Projet M1

Année scolaire 2023/2024

Institut Supérieur de l'Électronique et du Numérique
Tél. : +33 (0)2.98.03.84.00
Fax : +33 (0)2.98.03.84.10
20, rue Cuirassé Bretagne
CS 42807 - 29228 BREST Cedex 2 - **FRANCE**

F-City

Proposé par : Franck LE GALL

Thématique : Système embarqué

Maxime ALBERT — Domaine professionnel : Génie Logiciel
Antoine PROVAIN — Domaine professionnel : Génie Logiciel

1. Résumé

L'ISEN s'est doté d'un véhicule électrique, le « F-City », qui est mis à disposition des personnels ISEN.

Les étudiants de M2 Mobilité Électrique ont un module d'électronique embarquée dans lequel ils travaillent sur l'extraction de données de mesures électriques pour fournir un outil d'analyse de la conduite.

Dans ce projet, il s'agira de compléter ce dispositif de mesures en y adjoignant :

1. D'autres types de mesures (accélération, éclairement…)

2. De quoi afficher les paramètres d'aide à la conduite sur une IHM dans le véhicule

3. De quoi stocker ces données pour une analyse ultérieure

4. Un site internet afin de réserver le véhicule, de visualiser les données recueillies lors de différents trajets et d'avoir un retour d'expérience via un classement des utilisateurs les plus économes.

Ce projet a pour but d'aider le conducteur de la F-City lors de l'utilisation du véhicule, en y affichant toutes les données nécessaires à la conduite.

2. Sommaire

Projet M1 1

1. Résumé 2

2. Sommaire 3

3. Glossaire 5

4. Table des figures 7

5. Introduction 8

6. Cahier des charges 9

6.1 QQOQCCP 9

6.2 Taxonomie des fonctions 9

6.3 Diagramme de Pieuvre 9

6.4 Bête à corne 10

7. Gestion de projet 11

7.1 Dates clés 11

7.2 Contexte, définition et objectif du projet 11

7.3 Périmètre du projet 11

7.4 Descriptions fonctionnelles des besoins 11

7.5 Enveloppe budgétaire 12

7.6 Risques 12

7.7 Inventaire 12

7.8 Diagramme de Gantt 13

7.8.1 Diagramme de Gantt en début de projet 13

7.8.2 Diagramme de Gantt à postériori 13

7.9 Objectif du projet 13

7.9.1 Réalisation de l'IHM 14

7.9.2 Réalisation du site internet pour la réservation du F-City 14

7.9.3 Projet « fini » et utilisable en condition réelle 15

8. Le développement technique 16

8.1 Etude de l'existant 16

8.1.1 Etude des capteurs 16

8.1.2 Etude des programmes 16

8.1.3 Etude de la Raspberry Pi 16

8.2 Spécifications techniques 17

8.2.1 IHM 17

8.2.1.1 Microcontrôleur 17

8.2.1.2 Connexion entre la Raspberry et la carte FPGA 18

8.2.1.3 Interface graphique 18

8.2.1.4 Accès WIFI 19

8.2.1.5 UART et I2C 19

8.2.1.6 Stockage carte micro SD 19

8.2.1.7 Alimentation de 5V 20

8.2.2 Site internet 20

8.2.2.1 Fonctionnalités 20

8.2.2.2 Hébergement 20
8.3 Etude énergétique 20
8.4 Etude des communications 21
8.5 Choix des langages 22
8.5.1 IHM 22
8.5.2 Site internet 22
8.6 Base de données 23
8.6.1 Base de données externe 23
8.6.2 Base de données interne à la Raspberry 24
8.7 Conception de l'IHM 24
8.7.1 Maquette de l'IHM 24
8.7.2 Librairies installées 25
8.7.3 Gestion des capteurs 26
8.7.3.1 GPS 26
8.7.3.2 Capteurs de batterie et moteur 27
8.7.3.3 Autres capteurs 27
8.7.4 Affichage 27
8.7.5 Envoie des données 29
8.7.6 Procédure d'utilisation de l'IHM 29
8.7.7 Lancement automatique 30
8.8 Site internet 30
8.8.1 Connexion de l'utilisateur 30
8.8.2 Création d'un compte utilisateur 31
8.8.3 Réservation d'un trajet 32
8.8.4 Protocole d'utilisation du véhicule 33
8.8.5 Visualisation d'un trajet 34
8.8.6 Modification/ Suppression d'un trajet 36
8.8.7 Visualisation des graphiques 37
8.8.8 Visualisation du classement des utilisateurs les plus économes 39
8.8.9 Déconnexion 40
9. Conclusion 41
10. Remerciement 42
11. Bibliographie 43
12. Annexe 44
12.1 Taxonomie des fonctions 44
12.2 Diagramme de Gantt en début de projet 45
12.3 Diagramme de Gantt à postériori 46
12.4 Diagramme de Pert 47
12.5 Diagramme UML 48

3. Glossaire

IHM : Interface Homme Machine.

BDD : Base De Données

Twig : Moteur de template pour le langage PHP

Template : C'est un modèle d'élément facilement modifiable contenant des images, du texte, couleurs de fond…

Apache : Logiciel de serveur web qui écoute les requêtes émises par les navigateurs

MySQL : Système de gestion de base de données

API : Application Programming Interface. Permet à deux applications de discuter entre elles

GPS : Global Positioning System. Système de positionnement par satellites.

GPRMC : trame simplifiée pour fournir une localisation facilement décodable

FPGA : Circuit intégré de génération de composants logiques programmables

URL : Adresse d'un site ou d'une page internet

I2C : Inter-Integrated Circuit, est un bus informatique reliant un microprocesseur à différents circuits

SCL : Ligne d'horloge série

SDA : Ligne de données série

RxD : Récepteur de signal

UART : Récepteur/ émetteur asynchrone qui définit un protocole, ou un ensemble de règles, dédié à l'échange de données séries entre deux appareils

USB : Universal Serial Bus. Interface entre différents matériels informatiques

SSH : Secure Socket Shell, est un protocole réseau qui permet à un utilisateur d'accéder à une machine distante de manière sécurisée

Virtual Host : Méthode pour accueillir plus d'un nom de domaine sur une même machine

Python : Langage de programmation pour les applications web, le développement logiciel, la science des données et le machine learning

C : Langage de programmation qui s'écrit dans un fichier source

PHP : Hypertext Preprocessor, langage de programmation pour produire des pages web dynamiques via un serveur web

DNS : Domain Name System, permet d'associer un nom de domaine internet avec une adresse IP

Git : Gestion de version centralisée

Controller : S'occupe de recevoir les données entrées par l'utilisateur et de communiquer les changements au modèle, afin de modifier la vue

Service : Système standard qui permet via des scripts d'exécuter une application en mode service

Port série : C'est un port qui permet de recevoir et de transmettre des données bit par bit

HDMI : High Definition Multimedia Interface, permet la retransmission de flux chiffré

GPIO : General Purpose Input/Output, sont des ports d'entrée et de sortie des microcontrôleurs

SD : Sécure Digital, est une carte mémoire amovible de stockage de données

RAM : Random Access Memory, représente la mémoire vive de l'appareil

Pin : Pièce métallique solidaire d'un boitier ou circuit intégré, assurant une connexion électrique

CPU : Central Processing Unit, est un microprocesseur installé sur la carte mère

SQL : Structure Query Language, est un langage de programmation permettant de manipuler des données dans une base de données

Kivy : c'est une bibliothèque pour créer des applications tactiles, avec une interface utilisateur.

4. Table des figures

Figure 1 - Diagramme de Pieuvre 10
Figure 2 - Bête à corne 10
Figure 3 - PIN Raspberry Pi 4 17
Figure 4 - Raspberry Pi 4 18
Figure 5 - Micro SD de 32 Go 19
Figure 6 - Base de données 23
Figure 7 - Maquette de l'IHM 25
Figure 8 - Description GPRMC 26
Figure 9 - IHM finale 28
Figure 10 - Connexion à l'IHM 28
Figure 11 - Page de validation du transfert des données 29
Figure 12 - Page de connexion 31
Figure 13 - Page d'erreur de connexion 31
Figure 14 - Page d'inscription 32
Figure 15 - Numéro de badge 32
Figure 16 - Insertion dans la table user 32
Figure 17 - Page de réservation 33
Figure 18 - Insertion dans la table trajet 33
Figure 19 - Page de protocole d'utilisation 34
Figure 20 - Page de profil 34
Figure 21 - Page d'affichage des trajets 35
Figure 22 - Page sans réservation 35
Figure 23 – Affichage d'un trajet 36
Figure 24 - Page de suppression/ modification 36
Figure 25 - Modification du trajet 36
Figure 26 - Suppression d'un trajet 37
Figure 27 - Suppression d'un trajet en base de données 37
Figure 28 - Bouton de visualisation des graphes 37
Figure 29 - Table des capteurs 38
Figure 30 - Visualisation des graphiques 38
Figure 31 - Axe des abscisses des graphes 39
Figure 32 - Page du classement 39

5. Introduction

Avec le F-City, FAM Automobiles s'inscrit dans une démarche originale et unique. Cette solution a été apportée pour développer une autre manière de se déplacer en ville. Ce véhicule électrique est utilisable par 2 personnes adultes et mesure 2.5 mètres de long. Sur le même principe que le vélib', le F-City souhaite prendre une place dans le marché des nouveaux transports électriques.

Ce véhicule est déjà équipé d'un tableau de bord, permettant d'y afficher la vitesse, la batterie… Cependant, nous voulons plus. L'atout majeur de celui-ci est qu'il est adaptable aux besoins du client, tout en restant performant et robuste.

L'objectif est d'analyser les informations reçues par les différents capteurs (capteurs de batterie, capteurs moteurs, accéléromètre…) afin de les traiter et les afficher sur l'IHM.

Les tâches ont été réparties en deux grosses parties :

- Site internet et serveur associé, connecté avec une BDD (1)
- Réalisation de l'IHM et récupération/ traitement des données des capteurs (2)

Partie 1 :

Création d'un site internet permettant à chaque utilisateur de réserver le F-City, en fonction des contraintes de réservation. Possibilité de se connecter, de se créer un compte, de voir ses réservations et d'y apporter des modifications ou de les supprimer. Il y a aussi les graphes représentant les données reçues lors d'une réservation (graphe de puissance, batterie, vitesse et d'accélération). Grâce à la BDD, possibilité de faire un classement en fonction de l'utilisateur le plus économe en énergie.

Prise en main d'un serveur afin d'y héberger le site internet ainsi que la BDD.

Création de la BDD ainsi que sa connexion au serveur.

Partie 2 :

Réalisation de l'IHM de la voiture en temps réel afin d'afficher les données reçues par les différents capteurs. La partie GPS permet à l'utilisateur de se repérer lors de ses trajets.

Une fois les données reçues et traitées, elles sont envoyées dans la BDD du serveur afin de fournir les informations nécessaires à l'affichage des graphes et des classements sur le site internet.

Envoi des données de la BDD interne vers la BDD du serveur de l'ISEN.

Ces deux parties ont été réalisées séparément, mais rassemblées au moment de la mise en commun des codes. Pensées pour être simples et faciles d'utilisation, plusieurs documents d'explications sont laissés à l'utilisateur afin qu'il puisse avoir une meilleure connaissance de la procédure d'utilisation du véhicule et du site interne.

6. Cahier des charges

6.1 QQOQCCP

Quoi : réalisation d'un système embarqué 'fini' permettant l'affichage et le stockage des données concernant la F-City.

Qui : Le personnel de l'ISEN.

Où : à l'ISEN et aux alentours de Brest.

Quand : mise en place à partir de fin avril 2024.

Comment : création d'un serveur permettant la sauvegarde des données et le traitement afin de permettre l'affichage sur le tableau de bord.

Combien : budget de 500 euros.

Pourquoi : faciliter la prise d'informations électriques de l'utilisateur lors de l'utilisation du véhicule et analyser les données de conduite.

6.2 Taxonomie des fonctions

Fonctions principales FP, fonctions de contraintes FC, fonctions secondaires FS et fonctions d'estimes FE :

FP : Récupération, affichage et stockage de l'information électrique.

FC : Alimentation puis interprétation et traitement des données.

FS : Adaptation et transmissions des données vers le serveur.

FE : Boitier, esthétique, écran tactile et cryptage des données.

Voir ANNEXE

6.3 Diagramme de Pieuvre

Fonctions principales FP et fonctions de contraintes FC :

FC1 : Permettre d'interpréter les valeurs des capteurs
FC2 : Permettre de stocker les valeurs calculées des capteurs
FC3 : Permettre la transmission des données à un serveur
FC4 : Permettre le calcul de la vitesse

FP1 : Récupérer les informations sur puissance de la batterie
FP2 : Récupérer les informations sur puissance du moteur
FP3 : Permettre l'affichage des données

Après avoir analysé les différentes fonctions que requièrent ce projet, il a été choisi de mettre la carte Raspberry Pi 4 au cœur de notre Diagramme de Pieuvre car c'est elle le centre de ce projet, aussi bien par la récupération que l'affichage des données.

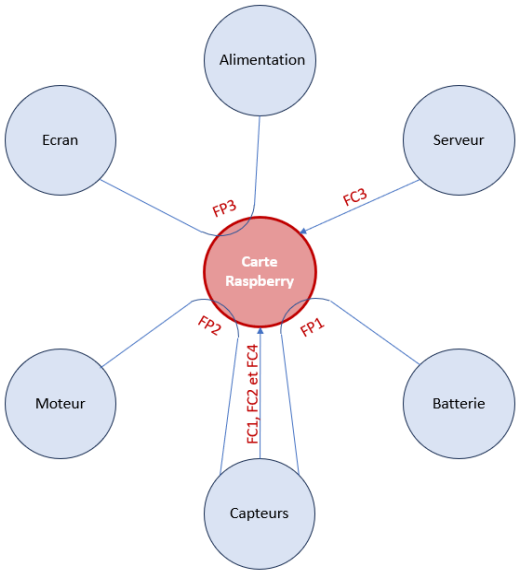


Figure 1 - Diagramme de Pieuvre

6.4 Bête à corne

Ce projet met au centre la carte Raspberry car elle offre un service à l'utilisateur tout en agissant sur la voiture. Ce service apporte une aide à la conduite à l'utilisateur.

