

## Note de synthèse



BTS – 2<sup>ème</sup> année – Service Informatique aux Organisations

Anthony AFONSO

# Table des matières

<b>INTRODUCTION.....</b>	<b>3</b>
Présentation de Actia.....	3
Localisation.....	4
<b>SEMAINE D'INTÉGRATION.....</b>	<b>5</b>
1.1 Outils de pratiques professionnelles.....	5
1.2 Matériel mis à disposition.....	5
1.3 OBJECTIF :.....	5
<u>1.31 Compétences requises / Technologies utilisées :</u> .....	5
1.4 Logiciels administrés pour mon projet.....	6
<b>2 MISE EN PLACE DU PROJET.....</b>	<b>7</b>
<b>3 RÉALISATION DU PROJET.....</b>	<b>9</b>
3.1 Tables créés.....	9
<u>3.11 Table sla_intervention :</u> .....	10
<u>3.12 Table sla_parameters :</u> .....	10
3.2 Préparation de l'application web.....	11
3.3 Les 3 pages web :.....	11
3.4 Page configuration :.....	11
<u>3.41 Représentation du CR dans Visual Studio Code :</u> .....	12
<u>3.42 Représentation du UD dans Visual Studio Code :</u> .....	13
<u>3.43 Résultat en base.....</u>	14
3.5 Page alertes en cours.....	15
<u>3.51 Résultat en base :</u> .....	17
<u>3.52 Problème rencontrés :</u> .....	17
3.6 Page historique des Justifications.....	18
<b>CONCLUSION.....</b>	<b>19</b>
<b>REMERCIEMENTS.....</b>	<b>19</b>

