

***Module: Programmation orienté objet (java)***

***Conception et réalisation d’une application de gestion d’un parc automobile***

*Réalisé par :*

*Zakaria HADRAOUI*

Encadré par :

**Professeur Youssef ES-SAADY**

*Remerciement*

Je tiens à remercier toutes les personnes qui m’ont aidé à réaliser ce mini projet, et d'apprendre des nouvelles technologies et d’approfondir mes connaissance en d’autre technologies .

Tout d'abord, j'adresse mes remerciements à mon professeur, **Professeur Youssef Es-Saady** qui nous a orienté vers une nouvelle méthode universitaire d' apprentissage. Je tiens à le remercier pour son écoute et ses conseils qui m'ont permis de cibler les outils et les technologies à connaître ou à approfondir .

Enfin, je tiens à remercier toutes mes collègues d'équipe qui m'ont conseillé et relu lors de la rédaction de ce rapport de stage

**Problématique**

Développer une application en java pour la gestion des parcs automobiles. Les fonctionnalités requises

* Lecture de matricule à l'aide d’un caméras
* Enregistrement des informations de la voiture dans une base de données Mysql
* Tableaux de bord produisant l’état du parc

1. Consultations et gestion des tables
2. Moteur de recherches multicritères.
3. Statistiques des visites
4. analyse multidimensionnelle
5. historiques
6. listes

Le parc automobile, qu’il soit acheté ou loué, occupe une place importante dans la vie de l’entreprise.

L’utilisation d’un logiciel est nécessaire afin d’accéder à une **gestion de parc automobile** optimale et rigoureuse. Le logiciel de **gestion de parc automobile** va nous permettre de centraliser et d’analyser toutes les données de notre parc.

Le suivi de la maintenance et des entretiens de notre parc sera bien plus simple et efficace si nous pouvons nous appuyer sur un logiciel dédié.

Outils utilisés

IntelliJ IDEA

pour la programmation des applications java j'utilise eclipse ou bien netbeans mais avec ce nouveau défis j’ai appris à utiliser et à manipuler un nouveau IDE IntelliJ IDEA avec une licence universitaire

cet outil très puissant et très efficace a une grande communauté dans le monde il facilite la programmation et la [gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions#Gestion_de_versions_d.C3.A9centralis.C3.A9e) à l'aide du git intégré dans ce IDE

MySQL

un [système de gestion de bases de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) relationnelles (SGBDR). Il est distribué sous une double licence [GPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) et [propriétaire](https://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire). pour notre cas on a utilisé la version gratuite sous licence [GPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU).

Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

j’ai appris que ce SGBDR est très légères et il contient plusieurs type de moteur de gestion tel que InnoDB ,un moteur de gestion récent et qui gère les relations entre les tables, et MyISAM qui est plus ancien que InnoDB mais son inconvénient c'est qu’il ne support pas les relations entre les tables ,or il très rapide,

j’ai appris à utiliser les méthodes et les fonctionnalités de ce SGBDR, tel que les calculs sur les dates ,le temps et du timestamp, la création de déclencheur afin de manipuler les 3 opérations (insert,update,delete).

# PowerAMC/PowerDesigner

un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

j’ai appris à créer des diagrammes et du MCD et MLD à l'aide de ce logiciel qui contient plusieurs fonctionnalité,

* la génération du diagramme de class a l'aide des fichiers \*.java
* la génération du du code java à partir un Diagramme de class établie
* la création automatique du MPD a partir d’un MLD établie
* la création automatique du MLD à partir d’un MCD établie
* la génération du MPD,MLD a partir d’une base données il support la plupart du base données, et presque tous les langages de programmations)
* la génération d’un rapport détaillé selon le workstation du travail

HeidiSQL

Pour l' administration du SGBDR MySQL j’ai préféré utiliser HeidiSQL parce qu' il est léger et gratuit sous Licence publique générale GNU,

Il possède un éditeur SQL et un constructeur de requête. Il a été développé et optimisé pour être utilisé avec le [SGBD](https://fr.wikipedia.org/wiki/SGBD) relationnel [MySQL](https://fr.wikipedia.org/wiki/MySQL) . HeidiSQL est capable de se connecter à des bases MySql, SQL Server ainsi que Postgresql .

