



SFL 1 - Gestion d'accès parking

18.01.2021

Contexte et définition du problème :	2
Présentation de l'entreprise	2
Le problème	2
Objectif du projet :	3
Exigences du projet :	3
Diagramme d'exigence :	4
Analyse :	5
Diagramme de cas d'utilisation	5
Diagramme de déploiement	5
Les tâches	6
Répartition des tâches :	6
Présentation des tâches :	7
Etudiant 1 :	7
Etudiant 2 :	8
Etudiant 3 :	12
Contraintes et spécification	13
Contraintes financières	13
Spécification technique	13
Ordinateur embarqué (PLC embarqué):	13
Choix du langage :	14
Capteur RFID :	14
Serveur Linux	14
Application PC d'administration des utilisateurs :	16
Page Web et base de donnée :	18
Système LoRa	19
Autres solutions utilisé pour le développement	20
Ressources disponible	21
Gestion du temps	22
Etudiant 1	23
Etudiant 2	23
Etudiant 3	25
Annexe	26
Glossaire	26

I. Contexte et définition du problème :

a. Présentation de l'entreprise

L'ensemble St Félix LASALLE est un établissement regroupant deux collèges, un lycée général et technique, ainsi qu'un campus. L'établissement propose un large choix de filières en alternance ou en cursus traditionnel. Le groupement existe depuis 1848. L'établissement regroupe 2500 étudiants. Il se situe au cœur de Nantes.



b. Le problème

Dans l'ensemble St Félix LASALLE suite à la fusion des établissements. De nombreux travaux ont été engagés à court et moyen terme. Ces travaux prévoient la construction d'un parking réservé au personnel de l'établissement. La construction de ce parking disposera d'un portail motorisé mais aucune solution de commande et de gestion n'a été prévue.

Ce projet est proposé par le directeur du lycée Saint Félix LaSALLE Monsieur David Bourguin.

II. Objectif du projet :

Le but de ce projet est de réaliser la gestion d'accès et d'occupation de ce parking.

En effet, la solution à développer devrait nous permettre de gérer l'accès au parking en plus de connaître le nombre de places disponibles.

Cela signifie que :

- Si aucune place n'est libre, le panneau d'affichage nous indique que le parking est complet et la barrière ne doit donc pas s'ouvrir.
- S'il reste des places, alors, l'accès se fait par badge et l'affichage indique alors le nombre de places disponibles.

Une interface de gestion des utilisateurs est aussi demandée.

III. Exigences du projet :

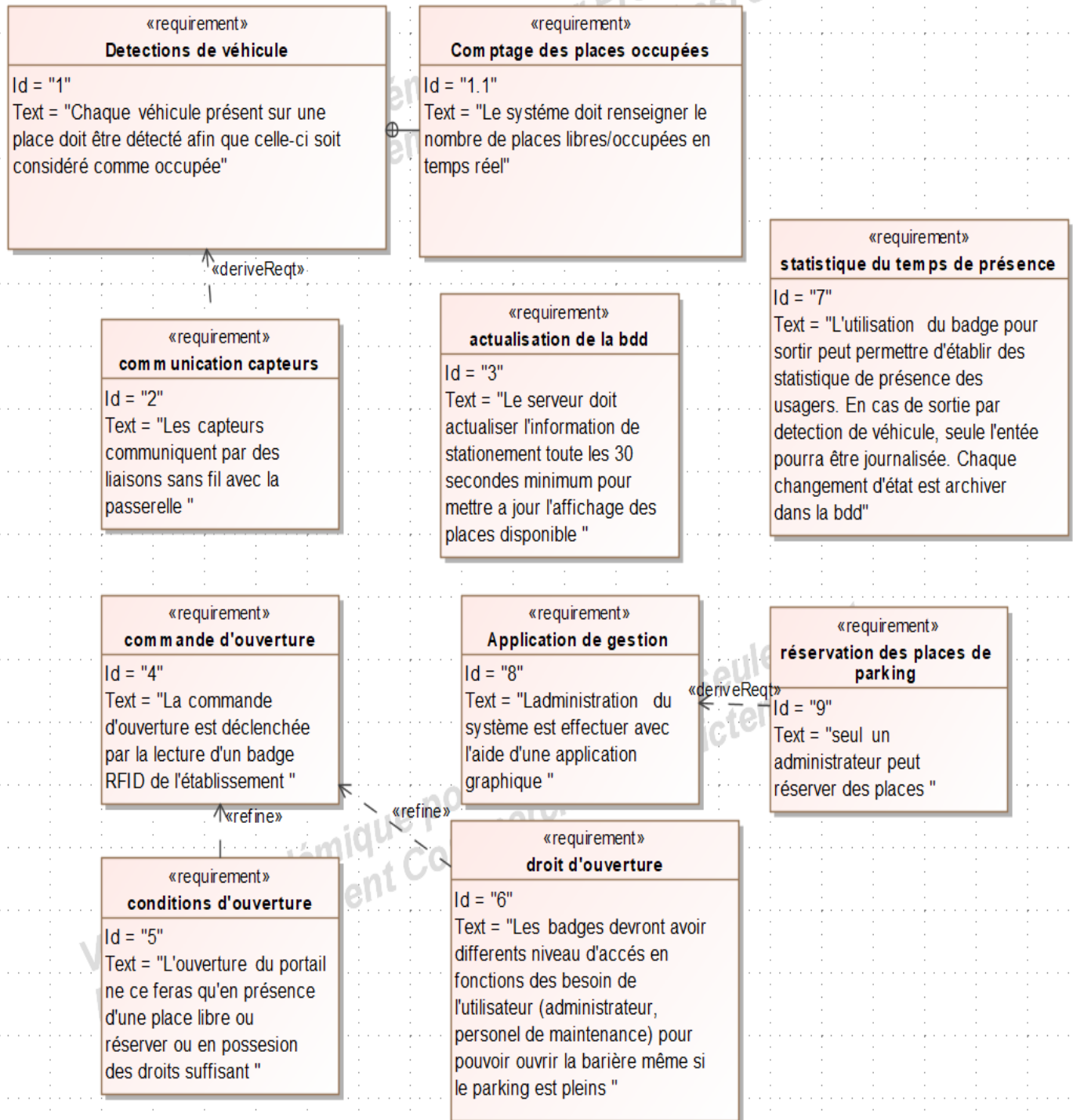
Pour le projet, l'une des exigences est de réaliser la détection de véhicule. Chaque voiture garée sur une place doit être détectée. Ce procédé est réalisé via la communication de capteurs qui communiquent par liaison sans fil.

La place se doit donc d'être considérée comme "occupée". Le système doit connaître le nombre de places libres et occupées en temps réel.

Le système doit pouvoir ouvrir la barrière du portail via la lecture d'un badge RFID de l'établissement.

