MPSI JEAN XXIII
Mai 2024

Travail d'Initiative Personnelle Encadré

Gr. 3

metz campus

GINOT Louis
PONCHIN Alban
SANGARIEVA Safiat
SARRA Alexis

Problématique

Permettre à un pilote de connaître précisément la température, l'humidité la pression, l'altitude ainsi que le tracé de son vol.

Ce que nous avons réalisé

à l'intérieur du cockpit

Recueillir — Récupération des données physiques (température, pression, humidité, altitude, position GPS)

Informer — Affichage en temps réel sur un écran + enregistrement

Analyser — Récapitulatif du vol et détails sur un site internet

Analyse de l’existant

Batterie intégrée, autonomie	1400mAh 2.5h
GPS Intégré	Oui
Capteur de pression	Non
Taille	135*86*14 mm

Oudie 2 - Naviter

1. Interrupteur d'alimentation
2. Indicateur de charge
3. Prise casque
4. Emplacement pour carte microSD
5. connecteur mini USB
6. Stylus
7. Haut-parleur
8. Bouton de réinitialisation

Diagrammes

Diagramme des cas d'utilisation

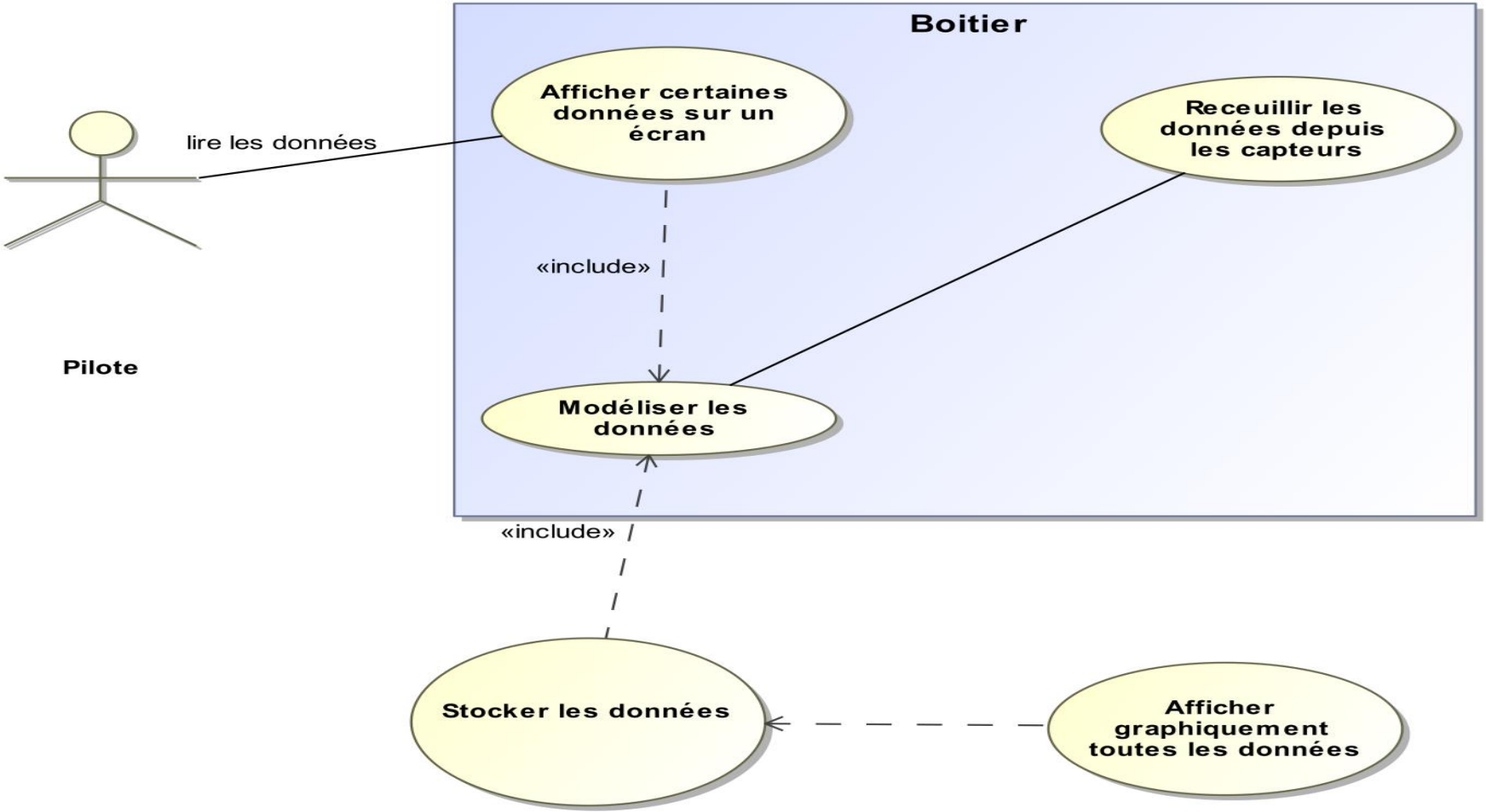


Diagramme des exigences

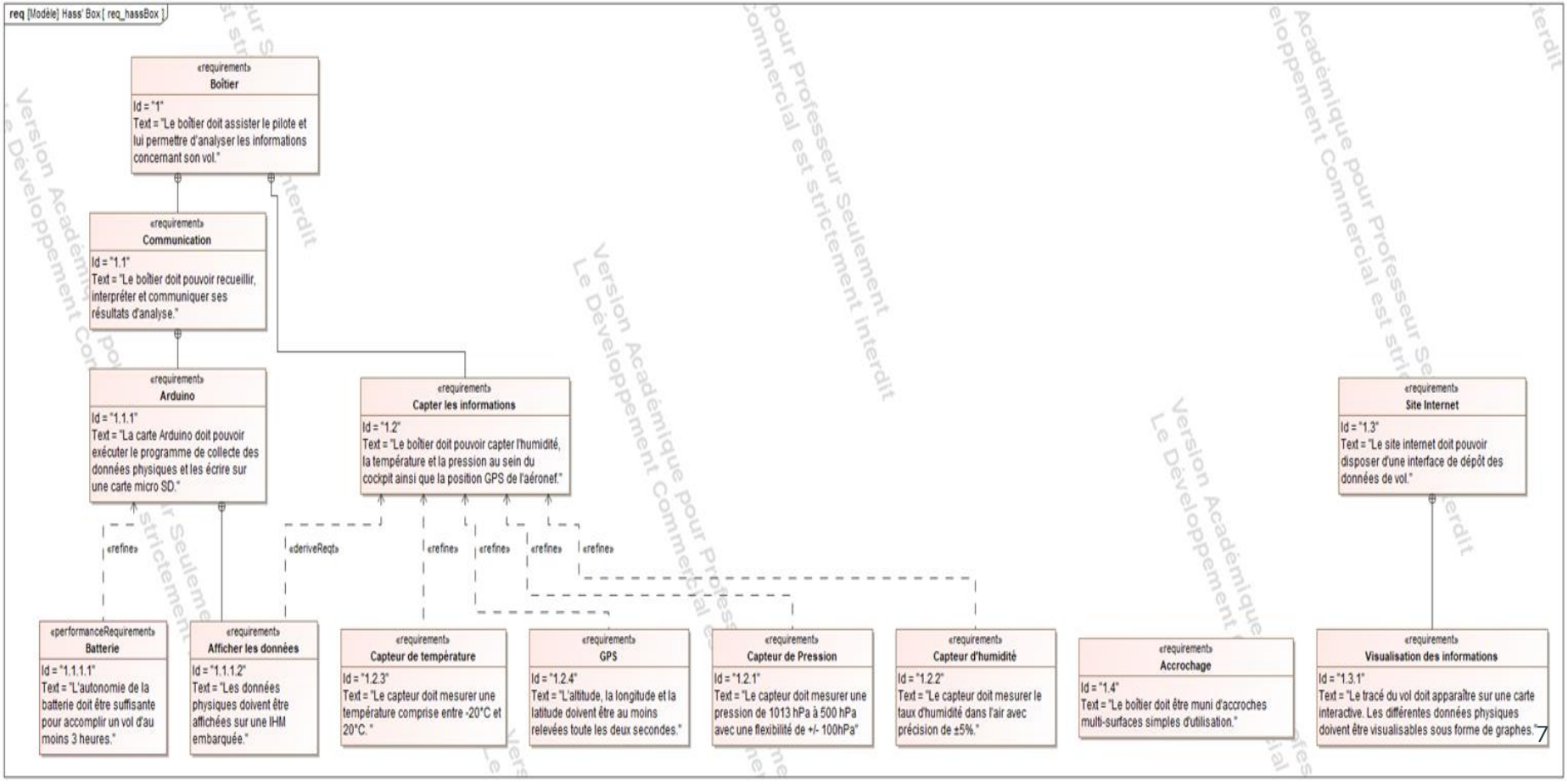


Diagramme de blocs

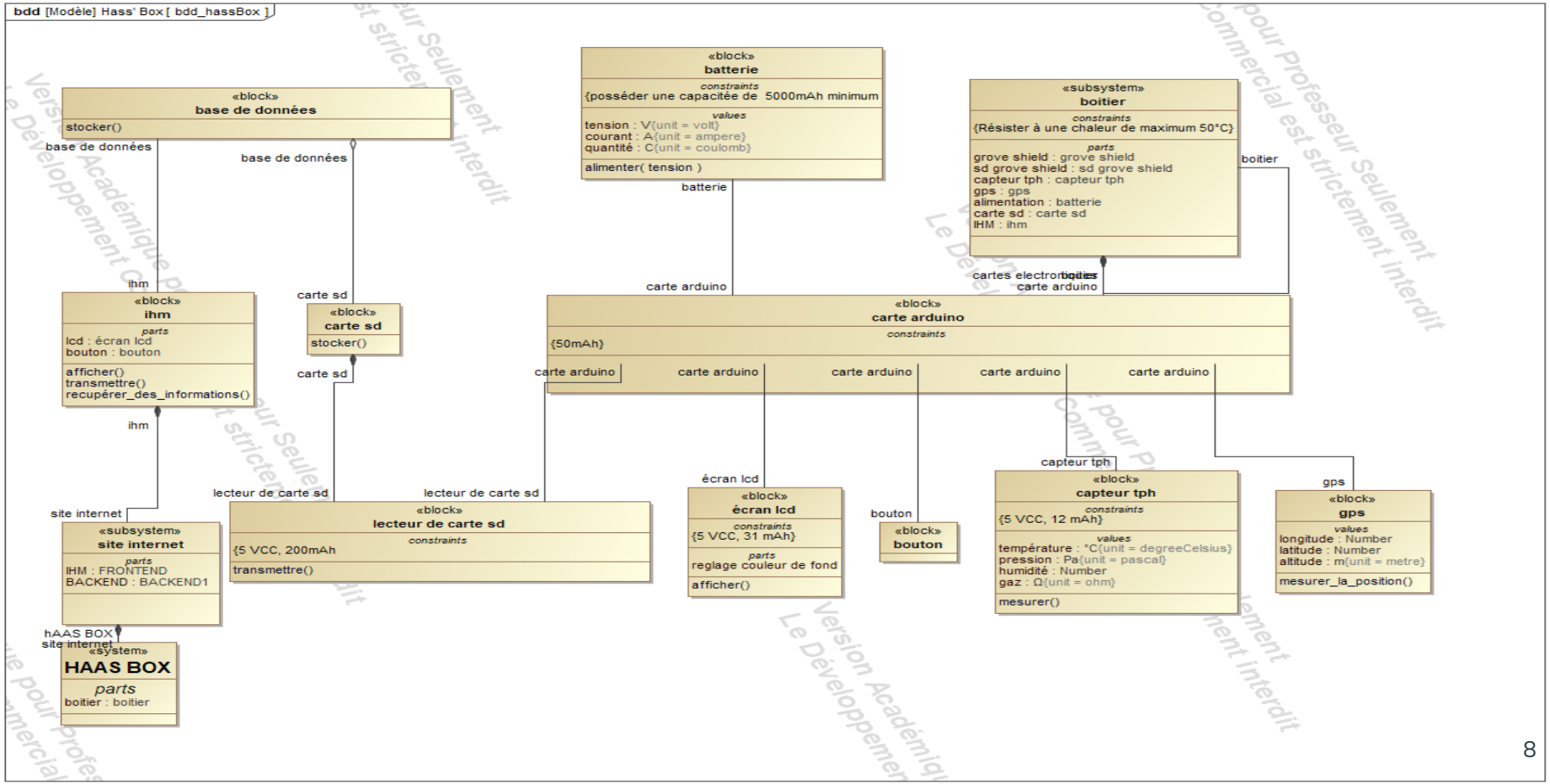


Diagramme de blocs internes : boîtier

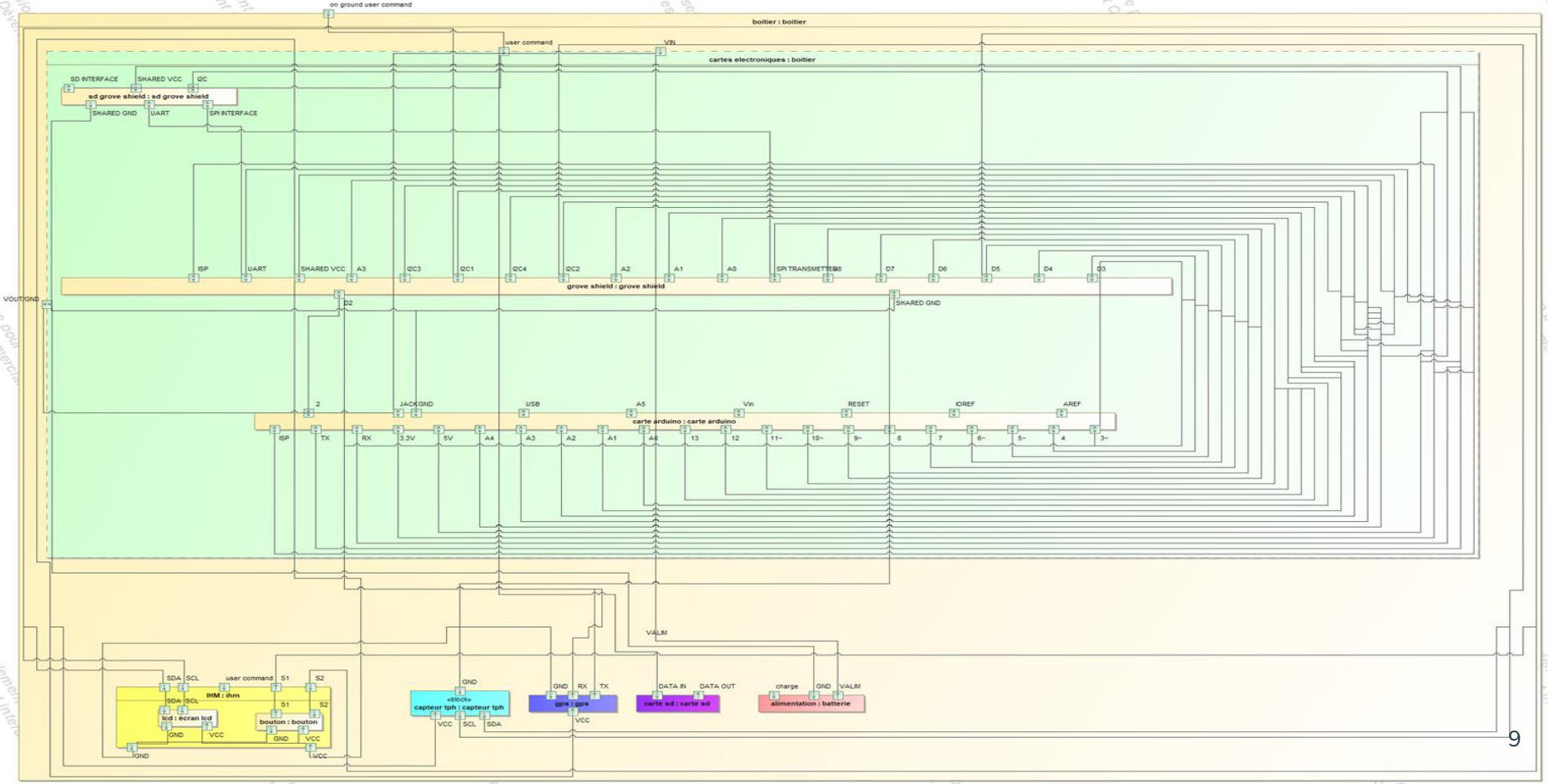
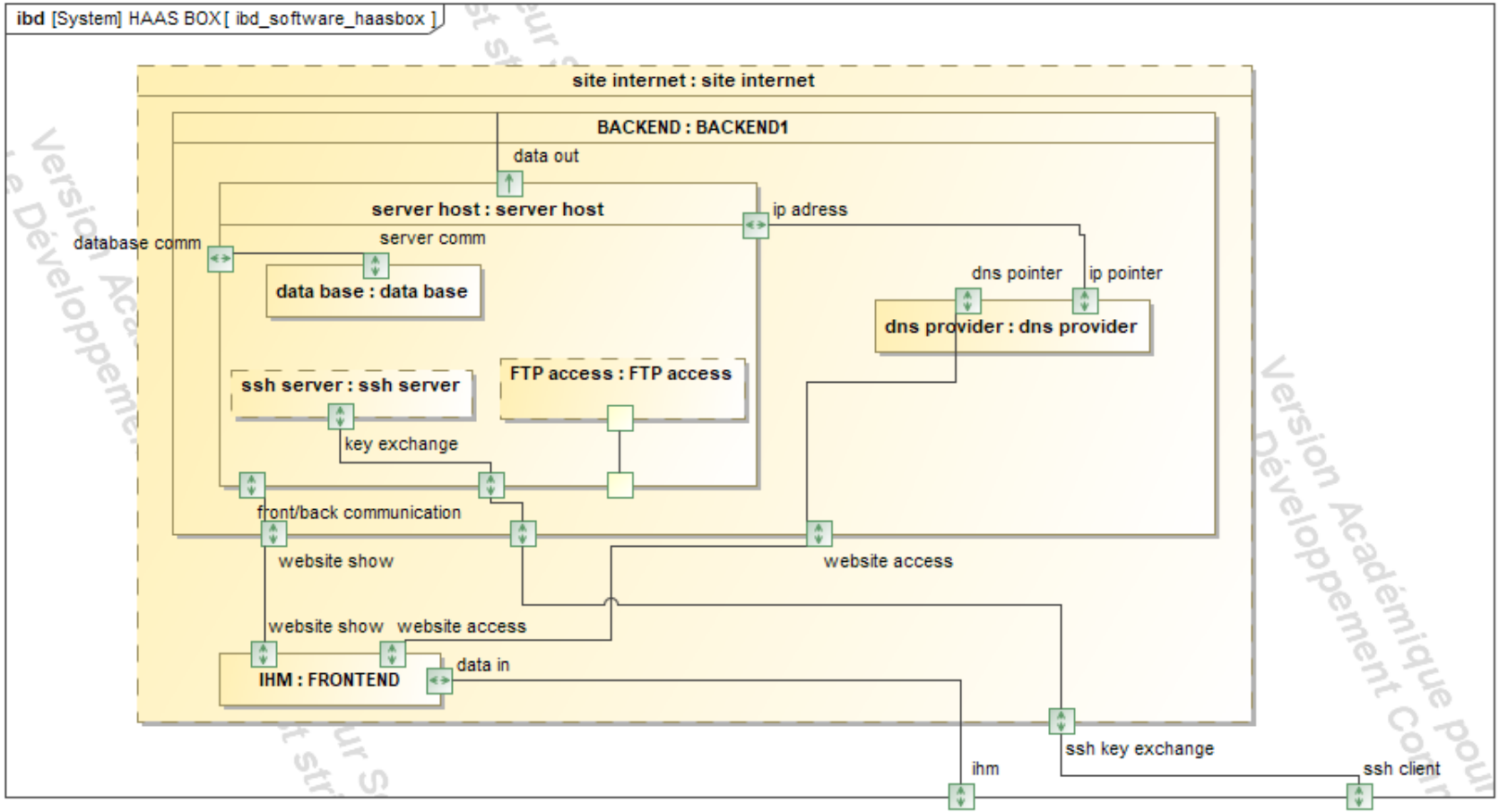


