

**MÔN HỌC: ĐỒ ÁN CHUYÊN NGÀNH**

*GV: Lê Phi Hùng*

TÊN ĐỀ TÀI: **HỆ THỐNG ĐẶT VÉ XE**

DANH SÁCH THÀNH VIÊN NHÓM:

18130284 NGUYỄN HOÀNG VINH

18130085 NGUYỄN HỮU VINH

18130292 TRƯƠNG VĂN XINH

18130295 LÂM HÀ YẾN

18130293 NGUYỄN NGỌC THANH XUÂN

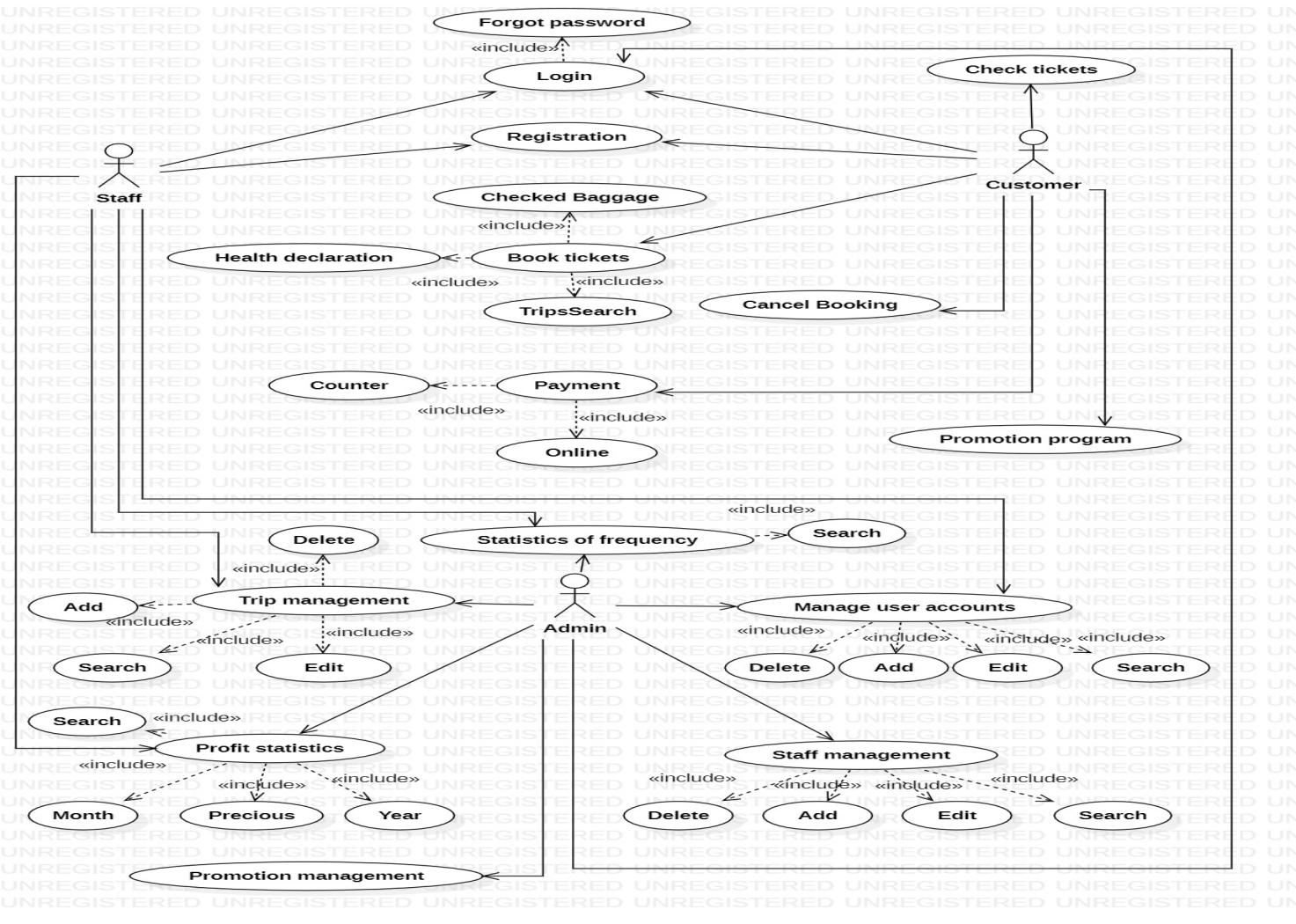
1. **PHÁT BIỂU:**

Currently, in the era of industrialization and modernization, applications are born and developed strongly, and human requirements are also rich. The car booking application is built based on the needs of people to move, helping people to use it easily in booking tickets, without crowding around at bus stations, especially in the stressful epidemic situation. like nowadays.

The ticket booking application we built includes the following benefits:

* Convenient in booking tickets.
* Payment via bank account.
* Easy and cheap consignment.
* There are many promotions.
* There is a transparent medical declaration for customers taking part in the move.
* Easy to choose a seat without fear of duplicating places like a traditional car reservation.
* Tickets can be canceled within the specified time.
* Book multiple tickets at the same time when traveling with relatives…..

