

# PROJET TECHNIQUE LA BORNE CONNECTEE



Baptiste POCHEBONNE

Fénelon Sup

9 Boulevard de Courcelles

Paris, 75008



# I. TABLE DES MATIERES

- II. Introduction
- III. Membres du Projet
- IV. Cahier des charges
  - a. Contexte
  - b. Public visé
  - c. Contraintes
    - i. Contraintes générales
    - ii. Contraintes techniques
  - d. Objectif final
- V. Organisation des taches
  - a. Gantt
- VI. Diagrammes
- VII. Analyses des besoins matériels
  - a. Choix de la Raspberry
  - b. Choix de l'imprimante thermique
  - c. Choix de l'écran
- VIII. Développement de l'interface Web
- IX. Réalisation de mes taches au sein du projet
  - a. Création de la base de données
    - i. Gestion des différentes tables.
      - 1. Table User
      - 2. Table Comptes
      - 3. Table Ticket
    - ii. Intégration de la base de données sur les serveurs MAMP & OVH.
    - iii. Lier la base de données au projet
  - b. Intégration et réalisation du Text To Speech
- X. Difficultés rencontrées
- XI. Bilan du projet
- XII. Conclusion
- XIII. Annexes
- XIV. Glossaire

## II. INTRODUCTION

Dans le cadre de mon BTS en Système Numérique en Informatique et Réseau, j'ai dû réaliser à l'aide de mon équipe une borne interactive qui devait répondre au besoin de la CAF.

Ce projet nous imposait de faciliter l'accès à la CAF à l'aide d'impressions de billet tous en étant totalement autonome.

Il était également demandé que cette borne puisse être accessible à tous dans le but d'aider les personnes en difficultés telle que les personnes à visibilité réduite.

Nous nous sommes donc lancés dans le développement de ce projet ou nous avons eu 5 mois pour le réaliser.

Nous avons donc choisi de créer ce projet en plusieurs parties, une interface utilisateur permettant de lier les besoins du client à la CAF tout deux liés à l'aide d'une base de données.

Nous avons donc par la suite commandé puis utilisé tout le matériel nécessaire pour monter cette borne.

Ce projet nous a été proposé par monsieur Ahmed Gabal professeur de développement informatique ainsi qu'avec l'aide de notre professeur de Mathématique monsieur Jacques Lerebours.

## III. Membres du projet

Ce projet nous demandait d'être en groupe de 3 à 4 personnes par groupes. Nous avons également la liberté de choisir la constitution de groupes.

Nous avons donc choisi de former un groupe constitué de :

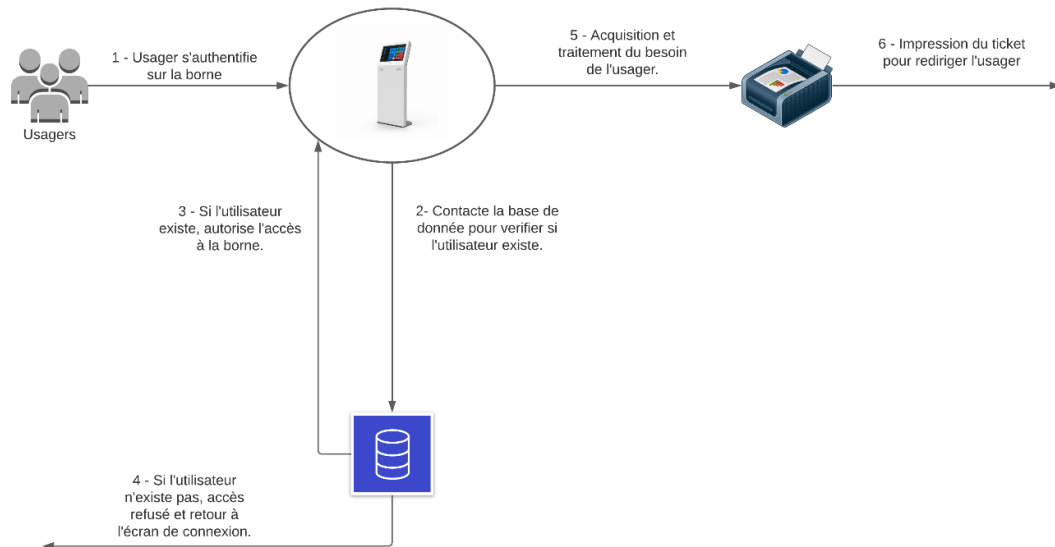
- Maxime DUHAMEL
- Théo FRANCOIS
- Nilson FERGE
- Baptiste POCHEBONNE

## IV. Cahier des charges

### a. Contexte

Notre projet consiste à créer de toute pièce une borne interactive pour impression de ticket de file d'attente pour l'accueil d'une administration publique. Sur cette borne, l'écran devra permettre de déterminer le motif de sa visite, de remplir ses informations personnelles afin de se faire orienter vers le guichet adéquat, munis d'un ticket avec un numéro de passage.

Cette borne doit être accessible aux personnes à mobilité réduites, aux malvoyants et personnes souffrant de tout autre handicap nécessitant la possibilité d'une interaction autre que par le touché avec la borne. Une commande vocale est donc nécessaire.



Nous avons donc choisi de créer une interface graphique qui permettrait de renseigner le client ou lui faire compléter un formulaire. Cette interface serait directement affichée sur la borne qui enverrait directement les données vers une base de données pour que les guichets de la CAF puissent voir quels besoins puissent être renseignés au client.

Le client a donc besoin d'un compte pour pouvoir interagir avec la CAF grâce à la base de données en s'authentifiant.

Une fois l'ensemble renseigné, l'imprimante rentre en action et imprime un ticket indiquant le guichet qui concerne le client.

## **b. Public concerné par cette borne**

Cette borne est destinée à tous les usagers de l'administration publique dans laquelle elle sera installée.

