



Université Abdelmalek Essaadi
Ecole Nationale des Sciences Appliquées Al
Hoceima Département Mathématiques et
Informatique



Génie Informatique

Module : Programmation JAVA

Mini-projet

Rapport application Gestion grand Taxi

Réaliser par : Bouhchicha Adnan &
Mouad Karim
Encadrée par : M. Bahri Abdelkhalek
Génie informatique 2

Table des matières

1.INTRODUCTION.....	4
2.DESCRPTION DU PROJET	6
Exigences fonctionnelles	5
Exigences non fonctionnelles.....	7
3.ANALYSE ET CONCEPTION	9
Spécification des exigences : lescas d'utilisations.....	9
Analyse du domaine : le diagrammede classe	10
La conception fonctionnelle.....	13
La conception technique.....	13
La conception graphique.....	15
4.REALISATION TECHNIQUE	17
4.1 Les langages utilisés.....	17
4.2 Les tests	18
5.BILAN DU PROJET.....	21
Résultat.....	21
Améliorations envisagées	21
6.LE Mode d'Emploi	22
Courtier	22
Admin	24
7.CONCLUSION	26

1. Introduction

1.1 Pourquoi réaliser un projet de gestion?

La gestion de projet vise à organiser, dans son intégralité, le bon déroulement du projet. Elle fait appel à des techniques d'organisation pendant le cycle de vie de celui-ci dans l'unique but d'atteindre les objectifs fixés au départ.

Pour travailler efficacement et atteindre un certain niveau de qualité, nous pouvons distinguer différentes étapes au cours de la gestion de projet :

- Réalisation d'un bref initial avec toute l'équipe projet
- Suivi et contrôle régulier des différentes tâches
- Anticipation des éventuels problèmes et des solutions pour les contrer
- Adaptation des plannings et/ou des ressources

1.2 Problématique.

Les grands taxis offrent une pluralité de solutions adaptées aux besoins. Leur usage est simple : dans la majorité des cas, ils suivent des itinéraires fixes et s'arrêtent au gré des personnes, les différentes stations (points de départ et d'arrivée) correspondant à des itinéraires précis. Néanmoins, ils peuvent effectuer tous types de déplacements interurbains à la demande des clients. Les stations des taxis collectifs sont rarement éloignées des centres-villes et les véhicules y sont regroupés selon des destinations prédéfinies. Les chauffeurs organisent leur activité et celle des stations sur la base de départs à tour de rôle en appliquant le principe du remplissage des véhicules avant départ. Ce remplissage est mal organisé et on se trouve devant plusieurs problèmes soit au niveau d'organisation ou bien au niveau de manque d'information sur ce grand marché chez l'état.

1.3 Solution Proposée.

Réalisation d'une application permettant de faciliter l'organisation des taxis selon le tour de rôle en prenant en considération la ville d'arrivée.

Pouvoir enregistrer, reprendre et imprimer les données des taxis dans chaque ville.

Ces données seront :

- Les informations de courtier des provinces (Nom, Prénom, CIN, âge, numéro de tel).
- Les informations de taxi de chaque province (conducteur, permis, matricule, marque).
- Admission mensuel ou annuel de chaque station.
- Le nombre d'employé, taxi, conducteur et courtier dans chaque station.

1.4 Bénéfices.

L'analyse de ces données va nous offrir une visualisation sur le marché et va aider l'état d'avoir les résultats exacts pour bien traiter le financement de ce marché selon chaque ville et chaque station de taxi.

Ceci va sûrement aider aussi les chauffeurs de Taxis d'améliorer les performances du transport et procurer la rapidité des déplacements ainsi que la souplesse dans la desserte et les horaires.

2. Description du projet

Puisqu'il s'agit d'une application desktop de gestion de taxi, on a choisi pour le nom de l'application : « Gestionnaire de taxi ». On pense qu'il résume bien le sujet de l'application.

Gestionnaire_de_taxi devra donc regrouper toutes les fonctionnalités nécessaires aux courtiers ainsi qu'aux admins : consulter et comparer le gain mensuels de chaque station, accéder à la liste des trajets faite chaque jours, contacter le courtier au cas d'une contravention, modifier, ajouter ou supprimer des taxis, ..., et toutes les fonctionnalités techniques comme : s'inscrire, s'authentifier, se déconnecter... Mais aussi, il devra répondre à des exigences non fonctionnelles par sa qualité et ses performances.

Exigences fonctionnelles

Inscription

Un courtier doit s'inscrire pour pouvoir faire la gestion de tour de rôle des taxis, sinon l'application ne s'exécutera jamais, dans un certain temps cette inscription doit être validé par un admin ce dernier va vérifier les documents fournis par le courtier. Lors de cette inscription, il saisit des informations personnelles, crée son profil afin de devenir utilisateur de l'application.

Authentification

Afin d'accéder à l'application et pouvoir accéder aux différentes fonctionnalités (ajouter le taxi, modifier, supprimer...), tout courtier doit s'authentifier.

Admin

Tout membre Admin de l'application possède un Dashboard, auquel il peut consulter les variations de bénéfices de chaque mois et de chaque station, gérer les taxis, et les courtiers, ajouter des stations.

Nous contacter

Par le biais de son email, un courtier peut nous contacter pour faire des réclamations si une enfreint de la pars d'un chauffeur de taxi a été faite a son égare.

Déconnection

Le compte de courtier ou d'admin est protégé grâce au traitement de certaine faille au niveau du code.

Consulter

Ajouter les informations des taxis, chaque courtier est affiché avec son image et ses informations. Chaque courtier fait partie du system d'une ville

Organiser

Un courtier organise le tour de rôle des taxis selon la ville d'arriver et le principe du FIFO First in First out, ainsi si un des taxis a eu un problème et ne peux pas servir ces clients le courtier a la possibilité de supprimer ou de modifier dans le tableau de rôle et il sera modifier directement dans notre base de donnée.

Exigences non fonctionnelles

Exigences de qualité

Afin que l'application donne envie aux courtiers de l'utiliser et ce débarrasser de leurs tableau et stylo, et de proposer la version améliorée de cette application ou on va faire des traitements des données et donné des interprétations et deviner le gain du mois prochain, détecter la période de la journée ou on a beaucoup de clients et pourquoi pas réaliser une partie en ligne de réservation il est important de répondre aux exigences de qualité suivantes :

Ergonomie efficace

Consulter et gérer les taxis est un travail pourquoi pas en être un plaisir. La mise en page de l'application doit faciliter un maximum la démarche à l'aide d'une présentation claire et intuitive. Nous reviendrons sur ce facteur de qualité dans la partie conception.

Interface graphique

Les différentes couleurs et choix typographiques doivent permettre à un utilisateur de repérer les différentes fonctionnalités qui s'offrent à lui pour cela que le design de l'application été float et simple.

Charte d'utilisation

Puisque les utilisateurs de l'application (les courtiers) sont moins éduqués, une fiche d'utilisation doit être consultable par tout utilisateurs, afin qu'ils connaissent les règles d'utilisation de l'application.

Aide en ligne

A tout moment, le courtier doit pouvoir consulter des pages d'aide.

