

RAPPPORT DE PROJET



Gulde Louka
Marty Théo

Agogue Mathis
Flavien

El Bargui Anass

Marwan Bouhadida

Barré



Sommaire

I.	Introduction :	3
II.	Besoins et objectifs du projet	4
1.	Contexte	4
2.	Besoins et Objectifs	4
III.	Gestion de Projet et outils de gestion :	5
1.	Gestion humaine	5
2.	Diagramme de Gantt	5
3.	Tableau RACI	6
4.	Serveur Discord	6
5.	Google Drive	6
IV.	Risques et Opportunité :	7
1.	Les risques :	7
A.	Les risques humains et internes au groupe	7
B.	Les risques concernant l'application en Python	7
C.	Les risques concernant le support web	7
D.	Les risques concernant la base de données	7
2.	Les Opportunités :	8
3.	Evaluation des familles de risques :	8
A.	Les risques humains et sociaux :	8
B.	Risque de l'application Python	9
C.	Risque support Web	11
D.	Risque Base de données	12
V.	Exigences du projet :	13
A.	Exigences Globales :	13
B.	Exigences Web :	13
C.	Exigences Base de Données :	14
D.	Exigences Application :	15
VI.	Plan de Validation	17
1.	Procédures de test	17
A.	Procédures de l'application Python :	18
B.	Procédure du support web :	18
	Page principale :	18



Page Secondaire :	18
VII. Supports et application :	19
1. Support web :	19
A. Page principale :	19
2. Base de données phpMyAdmin :	24
3. Application Python	24
A. Lecture fichier binaire :	25
B. Les extracteurs.....	26
B. Fonctions transfert :	27
C. Fonction principale du code	28
Conclusion	29
Annexes	30
Annexe 1 : Procédures Python fichiers Binaires et Rep	30
Annexe 2 : Connectivité Base de données et intégrité des données.....	31
Annexe 3 : Historiques et liens	32
Annexe 4 : Fonctionnalité barre de recherche et affichage résultats.....	32
Annexe 5 :Nombre de données par page / nombre de test historique.....	33
Annexe 6 : Changement de page et noms des colonnes	34
Annexe 7 : intégrité des données de la base de données.....	34
Annexe 8 : Schéma de la base de données	35

I. Introduction :

Dans le cadre de notre première année à ***l'Institut Universitaire de Technologie de Sophia-Antipolis*** et d'un partenariat entre l'Université et l'entreprise ***Thalès Alenia Space***, il nous est proposé un projet étalé sur les 2 semestres de 1^{ère} année du BUT Réseaux et Télécommunications.

Le premier semestre concerne la partie *théorique* du projet : algorigrammes, définition des exigences, des risques, conception de maquettes web, structuration de la base de données.

Le second semestre concernera la mise en application et le développement de la solution, le tout devant fonctionner sous une Raspberry Pi.



Notre groupe, composé de Louka Gulde, Anass El Bargui, Théo Marty, Mathis Agogue et Flavien Barré, tous passionné d'informatique, nous lançons dans ce projet sous la supervision de **M. Philippe Cam**, *Chef du service IVVQ Avionique*, sur le site de Cannes.

Ce projet vise à développer une solution logicielle pour accéder aux données renvoyées par les bancs de test satellitaire, les traiter et en obtenir une version lisible et compréhensible pour les équipes d'ingénierie.

II. Besoins et objectifs du projet

1. Contexte

Thalès Alenia Space se positionne en tête des entreprises concernant le développement et l'innovation satellitaire. Fruit de la fusion entre Thalès (Français) et Leonardo (Italien), la multinationale ne cesse d'innover pour aller toujours plus haut, et toujours plus loin.

Les satellites sont de mise dans la course à la conquête spatiale, et pour ce faire, la création de satellites toujours plus petits, léger, abordables financièrement ; mais toujours plus efficaces, autonomes et fiables est primordial.

Ce projet s'inscrit donc dans une logique constante d'amélioration de l'efficacité, de la rapidité et de la fiabilité des tests effectués sur les engins spatiaux.

2. Besoins et Objectifs

Les besoins du projet sont clairs et définis. Nous les avons regroupés en 2 grandes parties :

- Une application permettant de lire et traiter les données en entrée
- La gestion de fonctions transferts permettant l'insertion de données
- Un site web utilisé en tant que support des résultats de l'application.
- Toute ressource utilisée sera libre de droit et perdurera dans le temps
- Le tout doit impérativement être accessible sous distribution Linux
- La solution doit correctement fonctionner sous une Raspberry Pi

Ce projet comporte de nombreux objectifs que nous sommes parvenus à rassembler et réduire :

- L'application devra recevoir des trames Ethernet binaires en entrée, traiter ces trames dans un langage de programmation de notre choix (C, C++, Python, Javascript)

afin d'en obtenir des tableaux des informations critiques du test et finalement retourner ces résultats critiques pour une utilisation extérieure.

- Le support web devra afficher les résultats de l'application python et les stocker dans une base de données avec une grande disponibilité.

III. Gestion de Projet et outils de gestion :

Le projet nécessitant la mise en place d'une solution complexe – le projet nécessite la création d'une application, d'un support d'affichage et d'une base de données – il nous était impérativement nécessaire d'avoir une gestion de projet solide et efficace, pour nous permettre de nous comprendre, de gérer les tâches et d'atteindre nos objectifs en temps.

Afin de réussir le projet dans les temps nous avons donc décidé, ensemble, de mettre en place des solutions de co-working que nous avons accepté de suivre et de respecter, ce qui nous a permis d'avancer efficacement et sans mésentente.

Nous avons d'ailleurs, ensemble, désigné un Chef de Projet, qui nous permettrait de gérer les tâches, les échéances, les communications inter-tâches.

1. Gestion humaine

Etant en groupe de 5, il nous a paru, très rapidement, assez important de pouvoir rester en contact et se voir régulièrement pour partager nos idées, avancées, problèmes autour d'une table.

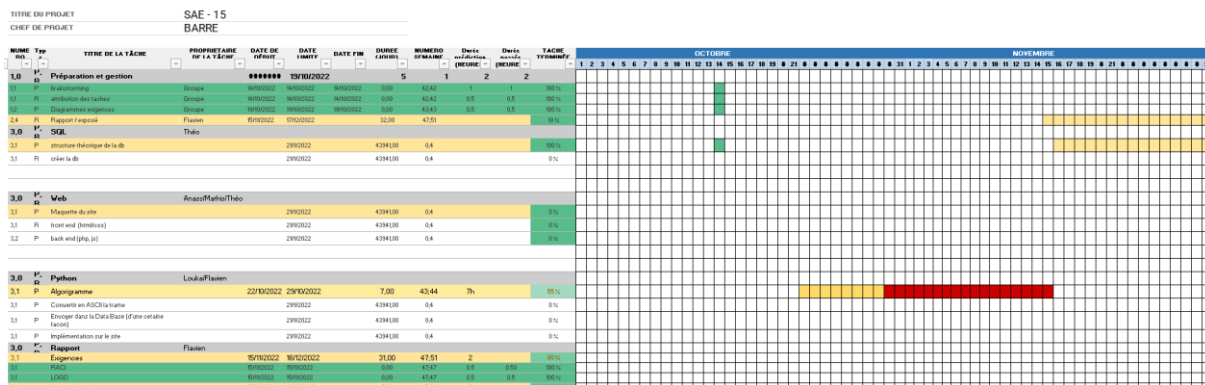
Cette solution nous paraissait importante, car malgré la possibilité de rester connectés par téléphone, nous avons besoin de se retrouver dans un endroit où chacun aurait la parole. De plus, se retrouver pour échanger autour du projet nous permettait de le rendre plus réaliste.

2. Diagramme de Gantt

La première solution mise en place au sein du groupe a été celle d'un diagramme de Gantt. Réalisé sous Microsoft-Excel, ce dernier se présente comme un tableau des tâches, en y regroupant : dates, avancement, décideurs.

Ce diagramme nous permet de prévoir et apercevoir l'avancement des tâches de chacun, de prendre des décisions pour apporter son aide dans les tâches en retard/difficulté.

DIAGRAMME DE GANTT



3. Tableau RACI

Une fois les tâches définies, datées et attribuées, il convenait de définir les rôles de chacun au sein des tâches. Cette solution nous permet de s'organiser et de se concentrer sur les tâches qui nous concerne, avant d'aider un collègue.

Colonne1	Louka	Annas	Mathis	Marwan	Theo	Flavien
Python	R	x	x	x	I	A/R
BDD	I	I	I	I/C	A/R	I/C
WEB	x	A/R	R	R	x	I/C
Rapport	I/C	I/C	I/C	I/C	I/C	A/R

4. Serveur Discord

Afin de communiquer efficacement au sein du groupe, bien que le diagramme de Gantt nous permette de voir l'avancée du projet et des tâches de manière individuelle ; il manquait un moyen de communiquer entre nous, durant nos périodes de travail individuel au domicile.