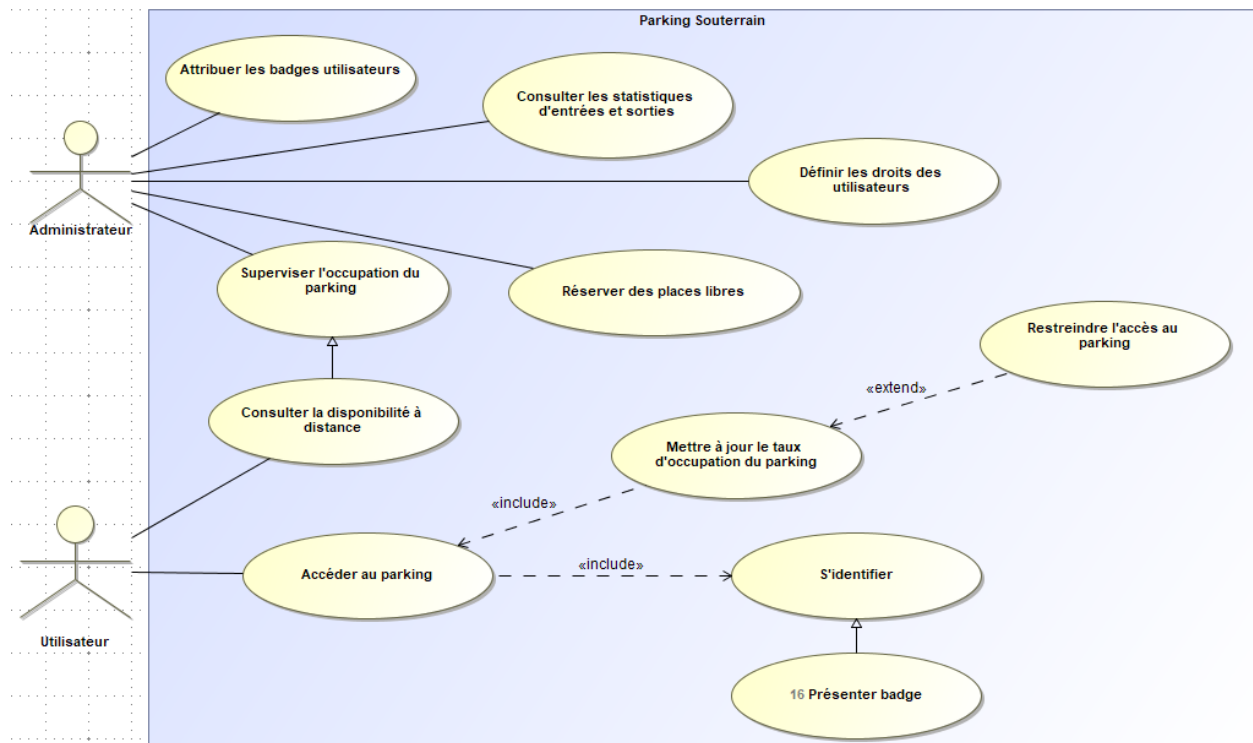
L'une des demandes de Monsieur David Bourgoïn est d'avoir des données statistiques.

Diagramme d'exigence :

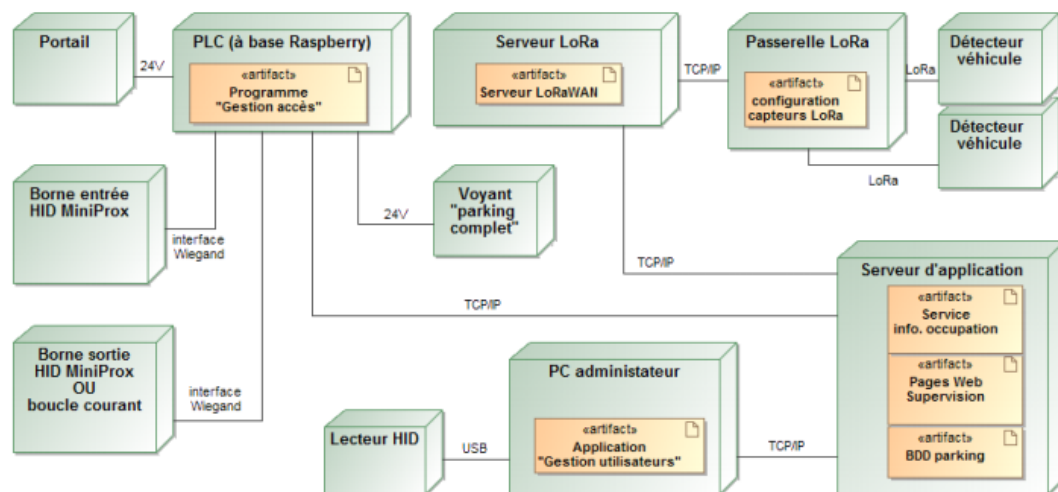


IV. Analyse :

a. Diagramme de cas d'utilisation



b. Diagramme de déploiement



V. Les tâches

Répartition des tâches :

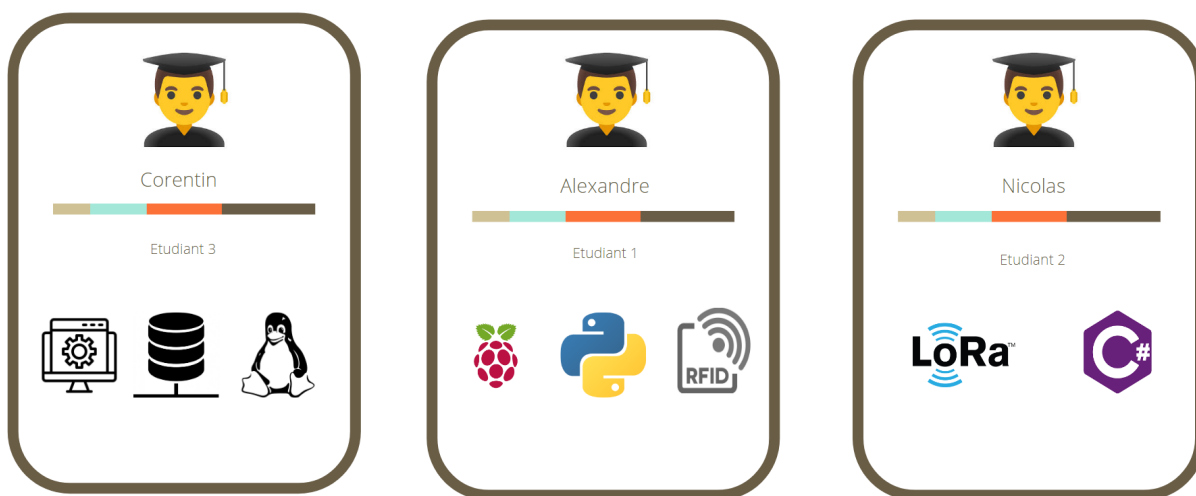


Schéma de la répartition des tâches

Présentation des tâches :

Etudiant 1 :

