

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid- Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option : Réseaux et Systèmes Distribués (R.S.D)

Thème

**Réingénierie d'un parking intelligent en
utilisant la découverte des services IoT pour
l'université Abou Bakr Belkaid -Tlemcen-**

Réalisé par :

- M. RADI Badreddine
- M. SALAH Hichem

Présenté le 04 Juillet 2022 devant le jury composé de :

- | | |
|------------------------------------|------------------|
| • M. MANA Mohammed | (Président) |
| • M. MERZOUG Mohamed | (Encadreur) |
| • M. BENAHMED Hamed | (Co - Encadreur) |
| • M. SETTOUTI Ahmed Khalid Yassine | (Examineur) |

Remerciements

Avant tout, nous tenons à remercier ALLAH le plus miséricordieux, pour son aide et sa générosité.

Par la suite, nous aimerions remercier nos encadrants, M. MERZOUG Mohamed et M. BENAHMED Hamed, pour leurs encadrements, leurs soins constants et leurs discussions d'orientation.

Nous remercions également les membres du jury pour leur temps et leur dévouement.

Nous en profitons pour remercier nos amis : Houari, Rida, Nadir, Mohammed et Sidi Mohammed pour leur aide matérielle, scientifique et morale, sans qui nous ne serions pas en mesure d'exécuter ce travail.

Enfin, nos gratitudeles totales à nos familles pour leur amour, leur aide et leur soutien continus et inégalés. Nous sommes reconnaissants à nos parents pour leur soutien et leurs encouragements en tout temps.

ملخص:

تلعب إنترنت الأشياء دورًا أساسيًا في ربط العناصر البيئية بالشبكة، مما يسهل الوصول إلى هذه الأشياء غير المتصلة من أي موقع. من المرجح أن يقوم الناس بتحديث تقنية متنامية. يواجه الناس مشاكل في وقوف السيارات في مواقف السيارات.

الغرض من عملنا هو إنشاء نموذج أولي لمواقف السيارات الذكية في جامعة أبو بكر بلقايد - تلمسان - لمساعدة المستخدمين على مشاهدة أماكن وقوف السيارات المتاحة باستخدام نهج اكتشاف خدمة إنترنت الأشياء، عبر تطبيق الهاتف المحمول (Android) الذي يسمح لك بعرض أكثر مواقف السيارات صلة في الوقت الفعلي بعد طلب المستخدم.

كلمات مفتاحية: إنترنت الأشياء، تطبيق الهاتف، موقف سيارات ذكي، أردوينو، أندرويد.

Résumé :

L'internet des Objets joue un rôle essentiel dans la connexion des éléments environnementaux au réseau, facilitant l'accès à ces objets non connectés depuis n'importe quel endroit. Les gens sont susceptibles de mettre à jour une technologie en pleine croissance. Les gens affrontent des problèmes de stationnement dans les parkings.

Le but de notre travail est de réaliser un prototype de parking intelligent à l'Université Abu Bakr Belkaid - Tlemcen - pour aider les utilisateurs à consulter les places de parking disponibles en utilisant l'approche de découverte des service IoT, via une application mobile (Android) qui permet d'afficher les positions de stationnement les plus pertinente en temps réel suite à la requête utilisateur.

Mots-clés : IoT, application mobile, smart parking, Arduino, Android.

Abstract:

The Internet of Things plays a key role in connecting environmental elements to the network, making it easy to access these unconnected objects from anywhere. People are likely to upgrade a growing technology. People are facing parking problems in parking lots.

The aim of our work is to realize a prototype of smart parking at the University Abu Bakr Belkaid - Tlemcen - to help users to consult the available parking spaces using the IoT service discovery approach, via a mobile application (Android) that displays the most relevant parking positions in real time following the user request.

Keywords: IoT, Mobile app, smart parking, Arduino, Android.

Table des matières

Remerciements

Résumé

Liste des figures **I**

Liste des tableaux **IV**

Liste d'abréviations **V**

Introduction générale **1**

Chapitre I: Généralités sur l'internet des objets et la découverte de services **3**

I.1. Internet des objets **3**

I.1.1. Introduction 3

I.1.2. Définition de l'internet des objets 3

I.1.3. Architecture de l'internet des objets 4

I.1.4. Composants d'Internet des objets 5

I.1.5. Domaines d'application 6

I.1.6. Caractéristiques d'Internet des objets 7

I.1.7. Avantages et inconvénients d'Internet des objets 8

I.2. La découverte des services **9**

I.2.1. Introduction 9

I.2.2. La découverte des services IoT 9

I.2.3. Différences entre les services IoT et les service Web Traditionnels ____ 10

I.2.4. Défis et exigences de la découverte des services IoT 10

I.2.5. Différentes Approches de la découverte des services IoT 11

I.3. Conclusion **11**

Chapitre II: Généralités sur le parking intelligent **13**

II.1. Introduction **13**

II.2. Parking traditionnel **13**

II.2.1. Définition 13

II.2.2. Problématique de parking traditionnel 13

II.3. Parking intelligent : **14**

II.3.1. Définition 14

II.3.2.	Fonctionnement du système de parking intelligent	14
II.3.3.	Capteurs de parking intelligent	15
II.3.4.	Avantages et désavantages de parking intelligent	18
II.4.	Conclusion :	19
Chapitre III:	Conception et réalisation de projet	21
III.1.	Introduction	21
III.2.	Architecture proposée	21
III.2.1.	Couche physique et réseau	22
III.2.2.	Couche middleware	22
III.2.3.	Couche applicative	22
III.3.	Les matériaux nécessaires (hardwares)	23
III.3.1.	Les composants électroniques utilisés	23
III.3.2.	Arduino Mega2560	23
III.3.2.1.	Description de Mega2560	23
III.3.3.	NodeMcu/ESP8266	24
III.3.3.1.	Description d'ESP8266	24
III.3.4.	Servomoteur SG90	25
III.3.4.1.	Description de Servomoteur	25
III.3.4.2.	Fonctionnement de servomoteur	25
III.3.4.3.	Câblage et programmation de servomoteur	26
III.3.5.	Capteur IR	26
III.3.5.1.	Description de capteur IR	26
III.3.5.2.	Fonctionnement de capteur IR	27
III.3.5.3.	Câblage et programmation de capteur IR	28
III.3.6.	RFID RC-522	28
III.3.6.1.	Description de RFID RC-522	28
III.3.6.2.	Fonctionnement de RFID	28
III.3.6.3.	Câblage et programmation de RFID RC-522	29
III.3.7.	Clavier matricielle 3*4	29
III.3.7.1.	Description de clavier matricielle	29
III.3.7.2.	Câblage et programmation de clavier	30
III.4.	Softwares applications	30
III.4.1.	Logiciel PROTEUS	30
III.4.2.	Arduino IDE	31
III.4.3.	MIT App Inventor	32
III.4.4.	Firebase	33
III.4.4.1.	Service de Firebase	33

III.5.	Prototype de projet	34
III.5.1.	Architecture de prototype	34
III.5.2.	Description de prototype	35
III.5.3.	Implémentation de prototype	36
III.5.3.1.	Implémentation de parking visiteurs	37
III.5.3.2.	Implémentation de parking de département 1	37
III.5.3.3.	Implémentation de parking de département 2	38
III.5.3.4.	Implémentation de l'entrée et la sortie principale	39
III.6.	Interconnexion entre les différents modules de système	39
III.7.	Application mobile	40
III.7.1.	Analyse de besoins	40
III.7.1.1.	Besoins fonctionnels	40
III.7.1.2.	Besoins non-fonctionnels	41
III.7.2.	Modélisation	41
III.7.2.1.	Diagramme de cas d'utilisation	41
III.7.2.2.	Diagramme d'activité	42
III.7.2.3.	Diagramme de classe	43
III.7.3.	Présentation de l'application	43
III.7.3.1.	Interface visiteur	44
III.7.3.2.	Interface associe	45
III.8.	Scénarios d'utilisation	47
III.8.1.	Initialisation et authentification	47
III.8.2.	Découverte	48
III.8.3.	Mise à jour de la base Firebase	53
Conclusion générale		54
Références Bibliographiques		56
Annexe		60

Liste des figures

Figure I.1: Internet of Things.	3
Figure I.2: Principaux éléments fondamentaux de l'IoT.	4
Figure I.3: Principaux composants d'un système IoT.	5
Figure I.4: Applications de l'IoT.	7
Figure I.5: Boucle de découverte dans le monde IoT.	9
Figure II.1: Smart parking.	14
Figure II.2: Fonctionnement du système de smart parking.	15
Figure II.3: Capteur PIR.	15
Figure II.4: Capteur IR actif.	16
Figure II.5: Capteur ultrasonique.	16
Figure II.6: Capteur d'identification par radiofréquence (RFID).	17
Figure II.7: Capteur sous-sol (Magnétomètre).	17
Figure III.1: L'architecture proposée.	21
Figure III.2: Carte Arduino Mega2560.	24
Figure III.3: NodeMCU/ESP8266.	24
Figure III.4: Servomoteur SG90.	25
Figure III.5: Fonctionnement de servomoteur.	26
Figure III.6: Câblage et la programmation de Servomoteur SG90 par la plateforme TinkerCad.	26
Figure III.7: Capteur IR	27
Figure III.8: Principe de fonctionnement d'un capteur IR.	27
Figure III.9: Câblage et la programmation de capteur IR par la plateforme Wokwi.	28
Figure III.10: Fonctionnement de RFID.	29
Figure III.11: Câblage de RFID-RC522 par le logiciel fritzing.	29
Figure III.12: Clavier matricielle 3*4.	30
Figure III.13: Le câblage et la programmation de Keypad par la plateforme TinkerCad.	30
Figure III.14: Interface de logiciel PROTEUS.	31
Figure III.15: Interface de Arduino IDE.	32
Figure III.16: Fonctionnement d'un programme Arduino.	32
Figure III.17: Interface de MIT App Inventor.	33
Figure III.18: Logo de Firebase.	33
Figure III.19: Service de Firebase.	34

Figure III.20: Architecture des parkings sur logicielle LibreCAD.	35
Figure III.21: Schéma générale de branchement du système prototype.	35
Figure III.22: Représentation schématique du prototype.	36
Figure III.23: Implémentation des composants de prototype sur le logiciel PROTEUS.	36
Figure III.24: Le branchement de composants de parking visiteur.	37
Figure III.25: Le branchement de composants de parking de département 1.	38
Figure III.26: Le branchement de composants de parking de département 2.	38
Figure III.27: Le branchement de composants de l'entrée et de sortie principale.	39
Figure III.28: Création des capteurs IR sur Firebase.	39
Figure III.29: Schéma représente la communication entre les différents modules de système.	40
Figure III.30: Diagramme de cas d'utilisation global.	42
Figure III.31: Diagramme d'activité global.	42
Figure III.32: Diagramme de classe globale.	43
Figure III.33: 1 ^{er} interface de le lancement de l'application.	43
Figure III.34: L'interface de choix le type d'utilisateur.	44
Figure III.35: Interface de parking visiteur.	45
Figure III.36: Interface de parking visiteur plein.	45
Figure III.37: Page d'authentification.	45
Figure III.38: Page d'inscription	46
Figure III.39: Page de sélection.	46
Figure III.40: Réponse à la requête.	46
Figure III.41: Échange des messages entre l'application et BD en cas d'authentification.	47
Figure III.42: Échange des messages entre l'application et BD en cas de création de compte.	47
Figure III.43: Echange d'information entre PMS et l'application (Visiteur).	48
Figure III.44: L'application envoi une requête à PMS.	49
Figure III.45: PMS répond à la requête de l'application (parking dep1).	49
Figure III.46: Parking dep 1.	49
Figure III.47: PMS répond à la requête de l'application (parking dep 2).	50
Figure III.48: Parking dep 2.	50
Figure III.49: PMS répond à la requête de l'application (Parking visitor).	50
Figure III.50: Parking visiteur.	50
Figure III.51: L'application envoi une requête à PMS.	51
Figure III.52: PMS répond à la requête de l'application (parking dep2).	51
Figure III.53: Parking dep 2.	51

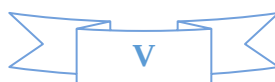
<i>Figure III.54: PMS répond à la requête de l'application (parking dep 1).</i>	52
<i>Figure III.55: Parking dep 1</i>	52
<i>Figure III.56: PMS répond à la requête de l'application (parking visitor).</i>	52
<i>Figure III.57: Parking visitor</i>	52
<i>Figure III.58: PMS répond à la requête de l'application (NULL).</i>	53
<i>Figure III.59: Plein</i>	53
<i>Figure III.60: La requête de contrôleur à PMS.</i>	53

Liste des tableaux :

<i>Tableau I.1: Domaine d'application d'IoT.</i>	6
<i>Tableau I.2: Services IoT et Services Web traditionnel.</i>	10
<i>Tableau I.3: Exigences et les défis de la découverte de services IoT.</i>	10
<i>Tableau III.1: Composants électroniques utilisés.</i>	23

Liste d'abréviations :

IoT	I nternet o f T hings
Wi-Fi	W ireless F idelity
IA	I ntelligence A rtificielle
IR	I nfrared
PIR	P assive I nfrared
RFID	R adio- F requency I dentification
ICSP	I n- C ircuit S erial P rogramming
I2C	I nter- I ntegrated C ircuit
SoC	S ystem o n a C hip
IDE	I ntegrated D evelopment E nvironment
SDK	S oftware D evelopment K it
SPI	S erial P eripheral I nterface
GPIO	G eneral- p urpose I nterface O utput
UART	U niversal A synchronous r eciever- t ransmitter
MIT	M assachusetts I nstitute of T echnologies
PMS	P arking M anagement S erver
RAD	R apid A pplication D evelopment
WYSIWYG	W hat Y ou S ee I s W hat Y ou G et
PCB	P rinted C ircuit B oard



Introduction générale

La croissance rapide de la population urbaine a créé l'augmentation du nombre de véhicules dans le monde. Cela peut provoquer des problèmes d'encombrement. L'un des problèmes les plus fréquents en ville est le parking. Le stationnement est un sujet d'actualité, car il affecte la mobilité de la plupart des gens. Le conducteur doit toujours s'arrêter pour atteindre sa destination. Pour ce faire, nous avons besoin d'un système alternatif, qui peut aider à gérer et à réduire le trafic routier. Le système assiste les gens à gagner du temps lors de la recherche d'une place de parking. Le besoin de ces options a conduit à l'émergence d'un terme moderne appelé « Smart Parking ».

Le parking intelligent est considéré comme l'installation d'appareils et de systèmes connectés pour collecter des données à l'aide de capteurs, ainsi que d'autres objets physiques connectés à Internet pour assurer le suivi et la gestion, et même des systèmes qui permettent aux conducteurs de réserver des places de stationnement.

En effet, notre projet comprend la réingénierie d'un parking intelligent de l'université de Tlemcen (UABT) en utilisant l'approche « Service Discovery » de l'internet des objets. L'IoT nous offre la possibilité de développer des applications et d'élaborer des systèmes connectés utilisant des capteurs, des actionneurs et d'autres objets physiques (lié via des protocoles de communication); permettant l'interconnexion des données et l'échange des informations.

Notre projet de fin d'études s'organise autour de trois chapitres :

- **Le premier chapitre :** fait l'objet d'une généralité sur l'internet des objets et la découverte des services.
- **Le deuxième chapitre :** concerne à la définition des parkings et de leurs problèmes. On a également parlé de la solution qui est le smart parking et leurs capteurs.
- **Le troisième chapitre :** divisée en deux parties, la première partie consacrée la description des composants et des logiciels utilisés pour la réalisation du prototype. Par la suite, nous allons passer à la réalisation du prototype à côté de l'application Android.

Chapitre | 1

*GÉNÉRALITÉS SUR L'INTERNET DES
OBJETS ET LA DÉCOUVERTE DE SERVICES*

Chapitre I : Généralités sur l'internet des objets et la découverte de services

lors, le terme a été utilisé pour décrire tout appareil qui interagit avec le monde physique qui l'entoure, qu'il s'agisse de collecter des données à partir de capteurs ou d'alimenter le monde réel via des actionneurs interactifs, généralement connectés à d'autres appareils ou à Internet. [1]

Internet des objets représente un monde où tout ce qui nous entoure, comme les téléphones, les réfrigérateurs, les lampes, etc. surveillent, détectent et interagissent les uns avec les autres avec ou sans contrôle humain. Beaucoup voient l'IoT comme la prochaine révolution technologique d'Internet.