## c. Contraintes du projet

### i. Contraintes générales :

Notre borne, devant être placée dans une administration publique doit être accessible à tout sorte d'utilisateur. Sa taille ne doit pas être excessive, ainsi que sa largeur. Mais également accessible aux personnes malvoyantes ou peu à l'aise avec une interface. Elle doit être capable d'identifier le besoin de l'utilisateur, de le renseigner si besoin puis de lui fournir un ticket de file d'attente avec un guichet correspondant.

Nous avons aussi une contrainte budgétaire. Nous avons besoin que notre borne soit peu coûteuse mais sans avoir à impacter le matériel choisi. En plus de cela, une contrainte de temps nous est posée car pour clôturer ce projet, plusieurs oraux répartis sur la durée du projet sont placés afin de vérifier l'avancement de celui-ci, ainsi qu'une soutenance finale devant un jury technique.

### ii. Contraintes techniques :

Notre dispositif doit comprendre trois modules backend :

- Gestion de file d'attente
- Gestion de base de données
- Intégration de l'assistant vocale

Nous avons décidé de créer une base de données en MySQL servant à stocker les heures de visites pour en faire des statistiques affins de réinitialiser chaque jour le numéro d'impressions de tickets.

Un frontend servant d'interface devant être développé en HTML, CSS, Javascript et PHP.

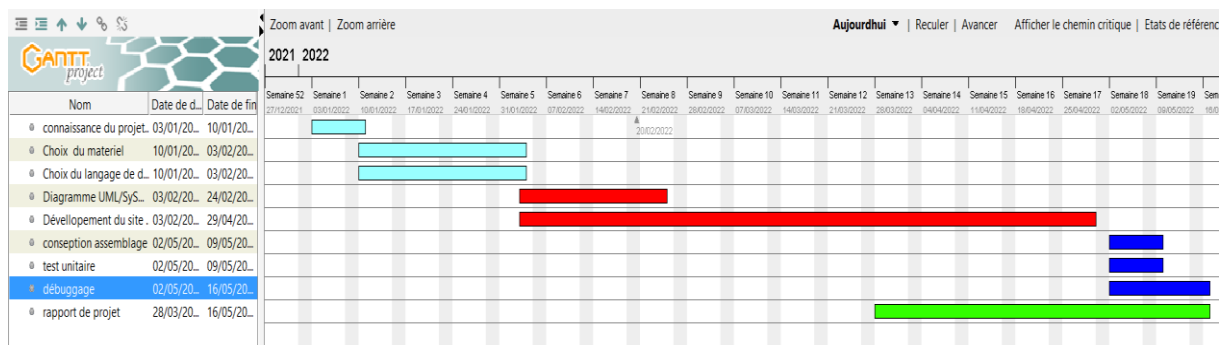
## d. Objectif final :

Le principal objectif de ce projet est de reproduire conformément à ce qui est demandé une borne qui serait à la fois simple d'utilisation mais également simple de compréhension avec l'aide d'une interface Web procédant toutes les informations nécessaires pour orienter l'utilisateur.

## V. Organisations et attributions des taches

Nous avons choisi de créer un planning afin de suivre un programme nous dictant notre avancer dans ce projet.

### a. Diagramme GANTT



Un diagramme de Gantt est un outil pratique pour planifier des projets. Grâce à une vue d'ensemble des tâches planifiées, chaque personne concernée sait quelle activité doit être effectuée et à quelle date précise. Il montre les dates de début et de fin d'un projet.

Pour faciliter notre travail, nous avons décidé de nous répartir les tâches afin d'être le plus productif possible. Voici la répartition finale :

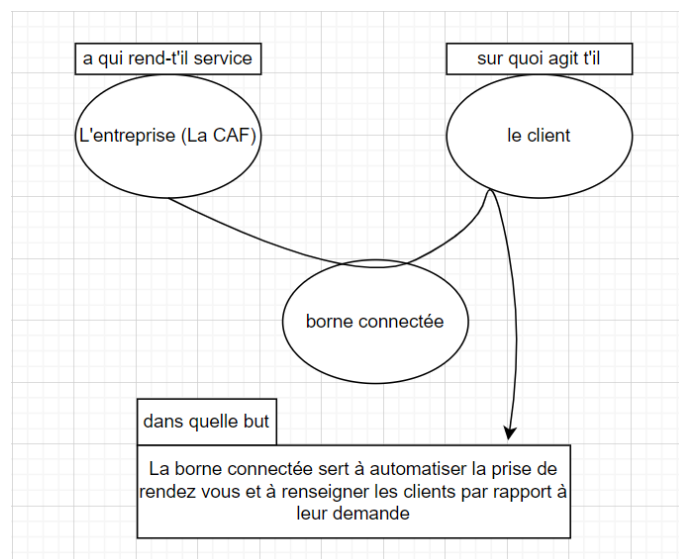
- Théo FRANCOIS (Chef de projet) :
  - Concepteur de l'interface utilisateur et login
  - Assistant de la conception de l'assistance vocal.
  
- Maxime DUHAMEL :
  - Assistant de l'interface utilisateur
  - Concepteur de la gestion du code pour l'imprimante
  
- Nilson FERGE :
  - Dirigeant du matériel (Imprimante ; Raspberry Pi etc.)
  - Conceptions intégration de l'imprimante de ticket avec la Raspberry
  - Assistant de l'interface login.
  
- Baptiste POCHEBONNE
  - Concepteur des diagrammes UML/SySML
  - Concepteur de la base de données
  - Assistant de la conception de l'assistant vocal.