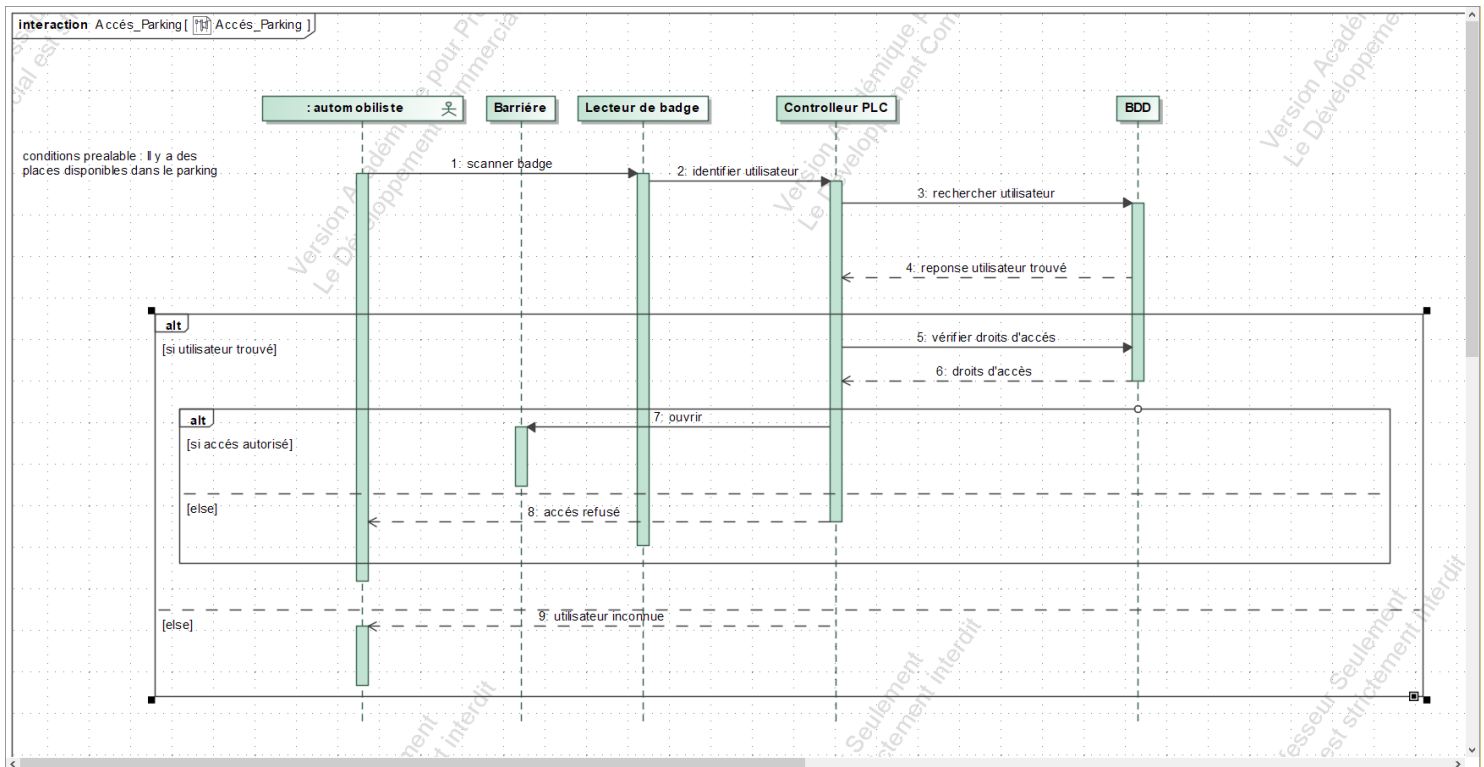
L'étudiant 1 devra installer un OS sur la PLC Raspberry pour programmer en Python afin de commander :

- L'ouverture de la barrière.
- Un voyant pour signaler que le parking est plein.
- Lecture des badges RFID.

Le programme permettra de communiquer avec la base de données pour récupérer certaines informations :

- Taux d'occupation du parking,
- Identification de l'utilisateur.

Voici le diagramme de séquence qui correspond à cette partie :



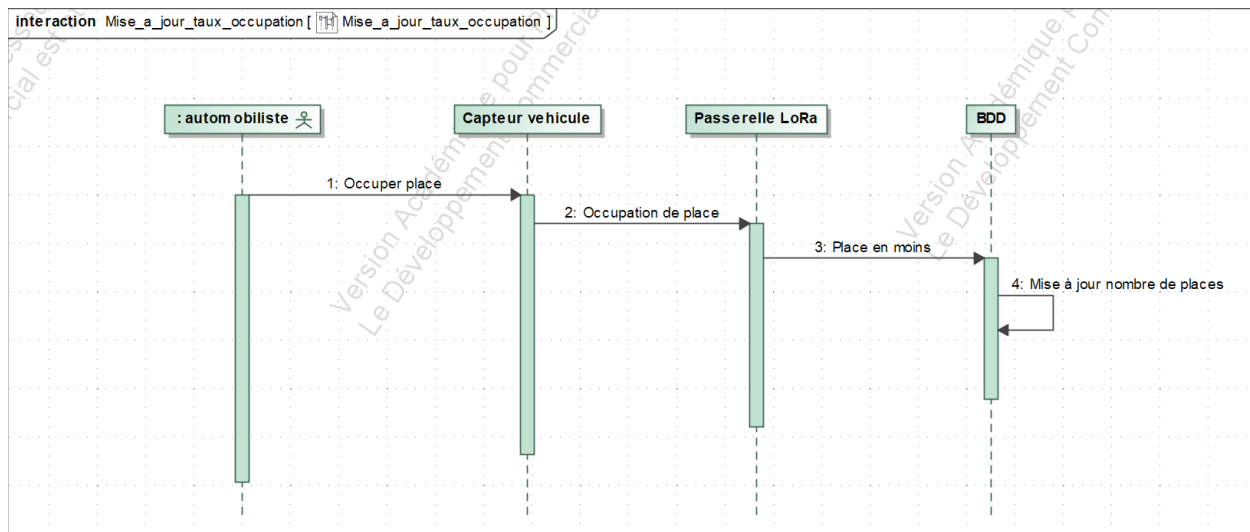
Etudiant 2 :

L'étudiant 2 sera chargé dans un premier temps de l'installation et la configuration du réseau LoRa ainsi que le développement du serveur d'application.

Ce qui comprend :

- La configuration de la passerelle LoRa et du serveur LoRaWAN
- L'Appairage des capteurs véhicules et l'intégration dans le serveur
- La configuration du serveur LoRaWAN pour renvoyer les données.
- Le développement du serveur d'application qui comprend le traitement des données reçu au format JSON et la mise en base de données

Voici le diagramme de séquence qui correspond à cette partie :



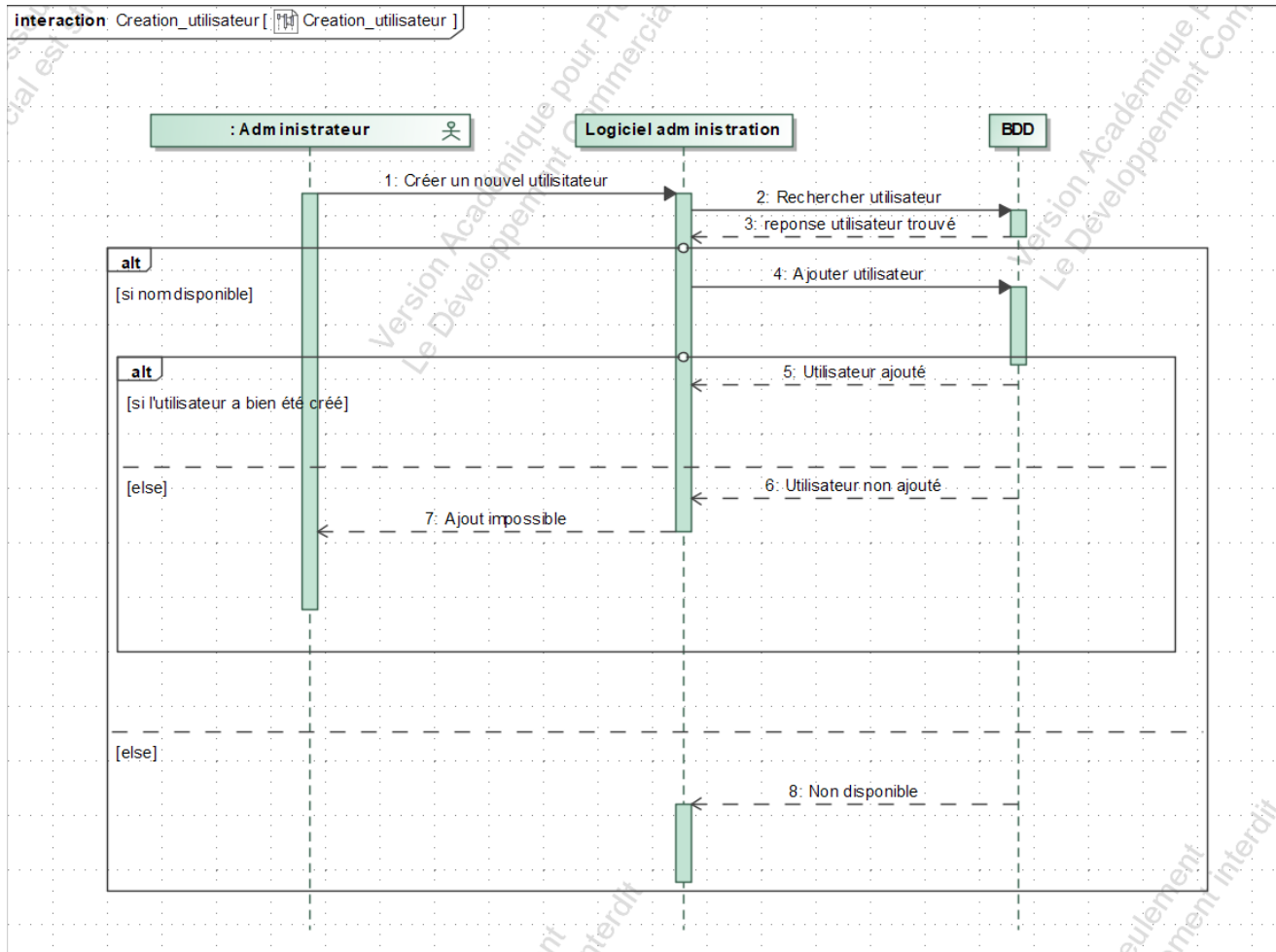
Dans un second temps, il sera chargé du développement d'une application pour gérer l'interface d'administration des usagers du parking en C#.

