| Software design |
| --- |
| Train booking System |
| Version 1.0.4 |
| Prepared by class SE1735, Group 4  Phạm Quang Thắng  Phạm Hoàng Tùng Nguyễn Thế Anh Trường  Đồng Trung Hiếu  Nguyễn Văn Đạt  Oct 27, 2023 |

Lecturer: Nguyễn Thúy Hường

Class SE1735\_Net  
Group 4

**Contents**

[**Revision history**](#_heading=h.gjdgxs) **5**

[**1. Document description**](#_heading=h.30j0zll) **6**

[**2. SRS overview**](#_heading=h.1fob9te) **6**

[2.1. List of system users](#_heading=h.3znysh7) 6

[2.2. List of use cases](#_heading=h.2et92p0) 6

[2.3. Use case diagram](#_heading=) 7

[2.3.1. Preliminary use case diagram](#_heading=h.3dy6vkm) 7

[2.3.2. Use case detail diagrams](#_heading=) 7

[2.3.2.1. Use case “Book Trains Ticket”](#_heading=) 7

[2.3.2.2. Use case “Cancel Ticket”](#_heading=) 8

[2.3.2.3. Use case “VIew Ticket Details”](#_heading=) 8

[2.3.2.4. Use case “Check Train Schedule”](#_heading=) 8

[2.3.2.5. Use case “Add Ticket”](#_heading=h.xgi26jy01p1h) 8

[2.4. Activity diagrams of use case scenarios](#_heading=) 9

[2.4.1. Activity diagram for use case “Book Trains Ticket”](#_heading=h.2s8eyo1) 9

[2.4.2. Activity diagram for use case “Cancel Trains Ticket”](#_heading=h.17dp8vu) 10

[2.4.1. Activity diagram for use case “Add Ticket”](#_heading=h.3rdcrjn) 10

[**3. Data model**](#_heading=h.26in1rg) **11**

[3.1. Entity Relationship Diagram](#_heading=h.lnxbz9) 11

[3.2. Table details/Chi tiết các bảng](#_heading=) 11

[3.2.1. Table name: Ticket](#_heading=h.44sinio) 11

[3.2.2. Table name: Train](#_heading=h.2jxsxqh) 12

[3.2.3. Table name: Wagon](#_heading=h.z337ya) 12

[3.2.4. Table name: Bookings](#_heading=h.3j2qqm3) 12

[3.2.5. Table name: Type](#_heading=h.1y810tw) 12

[3.2.6. Table name: Ticket\_Class](#_heading=h.ps81clqpcnoz) 13

[**4. Class diagram (p.61 – p70)**](#_heading=h.4i7ojhp) **14**

[4.1. List of classes in the Layers](#_heading=h.2xcytpi) 14

[4.2. Modeling the Class Behaviors - sequence diagrams ( p. 66, 77)](#_heading=h.1ci93xb) 14

[4.2.1. Sequence diagram for use case “Borrow copy”](#_heading=) 15

[4.2.2. Sequence diagram for use case “Return copy”](#_heading=h.2bn6wsx) 16

[4.2.3. Sequence diagram for use case “Reserve Book”](#_heading=h.qsh70q) 17

[**5. User Interface Model Design (p. 70 – p.74)**](#_heading=) **18**

[5.1. User interface model design](#_heading=h.1pxezwc) 18

[51.1. Screen “Register books”](#_heading=) 18

[51.2. Screen “Register members”](#_heading=h.49x2ik5) 18

[51.3. Screen “Borrow books”](#_heading=) 19

[51.4. Screen “Return books”](#_heading=) 19

[51.5. Screen “Reserve books”](#_heading=) 20

[5.2. Interface Flow Diagrams (p. 51)](#_heading=) 20

# Revision history

| **Date** | **Authors** | **Modifications** | **Version** |
| --- | --- | --- | --- |
| 25/10/2023 | Phạm Quang Thắng | Create Documents | 1.0.1 |
| 26/10/2023 | Phạm Hoàng Tùng | Create Section 1 , Section 2 | 1.0.2 |
| 27/10/2023 | Phạm Hoàng Thắng,  Nguyễn Văn Đạt | Create Section 3 , Section 4 | 1.0.3 |
| 28/10/2023 | Phạm Hoàng Tùng | Create Section 5 | 1.0.4 |

# 1. Document description

This document describes OO solution version 1.0 for the Train booking system. It is divided into the following sections:

* In section 2 an overview of the Train booking system’s requirements will be provided, including a list of system users, use cases, use case diagrams, and activity diagrams that describe the most important use case scenarios.
* In section 3 an E-R diagram and all tables of the Train booking system in detail will be presented.
* In section 4 all classes organized in the layers and their behaviors (sequence diagrams) for the most important use cases will be presented.
* In section 5 the most screen prototypes will be presented.