Sérialisation

Le cas d'utilisation typique se déroule donc de la manière suivante :

- Dans un premier temps, le courtier utilise la phase d'inscription pour entrer ses informations. Il remplit l'ensemble des champs concernant ses informations personnelles (nom, prénom, numéro de téléphone, CIN,...).
- Après, il passe à une autre phase pour finaliser son inscription. (Photo de profile, carte nationale et un certificat de travail recto et verso).
- Ses données sont enregistrées dans une base de données "gestionTaxi" qui a une table principale du courtier et des multi tables qui se varient selon le nom de chaque ville.
- Après l'authentification du courtier, il passe à une interface d'accueil où il peut ajouter, modifier et supprimer les infos de taxis dans un tableau trier selon le nom de la ville d'arrivée pour qu'il puisse faire la gestion de taxi et le principe FIFO.
- Face à ce projet, nous avons premièrement pensé à saisir les données dans une base MySQL, ceci aurait permis de pouvoir facilement ajouter, modifier et supprimer les données.
- Les langages orientés objet proposent une solution efficace pour la sauvegarde / restauration d'objets à propriétés multiples, il s'agit de la sérialisation. Sérialiser consiste à transformer un objet instancié en un fichier dont les données seront donc conservées au-delà de l'exécution du programme.
- Par la suite, on pourra ouvrir ce fichier, et instancier de nouveau les objets. Il est aussi possible d'utiliser la sérialisation pour transférer par exemple des objets à travers un réseau, mais cela dépasse le cadre de notre projet, et nous n'aborderons ici que la sérialisation vers un flux stocké dans un fichier.

3. Analyse et conception

Dans cette partie, on a utilisé la modélisation UML pour représenter les spécifications des exigences grâce au diagramme de cas d'utilisation, mais aussi pour analyser le domaine avec le diagramme de class, le diagramme d'activité et le diagramme de séquence. Par la suite, nous abordons la conception, d'un point de vue fonctionnel, technique et graphique.

Spécification des exigences : les cas d'utilisations

On va répondre aux questions suivantes : Quels sont les utilisateurs du système ? Quelles sont leurs interactions avec celui-ci ? Il faut donc identifier les différents acteurs ainsi que les cas d'utilisation c'est-à-dire les différentes fonctionnalités du système.

Les acteurs pour l'application Gestion de taxi sont les suivants :

- **Le courtier** : personne qui est membre de l'application, gérer les trajets.
- **Le Technicien** : rôle des personnes qui ont en charge la maintenance de l'application.
- **L'admin** : personne qui possède un compte admin, qui Gérer les Taxis, Courtier, ville, consulter les données...

Les principaux cas d'utilisation des acteurs précédemment identifiés, ont été bien mis en évidence dans la partie précédente. Voici donc le diagramme de cas d'utilisation.

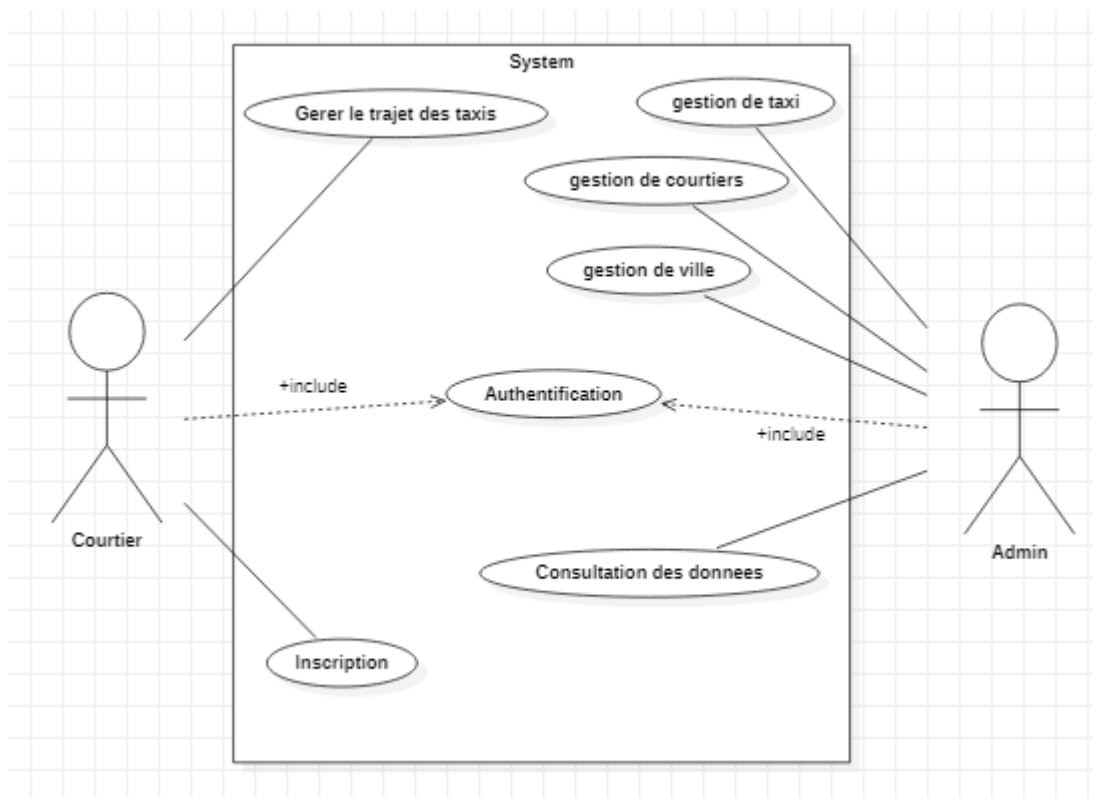


Diagramme de cas d'utilisation.

Remarque : par la suite, le technicien ne sera plus pris en compte.

Analyse du domaine : le diagramme de classe

Dans cette partie, nous étudierons les entités statiques du système. Ceci est illustré par le diagramme de classes suivant :

