

**Loup Sonneville,**

**Jean-Rémy Dion,**

**Nelson Graveau**

***Rapport***

***Personnel***

***partie BDD***

**Tables des matières**

[I. Introduction 3](#_Toc132033364)

[II. Planification 6](#_Toc132033365)

[A. GitHub 6](#_Toc132033366)

[B. Discord 6](#_Toc132033367)

[C. GANTT 7](#_Toc132033368)

[III. Mise en œuvre 7](#_Toc132033369)

[D. Répartition des taches 8](#_Toc132033370)

[E. Partie IHM 11](#_Toc132033371)

[F. Partie Interface Web 11](#_Toc132033372)

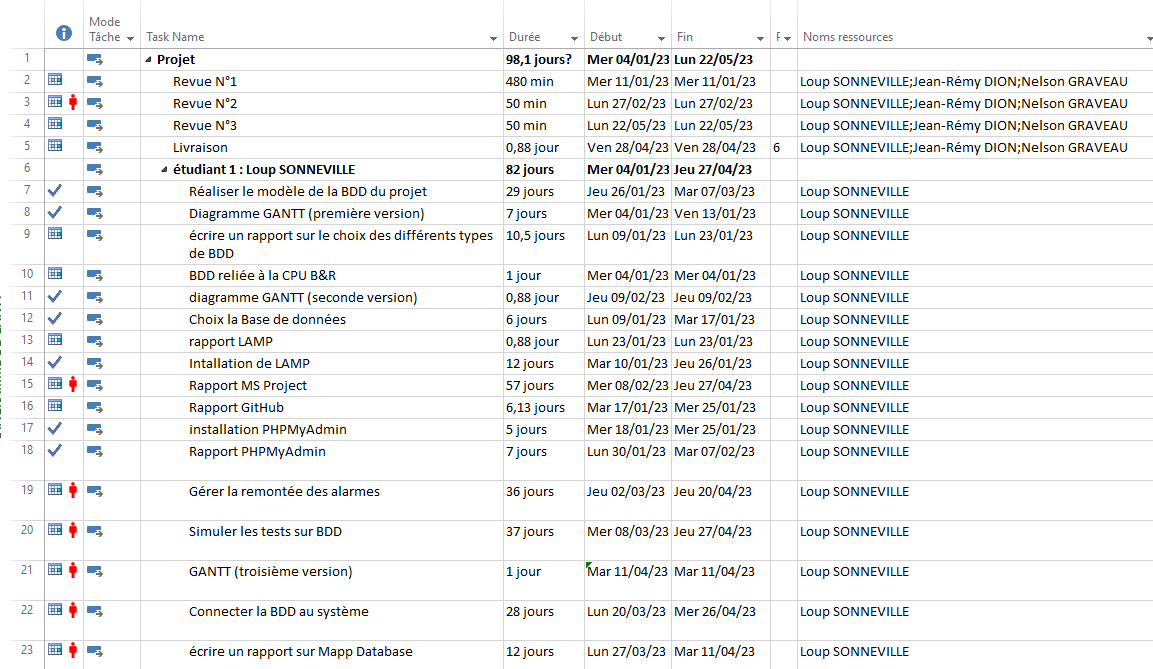
[G. Partie BDD 11](#_Toc132033373)

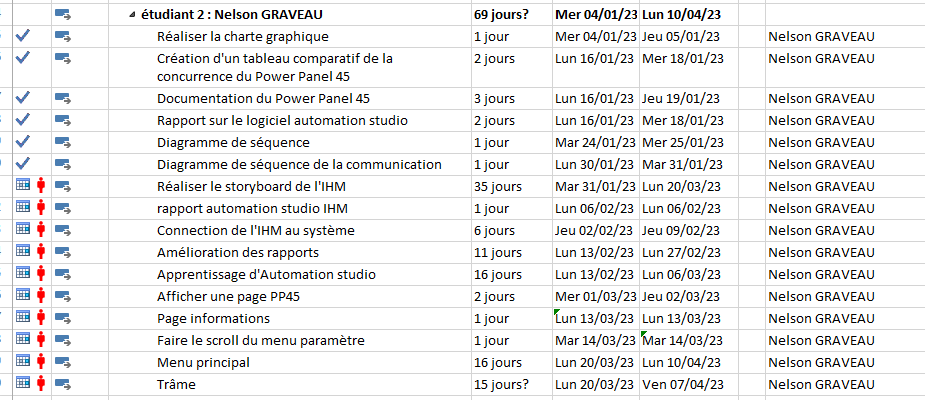
[IV. Automation Studio 11](#_Toc132033374)

