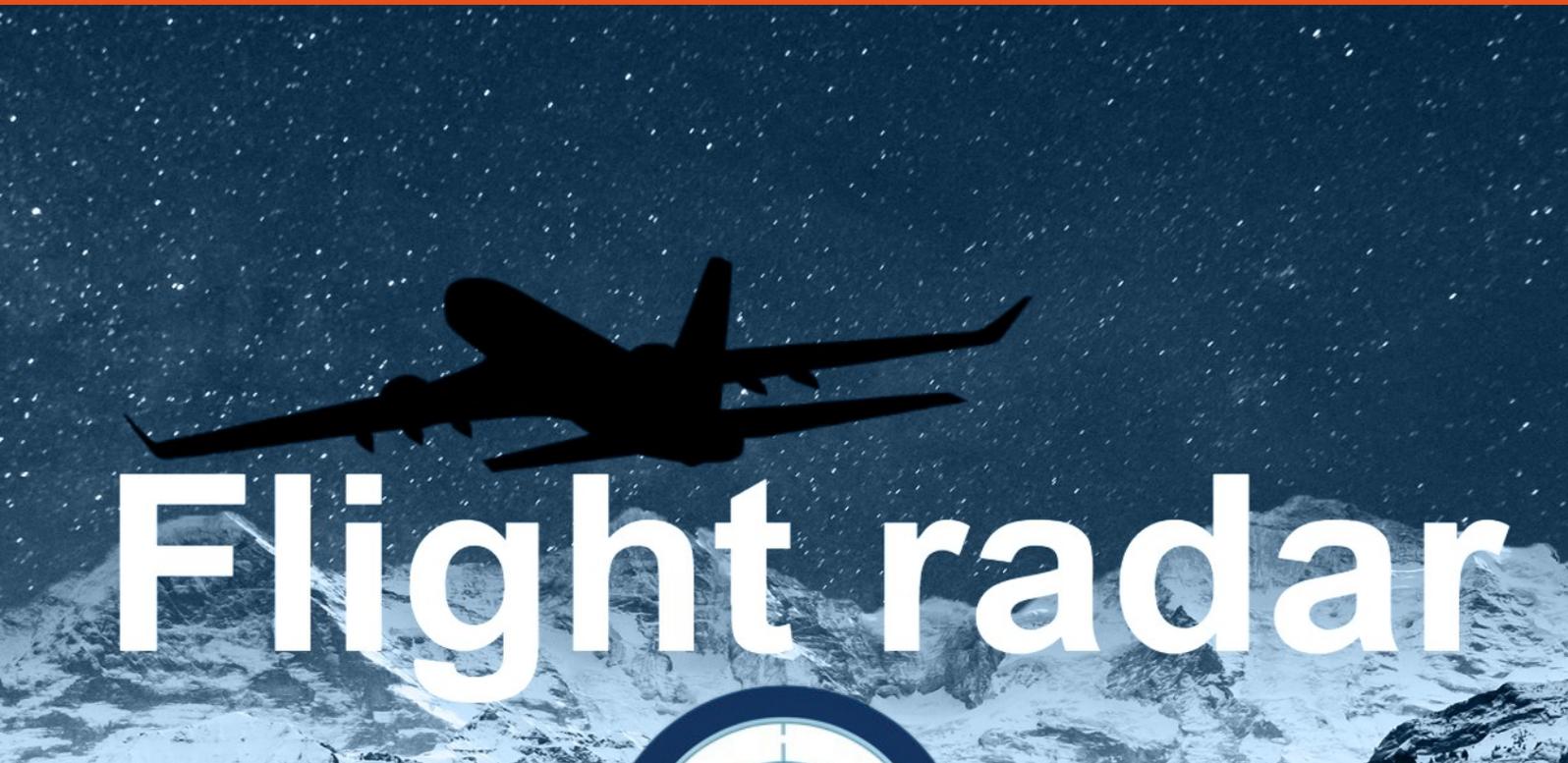




LES
TROPHÉES NSI

Édition 2023

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET



Flight radar



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

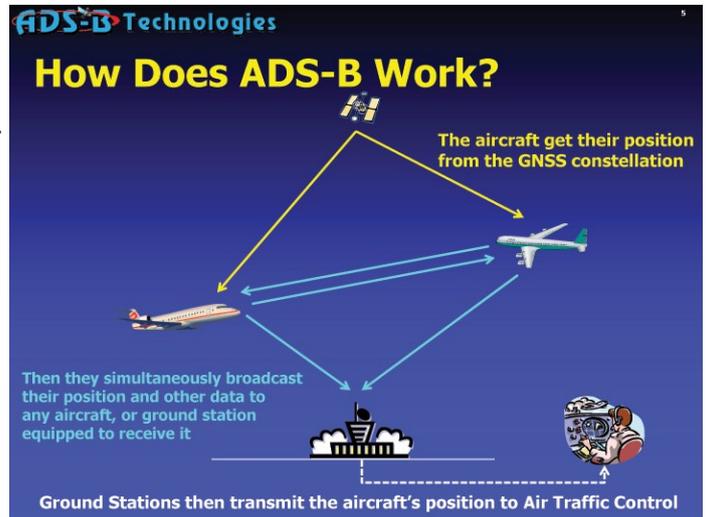
NOM DU PROJET : Flight Radar

> PRÉSENTATION GÉNÉRALE :

- Idée et objectifs
- Origines et intérêts du projet
- (...)

L'idée initiale du projet est de pouvoir placer sur une carte en temps réel les avions et leur trajectoires ainsi que leurs informations (identification du vol / ICAO / Longitude / Latitude). Afin de pouvoir récupérer ces informations nous utiliserons une clé RTL SDR qui permet de capter les signaux émis par les avions. Nous nous intéresserons à l'ADS-B et à l'ACARS.

L'ADS-B est un système de surveillance automatique - Broadcast. Il est utilisé pour afficher les mouvements de l'espace aérien. Alors que le radar de vol traditionnel suit les avions par la réflexion des ondes radar émises, l'ADS-B transmet activement les données de position des avions à l'environnement. Ils déterminent eux-mêmes leur propre position à l'aide de satellites de navigation et envoient ces données dans l'environnement sous une forme normalisée sur la fréquence de 1090 MHz. Les signaux peuvent être entendus jusqu'à 370 km - encore plus à des altitudes plus élevées.



L'ACARS permet de récupérer d'autre type d'information que l'ADS-B. L'ACARS (Aircraft Communication Addressing and Reporting System) est un système de communication codées (selon la norme ARINC) entre un aéronef et une station au sol. C'est un système de communication et de surveillance par radio utilisé en aviation par les exploitants d'aéronefs. Il permet le contrôle automatique de l'état de l'avion en