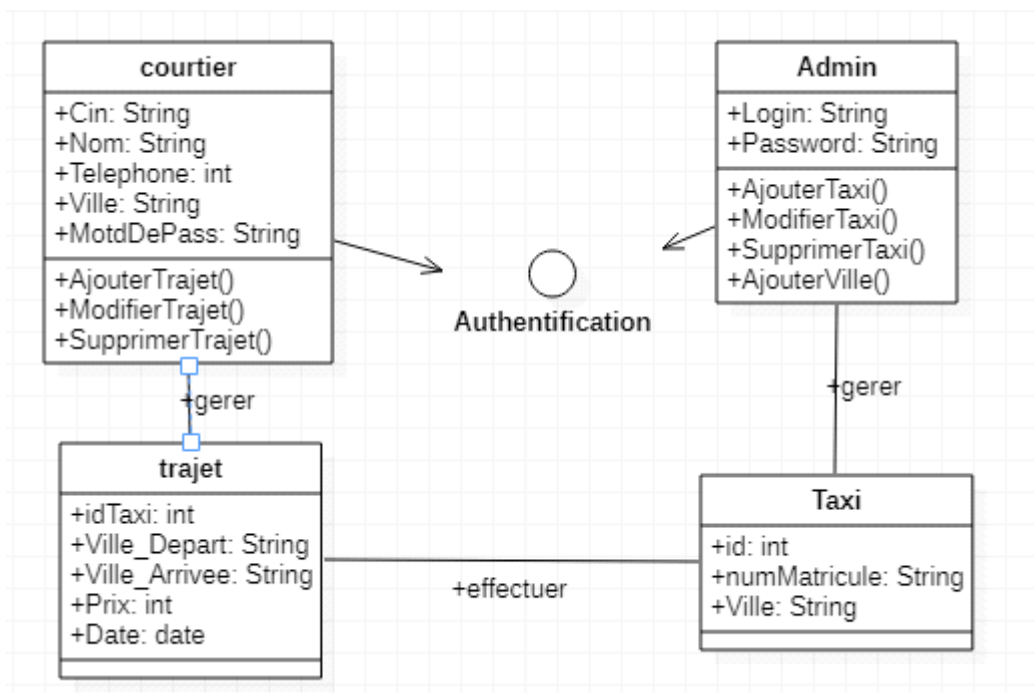
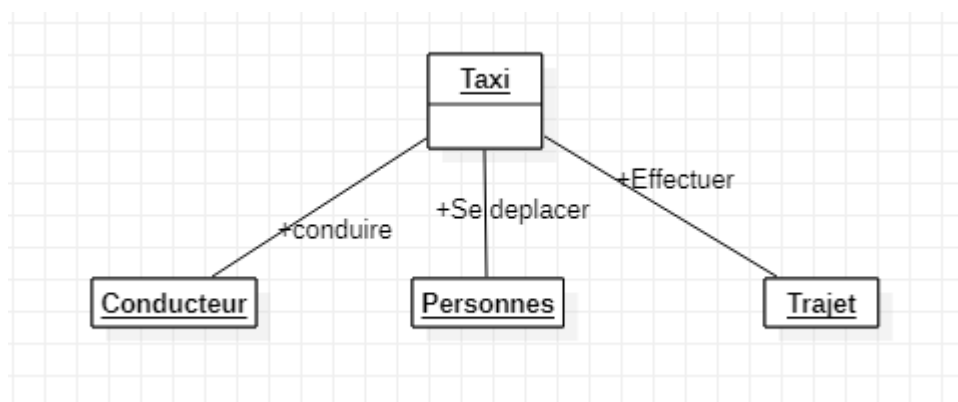


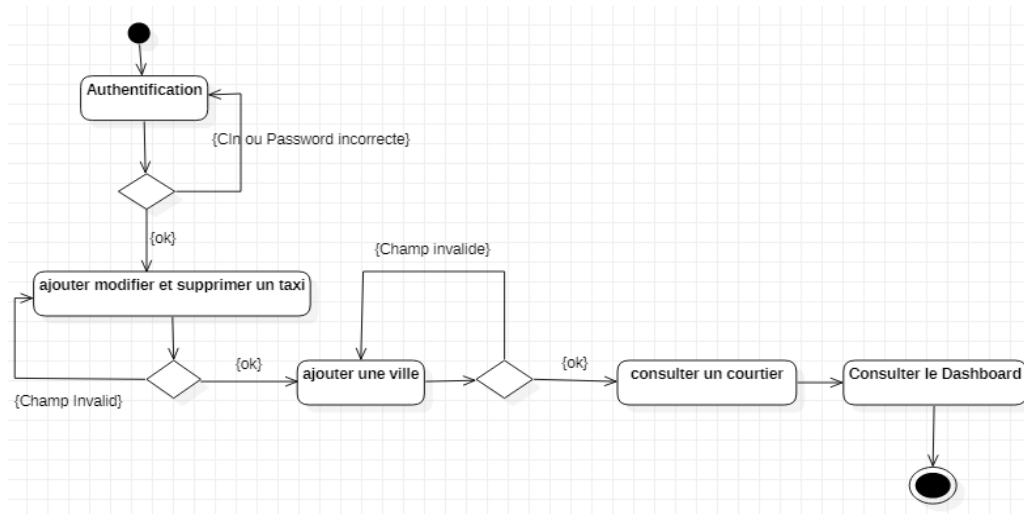
Diagramme de classes.

Le diagramme d'objet

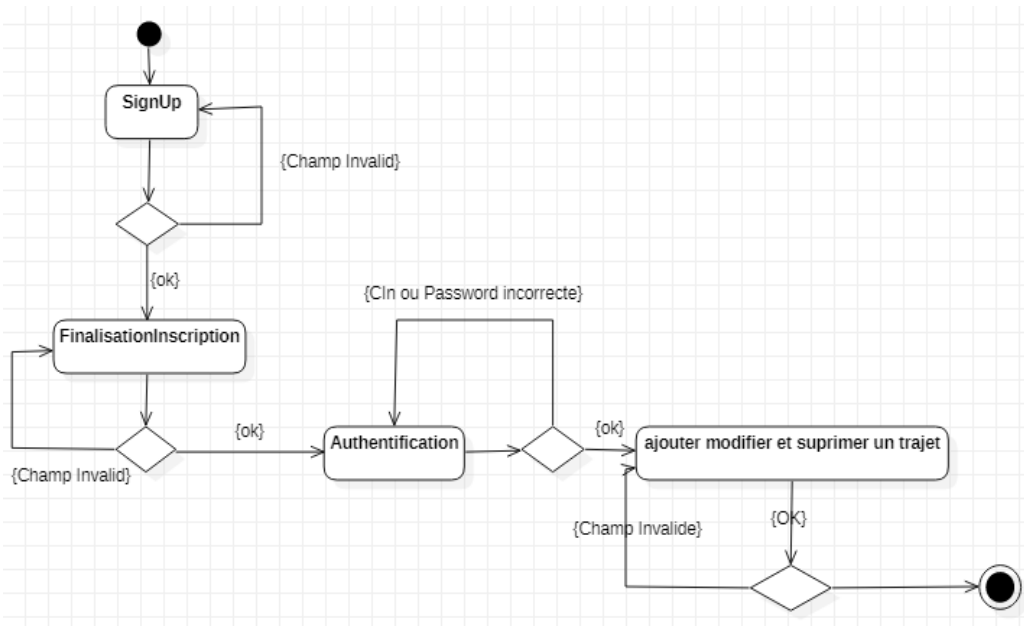


Le diagramme d'activité

Admin :