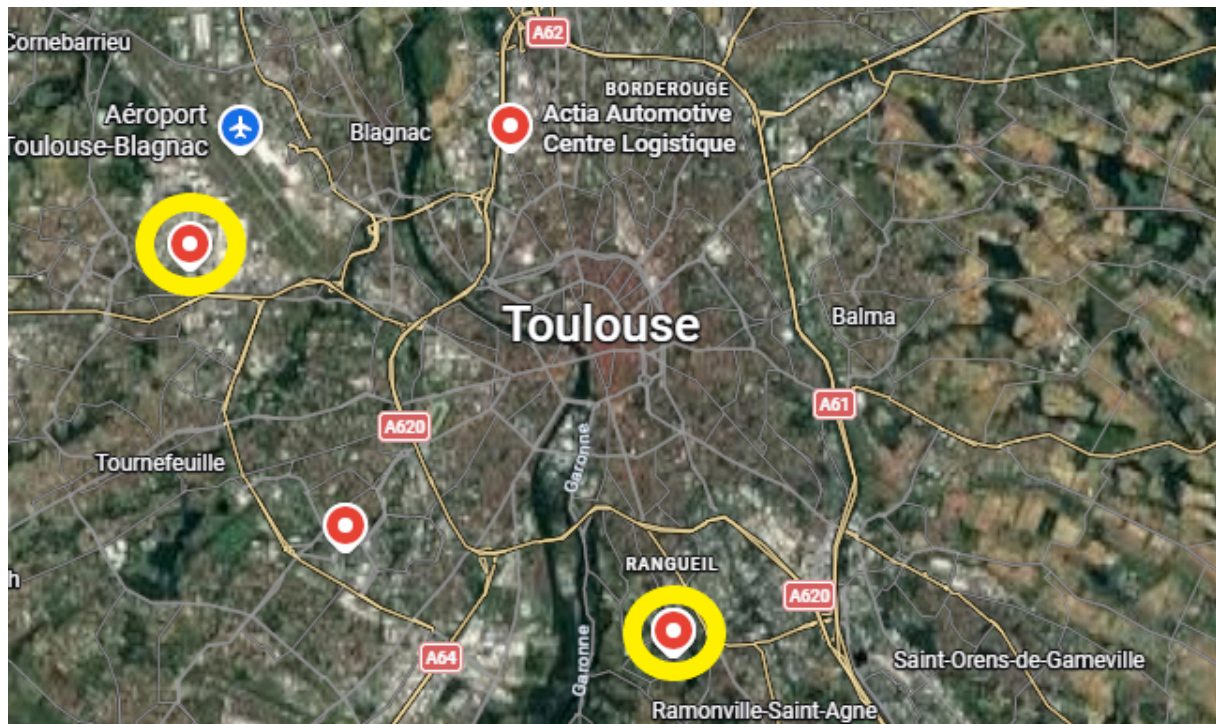
# INTRODUCTION

## Présentation de Actia



ACTIA est une entreprise française créée en 1986 qui conçoit et fabrique des équipements électroniques utilisés dans de nombreux secteurs, notamment l'automobile, l'aéronautique, le spatial, l'énergie et les télécommunications. Elle développe par exemple des systèmes embarqués pour les véhicules, des outils de diagnostic pour les garages, ou encore des équipements pour satellites et réseaux intelligents. ACTIA est une entreprise familiale de taille intermédiaire (ETI), basée à Toulouse, et présente dans une quinzaine de pays, avec plusieurs milliers de collaborateurs, ce qui lui permet de combiner expertise technologique, innovation et présence internationale. Actia investissements soutenus en R&D, partenariats technologiques stratégiques, excellence industrielle, et anticipation des ruptures technologiques.

## Localisation



Durant ce stage, j'ai dû me rendre mobile puisque mes points de rendez-vous se situaient de part et d'autre de Toulouse. À Colomiers, nous avons le centre industriel dans lequel j'ai passé la majorité du temps car c'est là qu'est positionné l'équipe qui m'a aidé durant ce travail, puis le siège à Pouvourville qui, lui, était surtout pour les papiers administratifs et ma semaine d'intégration.

Durant ce stage j'ai pu bénéficier d'un jour de télétravail afin de simplifier les déplacements. j'ai donc dû m'adapter et découvrir le monde du travail à domicile.

# SEMAINE D'INTÉGRATION

Lors de ma première semaine, j'ai découvert en profondeur l'organisation et le fonctionnement du site industriel d'Actia.

## 1.1 Outils de pratiques professionnelles

- J'ai pu retrouver l'organisation des réunions via Teams et Outlook, et leur planification selon les emplois du temps des collaborateurs.
- Le travail à domicile s'est effectué grâce à l'accès à un VPN pour avoir accès au Teams de l'organisation.

## 1.2 Matériel mis à disposition

- Un ordinateur portable performant et un écran m'ont été fournis ainsi que des périphériques tels qu'un clavier et une souris afin de travailler efficacement.

## 1.3 OBJECTIF :

**Le problème** : comme le calcul se fait sur *X heures glissantes*, il peut arriver qu'après résolution du problème, l'alerte visuelle reste active jusqu'à ce que le nombre de défauts sur ces *X heures* repasse sous le seuil.

**Le projet** : développer une mini application web à intégrer sous forme de « frame » dans notre application *Manufacturing Follow Up*, permettant :

- de remettre à zéro le nombre de défauts par l'intervenant qui résout le problème,
- de tracer qui effectue la remise à zéro,
- d'ajouter des commentaires / indications sur le défaut.

### 1.31 Compétences requises / Technologies utilisées :

- Base de données : PostgreSQL
- Frontend : Vue.js / JavaScript
- Backend : Node.js – Express / JavaScript

## 1.4 Logiciels administrés pour mon projet



DBeaver est un outil **pratique et polyvalent** pour administrer et interroger des bases de données sans utiliser uniquement la ligne de commande. Il facilite le travail des développeurs, administrateurs et étudiants grâce à son interface simple et compatible avec presque tous les systèmes de bases de données.



PostgreSQL est une base de données fiable et moderne, capable de gérer aussi bien de petits projets que des systèmes importants. Par rapport à MySQL, il est plus puissant, plus complet et plus strict dans la façon de gérer le SQL, tout en restant entièrement gratuit et open-source.

## 2 MISE EN PLACE DU PROJET

- Dans le cadre de mon projet, j'ai dû me former à des outils tels que DBeaver et PostgreSQL, que j'ai utilisés pour la conception et la gestion de la base de données.
- J'ai dû m'adapter efficacement à la base de données qui m'a été présentée, ce qui m'a permis d'en exploiter pleinement les fonctionnalités et de réaliser mon projet dans de bonnes conditions.
- Par la suite j'ai pu expliquer mon idée d'organisation pour le projet grâce à une réunion Teams dans laquelle j'ai présenté une affiche pour synthétiser celle-ci.

