation électrique	13
5.7	Infrastructure serveur et cloud	13
5.8	Matériel complémentaire	13
5.9	Récapitulatif global du matériel	14

6 Logiciels requis et Stack Technologique	14
6.1 Présentation générale	14
6.2 Logiciels requis	14
6.3 Stack technologique (React-based)	15
6.4 Fonctionnalités logicielles supportées	15
7 Analyse de l'Architecture pour le Système de Parking Intelligent (LoRaWAN + RFID)	16
8 Répartition des tâches	18
9 Analyse des Risques et Mitigations	18
9.1 Risques Techniques	18
9.2 Risques Financiers	19
9.3 Risques Technologiques	19
9.4 Risques Opérationnels	20
9.5 Risques Stratégiques	20
9.6 Risques Sécuritaires	21
9.7 Risques Logistiques	21
10 References	23

Projet de Système de Parking Intelligent

1 Introduction

1.1 Contexte

Avec l'augmentation de la population urbaine et la croissance des véhicules, la gestion des parkings devient un défi majeur dans les villes modernes. Les conducteurs passent souvent un temps considérable à chercher des places de stationnement, ce qui entraîne une congestion accrue, une pollution de l'air et une frustration chez les utilisateurs. Les systèmes de parking intelligents cherchent à optimiser l'utilisation des espaces de stationnement en intégrant des technologies avancées telles que les capteurs, les applications mobiles et les systèmes de gestion de données.

1.2 Enjeux

- **Optimisation de l'Espace :** Maximiser l'utilisation des places de stationnement disponibles pour réduire le besoin de nouvelles infrastructures.
- **Réduction de la Congestion :** Diminuer le trafic lié à la recherche de places de stationnement, ce qui contribue à la fluidité du trafic urbain.
- **Amélioration de l'Expérience Utilisateur :** Offrir aux conducteurs une solution simple et efficace pour trouver et réserver des places de stationnement.
- **Impact Environnemental :** Réduire l'empreinte carbone en diminuant le temps que les véhicules passent à circuler sans but.

1.3 Objectifs

- **Développer une Application Mobile :** Créer une application qui permet aux utilisateurs de localiser, réserver et payer leur place de stationnement en temps réel.

- **Intégration de Capteurs** : Installer des capteurs dans les parkings pour détecter la disponibilité des places et transmettre ces informations à l'application.
- **Analyse des Données** : Utiliser des outils d'analyse de données pour surveiller l'utilisation des parkings et prévoir les besoins futurs.

1.4 Portée du Projet

- **Zone Géographique** : Définir les zones urbaines spécifiques où le système sera déployé, en tenant compte des besoins locaux.
- **Fonctionnalités** : Inclure des fonctionnalités telles que la réservation de places, le paiement en ligne, des notifications en temps réel sur la disponibilité, et des statistiques sur l'utilisation des parkings.
- **Technologies Utilisées** : Intégrer des technologies telles que l'IoT (Internet des Objets), des applications web et mobiles, des bases de données pour le stockage des informations, et des algorithmes d'analyse de données.
- **Durée du Projet** : les phases de conception, développement, et tests devront durer environ 3 mois

2 Description des fonctionnalités du Système de Gestion de Parking Intelligent

2.1 Présentation générale

Le Système de Gestion de Parking Intelligent est une solution automatisée basée sur l'Internet des Objets (IoT) et les microcontrôleurs (Arduino/ESP32) permettant de superviser, contrôler et automatiser la gestion d'un parking en temps réel. Il vise à optimiser l'utilisation des places, réduire la congestion et faciliter le paiement grâce à une architecture connectée composée de capteurs, caméras, modules IoT et d'interfaces utilisateur.

2.2 Fonctionnalités principales

2.2.1 Détection et surveillance des places

- Chaque place de parking est équipée d'un capteur à infrarouge (IR) capable de détecter la présence ou l'absence d'un véhicule.
- Les données sont transmises à une carte Arduino/ESP32 qui met à jour automatiquement l'état de chaque place dans la base de données cloud.

- États possibles d'une place :
 - Disponible
 - Occupée
 - Réservée
- Informations visibles en temps réel sur un écran LCD à l'entrée du parking et sur l'application mobile/web.