# 2. SRS overview

## 2.1. List of system users

1. Admin: The person is working in the Train booking room

2.Passenger: The person uses services of the Train booking system

## 2.2. List of use cases

1. Login

2. Register

3. Book Trains Ticket

4. Cancel Trains Ticket

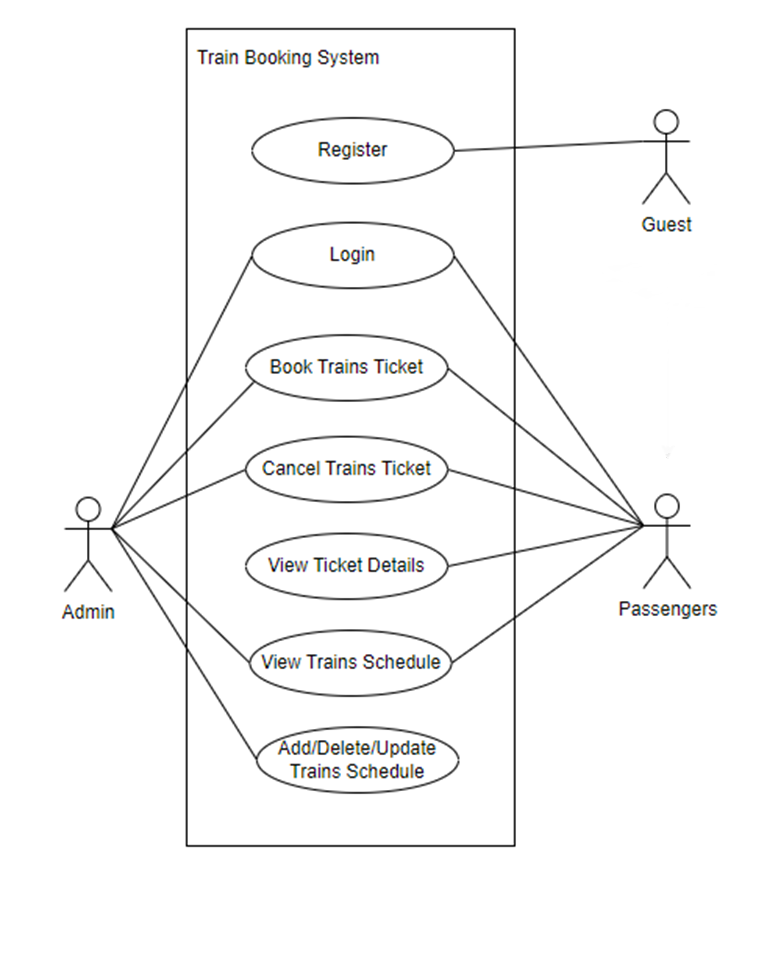
5. View Ticket Details

6. View Trains Schedule

7. Add/Delete/Update Trains Schedule

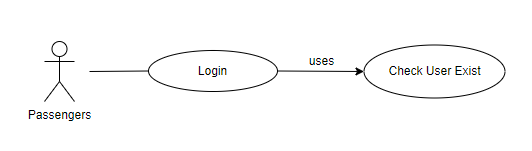
## 2.3. Use case diagram

### 2.3.1. Preliminary use case diagram

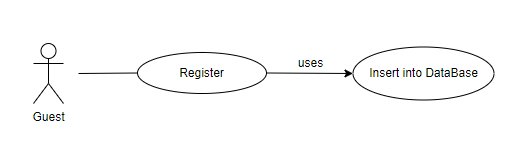


### 2.3.2. Use case detail diagrams

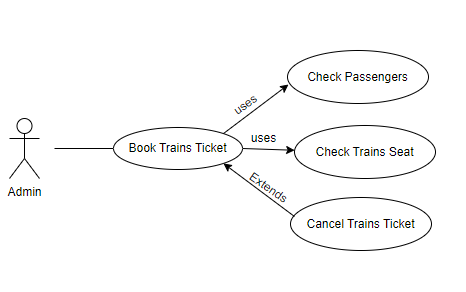
#### 2.3.2.1. Use case “Login”



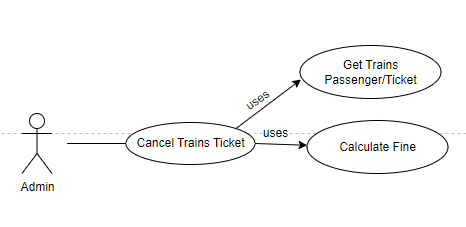
#### 2.3.2.2. Use case “Register”



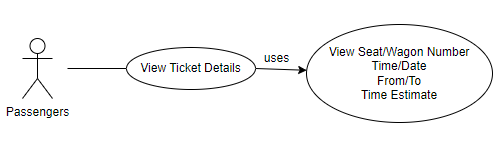
#### 2.3.2.3. Use case “Book Trains Ticket”



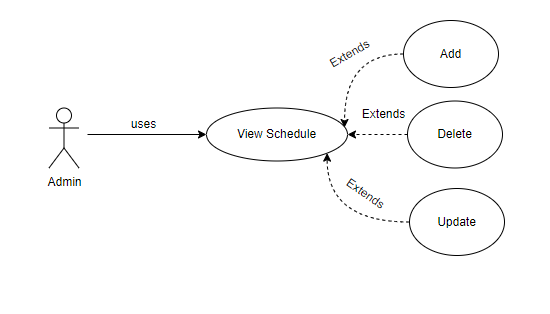
#### 2.3.2.4. Use case “Cancel Trains Ticket”