### Gestion des Alertes de Défauts — Projet base de données

Objectif et tables à créer

#### Objectif du projet

**Le problème actuel**  
L'alerte visuelle se déclenche quand 5 défauts similaires se produisent sur la même machine en 8 heures glissantes. Après résolution, l'alerte peut rester active plusieurs heures jusqu'à ce que le compteur descende naturellement.

**Ce qu'on veut pouvoir faire**

- Réinitialiser le compteur manuellement après résolution
- Tracer le nom de la personne qui réinitialise
- Ajouter des commentaires sur chaque défaut (popup)
- Configurer des seuils différents selon l'îlot

#### 2 nouvelles tables à créer

Approche ultra-simplifiée

##### 1. alert\_config

Configuration des seuils par îlot de production

îlot	Nom de l'îlot de production
seuil_defaults	Nombre de défauts avant alerte (ex : 3, 5...)
fenetre_heures	Fenêtre temporelle en heures (ex : 8)

Exemples de configuration :

- ASSEMBLAGE : 3 défauts / 8 heures
- CONTROLE : 2 défauts / 4 heures

##### 2. alert\_history

Historique de toutes les alertes avec début et résolution

ressource / nc_code / îlot	Identification
heure_debut_alerte	Quand l'alerte a débuté
heure_resolution	Quand résolu (NULL si en cours)
resolu_par	Nom de l'intervenant
commentaire	Commentaire de résolution
nb_defaults	Nombre de défauts détectés

#### Exemple concret — Cycle de vie d'une alerte

1. **ALERTE DÉTECTÉE**  
5 défauts sur MACHINE\_01 (NC\_CAPTEUR)  
îlot ASSEMBLAGE (seuil = 3)  
→ Création dans alert\_history

2. **EN COURS**  
heure\_debut\_alerte: 10h30  
heure\_resolution: NULL  
→ Alerte visible à l'écran

3. **RÉSOLU**  
heure\_resolution: 14h35  
resolu\_par: "Pierre Dubois"  
→ Alerte disparaît

✓ Résultat final stocké dans alert\_history

- Durée totale : 4h05 (calculée automatiquement : 14h35 - 10h30)
- Traçabilité complète : QUI (Pierre Dubois) / QUAND (14h35) / POURQUOI (commentaire)

Malheureusement l'idée proposée n'a pas convenu pour réaliser mon projet et l'équipe qui me suivait sur celui-ci m'a aiguillé jusqu'à trouver une solution finale pour réaliser le projet.

Page 7 sur 19

Avant de commencer réellement le projet, mon tuteur m'a demandé de réaliser un test sous le format d'un TP en me mettant sur une nouvelle base de données.

L'objectif de ce test était d'évaluer mon niveau pour savoir comment m'accompagner dans ce stage.

Ce TP est composé de 20 questions dans lesquelles je devais faire des requêtes afin d'afficher chaque réponse.

### **Sujet - TP Postgre SQL**

Trouver les requêtes qui permettent d'afficher les informations demandées ci-dessous :

Requête 1 : La liste des chercheurs

Requête 2 : Les noms et les numéros des Chercheurs

Requête 3 : Les noms des chercheurs et le nom de leur équipe

Requête 4 : Les noms des Chercheurs de l'équipe 'QUIZZ 2'

Requête 5 : Les équipes dont le budget est compris entre 10 000 et 20 000 euros

Requête 6 : Equipes dont budget est supérieur à l'équipe 'COVOIRUTRAGE 2'

Requête 7 : Le nombre de chercheurs de l'équipe 'QUIZZ 2'

Requête 8 : Lister par chercheur, le nombre de thèmes sur lesquels ils travaillent

Requête 9 : Lister le nombre d'équipes ayant au moins 5 chercheurs

Requête 10 : Lister les thèmes de recherche des responsables d'équipe

Requête 11 : Lister les chercheurs travaillant sur le thème 'BDD' ou sur le thème 'INFRASTRUCTURE'