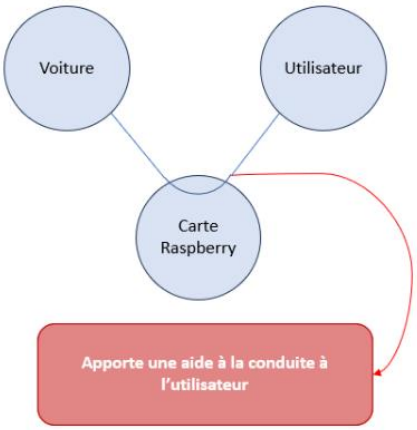


Figure 2 - Bête à corne

7. Gestion de projet

7.1 Dates clés

Début de projet : 8 janvier 2024
Fin du projet : 19 avril 2024
Remise du rapport : 14 avril 2024
Démonstration : 18 avril 2024
Soutenance orale : 19 avril 2024

7.2 Contexte, définition et objectif du projet

L'utilisateur a besoin d'avoir les informations relatives à sa conduite lorsqu'il utilise le véhicule et ainsi stocker ces informations pour pouvoir les analyser ultérieurement.

Voici la liste ci-dessous des différents objectifs à atteindre :

- Analyser la carte informatique et les capteurs déjà existants
- Choisir les composants à utiliser pour parvenir à ce but
- Afficher les paramètres d'aide à la conduite sur une interface homme machine
- Stocker les données en interne et sur serveur
- Rendre le module fini opérationnel sur le véhicule

7.3 Périmètre du projet

Utilisation par le personnel de l'ISEN dans un périmètre proche de l'école déterminé par les contraintes de batterie du véhicule.

Le rendu du prototype final devra être fini pour la fin du mois d'avril 2024.

7.4 Descriptions fonctionnelles des besoins

Pour décrire au mieux les besoins fonctionnelles du véhicule, les fonctions seront divisées en deux parties : fonctions principales et fonctions de contrainte.

Fonctions principales :

- Permettre d'interpréter les valeurs des capteurs
- Permettre de stocker les valeurs calculées des capteurs
- Permettre la transmission des données à un serveur
- Permettre le calcul de la vitesse
- Permettre le calcul de puissance
- Récupérer les informations électriques sur le courant de la batterie
- Récupérer les informations électriques sur la tension de la batterie
- Récupérer les informations électriques sur le courant du moteur
- Récupérer les informations électriques sur la tension du moteur

Fonctions de contraintes :

- Permettre l'interprétation des valeurs des capteurs
- Permettre le stockage des valeurs calculées des capteurs
- Permettre la transmission des données au serveur
- Permettre le calcul de la vitesse
- Sécuriser la transmission des données

7.5 Enveloppe budgétaire

Le budget concernant le projet s'élève à 500 euros.

7.6 Risques

- Manque d'espace de stockage sur la carte SD
- Soucis d'alimentation
- Problème de transmission avec le serveur
- Risques matériels (surchauffe, humidité, soudure…)
- Problème d'affichage (problème de puissance graphique de la carte)
- Problème de rafraîchissement de la carte Raspberry Pi en temps réel
- Echange de données entre les BDD interne et externe
- Perte de données lors d'une pause pendant le trajet
- Confidentialité des données
- Authentification de l'utilisateur avant l'utilisation du véhicule

7.7 Inventaire

Au commencement du projet, il a été mis à disposition du matériel afin de parvenir à la réalisation finie de celui-ci, que voici :

- Voiture F-City équipée d'une carte FPGA
- Un câble HDMI vers un port universel
- Bloc d'alimentation USB-C
- Kit de dissipation thermique pour Raspberry pi 4B couleur argent aluminium
- Carte SD de 32Go et adaptateur micro-SD
- Raspberry Pi 4 modèle B 8 Gbits de RAM
- Boitier pour Raspberry Pi 4
- Guide de démarrage rapide Pi 4B
- Ancien projet :
    - Écran pour Raspberry Pi
    - Système embarqué connecté à un Raspberry Pi 3

Pour permettre le déploiement du site internet et le bon fonctionnement de l'IHM, voici le matériel qui a été nécessaire pour la réalisation du produit pendant son développement :

- Serveur ISEN
- Câble USB-C vers USB
- Lecteur de carte SD (1)
- Batterie 12V pour la voiture (3)
- Boitier Raspberry Pi touch compatible avec les Raspberry Pi 4 (2)

Commande	Prix
1 + 2	23.24 €
3	107.94 €
Total	135.83 €

7.8 Diagramme de Gantt

7.8.1 Diagramme de Gantt en début de projet

L'approche du diagramme de Gantt en début de projet a donné une première approche des tâches à effectuer et de la répartition de celles-ci au sein du groupe. Cependant, certaines missions pouvaient être superposées, et/ ou mieux organisées et de manière plus séparées.

Voir ANNEXE

7.8.2 Diagramme de Gantt à postériori

Ce nouveau diagramme a pris forme au fur et à mesure que les tâches s'effectuaient durant la partie de développement de l'IHM. Ces durées sont donc beaucoup plus représentatives sachant que ce sont des valeurs relatives au temps de développement.

Voir ANNEXE

7.9 Objectif du projet

Ce projet peut être séparé en 3 parties distinctes :

- Réalisation de l'IHM
- Réalisation du site internet pour la réservation du F-City
- Créer un projet « fini » et utilisable en conditions réelles

Ces parties seront développées séparément, mais se réuniront pour rendre l'IHM utilisable par n'importe quel utilisateur.

7.9.1 Réalisation de l'IHM

Afin d'obtenir un affichage compréhensible et simple d'utilisation pour l'utilisateur, l'IHM comporte certains points que voici :

- Connexion au réseau WIFI de l'ISEN grâce à une borne WIFI installée dans le garage de l'école
- Lancement de l'IHM au démarrage de la voiture
- Demande du numéro de trajet associé à l'utilisateur pour que l'utilisateur démarre l'IHM
- Récupération de ce numéro de trajet afin de regarder s'il est bien enregistré dans la base de données
- Affichage en temps réel des données reçues des différents capteurs (moteur, batterie, accéléromètre…)
- Insertion des données concernant le trajet dans la base de données interne à la Raspberry Pi
- A l'arrivée du véhicule, les données de la base de données interne sont envoyées vers la base de données du serveur ISEN
- La consommation énergétique du tableau de bord doit être faible afin d'assurer une autonomie plus souple pour le véhicule

7.9.2 Réalisation du site internet pour la réservation du F-City

Comme pour toute location de véhicule, il est primordial de faire une réservation. Sans cela, chaque utilisateur serait amené à vouloir utiliser le véhicule, sans même savoir s'il est disponible. Pour cela, « fcity.isen » est là.

Afin de répondre aux besoins, ce site doit comprendre les fonctionnalités essentielles suivantes :

- Authentification/ Création d'un utilisateur
- Visualisation des trajets passés et futurs
- Réservation sur un créneau horaire du véhicule
- Visualisation graphique des données reçues lors d'une réservation
- Classement des utilisateurs en fonction de leur consommation
- Modification d'une réservation
- Suppression d'une réservation

Pour répondre à cette demande, la création de ce site internet s'est fait à l'aide du Framework PHP Symfony, qui est le plus a adapté en termes de robustesse, stabilité et modularité (important dans ce type de projet en constante évolution).

En ce qui concerne l'hébergement de ce site internet, il a été décidé de l'héberger sur le serveur interne de l'ISEN. L'installation d'Apache et sa configuration ont été une nécessité afin que ce site soit disponible sur les différents moteurs de recherche.

C’est aussi sur ce serveur qu’est abrité la base de données, contenant les données des utilisateurs, des trajets et des capteurs. Le choix s’est porté sur l’utilisation de MySQL car open-source, simple d’utilisation et permettant de grande performance.

7.9.3 Projet « fini » et utilisable en conditions réelles

Pour que le projet soit « fini » et utilisable en conditions réelles, il faut un boitier contenant tous les câbles électriques ainsi que les différents capteurs utilisés. De plus, il faut pouvoir exécuter le code au démarrage de la voiture.

8. Le développement technique

8.1 Etude de l'existant

8.1.1 Etude des capteurs

Au démarrage du projet, la partie d'analyse du véhicule et des éléments fournis était essentielle pour commencer correctement le projet et éviter des tâches inutiles.

En ce qui concerne les capteurs associés au véhicule F-City, il a été fourni un certain nombre de capteurs répertoriés ci-dessous :

- 1 GY-NEO6MV2 (GPS)
- 1 MPU6050 (accéléromètre, gyroscope et température)
- 1 capteur d'humidité
- 2 capteurs de tension moteur (en phase 1 et 2)
- 2 capteurs d'intensité de courant du moteur (en phase 1 et 2)
- 1 capteur de tension batterie
- 1 capteur d'intensité de courant de batterie

Ce sont les valeurs de ces capteurs qu'affichera l'IHM durant le trajet des utilisateurs du F-City.

8.1.2 Etude des programmes

Après une étude approfondie des programmes transmit par M**. LE GALL**, il a été décidé de refaire la totalité des fonctionnalités du projet car elles n'étaient pas optimales et prenaient beaucoup trop de ressources.

L'exception se fait pour le fichier de récupération des données de la carte FPGA nommé :

- ***read-vehicle-parameters-usb-serial.c***

L'explication de ces fonctionnalités se fera ultérieurement.

8.1.3 Etude de la Raspberry Pi

Dans un premier temps, pour faire fonctionner la Raspberry Pi et toutes ses applications, il a été nécessaire d'installer et de configurer le système d'exploitation grâce à la carte micro SD fournie. Pour cela, l'étude des PIN de la Raspberry Pi a été primordiale.

Vous pouvez voir ci-dessous les fonctions des différentes PIN de la Raspberry Pi 4. C'est grâce à ce schéma que la configuration des PIN a été possible.

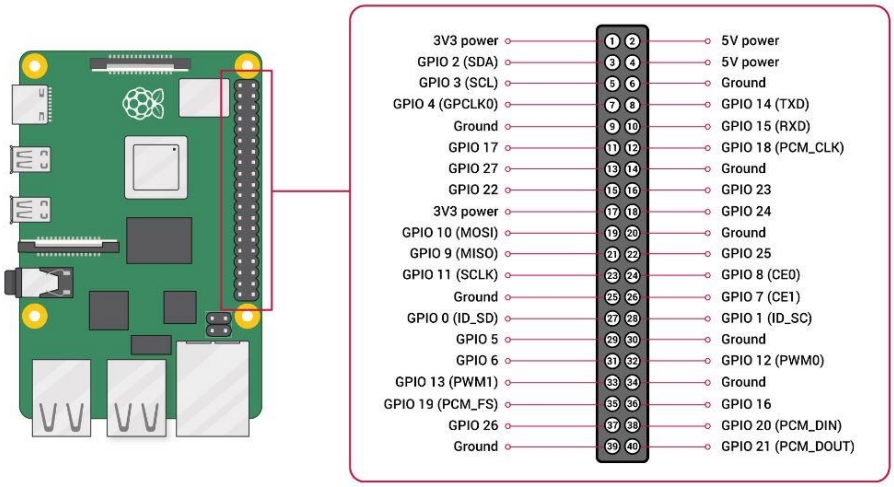


Figure 3 - PIN Raspberry Pi 4

8.2 Spécifications techniques

8.2.1 IHM

Dans le cadre de ce projet, l'IHM doit avoir un certain nombre de caractéristiques suivantes :

- Microcontrôleur à faible consommation
- 1 port USB minimum
- Interface graphique tactile
- Accès WIFI
- UART et I2C
- Stockage carte micro SD
- Alimentation de 5V

8.2.1.1 Microcontrôleur

Les anciens projets qui utilisaient un Raspberry Pi 3b comme microcontrôleur étaient limitées par la puissance de la carte. Cette année, parmi le matériel fourni, il y avait un Raspberry Pi 4, permettant ainsi une plus grosse capacité de rafraichissement et de traitement de données. Cette carte possède les caractéristiques ci-dessous :

Microcontrôleur	Raspberry Pi 4
CPU	ARM Cortex-A72 64 bits quatre cœurs à **1,5 GHz**
Mémoire RAM	4 GB de mémoire RAM
Ports USB	2 ports USB 2.0 / 2 ports USB 3.0
Ethernet	1 port Ethernet Gigabit
WIFI	1 interface WIFI
Bluetooth	1 interface Bluetooth (5.0)
HDMI	2 ports micro HDMI (jusqu'à 60 FPS)

SD	1 port micro SD
GPIO	1 connecteur GPIO 40 broches E/S
UART	OUI
I2C	OUI
Tension	5V
Courant	2.5 A
Puissance	12.5 W
Alimentation	5 VCC/ 3 A
Poids	46 g

Figure 4 - Raspberry Pi 4

8.2.1.2 Connexion entre la Raspberry et la carte FPGA

Grâce à ces nombreux ports USB (4 ports USB), la connectivité entre la carte FPGA et la Raspberry est possible. C'est par ce port USB que seront transmises toutes les données concernant les informations sur le moteur et la batterie.