Diagramme de blocs internes : site internet



Organisation

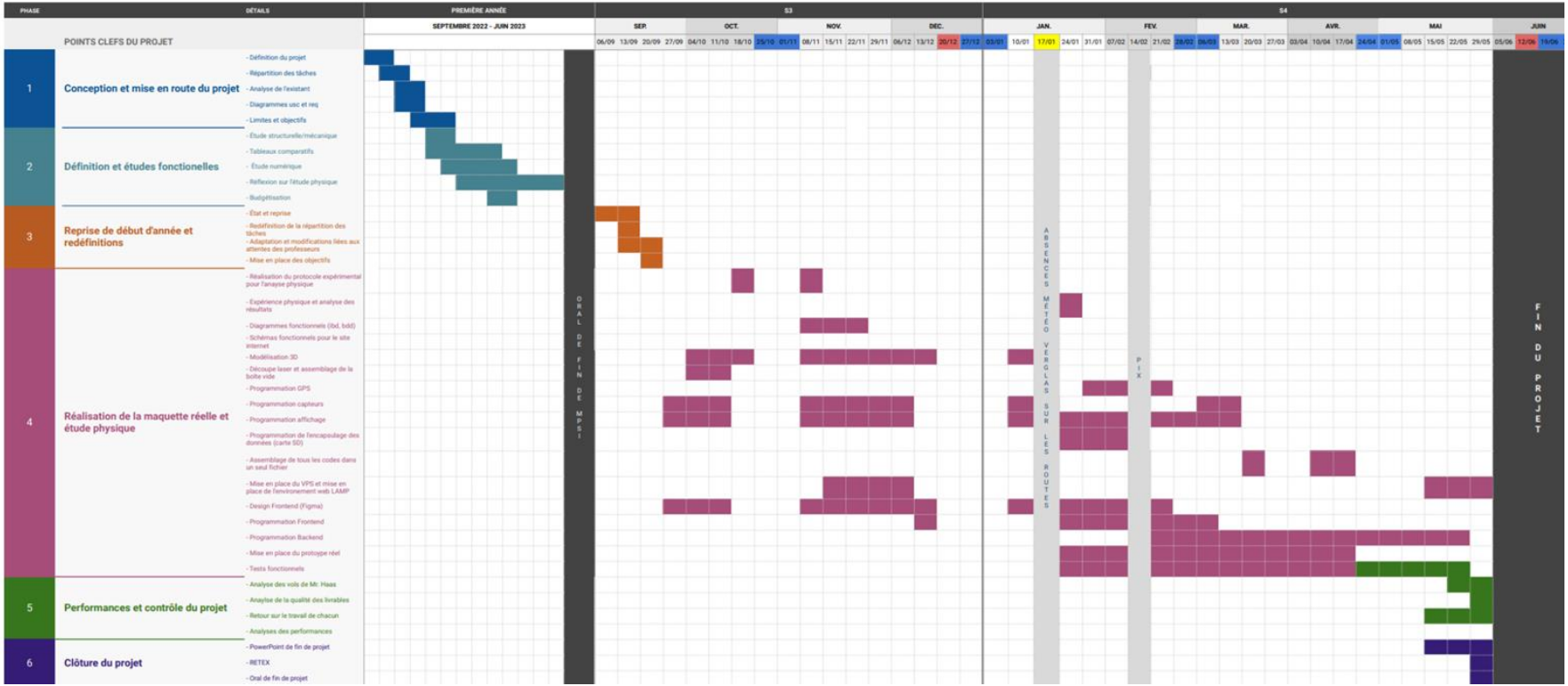
Répartition des tâches

GINOT Louis	Recherche du thème de projet Élaboration des fondements du scénario Répartition des tâches	Etude de la structure, modélisation et conception de la maquette 3D. Programmation du système GPS.	Étude de la structure pour l'intégration du matériel nécessaire.	Modélisation de la maquette sur Inventor.	Réalisation du code pour décomposer les trames NMEA	Réalisation de la maquette. Etude expérimentale du prototype
PONCHIN Alban		Etude, choix et programmation des capteurs d'humidité, pression et température.	Étude et choix des différents capteurs.	Préparation de l'étude physique et programmation des différents capteurs.	Programmation du code capteurs + système de stockage.	Réalisation de la maquette. Etude expérimentale du prototype
SANGARIEVA Safiat		Étude énergétique et choix du système d'alimentation. Choix et étude de la structure de la maquette 3D. Etude et design du site internet.	Etude et choix du système d'alimentation par batterie externe. Etude et choix du système de GPS.	Préparation de l'étude physique et simulation des différents capteurs.	Etude et design du site internet.	Réalisation de la maquette. Etude expérimentale du prototype
SARRA Alexis		Etude, choix et conception du stockage par carte micro SD. Etude, choix et conception de la retransmission des données sur un site internet.	Etude et choix du système de stockage par carte micro SD.	Choix et étude du microcontrôleur et des interfaces numériques.	Etude du système de communication. Design et programmation du site internet.	Réalisation de la maquette. Etude expérimentale du prototype

CALENDRIER DE PROJET

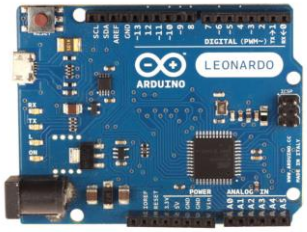
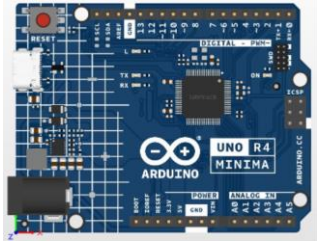
TITRE DU PROJET	HAAS' BOX	DÉTAILS	TIPE JEAN XXIII - MPSI/PSI	LÉGENDE	VACANCES SCOLAIRES
CHEF DE PROJET	/	ANNÉES	2022-2024		PARTIELS
GROUPE DE PROJET	Gr. 3	ÉTUDIANTS	GINOT Louis, PONCHIN Alban, SANGARIEVA Safiat, SARRA Alexis		ANNULÉ

metz campus



Tableaux Comparatifs

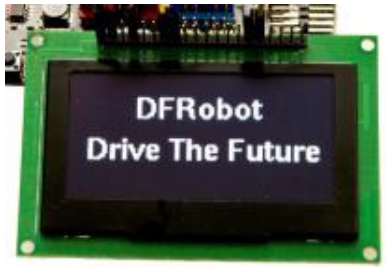
Comparatif Microcontrôleurs

	Arduino UNO ●	**Arduino Leonardo●**	**Arduino R4 minima□**
Type	Microcontrôleur □	Microcontrôleur □	Microcontrôleur □
Compatible Grove	OUI □	OUI□	OUI□
Microcontrôleur	ATMega328	ATMega32U4	R7FA4M1AB3CFM#AA0
Mémoire	2Kb ●	2,5Kb ●	8Kb□
Adapté au projet	NON, mémoire insuffisante●	NON, mémoire insuffisante●	OUI□
Prix (TTC)	23,90€□	25,90€□	22,50€□
Image			

Comparatif Capteurs

	Capteur BME	Capteur PT 101020812	Capteur TH 101020019
Alimentation	3,3 ou 5 Vcc	3,3 ou 5 Vcc	3,3 ou 5 Vcc
Plage de mesure	- température: -40 à 85 °C - humidité: 0 à 100 % - pression atmosphérique: 30 à 110 kPa	- Pression: 300 à 120 hPa - Température: - 40 à 85 °C	- température: -40°C à 80°C - humidité: 5 à 99%
Résolution	- température: 0,01 °C - humidité: 0,008 % - pression atmosphérique: 0,18 Pa	0,06 Pa	/
Précision	- température: ± 1°C - humidité: ± 3 % - pression atmosphérique: ± 0,6 hPa	± 1 hPa	- température: ±0,5°C - humidité: ±2%
Dimensions	40 x 20 x 7 mm	42 x 21 x 10 mm	40 x 20 x 11 mm
Prix (TTC)	24,90€	6,90€	11,40€
Image			

Comparatif Afficheurs

	Grove-LCD RGB Backlight V4.0 □	Afficheur OLED 2x16 □	afficheur OLED 2,7" FIT0328 ●
Écran	LCD alphanumérique 2 x 16 □	OLED alphanumérique 2 x 16 □	2,7" Oled, résolution 128 x 64 pixels □
Interface	I2C □	SPI ou parallèle □	SPI □
Compatible Grove	OUI □	NON □	NON □
Alimentation	5 Vcc □	3 - 5 Vcc □	3,3 - 5 Vcc □
Consommation	60mA max □	31mA max □	100mA max ●
Dimensions	84 x 45 x 13 mm □	80 x 36 x 10 mm □	85 x 58 x 7 mm □
Prix (TTC)	18,50 € □	21,50 € □	45,80 € ●
Image			

Comparatif GPS

	Module GPS grove 113020003 ●	Module GPS grove 109020022 □
Alimentation	3,3 ou 5 Vcc □	3,3 ou 5 Vcc □
Consommation	40 mA sous 3 Vcc	60 mA max
Sensibilité	-160 dBm □	-160 dBm □
Vitesse de transmission	4800 à 57600 bauds □	4800 à 57600 bauds □
Dimensions	40 x 20 x 13mm □	40 x 20 x 13 mm □
Compatibilité	NMEA et U-blox 6 □	GPS, Beidou, Glonass, Galileo, QZS, SBAS □
Timestamp	NON ●	OUI □
Prix (TTC)	26,90€ □	13,70€ □
Image		

Étude énergétique

ALIMENTATION

Capteur pression-température-humidité : 5 VCC
Capteur GPS : 5 VCC
Afficheur OLED : 5 VCC
Lecteur carte sd : 5.5 VCC

CONSOMMATION

Capteur pression-température-humidité : 0,09-12 mAh
Capteur GPS : 60 mAh
Afficheur OLED : 31 mAh
Lecteur carte sd: 200mAh
Carte arduino R4 minima : 60mAh

Total: 370mAh environ

Ainsi, suivant la consommation totale, nous pouvons choisir une batterie. Une batterie de 5000mAh convient à notre projet. Cela donne une autonomie minimale de 12h.