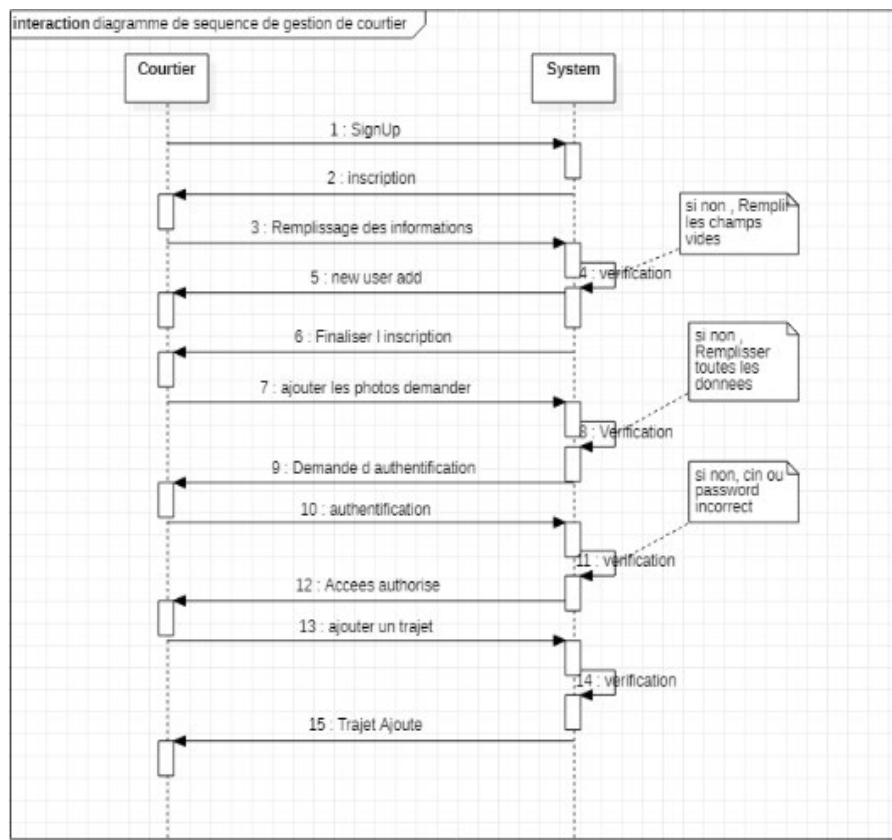


Courtier :

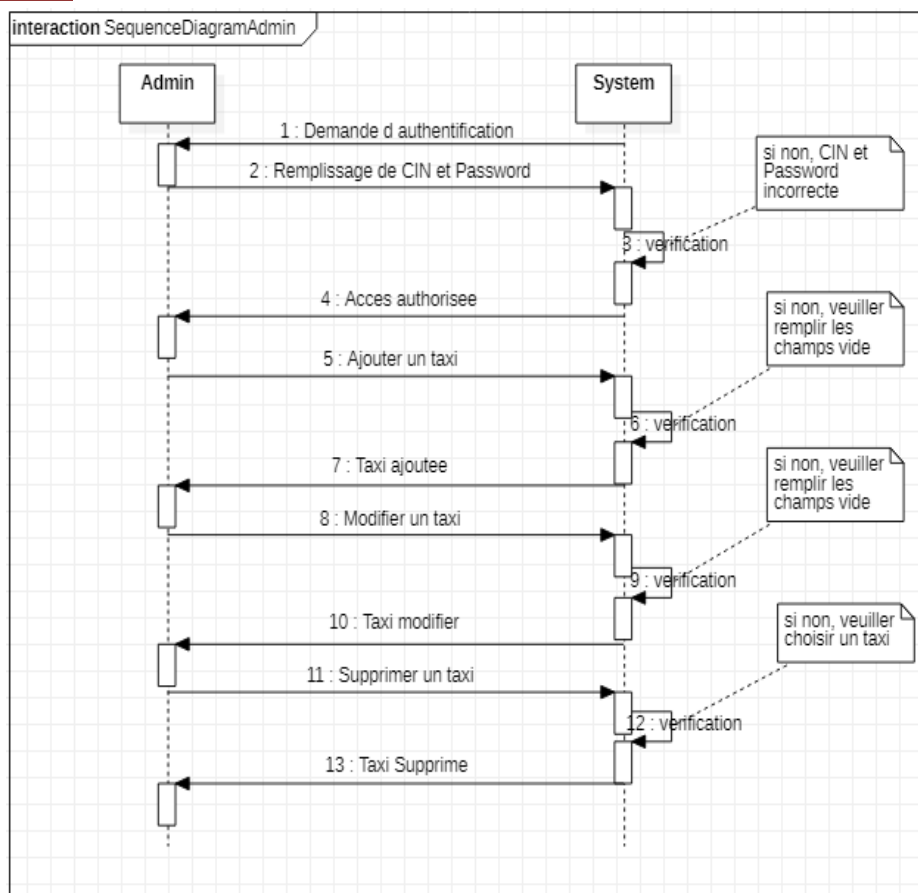


Le diagramme de séquence :

Courtier :



Admin :



La conception fonctionnelle

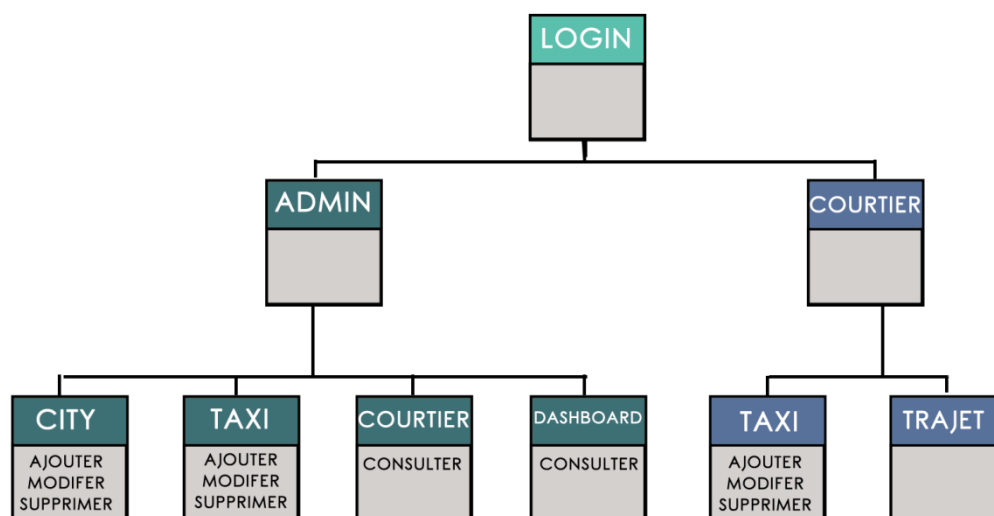
La structure de l'application

L'application est basée sur une structure que l'on appelle structure hiérarchisée. Il s'agit d'une structure découpée en blocs d'informations, ou de fonctionnalités, reliés à un même nœud.

La page d'authentification fait office de nœud principal qui donne racine à diverses fonctionnalités une fois authentifier.

Ces fonctionnalités sont présentes dans 3 barres d'outils regroupées par thèmes:

