

Application Desktop de réservation  
KOUSTA Khalid & El Arssi Zakaria



ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT  
TECHNIQUE DE MOHAMMEDIA  
UNIVERSITÉ HASSAN II DE CASABLANCA

## RAPPORT

# GESTION DES RÉSERVATIONS DES TRAINS



Java™



eclipse

**Réalisé par :**

KOUSTA Khalid

EL ARSI Zakariae

**Encadré par :**

Mme S. BELHOUS

## REMERCIEMENTS

Je tiens tout d'abord à remercier, toute l'équipe pédagogique de l'Ecole Normale Supérieure de l'Enseignement Technique de Mohammedia.

Avant d'entamer ce rapport, nous profitons de l'occasion pour remercier nos professeurs Madame **Safa BELHAOUS**.

Un grand merci à ma **mère** et mon **père**, pour leurs conseils, ainsi que pour leur soutien inconditionnel, à la fois moral et économique.

Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de mon stage. Un grand merci à **Najib Beddou** pour ses conseils concernant le style de mon rapport de stage.

Nous tenons à remercier nos professeurs de nous avoir incités à travailler en mettant à notre disposition leurs expériences et leurs compétences.

## DÉDICACE

Je dédie ce travail : A mes parents

EL Arssi Zakaria

Et

KOUSTA Khalid

Ceci est un pas vers l'accomplissement d'un acte encore plus grand

## Sommaire

INTRODUCTION GENERAL.....	7
Chapitre I : Traitement de cahier de charges.....	8
1. Cahier de charge de l'application à réaliser.....	8
2. Les tâches ajoutées au cahier de charges.....	8
2.1 La gestion des réservations des clients.....	9
3. Conclusion.....	9
CHAPITRE II : Conception et modélisation.....	10
1. Introduction : .....	10
2. Définition STAR UML : .....	10
3. Diagramme de cas d'utilisation : .....	10
3.1 Acteur.....	10
3.2 Cas d'utilisation.....	11
4. Diagramme de cas d'utilisation pour Les Clients : .....	11
5. Diagramme de cas d'utilisation pour les Employés : .....	12
6. Diagramme de cas d'utilisation final : .....	12
7. Diagramme de classe : .....	13
8. Conception Graphic .....	13
.....	14
9. Conclusion : .....	14
CHAPITRE III : Intégration et test du système.....	15
1. Calendrier de développement .....	15
2. Détails de la mise en œuvre.....	15
a. Aperçu du système.....	15
b. JAVA .....	15
c. Eclipse .....	16
d. Base de données .....	17
e. XAMPP.....	18
f. JDBC .....	19
g. Tableaux de bases de données .....	20
h. API JavaMail .....	21

Application Desktop de réservation  
KOUSTA Khalid & El Arssi Zakaria

3. Résultat du Mise en œuvre.....	22
a. Espace administration.....	22
b. Espace Client .....	25
CHAPITRE IV : Résultats et discussion.....	28
1. Résultats des tests .....	28
1.1 Tests unitaires .....	28
CHAPITRE V : Résumé et conclusions .....	30
1. Résumé des résultats .....	30
2. Principales difficultés rencontrées.....	30

## Liste des Figure

Figure 1 : LOGO du StarUML .....	10
Figure 2: Acteur.....	10
Figure 3 : Cas d'utilisation .....	11
Figure 4 : Use-Case du Client .....	11
Figure 5 : Use-case s'administrateur.....	12
Figure 6 : Use-case général .....	12
Figure 7 : Diagramme des classes .....	13
Figure 8 : Diagram des interfaces .....	14
Figure 9 : Calendrier de développment .....	15
Figure 10: LOGO du JAVA.....	15
Figure 11 : LOGO du Eclipse .....	16
Figure 12 : Hello world avec Eclipse.....	16
Figure 13 : Eclipse Description de la fenêtre .....	17
Figure 14 : LOGO du XAMPP .....	18
Figure 15 : XAMPP Control pane .....	19
Figure 16 : utilisation de l'API JDBC pour accéder aux MySQL .....	20
Figure 17 : Code de connexion au MySQL .....	20
Figure 18 : Utilisation de l'API JavaMail .....	21
Figure 19 : Initialisation du JavaMail API .....	21
Figure 20 : Fenêtre principale .....	22
Figure 21 : Fenêtre d'identification .....	22
Figure 22 : Fenêtre principale d'administration .....	23
Figure 23 : Fenêtre de gestion des trains.....	23
Figure 24 : base de données des trains.....	24
Figure 25 : Fenêtre de gestion des gares .....	24
Figure 26 : base de données des gares .....	25
Figure 27 : : Fenêtre du choix du Train .....	25
Figure 28 : Fenêtre principale du client .....	25
Figure 29 : Fenêtre de paiement.....	26
Figure 30 : Message de confirmation .....	26
Figure 31 : Base de données des clients .....	26
Figure 32 : Réception d'email avec ticket .....	27
Figure 33 : Billet reçu .....	27

## INTRODUCTION GENERAL

De nos jours, la gestion des données d'une façon automatisée occupe une place privilégiée dans le monde de l'informatique. Grâce à cette technologie, le temps de recherche des données a chuté de manière considérable, ce qui laisse croire qu'elle a encore de beaux jours devant elle. C'est dans ce contexte que plusieurs sociétés essayent de profiter au maximum possible de ces technologies afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression.

Objectif :

Grâce à ce système, une approche pratique a été adoptée pour comprendre le fonctionnement du système de réservation des trains. Ce système comprendra tous les champs nécessaires requis lors du système de réservation d'un train. Ce système sera facile à utiliser et peut être utilisé par n'importe qui. L'idée de base de ce projet est de sauvegarder les données dans une base de données centrale à laquelle toute personne autorisée à accéder puisse accéder pour obtenir des informations et économiser du temps et de la charge de travail pour ses clients.

Système proposé :

Grâce à ce système, les clients n'ont pas à faire la queue et ils pourront obtenir leurs réponses en un clic. Grâce à ce système, un formulaire en ligne sera mis à la disposition des passagers pour leur permettre de remplir leurs coordonnées avec les détails de leur voyage. Pour réserver leurs places, les utilisateurs devront d'abord fouiller leurs trains selon leurs besoins et, après avoir obtenu le bon train, ils devront fournir le numéro du train. Après avoir appuyé sur le bouton Suivant, il affichera son statut. Un formulaire d'annulation sera également disponible et pour annuler leur réservation,

## Chapitre I : Traitement de cahier de charges

### 1. Cahier de charge de l'application à réaliser

Cette étude de cas concerne un système simplifié de réservation de train pour une agence de voyage. Les rencontres avec les futurs utilisateurs et les experts ont permis de résumer la connaissance du domaine avec les phrases suivantes :

- La compagnie propose différents trains
- Une compagnie a un nom, chiffre d'affaire, type (Sarl, SA...), adresse
- Un train est ouvert à la réservation et refermé sur ordre de la compagnie
- Un train a un numéro, une date de départ et d'arrivée, une heure de départ et d'arrivée.
- Un client peut réserver un ou plusieurs trains pour des passagers différents. Il possède un numéro de CIN, un nom, un prénom, une adresse et un numéro de téléphone.
- Une réservation concerne un seul train et un seul passager
- Une réservation peut être annulée ou confirmée
- Un train a une gare de départ ou une gare d'arrivée
- Une gare a un nom, une adresse et nombre des trains par jours
- Une réservation a un jour et une heure de départ, et un jour et une heure d'arrivée
- Une gare a un nom, une adresse et nombre des trains par jours
- Une réservation a un jour et une heure de départ, et un jour et une heure d'arrivée
- Un train peut comporter des correspondances dans des gares.

### 2. Les tâches ajoutées au cahier de charges

Ainsi en ajoute un système pour envoyer des **e-mails** aux clients sur **Google Gmail**.

- Une interface de paiement (soit par carte bancaire soit paiement manuel dans les agences associées).
- Un ticket porte les informations de passager et le Prix de voyage.

On peut diviser le travail donné par le cahier de charges en deux parties :

La première partie concerne la gestion des réservations des clients,

Et la deuxième partie pour l'espace administratif,



## 2.1 La gestion des réservations des clients

### 2.1.1 *Choix de train :*

Dans cette partie quand l'utilisateur appuyé sur le bouton d'espace client.

Le client choisir un train parmi les trains disponibles, il choisit la date de départ et d'arrivée ainsi que la gare de départ et la gare d'arrivée(ville), ainsi que la classe.

Ensuite il doit remplir leurs informations personnelles, NOM, PRENOM, CIN, TELE, ADRESSE E-MAIL, ADRESSE.

