Text

Description automatically generated with low confidenceA picture containing application

Description automatically generated

Module Technologie Objet/IoT

Filière d’ingénieur:

**I**ngénierie **L**ogicielle et **I**ntégration des **S**ystèmes **I**nformatiques

Projet :

**Système de Réservation de ticket du**

**busway**

v

Encadré par :

*Année universitaire : 2023-2024*

* Pr O. EL BEGGAR Encadrant Pédagogique

Réalisé par :

Naima DAOUI

Khaoula ABASSI

Table des matières

[Introduction : 3](#_Toc162184259)

[I. Définition des acteurs et les besoins : 4](#_Toc162184260)

[1. Diagramme d’acteur : 4](#_Toc162184261)

[2. Diagramme de contexte statique : 5](#_Toc162184262)

[3. Recensement des exigences 5](#_Toc162184263)

[4. Regroupement des exigences par intention des acteurs : 6](#_Toc162184264)

[II. Analyse : 7](#_Toc162184265)

[1. Diagramme de cas d’utilisation 7](#_Toc162184266)

[2. Description textuelle de haut niveau : 7](#_Toc162184267)

[3. Description textuelle de bas niveau : 8](#_Toc162184268)

[III. Conception : 9](#_Toc162184269)

[1. Les diagrammes de communication : 9](#_Toc162184270)

[2. Diagramme de classe de conception : 11](#_Toc162184271)

[IV. Réalisation : 12](#_Toc162184272)

[1. Stockage des données : 12](#_Toc162184273)

[2. Interfaces : 15](#_Toc162184274)

[2-1 L’interface de réservation 16](#_Toc162184275)

[2-2 L’interface de confirmation : 16](#_Toc162184277)

[2-3 L’interface d’exportation du ticket : 17](#_Toc162184278)

[2-4 Le ticket exporté sous forme pdf : 18](#_Toc162184280)

[2-5 L’interface d’authentification de l’administrateur : 19](#_Toc162184281)

[2-6 Interface d’accueil d’administrateur : 19](#_Toc162184282)

[2-7 L’interface des voyages disponibles : 20](#_Toc162184283)

[2-8 L’interface de saisie des informations d'un bus 21](#_Toc162184284)

[2-9 Interface pour l'ajout des arrêts au bus créé 22](#_Toc162184285)

[2-10 L’interface des bus disponibles : 24](#_Toc162184286)

# Introduction :

Dans le monde dynamique et interconnecté d'aujourd'hui, la réservation de billets de transport en temps réel est devenue une nécessité incontournable pour des millions de personnes à travers le monde. La possibilité de réserver rapidement et efficacement un ticket de transport est cruciale. Dans le contexte spécifique de Casablanca, l'une des métropoles les plus animées du Maroc, la demande pour un système de réservation de Tickets de bus en temps réel est particulièrement palpable. C'est dans ce cadre que notre projet de réservation des tickets en temps réel trouve sa pertinence et son importance.

Notre projet vise à mettre en place un système de réservation de tickets de busway à Casablanca qui offre une expérience fluide et efficace pour les utilisateurs. Nous entendons réaliser cela en développant un système de réservation de tickets spécialement conçu pour gérer les demandes de réservation en temps réel. Concrètement, notre système permettra aux utilisateurs de rechercher, sélectionner et réserver des voyages de busway de manière rapide et intuitive, tout en assurant une optimisation maximale des performances et une utilisation efficiente des ressources.

En outre, notre système prendra en compte les spécificités du réseau de busway à Casablanca, en tenant compte des contraintes de disponibilité des sièges, des horaires de départ et d'arrivée, ainsi que des différentes options de trajet disponibles. Cela garantira que les utilisateurs bénéficient d'une expérience de réservation transparente et personnalisée, répondant parfaitement à leurs besoins individuels.