Ce serveur nous permet de garder une trace des discussions, de faire des réunions vocales pour se tenir au courant de l'avancée du projet ainsi que d'avoir un espace vocal pour se retrouver durant des périodes de travail hors de l'IUT.

5. Google Drive

Finalement, dans le but de centraliser tous les fichiers en relation avec le projet, nous avons mis en place une solution de stockage dans le cloud. Cette solution est entrée en jeu après des problèmes évident d'accès aux fichiers des tâches terminées.

Il était en effet nécessaire, pour certain groupe, d'être informé de l'avancée d'une tâche pour avancer sa propre tâche.

Nous avons alors décidé ensemble, de réunir chaque tâche dans un Google Drive, sur le Cloud, pour pouvoir y accéder peu importe où et quand.

IV. Risques et Opportunité :

1. Les risques :

Après une première réflexion, nous avons trouvés une quantité trop importante de risque. Après avoir réunis les doublons et éliminé les risques trop peu pertinents, nous avons réussi à obtenir une liste exhaustive des risques encourus pendant toute la durée du projet.

Nous sommes aussi parvenus à classer ces risques en 4 grandes parties, en se basant sur nos tâches :

A. Les risques humains et internes au groupe

- Un abandon du cursus par un des membres doit être envisagé
- Un membre ayant une maladie longue est un problème à prévoir
- Un manque de communication, des tensions sont probables, il faut agir en conséquence afin de correctement communiquer

B. Les risques concernant l'application en Python

- Une corruption de la conversion si plusieurs conversions sont lancées en même temps
- Fichier non disponible ou mauvais chemin inséré
- Un crash de l'application si les calculs sont trop complexes/volumineux
- Un ralentissement de l'application et de la machine, si celle-ci est utilisée en parallèle
- Une perte totale des données résultats si l'application est brutalement interrompue
- Un problème de compatibilité entre la version de python utilisée et celle installé
- Un problème de compatibilité entre des librairies non mises à jour

C. Les risques concernant le support web

- Ralentissement de la page web si une grande quantité de données doit être affichée

D. Les risques concernant la base de données

- Limitation de stockage de la base de données dans le temps
- Perte des données stockées

2. Les Opportunités :

S'il est important de savoir prendre conscience des risques liés au projet et les conséquences qu'ils pourraient avoir sur l'avancement et la réalisation du dit projet, il faut aussi rendre compte de possibles opportunités qui pourraient impacter positivement le groupe.

Ces opportunités peuvent par exemple être :

- Un allongement du temps pour la réalisation du projet
- Un nouveau membre au sein de l'équipe
- Des moyens techniques en plus pour réaliser le projet
- Un allègement des exigences

Concernant notre groupe, nous avons eu la chance de voir arriver un 6^{ème} membre au cours du projet.

Cette arrivée nous a notamment permis d'ajouter une force sur la partie web afin de décharger les autres membres et d'ainsi réduire leur charge de travail.

3. Evaluation des familles de risques :

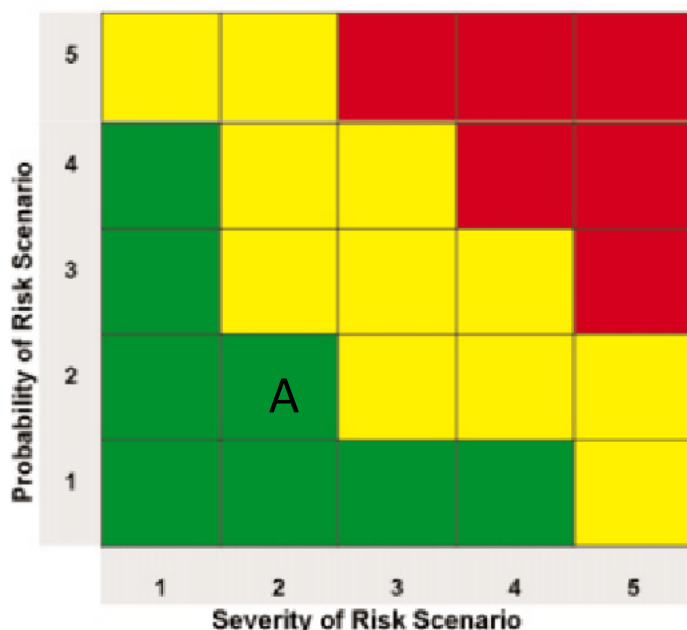
A. Les risques humains et sociaux :

Aucune exigence n'est spécifiée concernant les risques interne au groupe de travail. Cependant, après concertation, l'abandon de cursus au cours du 2^{ème} semestre n'était envisagée par aucun étudiant.

Nous avons tout mis en place pour limiter voir annihiler les risques de tensions à cause de manque de communication (réunion hebdomadaire, channel de discussion, prise en compte de chacun et de ses idées)

Concernant une longue maladie, heureusement aucun membres n'est tombé malade. Le travail a donc pu se faire continuellement, de chaque côté.

Après cette concertation nous avons donc conclu sur le diagramme suivant :



Au vu des réflexions, la conclusion reste que la probabilité d'un problème humaine au sein du groupe est peu probable (niveau 2). Le seul risque encouru est la maladie. Ce problème est décelé et une procédure est prête en cas de maladie, courte ou longue durée. La sévérité du risque est donc peu élevé (niveau 2)

B. Risque de l'application Python

Afin de réduire au maximum les problèmes rencontrés par l'application, une série d'exigences ont été mise en place. En concertation et suite à des test multiples, nous sommes parvenus aux conclusions suivantes, en considérant que les exigences soient respectées :

- I. Une corruption de la conversion si plusieurs conversions sont lancées en même temps est **très peu probable**(niveau 1). En effet, selon l'exigence **GEMAB_APPLICATION -accessibilite-360**, un seul test doit être effectué à la fois.
La sévérité d'un tel risque est **assez élevé (niveau 3)** car il touche à l'intégrité des données.
- II. Un crash de l'application si les calculs sont trop complexes/volumineux est assez probable. Les fichiers testés n'étant pas de grande taille, le risque de crash de l'application sur le traitement de fichiers volumineux est inconnu. Cependant au vu de nos test effectués et de l'efficacité du code python ainsi que des calculs théorique de durée de traitement. En concordance avec l'exigence **GEMAB_APPLICATION -accessibilite-440**, nous qualifions ce risque de **assez probable** (niveau 4)
Concernant la sévérité du risque, aucun dommage technique ne pourrait être

causé par le crash de l'application. Seul les données manquantes dans la base de données serait la conséquence. La sévérité est donc qualifiée au niveau **peu élevé** (niveau 2)

- III. Un ralentissement de l'application et de la machine, si celle-ci est utilisée en parallèle est un risque majeur dans la solution utilisée. Le code python nécessitant beaucoup de ressource, toutes autres applications ouverte ralentira inévitablement le traitement du fichier binaire. Au vu de ces éléments, en partant du principe que nos exigences sont respectées (exigence **GEMAB_APPLICATION -accessibilite-400** , nous qualifions ce risque de très **peu probable** (niveau 1)

Pour la sévérité, un ralentissement ne causera pas de dommages physiques, mais peut causer des retards importants. C'est pourquoi nous qualifions la sévérité à **assez élevé** (niveau 3)

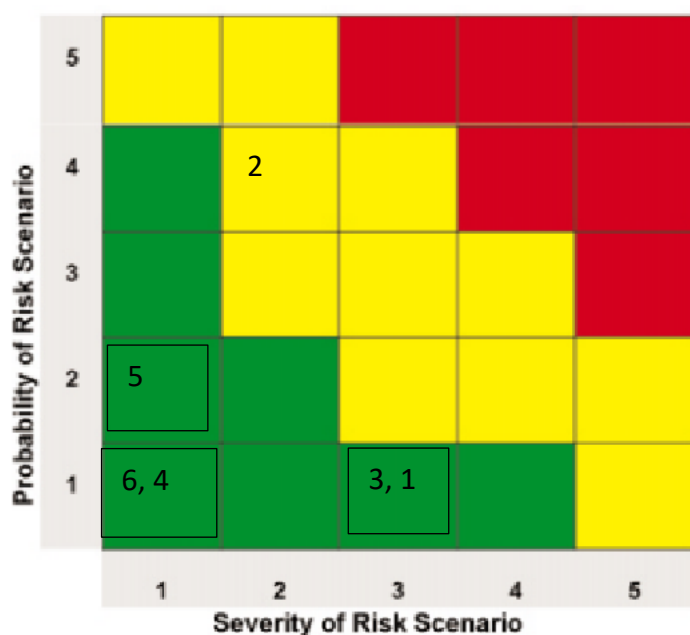
- IV. Une perte totale des données résultats si l'application est brutalement interrompue est peu probable. En effet, lors de l'interruption de l'application des données pourraient être manquantes, le processus de traitement n'étant pas arrivé à terme. Cependant, notre solution ne permettra pas la perte de données, celles-ci étant directement envoyées dans la base de données. Si le test s'interrompt subitement, il suffira de relancer l'application, et le test. Au vu de ces éléments et de l'exigence **GEMAB_APPLICATION -accessibilite-400**, le probabilité d'occurrence de perte de données est qualifié de **très peu probable** (niveau 1)

Concernant la sévérité du manque des données, elle est qualifiée de **très peu élevée**. (niveau 1) En effet, les données n'étant pas perdu, ni le fichier binaire original, pour recouvrer les informations manquantes il suffit de relancer le test.