Quand il choisit et terminer un message de confirmation s'affiche, et s'il appui sur NO la fenêtre de choix de train reste ouverte, et s'il appui sur OUI une autre fenêtre s'affiche.

### 2.1.2 *Paiement :*

Après que le client entrer ses informations, une fenêtre de paiement s'affiche.

Le client doit choisir le type de paiement soit par carte bancaire ou bien par agence, si le client choisit le paiement par carte il doit l'entrer les informations de la carte bancaire.

Et à la fin de cette tâche un message de confirmation s'affiche, et un email qui contient le ticket s'adresse à la boîte e-mail de client.

### 2.1.3 *L'espace administratif*

#### ➤ Authentification :

Dans cette partie l'employé doit entrer un Nom d'utilisateur et un mot passe pour consulter la gestion des trains, pour que l'application reste sécurisée,

#### ➤ Gestion des trains :

Dans cette partie l'employé à l'autorisation de supprimer / ajouter, ouvrir/fermer un train. Et à la fin de cette tâche un message de confirmation s'affiche.

## 3. Conclusion

A la fin de ce chapitre, on a bien étudié les besoins du client ; on a présente l'ensemble des fonctionnalités du futur portail de manière organisée dans les différents cycles de l'application

## CHAPITRE II : Conception et modélisation

### 1. Introduction :

Dans ce chapitre on va étudier la modélisation de cahier de charges, on utilise des logiciels pour la modélisation afin de construire un diagramme de cas d'utilisation ainsi qu'un diagramme de classe.

Dans notre travail nous avons utilisé le logiciel STAR UML pour faire la modélisation, Alors c'est quoi ce logiciel et à quoi sert ?

### 2. Définition STAR UML :

StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible.



Figure 1 : LOGO du StarUML

### 3. Diagramme de cas d'utilisation :

#### 3.1 Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom (son rôle) inscrit dessous.

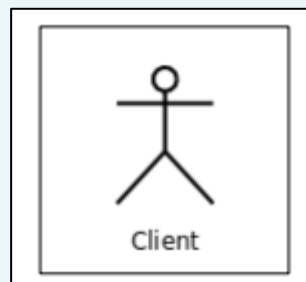


Figure 2: Acteur

### 3.2 Cas d'utilisation

Un cas d'utilisation est un service rendu à un acteur : c'est une fonctionnalité de son point de vue. Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.