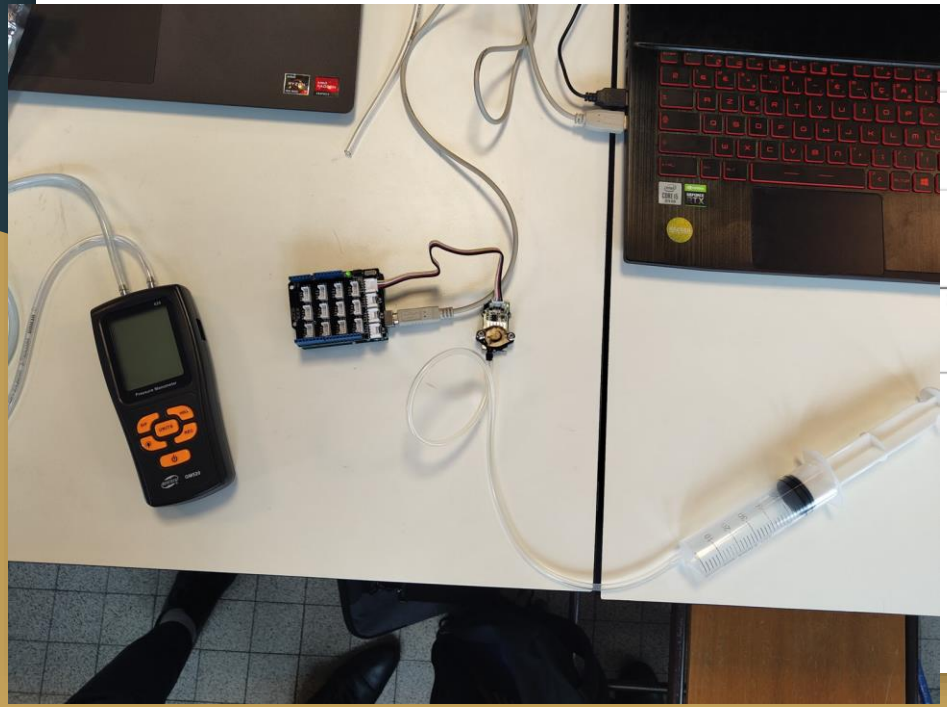
Comparatif Alimentation

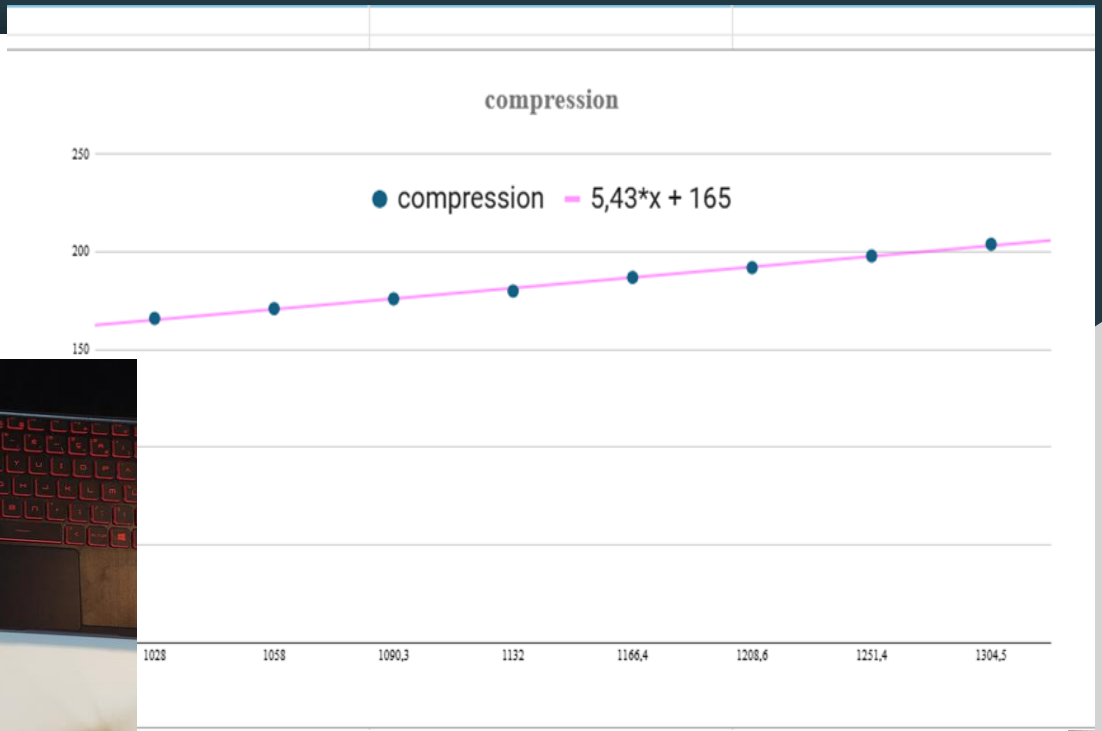
	Batterie externe USB 57975□	Pile alcaline 9V 6LR61●	Batterie externe USB V25●
Capacité	5000mAh□	700mAh●	8000mAh□
Type	Lithium-Ion	6LR61	Lithium-Ion
Dimensions	138 x 73 x 11 mm□	26,5 x 18 x 49 mm□	159 x 74 x 14 mm□
Entrée/Sortie	Micro-USB // USB Type-C | USB Type-A// USB Type-C	/	Micro-USB/ USB femelle type A
Poids	150 g□	43,3 g□	195 g□
Prix	19,90€□	2,85€□	22,90€□
Détails	- Cordon de charge micro-USB vers USB inclus Bouton-poussoir marche-arrêt	/	Cordon de charge 40 cm micro-USB vers USB inclus Bouton-poussoir marche-arrêt
Images			

Études et Analyses

	PLA	ABS	Découpe Laser (plexiglas)
facilité d'impression	☆ ☆ ☆ ☆	☆ ☆	☆ ☆ ☆ ☆ ☆ ☆
Résistance à la chaleur	☆	☆ ☆ ☆	☆ ☆ ☆ ☆ ☆
Résistance mécanique	☆	☆ ☆ ☆	☆ ☆ ☆ ☆ ☆
Résistance à l'humidité	☆	☆ ☆ ☆	☆ ☆ ☆ ☆ ☆
Remarques	• Température à l'intérieur du cockpit : 40°C ; • Le PLA n'est pas résistant à la chaleur (40-50°C max) ni à l'humidité. ➢ **Non adapté**	• ABS : un matériaux très intéressants pour notre structure; • Nombreux points positifs; • Impression complexe. ➢ **Non adapté**	• Résistance de -20 à + 80°C • Découpe de haute précision • Processus rapide ➢ **Adapté** ➢ **Type de plexi choisi : plexiglas incolore**

Étude physique





capteur_pression.ino

```ino
int rawValue; // A/D readings
int offset = 410; // uyzero pressure adjust
int fullScale = 9630; // max pressure (span) adjust
float pressure1;
float pressure2;
#define SERIAL Serial

void setup() {
  SERIAL.begin(9600);
}

void loop() {
  rawValue = 0;
  for (int x = 0; x < 10; x++) rawValue = rawValue + analogRead(A0);
  pressure1 = (rawValue - offset) * 700.0 / (fullScale - offset); // pressure conversion
  pressure2 = (5.43* analogRead(A0)) + 165 ;

  SERIAL.print("Pression1 : ");
  SERIAL.print(pressure1*10,1);
  SERIAL.print(" hPa");
  SERIAL.print("Pression2 : ");
  SERIAL.print(pressure2, 1);
  SERIAL.println(" hPa");
  delay(1000);
}

```

Etude physique

code capteur	code étude	écart	delta
877,7	1017,5	139,8	0,14
902	1033,8	131,8	0,13
914,9	1044,7	129,8	0,12
990	1099	109	0,1
1068,2	1153,3	85,1	0,07
1327,1	1332,5	5,4	0
1519,2	1473,6	45,6	0,03
1720,4	1614,8	105,6	0,07
2021,8	1832	189,8	0,1
2162,6	1935,2	227,4	0,12
1815,3	1685,4	129,9	0,08

Formule du programme : **p =((10 x - 410) * 700)/9220**

Notre formule : **p = 5,43 x + 165**

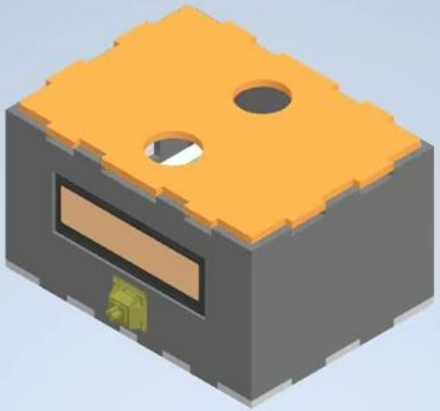
Delta d'erreur : 0,036

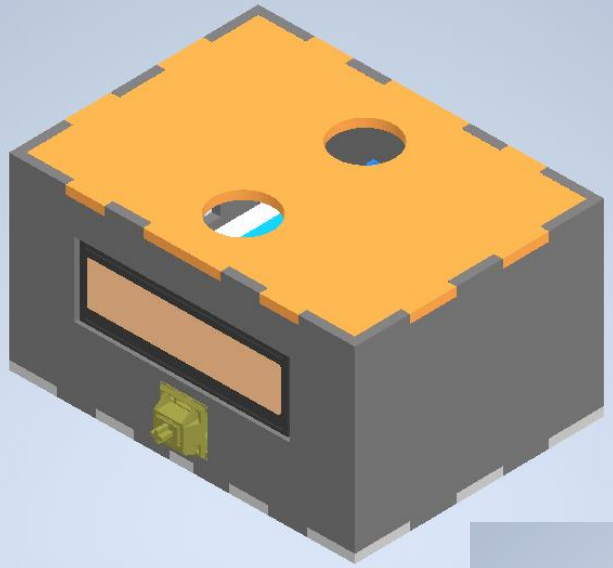
Coûts de conception

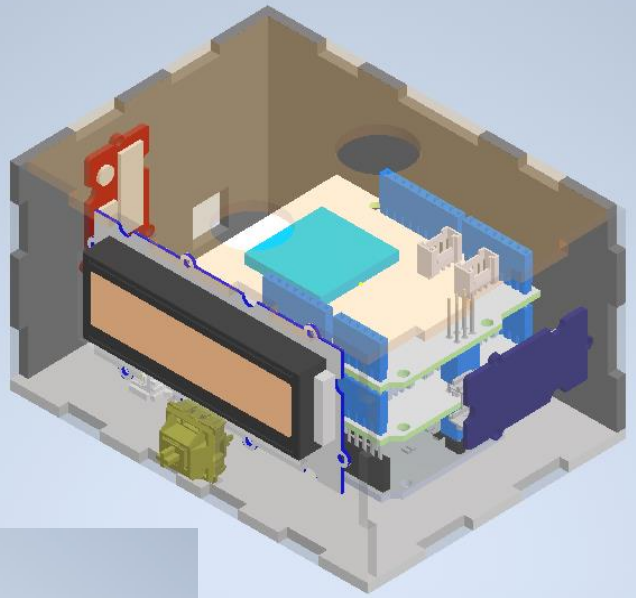
Nomination	Référence	Quantité	Coût TTC
Capteur PTH	BME680 / 101020513	1	24,90 €
GPS Grove	109020022	1	13,70 €
Shield carte SD V4	103030005	1	12,95 €
Arduino R4 minima		1	22,50 €
Module Grove Base Shield	103030000	1	4,80 €
Afficheur OLED 2 x 16	COM-OLED16x2	1	21,50€

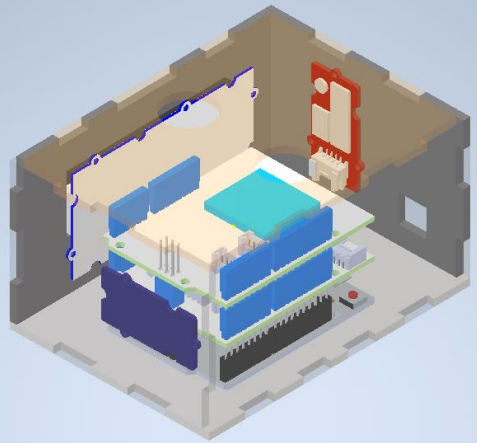
Total TTC
100,35 €

Modélisation 3D

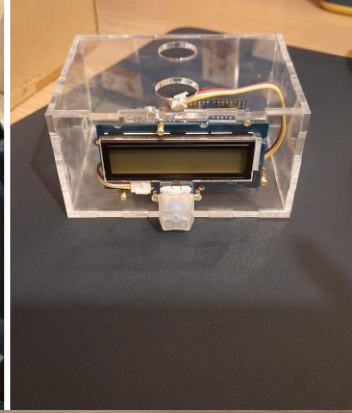


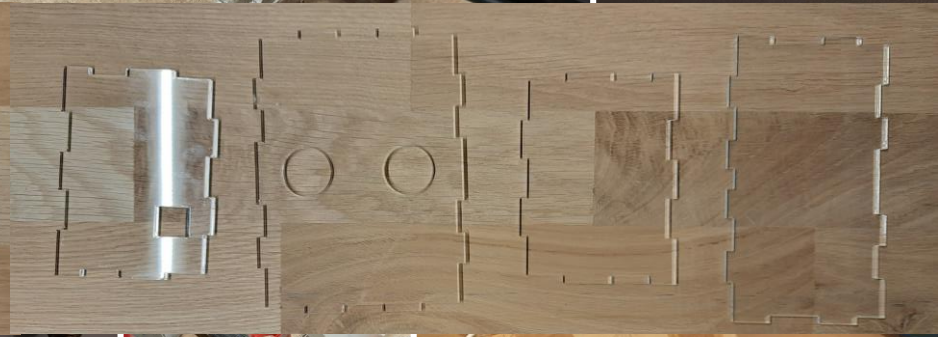




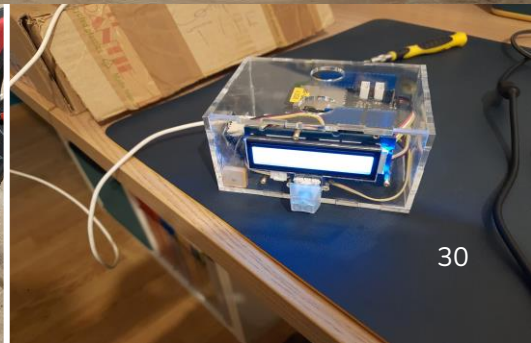












Programmation Arduino

```
#include <SoftwareSerial.h>
#include <TinyGPS++.h>
#include <SD.h>
#include <Wire.h>
#include "seeed_bme680.h"
#include "rgb_lcd.h"

// Broches pour le capteur GPS Grove
#define RX_PIN 2
#define TX_PIN 3
// Broche pour la carte SD
#define SD_CS_PIN 4
// Pin où le bouton est connecté
#define BUTTON_PIN 8
// Broches pour le capteur BME
#define BME_SCK 13
#define BME_MISO 12
#define BME_MOSI 11
#define BME_CS 10
#define IIC_ADDR  uint8_t(0x76)
```