2.2.2 Réservation à distance

- L'utilisateur peut réserver une place via une application mobile ou un site web.
- Lors de la réservation, l'utilisateur choisit :
 - La durée souhaitée (ex. 2h, 4h)
 - Le mode de paiement (Mobile Money)
- Une fois confirmée :
 - La place passe au statut "Réservée"
 - L'utilisateur reçoit la confirmation
- Non-présentation : la réservation est annulée automatiquement.

2.2.3 Entrée du véhicule et identification

perspective du prototypes proposées :

1. **Identification par badge RFID**
 - Chaque utilisateur possède une carte RFID.
 - Scan à l'entrée → accès si place disponible.
2. **Code ou QR Code de réservation**
 - Vérification du code sur lecteur optique avant ouverture de barrière.

2.2.4 Gestion automatique du stationnement

- Heure d'entrée enregistrée automatiquement.
- Suivi en temps réel de la durée de stationnement.
- Capteurs indiquent si le véhicule est présent.
- Départ → heure de sortie enregistrée et disponibilité mise à jour.

2.2.5 Facturation et paiement automatisés

- Utilisateur sans réservation
 - Heure d'entrée enregistrée automatiquement
 - Montant = Durée (minutes) × Tarif unitaire
 - Paiement via Mobile Money
 - Après paiement, barrière s'ouvre
- Utilisateur avec réservation
 - Tarif selon durée réservée
 - Dépassement → amende automatique

2.2.6 Gestion des dépassements et amendes

- Surveillance du temps restant pour chaque véhicule
- Dépassement → statut “En infraction” et amende :

$$Amende = Tarif_amende \times Temps_dpassement$$

- Notification utilisateur via application ou écran
- Paiement obligatoire avant sortie

2.2.7 Interface d'administration

- Visualisation globale du parking (libre, occupé, réservé, en infraction)
- Gestion des utilisateurs, véhicules, abonnements
- Suivi en temps réel des entrées, sorties, paiements
- Consultation des rapports et statistiques
- Gestion des tarifs, amendes et plages horaires

2.2.8 Signalisation et affichage local

- Afficheurs lumineux pour l'état des places :
 - Libre
 - Occupée
 - Réservée
- Écran d'entrée : nombre total de places, libres, tarif horaire, instructions
- Buzzer pour erreurs ou accès non autorisé

2.3 Fonctionnalités supplémentaires et perspectives

- Historique complet des entrées, sorties et paiements
- Notifications automatiques (SMS, push, e-mail)
- Statistiques analytiques : taux d'occupation, recettes, amendes
- Sécurité renforcée : journalisation des accès, détection d'anomalies

2.4 Avantages du système

- Réduction du temps de recherche d'une place
- Diminution de la congestion et de la pollution
- Gestion efficace et automatique des paiements
- Expérience utilisateur fluide
- Possibilité d'intégration future dans une infrastructure Smart City

3 Livrables Attendus

3.1 Phase 1 : Étude de Faisabilité et Conception Détaillée

Cette phase initiale vise à définir précisément le périmètre, les objectifs, l'architecture et les technologies du projet.

Phase 1 — Étude & conception (Semaines 1–2)

Objectif : Comprendre le projet, choisir les composants, planifier l'architecture.

Tâches :

- Analyse des besoins détaillée (fonctions, contraintes, acteurs).
- Étude de faisabilité (composants nécessaires, coûts, LoRa vs Wi-Fi).
- Schéma d'architecture : capteurs → passerelle LoRa → serveur → application web.
- Choix du matériel : capteurs ultrason, RFID, module LoRa, microcontrôleur (Arduino Uno).
- Installation des environnements : Arduino IDE, serveur local (XAMPP / Node.js), sandbox Orange/MoMo.

Livrables :

- Diagramme d'architecture.
- Cahier de charges finalisé.
- Liste des composants.

Phase 2 — Prototype matériel local (Semaines 3–4)

Objectif : Faire fonctionner localement les capteurs et les barrières.

Tâches :

- Montage des capteurs de présence .
- Contrôle de la barrière avec servo-moteur.
- Test de détection (affichage sur le moniteur série).
- Lecture RFID (identification d'un véhicule).

Livrables :

- Prototype matériel local fonctionnel.
- Programme Arduino de base.

Phase 3 — Communication LoRa (Semaines 5–6)

Objectif : Faire remonter les données capteurs via LoRa.

Tâches :

- Connexion et configuration du module LoRa (émetteur/récepteur).
- Envoi/réception des données capteurs (état des places).
- Développement d'un script côté passerelle (Python ou Node.js).
- Validation : affichage des données capteurs côté serveur.