Pour que les informations soient envoyées par la carte FPGA et reçues par la Raspberry Pi 4, il faut que le moteur de la F-City soit allumé. De plus, la Raspberry doit être alimentée en tension, sinon aucune action ne sera faite lors de l'écoute des ports USB de celle-ci.

8.2.1.3 Interface graphique

L'IHM repose sur l'écran de notre Raspberry Pi 4, et qui se caractérise comme ceci :

Ecran utilisé	Ecran LCD officiel Raspberry Pi
Fréquence de rafraîchissement	60 Hz
Résolution	800 X 480
Tension	5V
Courant	500 mA
Puissance	2.5 W
Interface de connexion	DSI, GPIO

L'intérêt d'avoir un écran LCD officiel Raspberry Pi est qu'il permet une autonomie et une liberté dans l'affichage des informations, qu'il n'est pas possible d'avoir avec le tableau de bord de la F-City, car l'écran d'origine ne peut être modifié.

8.2.1.4 Accès WIFI

Parmi les caractéristiques du microcontrôleur, l'interface WIFI permet la connexion au réseau de l'ISEN et ainsi vérifier à travers une API si le numéro de trajet rentré par l'utilisateur est correct. Sans cet accès, l'IHM serait dans l'incapacité de détecter si le numéro de trajet rentré par l'utilisateur existe ou non.

De plus, lors de l'envoi des données entre le serveur interne à la Raspberry Pi et le serveur de l'ISEN, il faut être connecté au WIFI, sinon les données ne pourront pas être insérées dans la base de données du serveur ISEN.

8.2.1.5 UART et I2C

Ces deux protocoles permettent la communication entre la Raspberry Pi et les différents capteurs (accéléromètre, humidité, GPS…).

Capteurs	GY-NEO6MV2	MPU6050	AM2320
Liaisons	UART	I2C	I2C
PIN utilisée	GPIO15 (RxD)	SDA et SCL	SDA et SCL

8.2.1.6 Stockage carte micro SD

Grâce à son port micro SD, il est possible d'installer/ stocker le système d'exploitation de la Raspberry Pi.

Fourni au début de ce projet, la carte micro SD de 32 Go permet de pouvoir installer et stocker plus confortablement les programmes permettant à l'IHM de fonctionner. Le système d'exploitation, d'une taille d'environ 4 Go, prend une place considérable sur la carte SD. Le choix d'une carte de plus grande capacité, comme celle qui a été fournie, est la meilleure option.

Dans le cadre de ce projet, pour assurer le bon fonctionnement du système d'exploitation, il a été nécessaire de réinitialiser la carte micro SD, et par la suite d'y installer le nouveau système d'exploitation.

Figure 5 - Micro SD de 32 Go

8.2.1.7 Alimentation de 5V

Pour que l'IHM soit fonctionnelle, il faut que la carte Raspberry Pi soit alimentée avec une tension de 5V. Pour parvenir à cette problématique, il a été décidé de la brancher, grâce à un adaptateur USB de 5V dans la prise allume-cigare, un câble USB vers USB-C afin de pouvoir alimenter la carte Raspberry.

8.2.2 Site internet

La partie du site internet est primordiale, car sans elle, il serait impossible d'utiliser correctement la F-City. Effectivement, s'assurer de réserver le véhicule avant de l'utiliser est nécessaire.

Pour cela, **fcity.isen** a été conçu. Mais quelles sont ses fonctionnalités ?

8.2.2.1 Fonctionnalités

Voici une liste exhaustive des tâches réalisées par le site internet :

- **Création** d'un compte utilisateur
- **Connexion** au site avec ses identifiants
- **Visualisation** de ses trajets passés et futurs
- **Modification** et/ ou **suppression** de ses trajets
- **Visualisation** de graphiques concernant le trajet
- **Visualisation** du classement des utilisateurs les plus économes en énergie
- **Création** d'un nouveau trajet
- **Visualisation** du protocole d'utilisation du véhicule

8.2.2.2 Hébergement

Dans un premier temps, ce site internet a été programmé sur une machine locale et ensuite déployé sur une machine distante. Pour cela, il a été préférable d'utiliser un serveur ISEN afin de l'héberger.

8.3 Etude énergétique

Pour s'assurer de la bonne intégration de notre IHM dans le véhicule F-City, il a été obligatoire de s'orienter vers l'étude énergétique de l'influence de l'IHM sur la voiture. Pour cela, l'étude de la documentation du constructeur du véhicule a été nécessaire. Voici les deux données principales à relever de ces documents constructeurs :

- Batterie de servitude de 12V/ 40 Ah
- Pic de courant maximum à ne pas dépasser par la batterie de servitude : 10A

Ce sont les données qui permettront par la suite de vérifier si l'autonomie de la voiture est suffisante avec l'utilisation de l'IHM.

Il a été entreprit une analyse de la consommation électrique maximale de tous les éléments constitutifs de l'IHM. L'intensité électrique maximale requise par l'IHM est de 1,5A lorsqu'elle est alimentée sous 5V. Cependant, pour une alimentation de 12V provenant de la batterie de servitude, une adaptation est nécessaire pour évaluer l'intensité électrique nécessaire à l'IHM. Cette adaptation prend en compte le convertisseur 12V → 5V intégré à la prise allume-cigare, avec un rendement de 96%.

$$I_{12v} = I_{5v}\frac{5}{12\eta} = 0{,}651A$$

Afin de prévenir une décharge excessive de la batterie de servitude, il a été choisi d'utiliser la moitié de sa capacité nominale de 40 Ah pour les estimations d'autonomie énergétique. Cette approche est basée sur l'hypothèse que la batterie de servitude ne bénéficie pas d'une recharge via la batterie principale.

$$\Delta t = \frac{20}{0.651} = 30\ heures\ 43\ minutes$$

En étudiant ce résultat, il a été conclu que l'intégration de l'IHM avec le F-City permettrait une autonomie de 30 heures et 43 minutes.

Quant à l'autonomie du véhicule, elle est environ égale à 8h. Il est donc totalement possible d'utiliser l'IHM pendant un trajet avec le F-City.

8.4 Etude des communications

Pour décrire au mieux les communications au sein de ce projet, étudions d'abord quelles communications sont nécessaires pour son fonctionnement.

Le premier élément essentiel est la communication entre l'IHM est le réseau **WIFI** de l'ISEN. En effet, voici les fonctionnalités qui nécessitent une connexion WIFI :

- Récupération du numéro de trajet rentré par l'utilisateur et vérification de son existence dans la base de données externe, sur le serveur ISEN.
- Envoie des données du trajet dans la base de données du serveur ISEN.

Dans un second temps, il faut pouvoir recevoir les données des capteurs installés sur la batterie et le moteur. Les informations reçues des capteurs sont traitées à l'aide d'une carte FPGA qui renvoie ensuite ces données à l'aide d'un câble **USB**. Cette méthode permet :

- La récupération des données des capteurs concernant le moteur et la batterie
- Intégration en temps réel de ces valeurs

Pour finir, la récupération des données des capteurs d'accélération, d'humidité et GPS se fait grâce à la connexion par les PIN de la carte Raspberry en **UART** et **I2C**. Ce protocole de communication a permis :

- Récupération des données des capteurs ci-dessus en fonction de leur protocole
- Intégration en temps réel de ces valeurs

En conclusion, il y a 3 protocoles principaux qui gèrent la communication entre les capteurs, le WIFI et l'IHM.

8.5 Choix des langages

8.5.1 IHM

Python 3.11 :

- Récupération des données des capteurs d'accélération, d'humidité et de GPS
- Création de l'interface graphique visible par l'utilisateur
- Connexion à la base de données
- Envoie des données reçues des capteurs
- Interpréter la réponse du programme C de lecture de la carte FPGA

C :

- Récupération des valeurs de la FPGA par le port USB concernant les capteurs sur le moteur et la batterie

8.5.2 Site internet

HTML, **CSS** :

- Constitution des pages web (titres, boutons, formulaire, menu…)
- Mise en forme (couleur, alignement, dynamisme…)

JS :

- Gestion de l'affichage des graphes en fonction du bouton sélectionné par l'utilisateur (puissance, batterie, accélération, vitesse)

PHP :

- Récupération des données rentrées par l'utilisateur dans les formulaires
- Gestion des données avec la base de données
- Gestion des url et des routes du site internet
- Gestion de connexion et de création des comptes et des trajets

Framework Symfony :

- Gestion des liaisons entre les pages
- Gestion de l'affichage des données entre les pages
- Liaison à la base de données

8.6 Base de données

8.6.1 Base de données externe

Pour faire fonctionner ce projet correctement, il est important de bien configurer la base de données externe (sur le serveur ISEN), pour ces différentes raisons :

- Gestion des utilisateurs
- Gestion des trajets
- Gestion des données des capteurs
- Bonne insertion des données lors de la fin d'un trajet

Ces différentes fonctions sont alors réparties dans 3 tables différentes :

- User
- Trajet
- Capteur

La table user comporte toutes les données relatives aux utilisateurs (nom d'utilisateur, numéro de badge et mot de passe). La table trajet se repose sur l'identification précise de chaque trajet (un numéro de trajet, numéro de badge, date du trajet, heure de départ et de fin). Et pour la table capteur, c'est elle qui reçoit toutes les données des capteurs pendant un trajet. Voici en image leurs représentations.

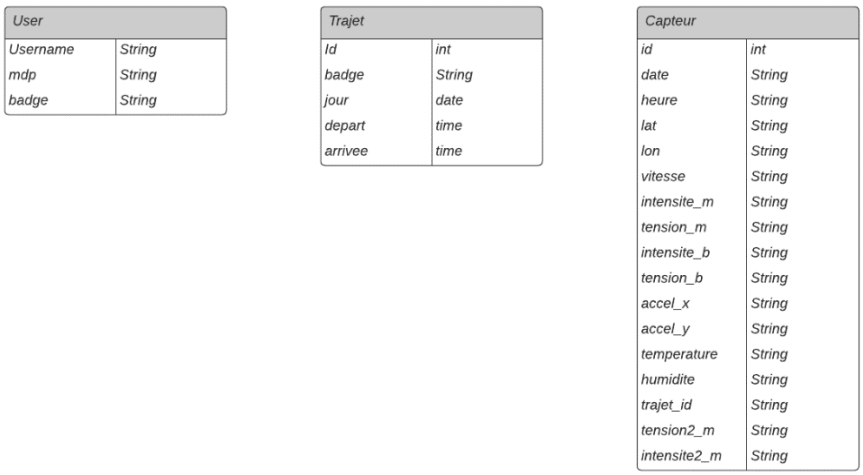


Figure 6 - Base de données

8.6.2 Base de données interne à la Raspberry

La Raspberry Pi a été configurée de manière à pouvoir initialiser une base de données. Mais pour quelles raisons ?

La première raison s'explique par la perte de données. En effet, il existe plusieurs raisons possibles qui amèneraient à la perte des données du trajet.

- La voiture s'éteint
- L'utilisateur fait une pause pendant le trajet

Ces différentes hypothèses mènent à transmettre les informations dans une base de données interne afin de perdre le moins de données possibles.

De plus, lors de la transmission des données du serveur interne à la Raspberry vers le serveur ISEN externe, il est plus facile de transmettre uniquement les valeurs de notre trajet directement d'une base de données à une autres base de données à l'aide de la commande SQL « Insert Into ». Cette commande permet d'insérer dans notre table « capteur » externe les données du trajet une à une.

Sur la Raspberry, il n'est pas nécessaire d'avoir les tables d'utilisateurs et de trajets, c'est pourquoi il n'y a que la table capteur qui y figure.

La décision d'une base de données plutôt qu'un fichier texte réside dans le fait que l'utilisation d'une base de données est plus sûre et offre de meilleures performances.

8.7 Conception de l'IHM

8.7.1 Maquette de l'IHM

Au début du projet, il a été demandé de réaliser la maquette de l'IHM afin de prévoir quelles informations affichées sur l'écran. Parmi les prérequis à avoir :

- Date et heure
- Vitesse
- Batterie
- Puissance

L'objectif étant d'aider l'utilisateur pendant sa conduite, il était donc préférable d'opter pour une IHM contenant le plus d'informations possible. Voici le rendu final de cette maquette.

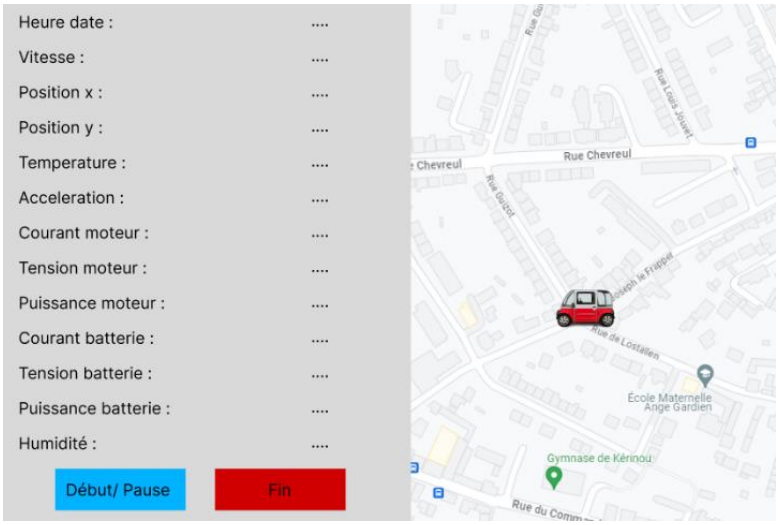


Figure 7 - Maquette de l'IHM

Comme vous pouvez le voir, cette nouvelle version comporte les attributs suivants :

- Date et heure
- Vitesse
- Position x/ Position y
- Température
- Accélération
- Courant/ Tension/ Puissance moteur
- Courant/ Tension/ Puissance batterie
- Humidité

8.7.2 Librairies installées

Pour que le programme compile correctement, il a fallu installer les librairies suivantes, qui ont toutes des tâches particulières. Pour une meilleure compréhension, elles seront organisées en fonction des tâches à effectuer.