Importation des librairies et définition des broches

Initialisation des composants

```
22  // Initialisation de l'objet TinyGPS++
23  TinyGPSPlus gps;
24
25  // Initialisation de l'objet SoftwareSerial
26  SoftwareSerial ss(RX_PIN, TX_PIN);
27
28  // Nom du fichier CSV
29  File dataFile;
30
31  rgb_lcd lcd;
32  const int colorR = 37;
33  const int colorG = 253;
34  const int colorB = 250;
35
36  int buttonState = 0;
37  unsigned long lastUpdate = 0;
38  const unsigned long updateInterval = 1000; // Intervalle de mise à jour des données en millisecondes (ici toutes les secondes)
39
40  enum DisplayMode {
41    PRESSURE,
42    TEMPERATURE,
43    HUMIDITY
44  };
45
46  DisplayMode currentMode = PRESSURE;
47  Seeed_BME680 bme680(IIC_ADDR);
```

Setup

```
49  void setup() {
50    Serial.begin(9600);
51    ss.begin(9600);
52
53    lcd.begin(16, 2);
54    lcd.setRGB(colorR, colorG, colorB);
55
56    if (!bme680.init()) {
57      Serial.println("Impossible de détecter le BME680. Vérifiez la connexion!");
58      while (1);
59    }
60
61    Serial.println("Initialisation terminée. Appuyez sur le bouton pour alterner entre la pression, la température et l'humidité.");
62
63    // Initialisation de la carte SD
64    if (SD.begin(SD_CS_PIN)) {
65      Serial.println("Carte SD initialisée avec succès.");
66      dataFile = SD.open("gps_data.csv", FILE_WRITE);
67
68      // Écrire l'en-tête du fichier CSV
69      dataFile.println("pressure,temperature,humidity,altitude,latitude,longitude,date,time");
70
71      dataFile.close();
72    } else {
73      Serial.println("Erreur lors de l'initialisation de la carte SD.");
74    }
75  }
76
77  unsigned long lastSaveTime = 0;
78  const unsigned long saveInterval = 2000; // Intervalle d'enregistrement des données en millisecondes (2 secondes)
```

```
void loop() {
  unsigned long currentMillis = millis();
  // Lire les données du capteur GPS
  while (ss.available() > 0) {
    gps.encode(ss.read());
    if (gps.location.isUpdated() && gps.date.isUpdated() && gps.time.isUpdated()) {
      // Enregistrer les données toutes les 2 secondes
      if (currentMillis - lastSaveTime >= saveInterval) {
        lastSaveTime = currentMillis;
        // Ouvrir le fichier CSV en mode écriture
        dataFile = SD.open("gps_data.csv", FILE_WRITE);

        // Enregistrer les données dans le fichier CSV
        dataFile.print(gps.altitude.meters());
        dataFile.print(",");
        dataFile.print(gps.location.lat(), 6);
        dataFile.print(",");
        dataFile.print(gps.location.lng(), 6);
        dataFile.print(",");
        dataFile.print(gps.date.year());
        dataFile.print("-");
        dataFile.print(gps.date.month());
        dataFile.print("-");
        dataFile.print(gps.date.day());
        dataFile.print(",");
        dataFile.print(gps.time.hour());
        dataFile.print(":");
        dataFile.print(gps.time.minute());
        dataFile.print(":");
        dataFile.println(gps.time.second());

        dataFile.print(bme680.sensor_result_value.pressure);
        dataFile.print(",");
        dataFile.print(bme680.sensor_result_value.temperature);
        dataFile.print(",");
        dataFile.print(bme680.sensor_result_value.humidity);
        dataFile.print(",");
        // Fermer le fichier
        dataFile.close();
      }
    }
  }
```

Void loop() : code de la carte SD avec le GPS

```
122   // programme affichage
123   int buttonState = digitalRead(BUTTON_PIN);
124
125   if (buttonState == HIGH) {
126     lcd.clear();
127
128     switch (currentMode) {
129       case PRESSURE:
130         lcd.setCursor(0, 0);
131         lcd.print("P: ");
132         lcd.print(bme680.sensor_result_value.pressure / 100.0);
133         lcd.print(" hPa");
134         lcd.setCursor(0, 1);
135         lcd.print("A: ");
136         lcd.print(gps.altitude.meters());
137         lcd.print(" m");
138         currentMode = TEMPERATURE;
139         break;
140
141       case TEMPERATURE:
142         lcd.setCursor(0, 0);
143         lcd.print("Temp: ");
144         lcd.print(bme680.sensor_result_value.temperature);
145         lcd.print('\xDF');
146         lcd.print("C");
147         currentMode = HUMIDITY;
148         break;
149
150       case HUMIDITY:
151         lcd.setCursor(0, 0);
152         lcd.print("Humidite: ");
153         lcd.print(bme680.sensor_result_value.humidity);
154         lcd.print("%");
155         currentMode = PRESSURE;
156         break;
157     }
158     delay(250); // Attendre 250 millisecondes pour éviter les pressions multiples du bouton
159   }
```

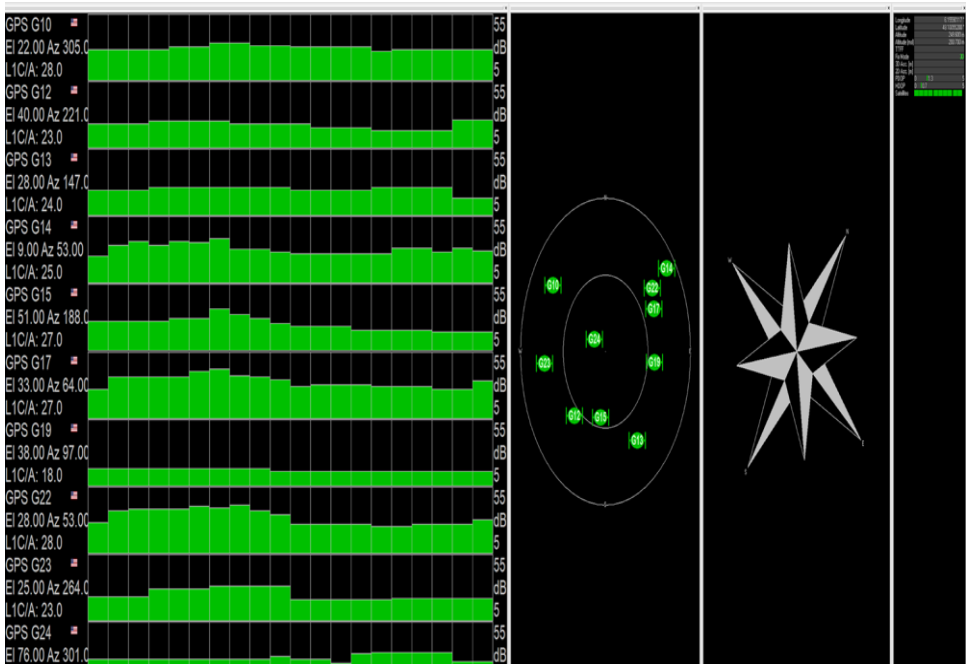
Void loop() : code de l'afficheur avec les capteur BME

```
  unsigned long currentMillis2 = millis();
  if (currentMillis2 - lastUpdate >= updateInterval) {
    lastUpdate = currentMillis;

    if (bme680.read_sensor_data()) {
      Serial.println("Failed to perform reading :(");
      return;
    }

    // Mise à jour de l'écran LCD avec les données actuelles en fonction du mode
    lcd.clear();
    switch (currentMode) {
      case PRESSURE:
        lcd.setCursor(0, 0);
        lcd.print("P: ");
        lcd.print(bme680.sensor_result_value.pressure / 100.0);
        lcd.print(" hPa");
        lcd.setCursor(0, 1);
        lcd.print("A: ");
        lcd.print(gps.altitude.meters());
        lcd.print(" m");
        break;

      case TEMPERATURE:
        lcd.setCursor(0, 0);
        lcd.print("Temp: ");
        lcd.print(bme680.sensor_result_value.temperature);
        lcd.print('\xDF');
        lcd.print("C");
        break;

      case HUMIDITY:
        lcd.setCursor(0, 0);
        lcd.print("Humidite: ");
        lcd.print(bme680.sensor_result_value.humidity);
        lcd.print("%");
        break;
    }
  }
}
```

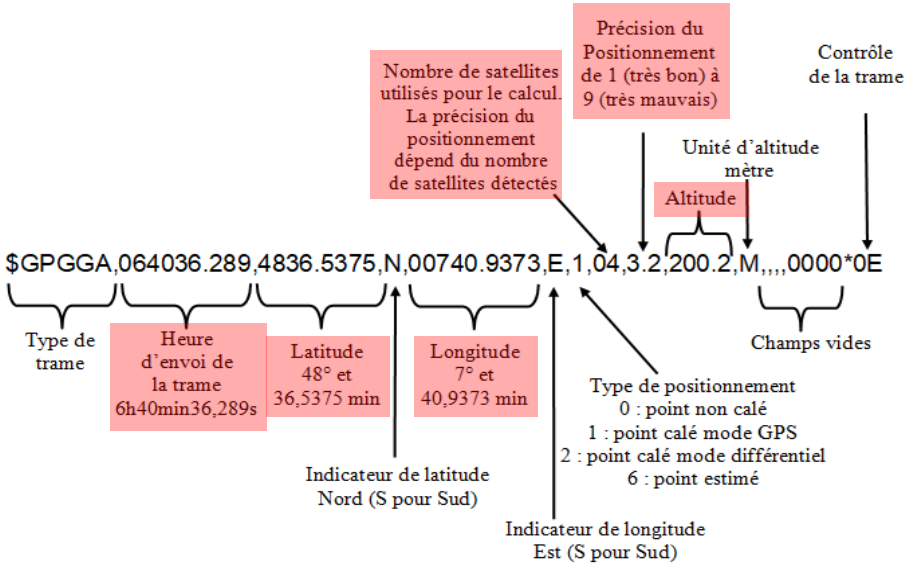
Void loop() : actualisation des données sur l'afficheur

L’utilisation du GPS

Récupérer la trame GGA — **Lire et traiter** — Stocker



Exemple de trame GGA :



Site Internet

Démarche

Définir les besoins et inventer le design → Reporter le design visuel en code informatique → Coder les algorithmes de traitement et gérer l'hébergement

DÉFINIR LES BESOINS

- identifier les objectifs du site;
- établir les exigences fonctionnelles.

INVENTER LE DESIGN

- Développer un prototype interactif.

FRONTEND

- Baliser les éléments principaux;
- Reporter les visuels en code informatique;
- Revoir les besoins et réadapter les visuels.