IoT est un réseau de systèmes embarqués (dispositifs) dotés d'une connectivité internet, leur permettant de se connecter et d'interagir à grande échelle avec d'autres appareils, services et personnes embarqués. [3]

I.1.3. Architecture de l'internet des objets :

Il n'existe pas d'architecture IoT standard. Mais quel que soit le cas d'utilisation et le nombre de couches, les éléments essentiels constitutifs de toute structure IoT sont toujours les mêmes, à savoir : [4]

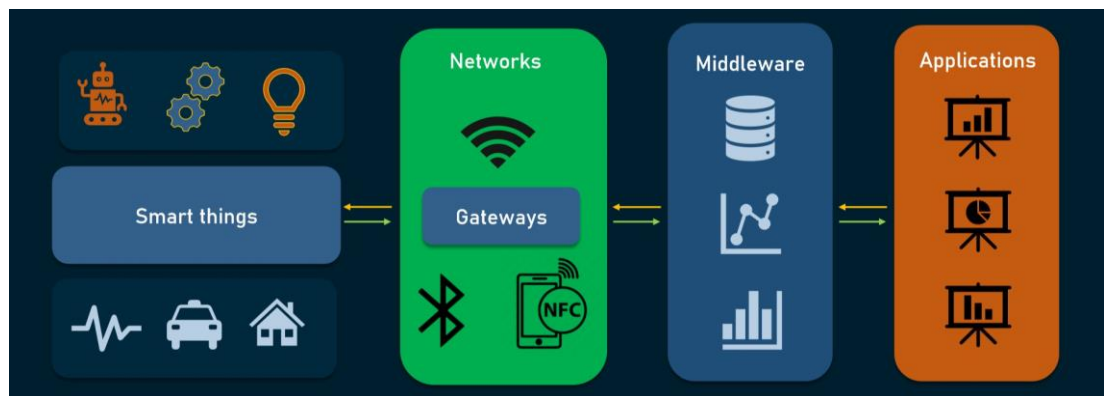


Figure I.2: Principaux éléments fondamentaux de l'IoT. [4]

- 1. Smart things :** il se compose de capteurs qui collectent des données de l'environnement, d'un réseau qui collecte et transmet les données des capteurs ; et d'actionneurs utilisés pour aborder les objectifs fixés par les données des capteurs.
- 2. Réseaux et passerelles :** permettre aux appareils à consommation réduite (scénario courant dans l'IoT) d'accéder à l'internet plus large.

3. Middleware (plateformes IoT) : fournis un espace de stockage de données et des capacités avancées de calcul et de moteur d'analyse.

4. Applications : permettre aux utilisateurs de bénéficier de l'IoT et de manipuler le monde physique.

Ces éléments constituent l'épine dorsale de tout système IoT sur lequel une architecture multicouche efficace peut être développée. Généralement, ces couches sont :

- **Couche de perception :** (couche de monde physique) héberge des objets intelligents.
- **Couche de connectivité :** (couche de transport) qui transportent les données de la couche physique vers le cloud et vice-versa via des réseaux et des passerelles.
- **Couche de traitement** utilise une plateforme IoT pour accumuler et gérer tous les flux de données.
- **Couche applicative** qui fournit aux utilisateurs des solutions telles que l'analyse, la création de rapports et le contrôle des appareils.

I.1.4. Composants d'Internet des objets :

Dans un système IoT, il existe quatre composants de base, qui sont : [5]

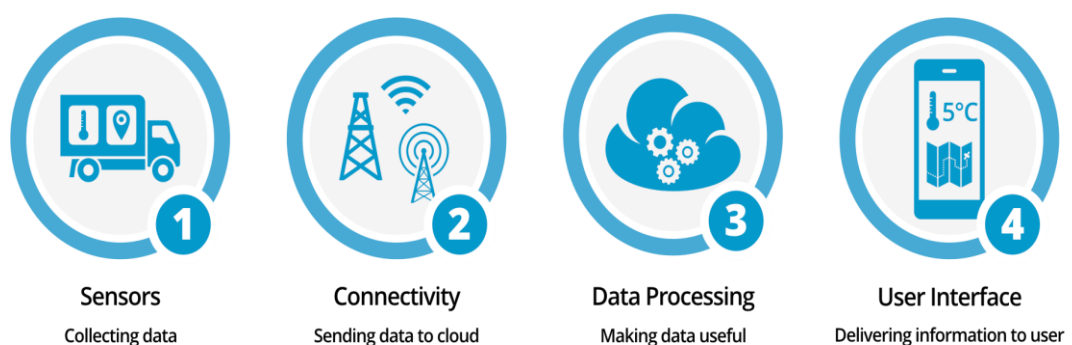


Figure I.3: Principaux composants d'un système IoT. [7]

- 1. Capteurs :** les capteurs peuvent collecter des données environnementales très précises. Toutes ces données collectées, peuvent obtenir différents degrés de complexité. Un appareil peut posséder plusieurs capteurs qui peuvent être

Chapitre I : Généralités sur l'internet des objets et la découverte de services

combinés pour faire plus que simplement détecter des choses (les smartphones sont obsolètes avec plusieurs capteurs, tels que le GPS, l'accéléromètre et l'appareil photo.).

2. **Réseaux** : les données collectées (par des capteurs) sont envoyées dans le cloud, mais il faut un moyen de transport. Les capteurs peuvent être connectés au cloud via diverses méthodes de communication et de transmission telles que le réseau cellulaire, le réseau satellitaire, le Wi-Fi, le Bluetooth, le WAN....
3. **Traitement de données** : une fois que les données collectées atteignent le cloud, le programme traitera les données obtenues. Cela peut être aussi simple que de vérifier que les lectures de température sur des appareils tels que les climatiseurs ou les radiateurs sont dans des limites acceptables. Parfois, cela peut aussi être extrêmement complexe, comme utiliser la vision par ordinateur sur la vidéo pour identifier des objets (comme un intrus dans une maison).
4. **Interface utilisateur** : les informations sont mises à la disposition des utilisateurs finaux d'une manière ou d'une autre. Cela peut être fait en réglant une alarme sur leur téléphone ou en les notifiant par SMS ou par mails.

I.1.5. Domaines d'application :

L'IoT aura un impact sur de nombreux aspects de la vie quotidienne. Voici quelques-uns des secteurs fondamentaux : [2]

Domaine	Utilisation
Maison	Contrôle du verrouillage des portes, de l'éclairage, de climatisations, etc.
Santé	Surveillance à distance des patients, etc.
Industrie	Contrôler et gérer des machines à grande échelle, etc.
Agriculture	Arrosage automatique, etc.
Smart cities	Éclairage public, gestion du trafic, suivi et disponibilité des places de stationnement.

Tableau I.1: Domaine d'application d'IoT.

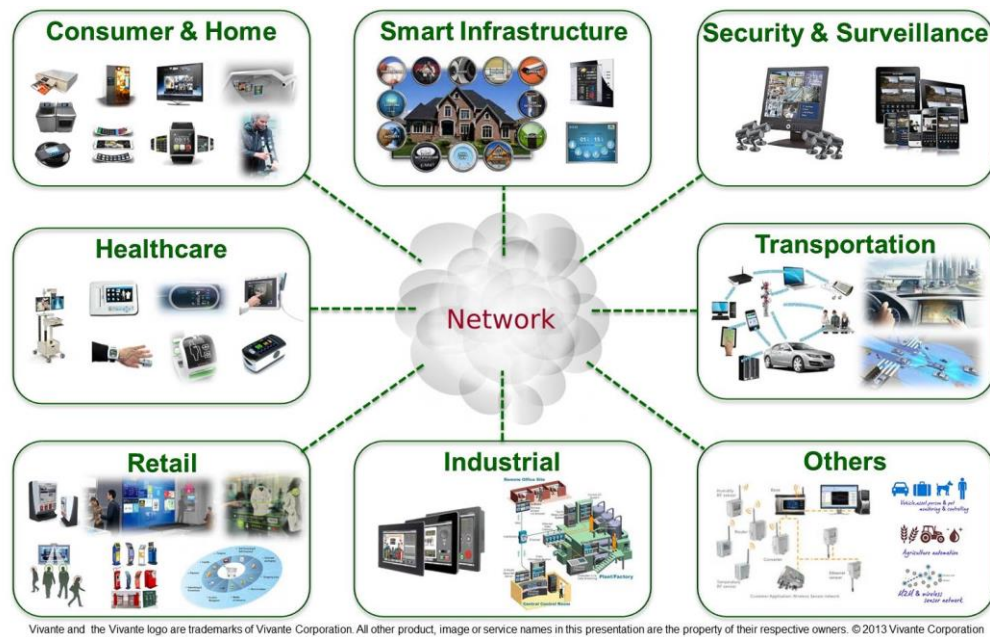


Figure I.4: Applications de l'IoT .[3]

I.1.6. Caractéristiques d'Internet des objets :

Les sept caractéristiques fondamentales de l'Internet of Things sont : [6]

1. **Connectivité** : permet l'accessibilité au réseau (fournis par le réseau) et la compatibilité (fournis des fonctionnalités communes pour l'utilisation et la génération de données).
2. **Objets** : tout ce qui peut être étiqueté ou connecté de la manière dont il est conçu pour être connecté. Des capteurs aux appareils électro-ménagers en passant par le bétail marqué. Les appareils peuvent contenir des capteurs ou des matériels de détection qui peut être attachée aux appareils et aux articles.
3. **Données** : les données sont le ciment de l'IoT, le premier pas vers l'action et l'intelligence.
4. **Communication** : connecter des appareils afin qu'ils puissent communiquer des données et analyser ces données. La communication peut être de courte portée ou de longue a grandement longue portée (Wi-Fi).
5. **Intelligence** : aspect de l'intelligence, tel que les capacités de détection des dispositifs IoT et l'intelligence tirée à partir de l'analyse des Big Data (IA).
6. **Action** : il peut s'agir d'actions manuelles, d'actions basées sur de discussions de phénomènes et d'automatisation, c'est généralement la partie essentielle.

7. **Écosystème** : là où l'Internet des Objets s'intègre dans d'autres technologies, communautés, objectifs IoT.

I.1.7. Avantages et inconvénients d'Internet des objets :

- **Avantages :**

- Il encourage la communication entre les dispositifs, ce qui permet d'obtenir une plus grande transparence avec moins d'efficacité et plus de qualités pour fournir les résultats finaux requis. [3]
- Grâce à la communication directe entre les appareils, sans intervention personnelle, l'automatisation constitue une réalité, favorisant une production rapide et ponctuelle. [3]
- Internet des Objets ouvre la porte à un vaste éventail d'informations, en particulier du monde réel, qui permettent une prise de décision rapide. [3]
- Avec l'aide de l'IoT, des données plus exactes et plus précises peuvent désormais être suivies grâce à une surveillance efficace, permettant des réponses instantanées basées sur des paramètres surveillés. [3]
- L'IoT fournit davantage d'informations du monde réel, pour une gestion efficace des ressources. [3]

- **Inconvénients :**

- **Compatibilité** : à l'heure actuelle, il n'existe pas de norme de compatibilité unifiée entre les appareils IoT, et les opérations ne peuvent être effectuées que selon des normes communes. [3]
- **Sécurité** : avec un écosystème d'appareils fréquemment connectés communiquant sur le réseau, les données sont plus vulnérables aux intrus ou aux pirates. [3]
- **Complexité** : l'utilisation de plusieurs technologies et de la diversité dans l'IoT est complexe en raison de la complexité de la conception, du déploiement et de la maintenance. [3]

I.2. La découverte des services :

I.2.1. Introduction :

Avec l'émergence de l'internet des objets, des données riches sur les objets connectés et des événements du monde réel seront générés en grande quantité et stockées dans des systèmes d'information hétérogènes largement distribués. [13] [8]

En raison du nombre excessif d'objets connectés, rechercher et découvrir des services spécifiques offerts par ces objets devient une tâche difficile [12].

Le principal défi est de savoir comment trouver le service requis dans l'IoT en respectant la grande quantité, la mobilité, l'hétérogénéité et la large distribution des services déployés dans des appareils contraints et en même temps souhaitable pour les utilisateurs [10].

I.2.2. La découverte des services IoT :

La découverte des services IoT représente un outil indispensable pour permettre aux clients tels que les plates-formes middleware, les utilisateurs finaux et les applications de localiser les ressources disponibles pour l'IoT (Capteurs, actionneurs, services, données de contexte), répondant aux changements de l'environnement IoT et à la situation de l'utilisateur.

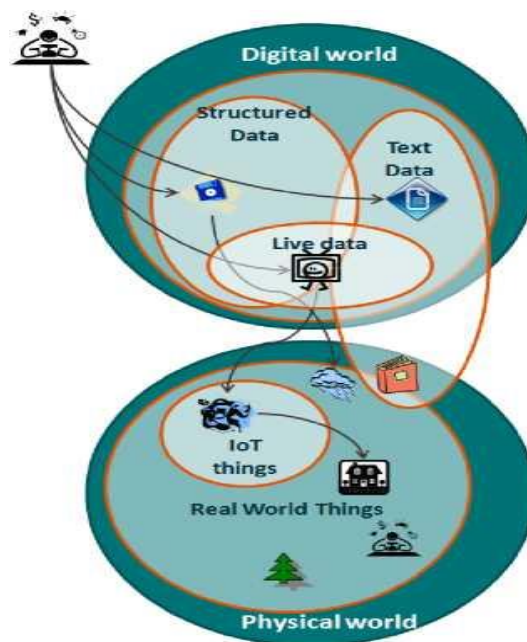


Figure I.5: Boucle de découverte dans le monde IoT. [8]

Chapitre I : Généralités sur l'internet des objets et la découverte de services

I.2.3. Différences entre les services IoT et les service Web Traditionnels :

Même si la découverte des services représente un sujet bien étudié dans les systèmes distribués, les approches traditionnelles ne sont pas adaptées à l'IoT, à cause des différences entre les services du monde réel et les services Web traditionnels [10] [12] :

	Services IoT	Services Web Traditionnel
Ressources	<u>Limitées</u> : capacité du calcul, bande passante de réseau, batterie et capacité de stockage...	<u>Riches</u> : puissance de calcul, capacité énorme de stockage, Bande passante importante...
Environnements	<u>Très dynamiques</u> : déconnexion intermittente du réseau sans fil et la mobilité des appareils	<u>Très stables</u> : connexion câblé, dispositifs de stockage stables et sécurisés
Etat d'objets	<u>Quasi temps réel</u> (Near real-time object state) ,	<u>Entièrement virtuelles.</u>

Tableau I.2: Services IoT et Services Web traditionnel.

I.2.4. Défis et exigences de la découverte des services IoT :

La nature des objets et la dynamique de l'environnement IoT ; apporte à la découverte du service de nouveaux des défis et exigences tels que c'est décrit dans le tableau suivant [11] :

Exigences IoT	Défis IoT
Grande quantité	Fiabilité
Mobilité	Dynamacité
Hétérogénéité	Performance
Large distribution	Evolutivité
Appareils contraints	Interopérabilité
Sensibilité au contexte	Sécurité et Confiance

Tableau I.3: Exigences et les défis de la découverte de services IoT.

I.2.5. Différentes Approches de la découverte des services IoT :

En se basant sur les mécanismes de découvertes existants, les approches de découverte des services IoT peuvent être classifiées en principales catégories [10] :

- Approches sémantique.
- Approches basées sur le contexte.
- Approches basées sur la qualité de service.
- Approches bio-inspirées.
- Approches de recherche fédérale.

I.3. Conclusion :

En un mot, l'Internet des Objets peut représenter une bénédiction s'il est utilisé efficacement. Cela nous permet de faire le meilleur usage des informations obtenues. il améliore notre mode de vie et nos connaissances. Cependant, il peut aussi être critiqués s'il est mal utilisé.

Chapitre | 2

*GÉNÉRALITÉS SUR LE PARKING
INTELLIGENT*

II.1. Introduction :

En raison de l'urbanisation croissante, l'utilisation de véhicules automobiles a augmenté, ce qui provoque des problèmes de circulation et de stationnement, où la plupart des encombrements urbains sont causés par les propriétaires de véhicules qui cherchent à se garer.

En raison du nombre remarquable de voitures, de l'étalement urbain et de l'insuffisance des places de stationnement, entraînant la propagation du chaos dans les grandes villes, nous avons conduit à la spéculation et à la création d'autres parkings. Cette confusion s'est déplacée vers les universités où les individus ne peuvent pas trouver de places de stationnement, en particulier les enseignants, le personnel et les étudiants, en raison du stationnement aléatoire, prenons l'exemple de l'Université de Tlemcen, vous observez des voitures garées devant l'entrée de l'université, à l'intérieur du campus aussi chaotique que c'est à l'extérieur, les gens perdent beaucoup de temps à chercher une place de parking convenable. Malgré les grands efforts déployés pour fournir des places de stationnement et des parkings. Par conséquent, les parkings intelligents doivent être généralisés pour éviter ce problème.

II.2. Parking traditionnel :

II.2.1. Définition :

Le parking est une espace permettant de laisser temporairement les véhicules, généralement dans un parking ou sur le bord de la route. [14]

Le stationnement est l'action d'arrêter et de débrayer un véhicule et de le laisser inoccupé. Le stationnement sur un côté ou sur les deux côtés d'une route est souvent autorisé, mais parfois avec des restrictions. Certains bâtiments disposent de parkings à l'usage des utilisateurs du bâtiment. [15]

II.2.2. Problématique de parking traditionnel :

- Les places de stationnement sont insuffisantes pour toutes les voitures.
- Le stationnement interdit au hasard rend la mobilité difficile.
- La recherche continue de stationnement augmente le trafic et la pollution.

Chapitre II : Généralités sur le parking intelligent

- Problèmes de sécurité (vols des voitures, ...).

II.3. Parking intelligent :

II.3.1. Définition :

Le parking intelligent est une solution de stationnement qui peut inclure des capteurs, des caméras ou des capteurs de comptage. Ces dispositifs sont généralement intégrés ou placés à côté d'une place de parking pour détecter si une place de parking est libre ou occupée. Cela se fait en collectant des données en temps réel. Les données sont ensuite transmises à une application mobile ou à un site web de parking intelligent, qui communique la disponibilité à l'utilisateur. [16]



Figure II.1: Smart parking.

