



Université Abdelmalek Essaâdi



Faculté des Sciences Tétouan

# Projet de fin d'étude

**Licence Fondamentale :**  
**Science Mathématique et Informatique**

## Conception et réalisation d'une application Web

### De réservation bus

Réalisé par :

**M. EL HARTI Mohamed Asad**  
**M. AOUTAH Bilal**

Soutenu le 23/07/2020 devant le jury :

Pr.	ATTARIAUS Hicham	Département d'Informatique	Encadrant
Pr.	BENMAATI Mohamed Larbi	Département d'Informatique	Examineur
Pr.	YOUNES Ali	Département d'Informatique	Examineur

## **Dédicaces**

***Au nom d'ALLAH, le miséricordieux et le très miséricordieux***

***A nos très chers parents : qu'ils trouvent ici l'hommage de notre gratitude qui, si grande qu'elle puisse être, ne sera à la hauteur de leurs sacrifices et leurs prières pour nous ;***

***A nos chers frères et sœurs, pour l'amour et le support perpétuel qu'ils nous ont prodigués ;***

***A tous nos amis qui nous sont chers : qu'ils trouvent ici l'expression de nos sentiments les plus dévoués et nos vœux les plus sincères ;***

***Que Dieu, le tout puissant, vous préserve tous et vous procure sagesse et bonheur.***

## **Remerciements**

« Si j'ai vu plus loin, c'est en me tenant sur les épaules des géants qui m'ont précédé » **Isaac  
NEWTON**

La difficulté tient d'abord dans le fait de n'oublier personne. C'est pourquoi, je remercie par avance tous ceux dont le nom n'apparaît pas dans cette page et dont l'intervention au cours de nos Études a favorisé son aboutissement. C'est avec un immense plaisir que je réserve ces lignes en gage de gratitude à tous ceux qui ont contribué de près ou de loin à l'élaboration et la réussite de ce travail.

Mes vifs remerciements vont tout d'abord à la direction de la Faculté des Sciences de Tétouan, et tout le corps professoral de la faculté.

Je tiens à remercier mon encadrant de projet Monsieur ATTARIAUS Hicham pour sa disponibilité, son orientation avisée, ainsi que pour les précieux conseils qu'il n'a cessé de nous prodiguer et qui ont fort contribué à l'accomplissement de ce travail.

## Résumé

---

Dans le but d'obtenir la licence des études fondamentales et comme projet de fin d'étude, notre choix a tombé sur la conception d'une application dédiée à la gestion des réservations des bus.

Depuis l'apparition de la gestion informatique, les agences d voyages et tous type d'organisations optent pour une gestion totale de leurs tâches quotidiennes afin d'économiser le temps, l'énergie et les couts générales, ainsi que l'amélioration de leur niveau de service. Le domaine de la réservation a été l'élément sur lequel nous avons concentré afin de montrer à quel point la gestion informatisée de toutes activités est essentielle à travers la conception d'une application qui vise principalement à gérer et faciliter les tâches quotidiennes des agences grâce aux options qu'elle offre allant de la réservation en ligne jusqu'à la gestion des clients et des destinations.

### **Mots-clés:**

Application, Java EE, Eclipse, XAMP, MCD, MLD, MySQL, XHTML, HTML, CSS, Bootstrap, Spring, Spring IOC, Model MVC, JSF, PrimeFaces, GitHub ...

## Liste des Figures

---

**FIGURE 1** : L'échange des requêtes et des réponses entre les utilisateurs et la base de données

**FIGURE 2** : Organigramme des étapes de la démarche mise en œuvre

**FIGURE 3** : Cycle de vie en cascade présentant les étapes de réalisation d'application

**FIGURE 4** : Organigramme des tâches de la démarche suivie

**FIGURE 5** : L'unification et la normalisation des méthodes

**FIGURE 6** : Diagramme des cas d'utilisation gestion des réservations

**FIGURE 7** : Diagramme des cas d'utilisation gestion d'authentification

**FIGURE 8** : Diagramme des cas d'utilisation gestion des destinations

**FIGURE 9** : Diagramme de séquence de LOGIN

**FIGURE 10** : Diagramme de séquence d'ajouter une destination

**FIGURE 11** : Diagramme d'activité d'affichage des destinations

**FIGURE 12** : Diagramme de classe de l'application

**FIGURE 13** : Modèle conceptuel des données

**FIGURE 14** : Modèle logique des données

**FIGURE 15** : Patrons de conception (MVC)

**FIGURE 16** : Interface d'authentification d'administrateur

**FIGURE 17** : Interface de la liste des clients

**FIGURE 18** : Interface gestion des bus

**FIGURE 19** : Interface gestion des stations

**FIGURE 20** : Interface gestion des destinations

**FIGURE 21** : Interface gestion des réservations

**FIGURE 22** : Interface bienvenue client

**FIGURE 23** : Interface réserver votre billet

## Table des matières

<i>Dédicaces</i> .....	2
<i>Remerciements</i> .....	3
Résumé .....	4
Liste des figures.....	5
Table des matières .....	6
Liste des observations .....	8
Introduction Générale.....	9
<b>Chapitre I : Contexte général du projet</b> .....	10
1. Introduction.....	11
2. Problématique.....	11
3. Solution proposée .....	11
4. Spécification des besoins .....	12
4.1 Les besoins fonctionnels.....	12
4.2 Les besoins non fonctionnels.....	13
4.3 A qui s'adresse l'application.....	13
4.3.1 - Les droits d'accès de l'admin .....	14
4.3.2- Les droits d'accès de client.....	14
4.4. Principe de fonctionnement.....	14
5. Planification opérationnelle.....	15
8. Conclusion.....	15
<b>Chapitre II : Analyse et conception</b> .....	16
1.Introduction.....	17
2.Processus de développement.....	18
3.Cycle de vie.....	19
4.Choix d'UML.....	20
2.1. Diagramme de cas d'utilisation.....	20
2.1. 1- Cas d'utilisation : <<authentication>>.....	22
2.1.2- Cas d'utilisation : <<Gestion des destinations>>.....	23
2.2. Diagramme de séquence.....	24

2.2.1. Diagramme de séquence « Authentification » .....	24
2.2.2. Diagramme de séquence « Ajouter destination » .....	25
2.3. Diagramme d'activité.....	26
2.3.1. Diagramme d'activité « Affichage de la liste des destinations » .....	26
2.4. Diagramme de classes.....	27
3. Conception de la base de données.....	28
3.1. Le modèle conceptuel des données.....	28
3.2. Le modèle logique des données.....	29
3.3. Les tables de la base de données et leur description.....	30
4. Conclusion.....	32
<b>Chapitre III : Réalisation.....</b>	<b>33</b>
1. Introduction.....	34
2. Exigence technique.....	34
3. Technologies de développement.....	35
3.1 Partie Serveur.....	35
3.2 Partie Client.....	38
4. Capture d'écran des interfaces.....	40
4.1 Interface d'authentification de l'admin .....	41
4.2 Interface de la liste des clients .....	42
4.3 Interface gestion des bus .....	43
4.4 Interface gestion des stations .....	44
4.5 Interface gestion des destinations.....	45
4.6 Interface gestion des réservations.....	46
4.7 L'interface bienvenue client.....	47
4.8 Interface réservation.....	48
5. Conclusion.....	49
Conclusion générale.....	50
Références.....	51

## Liste des observations

---

	Abréviation	Désignation
<b>C</b>	CSS	Cascading Style Sheets
<b>D</b>	DB	Database
<b>J</b>	JAVA	Just Another Vague Acronym
	JEE	Java Entreprise Edition
	JSF	JavaServer Faces
<b>H</b>	HTML	HyperText Markup Language
	HTTP	HyperText Transfer Protocol
<b>I</b>	IDE	Integrated Development Environment
<b>M</b>	MVC	Model View Controller
	MYSQL	My Structured Query Language
<b>U</b>	UML	Unified Modeling Language
	URL	Uniform Resource Locator
<b>X</b>	XML	eXtensible Markup Language



## Introduction Générale

---

Personne ne peut nier que l'informatique est une révolution fondamentale et innovante qui a touché considérablement la vie humaine durant les dernières années. En effet, loin d'être un phénomène effervescent, ou une tendance passagère, l'informatique vient d'être exploitée dans tous les aspects de la vie. Aucun domaine n'y est resté à l'abri, ce qui facilite les tâches aussi bien pour l'entreprise que pour le personnel. En effet, les systèmes d'informations ont répondu à un besoin vif pour n'importe quel type d'organisation, c'est la gestion d'information qui est parmi les enjeux les plus primordiaux pour les entreprises et touche pratiquement toutes les activités quotidiennes de ces dernières.

Dans la vision de faciliter et d'améliorer le système de réservations des bus, nous avons proposé de lancer un projet qui consiste à analyser, concevoir et réaliser une application web pour le compte de département d'informatique de la faculté des sciences de Tétouan. L'objectif de ce projet est d'appliquer nos savoirs et nos connaissances concernant les applications web surtout en Java EE. Pour atteindre ces objectifs, nous allons donc essayer de développer cette application en utilisant le langage de modélisation UML, le langage Java EE pour l'implémentation de l'interface graphique en particulier JSF, Spring IOC, MySQL comme système de gestion de bases de données relationnelles.

L'application offre aux clients la possibilité de consulter les voyages qui existent et de réserver en ligne leurs billets.

## CHAPITRE I

# Contexte général du projet

## **1. Introduction**

Ce chapitre constitue une ouverture sur le projet dans le but d'informer sur son environnement organisationnel et contextuel. Il a pour objectif de présenter l'organisme d'accueil, ainsi de présenter le sujet de projet et le cahier de charge relatif à projet.

## **2. Problématique**

Le voyage en bus reste une alternative assez populaire à l'avion, l'automobile et le train. Pour ceux qui veulent profiter du paysage ou qui angoissent à l'idée de prendre l'avion, le bus est un bon choix. Avec le bus on prend également pleinement conscience des distances parcourues, on ne ressent pas le choc de se retrouver à l'autre bout du monde dans un environnement complètement différent en seulement quelques heures.

L'autre avantage du bus est qu'il n'y a pas de restriction pour les bagages, vous pourrez donc transporter des grosses charges sans vous ruiner. (Pas de supplément comme en avion)

Grace à l'internet, les habitudes des consommateurs ont bien changé au fil des années. Tout le monde veut tout, tout de suite, et le premiers à avoir notés ce changement, c'est bien les agences de voyages.

Depuis des années, la réservation des billets représente un inconvénient majeur du voyage en bus, puisqu'il faut obligatoire consulter une agence de voyage pour se renseigner sur les disponibilités.