Figure 3 : Cas d'utilisation

## 4. Diagramme de cas d'utilisation pour Les Clients :

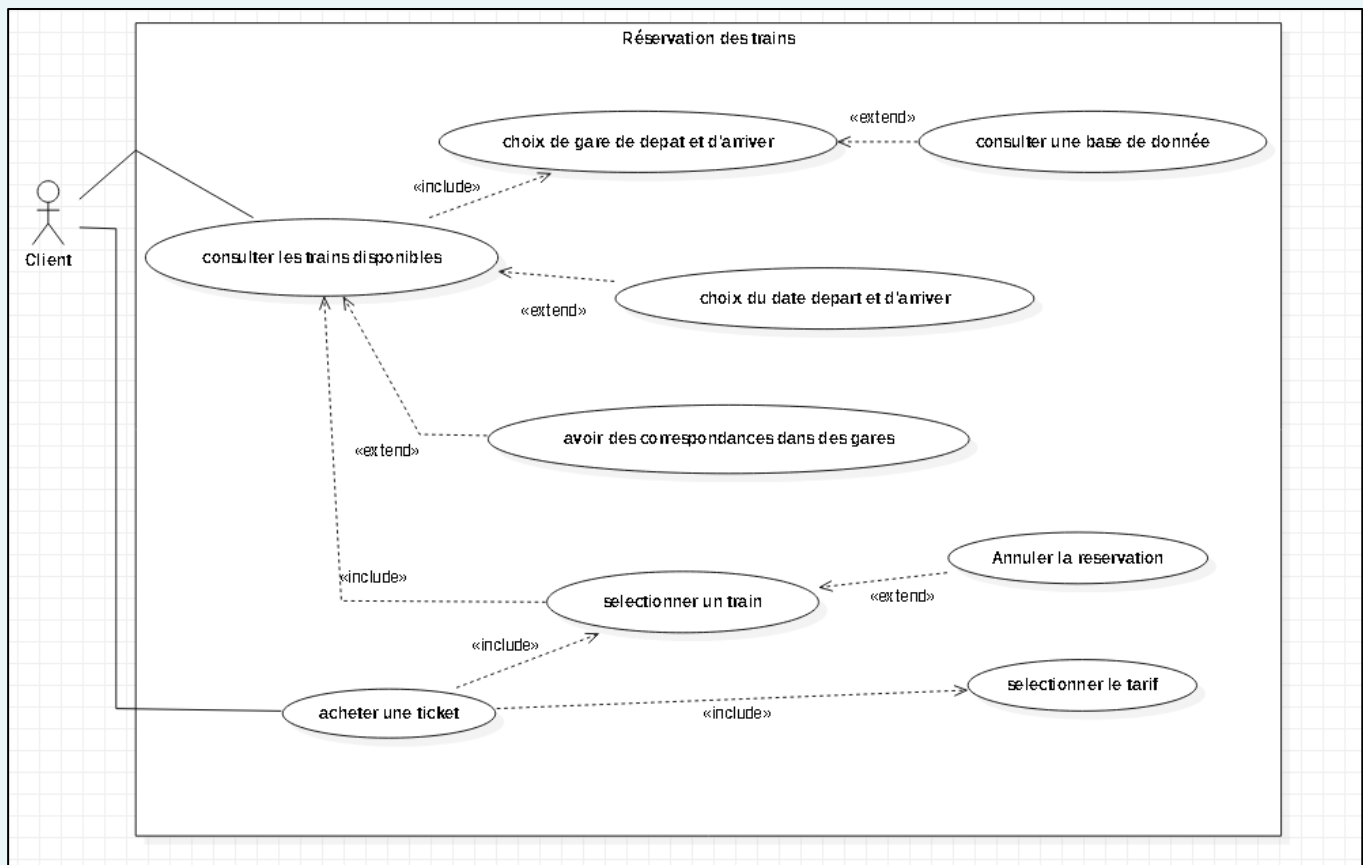


Figure 4 : Use-Case du Client

## 5. Diagramme de cas d'utilisation pour les Employés :

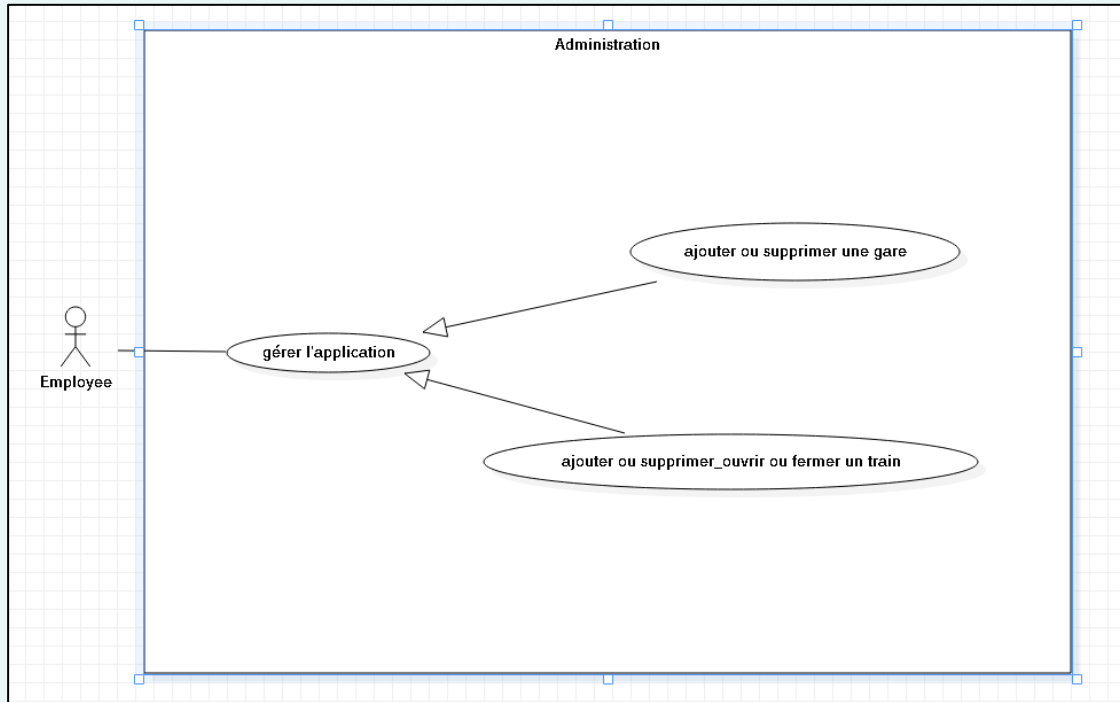


Figure 5 : Use-case s'administrateur

## 6. Diagramme de cas d'utilisation final :

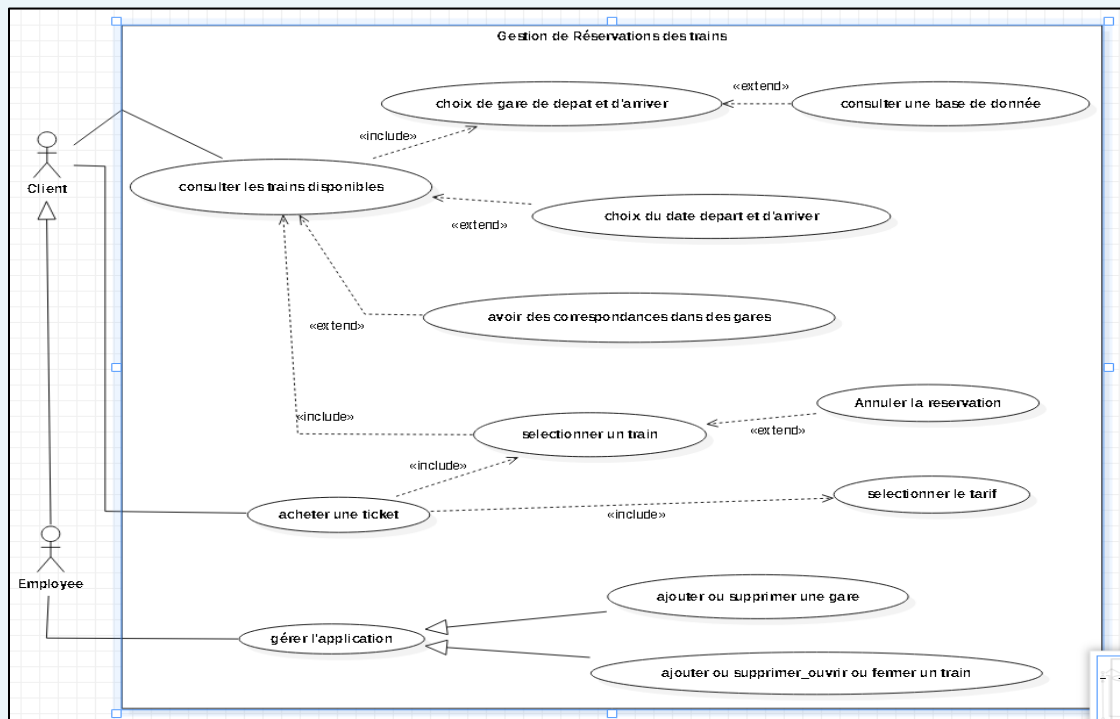


Figure 6 : Use-case général

## 7. Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

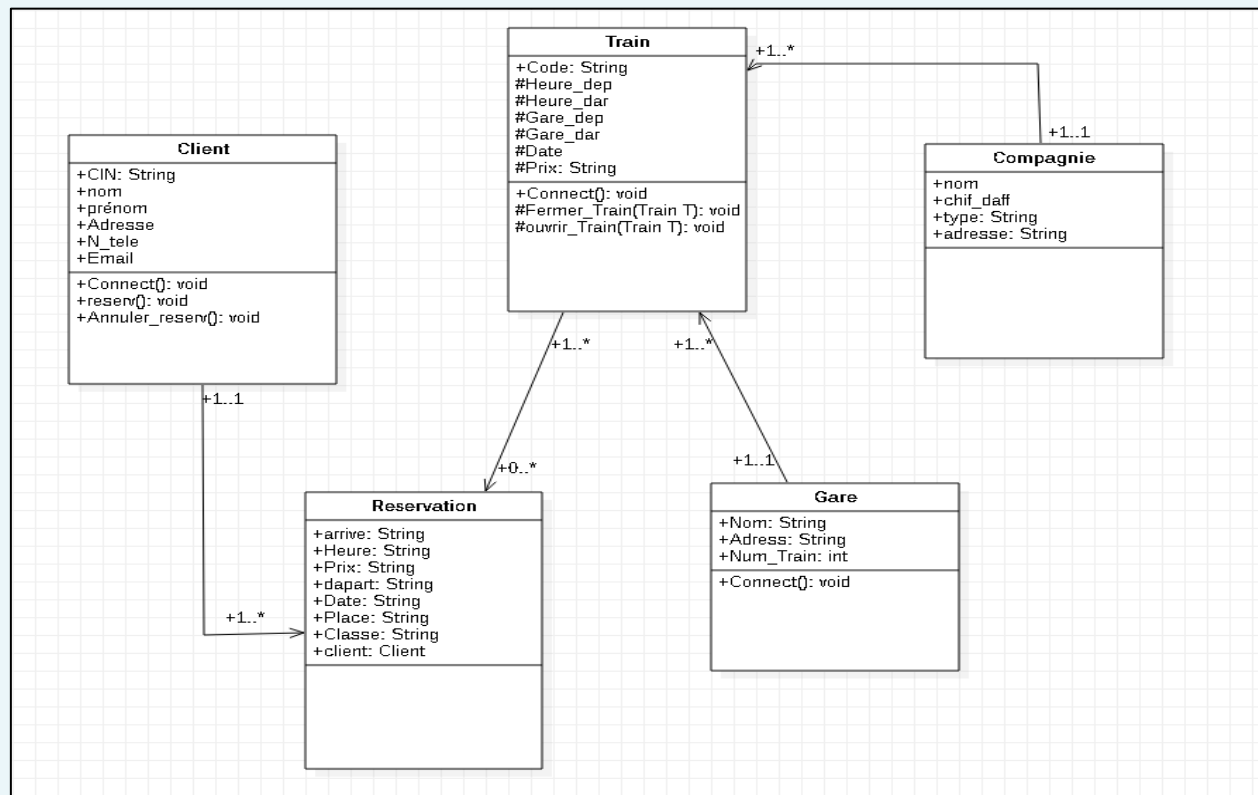


Figure 7 : Diagramme des classes

## 8. Conception Graphic

En commençant par le côté GUI, l'application aura un total de 7 fenêtres

Application Desktop de réservation  
KOUSTA Khalid & El Arssi Zakaria

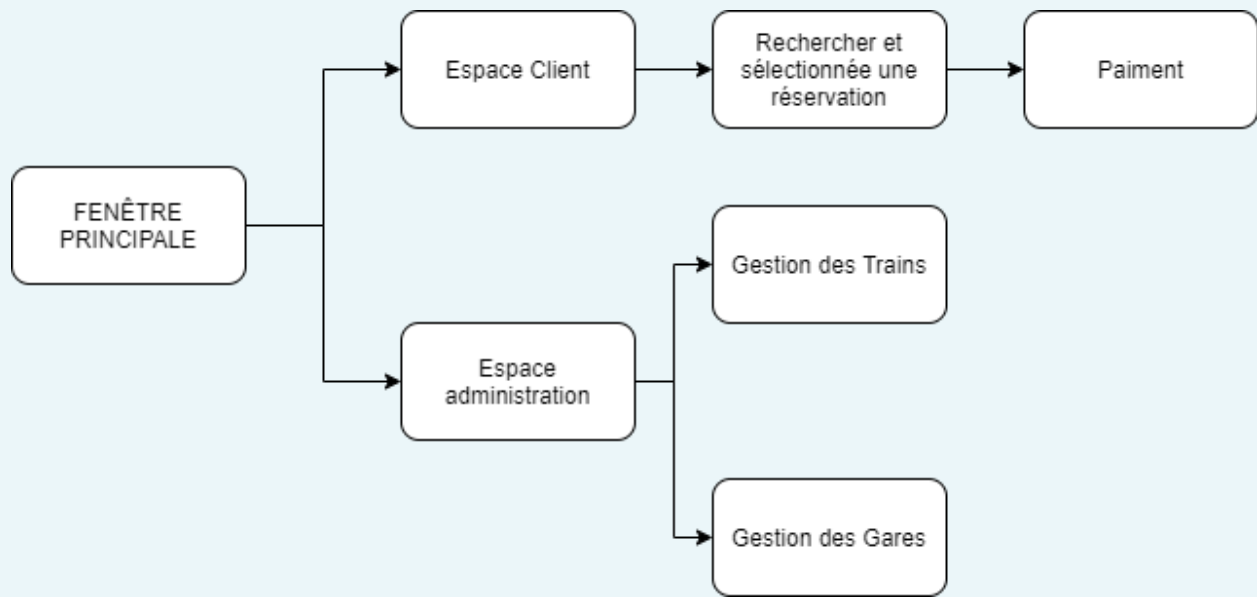


Figure 8 : Diagram des interfaces

## 9. Conclusion :

Dans ce chapitre, nous avons conçu et documenté le code que nous devons produire. Dans cette phase, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. Dans le chapitre suivant nous allons étudier en détails les outils et les langages utilisés durant la phase de construction.

