

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE DE CONSTRUCTION
DEPARTEMENT ELECTROMECHANIQUE

Projet de fin d'étude pour l'obtention du diplôme de Master académique.

Domaine : **Sciences et Technologies**
Filière : **Electromécanique**
Spécialité : **Maintenance Industrielle**

Thème

**Implémentation d'un logiciel GMAO dans une
entreprise de travaux publics**

Présenté Par :

Mr : Nacim Mansouri

Mr : Salem Annache

Mémoire soutenu publiquement le 30 / 09 / 2024 devant le jury composé de :

Mr. BOURAS Rachid

UMMTO, Président

Mr. SAMAH Boubeker

UMMTO, Encadreur

Mr. MOUHELBI Mouloud

UMMTO, Examineur

Promotion 2023/2024.

Remerciements

*Nous exprimons tout d'abord, notre gratitude à
Dieu le tout puissant de nous avoir donné la
Patience, la volonté et l'énergie pour réaliser ce
Travail*

*Nous remercions profondément notre promoteur Monsieur
Samah Boubkeur, pour avoir accepté de diriger notre travail
et MR AHMED ZEROUK pour nous avoir guidés tout au
long de ce mémoire pour ses conseils, ses critiques et ses
attitudes exemplaires tout au long de la réalisation de ce
travail de recherche.*

*Nos vifs remerciements vont également aux membres du jury
pour l'intérêt qu'ils ont porté à notre recherche en acceptant
d'examiner notre travail et de l'enrichir par leurs propositions*

*Enfin, nous tenons à remercier tous les membres de notre
famille pour leur soutien et leur appui moral. Nous leur
serons reconnaissants pour le reste de notre vie.*

*Nous tenons à exprimer notre profonde reconnaissance et nos
sincères remerciements à toutes les personnes qui ont
contribué de près ou de loin à l'avancement de notre travail de
recherche.*

Dédicaces

Je dédie le rendement de ces années d'études à mes chers parents qui m'ont toujours soutenu tout au long de ma vie, qui m'ont orienté dans le bon chemin et qui' ont tout fait pour que je réussisse, que Dieu vous bénisse et vous garde.

A toute ma famille (spécialement mon grand-père que dieu l'accueil dans son vaste paradis)

A mes chères frères et sœurs

A mon binôme Salem

A tous mes amis(es)

A toute la promotion 2023/2024

A toute personnes ayant contribué de près ou de loin à l'accomplissement de ce travail.

Nacim

Dédicaces

Je dédie ce mémoire à mes chers parents pour leur soutien indéfectible et leur amour inconditionnel tout au long de ces longues années d'études, leur confiance en moi a été une source de motivation pour atteindre mes objectifs et réussir cette étape importante de ma vie. Que dieu les protège.

À mes chers Sœurs et mes frères

Mon grand-père et ma grand-mère

À toute ma famille sans exception et mes proches

A mon binôme Nacim

Mes chers amis Azouaou.Nassim

À toutes autres personnes qui ont été une source inépuisable de soutien, d'amour et d'encouragement tout au long de mon parcours. Merci beaucoup. Vous êtes mes rayons de soleil éclatant.

Salem

Liste des abréviations

SGBD	SYSTEME GESTION DE BASE DE DONNEES.
ID	IDENTIFIANT
OI	ORDRE D'INTERVENTION DEMANDE D'INTERVENTION
DI	Demande d'intervention
RH	RESSOURCE HUMAIN
GMAO	gestion de maintenance assistée par ordinateur
MRP	MATERIEL REQUIREMENT PLANING
SGDT	SYSTEME DE GESTION DES DONNEES TECHNIQUES
ERP	ENTREPRISE RESSOURCE PLANING
SQLite	STRUCTURED QUERY LANGUAGE
MTBF	MEAN TIME BETWEEN FAILURES (TEMPS MOYEN ENTRE PANNES)
MTTR	mean time to repair (temps moyen de réparation)
UI	Utilisateur interface

Liste des figures & Tableaux

FIGURE 1.1 TYPE DE MAINTENANCE.....	8
FIGURE 1.2 ACTIONS DE LA MAINTENANCE	10
FIGURE 2.1 GESTION ITERATIVE DE LA MAINTENANCE.....	20
FIGURE 3.1 BESOINS FONCTIONNELS RELIE AU SYSTEME.....	29
FIGURE 3.2 SCHEMA DE BASE DE DONNEES.....	31
FIGURE 3.3 INTERFACE UTILISATEUR	32
FIGURE 3.4 LOGO DE KOTLIN	40
FIGURE 4.1 FENETRE D’AUTHENTIFICATION	44
FIGURE 4.2 PAGE D’ACCUEIL	45
FIGURE4.3 COURBE GRAPHIQUE SUR LES STATISTIQUES DU STOCK	45
FIGURE 4.4 DIAGRAMME CIRCULAIRE DE L’ETAT DES FACTURES DES PIECES	46
FIGURE 4.5 DIAGRAMME CIRCULAIRE SUR L’ETAT DES EQUIPEMENTS.....	46
FIGURE 4.6 HISTOGRAMME DES INTERVENTIONS SUR LES EQUIPEMENTS	47
FIGURE4.7 FENETRE DES EQUIPEMENTS	47
FIGURE 4.8 FENETRE D’AJOUT D’UN EQUIPEMENT	48
FIGURE 4.9 FENETRE DE DETAILS SUR LES EQUIPEMENTS.....	49
FIGURE 4.10 FENETRE D’ASSURANCE.....	50
FIGURE 4.11 FACTURE D’ACHAT.....	51
FIGURE 4.12 CONTROLE TECHNIQUE	52
FIGURE4.13 STOCK DES PIECES	53
FIGURE 4.14 CONTENU DE STOCK	53
FIGURE 4.15 HISTORIQUE DU STOCK	54
FIGURE4.16 FENETRE D’AJOUT D’UNE PIECE.....	54
FIGURE 4.17 FENETRE DE DI	56
FIGURE 4.17 FENETRE D’AJOUT D’UNE DI.....	57
FIGURE 4.18 DETAILS DE DI	58
FIGURE 4.19 INFORMATIONS SUR DI	59
FIGURE 4.20 CREATION D’UN OI	60
FIGURE 4.21 FENETRE OI	61
FIGURE 4.22 FENETRE RH	62
FIGURE 4.23 FENETRE D’AJOUT D’UN EMPLOYE	63
FIGURE 4.24 FENETRE DE NOTIFICATIONS	64

LISTE DES TABLEAUX

TABEAU 1.1 TABLEAU DES NIVEAUX DE MAINTENANCE.....	7
TABEAU 2.1 : LES FONCTIONS DE LA GMAO SUIVANT SES UTILISATEURS	25
TABEAU 3.1 RELATION ENTRE LES BESOINS.....	30
TABEAU3.2 TABLE D’EQUIPEMENT DANS LA BASE DE DONNEES	ERREUR ! SIGNET NON DEFINI.
TABEAU3.3 TABLE DES TECHNICIENS DANS LA BASE DE DONNEES.....	38

Table des matières

<i>Remerciements</i>	I
<i>Dédicaces</i>	II
Liste des abréviations.....	IV
Liste des figures & Tableaux.....	V
Liste des tableaux.....	V
1-Introduction :	5
2-Importance de la maintenance dans l'entreprise :	5
3-Définition de la maintenance:	5
4- Objectif de la maintenance	5
4.1 Objectifs financiers	6
4.2 Objectifs opérationnels	6
5-Les cinq niveaux de la maintenance	6
6-Types de maintenance :	8
6.1 Maintenance corrective :	8
a) Maintenance palliative :	8
b) Maintenance curative :	8
6.2 Maintenance préventive :	9
a) Maintenance systématique	9
b) Maintenance conditionnelle	9
c) Maintenance prédictive :	9
7-Les formes d'actions d'un service maintenance	10
8-Domaine d'utilisation de la maintenance :	10
8.1 Industrie	11
8.2 Secteur de l'énergie	11
8.3 Transport et logistique	11
8.4 Technologies de l'information :	11
10-Conclusion :	12
CHAPITRE 2 : Généralité sur la gestion de maintenance assistée par ordinateur GMAO	13
1-Introduction :	14
3-Définition d'un logiciel de GMAO	15
3.1 Fonctionnalités d'un logiciel de GMAO	16
3.2 Nécessité de l'utilisation d'un logiciel de GMAO	16
3.3 Fonctions principales d'un logiciel de GMAO	17
3.4 Les différents modules fonctionnels d'une GMAO	18

3.4.2 Module "gestion du suivi opérationnel des équipements"	18
3.4.3 Module "gestion des interventions"	18
3.4.4 Module "gestion du préventif"	18
3.4.5 Module "gestion des stocks"	18
3.4.6 Module "gestion des approvisionnements et des achats"	18
3.4.7 Module "analyse des défaillances"	18
3.4.8 Module "budget et suivi des dépenses"	19
3.4.9 Module "gestion des ressources humaines"	19
3.4.10 Module "tableaux de bord et statistiques"	19
4-La conduite d'un projet GMAO.....	19
4.1 Les étapes d'un projet GMAO.....	19
4.1.1 Étude de faisabilité	19
4.1.2 Rédaction du cahier des charges de consultation	19
4.1.3 Choix de l'outil GMAO et de ses modules nécessaires	19
4.1.4 Implantation, plan de formation et démarrage	19
4.2 Le modèle itératif de la gestion.....	20
4.3 Application à la gestion du service maintenance.....	20
5.1 Les conditions de réussite d'un GMAO :	21
5.1.1 La convivialité de l'outil informatique :	21
5.1.2. Volonté de réussite de consensus :	21
5.1.3. Facteur temps :	21
6- comparaison des quelques logiciels GMAO :	21
7-Les utilisateurs de GMAO :	24
8-Conclusion :	25
Chapitre 3 :	26
Conception et architecture du système	26
1-Introduction :	27
2-les besoins fonctionnels :	27
2.1 Gestion des équipements	27
2.2 Gestion de ressources humaines	27
2.3Gestion de stock	28
2.4Gestion d'historique et contenu de stock	28
2.5Gestion des interventions	28
2.6 Gestion des moyens économiques	28
2.7 Gestion des demandes d'interventions.....	29
2.8 Gestion des notifications	29

2.9 Gestion de tableau de bord	29
4-Architecture logicielle :	30
4.1 Base de données	30
4.3 Conception front end :	32
5.1-Model Layer :	33
5.2-ViewModel Layer :	33
5.3-View Layer :	34
6- Fonctionnement Général :	34
7- Avantage de MVVM :	35
8- Relation MVVM avec le Frontend:	35
9- SQLite comme backend :	35
9.1Simplicité et légèreté.....	36
9.2 Architecture de SQLite dans un logiciel GMAO.....	36
a) Base de données locale :	36
b) Requêtes SQL dans l'application GMAO :	36
c) Synchronisation et gestion des données :	37
9.3 Tables et modèles dans une GMAO	37
9.4 Fonctionnement global de l'architecture SQLite :	38
9.5 Fonctionnement de MVVM avec SQLite.....	39
a) Model (Modèle)	39
b) ViewModel (Modèle de Vue)	39
c)Interaction avec SQLite via le modèle.	39
d) View (Vue) :	39
10-Technologies utilisée :	40
10.1 Kotlin	40
10.3 L'utilité SQLite dans la base de données et le backend :	41
10.4 Avantages de SQLite :	42
11- Conclusion :	42
Chapitre 4 :	43
Réalisation de l'application GMAO.....	43
1-Introduction :	44
2- Fenêtre d'authentification :	44
5 Fenêtre « équipements » :	47
5.1 Fenêtre d'ajout d'un équipement :	48
5.2 Détails sur les équipements :	49
5.2.1 Fiche d'assurance d'un équipement :	50

5.2.2	Facture d'achat :	51
5.2.3	Fiche de contrôle technique :	52
6-	Fenêtre de stock :	53
6.1	Contenu de stock :	53
6.2	Historique du stock :	54
6.3	Informations sur la pièce :	54
7-	Fenêtre de demande d'intervention «DI» :	55
7.1	Ajout d'une demande d'intervention :	57
7.4	Création d'un OI :	60
8-	Fenêtre d'ordres d'intervention «OI» :	61
9-	Fenêtre «RH» :	62
9.1	Fenêtre d'ajout d'un employeur :	63
10-	Notifications :	64
	Conclusion Général	65
	Conclusion générale	66
	Annexe :	67
	Bibliographie	Erreur ! Signet non défini.
	Résumé:	Erreur ! Signet non défini.

Introduction général

De nos jours, le formidable développement de l'informatique et surtout de la micro-informatique avec des systèmes d'exploitation de plus en plus conviviaux permettant d'avoir une interface homme-machine d'une simplicité extrême, il n'est plus besoin d'avoir de solides connaissances en informatique pour utiliser une GMAO.

En réalité, l'approche à privilégier est que la GMAO doit être un outil performant, conçu pour permettre aux techniciens de maintenance de gagner du temps et d'optimiser leurs activités quotidiennes. Grâce à un accès facile aux historiques d'intervention et aux analyses pertinentes, cet outil favorise une amélioration continue des processus de maintenance.

L'informatique en tant que technique de gestion et d'organisation, est devenue un outil indispensable pour toute organisation qui gère un volume important d'information. [01]

Dans notre travail nous allons faire la conception et la réalisation d'un logiciel de gestion de la maintenance assistée par ordinateur, dont le but principal est d'automatiser le traitement et la présentation des informations nécessaires, pour mieux gérer la gestion des travaux.

Pour mener à bien ce projet, nous avons structuré ce mémoire en quatre chapitres distincts :

Chapitre 1 : Nous aborderons les concepts généraux de la maintenance, en expliquant ses principes fondamentaux et en présentant les différents types de maintenance (préventive, corrective, conditionnelle, etc.).

Chapitre 2 : Nous retracerons brièvement l'histoire de la maintenance industrielle, puis nous définirons la GMAO en explorant ses différents modèles et logiciels.

Chapitre 3 : Ce chapitre sera consacré à la description de l'architecture logicielle de la GMAO, ainsi qu'aux technologies employées pour la réalisation des diverses tâches de maintenance.

Chapitre 4 : Enfin, nous présenterons le logiciel développé dans le cadre de ce projet, illustré par des captures d'écran, et expliquerons son fonctionnement spécifique.

La gestion de maintenance assistée par ordinateur (GMAO) est un ensemble d'outils et de méthodes qui vise à optimiser la gestion des opérations de maintenance.

En somme, un progiciel de GMAO est un outil essentiel pour toute organisation souhaitant améliorer sa stratégie de maintenance, garantir la disponibilité de ses équipements et optimiser ses ressources.

Chapitre 1 :
Introduction à la maintenance

1-Introduction :

La maintenance est l'ensemble des actions techniques et organisationnelles mises en œuvre pour maintenir ou rétablir des équipements industriels dans un état fonctionnel optimal. Elle joue un rôle clé dans l'efficacité des entreprises, en assurant la disponibilité, la fiabilité et la durabilité des machines et des infrastructures. En d'autres termes, la maintenance industrielle vise à prévenir les pannes, minimiser les interruptions de production et prolonger la durée de vie des équipements.

2-Importance de la maintenance dans l'entreprise :

Elle garantit la sûreté des systèmes et la sécurité des employés en anticipant et corrigeant les dysfonctionnements potentiels des équipements, susceptibles de causer des blessures aux utilisateurs. Sur le plan économique, la maintenance est également une fonction essentielle pour les entreprises, contribuant à la continuité des opérations et à la réduction des coûts liés aux pannes.

3-Définition de la maintenance:

Le maintien des équipements de production est un enjeu clé pour la productivité des entreprises aussi bien que pour la qualité des produits. C'est un défi industriel impliquant la remise en cause des structures figées actuelles et la promotion de méthodes adaptées à la nature nouvelle des matériels.

4- Objectif de la maintenance

L'objectif des équipes de maintenance est de maintenir les installations de production en parfait état et d'assurer le rendement global maximum tout en optimisant le coût. L'obtention du meilleur rendement passe par la prévention des pannes, le respect des cadences de production et l'amélioration continue de la qualité des produits. Maintenir, ce n'est plus subir les pannes mais maîtriser les défaillances par l'optimisation de la politique de Maintenance, par une bonne prévention, par des réparations rapides et efficaces, enfin par l'amélioration du matériel. [2]

Pour assurer correctement cette mission, il est nécessaire de se doter en plus de la compétence technique des hommes, d'une organisation efficace et d'outils adéquats. C'est la nature de l'entreprise qui fixe les l'objectifs, des services de la maintenance. On peut classer les objectifs de la maintenance on deux catégories:

- 1- Les objectifs financiers,
- 2- Les objectifs opérationnels.

Le fait que ces deux objectifs sont différents expliquera pourquoi la production et la maintenance sont souvent à couteau tirés et pourquoi les deux attitudes sont apparemment opposées.

4.1 Objectifs financiers

- 1- Réduire au minimum les dépenses de la maintenance.
- 2- Augmenter au maximum les profits.
- 3- Avoir des dépenses de maintenance en fonction de l'âge des installations et de son taux d'utilisation.

4.2 Objectifs opérationnels

- 1- Maintenir les équipements.
- 2- Assurer la disponibilité maximale des installations et des équipements.
- 3- Fournir un service qui élimine la panne à tous les moments à tout prix.
- 4- Pousser à la dernière limite la durée de vie de l'installation.
- 5- Assurer une performance (rendement) de haute qualité.

5-Les cinq niveaux de la maintenance

1 er niveau : Réglages simples prévus par le constructeur au moyen d'éléments accessibles sans aucun démontage ou ouverture de l'équipement, ou échanges d'éléments consommables accessibles en toute sécurité, tels que voyants ou certains fusibles, etc...Ce type d'intervention peut être effectué par l'exploitant du bien, sur place, sans outillage et à l'aide des instructions d'utilisation, le stock des pièces consommables nécessaires est très faible.

2ème niveau: Dépannage par échange standard des éléments prévus à cet effet et opérations mineures de maintenance préventive, telles que graissage ou contrôle de bon fonctionnement. Ce type d'intervention peut être effectué par un technicien habilité de qualification moyenne, sur place, avec l'outillage portable défini par les instructions de maintenance, et à l'aide de ces mêmes instructions. On peut se procurer les pièces de rechange transportables nécessaires sans délai et à proximité immédiate du lieu d'exploitation.

3ème niveau: Identification et diagnostic des pannes, réparations par échange de composants ou d'éléments fonctionnels, réparations mécaniques mineures et toutes opérations courantes de maintenance préventive telles que réglage général ou réaligement des appareils de mesure. Ce type d'intervention peut être effectué par un technicien spécialisé, sur place ou dans le local de maintenance, à l'aide de l'outillage prévu dans les instructions de maintenance ainsi que des appareils de mesure et de réglage, et éventuellement des bancs d'essais et de contrôle des équipements et en utilisant l'ensemble de la documentation nécessaire à la maintenance.

4^{ème} niveau: Tous les travaux importants de maintenance corrective ou préventive à l'exception de la rénovation et de la reconstruction. Ce niveau comprend aussi le réglage des appareils de mesure utilisés pour la maintenance, et éventuellement la vérification des étalons du travail par les organismes spécialisés. Ce type d'intervention peut être effectué par une équipe comprenant un encadrement technique très spécialisé dans un atelier spécialisé.