II.3.2. Fonctionnement du système de parking intelligent :

Les systèmes de stationnement sont installés à l'extérieur ou à l'intérieur des bâtiments. Lorsqu'un véhicule pénètre dans l'espace, des capteurs détectent sa présence et calculent les places de stationnement disponibles. Ces informations sont ensuite envoyées au téléphone du conducteur via une application.

Le système de stationnement intelligent dispose également de données en temps réel sur l'occupation, qui peut être trouvée sur l'application.

Les données sont collectées à partir de chaque capteur et mises à jour à chaque moment. [17]

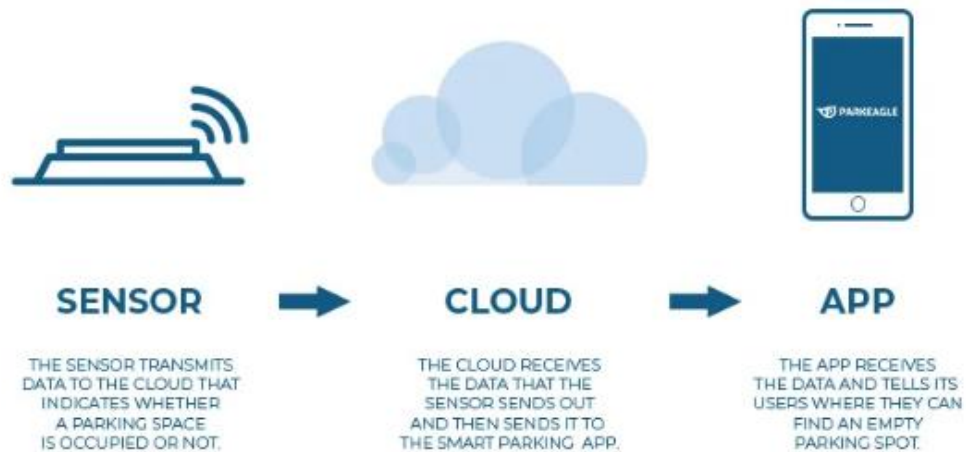


Figure II.2: Fonctionnement du système de smart parking. [13]

II.3.3. Capteurs de parking intelligent :

Il existe plusieurs capteurs qui facilitent la détection des informations relatives à l'occupation des places de stationnement. Chaque capteur a des caractéristiques et des avantages.

Les capteurs les plus largement adoptés sont les suivants :

- **Capteur infrarouge (IR)** : les capteurs infrarouges plus courants utilisent dans les smart parkings actuellement, il y a deux types :
 - **Capteur infrarouge passif (PIR)** : ces capteurs détectent les changements d'énergie thermique lorsqu'un véhicule occupe une place de stationnement ; ces capteurs conviennent aux parkings fermés situés à l'intérieur des bâtiments et ne conviennent pas aux parkings extérieurs ou ouverts. Les PIR doivent être placés sous le sol ou au plafond.

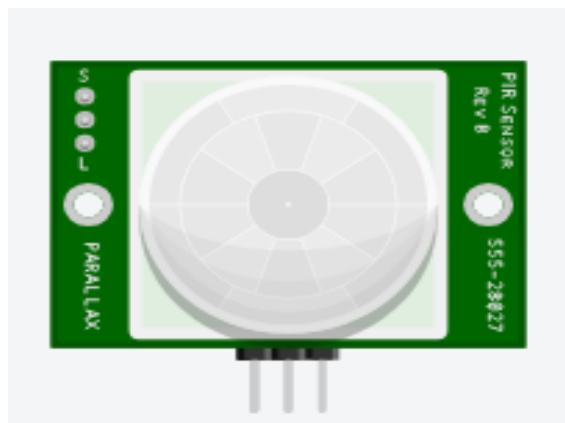


Figure II.3: Capteur PIR.

- **Capteur infrarouge actif** : ces capteurs émettent de l'énergie infrarouge et permettent de détecter tout objet ou véhicule par la quantité d'énergie thermique réfléchiée. Ils sont également sensibles au changement climatique (pluies, neige,). Le déploiement de capteurs dans toutes les places de stationnement permet d'obtenir le taux d'occupation du parking. Ces capteurs sont généralement placés en hauteur et conviennent aux parkings intérieurs fermés. [18]



Capteur IR

Figure II.4: Capteur IR actif.

- **Capteur ultrasonique** : ces capteurs émettent des ondes sonores entre 25 et 50 KHz et détectent les objets en fonction des rebonds sonores réfléchies. Ils sont généralement montés au plafond et sont sensibles aux changements environnementaux (pluies, neige,). En fonction de la distance à laquelle les ondes sont réfléchies, ils peuvent faire la distinction entre un véhicule et une personne. Afin d'obtenir l'état d'occupation du parking, ces capteurs doivent être placés au-dessus de chaque place de parking. [18]

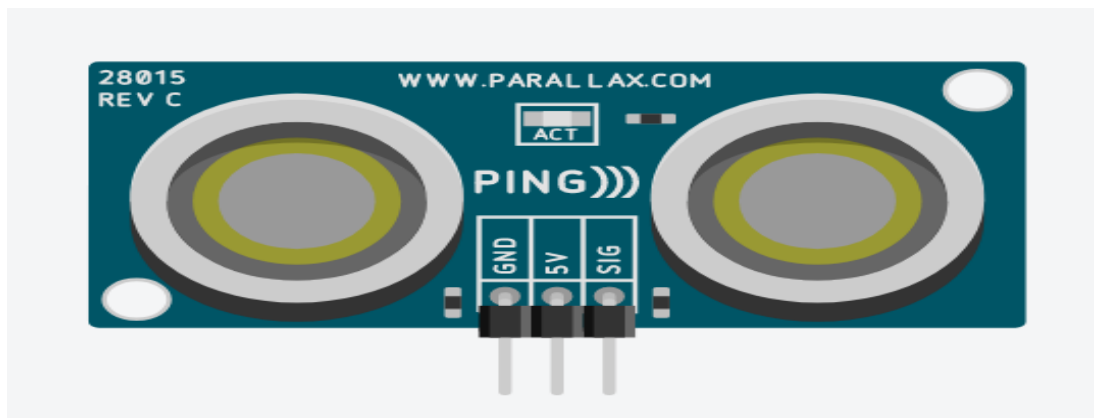


Figure II.5: Capteur ultrasonique.

Chapitre II : Généralités sur le parking intelligent

- **Identification par radiofréquence (RFID)** : chaque véhicule recevra une étiquette la radiofréquence pour identifier. Un émetteur-récepteur et une antenne sont installés à l'entrée du parking. Ces systèmes sont adoptés aux parkings fermés et couverts. Ils ne conviennent pas aux parkings ouverts, car ils sont librement disponibles. [18]



Figure II.6: Capteur d'identification par radiofréquence (RFID).

- **Magnétomètre** : ces capteurs détectent la présence d'un véhicule en détectant la variation du champ électromagnétique. Ils doivent être proches du véhicule, c'est pourquoi ils sont placés sous la surface. Ils conviennent aux parkings ouverts et fermés. Il existe des capteurs sans fil qui ont une autonomie de quelques années et qui permettent de détecter en temps réel l'occupation d'un parking. Les capteurs doivent être placés sous chaque place de stationnement pour connaître l'occupation des places. [18]



Figure II.7: Capteur sous-sol (Magnétomètre). [19]

- **Caméras** : les dispositifs de solution IoT de stationnement basés sur la caméra sont une approche technologique relativement nouvelle qui a évolué avec les progrès de la vision par ordinateur. Cette méthode de traitement d'image/vidéo nécessite un support de surveillance informatique pour identifier toutes les voitures dans un espace de stationnement. Alors que les caméras peuvent surveiller plusieurs places de stationnement, cette approche peut avoir des angles morts. De plus, le traitement de l'image ou la vidéo en temps réel nécessite un matériel de traitement Cloud de plateforme. Il s'agit d'une solution coûteuse étant donné que les caméras et les systèmes informatiques ne peuvent pas être utilisés de manière évolutive. Les solutions basées sur des caméras ne peuvent vraiment être considérées comme une approche viable que dans des champs ouverts relativement grands. [19]

II.3.4. Avantages et désavantages de parking intelligent :

- **Avantages** :
 - **Efficacité** : utilisation des logiciels et des smartphones, qui sont moins susceptibles de faire des erreurs. Ainsi, le niveau d'efficacité augmente et peut gérer les protocoles de stationnement sans problème. Éviter la longue file d'attente à l'entrée et aller directement à l'entrée du bâtiment où utiliser l'espace de stationnement par eux-mêmes à cause de protocoles de stationnement qui déterminent le moment où entrer dans le bâtiment sans délai. [21]
 - **Sécurité** : une fonction de barrière et de réservation permet de contrôler les véhicules qui ont accès à un lieu. Ainsi, il peut en verrouiller en certains moments, s'il est programmé pour le faire. Il y a aussi une caméra de surveillance qui contrôle les véhicules. [21]
 - **Environnement** : réduction des émissions de CO₂, du bruit et d'autres polluants.

- **Désavantages :**

- **Le coût élevé de la construction ou de l'installation :** Le coût d'un système de gestion du stationnement efficace et fonctionnel est généralement élevé. Cela s'explique par les divers éléments qui entrent dans le fonctionnement du système (la fonction statistique, la billetterie automatisée, les rapports statistiques, ...). [21]
- **Maintenance régulière :** Le système est automatisé, mais il nécessite néanmoins plusieurs maintenances régulières. Il s'agit de s'assurer que le système fonctionne parfaitement. La maintenance peut être effectuée une fois par mois. [21]

II.4. Conclusion :

Le parking intelligent joue un rôle important, car il contribue à réduire les embouteillages et de nombreux problèmes d'accessibilité urbaine. Dans ce chapitre, nous avons introduit le parking traditionnel et leurs problèmes, et nous parlons de parking intelligent et comment fonctionnons, et les capteurs utilisés.

Chapitre | 3

*CONCEPTION ET RÉALISATION DE
PROJET*

III.1. Introduction :

Le but de notre projet est de mettre en place un prototype de parking intelligent qui détermine l'emplacement idéal, en aidant les utilisateurs à trouver une place de parking de manière simple et facile sans perdre trop de temps.

Nous avons développé une application Android pour exécuter un modèle basé sur des requêtes d'un système de stationnement intelligent qui permet aux utilisateurs de découvrir les places de stationnement vacantes.

L'objectif principal du système proposé est de trouver des places de stationnement en utilisant l'Internet des Objets.

III.2. Architecture proposée :

Notre projet consiste à réaliser un prototype de parking intelligent en utilisant la technologie de l'internet des objets. Et pour cela ont proposé une architecture basée sur trois couches (**Figure III.1**) :

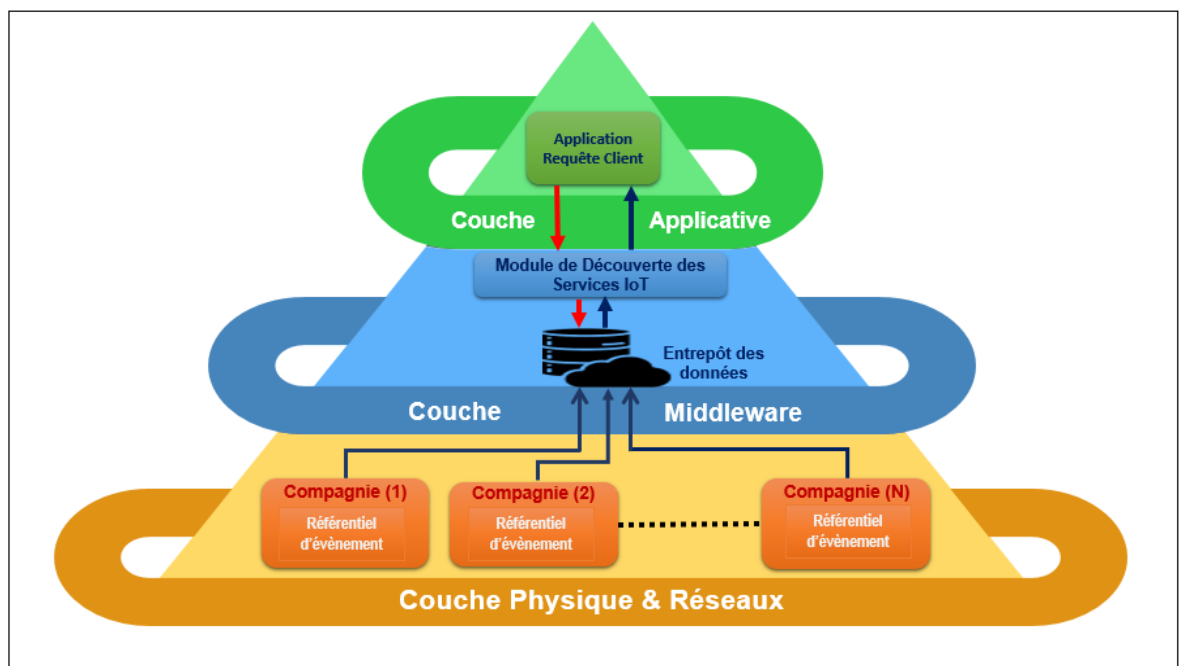


Figure III.1: L'architecture proposée.

III.2.1. Couche physique et réseau :

Dans cette couche on trouve des compagnies qui représente chacune l'ensemble des capteurs installé dans les positions de stationnement d'un parking lié à un établissement (département, administration, bibliothèque, ...), ainsi que les moyens réseaux nécessaire pour assurer la connexion des capteurs avec l'internet.

Chaque compagnie contient un référentiel d'événements qui stock les adresses des capteurs de la compagnie ainsi que les états des positions dont ils sont liés.

A chaque changement d'état (occupé, libre) un message est envoyé via la couche réseau vers la couche middleware.

III.2.2. Couche middleware :

C'est une couche intermédiaire qui assure permettre la communication entre le monde virtuel (Couche applicative) et le monde physique (Couche Physique et réseaux). Dans notre architecture cette tâche est assurée par deux modules :

- **Module entrepôt des données :** Il joue le rôle d'une base de données où ils sont stockés les états des positions de stationnement envoyés par la couche physique et réseau.
- **Module de découverte des services IoT :** Ce module représente le noyau de notre système car c'est lui qui répond à la requête client.

Pour répondre au requête client notre système utilise une approche de découverte des services par compagnie (découverte horizontale), en interrogeant l'état des positions liés au parking demandé par le client au niveau de l'entrepôt des données, et en envoyant au client un état de parking contenant les positions libres et occupées.

III.2.3. Couche applicative :

Cette comprend l'interface utilisateur de notre système représenté par une application de requête utilisateur (Application Android) qui permet aux clients de notre système de s'inscrire pour bénéficier des services offerts en temps réel.

Le client introduit sa requête en choisissant le parking destination, cette dernière sera envoyée vers le module de la découverte des services au niveau de la couche

Chapitre III : Conception et réalisation de projet

middleware, qui renvoi une réponse de l'état de parking qui va être présenté au client par l'application sous forme graphique.

III.3. Les matériaux nécessaires (hardwares) :

III.3.1. Les composants électroniques utilisés :

Pour notre prototype, nous nous sommes servis de plusieurs éléments.

Ces équipements sont indiqués au Tableau III.1:

Composant	Quantité
Arduino Méga2560	01
NodeMCU/ESP8266	01
Servomoteur SG90	05
Capteur IR	11
RFID-RC522	01
Clavier 3 * 4 touches Matrix	01
Buzzer	04
LED 5 mm	08
Bouton	02
Mini Breadboard (4.5X3.5CM)	02
Jumpers	320

Tableau III.1: Composants électroniques utilisés.