- *Affichage des données*

Kivy : Framework pour l'affichage graphique sur python avec une multitude de librairies internes, permettant la création de rendu varié.
Time : gestion du temps
Board : gestion de la lecture du port I2C
Math : utilisé pour arrondir à l'inférieur grâce à la fonction « floor »

- *Capteur d'humidité et température*

Adafruit_am2320 : utilisé pour lire les données reçues par l'am2320

- *Capteur GPS*

Serial : utilisé pour lire les ports séries de la Raspberry Pi

- *Capteur accélération/ gyroscope/ température*

Mpu6050 : utilisé pour interpréter les données reçues par la mpu6050

- *Base de données*

Request : utilisé pour faire des requêtes API au serveur
Json : utilisé pour le format des données envoyées via l'API
Mysql.connector : utilisé pour remplir ou récupérer des informations dans la base de données interne de la Raspberry Pi

- *FPGA*

Subprocess : utilisé pour exécuter du code C depuis le code Python
Statistics : utilisé pour la fonction « median » permettant de calculer la médiane d'un tableau de valeurs
Collections : utilisé pour stocker des données, facilement modifiables

8.7.3 Gestion des capteurs

8.7.3.1 GPS

Le capteur GPS récupère différents types de valeurs :

- GPGSV, GPRMC, GPGSA, GPGGA, GPGLL, et GPVTG

Après quelques recherches, il a été convenu que le type de valeurs correspondant le plus aux besoins était le GPRMC car il contient longitude, latitude, date, heure, vitesse et la validité de la trame.

Field No.	Name	Unit	Format	Example	Description
0	xxRMC	-	string	$GPRMC	RMC Message ID (xx = current Talker ID, see NMEA Talker IDs table)
1	time	-	hhmmss.ss	083559.00	UTC time. See the section UTC representation in the Integration manual for details.
2	status	-	character	A	Data validity status, see position fix flags description
3	lat	-	ddmm.mmmmm	4717.11437	Latitude (degrees and minutes), see format description
4	NS	-	character	N	North/South indicator
5	lon	-	dddmm.mmmmm	00833.91522	Longitude (degrees and minutes), see format description
6	EW	-	character	E	East/West indicator
7	spd	knots	numeric	0.004	Speed over ground
8	cog	deg	numeric	77.52	Course over ground

Figure 8 - Description GPRMC

Pour pouvoir exploiter correctement les valeurs, certaines ont dû être modifiées, comme le format de la date, pour le passer au format aaaa-mm-dd, passer les heures en UTC+01 :00 et passer la vitesse en km/h.

8.7.3.2 Capteurs de batterie et moteur

Pendant une période de 0.5 secondes, les valeurs récupérées grâce au programme « ***read-vehicle-parameters-usb-serial.c »*** sont chacunes triées pour ne garder que la médiane de ces valeurs, et ainsi limiter les valeurs incohérentes.

Les données récupérées sont :

- Intensité de la batterie (**Ibat**)
- Intensité du moteur en phase 2 (**I2**)
- Intensité du moteur en phase 1 (**I1**)
- Tension du moteur en phase 2 (**U23**)
- Tension du moteur en phase 1 (**U12**)
- Tension de la batterie (**Ubat**)

8.7.3.3 Autres capteurs

La récupération des valeurs des autres capteurs est simplifiée via des librairies déjà existantes, comme l'adafruit_am2320, pour l'am2320 et la librairie mpu6050 pour le capteur qui porte le même nom.

8.7.4 Affichage

Cet affichage, séparé en 2 parties, permet à l'utilisateur de pouvoir s'orienter grâce à une carte GPS sur sa droite et connaître les principales informations concernant les capteurs sur sa gauche.

Chacune de ces données, comprenant les données des capteurs et celles de la carte GPS, sont actualisées en temps réel pendant le trajet.

Sur la partie de gauche, vous pouvez aussi apercevoir 2 boutons :

- Le bouton départ/ pause
- Le bouton de fin

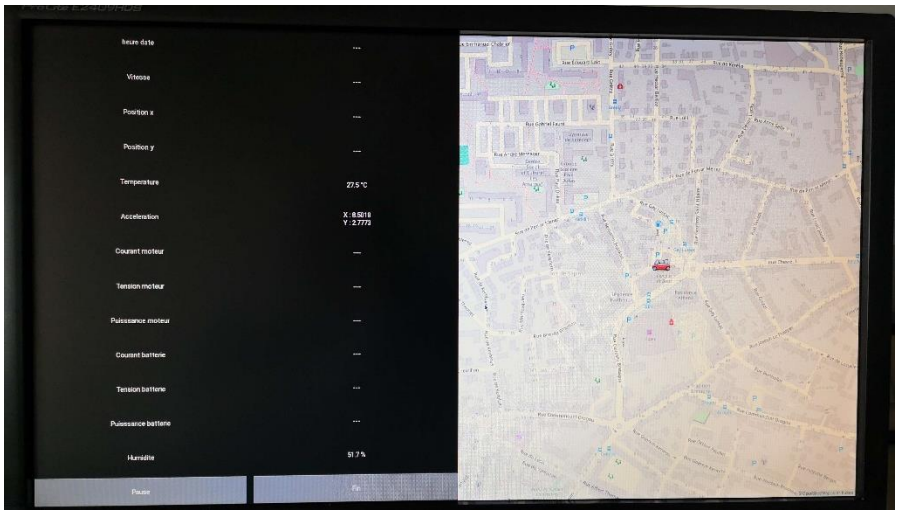


Figure 9 - IHM finale

Cependant, pour pouvoir afficher cette page-là, il faut avant tout vérifier si le numéro du trajet rentré par l'utilisateur correspond bien au trajet qui va être effectué. Pour cela, une deuxième interface graphique permet cette vérification.

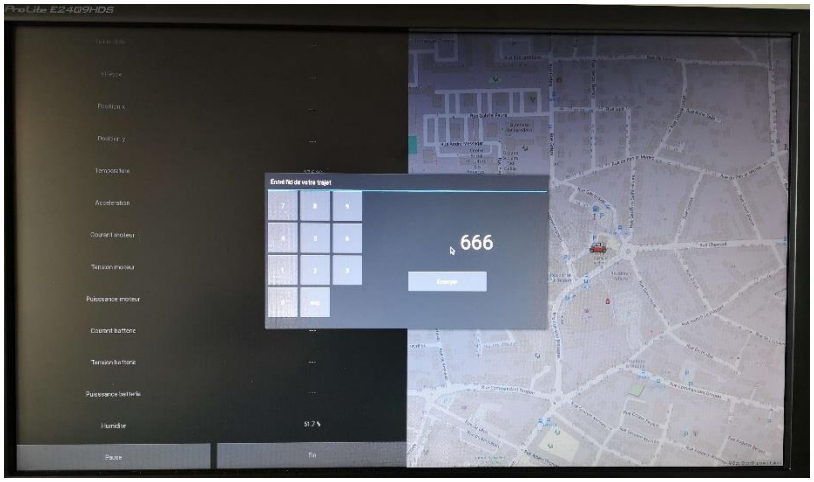


Figure 10 - Connexion à l'IHM

Comme vous pouvez le voir ci-dessus, au lancement du programme, un programme est lancé afin de demander à l'utilisateur de rentrer le numéro de son trajet.

Si le numéro de trajet correspond à un trajet enregistré pour la date actuelle, le trajet est validé et pourra commencer. Dans le cas contraire, la page vous indiquera un numéro incorrect, et ne vous permettra pas d'atteindre la page d'après.

Lors de l'appuie sur le bouton de **Fin** de trajet, une nouvelle page s'affiche afin de préciser à l'utilisateur l'état d'avancement du transfert des données relative au trajet vers le serveur. Pour cela, il existe 4 états d'avancement :

- Confirmation de fin de trajet (début du transfert)
- L'envoie des données en cours (attente)
- Bonne transmission des données (fin de transfert)
- Echec d'envoi

Si le transfert s’est bien passé, vous devriez voir s’afficher cette page-là.

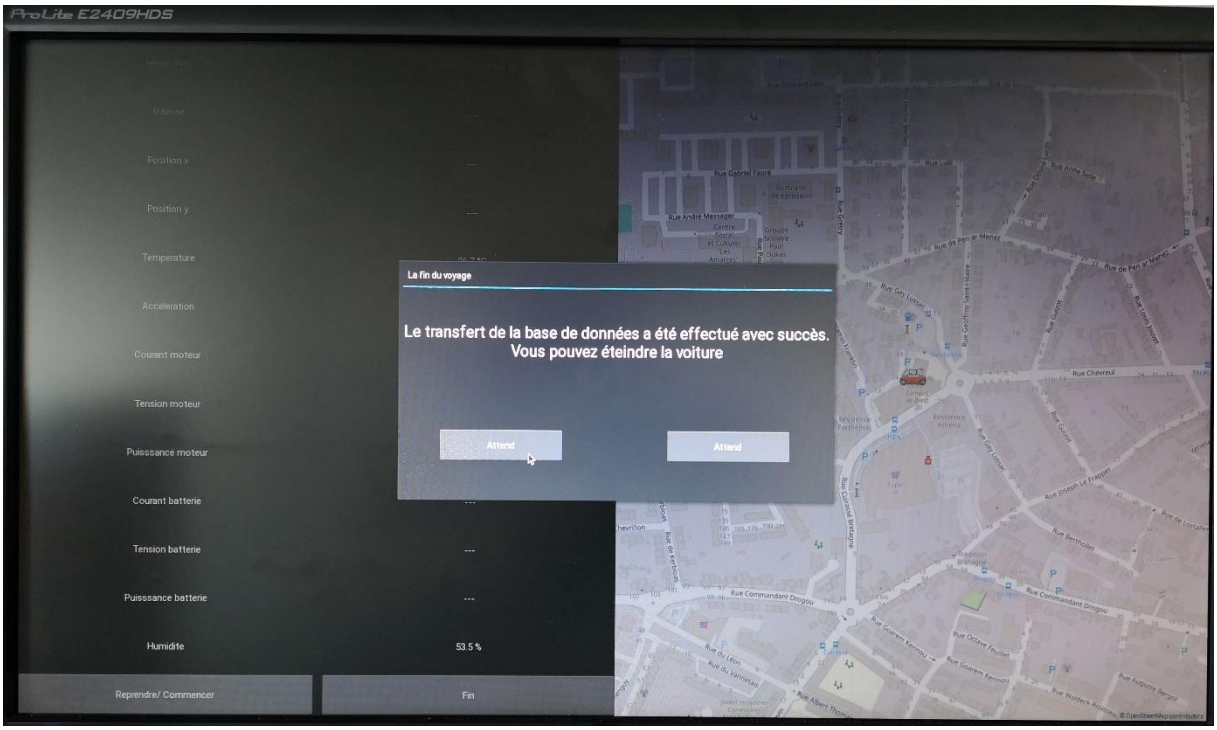


Figure 11 - Page de validation du transfert des données

Vous pouvez désormais éteindre le véhicule. Les données concernant le trajet sont maintenant enregistrées sur la base de données externe du serveur ISEN, et seront réutilisables depuis le site internet.

8.7.5 Envoi des données

En ce qui concerne l’envoi des données, 2 liens de connexion doivent être créés. Un avec la base de données interne à la Raspberry, et un autre avec la base de données du serveur distant.

La fonction du premier lien est de permettre la récupération des données, liées au numéro de trajet, qui seront ensuite envoyés dans la base de données du serveur grâce au deuxième lien.

8.7.6 Procédure d’utilisation de l’IHM

Pour comprendre au mieux le principe de fonctionnement de l’IHM, voici un diagramme UML décrivant ses différentes fonctionnalités.

Voir ANNEXE

8.7.7 Lancement automatique

Pour permettre le lancement automatique du programme, il a été nécessaire de créer un service Linux qui tente d'exécuter le programme et qui, si le programme est fermé ou n'a pas pu être lancé, tente de le relancer toutes les 3 secondes.

Afin que ce service soit exécuté au lancement de la Raspberry, il a fallu autoriser le lancement automatique du service.

8.8 Site internet

Dans le cadre de ce projet, l'utilisation du véhicule électrique de l'ISEN nécessite sa réservation. C'est pourquoi, le site **fcity.isen** a été programmé. Mais comment faire pour que ce site soit visible de tous ?

En effet, afin de rendre cette partie du projet accessible à toutes et tous, il a fallu héberger ce site internet sur le serveur de l'ISEN. Par conséquent, n'importe quelle personne étant reliée au réseau de l'ISEN a la capacité de s'y connecter.

Cependant, cela n'est pas possible sans passer par ces différentes étapes :

- Connexion au serveur en ssh
- Installation de mysql, apache, symfony et git
- Copie sécurisée du programme réalisé sur la machine locale vers le serveur
- Création d'un Virtual Host afin de configurer le site internet
- Configuration du DNS pour accéder au site internet
- Configuration de la base de données

Après avoir réalisé ce protocole, vous pouvez désormais vous connecter en toute simplicité au site internet via le lien ci-dessous : fcity.isen.

8.8.1 Connexion de l'utilisateur

Le lien ci-dessus, permettant la connexion au site internet, renvoie l'utilisateur sur la page de connexion au site, que voici.