- V. Un problème de compatibilité entre la version de python utilisée et celle installée est très peu probable. Au vu de l'exigence **GEMAB_APPLICATION -accessibilite-420-A**, et de la version de Python obligatoire, la probabilité d'erreur de version est qualifiée de **peu probable** (niveau 2).

Concernant la sévérité d'une telle erreur de compatibilité, elle est jugée moindre, car n'entraîne en aucun cas une perte de données ou des dommages physiques. Il faudra seulement mettre à jours les librairies employées. La sévérité est donc qualifiée de **très peu élevée** (niveau 1).

- VI. Un problème de compatibilité entre des librairies non mises à jour est très peu probable, au vu de l'exigence **GEMAB_APPLICATION -accessibilite-420-B**. Si une mise à jour régulière des paquets est assurée, la probabilité d'erreur de compatibilité qualifiée de **très peu probable** (niveau 1)
Concernant la sévérité, elle est analogue au risque précédent. Elle est donc qualifiée de **très peu élevée** (niveau 1).



C. Risque support Web

Le support web est le cœur de la solution. Sans lui, impossible d'observer les données traitées par l'application, ni même d'accéder au contenu de la base de données. Afin de se prémunir au maximum, les risques majeurs suivants ont été révélés :

- I. Le ralentissement de la base de données à cause d'une trop grande quantité de données à afficher est un risque important. Le problème réside dans la probabilité que cet événement ait lieu. En coercition avec l'exigence **GEMAB_WEB-accessibilite-150**, nous limitons le nombre de résultats par page, pour limiter les risques d'apparition du risque. Au vu des moyens mis en place, nous qualifions la probabilité d'apparition à **peu probable** (niveau 2).
La sévérité du risque a été jugé relativement minime. Le ralentissement de l'affichage ne cause pas de dommages physique ou de perte de données. Le risque est donc qualifié de **peu élevé** (niveau 2).

D. Risque Base de données

La base de données stocke toutes les données traitées par l'Application Python. Sa disponibilité est critique pour pouvoir accéder aux données en temps et en heure. Afin de prévoir toutes période noire pour la base de données, nous avons identifié les risques majeurs suivants :

- II. La limitation de stockage de la base de données dans le temps n'est pas impossible. Selon la taille des fichiers traités par l'application python, le système d'exploitation et la base de données peuvent être limités en stockage. La probabilité que ça arrive a été jugée très faible (niveau 1). Avec la prise en compte de l'exigence **GEMAB_BDD -accessibilite-260-A**, le risque de manque d'espace de stockage est quasi-nul.
En revanche la sévérité a été jugée plus haute, car elle peut entraîner une indisponibilité de la base de données et des retards.

- III. La perte des données stockées est un risque sévère. Le cas échéant, tous les tests seraient à refaire et leurs résultats à interpréter, impliquant une perte de temps considérable. L'exigence **GEMAB_BDD -accessibilite-260-B** couvre ce risque, en exigeant des backups hebdomadaires de la base de données, sous n'importe quelle forme. L'apparition du problème reste tout de même peu probable (niveau 2).
La sévérité du risque est **très élevée** (niveau 5), car la perte des données serait une catastrophe. C'est pourquoi il faut prêter une attention très particulière à la base de données et **effectuer les backups de manière assidue**.

V. Exigences du projet :

A. Exigences Globales :

Reference _GEMAB-GLOBALE-0

Tous les outils développés doivent être accessible et disponibles sous une distribution GNU/LINUX.

Reference _GEMAB-GLOBALE-1

La totalité des solutions utilisées pour réaliser le projet doivent être libre de droits.

B. Exigences Web :

Reference GEMAB_WEB-accessibilite-100

Le support web doit impérativement être disponible localement.

Reference GEMAB_WEB -accessibilite-110

Le support web aura pour unique fonction l'affichage du contenu de la base de données.

Référence GEMAB_WEB -accessibilite-120

Le support web sera lié à une base de données du type phpMyAdmin.

Référence GEMAB_WEB -accessibilite-130

Le support web contiendra un historique des tests effectués précédemment.

Référence GEMAB_WEB-accessibilite-140

Les labels des données affichées de la page web devront être modifiables par l'utilisateur.

Référence GEMAB_WEB-accessibilite-150

Un nombre limité mais personnalisable de test sera affiché sur la page dédiée

Référence GEMAB_WEB -securite-10

Le support web ne sera pas protégée par un système d'authentification.

C. Exigences Base de Données :

Référence GEMAB_BDD -accessibilite-200

La base de données sera hébergée sous l'utilitaire phpMyAdmin.

Référence GEMAB_BDD-accessibilite-210

La base de données contiendra au moins les 3 champs suivants : Adresses MAC, Adresses IP et Fonctions de transfert 0 à 7.

Référence GEMAB_BDD -accessibilite-220

La base de données contiendra des tables repérables par un identifiant unique pour chaque Test.

Référence GEMAB_BDD -accessibilite-230

Chaque table de la base de données contiendra une clé étrangère/primaire afin de lier les tables entre elles.

Référence GEMAB_BDD -accessibilite-240

La base de données sera liée au support web et accessible sur serveur local en php.

Référence GEMAB_BDD -accessibilite-250

La base de données sera compatible avec Python 3.10.

Référence GEMAB_BDD -accessibilite-260-A

Le stockage de la base de données devra faire l'objet d'une attention particulière, sans amoindrir les besoins de stockages

Référence GEMAB_BDD -accessibilite-260-B

La base de données fera l'objet de backup au moins tous les 2 jours.

Référence GEMAB_BDD -securite-21

La base de données n'aura pas de système d'authentification



D. Exigences Application :

Référence GEMAB_APPLICATION-accessibilite-300

L'application sera exclusivement codée dans le langage de programmation Python.

Référence GEMAB_APPLICATION -accessibilite-310

L'application doit pouvoir communiquer et récupérer/envoyer des données à la Base de Données phpMyAdmin.

Référence GEMAB_APPLICATION -accessibilite-320

L'application doit impérativement être sous forme d'un exécutable.

Référence GEMAB_APPLICATION -accessibilite-330

Le code source du code, avec des commentaires, doit être disponible en plus de l'exécutable.

Référence GEMAB_APPLICATION -accessibilite-340

L'application n'accepte que les fichiers binaires ou résumé de test en entrée.

Référence GEMAB_APPLICATION -accessibilite-350

L'application doit convertir les données binaires en données lisibles et compréhensibles par l'équipe d'ingénierie.

Référence GEMAB_APPLICATION -accessibilite-360

Une seule conversion à la fois sera possible.

Référence GEMAB_APPLICATION -accessibilite-370

L'application doit pouvoir, au moyen des fonctions transferts, changer des données s'ils correspondent à la syntaxe attendue.

Référence GEMAB_APPLICATION -accessibilite-380

Les fichiers étant traités par la machine doivent être aux formats ".rep" pour le fichier de configuration du test et ".bin" pour les trames du banc de test.



Référence GEMAB_APPLICATION -accessibilite-390

Le code étant sensible à la casse, la syntaxe des fichiers ne doit pas changer.

Référence GEMAB_APPLICATION -accessibilite-400

Afin de générer les données de la meilleure façon possible, uniquement l'application python doit être en cours d'exécution sur la machine, tout le long de son processus.

Référence GEMAB_APPLICATION -accessibilite-410

L'application Python ne doit en aucun cas être interrompue dans son processus.

Référence GEMAB_APPLICATION -accessibilite-420-A

La version de python utilisée est Python 3.10 et ses librairies compatibles.

Référence GEMAB_APPLICATION -accessibilite-420-B

Afin de s'assurer de la compatibilités des librairies, il est demandé de mettre à jour les librairies de manière bimensuelle.

Référence GEMAB_APPLICATION -accessibilite-430

Les chemins de fichiers doivent être correctement inséré selon le système Linux. Un fichier non trouvé ou un mauvais chemin terminera l'exécution de l'application

Référence GEMAB_APPLICATION -accessibilite-440

Les fichiers binaires en entrées ne doivent pas excéder une taille de 100Mo pour assurer l'intégrité du processus de traitement du fichier.

Référence GEMAB_APPLICATION -accessibilite-450

Les données extraites du fichier binaires sont immédiatement envoyées dans la base de données pour éviter toute perte potentielle.

Référence GEMAB_APPLICATION -securite-500

L'application ne sera pas protégée par un système d'authentification.