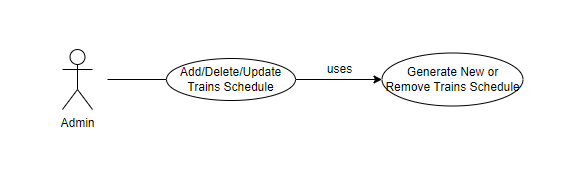
#### 2.3.2.5. Use case “VIew Ticket Details”



#### 2.3.2.6. Use case “View Train Schedule”



#### 2.3.2.5. Use case “Add/Delete/Update Trains Schedule”



## 2.4. Activity diagrams of use case scenarios

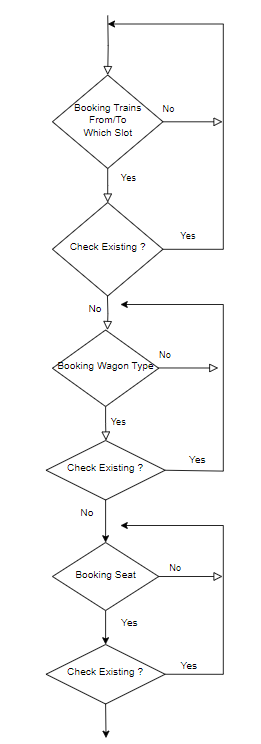
### 2.4.1. Activity diagram for use case “Book Trains Ticket”

Get Train From/To and Slot

Get Wagon Type

Get Seat

Get Ticket (Update Ticket Into Server)



### 

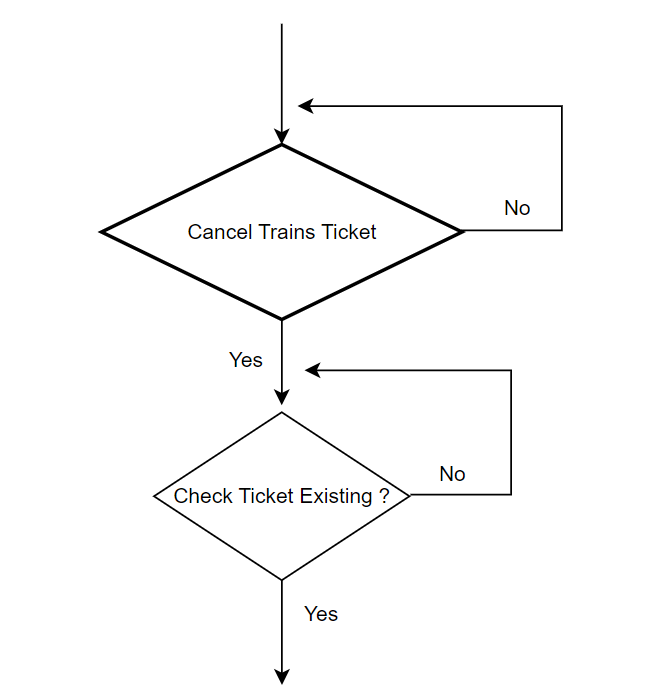
### 2.4.2. Activity diagram for use case “Cancel Trains Ticket”

Get Ticket Code

Check if it exist

Calculate Fine Amount

Return (The seat will be deleted if cancel successfully)