- fonctionnalités d'un admin
- fonctionnalités d'un courtier

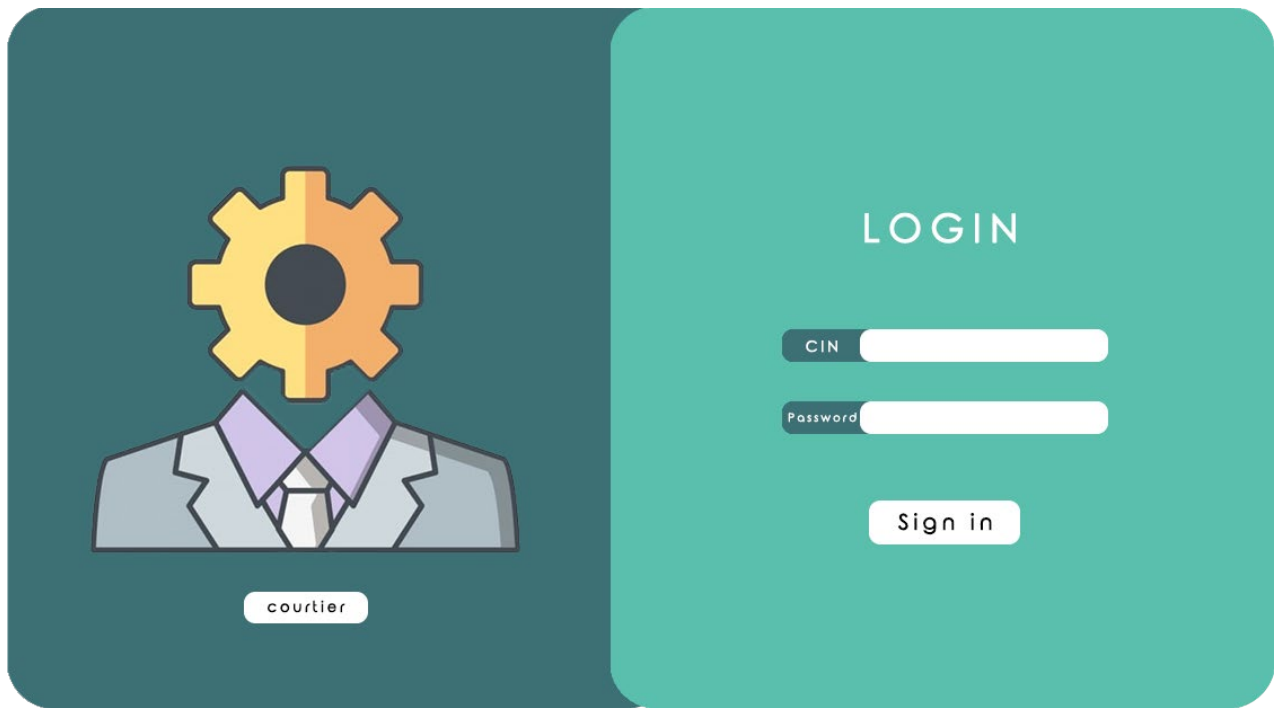


Structure hiérarchisée simplifiée.

On a donc choisi ce type de schéma d'organisation puisqu'il s'adapte à nos objectifs, dans un lieu, une facilité de navigation pour l'utilisateur en se familiarisant très rapidement à ce schéma, et d'un autre une bonne organisation de nos informations et fonctionnalités.

La structure des pages

Toutes les pages de l'application ont la même structure.



Trame simplifiée des pages.

Les éléments changeant entre les différentes pages sont le contenu, ainsi que la barre d'outils. On reviendra sur cela lors de la conception graphique.

L'application s'organise d'un conteneur principal, LOGIN.

Remarque : le mot de passe de l'utilisateur est hasher pour assurer la protection de ses informations personnelles

La conception technique

Les règles de nommage et de codage

Nous devons respecter certaines règles de nommage et de codage que nous nous sommes définies. Elles permettront une compréhension plus rapide du code par l'encadrant.

La conception graphique

L'ergonomie est un point important. Son objectif principal est de faciliter l'utilisation de l'application par les utilisateurs. Pour cela, nous nous imposerons certaines contraintes :

Accessibilité des utilisateurs

L'application doit être fonctionnelle sous les systèmes les plus utilisés, c'est-à-dire : Windows la plus part des version . Les différentes fonctionnalités développées devront donc être testées sur Windows.

****Courtier & Admin**

L'utilisation doit être simple, même pour les usagers débutants. Il est donc indispensable que l'apprentissage de l'application ne demande pas des efforts à l'utilisateur. L'utilisation de menu qui répertorie les principales fonctionnalités permet à l'utilisateur de minimiser son effort mental.

Apparence de l'application

➤ La charte graphique

Afin de donner à l'utilisateur des repères au sein de l'application et de faciliter l'utilisation, il doit y avoir une cohérence entre chaque page. La charte graphique détermine les différentes règles graphiques et ergonomiques.

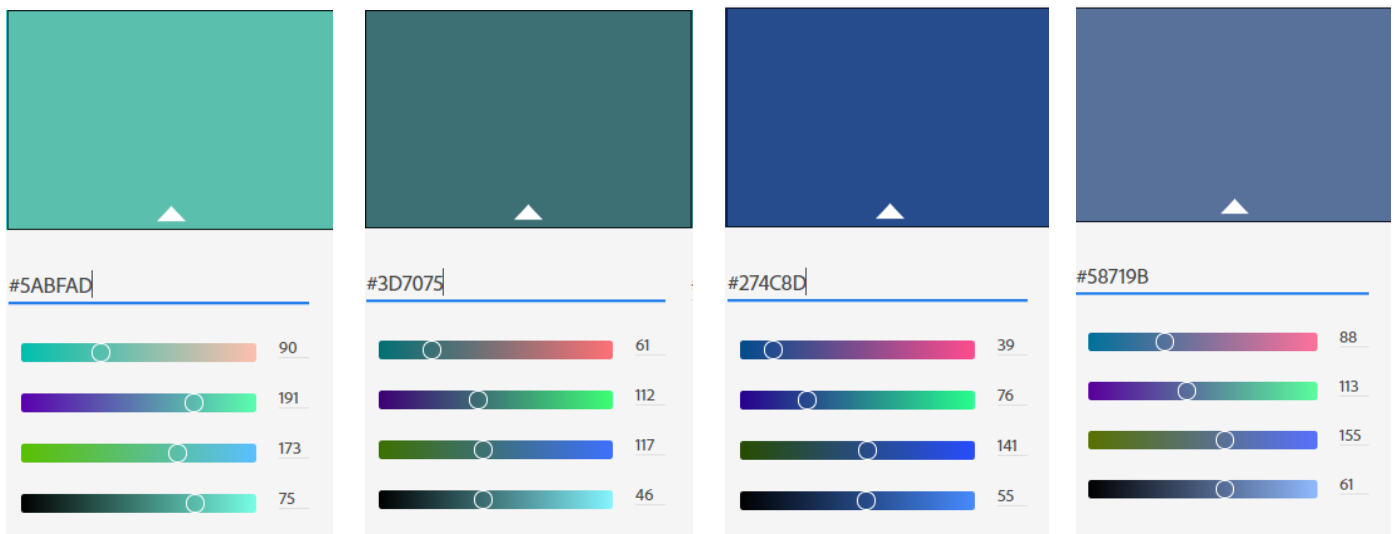
Elle contient :

- les dimensions des pages
- les couleurs à employer (le texte, les liens, les liens actifs, les boutons...)
- les styles à employer (famille, taille...)
- les types d'images...

➤ Choix des couleurs

Pour réaliser l'image du nom « Gestion de taxi », On a cherché les tendances actuelles. Il en a découlé qu'une image avec des formes carré ronde avec des couleurs bleu grise et vert correspondait le mieux, mais le choix était encore assez vaste pour la création.

Alors grâce a adobe colore On a choisi la palette suivante :

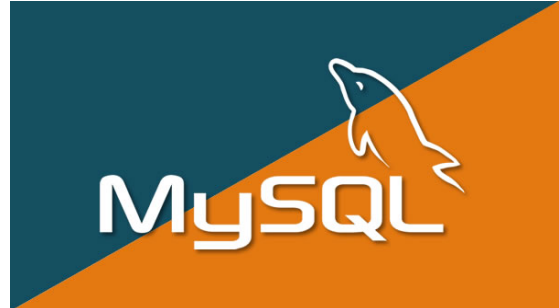


Avec Photoshop, On a designer les interfaces de l'application en respectant la palette de couleur et en réalisant des interfaces simple et signifiant.