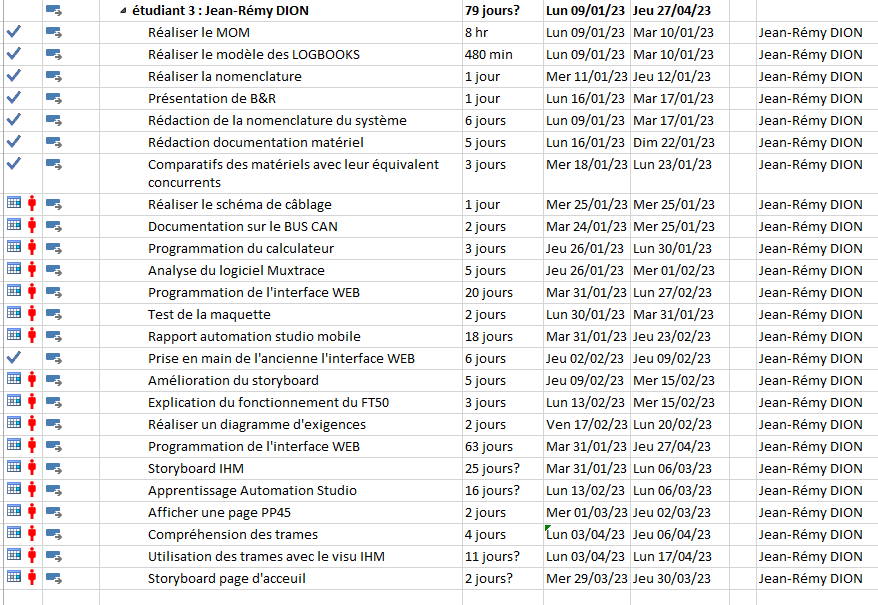
[V. Conclusion 11](#_Toc132033375)