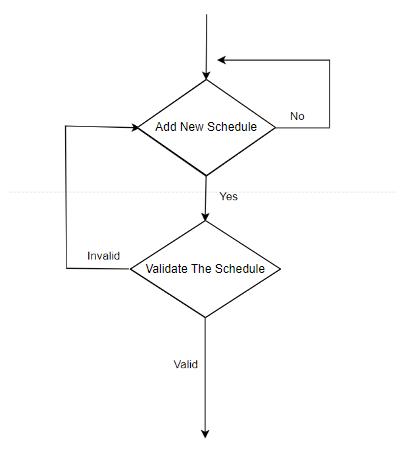


### 2.4.1. Activity diagram for use case “Add Trains Schedule”

Select the new Schedule

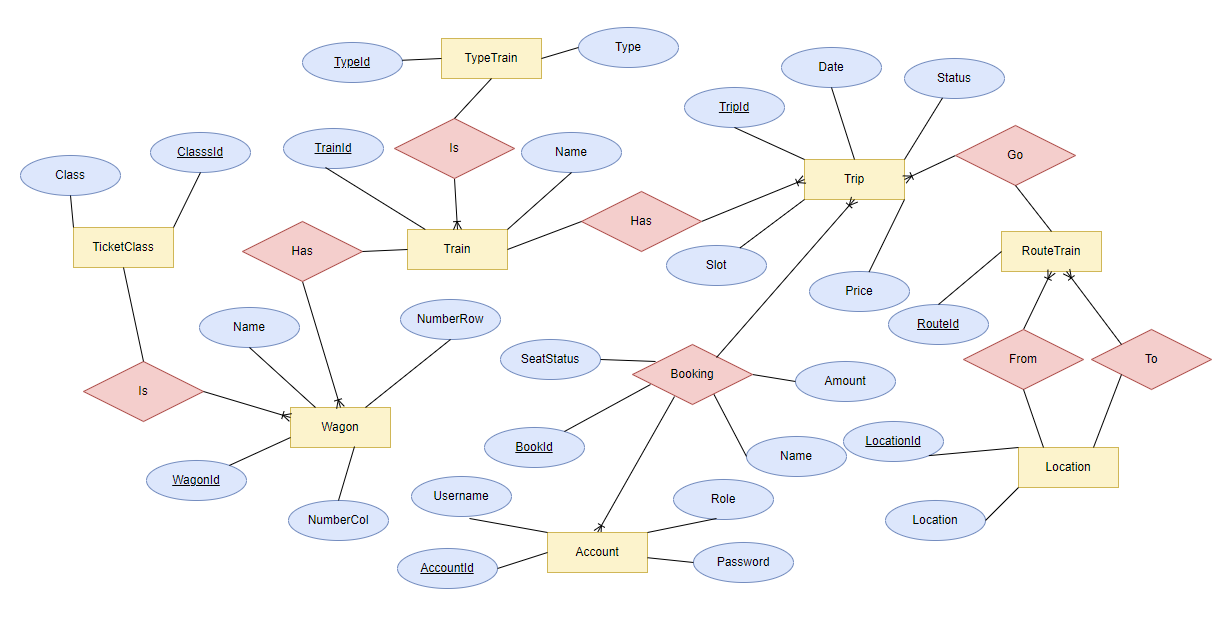
Set the new Trains Schedule

Update Into Server

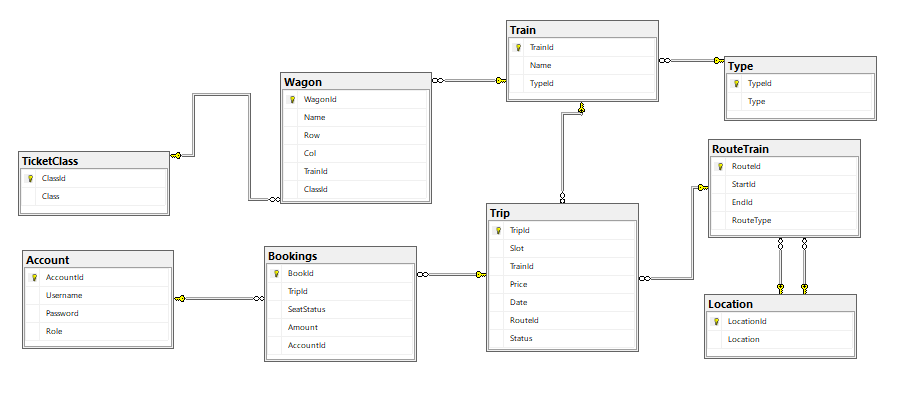


# 3. Data model

## 3.1 Entity Relationship Diagram



## 3.2 Table details



### 3.2.1 Table name: Trip

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| TripId | int | Primary key |
| Slot | int | Slots start at 0:00 PM, each slot is 2 hour apart |
| RouteId | int | Foreign key, refers to Route table (RouteId) |
| Status | bin | Status of whether tickets are still available or sold out for that trip (0 is sold out and 1 is avaiable) |
| TrainnId | int | Foreign key, refers to Train table |
| Price | money | Prices are based on train type, class of wagon and distance traveled |
| Date | date | Day of the trip |

### 3.2.2 Table name: Train

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| TrainId | int | Primary key |
| Name | nvarchar(100) | Train code and train name |
| TypeId | int | Foreign key, refers to Type table |

### 3.2.3 Table name: Wagon

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| WagonId | int | Primary key |
| Name | nvarchar(100) | Train car code |
| Row | int | Number of rows of seats in a train |
| Col | int | Number of columns of seats in a train |
| TrainId | int | Foreign key, refers to Train table |
| ClassId | int | Foreign key, refers to Ticket\_Class table |

### 3.2.4 Table name: Bookings

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| BookId | int | Primary key |
| TriptId | int | Foreign key, refers to Ticket table (TicketId) |
| AccountId | int | Foreign key, refers to Account table (AccountId) |
| SeatStatus | nchar(100) | Status of seat |
| Amount | float | Total amount of tickets purchased |

### 3.2.5 Table name: TypeTrain

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| TypeId | int | Primary key |
| Type | nvarchar(100) | Type of the train |

### 3.2.6 Table name: TicketClass

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| ClassId | int | Primary key |
| Class | nvarchar(100) | Ticket class of each train car |

### 3.2.7 Table name: RouteTrain

### 

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| RouteId | int | Primary key |
| StartId | int | Foreign key, refers to Location table (LocationId) |
| EndId | int | Foreign key, refers to Location table (LocationId) |
| RouteType | int | Type of route, which will affect price of trip |

### 

### 3.2.8 Table name: Location

### 

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| LocationId | int | Primary key |
| Location | nvarchar(100) | Locations included in the trip |

### 

### 3.2.9 Table name: Account

### 

| **Field name** | **Type** | **Note** |
| --- | --- | --- |
| AccountId | int | Primary key |
| Username | nvarchar(100) | Username of this account |
| Password | nvarchar(100) | Password of this account |
| Role | bit | Role of this account (0 is user and 1 is admin) |