# Définition des acteurs et les besoins :

## Diagramme d’acteur :

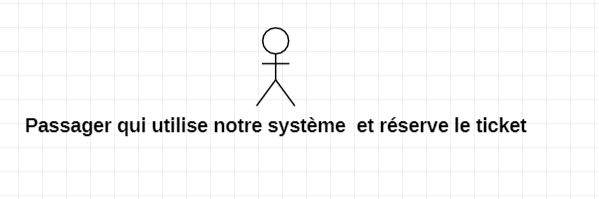


Figure 1 : Diagramme d’acteurs

## Diagramme de contexte statique :

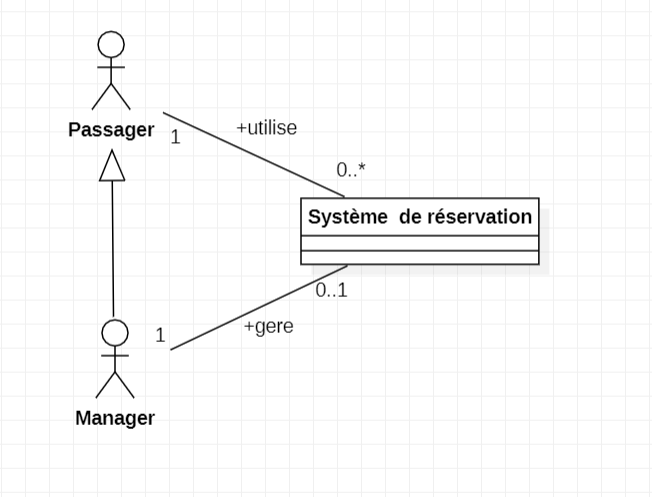


Figure 2 : Diagramme de contexte statique

## Recensement des exigences

|  |  |
| --- | --- |
| Référence | Fonctionnalités |
| R1 | Sélectionner un voyage parmi les voyages disponibles |
| R2 | Choisir la station d’arrivée |
| R3 | Afficher les voyages disponibles |
| R4 | Afficher les bus disponibles |
| R5 | Exporter ticket |
| R6 | Vérifier disponibilité d’un voyage à une station |
| R7 | Sélectionner un voyage |
| R8 | Chercher les voyages disponibles pour une station de départ et une station d’arrivée |
| R9 | Afficher les stations les plus proches au passager |
| R10 | Saisir le nom complet du passager |
| R11 | Générer QR code correspondant à la réservation |
| R12 | Générer ticket |
| R13 | Sélectionner la station de départ |
| R14 | Installer les informations sur le bus |
| R15 | Exporter Ticket |

## Regroupement des exigences par intention des acteurs :

|  |  |  |
| --- | --- | --- |
| Références Croisées | Intention (UC) | Acteur principal |
| R1, R2, R6, R7, R8, R9, R10,  R13 | Réserver un ticket | Passager |
| R11, R12, R15 | Exporter Ticket | Passager |
| R3 | Afficher les voyages disponibles | Passager |
| R4 | Afficher les bus disponibles | Passager |
| R14 | Installer les informations sur le bus | Manager |

# Analyse :

## Diagramme de cas d’utilisation

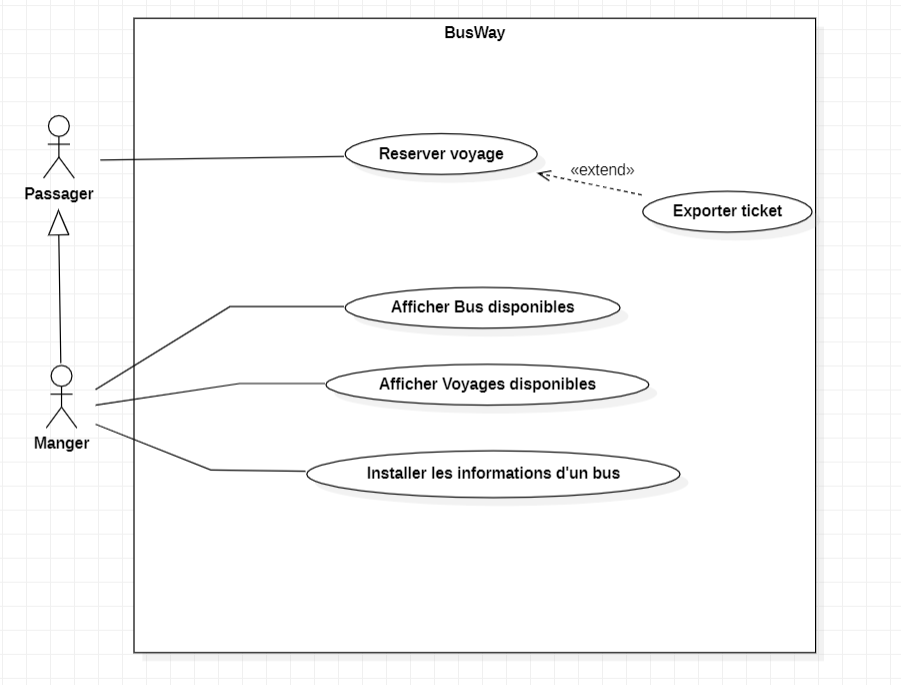


Figure 3 : Diagramme de cas d’utilisation

## Description textuelle de haut niveau :

* Nom de cas d’utilisation : **Réserver Ticket.**
* Objectif : Un passager réserve un voyage
* Acteur Principal : Passager.
* Responsable : DAOUI Naima, ABASSI Khaoula
* Version 2.0
* Préconditions : \* Système fonctionnel
* Postconditions : Une réservation d’un voyage est ajoutée et le nombre de place disponible de ce dernier sera décrémenter.