4. Réalisation technique

4.1 Les langages utilisés



JAVA

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles applicatives multiplate-forme (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

MySQL

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

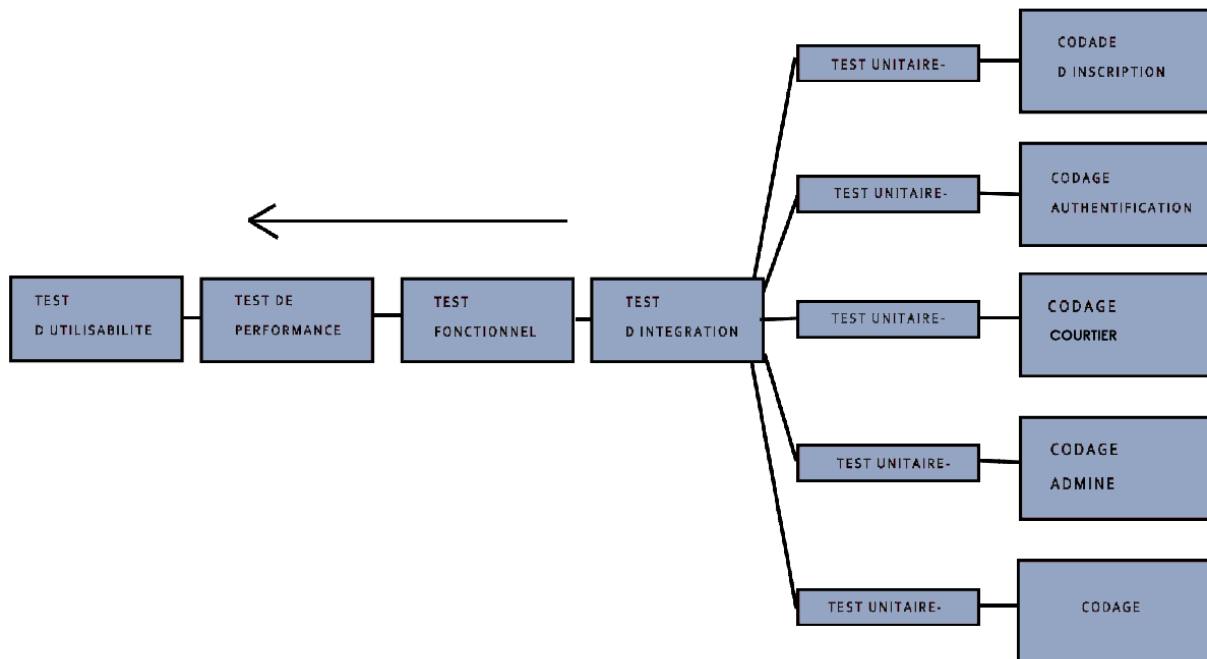
C'est un logiciel libre, open source, développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est la licence publique générale GNU (GPL) qui s'applique. Un logiciel qui intègre du code MySQL ou intègre MySQL lors de son installation devra donc être libre ou acquérir une licence payante. Cependant, si la base de données est séparée du logiciel propriétaire qui ne fait qu'utiliser des API tierces (par exemple en C# ou php), alors il n'y a pas besoin d'acquérir une licence payante MySQL. Ce type de licence double est utilisé par d'autres produits comme le framework de développement de logiciels Qt (pour les versions antérieures à la 4.5).

4.2 Les tests

Afin que l'application soit de qualité, il est important de réaliser des tests. Ces tests ont pour but de détecter les erreurs éventuelles et permettent de les corriger pour qu'il n'y ait pas d'impact sur le fonctionnement de l'application.

Pour cela au cours du développement, On a effectué 5 types de tests :

- des tests unitaires
- des tests d'intégrations
- des tests fonctionnels
- des tests de performances
- des tests d'utilisabilité.



Etapes des tests.

Les tests unitaires

Ces tests permettent de valider chaque fonctionnalité développée indépendamment des autres. L'objectif de ce test est de vérifier que chaque fonctionnalité développée correspond aux attentes des spécifications. Une fois les tests validés, le programme va être testé son intégration.

Résultats des tests

Toutes les fonctionnalités : inscription, insertion des images, pour le courtier l'ajout, la suppression et la modification de tour de rôle des taxi, pour l'admin, l'ajout de taxi et de ville, de supprimer ou modifier, consulter le Dashboard, gérer les courtiers... ont été testées. Les cas d'erreurs ont été traités. Une fois les images reçus, ensuite On a réalisé des tests d'intégration.

Les tests d'intégrations

Les tests d'intégration consistent en l'assemblage des fonctionnalités testées unitairement. L'objectif de ces tests est donc de valider l'intégration des différents modules entre eux, et de vérifier que le résultat est conforme à la conception.

L'intégration d'un module dans l'application est une phase délicate. Elle peut soulever des problèmes non prise en compte durant la phase d'analyse du module tout comme conduire à de simples bugs. C'est pourquoi les tests d'intégrations sont très importants puisqu'ils peuvent déceler ces bugs qui ne se sont pas apparus lors des tests unitaires, et qu'il faut corriger.

Mise en place des tests

On teste tout d'abord la compatibilité du module avec les autres. Ensuite on fait des tests plus poussés sur les fonctionnalités du module. Puis on teste les autres modules qui peuvent interagir avec celui-ci, en exécutant des tâches plus précises qui pourrait avoir des répercutions sur ce module. Bien entendu des corrections sont faites pour passer à des tests plus poussés.

Problèmes souvent rencontrés lors de l'intégration

Les problèmes les plus souvent rencontrés dans ces tests sont la redondance des importations de bibliothèques en JAVA, redondance des variables de session, le tableau de gestion des taxi et l'ajout du mouslistener de la recherche du tableau, l'affichage du Dashboard ainsi que d'autre fonctions.

Les tests fonctionnels

Ces tests permettent de savoir si les différentes fonctionnalités sont opérationnelles, et se déroulent correctement.

L'objectif de ce test est de vérifier que l'application est conforme au objectifs donne par l'encadrant et s'il résiste à des comportements anormaux. Si le système réagit bien à ces erreurs, cela veut dire que l'application a une bonne aptitude à résister aux erreurs.

Mise en place des tests

Pour chaque fonctionnalité développée, j'ai regardé si les différentes erreurs que peut rencontrer le système sont gérées.

Les tests de performance

L'objectif de ces tests est donc de détecter les pages qui peuvent être améliorées, notamment en modifiant les requêtes. Ces tests sont importantes pour les pages les plus fréquentées, puisqu'elles se doivent d'être le plus fluides possible.

Les tests d'utilisabilité

Ces tests permettent d'observer le comportement de l'utilisateur lorsqu'il utilise l'application : connaître ses impressions sur le graphisme utilisé, voir les problèmes qu'il rencontre, les questions qu'il peut se poser, les fonctionnalités qu'il apprécie.

Mise en place des tests

On a réalisé ces tests grâce à vingt personnes de mon entourage. Ces personnes ont été choisies selon leur âge, entre 30 et 55 ans, qui reflètent l'âge des courtiers au MAROC.