### 

# 4. Class diagram (p.61 – p70)

## 4.1 List of classes in the Layers

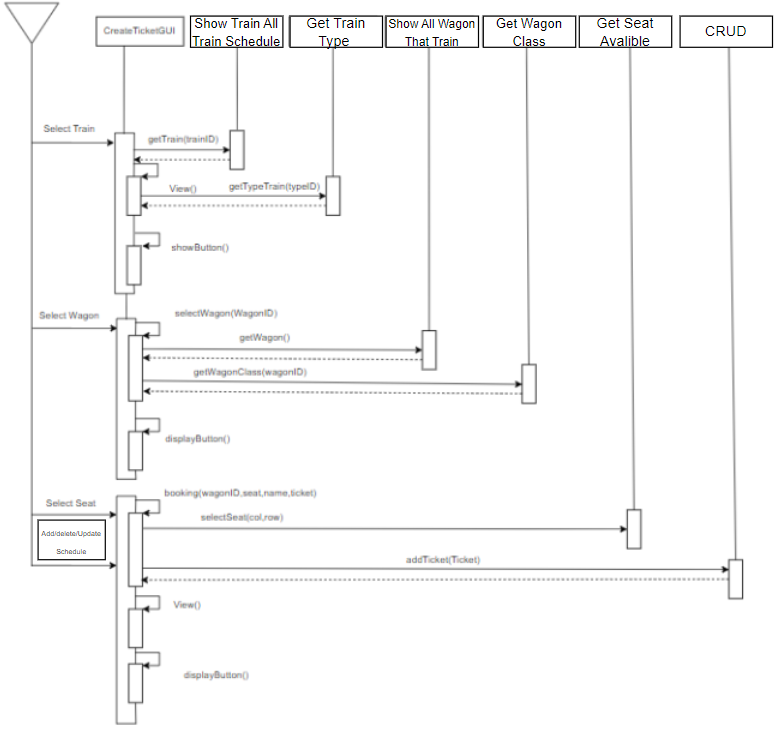
## 

## 

## 

## 4.2 Modeling the Class Behaviors - sequence diagrams ( p. 66, 77)

### 4.2.1 Sequence diagram for use case “Booking Train”



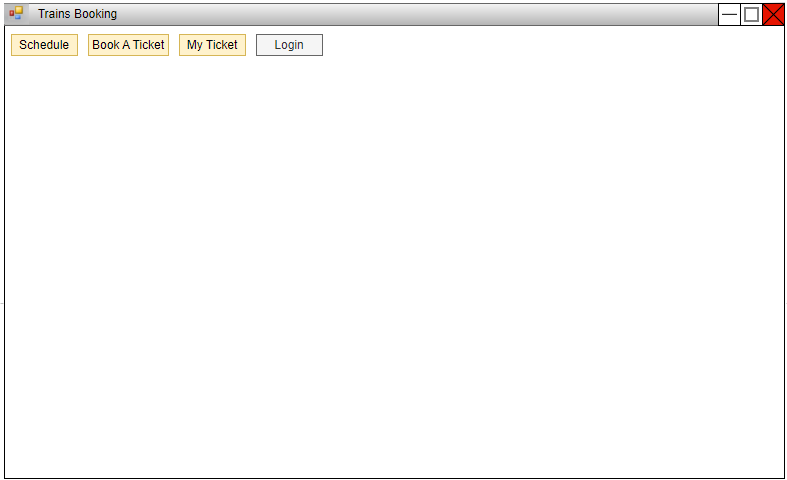
### 

# 5. User Interface Model Design (p. 70 – p.74)

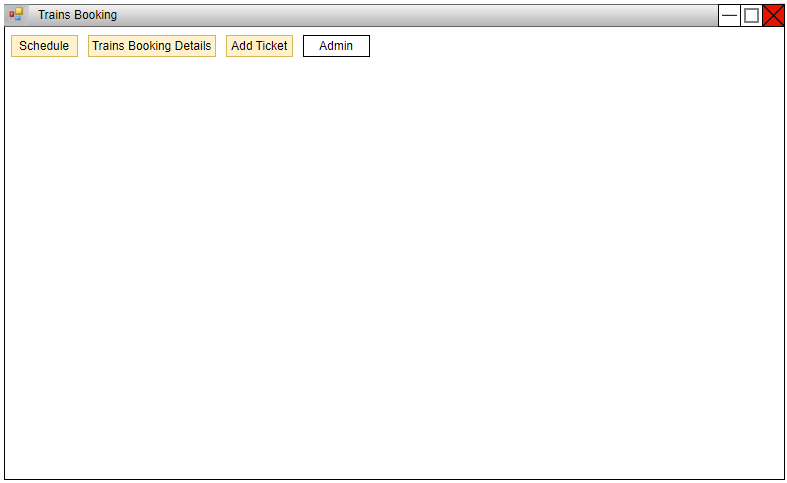