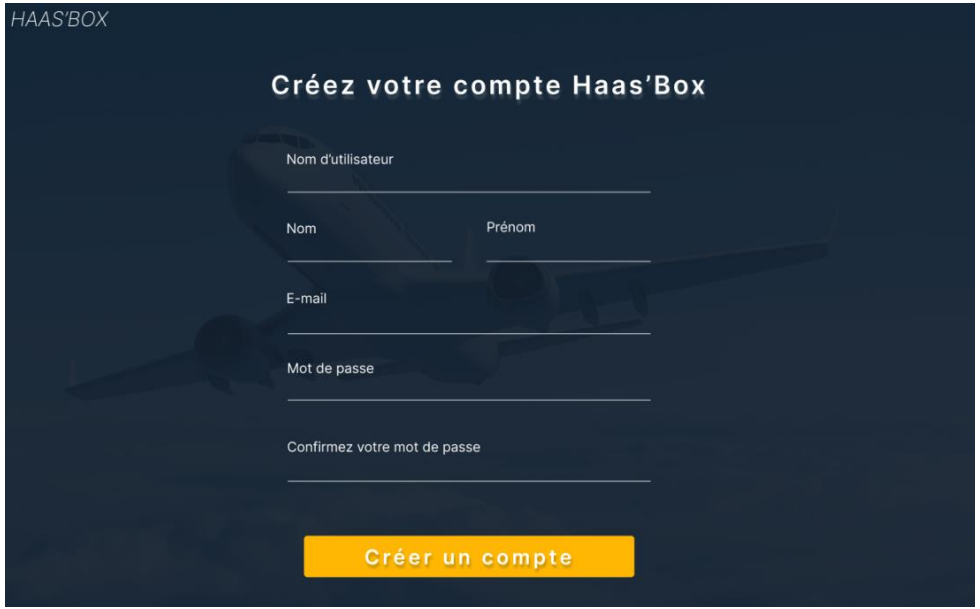
BACKEND

- Mettre en place les bases de données;
- Créer les interactions utilisateur;
- Concevoir la gestion des données;
- Mettre le site en ligne.

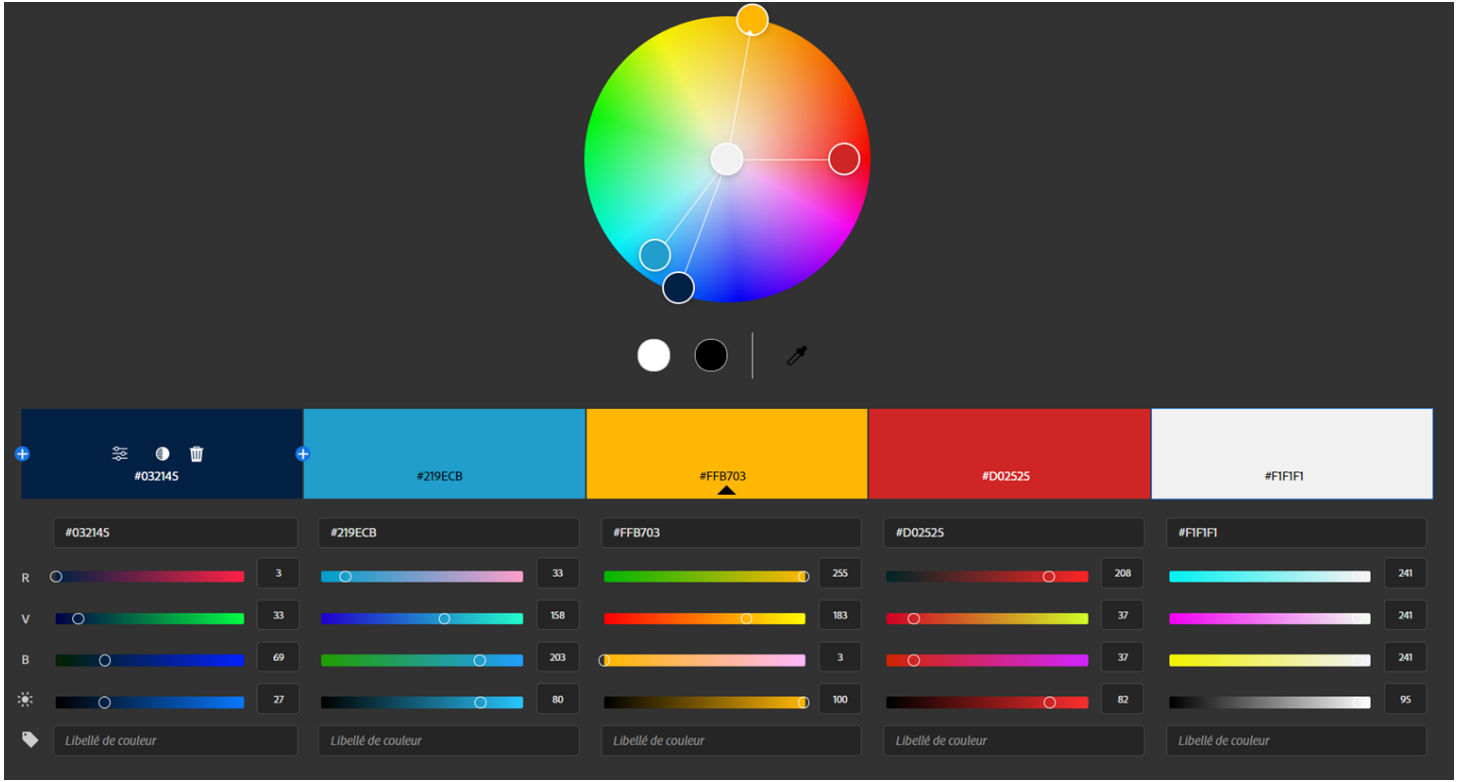
Définir les besoins

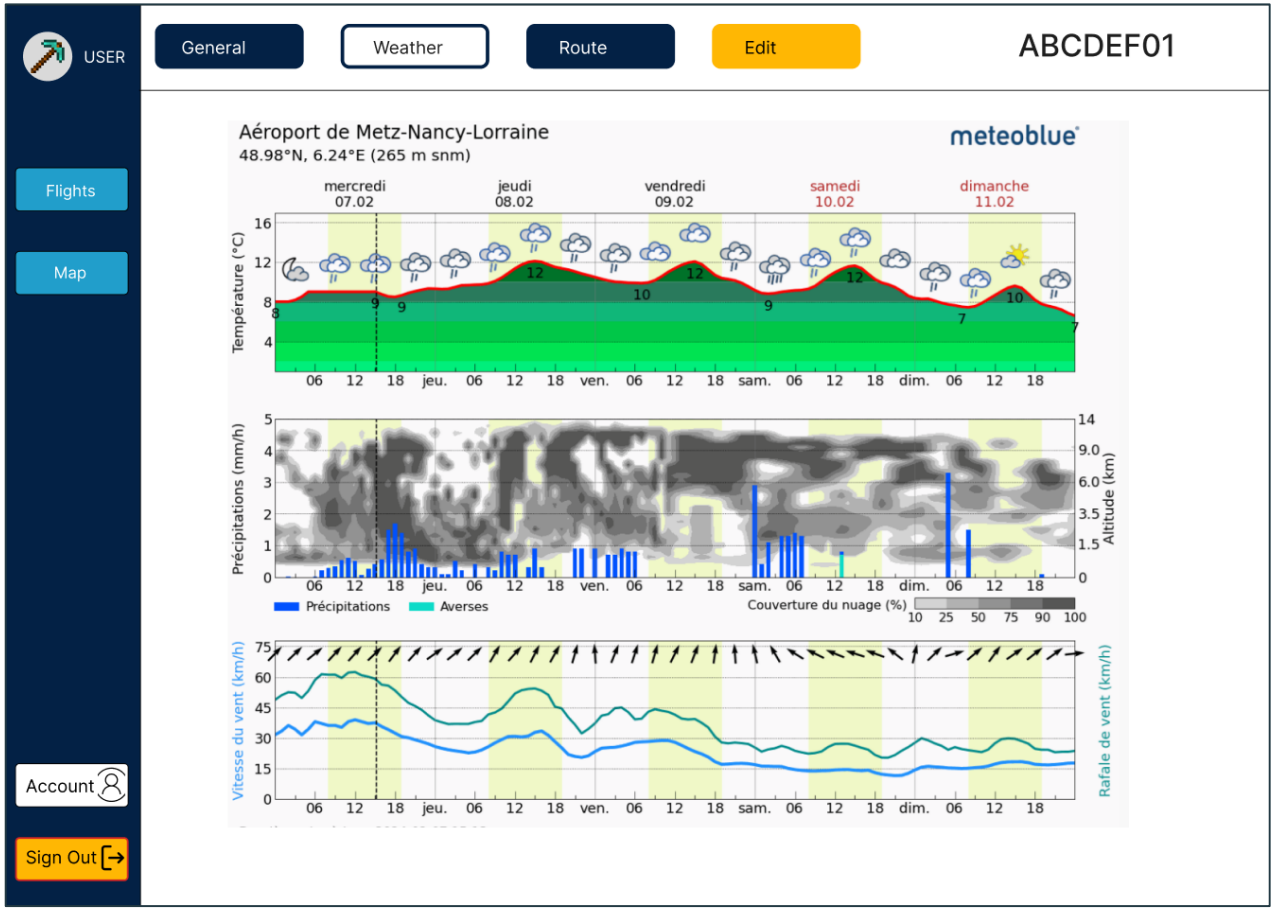
	barre de navigation latérale	barre de navigation supérieure	tableau de bord	liste des vols	page info vols
Fonctionnalités	- Boutons de navigation; - Déconnexion.	- Barre de recherche; - Boutons d'interaction vols.	Récapitulatif général.	Lister tous les vols.	- Données générales; - météo; - tracé; - modifications.
Organisation	- Boutons placés les uns en-dessous des autres; - Séparations des boutons d'interaction compte.	- Boutons placés les uns après les autres; - Barre de recherche en fin d'ensemble.	- Placer chaque information dans une bulle; - Placer un graphique pour illustrer.	Tableau.	Lisibilité accrue.