## **3. Solution proposée**

Après l'étude que nous avons effectuée, la solution proposée était de créer une application qui facilite la réservation puisqu'elle permet de chercher, réserver votre billet depuis votre PC ou Téléphone.

## **4. Spécification des besoins**

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif, mais elle-même repose sur une bonne spécification des besoins qui n'est autre que la question que doit se poser tout ingénieur au début de son travail " Qu'est ce qu'on veut que nous fassions ? ". Dans cette section nous allons présenter notre réponse à cette question. Donc, cette phase consiste à mieux comprendre le contexte du système, il s'agit de déterminer les fonctionnalités et les utilisateurs du notre application.

### **4.1. Les besoins fonctionnels**

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces besoins proviennent généralement des utilisateurs du système. Cette application devra permettre :

- **Gestion des clients** : Cette partie intégrera la gestion des clients, cette à dire le faite d'ajouter et d'archiver les informations des clients, et aussi faire des manipulations sur ces données : ajout, modification et suppression
- **Gestion des Bus** : Il est bien conseiller de conserver la liste des bus dans un tableau pour donner au client la possibilité de choisir le bus qui le préfère, les manipulations possibles : ajout, modification et suppression
- **Gestion des Stations** : Comme on a enregistré la liste des bus il est plus significatif d'enregistrer aussi les stations, on peut aussi modifier et supprimer une station
- **Gestion des destinations** : Ajouter une destination entre deux stations en précisant le prix de voyage, le temps et le bus, les manipulations possibles : ajout, modification et suppression.

- **Gestion des réservations** : C'est la partie la plus importante car dans cette partie le client peut réserver ses billets, l'admin peut ajouter, modifier et supprimer une réservation.

## 4.2. Les besoins non fonctionnels

Ce sont des besoins qui permettraient d'améliorer la qualité des services d'application, notre application doit répondre aux critères suivants :

- **La sécurité** : l'application doit respecter la confidentialité des données.
- **La convivialité** : l'application doit être facile à utiliser. Il doit présenter enchainement logique entre les interfaces, c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.
- **Les erreurs** : L'application doit signaler les erreurs par des messages d'erreurs compréhensibles.
- **La performance** : C'est-à-dire une application doit être avant tout performante, à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.
- **La stabilité** : L'application doit être stable dans son environnement d'exécution et ne doit pas contenir des bogues bloquants.

## 4.3. A qui s'adresse l'application ?

- **L'administrateur** : A le droit de faire tous les manipulations sur l'application (mise à jour...)
- **Client** : A le droit de consulter les voyages, et réserver un billet

#### 4.3. 1 - Les droits d'accès de l'admin :

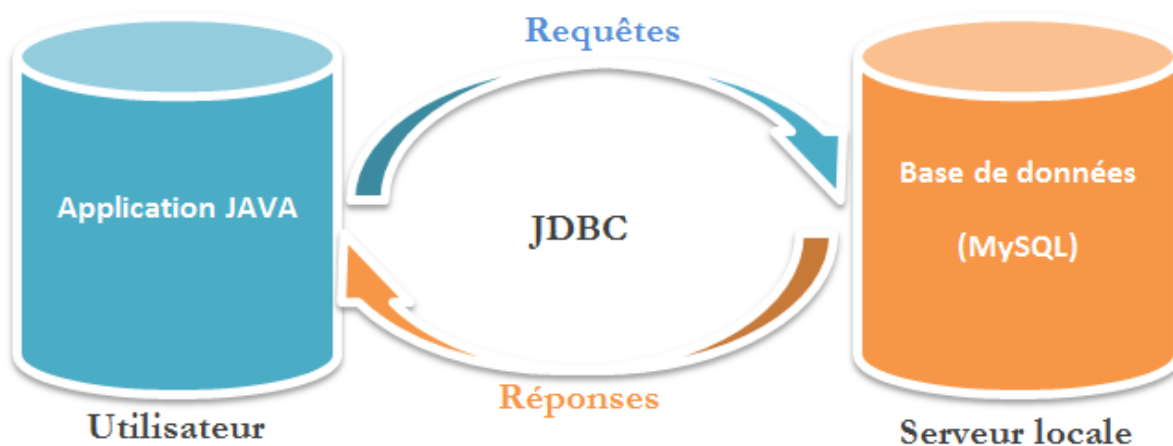
- ✓ Ajouter, modifier, supprimer un client
- ✓ Ajouter, modifier, supprimer un bus
- ✓ Ajouter, modifier, supprimer une station
- ✓ Ajouter, modifier, supprimer une destination
- ✓ Modifier, supprimer une réservation

#### 4.3.2 – Les droits d'accès de client

- ✓ Consulter les voyages disponible
- ✓ Réserver un ou plusieurs billets

### 4.4. Principe de fonctionnement

Le principe de cette application est l'échange des requêtes et de réponses entre : (l'administrateur ou le client) et la base de données. Cet échange permet au serveur de satisfaire les besoins de l'utilisateur.



**FIGURE 1 :** L'échange des requêtes et des réponses entre les utilisateurs et la base de données

## 5. Planification opérationnelle

Taches	Durée	Date de début	Date de fin	Prédécesseur
A. La réception du projet	4	01/04/2020	04/04/2020	/
B. Etude de cahier des charges	2	06/04/2020	07/04/2020	A
C. Etude fonctionnelle	9	09/04/2020	17/04/2020	B
D. Etude technique	9	20/04/2020	28/04/2020	C
E. Conception	15	02/05/2020	20/05/2020	D
F. Autoformation	9	21/05/2020	29/05/2020	E
G. Réalisation de l'application web	30	30/05/2020	28/06/2020	F
H. Evaluation	3	29/06/2020	02/07/2020	H
I. Rédaction du rapport	75	09/04/2020	03/07/2020	B

**FIGURE 4 : ORGANIGRAMME DES TACHES DE LA DEMARCHE SUIVIE**

## 6. Conclusion

Après qu'on a décrit le contexte, les objectifs du projet, les besoins fonctionnels ainsi que les besoins non fonctionnels de notre projet, on a essayé de fixer les échéances pour livrer le projet dans ses délais.

Dans la prochaine partie j'entame une analyse de besoins et des spécifications avant de passer à la conception de notre solution.

## CHAPITRE II

# Analyse et conception

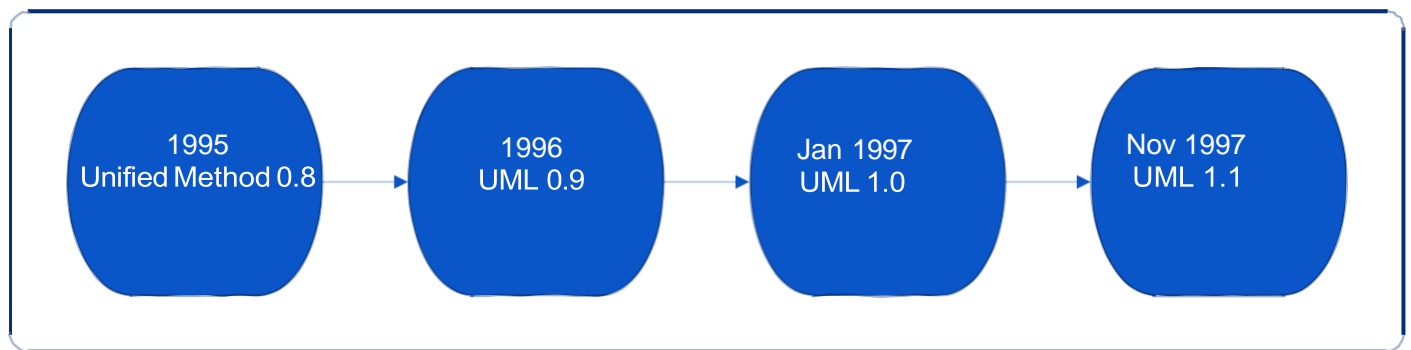


## 1. Introduction

Face à cette diversité méthodologique, nous avons choisi l'approche orientée objet pour la conception de l'application. Ce choix revient essentiellement aux diverses opportunités qu'offre cette approche, d'une part, et aux problèmes, surtout dans la phase de codage, que peut causer l'approche fonctionnelle, d'autre part.

Les méthodes orientées objets les plus connus sont OMT (Object Modeling Technology), Booch et OOSE (Object Oriented Software Engineering).

UML (Unified Modeling Language : langage de modélisation objet unifié) est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE.



**FIGURE 5 : L'UNIFICATION ET LA NORMALISATION DES METHODES**

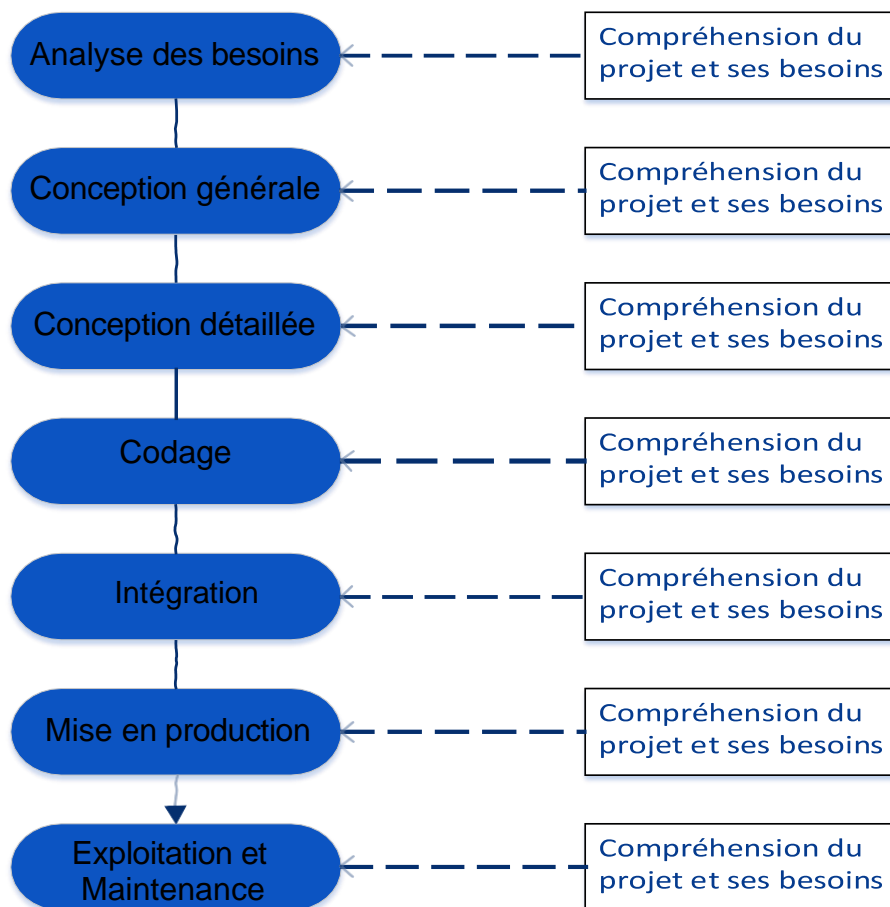
## 2. Processus de développement

Une bonne conduite du projet nécessite plusieurs critères.