Référence GEMAB_APPLICATION -traitement-600

Les trames dont le champs « Ethernet Type » vaut 0x800 doivent être traité de la manière suivante :

	bench_1	Frame Date	bench_3	bench_4	bench_5	bench_6	Frame Size
en octet	8	8	4	4			4
en bits	64	64	32	12	4	16	32
valeur fixe						0	
fonction transfert					FT_0		
commentaire		nb sec					nb octets
encodage	N.A	IEEE754 en double précision	entier	N.A	entier	N.A	entier

Ether Type = 0x800 (trame UDP)	MAC Dest @	MAC Source @	field_1	field_2	field_3	field_4	field_5	field_6	field_7	field_8	Source IP @	Dest IP @	field_9	field_10	field_11	field_12	field_13	field_14	field_15	field_16	field_17	field_18
en octet	6	6	2	2	2	2	2	1	1	2	4	4	2	2	2	2	3	1	1	3	3	5
en bits	48	48	16	16	16	16	16	8	8	16	32	32	16	16	16	16	000					
valeur fixe																			1			
fonction transfert	MAC @	MAC @									IP @	IP @						FT_7			FT_5	FT_2
commentaire																						
encodage	6 entiers	6 entiers	entier	entier	entier	entier	entier	entier	N.A	N.A	4 entiers	4 entiers	entier	entier	entier	N.A	N.A	entier	N.A	entier	entier	entier

Ether Type = 0x800 (trame UDP)	MAC Dest @	MAC Source @	field_1	field_2	field_3	field_4	field_5	field_6	field_7	field_8	Source IP @	Dest IP @	field_9	field_10	field_11	field_12	field_13	field_14	field_15	field_16	field_17	field_18
en octet	6	6	2	2	2	2	2	1	1	2	4	4	2	2	2	2	3	1	1	3	3	5
en bits	48	48	16	16	16	16	16	8	8	16	32	32	16	16	16	16	000					
valeur fixe																			1			
fonction transfert	MAC @	MAC @									IP @	IP @						FT_7			FT_5	FT_2
commentaire																						
encodage	6 entiers	6 entiers	entier	entier	entier	entier	entier	entier	N.A	N.A	4 entiers	4 entiers	entier	entier	entier	N.A	N.A	entier	N.A	entier	entier	entier

Référence GEMAB_APPLICATION -traitement-510

Les trames dont le champs « Ethernet Type » vaut 0x806 doivent être traité de la manière suivante :

Ether Type = 0x806 (trame ARP)	bench_1	Frame Date	bench_3	bench_4	bench_5	bench_6	Frame Size	MAC Dest @	MAC Source @	field_1	field_2	field_3	field_4	field_5	field_6	MAC Sender @	Sender IP @	MAC Target @	Target IP @	
en octet	8	8	4		4		4	6	6	2	2	2	1	1	2	6	4	6	4	n
en bits	64	64	32	12	4	16	32	48	48	16	16	16	8	8	16	48	32	48	32	8 x n
valeur fixe						0														
fonction transfert					FT_0			MAC @	MAC @							MAC @	IP @	MAC @	IP @	
commentaire		nb sec					nb octets													
encodage	N.A	IEEE754 en double précision	entier	N.A	entier	N.A	entier	6 entiers	6 entiers	entier	entier	entier	entier	entier	entier	6 entiers	4 entiers	6 entiers	4 entiers	N.A

VI. Plan de Validation

Le plan de validation vise à assurer de l'intégrité et de la conformité du produit livré au client, avec ses attentes et exigences. Il permet aussi la vérification de la fonctionnalité du produit, et permet d'avoir le périmètre d'utilisation sûr de la solution (car testé et validé via une procédure adaptée).

1. Procédures de test

Les tests visent à assurer de la fonctionnalité, de l'ergonomie et de l'intégrité des données traités.

Ces test sont soumis à des procédures définies par notre équipe, afin d'assurer la qualité et le périmètre d'action du test. Ces procédures permettent de prendre en compte les spécificités de chacune des solutions, permettant un test plus poussé, approfondi et adapté.

A. Procédures de l'application Python :

Afin de s'assurer de la conformité du produit Python avec les attentes du client (Thalès Alenia Space), de la fonctionnalité du code et du champs d'application de la solution, les procédures de test suivantes ont été mises en place :

- IV. Vérification de la **prise en charge des fichiers REP et BIN en entrée** (annexe 1)
- V. Vérification de la **connectivité avec la Base de Données** et de **l'intégrité des données transférées** (annexe 2)

B. Procédure du support web :

Page principale :

- VI. Vérification de la fonctionnalité de **l'Historique** et **des liens** (annexe 3)
- VII. Vérification de la fonctionnalité de **la barre de recherche** et de **l'affichage des résultats** (annexe 4)
- VIII. Vérification de la fonctionnalité du **paramétrage du nombre de test par page** et du **nombre de test dans l'historique** (annexe 5)

Page Secondaire :

- IX. Vérification de la fonctionnalité du **système de choix de page** et de la **sauvegarde du titre des colonnes** (annexe 6)
- X. Vérification de **l'intégrité de l'affichage des données de la Base de Données** (annexe 7)

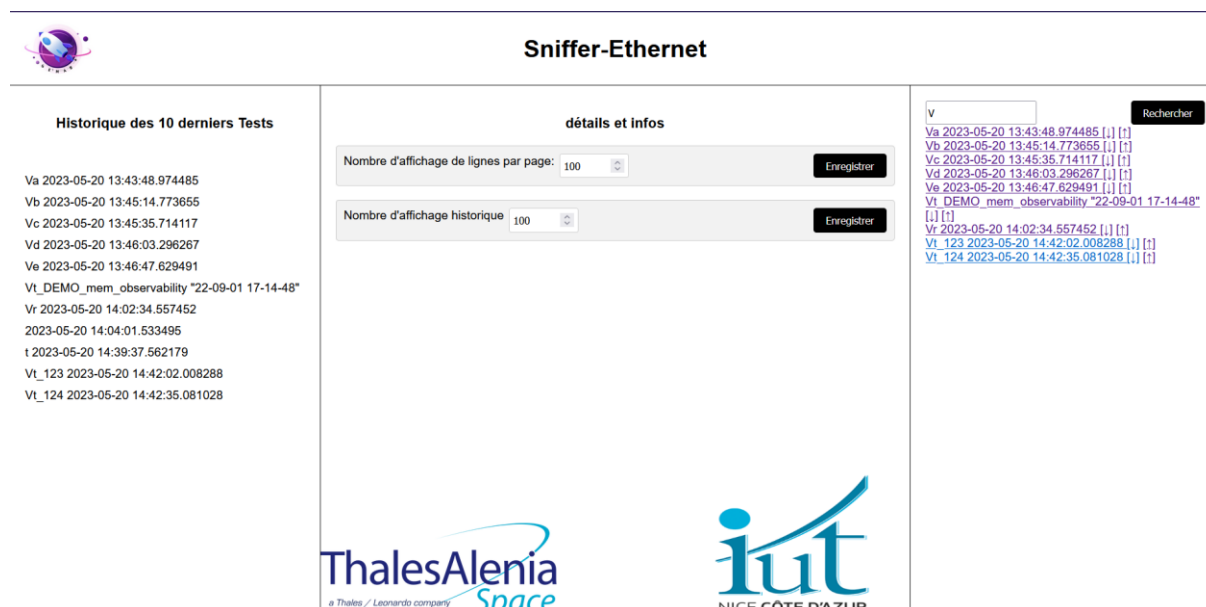
VII. Supports et application :

1. Support web :

Le support web affiche les tests effectués ainsi que leurs multiples exécutions.

Afin d'afficher toutes les données de manière claire et explicite, nous avons décidé de coder le site sur **2 pages web différentes** : une page principale où s'affichent les différents tests et où nous pouvons effectuer des recherches ; une seconde page sur laquelle sont affichés les données contenues dans chaque exécution du test.

A. Page principale :



Dans la première case, le cadre **historique** permet l'affichage des **10 dernières exécutions** faites. Ce cadre offre à l'utilisateur un accès rapide et direct aux données de test.

Au centre, un cadre de **configuration** permet à l'utilisateur de définir les **paramètres généraux** pour le site web comme le nombre de trames par pages, le nombre de test affichés dans l'historique.

Dans le dernier cadre, une **barre de recherche** permet à l'utilisateur de rechercher un test précis dans la base de données et d'en afficher les multiples exécutions.

Cette **barre de recherche** permet de rechercher **des tests selon plusieurs mots**. Cependant les mots entrés pour effectuer la recherche ne peuvent uniquement **concerner le nom du test**. Le nom des tests est cliquable pour en afficher les exécutions, comme pour les cacher.

Nous projetons d'améliorer l'affichage du cadre à droite, en utilisant une meilleure police d'écriture pour que le tout soit plus agréable à lire.

Dans le cadre situé au centre de page, **nous projetons** d'y inscrire des informations sur la base de données, telle que le nombre de test et le nombre d'exécution enregistrés.