5^{ème} niveau: Rénovation, reconstruction ou exécution des réparations importantes confiées à un atelier central ou à une unité extérieure. Par définition, ce type de travaux est donc effectué par le constructeur, ou par le reconstruteur, avec des moyens définis par le constructeur et donc proches de la fabrication. [2]

Commentaire

Niveau	Personnel d'Intervention	Moyens
1 ^{er}	Exploitant, sur place.	Outillage léger défini dans les instructions d'utilisation
2 ^{ème}	Technicien habilité, sur place	Idem, plus pièces de rechanges trouvées à proximité, sans délai.
3 ^{ème}	Technicien spécialisé, sur place ou en local de maintenance	Outillage prévu plus appareils de mesure, banc d'essais, contrôle...
4 ^{ème}	Equipe encadrée par un Technicien spécialisé, en atelier central	Outillage général plus spécialisé, matériel d'essais, contrôle...
5 ^{ème}	Equipe complète, polyvalente, en atelier central	Moyens proches de la fabrication par le constructeur


Tableau 1.1 tableau des niveaux de maintenance

6-Types de maintenance :

Nous distinguons plusieurs types de maintenance (figure 1.1) :

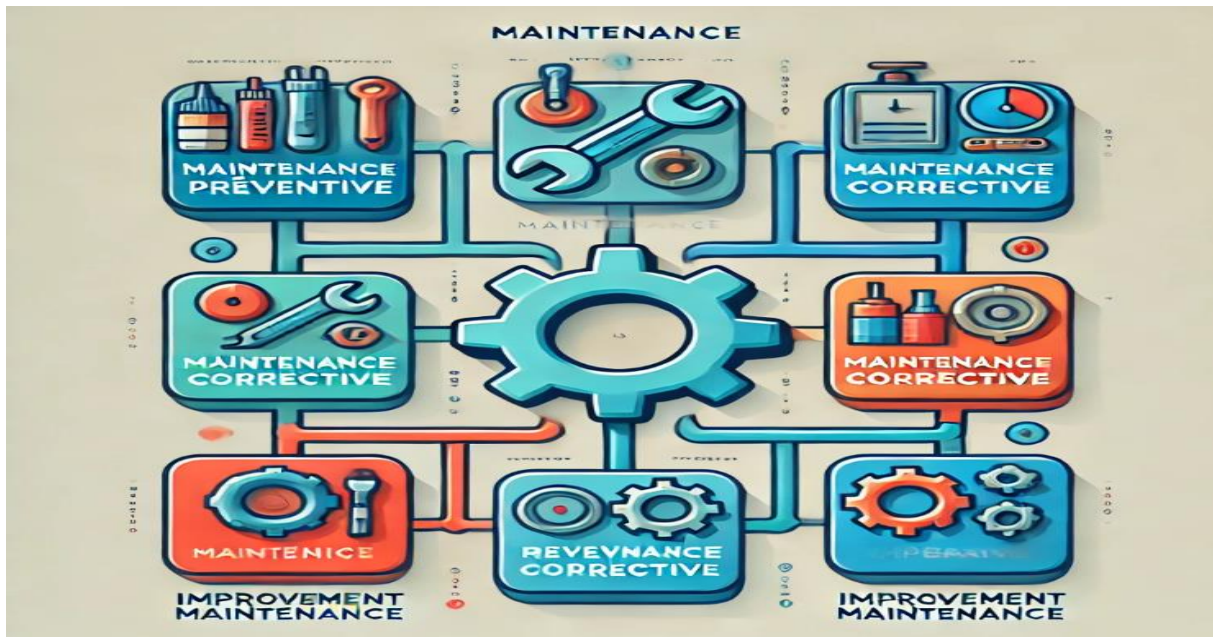


Figure 1.1 Type de maintenance

6.1 Maintenance corrective :

Elle consiste à intervenir à la suite des pannes. Cela ne veut pas dire que toutes ces pannes n’ont pas été prévisibles. Ce type de maintenance sera facilité par une bonne maintenabilité (aptitudes à maintenir ou à rétablir un dispositif, dans un état, lui permettant d’accomplir sa fonction). Il pourra permettre d’améliorer la fiabilité globale en analysant les problèmes rencontrés en contrôle techniques ou cercles de qualité ou avec les constructeurs. [2]

a) Maintenance palliative :

La maintenance palliative correspond au dépannage. Elle consiste à remettre provisoirement une machine ou un outil en état, dans l'attente de sa réparation. La maintenance palliative permet d'éviter l'arrêt totale de production, on parle alors de reprise de production en mode « dégradé ».

b) Maintenance curative :

La maintenance curative est un type de maintenance utilisé pour réparer des équipements, des systèmes ou des infrastructures qui ont subi une panne ou une défaillance. Elle se concentre sur la restauration rapide du fonctionnement normal de l'équipement en identifiant et en corrigeant les problèmes qui ont causé la défaillance. Cette forme de maintenance est souvent utilisée en réponse à des pannes imprévues ou à des arrêts inattendus des équipements. La maintenance

curative peut être effectuée par des techniciens internes ou des fournisseurs de services externes en fonction des besoins et des ressources disponibles.

6.2 Maintenance préventive :

Ce type de maintenance vise à diminuer la probabilité de défaillance d'un système. Elle s'appuie sur :

a) Maintenance systématique

La maintenance systématique est une opération de maintenance réalisée par un technicien de manière régulière selon un certain cycle de temps. En général, sa fréquence est déterminée par le constructeur, mais peut aussi être ajustée selon l'utilisation qui en est faite.

b) Maintenance conditionnelle

Ce type de maintenance consiste à effectuer un diagnostic avant de remplacer l'organe en question. Elle s'applique particulièrement aux machines tournantes. Elle est basée sur l'analyse : - externe des machines (corrosion, ancrages, etc.....) - des paramètres tels que (la pression, le débit) - des huiles de lubrification et étanchéité. - des températures d'organes des machines - des vibrations et des bruits au niveau des organes. La maintenance conditionnelle permet donc d'éviter les inconvénients de la maintenance systématique. Par le fait qu'elle permet de déterminer quel organe défaillant devra être remplacé et la date à laquelle s'impose l'intervention en se basant uniquement sur les analyses en temps réel de la machine et non sur des données statistiques. Bien évidemment, pour faire de telles prédictions, il est nécessaire d'accéder à une bonne compréhension du rôle de la machine dans les procès et de son fonctionnement interne, car il n'est pas possible de prédire l'avenir d'un équipement sans connaître son état de santé au temps présent.

c) Maintenance prédictive :

La maintenance prédictive est une stratégie de gestion de la maintenance qui consiste à surveiller l'état des équipements en temps réel, grâce à l'utilisation de capteurs et de données, afin de prédire quand une panne ou un dysfonctionnement est susceptible de se produire. Contrairement à la maintenance corrective (réparations après une panne) ou à la maintenance préventive (interventions à intervalles réguliers), la maintenance prédictive permet d'intervenir seulement lorsque cela est nécessaire, optimisant ainsi les coûts et réduisant les temps d'arrêt.

6.3 Maintenance améliorative (improvement):

La maintenance améliorative (ou maintenance évolutive) concerne les actions qui visent à améliorer ou à enrichir un système existant sans corriger un dysfonctionnement spécifique.

Ce type de maintenance se concentre sur l'ajout de nouvelles fonctionnalités, l'amélioration des performances, ou l'adaptation du système pour répondre à de nouvelles exigences technologiques ou législatives

7-Les formes d'actions d'un service maintenance

Représentants graphiquement, ci-après, les différentes formes d'action possible

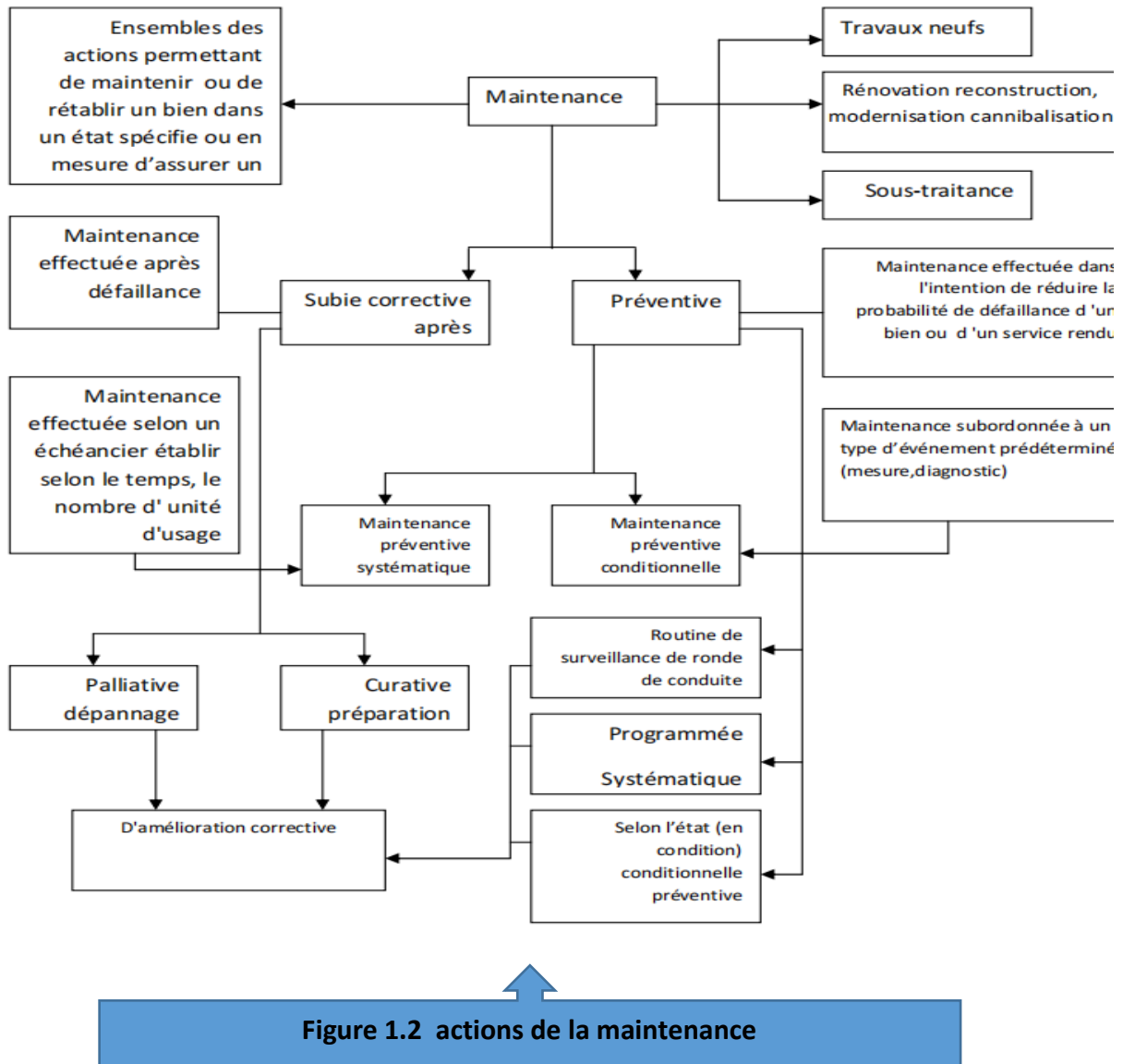


Figure 1.2 actions de la maintenance

8-Domaine d'utilisation de la maintenance :

La maintenance est utilisée dans une grande variété de domaines, car elle est essentielle pour garantir le bon fonctionnement des machines, des infrastructures, et des systèmes. Voici quelques domaines d'utilisation :

8.1 Industrie

-Machines de production :

Maintenance des machines-outils, des lignes de production automatisées, des robots industriels, etc.

-**Équipements spécialisés** : Préventive et corrective pour éviter les pannes et maintenir un niveau de production élevé. [3]

8.2 Secteur de l'énergie

-**Centrales électriques** : Maintenance des turbines, des alternateurs, des équipements de production et distribution d'électricité.

-**Énergies renouvelables** : Maintenance des éoliennes, panneaux solaires, barrages hydrauliques, etc.

-**Réseaux de distribution** : Maintenance des lignes électriques, des transformateurs, et autres infrastructures. [4]

8.3 Transport et logistique

-**Transports terrestres** : Entretien des véhicules (automobiles, camions, trains), maintenance des routes et des voies ferrées.

-**Aéronautique** : Maintenance préventive et corrective des avions, des systèmes de navigation, et des infrastructures aéroportuaires.

-**Transport maritime** : Entretien des navires, des systèmes de propulsion, des ports, etc.

8.4 Technologies de l'information :

-**Infrastructures informatiques** : Maintenance des serveurs, des équipements réseau, des systèmes de stockage, et des ordinateurs.

-**Logiciels** : Maintenance logicielle (corrective, évolutive, adaptative) pour corriger les bugs et améliorer les performances des systèmes. [5]

9-Les avantages de la maintenance :

La maintenance industrielle garantit que votre propriété est en bon état. Faire des améliorations et des travaux d'entretien au sein de votre installation gardera votre propriété en bon état et vous fera économiser de l'argent en même temps.

10-Conclusion :

La maintenance, en tant que fonction clé au sein des entreprises et des organisations, joue un rôle déterminant dans la préservation de la performance, la durabilité des équipements et des infrastructures. En s'appuyant sur diverses stratégies comme « corrective, préventive, et améliorative », elle vise à minimiser les interruptions de production, à prolonger la durée de vie des actifs, et à optimiser les coûts d'exploitation

CHAPITRE 2 :
**Généralité sur la gestion de maintenance
assistée par ordinateur GMAO**

1-Introduction :

La gestion de la maintenance assistée par ordinateur (GMAO) est un système informatisé qui facilite la gestion des activités de maintenance des entreprises. Elle permet de centraliser, d'organiser et de suivre toutes les informations liées à la maintenance des équipements, des infrastructures, et des installations industrielles. La GMAO aide les organisations à planifier, exécuter et contrôler les tâches de maintenance de manière plus efficace. [6]

2-Bref historique sur l'évolution de l'informatique industrielle :

Le paysage informatique des entreprises industrielles a été marqué par l'évolution parallèle de l'informatique pour l'ingénierie et de l'informatique pour la gestion, qu'il s'agisse de gestion de production ou de gestion administrative (ventes, personnel, comptabilité etc.). Ces applications ont été développées de manière indépendante, pour un domaine fonctionnel précis, sans avoir une vision intégrée des besoins. Les premiers outils développés pour les départements d'ingénierie ont été les outils de calcul et d'aide à la conception dans les années 60. Puis, dans les années 80, les systèmes de gestion des données techniques (SGDT) pour des départements d'ingénierie sont apparus. Ils étaient d'abord destinés à gérer les plans des pièces puis ils ont évolué pour gérer les données provenant des différents systèmes utilisés par les départements d'ingénierie, à savoir les calculs, les programmes de fabrication, les instructions de contrôle, les publications techniques, les notices etc.

L'informatique de gestion s'est développée depuis le début des années 60, évoluant depuis les systèmes de gestion de production MRP permettant la planification des besoins en composants (Material Requirement Planning) aux systèmes de gestion de production MRP permettant la planification de toutes les ressources associées à la production (manufacturions resource planning).

Aujourd'hui les ERP « Enterprise Resource Planning » ne se contentent plus de gérer la production, ils intègrent des modules permettant de gérer l'ensemble de l'entreprise de façon intégrée. Les ERP regroupent en effet des modules de gestion des achats et des ventes, de gestion du personnel, de comptabilité, de gestion de la qualité etc. [7]

Définition de la GMAO :

- **Gestion** : prise de décisions suivant une approche
- **Maintenance**: ensemble des activités permettant de maintenir ou de rétablir un bien dans un état spécifié.
- **Assisté par Ordinateur**: l'utilisation de l'ordinateur comme aide à la décision. [8]

La GMAO est un système informatisé qui centralise les informations relatives aux équipements, planifie les interventions de maintenance, gère les stocks de pièces de rechange, et permet de suivre l'historique des interventions. Elle vise à réduire les temps d'arrêt des équipements, prolonger leur durée de vie et optimiser les coûts de maintenance en automatisant plusieurs processus liés à la gestion de la maintenance.

En d'autres termes, la GMAO améliore la gestion des ressources humaines, matérielles, et techniques nécessaires pour garantir le bon fonctionnement des équipements au sein d'une entreprise ou d'une organisation. [9]

3-Définition d'un logiciel de GMAO

On peut définir la GMAO comme un progiciel permettant une aide à la décision dans une entreprise dans le but de :

- Augmenter la disponibilité de l'outil de production.
- Diminuer les coûts de maintenance.
- Rationaliser les coûts du stock de pièces détachées.
- Automatiser le processus d'achat et réduire les coûts d'approvisionnement.
- Simplifier la gestion des flux de données.

Les produits de GMAO ont été développés afin de rassembler dans une base de données interfacée avec le système interne de l'entreprise, toutes les informations nécessaires pour enregistrer, traiter et actualiser les différents processus et données des entreprises.

Pour essayer de mieux comprendre l'impact d'une GMAO, on peut par simplification assimiler l'activité d'un service de maintenance à :

- Un système de file d'attente avec des demandes de service réparties selon une loi de probabilité.
- Des durées d'intervention qui suivent une distribution exponentielle négative.

En fonction des ressources disponibles, on peut déterminer un temps moyen d'attente provoqué par une panne ainsi que son coût. Sans l'informatique il est difficile, même impossible, d'optimiser en termes de coûts et de performance, la relation entre la disponibilité des intervenants et la probabilité de l'occurrence d'une panne. [10]

3.1 Fonctionnalités d'un logiciel de GMAO

La mise sous informatique de la gestion d'un service maintenance de PME apparaît aujourd'hui incontournable dans la plupart des secteurs d'activité.

Dès lors qu'un service maintenance est structuré, il a fait la preuve de l'efficacité de son organisation,

l'outil GMAO se révèle indispensable par sa capacité de mise en mémoire, par ses possibilités de traitement d'informations, par ses interfaces et par sa réactivité.

Il appartient à chaque service maintenance de déterminer ses besoins internes en matière d'informatisation, mais également ses besoins de communication externes, présents et à venir.

Le choix d'un outil GMAO passe par son adéquation

- À la stratégie globale du système informatique de l'entreprise (problème d'intégration)
- Aux besoins exprimés par le service maintenance

3.2 Nécessité de l'utilisation d'un logiciel de GMAO

Pour se distinguer, il est nécessaire d'utiliser les ressources disponibles mieux que les autres. Il faut donc éliminer toute forme de gaspillage. L'idéal c'est d'éliminer toutes les pertes et atteindre les cinq zéros :

- Zéro pannes : c'est l'objectif naturel de la maintenance.
- Zéro défauts : garder tout outil de production en parfait état : tout défaut entraîne un arrêt de production, donc augmentation des coûts et des délais.
- Zéro stocks, zéro délais : un outil de fabrication fiable permet une fabrication sans stock (flux tendu) et une livraison sans délai.
- Zéro papier : il faut comprendre par ce mot « zéro papier inutile », c'est à dire les papiers engendrés par les erreurs, les défauts, les défaillances, les retards qui viennent alourdir le travail et l'organisation.

Pour réduire puis éliminer ces gaspillages, il faut : minimiser les arrêts machines, améliorer les machines du parc existant, et maximiser l'utilisation de ces machines.

Les gains réalisés permettent de réduire les frais financiers et d'introduire de nouveaux équipements en tenant compte de l'expérience du passé : ne pas refaire les mêmes erreurs.

Pour atteindre ces objectifs, il faut savoir calculer les différents ratios et pratiquer les analyses

nécessaires pour pouvoir prendre des décisions.

Mais la difficulté réelle de la maintenance est de collecter toutes les informations nécessaires pour déterminer le MTBF ou le MTTR ou autres. En effet, pour résoudre un problème de remplacement d'une pièce soumise à l'usure ou encore d'investissement de remplacement d'une machine de production, il faut des statistiques exactes des coûts de consommation, de pannes et de réparations, de pourcentages de rebuts en production, etc.

Il apparaît donc un besoin d'un produit informatique qui rassemble toutes ces informations. Cet outil peut être un logiciel de GMAO.

3.3 Fonctions principales d'un logiciel de GMAO

Le rôle principal d'un logiciel de GMAO est d'assurer le bon fonctionnement des installations, et de réduire les temps d'arrêts des machines en fournissant des outils d'aide à la décision qui facilitent la planification et l'organisation des opérations de maintenance.