La garantie de cette dernière s'effectue, à travers une bonne spécification des besoins, une minutie des enjeux et une estimation des gains potentiels.

Ce projet de fin d'étude comporte plusieurs parties, permettant l'atteinte de l'objectif estimé .

Nous allons étaler cela en utilisant l'organigramme suivant :



**FIGURE 2 :** Organigramme des étapes de la démarche mise en œuvre

### 3. Le cycle de vie

La réalisation de notre application peut être vue comme une suite ordonnée d'étapes à atteindre, dont chacune se termine par une étape de vérification et de validation destinée à éliminer le plus possible d'anomalies. C'est pour cela que nous avons choisi le Modèle en Cascade comme cycle de vie.

Le modèle Cascade comporte sept phases : spécifications, conception générale, conception détaillée, codage, intégration, mise en produit et maintenance. Chacune de ces sept phases doit produire un ou plusieurs livrables définis à l'avance et a une date de délai fixée. On ne peut précéder d'une étape à l'autre que lorsque les livrables de l'étape en cours sont jugés satisfaisants. Si tout se passe bien on précéder à la phase suivante, sinon on remonte à la phase précédente.

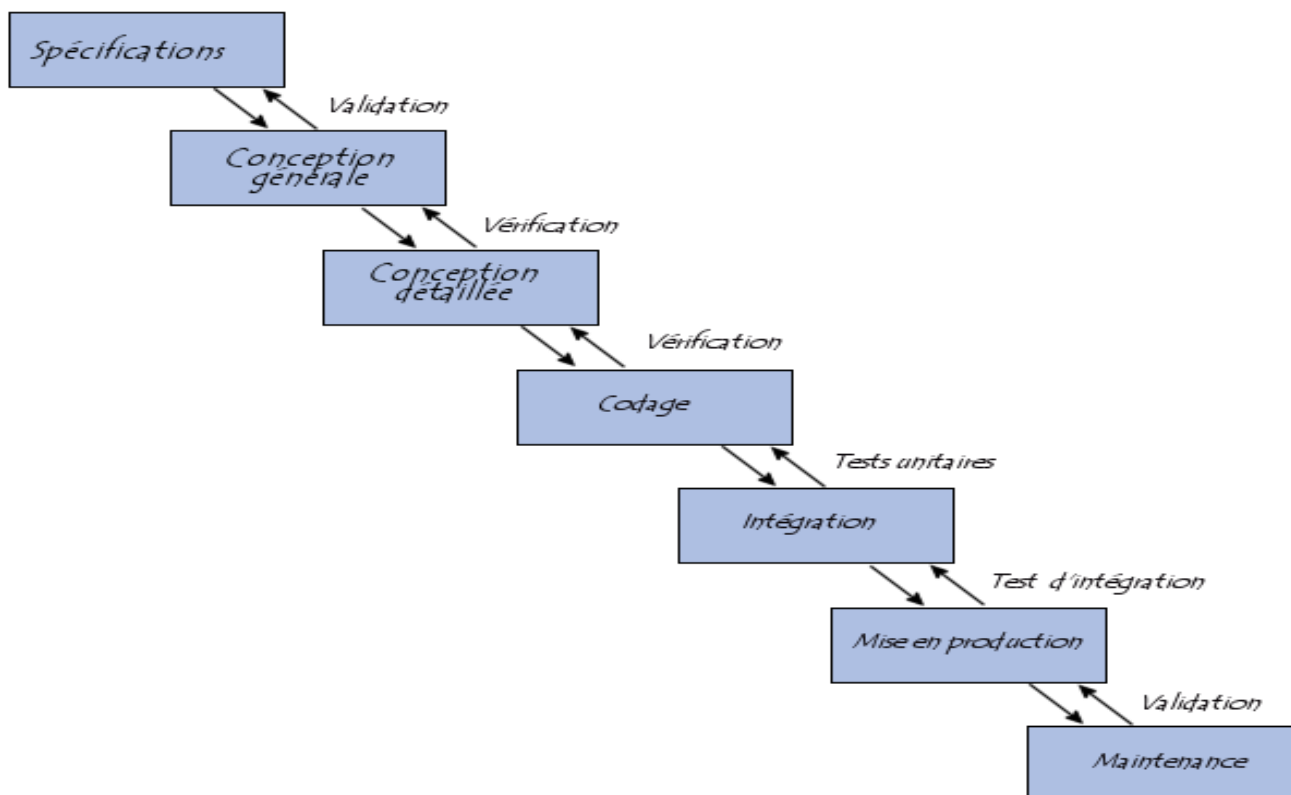


FIGURE 3 : CYCLE DE VIE EN CASCADE PRESENTANT LES ETAPES DE REALISATION D'APPLICATION

## 4. Choix d'UML

Notre choix s'est porté sur la méthode UML vue :

- Sa richesse : il couvre toutes les phases d'un cycle de développement.
- Son indépendance : par rapport aux langages de programmation, aux domaines d'application et aux processus. C'est un langage universel.
- L'aspect formel de sa notation : limite les ambiguïtés et les incompréhensions.
- Sa notation graphique : permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- Le langage UML considère le développement de plusieurs diagrammes. Pour les besoins de notre système, nous allons nous limiter dans notre travail à quatre diagrammes, seulement le diagramme de cas d'utilisation, le diagramme d'activité, le diagramme de classe et le diagramme de séquence.

### 2.1. Diagramme de cas d'utilisation

Les Diagrammes de cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés. Ils se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système, mais il permet de faire une description d'un ensemble de séquences d'actions qu'un système effectue pour produire un résultat observable à un acteur.

Alors pour faire simple on a choisi de modéliser la manière de fonctionnement de quelques tâches de notre application.

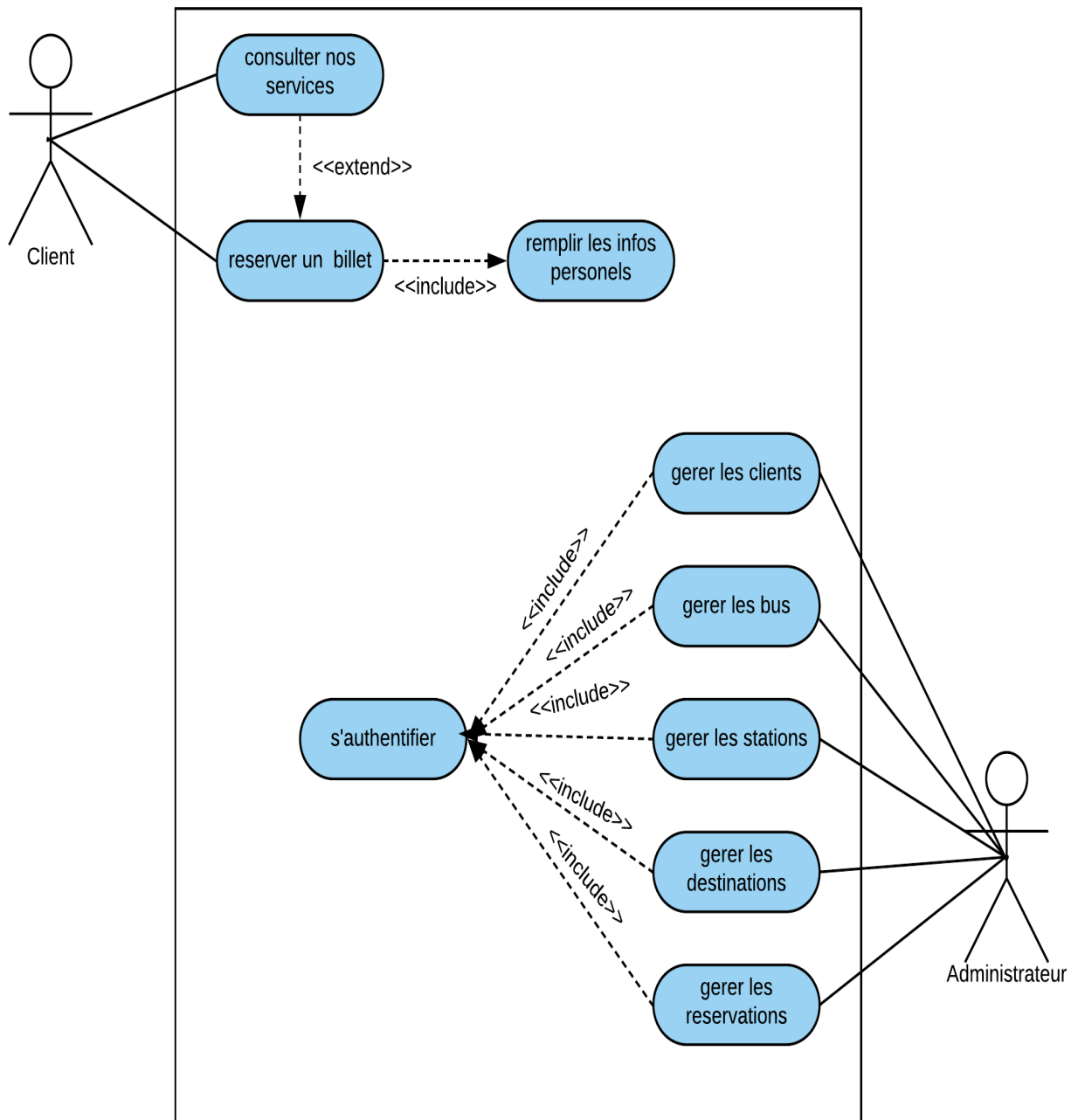
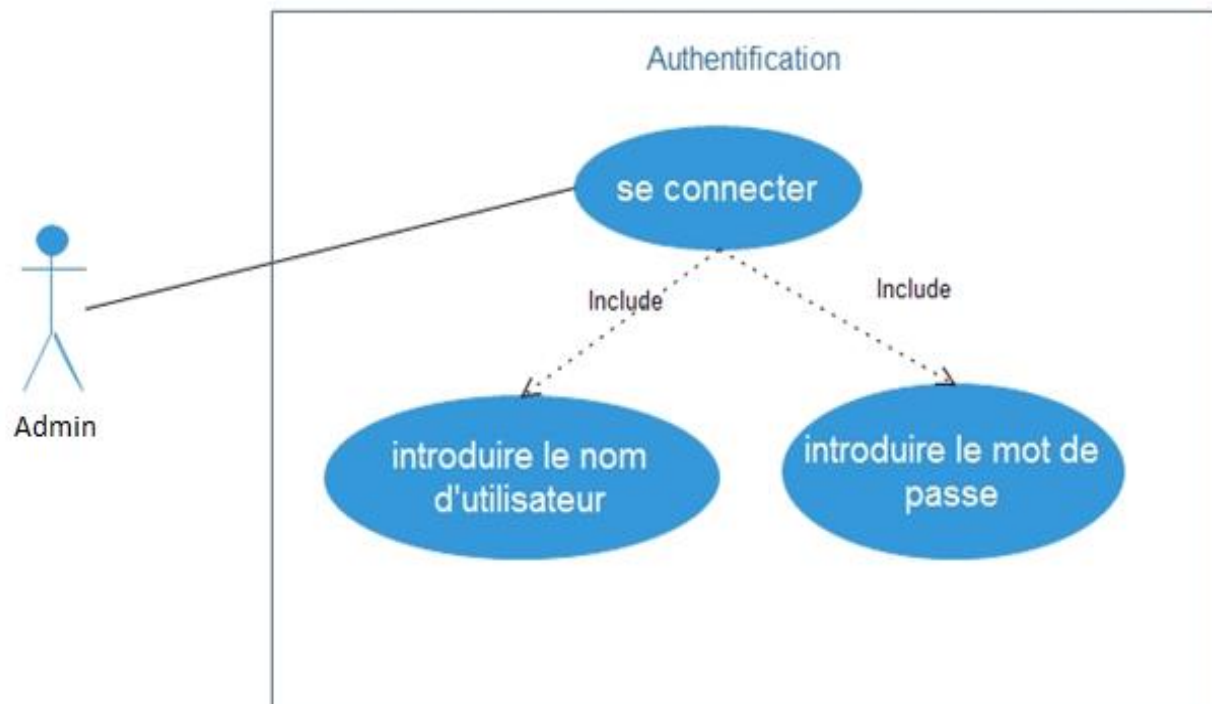