Livrables :

- Lien fonctionnel LoRa → passerelle → PC.
- Code Arduino et script serveur.

Phase 4 — Interface web (Semaines 7–8)

Objectif : Afficher et gérer les places disponibles.

Tâches :

- Création d'un tableau web des places.
- Rafraîchissement automatique selon les données reçues (API ou base de données).
- Simulation d'un écran numérique pour le portail.

- Mise en base de données : places, utilisateurs, réservations.

Livrables :

- Interface web fonctionnelle.
- Connexion serveur base de données.

Phase 5 — Paiement & RFID (Semaines 9–10)

Objectif : Intégrer le paiement et l'identification du véhicule.

Tâches :

- Liaison RFID utilisateur véhicule.
- Intégration de l'API Orange Money / MTN MoMo (mode sandbox).
- Association du montant selon la durée de stationnement.
- Tests de paiement simulé.

Livrables :

- Paiement en sandbox fonctionnel.
- Liaison RFID – paiement validée.

Phase 6 — Finalisation & tests (Semaines 11–12)

Objectif : Démonstration finale stable et documentée.

Tâches :

- Tests d'intégration de tous les modules (LoRa, capteurs, interface, paiement).
- Correction des bugs et optimisation des temps de réponse.
- Rédaction du rapport et préparation de la soutenance (slides, vidéo, démo).

Livrables :

- Prototype final complet.
- Rapport de projet et présentation PowerPoint.

Résumé global du planning

Phase	Période	Objectif principal	Livrable clé
1	Sem. 1–2	Étude & conception	Cahier de charges + schéma
2	Sem. 3–4	Prototype matériel	Montage + code Arduino
3	Sem. 5–6	Communication LoRa	Transmission valide
4	Sem. 7–8	Interface web	Tableau affichage dynamique
5	Sem. 9–10	Paiement & RFID	Paiement simulé
6	Sem. 11–12	Finalisation & rapport	Prototype complet

4 Chronogramme du projet

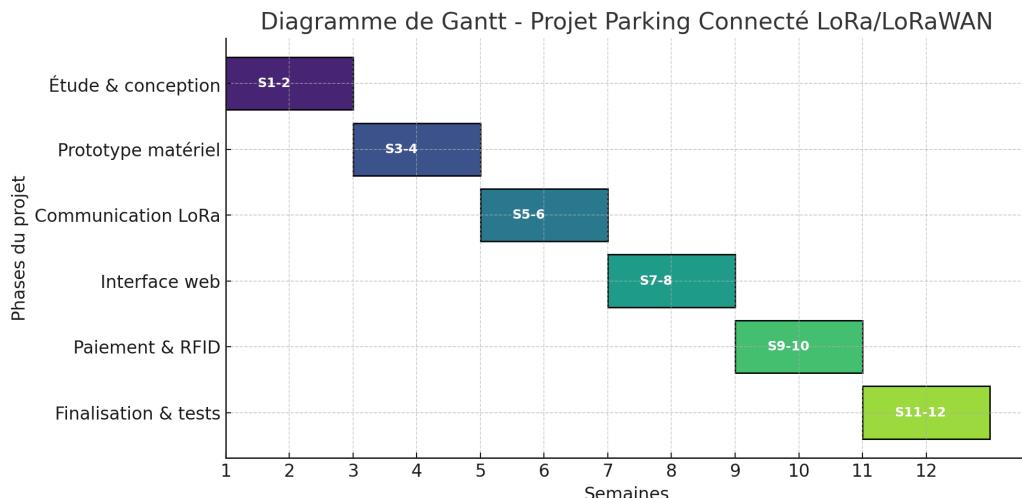


FIGURE 1 – Chronogramme global du projet de parking connecté

5 Resource nécessaires

Dans le cadre de la conception d'un système de parking intelligent, le matériel doit permettre de détecter les véhicules, de contrôler les barrières, de gérer la communication entre les modules et de faciliter le paiement automatisé. Le présent rapport liste et décrit les principaux composants nécessaires à la mise en œuvre du système.

5.1 Microcontrôleur principal

Composant / Référence	Description / Rôle
ESP32	Microcontrôleur 32 bits avec Wi-Fi et Bluetooth intégrés. Sert à gérer les capteurs, à traiter les données et à communiquer avec le serveur IoT.
Rôle	Lecture des capteurs d'entrée et de sortie. Gestion des signaux de la barrière. Communication via Wi-Fi ou LoRa. Envoi et réception de données vers le serveur.