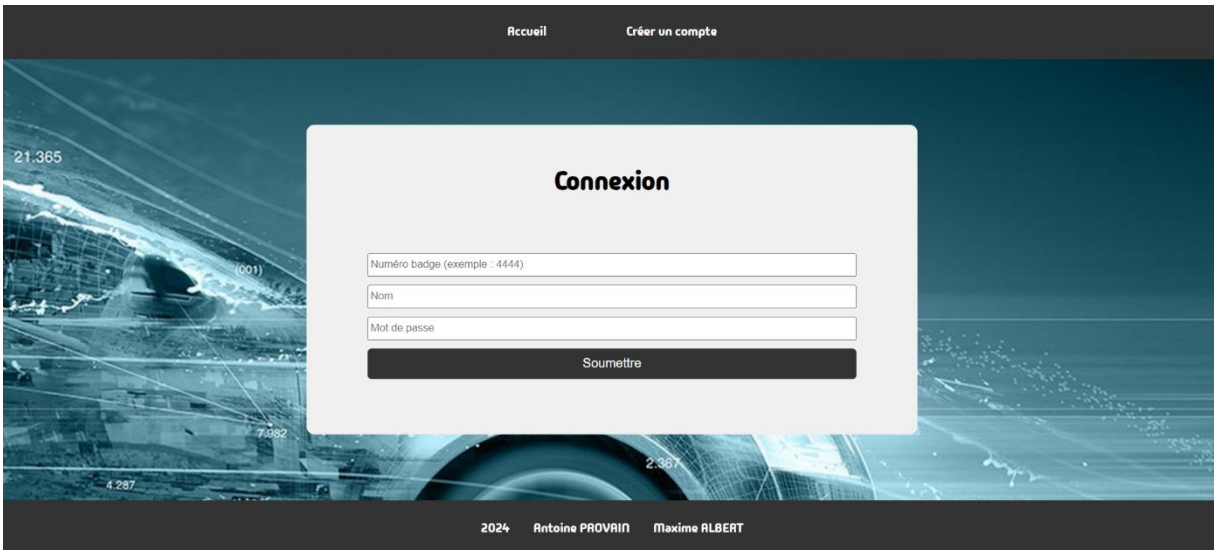


Figure 12 - Page de connexion

Cependant, la connexion sera impossible sans la création du compte utilisateur. Le test suivant est fait avec l'utilisateur « maxime », numéro de badge 5432 et de mot de passe azerty, qui servira d'exemple pendant toute la démonstration. Cet utilisateur n'existe pas et renvoie donc l'erreur suivante :

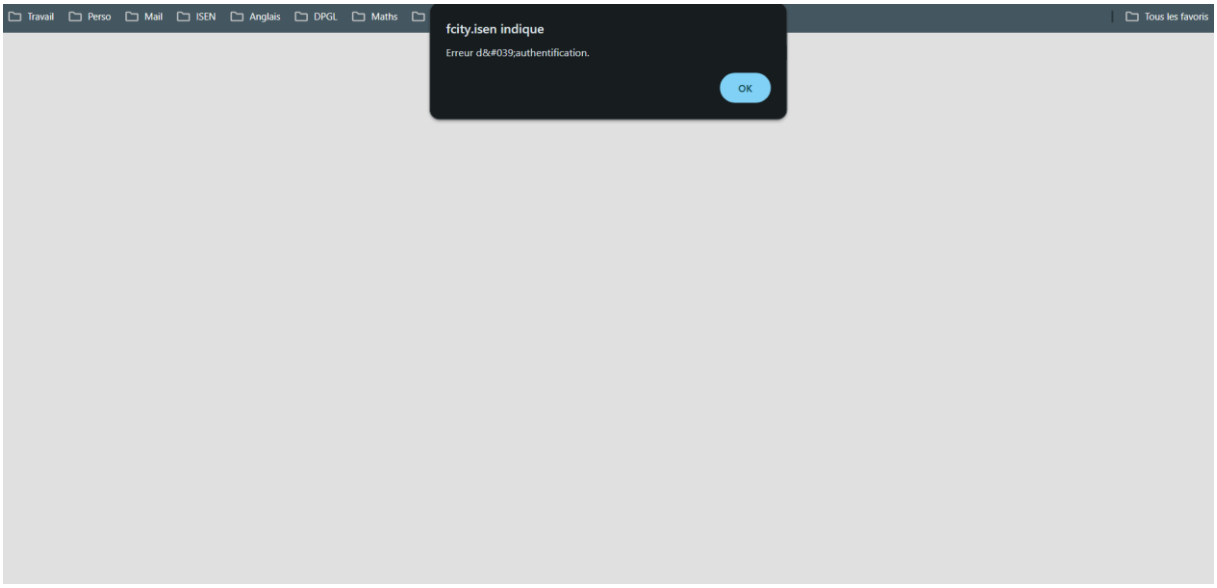


Figure 13 - Page d'erreur de connexion

Dans le cas contraire, vous êtes renvoyé à la page d'accueil du site. Pour remédier à ce problème, il suffit de créer le compte pour l'utilisateur « maxime ».

8.8.2 Création d'un compte utilisateur

La page d'inscription est la suivante.

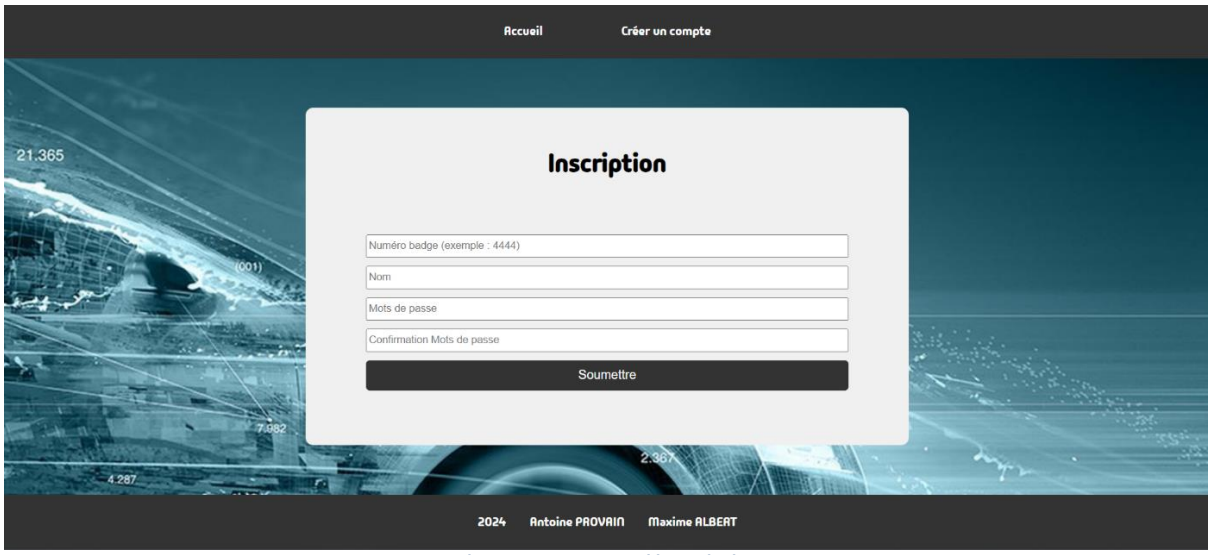


Figure 14 - Page d'inscription

Pour s'inscrire, rien de plus facile. Il suffit de rentrer son numéro de badge, numéro noté au dos de votre carte ISEN, comme le montre la figure ci-dessous :

Figure 15 - Numéro de badge

Il ne reste plus qu'à remplir le nom d'utilisateur, de donner un mot et de le reconfirmer.

Une fois la création de compte terminée, il est possible de voir affichées les nouvelles informations concernant l'utilisateur « maxime » dans la base de données.

username	mdp	badge
hun	$2y$10$uILt/tTILt0B6GBWuDMizeWVvzhS27.B.y.1Nw9kl7JchlFy4Zxfa	1111
antoine	$2y$10$Qro54L5rQY3qbAvjYe5bNei5xL68Ic0KmTr0bBQOTIdG/p43cPKPy	1234
john_doe	motdepasse123	123456
john_doe	motdepasse123	2345
maxime	$2y$10$o5QPVq.6mYSQetMzFuEdjeWnzecmUfB3HM1D4RWqitdp/kW2Hpyzq	5432

Figure 16 - Insertion dans la table user

Pour une meilleure sécurité, le mot de passe est crypté en base de données.

8.8.3 Réservation d'un trajet

Une fois le compte de « maxime » créé, et que sa connexion a bien été réussie, voyons voir pour réserver son premier trajet.

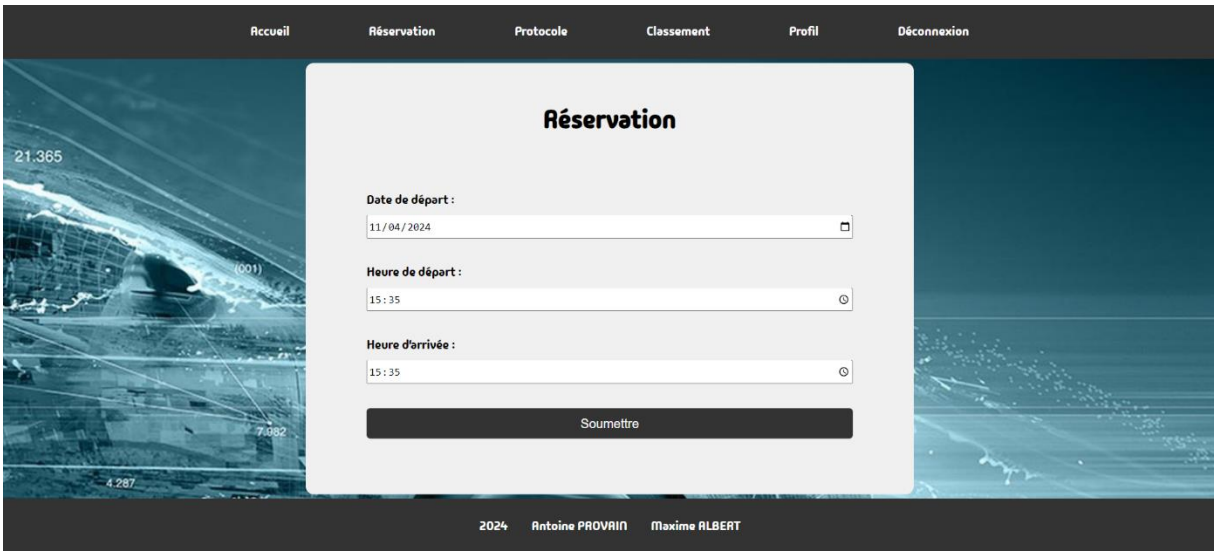


Figure 17 - Page de réservation

Comme montré sur l'image ci-dessus, lors de la réservation, « maxime » n'a qu'à rentrer la date de départ de son trajet, ainsi que son heure de départ et de retour. Grâce à des fonctions codées en PHP, il y a certaines vérifications à faire avant de valider le trajet de « maxime » :

- Vérification de la date de départ (pas avant la date actuelle)
- Vérification de l'heure de départ (pas avant l'heure actuelle si le trajet est le jour même)
- Vérification de l'heure d'arrivée (pas de retour avant l'heure de départ)
- Vérification des réservations déjà en base de données

Si toutes ces informations sont vérifiées, la réservation est enregistrée dans la base de données de cette manière.

Prenons l'exemple d'un trajet le 13/04/2024, de 16h51 à 17h51.

id	badge	jour	depart	arrivee
22	1234	2024-03-13	08:00:00	09:00:00
24	1234	2024-03-28	15:36:00	16:34:00
25	1234	2024-04-11	13:53:00	16:53:00
26	1234	2024-04-10	11:15:00	12:00:00
27	1234	2024-04-10	15:05:00	18:05:00
28	5432	2024-04-13	16:51:00	17:51:00

Figure 18 - Insertion dans la table trajet

Le trajet à bien été implémenté dans la table trajet avec le numéro de badge de maxime (le numéro 5432). « maxime » à son premier trajet !

8.8.4 Protocole d'utilisation du véhicule

Lorsque le trajet est validé, le site renvoie « maxime » directement vers une page de protocole d'utilisation du véhicule.

Figure 19 - Page de protocole d'utilisation

Pour que « maxime » comprenne au mieux la manière de faire fonctionner le véhicule, voici une page consacrée à l'explication des différentes étapes pour y arriver. Allant de la réservation jusqu'à l'arrêt total de la voiture.

8.8.5 Visualisation d'un trajet

La visualisation d'un trajet s'apparente à une page de profil, c'est d'ailleurs le nom que « maxime » sera amené à utiliser s'il souhaite visualiser les informations de son compte. Il y a une deuxième manière d'y accéder, en appuyant sur le bouton « Voir mes réservations » sur la page d'accueil du site.

Cette page affiche les informations telles que le nom d'utilisateur et le numéro de badge de « maxime ». Dans son cas, « maxime » et 5432.

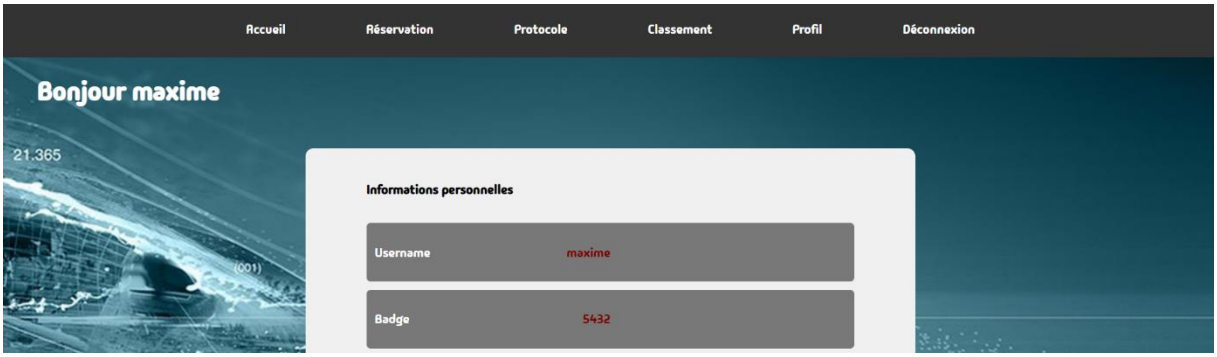


Figure 20 - Page de profil

Il y a ensuite deux sections différentes pour les trajets :

- Trajets passés
- Trajets futurs

En ce qui concerne l'affichage des trajets passés, il s'agit d'une requête PHP vers la base de données permettant de comparer les dates des trajets à la date actuelle.