## CHAPITRE III : Intégration et test du système

### 1. Calendrier de développement

Le travail sur Le SYSTEME DE RESERVATION a commencé le 1 Janvier 2021 et devrait être terminé le 29 Janvier 2021. Le diagramme suivant estimée la durée des différentes phases du travail de développement :

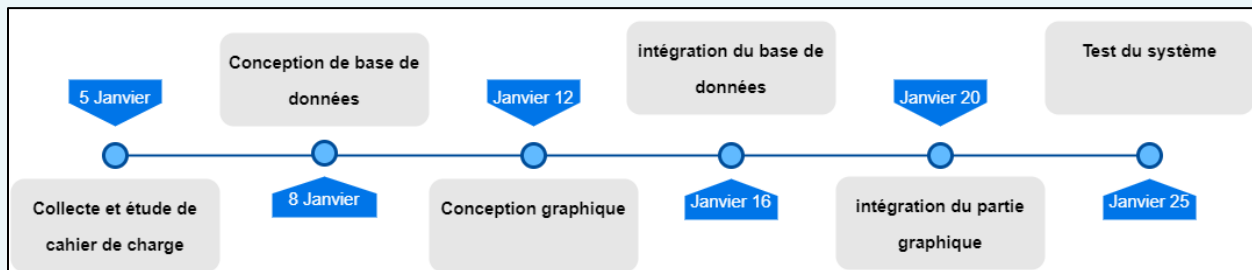


Figure 9 : Calendrier de développment

### 2. Détails de la mise en œuvre

#### a. Aperçu du système

Le système de réservation des billets du trains est une application Desktop. Elle a été développée en utilisant JAVA. Une base de données MySQL composée de plusieurs tables est utilisée pour le stockage des données.

Cette application est une application de réservation des voyages en train, qui est capable de réserver des voyages nationaux et internationaux en quelques étapes simples et au meilleur prix.

#### b. JAVA



Figure 10: LOGO du JAVA

**JAVA** est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionne pas si Java

n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

Java est un langage de programmation inspiré du langage C++, avec un modèle de programmation orienté objet. Java permet de créer des applications complètes. Il peut également servir à créer un petit module d'application, dit applet, à intégrer dans une page Web.

Les principales caractéristiques de Java sont les suivantes :

- Les programmes créés sont portables. Le programme source est compilé dans un « code », qui peut être exécuté sur un serveur ou un client doté d'une machine virtuelle Java.
- Le code est robuste.
- Java est orienté objet.

### c. Eclipse



Figure 11 : LOGO du Eclipse

**Eclipse IDE** est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

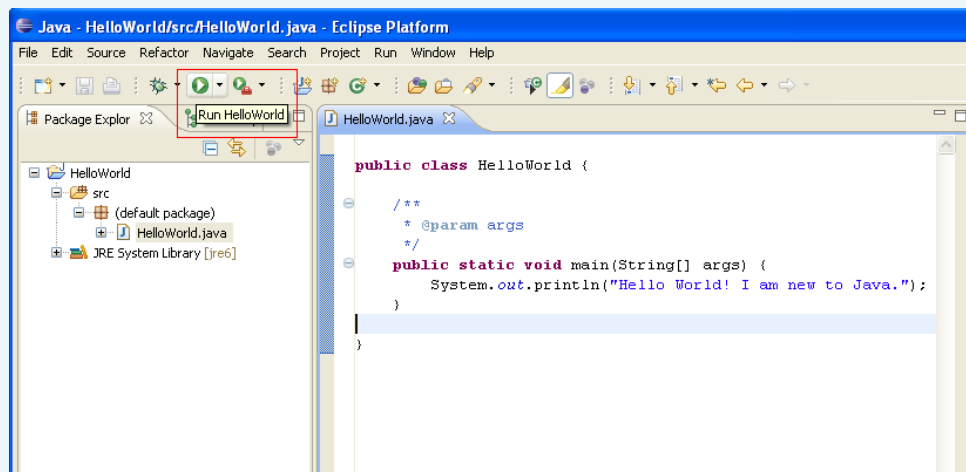


Figure 12 : Hello world avec Eclipse



## Application Desktop de réservation KOUSTA Khalid & El Arssi Zakaria

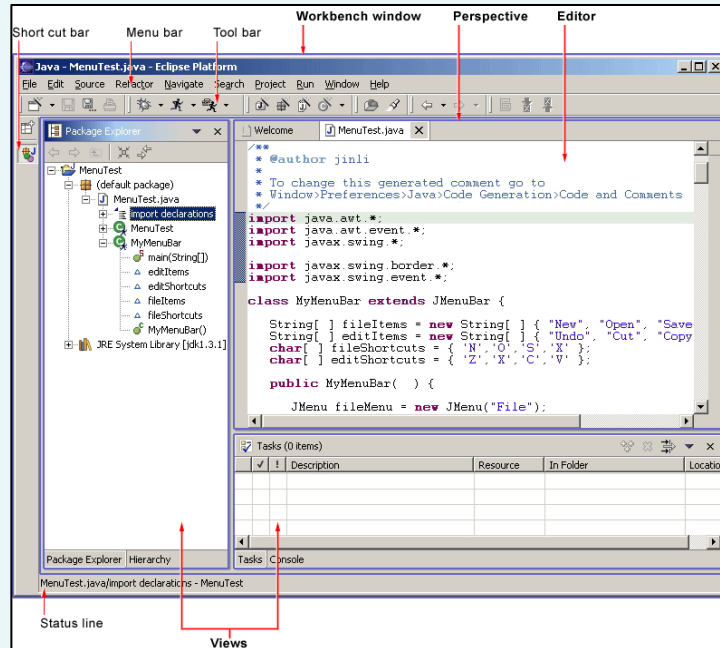


Figure 13 : Eclipse Description de la fenêtre

### d. Base de données

La base de données (BDD, ou DB en anglais) est un système qui enregistre des informations, mais pas n'importe comment : ces informations sont toujours classées. Et c'est ça qui fait que la BDD est si pratique : c'est un moyen extrêmement simple de ranger des informations ! Grossièrement, une BDD peut être vue comme un ensemble de tableaux, des structures contenant donc des lignes et des colonnes et dans lesquelles nos données sont rangées.

Il existe un vocabulaire spécifique pour désigner les différents éléments composant une BDD :

- La base désigne le volume englobant l'ensemble, la boîte qui contient tous les tableaux ;
- Une table désigne un tableau de données, elle contient des lignes et des colonnes ;
- Une entrée désigne une ligne ;
- Un champ désigne une colonne.

Il n'existe pas qu'une seule solution pour créer et gérer des bases de données. Voici une liste des principaux acteurs de ce marché :

- MySQL : solution libre et gratuite, c'est le SGBD le plus répandu. C'est d'ailleurs celui que nous allons utiliser dans ce cours !
- PostgreSQL : solution libre et gratuite, moins connue du grand public mais proposant des fonctionnalités inexistantes dans MySQL ;
- Oracle : solution propriétaire et payante, massivement utilisée par les grandes entreprises. C'est un des SGBD les plus complets, mais un des plus chers également ;
- SQL Server : la solution propriétaire de Microsoft ;

- DB2 : la solution propriétaire d'IBM, utilisée principalement dans les très grandes entreprises sur des *Mainframes*.

e. XAMPP



Figure 14 : LOGO du XAMPP

XAMPP est synonyme de : **Multi-plateforme (X), Apache (A), MySQL (M), PHP (P) Perl (P).**

C'est une distribution Apache simple et légère qui permet aux développeurs de créer facilement un serveur web local à des fins de test. Tout ce dont vous avez besoin pour configurer un serveur web – application serveur (Apache), base de données (MySQL) et langage de script (PHP) – est inclus dans un simple fichier extractible. XAMPP est également multi-plateforme, ce qui signifie qu'il fonctionne aussi bien sur Linux, Mac et Windows.