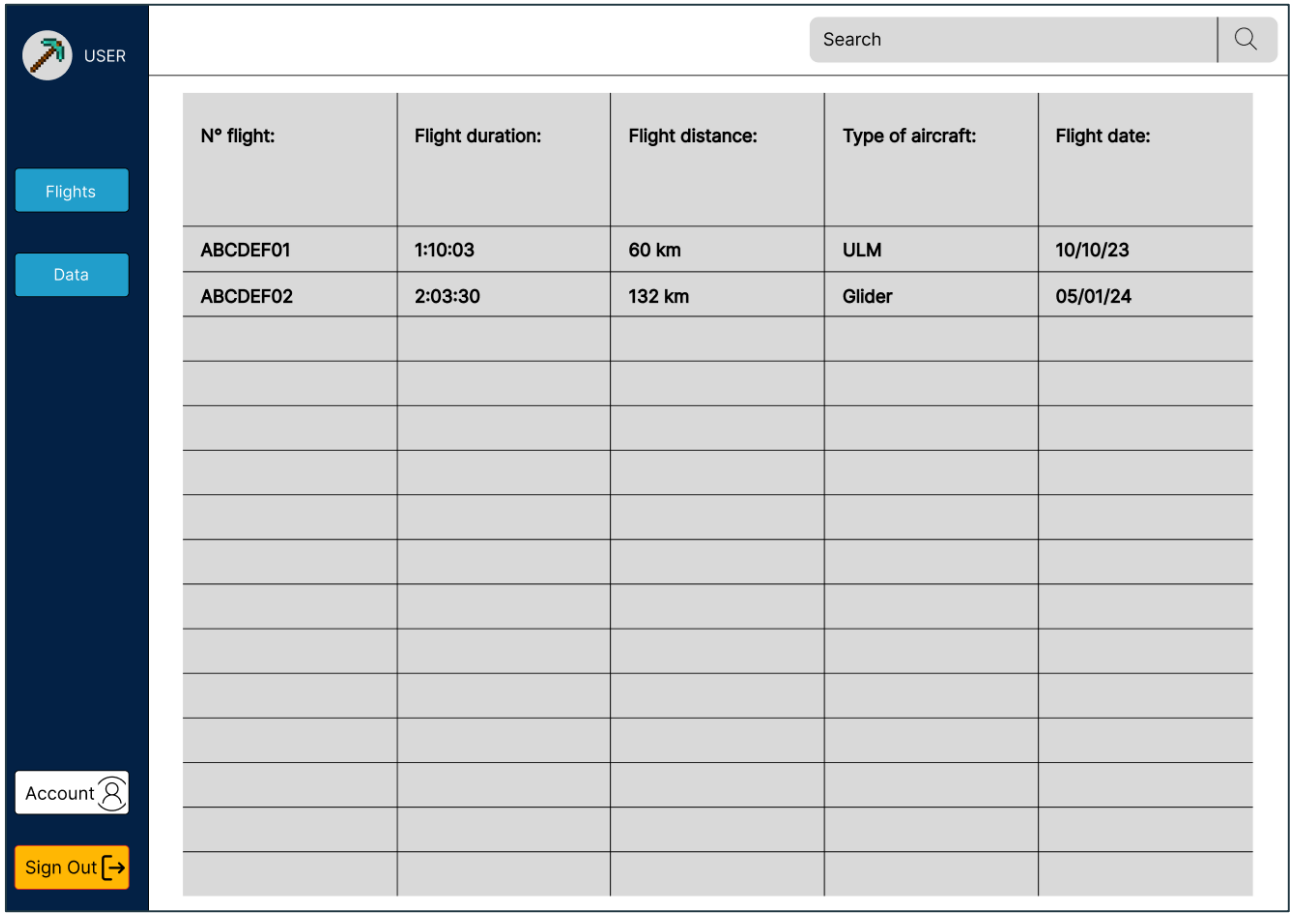
Inventer le Design

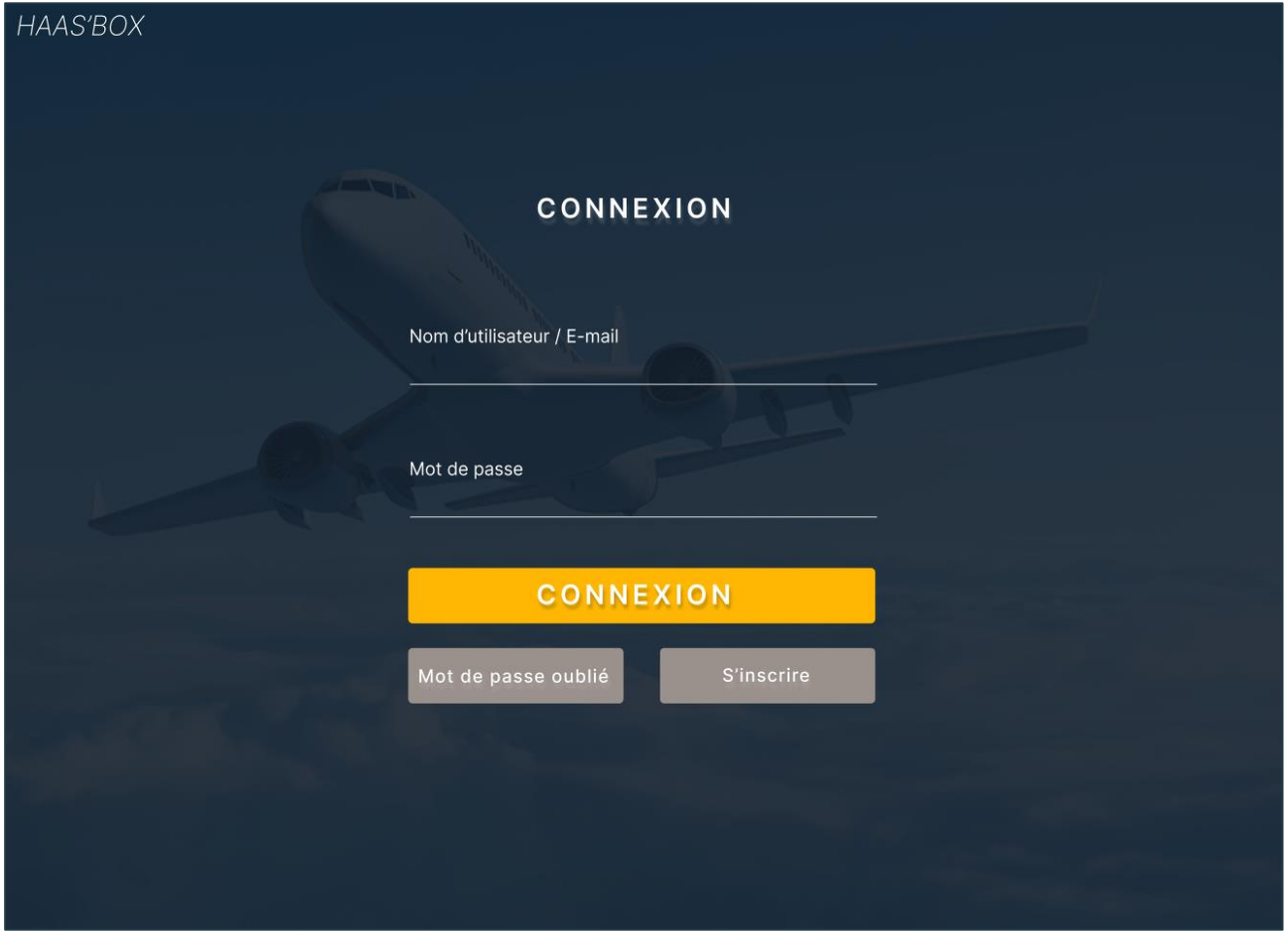


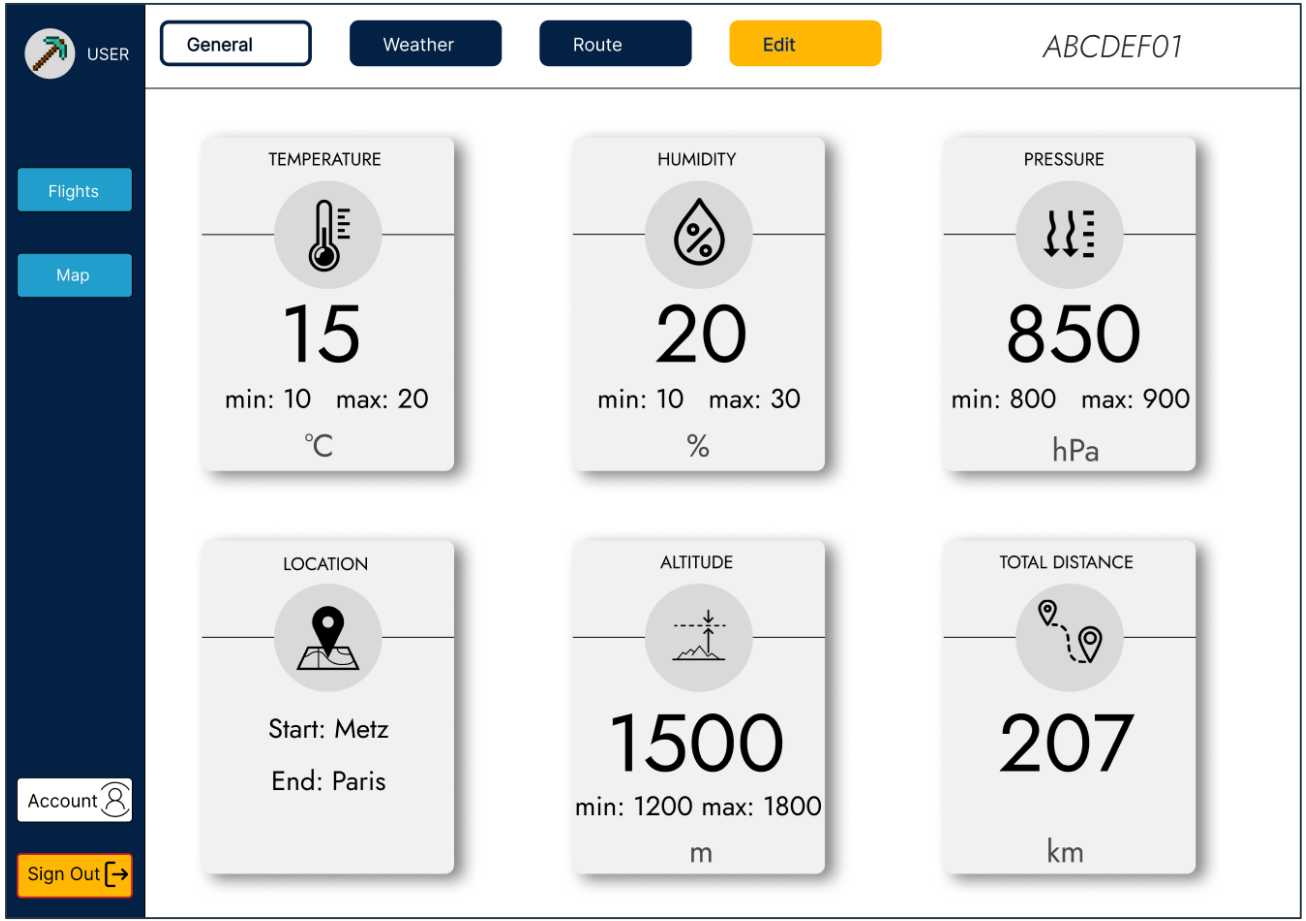
Figma


#032145
#219ECB
#FFB703
#D02525
#F1F1F1
#032145
#219ECB
#FFB703
#D02525
#F1F1F1
R
V
B
3
33
255
208
241
33
158
183
37
241
69
203
3
37
241
27
80
100
82
95
Libellé de couleur
Libellé de couleur
Libellé de couleur
Libellé de couleur
Libellé de couleur