FIGURE 6 : DIAGRAMME DES CAS D'UTILISATIONS GESTION DE RESERVATIONS

### 2.1. 1- Cas d'utilisation : <<authentication>>

Pour accéder au système, l'admin introduit son nom et son mot de passe. Si les informations sont correctes une page d'accueil, sinon le système renvoie un message d'erreur



**FIGURE 7 :** DIAGRAMME DE CAS D'UTILISATION GESTION D'AUTHENTIFICATION

### 2.1.2- Cas d'utilisation : <<Gestion des destinations>>

L'admin peut consulter tous les destinations, ajouter, modifier et supprimer une destination, le client a le droit de chercher une destination

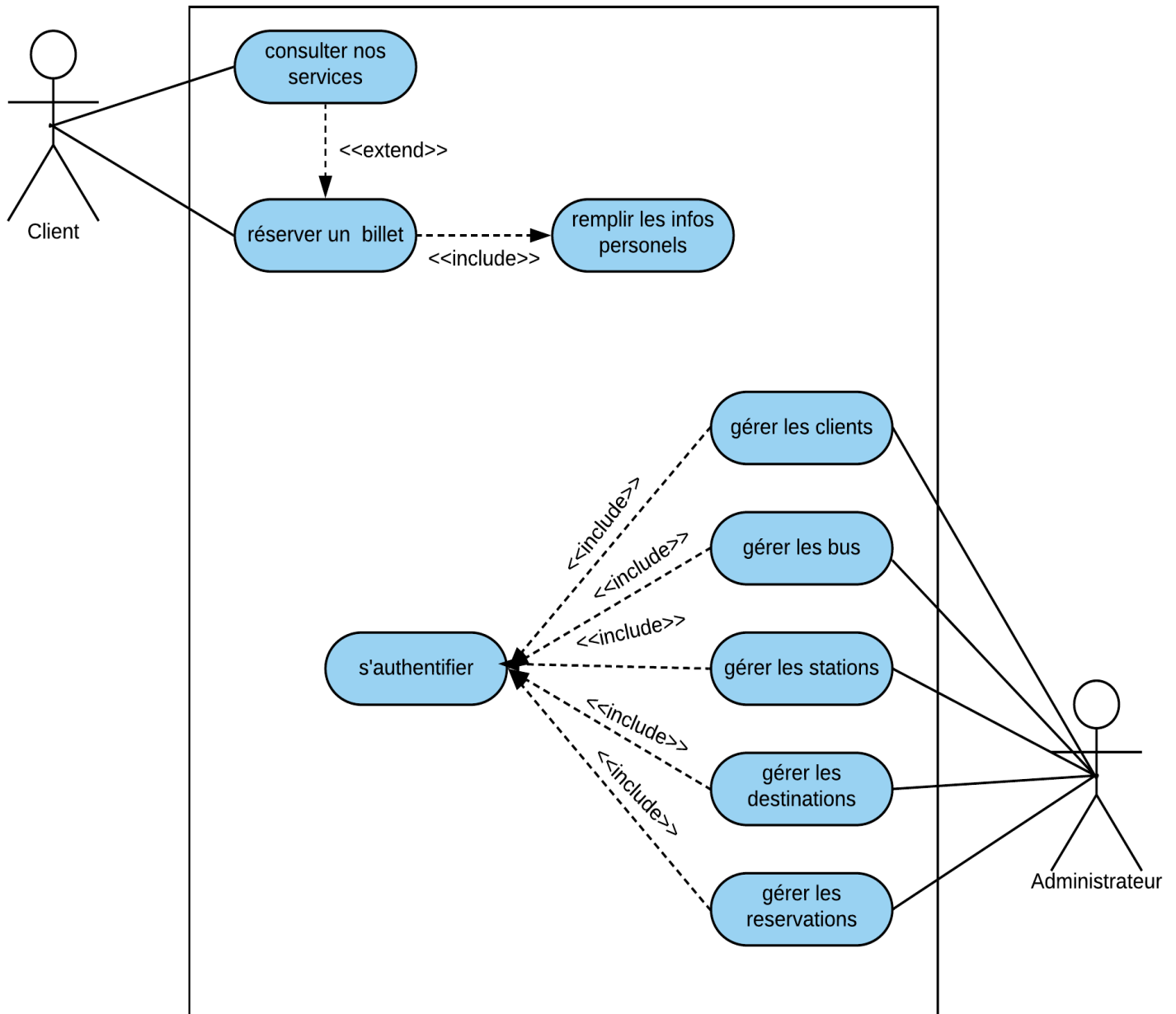
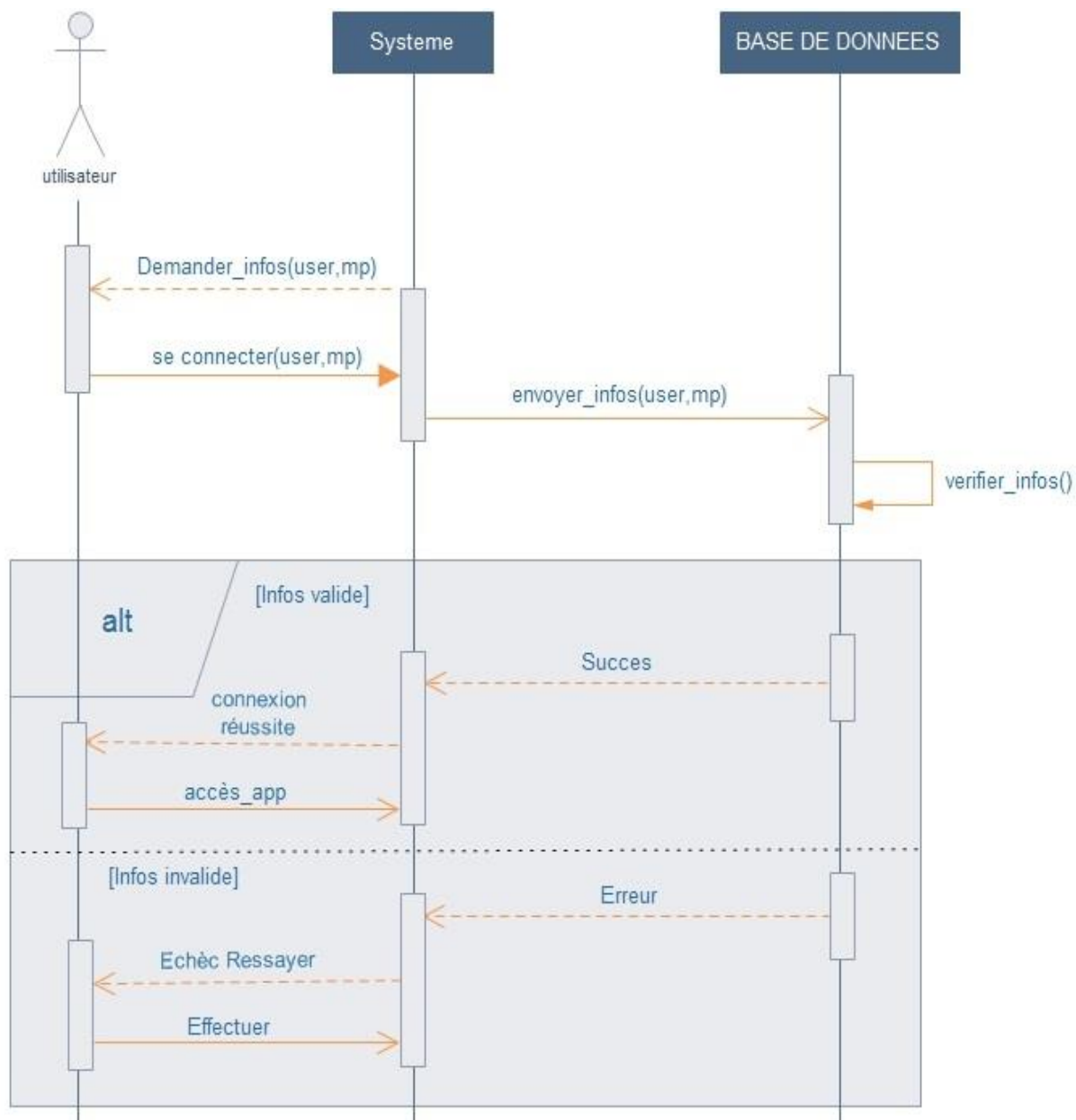


FIGURE 8 : DIAGRAMME DE CAS D'UTILISATION GESTION DES DESTINATIONS

## 2.2. Diagramme de séquence

### 2.2.1. Diagramme de séquence « Authentification »

Suite à une demande d'accès à l'application, l'admin va passer à une interface d'authentification dans laquelle il va introduire son propre nom d'utilisateur et son mot de passe et selon le résultat de la vérification de la validité de ces coordonnées il aura l'accès ou non à l'interface qui le concerne.