Requête 12 : Lister les chercheurs travaillant sur le thème 'INFRASTRUCTURE' mais pas sur le thème 'BDD'

Requête 13 : Lister les chercheurs travaillant sur le thème 'INFRASTRUCTURE' et sur le thème 'BDD'

Requête 14 : Lister les chercheurs travaillant dans la même équipe que le chercheur 'SERANDOUR'

Requête 15 : Lister les chercheurs travaillant sur au moins les mêmes thèmes que le chercheur 'SERANDOUR'

Requête 16 : Lister les chercheurs travaillant sur exactement les mêmes thèmes que le chercheur 'SERANDOUR'

Requête 17 : Lister les chercheurs travaillant sur au plus les mêmes thèmes que le chercheur 'SERANDOUR'

Requête 18 : Lister les équipes qui travaillent sur tous les thèmes de recherches

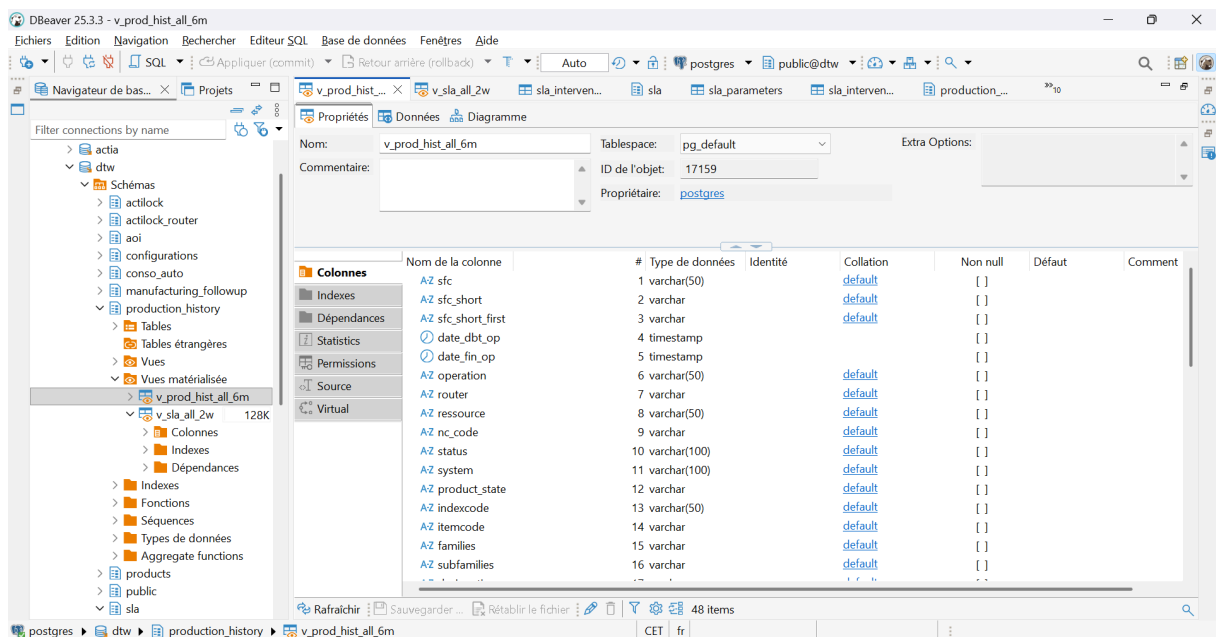
Requête 19 : Lister les chercheurs n'ayant pas publié

Requête 20 : Lister par chercheur, le nombre d'articles qu'ils ont publiés

Dans le cadre de ce TP, l'objectif était d'utiliser aucune intelligence artificielle et de pouvoir comprendre et expliquer chaque réponse et la technique utilisée. Dans ce principe là, je n'ai pas su répondre à toutes les questions notamment celles liées aux requêtes imbriquées pour lesquelles j'ai dû m'exercer.