Implémentation barre de recherche :

```
<form name="fo" method="get" action="">
  <input type="text" name="keywords" placeholder="Mots-clés" value="<?php echo $keywords ?>" />
  <input type="submit" name="valider" value="Rechercher" />
</form>
```

Système d'affichage minimum de résultats :

Afin de résoudre des erreurs d'affichage de résultats vide (suite l'affichage de valeurs contenues dans une variable vide, car remplie après, avec la recherche) lorsqu'aucune recherche n'a été effectuée, la boîte de la barre de recherche contient les 10 premiers tests de la base. On y intègre la connexion à la BDD.

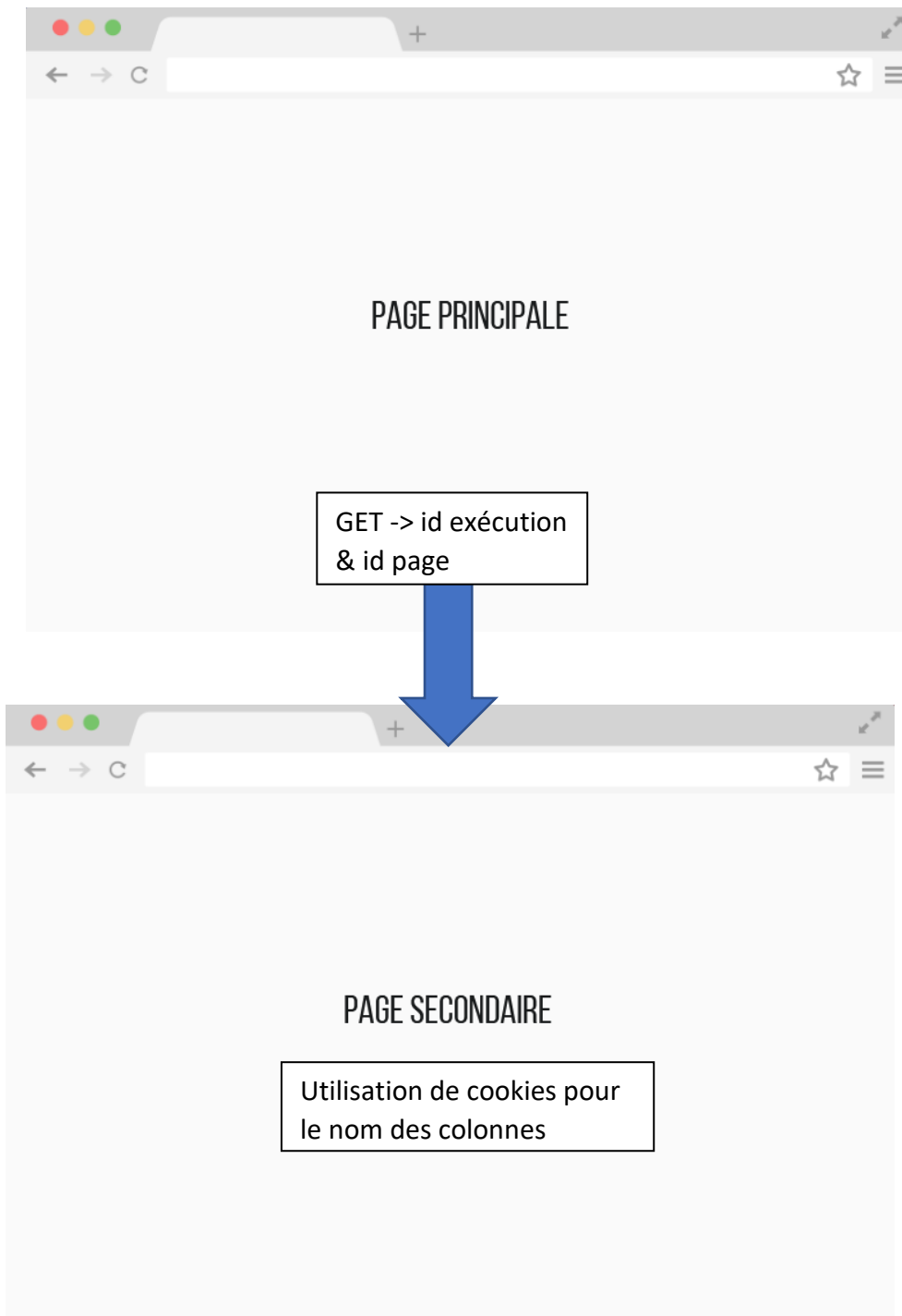
```
<?php for ($i=0;$i<count($tab_recherche);$i++){?>
  <?php echo "<a href='#afficher{$i}'>{$tab_recherche[$i]['nom_test']} {$tab_recherche[$i]['date']} [ &darr;]</a> <a href='#'>[&uarr;]</a>";
  try {
    $bdd = new PDO ( "mysql:host=localhost;dbname=thales",
      "root", "" );
    $bdd->exec('SET NAMES utf8');
  }
  catch (Exception $e) {
    die ("Erreur: Connexion à la base impossible");
  }
}
```

Affichage des résultats de la recherche :

Finalement, lors d'une recherche, est recherché dans la base de données une correspondance concernant le nom du test, puis on affiche tous les tests qui correspondent :

```
$sql = "SELECT id_test, date_exec, exec_id FROM execution
INNER JOIN test ON execution.id_test = test.test_id; "; // Stocke le code SQL de la requête
$req = $bdd->prepare ($sql); // Requête préparée
$req->execute (); // Requête exécutée
$LesExecs = $req->fetchall (); // Requête exécutée
$req->closeCursor (); // Requête détruite
echo "<p id='afficher{$i}'>";
for ($w=0;$w<sizeof($LesExecs);$w++){
  if ($LesExecs[$w]['id_test'] == $tab_recherche[$i]['test_id']){
    echo "<a href='resultats.php?exec_id={$LesExecs[$w]['exec_id']}&pageid=0'>{$LesExecs[$w]['id_test']} {$LesExecs[$w]['date_exec']}</a>";
    ?> <br> <?php
  }
}
```

Il faut noter que, dans les 2 cas d'affichages, les résultats sont affichés avec des flèches à côté, permettant d'afficher les exécutions du test.



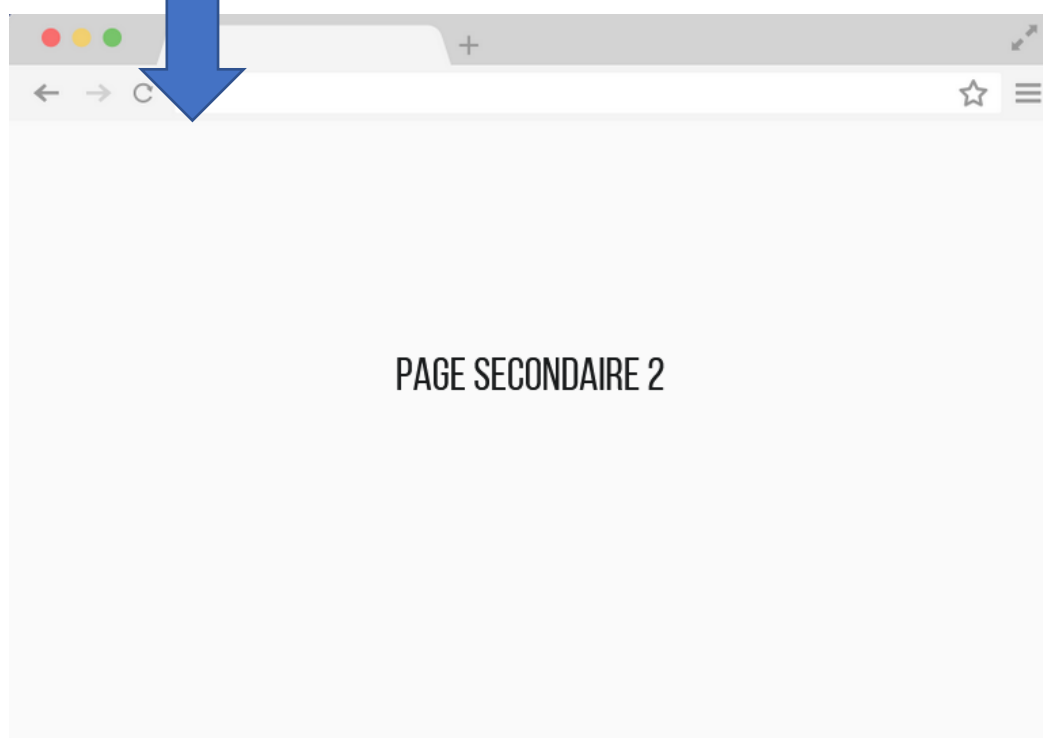
Grace aux cookies, les titres sont persistant et permanents.