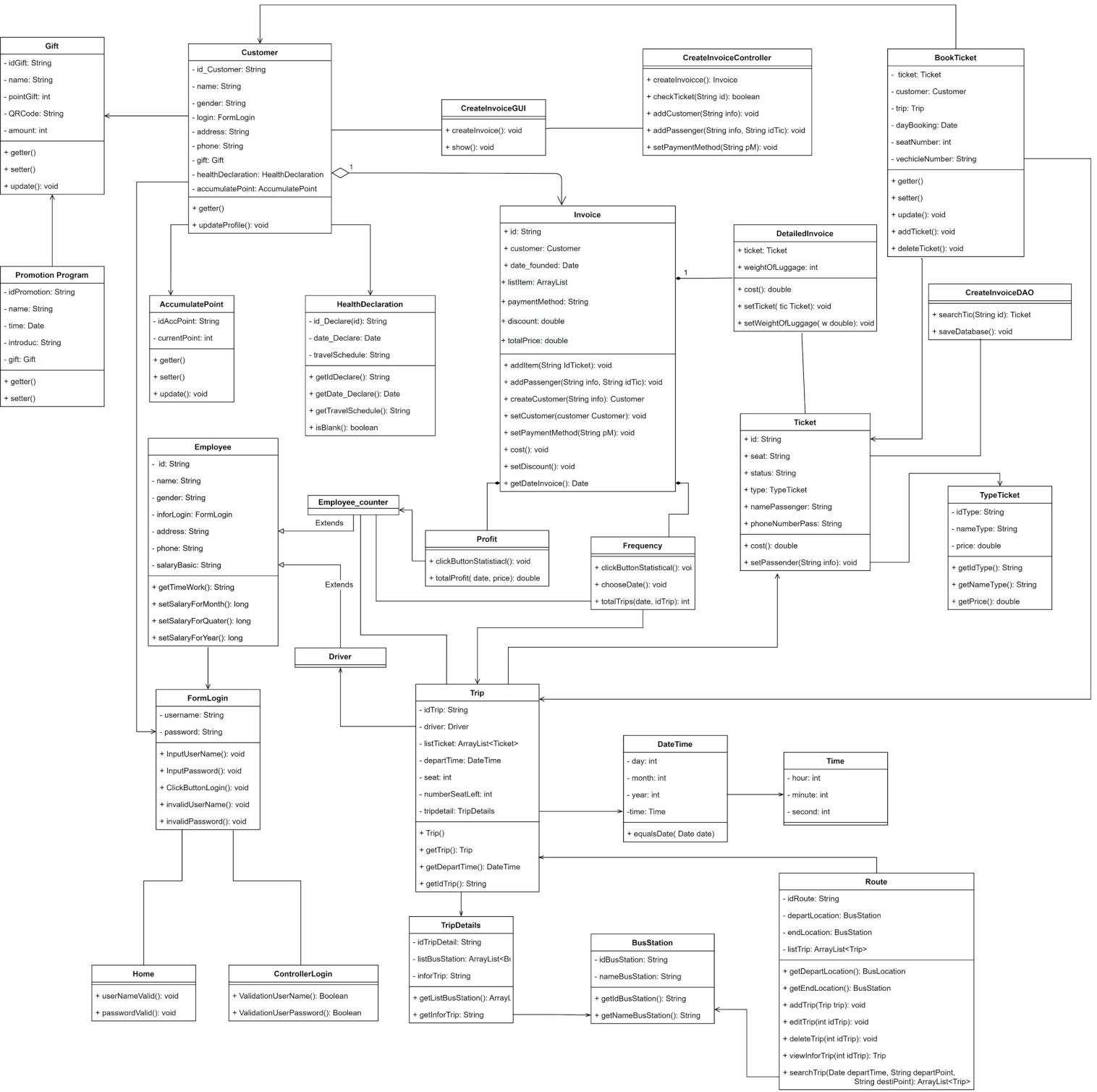
1. **USECASE DIAGRAM:** CHUNG CHO TOÀN HỆ THỐNG

****

LIỆT KÊ ĐÁNH SỐ THỨ TỰ CÁC USECASE

|  |  |
| --- | --- |
| **STT** | **USECASE** |
| 1 | Registration |
| 2 | Checked Baggage |
| 3 | Promotion |
| 4 | Login |
| 5 | Profit statistics |
| 6 | Statistics of frequency |
| 7 | Health Declaration |
| 8 | Promotion Program |
| 9 | Check ticket |
| 10 | Payment |
| 11 | Employee Management |
| 12 | Booking Ticketing |
| 13 | Cancel Booking |
| 14 | Manage Trip |
| 15 | Search Trip |

1. **CLASS DIAGRAM CHUNG CHO BÀI TOÁN**

****

1. **DATABASE DIAGRAM CHUNG CHO BÀI TOÁN**

**Qr code

Description automatically generated**

1. **TỔNG HỢP.**

**1. Registration.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Registration | Trương Văn Xinh |  |  |  |

**1.1 Use case Registration.**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Registration | |
| **Actor(s)** | Staff, customer | |
| **Description** | * For customers who register as a member to have more functions for booking tickets, or new employees can register here and are granted user rights to use by the admin. | |
| **Trigger event** | * Staff and customers register. | |
| **Trigger type** | xExteal | Temporal (Timing) |
| **Precondition** | * Staff, customers registered successfully. | |
| **Step performed** | Information per step.  1. Click register on the login form.  2. Fill out information marked with (\*)  + Password  + Full name  + Email  + Phone number  3. Click registration | |
| **Post condition** | * The account registered will not be identical with the existing accounts. | |
| **Assumptions** | * No new account registration customers will not have any new accounts added to the database. | |
| **Requirement met** | * In the database, there must be customer information that has just been registered and must be clearly authorized. | |
| **Outstanding issues** | * When registering a new account, if you have the same username, you will not be registered and must change your username again. | |
| **Priority** |  | |
| **Risk** |  | |

* 1. **Prototype**

**Graphical user interface, application

Description automatically generated**

* 1. **Activity**

**Graphical user interface, diagram

Description automatically generated**

* 1. **Sequence**

**Diagram

Description automatically generated**

* 1. **Database**

**Text

Description automatically generated**

**1.6 Class diagram**

**Timeline

Description automatically generated**

**2. Checked Baggage.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Checked Baggage | Truong Van Xinh |  |  |  |

* 1. **Use case**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Checked Baggage | |
| **Actor(s)** | Customer, staff | |
| **Description** | Help customers transport goods to where they need to ship. | |
| **Trigger event** | The customer conducts a system request to transport goods. | |
| **Trigger type** | x Extenal | Temporal (Timing) |
| **Precondition** | When customers request to deposit goods successfully. | |
| **Step performed** | Information per steps.  1. Customers click on Checked Baggage.  2. Customers enter information marked with (\*)  + Full name of the recipient  + Address  + Phone number  + Quantity of goods  + Weight of goods  + Size of goods  3. Customers click to send the goods, then the staff will reply to the customer later. | |
| **Post condition** | Checked Baggage consignment has been successfully deposited | |
| **Assumptions** | Without the customer consignment data will not be available in the database. | |
| **Requirement met** | Customers must log in to the system to send goods. | |
| **Outstanding issues** | The goods were not identified. | |
| **Priority** | Priority is given to customers who deposit their goods in advance from time to time | |
| **Risk** | Goods could not be delivered to receiver | |

**2.2 Prototype**

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. **Activity**

**Graphical user interface, application

Description automatically generated**

* 1. **Sequence**

**Diagram

Description automatically generated with medium confidence**

* 1. **Database**

Graphical user interface, application, Teams

Description automatically generated

* 1. **Class diagram**

**Graphical user interface, text, application

Description automatically generated**

**3. Promotion.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Promotion | Truong Van Xinh |  |  |  |

* 1. **Use case**

|  |  |  |
| --- | --- | --- |
| **Use case name** | Promotion | |
| **Actor(s)** | Customer | |
| **Description** | The tickets will be discounted during the promotion period, and the tickets will be reduced from 20% to 30%. | |
| **Trigger event** | Customer proceeds to open a promotion. | |
| **Trigger type** | x Extenal | Temporal (Timing) |
| **Precondition** | When the promotion is opened successfully. | |
| **Step performed** | Information per steps.  1. Customers click on "Promotions" in the system to view promotions for themselves.  2. Customers click the "Get" button to receive the promotion | |
| **Post condition** | The promotional price does not exceed 30%. | |
| **Assumptions** | If there are no promotions, the fare is still the same | |
| **Requirement met** | Admin must login to the system to open promotions. | |
| **Outstanding issues** | No | |
| **Priority** | No | |
| **Risk** | No | |