vol, envoyé vers le centre de maintenance de la compagnie aérienne propriétaire de l'avion, mais aussi l'acheminement de communications opérationnelles et logistique. Ainsi le service de maintenance sait bien avant l'arrivée de l'avion son état, ses pannes éventuelles et donc les interventions à effectuer.

Avec la clé RTL SDR nous pourrions donc récupérer ces informations, les décoder en python, les mettre dans une base de donnée et enfin les afficher sur notre site et notre application.

> ORGANISATION DU TRAVAIL :

- *Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)*
- *Répartition des tâches*
- *Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)*

Nous avons organisé le projet en 3 grandes parties que nous nous sommes attribués en fonction de nos affinités. Chaque partie est différente est requiers des compétences bien différentes.

L'objectif de notre projet étant de recréer une tour de contrôle, nous avons décidé de créer un site et une application où l'on pourra placer en temps réel les avions et avoir accès à leurs informations (ICAO, destination, localisation, identification du vol).

La première se fut attribué à Eliott SEVIN. Cette partie consiste à configurer la clé RTL SDR, récupérer les signaux émis par les avions (trame), de les décoder et de les envoyer sur la base de données.

La deuxième partie est donc la base de données qui sera l'élément clé entre l'application et le site. Nicolas LE MILINAIRE se verra donc responsable de la base de données ainsi que le site internet où se trouvera la carte interactive.

Et enfin la dernière partie sera assurée par Robin BECKER. Il est chargé de créer l'application en Tkinter, ayant les même objectifs que la carte interactive sur le site.