5.2 Module de communication IoT

Composant / Référence	Description / Justification
Module LoRa SX1278 433MHz	Transmission longue portée et basse consommation. Permet la communication sans fil entre les points d'entrée/sortie et le serveur.
Justification	Portée jusqu'à 2 km, faible consommation, communication fiable point à point. LoRa est préféré à LoRaWAN pour la simplicité de mise en œuvre dans un projet scolaire.

5.3 Capteurs de détection de véhicules

Capteur / Référence	Description / Rôle
Capteur infrarouge IRSensor	Détection plus précise du passage sous la barrière.
Rôle général	Déterminer si une place est occupée. Déetecter un véhicule à l'entrée ou à la sortie. Déclencher l'ouverture de la barrière.

5.4 Mécanisme de barrière

Élément / Référence	Description
Servomoteur	Actionne mécaniquement la barrière selon les signaux envoyés par le microcontrôleur.
Structure mécanique	Support et bras mobile pour la barrière. Modèle imprimé en 3D.

5.5 Module de paiement automatisé

Composant / Référence	Description
Module RFID MFRC522	Permet à l'utilisateur de payer ou de s'identifier à l'aide d'une carte prépayée.
Écran LCD	Affiche le montant à payer et les instructions.
Lecteur QR code (optionnel)	Paiement via téléphone mobile.

5.6 Alimentation électrique

Élément	Description / Référence
Bloc d'alimentation	Fournit l'énergie aux composants (5V ou 12V). Adaptateur 12V-2A
Batterie (optionnelle)	Alimentation de secours en cas de coupure. Batterie 12V 2000mAh

5.7 Infrastructure serveur et cloud

Élément	Description / Exemple
Base de données	Stockage des entrées, sorties et paiements. Exemple : Firebase, MySQL
Dashboard IoT	Affichage en temps réel des informations. Exemple : ThingsBoard, Node-RED
API Backend	Interface entre système IoT et base de données

5.8 Matériel complémentaire

- Breadboard et câbles Dupont
- Résistances, LED et boutons poussoirs
- Boîtiers de protection pour modules électroniques
- Connecteurs et adaptateurs divers

5.9 Récapitulatif global du matériel

Catégorie	Composant	Quantité	Prix (CFA)
Microcontrôleur	ESP32/ arduino uno	2	10k
Communication	LoraWAN	1	5k
Capteurs	IR Distance sensor	2	5k
Actionneurs	Servomoteur	2	5k
Identification	RFID MFRC522	4	6k
Caméra	ESP32-CAM (optionnelle)	1	5k
Alimentation	Bloc 12V + câbles	1	3k
Display	LCD	1	2k

6 Logiciels requis et Stack Technologique

6.1 Présentation générale

Le Système de Gestion de Parking Intelligent nécessite un ensemble cohérent de logiciels pour assurer la supervision, le contrôle et l'automatisation du parking en temps réel. Le système doit être accessible via une application web, une application mobile et intégrer la communication avec les capteurs et dispositifs IoT (Arduino/ESP32).

6.2 Logiciels requis

- **Système d’exploitation :** Windows 10/11, macOS ou Linux pour le développement, Android et iOS pour les mobiles.
- **Environnement de développement intégré (IDE) :**
 - Visual Studio Code (web et mobile React/React Native)
 - Arduino IDE (programmation ESP32 / capteurs)
- **Base de données :** PostgreSQL (stockage des utilisateurs, véhicules, réservations, paiements et capteurs)
- **Serveur d’API :** Node.js avec Express.js
- **Gestion des communications temps réel :**

- Socket.IO pour les mises à jour en temps réel sur les interfaces web et mobiles
- **Gestion des notifications :** Firebase Cloud Messaging (push), Twilio (SMS)
- **Plateformes de paiement :** Intégration Mobile Money (MTN, Orange Money), API bancaire

6.3 Stack technologique (React-based)

- **Frontend Web :** React.js + Tailwind CSS
- **Frontend Mobile :** React Native (iOS et Android)
- **Backend :** Node.js + Express.js (REST API)
- **Base de données :** mySQL
- **IoT et communication temps réel :**
 - ESP32 / Arduino pour capteurs et actionneurs
 - Socket.IO pour mises à jour en temps réel vers les applications
- **Stockage Cloud et hébergement :**
 - Vercel ou Netlify (frontend)
 - Heroku ou Railway (backend)
- **OCR et reconnaissance de plaque :** OpenALPR ou EasyOCR pour la lecture automatique des plaques