# Gantt

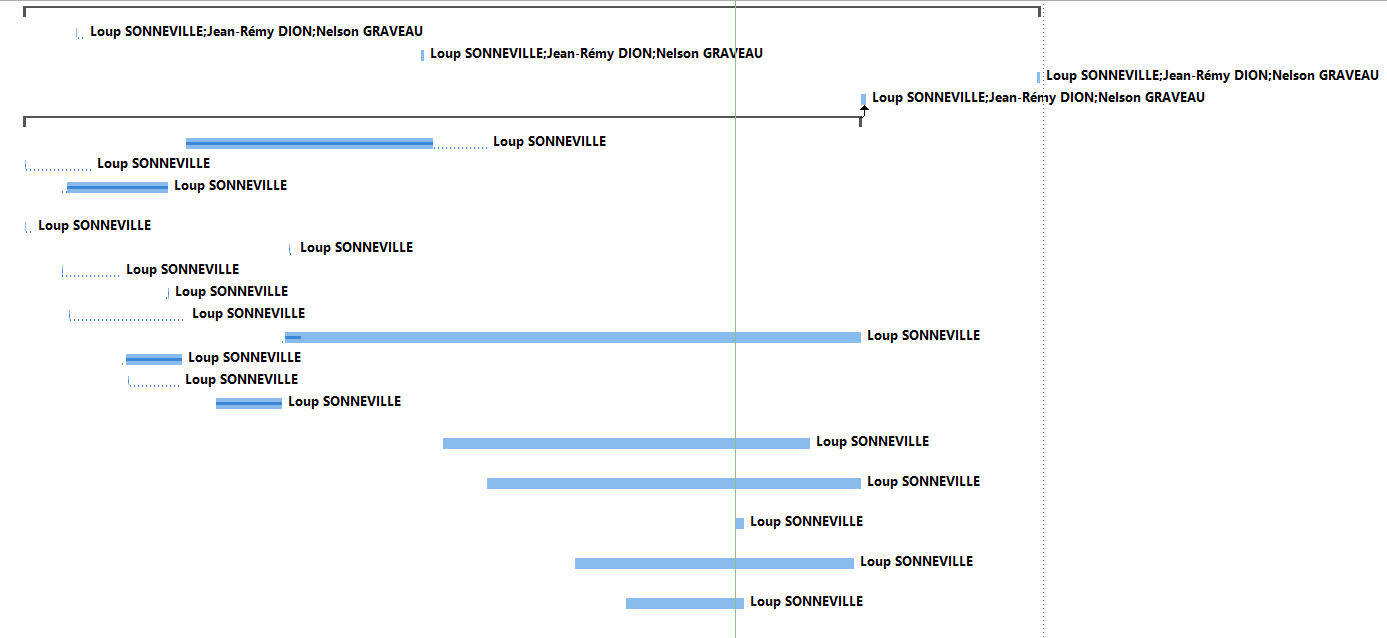
Afin de planifier nos tâches, nous nous sommes tournés vers Microsoft Project, une fois dedans nous avons rajouté les tâches à faire une à une tout en précisant les étudiants assignés à la tâche ainsi que les estimations du temps qu’elles prendront



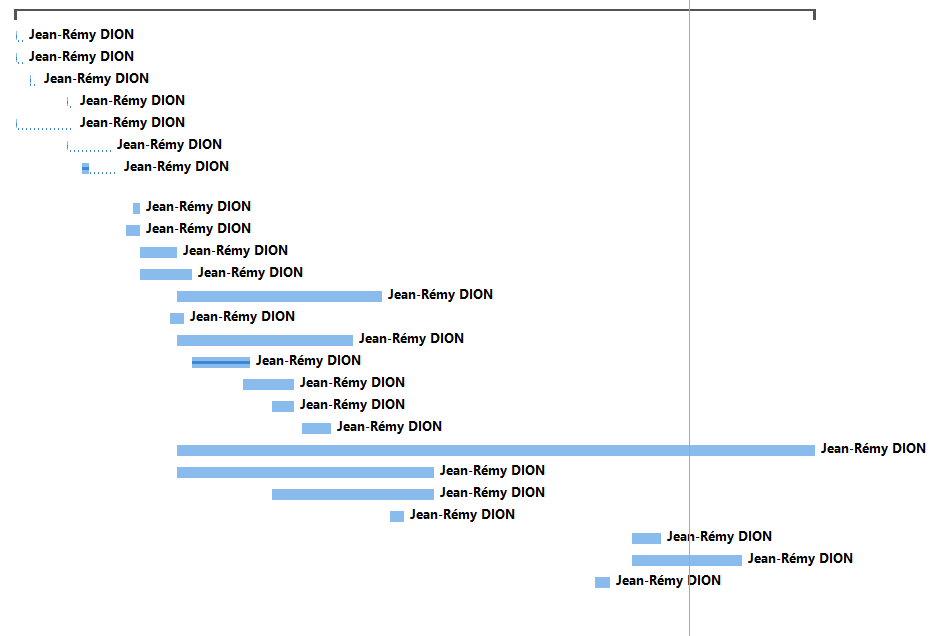
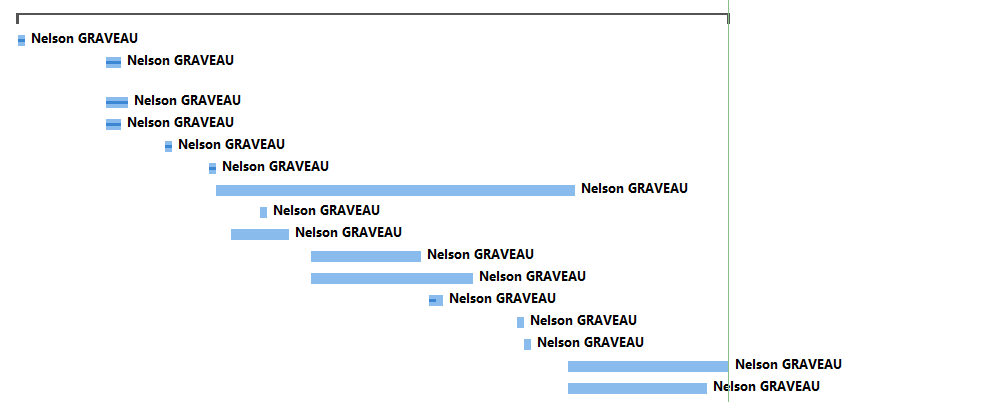