Les fonctions d'un tel logiciel peuvent être résumées comme suit :

- Établissement d'un fichier de base qui recense la définition et la codification des éléments de l'installation concernée, les fréquences d'intervention, les intervenants par qualification et affectation des taux horaires.
- Organisation de la maintenance préventive qui est établie à partir du fichier de base, avec édition des programmes d'intervention, des chiffrages (temps et coûts), des interventions hebdomadaires, des fiches de travail par équipement et/ou par composant et qualification des intervenants.
- Organisation de la maintenance corrective avec édition des ordres d'exécution établissent tenant compte de l'équipement, de l'objet de l'intervention, de la qualification de l'intervenant, de l'échéance de l'intervention, et du degré d'urgence.
- Contrôle de gestion exercé à partir de la saisie des informations sur les fiches de travail en retour, de l'exécution du contrôle et la validation des interventions effectuées, de l'exécution du suivi et de la remise à jour du planning.
- Fonctions complémentaires: le logiciel doit pouvoir, en plus des quatre fonctions précédentes : enregistrer des événements et éditer des historiques, éditer des ratios économiques dont l'évolution pourra être traduite graphiquement (courbes de Pareto, analyses AMDEC, etc.), assurer la gestion du stock des pièces de rechange, etc. [10]

3.4 Les différents modules fonctionnels d'une GMAO

Tous les logiciels de GMAO ont en commun la même structure modulaire proposant les mêmes fonctions. Mais, selon les logiciels, les fonctions remplies sont diversement dénommées, diversement réparties et diversement organisées.

Dans les bureaux techniques d'une entreprise (méthodes, ordonnancement, logistique et travaux neufs), on pourra effectuer la gestion par exploitation des 10 modules suivants:

3.4.1 Module "gestion des équipements"

Il s'agit de décrire et de coder l'arborescence du découpage allant de l'ensemble du parc à maintenir aux équipements identifiés et caractérisés par leur DTE (dossier technique d'équipement) et leur historique, puis à leur propre découpage fonctionnel.

3.4.2 Module "gestion du suivi opérationnel des équipements"

Ce module permettra de suivre les performances d'un équipement à partir des indicateurs de fiabilité, de maintenabilité, de disponibilité.

3.4.3 Module "gestion des interventions"

Ce module doit permettre un enregistrement rapide de la durée, de la localisation, et de la nature d'une intervention.

3.4.4 Module "gestion du préventif"

Ce module doit permettre de gérer la maintenance systématique à travers un planning calendaire par équipement, les dates doivent être déterminées à partir d'un relevé de compteur (ou d'une mesure dans le cas de la maintenance conditionnelle). Ce qui fait que le déclenchement sera automatique, par listing hebdomadaire des opérations prévues dans la semaine. Chaque ation sera prévue par sa gamme préventive.

3.4.5 Module "gestion des stocks"

Le système repose sur le fichier des articles en magasin comprenant les "lots de maintenance" par équipement et sur les mouvements entrées/sorties du magasin.

3.4.6 Module "gestion des approvisionnements et des achats"

Les caractéristiques de la fonction de maintenance sont beaucoup de références et de fournisseurs pour des quantités faibles et des délais courts. Ce module doit permettre la gestion des achats.

3.4.7 Module "analyse des défaillances"

La base de ce module est constituée des historiques automatiquement alimentés par chaque saisie de BT (bons de travaux) ou OT (ordre de travaux). Ce qui permet une analyse quantitative ou qualitative des défaillances.

3.4.8 Module "budget et suivi des dépenses"

L'objectif de ce module est le suivi de l'évolution des dépenses par activité dans un budget donné.

3.4.9 Module "gestion des ressources humaines"

Spécifiquement adapté au service maintenance, ce module sera principalement une aide à l'ordonnancement.

3.4.10 Module "tableaux de bord et statistiques"

Les tableaux de bord concernent la mise en forme de tous les indicateurs techniques, économiques, et sociaux sélectionnés pour assurer la gestion et le management du service maintenance.

4-La conduite d'un projet GMAO

Le projet GMAO est pour le service maintenance un projet "structurant" remettant en cause des habitudes de travail, donc susceptible de modifier en profondeur l'état d'esprit et la motivation des acteurs. C'est une opportunité pour réorganiser un secteur, pour optimiser des procédures, pour élever le niveau de sensibilité à la gestion de tous les acteurs et pour en promouvoir certains. Il ne faut pas négliger ni sous-estimer le poids de la formation dans le coût du projet : l'acceptation de l'outil est la condition incontournable de réussite du projet et son refus aboutit à l'échec total.

4.1 Les étapes d'un projet GMAO**4.1.1 Étude de faisabilité**

Elle passe par le dimensionnement du projet en termes de ressources matérielles, humaines et financières, menée à l'aide d'un consultant expérimenté en GMAO. [10]

4.1.2 Rédaction du cahier des charges de consultation

L'idée du "juste nécessaire" doit éviter d'investir dans des fonctions inutiles, inadaptées, superflues et coûteuses.

4.1.3 Choix de l'outil GMAO et de ses modules nécessaires

A partir d'un problème bien posé, la réponse est supposée aisée. Le choix du logiciel se fera après appel d'offre détaillé et essais sur les produits sélectionnés.

4.1.4 Implantation, plan de formation et démarrage

Le moment de l'implantation doit être bien choisi, en dehors d'une période de forte activité, et doit être précédé d'une forte information. Mais la crainte "à priori de l'informatique" étant toujours forte.

Cette information doit porter à la fois sur les objectifs généraux de l'informatisation, sur les caractéristiques de l'outil sélectionné et sur le rôle de chaque secteur.

4.2 Le modèle itératif de la gestion

Le modèle itératif « Observer, Réfléchir et Agir », toujours recommencé, est un modèle de gestion « naturel », puisque calqué sur le modèle de fonctionnement de l'homme. Il est important de noter qu'il contient une potentialité de progrès, à partir de l'observation des résultats de l'action.

4.3 Application à la gestion du service maintenance

La Figure 2.1 illustre la place omniprésente de la « base de données » qu'est une GMAO. Il ne faut pas oublier que les phases « productives » sont l'action, la connaissance et la décision. Les structures d'analyse des informations puis de prise de décisions sont réparties entre le bureau des méthodes, le bureau d'ordonnancement et de logistique et la direction du service, en fonction de l'organisation en place.

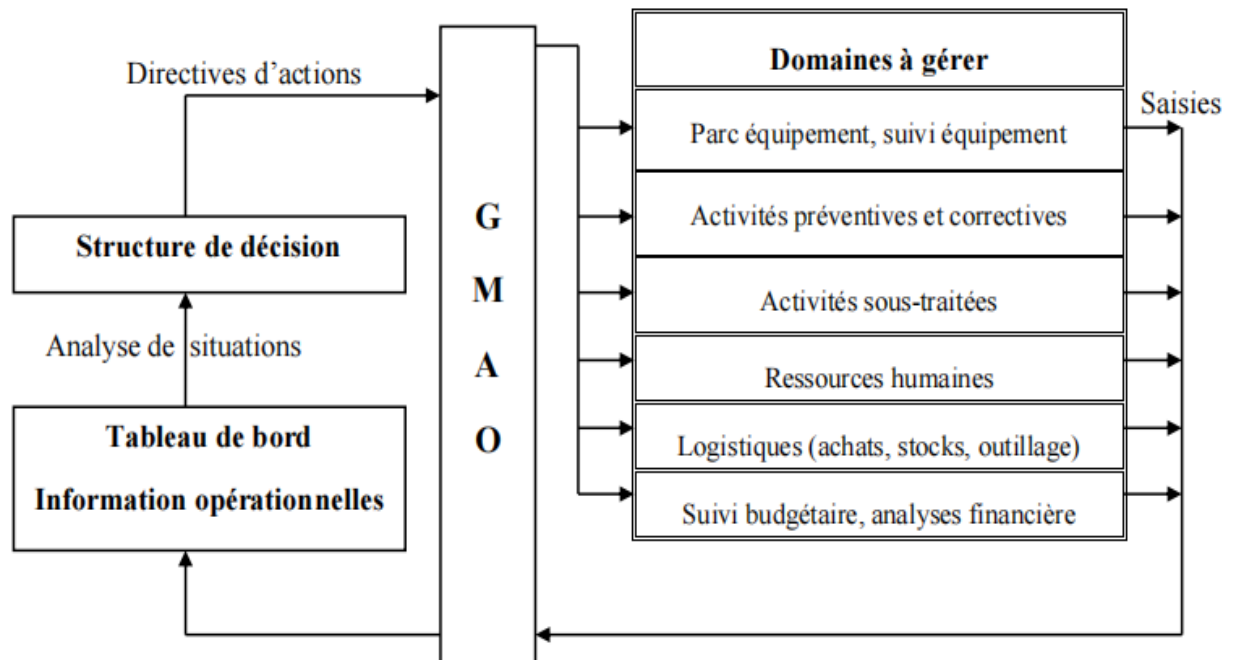


Figure 2.1 Gestion itérative de la maintenance

5-La réussite de la GMAO :

Pour aboutir à la réussite du projet, l'implantation d'une GMAO nécessite obligatoirement au préalable une analyse fine et fouillée des besoins, une définition précise des objectifs, une préparation soignée des acteurs et l'adhésion de tous. En effet, une telle

démarche doit se faire sous l'impulsion de la direction et la participation du personnel de l'entreprise. La décision d'investir dans un outil de GMAO remet en question les habitudes acquises par les différents services de l'entreprise qui utilisent pour des raisons historiques des systèmes de codages différents pour un même équipement en fonction de leur domaine de spécificité propre (achats, méthodes, travaux neufs, maintenance, conduite). L'évolution de l'offre GMAO et du marché va vers l'utilisation des technologies nomades, ce qui va entraîner une nouvelle remise en question des habitudes qui ont été prises depuis la première installation.

[10]

5.1 Les conditions de réussite d'un GMAO :

5.1.1 La convivialité de l'outil informatique :

Le système va être donc utilisé par tout le personnel du service maintenance, et non réservé à certains. Il faut donc bannir l'ésotérisme, exigé absolument que la totalité de l'information soit disponible en langage connu, s'efforcer de trouver un bon compromis entre langage normalisé et terme de métier habituel de l'entreprise.

5.1.2. Volonté de réussite de consensus :

C'est une question de motivation de la part de tous les intéressés, qu'il convient donc de faire participer à la définition générale du projet GMAO

5.1.3. Facteur temps :

Le facteur temps doit être pris en considération. La mise en place d'un système De GMAO peut durer jusqu'à 2 ans.

6- comparaison des quelques logiciels GMAO :

6.1. IBM Maximo

Fonctionnalités :

- Gestion des actifs : Suivi complet des actifs, gestion des cycles de vie.
- Maintenance Préventive et Corrective : Planification des tâches, gestion des ordres de travail.
- Gestion des stocks : Suivi des pièces de rechange, gestion des approvisionnements.
- Analyse et rapports : Outils d'analyse avancés.

Efficacité :

- Points forts : Très complet et personnalisable, idéal pour les grandes entreprises.
- Points faibles : Complexité élevée, courbe d'apprentissage importante. Coût élevé, nécessite des ressources pour la configuration.

Prix :

- Modèle de tarification : 10 000 à 100 000 USD par an, selon les fonctionnalités et le nombre d'utilisateurs.[11]

6.2. SAP PM (Plant Maintenance)**Fonctionnalités :**

- Gestion des actifs: Gestion des équipements, maintenance préventive et corrective.
- Gestion des stocks et Achats: Intégration avec les modules SAP pour les achats et stocks.
- Analyse et rapports: Outils analytiques avancés.

Efficacité :

- Points forts: Intégration robuste avec SAP, puissant pour les grandes entreprises.
- Points faibles : Coût élevé, mise en œuvre complexe, personnalisation importante requise.

Prix :

- Modèle de tarification: 20 000 à 100 000 USD par an, en fonction de la taille et des modules. [12]

6.3. Infor EAM**Fonctionnalités :**

- Gestion des actifs: Gestion des cycles de vie, maintenance préventive et corrective.
- Gestion des stocks: Suivi des pièces et des inventaires.
- Mobilité: accès mobile pour la gestion des tâches de maintenance.

Efficacité :

- Points forts: Flexible et personnalisable, bon compromis entre fonctionnalités et coût.
- Points faibles: Peut nécessiter des ajustements pour des besoins très spécifiques.

Prix :

- Modèle de tarification: 5 000 à 30 000 USD par an, selon le nombre d'utilisateurs.[13]

6.4. Asset Panda**Fonctionnalités :**

- Gestion des actifs: Suivi des actifs, gestion des réparations et de la maintenance.
- Gestion des stocks: Suivi des pièces et des fournitures.
- Mobilité: Application mobile développée.

Efficacité :

- Points forts: Interface conviviale, coût plus accessible, bon pour les PME.
- Points faibles: Fonctionnalités moins avancées, moins adapté aux configurations complexes.

Prix :

- Modèle de tarification: 500 à 5 000 USD par an, selon le nombre d'utilisateurs.[14]

6.5 UpKeep**Fonctionnalités**

- Gestion des actifs : Gestion des ordres de travail, planification de la maintenance.
- Gestion des stocks : Suivi de l'inventaire et des pièces de rechange.
- Mobilité : Application mobile développée.

Efficacité :

- Points forts : Interface moderne et facile à utiliser, bon rapport qualité-prix.
- Points faibles : Fonctionnalités moins avancées, moins adapté aux configurations complexes.

Prix :

- Modèle de Tarification : 600 à 3 000 USD par an par utilisateur. [15]

Bilan :**Forces et Faiblesses :**

- IBM Maximo et SAP PM : Ces solutions sont très robustes et adaptées aux grandes entreprises avec des besoins complexes, mais elles présentent des coûts élevés et une complexité de mise en œuvre importante.
- Infor EAM : Offre un bon compromis entre coût et fonctionnalités, avec une flexibilité appréciée, mais peut nécessiter des ajustements pour des besoins spécifiques.
- Asset Panda et UpKeep : Ces options sont plus accessibles et conviennent bien aux PME, offrant des interfaces conviviales et des coûts plus bas, mais elles peuvent manquer de certaines fonctionnalités avancées requises pour des configurations très complexes.

Chaque logiciel a ses propres avantages selon les besoins spécifiques de l'entreprise, et le choix doit se faire en tenant compte des fonctionnalités nécessaires, du budget disponible, et de la capacité à gérer la complexité du système.

7-Les utilisateurs de GMAO :

La GMAO est utilisée par les techniciens, les opérateurs de production, de maintenance et le service des achats (hors matières premières).

La GMAO est utilisée par...	Pour réaliser...
Les techniciens de maintenance	- La préparation des travaux, comptes rendus - La recherche des informations techniques - L'exécution des diagnostics, la consultation de l'historique
Les responsables de maintenance	- Le contrôle et le suivi des coûts - Le contrôle des factures et leur envoi à la comptabilité
Le service méthodes	- Un Pareto des pannes et défaillances - La préparation et la planification des travaux
Le service des travaux neufs	- La gestion des travaux - Le suivi budgétaire des dépenses engagées
Les gestionnaires	- Le suivi des coûts par machine, ligne de production, installations - Le suivi d'évolution des performances, optimisation des charges - La participation au tableau de bord de l'activité
Les magasiniers	- La réception des matériels et leur gestion - L'identification des demandes à l'avance et leur préparation - La gestion des stocks - Les inventaires
Le contrôleur de gestion	- Le recueil dans la GMAO des composantes du coût de fabrication et des causes de surcoûts
Les opérateurs et responsables de production	- La création de demandes d'intervention, les relevés opérationnels et performances équipements
Les comptables	- La réception, identification des factures contrôlées
Les acheteurs	- La réception des demandes d'achats et services - L'utilisation du module achat de la GMAO

Tableau 2.1 : Les fonctions de la GMAO suivant ses utilisateurs

8-Conclusion :

Dans ce chapitre, on a défini les généralités de la GMAO. En effet nous y avons ressort une ébauche de définition général d'un progiciel de GMAO pour en déduire les conditions de réussite. Nous avons ensuite présenté quelques progiciels avec leurs caractéristiques.

Chapitre 3 :
Conception et architecture du système

1-Introduction :

Ce 3ème chapitre est consacré à la présentation de l'architecture logicielle et la conception technique du logiciel de GMAO.

L'objectif général est d'exposer les choix technologiques effectués, notamment KOTLIN pour le développement de l'application et de SQLite comme système de gestion de base de données.

2-les besoins fonctionnels :

2.1 Gestion des équipements : permet de gérer les différents types des équipements (camion, fourgon, machine,...) avec possibilité de les supprimer, modifier, rechercher et voir détails.

Pour chaque équipement, il sera mis en place une fiche technique, fiche d'assurance, fiche de la facture et un dossier historique qui se contient les informations suivantes :

- L'année de mise en service
- numéro d'identifiant et numéro de chassé
- type d'équipement
- marque d'équipement
- série et année de fabrication
- Prix
- l'inventaire
- MTBF de l'équipement

2.2 Gestion de ressources humaines : permet gérer un ou plusieurs employées avec possibilité d'ajouter supprimer, recherché, et modifier et voir les détails historique comme :

- nom et prénom
- l'email
- service
- poste occupée
- Listes des taches en cours
- Listes des taches accomplis

2.3 Gestion de stock : permet de gérer les pièces détachées de différents types (moteur, suspension,..) avec possibilité d'ajouter et voir leur détails tels que :

- type
- criticité
- quantité stockée

2.4 Gestion d'historique et contenu de stock : sert à voir les détails des pièces dans le stock (date d'entrée, date de sortie, la quantité disponible).

2.5 Gestion des interventions : permet de gérer l'ordre d'intervention de le modifier, supprimer et détailler.

Pour chaque ordre d'intervention une fiche d'historique avant et après la planification qui contient :

- employées affectées pour la tâche
- pièces demandées.
- étapes de résolution de problème
- possibilité de changement de la date
- annuler l'OI
- Rapport PDF de l'OI

2.6 Gestion des moyens économiques : Toutes les activités de maintenance impliquent des coûts qui peuvent être très variés :

- Les coûts d'achats et d'installation des équipements.
- Les coûts de main d'œuvre : corrective, préventive (systématique, conditionnelle...).
- Les coûts des rechanges : sorties magasins et achats directs.
- Les coûts des sous-traitances.
- Les coûts indirects (des pertes de production).
- Les coûts des modifications et améliorations.