## 5.1 User interface model design

### 5.1.1. Screen “Main”

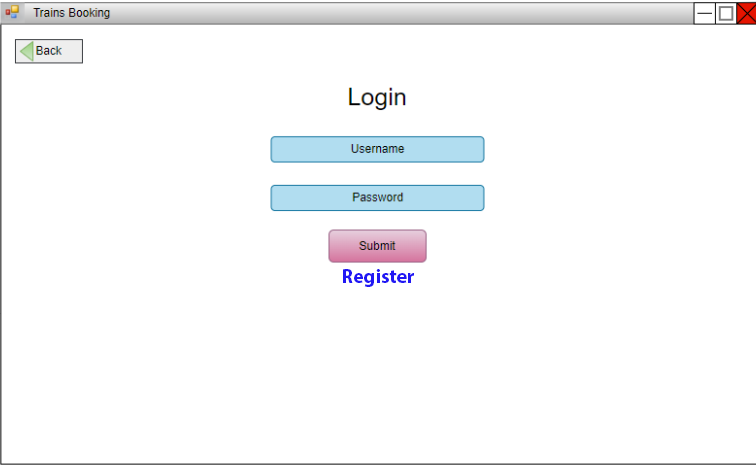
* Users Screen



* Admin Screen



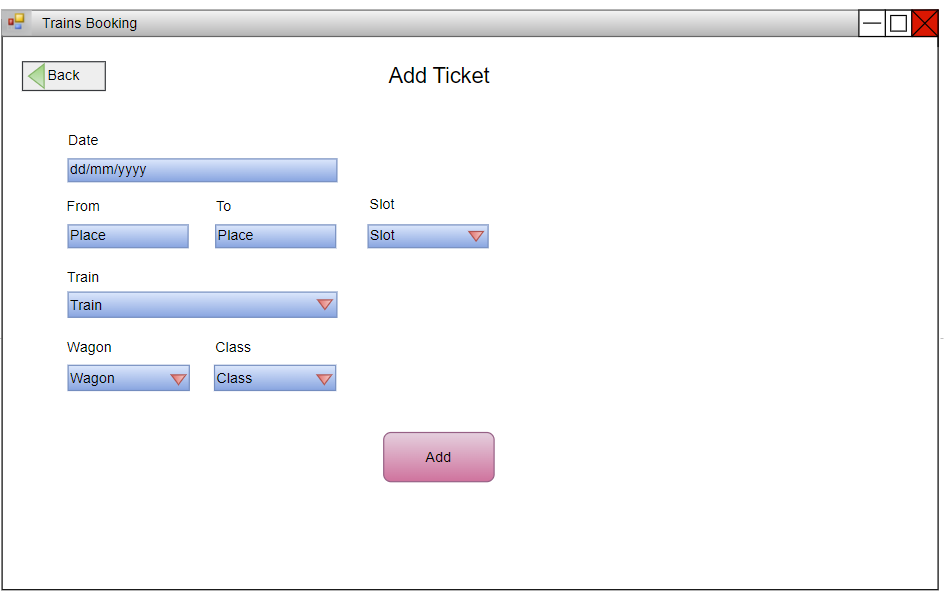
### 5.1.2. Screen “Login”



### Screen “Register”

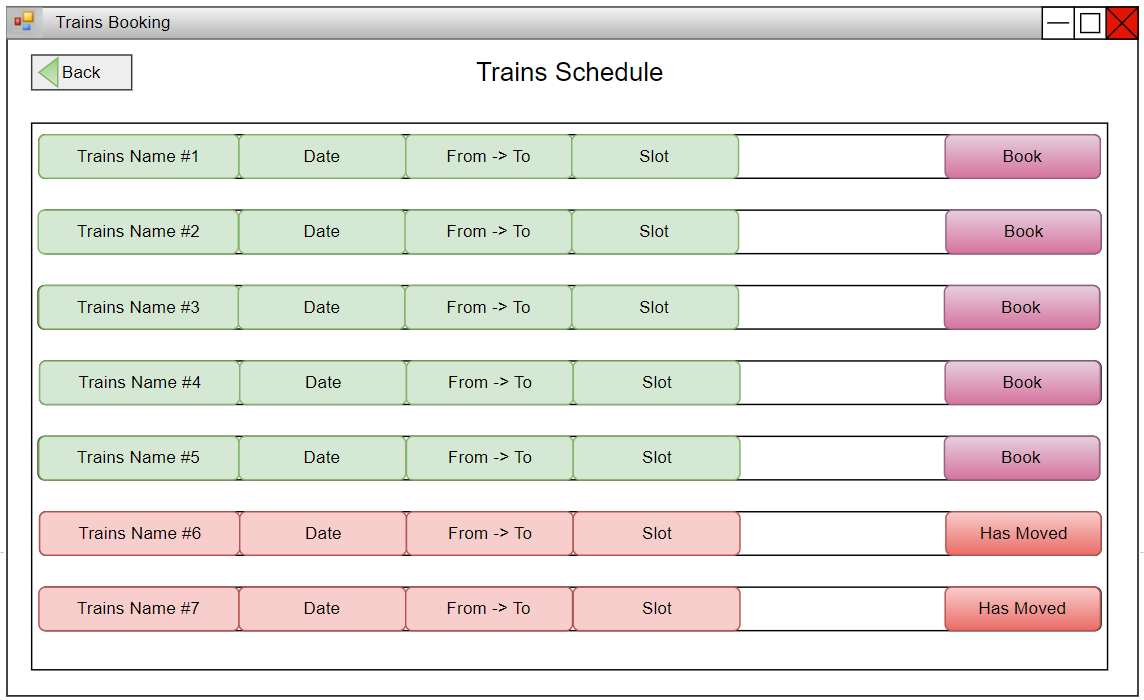
### 5.1.3. Screen “Add Ticket”