## VI. Diagrammes

*PS : La majorité des diagrammes a été conçus avant l'intégration de la base de données mais aussi de l'assistant vocal qui ne seront donc pas visible dans certains diagrammes mais seront pris en compte sur l'explication des diagrammes.*

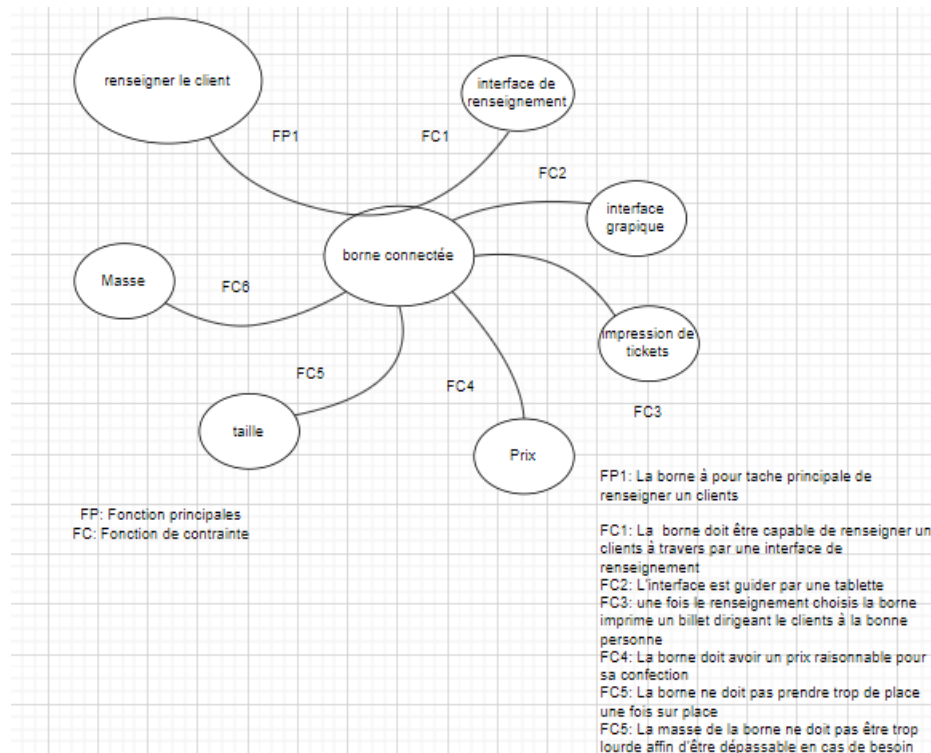
### a. Diagrammes bêtes à corne



Le diagramme bête à corne permet de s'interroger sur l'existence et la fonction du produit.

Les différentes questions principales sont « sur quoi agit-il ? » ; « à qui rend-il service ? » ou encore « dans quel but ? ».

## b. Diagrammes poulpes



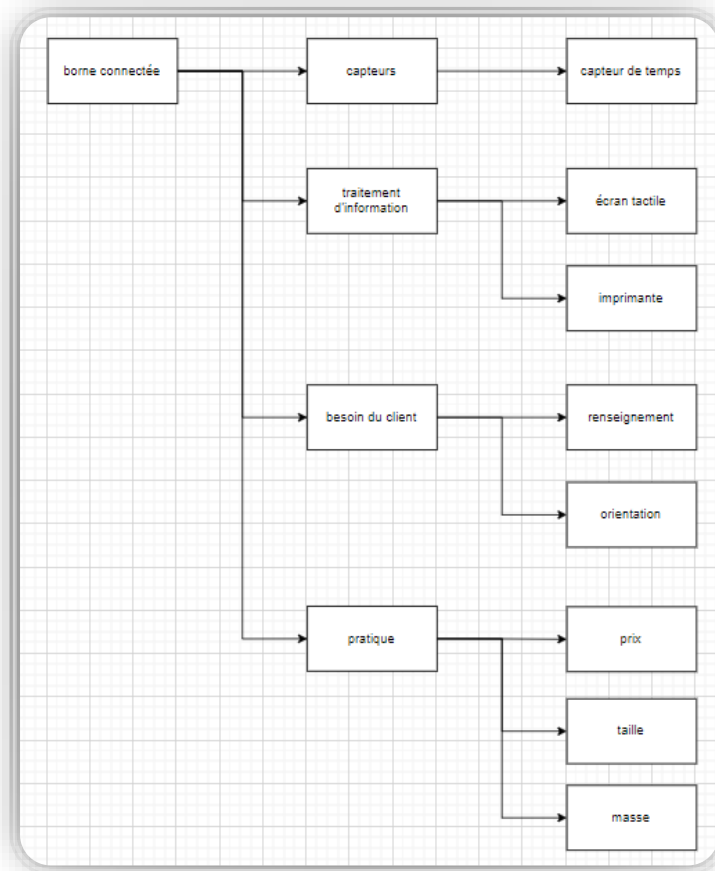
Le diagramme poulpe permet de mettre en lumière l'ensemble des fonctions principales ou de contraintes imposées par le projet.

En effet la borne doit en priorité renseigner l'utilisateur grâce à son interface Web, cette borne doit donc également faire une impression de ticket facilitant les différents guichets.

Le prix ; la taille mais aussi masse doivent permettre un accès plus facile à toute personne à visibilité ou mobilité réduite.

*PS : Ce diagramme a été conçu avant l'intégration de la base de données mais aussi de l'assistant vocal qui sont tous deux des fonctions de contraintes.*

### c. Diagrammes FAST

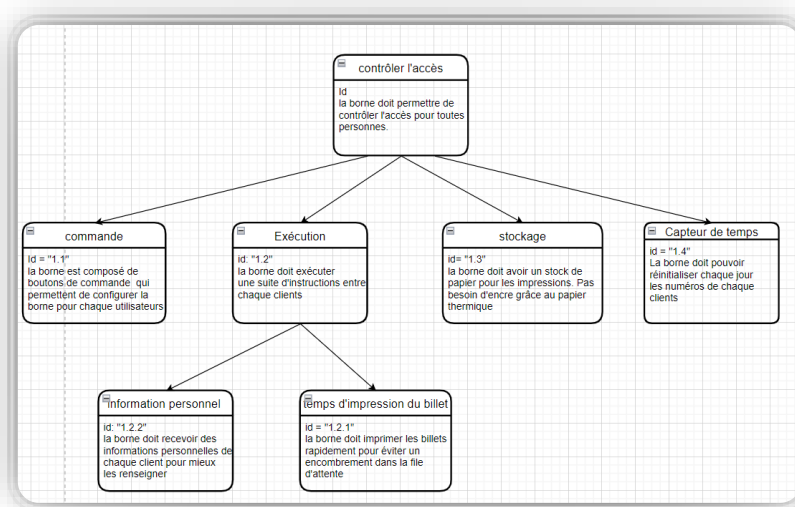


Le diagramme FAST divise en 4 parties la borne connectée :