XAMPP a quatre composants principaux. Ceux-ci sont :

- Apache : Apache est l'application de serveur Web qui traite et livre le contenu Web à un ordinateur. Apache est le serveur web le plus populaire en ligne, alimentant près de 54% des sites web.
- MySQL : Chaque application web, simple ou compliquée, nécessite une base de données pour stocker les données collectées. MySQL, qui est open source, est le système de gestion de base de données le plus populaire au monde.
- PHP : PHP signifie Hypertext Preprocessor. C'est un langage de script côté serveur qui alimente certains des sites Web les plus populaires au monde, y compris WordPress et Facebook. Il est open source, relativement facile à apprendre, et fonctionne parfaitement avec MySQL, ce qui en fait un choix populaire pour les développeurs web.
- Perl : Perl est un langage de programmation dynamique de haut niveau largement utilisé dans la programmation réseau, l'administration système, etc. Bien que moins populaire à des fins de développement web, Perl a beaucoup d'applications de niche.

- Contrôleur de gestion XAMPP

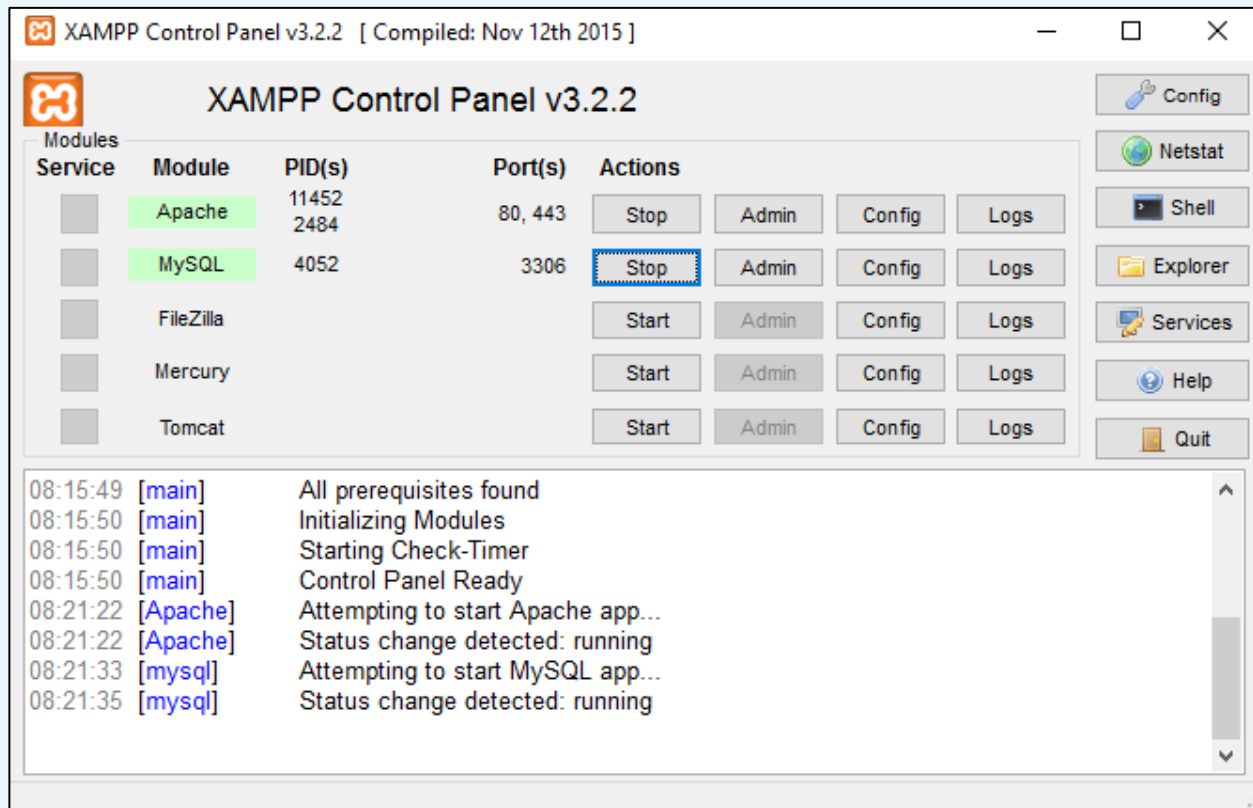


Figure 15 : XAMPP Control pane

Le panneau de contrôle XAMPP vous donne un contrôle complet sur tous les composants XAMPP installés. Vous pouvez utiliser le CP pour démarrer / arrêter différents modules, lancer le shell Unix, ouvrir l'explorateur Windows et voir toutes les opérations s'exécutant en arrière-plan. Les actions proposées par cette interface sont les suivantes :

- Config : Vous permet de configurer XAMPP ainsi que des composants
- NetStats : Affiche les processus courants sur votre ordinateur local
- Shell : Ouvre une cellule UNIX
- Explorer : Ouvre le dossier XAMPP dans Windows
- Services : Affiche les services fonctionnant en arrière-plan
- Aide : propose des liens d'aide par le biais du forum utilisateur
- Terminer : Fermer le Contrôleur de gestion XAMPP

#### f. JDBC

C'est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis vos applications Java à des données rangées sous forme de tables. Dans la très grande majorité des cas, il s'agira bien entendu de bases de données stockées dans un SGBD ! Les actions rendues possibles par cette API sont :

Application Desktop de réservation  
KOUSTA Khalid & El Arssi Zakaria

- La connexion avec le SGBD ;
- L'envoi de requêtes SQL au SGBD depuis une application Java ;
- Le traitement des données et éventuelles erreurs retournées par le SGBD lors des différentes étapes du dialogue (connexion, requête, exécution, etc.).

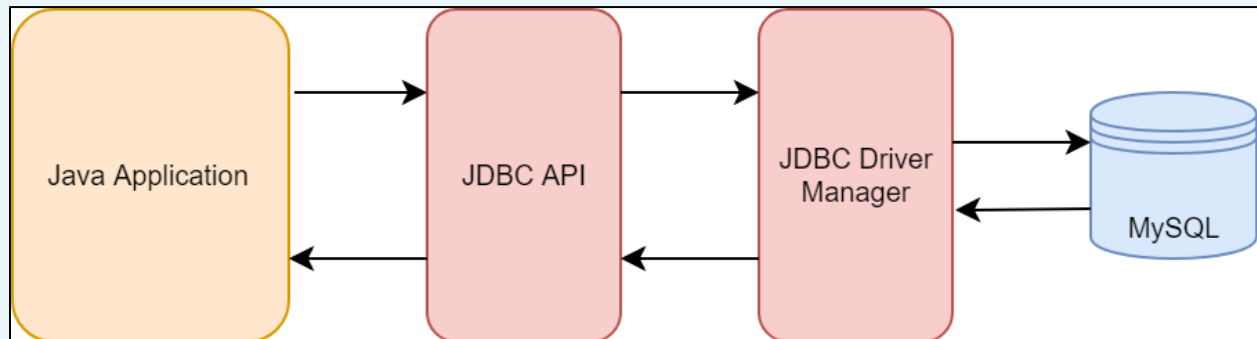


Figure 16 : utilisation de l'API JDBC pour accéder aux MySQL

En utilisant le code suivant, nous pouvons nous connecter avec succès à la base de données MySQL

```
678 public void Connect(){
679     try
680     {
681         // create a mysql database connection
682         String myDriver = "com.mysql.jdbc.Driver";
683         String myUrl = "jdbc:mysql://localhost/africatgv";
684         Class.forName(myDriver);
685         conn = DriverManager.getConnection(myUrl, "root", "");
686     }
687 }
688 catch(Exception e)
689 {
690     JOptionPane.showMessageDialog(null, "Impossible de se connecter à la base de données.", "Information", JOptionPane.WARNING_MESSAGE);
691 }
692 }
693 }
694 }
```

Figure 17 : Code de connexion au MySQL

g. Tableaux de bases de données

Nom de la table	Description
Trains	Enregistrer les détails de chaque train
Gares	Enregistrer les détails de chaque gares
Clients	Enregistrer les détails de chaque client

#### h. API JavaMail

JavaMail est une API qui permet d'utiliser le courrier électronique (e-mail) dans une application écrite en Java (application cliente, applet, servlet, EJB, ... ). Son but est d'être facile à utiliser, de fournir une souplesse qui permette de la faire évoluer et de rester le plus indépendant possible des protocoles utilisés.