Ce qui comprend :

- **L'attribution des badges**

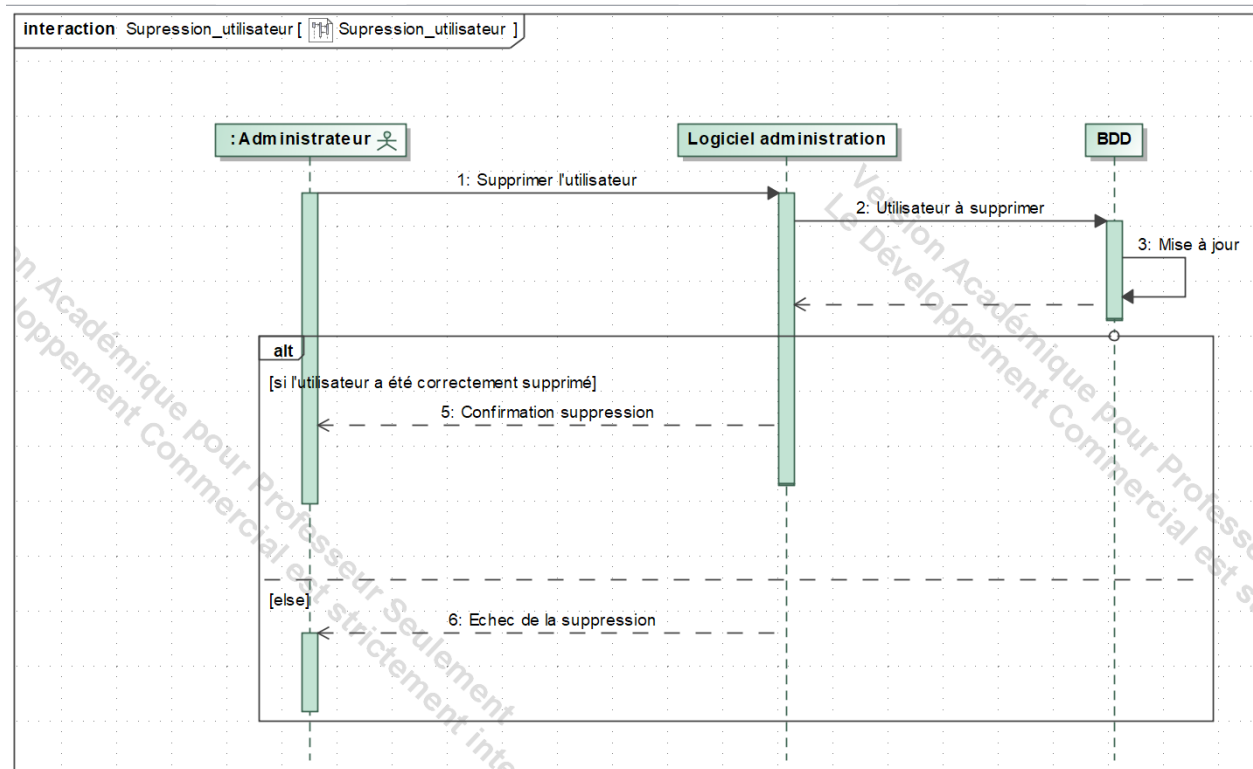
- La création d'utilisateurs

Voici le diagramme de séquence qui correspond à la création :



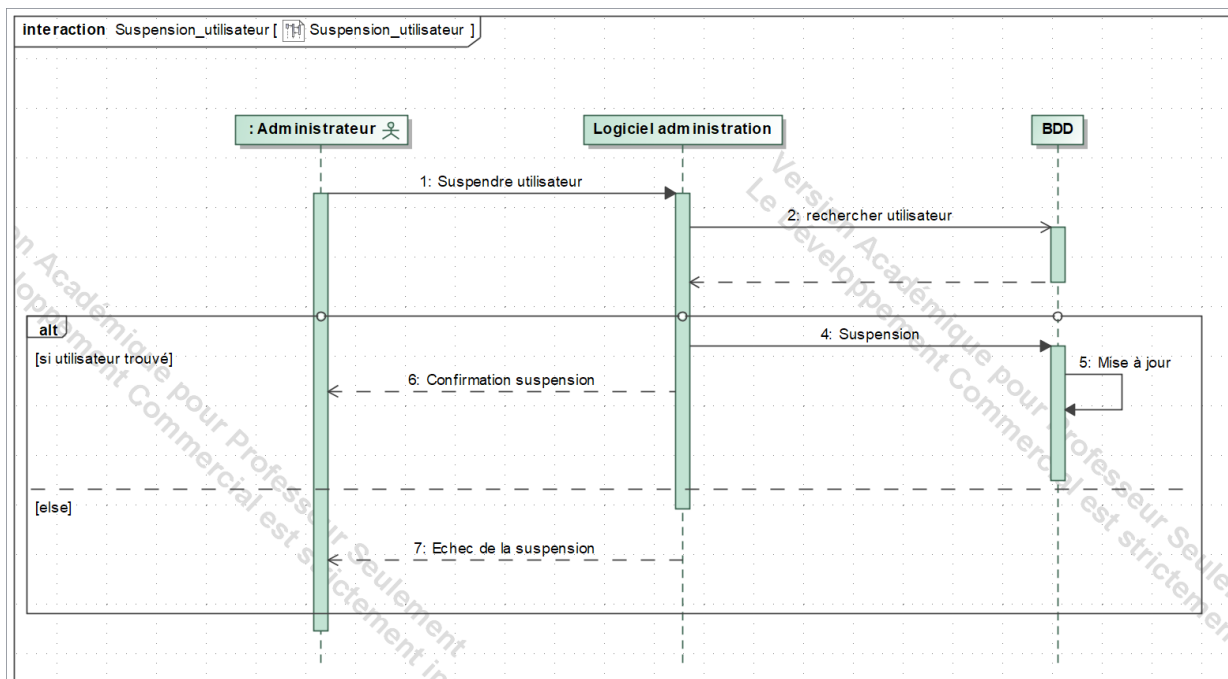
- La suppression d'utilisateur

Voici le diagramme de séquence qui correspond à la suppression :



