

**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)

# Introduction

Notre objectif est de créer un banc de test pour les tableaux de bord numériques des chariots élévateurs du groupe MANITOU. Nous devrons finaliser l'interface tableau de bord en ajoutant un écran de 5,7 pouces et développer une interface graphique sur dalle tactile pour le banc de test. Le banc sera relié à une base de données pour suivre les tests et générer des statistiques. Les informations seront transmises par les liaisons Ethernet et bus CAN norme J1939 pour le temps réel.

La société MANITOU souhaite améliorer ses systèmes de commande en pilotant ses chariots élévateurs à l'aide de nouveaux IHM et en effectuant un meilleur suivi des ordinateurs de bord de ces chariots avec un banc de test. Pour cela, nous utiliserons les calculateurs de la gamme X90 et les écrans associés de la société B&R pour simuler le comportement du chariot élévateur et supporter le banc de tests.

Le projet nécessite la création d'un système permettant de superviser et contrôler le bon fonctionnement d'un chariot élévateur via une interface web. Les besoins identifiés sont les suivants :

* Administrer les tests du système : le responsable technique doit pouvoir effectuer des tests pour vérifier le bon fonctionnement de l'interface homme-machine (IHM) du tableau de bord, en utilisant une interface web sur son poste de contrôle et un émulateur de la partie opérative du chariot.
* Superviser les données du système : le responsable technique doit pouvoir consulter les résultats des tests sur une interface web dédiée, qui remonte les différentes informations du chariot élévateur (vitesse, carburant, etc.…) du jour et des historiques grâce à une base de données.
* Paramétrer le système : le responsable technique doit être capable de configurer les entrées/sorties de l'émulateur avec l'IHM du chariot élévateur en utilisant l'interface web, afin de mettre en relation l'émulateur du chariot avec l'IHM du chariot élévateur.
* Calibrer les pupitres : le responsable technique doit calibrer les configurations IHM en fonction de données telles que le type du chariot et la langue. L'opérateur doit vérifier les indicateurs visuels de l'IHM avec les conditions initiales requises pour un bon fonctionnement du chariot élévateur, déterminer une action, et paramétrer les consignes nécessaires à ces actions pour qu'un cycle de fonctionnement du chariot se déroule correctement.
* Démarrer un cycle : le responsable technique doit être en mesure de mettre en route le système grâce à l'interface de l'émulateur du chariot et de visualiser à l'aide d'un pupitre IHM la mise en route du chariot. Le système doit passer d'un état "repos" à un état "actif", et l'IHM doit renseigner l'opérateur de la situation par un message écrit et un suivi dynamique sur le tableau de bord de l'évolution des consignes en fonction des actions du chariot. Le responsable doit être en mesure d'interrompre ou de mettre fin à un processus si nécessaire.
* Superviser le tableau de bord : l'opérateur doit pouvoir suivre l'évolution des consignes en fonction des actions du chariot sur son IHM tableau de bord (temps, température, feux, clignotants, gyrophare, claxone, etc..).

Il y aura deux acteurs humains directement impliqués dans l'utilisation du système : l'opérateur, qui sera en relation directe avec l'IHM du tableau de bord du chariot élévateur et chargé de démarrer et superviser le cycle des actions du chariot, et le responsable technique, qui sera chargé d'administrer le bon fonctionnement de l'IHM du tableau de bord du chariot élévateur, gérer les différentes recettes de test du système, paramétrer, calibrer la partie opérative et visualiser l'état de la machine en temps réel. Les deux acteurs pourront interrompre un cycle si visuellement, ils constatent un dysfonctionnement.