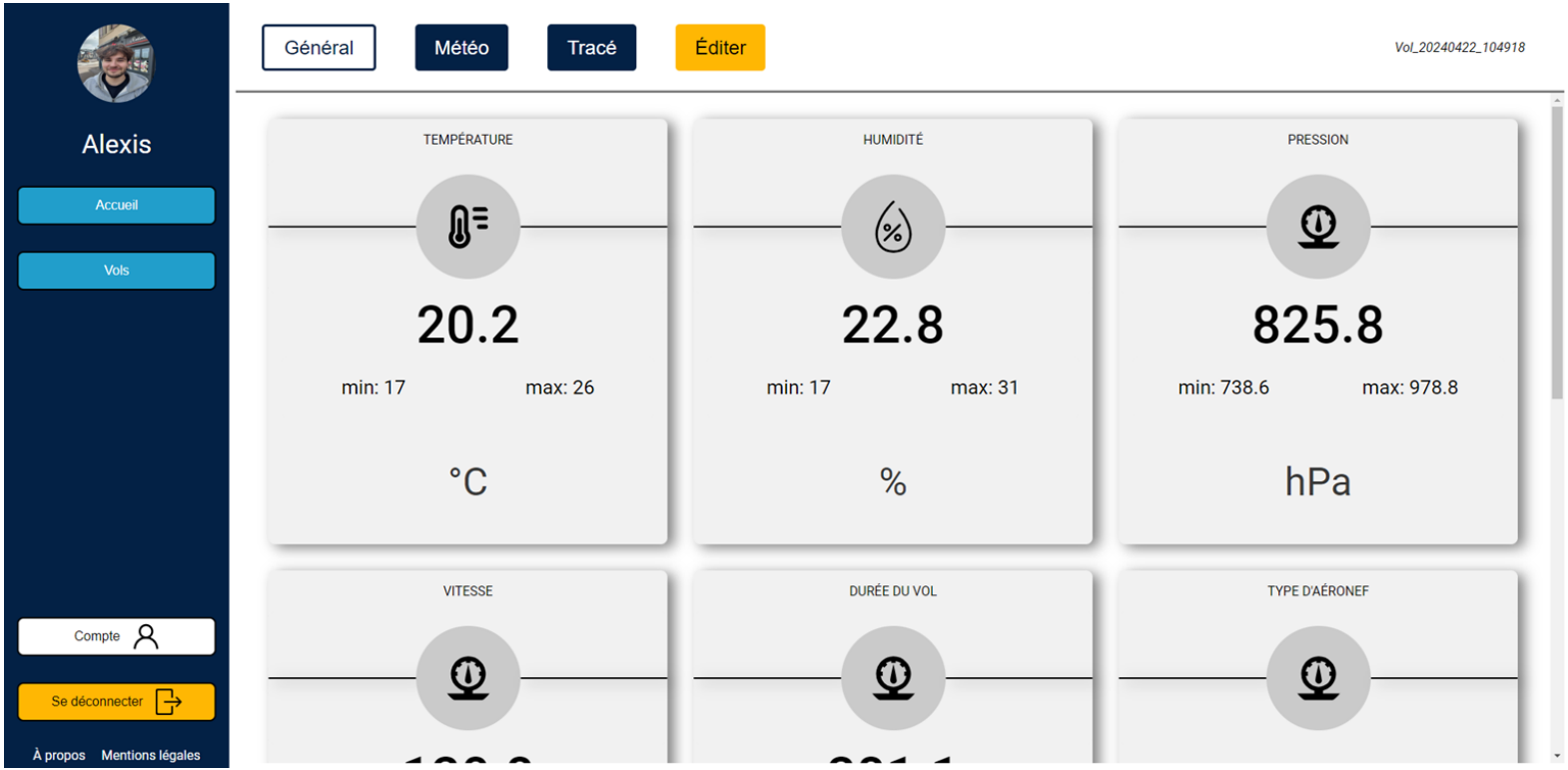

USER
General
Weather
Route
Edit
ABCDEF01
Flights
Map
Account
Sign Out
Aéroport de Metz-Nancy-Lorraine
48.98°N, 6.24°E (265 m snm)
meteoblue
mercredi 07.02
jeudi 08.02
vendredi 09.02
samedi 10.02
dimanche 11.02
Température (°C)
Précipitations (mm/h)
Altitude (km)
Précipitations
Averses
Couverture du nuage (%)
Vitesse du vent (km/h)
Rafale de vent (km/h)
jeu.
ven.
sam.
dim.


USER
Search
Flights
Data
Account
Sign Out
N° flight:
Flight duration:
Flight distance:
Type of aircraft:
Flight date:
ABCDEF01
1:10:03
60 km
ULM
10/10/23
ABCDEF02
2:03:30
132 km
Glider
05/01/24


HAAS'BOX
CONNEXION
Nom d'utilisateur / E-mail
Mot de passe
CONNEXION
Mot de passe oublié
S'inscrire


USER
General
Weather
Route
Edit
ABCDEF01
Flights
Map
TEMPERATURE
15
min: 10 max: 20
°C
HUMIDITY
20
min: 10 max: 30
%
PRESSURE
850
min: 800 max: 900
hPa
LOCATION
Start: Metz
End: Paris
ALTITUDE
1500
min: 1200 max: 1800
m
TOTAL DISTANCE
207
km
Account
Sign Out

Reporter les visuels en code informatique



Reporter les visuels en code informatique

HTML

Baliser les éléments

CSS

Créer le “style” des éléments

JS

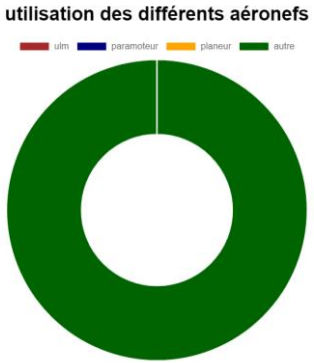
npm

AJAX

Leaflet

Bibliothèques

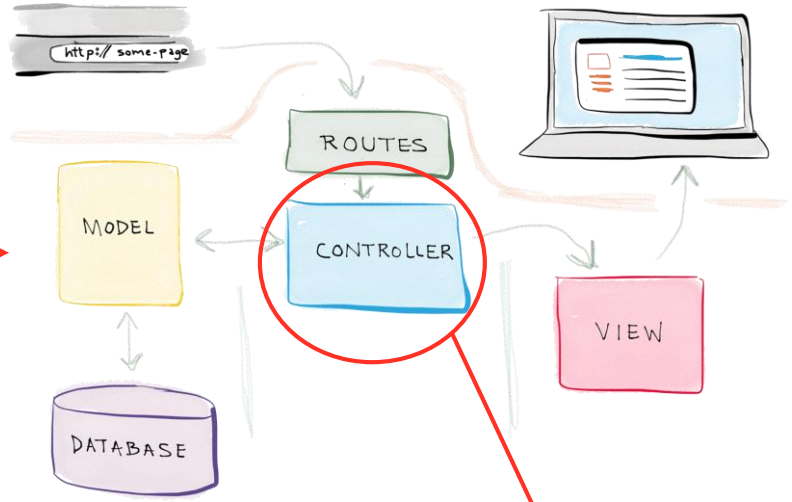
Chart.js : utilisé pour les graphiques



Coder les algorithmes de traitement

Rendre les pages dynamiques

Framework PHP



OpenStreetMap

Afficher la carte du tracé et faire du géocodage

Open-Meteo

Fournit des données météorologiques mondiales via une API

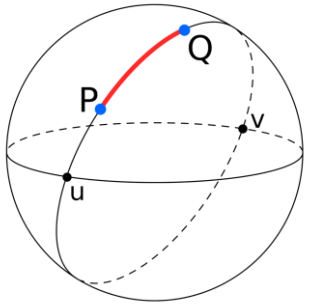
Récupérer les données météo jusqu'à 2 mois

EFFECTUE LES CALCULS

Coder les algorithmes de traitement

Formule de Haversine

Exemple : calculer la distance entre deux points




```
// Convert the latitude and longitude values to radians
$lat1 = deg2rad($previousRecord['latitude']);
$lon1 = deg2rad($previousRecord['longitude']);
$lat2 = deg2rad($record['latitude']);
$lon2 = deg2rad($record['longitude']);

// Calculate the difference between the coordinates
$dLat = $lat2 - $lat1;
$dLon = $lon2 - $lon1;

// Calculate the distance using the Haversine formula
$a = sin($dLat / 2) * sin($dLat / 2) +
       cos($lat1) * cos($lat2) *
       sin($dLon / 2) * sin($dLon / 2);
$c = 2 * atan2(sqrt($a), sqrt(1 - $a));

$distanceMadeAtPoint =  6371 * $c; // distance just made by the user
// Convert the distance to kilometers and add it to the total flight distance
$flightDistance += $distanceMadeAtPoint; // 6371 is the radius of the Earth in kilometers
```

$$\operatorname{hav}(\theta) = \operatorname{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\operatorname{hav}(\lambda_2 - \lambda_1)$$

$$\operatorname{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2}$$

Base de données

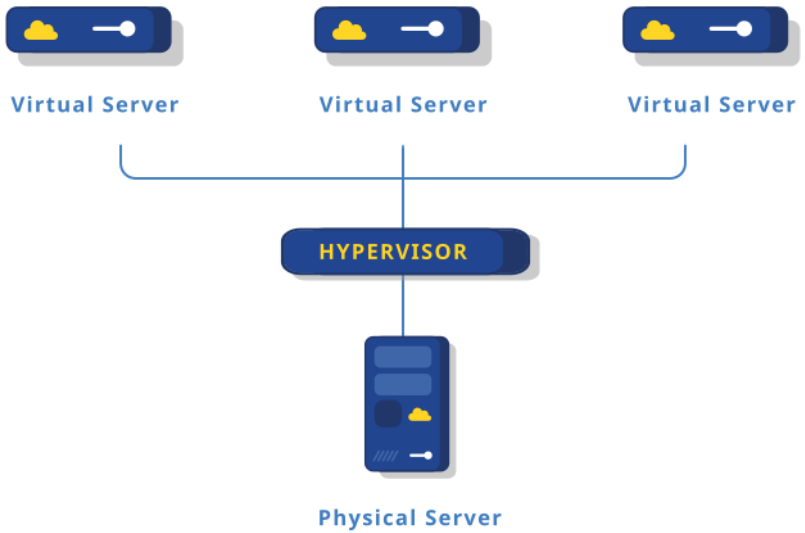
Système de Gestion de Bases de Données Relationnelles (SGBDR) → création de la base de données et des tables

Application de gestion pour les SGBD → interagir visuellement avec le tables

Table
flight_data
flight_lists
migrations
personal_access_tokens
users

id	flight_id	pressure	temperature	humidity	altitude	speed	latitude	longitude	weather_rain	weather_showers	surfacePressure	windSpeed	windDirection	date	time	created_at	updated_at
1	1	96685	46	12	305.6	0	49.10283300	6.73788300	0.00	0.00	0.00	0.00	0.00	2024-05-01	16:58:11	2024-05-25 11:08:20	2024-05-25 11:08:20
2	1	96695	46	12	304.9	2.2172032234812	49.10282200	6.73788500	0.00	0.00	0.00	0.00	0.00	2024-05-01	16:58:13	2024-05-25 11:08:20	2024-05-25 11:08:20
3	1	96685	46	12	303.8	5.4110921050373	49.10281100	6.73787300	0.00	0.00	0.00	0.00	0.00	2024-05-01	16:58:14	2024-05-25 11:08:20	2024-05-25 11:08:20

Hébergement



OVHcloud

OS / Distribution : Ubuntu 22.04
Localisation : Strasbourg (SBG)
vCores : 2
Mémoire : 4 Go
Stockage : 81 Go
DNS : OVH Web Cloud

Accéder au site internet

Adresse : https://asarra.fr/

Le projet en quelques images


F-PSIR
F-CLEN


F-PSIR

Planeur type ASH25 de 26 mètres d’envergure