Le suivi de ces différents postes de charges est très important pour les entreprises car ils interviennent directement dans la politique budgétaire. Aussi interviennent-ils dans les différents tableaux de bord que la maintenance est appelée à produire.[16]

CHAPITRE 3 conceptions et architecture du système

2.7 Gestion des demandes d'interventions_ : permet de gérer les demandes d'interventions avec des opérations d'ajout, suppression, modification, recherche et affichage d'historique de la demande qui contient :

- raison de la demande
- date
- importance de la demande
- type de demande (corrective, préventive ...)
- équipement concernée
- description détaillé
- état de la demande
- Rapport PDF

2.8 Gestion des notifications : Alerter si le temps de contrôle technique ou l'assurance d'un équipement bientôt sera fini

2.9 Gestion de tableau de bord : Permet d'afficher les statistiques de stocks (états de pièces dans le stock, et le nombre) et les statistiques des équipements (état des équipements, nombre d'intervention par équipement)

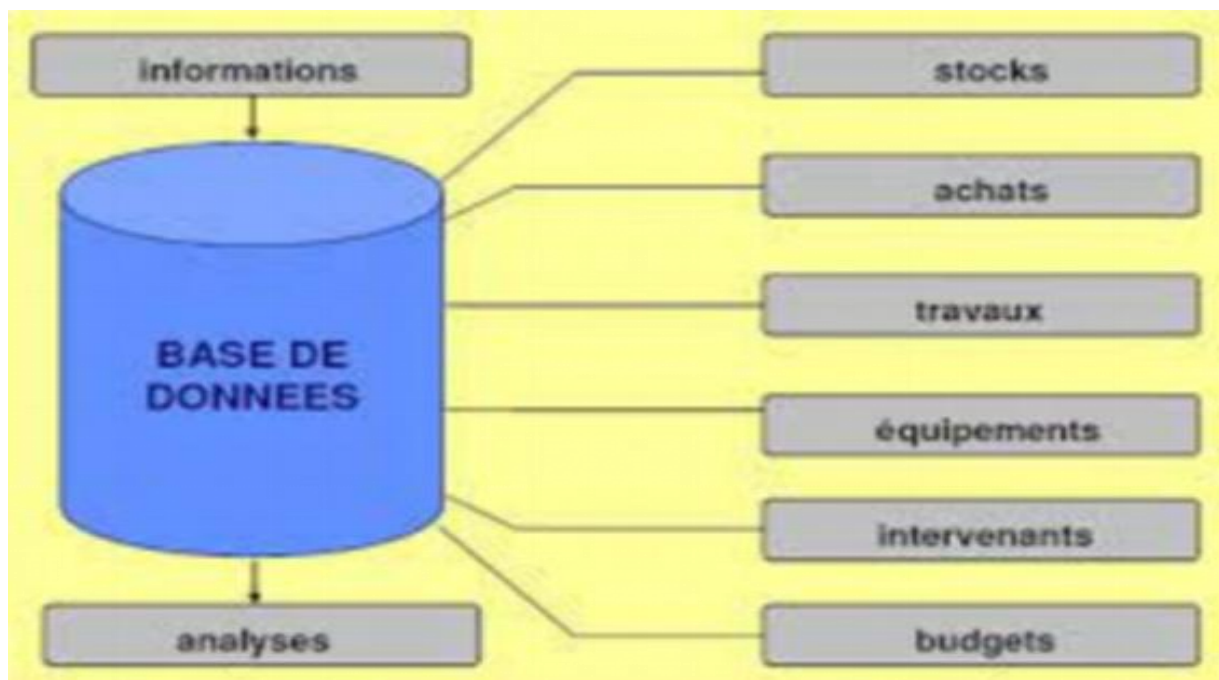
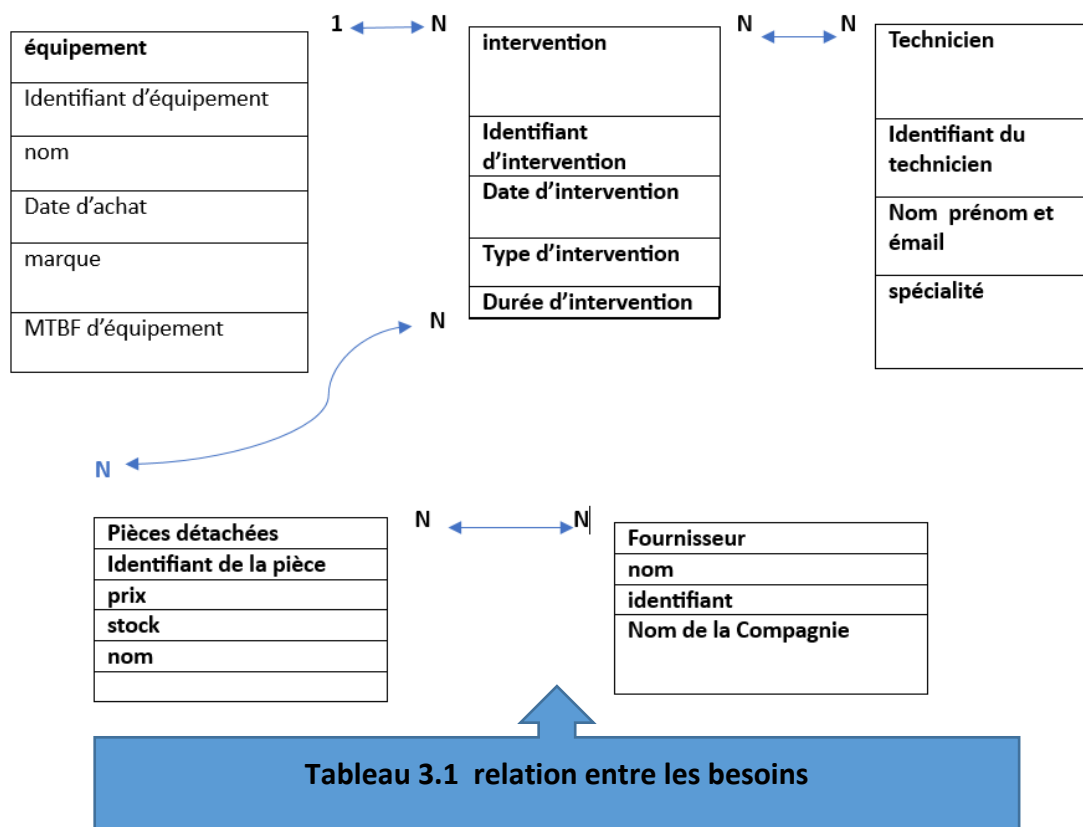


Figure 3.1 besoins fonctionnels relié au système [17]

3-Etablissement des relations entre les besoins :

Les relations définissent comment les besoins interagissent entre elles, on trouve plusieurs cas de relation comme :

- **Un a plusieurs (1, n)** : comme un équipement peut avoir une ou plusieurs intervention au même temps, mais par contre une intervention est associée à un seul équipement
- **Plusieurs à plusieurs (n, n)** : un technicien peut être assigné a plusieurs interventions, et une intervention peut nécessite plusieurs techniciens à la fois.
- **Un à un (1,1)** : c'est moins fréquent, mais pourrait être utilisé si un équipement spécifique ne peut être associé qu'un seul fournisseur.



4-Architecture logicielle :

Dans cette étape on va voir la structure de logiciel en différents composants et définir comment ces composants vont interagir pour répondre aux besoins du système.

4.1 Base de données : c'est la partie qui stocke et gère l'information ou les données.

CHAPITRE 3 conceptions et architecture du système

C'est ici que toutes les informations comme équipements, ressources humains, interventions,... sont enregistrées.

Pour effectuer cette tâche on a utilisé **SQLITE** qui est un choix très judicieux pour les applications légères nécessitant une base de données intégrée.

Voici un diagramme de base de données qui présente les classes et les interfaces des systèmes ainsi que les différentes relations entre eux.

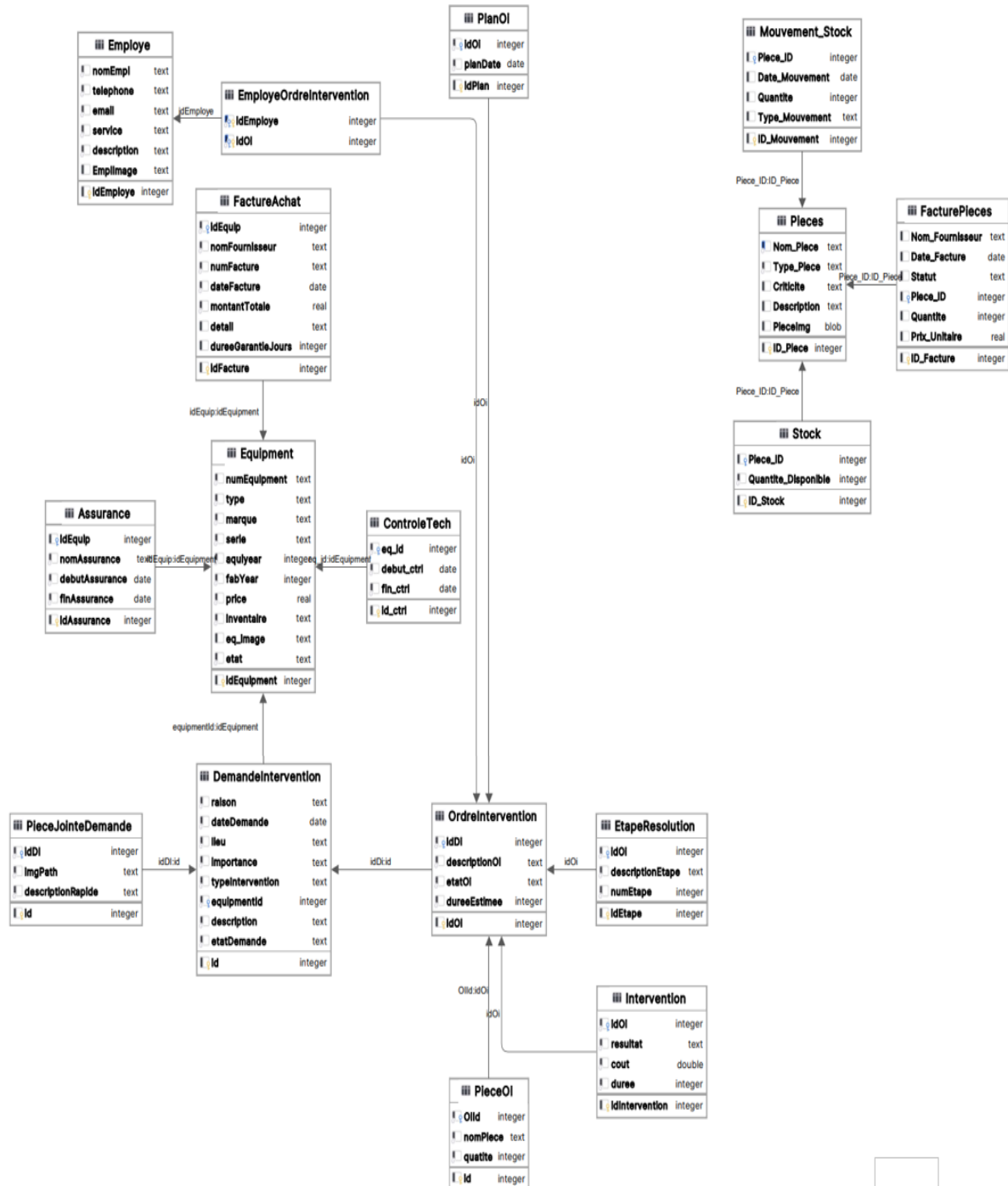


Figure 3.2 schéma de base de données

4.2 Coté client (front end) :

Le coté client (frontend) du logiciel en général constitue l'interface ou les utilisateurs interagissent pour y accéder aux fonctionnalités du système.

L'interface est responsable de la présentation des données, de gérer les actions des utilisateurs afin que les envoyer au système (backend) et renvoyer par le système.

Objectif :

L'objectif du frontend c'est de fournir une interface intuitive, performante adapté aux besoins des utilisateurs du ce logiciel GMAO.

4.3 Conception front end :

Pour réussir cette tâche on a utilisé un langage de programmation qui s'appelle KOTLIN qui est interprétable avec la machine virtuelle Java, les bibliothèques Java et Android.

KOTLIN représente un gain du temps pour les développeurs : plus concis, ce langage permet d'écrire moins de code et d'éviter les redondances.

L'interface utilisateur a été conçu de manière à être accessible, les différentes fonctionnalités sont regroupées en section claires, avec un menu simple pour faciliter l'accès aux diverses parties de l'application.

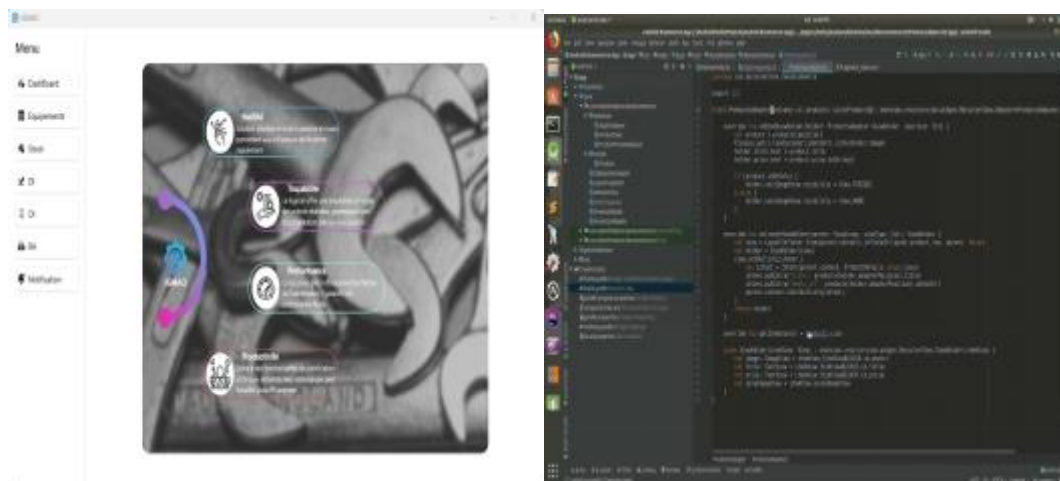


Figure 3.3 interface utilisateur

5-Structure MVVM :

Dans notre projet on a suivi l'architecture MVVM (model view viewModel) qui est :

- **Model** : est la couche où se trouve la logique métier et la gestion des données.
- **View** : la partie de l'interface qui est responsable de l'affichage des données.
- **ViewModel** : qui gère la logique de l'interface. Elle récupère les données du Model les prépare pour la View et notifie la View des changements.[18]

L'architecture globale pourrait être structurée ainsi :

5.1-Model Layer :

StockRepo et autres repositories : Les repositories comme StockRepo interagissent directement avec la base de données (ici SQLite). Ils contiennent les méthodes nécessaires pour exécuter les requêtes SQL, récupérer les données, et les transmettre au ViewModel. Chaque repository est dédié à un module (comme le stock) et gère les opérations CRUD associées à ce domaine particulier.

Autres Repositories : Il y aurait d'autres repositories similaires à StockRepo, chacun responsable d'un module spécifique (par exemple, EmployeRepo, OrdreInterventionRepo).

5.2-ViewModel Layer :

StockViewModel : Le ViewModel est responsable de la gestion des données pour l'interface utilisateur. Il reçoit les informations des repositories et les expose sous une forme que la vue (UI) peut observer.

- Le StockViewModel interagit avec le StockRepo pour effectuer des actions comme obtenir la liste des pièces en stock, ajouter de nouvelles pièces, ou gérer les mouvements de stock.

- Il utilise souvent des LiveData ou des StateFlow pour transmettre les données aux composants UI de manière réactive.

5.3-View Layer :

UI avec Jetpack Compose : La vue est construite en utilisant Jetpack Compose pour Desktop. Elle observe les données exposées par le StockViewModel et les affiche de manière dynamique.

- Par exemple, une liste d'articles en stock pourrait être affichée avec des composants Composables, qui se mettent à jour automatiquement lorsque les données dans le ViewModel changent.

6- Fonctionnement Général :

1. L'utilisateur interagit avec l'interface utilisateur (par exemple, en ajoutant un nouvel article ou en vérifiant les stocks).
2. Cette interaction est capturée par les composants UI qui appellent des fonctions dans le StockViewModel.
3. Le StockViewModel contacte le StockRepo pour exécuter des actions sur la base de données.
4. Le StockRepo interagit avec SQLite via des requêtes SQL et retourne les données au ViewModel.
5. Le ViewModel expose les données mises à jour à l'interface utilisateur, qui les affiche de manière réactive.

Cette architecture découple clairement la logique d'accès aux données (via les repositories) de la logique de gestion des données (ViewModel) et de la présentation (Vue) MVVM précédent présente une description concernant un module de notre logiciel qui est la gestion de stock, il explique le fonctionnement de l'architecture MVVM, on a donnée celui-ci uniquement car les autre modules comme employé, DI, OI,... fonctionne de la même maniéré.

7- Avantage de MVVM :

-MVVM sépare clairement les responsabilités entre la logique de présentation (viewmodel), la logique métier (model) et l'interface utilisateur (view), cela rend le code facile à maintenir.

-la séparation des responsabilités facilite les tests unitaires.

-MVVM est adapté pour les applications complexes, car il facilite l'ajout de nouvelles fonctionnalités tout en maintenant une architecture claire et organisée.

En résumé MVVM améliore la structure du code facilite les tests et la maintenance et offre une meilleure flexibilité dans le développement d'application.[19]

8- Relation MVVM avec le Frontend:

MVVM permet une bonne liaison entre la vue et la vue model. Donc cela signifie que les changements dans la vue reflètent dans la vue model et aussi le contraire. Cela simplifie la gestion d'état d'UI.

MVVM rendent le code plus modulaire et facile à tester car cette architecture sépare les données.

Et aussi frontend exploite MVVM pour crée des interfaces réactive ou les données qui changent entraînent des mises a jours automatique de l'interface utilisateur.

En résumé, l'architecture MVVM facilite la collaboration entre le frontend (vue et viewmodel) et le backend (modèle). Elle permet une séparation claire des responsabilités et améliore la maintenabilité de l'application.

9- SQLite comme backend :

Dans le cadre de la création d'un logiciel de **GMAO** (gestion de maintenance assistée par ordinateur), **SQLite** est une base de données légère, idéale pour des systèmes embarqués ou des applications avec des besoins en base de données locales et peu complexes.

9.1 Simplicité et légèreté

- **Facilité d'utilisation** : SQLite est extrêmement facile à installer et à utiliser. Aucune configuration serveur n'est nécessaire, il suffit d'intégrer le fichier de base de données à l'application.
- **Pas de processus serveur** : Contrairement à d'autres bases de données qui nécessitent un processus serveur en arrière-plan, SQLite fonctionne directement dans l'application qui l'utilise, ce qui réduit la complexité.

9.2 Architecture de SQLite dans un logiciel GMAO

a) Base de données locale :

SQLite fonctionne avec un fichier unique pour stocker l'intégralité de la base de données. Cela inclut les tables, les index, les vues, et les triggers nécessaires pour gérer les données de maintenance.

Ce fichier de base de données est souvent stocké localement sur l'appareil où le logiciel de GMAO est installé (ordinateur ou appareil mobile). Chaque application peut avoir son propre fichier SQLite ou partager une base de données si plusieurs instances de l'application accèdent aux mêmes données.

b) Requêtes SQL dans l'application GMAO :

Le backend du logiciel de GMAO utilise des requêtes SQL pour manipuler les données dans SQLite. Ces requêtes sont utilisées pour :

- Créer des ordres de travail : Lorsque des équipements nécessitent une intervention.
- Suivre l'historique de maintenance : Récupérer les interventions passées pour une machine donnée.
- Gérer les stocks de pièces détachées : Suivre la quantité de pièces restantes et générer des alertes lorsque le stock est faible.
- Générer des rapports : Par exemple, un rapport sur les temps d'arrêt des machines ou sur la performance des techniciens.

c) Synchronisation et gestion des données :

Mode hors ligne : Un des avantages de SQLite est qu'il peut être utilisé pour stocker des données localement et permettre au logiciel de GMAO de fonctionner en mode hors ligne. Cela est particulièrement utile pour des techniciens travaillant dans des environnements où la connectivité réseau est limitée.

Synchronisation des données : Dans un logiciel GMAO utilisant SQLite, les données peuvent être synchronisées avec une base de données centrale (comme MySQL, PostgreSQL, ou une base de données cloud) lorsque la connexion est disponible. Cela se fait via des APIs REST ou d'autres mécanismes de synchronisation de données.

9.3 Tables et modèles dans une GMAO

Dans une GMAO, plusieurs types de données sont stockés dans SQLite. Voici un exemple d'architecture de tables :

Table des équipements :

Contient les informations sur chaque équipement maintenu (nom, modèle, numéro de série, localisation, etc.).

Exemple de schéma :

Equipment		CREATE TABLE "Equipment" (idEquipment INTEGER primary key autoincrement, numEquipment text not null, type TEXT not null, marque TEXT not null, serie TEXT not null, aqyear INTEGER not null, fabYear INTEGER not null, price REAL not null, inventaire TEXT not null, eq_image TEXT , etat TEXT default unknown not null)
idEquipment	INTEGER	"idEquipment" INTEGER
numEquipment	text	"numEquipment" text NOT NULL
type	TEXT	"type" TEXT NOT NULL
marque	TEXT	"marque" TEXT NOT NULL
serie	TEXT	"serie" TEXT NOT NULL
aqyear	INTEGER	"aqyear" INTEGER NOT NULL
fabYear	INTEGER	"fabYear" INTEGER NOT NULL
price	REAL	"price" REAL NOT NULL
inventaire	TEXT	"inventaire" TEXT NOT NULL
eq_image	TEXT	"eq_image" TEXT
etat	TEXT	"etat" TEXT NOT NULL DEFAULT unknown


Tableau3.2 table d'équipement dans la base de données

Table des techniciens :

- Contient les informations sur les techniciens (nom, compétence, disponibilité, etc.).
- Exemple de schéma :

Employe		CREATE TABLE "Employs" (idEmploys integer primary key autoincrement, nomEmpl text not null, telephone text not null, email text not null, service text not null, description text not null, EmplImage TEXT)
idEmploye	integer	"idEmploye" integer
nomEmpl	text	"nomEmpl" text NOT NULL
telephone	text	"telephone" text NOT NULL
email	text	"email" text NOT NULL
service	text	"service" text NOT NULL

Name	Type	Schema
description	text	"description" text NOT NULL
EmplImage	TEXT	"EmplImage" TEXT

Tableau3.3 table des techniciens dans la base de données

Tous les tables qu'on a besoins dans notre logiciel a des caractéristique et un code spécifique comme des interventions, les ressources humains, le stock,...etc. (voir l'annexe).