III.3.2. Arduino Mega2560 :

III.3.2.1. Description de Mega2560 :

L'Arduino Mega2560 est une carte microcontrôleur basée sur le microcontrôleur ATmega328P 8 bits. Outre l'ATmega328P, il se compose d'autres composants tels qu'un

Chapitre III : Conception et réalisation de projet

oscillateur à cristal, une communication série, un régulateur de tension, etc. le microcontrôleur est pris en charge. La carte Arduino Mega2560 possède 54 broches d'entrée-sortie numériques, 16 broches d'entrée analogiques, une connexion USB, une prise d'alimentation A, un connecteur ICSP et un bouton de réinitialisation. [22]

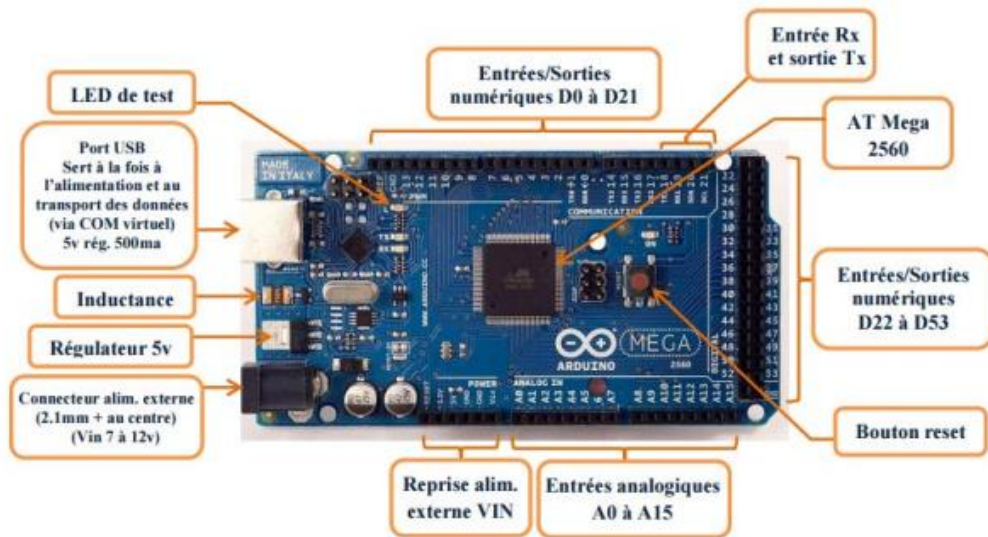


Figure III.2: Carte Arduino Mega2560. [22]

III.3.3. NodeMcu/ESP8266 :

III.3.3.1. Description d'ESP8266 :

NodeMCU est un environnement de développement logiciel et matériel open source construit autour d'un système sur puce (SoC) très économique appelé ESP8266. Conçu et fabriqué par Espressif Systems, l'ESP8266 contient tous les éléments clés d'un ordinateur moderne : CPU, RAM, réseau (sans-fil), et même un système d'exploitation et un SDK modernes. Le NodeMCU de ce prototype utilise un module de microcontrôleur avec un système Wi-Fi programmé à l'aide de l'IDE Arduino. [23]

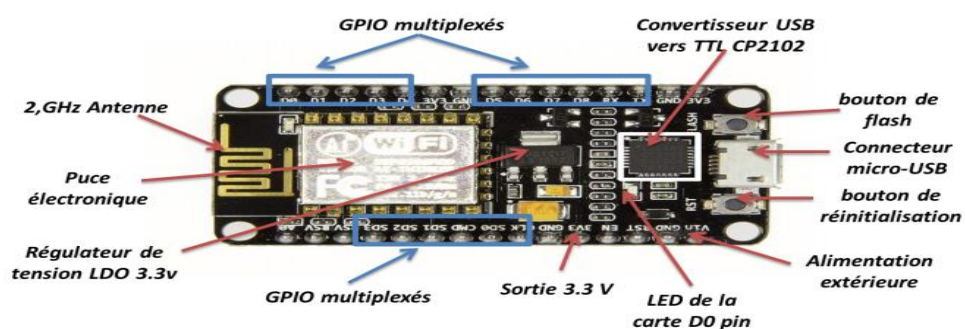


Figure III.3: NodeMCU/ESP8266.

III.3.4. Servomoteur SG90 :

III.3.4.1. Description de Servomoteur :

Un servomoteur est un actionneur rotatif ou linéaire qui permet un contrôle précis de la position angulaire ou linéaire, de la vitesse et de l'accélération.

Il se compose d'un moteur couplé à un capteur pour le retour de position. Il nécessite également un contrôleur relativement sophistiqué. [24]



Figure III.4: Servomoteur SG90.

III.3.4.2. Fonctionnement de servomoteur :

Les servomoteurs sont actionnés par un servo-contrôleur, qui forment tous deux un servo-entraînement. Grâce à cette combinaison, le moteur peut fonctionner avec un mouvement très dynamique, un couple effectif élevé et une densité de sortie élevée.

Cependant, les plus grandes forces des servomoteurs et leurs fonctions ne sont présentées qu'en interagissant avec le servo-contrôleur. La combinaison de ces éléments permet une programmation assistée par logiciel qui garantit une précision exceptionnelle. Le couple, la position et la vitesse prédéfinis peuvent être contrôlés avec la plus grande précision et s'adapter dynamiquement au mouvement requis. [25]

Les servomoteurs sont utilisés pour contrôler très précisément la position et la vitesse, mais dans un cas simple, seule la position peut être contrôlée. La position mécanique de l'arbre peut être détectée en utilisant un potentiomètre, qui est couplé à l'arbre du moteur par des engrenages. La position actuelle de l'arbre est convertie en signal électrique par le potentiomètre, puis comparée au signal d'entrée de commande. Dans les servomoteurs modernes, des encodeurs ou des capteurs électroniques sont utilisés pour détecter la position de l'arbre. [26]

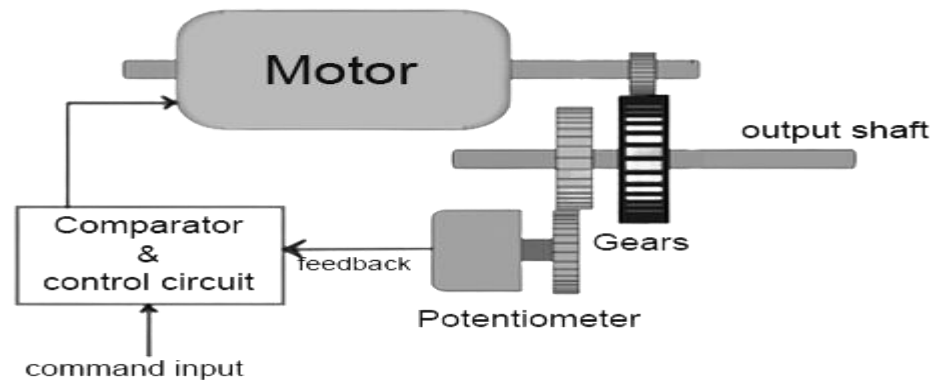


Figure III.5: Fonctionnement de servomoteur. [26]

III.3.4.3. Câblage et programmation de servomoteur :

Le câblage d'un servomoteur est montré dans la **figure III.5**. Dans le code, Il faut qu'on utilise la bibliothèque « servo », puis crée un objet servo (monservo), ensuite attacher à une broche Arduino à l'aide de la fonction « *attach()* » et après utiliser la fonction « *write()* » qui permet de modifier l'angle du bras du servomoteur.

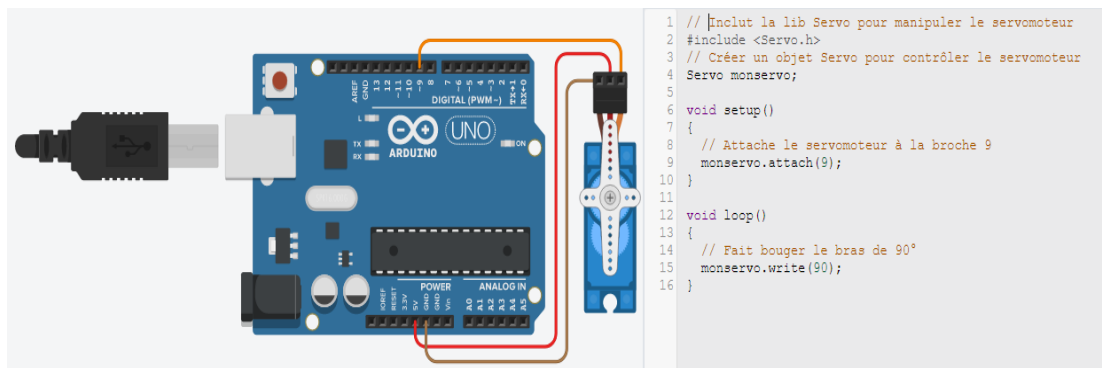


Figure III.6: Câblage et la programmation de Servomoteur SG90 par la plateforme TinkerCad.

III.3.5. Capteur IR :

III.3.5.1. Description de capteur IR :

Le capteur infrarouge (IR) est un capteur électronique qui émet et détecte l'infrarouge. Ce capteur utilise pour détecter les objets en détectant la réflexion du faisceau. Il a un émetteur infrarouge et récepteur qui explore les couleurs claires dans

Chapitre III : Conception et réalisation de projet

une plus grande mesure que les objets sombres qui ne réfléchissent habituellement pas la lumière. Il peut détecter un objet à une distance comprise entre 2 et 30 cm avec un angle de détection de 35°, cette distance peut être réglée par la résistance variable.

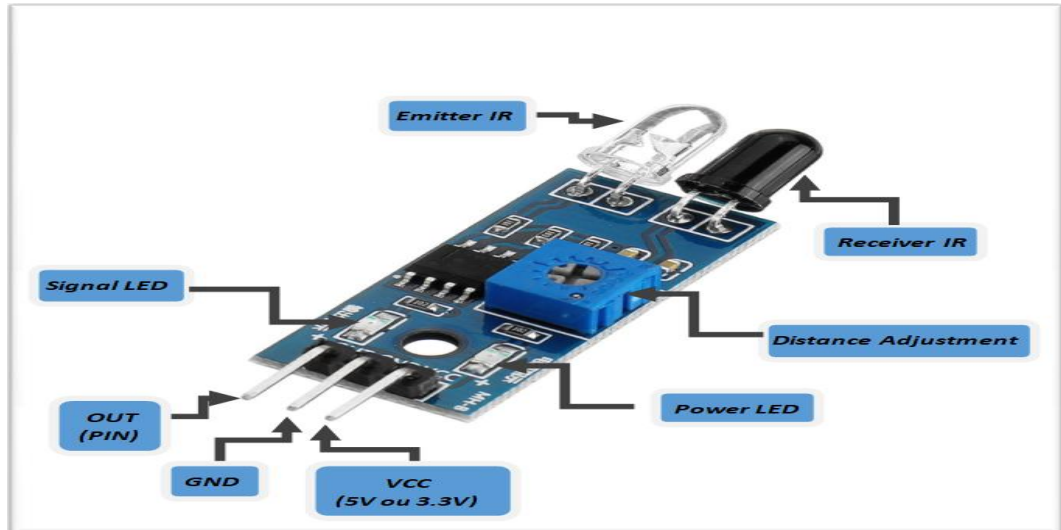


Figure III.7: Capteur IR

III.3.5.2. Fonctionnement de capteur IR :

Les capteurs infrarouges fonctionnent en utilisant un capteur de lumière spécifique pour détecter une longueur d'onde de lumière spécifique. En utilisant une lampe LED qui produit de la lumière à la même longueur d'onde que le capteur recherche, vous pouvez regarder l'intensité de la lumière reçue.

Lorsque l'objet est proche du capteur, la lumière de la LED rebondit sur l'objet et dans le capteur de lumière. Parce que le capteur fonctionne en recherchant la lumière réfléchie, il est possible d'avoir un capteur qui peut renvoyer la valeur de la lumière réfléchie.

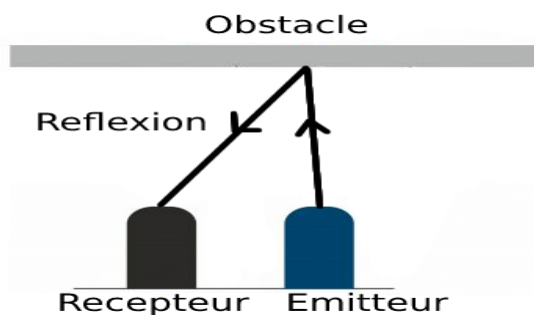


Figure III.8: Principe de fonctionnement d'un capteur IR.

III.3.5.3. Câblage et programmation de capteur IR :

Le câblage de capteur IR à une carte Arduino est montré dans la **figure III.8**. Pour le code, on utilise la fonction « *DigitalRead* » pour lire la valeur du capteur.

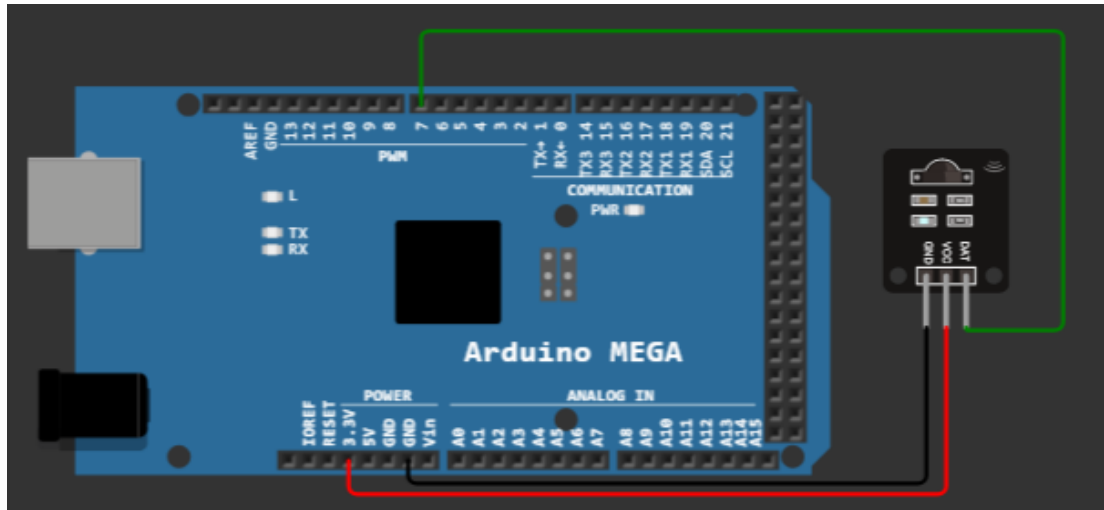


Figure III.9: Câblage et la programmation de capteur IR par la plateforme Wokwi.

III.3.6. RFID RC-522 :

III.3.6.1. Description de RFID RC-522 :

Le module RFID RC522 basé sur le circuit MFRC522. Ce module est conçu pour créer un champ électromagnétique de 13,56 MHz qu'il utilise pour communiquer avec les balises RFID. Ainsi que le lecteur peut communiquer avec un microcontrôleur via une interface périphérique série (SPI) à 4 broches avec un débit maximal de 10 Mbps. Il permet aussi de communiquer via les protocoles I2C et UART. [27]

III.3.6.2. Fonctionnement de RFID :

L'étiquette RFID (carte ou tag) est elle-même équipée d'une puce reliée à une antenne, l'antenne permet à la puce de transmettre les informations (numéro de série) qui sont lues par un lecteur. [27]

Une fois les informations transmises au lecteur RFID équipée d'une antenne intégrée ou externe, celui-ci n'a plus qu'à convertir les ondes-radios en données et ceux-ci pourront être lus par un logiciel RFID. [27]

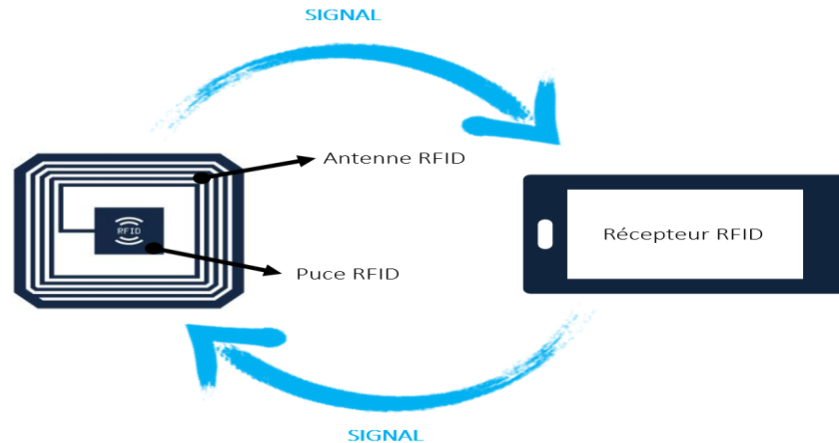


Figure III.10: Fonctionnement de RFID. [28]

III.3.6.3. Câblage et programmation de RFID RC-522 :

Le câblage de RFID à une carte Arduino est montré dans la **figure III.9**. Pour utiliser le module RFID RC522, nous utilisons la bibliothèque « *SPI.h* », qui nous permet d'établir une communication entre la carte Arduino et le module ; et la bibliothèque « *MFRC522.h* », qui nous permet de parler au module.

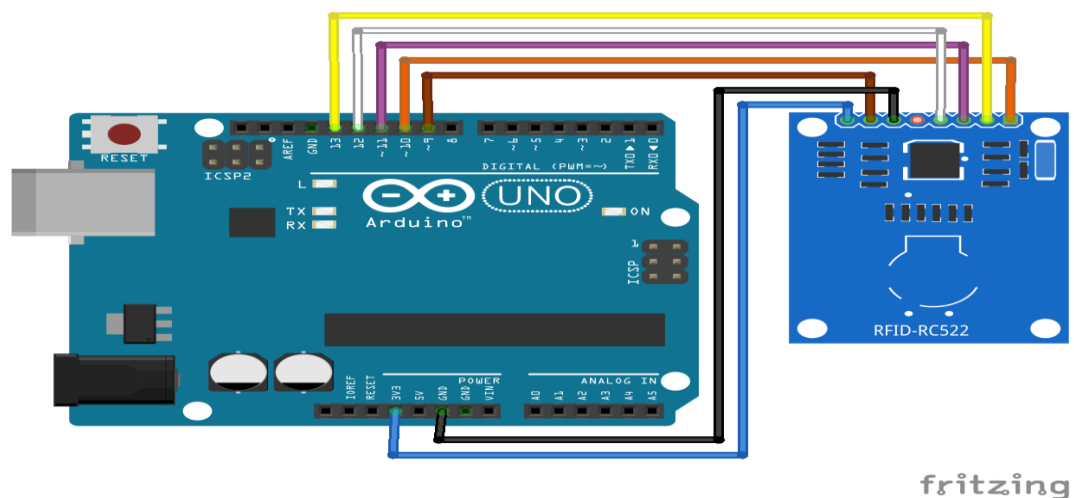


Figure III.11: Câblage de RFID-RC522 par le logiciel fritzing.