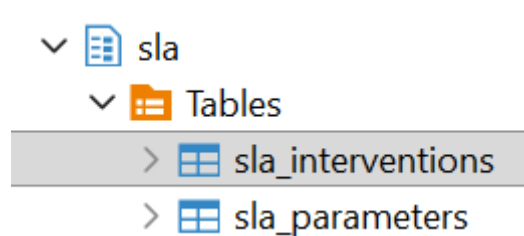
## 3 RÉALISATION DU PROJET

Ici nous avons la vue matérialisée d'une base de données basée sur 6 mois. Cette BDD référence tout ce qui correspond à un produit durant sa fabrication. Autrement dit, on y retrouve des informations telles que la ressource qui a pu fabriquer le produit, des codes d'erreur en cas de défaut ce qui nous sera très utile par la suite, ou encore le produit en lui-même défini par un ID.




Par la suite, j'ai pu me baser sur la même base de données mais axée sur les 2 dernières semaines afin de simplifier les requêtes.

### 3.1 Tables créées



Grâce à cette vue matérialisée j'ai pu créer ma propre table avec les données dont j'aurai besoin.

### 3.11 Table sla\_intervention :

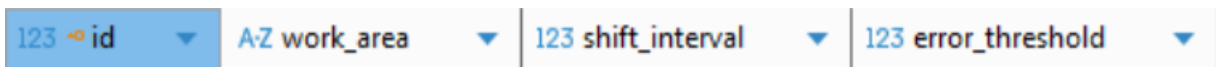


123 id	AZ work_area	AZ operation	AZ ressource	AZ nc_code	123 error_quantity	AZ comment	reset_date	AZ user
--------	--------------	--------------	--------------	------------	--------------------	------------	------------	---------

Détail de la table :

- work\_area : Représente les îlots. Chaque îlot représente une tolérance de nombre d'erreurs sur X heures glissantes.
- ressource : Représente la machine qui à réalisé le produit.
- operation : Représente l'opération réalisée par la ressource.
- nc\_code : Définit le code de l'erreur commise.
- error\_quantity : Calcule le nombre d'erreurs similaires .
- reset\_date : Place l'heure automatiquement dès la résolution de l'alerte .
- comment : Affiche le commentaire qui devra être écrit au préalable sur l'application et devra comprendre la résolution de l'alerte.
- user : Représente le nom de la personne qui à résolu le problème.

### 3.12 Table sla\_parameters :



123 id	AZ work_area	123 shift_interval	123 error_threshold
--------	--------------	--------------------	---------------------

Détail de la table :

- work\_area : Représente les îlots.
- shift\_interval : Défini l'intervalle d'heure.
- error\_threshold : Présente la limite d'erreurs similaires autorisée.

## 3.2 Préparation de l'application web

Dans le cadre de la réalisation visuelle de la page web, on m'a fourni la bande en haut de la page afin de respecter la charte de couleurs de l'entreprise.



## 3.3 Les 3 pages web :

Pour réaliser la première page, j'ai pu effectuer un **CRUD** (Create, Read, Update, Delete)

## 3.4 Page configuration :



### Configuration des alertes

#### Ajouter une configuration

Nom de l'îlot	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="Créer"/>
---------------	--------------------------------	--------------------------------	--------------------------------------

#### Configurations existantes

ID	îlot	Seuil défauts	Fenêtre (heures)	Actions
10	tgw5	5	9	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
8	ACU7	5	5	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
6	ECUB	6	3	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
2	LINKY	4	3	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
1	TGW3	1	8	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>

**Create** : Ici représenté par la création d'une nouvelle configuration directement sur l'application qui renverra directement les données dans la base .

**Read** : Chaque configuration en base de données est lue pour pouvoir être reportée dans le tableau.

### 3.41 Représentation du CR dans Visual Studio Code :

Create :

```
const create = (data, result) => {
  const query = {
    text: `
      INSERT INTO sla.sla_parameters (work_area, shift_interval,error_threshold )
      VALUES ('${data.work_area}','${data.shift_interval}','${data.error_threshold}')
      RETURNING *
    `;
  };
};
```

Read :

```
const getAll = (data, result) => {
  const query = {
    text: `SELECT * FROM sla.sla_parameters ORDER BY id DESC `;
  };
  db_config.pool
    .query(query)
    .then((results) => {
      result(null, results.rows);
    })
    .catch((e) => {
      console.error(e.stack);
      result(e, null);
    });
};
```

Lecture de tous les éléments.

```
const getOne = (data, result) => {
  const query = {
    text: `SELECT * FROM sla.sla_parameters where id = ${data.id}`;
  };
};
```

Lecture d'un élément précis.