## Description textuelle de bas niveau :

* Scénario Nominal :

|  |  |
| --- | --- |
| Actions réalisées par l’acteur principal | Actions réalisées par le système |
| 1 – Passager se connecte au système  3-Demande une réservation d’un ticket  6- sélectionner une station départ et station d’arrivée  9- Sélectionner un voyage  11- Saisir nom et prénom  12-Valider réservation  20-Exporter le ticket (dont le QR code est généré) | 2-Dériger le passager vers la page d’accueil  4- chercher et afficher les stations les plus proches au passager  5- afficher les stations d’arrivées possibles.  7-Chercher les voyages disponibles et qui correspondent aux stations choisis par le passager.  8- Afficher les voyages disponibles à cette réservation.  10-Demander nom et prénom du passager  13-Enregistrer la réservation  14-création du ticket  15-Demande d’exportation du ticket  21-Mettre fin à la session |

* Enchainement d’exceptions :
* E1 : L'indisponibilité des voyages correspondant aux critères choisis par le client.

Point de départ : point 7 du scénario nominal.

* E2 : Aucune station de départ n'est située à proximité de la localisation du passager.

Point de départ : point 4 du scénario nominal.

# Conception :

## Les diagrammes de communication :

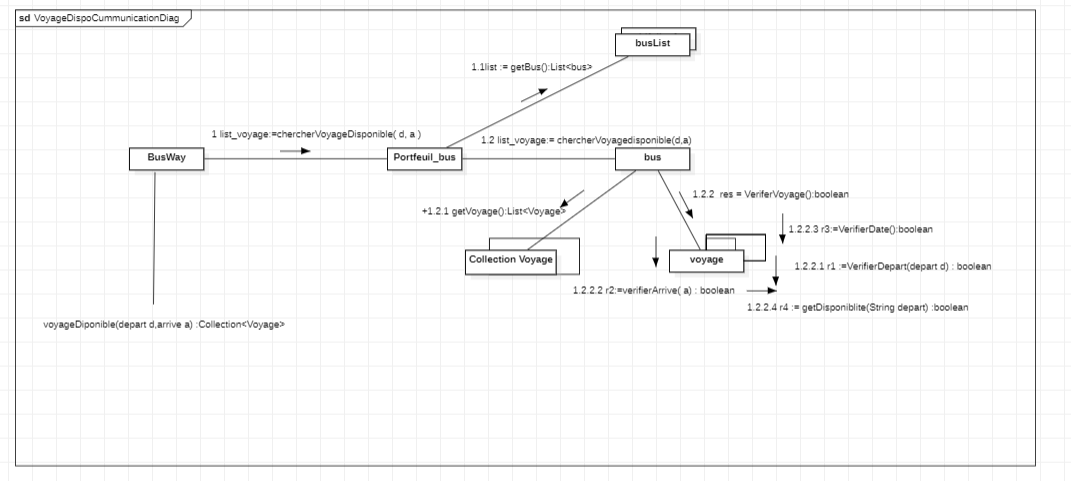
* + - * Contrat d’opération :
* Opération : **Chercher les voyages disponibles**
* Préconditions :
* Un objet BusWay est créé (classe contrôleur).
* Au moins un objet **Bus** est créé.
* Au moins un **objet Conducteur** est créé.
* Au moins deux **stations** sont créées.
* Postconditions :
* Liste voyages disponibles seront retrouvées.

Figure 4 : Diagramme de communication <<chercher voyages disponibles>>

* + - * Contrat d’opération :
* Opération **: Réserver un Ticket**
* Préconditions :
* Un objet **Voyage** est créé.
* Un objet **Bus** est créé.
* Un objet **Conducteur** est créé.
* Au moins deux **stations** sont créées.
* Postconditions :
* Création d'un objet **Réservation**.
* **Association** de la réservation créée au voyage correspondant.
* Création d'un objet **Passager**.
* **Association** du **passager** créé à l'objet **réservation** nouvellement créé.
* **Association** des deux stations de départ et d'arrivée à l'objet réservation.
* Création d'un objet **Paiement**.
* **Association** du paiement à la réservation.
* Création d'un **ticket**.
* Association du ticket à la réservation.