GIT

j’ai compris utilité du git et de la [gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions#Gestion_de_versions_d.C3.A9centralis.C3.A9e), le travail individuel et surtout en équipe pose souvent des problème s d'intégration et de versionning , mais on manipulant le commande le plus populaire du gît tel que

git ini -- initialiser un projet git

git add \* -- ajouter les fichiers et les dossiers

git commit -m “” -- permet de commit une tache données

git branch name --créer une branche données pour une tâche réalisée

git checkout name branch -- changement de branch

git clone url -- copie un projet git localement

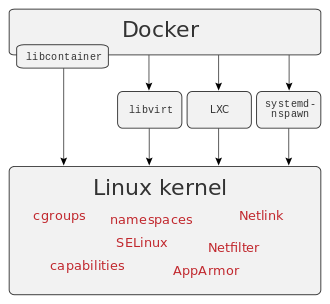
git push url -- envoyer les données vers le serveur du git

et pas mal d'autres commandes , et comme hébergeur de projet git on a choisit github.com un serveur gratuit , rapide et le plus populaire ainsi que il a été acheté récemment par microsoft

# Docker

Pour éviter le problème de la configuration software et des dépendances requises, j’ai appris à utiliser Docker, c’est un outil qui permet d'empaqueter une application et ses dépendances dans un conteneur isolé, qui pourra être exécuté sur n'importe quel serveur . Il ne s'agit pas de virtualisation, mais de conteneurisation, une forme plus légère qui s'appuie sur certaines parties de la machine hôte pour son fonctionnement. Cette approche permet d'accroître la flexibilité et la portabilité d’exécution d'une application, laquelle va pouvoir tourner de façon fiable et prévisible sur une grande variété de machines hôtes, que ce soit sur la machine locale, un cloud privé ou public.

ci dessus un schéma explicatif des interfaces de Docker



Conception et programmation

**Ma tâche de l’équipe consiste à créer**

* Créer et gérer la bases de données MySQL
* la programmation des interfaces et du code en java

dans ce sens j’ai appris à utiliser les patterns de conceptions suivants

# **pattern Fabrique abstraite**

# **pattern DAO**

# **Singleton**

Modèle-vue-contrôleur le motif d'architecture logicielle,destiné aux interfaces graphiques , le MVC est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

* Un modèle (Model) contient les données à afficher.
* Une vue (View) contient la présentation de l'interface graphique.
* Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

pour la parti GUI j'ai appris à utiliser la bibliothèque JavaFX, cette bibliothèque a été conçue pour remplacer les deux bibliothèques du java Swing et AWT, ce framework GUI issue du projet OpenJFX, il permet aux développeurs Java de créer une interface graphique pour des applications de **bureau**, des applications **internet riches** et des applications **smartphones** et **tablettes tactiles.**

##### **Avantages de JavaFX :**

* Permet aux développeurs d'intégrer des graphiques vectoriels, des animations, du son et des vidéos récupérés sur Internet dans une application riche, dense et interactive.
* Étend la technologie Java grâce à l'utilisation de n'importe quelle bibliothèque Java au sein d'une application JavaFX.
* Améliore l'efficacité du travail entre concepteurs et développeurs, les premiers pouvant utiliser les outils de leur choix tout en collaborant avec les seconds.
* la séparation du code java, et XML pour les interfaces

# **inconvénients de JavaFX**

* moins d'outils de designer
* nécessite l'installation du SDK JAVAFX sur le poste client
* avoir une version récente du jdk au min 11
* JavaFX n'est pas intégré au sein de JDK Embedded
* Manque d’un contrôleur pour la gestion du temps
* la gestion et le appels des windows n’est pas facile
* manque de communication entre les interfaces

**Programmation langage Java**

* j’ai approfondis mais concessaince en java surtout côté POO
* Maîtriser la syntaxe du langage Java
* la gestion des exceptions des entrées / sorties en Java
* approfondir mes connaissances en JDBC version 8 pour accéder aux bases de données, …
* utilisation et la gestion des procédures stockées en niveau du code java
* utilisation des interfaces et des classe abstract
* j’ai appris a utilisé la programmation parallèle à l'aide du threads
* utilisation timer
* utilisation du module-info.java anfin d’inclure les packages nécessaires à l'aide du mot requise
* définir les packages à l'aide du mot export …
* utilisation du régulière expression en java afin de manipuler des données saisies dans les contrôles (textfield…..)
* apprendre à utiliser expression lambda java inclure à partir du java 8
* utilisation du foreach et l’itération en java
* utilisation des algorithmes de hash pour crypter les passwords stockés dans la BD et plus précisément md5
* utilisation des Test Unitaires grâce au junit
* gestion des dependance grâce au maven et gradle