9.4 Fonctionnement global de l'architecture SQLite :

1. **Requête envoyée** : L'application envoie une requête SQL via l'API SQLite.
2. **Analyse syntaxique** : Le parser de SQLite analyse la requête et la convertit en une structure interne.
3. **Optimisation** : L'optimiseur de requêtes analyse la meilleure façon d'exécuter la requête, en tenant compte des index, des statistiques de la base de données et des opérations à effectuer.
4. **Exécution** : Le moteur d'exécution exécute la requête. Si des données doivent être modifiées ou lues, elles sont récupérées via le pager et éventuellement stockées en cache pour un accès plus rapide.
5. **Accès disque** : Les données sont finalement récupérées ou stockées dans le fichier de la base de données.

6. **Résultat retourné** : Le résultat de la requête (dans le cas d'une sélection, par exemple) est renvoyé à l'application.

9.5 Fonctionnement de MVVM avec SQLite

a) Model (Modèle)

Rôle : Le modèle est responsable de **gérer les données** et de fournir les méthodes pour interagir avec la base de données SQLite. Il contient des classes qui représentent les entités et des méthodes pour exécuter des requêtes SQL via SQLite.

b) ViewModel (Modèle de Vue)

Rôle : Le ViewModel gère les données et la logique métier entre la vue et le modèle. Il appelle les méthodes du modèle pour accéder aux données et les formater ou les préparer avant de les exposer à la vue.

Le ViewModel ne doit pas connaître les détails spécifiques de l'interface utilisateur, mais il contient la logique pour répondre aux événements (comme les clics de boutons) ou pour fournir des données à afficher.

c) Interaction avec SQLite via le modèle : Le ViewModel fait appel aux méthodes du modèle pour lire et écrire dans SQLite. Il expose les données au format nécessaire pour la vue (par exemple, sous forme de listes ou d'objets liés aux widgets UI).

d) View (Vue) :

Rôle : La vue est **responsable de l'interface utilisateur**. Elle observe le ViewModel pour obtenir les données et les afficher, sans interagir directement avec le modèle ou SQLite.

La vue écoute les événements utilisateur (comme les clics de boutons) et transmet ces événements au ViewModel. Elle peut être liée au ViewModel pour que l'UI se mette à jour automatiquement en fonction des données fournies par le ViewModel. **d) Interaction avec le ViewModel** : La vue utilise des liaisons de données (data binding) ou des événements pour obtenir et mettre à jour les données via le ViewModel.

10-Technologies utilisée :

Pour la création du logiciel on a utilisé langage de programmation moderne qui est « KOTLIN ».

10.1 Kotlin : est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java, JavaScript, et vers plusieurs plateformes. Son développement provient principalement d'une équipe de programmeurs chez JetBrains basée à Saint-Petersbourg en Russie (son nom vient de l'île de Kotlin, près de St. Pétersbourg).[20]

Google annonce pendant la conférence Google I/O 2017 que Kotlin devient le second langage de programmation officiellement pris en charge par Android après Java. Le 8 mai 2019, toujours lors de la conférence Google I/O, Kotlin devient officiellement le langage de programmation voulu et recommandé par le géant américain Google pour le développement des applications Android.[21]

Pivotal Software annonce le 4 janvier 2017 le support officiel de Kotlin sur la cinquième version du Framework Spring



Figure 3.4 logo de KOTLIN

10.2 Avantage de KOTLIN :

KOTLIN est totalement compatible avec l'ancien langage JAVA qui est l'un le plus utilisé dans le développement Android et backend, cela signifie que les projets existant déjà en JAVA peuvent facilement intégrer du code KOTLIN sans réécriture complète.

Et aussi on peut utiliser les bibliothèques JAVA en KOTLIN ce qui donne la facilité d'utilisation de KOTLIN.

KOTLIN est plus concise que JAVA, le code généralement nécessite moins de ligne pour accomplir les mêmes tâches qu'en JAVA ou d'autre langage.

La concision de Kotlin permet un code plus lisible, maintenable, et moins sujet aux erreurs humaines.

L'une des principales causes de bugs dans Java est l'exception NullPointerException (NPE), qui se produit lorsque l'on tente d'accéder à une référence d'objet null. Kotlin introduit une gestion stricte de la nullabilité, en forçant les développeurs à gérer explicitement les références nulles.

Ce système réduit considérablement le risque de bugs liés à la gestion des nulls, offrant une plus grande sécurité et stabilité du code.

Kotlin est considéré comme facile à apprendre, surtout pour les développeurs Java, car sa syntaxe est simple et son modèle de programmation est intuitif. Il élimine beaucoup de complexités que les développeurs rencontrent souvent en Java.

10.3 L'utilité SQLite dans la base de données et le backend :

SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL et des propriétés ACID.

Contrairement aux serveurs de bases de données traditionnels, tels que MySQL ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.[22]

CHAPITRE 3 conceptions et architecture du système

Le créateur de SQLite, a choisi de mettre cette bibliothèque ainsi que son code source dans le domaine public, ce qui permet son utilisation sans restriction aussi bien dans les projets open source que dans les projets propriétaires. Le créateur ainsi qu'une partie des développeurs principaux de SQLite sont employés par la société américaine Hwaci.

SQLite est le moteur de base de données le plus utilisé au monde, grâce à son utilisation :

- dans de nombreux logiciels grand public comme Firefox, Skype.
- dans certains produits d'Apple, d'Adobe et de McAfee,
- dans les bibliothèques standards de nombreux langages comme PHP ou Python.

De par son extrême légèreté, il est également très populaire sur les systèmes embarqués, notamment sur la plupart des smartphones et tablettes modernes : les systèmes d'exploitation mobiles iOS, Android et Symbian l'utilisent comme base de données embarquée. Au total, on peut dénombrer plus d'un milliard de copies connues et déclarées de la bibliothèque.

10.4 Avantages de SQLite :

- Simplicité : Pas besoin de serveur externe ou de configuration supplémentaire.
- Fonctionne hors ligne : Idéal pour les scénarios où la connexion réseau est limitée ou inexistante (par exemple, des techniciens travaillant dans des environnements isolés).
- Installation rapide : L'application est autonome et SQLite est facile à déployer.

11- Conclusion :

Dans ce chapitre, nous avons détaillé la conception technique et l'architecture logicielle de notre application de GMAO.

Chapitre 4 :
Réalisation de l'application GMAO

1-Introduction :

Ce quatrième chapitre est consacré à la mise en œuvre pratique du logiciel de gestion de maintenance assisté par ordinateur.

Après avoir cité les différents besoins et les outils qu'on a utilisé pour rendre ce logiciel en marche dans le chapitre précédent, maintenant c'est la réalisation et les tests de ce système.

Ici on va expliquer comment le logiciel fonctionne à partir des captures d'écran pour bien expliquer le rôle de ce système.

2- Fenêtre d'authentification :

Pour accéder à cette fenêtre il suffit que faire un double clic gauche sur l'icône de l'application qui se trouve dans le bureau. Par la suite il faut saisir les informations (nom d'utilisateur et le mot de passe) pour rentrer à l'application.

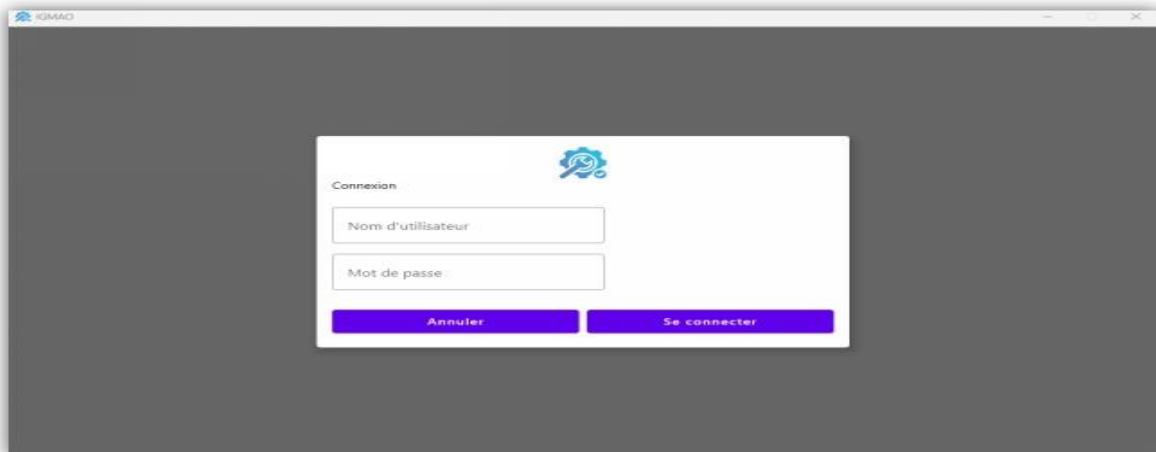


Figure 4.1 fenêtre d'authentification

3-Fenêtre d'accès principale :

Après avoir exécuté la première étape, c'est la deuxième qui suit c'est celle de l'affichage de la fenêtre d'accès qui affiche les différents besoins d'utilisateur tels que le « Dashboard », « Equipements », « Stocks », « OI », « DI », « RH », et « Notifications ».



Figure 4.2 page d'accueil

4-Fenêtre «Dashboard» :

Elle contient les statistiques du stock et les différents équipements, les traduit en courbes graphiques, diagramme circulaire, et histogramme.

Commençant avec la courbe graphique : représente le nombre de pièces dans le stock en fonction du temps (date)



Figure4.3 courbe graphique sur les statistiques du stock

2eme figure qui est le diagramme circulaire qui représente l'état de facture des pièces (payée, impayée, en cours de paiement, annulé).

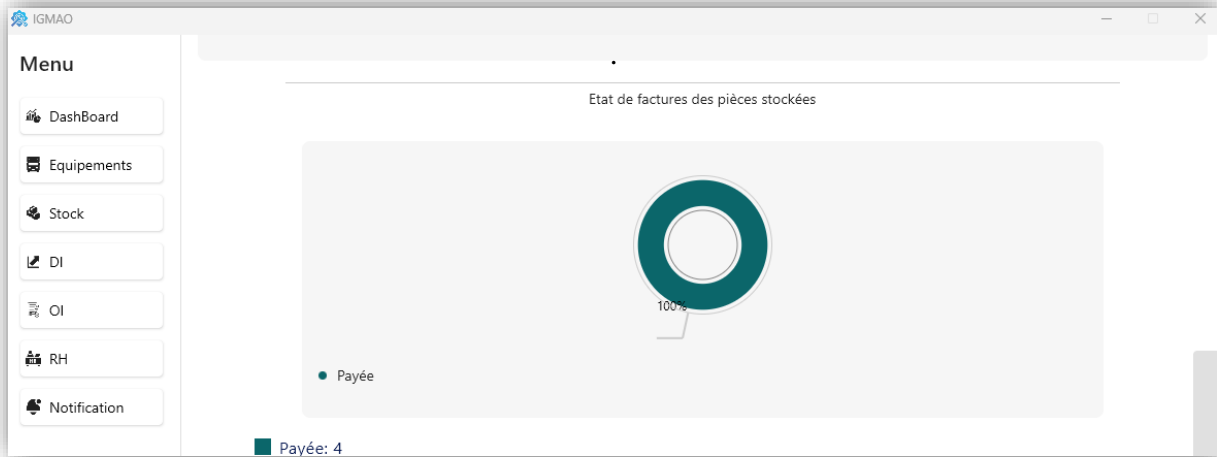


Figure 4.4 Diagramme circulaire de l'état des factures des pièces

Cette 3ème figure représente les statistique des équipements présentées par un diagramme circulaire affichant si un équipement est en marche ou en maintenance ou en panne.

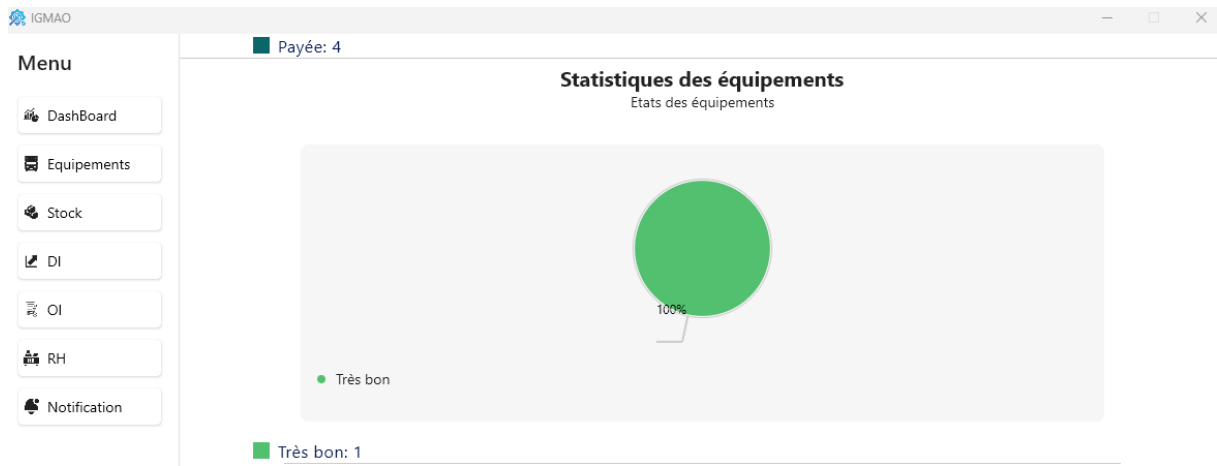


Figure 4.5 diagramme circulaire sur l'état des équipements

La dernière figure dans le Dashboard c'est les barres graphiques qui présentent le nombre d'intervention par équipement (nombre d'intervention en fonction des équipements).

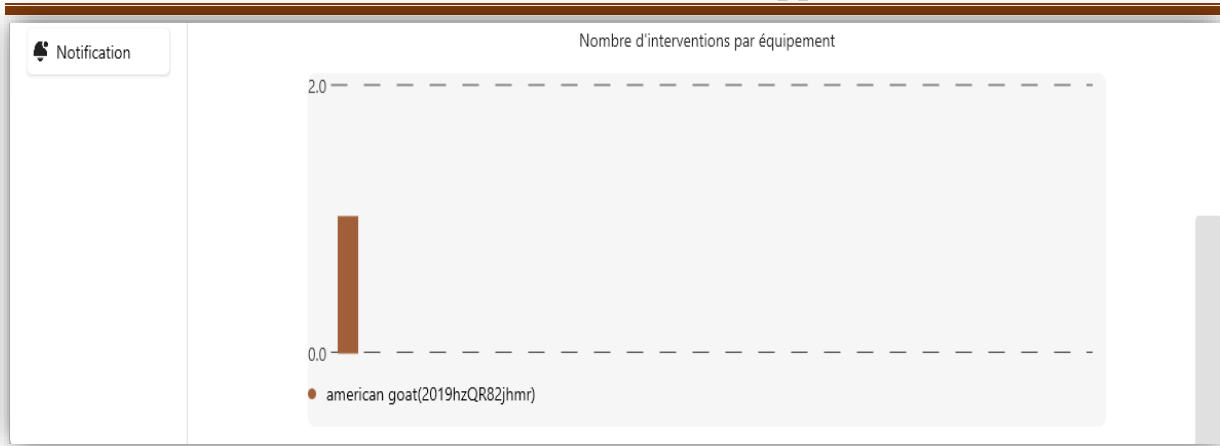


Figure 4.6 histogramme des interventions sur les équipements

5 Fenêtre « équipements » :

C'est ici qu'on peut voir tous les équipements qu'on a dans une entreprise avec capacité de la suppression et modification et aussi voir détails d'un équipement, Et de la recherche.

Quand on ouvre cette fenêtre elle nous affiche directement les équipements qu'on a avec une petite fiche qui affiche le numéro de chassé, type, marque, série d'un équipement.

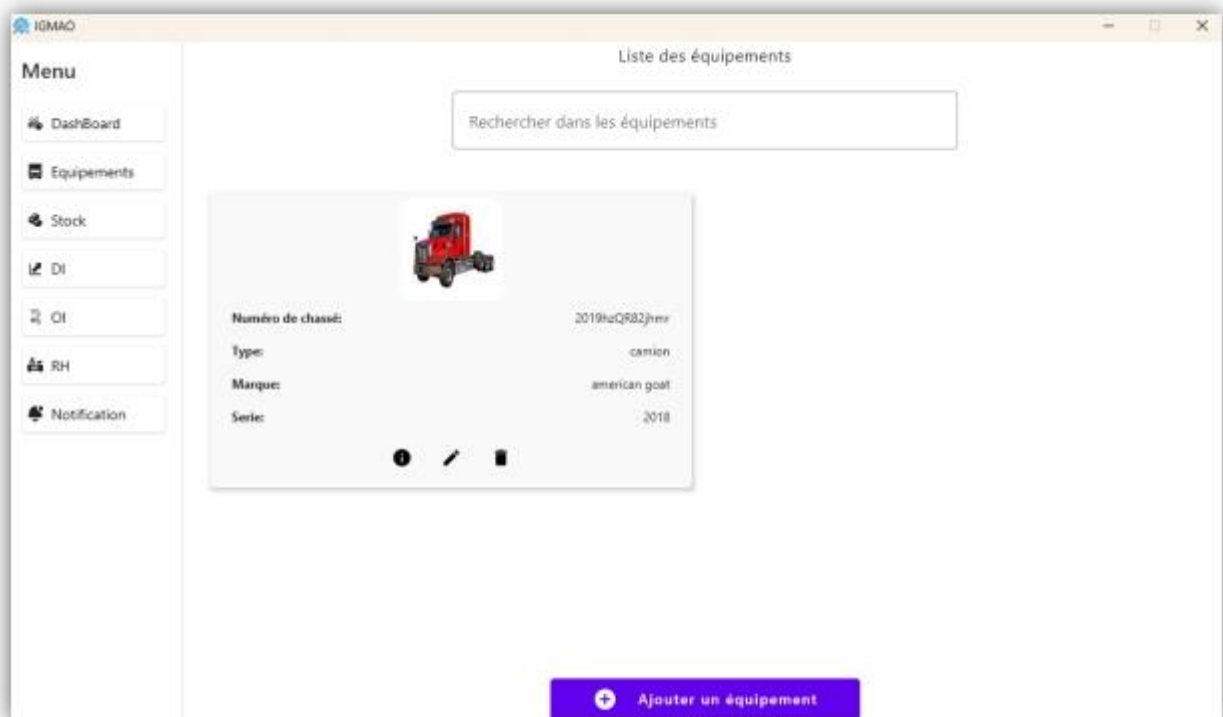


Figure4.7 fenêtre des équipements

5.1 Fenêtre d'ajout d'un équipement :

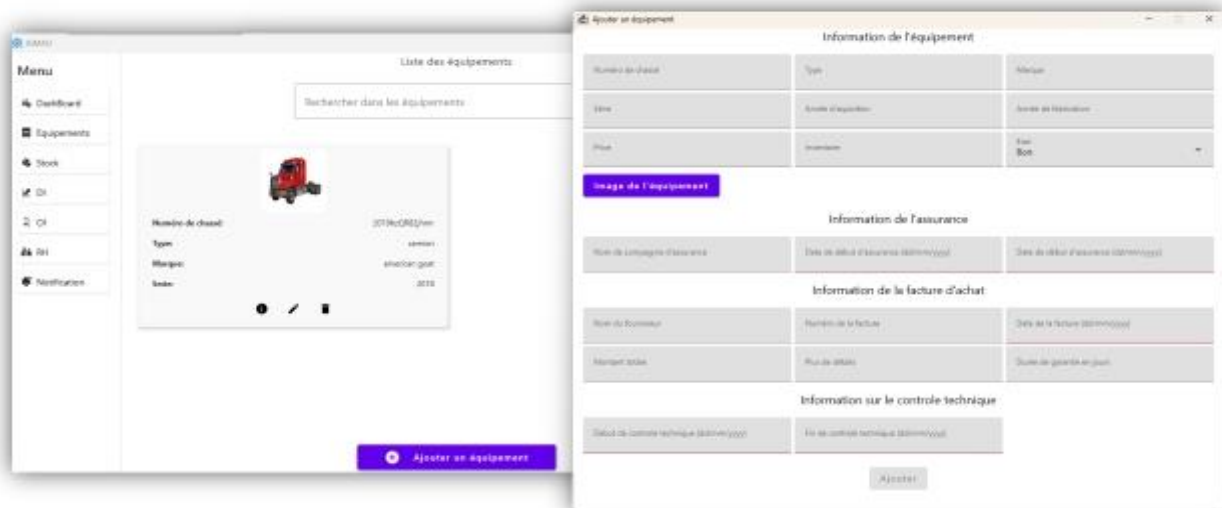


Figure 4.8 fenêtre d'ajout d'un équipement

Cette fenêtre est responsable d'ajout d'un équipement, mais avant de les ajouter il faut remplir les informations de l'équipement tels que :

Numéro de châssé, type (camion, machine, fourgon,...), marque (Renault, Volvo, Ford,...), série, année d'acquisition, année de fabrication, prix, inventaire et son état (bon, très bon, en maintenance, en panne).