Dès qu’une tâche est entrée, elle est également affichée dans une autre partie de Project qui est ce que l’on pourrait qualifier de calendrier horizontal



Celà permet une meilleure visualisation des tâches à effectuer



# Justification des choix

Afin que l’interface WEB puisse stocker des données comme l’historique des chariots en service ou que les alarmes remontent, il était nécessaire de créer une base de données

En raison de divers préférences personnelles et pour des questions de sécurité ainsi que de performances, nous avons choisi de la faire sur LINUX, plus précisément sur UBUNTU

Nous avons donc pris une machine et installé UBUNTU dessus (voir l’annexe sur UBUNTU pour plus de détail)

L’étape suivante consistait aux choix de quelle pile de programme choisir, ce qui engagea une étude des qualités et inconvénients des piles

## Tableau comparatifs des SGBDR

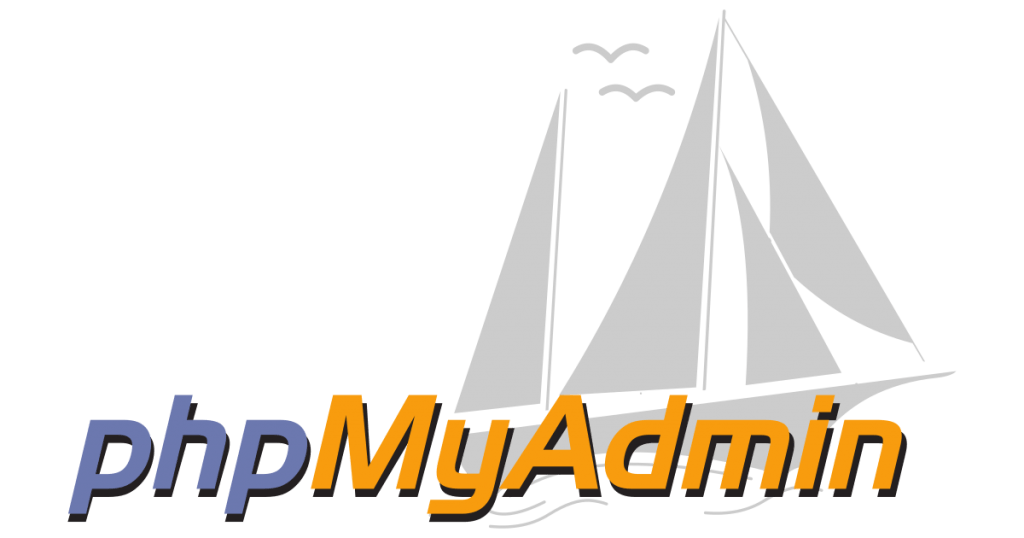
|  |  |  |
| --- | --- | --- |
|  | -La base de données la plus utilisée au monde  -Facile à utiliser  -De bonnes performances  -Plusieurs fonctionnalités pour sécuriser ses données  -Open-source | -Difficilement scalable, les performances du système se détériorent à partir d’un certain volume de données |
|  | -plus rapide que MySQL  -supporte jusqu’à 200 000 connexions  -compatibilité  -en constante évolution (mises à jour fréquentes) | -Système de mise en cache peu performant  -Mauvaise gestion du chargement  -Manque de fonctionnalités avancées  -Supporte mal un haut volume de données |
|  | -Open-source  -Facile à utiliser  -Possède un type de données défini par l’utilisateur  -Une grande communauté | -L’un des moins bien noté pour ce qui est des performances  -La réplication est complexe  -L’installation est très peu intuitive |
|  | -Bonne sécurité des données  -Facile à installer et à configurer ·  -De nombreux outils pour gérer l’ensemble des tâches en entreprise | -payant  -incompatible avec les logiciels ne provenant pas de Microsoft |
|  | -Bonne capacité de sauvegarde et de récupération des données  -Régulièrement mis à jour  -Grande portabilité  -Gère facilement plusieurs bases de données au sein d’une même transaction  -La base de données la plus populaire selon le classement DB-Engines | -payant  -Difficile à maîtriser |