Nous avons dus tous baser notre code sur la base de données qui est alimentée en continue par les informations. Tout notre code ce doit donc de pouvoir fonctionner en continue lorsque l'utilisateur le veut.

LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)*

Étape	Description	Difficultés	Durée
Idée	Le fonctionnement d'une tour de contrôle semble a la foi simple et complexe. Par curiosité de ce fonctionnement, nous avons donc décider de recréer une tour de contrôle.	Aucune	Quelques jours
Renseignement	Nous avons donc cherché ce que les avions émettent (ADS-B / ACARS) et comment les récupérer (cle RTL SDR). Puis nous avons cherché comment communiquer avec la clé. Python est donc la meilleure option ayant des bibliothèques pour pouvoir avoir accès à la clé.	Problème avec la bibliothèque pyrtlsdr, peu de documentation en ligne sur le sujet. Connaissance en physique requise.	Deux semaines

Première version	Après avoir compris comment cela fonctionne, nous avons créés l'algorithme qui permet de recevoir et décoder les trames. Nous avons aussi l'application et le site qui sont prêt à recevoir les informations de la base de données.	Peu d'avions dans la région / interférence due aux autres signaux.	Deux semaines
Finalisation	Correction de bug, pas encore eu le temps a cause des épreuves de spécialités pour connecter la base de données au reste du code.	Problème de temps.	Une semaine

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)*
- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*
- *Difficultés rencontrées et solutions apportées*

Ce qui est terminé	En cours de réalisation	Reste à faire
-Base de Données -Page d'accueil du site web -Code Python ADS-B -Carte interactive de l'application Tkinter	-Connexion à la base de données -Mise en temps réel de la carte interactive de l'application	-Code Python ACARS -Carte interactive du site web -Mise en ligne du site web

Approches pour vérifier l'absence de bugs :

- Mise en place d'un fausse base de données pour tester le positionnement des avions sur la carte interactive.
- Mise en place d'un fonction qui vérifie la validité de la trame récupérer à partir des signaux ADS-B

Difficultés rencontrées	Solutions apportées
-Fonction affichant les avions sur l'application ne fonctionnant que associé à un bouton Tkinter.	-Aucune
-La bibliothèque Pyrtlsdr ne fonctionnant pas.	-Modification du path directement dans la bibliothèque.
-Interférence dans signaux reçu et manque d'avions.	- Aucune (Pas de notre ressort)

> OUVERTURE :

- *Idées d'améliorations (nouvelles fonctionnalités)*
- *Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)*
- *Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)*

Idées d'améliorations :

- A l'heure actuelle le code ne permet pas de lire l'ACARS, le projet se montre beaucoup plus difficile que prévu. Nous pourrions donc avec plus de temps ajouter l'ACARS.
- Ajout de la carte interactive sur le site web
- Mise en temps réel de la carte interactive de l'application.
- Mettre en ligne le site web
- Connexion à la base de données pour lié l'intégralité du projet.

Stratégie de diffusion :

- Présentation du projet à de petits aérodromes locaux.

DOCUMENTATION

- *Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)*
- *Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)*
- *Illustrations, captures d'écran, etc*

Spécifications fonctionnelles

Pour utiliser l'application :

- il faut télécharger le dossier AppFlightRadar depuis le [Github](#)
- ainsi que les bibliothèques python : (tkinter, customtkinter, tkintermapview, Pillow, pymysql)
- et puisque la base de données n'est pas liée, installer la fausse base de données en localhost avec un logiciel comme xamp (voir Annexe)
- Il ne reste plus qu'à exécuter le code python et appuyer sur le bouton rafraîchir pour faire apparaître les avions

Pour utiliser le code python ADSB :

- Brancher la clé RTL SDR, avec des antennes.
- Executer le programme python (après avoir installer les bibliothèque, maths, Pyrtlsdr, time, copy et codecs, il est possible qu'il y ait des problème de path avec Pyrtlsdr, dans ce cas la, il faudra changer le path directement dans la bibliotèques).
- Le programme return l'ICAO, l'identification du vol, si la trame est valide et le type.