### 2.2.2. Diagramme de séquence « Ajouter destination »

Si l'admin veut injecter une nouvelle destination dans la base de données, il accède à son interface, il clique sur le bouton « liste des destinations », le formulaire affiche sur l'écran. Le SGBD doit vérifier l'existence de cette destination et suivant le résultat de la requête, la destination sera ajoutée ou un dialogue d'information va s'affiché dans le but d'informer l'admin que cette destination déjà existe.

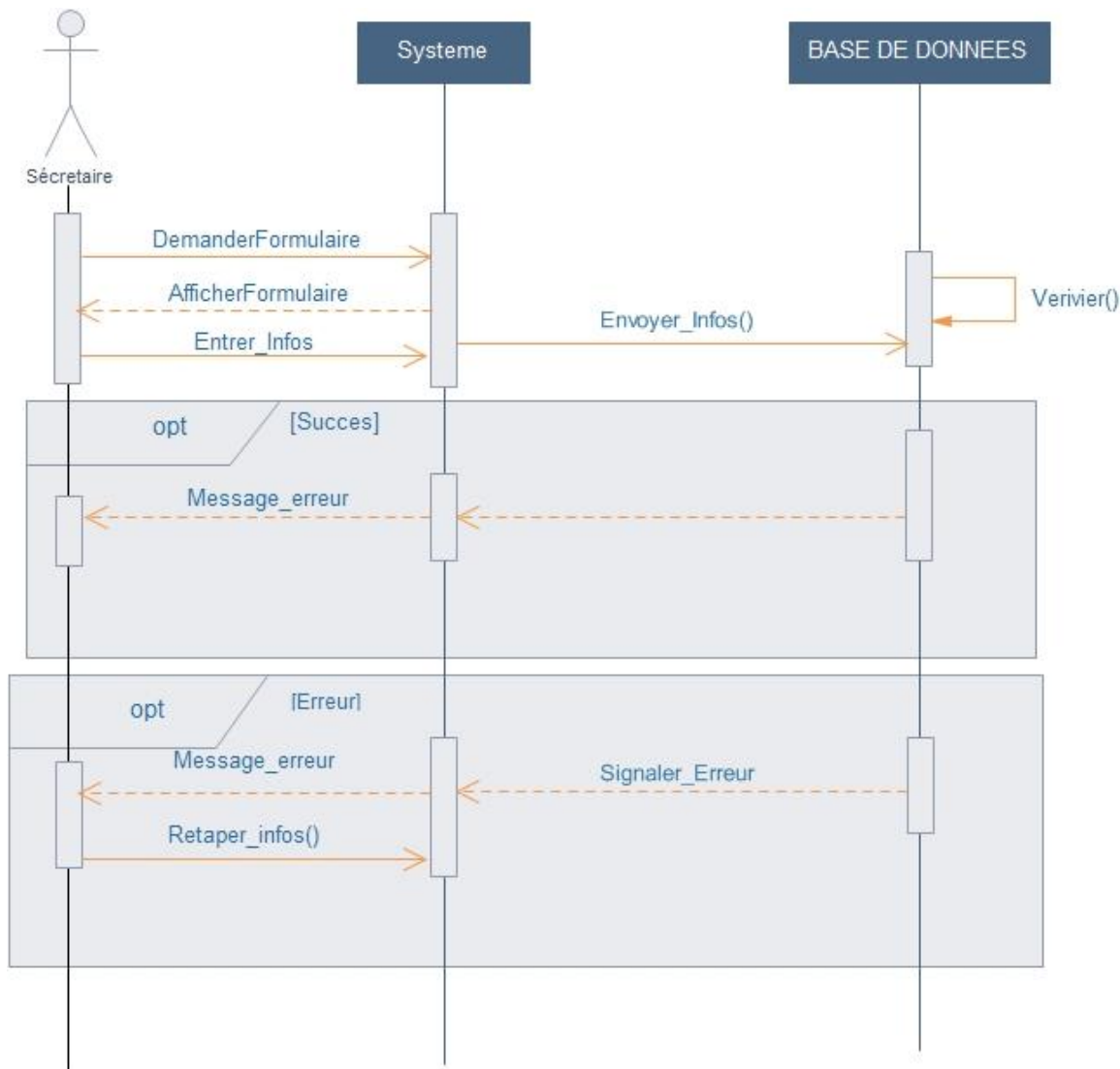


FIGURE 10 : Diagramme de séquence d'ajouter une destination

## 2.3. Diagramme d'activité

### 2.3.1. Diagramme d'activité « Affichage de la liste des destinations »

Après le démarrage de l'application l'admin doit s'authentifier si les informations sont correctes, il peut afficher la liste des destinations en cliquant sur « liste des destinations »

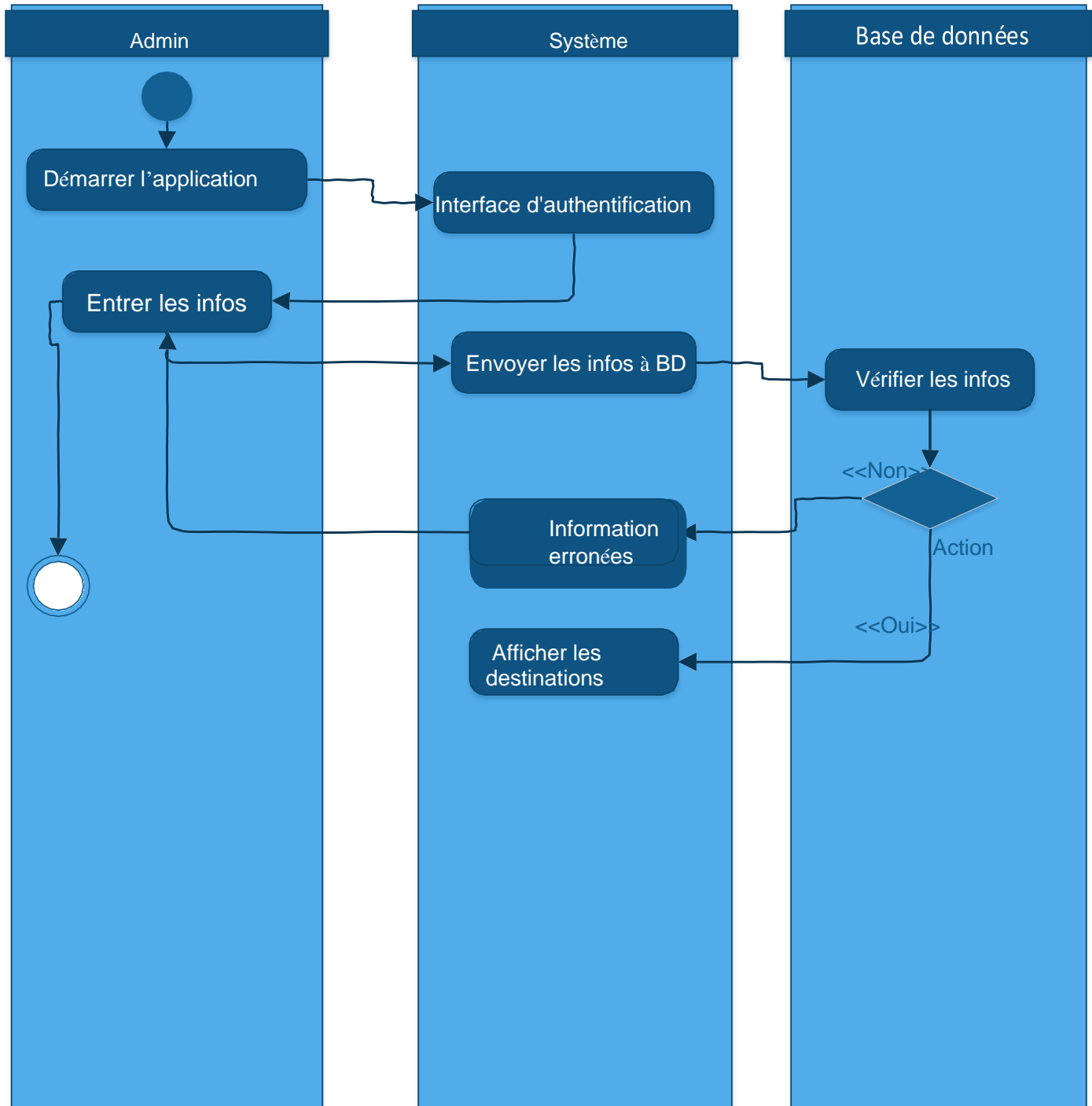


FIGURE 11 : Diagramme d'activité d'affichage des destinations

## 2.4. Diagramme de classes

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classe représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage du développement.