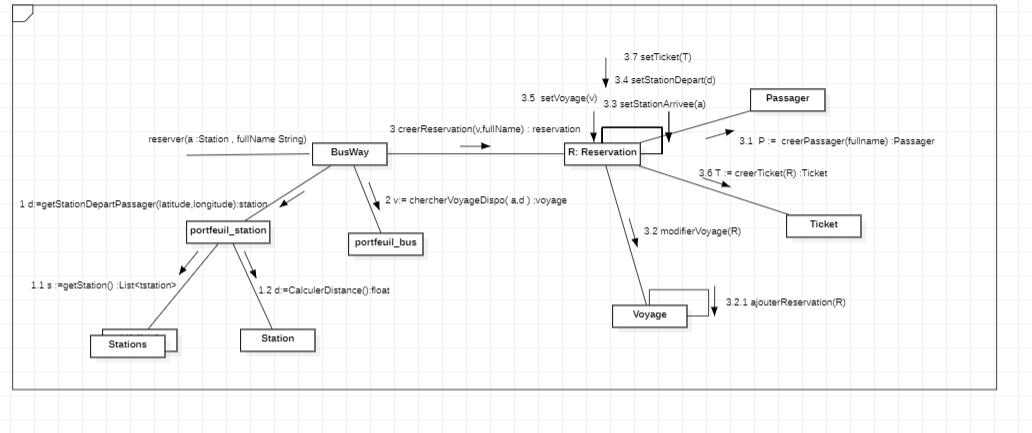


Figure 5 : Diagramme de communication <<Réserver un voyage>>

## Diagramme de classe de conception :

Figure 6 : Diagramme de classe de conception

# Réalisation :

## Stockage des données :

Dans notre projet de réservation de tickets de voyage, nous avons opté pour l'utilisation de Neo4j comme système de stockage des données concernant les bus et les stations. Cette décision a été prise en raison des avantages qu'offre Neo4j pour la représentation et la gestion des données dans un modèle de graphe.

Pour le stockage des données dans Neo4j, nous avons organisé notre schéma de la manière suivante :

#### Les nœuds du graphe :

**Nœud Station :**

Ce nœud représente une station dans notre système de réservation de tickets de voyage. Il est défini par les attributs suivants :

* **Nom** : nom de la station.
* **Latitude et Longitude** : coordonnées géographiques de la station.
* **End :** un booléen indiquant si la station est le terminus d'un voyage.

#### **Noeud Bus :**

Ce nœud représente un bus dans notre système. Il est caractérisé par les attributs suivants :

* **Matricule :** numéro de matricule du bus.
* **Capacité :** le nombre maximum de passagers que le bus peut transporter.
* **Conducteur :** identifiant du conducteur associé au bus.

#### Les relations du graphe :

**Relation Arret :**

Cette relation lie un bus à une station, indiquant qu'un bus s'arrête à cette station. Elle est caractérisée par les attributs suivants :

* **Heure :** l'heure à laquelle le bus s'arrête à cette station.
* **NbreAsc :** le nombre de passagers montant dans le bus à cette station.
* **NbreDesc :** le nombre de passagers descendant du bus à cette station.

**Relation Next :**

Cette relation connecte deux stations, indiquant qu'une station est suivie d'une autre dans un trajet de voyage.

#### La représentation des voyages

* Chaque voyage est représenté par une liste de relations entre les stations, utilisant la relation **"Next".** Cette approche permet de définir un itinéraire clair pour chaque trajet, en indiquant l'ordre des stations à parcourir.
* Le voyage débute à partir d'une station terminus, identifiée par l'attribut "End" égal à 1, et se poursuit jusqu'à atteindre une autre station terminus, également marquée par "End" égal à 1.
* Chaque arrêt effectué au cours du voyage est représenté par la relation "Arret" entre le bus et les stations concernées.
* Cette relation enregistre des détails importants tels que l'heure de l'arrêt, le nombre de passagers montant dans le bus à cette station (NbreAsc), ainsi que le nombre de passagers descendant du bus à cette station (NbreDesc).
* Pour déterminer la disponibilité d'un bus à une station donnée, nous utilisons la capacité du bus comme base de calcul. Pour ce faire, nous soustrayons le nombre de passagers montés à bord du bus à toutes les stations précédentes du voyage (NbreAsc) et nous ajoutons le nombre de passagers descendus du bus à ces mêmes stations (NbreDesc).
* Cette approche nous permet d'obtenir une estimation précise du nombre de places disponibles à la station actuelle du voyage. En ajustant ce calcul à chaque arrêt du trajet, nous pouvons fournir des informations en temps réel sur la disponibilité des sièges à bord du bus à chaque étape du voyage.

#### Persistance des réservations et les conducteurs des bus :

Pour persister les données des réservations effectuées par les passagers et les conducteurs des bus, nous avons utilisé une base de données MySQL.

**Réservation :**

Les détails de chaque réservation sont enregistrés dans une table appelée "réservation". Cette table contient les champs suivants :

* **Nom du passager** : le nom complet du passager effectuant la réservation.
* **Station de départ** : le nom de la station de départ choisie par le passager.
* **Station d'arrivée** : le nom de la station d'arrivée choisie par le passager.
* **Heure de départ** : l'heure d’arrivé du bus à la station de départ de passager.
* **Heure d’arrivé :** l’heure d’arrivé à la station destination du passager

**Conducteur :**

nous avons également utilisé une table, appelée "conducteurs", contient les champs suivants :

* **Nom :** le nom du conducteur.
* **Prénom** : le prénom du conducteur.
* **Âge :** l'âge du conducteur.
* **CIN** (Carte d'Identité Nationale) : le numéro de la carte d'identité nationale du conducteur, servant d'identifiant unique.

Dans l'illustration ci-dessous, trois bus sont représentés, identifiés par leurs matricules respectifs (123ABC, 456DEF, 789GHL). Chacun de ces bus est assigné à un voyage spécifique.