- La première partie concerne la prise en compte du capteur temps géré par la base de données permettant de réinitialiser le numéro du ticket à chaque jour afin d'éviter une incrémentation vers des nombres infiniment élevés
- La seconde partie concerne l'aspect traitement d'informations géré par l'écran tactile et l'imprimante mais aussi la base de données
- Le besoin du client est un élément à prendre en considération pour l'existence de ce projet en orientant le client ; en renseignant ou encore en utilisant l'assistant vocale.

- Afin de faciliter l'accès de la borne pour les personnes souffrant d'un handicap, la borne doit se montrer pratique, de masse légère, de taille moyenne est d'un prix abordable pour pouvoir réaliser se projet sans trop de difficultés.

## d. Diagrammes de définitions de bloc



Le diagramme de définitions de bloc permet de différencier 4 aspects du produit. Dans celui de la borne interactive, la mission principales est de contrôler l'accès des utilisateurs au différents guichet de la CAF. C'est pourquoi cette mission est séparer en 4 blocs.

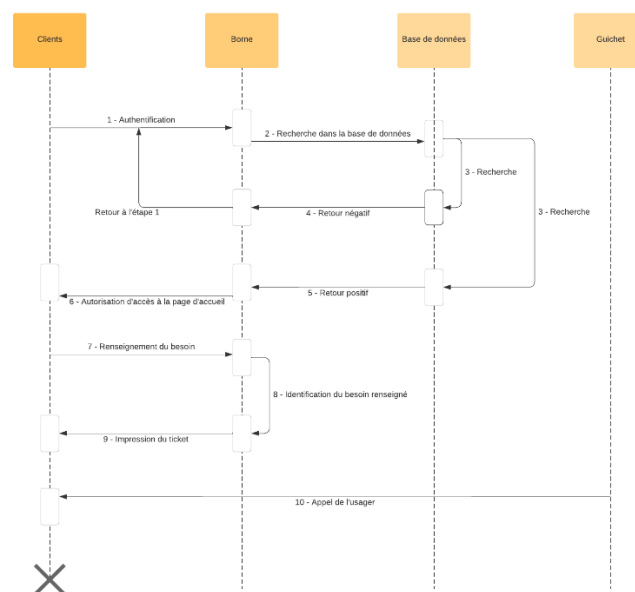
Le premier explique l'utilisation de bouton pour l'utilisation de la borne qui seront directement intégrer sur le site internet

Le second est une contrainte d'exécutions, celle-ci à un temps d'exécutions entre la borne est l'utilisateur ou encore entre la borne et la base de données. Cette contrainte est également existante lors du temps d'impressions de billet.

Le stockage est également un élément à prendre en compte, il est important de prévoir de quoi stocker du papier pour l'impressions de ticket.

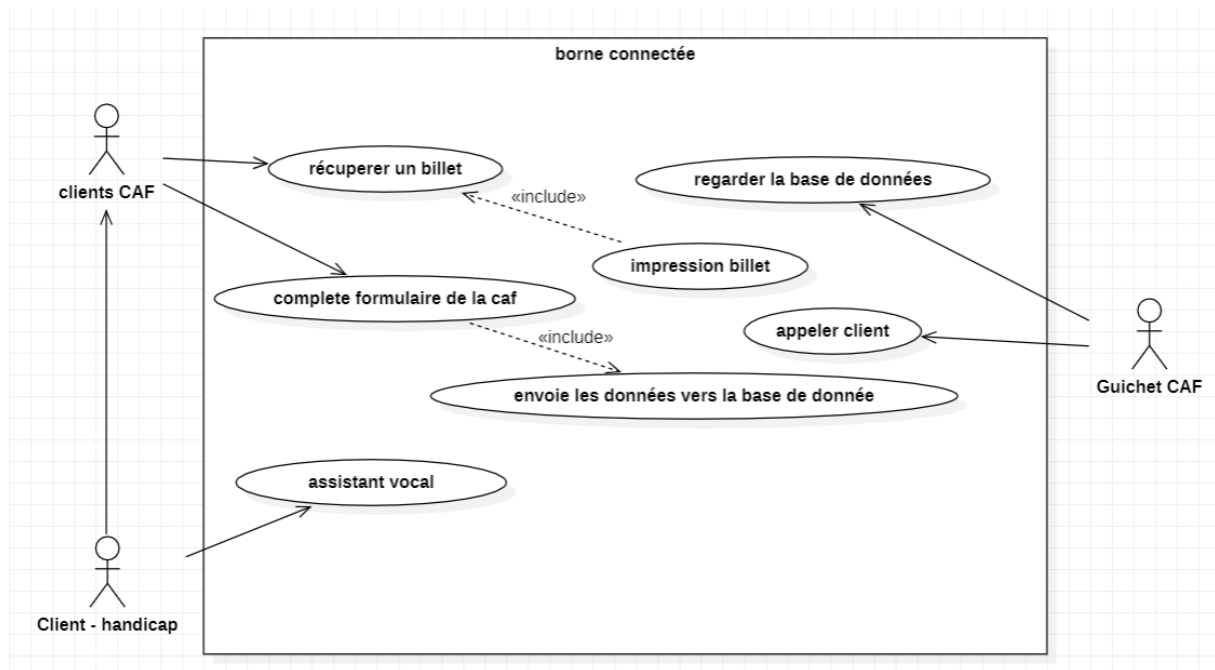
Pour finir un capteur de temps est défini et envoyé sur la base de données afin de réinitialiser le numéro de ticket chaque jour.