Spécifications techniques

Bibliothèques utilisés :

-maths -Pyrtlsdr -time -copy -codecs
-tkinter -customtkinter -tkintermapview -Pillow
-pymysql

Matériel :

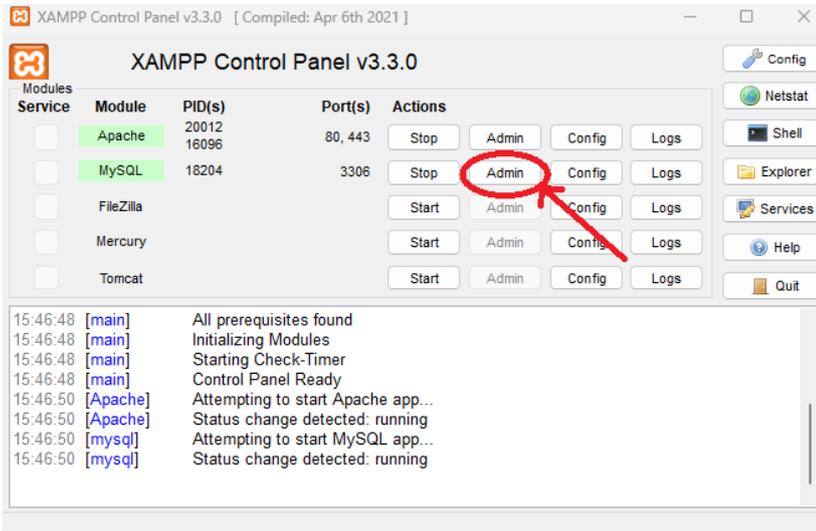
-Quatre antennes (2 paires, une pour l'ADS-B et une pour l'ACARS) -une Clef RTL SDR

Choix techniques :

Le projet est programmé en python.

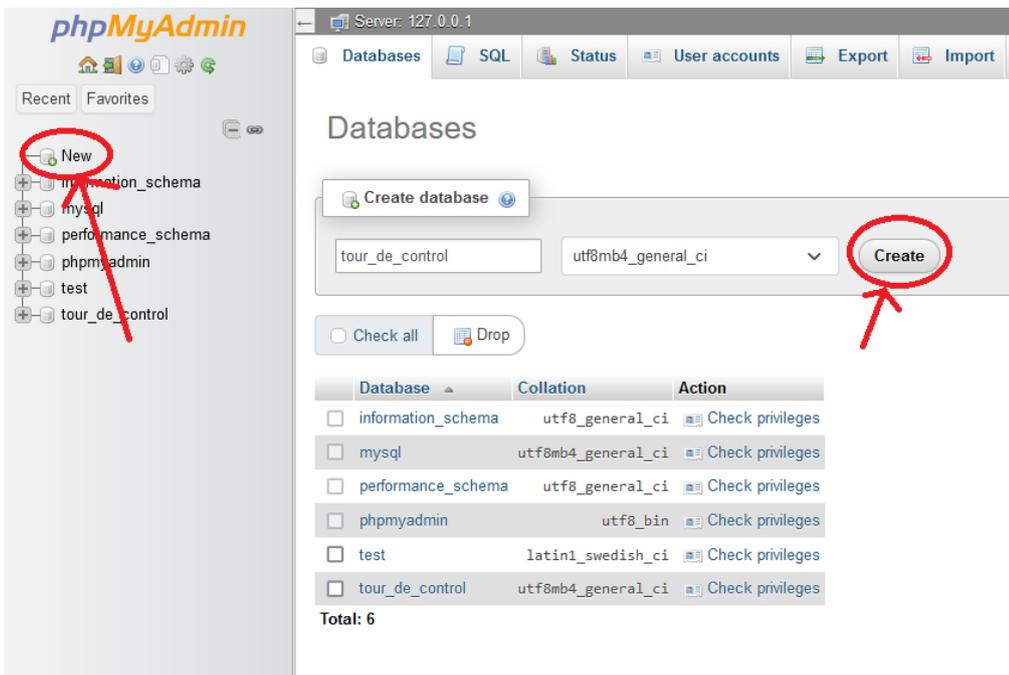
ANNEXE :

Ici on utilise Xamp, une fois installer et lancer, on active les modules Apache et MySQL :