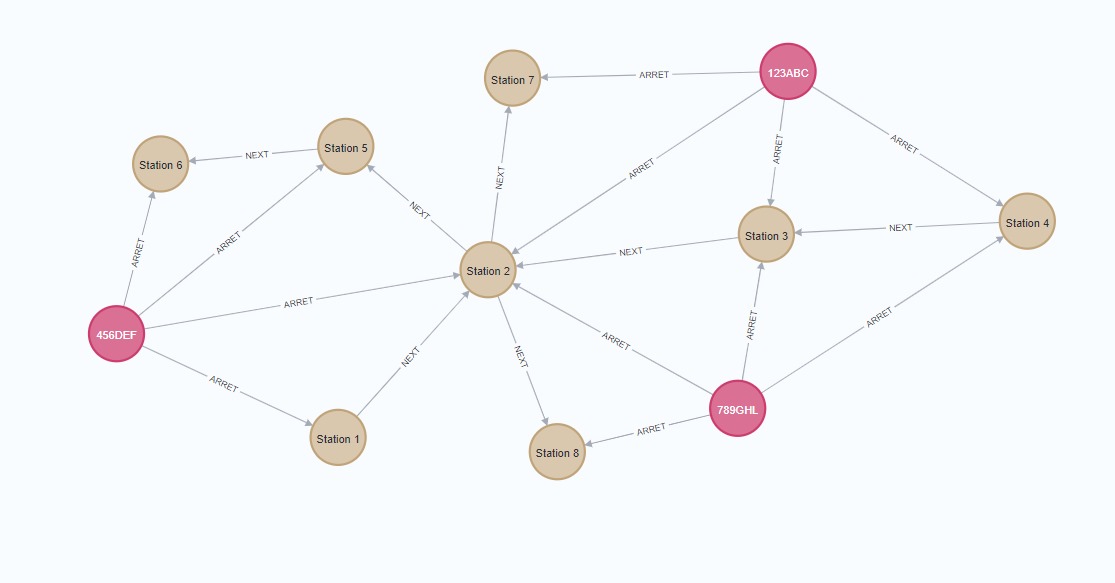


Figure 7 : Représentation du graphe

## Interfaces :

Dans cette phase de réalisation du projet, nous avons utilisé la bibliothèque Swing pour développer les interfaces utilisateur. Dans cette partie du rapport, nous présenterons en détail les différentes interfaces conçues, en expliquant leur fonctionnalité.

Interface Accueil :

* Dans cette page d'accueil, nous avons deux options :

1. **Réserver un ticket** Si on souhaite réserver un ticket en tant que passager
2. **Se Connecter en tant que Manager** Si on souhaite accéder aux fonctionnalités de manager   
   « consulter les voyages disponibles », « afficher les informations des bus ainsi que la réservation effectuée par les passagers »et enfin, << Installer les informations d'un bus>>

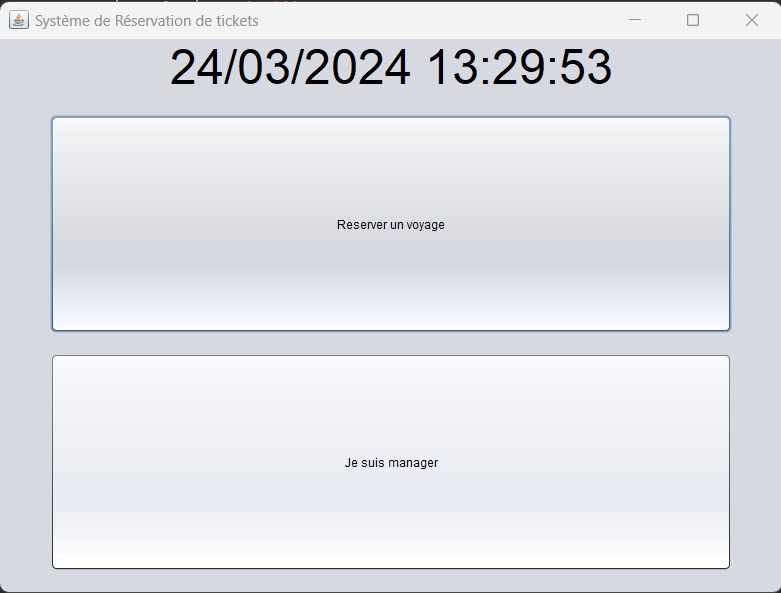


Figure 8 : L’interface d’accueil

Espace Passager :

### **L’interface de réservation**

* Dans cette interface, le contrôleur est responsable de déterminer la station la plus proche du passager.
* Ensuite, une liste déroulante des stations est affichée, permettant au passager de sélectionner sa station de destination. Cette liste est générée en chargeant les données des stations à partir du graphe à l'aide de requêtes Cypher.
* Une fois que le passager a sélectionné sa destination, le système recherche les voyages disponibles à partir de la station de départ (la plus proche du passager) qui assurent le voyage souhaité.
* Ces voyages sont ensuite affichés dans une liste déroulante, permettant au passager de choisir celui qui lui convient le mieux.
* Enfin, le passager est redirigé vers une page de confirmation une fois sa sélection effectuée.