III.3.7. Clavier matricielle 3*4 :

III.3.7.1. Description de clavier matricielle :

Les claviers à membrane sont constitués d'une membrane fine et flexible. Ils existent en plusieurs tailles : 4×3 , 4×4 , 4×1 , etc. Quelle que soit leur taille, ils fonctionnent tous de la même manière.



Figure III.12: Clavier matricielle 3*4.

III.3.7.2. Câblage et programmation de clavier :

Le câblage de clavier à une carte Arduino est représenté dans la **figure III.11**.

Pour utiliser le clavier, nous utilisons la bibliothèque « *Keypad.h* », qui nous permet de décoder la matrice.

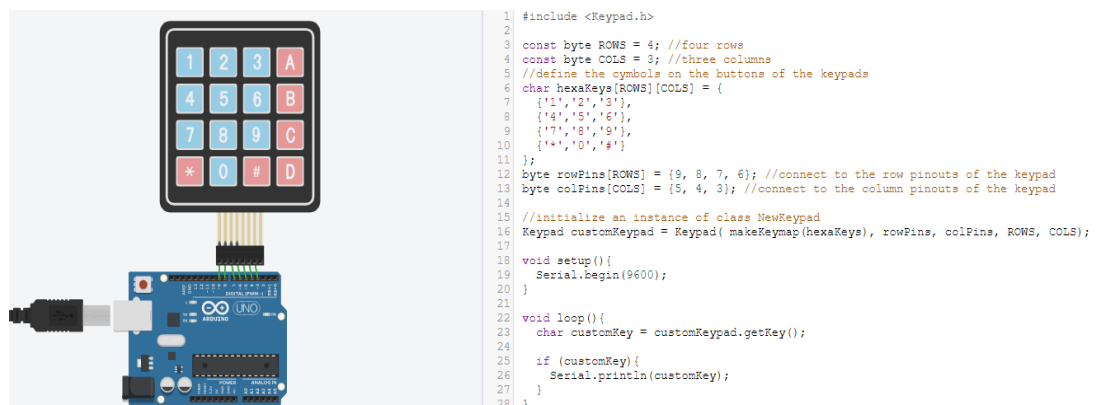


Figure III.13: Le câblage et la programmation de Keypad par la plateforme Tinkercad.

III.4. Softwares applications :

III.4.1. Logiciel PROTEUS :

Proteus est un outil logiciel de simulation et de conception développé par Labcenter Electronics pour la conception de circuits électriques et électroniques. Il s'agit d'une suite logicielle contenant le schéma, la simulation ainsi que la conception de PCB.[29]

En général, le logiciel PROTEUS se compose de deux modules : ISIS et ARES.

Chapitre III : Conception et réalisation de projet

- **ISIS** : ce logiciel de PROTEUS est principalement connu pour la création de schémas et la simulation électrique.
- **ARES** : ce logiciel de PROTEUS est principalement utilisé pour créer des schémas de circuits imprimés. [29]

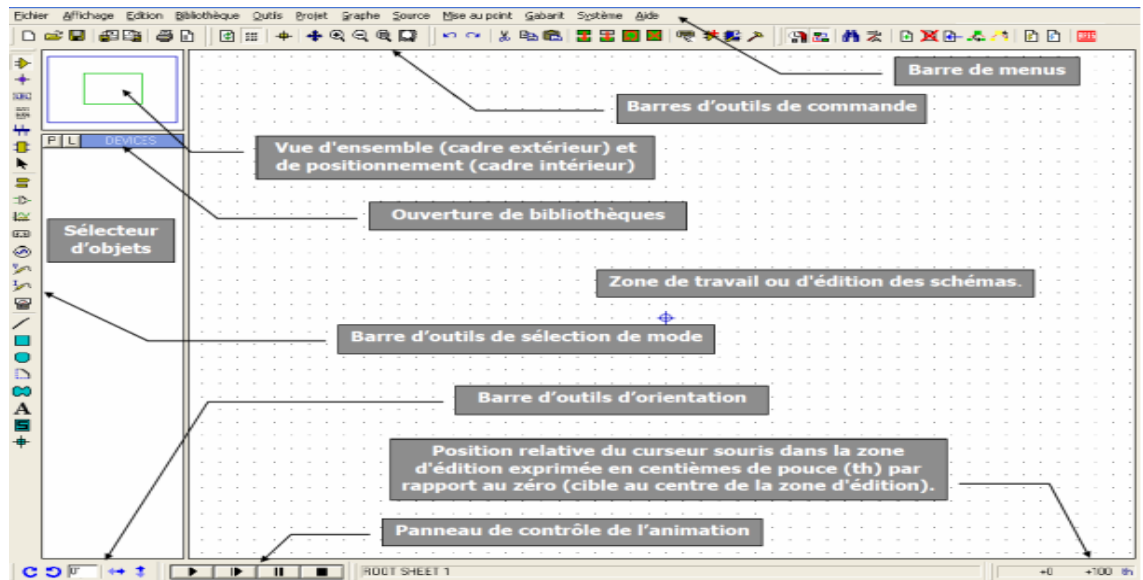


Figure III.14: Interface de logiciel PROTEUS.

III.4.2. Arduino IDE :

Le logiciel Arduino IDE est un programme spécial exécutable sur votre ordinateur qui vous permet d'écrire des croquis pour les cartes Arduino. Un langage simple écrit en Java et basé sur des modèles de langage de traitement, contient un éditeur de texte pour l'écriture de code, une boîte de message, une console de texte, une barre d'outils avec des boutons pour les fonctions fréquemment utilisées et une série de menus. Il se connecte au matériel Arduino et Genuino pour télécharger des programmes et communiquer avec eux. Les programmes Arduino, appelés "sketches", sont écrits dans un langage de programmation basé sur C ++. [30]

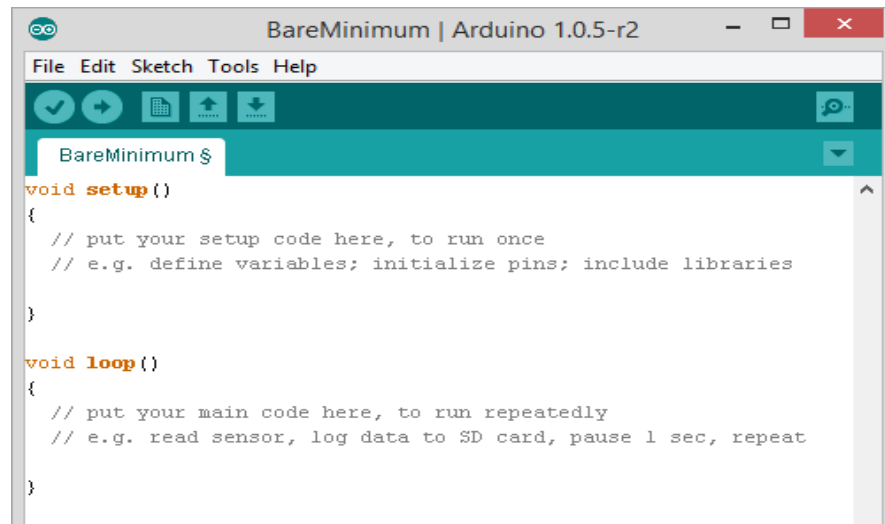


Figure III.15: Interface de Arduino IDE.

Chaque croquis doit avoir deux fonctions.

- « *Setup ()* » : fonction exécutée à l'initialisation de la carte.
- « *Loop ()* » : fonction exécutée en boucle infinie.



Figure III.16: Fonctionnement d'un programme Arduino.

III.4.3. MIT App Inventor :

MIT App Inventor est un RAD (Rapid Application Development) IDE qui a été fournie par Google, et le Massachusetts Institute of Technology (MIT). L'éditeur de MIT App Inventor est un drag-and-drop WYSIWYG editor. Cet éditeur fait partie d'un composant appelé « Graphical user interface builder ».

Chapitre III : Conception et réalisation de projet

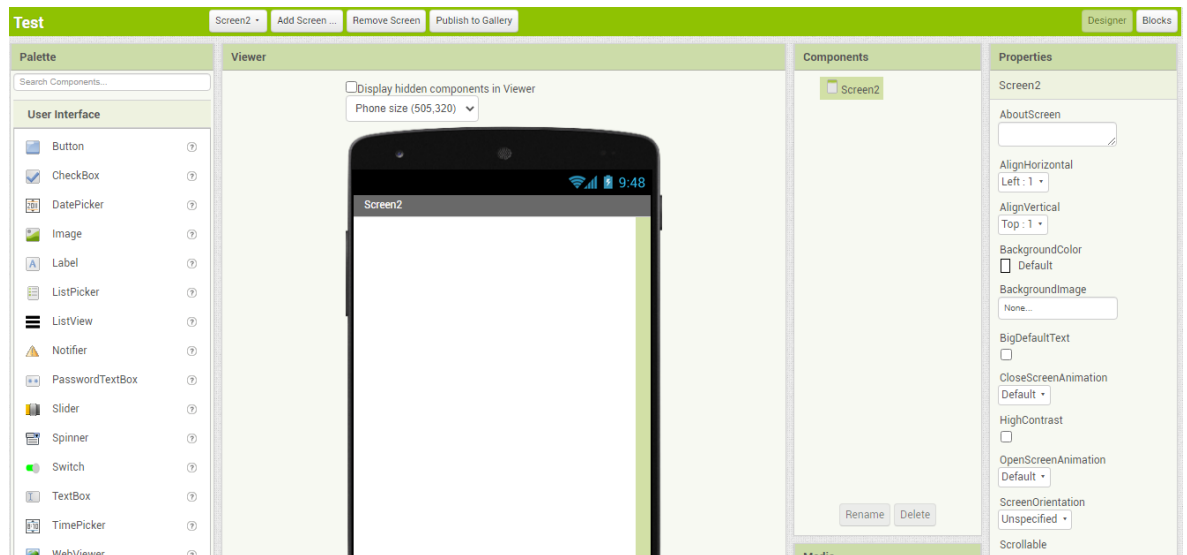


Figure III.17: Interface de MIT App Inventor.

III.4.4. Firebase :

Firebase est la plateforme de Google qui vous permet de développer rapidement des applications pour mobile et Web. Il fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à développer leur base d'utilisateurs et à générer plus de profits. [31]

Firebase est classé comme un programme de base de données NoSQL, qui stocke les données dans des documents de type JSON. [31]



Figure III.18: Logo de Firebase. [30]

III.4.4.1. Service de Firebase :

- **Base de données en temps réel** : les données sont synchronisées en temps réel sur tous les clients et restent disponibles même lorsque l'application est hors ligne. [32]
- **Authentification** : une fonctionnalité d'authentification Google pour les applications utilisant Firebase. Il vous permet d'authentifier les utilisateurs à

Chapitre III : Conception et réalisation de projet

l'aide d'une interface utilisateur prédéfinie ou personnalisée et de connecter les utilisateurs via des informations d'identification personnalisées, un e-mail ou des médias sociaux. [32]

- **Hosting** : Firebase Hosting fournit un hébergement rapide pour les applications Web ; le contenu est mis en cache dans les réseaux de diffusion de contenu du monde entier. [32]



Figure III.19: Service de Firebase. [31]

III.5. Prototype de projet :

III.5.1. Architecture de prototype :

La maquette est composée d'une entrée et d'une sortie et de trois parkings.

Le principe de fonctionnement est le même pour chaque parking.

Le parking de département 1 est composé de cinq places de stationnement, le parking de département 2 et le parking de visiteur est composé de six places de stationnement.

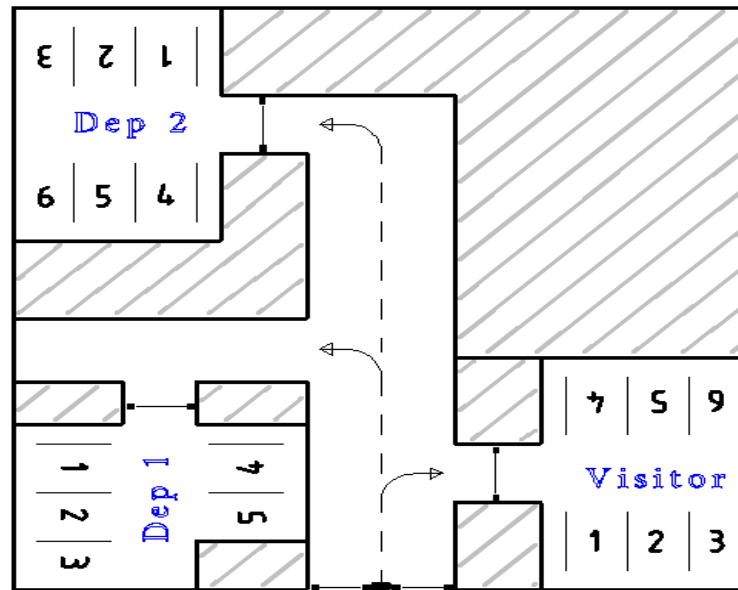


Figure III.20: Architecture des parkings sur logicielle LibreCAD.

III.5.2. Description de prototype :

Puisque le fonctionnement est le même pour chaque place de stationnement on n'utilise que six places pour découvrir le fonctionnement de notre système, donc deux places pour chaque parking.

La figure suivante représente le schéma de branchement pour le système de stationnement intelligent typique :

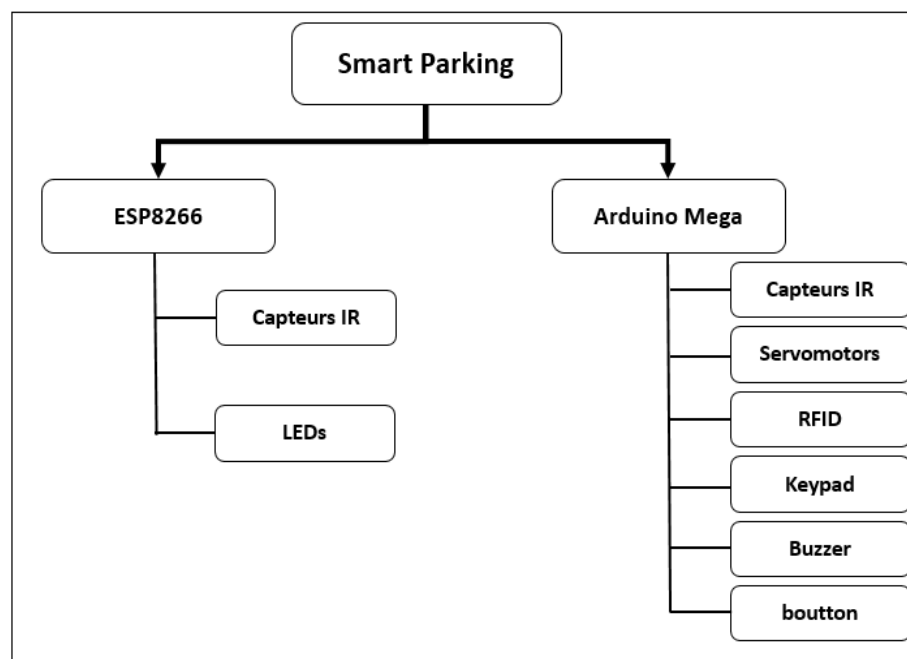


Figure III.21: Schéma générale de branchement du système prototype.

Chapitre III : Conception et réalisation de projet

Pour mieux comprendre le fonctionnement de notre prototype, nous avons décomposé notre projet trois parties :

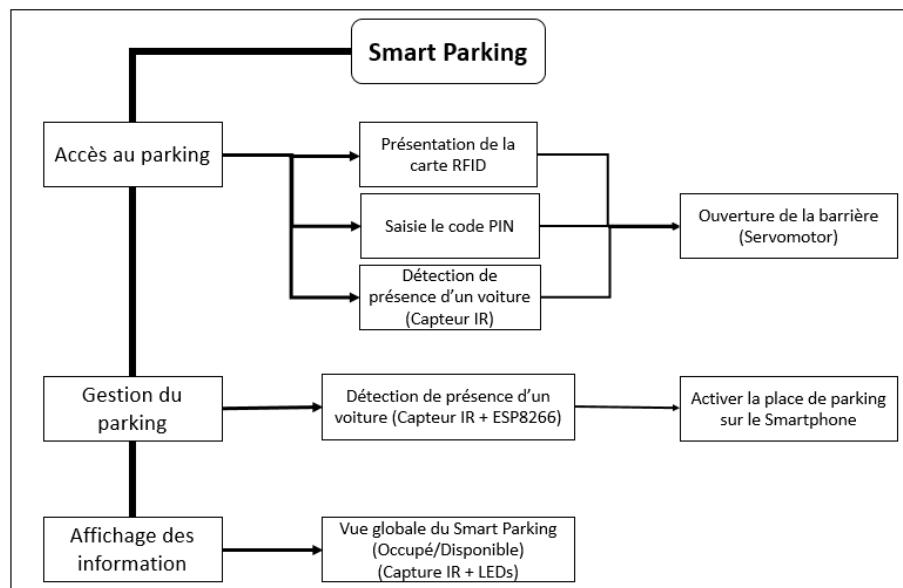


Figure III.22: Représentation schématique du prototype.