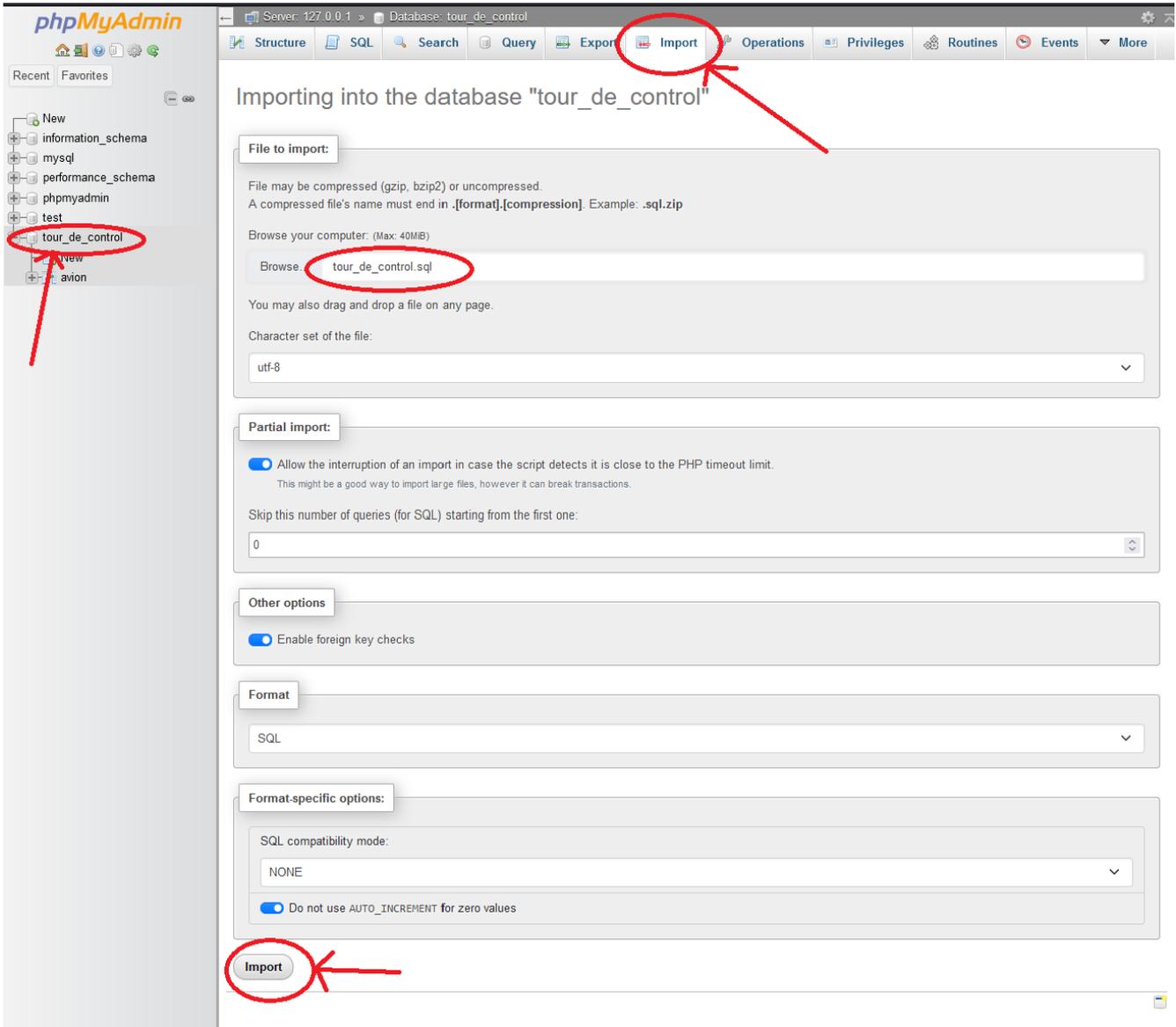


Ici les noms des modules sont en vert, ils sont donc activés. En suite, on appuis sur admin sur la ligne MySQL pour lancer une page phpMyAdmin

En suite, on créer une nouvelles base de données que l'on nomme : « tour_de_control ».



On se rend en suite sur la base de données et on importe le fichier : « tour_de_control.sql » qui viens avec le code python.



L'application est maintenant prête à être testée.