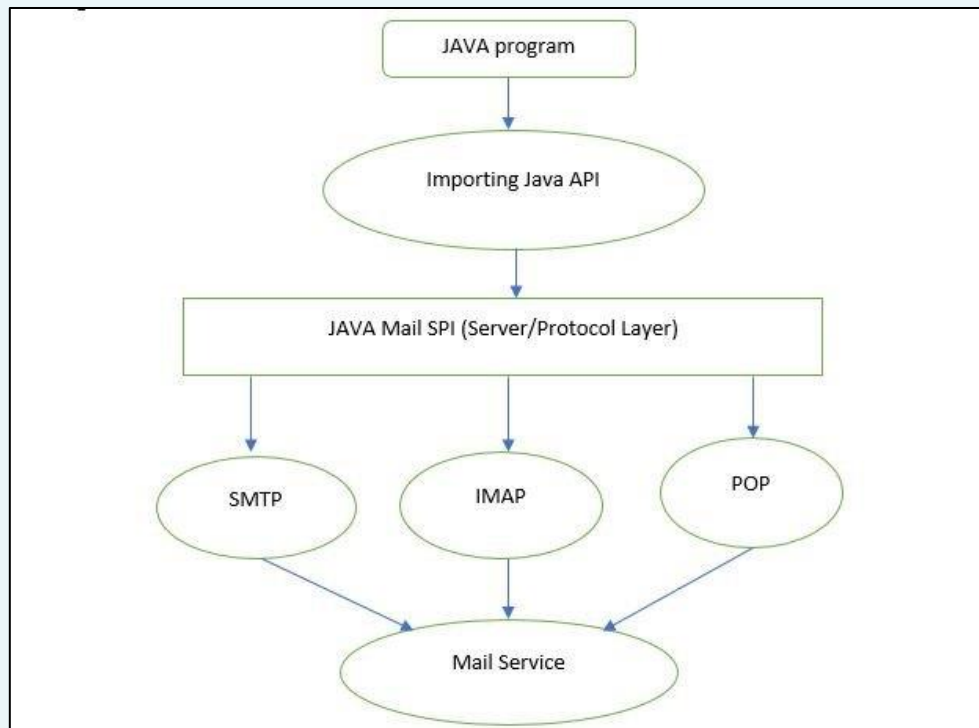


Figure 18 : Utilisation de l'API JavaMail

En utilisant le code suivant, nous pouvons initialiser l'API du courrier électronique

```
33 public void run() {
34     // TODO Auto-generated method stub
35
36     String host="smtp.gmail.com";
37     final String user="enset.tgv@gmail.com";
38     final String password="TGVTVGTGV";
39
40     String to = Email;
41
42     //imported code
43     Properties props = System.getProperties();
44     props.put("mail.smtp.starttls.enable", "true");
45     props.put("mail.smtp.host", host);
46     props.put("mail.smtp.user", user);
47     props.put("mail.smtp.password", password);
48     props.put("mail.smtp.port", "587");
49     props.put("mail.smtp.auth", "true");
50     props.put("mail.smtp.ssl.trust", "smtp.gmail.com");
51
52
53     Session session = Session.getDefaultInstance(props,
54     new javax.mail.Authenticator() {
55     protected PasswordAuthentication getPasswordAuthentication() {
56         return new PasswordAuthentication(user,password);
57     }
58     });
```

EMAIL

Mot de passe

Figure 19 : Initialisation du JavaMail API

### 3. Résultat du Mise en œuvre

Après le démarrage de l'application, la Fenêtre principale s'affiche, sur laquelle l'utilisateur peut cliquer pour accéder à l'espace client ou à l'espace d'administration.

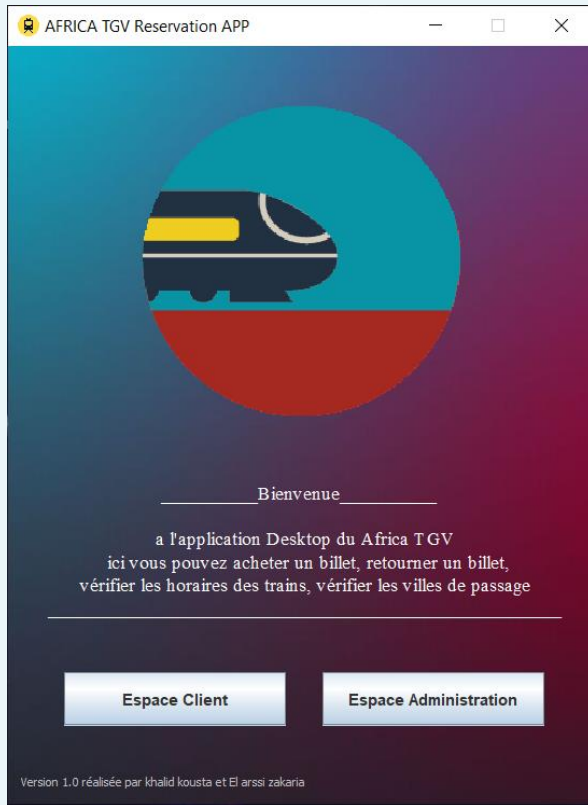


Figure 20 : Fenêtre principale

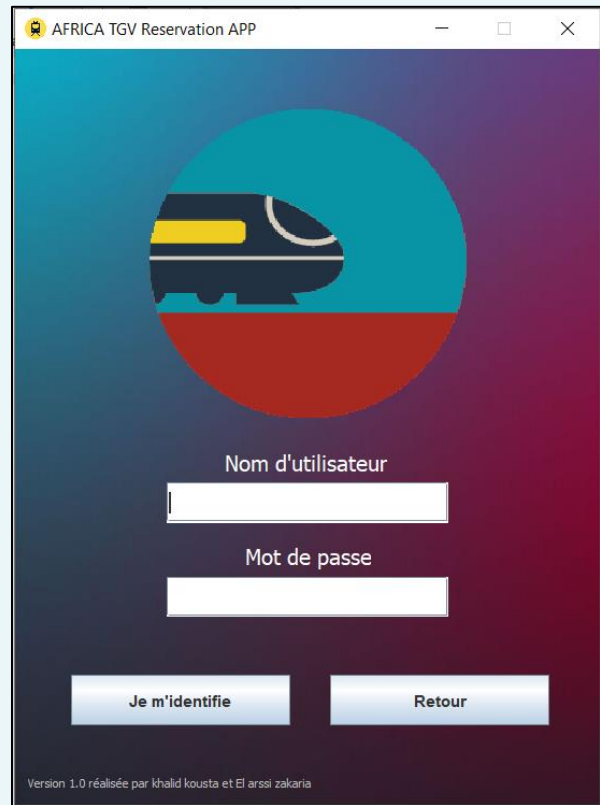


Figure 21 : Fenêtre d'identification

Si l'utilisateur click sur espace Administration, Une fenêtre d'identification sera affichée. Chaque employée a sont propre compte, Ici on va essayer d'accès come un employé appeler Ahmed avec un Nom utilisateur «ENSET » et un Mot de passe « ENSET ».

#### a. Espace administration

Dans la fenêtre d'administration principale on peut modifier certaines options liées aux la compagnie telles que :

- Gestion des Train :
  - Ajoute des trains
  - Suppression des trains
  - Ouvrir ou fermée un train
  - Modifier les trains
- Gestion des Gares :
  - Ajoute des gares
  - Suppression des gares