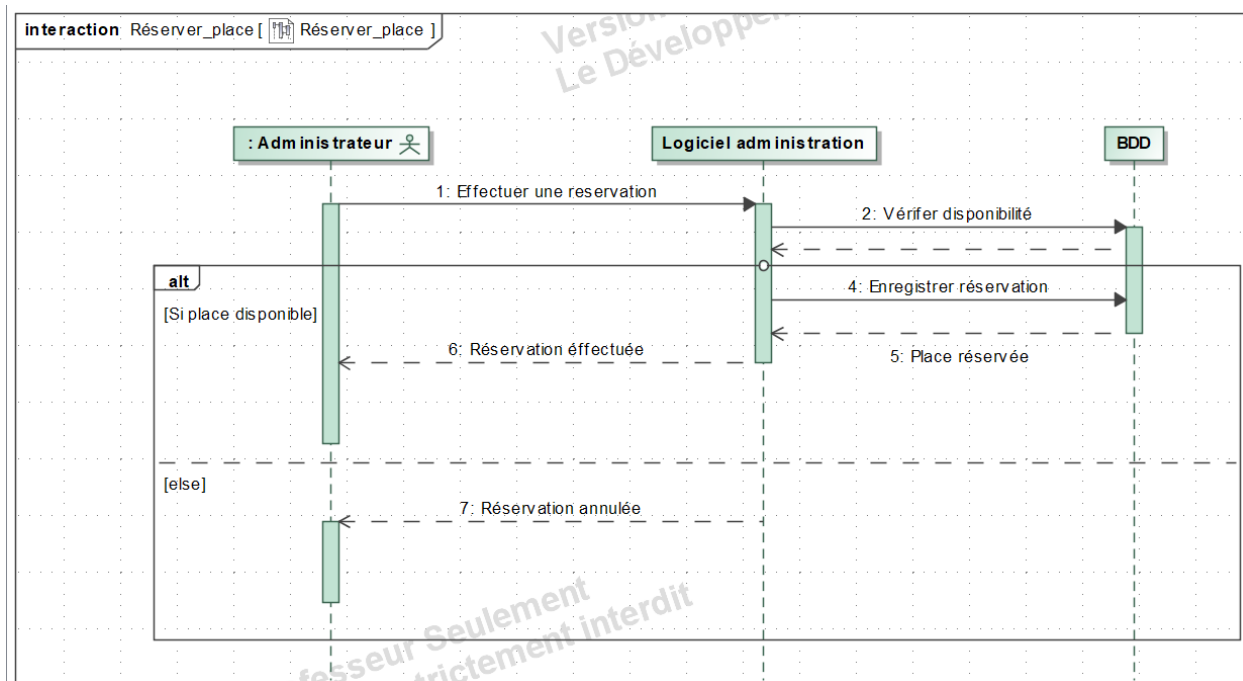
- La suspension d'utilisateur

Voici le diagramme de séquence qui correspond à la suspension :



- La réservation de place de parking

Voici le diagramme de séquence qui correspond à cette partie :



Etudiant 3 :

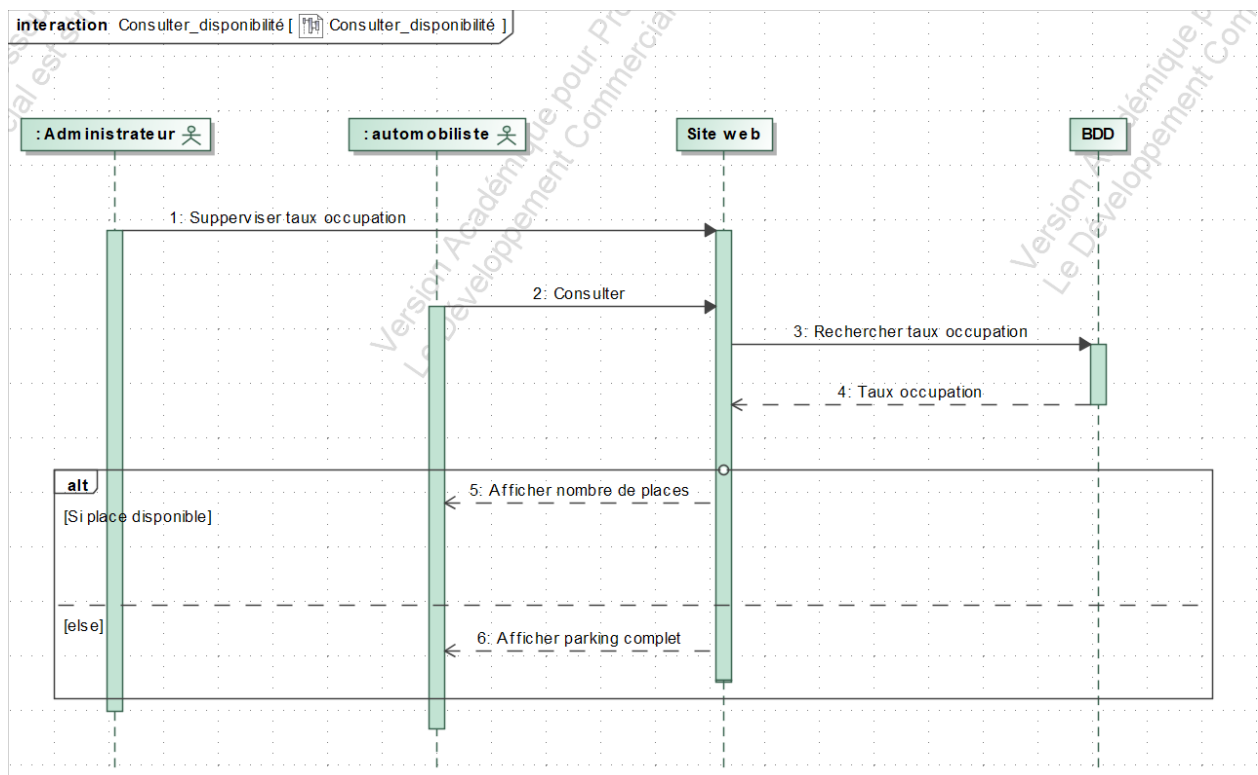
L'étudiant 3 sera chargé dans un premier temps de chercher et d'installer une distribution Linux sur laquelle pourra être installée une version d'Apache/MySQL.

L'étudiant 3 sera également chargé de la base de données ce qui comprend :

- La création du modèle entité-association de la base de données,
- Le déploiement de la base de données sur le serveur.

Enfin il sera chargé de développer une page web dynamique qui aura pour but de superviser, gérer le parking en temps réel ainsi que de récupérer le taux d'occupation.

Voici le diagramme de séquence qui correspond à cette partie :



VI. Contraintes et spécification

a. Contraintes financières

Nous avons 2000€ à disposition pour la réalisation de ce projet. Ce budget est donné par le commanditaire, l'établissement Saint Félix LASALLE.

Ce budget permettra l'achat de matériel nécessaire au projet comme :

- Une passerelle LoRa
- Trois capteurs d'occupation
- Une raspberry PLC PI

b. Spécification technique

1. Ordinateur embarqué (PLC embarqué):

Nous avons besoin d'une carte Raspberry PLC pour commander le portail, afficher la disponibilité du parking, gérer une led ainsi que gérer un lecteur de badge RFID.