Ensuite faut insérer les informations sur l'assurance de l'équipement comme nom de compagnie d'assurance, date de début et de fin d'assurance

Et aussi la facture d'achat de cette Equipment qui contient :

Nom de fournisseur, numéro de la facture, date de la facture, montant totale (contient tous les dépenses sur l'assurance et tout,...), plus de détails si y'on a et vers la fin on trouve la garantie en jours

Après l'assurance c'est le control technique de l'équipement faut insérer la date de début et de fin de control technique

5.2 Détails sur les équipements :



Figure 4.9 fenêtre de détails sur les équipements

En cliquant sur plus de détails sur l'équipement cette fenêtre sera affichée qui nous donne les informations sur l'équipement concernée comme :

- Identifiant : est un numéro aléatoire jamais répéter sur les autres équipements
- Numéro de châssé : qui est le numéro de l'équipement sortie de la maison qui a fabriquée l'équipement.
- Type : camion, fourgon, voiture, machine,...
- Marque
- Série : c'est l'année ou l'usine d'équipement a sorti son premier modèle.
- Année d'acquisition : c'est l'année quand l'entreprise a acheté cet équipement
- Année de fabrication
- Prix de l'équipement et l'inventaire
- Etat d'équipement, et l' **MTBF** : chaque équipement possède son propre MTBF qui est :

$MTBF = \sum \text{temps de bon fonctionnement} / \text{nombre de défaillances}$.

5.2.1 Fiche d'assurance d'un équipement :

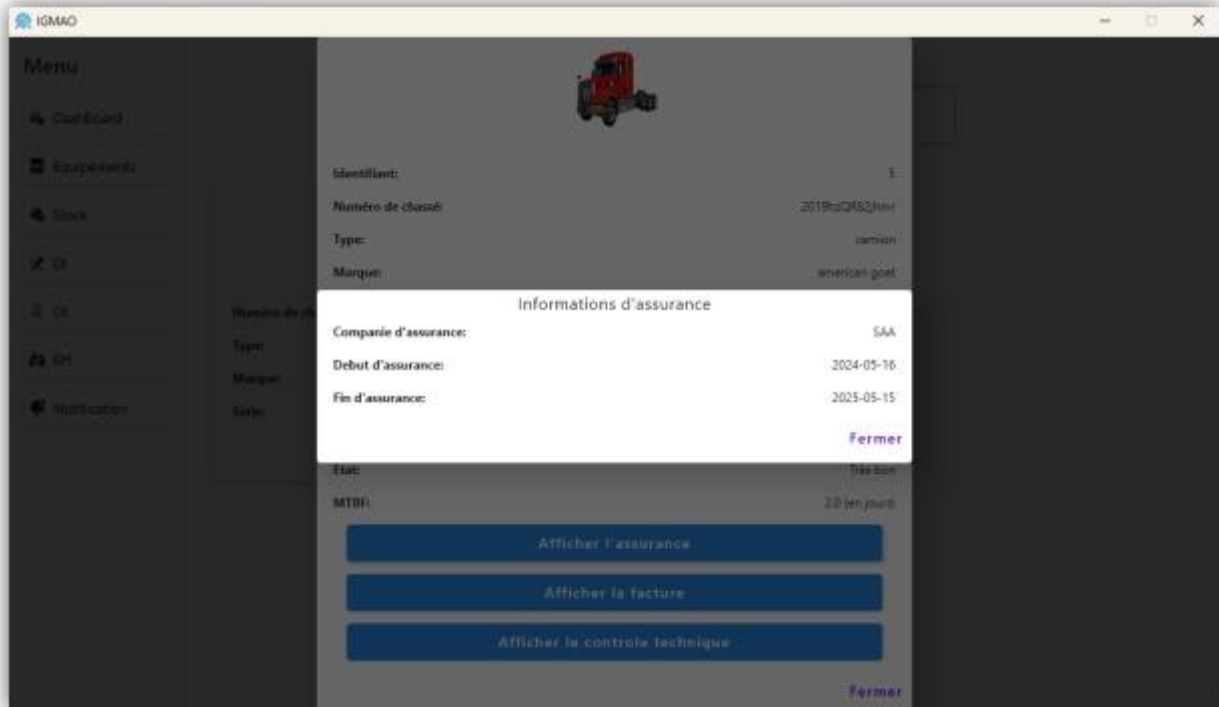


Figure 4.10 fenêtre d'assurance

En cliquant sur détails d'équipement c'est ici ou on peut voir l'affiche qui concerne l'assurance, et on trouve dedans :

- Compagnie d'assurance
- Date début d'assurance
- Date fin d'assurance
- Jours restant pour que l'assurance expire

5.2.2 Facture d'achat :

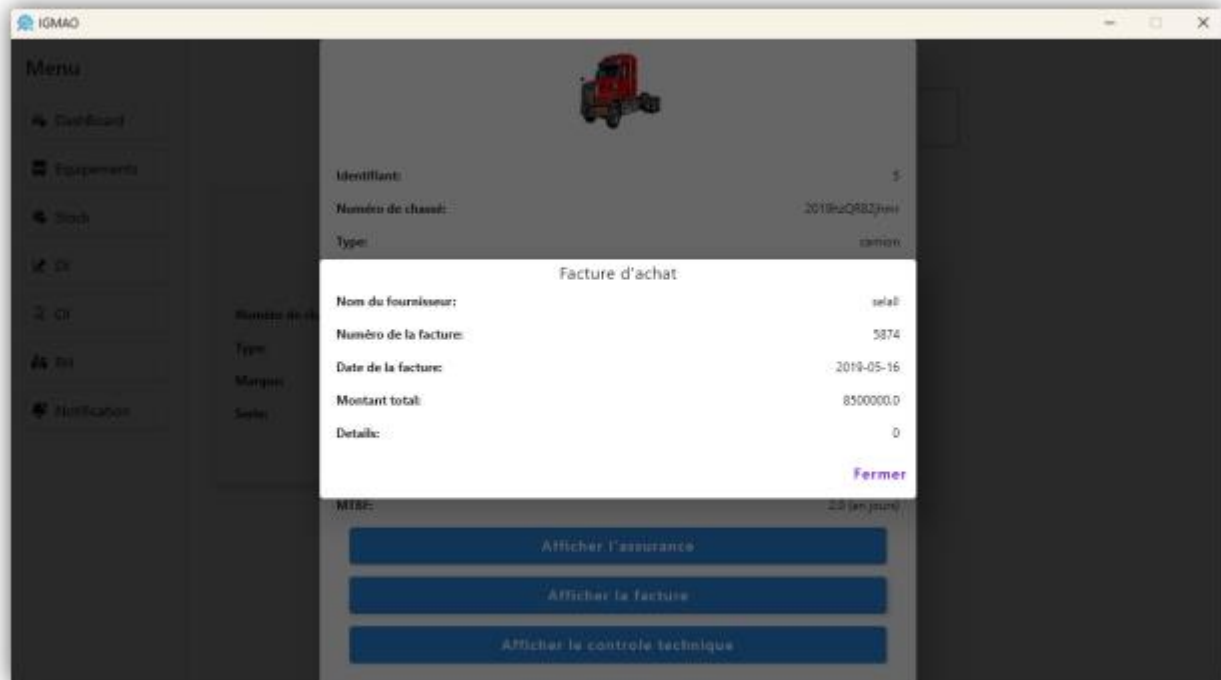


Figure 4.11 facture d'achat

Restons toujours dans la fenêtre détails d'équipement, on trouve la facture d'achat d'équipement qui contient :

- Nom du fournisseur (personne ou compagnie qui nous a vendu l'équipement)
- Numéro de la facture
- Date de facture
- Montant total
- Autre détails

5.2.3 Fiche de contrôle technique :

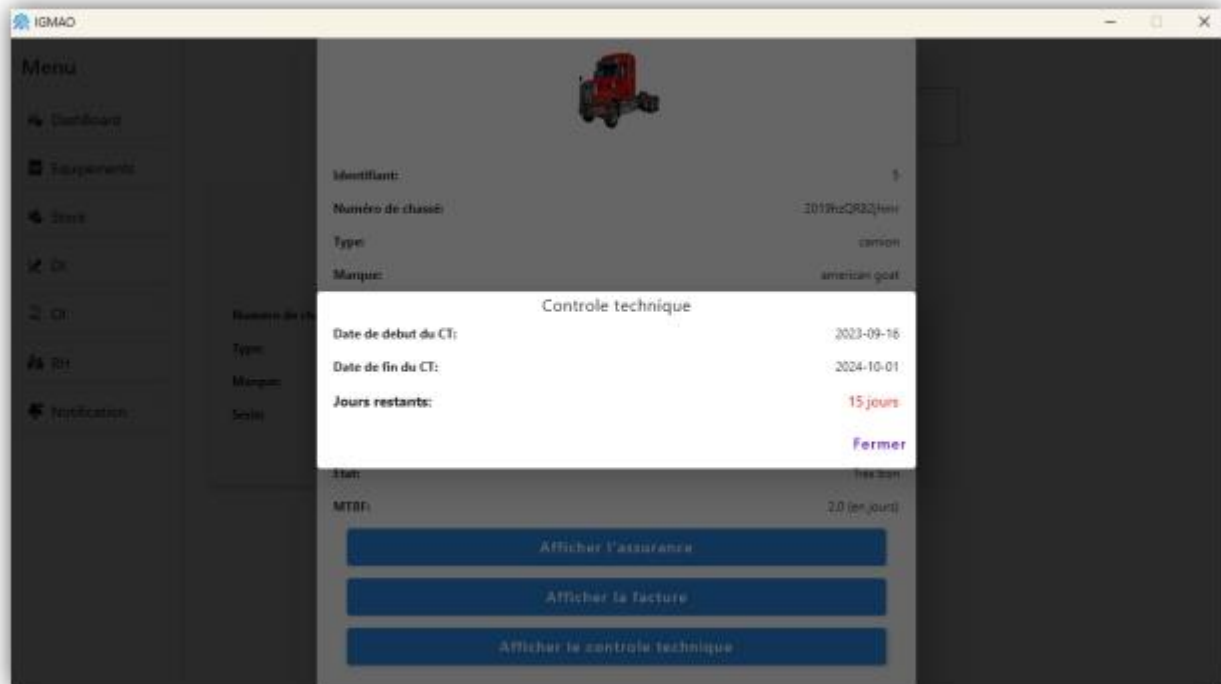


Figure 4.12 contrôle technique

Le dernier bouton dans détails d'Equipment c'est le control technique ou on trouve :

- Date de début de control technique
- Date de fin de control technique
- Jours restants pour le control technique (si la durée qui nous reste le système va l'afficher sinon non)

6-Fenêtre de stock :

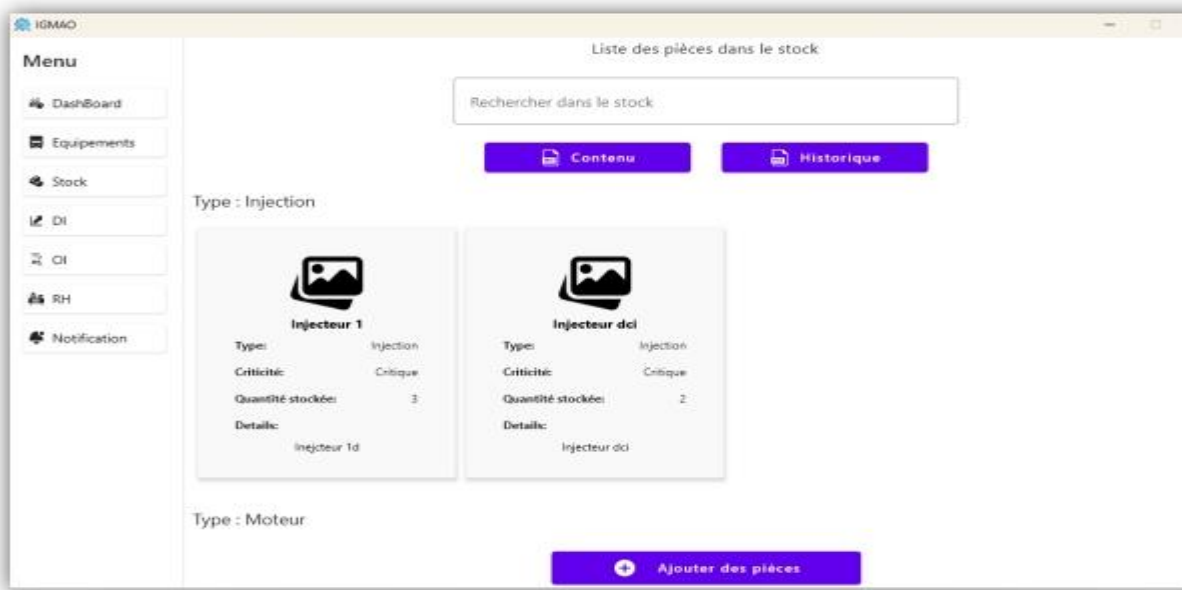


Figure4.13 stock des pièces

Cette fenêtre est responsable de nous afficher les pièces qu'on a dans le stock avec capacité d'ajout, de recherche et voir l'historique et le contenu de stock

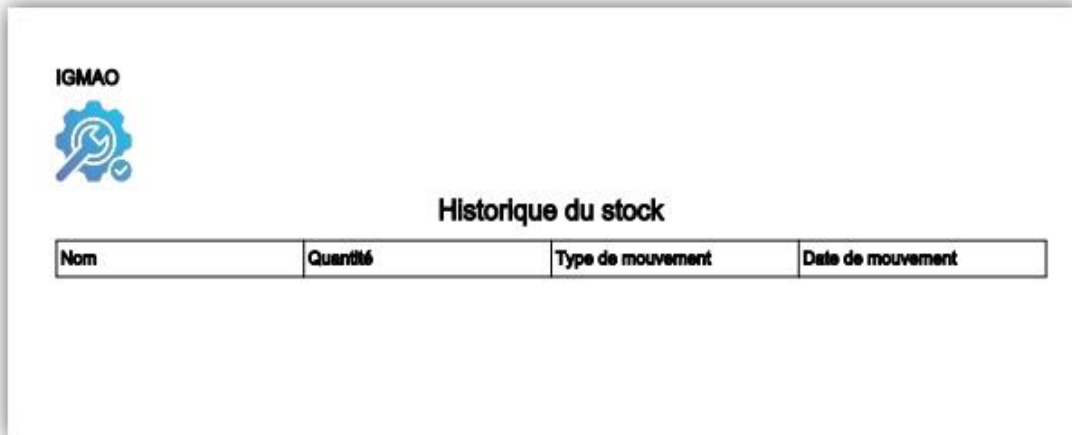
6.1 Contenu de stock :

Nom de la pièce	Quantité dans le stock
Injecteur 1	3
Injecteur dci	2
Moteur	1
moteur scania	2

Figure 4.14 contenu de stock

Dans le contenu de stocke on trouve tous les pièces qu'on a avec leurs nom et quantités

6.2 Historique du stock :



IGMAO

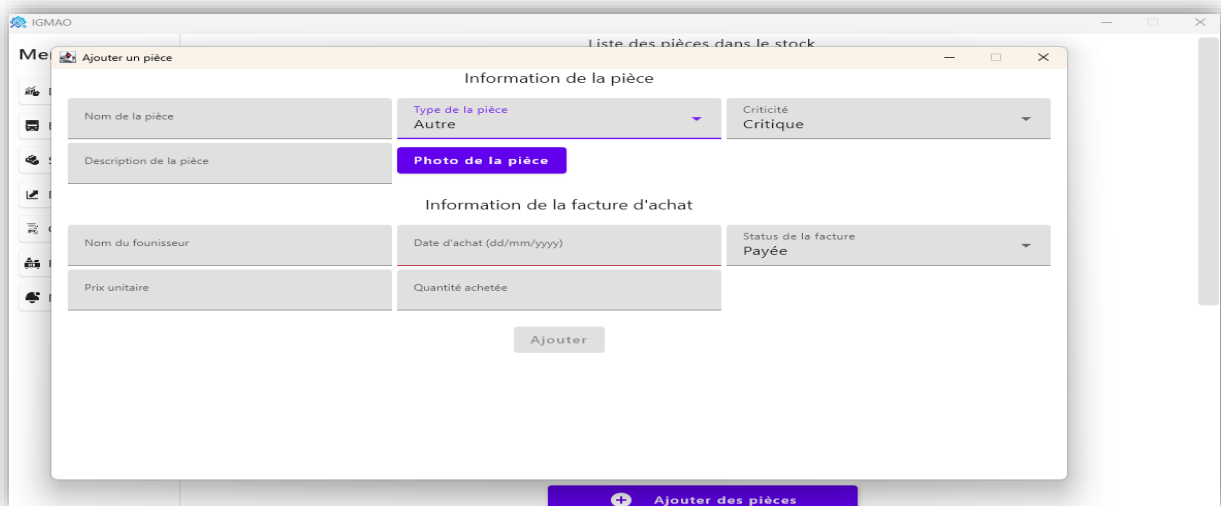
Historique du stock

Nom	Quantité	Type de mouvement	Date de mouvement
-----	----------	-------------------	-------------------

Figure 4.15 historique du stock

Cette fenêtre représente l'historique de stock ou le mouvement des pièces, quand on les a utilisés et aussi quand on les a achetées avec la quantité utilisée et leurs noms. Type de mouvement c'est entrée ou sortie.