* 1. **Prototype**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

* 1. **Activity**

**Graphical user interface, application

Description automatically generated**

* 1. **Sequence**

**Diagram

Description automatically generated**

* 1. **Database**

A picture containing timeline

Description automatically generated

**3.6 Class diagram**

**Graphical user interface, application

Description automatically generated**

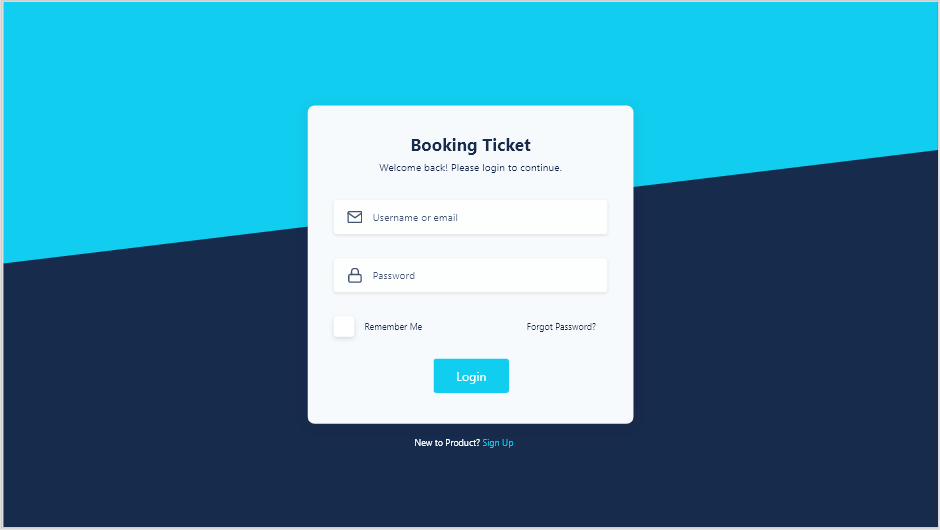
1. **Login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Login | Lam Ha Yen |  |  |  |

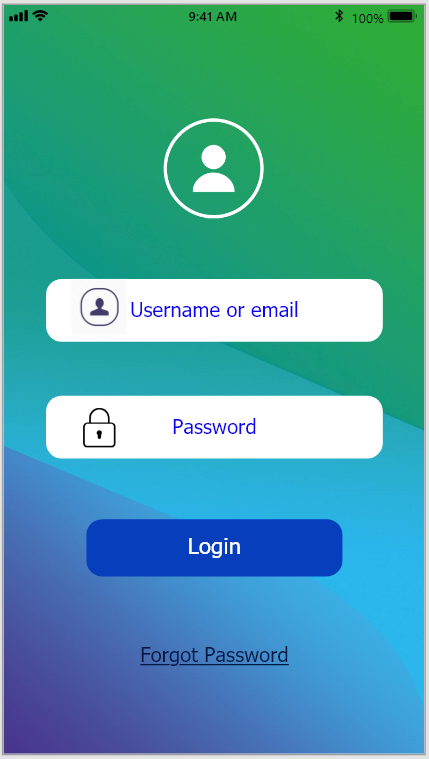
* 1. **Use case Login.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Login ID: UC\_001 | |
| **Actor(s)** | | Customer, Employee. Admin | |
| **Description** | | * The login function helps customer and admin employee log into the application to use services from the application | |
| **Trigger event** | | * Customer, employee, admin want to login to the app | |
| **Trigger type** | | x Extenal | Temporal (Timing) |
| **Pre-condition** | | * Already created a user account. * User account has permissions. * The user's device is already connected to the internet when logging in. | |
| **Step performed** | | | Information per step. |
| **1** | Enter the name registered earlier. | |  |
| **2** | Enter the password. | |  |
| **3** | Click button login | | The system displays the home page screen. |
| **Post condition** | | * The system records successful logging in Activity Log. | |
| **Assumptions** | | * If the user enters the wrong account or password, repeat the steps on the login section | |
| **Requirement met** | | * In order to log in, users need to register for an account successfully. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | |  | |