* Admin’s Screen

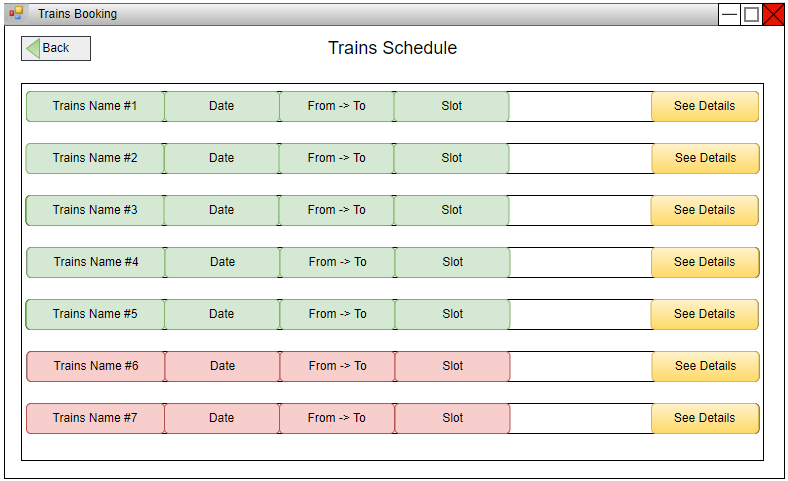


### 5.1.4. Screen “Trains Schedule”

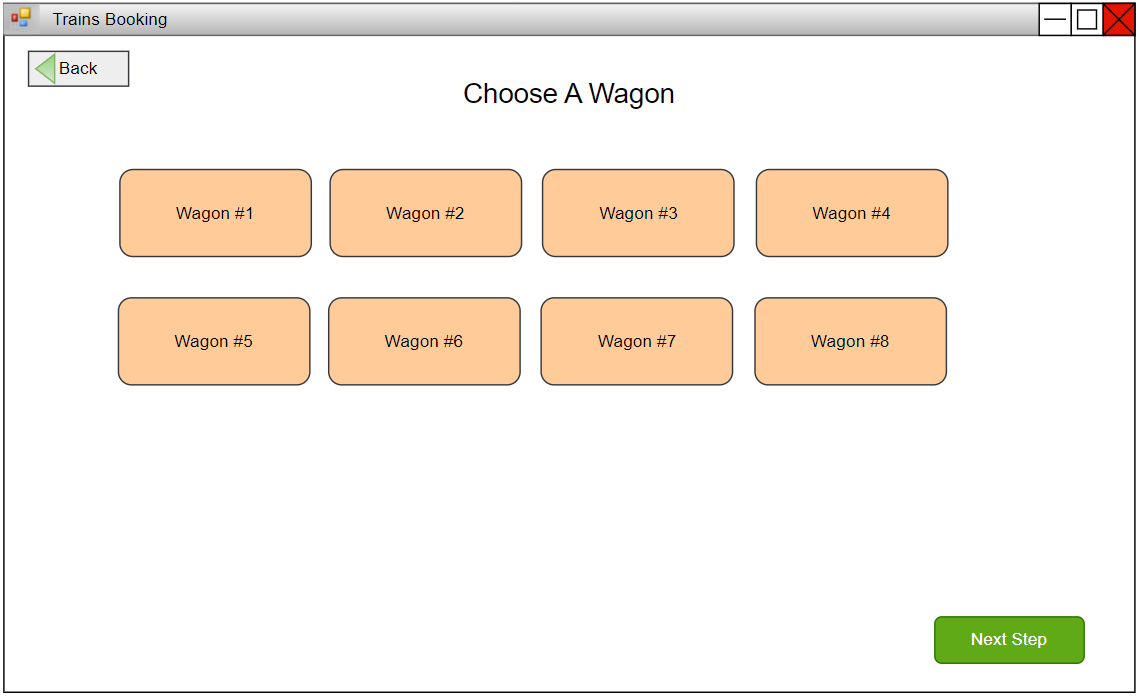
* Users Screen



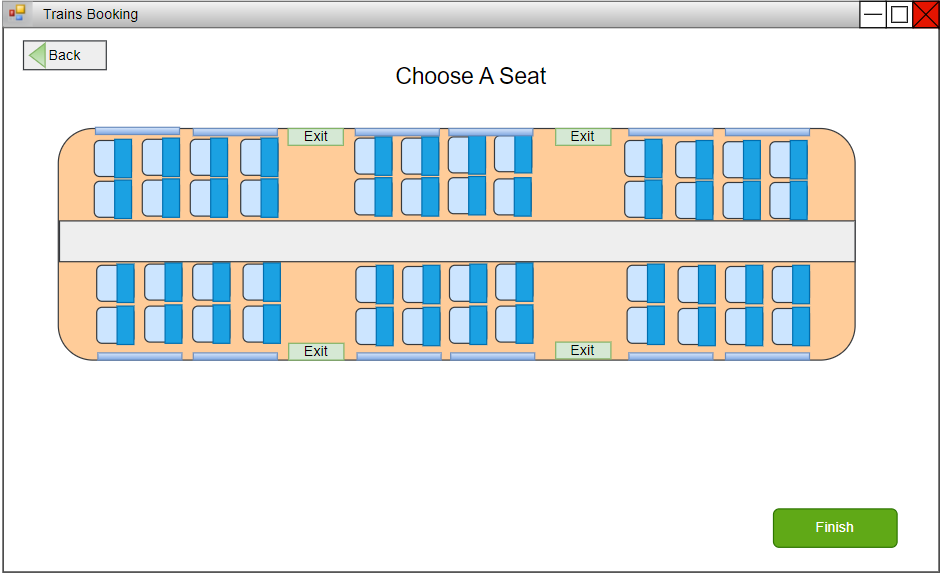
* Admin’s Screen