III.5.3. Implémentation de prototype :

Avant de passer à la réalisation de notre circuit, il est très intéressant de simuler notre prototype, et la figure ci-dessous illustre le circuit de notre maquette :

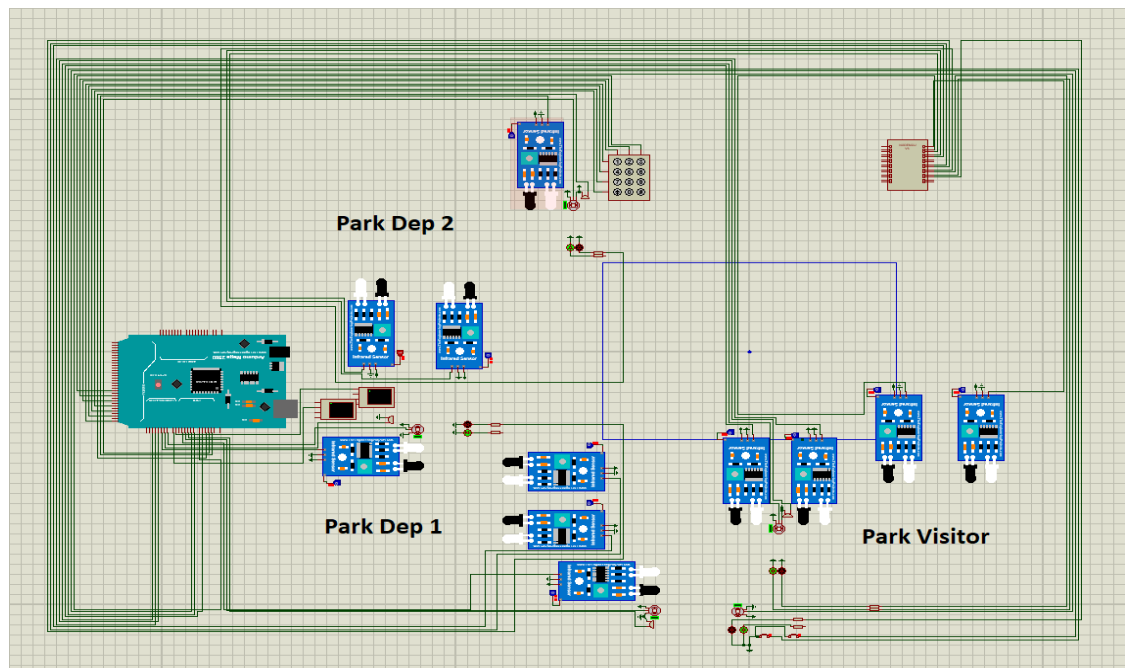


Figure III.23: Implémentation des composants de prototype sur le logiciel PROTEUS.

III.5.3.1. Implémentation de parking visiteurs :

Pour le parking de visiteur, l'accès est assuré par capteur IR et un servomoteur. Ces deux éléments sont reliés à la carte Arduino dont la sortie capteur IR est reliée à l'entrée analogique de la carte Arduino tout en étant reliée à la sortie numérique de l'Arduino Mega.

En fait, s'il y a un véhicule à l'entrée, le générateur de tension à la sortie du module IR permet à l'Arduino Mega de contrôler le servomoteur et la position de la barrière.

Pour la sortie, c'est le même principe que l'entrée.

Pour chaque place de stationnement, on utilise un capteur IR relié à la carte ESP8266. Le capteur IR dès qu'il détecte la présence d'un véhicule ; il délivre un signal électrique à la carte ESP8266.

Si toutes les places de stationnement sont occupées, le led rouge sera allumé et le led vert sera éteint.

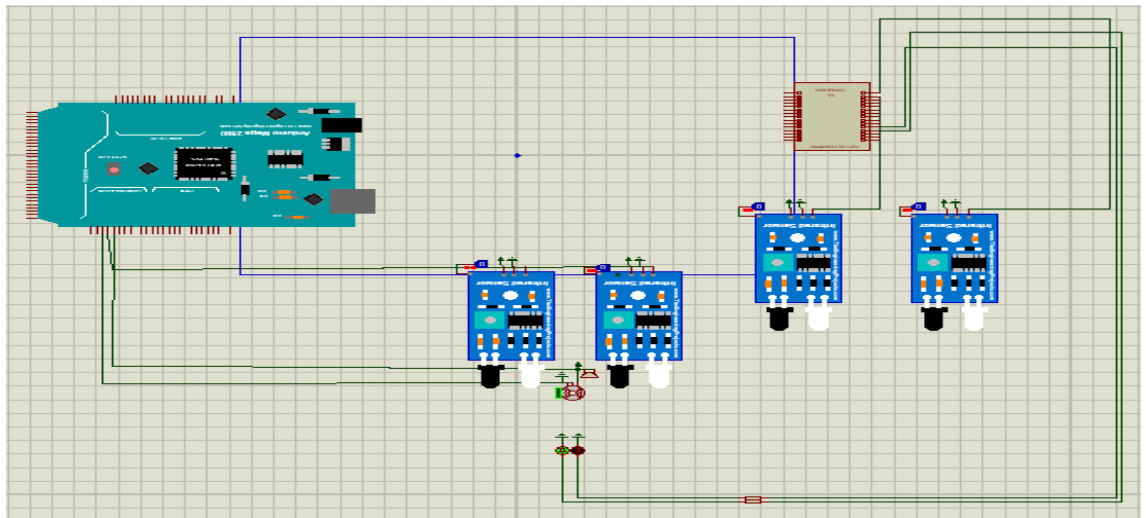


Figure III.24: Le branchement de composants de parking visiteur.

III.5.3.2. Implémentation de parking de département 1 :

Pour le parking de département 1, l'accès est effectué via un système RFID (carte RFID + lecteur). Lorsque la carte RFID est placée devant le lecteur, la barrière va être ouverte pendant 5 secondes puis ferme.

Chapitre III : Conception et réalisation de projet

Pour la sortie, lorsque le capteur infrarouge détecte une voiture, la porte est ouverte.

Pour les places de stationnement, on a le même principe de parking visiteurs.

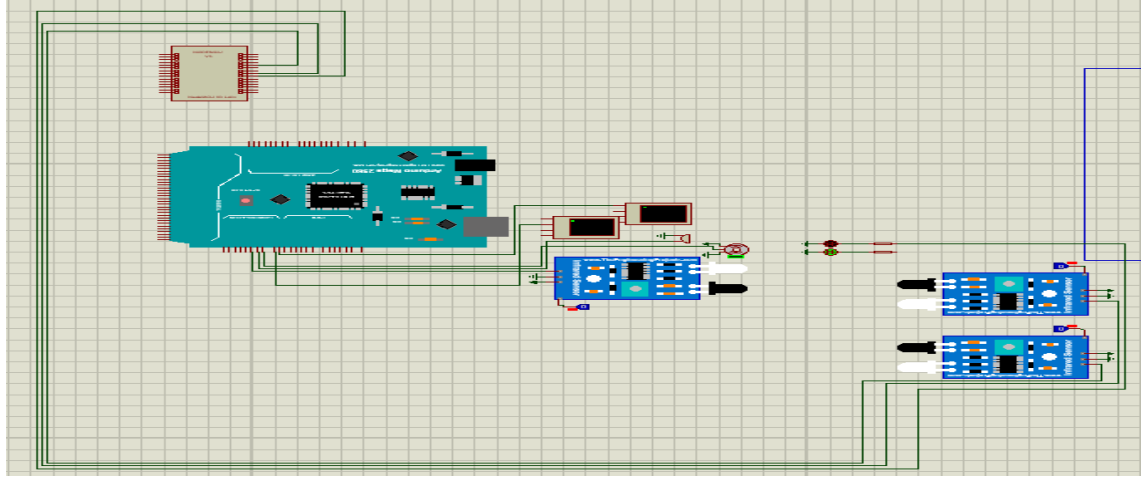


Figure III.25: Le branchement de composants de parking de département 1.

III.5.3.3. Implémentation de parking de département 2 :

Pour le parking de département 2, l'accès est effectué via un code PIN (Keypad). Lorsque le code PIN est saisi correctement, la barrière va être ouverte pendant 3 secondes puis ferme.

Pour la sortie, le même principe de parking de département 1.

Le même principe appliqué dans d'autres parkings est appliqué dans ce parking.

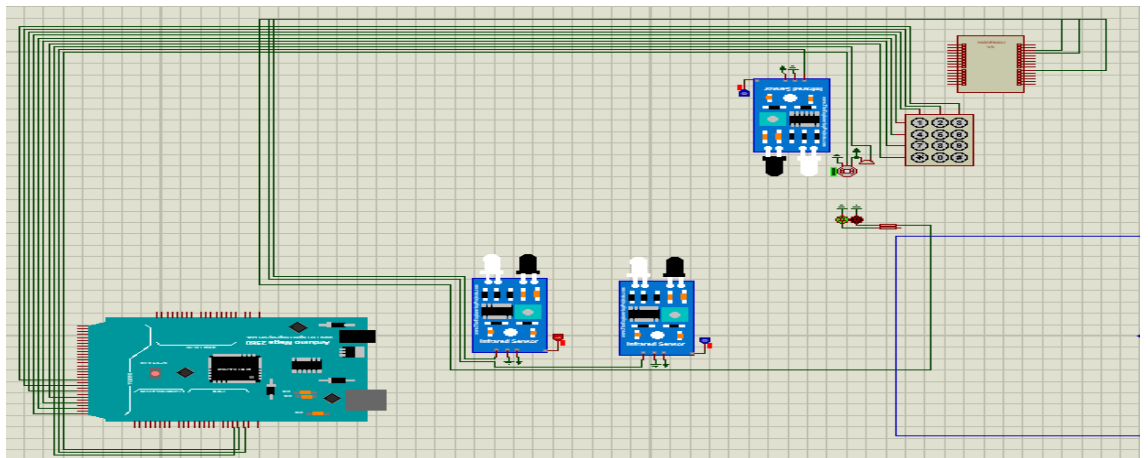


Figure III.26: Le branchement de composants de parking de département 2.

III.5.3.4. Implémentation de l'entrée et la sortie principale :

L'entrée principale est contrôlée manuellement par des boutons pour l'ouverture et la fermeture du porte.

Si tous les parkings (visitor, Dep1 et Dep2) sont occupés, le led rouge à l'entrée principale sera allumé.

Pour la sortie principale, lorsque le capteur infrarouge détecte la présence d'une voiture, la porte est ouverte.

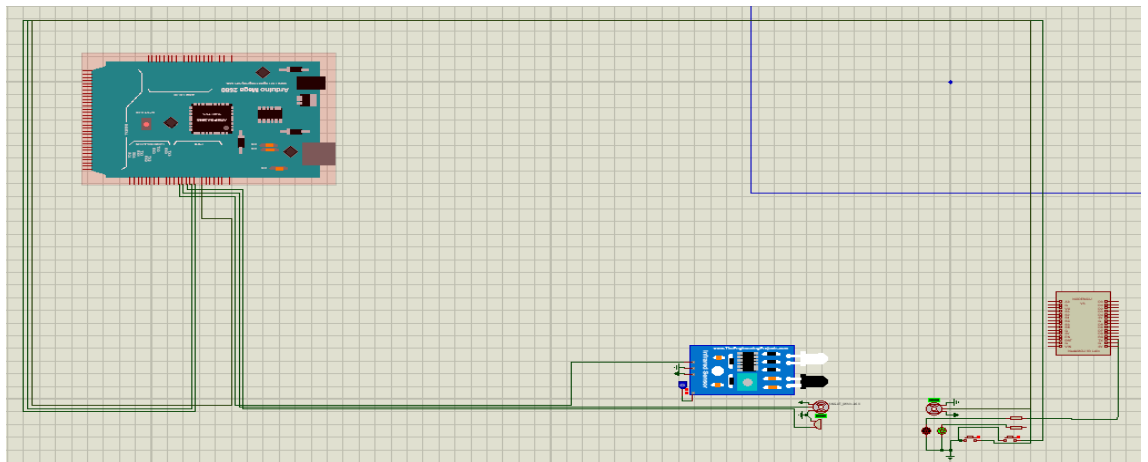


Figure III.27: Le branchement de composants de l'entrée et de sortie principale.

III.6. Interconnexion entre les différents modules de système :

Pour réaliser l'IoT dans notre projet, nous avons connecté les six capteurs IR à la carte ESP8266 qui est connectée à l'internet et qui communique l'état positions requises à partir des capteurs vers la base de données Firebase en temps réel.

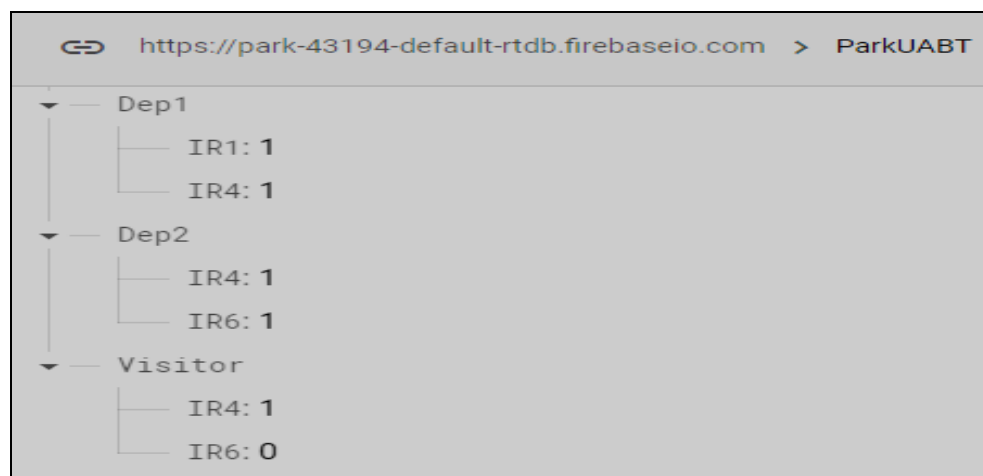


Figure III.28: Création des capteurs IR sur Firebase.

Chapitre III : Conception et réalisation de projet

La base de données qui a été créée sur Firebase est considérée comme un PMS et l'ensemble de la communication entre le PMS et hardware et PMS vers l'application mobile est alimenté par l'internet.

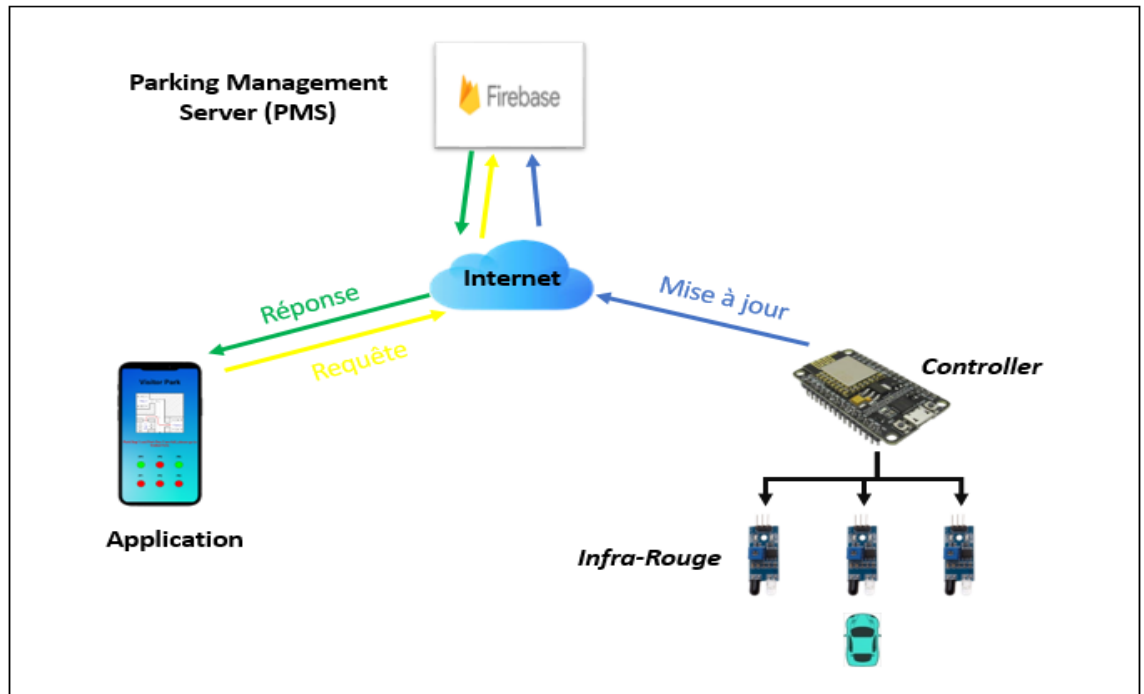


Figure III.29: Schéma représente la communication entre les différents modules de système.

ESP8266 est le contrôleur principal pour contrôler les places de stationnement. Il surveille périodiquement et obtient le statut de chaque place de stationnement, via des capteurs IR. Il interagit également avec PMS via Internet et publie le statut de parking.

Les messages échangés entre l'application et PMS sous forme requête - réponse.

III.7. Application mobile :

III.7.1. Analyse de besoins :

III.7.1.1. Besoins fonctionnels :

Cette partie est réservée aux besoins fonctionnels des différents acteurs de l'application. Le système doit assurer :

- L'authentification de l'utilisateur.