* 1. **Prototype Login**

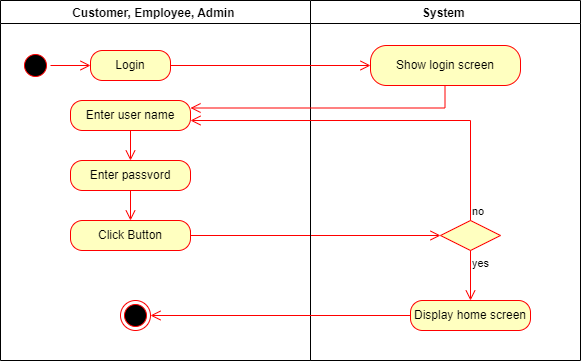
****

**Picture 1: Login of employee and admin.**

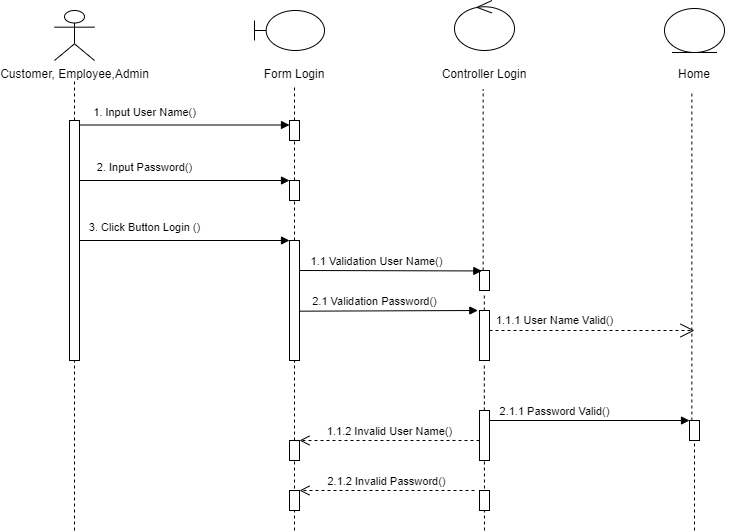
****

**Picture 1: Login of customer**

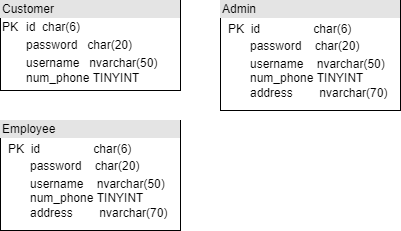
* 1. **Activity:**

****

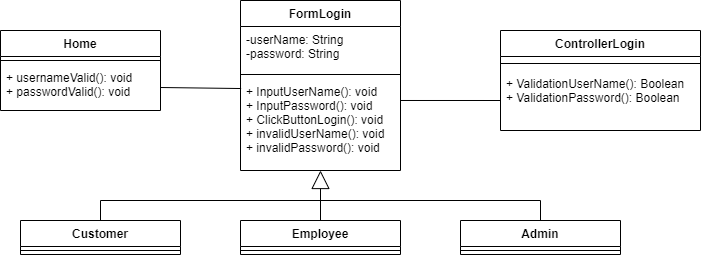
* 1. **Sequence:**

****

* 1. **Database:**

****

* 1. **Class diagram:**

****

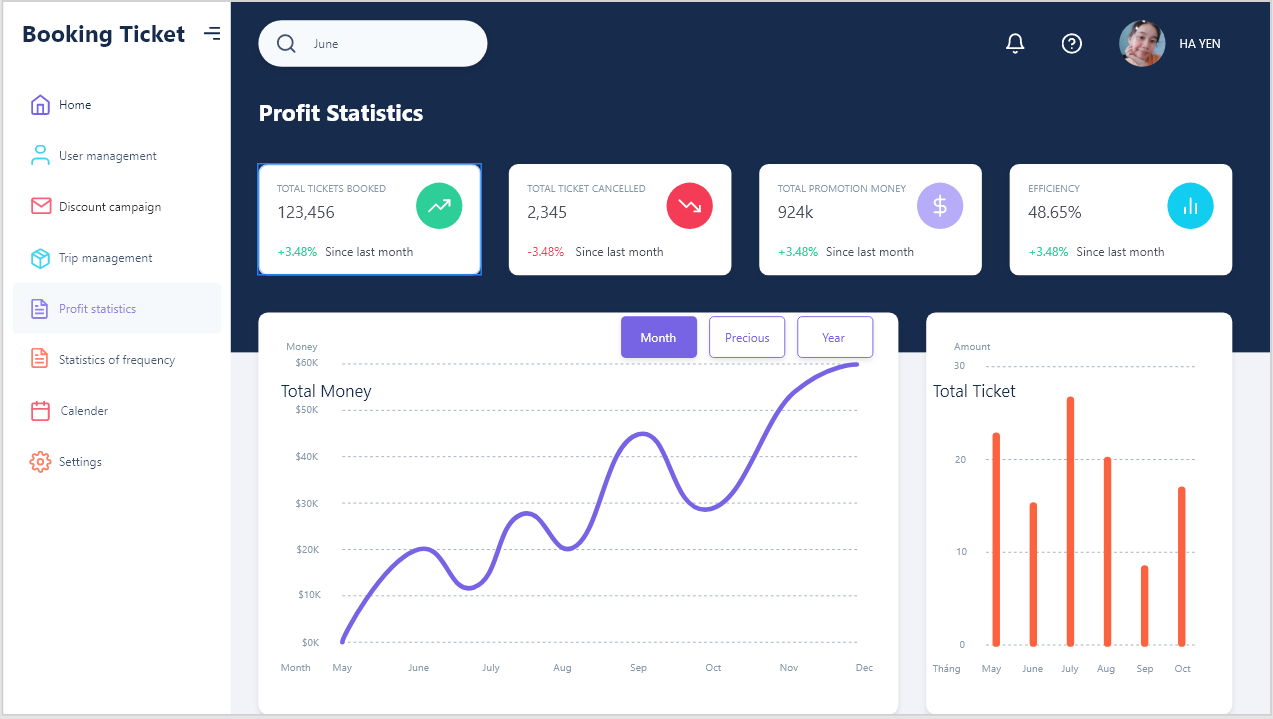
1. **Profit statistics**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Scenario - Profit statistics | Lam Ha Yen |  |  |  |

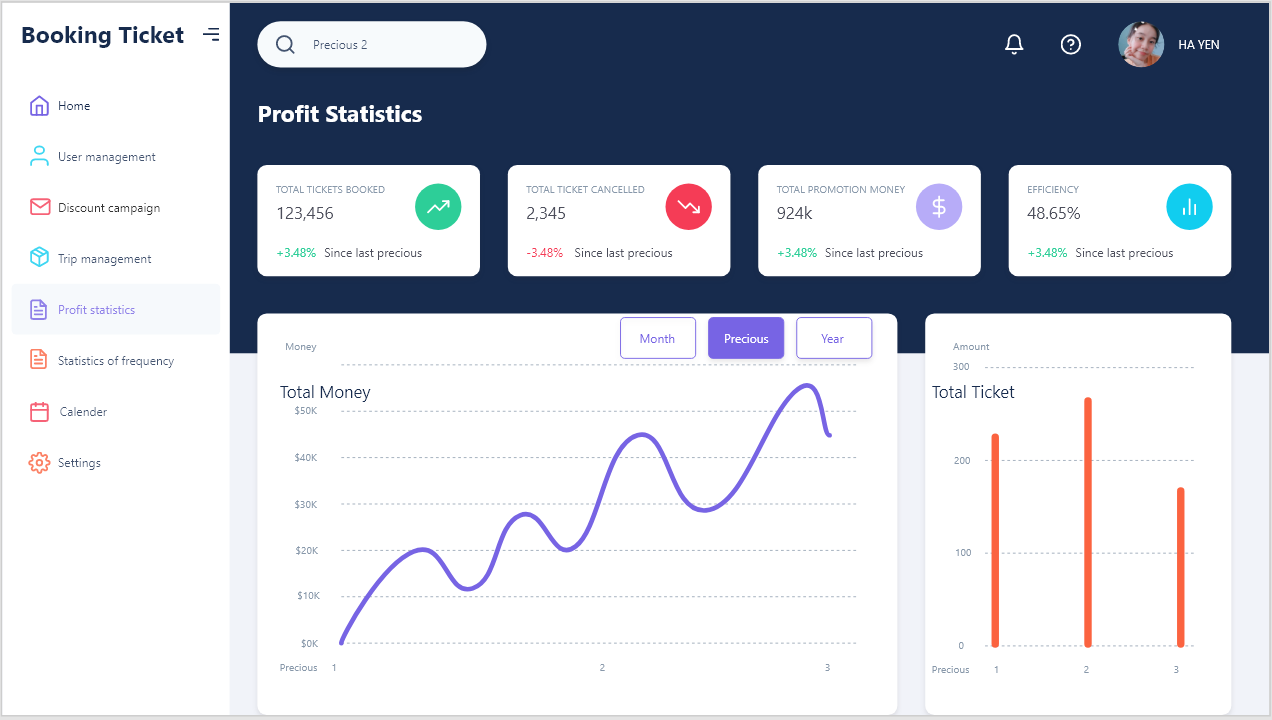
* 1. **Use case:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Profit statistics ID: UC\_002 | |
| **Actor(s)** | | Employee, Admin. | |
| **Description** | | * Profit statistics function makes it easy for the garage to manage revenue over a predetermined time. | |
| **Trigger event** | | * Admin, employee conducts profit statistics. | |
| **Trigger type** | | x Extenal | Temporal (Timing) |
| **Precondition** | | * Admin, employee successfully logged in. | |
| **Step performed** | | | Information per step. |
| **1** | Click on the Profit statistics function. | | * The system displays a Profit statistics screen. |
| **2** | Admin, employee chooses the time to make profit statistics | | * The system displays the results according to the corresponding time |
| **Post condition** | | * Complete profit statistics. | |
| **Assumptions** | | * If the admin or employee does not choose the time of statistics, the system will not be able to make statistics. so it is necessary to choose the time before statistics | |
| **Requirement met** | | * Logged in successfully. * Invoice information exists in the database, with date, month and year, as part of the statistical request. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | |  | |