Nous changeons de page pour voir la suite des tests, donc, pour communiquer avec la page d'affichage des données suivante :

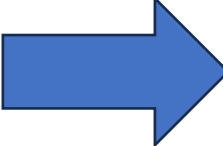
L'id exécution permet de récupérer les données de l'exécution que l'on souhaite observer, id page permet de changer de page et l'id données max permet de limiter le nombre de données par pages.



GET -> id exécution, id page, id
données max



Page secondaire

Champs		Champs	
Numero			FT6
Ligne	date	Champs 2, 3, 4, 5, 31, 32,	
0	04:03:2022:15:37:01		
1	04:03:2022:15:37:01		
2	04:03:2022:15:37:01		
3	04:03:2022:15:37:01		
4	04:03:2022:15:37:01		
5	04:03:2022:15:37:01		
6	04:03:2022:15:37:01		
7	04:03:2022:15:37:01		
8	04:03:2022:15:37:01		
9	04:03:2022:15:37:01		
10	04:03:2022:15:37:01		
11	04:03:2022:15:37:01		
12	04:03:2022:15:37:01		
13	04:03:2022:15:37:01		
14	04:03:2022:15:37:01		
15	04:03:2022:15:37:01		
16	04:03:2022:15:37:01		
17	04:03:2022:15:37:01		
18	04:03:2022:15:37:01		
19	04:03:2022:15:37:01		
20	04:03:2022:15:37:01		
21	04:03:2022:15:37:01		
22	04:03:2022:15:37:01		
23	04:03:2022:15:37:01		
24	04:03:2022:15:37:01		
25	04:03:2022:15:37:01		
			00000000000011001110000000101
			0110100000100010000000000000
			1000000000000000100100000000
			00001000000000000000001000100
			00100000000000000000000010001
			00000000000000000000000000001
			11111000111111101101101101101
			00000000000000000000000000000
			000000000000000011101000010000
			0000000000111110011001010101
			10000000010000000000001000101
			0010000101011100000101101010
			00000000000000000000000000000
			1100011010001001000010111001
			00000000000000000000000000000
			00000000000000000000000000000
			00000000000000000000000000000
			00000000000000000000000000000
			00000000000000000000000000000
			01010000000000000000000000000
			0000001111110101011000000101
			0000010010110101001011101000
			0111110000100000111010011011
			11010000000000000000000000000
			0100000110000100000000000100
			00000000000000000000000000000

La page secondaire n'a pour vocation que **d'afficher les résultats** des exécutions de chaque test.

Cet affichage se fait dans **un tableau** affichant l'entièreté des données, dans leur colonne respective. Afin **d'identifier au mieux** chaque ligne, **un numéro de ligne** est inséré en première colonne.

Toujours dans l'idée d'un affichage clair des données, **les résultats affichés par page sont limités, mais personnalisable sur la page principale**. Pour changer de page, des symboles « < » et « > » sont disponibles en bas de page. Les informations d'affichage sur les pages suivantes ou précédentes sont transmises par un formulaire « GET » (via l'URL).

Il faut noter que le remplissage du tableau se fait de manière automatique, ce qui permet d'adapter ledit remplissage à n'importe quelle taille de trame.

Comme exigé, afin de mieux se repérer dans les données, **le titre des colonnes est modifiable**. Le système utilisant **des cookies**, **il faut suivre la procédure** comme suit pour voir le **changement apparaître immédiatement** :

- XI. Taper le nom de la colonne
- XII. Sauvegarder les modifications avec le bouton dans la dernière colonne
- XIII. Rafraichir la page

2. Base de données phpMyAdmin :

La base de données contient toutes les données traitées par l'application Python. Elle **contient 5 tables**, permettant un classement idéal des données et ce, de manière hiérarchique.

En annexe 1, le schéma complet de la base de données.

En conclusion dudit schéma, la base de données est normalisée en 3NF :

- XIV. Tous les champs contiennent une unique information (unicité)
- XV. La table ne contient pas de clé composée (pas de contrainte de clé composée)
- XVI. Tous les champs sont indépendants

Cette normalisation permet d'assurer l'absence de doublon d'information au sein de la table de données.

La base de données est donc normalisée et fonctionnelle.

3. Application Python

L'application Python est codée pour traiter les données binaires en entrée, les convertir et les enregistrer dans une base de données. Le langage choisi est du python, autorisant une certaine flexibilité sur l'implémentation du code d'insertion des résultats dans la base de données.

Dans un premier temps, le choix de lire l'entièreté du fichier et de le stocker en RAM a été appliqué. Cela permet de gagner en rapidité d'exécution, mais en augmentant l'utilisation de la RAM. Cependant, les caractéristiques de cette dernière n'ont pas été retenues comme facteur limitant.

Les données sont stockées dans un tableau afin d'améliorer leur accessibilité.

Le code entier du programme se divise en 2 fichiers distincts, afin d'améliorer la lisibilité et la clarté du code.

- XVII. Un premier fichier contient **les fonctions** utilisées afin de traiter les données des trames. C'est dans ce fichier que se feront des modifications sur la lecture du fichier binaire en entrée, de l'envoi dans la base de données, des fields à trier.
- XVIII. Dans le second fichier, on trouve le **code principal du programme**, celui qui utilise les fonctions précédemment définies pour extraire, traiter et envoyer les données. C'est dans ce fichier que se feront les modifications concernant le déroulé du programme.

A. Lecture fichier binaire :

Fonction de lecture du fichier binaire en entrée :

```
def read_binary_file_bits(path) -> list:
    with open(path, 'rb') as f:
        binary_data = f.read()

    bit_array = []
    for byte in binary_data:
        for i in range(7, -1, -1):
            bit = (byte >> i) & 1
            bit_array.append(bit)
    return bit_array
```

Explications :

- XIX. La fonction commence par lire, en mode binaire le fichier spécifié en entrée (lignes 2 et 3). Il faut noter que les données lues sont sous la forme d'octet. Or nous exigeons de les avoir en bit, pour opérer aisément dessus.
- XX. On déclare ensuite une liste **bit_array**, que nous allons remplir en itérant sur la variable contenant les données du fichier **binary_data**.
- XXI. Cette boucle permet de passer de données sous forme d'octet en données sous forme de 8 bits, plus simple à manipuler, mais plus lourd à stocker.
- XXII. Finalement, la fonction renvoie une liste contenant les bits du fichier binaire.

Une fois le fichier en entrée lu, les données doivent être traitées et extraites comme le stipule les exigences **GEMAB_APPLICATION -traitement-500** et **GEMAB_APPLICATION -traitement-510**.

C'est le rôle de la fonction **extracteur**, qui traite toutes les données en faisant appel aux fonctions définies plus tôt, et les insère dans la base de données.

B. Les extracteurs

Un extrait de la fonction extracteur :

```
def extracteur(cursor, id_exec):
    decalage lec = 0
    file_bin = read_binary_file_bits(path) #on garde le fichier binaire en mémoire pour rapidement y accéder et ne le lire qu'une seule fois
    taille = len(file_bin)
    print(taille)
    while True:
        arp_udp = bin2hex(file_bin[(40+decalage lec)*8:(42+decalage lec)*8])
        ips = lire_addr_ip(file_bin, decalage lec)
        if ips == None:
            return 0
        date_exec = read_date(path)
        size = taille_paquet(file_bin, decalage lec)
        macs = lire_addr_mac(file_bin, decalage lec)
        mac_s = macs[0]
        mac_d = macs[1]
        ip_s = ips[0]
        ip_d = ips[1]
        fields = lire_fields(file_bin, decalage lec)
        FT = lire_FT(file_bin, decalage lec)
        FT6 = lire_FT6(FT, fields, file_bin)
        decalage lec = decalage lec + size + 28
```

Cet extrait de code s'occupe de la lecture et de l'extraction des données principales de la trame, en faisant appel à des fonctions définies plus tôt. On stocke le tout dans une liste python.

Concernant les trames ARP, la vérification se fait par le type de trame, 806 ou 800. Elle est à part de l'extracteur principal, car appelée occasionnellement.

```
def extracteur_ARP(fic, decalage, id_num, id_exec_arp):
    conn = connector.connect(host="localhost",user="root",password="", database="thales") #ajouter les valeurs du bench2 et bench3
    cursor = conn.cursor()
    date = read_date(path)
    size = taille_paquet(fic, decalage)
    macs = lire_addr_mac(fic, decalage)
    mac_s = macs[0]
    mac_d = macs[1]
    date_frame = packet_date(fic, decalage)
    fields = lire_fields(fic, decalage)

    valarp = (date, size, mac_d, mac_s, id_num, id_exec_arp) #on lit que jusqu'au field 30, rajouter field 33_34_35 et field 32

    sql = 'INSERT INTO arp1(frame_date, frame_size, adresse_mac_dest, adresse_mac_source, id_test, id_exec) VALUES(%s, %s, %s, %s, %s, %s)'

    cursor.execute(sql, valarp)
    conn.commit()
    conn.close()
```

La fonction établie une connexion à la base de données et envoie les données extraites des trames ARP dans la base de données.