## Application Desktop de réservation KOUSTA Khalid & El Arssi Zakaria

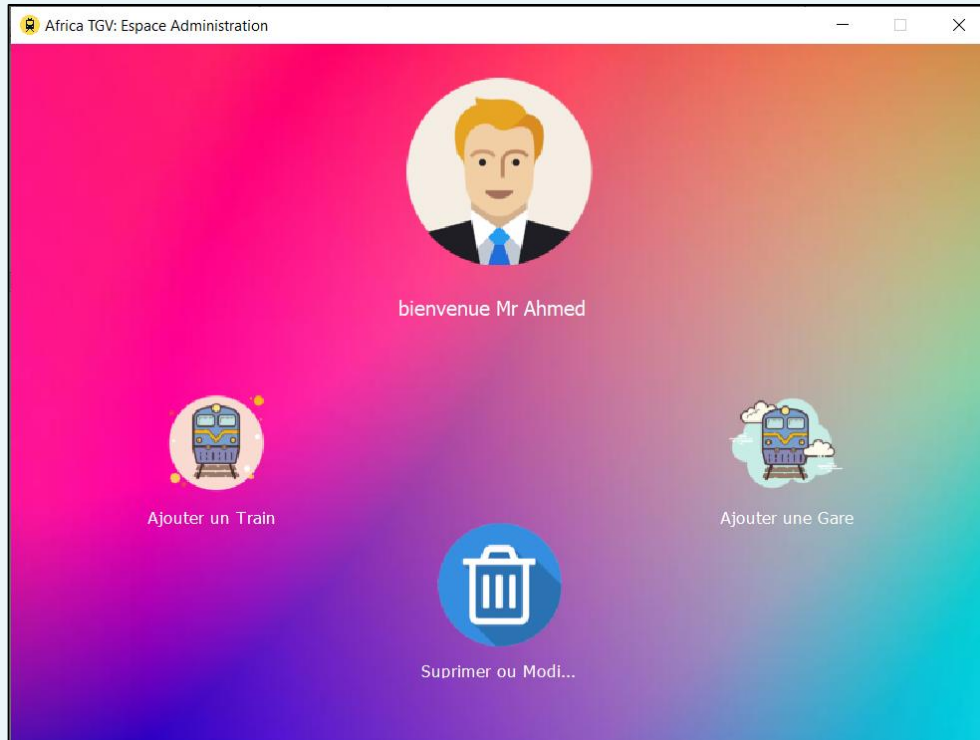


Figure 22 : Fenêtre principale d'administration

**Africa TGV: Espace Administration**

**Vous pouvez ajouter et modifier les Train...**

**Ajouter un Train**

Code\* Gare de dépa... Date\* Heure\* Gare d'arivée\* Heure\* Prix\*

OUJDA    OUJDA

**Supprimer ou Desactiver un Train**

Code

A001

**Train disponible**

Code	Gare de départ	Date de départ	Heure	Gare de d'arriv...	Heure	Prix	ETAT
A001	OUJDA	Jan 26, 2021	10:00	TAZA	14:22	120 DH	ACTIVER
A002	OUJDA	Jan 27, 2021	11:00	FES	15:00	150 DH	ACTIVER
A003	OUJDA	Jan 27, 2021	14:00	FIGUIG	17:00	150 DH	ACTIVER
A004	OUJDA	Jan 28, 2021	08:00	TANGER	15:00	220 DH	ACTIVER
A005	OUJDA	Jan 27, 2021	09:35	EL JADIDA	15:22	260 DH	ACTIVER
A006	OUJDA	Jan 29, 2021	14:19	MARRAKECH	22:30	300 DH	ACTIVER
A007	OUJDA	Jan 27, 2021	10:00	RABAT	16:00	210 DH	ACTIVER

Figure 23 : Fenêtre de gestion des trains



## Application Desktop de réservation KOUSTA Khalid & El Arssi Zakaria

Toutes les données saisies dans cette fenêtre sont directement enregistrées dans la base de données

	ID	Code	Gare_dep	Date	Heure_dep	Gare_dar	Heure_dar	Prix	statut
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	42	KM05	FES	Jan 27, 2021	01:15	EL JADIDA	08:16	260 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	50	BM33	OUJDA	Jan 29, 2021	07:55	CASABLANCA	13:35	360 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	35	A004	OUJDA	Jan 28, 2021	08:00	TANGER	15:00	220 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	46	ZT88	MARRAKECH	Jan 15, 2021	09:33	RABAT	15:36	270 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	36	A005	OUJDA	Jan 27, 2021	09:35	EL JADIDA	15:22	260 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	48	B457	OUJDA	Jan 27, 2021	09:45	CASABLANCA	18:16	330 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	32	A001	OUJDA	Jan 26, 2021	10:00	TAZA	14:22	120 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	38	A007	OUJDA	Jan 27, 2021	10:00	RABAT	16:00	210 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	33	A002	OUJDA	Jan 27, 2021	11:00	FES	15:00	150 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	49	H355	KENITRA	Jan 31, 2021	13:00	OUJDA	13:30	520 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	41	WT66	TANGER	Jan 27, 2021	13:35	OUJDA	18:56	300 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	34	A003	OUJDA	Jan 27, 2021	14:00	FIGUIG	17:00	150 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	52	FS85	OUJDA	Jan 31, 2021	14:00	CASABLANCA	19:20	360 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	37	A006	OUJDA	Jan 29, 2021	14:19	MARRAKECH	22:30	300 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	39	ZN12	RABAT	Jan 13, 2021	15:00	CASABLANCA	16:05	50 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	40	ZX63	CASABLANCA	Jan 27, 2021	15:00	TANGER	20:35	160 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	43	B001	OUJDA	Jan 21, 2021	15:00	TAZA	18:35	160 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	47	T888	SAFI	Jan 15, 2021	15:00	BENGUERIR	20:23	160 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	45	B638	EL JADIDA	Jan 28, 2021	15:33	CASABLANCA	20:16	160 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	51	KU59	OUJDA	Jan 31, 2021	16:00	CASABLANCA	23:11	400 DH	ACTIVER
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	44	C358	FIGUIG	Jan 29, 2021	16:35	CASABLANCA	01:33	320 DH	ACTIVER

Figure 24 : base de données des trains

Africa TGV: Espace Administration

**Vous pouvez ajouter ou Supprimer les Gare...**

**Ajouter Gare**

Nom

Adresse

Ajouter

**Supprimer Gare**

Gare

OUJDA

Supprimer

**Gares disponible**

NOM	ADRESSE
OUJDA	Boulevard Abdellah Chefcha...
CASABLANCA	AIN SEBAA
FIGUIG	Hay BERKOUKESS Ksar Z...
NADOR	Gare Nador-Ville
TAZA	bd Bir Anzarane TAZA
FES	bd des Almohades .Quartier: f...
TANGER	hay Mghogha
BOUARFA	Rt bouarfa
MEKNES	Nouvelle-Ville Hamria
KENITRA	bd Mohamed Diouri, Maamora...
AEROPORT	av @roport Mohammed V

**Carte**

Figure 25 : Fenêtre de gestion des gares



## Application Desktop de réservation KOUSTA Khalid & El Arssi Zakaria

		ID	Nom	Adress
<input type="checkbox"/>	Éditer	Copier	Supprimer	19 OUJDA Boulevard Abdellah Chefchaouni
<input type="checkbox"/>	Éditer	Copier	Supprimer	20 CASABLANCA AIN SEBAA
<input type="checkbox"/>	Éditer	Copier	Supprimer	22 FIGUIG Hay BERKOUKESS Ksar ZENAGA
<input type="checkbox"/>	Éditer	Copier	Supprimer	23 NADOR Gare Nador-Ville
<input type="checkbox"/>	Éditer	Copier	Supprimer	24 TAZA bd Bir Anzarane TAZA
<input type="checkbox"/>	Éditer	Copier	Supprimer	25 FES bd des Almohades .Quartier: fes
<input type="checkbox"/>	Éditer	Copier	Supprimer	26 TANGER hay Mghogha
<input type="checkbox"/>	Éditer	Copier	Supprimer	30 BOUARFA Rt bouarfa
<input type="checkbox"/>	Éditer	Copier	Supprimer	31 MEKNES Nouvelle-Ville Hamria
<input type="checkbox"/>	Éditer	Copier	Supprimer	32 KENITRA bd Mohamed Diouri, Maamora.Quartier
<input type="checkbox"/>	Éditer	Copier	Supprimer	33 AEROPORT aéroport Mohammed V
<input type="checkbox"/>	Éditer	Copier	Supprimer	34 SETTAT bd Changuit
<input type="checkbox"/>	Éditer	Copier	Supprimer	35 BENGUERIR Bd. Mohamed V BEN GUERIR
<input type="checkbox"/>	Éditer	Copier	Supprimer	36 SAFI Rue Sidi Oussel
<input type="checkbox"/>	Éditer	Copier	Supprimer	37 OUJDA hay salam

Figure 26 : base de données des gares

### b. Espace Client

La fenêtre du client est responsable de la recherche et réservation.

Pour effectuer une recherche, l'utilisateur du train doit entrer le départ et la destination, la date et la classe dans laquelle l'utilisateur veut aller.

Figure 28 : Fenêtre principale du client

Code	Gare de ...	Date de ...	Heure	Gare de ...	Heure	Prix
B457	OUJDA	Jan 27, 2...	09:45	CASABL...	18:16	330 DH
BM33	OUJDA	Jan 29, 2...	07:55	CASABL...	13:35	360 DH
KU59	OUJDA	Jan 31, 2...	16:00	CASABL...	23:11	400 DH
FS85	OUJDA	Jan 31, 2...	14:00	CASABL...	19:20	360 DH

Figure 27 : : Fenêtre du choix du Train

## Application Desktop de réservation KOSTA Khalid & El Arssi Zakaria

Une fois que le client a choisi le voyage qui lui convient, une fenêtre de paiement s'ouvre pour qu'il puisse saisir les informations relatives à son compte bancaire et confirmer le paiement

**Africa TGV: Paiement par carte bancaire**

**Total a payer**  
360 DH

**Selectionner une carte ci-dessous pour payer votr...**

☒ VISA ☐ MasterCard ☐ PayPal

**Numéro de carte\***

**Code de sécurité\***  **Expiration\*** January 2021

\*les champs son obligatoire

**Confirmation de paiement**

Figure 29 : Fenêtre de paiement

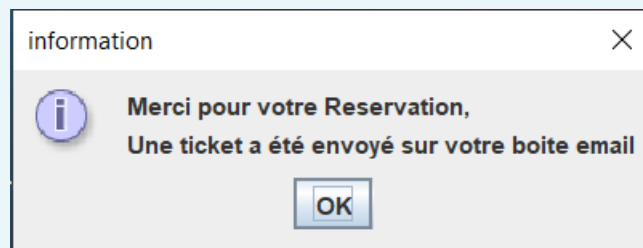


Figure 30 : Message de confirmation

les donnees des client sont totalment sauvgarder et securiser dans notre base de donees.