## e. Diagrammes de séquence.



Le diagramme de séquence représente graphiquement les interactions entre les différents acteurs et le système selon un ordre chronologique dans une formulation UML.

## f. Diagrammes de séquence.



Le diagramme de cas d'utilisation explique les différentes possibilités de droit par les acteurs agissant sur la borne d'un point de vue UML.

## VII. Besoin matérielle

Afin de choisir le matériel dont nous avons besoin, nous avons préféré de choisir plusieurs composant afin de les comparer entre eux et choisir un bon rapport qualité prix.

## a. Choix de la Raspberry

Nom des Raspberry	Modèle de CPU	Bluetooth	Mémoire (SDRAM)	Prix
Raspberry Pi 0w	1GHz ARM1176JZ	4.1 BLE	512Mo (intégré avec GPU)	10€
Raspberry Pi	700MHz ARM1176JZF-S core (ARM-11)	NON	256Mo (intégré avec GPU)	35€
Raspberry Pi 2	900MHz quadricœur ARM Cortex A7	NON	1Go	35€
Raspberry Pi 3	1.4GHzquadricœur ARM Cortex-A53	4.2 BLE	512Go	35€
Raspberry Pi 4	1.8GHzquadricœur	5.0 BLE	1Go/2Go/4Go/8Go	35€/45€/55€/75€

Nous avons finalement opté pour la Raspberry PI 4 avec 4Go de mémoire qui reste la plus complète nous laissant une marge d'erreur au cas où l'on rencontre un problème. Même si son prix est le plus élevé, il reste cependant abordable pour la conception de notre projet.

## b. Choix de l'imprimante

Imprimantes thermiques	Type de connecteurs	Bluetooth / Wifi	Vitesse d'impressions	Compatibilité	Batterie	Taille	Prix
MUNBYN WIFI Imprimante de reçus thermique 80 mm	USB, Ethernet	OUI	300 mm/s	Windows - MAC	Oui - Rechargeable	14.4 x 14.2 x 19 cm	180.99€
Imprimante thermique direct ZD421	USB	OUI	152 mm/s	PC - MAC	Oui Rechargeable	15 x 17,8 x 22,8 cm	491€
80Mn imprimante Thermique de reçu portable	USB	NON	50-80 mm/s	Android – IOS – PC - MAC	Oui – pile	11 x 5,6 x 9,8 cm 4,3 x 2,2 x 3,9 pouces	94.30€

Nous avons finalement choisi l'imprimante MUNBYN, qui malgré une vitesse faible par rapport aux autres, celle-ci avait de très bonne caractéristique pour intégrer notre projet.

L'imprimante thermique ZD421 et portable avaient également de bonne caractéristique mais le prix de la seconde était trop impactant par rapport à l'usage que nous voulions faire de plus pour faire des tests sur un mac (ordinateur utiliser par plusieurs de mes collègue), seul la MUNBYN pouvait ci prêter.

### c. Choix de l'écran

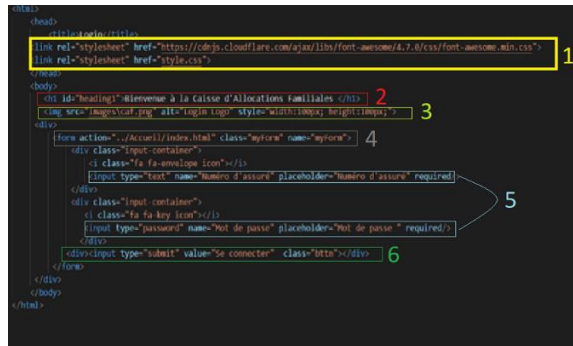
Ecrans Tactiles	Type de connecteur	Taille d'écran	Prix
Ecran tactile raspberry PI monitor	USB HDMI	10.1 pouces, LCD	176.99€
WIMAXIT monitor portable	USB HDMI	14 pouces, LED	240€
Ecran tactile raspberry PI magebok	USB HDMI	8 pouces, LCD	108.49€

Le choix de l'écran à été celui de 10.1 pouces, soit l'écran Raspberry PI monitor, qui permet une connexion en Raspberry avec un prix abordable par rapport au WIMAXIT et garde également une taille raisonnable pour être accessible au personne malvoyant.

## VIII. Interface Web



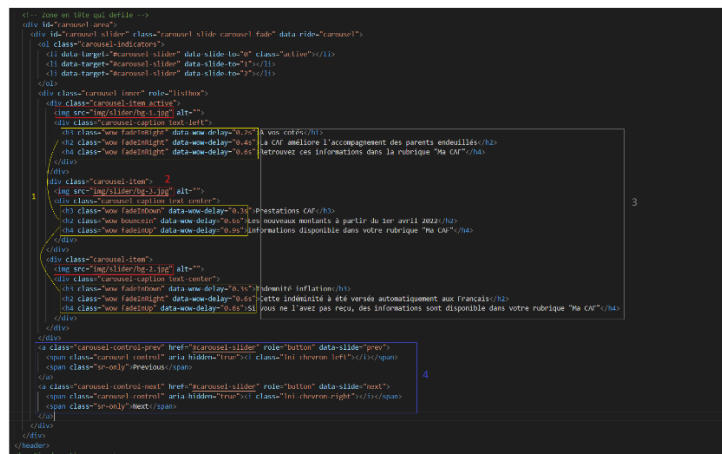
## a. Partie authentication



Bienvenue à la Caisse d'Allocations Familiales

Nous pouvons voir que le code HTML permet de mettre évidence la page de la caisse d'Allocations Familiales, avec la mise en place de l'ID et du mot de passe tous deux mis en pratique par le PHP de la page. Les 7 étapes expliquent comment la page mets en place le CSS ; puis le titre « Bienvenue à la Caisse d'Allocations Familiales » ; puis l'insertion du logo ; ensuite on peut retrouver l'ID et le mots de passe puis l'affichage du bouton « se connecter »