6.4 Fonctionnalités logicielles supportées

- Détection et suivi en temps réel des places de parking
- Réservation à distance et génération de QR Code pour l'accès
- Identification des véhicules via RFID
- Gestion automatique du stationnement et suivi des durées
- Facturation et paiement automatisé (avec amendes en cas de dépassement)
- Interface web d'administration pour gestion et statistiques
- Signalisation locale via écrans, alertes sonores et notifications
- Historique complet, notifications automatiques et statistiques analytiques

7 Analyse de l'Architecture pour le Système de Parking Intelligent (LoRaWAN + RFID)

L'adoption de la RFID implique que chaque véhicule autorisé devra être équipé d'un tag (une étiquette ou une carte) RFID. Le système est un modèle "fermé" ou "d'abonnés" (seuls les véhicules avec un tag sont gérés automatiquement).

A) Architecture Logique

L'architecture logique décrit le fonctionnement conceptuel du système, le flux des informations et les interactions entre les différents modules fonctionnels, sans spécifier les technologies précises. Elle peut se diviser en quatre couches principales.

Couche 1 : Perception et Action (Le terrain)

C'est la couche physique qui interagit directement avec les véhicules et l'environnement du parking.

- **Identification du Véhicule** : Un lecteur RFID à l'entrée et à la sortie scanne le tag du véhicule qui se présente. Il capture un identifiant unique (l'ID du tag).
- **Détection de Présence** : Des capteurs installés sur chaque place détectent si la place est physiquement occupée ou libre.
- **Action Physique** : Une barrière motorisée s'ouvre ou reste fermée en fonction des autorisations.
- **Information Visuelle** : Un écran à l'entrée affiche en temps réel le nombre de places disponibles.

Couche 2 : Collecte et Connectivité (Le réseau)

Cette couche assure la transmission des données brutes de la couche de perception vers le centre de traitement.

- **Réseau RFID** : Les lecteurs RFID communiquent les ID des tags lus, généralement via un réseau local (Ethernet ou Wi-Fi), au serveur d'application.
- **Réseau LoRaWAN** : Les capteurs de place de parking, optimisés pour une faible consommation, envoient des messages très courts ("occupé" ou "libre") via une passerelle LoRaWAN. La passerelle relaie ensuite ces messages vers le serveur via Internet.

Couche 3 : Traitement et Intelligence (Le cerveau)

C'est le cœur du système : le serveur applicatif (back-end) reçoit toutes les données, les traite et prend des décisions.

- **Gestion des Accès** : Vérifie l'ID RFID, valide l'abonnement, ouvre la barrière si autorisé et enregistre l'heure d'entrée.
- **Gestion de l'État du Parking** : Met à jour le statut de chaque place ("libre" / "occupé") et calcule le nombre total de places disponibles.
- **Logique Métier** : Calcule la durée de stationnement, gère la facturation, et génère des statistiques et rapports.
- **Stockage** : Toutes les informations (utilisateurs, tags RFID, état des places, historique) sont stockées dans une base de données.

Couche 4 : Application et Présentation (L'interface)

Cette couche permet aux humains d'interagir avec le système.

- **Interface Utilisateur (Conducteur)** : Une application mobile ou web pour visualiser la disponibilité des places, gérer le compte et consulter l'historique.
- **Interface Administrateur** : Un tableau de bord web permet de superviser l'état du parking, gérer les utilisateurs, et consulter les rapports.

B) Architecture Technique

L'architecture technique traduit les concepts logiques en choix concrets de matériel et de logiciels.

Composants Matériels (Hardware)

- **Identification** : Lecteurs RFID UHF longue portée et tags RFID UHF pour chaque véhicule.
- **Détection de Présence** : Capteurs LoRaWAN pour chaque place.
- **Connectivité** : Passerelle LoRaWAN, routeur/switch Ethernet.
- **Traitement central** : Serveur cloud (AWS, Google Cloud, Azure) ou local.
- **Contrôle d'accès et affichage** : Barrière motorisée, microcontrôleur ESP32, écran LCD.

En résumé, cette architecture est robuste et modulaire. La RFID simplifie et facilite l'identification, tandis que le LoRaWAN garantit une excellente autonomie pour les capteurs de place. La prochaine étape consiste à sélectionner le matériel spécifique et à commencer le développement logiciel.