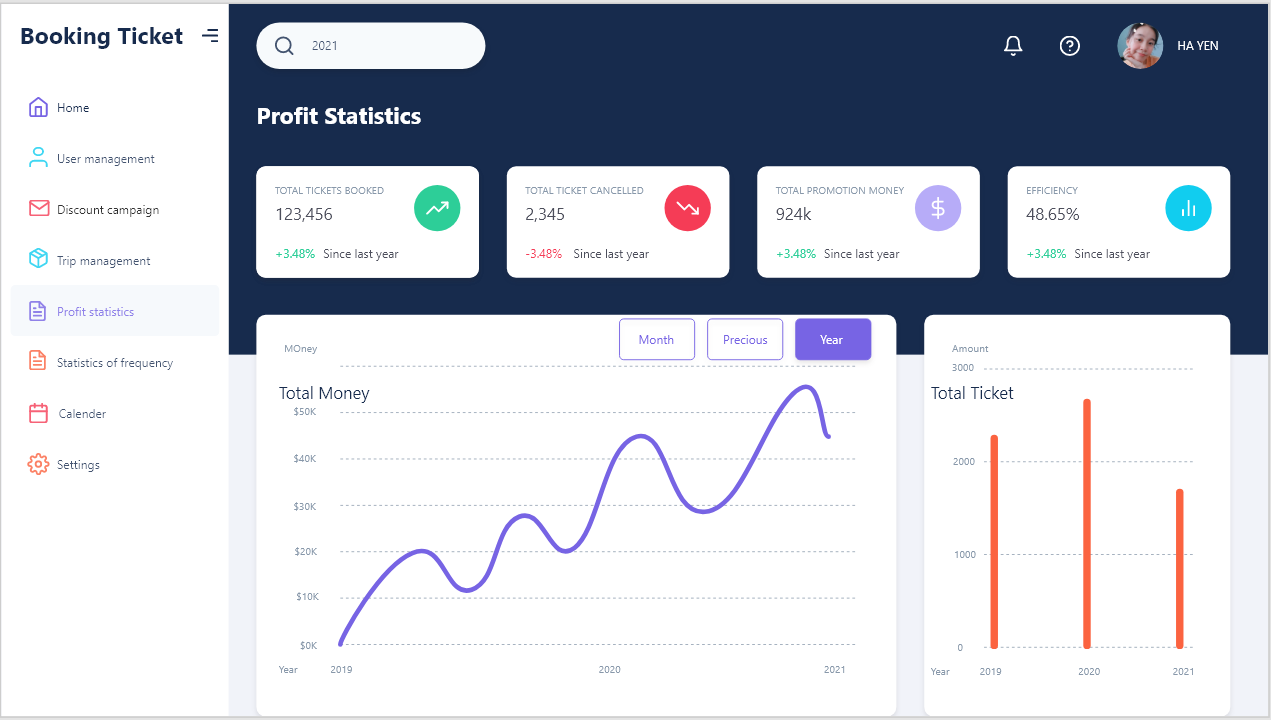
* 1. **Prototype:**



***Picture 1:Profit statistics by month***

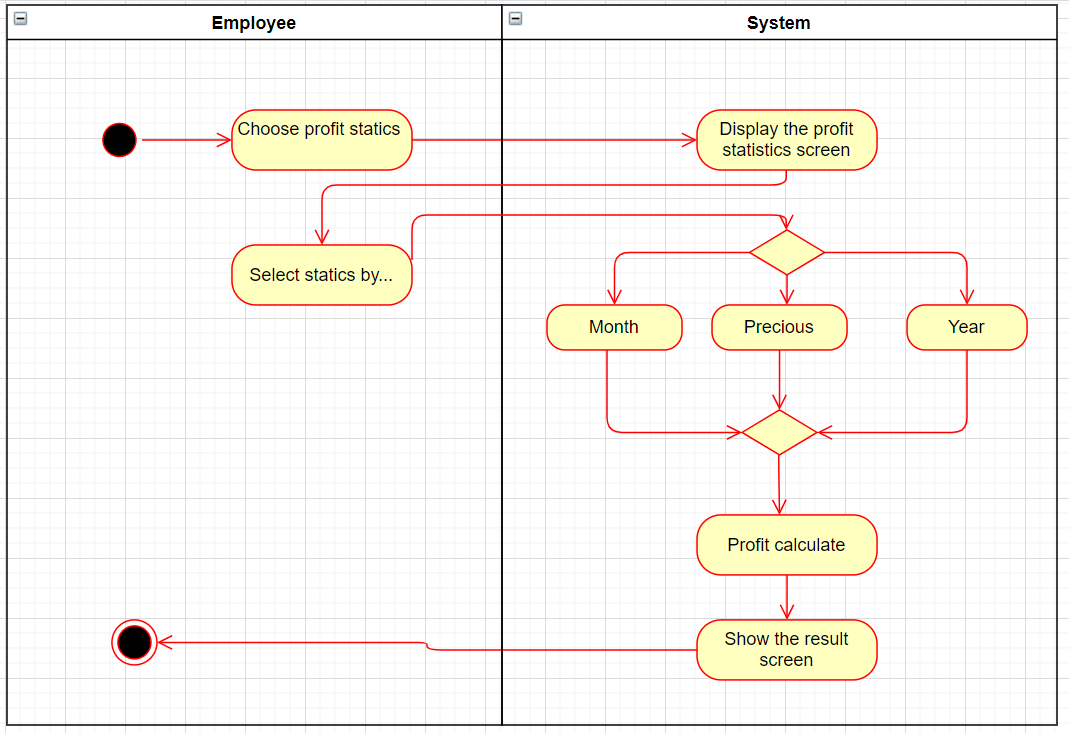


***Picture 2: Profit statistics by precious***

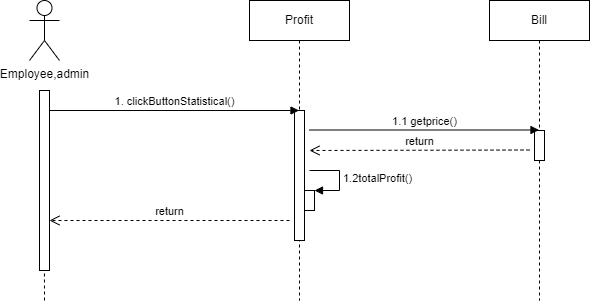


***Picture 3:Profit statistics by year***

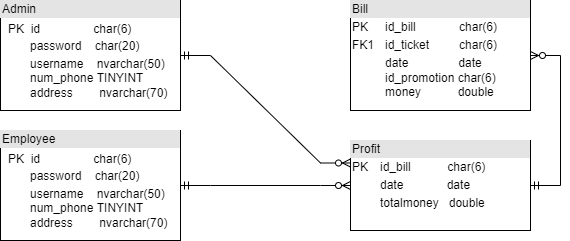
* 1. **Activity:**



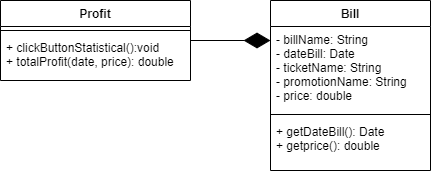
* 1. **Sequence:**

****

* 1. **Database:**

****

* 1. **Class diargam:**

****

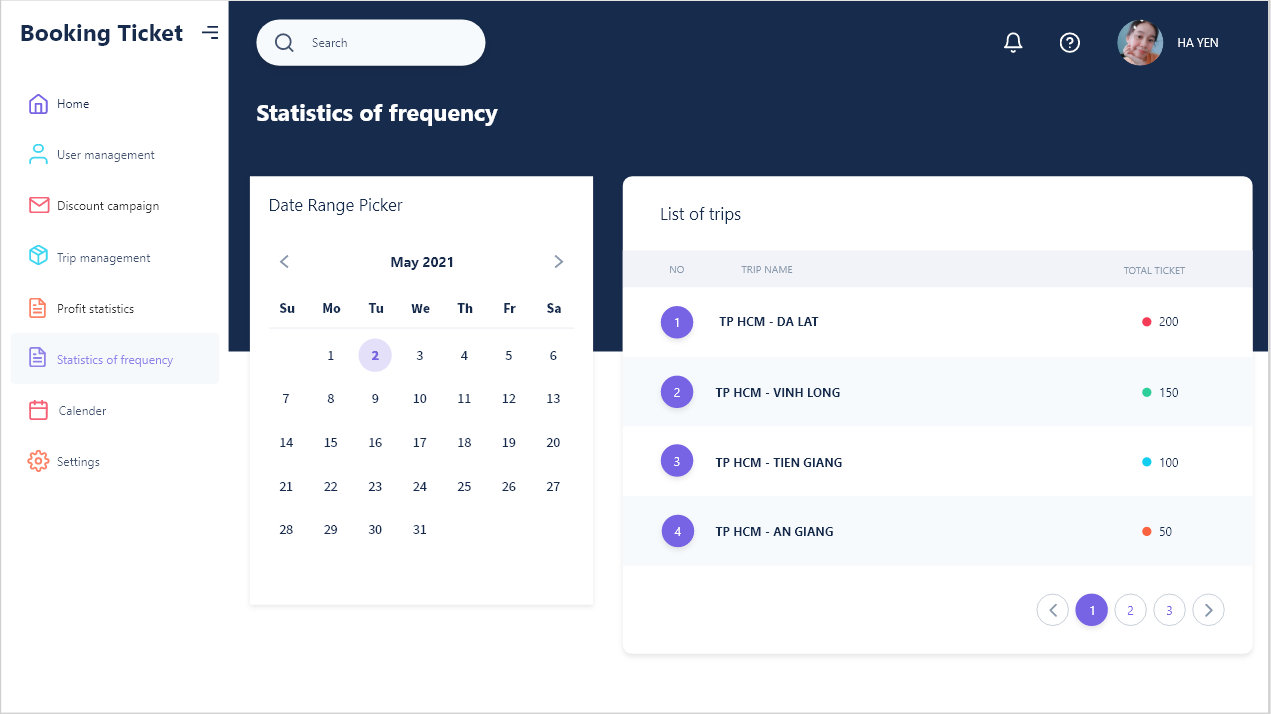
1. **Statistics of frequency.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Statistics of frequency | Lam Ha Yen |  |  |  |

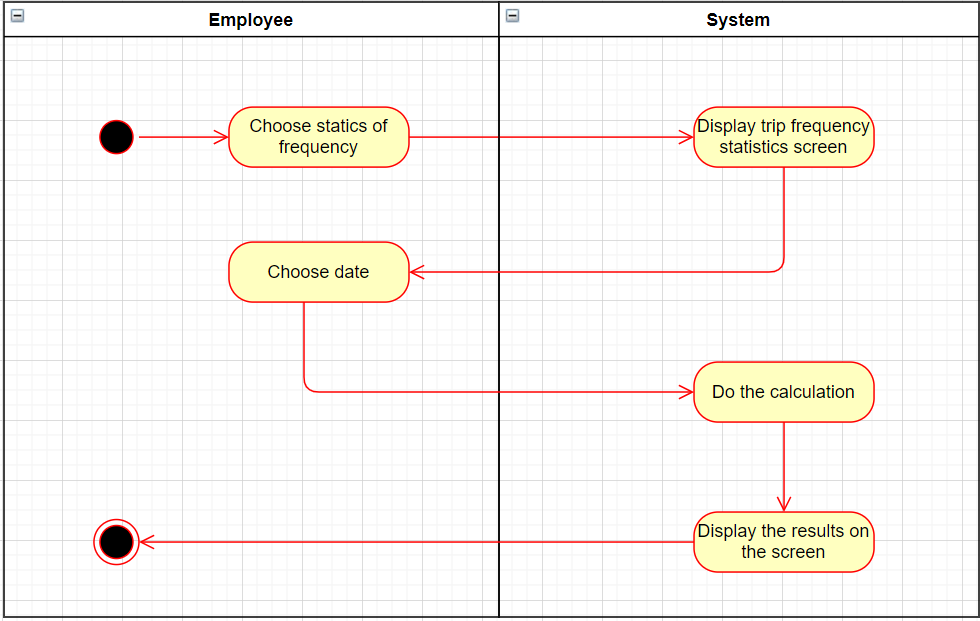
* 1. **Use case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Statistics of frequency ID: UC\_003 | |
| **Actor(s)** | | Employee, Admin. | |
| **Description** | | * The function of booking frequency statistics helps to easily manage passengers who book more or less, so the garage can easily allocate the shuttle reasonably, contributing to increased profits. | |
| **Trigger event** | | * Admin, Employee conducts frequency statistics | |
| **Trigger type** | | x Extenal | Temporal (Timing) |
| **Precondition** | | * Admin, Employee successfully logged in. | |
| **Step performed** | | | Information per step. |
| **1** | Click the frequency statistics function. | | * The system displays a frequency statistics screen. |
| **2** | Choose date statistics | | * Display the result on the screen. |
| **Post condition** | | * Complete frequency statistics. | |
| **Assumptions** | | * Without the customer booking tickets they will not be able to do statistics. | |
| **Requirement met** | | * Login in successful. * There must be successful booking information in the database for the system to perform more to less booked trip statistics. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | |  | |