Update : Modification sur le site qui fait effet en base.

## Configurations existantes

ID	Îlot	Seuil défauts	Fenêtre (heures)	Actions
10	<input type="text" value="tgw5"/>	<input type="text" value="5"/>	<input type="text" value="9"/>	<input type="button" value="Enregistrer"/> <input type="button" value="Annuler"/>
8	ACU7	5	5	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
6	ECUB	6	3	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
2	LINKY	4	3	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>
1	TGW3	1	8	<input type="button" value="Modifier"/> <input type="button" value="Supprimer"/>

Ici on peut choisir de tout modifier pour simplifier l'utilisation plutôt que de créer un nouvel îlot. On peut aussi annuler toute modification.

Delete : Supprime un îlot au choix.

### 3.42 Représentation du UD dans Visual Studio Code :

Update :

```
const saveEdit = (id, data, result) => {
  const query = {
    text: `
      update sla.sla_parameters
      SET work_area='${data.work_area}', shift_interval = ${data.shift_interval},
      error_threshold = ${data.error_threshold}
      where id = ${id} ;
    `
  };
};
```

Delete

```
const removeConfig = (data, result) => {
  console.log(data);
  const query = {
    text: `DELETE FROM sla.sla_parameters WHERE id = ${data.id}`,
  };
};
```

### 3.43 Résultat en base

Toutes ces informations sont enregistrées dans la table "sla\_parameters".


>  sla\_parameters

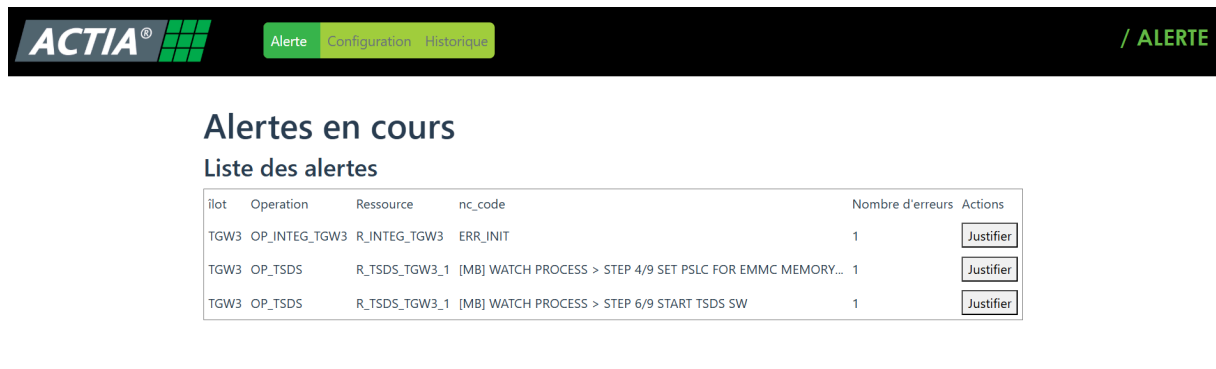
Table complète :

	123 id	A-Z work_area	123 shift_interval	123 error_threshold
1	8	ACU7	5	5
2	2	LINKY	3	4
3	6	ECUB	3	6
4	1	TGW3	8	1
5	10	tgw5	9	5

Ici on retrouve bien les éléments recherchés pour créer un îlot..