## b. Partie Page Principale



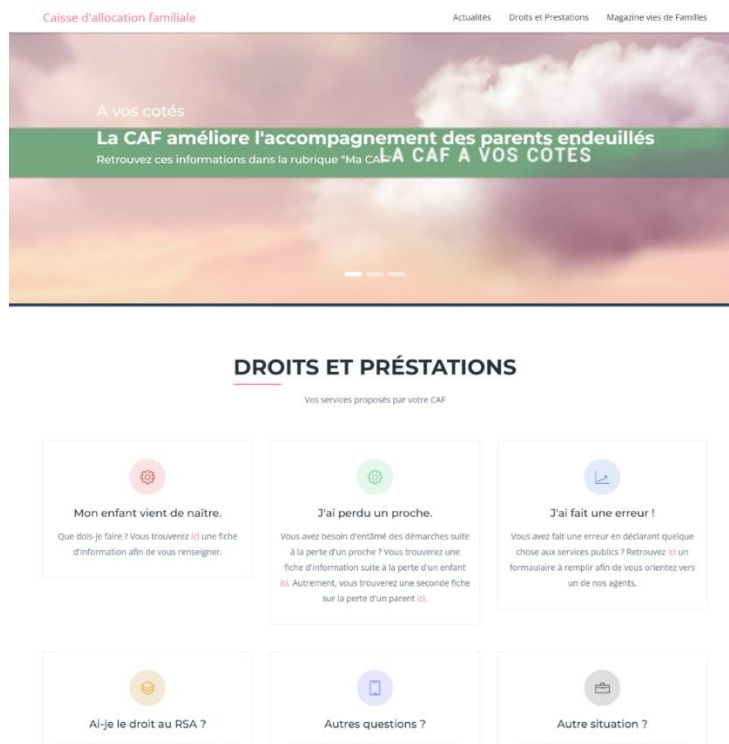


## Le Slider

Cette partie du code HTML permet de mettre en évidence 4 aspect du code

- 1-Créer un « slider » en bout de page.
- 2-Intégrer des images au « slider »
- 3-Intégrer un raccourcie sur le « slider » pour trouver les informations importantes
- 4-Ajouter un défilement manuel.

## Partie Accueil :



Voir sur le même code la partie accueil qui ajoute 6 fonctionnalités :

- Mon enfant vient de naître
- J'ai perdu un proche
- J'ai fait une erreur !
- Ai-je le droit au RSA ?
- Autres questions ?
- Autres situations ?

## IX. Réalisation de mes taches aux seins du projet

### a. Intégration de la base de données

#### i. Les différentes tables de la base sur MySQL

Afin de connecter le projet à la base de données, il a fallu créer différentes tables dans la base permettant de gérer l'accès à la CAF. C'est pourquoi nous avons créé différentes tables depuis MySQL.

# 1. La table User

Table Name:  Schema: **caf**

Charset/Collation:   Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Identifiant	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Mdp	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
compte	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:  Data Type:

Charset/Collation:   Default:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

*PK = Clef primaire ; NN = Non nul ; UQ = Index Unique ; B = Binaire ; UN = Donnée non signé ; AI = Auto incrémentation ; G = Colonne générale*

J'ai créé les différentes tables avec MySQL. Ici on a une représentation de la création de la table « User » qui comprend un identifiant et mots de passe. L'utilisateur a donc besoin d'un compte pour se connecter.

On définit un compte qui est l'élément principal de connexion, il est donc impossible que le compte soit nul.

On définit un mot de passe et un identifiant de valeur non nul et d'index Unique.

**Form Editor** | Navigate: 1 / 2 | Edit:

Identifiant:

Mdp:

Compte:

Dans la capture si dessus, voici un exemple d'enregistrement d'un compte. On defini un identifiant ; un mots de passe puis on defini une adresse mail de connection.

**Result Grid** | Filter Rows:  | Edit: | Export/Import: | Wrap Cell C

	Identifiant	Mdp	compte
	Baptiste	toto	bpoch...
	Théo	tata	tfranc...
	Maxime	tutu	mduha...
	Nilson	titi	nferge...
		NULL	NULL

**Result Grid**  
**Form Editor**

Une fois le compte enregistré, voici la base de données avec les différents comptes déjà enregistré. On peut également créer un compte depuis cette page en cliquant sur les « null ».

## 2. La Table Compte

comptes comptes user **comptes - Table**

Table Name: comptes Schema: calif

Charset/Collation: utf8mb4 utf8mb4\_09i Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
User	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
Password	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: User Data Type: VARCHAR(20)

Charset/Collation: Default Default

Comments:

Default: NULL

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☒ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output

Action Output

La table Compte prend en compte 3 éléments ; nous pouvons retrouver ID ; l’User ; et les mots de passe. Cette table ressemble à celle de l’utilisateur mais montre un point de vue interne à celui du compte.

De plus pour le compte, la clef primaire est attribuée à l’identifiant.

L’utilisateur lui est unique car il n’y a qu’un utilisateur par compte et inversement.

Pour ce qui est du mot de passe, lui ne peut pas être nul et reste unique afin de réduire le risque de fuite du compte.

La modification du compte s’effectue au niveau de la table « User » afin d’éviter de créer des doublons pour simplifier l’utilisation de cette borne.

### 3. La Table Guichet

comptes comptes user comptes - Table ticket - Table ticket ticket - Table x

Table Name: ticket Schema: caf

Charset/Collation: utf8mb4 utf8mb4\_0908 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
guichet	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
journée	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP
number	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Column Name: Data Type: Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Comments:

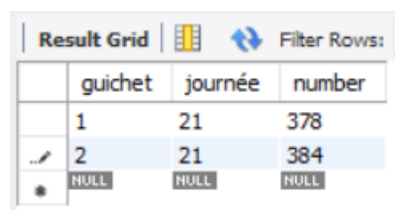
Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

La table ticket elle permet d'envoyé l'information à la borne ; un utilisateur imprime un ticket ce qui envoie une notification au guichet concerner par la demande au niveau de la borne.