Après plusieurs recherches, nous avons trouvé la Raspberry "RevPi Compact" :

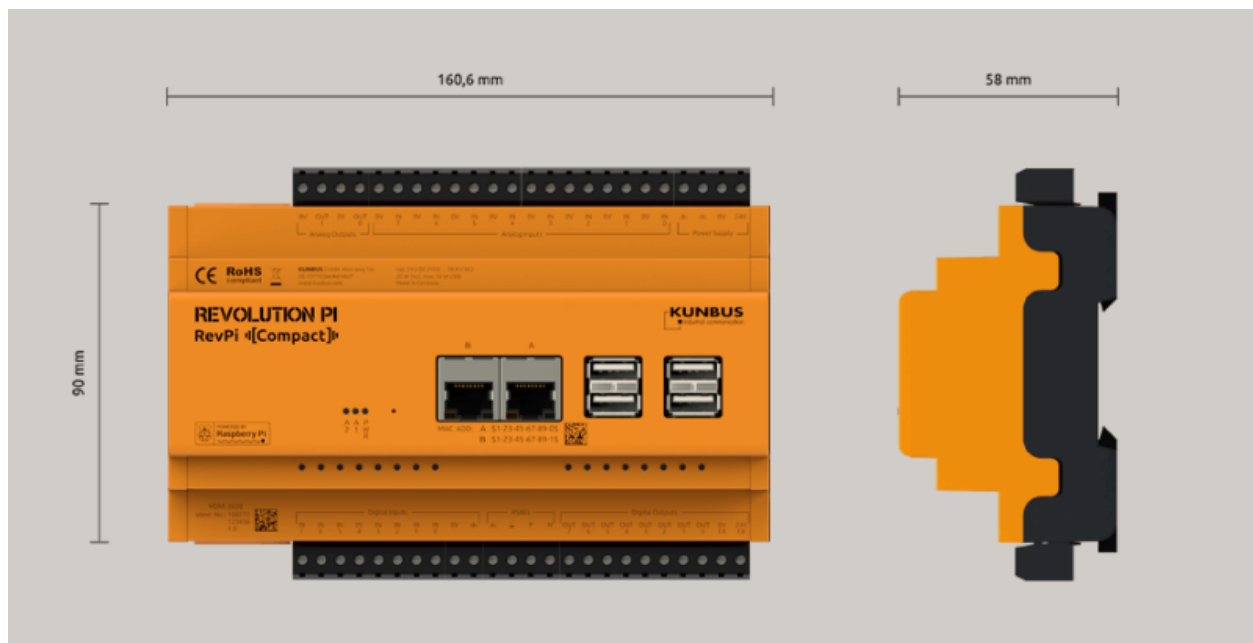


schéma de la ResPI Compact

Caractéristiques :

- 24 V
- 1x HDMI socket
- 1x RS485
- 1x ModBus protocole

a. Choix du langage :

Afin de développer la commande du portail, l'affichage de la disponibilité du parking, gérer une led et un lecteur de badge RFID, nous avons besoin de choisir un langage de développement adapté à nos besoins et à la Raspberry. Afin de choisir correctement, nous avons réalisé une fiche d'évaluation.

	Connaissances des élèves (0.5)	Aide possible du professeur (0.1)	Adapté à nos attentes (0.4)	Total : (/1)
C++	1	0.9	0.5	0.79
Java	0.8	1	0.3	0.62
Python	0.65	0.9	1	0.81

Nous avons donc décidé de partir sur un développement en Python pour la partie liée à la Raspberry.

b. Capteur RFID :

Nous avons besoin d'un capteur avec la technologie RFID :

Après plusieurs recherches nous avons trouvé le capteur OsiSense XG - XGCS850C201



Caractéristiques :

- Tension d'alimentation : 24 V
- Protocole de communication : Ethernet IP/Modbus TCP
- Port de communication : Ethernet

- Type d'étiquettes à associer : ISO 14443 / ISO 15693

2. Serveur Linux

Nous avons besoin d'un serveur sous Linux. Pour choisir la meilleure distribution Linux, nous avons fait un tableau comparatif des différents OS majeurs disponibles pour un serveur sous linux :

	Développement	Contenue	Stabilité	Communauté	Utilisation	Bibliothèque
Debian	Développement communautaire	Interface en ligne de commande mais possibilité d'avoir une interface graphique	Plus stable que Ubuntu , système considéré comme très stable	Très active et nombreuse, grande facilité pour trouver des réponses aux questions	Debian est une distribution destinée à la base aux serveurs	Bibliothèque un peu moins importante que Ubuntu
Ubuntu	Développement communautaire à partir de Debian	Interface en ligne de commande	Moins stable que Debian	Très active et nombreuse, grande facilité pour trouver des réponses aux questions	Debian est une distribution destinée à la base aux ordinateurs	Bibliothèque importante, accès à beaucoup de nouvelles fonctionnalités
CentOS	Contrairement à Ubuntu et Debian CentOS n'est pas développé par une communauté mais elle est éditée par la société Red Hat	Interface en ligne de commande mais installation d'une interface graphique possible et facile	Stabilité maximale	Très active et nombreuse, grande facilité pour trouver des réponses	CentOS est une distribution destinée à la base aux serveurs	Bibliothèque importante

Grâce à ce tableau comparatif, nous avons réalisé une fiche d'évaluation des différentes distributions Linux.

	Connaissances des élèves (0.7)	Aide possible du professeur (0.2)	Documentation et aide en ligne (0.1)	Total : (/1)
Debian	0.7	0.9	0.95	0.765
Ubuntu	0.65	0.8	0.9	0.705
CentOS	0.1	0.4	0.95	0.245

Après plusieurs recherches, nous avons décidé de partir sur une distribution Debian. En effet, cette solution répond au mieux à nos attentes et connaissances pour la réalisation de ce projet.

3. Application PC d'administration des utilisateurs :

Pour développer l'application de gestion des utilisateurs, nous avons choisi un langage de programmation en fonction de notre besoin. Pour faire un choix du langage, nous avons fait une fiche d'évaluation de chaque langage et de sa librairie graphique la plus importante.

	Connaissances des élèves (0.7)	Aide possible du professeur (0.2)	Documentation et aide en ligne (0.1)	Total : (/1)
Java Swing	0.7	1	0.9	0.85
QT Widget	0.85	1	1	0.895
WPF Visual C#	0.9	1	1	0.93
Python Tkinter	0.2	0.3	0.7	0.27

Avec l'aide de ce tableau, nous avons décidé de choisir de développer cette application en WPF sous Visual Studio en C#.

Nous avons réalisé un prototype de l'IHM de l'application.

Le menu :

Menu

RESERVATION PLACE

UTILISATEUR

Refresh icon

NOM	PRENOM	BADGE
TEST	DUPONT	54500550545

User icons: + and x

La Création d'un utilisateur :

Ajouter utilisateur

Nom :

Prénom :

Badge :

Ajouter Annuler

La Suppression/ suspension d'un utilisateur :

Supression/Suspension

Nom :

Prénom :

☐ Supression ☐ Suspension

4. Page Web et base de données :

Toutes les pages Web seront développées sous l'IDE *NetBeans* en *HTML/PHP/CSS*. Pour faire le choix de l'IDE, nous avons fait une fiche d'évaluation en prenant en compte les connaissances de Corentin et des autres étudiants, l'aide que pouvait lui apporter le professeur et la simplicité de l'interface.