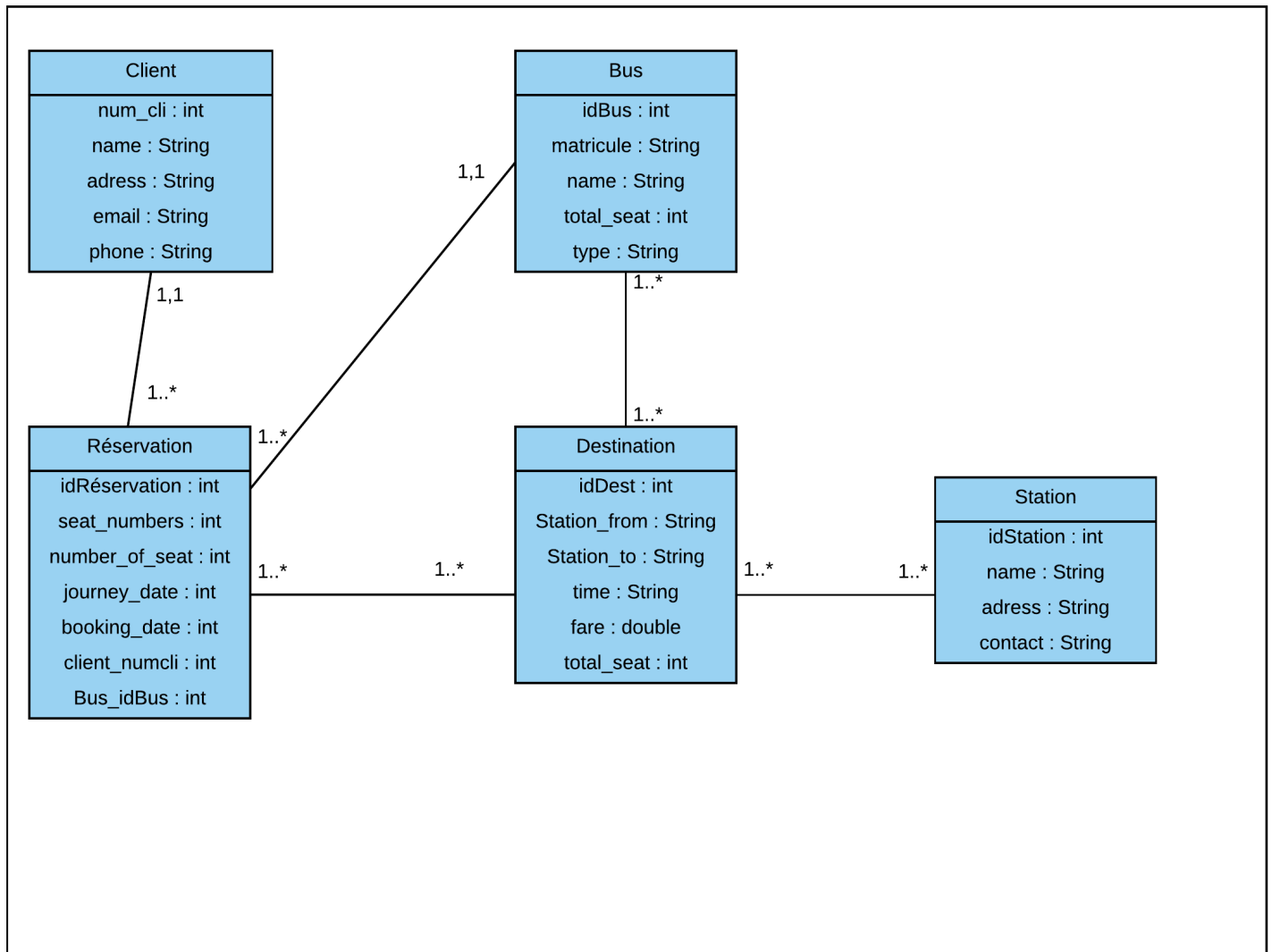


FIGURE 12 : Diagramme de classe de l'application

## 3. Conception de la base de données

### 3.1. Le modèle conceptuel des données

Un modèle conceptuel de données (MCD) est la représentation la plus abstraite des données d'un système d'information.

Les données sont représentées sous forme d'entités et d'associations entre entité.

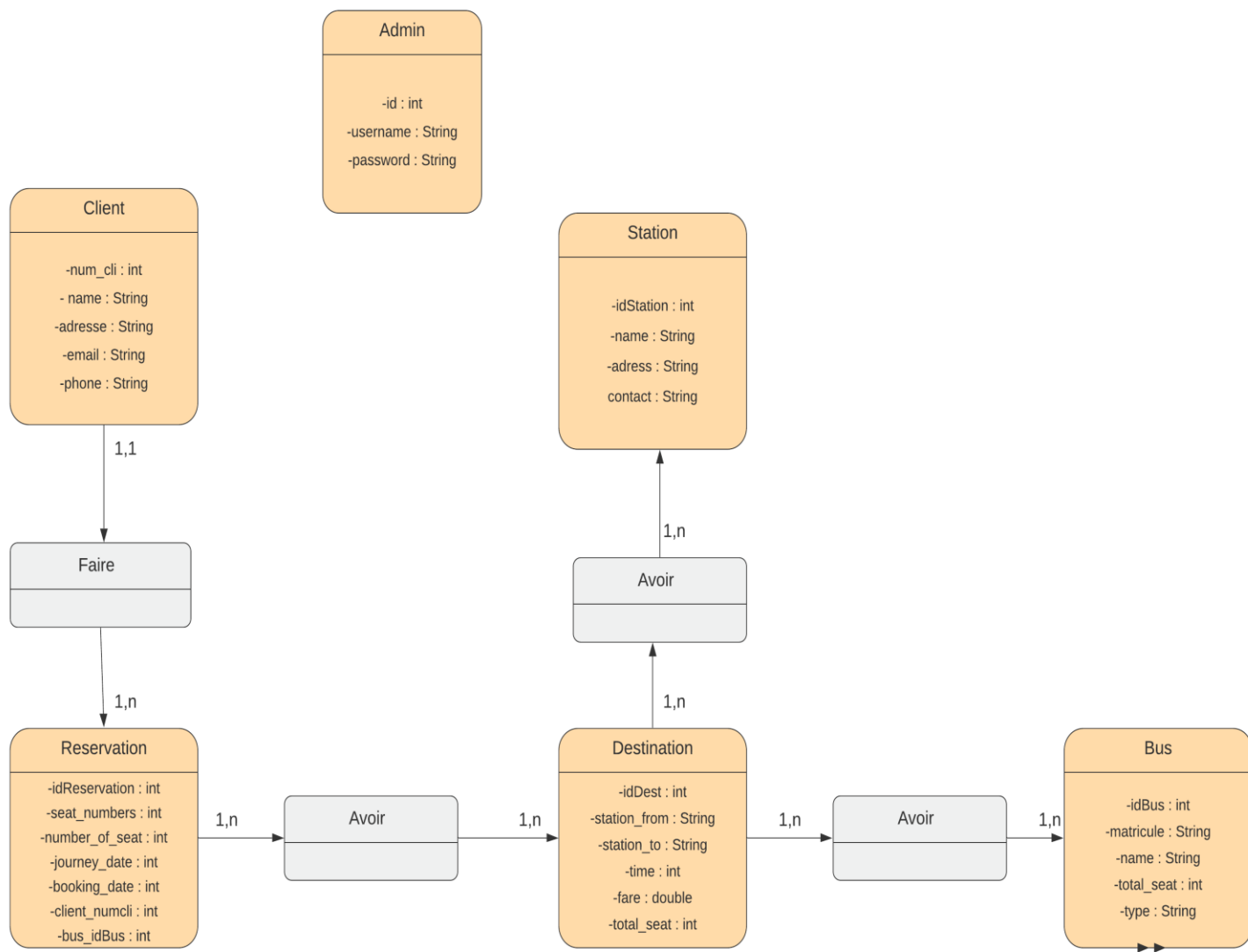


FIGURE 13 : Modèle conceptuel des données

## 3.2. Le modèle logique des données

Le MLD est un modèle relationnel qui reprend les associations, et transforme la forme du MCD pour montrer les différentes clés étrangères.

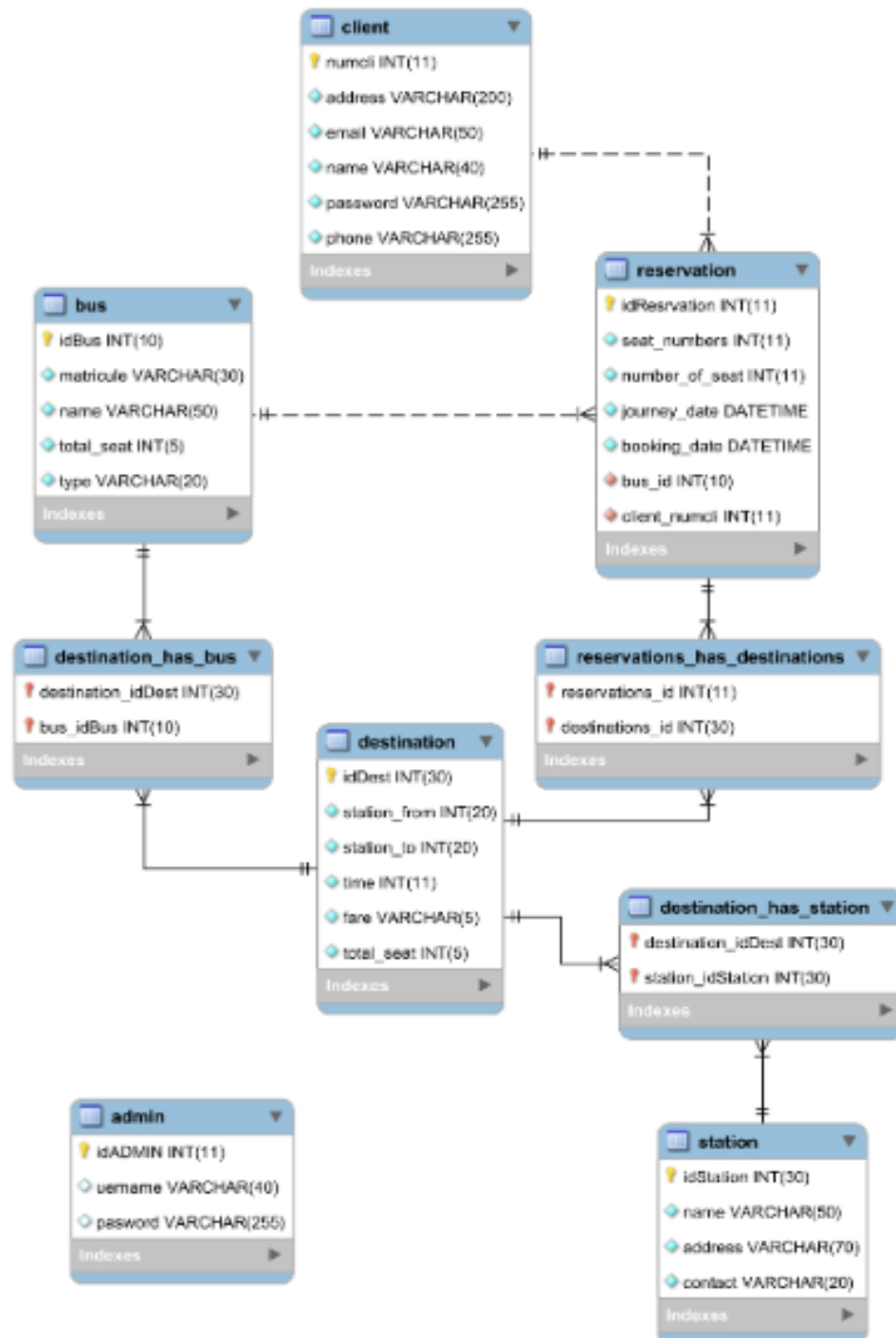



FIGURE 14 : Modèle logique des données

### 3.3. Les tables de la base de données et leur description


En se basant sur le MLD, nous avons créé la base de données suivante :

- Nom de la base de données : ticket system
- Nombre des tables : neuf tables
- Nom de l'utilisateur : (non communiqué pour raison de sécurité).
- Mot de passe : (non communiqué pour raison de sécurité).