On peut remarquer que la création des îlots est réussie par la présence des mêmes données sur le site et en base ✓

## 3.5 Page alertes en cours



îlot	Operation	Ressource	nc_code	Nombre d'erreurs	Actions
TGW3	OP_INTEG_TGW3	R_INTEG_TGW3	ERR_INIT	1	<input type="button" value="Justifier"/>
TGW3	OP_TSDS	R_TSDS_TGW3_1	[MB] WATCH PROCESS > STEP 4/9 SET PSLC FOR EMMC MEMORY...	1	<input type="button" value="Justifier"/>
TGW3	OP_TSDS	R_TSDS_TGW3_1	[MB] WATCH PROCESS > STEP 6/9 START TSDS SW	1	<input type="button" value="Justifier"/>

Cette page nous permet d'afficher toutes les alertes en cours, référencées par la vue matérialisée "v\_sla\_all\_2w" du schéma "production\_history" afin de pouvoir les justifier à la résolution du problème.

```
const getAllAlert = (data, result) => {
  const query = {
    text: `select toto.work_area, toto.ressource, toto.operation, toto.nc_code, toto.nb_error
from (
  SELECT nc_code, ressource, work_area, operation,
  count (*) as nb_error
  FROM production_history.v_sla_all_2w
  group by nc_code, ressource, work_area, operation
)as toto
inner join sla.sla_parameters on (toto.work_area = sla.sla_parameters.work_area)
where toto.nb_error >= sla.sla_parameters.error_threshold
;`
  }
```

Pour afficher ces alertes, j'ai dû faire une requête imbriquée qui, dans un premier temps, compte le nombre d'erreurs dans la vue matérialisée "v\_sla\_all\_2w" pour un même "nc\_code", "ressource", "work\_area", "opération". Partie qu'on appelle ici "toto". Ensuite, on compare les données de "toto" à celles de la table "sla\_parameters" et on affiche seulement les erreurs qui dépassent le seuil donné par l'îlot, sur une même ressource, pour une même opération, et qui correspond à un même code d'erreur.

Suite de la page

The screenshot shows the ACTIA software interface with a modal window titled "Justifier l'alerte". The background displays a table of alerts and a table of error counts. The modal window contains the following fields:

- Lot: TGW3
- Opération: OP\_INTEG\_TGW3
- Ressource: R\_INTEG\_TGW3
- Utilisateur: Anthony
- Commentaire: Réparation du rack SS

Buttons for "Annuler" and "Confirmer" are located at the bottom of the modal.

Quand on appuie sur le bouton “justifier”, un Pop-up se lance avec les 5 champs dont on aura besoin plus tard dans la partie historique. Les 3 premiers champs sont pré remplis avec les informations de la ligne sélectionnée pour s’assurer de justifier la bonne alerte. Ensuite, on peut remplir le champ “Utilisateur” avec le nom de la personne qui a résolu l’alerte puis le “Commentaire” avec la raison du problème et comment il a pu être résolu.

Le bouton “Confirmer”, lui, sert à enregistrer les informations dans la base de données.

### 3.51 Résultat en base :



Toutes les justifications d’alerte sont renseignées dans la table “sla\_interventions”.

	123 *id	AZ work_area	AZ operation	AZ ressource	AZ nc_code	123 error_quantity	AZ comment	🕒 reset_date	AZ user
1	9	TGW3	OP_CADS	R_CADS_TGW3_1	[MB] WATCH PROCESS > STEP 4/10 FLASHING CAD	1	<wx<	2026-02-26 13:34:28.117 +0100	wcq<w
2	10	TGW3	OP_TF	R_TGW3_UFT_124		1	bnvb	2026-02-26 14:09:45.749 +0100	ng b
3	11	TGW3	OP_TF	R_TGW3_UFT_124		1	titi est tombé du bateau	2026-02-26 15:05:29.503 +0100	toto
4	12	TGW3	OP_INTEG_TGW3	R_INTEG_TGW3	ERR_CAP_VER_NOK	1	fzfll	2026-02-26 16:01:00.477 +0100	tata
5	13	TGW3	OP_INTEG_TGW3	R_INTEG_TGW3	ERR_INIT	1	Réparation de la rack	2026-02-26 16:17:45.290 +0100	Anthony

Ici on retrouve l’historique des justifications rempli par les éléments demandés ce qui signifie que l’objectif de cette page est atteint ✓

### 3.52 Problème rencontrés :

Dans le cas actuel, le choix d’une vue matérialisée a été un choix peu judicieux car le principe de celle-ci est rapide, mais doit être rafraîchi pour être à jour ce qui pose problème dans notre cas. On a donc dû migrer la vue matérialisée sur une vue normale qui, elle, est parfois lente mais se met à jour automatiquement.