## Tableau comparatif des piles

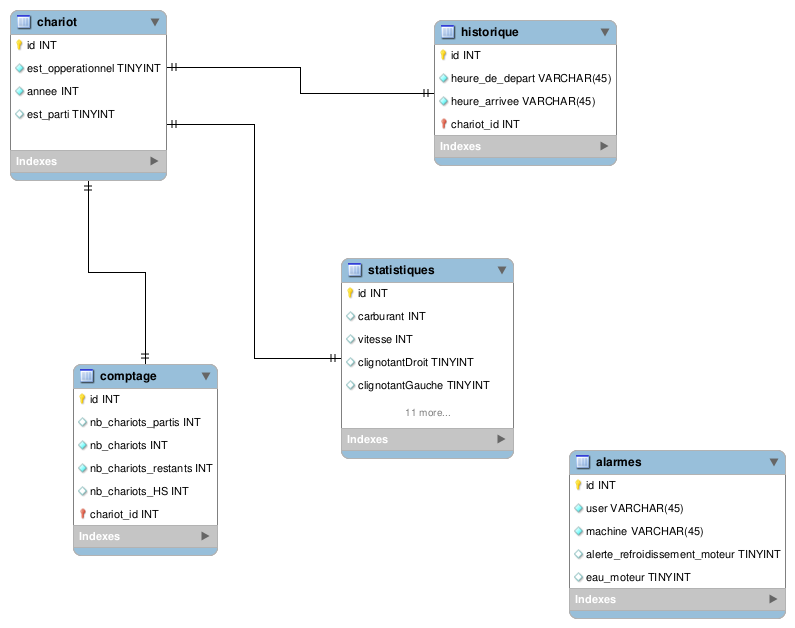
|  |  |  |
| --- | --- | --- |
|  | -simple à configurer | -difficile d’installation |
|  | -réputé pour sa sécurisé et sa flexibilité | -sensible aux lourdes charges |
|  | -crossplatforme (fonctionne aussi bien sur Windows, que sur mac ou linux) | -gourmand en ressources |
|  | -Fonctionne sur Windows comme sur MAC | -Payant, existe en version gratuite mais cette dernière est très insuffisante |

La pile choisi fut LAMP, notre prochain besoin était une interface graphique, deux choix s’offrait : PHPMyAdmin ou Adminer ?

N’ayant pas réussi à faire fonctionner Adminer, nous avons choisi de nous rabattre sur PHPMyAdmin que nous connaissons bien mieux.

Ce qu’il nous fallait par la suite était de quoi réaliser le modèle de la BDD de manière à pouvoir la visualiser de façon visuelle.

Ne connaissant pas d’autres alternatives, nous l’avons fait sur SQL Workbench



Nous avons exporté ce modèle en script SQL que nous avons importé dans la BDD via PHPMyAdmin.

La dernière étape est de relier BDD et CPU, les équipements étant de la gamme BR, nous avons choisi de faire la liaison via MappDatabase sur automation Studio.