#### 1) Table client:

Nom	Type
numcli 	int(11)
address	varchar(200)
email	varchar(50)
name	varchar(40)
phone	varchar(255)


#### 2) Table bus :

Nom	Type
idBus 	int(10)
matricule	varchar(30)
name	varchar(50)
total_seat	int(5)
type	varchar(20)



#### 3) Table station :

Nom	Type
idStation 	int(30)
name	varchar(50)
address	varchar(70)
contact	varchar(20)





#### 4) Table destination

Nom	Type
idDest 	int(11)
station_from	varchar(10)
station_to	varchar(10)
time	int(11)
fare	double
total_seat	int(5)





#### 5) Table réservation

Nom	Type
idReservation 	int(11)
seat_numbers	int(11)
number_of_seat	int(11)
journey_date	int(11)
booking_date	int(11)
client_numcli 	int(11)
bus_idBus 	int(10)





#### 6) Table destination\_has\_bus

Nom	Type
bus_idBus  	int(10)
destination_idDest  	int(11)


#### 7) Table destination\_has\_station

Nom	Type
station_idStation  	int(30)
destination_idDest  	int(11)

8) Table reservation\_has\_destination:

Nom	Type
reservation_idReservation  	int(11)
destination_idDest  	int(11)

9) Table admin:

Nom	Type
idADMIN 	int(11)
username	varchar(40)
password	varchar(255)

## 5. Conclusion

Ce chapitre a été dédié à la conception du système. Dans ce chapitre, nous avons présenté la conception réalisée pour les différents modules composant ce projet à savoir les diagrammes de cas d'utilisation, diagrammes de séquences et le diagramme de classes. Réalisation, Exigences techniques et technologies de développement sera l'objet du chapitre suivant.



## CHAPITRE III

# Réalisation

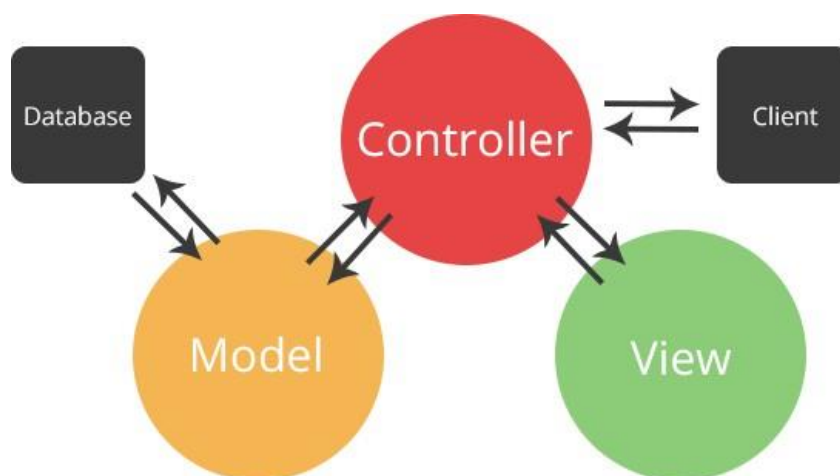
## 1. Introduction

Ce chapitre présente le résultat du travail effectué durant ce projet de fin d'études. Nous allons présenter, aussi, les technologies et l'environnement matériels utilisées, tel que le langage de programmation, le système de gestion des bases de données (SGBD). Nous clôturons ce chapitre par quelques captures d'écran de notre application.

## 2. Exigence technique

### Modèle MVC

- MVC est un patron de conception particulièrement adapté pour réaliser une application web.
- MVC est l'acronyme de Modèle-Vue-Contrôleur : le Modèle contient les données de l'application, la Vue (ou les Vues) contient le code pour afficher les pages de l'application, et le Contrôleur gère les interactions de l'utilisateur en faisant le lien avec le Modèle et la Vue.
- MVC permet une séparation claire des intentions et optimise la création d'une application web ou d'un site, sa maintenance et ses tests automatisés.



**FIGURE 15 : PATRONS DE CONCEPTION (MVC)**

### 3. Technologies de développement

#### 3.1 Partie Serveur

##### a. L'architecture JEE



La plateforme Java Entreprise (Java EE) est un ensemble de spécifications coordonnées et pratiques qui permettent des solutions pour le développement, le déploiement, et de la gestion des applications multi-tiers centralisées sur un serveur. Construite sur la plateforme de Java 2 édition standard (Java SE), la plateforme Java EE ajoute les possibilités nécessaires pour fournir une plateforme complète, stable, sécurisée, et rapide de Java au niveau entreprise.

##### b. Java



Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisé pour le développement d'applications d'entreprises et mobiles.

Il n'y a pas de compilation spécifique pour chaque plateforme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du bytecode.

### c. Apache Tomcat



Le conteneur de servlet JEE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur http.

### d. MySQL



MySQL est un Système de Gestion de base de données relationnelle qui a vu le jour en 1995 et très utilisée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

### e. XAMPP



XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

## f. Frameworks

- [Spring](#)



Spring est un socle pour le développement d'applications, principalement d'entreprises mais pas obligatoirement. Il fournit de nombreuses fonctionnalités parfois redondantes ou qui peuvent être configurées ou utilisées de plusieurs manières : ceci laisse le choix au développeur d'utiliser la solution qui lui convient le mieux et/ou qui répond aux besoins.

Spring est ainsi un des frameworks les plus répandus dans le monde Java : sa popularité a grandi au profit de la complexité de Java EE notamment pour ses versions antérieures à la version 5 mais aussi grâce à la qualité et la richesse des fonctionnalités qu'il propose : son cœur reposant sur un conteneur de type IoC assure la gestion du cycle de vie des beans et l'injection des dépendances l'utilisation de l'AOP des projets pour faciliter l'intégration avec de nombreux projets open source ou API de Java EE

- [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.

- [GitHub](#)



GitHub est une plateforme de développement inspirée par votre manière de travailler. De l'open source à l'entreprise, vous pouvez héberger et revoir du code, gérer des projets et construire des logiciels aux côtés de millions d'autres développeurs.

## 3.2 Partie Client

### a. HTML5



HTML5 (HyperText Markup Language 5) est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en parallèle sur le même document afin de maintenir une version unique de la technologie.

### b. CSS3



Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

## c. Primefaces



Prime Faces est une bibliothèque open source de composants JSF.

Il est basé côté serveur sur l'API standard de JSF 2.

Côté client les scripts de Prime Faces sont basés sur la librairie la plus populaire de JavaScript jQuery.

Prime Faces vise à garder le traitement propre, rapide et léger.

Prime Faces est maintenu par « Prime Teknoloji », une entreprise de logiciels Turque de

développement spécialisée dans Agile et Java EE conseil.

## d. Frameworks

- JSF



Java Server Faces (JSF) est une technologie dont le but est de proposer un framework qui facilite et standardise le développement d'applications web avec Java. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks.

- [Bootstrap](#)



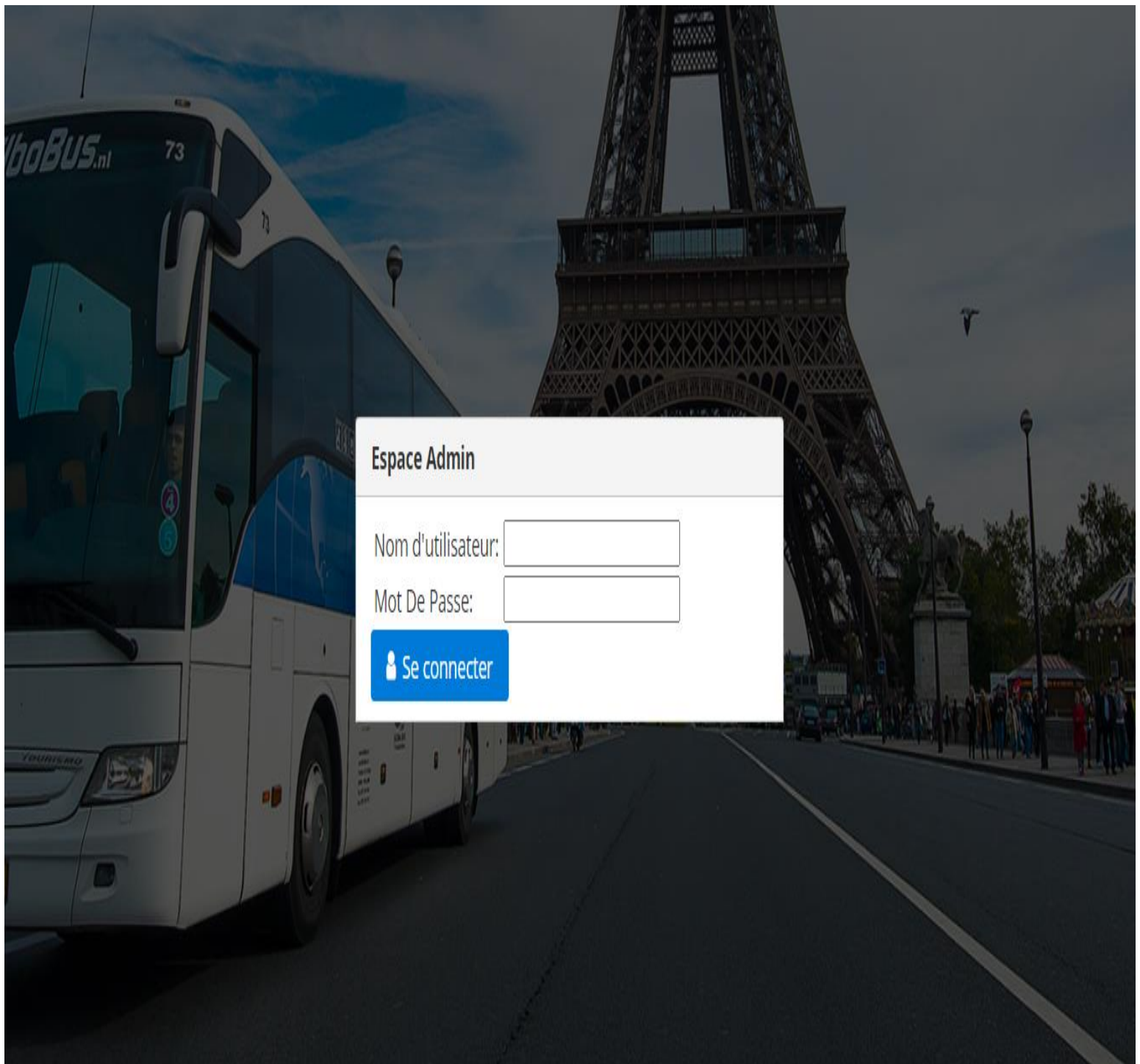
Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

## **4. Capture d'écran des interfaces**

Je vais présenter dans cette partie quelques interfaces utilisateurs de notre application



## 4. 1 Interface d'authentification de l'admin



**FIGURE 16 :** Interface d'authentification d'administrateur

## 4.2 Interface de la liste des clients :

Sur cette interface l'admin peut consulter la liste des clients, ajouter, modifier et supprimer un client.