B. Fonctions transfert :

Les fonctions transfert doivent permettre la modification de champs par des valeurs définies par l'opérateur. Pour gérer cette exigence, nous avons mis en place un fichier de configuration (ft_fonctions.py) qui permet la configuration des fonctions transfert sous la forme de dictionnaires python.

```

1  #Fichier cont
2  #Les fonction
3
4  FT0 = {}
5
6  FT1 = {}
7
8  FT2 = {}
9
10 FT3 = {}
11
12 FT4 = {}
13
14 FT5 = {}
15
16 FT6 = {}
17
18 FT7 = {}

```

Le changement de valeurs des fonctions transfert se fait au sein d'une fonction qui compare la clé de la fonction transfert (frome brute) avec celle extraite du fichier binaire. Si les 2 correspondent, alors la valeur associée à la clé est insérée.

De plus, concernant la Fonction Transfert 6 qui est positionnée sur plusieurs champs, on trouve ci-dessous le code nous permettant de l'assembler :

```

def lire_FT6(FT_liste, fields_liste, liste) -> str:
    FT6 = []
    FT6.append(FT_liste[6])
    FT6.append(FT_liste[2])
    FT6.append(FT_liste[3])
    FT6.append(FT_liste[4])
    FT6.append(FT_liste[7])
    ch = ""
    for i in range(0, len(FT6)):
        ch+= str(FT6[i])
    return ch

```

On fait ici une concaténation des Fonctions Transfert constituant la FT6. Pour pouvoir la lire correctement, nous la modifions en un string, au travers d'une boucle for, afin de simplifier son insertion dans la base de données ainsi que sa comparaison avec les éventuelles valeurs de sa Fonction Transfert.

C. Fonction principale du code

Finalement, le fichier python contient la directive `if __name__ == "__main__":`: ce qui permet, après cette directive, d'exécuter la séquence de fonctions pour extraire, traiter et envoyer les données dans la base de donnée.

Les exigences voulaient que des paramètres puissent être insérés à l'appel de la fonction, en ligne de commande. La fonction main se plie donc à cette exigence.

Début du code de la fonction principale :

```
if __name__ == "__main__":  
    if len(sys.argv) < 1:  
        print("Error : file path or name test required")  
        exit()  
    elif len(sys.argv) < 3:  
        try: #on vérifie si le paramètre est un chemin ou le nom d'u  
            open(sys.argv[1], "r")  
            data = lire_rep(sys.argv[1]) #Si le paramètre est un che  
            nom_test = data['name']  
            date_test = data['date']  
            is_path = True  
  
        except AttributeError: #Si on obtient AttributeError, alors  
            date_test = str(datetime.datetime.now()) #On récupère la  
            valtest2 = (sys.argv[1], date_test) #on met en Tuple le c  
            is_path = False  
    else:  
        print("Error : too much parameter (file-path, test-name)")  
        exit()
```

Conclusion

Dans ce rapport nous avons exposés les étapes de conception de la solution, du support web à l'application Python, en passant par la base de données. Les aspects techniques apportés permettent une compréhension approfondie de notre solution, consolidée par les preuves matérielles attestant de la conformité, de l'intégrité et de la fiabilité de la solution apportée.

Nous avons commencé par revoir les exigences du clients et en extraire les risques potentiels qu'il faut encadrer, surveiller et solutionner en mettant en place des exigences propres à l'équipe de conception, nécessaire au bon fonctionnement du projet.

Nous avons ensuite détaillé le plan de validation de ladite solution. Ce plan de validation permet d'attester de la fonctionnalité de tout les éléments de la solution par la justification de procédures de test menées par l'équipe développement.

Finalement, nous avons détaillé de manière concise les principales étapes de chaque objets de la solution : les algorithmes et fonctions principales de l'application, les éléments fondamentaux du support web et la structure de la base de données, ainsi que sa normalisation en 3NF.


Durant cette première année de BUT, ce projet en collaboration avec Thalès Alenia Space nous a fait plonger dans le monde professionnel, dans lequel nous serons plongé à la fin de nos études. La gestion d'un projet, depuis la réflexion sur tableaux blanc jusqu'à la mise en service de la solution, est commun dans le monde professionnel, puisque les contrats fonctionnent sous forme de projet.

Cette expérience a été plus qu'enrichissante pour nous tous, et nous nous rendons compte d'une franche évolution dans notre manière d'aborder les projets et de l'importance d'une démarche structurée pour ne pas se précipiter dans la conception immédiatement.

Annexes

Annexe 1 : Procédures Python fichiers Binaires et Rep

1 PRISE EN CHARGE DES FICHIER BIN ET REP


Python Verification / Validation Report				
Verification-Number: 10000000 - 00000001		Test Procedure: PCFBR-1	at Date, Time: Start: 08/05/23 12:10 End: 08/05/23 12:13	Verification Result: 
Requirement-Number: <requirement number> <requirement number> <requirement number>	Open NCRs:	Issue: <version> Short Description: Test d'insertion des fichiers binaires et rep Date: 08/05/23 Author: Louka Gulde	Test Report Executed by: Louka Gulde Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Vérification de la prise en charge correct du chemin du fichier binaire et du fichier rep.				
Test Preparation:				
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks
i	Ouverture du terminal	Réussite de l'ouverture du terminal	PASS	L'ouverture du terminal est lente
ii	Se rendre dans le répertoire du fichier python works.py	Réussite du déplacement avec la commande cd	PASS	
iii				
Test Execution:				
1	Lancement de l'application avec la commande : python3 chemin_fic_binaire chemin_fic_rep	Le code se lance, aucune erreur concernant les fichiers n'apparaît en console.	PASS	
2			PASS	
3				
Execution duration: 2min40				
Summary: Le test s'est déroulé sans problème apparents. L'application gère correctement la présence d'un nom à la place d'un chemin pour le fichier REP. Les exceptions sont bien gérées pas l'application. La chaîne de caractère renvoie une erreur et ferme correctement l'exécution du code.				

2. COMPORTEMENT NOM DE TEST FICHIER REP ET INSERTION CHAINE DE CARACTERE FICHIER BINAIRE

Python Verification / Validation Report					
Verification-Number: 10000000 - 00000002		Open NCRs:	Test Procedure: PCFBR-1	at Date, Time: Start: 08/05/23 12:15 End: 08/05/23 12:18	Verification Result: <div>PASS</div>
Requirement-Number: <requirement number> <requirement number> <requirement number>			Issue: <version> Short Description: Test d'insertion des fichiers binaires et rep Date: 08/05/23 Author: Louka Gulde	Test Report Executed by: Louka Gulde Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Vérification de la prise en charge d'un nom de test à la place du chemin du fichier REP puis d'une chaîne de caractère à la place du chemin du fichier binaire					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Ouverture du terminal	Réussite de l'ouverture du terminal	PASS	L'ouverture du terminal est lente	
ii	Se rendre dans le répertoire du fichier python works.py	Réussite du déplacement avec la commande cd	PASS		
iii					
Test Execution:					
1	Lancement de l'application avec la commande : python3 chemin_fic_binaire nom_test	Le code se lance, aucune erreur concernant les fichiers n'apparaît en console.	PASS		
2	Lancement de l'application avec la commande : python3 chaîne_de_caractère chemin_fic_rep	Le code se lance et se termine correctement, avec un message d'erreur.	PASS	Chaîne de caractère entrée : maison	
3					
Execution duration: 3min					
Summary: Le test s'est déroulé sans problème apparents. L'application gère correctement la présence d'un nom à la place d'un chemin pour le fichier REP. Les exceptions sont bien gérées pas l'application. La chaîne de caractère renvoie une erreur et ferme correctement l'exécution du code.					


Annexe 2 : Connectivité Base de données et intégrité des données

3. CONNECTIVITE A LA BASE DE DONNEES

Python Verification / Validation Report					Verification Result:
Verification-Number: 20000000-00000001		Open NCRs:	Test Procedure: CBD-1	at Date, Time: Start: 08/05/23 14:10 End: 08/05/23 14:15	
Requirement-Number: <requirement number> <requirement number> <requirement number>			Issue: <version> Short Description: Vérification de la connexion à la base de données Date: 08/05/23 Author: Théo MARTY	Test Report Executed by: Théo MARTY Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Test de la connectivité à la base de données avec le script python, avec la base de données connectées en ligne, puis déconnectée pour vérifier le fonctionnement du code en tout cas.					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Connexion en ligne de la base de données puis exécution du test				
ii	Déconnexion de la base de données, puis exécution du 2 ^{ème} test				
iii	Reconnexion en ligne de la base de données				
Test Execution:					
1	Exécution de l'application python avec la base de données en ligne	L'application doit marcher correctement, et le processus de traitement se terminer en entier	PASS		
2	Exécution de l'application python avec la base de donnée hors lignes	Le processus doit s'arrêter avec un message d'erreur de connexion à la base de donnée	PASS		
3					

Execution duration: 5min	
Summary: Lorsque la base de données est connectée en ligne, le processus se déroule normalement et entièrement. Lorsque la base de données est déconnectée, le processus s'arrête en levant une erreur. L'application gère correctement la connexion à la base de données.	