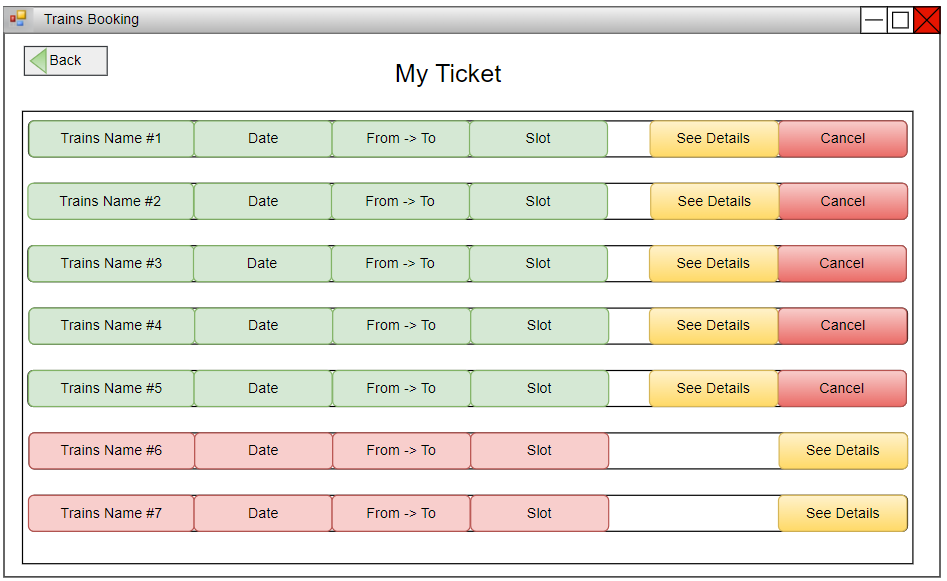
### 5.1.5. Screen “Book Wagon”



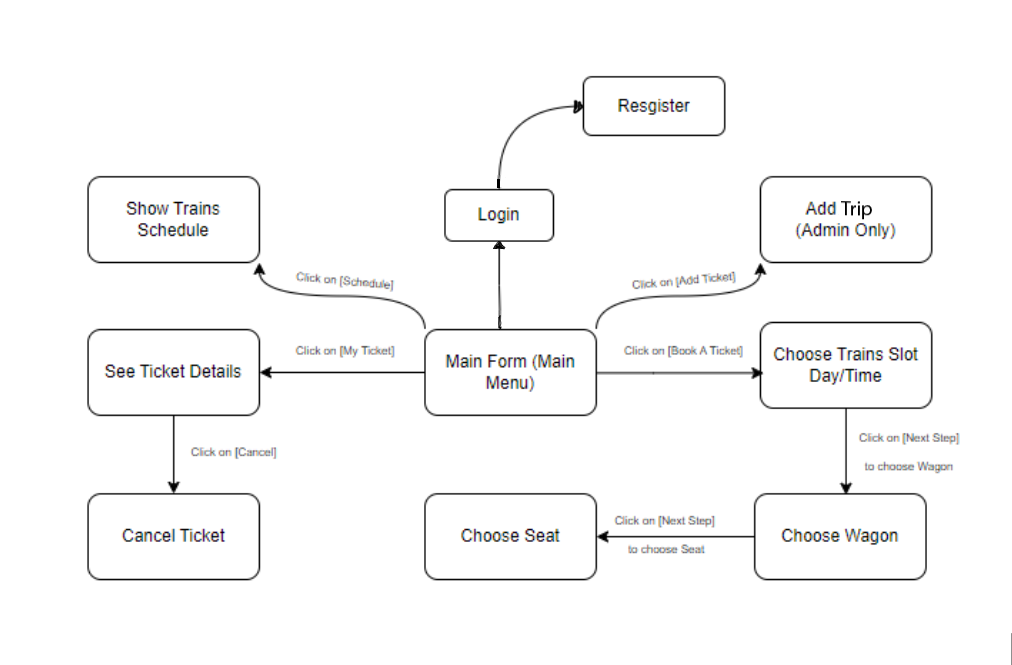
### 5.1.6. Screen “Book Seat”



### 5.1.7. Screen “My Ticket”



## 5.2 Interface Flow Diagrams (p. 51)

****