La valeur journée permet réinitialiser le numéro de ticket afin d'éviter une incrémentation infiniment grande.

La valeur du ticket est donc non nulle, unique et possède une auto incrémentation.

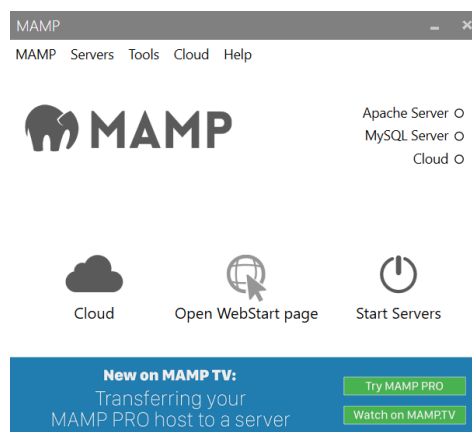


	guichet	journée	number
	1	21	378
	2	21	384
	NULL	NULL	NULL

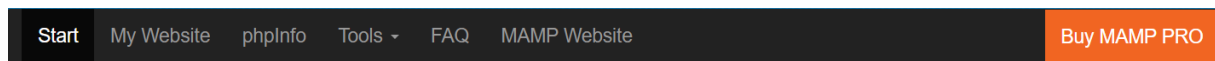
Ici nous avons différent guichet qui nous sont proposer tous deux lors de la 21nième journées du mois et les prochains numéros qui seront appelé par ces guichet sont le 378 pour le guichet 1 et 384 pour le guichet 2.

## ii. Intégration de la base de données sur le serveur MAMP & OVH.

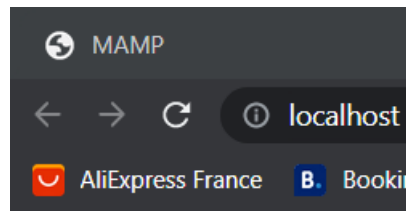
La base de données avait besoin de logiciel pour pouvoir diffuser le l'interface en local, c'est pourquoi nous avons utilisé MAMP.



Afin de s'y connecter il faut activer « MySQL serveur » et « Apache Server » puis faire « Open WebStart page » afin d'intégrer le programme.



Après avoir Intégré le projet dans la base il est possible d'ouvrir son site depuis MAMP sur l'encoche « My Website »



## Index of /

- [Projet.rar](#)
- [Projet/](#)

Nous pouvons Ducoup ouvrir le projet qui nous mènera vers la page d'authentification

## Bienvenue Ã la Caisse d'Allocations Familiales



ALLOCATIONS  
FAMILIALES

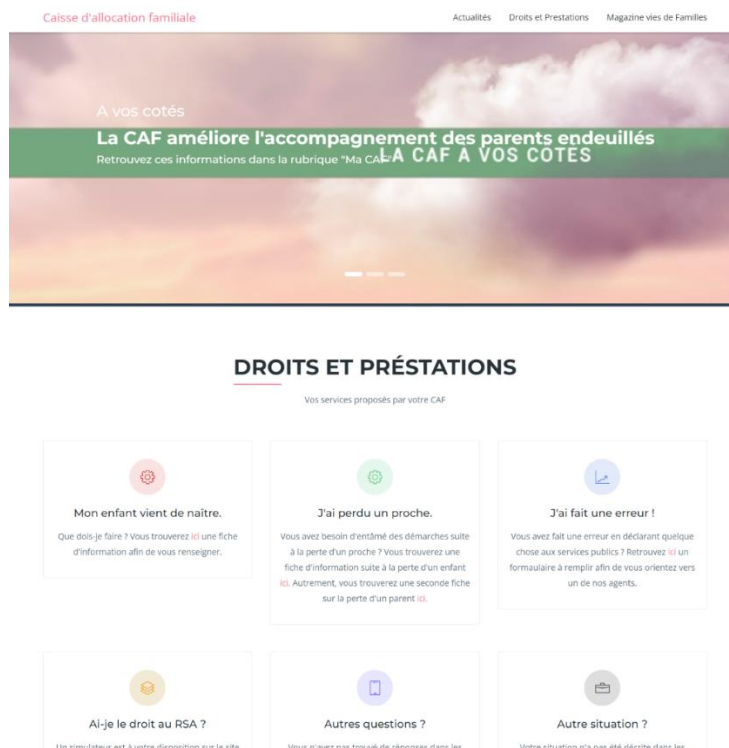
 054865128525

 .....

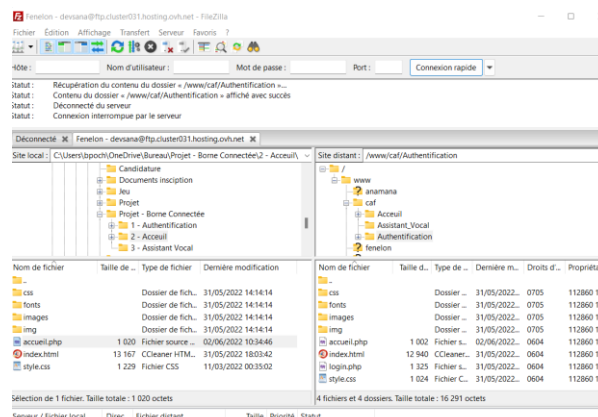
Se connecter



La page d'authentification demande un identifiant et un mots de passe déjà stocké sur la base de donnée afin d'accéder à la page d'accueil.



Par la suite nous avons déplacé notre projet sur un serveur OVH. Nous devons donc installer Filezilla, afin de transférer les dernières versions du programme afin d'évoluer la base de données.