Se déconnecter

Menu Generale

> Dashboard

▼ Client

Liste des clients

▼ Bus

Gestion des Bus

> Station

> Destination

> Reservation

Liste des Clients

« < 1 > »

Numero	Nom Complet	Adresse	Email	Telephone
18	bilal	tet	bilal@hbf	0678494478
19	asaad	tetouan	asaad-gubh@hbu	06456389403
20	ismael	tanger	ismael@gmail.com	0687754789
21	simo	ytfgyu	simo@reyg.com	0678494478
22	edc	edgvuc	redfdre	0988765

« < 1 > »

FIGURE 17 : Interface de la lise des clients

## 4.3 Interface gestion des bus :

Sur cette interface l'admin peut consulter la liste des bus, ajouter, modifier et supprimer un bus.

se deconnecter

Menu Generale

Dashboard

Statistiques

Client

Bus

Gestion des Bus

Station

Destination

Gestion des destination

Reservation

Nom de Bus : \*

Matricule : \*

Total Seat : \*

0

Bus Type : \*

Enregistrer

Liste des Bus

« < 1 > »

Numero	Matricule	places	Nom	Type
14	4352ab	45	vitalis	class A
15	ter456	65	atlas	class B
16	ct567	47	CTM	premium

« < 1 > »

RiAs @ 2020

FIGURE 18 : Interface gestion des bus

## 4.4 Interface gestion des stations :

Sur cette interface l'admin peut consulter la liste des clients, ajouter, modifier et supprimer un client

se deconnecter

Menu Generale

Dashboard

Client

Bus

Station

Destination

Reservation

Ajouter Station

Nom de Station : \*

Adresse : \*

Contact : \*

Enregistrer

Liste des stations

« < 1 > »

Numero	Nom	Adresse	Contact
9	tetouan	mhanech	068948249
10	tanger	medina	087783657
11	rabat	qamra	069848438
12	casablanca	wlad zian	0743647842

« < 1 > »

BIAs @ 2020

FIGURE 19 : Interface gestions des stations

## 4.5 Interface gestion des destinations

Sur cette interface l'admin peut consulter la liste des destinations, ajouter, modifier et supprimer une destinations.

se deconnecter

Menu Generale

Dashboard

Statistiques

Client

Bus

Gestion des Bus

Station

Gestion des Station

Destination

Gestion des destination

Reservation

Ajoute Destination

Station de départ :

tetouan

Station d'arrivée:

tanger

Bus:

vitalis

Prix: \*

0.0

Heure : \*

0

Enregistrer

Liste des Destinations

« < 1 > »

Départ	Arrivée	Heure	Prix	Total de places
tetouan	tanger	4	20.0	65
tetouan	rahat	7	70.0	17

RiAs © 2020

FIGURE 20 : Interface gestion des destinations

## 4.6 Interface gestion des réservations

Sur cette interface l'admin peut consulter la liste des réservations.

Se déconnecter

Menu Generale

> Dashboard

▼ Client

Liste des clients

▼ Bus

Gestion des Bus

▼ Station

Gestion des Station

▼ Destination

Gestion des destination

▼ Reservation

Gestion des Reservation

Statistiques

Liste des Reservations

« < 1 > »

ID Reservation	Heure	Adresse de client	Matricule de bus	Heure de Reservation	Numero de place	total de place
12	4	tet	ter456	10	1	65
13	7	tetouan	ct567	10	1	47
14	4	tanger	ter456	10	2	65
15	4	ytfgyu	ter456	10	3	65
16	4	edgvuc	ter456	10	4	65

« < 1 > »

FIGURE 21 : Interface gestion des réservations



## 4.7 L'interface bienvenue client :

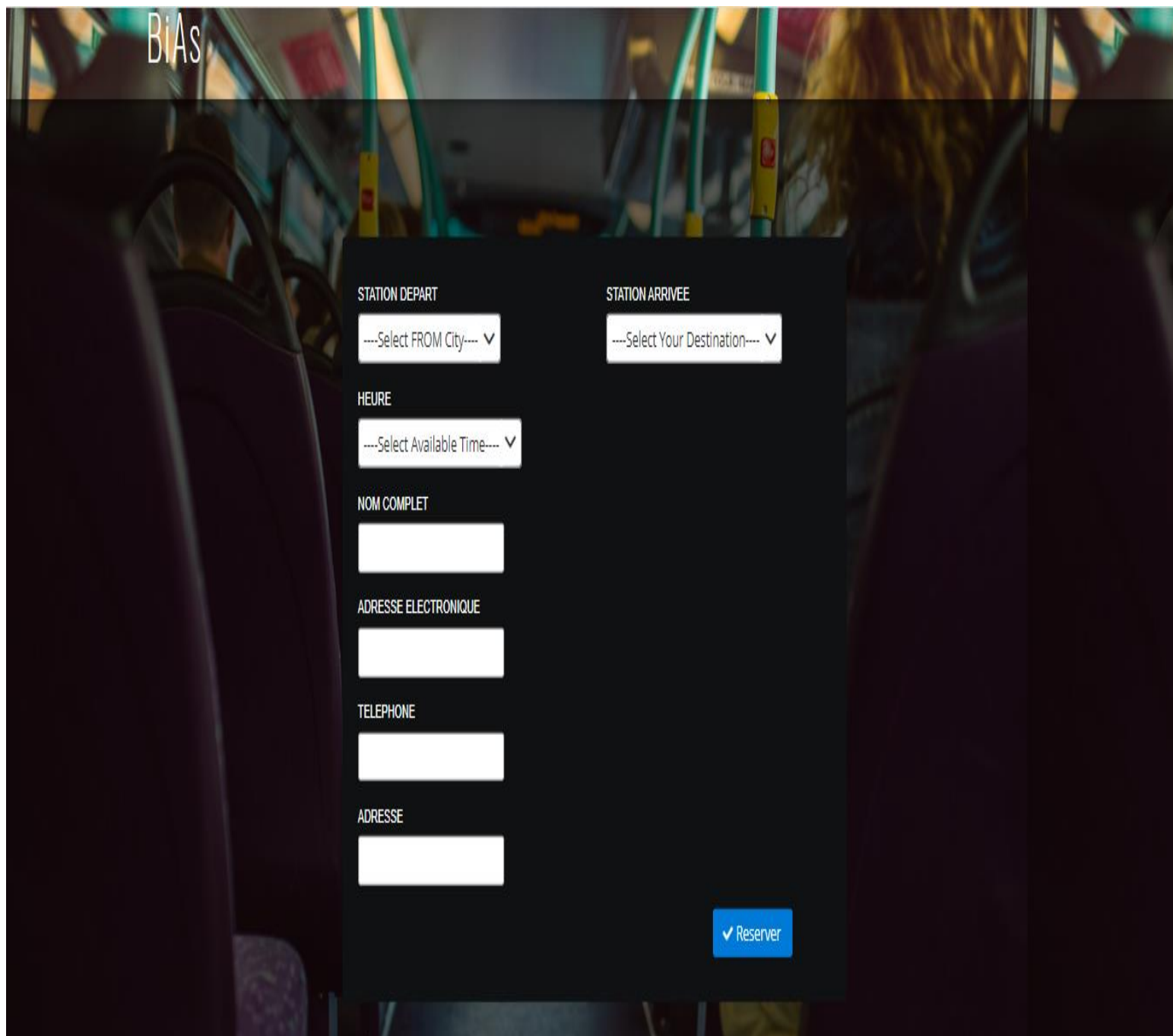
Sur cette interface le client peut accéder à la page de réservation et consulter notre service.



FIGURE 22 : Interface bienvenue client

## 4.8 Interface réservation :

Sur cette page le client peut chercher une destination et réserver un billet.



The image shows a dark-themed reservation form titled "BiAs" in the top left corner. The form is overlaid on a blurred background of a train interior. The form contains the following fields and labels:

- STATION DEPART**: A dropdown menu with the text "----Select FROM City----" and a downward arrow.
- STATION ARRIVEE**: A dropdown menu with the text "----Select Your Destination----" and a downward arrow.
- HEURE**: A dropdown menu with the text "----Select Available Time----" and a downward arrow.
- NOM COMPLET**: A text input field.
- ADRESSE ELECTRONIQUE**: A text input field.
- TELEPHONE**: A text input field.
- ADRESSE**: A text input field.
- Reserver**: A blue button with a white checkmark icon and the text "Reserver".

**FIGURE 23 :** Interface réserver votre billet



## **5. Conclusion**

Ce chapitre était consacré aux exigences techniques et technologies de développement de l'application web. Dans un premier temps, nous avons décrit les exigences techniques, ensuite nous avons présenté les Frameworks, les technologies de développement utilisées et quelques interfaces de l'application.

## Conclusion générale

---

Je tiens à témoigner de l'importance de ce projet, qui m'a donné l'occasion de mettre en exergue des concepts et des connaissances techniques et d'analyse acquis durant mes études à la faculté des sciences. Je me suis particulièrement investi dans ce projet, à cause de son caractère utile. Mon projet consiste de réaliser une application qui gère la réservation des bus.

Le fait de développer une application, basée sur des technologies web récentes est quelque chose de vraiment motivant car je me retrouve dans la situation, avec un cahier des charges, une "Dead-line", et un planning. Il m'a fallu d'améliorer mes techniques de documentation, prendre le temps pour les décisions importantes et bien sûr tout faire pour favoriser mes compétences.

Bien que la réalité démontre que la théorie n'est qu'une encre sur papier, et que la pratique est partiellement différente des connaissances acquises. La théorie reste l'aboutissement de toute pratique, sa satisfaction et sa limite, car j'ai eu l'occasion d'apprendre plusieurs choses nouvelles et aussi la façon d'affronter les problèmes inattendus.

Durant cette expérience enrichissante, j'ai eu l'occasion de rencontrer des gens de responsabilité qui n'ont pas hésité à me faire profiter de leur expérience dans le monde du travail et ainsi à m'encourager à avoir de l'initiative, à renforcer mon esprit de compétitivité et de performances.

Enfin, On pourrait dire que Dans la durée de mes études m'a fait découvrir des domaines d'activités, j'ai permis d'appliquer les concepts acquis durant ma formation sur la vie professionnelle. Et je suis également conscient qu'il me reste encore beaucoup à apprendre, même si je suis désormais plus confiant vis à vis de mes connaissances.

## RÉFÉRENCES

<http://docs.oracle.com/javaee/7/api/>

Dernière consultation le 20/06/2020

<https://javabrain.io/>

Dernière consultation le 15/06/2020

<https://spring.io/docs/>

Dernière consultation le 16/06/2020

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

Dernière consultation le 30/06/2020

<http://hibernate.org/>

Dernière consultation le 04/06/2020

<https://colorlib.com/>

Dernière consultation le 25/06/2020

<https://stackoverflow.com/>

Dernière consultation le 25/06/2020