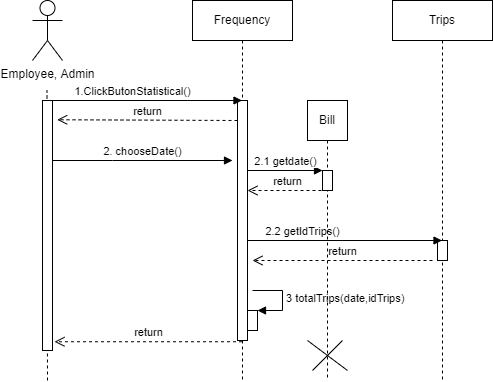
* 1. **Prototype:**

****

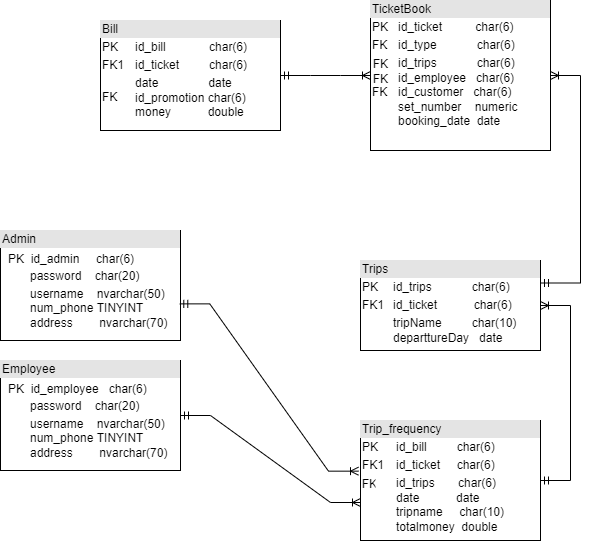
* 1. **Activity:**

****

* 1. **Sequence:**

****

* 1. **Database:**

****

* 1. **Class diargam:**

****

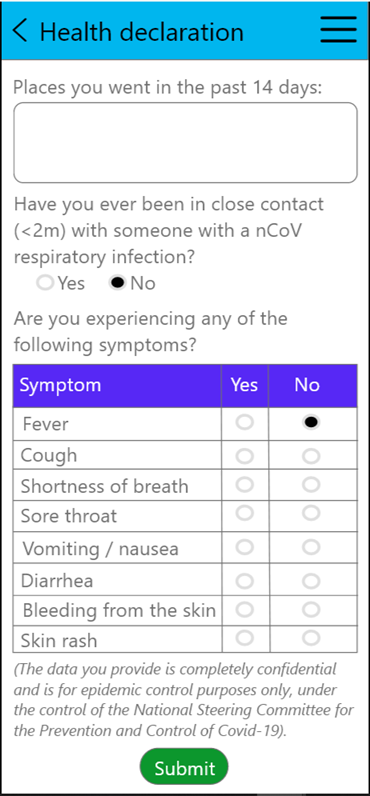
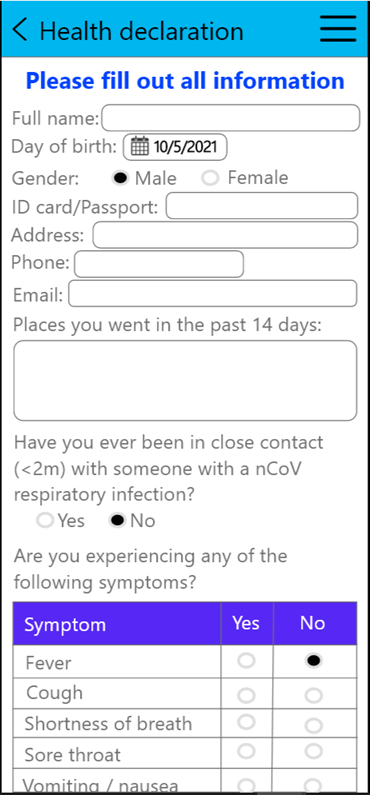
1. **Health Declaration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Health declaration | Nguyen Hoang Vinh |  |  |  |

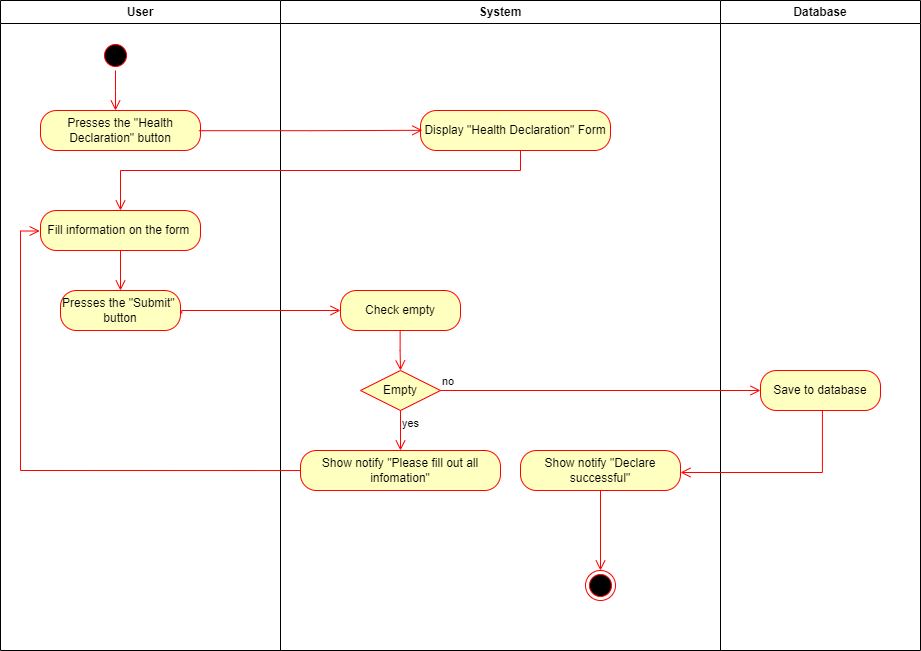
* 1. **Use case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Health declaration | |
| **Actor(s)** | | User | |
| **Description** | | Users must fill in medical declaration information prior to booking payment confirmation. | |
| **Trigger event** | | The user presses the button "Medical Declaration". | |
| **Trigger type** | |  | |
| **Precondition** | | * User successfully logged into the system. And complete the booking process. * Make sure the device is connected to the Internet. | |
| **Actor actions** | | | **System actions** |
| **1** | The user presses the "Health declaration" button. | |  |
| **2** |  | | The system displays the form with the following information:     -User name     -Year of Birth     -Gender     -Passport number / ID card number     -Address     -Phone number     -Email     -The places that the user went through the past 14 days.     -Are signs of Covid-related illness appearing?     -Have you been in contact with someone with nCoV-related respiratory infections?     -Confirm that the above declaration is true. |
| **3** | The user fill information on the form and then presses the Submit button. | |  |
| **4** |  | | The system saves the declaration form in the database. |
| **5** |  | | The system displays the message Declare successful. |
| **Post condition** | | Successful medical report. The declaration form will be saved in the database. | |
| **Assumptions** | |  | |
| **Requirement met** | | Users must fill out all the information in the form. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | | The system cannot verify whether  the declared information is correct or not. | |