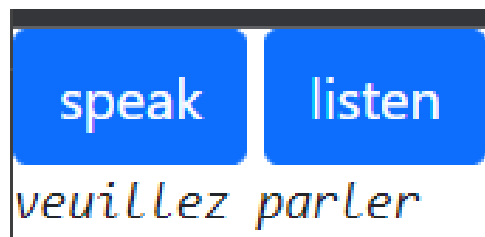


L'ensemble des différents projets est stocké sur Filezilla, il permet donc de stocker l'ensemble des projets et donner l'accès au différent membre du groupes les versions final, il suffit de taper l'adresse web suivant :

<https://discordapp.com/channels/968130440496042035/968956785501306890/981836084981936178>

### a. Intégration du texte to speak

Durant notre projet nous avons également dû développer un assistant vocal ou j'ai dû m'occuper de développer le code et l'améliorer afin de le rendre accessible pour la borne.



L'interface est divisée en 3 parties :

- Speak : L'assistant vocale se présente et présente les principales commandes de ses fonctions
- Listen : Ecoute la demande de l'utilisateur pour mieux orienter l'utilisateur dans les demandes qui lui sont proposer.
- La partie écrite dans ce cas : « veuillez parler » affiche le message de l'utilisateur après avoir cliqué sur listen pour éviter les confusions en cas d'erreurs.

On peut apercevoir 3 parties dans ce code :

```
let demande = ''
if (s.done.match(/bonjour|salut|salutations|bonsoir/gui)) demande = 'bonjour'
if (s.done.match(/étudiant|écolier/gui)) demande = 'etudiant'
if (s.done.match(/décéder|dêces|mort|trépasser/gui)) demande = 'mort'
if (s.done.match(/naissance|né|naitre|bébé|(nouveau né)/gui)) demande = 'naissance'

if (s.done.match(/(pardon|(pas entendu)|(pas compris)|(que dites-vous)|répéter|répète)/gui))
  demande = 'repeter'
```

Des mots clef sont défini afin d'identifier les différentes tel que  
bonjour et tous ses synonymes ; étudiant et tous ses synonymes etc.

Cette partie du code définit des mots clef qui vont être perçus comme des demandes ; ces demandes permettent de définir des éléments de réponses visible sur l'images ci-dessous :

```
<a name="" id="speak" class="btn btn-primary" href="#" role="button">speak</a>
<a name="" id="listen" class="btn btn-primary" href="#" role="button">listen</a>
<!-- Bootstrap JavaScript Libraries -->
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LCil5VnnvnwIqNvBxtM6ww6jpZ0UsyTCV6WtJgRm=>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-CVKIPhGWiC2Al4u+LWgxfKFTLTu>
<script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-7UuqjwUkH3rMBv7WkuTnD+rPna86JrsDadKPZ.==>
<script src="speech.js"></script>
<script>
let reponses = {
    bonjour: "Bonjour comment allez-vous?",
    etudiant: "Si vous êtes étudiants allez sur guichet numéro 3. Sinon lever la main, et un agent vous y conduira",
    mort: "Toutes mes condoléances, suite à la perte d'un proche, rendez vous au guichet 2. Sinon lever la main, et un agent vous y conduira",
    naissance: "Votre enfant vient de naître ? Félicitations ! Rendez vous au guichet numéro 1. Sinon lever la main, et un agent vous y conduira"
}
```

Si l'élément de demande est étudiant, la borne va retourner l'élément de réponse suivant : « Si vous êtes étudiants, allez sur guichet n°3, sinon levez la main et un agent vous y conduira »

L'ensemble des éléments de réponses est défini ci-dessus pour chaque éléments de demandes.

```

window.lastReponse = '..'

$(document).ready(() => {

    $('#speech').html('<i>veuillez parler</i>')

    $('#speak').click(() => {
        console.log('click!')
        window.tts.say("Bonjour, je suis la borne connectée de votre CAF ! Si vous avez beso
    })

    $('#listen').click(() => {
        console.log('listening...')
        // ASR = Autoamtic Speech Recognition ( Reconnaissance Automatique de la Parole)
        // TTS = Text To Speech (synthèse vocale)
        window.asr.start()

        window.asr.on('speech', s => {

            $('#speech').html('<b>' + s.done + "</b>" + '<i>' + s.speaking + '</i><BR/>')
        })
    })
})

```

Ce script permet le développement de l'assistant vocale avec la partie listening ; speech et speak.

```

console.log('==>>', window.lastReponse)
if (demande == 'repeter')
    window.tts.say(window.lastReponse)
else if (reponses[demande]) {
    window.lastReponse = reponses[demande]
    window.tts.say(window.lastReponse)
    console.log('last reponse', window.lastReponse)
}

```

Cette partie du code permet d'intégrer une demande « répéter » qui permet de répéter la dernière demande.

*Je me suis également occupé des diagrammes*

## X. Difficultés rencontrées :

J'ai rencontré diverse difficulté aux seins de ce projet notamment pour relier la base de données au projet car c'était une première pour moi, j'ai fini par réussir avec l'aide de mon professeur monsieur Gabal, professeur de développement de logiciel aux sein de

l'établissement Fénelon Sup, j'ai donc appris de mes erreurs pour reproduire les autres bases.

J'ai également rencontré des difficultés dans le développement de l'assistant vocale, car je ne maîtrisais pas parfaitement le langage HTML ; JavaScript et PHP.

## XI. Conclusion :

Durant ce projet, nous avons dû effectuer un système de borne connectée.

Afin de réaliser au mieux ce projet, nous avons dû anticiper le matériel nécessaire, réaliser des schémas adaptés à notre système.

Par ailleurs, une étude sur la partie programmation a été effectuée de pouvoir réaliser les tests adéquats.

Ce projet nous a permis de réussir à approfondir nos connaissances et comprendre le fonctionnement de différents composants électroniques.

Grâce à la Raspberry nous avons pu approfondir et découvrir de nouvelles

connaissances en matière de programmation du système embarqué. Nous sommes parvenus à réaliser notre projet dans les temps estimés.

Pour finir, nous voulons remercier Mr. Gabal et Mr Lerebours pour leurs aides, leurs conseils et leurs présences tout au long de notre projet.