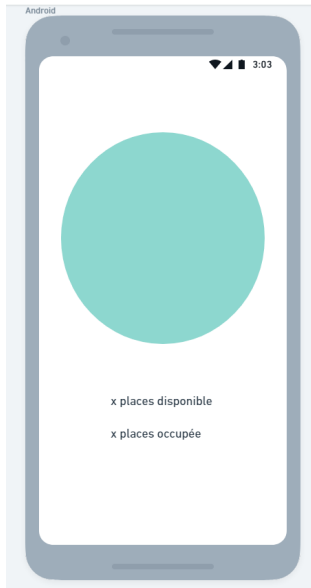
	Connaissance (0.7)	Aide possible du professeur (0.1)	Simplicité de l'interface (0.2)	Total : (/1)
Visual studio	0.6	0.9	0.85	0.68
NetBeans	0.7	1	0.9	0.77
Notepad ++	0.4	0.7	1	0.55

Nous utiliserons le framework **Bootstrap** pour simplifier le développement et rendre les pages adaptables à tous les supports(téléphone, tablette, PC).

Les pages Web et la base de données seront hébergées sur un serveur Apache/MySQL.

Cependant pendant la phase de production, on pourra utiliser **Wampserver** sous Windows pour simplifier le développement.

Nous avons réalisé un prototype de l'IHM du site web sur téléphone.



5. Système LoRa

LoRa est un protocole de communication fonctionnant via des ondes radio. Nous utilisons ce protocole car il permet une communication à longue distance et il offre aussi une consommation très réduite d'énergie.

Le système LoRa que nous allons utiliser est produit par la société Wi6labs. Wi6labs est une société rennaise qui développe des solutions pour un internet des objets performant et durable. Ces solutions s'appuient sur le déploiement de réseau privé LPWAN.

Les capteurs sous les voitures sont des capteurs TBS-220.



Les capteurs TBS-220 communiquent à l'aide du protocole LoRa avec la passerelle. Celle-ci interprète et organise les données reçues avant de les envoyer au serveur LoRa sous forme de paquet IP.

WI6labs fournit le serveur LoRa. Pour récupérer les infos, nous avons utilisé un serveur d'application qui demande les informations au serveur LoRa. Les informations récupérées seront ensuite envoyées à la BDD.

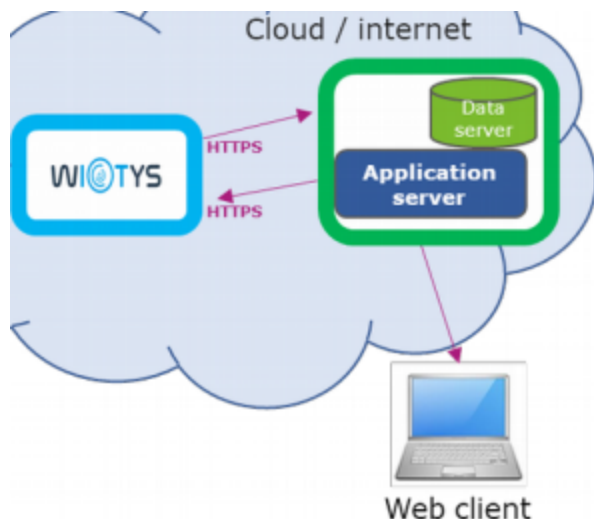


schéma du fonctionnement

6. Autres solutions utilisés pour le développement



GitHub est une solution de partage de code, sur lequel nous pouvons partager les différentes tâches entre chaque étudiant via la création d'un projet. Le dépôt de notre projet s'appelle "Sfl1_Gestion_d'accès_parking" (consultable via ce lien :https://github.com/NicolasDebras/Sfl1_Gestion_d_acces_parking). Chaque étudiant pourra donc accéder à son travail partout grâce à une connexion Internet.



Teams est un outil collaboratif et de communication unifié et polyvalent pour le travail d'équipe. Il propose de nombreux outils qui nous seront utiles au cours de notre projet comme le chat, visioconférence ou l'appel vocal. Cet outil nous permet d'échanger facilement avec notre professeur référent ainsi qu'entre nous. Teams est un outil déjà utilisé et intégré dans notre établissement.

c. Ressources disponible

Pour mener à bien ce projet, nous avons à notre dispositions le matériel suivant :

- Un ordinateur de développement par étudiant,
- Une Raspberry PLC Pi compatible Python,
- un lecteur de badge RFID,
- un ordinateur fourni par l'entreprise pour servir de serveur,
- deux capteurs LoRa pour la détection des voitures,
- Une passerelle LoRaWAN.

VII. Gestion du temps

Pour visualiser la gestion de temps, nous avons réalisé un diagramme de Gantt pour notre projet. Nous mettrons à jour le diagramme dans nos parties personnalisées en fonction de l'avancement du projet.

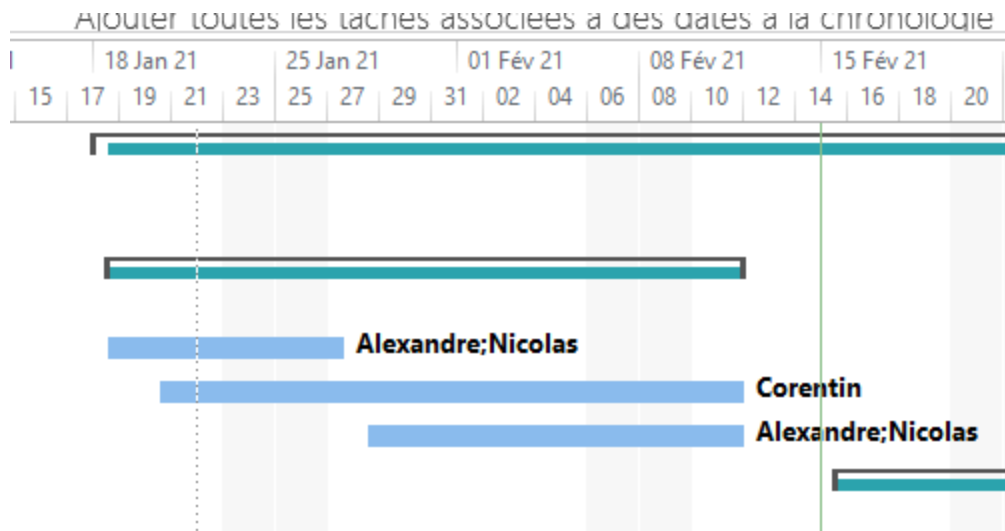
Nous avons réalisé le diagramme de Gantt via le solution MS Project. Ce logiciel nous est imposé par notre professeur.

(voir en annexe le Gantt en entier)

Pour faire de la phase d'analyse, nous avons découpé les tâches en fonction de nos compétences.