8 Répartition des tâches

Task	Member 1	Member 2
Frontend	Dareen	Hermann
IoT Dashboard	Kerman	Amandine
API backend	Bakotcha	Alan
Base de donnée	Bilou	Bakotcha
Montage des composants (hardware)	Kerone	Aghetseh
Capteur infrarouge	Fosso	Igor
LoraWAN	Aghetseh	Kerone
Servomoteur MG996R	Amandine	Josué
RFID/Camera	Alan	Fosso
LCD	Josué	Bilou
Capteur ultrason HC-SR04	Igor	Kerman
Test / Maintenance	Hermann	Dareen

TABLE 1 – Répartition des tâches entre les membres

9 Analyse des Risques et Mitigations

Le projet de parking intelligent vise à développer un système de gestion de parking innovant qui utilise des technologies avancées pour améliorer l'expérience utilisateur et optimiser l'utilisation des places de parking. Cependant, comme tout projet complexe, il comporte des risques qui doivent être identifiés et mitigés pour assurer son succès.

9.1 Risques Techniques

- **Problèmes de connectivité :** Les systèmes de parking intelligent reposent sur des technologies de communication telles que le Wi-Fi, le Bluetooth ou les réseaux cellulaires. Les problèmes de connectivité peuvent entraîner des dysfonctionnements du système.

- **Mitigation** : Mettre en place des systèmes de sauvegarde et de redondance pour assurer la continuité du service.
- **Conséquence** : Le système reste opérationnel même en cas de problème de connectivité, minimisant les pertes de données et interruptions de service.
- **Erreurs de détection** : Les capteurs et les caméras utilisés pour détecter les véhicules et les places de parking peuvent être sujets à des erreurs.
 - **Mitigation** : Calibration et maintenance régulières des capteurs et caméras.
 - **Conséquence** : Réduction des erreurs de détection, meilleure expérience utilisateur et réduction des pertes financières potentielles.
- **Problèmes de scalabilité** : Le système doit gérer un grand nombre de véhicules et de places.
 - **Mitigation** : Concevoir un système évolutif.
 - **Conséquence** : Gestion efficace de l'augmentation de la demande, satisfaction utilisateur améliorée et réduction des coûts de maintenance.

9.2 Risques Financiers

- **Coûts de développement** : Le développement peut être coûteux.
 - **Mitigation** : Établir un budget détaillé et suivre les coûts.
 - **Conséquence** : Respect du budget et réduction des risques financiers.
- **Coûts de maintenance** : Maintenance régulière nécessaire.
 - **Mitigation** : Plan de maintenance et budgétisation.
 - **Conséquence** : Réduction des coûts de réparation et prolongation de la durée de vie du système.
- **Risque de non-paiement** : Les utilisateurs peuvent refuser de payer.
 - **Mitigation** : Systèmes de paiement sécurisés et flexibles.
 - **Conséquence** : Amélioration des revenus et satisfaction des utilisateurs.

9.3 Risques Technologiques

- **Obsolescence technologique** : Les technologies peuvent devenir obsolètes rapidement.

- **Mitigation** : Choisir des technologies évolutives et mises à jour régulières.
- **Conséquence** : Système compétitif et à jour.
- **Problèmes de sécurité** : Vulnérabilité aux cyberattaques.
 - **Mitigation** : Chiffrement et authentification robustes.
 - **Conséquence** : Protection des données et prévention des pertes financières.
- **Problèmes d'intégration** : Intégration avec d'autres systèmes (paiement, gestion trafic).
 - **Mitigation** : Mise en place d'API.
 - **Conséquence** : Intégration fluide et réduction des coûts de maintenance.

9.4 Risques Opérationnels

- **Problèmes de déploiement** : Déploiement complexe.
 - **Mitigation** : Plan de déploiement détaillé et suivi.
 - **Conséquence** : Déploiement à temps, réduction des coûts et satisfaction utilisateur.
- **Problèmes de formation** : Les utilisateurs et administrateurs doivent être formés.
 - **Mitigation** : Programmes de formation.
 - **Conséquence** : Meilleure expérience utilisateur et réduction des erreurs.