**Diagramme d’exigence :**

A picture containing diagram

Description automatically generated

**Diagramme des cas d’utilisation :**

Diagram

Description automatically generated

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

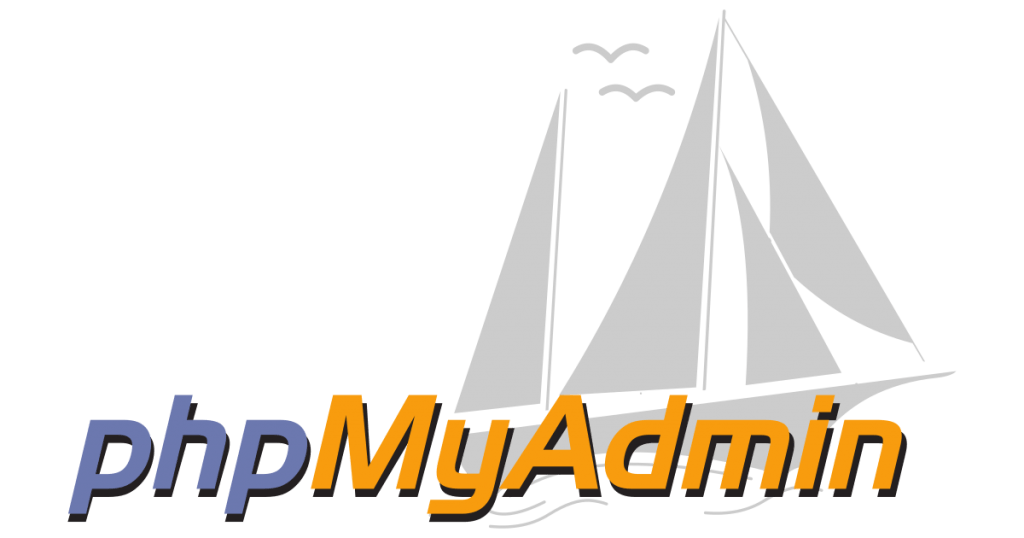
|  |  |  |
| --- | --- | --- |
|  | -La base de données la plus utilisée au monde  -Facile à utiliser  -De bonnes performances  -Plusieurs fonctionnalités pour sécuriser ses données  -Open-source | -Difficilement scalable, les performances du système se détériorent à partir d’un certain volume de données |
|  | -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

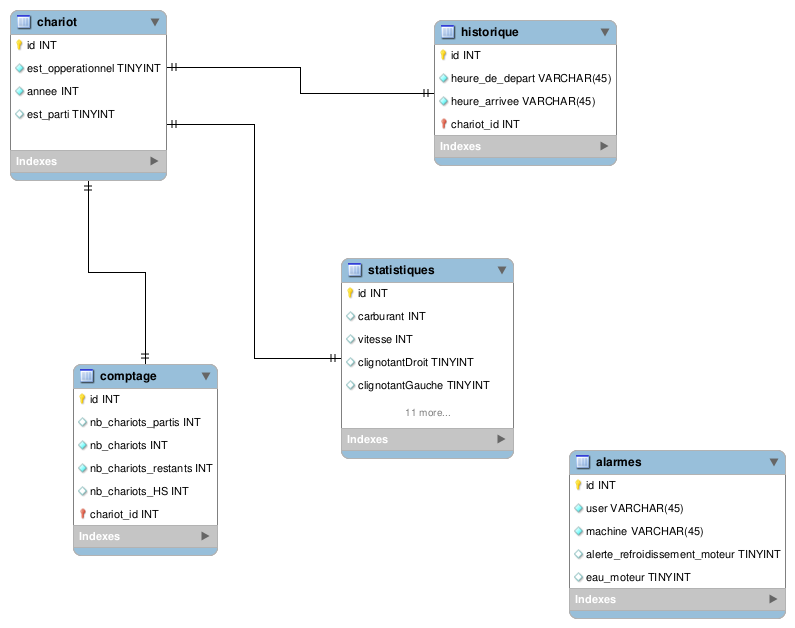
|  |  |  |
| --- | --- | --- |
|  | -simple à configurer | -difficile d’installation |
|  | -réputé pour sa sécurisé et sa flexibilité | -sensible aux lourdes charges |
|  | -crossplatforme (fonctionne aussi bien sur Windows, que sur mac ou linux) | -gourmand en ressources |
|  | -Fonctionne sur Windows 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.