Chapitre III : Conception et réalisation de projet

- La consultation des places libres.
- L'orientation de l'utilisateur vers les places disponibles.

III.7.1.2. Besoins non-fonctionnels :

- **Utilisabilité** : l'interface d'utilisateur est facile de compréhension, facile à utiliser et attractive.
- **Disponibilité** : possibilité d'utiliser l'application par n'importe quel utilisateur.
- **Performance** : les performances du système en termes de temps de réponse.
- **Maintenabilité** : facilité de modification, stabilité et testabilité.
- **Sécurité** : le système doit être sécurisé (une sécurité totale) surtout au niveau de gestion des accès aux données.
- **Ergonomie** : l'utilisateur peut réaliser les tâches de façon sécuritaire, confortable et efficace.

III.7.2. Modélisation :

Pour notre modélisation d'application, nous avons utilisé trois diagrammes : le diagramme de classe pour modéliser l'aspect statique, le diagramme de cas d'utilisation et d'activité pour modéliser l'aspect dynamique.

III.7.2.1. Diagramme de cas d'utilisation :

La figure ci-dessous représente le diagramme de cas d'utilisation donnant à apercevoir des fonctionnements du notre système :

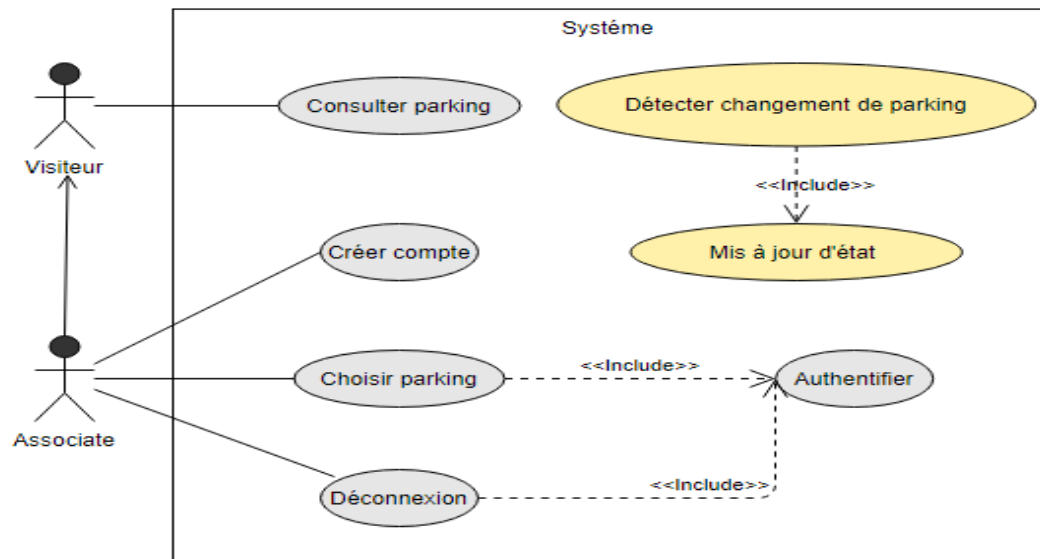


Figure III.30: Diagramme de cas d'utilisation global.

III.7.2.2. Diagramme d'activité :

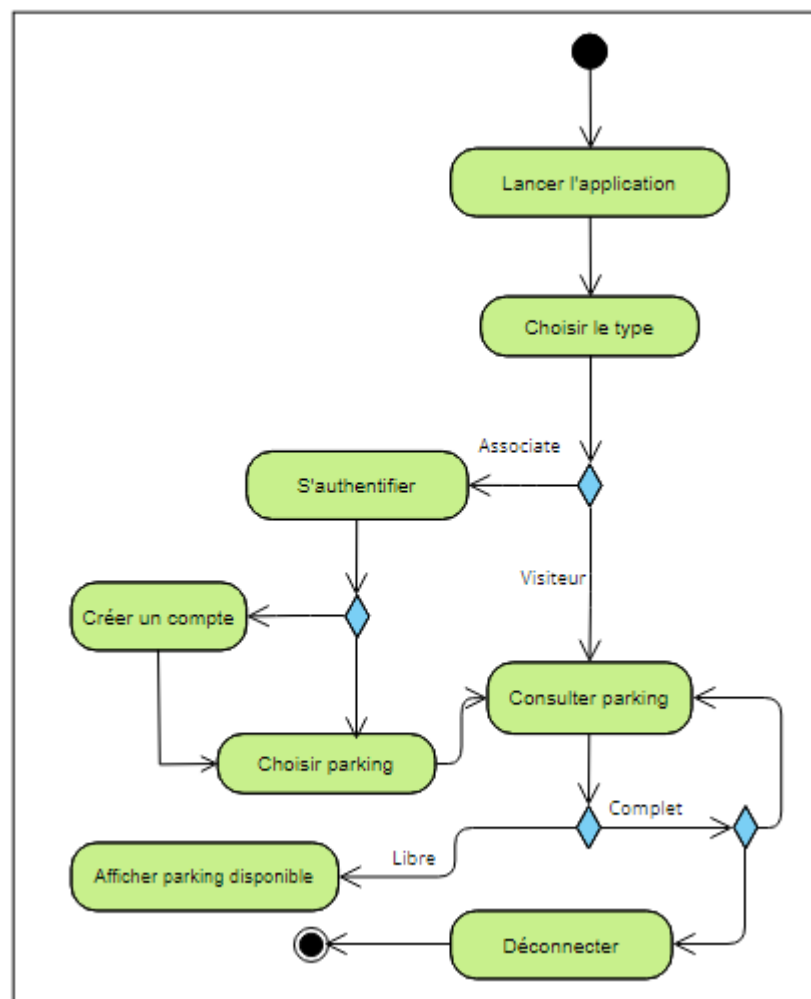


Figure III.31: Diagramme d'activité global.

III.7.2.3. Diagramme de classe :

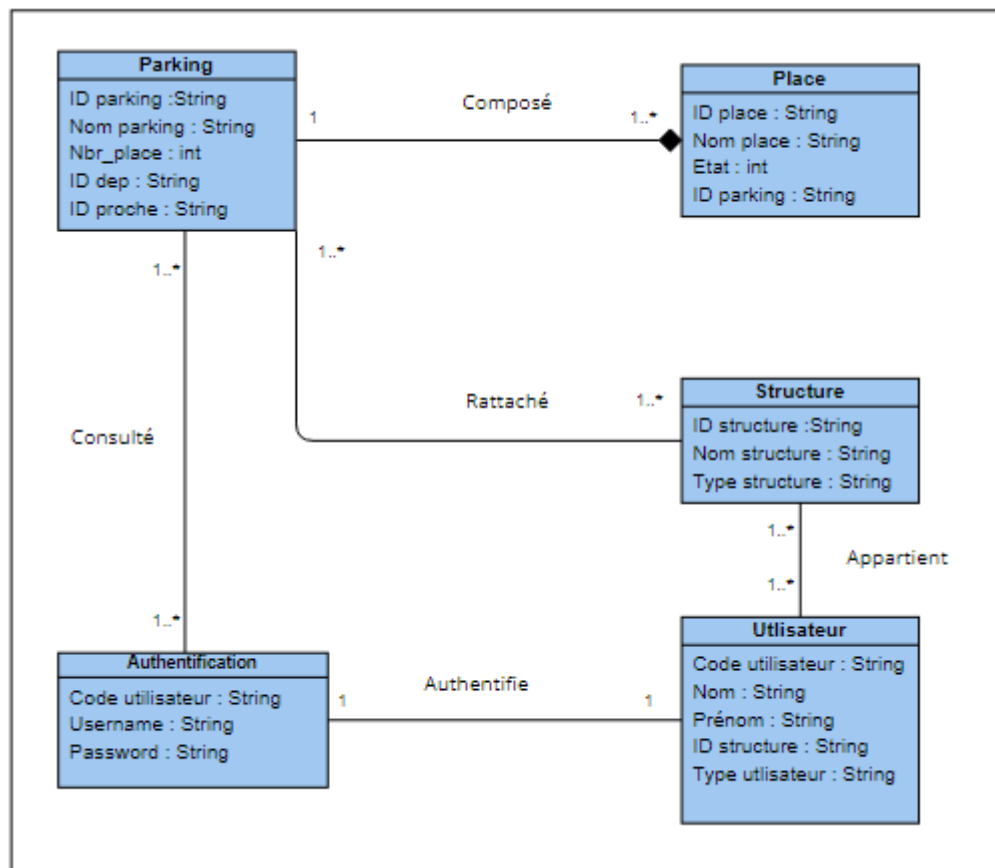


Figure III.32: Diagramme de classe globale.

III.7.3. Présentation de l'application :

Dans cette section, nous allons présenter notre application mobile et nous montrons son fonctionnement. Ceci assure une meilleure convergence entre le système et les besoins de ces utilisateurs. Pour accéder à l'application « ParkUABT », l'utilisateur doit télécharger et installer l'application sur son smartphone.



Figure III.33: 1^{er} interface de le lancement de l'application.

Chapitre III : Conception et réalisation de projet

La figure suivante illustre la première interface visualisée par l'utilisateur lors d'accès à l'application où l'utilisateur doit d'abord choisir son type :

- Le premier bouton « **Associate** » : permet aux utilisateurs (enseignants, personnels et étudiants) de créer un compte et consulter les parkings des départements.
- Le deuxième bouton « **Visitor** » permet à l'utilisateur d'accéder et de consulter parking visiteurs seulement.

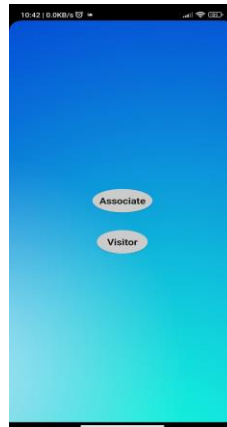


Figure III.34: L'interface de choix le type d'utilisateur.

III.7.3.1. Interface visiteur :

L'interface illustrée dans la **figure III.34** permet aux utilisateurs de consulter l'état de parking visiteurs (place libre en vert et place occupée en rouge). Si le parking visiteurs est plein, l'interface suivante s'affichera la **figure III.35**.

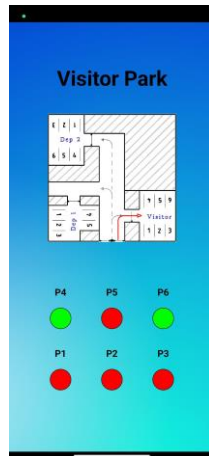


Figure III.35: Interface de parking visiteur.

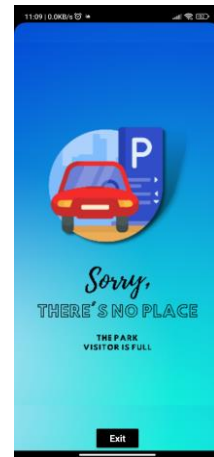


Figure III.36: Interface de parking visiteur plein.

III.7.3.2. Interface associe :

- Page d'authentification :

L'authentification par nom d'utilisateur et mot de passe via le formulaire de connexion est nécessaire pour accéder aux services de l'application.

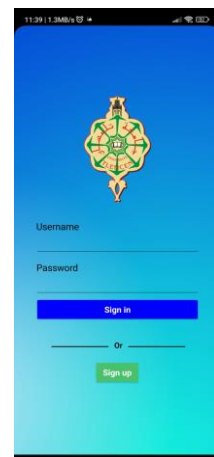


Figure III.37: Page d'authentification.

- **Page d'inscription :**

La page d'inscription vous permet de créer un compte pour accéder aux services offerts par l'application.

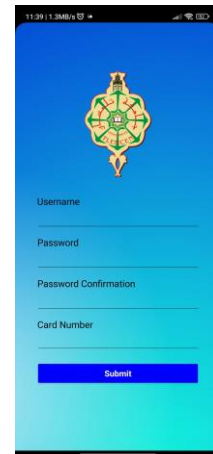


Figure III.38: Page d'inscription

- **Page de sélection :**

Une fois que l'utilisateur s'est authentifié, il doit sélectionner son département.

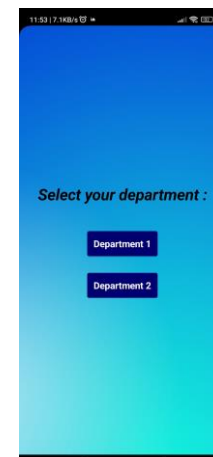


Figure III.39: Page de sélection.

- **Affichage d'un parking :**

Exemple : affiche de l'état de « Département1 ».

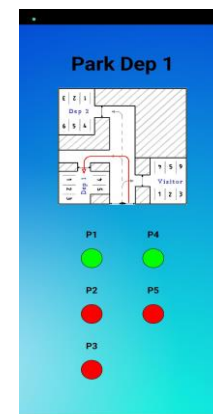


Figure III.40: Réponse à la requête.

III.8. Scénarios d'utilisation :

III.8.1. Initialisation et authentification :

- **1^{er} scénario** : initialisation d'application

Lorsque l'application est lancée après l'installation, elle demande à l'utilisateur de choisir son type (Associate ou Visitor). (**Figure III.33**)

- **2e scénario** : authentification d'utilisateur

Si l'utilisateur (le type associe.) a déjà un compte, il remplit le formulaire de connexion (Username, Password).

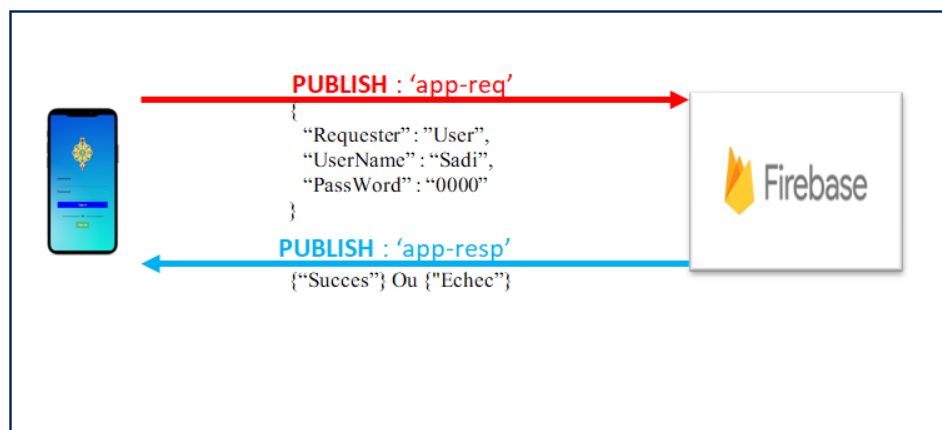


Figure III.41: Échange des messages entre l'application et BD en cas d'authentification.

- **3e scénario** : création de compte

Quand utilisateur (le type associe.) pour a première fois doit créer un compte.

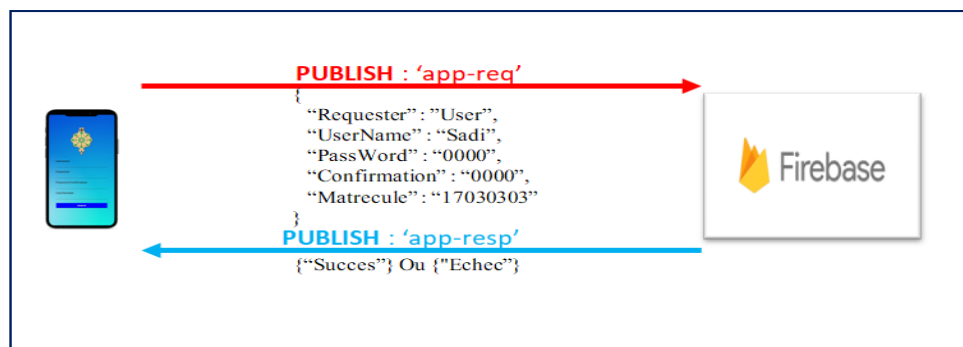


Figure III.42: Échange des messages entre l'application et BD en cas de création de compte.

III.8.2. Découverte :

- **1^{er} scénario** : type visiteur

Quand l'utilisateur choisit le type (visitor), l'application envoie une requête au PMS (Firebase) pour obtenir l'état de toutes les places de stationnement et l'itinéraire de parking visiteurs. Chaque place de stationnement est identifiée par un ID (P1, P2, ...).

Voici comment l'échange de message s'effectue entre le PMS et l'application :

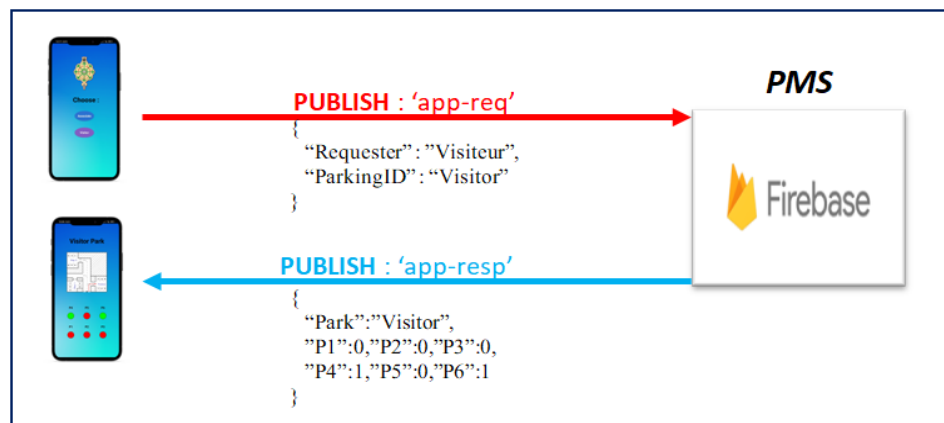


Figure III.43: Echange d'information entre PMS et l'application (Visiteur).