# 

Figure 9 : L’interface de réservation

### **L’interface de confirmation :**

* Dans cette interface, les passagers peuvent saisir leur nom complet et de visualiser les détails du voyage qu'ils souhaitent réserver, notamment la station de départ (la plus proche du passager), la station d'arrivée choisie par le passager, ainsi que le numéro de matricule du bus qui assure ce voyage.
* Après validation, la réservation est enregistrée, le nombre de passagers montant dans le bus est incrémenté dans la relation d'arrêt entre le bus et la station de départ du passager, tandis que le nombre de passagers descendant du bus est incrémenté dans la relation d'arrêt entre le bus et la station d'arrivée du passager et un code QR est généré et affiché dans l'interface d'exportation du ticket.

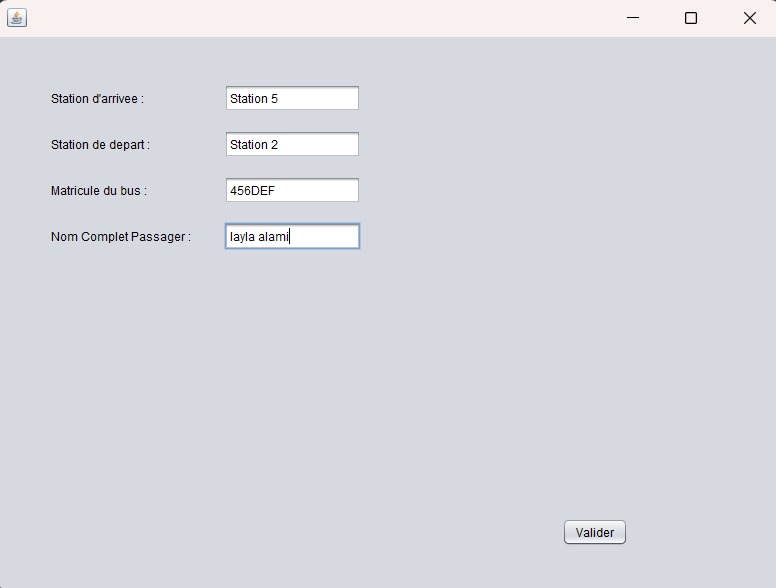


Figure 10 : L’interface de confirmation

### **L’interface d’exportation du ticket :**

Cette interface présente le code QR associé à la réservation du passager, accompagné d'un bouton "Exporter" qui permet de générer et de télécharger le billet au format PDF.

### 

Figure 11 : L’interface d’exportation du ticket

### **Le ticket exporté sous forme pdf :**

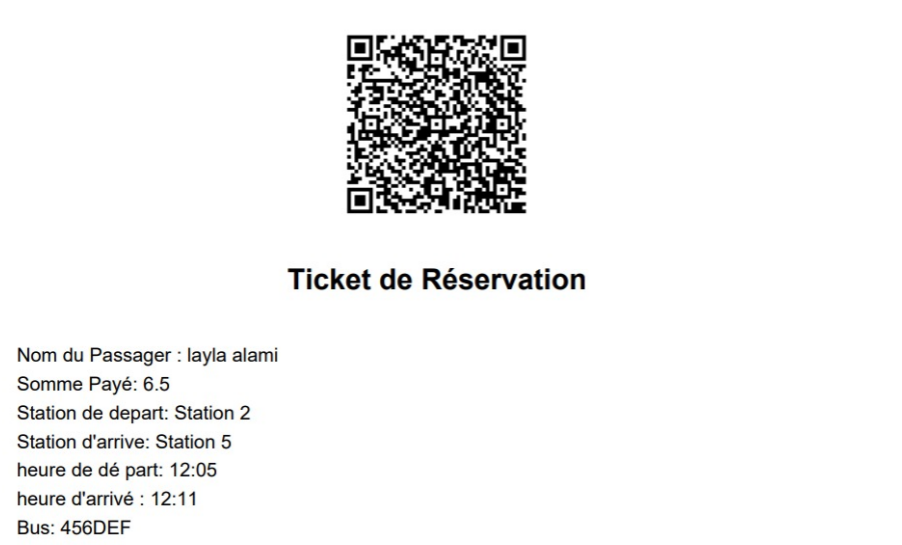


Figure 13 : Le ticket sous forme pdf

Espace Manager ou administrateur :