2. VERIFICATION DE L'INTEGRITE DES DONNEES TRANSFEREES LORS DU TRAITEMENT

Python Verification / Validation Report					
Verification-Number: 20000000-00000002		Open NCRs:	Test Procedure: VIDTT-1	at Date, Time: Start: 08/05/23 14:30 End: 08/05/23 14:35	
Requirement-Number: <requirement number> <requirement number> <requirement number>			Issue: <version> Short Description: Vérification de l'intégrité des données lors du transfert Date: 08/05/23 Author: Théo MARTY	Test Report Executed by: Théo MARTY Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Verification Result:					
Test Description: Vérification de l'intégrité des données lors du transfert de l'application python vers la base de données.					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Execution de l'application python				
ii	Création d'un fichier texte pour stocker les données traités				
iii	Vérification fonctionnement de la base de données				
Test Execution:					
1	Execution de l'application python	Exécution complète du processus	PASS		
2	Stockage dans un fichier texte des données traitées	Le nombre de ligne du fichier texte doit correspondre au nombre de trame du fichier	PASS		
3	Comparaison des données du fichier texte et de la base de données	Les données prises aléatoirement doivent correspondre dans les 2 fichiers	PASS	Les données ont été croisées et validées	

Execution duration: 30min	
Summary: Les données croisées, récupérées après un test effectué correctement, correspondent bien entre le fichier texte et la base de données.	


Annexe 3 : Historiques et liens

1 VERIFICATION FONCTIONNALITE HISTORIQUE ET LIENS

Website Verification / Validation Report					Verification Result:
Verification-Number: 30000000 - 00000001		Test Procedure: VFHL-1		at Date, Time: Start: 20/05/23 17:15 End: 20/05/23 17:20	<div></div> <div>PASS</div>
Requirement-Number: <requirement number> <requirement number> <requirement number>		Issue: <version> Short Description: Verification du fonctionnement de l'historique et des liens Date: 20/05/23 Author: Anass		Test Report Executed by: Anass EL-BARGUI Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Open NCRs:					
Test Description: Vérification du correct affichage des test de l'historique et du bon renvoi des liens vers les page d'affichage des données					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Vérification de la présence de données dans la base de données			Nécessite au moins 15 test différents présents dans la base de données	
ii					
iii					
Test Execution:					
1	Chargement de la page principale	Affichage correct de la page sans code d'erreur présent	PASS		
2	Vérification de l'affichage de N test et non de la totalité	Seuls les N derniers test doivent s'afficher	PASS		
3	Vérification de la page de renvoi lors du clic sur les liens	Les liens doivent être cliquable et renvoyer sur la page résultat	PASS		
Execution duration: 5min					
Summary: Les données affichées dans l'historique sont conformes et les liens renvois vers les pages d'affichage des résultats					


Annexe 4 : Fonctionnalité barre de recherche et affichage résultats

2. VERIFICATION FONCTIONNALITE BARRE DE RECHERCHE ET AFFICHAGE DES RESULTATS

Website Verification / Validation Report					
Verification-Number: 40000000-00000001		Test Procedure: VFBRAR-1		at Date, Time: Start: 20/05/23 17:50 End: 20/05/23 17:55	Verification Result:
Requirement-Number: <requirement number> <requirement number> <requirement number>	Open NCRs:	Issue: <version> Short Description: Test de fonctionnalité de la barre de recherche et de ses résultats Date: 20/05/23 Author: Anass		Test Report Executed by: Anass EL-BARGUI Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Test de vérification de la fonctionnalité de la barre de recherche et de l'affichage des résultats par une vérification manuelle dans la base de données					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Vérification de la présence de test dans la base de données			Au moins 15 test doivent être présents dans la base de données	
ii					
iii					
Test Execution:					
1	Recherche de test via la barre de recherche	Les résultats affichés doivent être conforme à la recherche effectuée	PASS		
2	Test du défilement des résultats (affichage exécutions)	Les test doivent contenir au moins une exécution	PASS		
3	Test du lien	Les test doivent être cliquable	PASS		
Execution duration: 5min					
Summary: Les données affichées sont bien conformes et la barre de recherche marche correctement. Les noms des test et exécution sont cliquables					


Annexe 5 :Nombre de données par page / nombre de test historique

3. VERIFICATION DE LA FONCTIONNALITE CHOIX NOMBRE TEST / PAGE ET NOMBRE TEST DANS L'HISTORIQUE

Website Verification / Validation Report				
Verification-Number: 50000000 - 00000001		Test Procedure: VFTPTH-1		at Date, Time: Start: 20/05/23 16:40 End: 20/05/23 16:45
Requirement-Number: <requirement number> <requirement number> <requirement number>	Open NCRs:	Issue: <version> Short Description: Test de la personnalisation de l'affichage Date: 20/05/23 Author: Marwan	Test Report Executed by: Marwan Bouhadida Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Test du bon fonctionnement de la personnalisation du nombre de test affichés dans l'historique et du nombre de données affichés par page				
Test Preparation:				
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks
i	Vérification de la présence de test dans la base de données			Il doit y avoir au moins 15 test dans la base de données
ii	Personnalisation de la valeur du nombre de données par page			La valeur par défaut est 100, il faut la changer
iii	Personnalisation de la valeur du nombre de test dans l'historique			La valeur par défaut est 100, il faut la changer
Test Execution:				
1	Observation et vérification des changements	Le changement doit être persistant même après rafraichissement	PASS	
2				
3				
Execution duration: 5min				
Summary: Le changement des valeurs est bien effectif sur les pages et persistant.				

Annexe 6 : Changement de page et noms des colonnes

4. VERIFICATION CHANGEMENT DE PAGE ET SAUVEGARDE NOM DES COLONNES

Website Verification / Validation Report					
Verification-Number: 6000000 - 0000001		Open NCRs:	Test Procedure: VCPSNC-1	at Date, Time: Start: 20/05/23 17:50 End: 20/05/23 18:00	Verification Result:
Requirement-Number: <requirement number> <requirement number> <requirement number>			Issue: <version> Short Description: Test du changement de page et de la sauvegarde du nom de colonne Date: 20/05/23 Author: Marwan	Test Report Executed by: Marwan BOUHADIDA Test Report Checked by: Flavien BARRE Test Report Approved by: Flavien BARRE	
Test Description: Vérification de la transmission des données lors du changement de page, de l'incrémentation de l'id des lignes et du nom des colonnes					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Vérification de la présence de test dans la base de données				
ii					
iii					
Test Execution:					
1	Modification des noms de colonnes	Les noms doivent être changés et persistant après redémarrage	PASS	Il faut enregistrer et rafraichir après le changement du nom d'une colonne	
2	Aller à page suivante, changer le nom d'une colonne	Les noms doivent apparaître sur cette nouvelle page	PASS		
3	Aller à la page précédente	Tout les noms ajoutés doivent apparaître même après redémarrage/rafraichissement	PASS		
Execution duration: 10min					

Summary: Les noms des colonnes sont bien changés sur chaque page et ce de manière persistante (après redémarrage du navigateur ou de l'ordinateur et changement de page). Les pages précédentes et suivantes sont bien disponibles. Aucune erreurs lors de la tentative d'aller à la page 0, -1 ...

Annexe 7 : intégrité des données de la base de données

5. INTEGRITE DE DONNEES RESULTATS AFFICHES

Website Verification / Validation Report					
Verification-Number: 70000000 - 0000001		Test Procedure: IDRA-1		at Date, Time: Start: 27/05/23 18:30 End: 27/05/23 19:00	Verification Result: <div>PASS</div>
Requirement-Number: <requirement number> <requirement number> <requirement number>	Open NCRs:	Issue: <version> Short Description: Vérification de l'intégrité des données affichées Date: 27/05/23 Author: Marwan		Test Report Executed by: Marwan BOUHADIDA Test Report Checked by: Anass EL-BARGUI Test Report Approved by: Anass EL-BARGUI	
Test Description: Vérification de la véracité des données affichées dans le tableau résultat, en comparant à la main avec ceux dans la base de données					
Test Preparation:					
No.	Activity Description	Pass/Fail Criteria	P/F/PWO Status	Remarks	
i	Présence d'au moins 15 test dans la base de données				
ii	Sélectionner un test depuis la page principale				
iii					
Test Execution:					
1	Vérification des données de manière aléatoire	Les données prises au hasard doivent correspondre	PASS	Vérification sur 150 Valeurs	
2	Changement de page et suivi des données	Les données dans les pages succincte doivent corréler avec celles dans la base de données	PASS		
3					
Execution duration: 30min					
Summary: Les données affichées sont bien conformes à celle présente dans la base de données. La continuité de l'intégrité des données est assurées. Les pages succinctes affichent les bonnes données.					

Annexe 8 : Schéma de la base de données