6.3 Informations sur la pièce :



IGMAO

Liste des pièces dans le stock

Ajouter une pièce

Information de la pièce

Nom de la pièce

Type de la pièce
Autre

Criticité
Critique

Description de la pièce

Photo de la pièce

Information de la facture d'achat

Nom du fournisseur

Date d'achat (dd/mm/yyyy)

Status de la facture
Payée

Prix unitaire

Quantité achetée

Ajouter

Ajouter des pièces

Figure4.16 fenêtre d'ajout d'une pièce

Ici en cliquant sur ajouter des pièces ce tableau sera affichée pour saisir les informations des pièces qu'on veut les ajouter comme :

- Nom de la pièce
- Type de la pièce (moteur, suspension,...)
- Criticité (critique, important, modéré)
- Description de pièce
- Photo de la pièce

Après avoir remplir le 1^{er} tableau on saute vers le 2eme qui est la facture d'achat de la pièce qui contient :

- Nom de fournisseur
- Date d'achat
- Statut de la facture (payée, non payée, en cours de paiement)
- Prix unitaire et quantité achetée

7-Fenêtre de demande d'intervention «DI» :

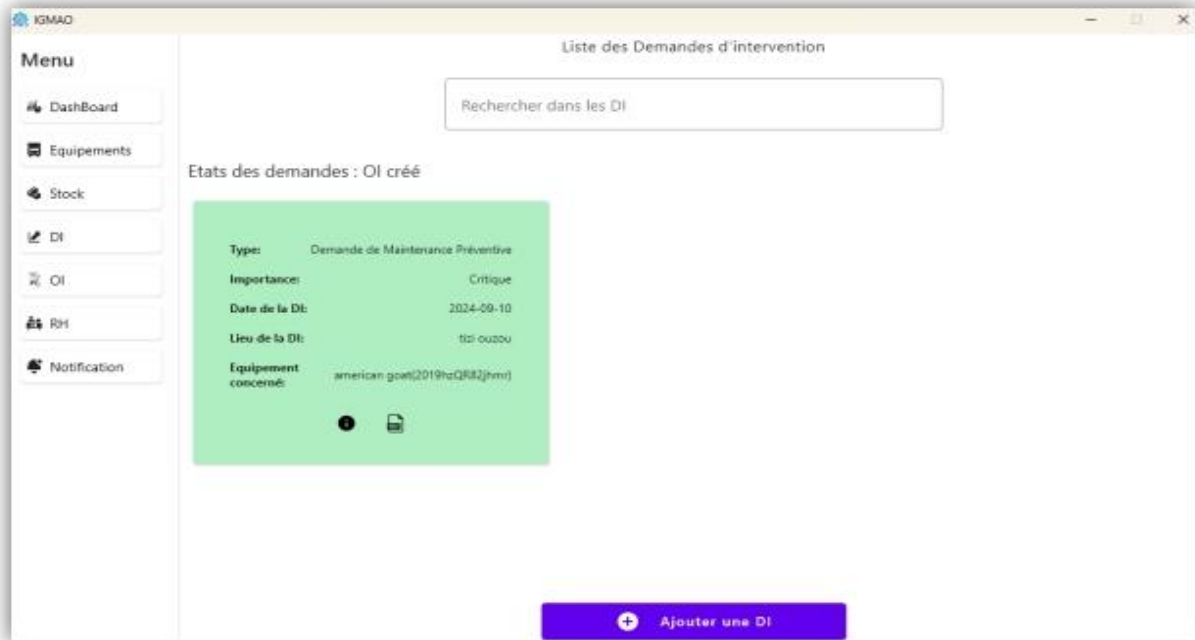


Figure 4.17 fenêtre de DI

Fenêtre demande d'intervention est responsable d'afficher les demandes qu'on a déjà fait et d'ajouter des autres demande en cas de besoin. Une fois qu'on a créé une on peut voir les détails de la demande et le rapport PDF de cette demande d'intervention, et aussi il affiche le statut de la demande si elle est confirmée ou en attente de confirmation si elle est en attente elle s'affiche en rouge avec un bouton confirmé au-dessus.

Dans cette fenêtre on peut voir qui est l'équipement concernée, le lieu la date de demande, l'importance, et le type de la demande (préventive, corrective,...).

7.1 Ajout d'une demande d'intervention :

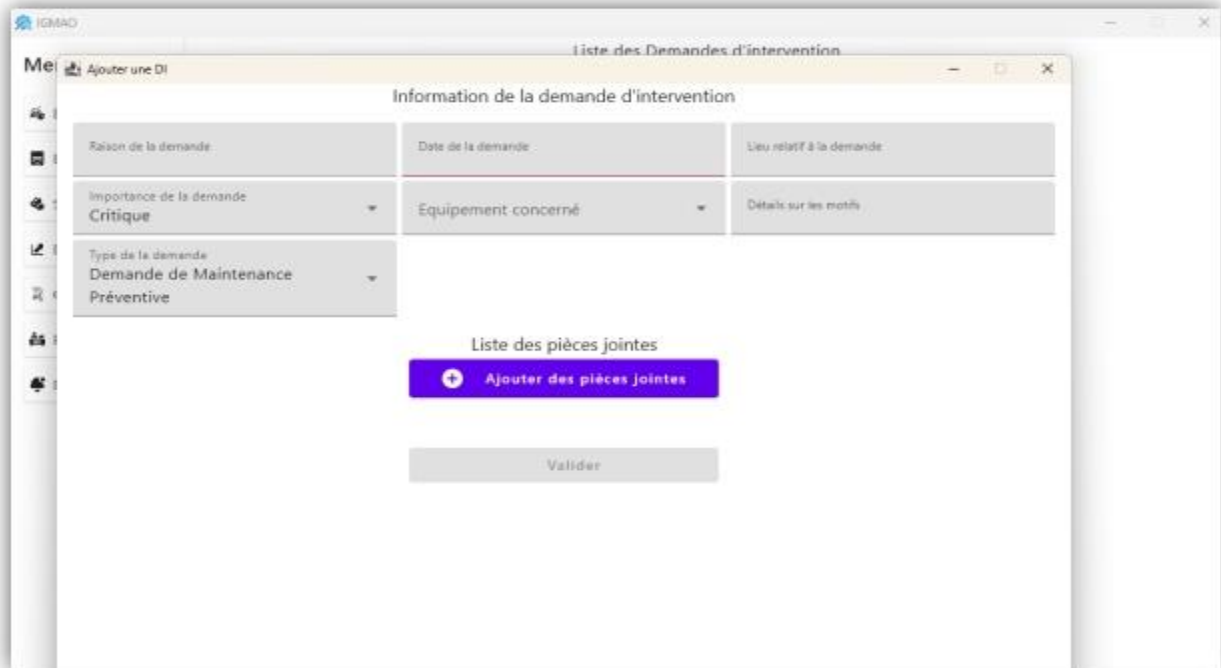


Figure 4.18 fenêtre d'ajout d'une DI

Une fois que tu cliques sur le bouton ajouter une demande cette figure sera affichée, et pour qu'on puisse ajouter une il nous faut d'abord remplir le tableau qui contient :

- Raison de la demande (panne,...)
- Date de la demande
- Lieu relatif à la demande
- Importance de la demande (critique, important, modéré)
- Equipement concernée : une fois tu cliques sur il te donne les équipements que tu as
- Détails sur les motifs
- Type de la demande (demande de réglage, de sécurité, demande de maintenance préventive, corrective,...)
- Photo jointe sur les pièces endommagées si on a

7.2 Détails de la demande :

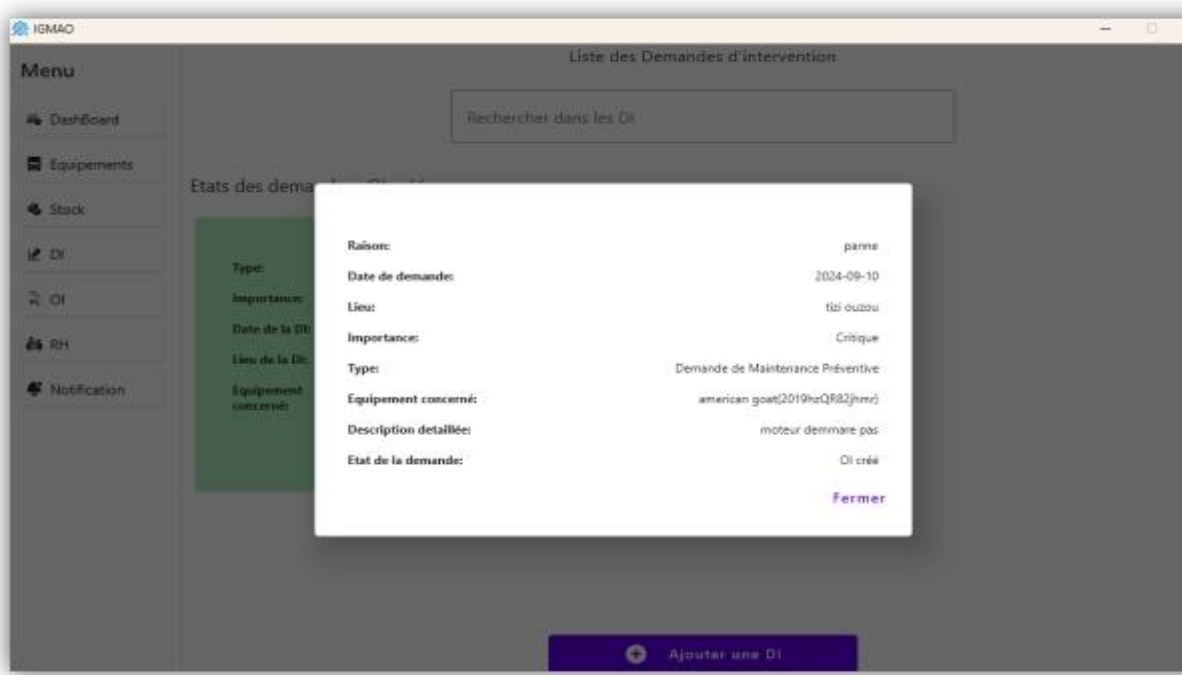



Figure 4.19 détails de DI

Ici c'est les détails de la demande qui nous donne l'historique des informations qu'on a saisi lors de remplissage de la demande d'intervention qui contient :

- Raison de la demande
- Date
- Lieu
- Importance
- Type
- Équipement concernée
- Description détaillées
- Etat de la demande (confirmé ou en attente)

6.3 Document de la demande :

IGMAO



Demande d'intervention

Identifiant de la demande	13
Raison	panne
Date de demande	10/09/2024
Lieu	tizi ouzou
Importance	Critique
Type d'intervention	Demande de Maintenance Préventive
Équipement	american goat(2019hzQR82jhmr)
Description	moteur demmare pas
État de la demande	OI créé




Figure 4.20 informations sur DI

7.4 Création d'un OI :

The screenshot shows a web application window titled 'Ajouter un OI' (Add Intervention Order) within a larger window titled 'Liste des Demandes d'intervention'. The form is titled 'Ordre d'intervention' and contains the following elements:

- A header bar with the title 'Ordre d'intervention'.
- A table with three columns: 'DI', 'Description', and 'Durée estimée en jours'. The 'DI' column contains the value '14(Demande de Maintenance Corrective)'. The 'Description' and 'Durée estimée en jours' columns are empty.
- A section titled 'Liste des pièces nécessaire' with a purple button labeled '+ Ajouter une pièce'.
- A section titled 'Liste des employés affectés' with a purple button labeled '+ Affecter des employés'.
- A section titled 'Etapas de resolution du problème' with a purple button labeled '+ Ajouter les étapes de resolution'.
- A grey button labeled 'Valider' at the bottom.

Figure 4.21 création d'un OI

Une fois que on a confirmé la demande il faut crée un OI pour que la demande tourne a un OI et pour qu'on puisse le crée il faut remplir les données suivant :

- Descriptions de la demande
- Durée estimée en jours (durée de maintenance)
- Liste des pièces nécessaire
- Listes des employés affectés pour réaliser une tache
- Etapes de résolution du problème

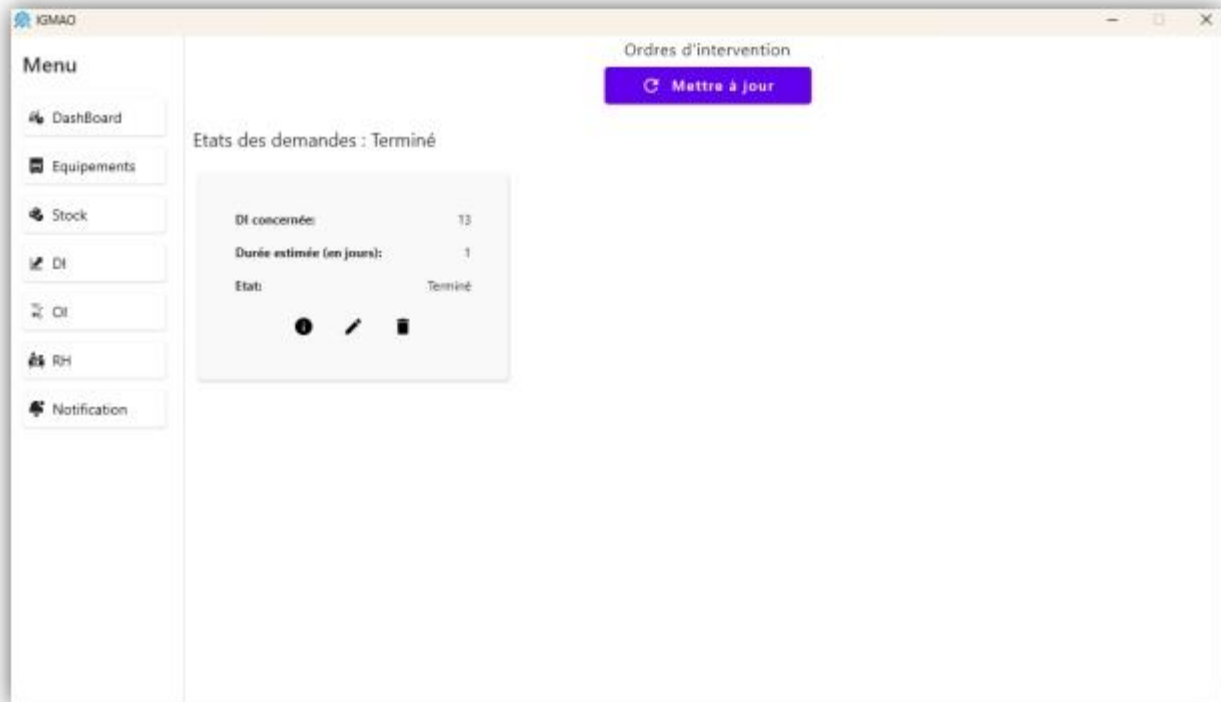
8-Fenêtre d'ordres d'intervention «OI» :

Figure 4.22 fenêtre OI

Fenêtre responsable d'afficher les ordres d'interventions et leur état si en cours, terminé, annuler avec possibilité de modifier ou supprimer et voir détails de l'ordre d'intervention

En haut y a un bouton « mettre a jour » pour que toutes les demandes crée par les différents utilisateurs s'affichent en quelque seconde après avoir appuyé sur.

9-Fenêtre «RH» :

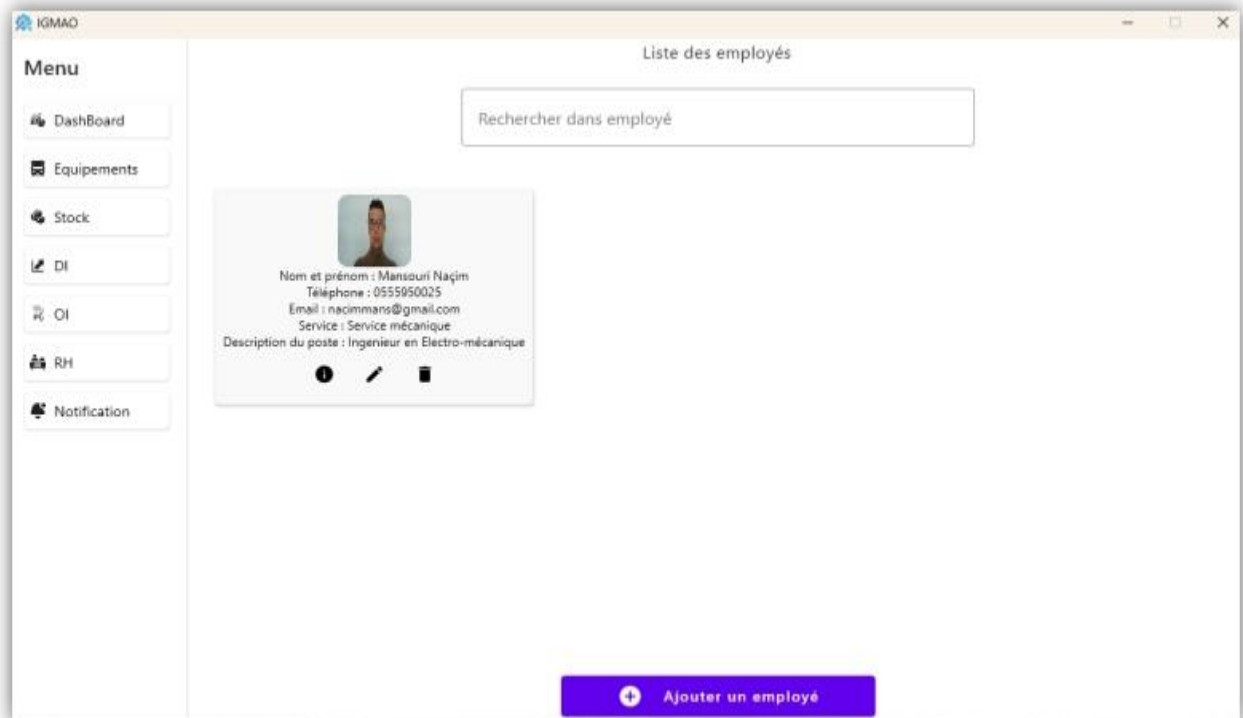


Figure 4.23 fenêtre RH

Fenêtre ressource humains : quand on appuie sur le bouton RH cette fenêtre sera affichée, elle affiche la liste des employées avec leur :

- Nom et prénom
- Téléphone
- Email
- Service de travail
- Description du poste
- Une photo d'employée

Avec possibilité d'ajout, suppression, modification et voir détails d'un employée.

9.1 Fenêtre d'ajout d'un employeur :

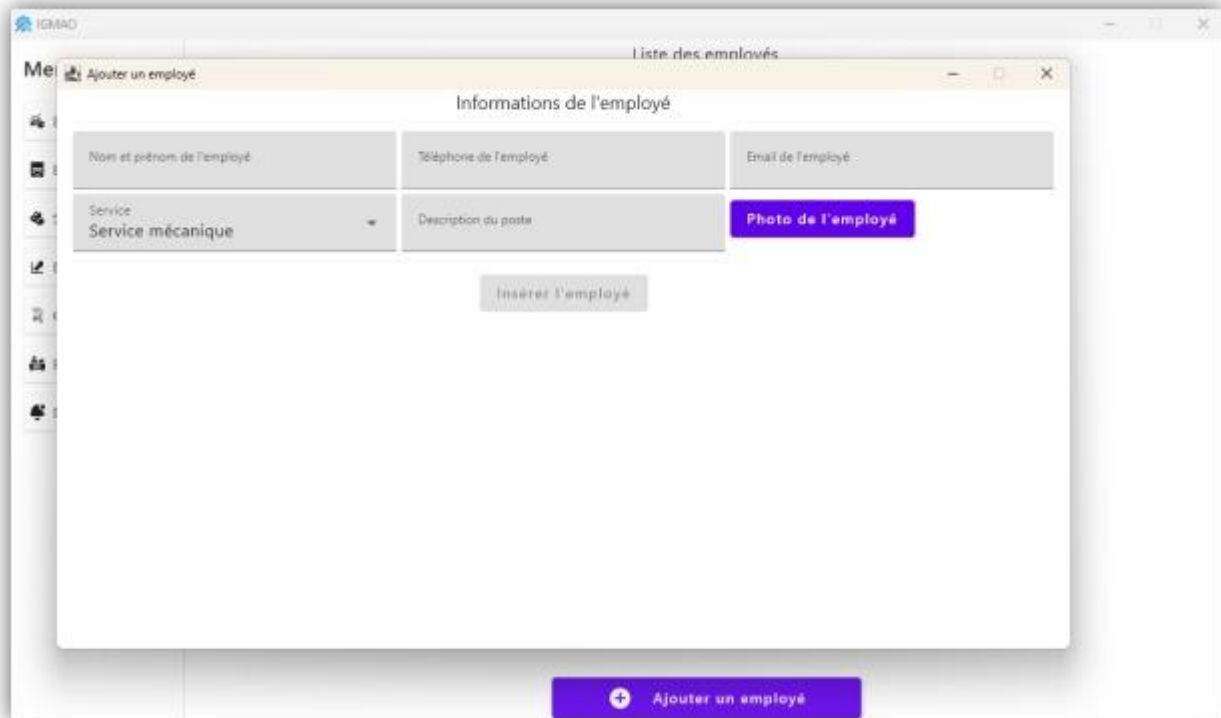


Figure 4.24 fenêtre d'ajout d'un employé

Pour ajouter un employé il faut remplir les informations qui suivent :

- Nom et prénom de l'employé
- Téléphone de l'employé
- Email d'employé
- Service (mécanique, électrique, hydraulique,...)
- Description de poste
- Photo de l'employé