5. Bilan duprojet

Comment s'est-on organisé pour mener à bien ce projet ? Quels sont les résultats du projet ? A-t-on réussi à atteindre les objectifs fixés ? Quelles sont les différences entre objectifs et résultats ? Quelles améliorations sont envisagées ?

Les clés de notre méthode:

- la propreté du code, les commentaires, et respect de la charte de codage ;

Travail Collaboratifs

Nous sommes une équipe de deux personnes ce qu'as facilité le travail et la communication chacun de nous a pris la responsabilité de travailler sur une partie un la partie du courtier et l'autre la parti admin, mais quand un d'entre nous rencontre un problème on essaye de le résoudre tous les deux, ainsi on a mis d'accord dès le départ de la structure du code, les commentaires et la clarté étant des outils très essentiels pour mener au bien notre projet, les diagrammes, le design et les couleurs étaient la première des choses à discuter, nous avons donc, dès le départ, sélectionné toutes les tâches de niveau normale aux niveaux difficiles et on a donné à chaque personne ses propres tâches à réaliser dans un temps précis.

Résultat

Les exigences fonctionnelles fixées dès le début par l'encadrant ont presque été menées à bien. En effet, les principaux modules caractérisant une application Desktop ont été intégrés.

Les exigences non fonctionnelles qui assurent la qualité et la performance de l'application ont toutes été développées. L'ergonomie de l'application est satisfaisante pour les utilisateurs. Les actions telles que contacter le technicien, accéder à l'aide en ligne, et à la charte d'utilisation seront possibles dans un autre temps.

Toutefois, certains points n'ont pu être développés par manque de savoir.

Améliorations envisagées

La version améliorée de cette application où on va faire des traitements des données et donner des interprétations et deviner le gain du mois prochain, détecter la période de la journée où on a beaucoup de clients et pourquoi pas réaliser une partie en ligne de réservation ceci sera réalisé le semestre prochain après avoir une idée sur l'IA, Machine Learning et Data Science.

6. Mode d'emploi de l'application : Gestion de Taxi



Le but de l'application s'agit d'entrer les données des chauffeurs et des courtiers (image, nom, CIN, numTel, photo...) dans une base de données afin de réaliser une bonne étude du marché ainsi de faire une gestion globale pour organiser ces stations de taxis et le développer d'après les statistiques.

Inscription

L'inscription du coursier se fait selon deux phases :

1er phase :

Inscription :

Le courtier entre ses informations personnelles : Nom, Prenom, Cin, Email, Mdp, Ville, NumTel.. Ses Données s'enregistrent dans la base de données "gestionTaxi" dans la table coursier.

Sign in Sign up

SIGN UP

nom

prenom

email

CIN

password

ville

phone

Sign up

2eme phase :

Finalisation d'inscription.

Dans ce champ là, le coursier finalise son Inscription en ajoutant sa propre photo de profile, sa carte nationale des deux cotes (Recto et Verso) et son certificat de travail.

SIGN UP

photo de profile

certificat

CIN Recto

CIN verso

Sign in

Authentification.

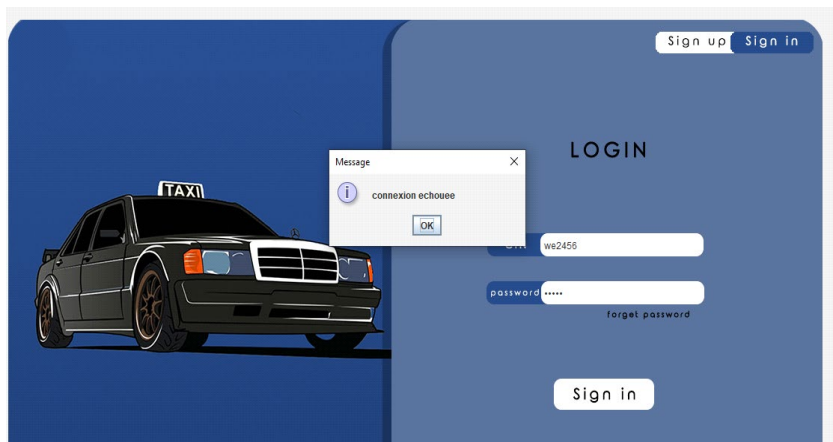
Après la connexion avec la base de données, par wampServer comme notre exemple la, l'authentification du coursier après son inscription se fait selon deux éléments principale : Son Cin et le Mot de Passe qui les aient entre dans la phase d'inscription.
On Reçoit deux résultats :

Connexion Valide :

Si le cin et le mdp sont correcte dont le courtier passe a l'interface d'accueil

Connexion Echouée :

Si le cin ou le mdp sont incorrecte.



Partie de gestion :

Ils ce constitue de deux parties essentiel :

Premier partie affiche les infos personnelles du courtier ainsi le courtier peut ajouter, modifier et supprimer le trajet d'un taxi.

Deuxième partie qui contient le tableau dont il affiche les taxis ajouter trier selon la ligne de ville d'arriver.



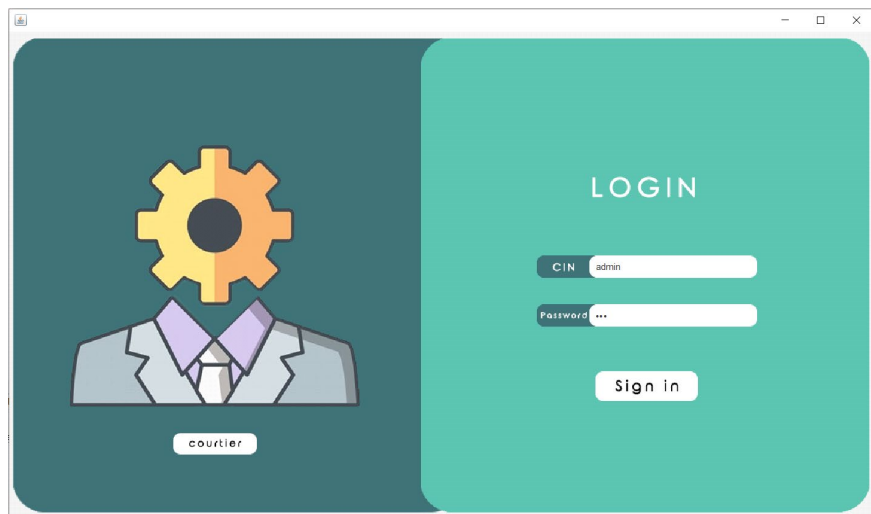
Partie Admin : Authentification :

Connexion Valide :

Si le cin et le mdp sont correcte dont
le coursier passe a l'interface d'accueil

Connexion Echoué :

Si le cin ou le mdp sont incorrecte.
Le cin est admin
Le mdp est 123



Taxi :

C'est la partie ou l'admin a le choix de
chercher, modifier, ajouter et supprimer
un taxi et il se compose de deux parties
une pour les traitement et l'autre pour
le tableau.