Malheureusement ce problème est survenu dans les derniers jours et je n’ai pas eu le temps de le faire moi même donc pour tester que mon système fonctionne bien, je me suis débrouillé en utilisant la commande “REFRESH MATERIALIZED VIEW” afin de pouvoir recharger la base et voir si elle est bien mise à jour.

## 3.6 Page historique des Justifications



### Historique des Justification

#### Liste des résolutions d'alerte

ID	Îlot	Ressource	Operation	nc_code	Nb d'erreurs	Commentaire	Date reset	Actions
13	TGW3	OP_INTEG_TGW3	R_INTEG_TGW3	ERR_INIT	1	Réparation de la rack	2026-02-26T15:17:45.290Z	<input type="button" value="Modifier"/>
12	TGW3	OP_INTEG_TGW3	R_INTEG_TGW3	ERR_CAP_VER_NOK	1	fzf	2026-02-26T15:01:00.477Z	<input type="button" value="Modifier"/>
11	TGW3	OP_TF	R_TGW3_UFT_124		1	titi est tombé du bateau	2026-02-26T14:05:29.503Z	<input type="button" value="Modifier"/>
10	TGW3	OP_TF	R_TGW3_UFT_124		1	bnvb	2026-02-26T13:09:45.749Z	<input type="button" value="Modifier"/>
9	TGW3	OP_CADS	R_CADS_TGW3_1	[MB] WATCH PROCESS > STEP 4/10 FLASHING CADS SW...	1	<wx<	2026-02-26T12:34:28.117Z	<input type="button" value="Modifier"/>

Ici, nous avons un tableau répertoriant toutes les alertes justifiées par la page "alerte" récupérés par la base de données.

Sur la page, on veut seulement afficher :

- ID
- Îlot
- Ressource
- date de résolution
- Code d'erreur
- Nombre d'erreurs
- Commentaire

Pas besoin d'afficher la personne qui a résolu l'alerte pour des questions de confidentialité.

Ici, le seul élément modifiable devait être le commentaire en cas de simple erreurs. Le reste des données ne devaient pas être modifiables pour des raisons d'intégrité mais par contrainte de temps, je n'ai donc pas pu finir cette partie là de mon côté.

## **CONCLUSION**

Ce stage au sein d'ACTIA m'a permis de découvrir un environnement industriel exigeant, structuré et orienté vers l'innovation technologique. Après une phase d'intégration qui m'a familiarisé avec l'organisation du site, les outils professionnels et les méthodes de travail, j'ai pu m'investir pleinement dans un projet concret répondant à un besoin réel de l'entreprise : améliorer la gestion et le suivi des alertes qualité liées aux défauts de production.

La réalisation de ce projet m'a conduit à mobiliser et approfondir mes compétences en base de données, en développement web et en logique applicative. J'ai conçu deux tables dédiées, développé une application web complète intégrée à l'outil Manufacturing Follow Up, et mis en place un système fiable permettant d'afficher les alertes, de les justifier et d'en conserver l'historique. Ce travail m'a également permis de comprendre les enjeux liés aux vues matérialisées, à la performance des requêtes SQL et à la cohérence des données dans un contexte industriel.

Au-delà des aspects techniques, ce stage m'a appris à travailler en équipe, à présenter mes idées, à m'adapter aux contraintes du terrain et à améliorer mes solutions grâce aux retours de mes tuteurs. Malgré certaines difficultés, notamment liées à la gestion des vues matérialisées ou à la finalisation de certaines fonctionnalités, ce projet a été une expérience formatrice et enrichissante qui a renforcé ma capacité à analyser un besoin, proposer une solution et la mettre en œuvre de manière autonome.

Ce stage représente ainsi une étape importante dans mon parcours, tant sur le plan professionnel que personnel.

## **REMERCIEMENTS**

Merci à toutes l'équipe de Actia de m'avoir accompagné tout au long de ce stage. Particulièrement au service informatique qui m'a aidé à approfondir mes connaissances et m'a formé à de nouveaux outils.

**Anthony AFONSO**