Mais cela ne suffit pas, il faut aussi comparer les heures de départ avec l'heure actuelle et la date d'arrivée avec l'heure actuelle afin de savoir si le trajet est passé, futur ou en cours.

Le rendu final de cette partie de profil avec la visualisation des trajets de « maxime » est la suivante :

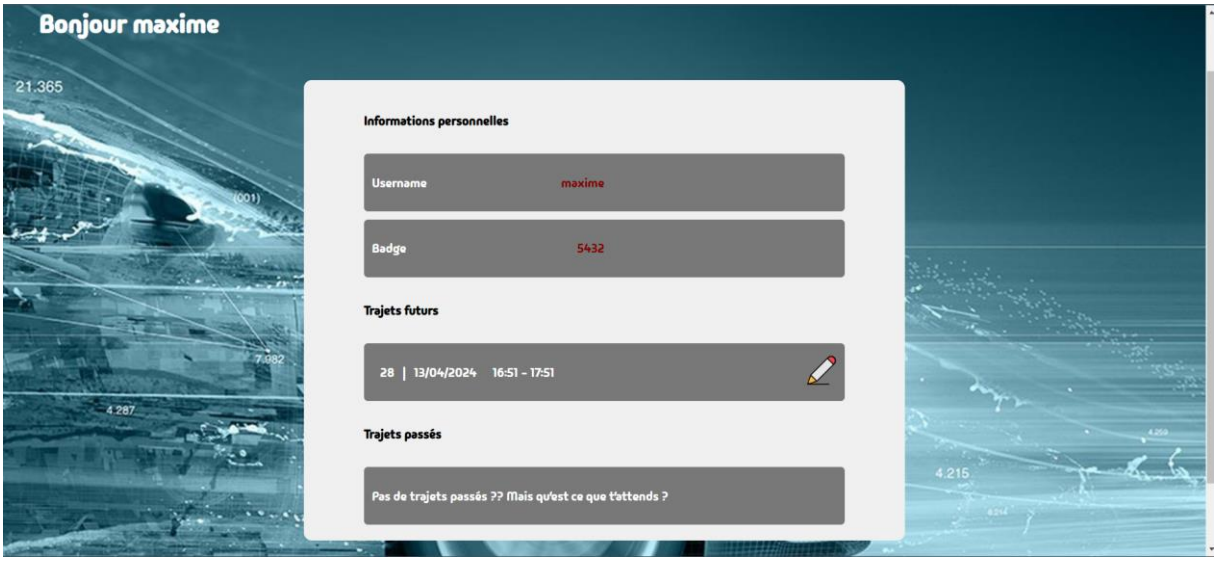


Figure 21 - Page d'affichage des trajets

Les informations que « maxime » peut voir concernant son trajet sont les suivantes :

- Le numéro de trajet
- La date de réservation du trajet
- Le créneau horaire (heure de début et de fin du trajet)

Dans le cas où « maxime » n'a ni trajets passés ni trajets futurs, voici l'affichage que cela donnerait.

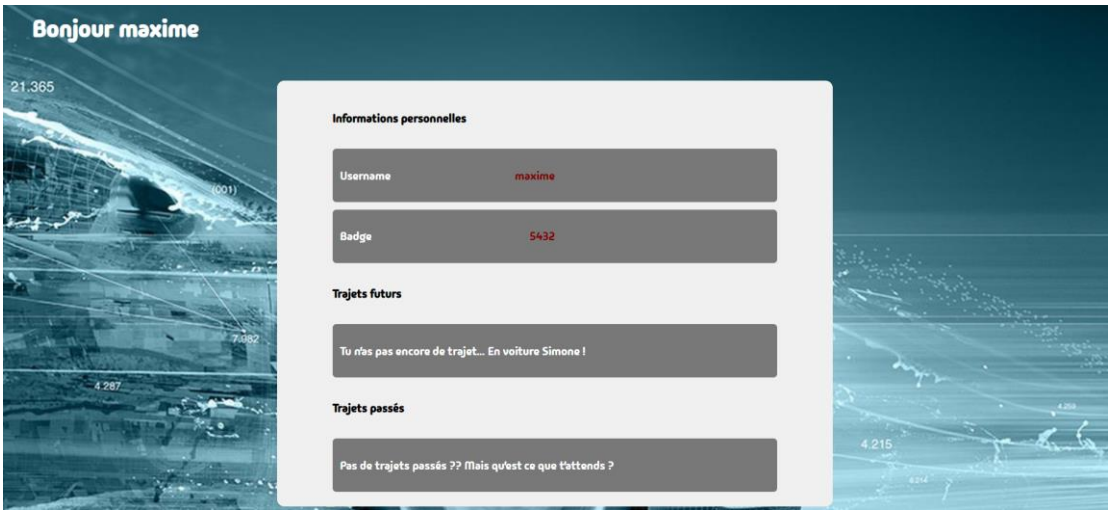


Figure 22 - Page sans réservation

8.8.6 Modification/ Suppression d'un trajet

La suppression et/ ou la modification d'un trajet peut se faire une fois la réservation faite. Le trajet réservé par « maxime » illustrera parfaitement cela.

Figure 23 – Affichage d'un trajet

Une fois la réservation effectuée, il est possible à « maxime » de visualiser les informations relatives à son trajet sur son profil. De plus, un bouton représentant un stylo a été placé de telle sorte que « maxime » puisse y effectuer des modifications. Voici à quoi cela ressemblerait.

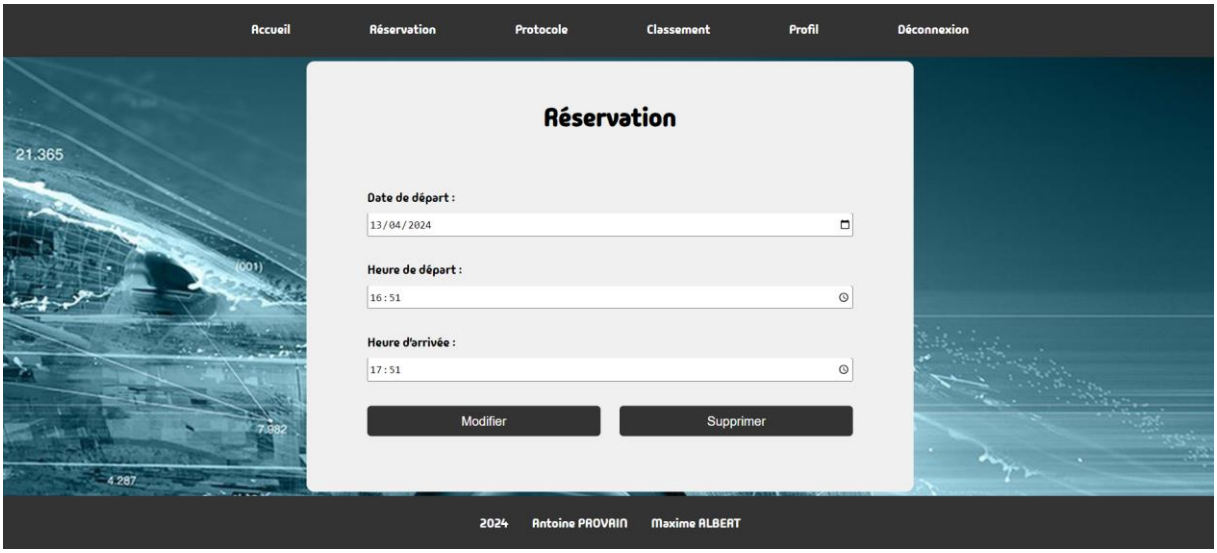


Figure 24 - Page de suppression/ modification

Pour reprendre les informations exactes du trajet que « maxime » veut modifier, le numéro de trajet a été volontairement placé dans l'URL de la page. C'est de cette façon que la date, l'heure de départ et de fin correspondent aux informations affichées sur la page de profil.

Pour modifier ce trajet, il suffit de changer les paramètres présents dans le formulaire ci-dessus. Cette opération une fois effectuée, il ne restera plus qu'à appuyer sur le bouton « Modifier ». Pour « maxime », il s'est trompé de jour, mais pas de panique ! Il souhaitait le 15/06/2024 plutôt que le 13/04/2024. Il n'a plus qu'à essayer.

Si la date choisie et les heures de départ et d'arrivée sont validées, le trajet est modifié. Pour vérifier si le nouveau trajet est conforme, voyons voir dans le profil.

Figure 25 - Modification du trajet

Pour la partie de suppression d'un trajet, cela part du même principe que pour la modification. En effet, un seul appui sur le bouton « Suppression » permet à « maxime » d'effacer son trajet de la base de données. Une fois le trajet supprimé, plus qu'à vérifier sur le profil de « maxime ».

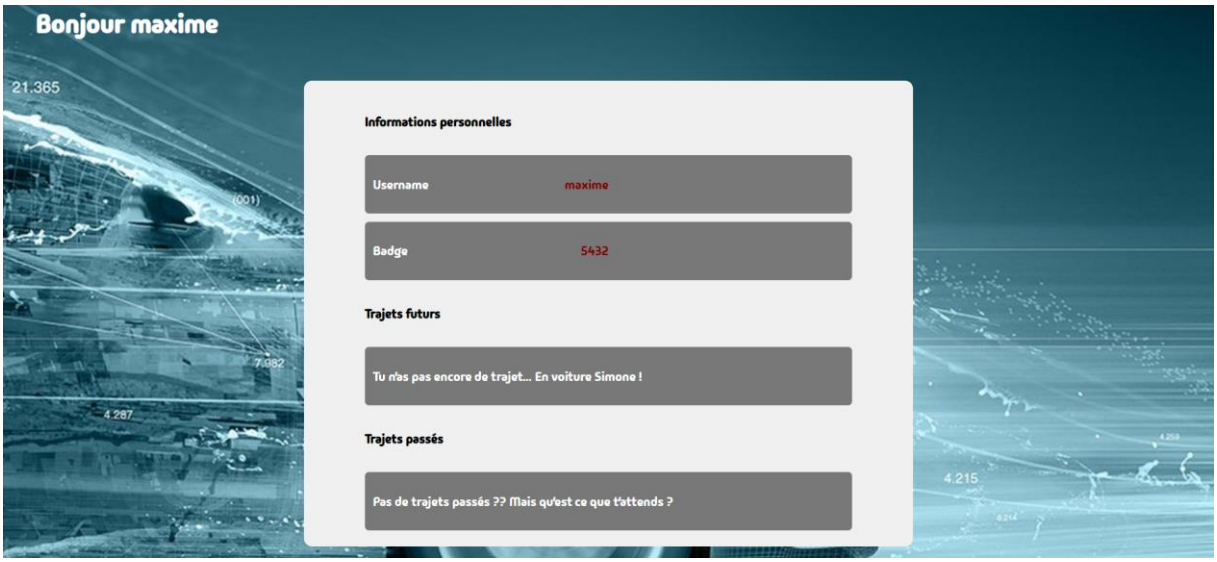


Figure 26 - Suppression d'un trajet

En retournant sur le profil de « maxime », il est possible de constater la suppression de son trajet du 15/06/2024. Cette vérification est aussi possible en base de données.

id	badge	jour	depart	arrivee
22	1234	2024-03-13	08:00:00	09:00:00
24	1234	2024-03-28	15:36:00	16:34:00
25	1234	2024-04-11	13:53:00	16:53:00
26	1234	2024-04-10	11:15:00	12:00:00
27	1234	2024-04-10	15:05:00	18:05:00

Figure 27 - Suppression d'un trajet en base de données

Le trajet avec le numéro de badge de « maxime » (numéro 5432) a bien été supprimé, comme attendu.

8.8.7 Visualisation des graphiques

La visualisation des graphiques d'un trajet est disponible une fois que celui-ci est considéré comme « passé ». Effectivement, si un trajet n'est pas encore passé, ces données en question n'existent pas encore, car elles sont rentrées dans la base de données une fois le trajet fini.

Afin d'accéder à ces graphes, même principe que pour la modification/ suppression.

Figure 28 - Bouton de visualisation des graphes

Comme « maxime » peut le voir sur la figure ci-dessus, un bouton, sous la forme d'un œil, a été placé pour qu'il puisse ainsi visualiser les données relatives à son trajet numéro 22.

Lorsque « maxime » appuie sur le bouton, le numéro de trajet a été renvoyé dans l'url du site, pour sélectionner les données du trajet en question. Pour cela, une fonction a été nécessaire afin de rassembler les données en base de données en fonction de leur numéro de trajet. Chaque numéro de trajet est unique, et permet donc d'être identifié de manière unique lorsque les données du trajet sont rentrées en base de données.