ID	Nom	Prenom	CIN	Adresse	télé	Email	Départ	Arrivée	Date	Heure	Classe
1	KOUSTA	khalid	FG12200	Hay berkouesse ksar zenaga Figuig	0643756936	kousta90@gmail.com	OUJDA	EL JADIDA	Jan 26, 2021	10:00	17re classe
2	brahimi	youssef	AB5563	oujda hay zaitoun	06542842	ibrajimi.yousef@gmail.com	OUJDA	TANGER	Jan 26, 2021	Matinée	17re classe
3	Amer	Fouad	KT000	sidi bernoussi casablanca	0615151515	kousta90@gmail.com	OUJDA	TAZA	Jan 26, 2021	Matinée	17re classe
4	kousta	ahmed	FG555	ain sebaa	06000111	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	Matinée	17re classe
6	kousta	khalid	fg2200	figuig	0655555555	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	Matinée	27me classe
8	KOUSTA	khalid	fg12200	dfd	552523652	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	09:45	17re classe
9	fafa	hakkou	hj1542	figuig	067585632	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	09:45	17re classe
17	edderkaoui	yassine	fg5585	usa	0645698574	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	09:45	17re classe
18	KOUSTA	WALID	FF	FF	FF	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 26, 2021	09:45	17re classe
20	EL arssi	zakaria	h5452	oujda	0515256135	zakariaeelarsi18@gmail.com	KENITRA	OUJDA	Jan 31, 2021	13:00	17re classe
21	KOUSTA	khalid	FG12200	FIGUIG berkouesse	0643756936	kousta90@gmail.com	OUJDA	CASABLANCA	Jan 31, 2021	14:00	17re classe

Figure 31 : Base de données des clients

## Application Desktop de réservation KOUSTA Khalid & El Arssi Zakaria

Après confirmation, un email sera envoyé au boîte email du client

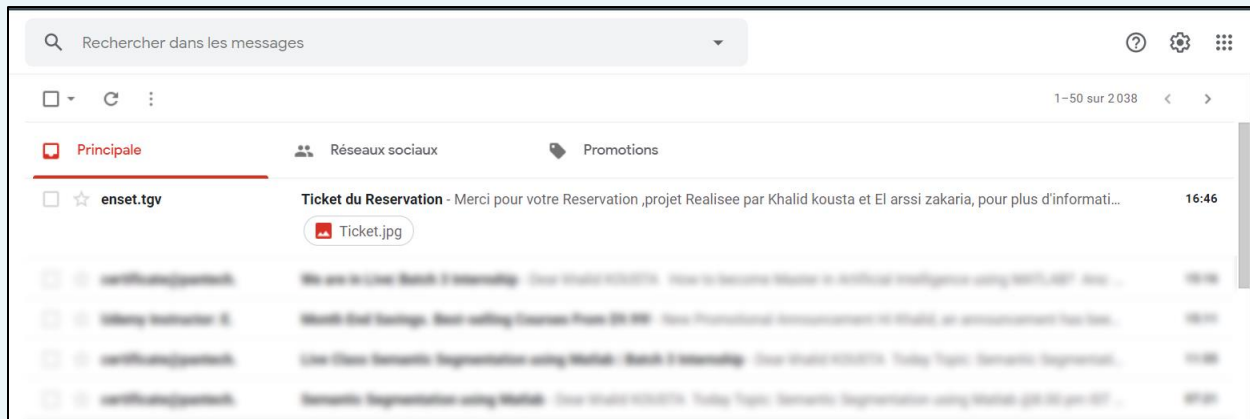


Figure 32 : Réception d'email avec ticket

Un billet est joint à cet e-mail, avec toutes les données nécessaires pour faire un bon voyage dans nos merveilleux trains



Figure 33 : Billet reçu

## CHAPITRE IV : Résultats et discussion

### 1. Résultats des tests

#### 1.1 Tests unitaires

En partant du bas, le premier niveau de test est le "test unitaire". Il s'agit de vérifier que chaque spécifiée dans la "Conception du composant" a été mise en œuvre dans le composant. En théorie, un testeur indépendant devrait le faire, mais en pratique, le développeur fait généralement car ce sont les seules personnes qui comprennent le fonctionnement d'un composant. Le problème avec un composant est qu'il n'assure qu'une petite partie des fonctionnalités d'un système, et il repose sur la coopération avec d'autres parties du système, qui n'ont peut-être pas encore été construites. Pour surmonter ce problème, le développeur construit ou utilise un logiciel spécial pour tromper le composant de croire qu'il fonctionne dans un système pleinement fonctionnel.

Le résumé des tests unitaires est fourni ci-dessous

- Pour l'administration :

UNITÉ	BUT	VÉRIFIÉE
Connexion de l'administrateur	Cette unité détecte l'autorisation d'accès l'administrateur	OUI
Ajouter et modifier les détails des trains	Cette unité permet à l'administrateur de modifier les détails des trains.	OUI
Ajouter et modifier les détails des gares	Cette unité permet à l'administrateur de modifier les détails des gares.	OUI

- Pour le Client :

UNITÉ	BUT	VÉRIFIÉE
Rechercher un train	Cette unité fouille les trains.	OUI
Détails du train	Cette unité montre les trains d'une source particulière vers une destination à une date donnée et un siège spécifique.	OUI
Réserver un billet	L'utilisateur peut sélectionner un train particulier et réserver un billet.	OUI
Remplissez le formulaire de réservation	L'utilisateur doit remplir le formulaire pour réserver son billet.	OUI
Paieement	L'utilisateur remplit le formulaire et paie à l'aide de sa carte de crédit.	OUI
Billet	Cette unité permet au client de recevoir son billet	OUI

## 1.2 Test d'intégration

Au fur et à mesure que les composants sont construits et testés, ils sont ensuite reliés entre eux pour vérifier s'ils travaillent ensemble. C'est un fait que deux composantes qui ont passé tous leurs tests, lorsqu'ils sont connectés entre eux, produisent un nouveau composant plein de défauts. Ces tests peuvent être par des spécialistes, ou par les développeurs.

Les tests d'intégration ne se concentrent pas sur ce que font les composants, mais sur la manière dont ils communiquent entre eux, comme spécifié dans la "Conception du système". La "conception du système" définit les relations entre les composants.

Les tests sont organisés pour vérifier toutes les interfaces, jusqu'à ce que tous les composants aient été construits et se soient interfacés les uns aux autres, produisant ainsi l'ensemble du système.

Ce test a donc été réalisé avec succès. Aucun conflit ou incohérence n'a été détecté.

## 1.3 Résultat

Le système de réservation de billets de train a été conçu et développé avec succès conformément aux spécifications. Il a été largement testé en utilisant une base de données qui contient des données similaires à celles que l'on peut attendre d'une base de données réelle. Il a été constaté que le système fonctionnait de manière satisfaisante, sans aucune erreur et dans toutes les conditions.

## CHAPITRE V : Résumé et conclusions

### 1. Résumé des résultats

Les réalisations suivantes ont été faites au cours du projet :

- Nous avons eu l'occasion d'apprendre une nouvelle technologie JAVA et MySQL
- A appris à travailler dans Eclipse.
- Nous avons appris à gérer un projet de manière efficace et correcte.
- Nous avons appris à faire face à diverses situations adverses tout en gérant et en développant des logiciels.
- Apprendre les différentes phases du développement de logiciels et les processus de génie logiciel impliqués dans le développement de logiciels.

### 2. Principales difficultés rencontrées

- La principale difficulté rencontrée au cours de ce projet était que nous devions apprendre une nouvelle technologie MySQL ainsi qu'un nouveau langage JAVA avec lequel nous n'étions pas du tout familiers auparavant.
- Contraintes de temps.
- Apprendre l'utilisation des threads.