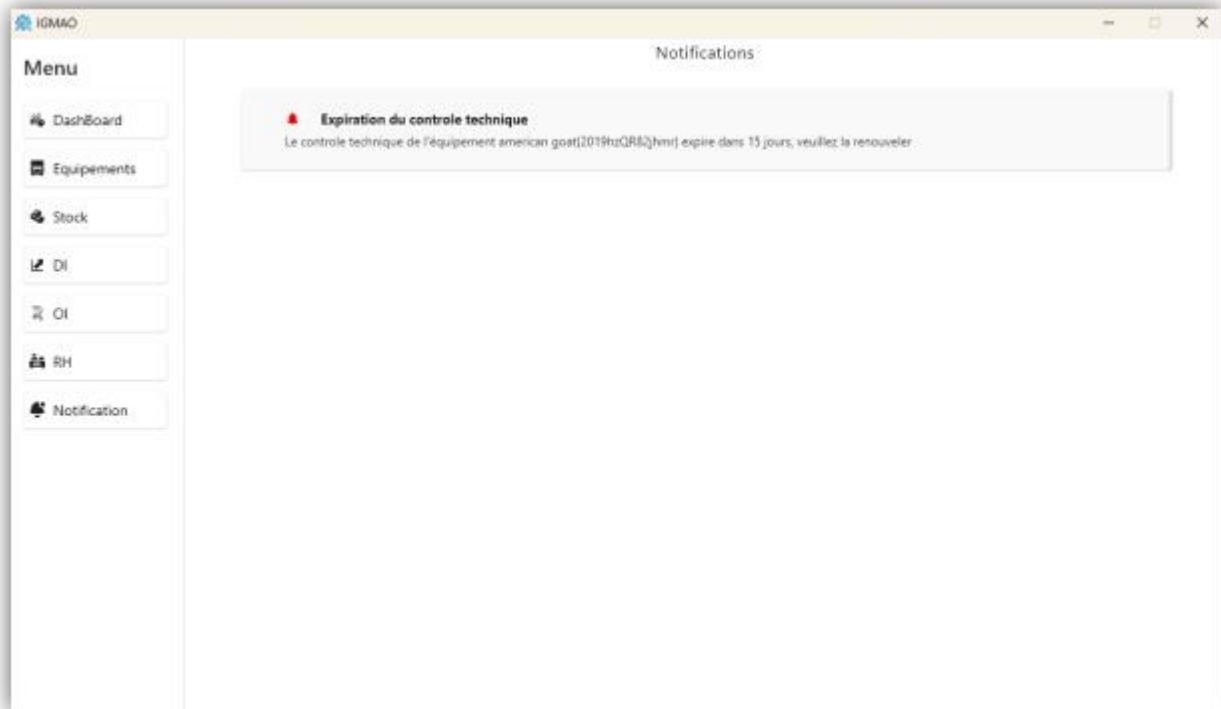
10-Notifications :

Figure 4.25 fenêtre de notifications

Cette fenêtre est responsable d'affichage les notifications concernant l'assurance de L'équipement et le control technique, une fois que la durée s'expire dans moins de 20 jours le Logiciel alerte l'utilisateur pour qu'il puisse les renouveler avant que ce soit trop tard.

11- Conclusion :

Nous concluons ce chapitre en présentant les résultats obtenus, les tests réalisés pour valider les fonctionnalités du logiciel et les éventuelles améliorations à envisager pour les versions futures.

Conclusion Général

Conclusion générale

Notre recherche avait pour but la conception et la réalisation d'un progiciel de maintenance pour la gestion de maintenance de matériel d'une entreprise.de travaux publics.

Avant d'avoir proposé notre solution, il nous a été impératif dans la phase étude de l'existant de bien comprendre et d'avoir une vision globale du système à automatiser, tout en fixant pour notre application ses limites de recueil des objets de gestion (fiches, données,...), ensuite on a proposé la solution informatique.

Dans le deuxième chapitre, en parlant des généralités sur la GMAO, nous nous sommes attachés à présenter l'importance de la GMAO et les conditions de la réussite de la GMAO ensuite on a cité quelques progiciels.

Le troisième chapitre, nous étudions l'architecture et la conception de notre logiciel GMAO, par la suite on a défini les technologies qu'on a suivies afin de réaliser notre logiciel.

Enfin dans le quatrième chapitre on a réussi d'expliquer le fonctionnement de notre logiciel en utilisant les captures d'écran.

Annexe :

Name	Type	Schema
Assurance		CREATE TABLE "Assurance" (idAssurance integer primary key autoincrement, idEquip integer references Equipment on update cascade on delete cascade, nomAssurance text not null, debutAssurance date not null, finAssurance date not null)
idAssurance	integer	"idAssurance" integer
idEquip	integer	"idEquip" integer
nomAssurance	text	"nomAssurance" text NOT NULL
debutAssurance	date	"debutAssurance" date NOT NULL
finAssurance	date	"finAssurance" date NOT NULL
ControleTech		CREATE TABLE ControleTech(id_ctrl integer primary key autoincrement , eq_id integer not null, debut_ctrl date not null , fin_ctrl date not null, foreign key (eq_id) references Equipment(idEquipment))
id_ctrl	integer	"id_ctrl" integer
eq_id	integer	"eq_id" integer NOT NULL
debut_ctrl	date	"debut_ctrl" date NOT NULL
fin_ctrl	date	"fin_ctrl" date NOT NULL
DemandeIntervention		CREATE TABLE DemandeIntervention id INTEGER PRIMARY KEY AUTOINCREMENT, raison TEXT NOT NULL, dateDemande DATE NOT NULL, lieu TEXT NOT NULL, importance TEXT NOT NULL, typeIntervention TEXT NOT NULL, equipmentId INTEGER NOT NULL, description TEXT NOT NULL, etatDemande TEXT not null, FOREIGN KEY (equipmentId) REFERENCES Equipment(idEquipment))
id	INTEGER	"id" INTEGER
raison	TEXT	"raison" TEXT NOT NULL
dateDemande	DATE	"dateDemande" DATE NOT NULL
lieu	TEXT	"lieu" TEXT NOT NULL
importance	TEXT	"importance" TEXT NOT NULL
typeIntervention	TEXT	"typeIntervention" TEXT NOT NULL
equipmentId	INTEGER	"equipmentId" INTEGER NOT NULL
description	TEXT	"description" TEXT NOT NULL
etatDemande	TEXT	"etatDemande" TEXT NOT NULL
Employe		CREATE TABLE "Employe" (idEmploye integer primary key autoincrement, nomEmpl text not null, telephone text not null, email text not null, service text not null, description text not null, EmplImage TEXT)
idEmploye	integer	"idEmploye" integer
nomEmpl	text	"nomEmpl" text NOT NULL
telephone	text	"telephone" text NOT NULL
email	text	"email" text NOT NULL
service	text	"service" text NOT NULL

Name	Type	Schema
description	text	"description" text NOT NULL
EmplImage	TEXT	"EmplImage" TEXT
EmployeOrdreIntervention		CREATE TABLE EmployeOrdreIntervention (idEmploye INTEGER NOT NULL, idOi INTEGER NOT NULL, FOREIGN KEY (idEmploye) REFERENCES Employe(idEmploye) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (idOi) REFERENCES OrdreIntervention(idOi) ON DELETE CASCADE ON UPDATE CASCADE, PRIMARY KEY (idEmploye, idOi))
idEmploye	INTEGER	"idEmploye" INTEGER NOT NULL
idOi	INTEGER	"idOi" INTEGER NOT NULL
Equipment		CREATE TABLE "Equipment" (idEquipment INTEGER primary key autoincrement, numEquipment text not null, type TEXT not null, marque TEXT not null, serie TEXT not null, acquiyear INTEGER not null, fabYear INTEGER not null, price REAL not null, inventaire TEXT not null, eq_image TEXT , etat TEXT default unknown not null)
idEquipment	INTEGER	"idEquipment" INTEGER
numEquipment	text	"numEquipment" text NOT NULL
type	TEXT	"type" TEXT NOT NULL
marque	TEXT	"marque" TEXT NOT NULL
serie	TEXT	"serie" TEXT NOT NULL
acquiyear	INTEGER	"acquiyear" INTEGER NOT NULL
fabYear	INTEGER	"fabYear" INTEGER NOT NULL
price	REAL	"price" REAL NOT NULL
inventaire	TEXT	"inventaire" TEXT NOT NULL
eq_image	TEXT	"eq_image" TEXT
etat	TEXT	"etat" TEXT NOT NULL DEFAULT unknown
EtapeResolution		CREATE TABLE EtapeResolution (idEtape INTEGER PRIMARY KEY AUTOINCREMENT, idOi INTEGER NOT NULL, descriptionEtape TEXT NOT NULL, numEtape integer not null , FOREIGN KEY (idOi) REFERENCES OrdreIntervention(idOi) ON DELETE CASCADE ON UPDATE CASCADE)
idEtape	INTEGER	"idEtape" INTEGER
idOi	INTEGER	"idOi" INTEGER NOT NULL
descriptionEtape	TEXT	"descriptionEtape" TEXT NOT NULL
numEtape	integer	"numEtape" integer NOT NULL
FactureAchat		CREATE TABLE "FactureAchat" (idFacture integer primary key autoincrement, idEquip integer not null references Equipment on update cascade on delete cascade, nomFournisseur text not null, numFacture text not null, dateFacture Date not null, montantTotale real not null, detail text , dureeGarantieJours integer)
idFacture	integer	"idFacture" integer
idEquip	integer	"idEquip" integer NOT NULL
nomFournisseur	text	"nomFournisseur" text NOT NULL
numFacture	text	"numFacture" text NOT NULL
dateFacture	Date	"dateFacture" Date NOT NULL

Name	Type	Schema
montantTotale	real	"montantTotale" real NOT NULL
detail	text	"detail" text
dureeGarantieJours	integer	"dureeGarantieJours" integer
FacturePieces		CREATE TABLE FacturePieces (ID_Facture INTEGER PRIMARY KEY AUTOINCREMENT, Nom_Fournisseur TEXT, -- Nom du fournisseur en tant que texte Date_Facture DATE, Statut TEXT, Piece_ID INTEGER, Quantite INTEGER, Prix_Unitaire REAL, FOREIGN KEY (Piece_ID) REFERENCES "Pieces_old"(ID_Piece) ON UPDATE CASCADE ON DELETE SET NULL)
ID_Facture	INTEGER	"ID_Facture" INTEGER
Nom_Fournisseur	TEXT	"Nom_Fournisseur" TEXT
Date_Facture	DATE	"Date_Facture" DATE
Statut	TEXT	"Statut" TEXT
Piece_ID	INTEGER	"Piece_ID" INTEGER
Quantite	INTEGER	"Quantite" INTEGER
Prix_Unitaire	REAL	"Prix_Unitaire" REAL
Intervention		CREATE TABLE Intervention(idIntervention integer primary key autoincrement , idOi integer not null, resultat text not null, cout double not null, duree integer not null, foreign key (idDi) references OrdreIntervention(idDi) on delete cascade on update cascade)
idIntervention	integer	"idIntervention" integer
idOi	integer	"idOi" integer NOT NULL
resultat	text	"resultat" text NOT NULL
cout	double	"cout" double NOT NULL
duree	integer	"duree" integer NOT NULL
Mouvement_Stock		CREATE TABLE Mouvement_Stock (ID_Mouvement INTEGER PRIMARY KEY AUTOINCREMENT, Piece_ID INTEGER, Date_Mouvement DATE, Quantite INTEGER, Type_Mouvement TEXT, -- Type de mouvement (par exemple, "Entrée", "Sortie") FOREIGN KEY (Piece_ID) REFERENCES "Pieces_old"(ID_Piece) ON UPDATE CASCADE ON DELETE SET NULL)
ID_Mouvement	INTEGER	"ID_Mouvement" INTEGER
Piece_ID	INTEGER	"Piece_ID" INTEGER
Date_Mouvement	DATE	"Date_Mouvement" DATE
Quantite	INTEGER	"Quantite" INTEGER
Type_Mouvement	TEXT	"Type_Mouvement" TEXT
OrdreIntervention		CREATE TABLE OrdreIntervention (idOi INTEGER PRIMARY KEY AUTOINCREMENT, idDi INTEGER NOT NULL, descriptionOi TEXT NOT NULL, etatOi TEXT NOT NULL, dureeEstimee integer not null, FOREIGN KEY (idDi) REFERENCES DemandeIntervention(id) ON DELETE CASCADE ON UPDATE CASCADE)
idOi	INTEGER	"idOi" INTEGER
idDi	INTEGER	"idDi" INTEGER NOT NULL
descriptionOi	TEXT	"descriptionOi" TEXT NOT NULL
etatOi	TEXT	"etatOi" TEXT NOT NULL

:

Name	Type	Schema
dureeEstimee	integer	"dureeEstimee" integer NOT NULL
PieceJointeDemande		CREATE TABLE PieceJointeDemande (id INTEGER PRIMARY KEY AUTOINCREMENT, idDI INTEGER NOT NULL, imgPath TEXT NOT NULL, descriptionRapide TEXT NOT NULL, FOREIGN KEY (idDI) REFERENCES DemandeIntervention(id))
id	INTEGER	"id" INTEGER
idDI	INTEGER	"idDI" INTEGER NOT NULL
imgPath	TEXT	"imgPath" TEXT NOT NULL
descriptionRapide	TEXT	"descriptionRapide" TEXT NOT NULL
PieceOI		CREATE TABLE PieceOI (id integer primary key, OIid integer not null, nomPiece text not null, quatite integer not null, FOREIGN KEY (OIid) REFERENCES OrdreIntervention(idOI) ON DELETE CASCADE ON UPDATE CASCADE)
id	integer	"id" integer
OIid	integer	"OIid" integer NOT NULL
nomPiece	text	"nomPiece" text NOT NULL
quatite	integer	"quatite" integer NOT NULL
Pieces		CREATE TABLE Pieces (ID_Piece INTEGER PRIMARY KEY AUTOINCREMENT, Nom_Piece TEXT NOT NULL UNIQUE, Type_Piece TEXT NOT NULL, Criticite TEXT, Description TEXT, PieceImg BLOB)
ID_Piece	INTEGER	"ID_Piece" INTEGER
Nom_Piece	TEXT	"Nom_Piece" TEXT NOT NULL UNIQUE
Type_Piece	TEXT	"Type_Piece" TEXT NOT NULL
Criticite	TEXT	"Criticite" TEXT
Description	TEXT	"Description" TEXT
PieceImg	BLOB	"PieceImg" BLOB
PlanOI		CREATE TABLE PlanOI (idPlan INTEGER PRIMARY KEY AUTOINCREMENT, idOI INTEGER NOT NULL, planDate DATE NOT NULL, FOREIGN KEY (idOI) REFERENCES OrdreIntervention(idOI) ON DELETE CASCADE ON UPDATE CASCADE)
idPlan	INTEGER	"idPlan" INTEGER
idOI	INTEGER	"idOI" INTEGER NOT NULL
planDate	DATE	"planDate" DATE NOT NULL
Stock		CREATE TABLE Stock (ID_Stock INTEGER PRIMARY KEY AUTOINCREMENT, Piece_ID INTEGER, Quantite_Disponible INTEGER NOT NULL, FOREIGN KEY (Piece_ID) REFERENCES "Pieces_old"(ID_Piece) ON UPDATE CASCADE ON DELETE SET NULL)
ID_Stock	INTEGER	"ID_Stock" INTEGER
Piece_ID	INTEGER	"Piece_ID" INTEGER
Quantite_Disponible	INTEGER	"Quantite_Disponible" INTEGER NOT NULL
User		CREATE TABLE User (id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT NOT NULL UNIQUE, password TEXT NOT NULL)
id	INTEGER	"id" INTEGER

4

Name	Type	Schema
username	TEXT	"username" TEXT NOT NULL UNIQUE
password	TEXT	"password" TEXT NOT NULL
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"

Bibliographie

- [1] M.Frédéric, Pratique de la maintenance industrielle Dunod, Octobre, 1997
- [2] Smith, A. M., & Hinchcliffe, J. (2004). *RCM: Gateway to World Class Maintenance*
- [3] "Maintenance industrielle : Stratégies, organisation et outils" de Patrick Penot, Dunod (édition 2013) .
- [4]International Energy Agency (IEA), "Energy Technology Perspectives 2020"
- [5]"Software Maintenance" de Penny Grubb et Armstrong A. Takang (Wiley, 2003)
- [6] Maintenance & Asset Management - "La GMAO : un levier pour la performance industrielle", 2023
- [7] V.Athané, « Appels d'offres dans les entreprises industriellesEnjeux, Pratiques et Informatisation », Mémoire du Magister www.hec.unil.ch 2000
- [8]M.Siemde Moussa, « cours de la GMAO », IST 2016
- [9]Léonard,J.,&Pichon,D.(2010).Maintenance des système industriels.Paris :édition technip
- [10] Mobley, R.K., *Maintenance Engineering Handbook*, McGraw-Hill Education, 2008.
- [11] Gershon, P. (2013). *Maximo for Dummies*. John Wiley & Sons.
- [12] Anderson, K. (2019). *SAP PM: Plant Maintenance: An Introduction to the SAP Maintenance Management System*
- [13] McCarthy, P. (2017). *Infor EAM: Enhancing Maintenance Management in a Digital World*
- [14] Williams, R. (2020). *Asset Tracking and Management with Asset Panda: Optimizing Maintenance and Life Cycle Management*
- [15] Miller, J. (2020). *UpKeep Maintenance Management: A Practical Guide to Implementing and Optimizing Your GMAO Solution*
- [16] M.elouissi,M.Boudjelal, « GMAO des dispositifs médicaux dans une administration de santé » mémoire master en génie biomédical, université de Tlemcen, 2015
- [17] <https://www.microsoft.com>
- [18]B.PRIEUR, WPF développez des applications structurées (MVVM, XAML,...), édition ENI depuis 2017,p59
- [19] Gossman, D. (2017). *MVVM Explained: The Model-View-ViewModel Design Pattern*
- [20] spring.io 20/11/2019
- [21]www.developpez.com 15/07/2019
- [22] Owens, M. (2010). *The Definitive Guide to SQLite*

Résumé

Le déploiement de ce logiciel GMAO est devenu de plus en plus indispensable, il tend de réduire les coûts et d'améliorer la qualité.

Ce mémoire présente le processus de conception, de développement et de mise en œuvre d'un logiciel de Gestion de Maintenance Assistée par Ordinateur (GMAO), destiné à optimiser la gestion des équipements et la planification des interventions de maintenance au sein d'une entreprise. L'objectif principal du projet est de proposer une solution logicielle permettant de centraliser et d'automatiser les tâches de maintenance, tout en améliorant l'efficacité des opérations et la traçabilité des interventions.

Le logiciel développé repose sur une architecture flexible et modulaire, intégrant une base de données SQLite pour le stockage local des données, avec la possibilité de synchroniser ces informations avec un serveur central pour le partage multi-utilisateurs. Le choix de SQLite comme base de données embarquée s'explique par sa légèreté et sa facilité d'intégration, idéale pour une utilisation dans des environnements déconnectés ou mobiles. Une architecture hybride est envisagée, permettant un fonctionnement hors ligne pour les techniciens sur le terrain, suivi d'une synchronisation avec un serveur central dès que la connexion est rétablie.

Abstract

The deployment of This Computerized Maintenance Management System (CMMS) software has become increasingly essential, aiming to reduce costs and improve quality. This thesis presents the design, development, and implementation process of a CMMS software aimed at optimizing equipment management and planning maintenance interventions within a company. The main objective of the project is to provide a software solution that centralizes and automates maintenance tasks while enhancing operational efficiency and traceability of interventions.

The developed software is based on a flexible and modular architecture, integrating an SQLite database for local data storage, with the ability to synchronize this information with a central server for multi-user sharing. The choice of SQLite as the embedded database is due to its lightweight nature and ease of integration, making it ideal for use in disconnected or mobile environments. A hybrid architecture is envisioned, allowing for offline operation for technicians in the field, followed by synchronization with a central server as soon as the connection is restored.