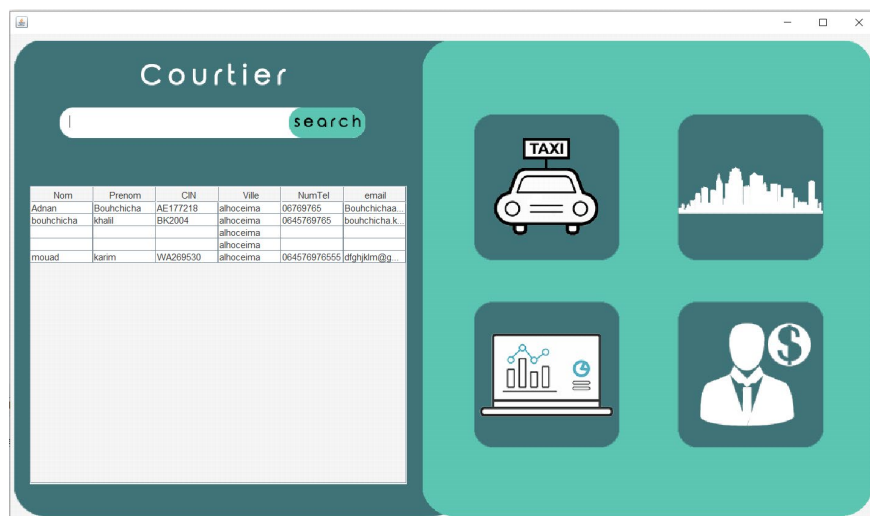
ville :

C'est la partie qui affiche toutes les villes
qui sont disponibles, ainsi que l'admin
peut ajouter une ville.



Courtier :

C'est la partie qui affiche tous les courtiers qui sont enregistrés dans le système avec leurs informations. (cin, nom, prénom, ville...)



The 'Courtier' interface displays a table of brokers with the following data:

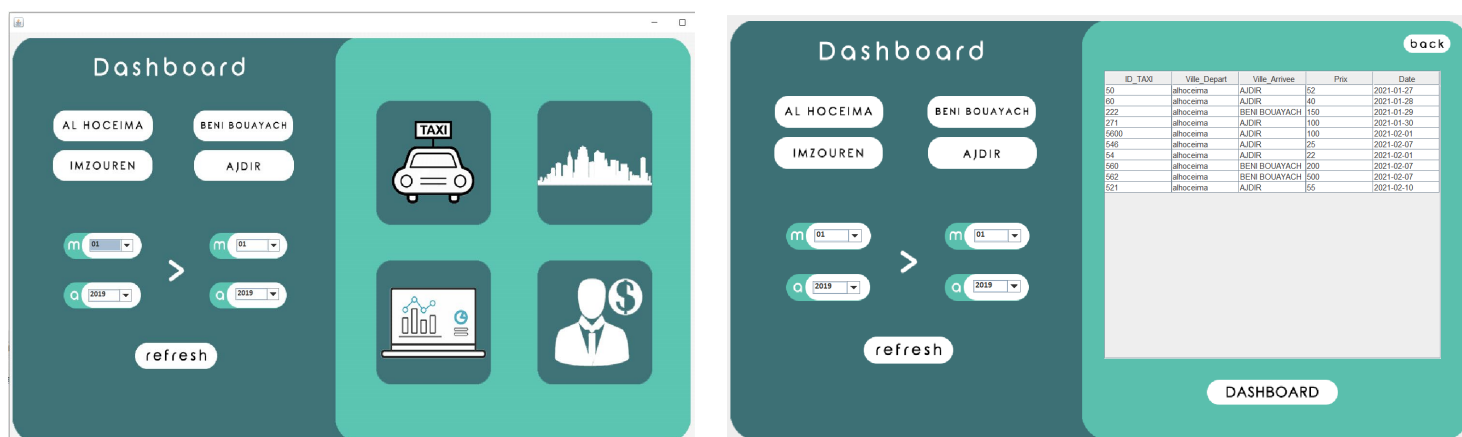
Nom	Prénom	CIN	Ville	NumTel	email
Adnan	Bouhchicha	AE177218	alhocema	06769765	Bouhchicha.k
bouhchicha	khali	BK2004	alhocema	0645769765	bouhchicha.k
			alhocema		
			alhocema		
mouad	karim	WA269530	alhocema	064576976555	dfghjklm@g

Navigation icons on the right include: TAXI (car icon), a city skyline, a laptop with a bar chart, and a person with a dollar sign.

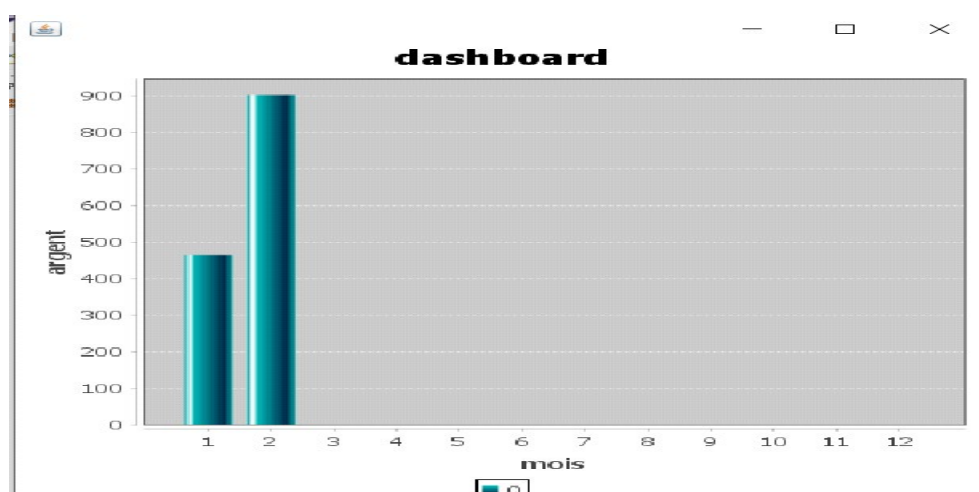
Dashboard :

C'est la partie qui affiche les données selon des courbes graphiques et qui se compose de deux champs principale :

Premier champ ou l'admin choisir la période de ce traitement en choisissant le mois et l'année.



Deuxième champ qui montre le graphe selon deux variables : l'argent et le mois.



7. Conclusion

Le projet qu'on a réalisé est beaucoup plus complexe que les tps que nous avons déjà faits. Il permet de mettre en relation plusieurs enseignements, utilisés sous une dimension professionnelle.

Tout d'abord, ce projet nous a permis d'appliquer les connaissances que nous avons acquises durant notre 3eme année d'étude, telles que la modélisation (UML), les différents langages de programmation.

Dans ce travail on a réalisé une application de gestion de taxis.

A travers des méthodes de travail, ce projet nous a permis de m'immerger dans un univers professionnel, même si notre application peut être encore meilleure.

Le projet nous a apporté donc une idée sur l'organisation dans le monde professionnel.

Ce rapport résume 15 jours de travail rigoureux.

Les sources d'information

<https://color.adobe.com/fr/create/color-wheel>

<https://fr.wikipedia.org/wiki/JAVA>

<https://fr.wikipedia.org/wiki/MySQL>

<https://www.eclipse.org/>

<https://www.java.com/fr/>

<https://www.phpmyadmin.net/>

<http://www.enseignement.polytechnique.fr/informatique/INF431/GrapheX/docgrapheX.pdf>