1		✦ SFL1 - GANTT	213,5 hr	Lun 18/01/21	Ven 21/05/21		
2							
3		✦ Analyse du projet	49,08 hr	Lun 18/01/21	Jeu 11/02/21		
4		reformulation	18 hr	Lun 18/01/21	Mer 27/01/21	Alexandre;Nicolas	
5		Analyse via diagramme	45 hr	Mer 20/01/21	Jeu 11/02/21	Corentin	
6		Préparation à la revue 2	30 hr	Jeu 28/01/21	Jeu 11/02/21	Alexandre;Nicolas	
7							

Détails des tâches



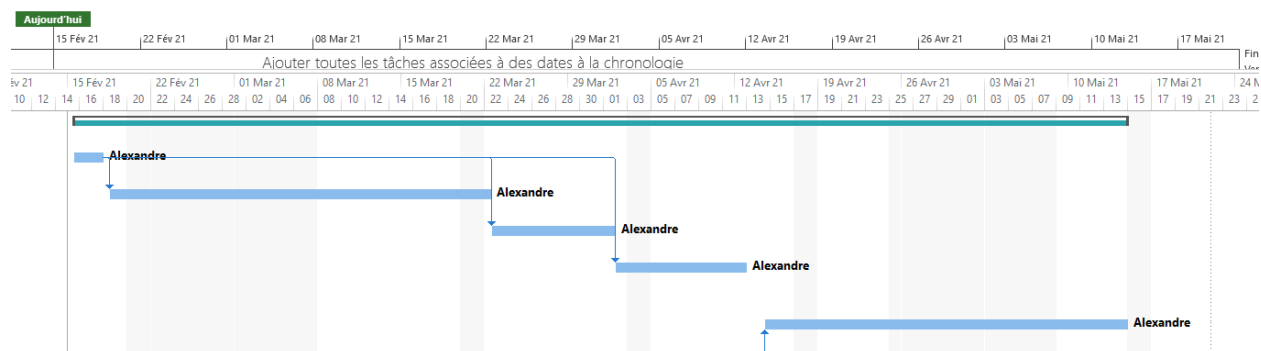
Représentation des tâches dans la durée

Dans notre diagramme de Gantt, nous avons réparti les tâches en fonction de chaque étudiant à partir de la fin de la revue 2 pour la phase de développement.

a. Etudiant 1

Etudiant 1 :	140,58 hr	Lun 15/02/21	Ven 14/05/21		
Installation d'un OS pour le PLC Raspberry	8 hr	Lun 15/02/21	Mer 17/02/21	Alexandre	
Lecture des badges RFID	38 hr	Jeu 18/02/21	Lun 22/03/21	Alexandre	8
Commande d'ouverture du portail	25 hr	Lun 22/03/21	Jeu 01/04/21	Alexandre	8
Commande du voyant "Parking plein"	22 hr	Jeu 01/04/21	Lun 12/04/21	Alexandre	8
redaction du rapport de projet					
Communication avec le serveur d'application (taux d'occupation, identification utilisateur)	41 hr	Mer 14/04/21	Ven 14/05/21	Alexandre	22
Etudiant 2 :	446 hr	Lun	Mer		

Détails des tâches

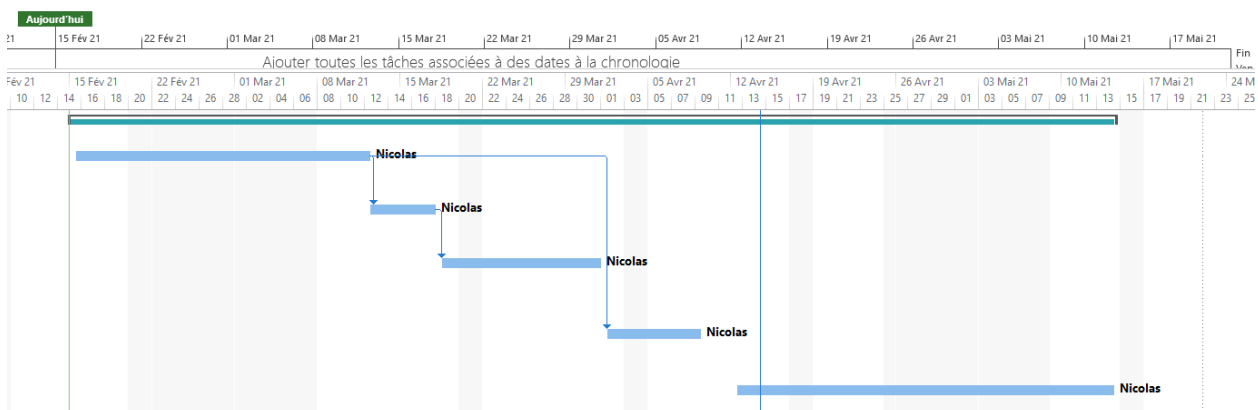


Représentation des tâches dans la durées

b. Etudiant 2

Etudiant 2 :	146 hr	Lun 15/02/21	Ven 14/05/21		
Configuration de la passerelle LoRaet du serveur LoRaWAN pour le contexte du projet	30 hr	Lun 15/02/21	Ven 12/03/21	Nicolas	
Appairage des capteurs véhicules, intégration dans le serveur	8 hr	Ven 12/03/21	Mer 17/03/21	Nicolas	15
Configuration du serveur LoRaWAN pour renvoyer les données des capteurs vers le serveur d'application	30 hr	Jeu 18/03/21	Mer 31/03/21	Nicolas	16
Développement du serveur d'application	22 hr	Jeu 01/04/21	Ven 09/04/21	Nicolas	15
Rédaction du rapport de projet					
Développement de l'interface admin des usagers du parking	45 hr	Lun 12/04/21	Ven 14/05/21	Nicolas	

Détails des tâches



Représentation des tâches dans la durées

♣ Etudiant 3 :	146,08 hr	Lun 15/02/21	Ven 14/05/21		
Installation d'un serveur(distribution Linux avec Apache/MySQL)	15 hr	Lun 15/02/21	Ven 19/02/21	Corentin	
Création de la base de données parking	12 hr	Lun 08/03/21	Jeu 11/03/21	Corentin	22
Déploiement de la BDD sur le serveur	15 hr	Ven 12/03/21	Jeu 18/03/21	Corentin	23
♣ Développement des pages Web de supervisiondu parking	99,5 hr	Ven 19/03/21	Ven 14/05/21		23
HTML	18,17 hr	Ven 19/03/2	Ven 26/03/2	Corentin	
Partie dynamique	34 hr	Lun 29/03/21	Lun 12/04/21	Corentin	26
gestion de place de parking	41,25 hr	Mer 14/04/2	Ven 14/05/2	Corentin	

The screenshot shows a Gantt chart interface with a timeline from February 15, 2021, to May 17, 2021. The chart displays several tasks for 'Corentin' as blue bars. The first task is a long bar from Feb 15 to Apr 12. A second task starts on Apr 12 and ends on May 15. A third task starts on May 15 and ends on May 23. There are also some shorter tasks and sub-tasks indicated by arrows and smaller bars. The interface includes a date picker at the top and a task list on the right.

Représentation des tâches dans la durées

VIII. Annexe

a. Glossaire

Technologie RFID :

- La radio-identification, le plus souvent désignée par le sigle RFID, est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés "Radio-étiquettes".

Technologie LoRa et protocole LoRaWAN :

- LoRaWAN est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa et connectés à l'internet via des passerelles, participant ainsi à l'internet des objets.

WampServer :

- Wampserver est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts en PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs, un interpréteur de script, ainsi que PhpMyAdmin pour l'administration Web des bases MySQL.

IDE :

- Environnement de développement.

MS Project :

- Microsoft Project est un logiciel de gestion de projet. Il permet au chef de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets.

MagicDraw :

- MagicDraw est un outil de modélisation visuel UML, SysML.

Netbeans :

- NetBeans est un environnement de développement intégré, placé en open source depuis 2000. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres par l'ajout de greffons.

Raspberry :

- Le Raspberry Pi est un nano-ordinateur monocarte à processeur ARM de la taille d'une carte de crédit conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi.

Linux :

- Linux ou GNU/Linux est une famille de systèmes d'exploitation open source de type Unix fondé sur le noyau Linux, créé en 1991 par Linus Torvalds. De nombreuses distributions Linux ont depuis vu le jour et constituent un important vecteur de popularisation du mouvement du logiciel libre.

Serveur Apache :

- Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est maintenu et développé par Apache Software Foundation.

Bootstrap :

- Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs.

OS :

- Système d'exploitation

IHM :

- Interfaces homme-machine

BDD :

- acronyme signifiant base de données