NULL	NULL	2.6791	3.9576	25.0	NULL	24	NULL	NULL
NULL	NULL	2.7414	3.8595	25.1	NULL	24	NULL	NULL
NULL	NULL	2.6983	3.948	25.0	NULL	24	NULL	NULL
NULL	NULL	2.7222	3.9385	25.0	NULL	24	NULL	NULL
NULL	NULL	2.6217	3.9145	25.0	NULL	24	NULL	NULL
NULL	NULL	2.7222	3.8666	25.1	NULL	24	NULL	NULL
NULL	NULL	2.7054	3.8906	25.0	NULL	24	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.3	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.3	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.4	NULL	25	NULL	NULL
NULL	NULL	NULL	NULL	24.3	NULL	25	NULL	NULL

Figure 29 - Table des capteurs

Lors de l'utilisation du véhicule, le numéro de trajet rentré par l'utilisateur sera inséré dans la base de données au même moment que les données des capteurs. De cette manière, toutes les lignes de la base de données, dans la table capteur, possédant le même numéro de trajet seront alors identifiés comme étant du même trajet.

Si « maxime » utilise le véhicule, avec comme numéro de trajet 29, et qu'il y a 120 valeurs dans la base de données portant le numéro 29, alors toutes ces données concernent ce parcours de « maxime ». C'est grâce à elles que les graphiques sur le site internet sont construits.

Voici ce que renvoie l'appui sur ce bouton.

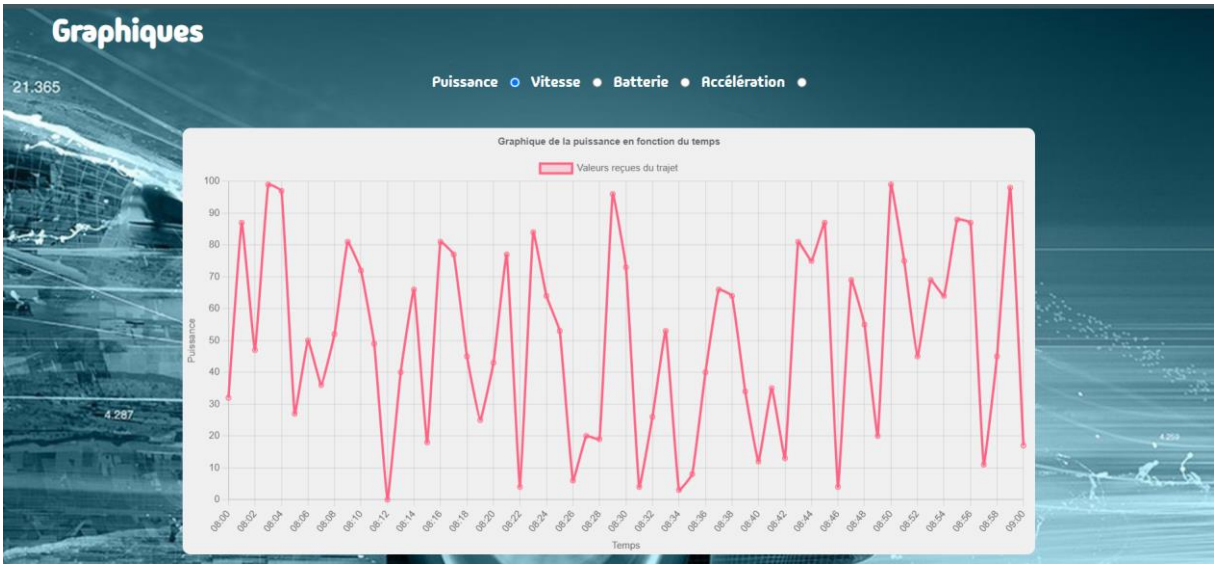


Figure 30 - Visualisation des graphiques

Sur cette page, « maxime » a le choix d'afficher les données de 4 variables différentes :

- La puissance du véhicule en fonction du temps
- La vitesse du véhicule en fonction du temps
- La batterie du véhicule en fonction du temps
- L'accélération du véhicule en fonction du temps

Pour afficher les données demandées par « maxime », il suffit de changer de tableau de données en fonction du bouton choisi par « maxime ». Si « maxime » choisit d'afficher « puissance », c'est le tableau des valeurs de puissance qui sera sélectionné. Si c'est « batterie » qui est choisi, alors le tableau des valeurs de la batterie sera sélectionné, et ainsi de suite.

L'axe des ordonnées représente les valeurs prisent par la puissance lors du trajet, tandis que l'axe des abscisses représente l'heure de départ et d'arrivée du trajet. Le trajet 22, l'illustre parfaitement. Enregistré entre 8h et 9h dans le profil, voici le résultat une fois le bouton cliqué, sur l'axe des abscisses.

Figure 31 - Axe des abscisses des graphes

Le créneau horaire est bien retranscrit sur le graphique.

8.8.8 Visualisation du classement des utilisateurs les plus économes

Lors de la récupération dans la base de données, chaque trajet est identifié par son numéro dans la table des capteurs. Une fois ces données séparées par numéro de trajet, pour connaître les utilisateurs les plus économes, il faut calculer la puissance de la batterie en multipliant la tension et l'intensité de la batterie, pour chaque ligne de la base de données, et en faire la moyenne. Ainsi nous retrouvons la liste des utilisateurs qui ont le moins consommé.

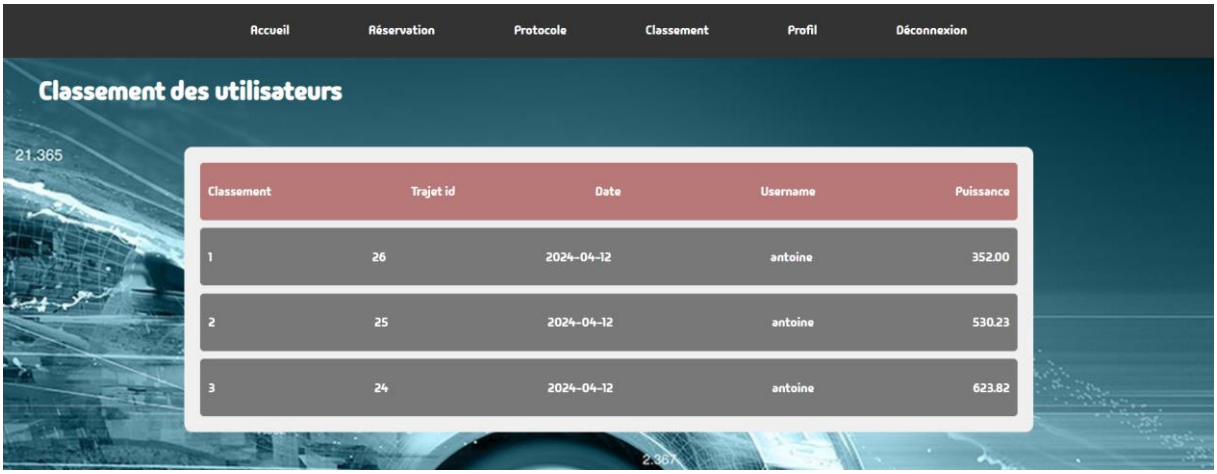


Figure 32 - Page du classement

Voici la liste des utilisateurs les plus économes, avec affichée la position, le numéro de trajet, la date du trajet, le nom d'utilisateur et la puissance moyenne lors de ces trajets.

8.8.9 Déconnexion

Quand « maxime » a fini sa réservation, « maxime » se déconnecte et est renvoyé vers la page de connexion.

Plus qu'à faire comme « maxime », lancez-vous !

9. Conclusion

En conclusion, les objectifs qui avaient été fixés en début de projet sont pratiquement atteint. Le site internet et l'IHM sont fonctionnels, cependant la partie de développement d'un boitier contenant les câbles et les capteurs, afin de les protéger, pour rendre le projet « fini » n'a pas pu être réalisée dans les temps.

La partie concernant le site internet fonctionne correctement et est une partie essentielle du projet, car sans elle, il n'est pas possible de se servir de l'IHM. En effet, sans le numéro de trajet, visualisable sur la page de profil du site internet, il est donc impossible pour l'utilisateur d'avoir accès aux informations de l'IHM. Parmi les fonctionnalités complémentaires, l'utilisateur peut se servir du site pour avoir un retour sur son expérience après utilisation du véhicule, par le biais de graphiques (vitesse, accélération, batterie et puissance). Mais aussi grâce à un classement des utilisateurs les plus économes. La mise en place du protocole d'utilisation du véhicule permet aussi à l'hôte de mieux comprendre le fonctionnement de la F-City.

Quant à elle, l'IHM remplit entièrement son rôle, qui est en outre d'afficher toutes les données reçues par les capteurs. Le fait de demander le numéro de trajet au conducteur du véhicule est une alternative simple afin de vérifier que n'importe qui ne puisse pas utiliser le véhicule à sa guise. Il suffit de se munir de son numéro de badge ISEN pour s'en servir, via la réservation sur le site internet. Malheureusement, une partie peut être encore obsolète. En effet, l'affichage du GPS est fait, mais il faudrait charger toutes les données des lieux de Brest pour pouvoir afficher correctement la carte. Parmi les choses qui sont à remarquer, l'envoi des données concernant le trajet de l'utilisateur vers la base de données externe est une réussite, et permet ainsi l'affichage des graphiques sur la partie du site internet. Le lancement automatique du programme au démarrage de la Raspberry est aussi une réussite.

Séparer ce projet en deux parties, une partie IHM et une partie site internet, était donc prolifique car il a permis d'avancer à bon rythme sans avoir eu besoin l'un de l'autre. La réunification des codes à la fin de ces deux parties était essentielle pour vérifier le fonctionnement complet des deux programmes.

Etant un projet de grande envergure, ce projet permet aussi de nombreuses possibilités d'amélioration. En effet, parmi les améliorations possibles, réaliser un affichage graphique sur l'IHM plus grand permettrait à l'utilisateur de mieux percevoir les informations, mais aussi d'en afficher plus. Il serait donc possible de rajouter des capteurs supplémentaires (capteurs et caméra de recul). Il serait aussi possible de rajouter des panneaux solaires sur le toit de la voiture, lui permettant ainsi une plus grande autonomie.

10. Remerciement

Nous tenons à remercier M. Franck LE GALL, responsable des équipes F-City, pour son soutien et pour la confiance qu'il nous a accordée dans la réalisation de ce projet.

Nous tenons aussi à remercier Philippe FORJONEL et Christophe VIGNAUD pour leurs aides vis-à-vis de la récupération des données de la carte FPGA et de l'installation du site internet sur le serveur de l'ISEN.

11. Bibliographie

11.1 Raspberry Pi 4

Caractéristiques : https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

11.2 Base de données

Les commandes : https://buzut.net/maitrisez-mysql-en-cli/

11.3 Partie Serveur/ Web

Installation et configuration d'apache : https://httpd.apache.org/docs/2.4/fr/install.html
Installation de git : https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Installation-de-Git
Installation symfony : https://symfony.com/doc/current/setup.html

Vhost : https://adventy.org/fr/tutoriel/comment-creer-et-configurer-un-hote-virtuel-sur-apache

Aide vue twig : https://openclassrooms.com/fr/courses/5489656-construisez-un-site-web-a-l-aide-du-framework-symfony-5/5517021-dynamisez-vos-vues-a-l-aide-de-twig

Aide php : https://www.php.net/manual/fr/index.php

Aide JS : https://developer.mozilla.org/fr/docs/Web/JavaScript

Controller PHP : https://symfony.com/doc/current/controller.html

11.4 Partie IHM

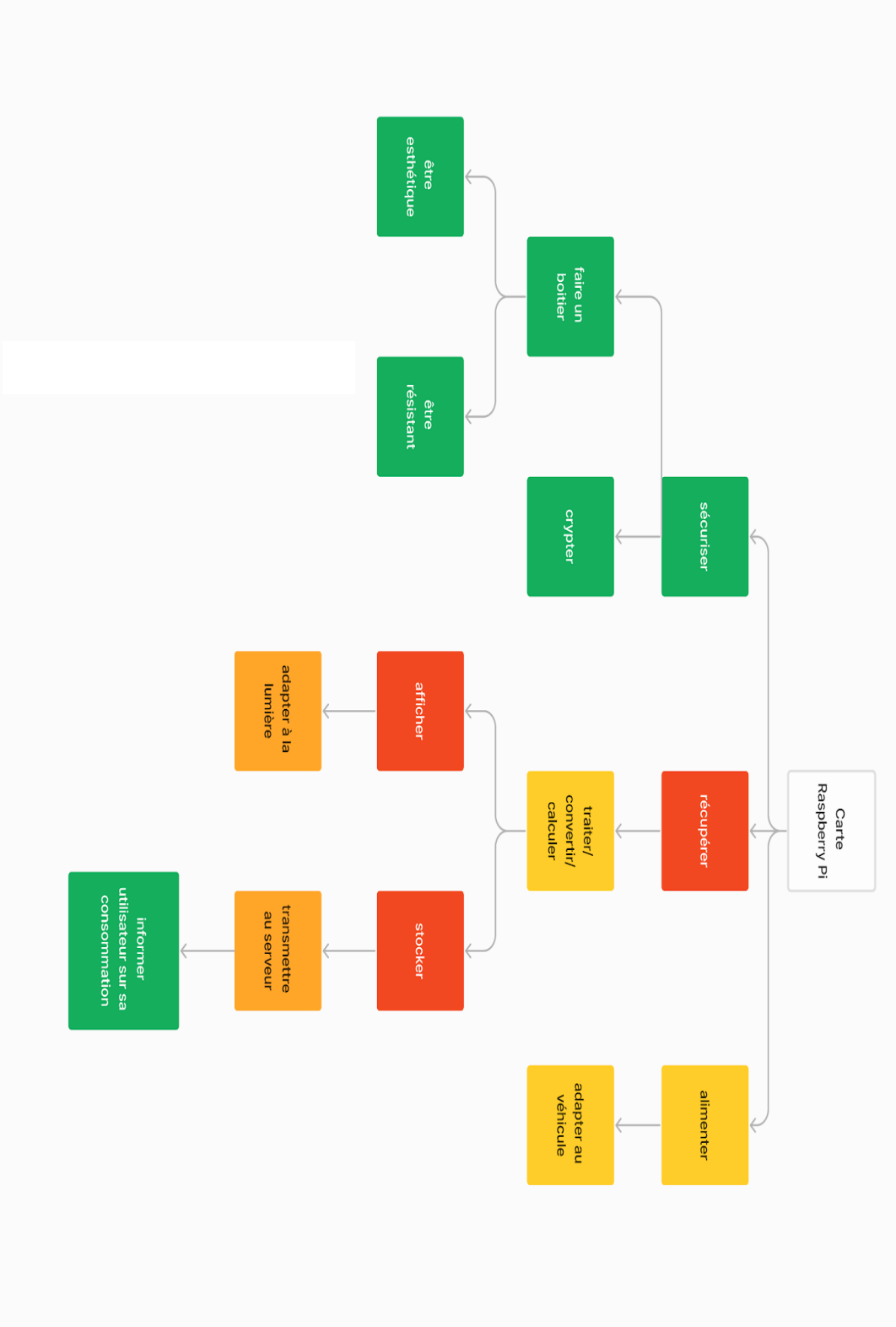
Configuration UART : Raspberry Pi Documentation - Configuration
Configuration I2C : https://www.siloged.fr/docs/raspberry/index.html?I2C.html