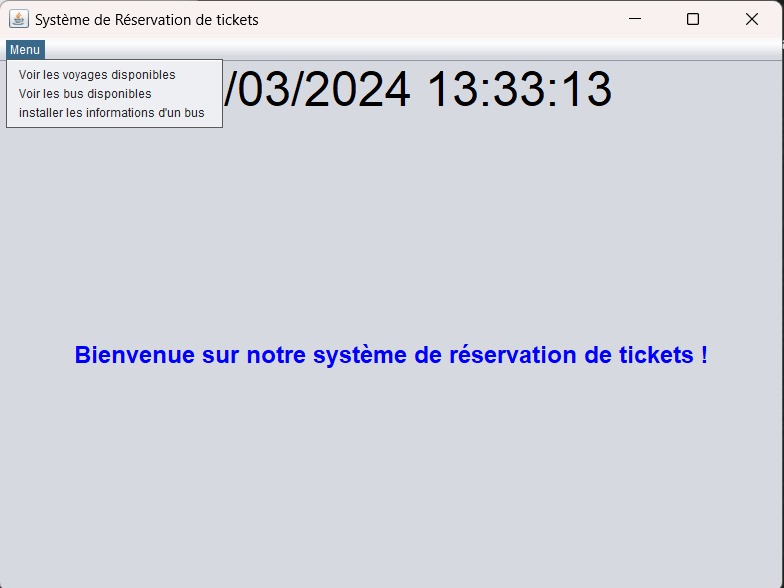
### **L’interface d’authentification de l’administrateur :**

### 

Figure 14 : Interface d’authentification

### **Interface d’accueil d’administrateur :**

La page d'accueil administrateur présente un menu offrant 3 options : consulter les voyages disponibles, afficher les informations des bus ainsi que les réservations effectuées par les passagers, et enfin, initialiser les informations d’un bus .

Figure 14 : Interface d’accueil administrateur

### **L’interface des voyages disponibles :**

Dans cette interface, nous affichons les voyages disponibles au niveau des stations. Pour chaque voyage, nous affichons le nom de la station où il est disponible, le nombre de places disponibles à cet arrêt, l'heure d'arrêt, ainsi que le matricule du bus.

Dans ce cas, dans notre graphe, le voyage assuré par le bus portant le matricule 789GHL est disponible aux stations « Station 2 »et « Station 8 », et est complet aux stations « Station 3 » et « Station 4 ». Ainsi, nous n'affichons que les arrêts du bus aux stations « Station 2 »et « Station 8 », où des places sont encore disponibles.

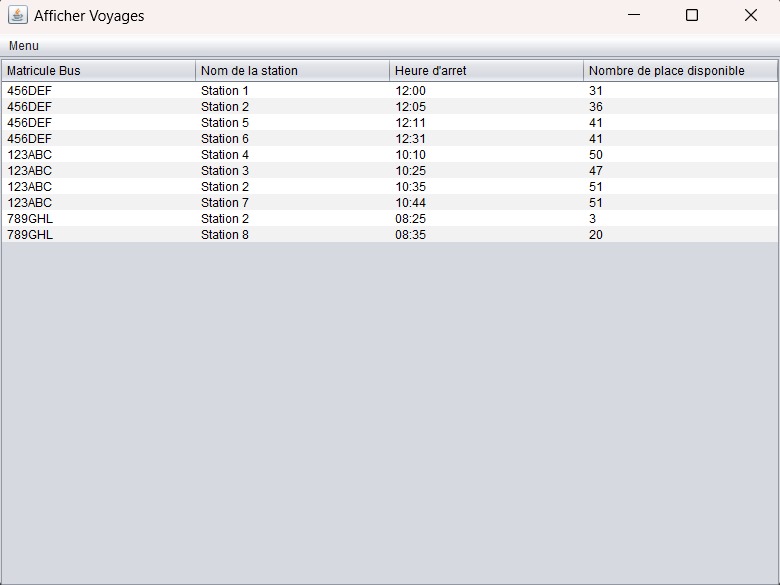


Figure 15 : Interface des voyages disponibles au niveau des stations

Haut du formulaire

### **L’interface de saisie des informations d'un bus**

Dans cette interface, le manager est autorisé à saisir la capacité du bus, son matricule et à choisir parmi les conducteurs disponibles dans notre base de données système. Deux boutons sont également présents :

* **"Ajouter",** pour enregistrer le bus en tant que nœud dans le graphe sans le lier à aucune station.
* **"Créer arrêt",** pour diriger le manager vers l'interface de création de relation d'arrêt entre le bus et une station du système, ce bouton est désactive jusqu’à ajout d’au moins un bus.