L'application affiche la carte de parking et les places vacantes de parking sont indiquées en vert, alors que celles qui sont occupées sont indiquées en rouge. (**Figure III.31**)

Si toutes les places de stationnement sont occupées, l'application affiche (**Figure III.32**).

- **2^e scénario** : le type associe.

Après l'authentification, l'utilisateur doit sélectionner son département :

- **1^{er} cas** : sélection de département 1

Après la sélection de département 1, l'application envoie une demande au PMS pour obtenir l'état de toutes les places de stationnement et l'itinéraire de parking Département 1.

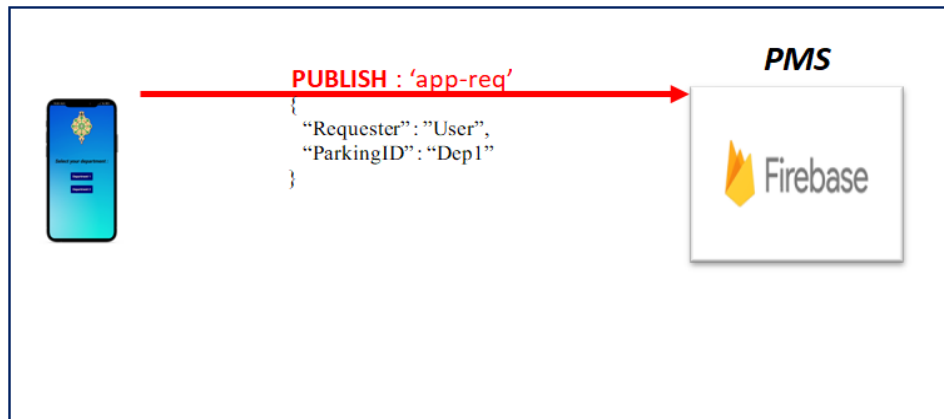


Figure III.44: L'application envoie une requête à PMS.

- Si le parking de département 1 a des places libres, le PMS répond :



Figure III.45: PMS répond à la requête de l'application (parking dep1).



Figure III.46: Parking dep 1.

- Sinon si le parking de département 1 est plein, le PMS répond par l'état de places de parking le plus proche parking de département 2.



Figure III.47: PMS répond à la requête de l'application (parking dep 2).

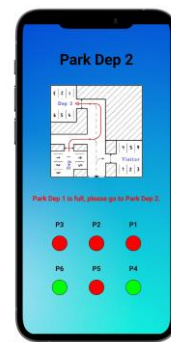


Figure III.48: Parking dep 2.

- Sinon si le parking de département 1 et le parking de département 2 sont pleins, le PMS répond par l'état de places de parking le plus proche parking de visiteur.



Figure III.49: PMS répond à la requête de l'application (Parking visitor).



Figure III.50: Parking visiteur.

➤ 2e scénario : sélection de département 2

Après la sélection de département 2, l'application envoie une demande au PMS pour obtenir l'état de toutes les places de stationnement et l'itinéraire de parking Département 2.



Figure III.51: L'application envoi une requête à PMS.

- Si le parking de département 2 a des places libres, le PMS répond :

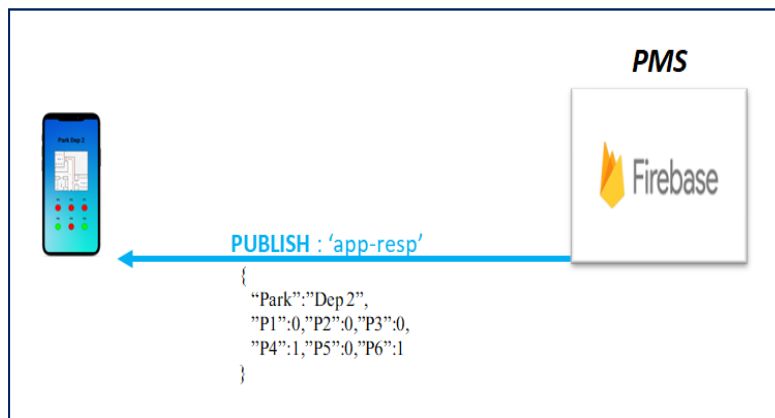


Figure III.52: PMS répond à la requête de l'application (parking dep2).

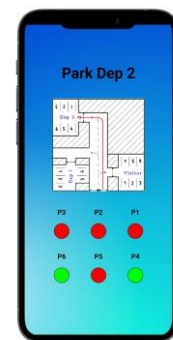


Figure III.53: Parking dep 2.

- Sinon si le parking de département 2 est plein, le PMS répond par l'état de places de parking le plus proche parking de département 1.

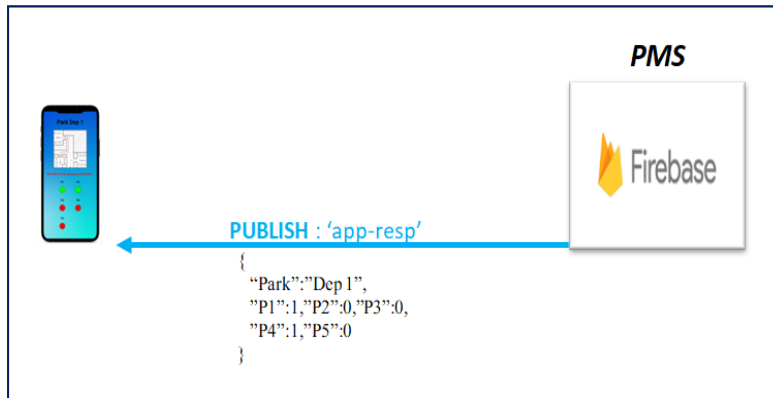


Figure III.54: PMS répond à la requête de l'application (parking dep 1).



Figure III.55: Parking dep 1

Dans les deux cas, quand les deux parkings (département 1 et département 2) sont pleins, le PMS répond à l'état de places de parking le plus proche parking de visiteur.



Figure III.56: PMS répond à la requête de l'application (parking visitor).



Figure III.57: Parking visitor

Si tous les parkings sont plein, PMS répond :

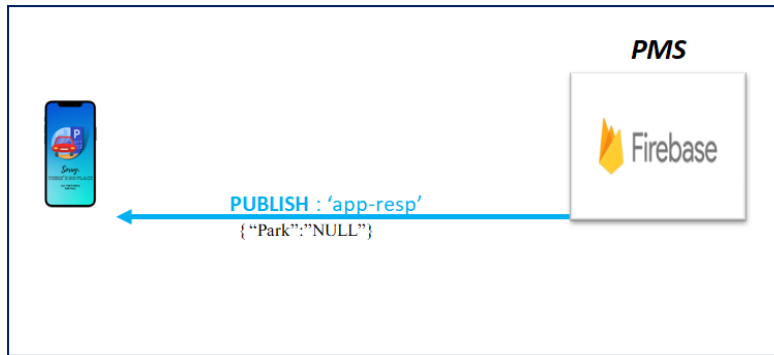


Figure III.58: PMS répond à la requête de l'application (NULL).



Figure III.59: Plein

Dans les deux cas, l'utilisateur a le choix de patienter en cas de libération d'une place où quitter l'application.

III.8.3. Mise à jour de la base Firebase :

Chaque fois que la place de stationnement détecte la présence ou l'absence d'une voiture, il signale immédiatement le PMS.

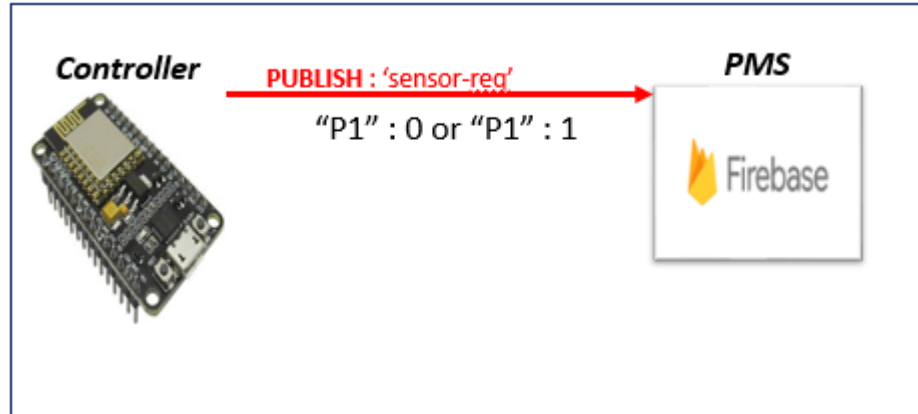


Figure III.60: La requête de contrôleur à PMS.

Conclusion générale :

Depuis l'aube de l'humanité, les êtres humains recherchent et utilisent des moyens qui permettent aux personnes d'accomplir plus facilement des tâches, et d'accomplir leurs tâches rapidement, facilement et en toute sécurité. Pour suivre le développement du monde, en particulier le rythme de développement continu technologie. Cela est dû à la nature d'humanité ambitieuse qui aime les changements positifs dans tous les aspects de la vie.

L'objectif de ce projet est de créer un exemple de parking intelligent pour l'Université de Tlemcen. En ce sens, nous avons pu développer un système qui utilise des outils d'orientation au stationnement via une application mobile utilisant l'IoT.

Au début, nous étudions l'Internet des objets et le stationnement et la relation entre les deux. Ensuite, présentons les dernières avancées en matière de parking intelligent. Puis, nous donnons une description des composants électroniques utilisés pour réaliser un parking intelligent.

Après cela, nous avons réalisé un prototype de parking intelligent, qui s'est décomposé en trois parties : parking visiteurs, parking de département 1 et parking de département 2. Dans un premier temps, nous avons développé une application mobile avec l'environnement MIT App Inventor, permettant aux clients d'interroger l'état des places de parking.

Au final, ce projet nous a été très bénéfique pour mettre à profit nos connaissances, notamment en informatique et en Arduino. Ce projet nous permet de valider les connaissances acquises tant sur le plan théorique que pratique.

- **Perspectives :**

Le travail que nous avons réalisé peut-être amélioré et enrichi pour en faire un système plus performant. Parmi les perspectives à prendre en compte pour améliorer le fonctionnement de notre projet, nous citons notamment :

- ❖ Création d'un site web pour l'application.
- ❖ Ajouter d'autres fonctionnalités utiles (éclairage automatique, système de guidage, ...).
- ❖ Ajouter un parking VIP.
- ❖ Ajouter de nouveaux moyens pour accéder aux parkings (caméra, capteurs plus performant, ...).



Références Bibliographiques

- [1] Microsoft. (2015). IoT-For-Beginners. GitHub. <https://github.com/microsoft/IoT-For-Beginners/blob/main/1-getting-started/lessons/1-introduction-to-iot/README.md#what-is-the-internet-of-things> [08/03/2022]
- [2] Steve (2020, 26 novembre). Beginners Guide to the Internet of Things (IOT). Steve's Internet Guide. <http://www.steves-internet-guide.com/internet-of-things/> [08/03/2022]
- [3] What is Internet of Things (IoT)? (2020, 13 janvier). Electronics Hub. <https://www.electronicshub.org/internet-of-things/> [08/03/2022]
- [4] IoT Architecture: The Pathway from Physical Signals to Business Decisions. (2020, août 12). AltexSoft. <https://www.altexsoft.com/blog/iot-architecture-layers-components/> [09/03/2022]
- [5] How IoT Works – 4 Main Components of IoT System. (2021, 9 mai). DataFlair. <https://data-flair.training/blogs/how-iot-works/> [09/03/2022]
- [6] i-SCOOP. (2022, 10 mai). What is IoT? The Internet of Things defined and explained. <https://www.i-scoop.eu/internet-of-things-iot/internet-of-things-what-definition/> [10/03/2022]
- [7] <https://trackinno.com/wp-content/uploads/2018/08/how-iot-works-summary.001.png#main> [11/03/2022]
- [8] Zaslavsky, A., & Jayaraman, P. P. (2015). Discovery in the Internet of Things. Ubiquity, 2015(October). <https://doi.org/10.1145/2822529> [21/06/2022]
- [9] Balakrishnan, S. M., & Sangaiah, A. K. (2016). Aspect-oriented middleware framework for resolving service discovery issues in Internet of Things.

International Journal of Internet Protocol Technology, 9(2/3).

<https://doi.org/10.1504/ijipt.2016.079546> [21/06/2022]

- [10] Aziez, M., Benharzallah, S., & Bennoui, H. (2017). A Comparative Analysis of Service Discovery Approaches for the Internet of Things. International Research Journal of Electronics and Computer Engineering, 3(1).
<https://doi.org/10.24178/irjece.2017.3.1.17> [21/06/2022]
- [11] Pourghebleh, B., Hayyolalam, V. & Aghaeianvigh, A. Service discovery in the Internet of Things: review of current trends and research challenges. Wireless Netw 26, 5371–5391 (2020) [21/06/2022]
- [12] Zorgati, H., Djemaa, R. B., & Amor, I. A. B. (2019, October). Service discovery techniques in Internet of Things: a survey. In 2019 IEEE international conference on systems, man and cybernetics (SMC) (pp. 1720-1725). IEEE. [21/06/2022]
- [13] Evdokimov, S., Fabian, B., Kunz, S., & Schoenemann, N. (2010, June). Comparison of discovery service architectures for the internet of things. In 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (pp. 237-244). IEEE. [21/06/2022]
- [14] Oxford University Press (OUP). parking. Lexico.Com.
<https://www.lexico.com/definition/parking> [05/07/2022]
- [15] WIKIPEDIA CONTRIBUTORS. (2022, 5 JUILLET). PARKING. WIKIPEDIA.
<https://en.wikipedia.org/wiki/Parking> [05/07/2022]
- [16] Karsten, J. (2019, 20 mars). What is Smart Parking? Parkeagle.
<https://www.parkeagle.com/2018/05/12/what-is-smart-parking/> [20/05/2022]
- [17] Ultimate Guide to IoT Based Smart Parking System. (2022, 13 janvier). CHINT Blog. <https://chintglobal.com/blog/iot-based-smart-parking-system/> [21/05/2022]
- [18] Paidi, V., Fleyeh, H., Håkansson, J., & Nyberg, R. G. (2018). Smart parking sensors, technologies and applications for open parking lots: a review. IET Intelligent Transport Systems, 12(8), 735-741. <https://doi.org/10.1049/iet-its.2017.0406> [21/05/2022]

- [19] Nieves, J. M. (2022, 26 avril). How to choose a Smart Parking sensor? ThingPark Market. <https://market.thingpark.com/blog/post/how-to-choose-smart-parking-sensor> [22/05/2022]
- [20] Vehicle Detection Sensors. (2022, 8 avril). Smart Parking. <https://www.smartparking.com/smartpark-system/smart-sensors> [25/05/2022]
- [21] Adaramola, S. (2020, 22 mars). Pros and Cons of Parking Management Systems | BookingNinjas. Booking Ninjas. <https://www.bookingninjas.com/blog/pros-and-cons-of-parking-management-systems/> [23/06/2022]
- [22] Arduino Mega 2560 Datasheet.Robotshop. [28/05/2022]
- [23] Insight into ESP8266 NodeMCU Features & Using It with Arduino IDE. (2020, 18 décembre). Last Minute Engineers. <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/> [28/05/2022]
- [24] What is a servomotor? (2020, 28 octobre). Electrical4U. <https://www.electrical4u.com/what-is-servo-motor/> [05/07/2022]
- [25] What is a servo motor ? How it works & application. (2021, août 2). Baumüller. <https://www.baumueller.com/en/insights/basics/servomotor-how-it-works-properties-areas-of-use#how-a-servo-motor-works> [05/07/2022]
- [26] Daware, K. electricaleasy.com: How does a servo motor work? www.electricaleasy.com. <https://www.electricaleasy.com/2015/01/how-does-servo-motor-work.html> [05/07/2022]
- [27] What is RFID? How It Works? Interface RC522 RFID Module with Arduino. (2022, 15 mai). Last Minute Engineers. <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/> [28/05/2022]
- [28] Redsen Consulting. (2021, 22 septembre). Comprendre la technologie RFID en 6 points. Redsen. <https://www.redsen-consulting.com/transformation-digitale/technologie-rfid-en-6-points/> [05/07/2022]
- [29] Proteus PCB Design and Simulation Software – Introduction. (2017, 8 mars). Electronic Circuits and Diagrams-Electronic Projects and Design.

<https://www.circuitstoday.com/proteus-software-introduction> [29/05/2022]

- [30] Arduino Integrated Development Environment (IDE) v1. (2022, 23 mai). Arduino Documentation. <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/> [30/05/2022]
- [31] A quoi sert Firebase, la plateforme mobile de Google ? (2020, 28 avril). Junto. <https://junto.fr/blog/firebase/> [01/06/2022]
- [32] The Good and the Bad of Firebase Backend Services. (2020, 18 février). AltexSoft. <https://www.altexsoft.com/blog/firebase-review-pros-cons-alternatives/> [01/06/2022]

Annexe :