Librairies python :

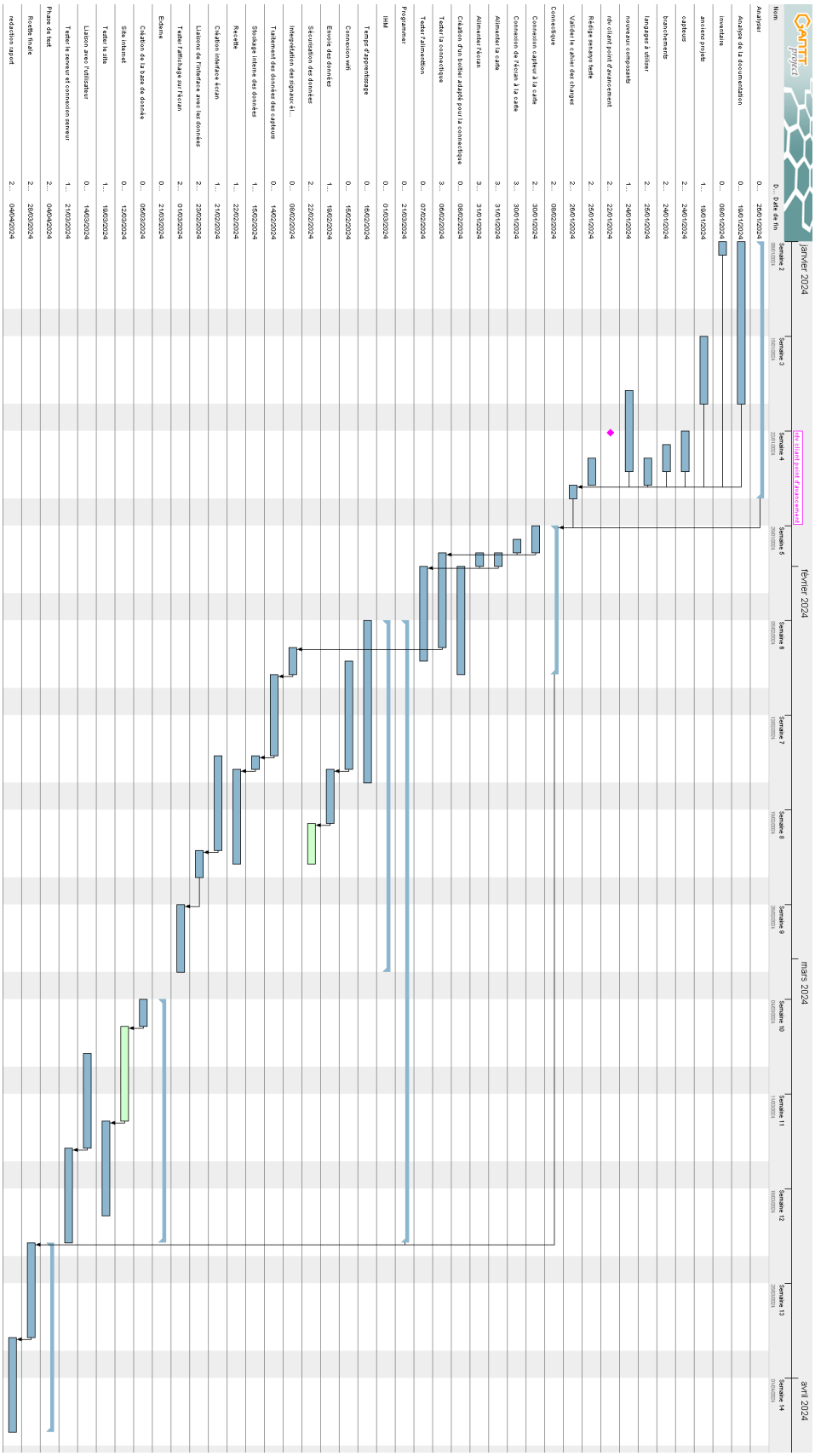
Kivy : https://docs.python.org/3/library/subprocess.html
Subprocess : https://docs.python.org/3/library/subprocess.html
Serial : https://pyserial.readthedocs.io/en/latest/
Adafruit_am2320 : https://docs.circuitpython.org/projects/am2320/en/latest/
Adafruit_mpu5060 : https://github.com/adafruit/Adafruit_MPU6050
GPS kivy : https://github.com/kivy-garden/garden.mapview

13. Annexe

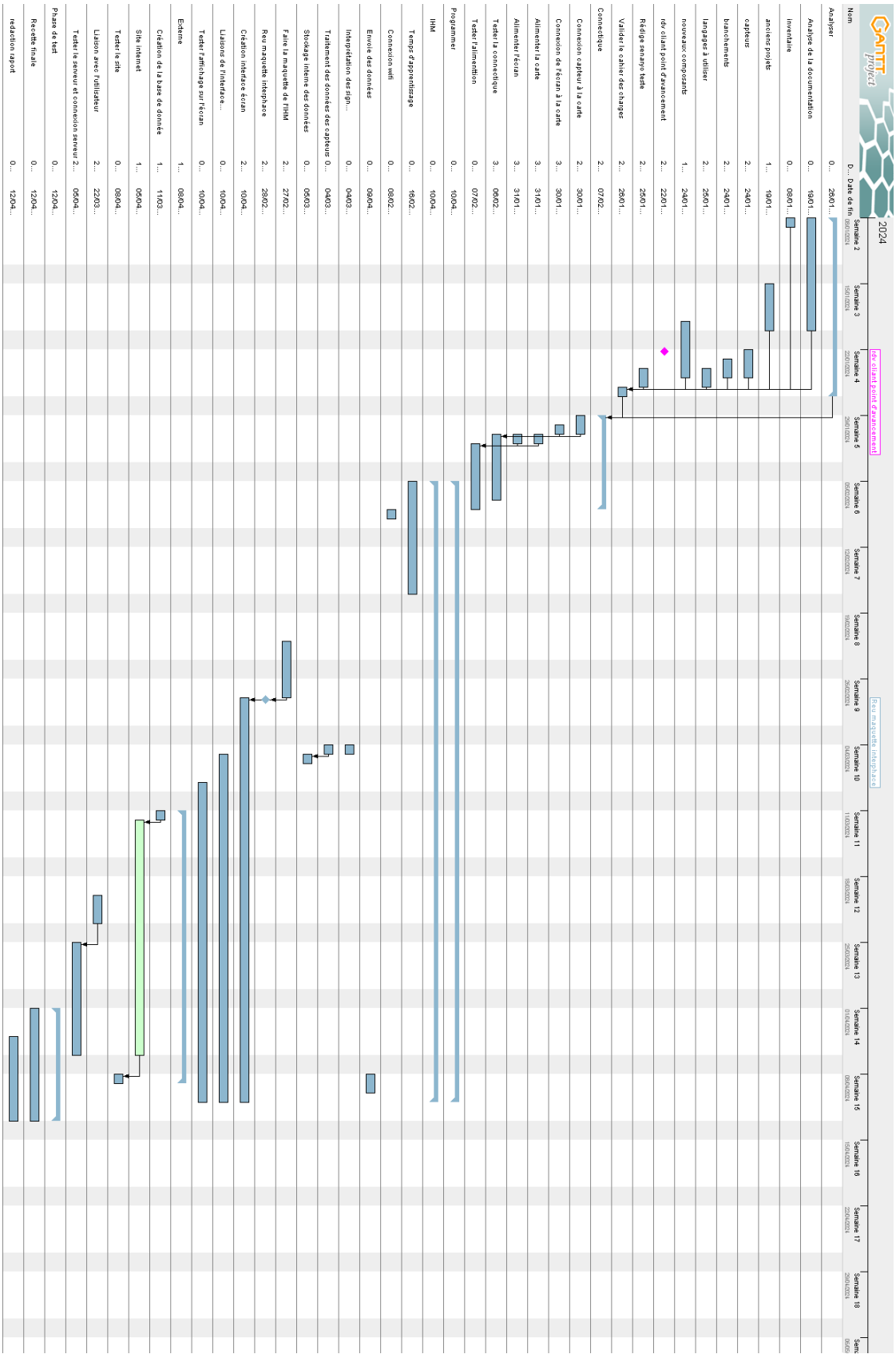
13.1 Taxonomie des fonctions



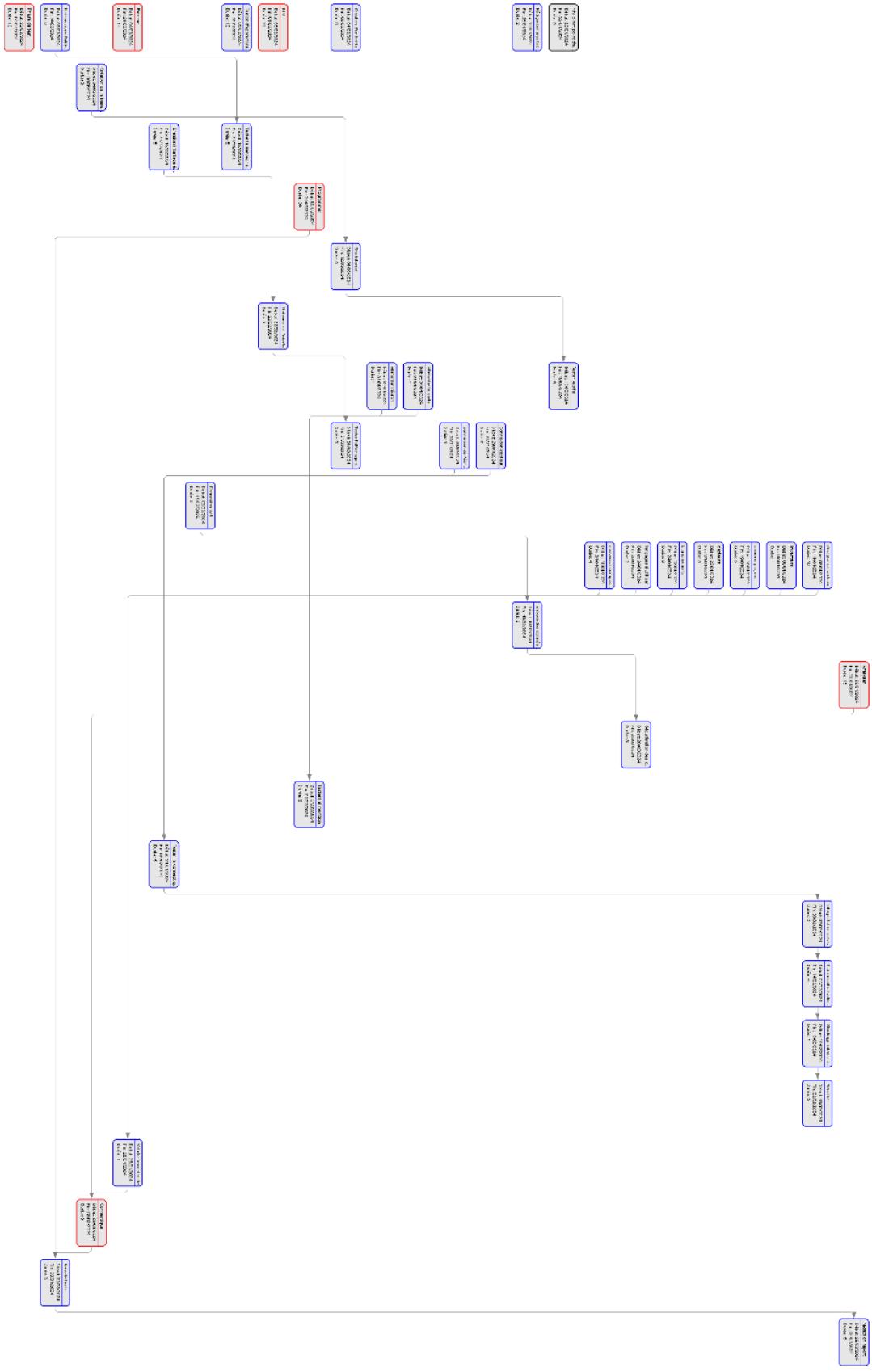
13.2 Diagramme de Gantt en début de projet



13.3 Diagramme de Gantt à postériori



13.4 Diagramme de Pert



13.5 Diagramme UML