9.5 Risques Stratégiques

- **Concurrence** : Marché concurrentiel.
 - **Mitigation** : Stratégie de différenciation.
 - **Conséquence** : Parts de marché et revenus améliorés.
- **Évolution des besoins des utilisateurs** : Besoins utilisateurs évolutifs.
 - **Mitigation** : Recueil des commentaires et adaptation du système.
 - **Conséquence** : Satisfaction utilisateur et réduction du risque de perte de clients.

9.6 Risques Sécuritaires

- **Sécurité des utilisateurs** : Les utilisateurs doivent se sentir en sécurité.
 - **Mitigation** : Caméras de surveillance et systèmes d'alarme.
 - **Conséquence** : Meilleure expérience utilisateur et réduction des risques de sécurité.
- **Protection des données** : Données personnelles à protéger.
 - **Mitigation** : Chiffrement et authentification.
 - **Conséquence** : Protection des données et prévention des pertes financières.

9.7 Risques Logistiques

- **Problèmes de maintenance** : Maintenance régulière nécessaire.
 - **Mitigation** : Plan de maintenance et budgétisation.
 - **Conséquence** : Réduction des coûts et amélioration de la durée de vie du système.
- **Problèmes de gestion des déchets** : Déchets générés par le système.
 - **Mitigation** : Processus pour gérer les déchets de manière responsable.
 - **Conséquence** : Réduction de l'impact environnemental et amélioration de l'image de l'entreprise.

En identifiant et en mitigeant ces risques, nous pouvons minimiser les impacts négatifs et assurer le succès du projet de parking intelligent. Il est essentiel de suivre et de mettre à jour régulièrement l'analyse des risques pour garantir que le projet reste sur la bonne voie.

Conclusion

Le projet de système de parking intelligent a permis de concevoir et de réaliser une solution complète alliant matériel, logiciels et services connectés pour optimiser la gestion des places de stationnement. Grâce à une approche progressive et structurée, chaque phase a été menée avec des objectifs précis, allant de l'étude et la conception initiale, à la mise en place du prototype matériel, jusqu'à l'intégration des interfaces web et mobiles, la communication IoT, le paiement automatisé et l'identification des véhicules.

L'utilisation d'une stack technologique moderne basée sur **React/React Native, Node.js, IoT (ESP32, LoRa)** et des solutions cloud a permis de garantir une interopérabilité efficace entre les différents modules. Les choix techniques ont également pris en compte la scalabilité, la sécurité et la fiabilité du système, tout en offrant une expérience utilisateur fluide et intuitive.

L'analyse des risques réalisée tout au long du projet a permis d'identifier les enjeux techniques, financiers, opérationnels et sécuritaires, et de mettre en place des stratégies de mitigation adaptées. Cette démarche a contribué à réduire les imprévus, à assurer la continuité du service et à optimiser la performance globale du système.

Enfin, le prototype final, incluant le paiement automatisé, l'identification RFID, le suivi des places et l'interface de gestion, constitue une base solide pour un déploiement réel et évolutif. Ce projet illustre l'importance de l'intégration cohérente entre le matériel et les logiciels, et fournit un cadre méthodologique applicable à d'autres systèmes connectés, tout en répondant aux besoins réels des utilisateurs et des gestionnaires de parkings.

10 References

- “MTNMoMo Open API– Sandbox User Provisioning,” *MTN Developer Portal*. Available : https://momodeveloper.mtn.com/API-collections#api=sandbox-provisioning-apioperation=post-v1_0 – apiuser
- “Unleashing the Power of the MTNMoMo Open API,” RaymondZian, *Medium*, 2023. Available : <https://medium.com/>
- A.AlMamun, A.Hasib, A.S.M.Mussa, R.Hossen, A.Rahman, “IoT-Enabled Smart Car Parking System through Integrated Sensors and Mobile Applications.” (In press)
- C.Biyik, Z.Allam, G.Pieri, D.Moroni, etal., “Smart Parking Systems : Reviewing the Literature, Architecture and Ways Forward.” *Journal of IoT Applications*, vol.4, no.2, 2023.
- S.Y.C.Hong, C.C.Kang, J.D.Tan, M.Ariannejad, “Smart Parking System Using IoT Sensors : prototype with ESP8266+NodeMCU, Cloud Data.” *International Journal of Embedded Systems*, 2022.
- H.Kaur J.Malhotra, “A Review of Smart Parking System based on Internet of Things.” *IJISAE*, vol.8, no.3, 2022.
- <https://www.youtube.com/shorts/Q5ibWQxTn5A>
- https://www.youtube.com/watch?v=6gccSyp_uJQ