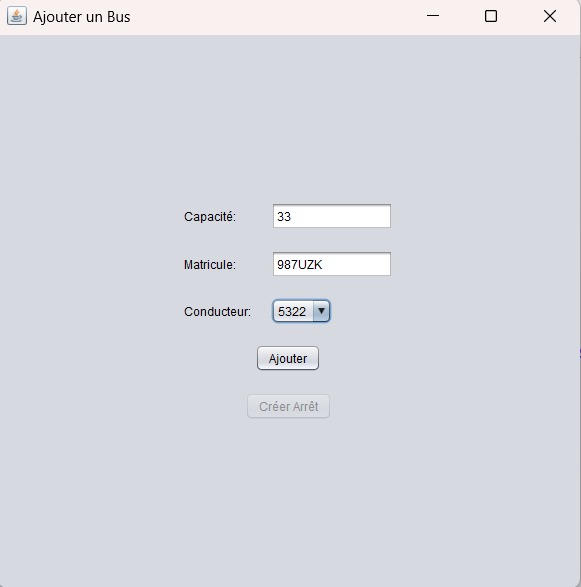


Figure 16 : Interface d’ajout d’un bus

### **Interface pour l'ajout des arrêts au bus créé**

* Dans cette interface, une liste déroulante des stations terminus, ayant au moins une relation NEXT avec une autre station, pour le premier arrêt a créé, ensuite on affiche les stations Next de ce terminus choisis et ainsi dessus jusqu’à arrive au dernier terminus. On Récupéré ces données à partir de notre graphe à l'aide d'une requête Cypher, est affichée.
* Un champ heure est également présent pour lier un bus à une station via la relation d'arrêt du graphe, indiquant l'heure à laquelle le bus a fait halte à cette station.
* Enfin, un bouton "Créer" permet d'ajouter la relation au graphe entre les nœuds bus et station.



Figure 17 : Interface d’ajout des arrêts au bus crée

Dans les figures ci-dessus, nous avons ajouté un bus avec le matricule 987UZK qui effectue le trajet de la Station 1 à la Station 2, puis à la Station 8. Ci-dessous, vous pouvez voir le graphe après cet ajout.

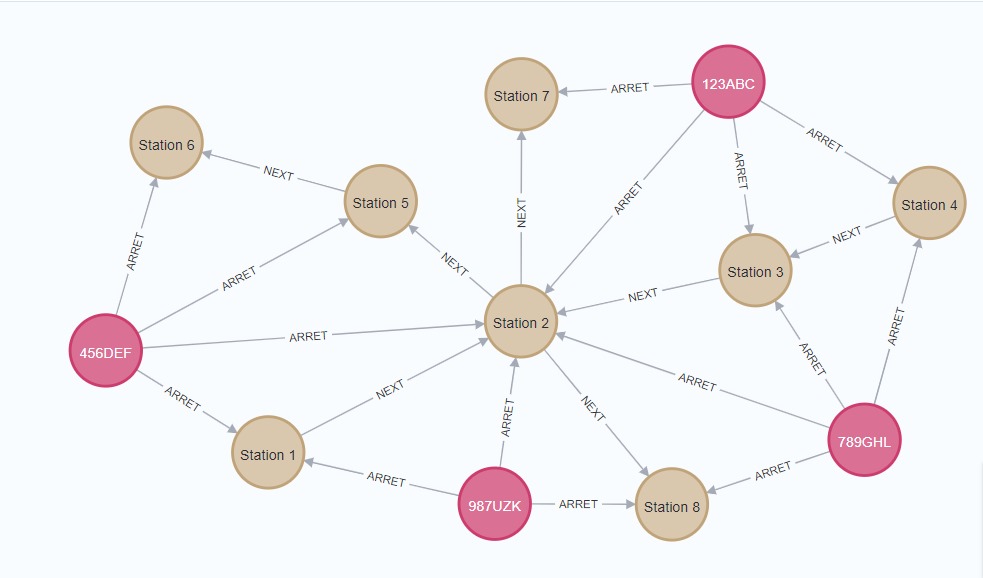


Figure 18 : Le graphe après l’ajout du bus « 987UZK »

### **L’interface des bus disponibles :**

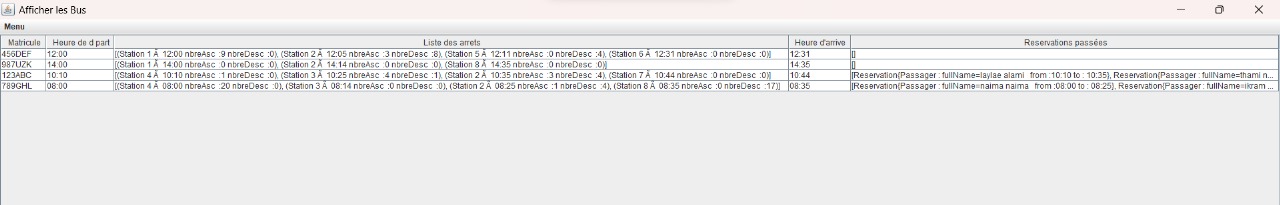
* Dans cette interface, nous présentons l'ensemble des bus disponibles dans notre système, chacun assurant au moins un voyage.
* Pour chaque bus, les détails des voyages sont affichés, notamment l'heure de départ, l'heure d'arrivée, ainsi que les stations d'arrêt du voyage. De plus, les réservations effectuées tout au long de chaque voyage sont également affichées.

Figure 19 : Le graphe après l’ajout du bus « 987UZK »