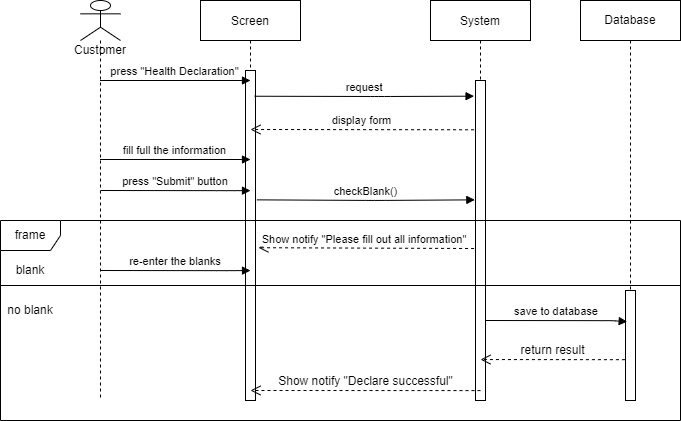
* 1. **Protype**



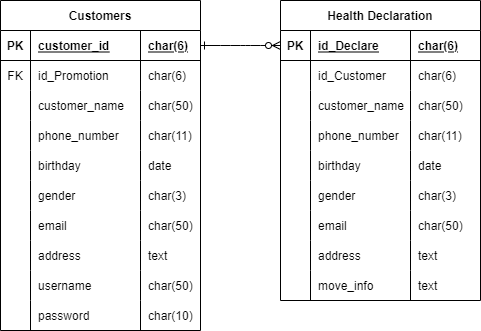
* 1. **Activity diagram**



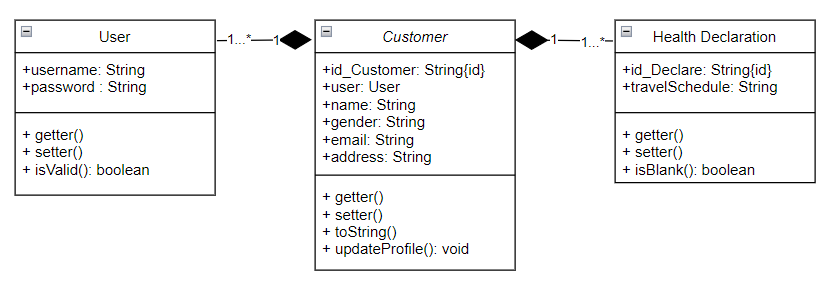
* 1. **Sequence**

****

* 1. **Database**



* 1. **Class Diagram**

****

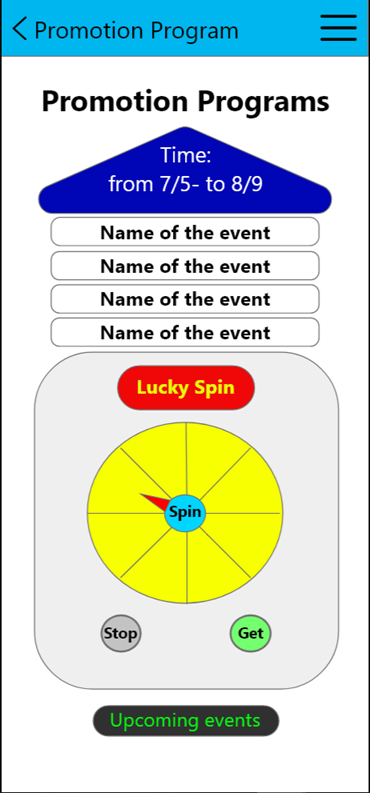
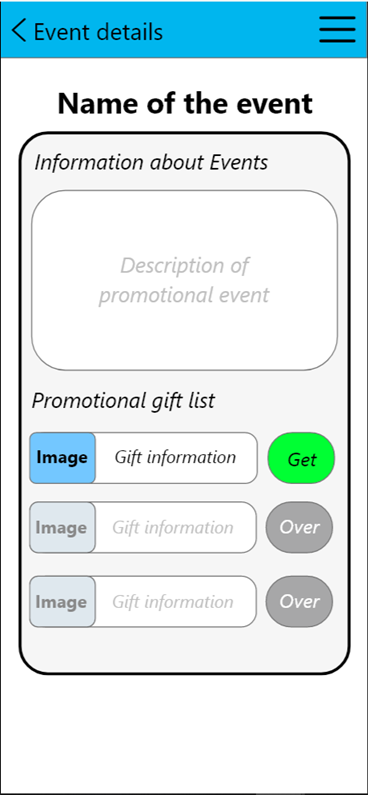
1. **Promotion Program**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Promotion program | Nguyen Hoang Vinh |  |  |  |

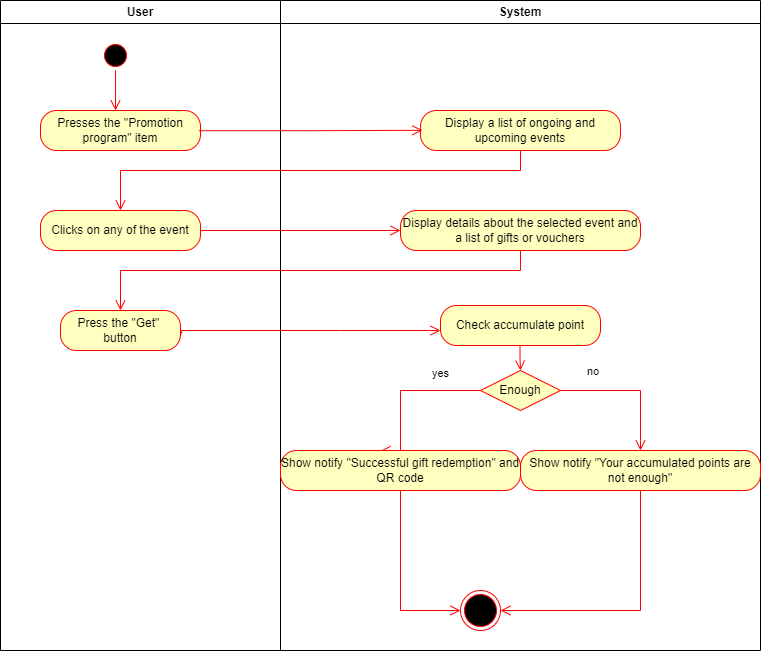
* 1. **Use case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Promotion program | |
| **Actor(s)** | | User | |
| **Description** | | Users want to see all the ongoing deals and upcoming deals. Users want to use their points to redeem discount tickets or receive gifts. | |
| **Trigger event** | | Users click on the “Promotion program” item in the main screen. | |
| **Trigger type** | |  | |
| **Precondition** | | -User successfully logged into the system.  - Make sure the device is connected to the Internet. | |
|  | | | |
| **Actor actions** | | | **System actions** |
| **1** | User selects the Promotion item. | |  |
| **2** |  | | Display a list of ongoing and upcoming events. |
| **3** | The user clicks on any of the promotions. | |  |
| **4** |  | | The system will display detailed information about the selected program:  - Name of the program  - Time  - List of gifts |
| **5** | The user presses the Get button. | |  |
| **6** |  | | The system will display a Gift code so that users can use it when they book tickets. |
|  | | | |
| **Post condition** | | Users receive a QR code to redeem gifts from accumulated points. | |
| **Assumptions** | |  | |
| **Requirement met** | | User must have the required number of points to redeem the gift. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | |  | |

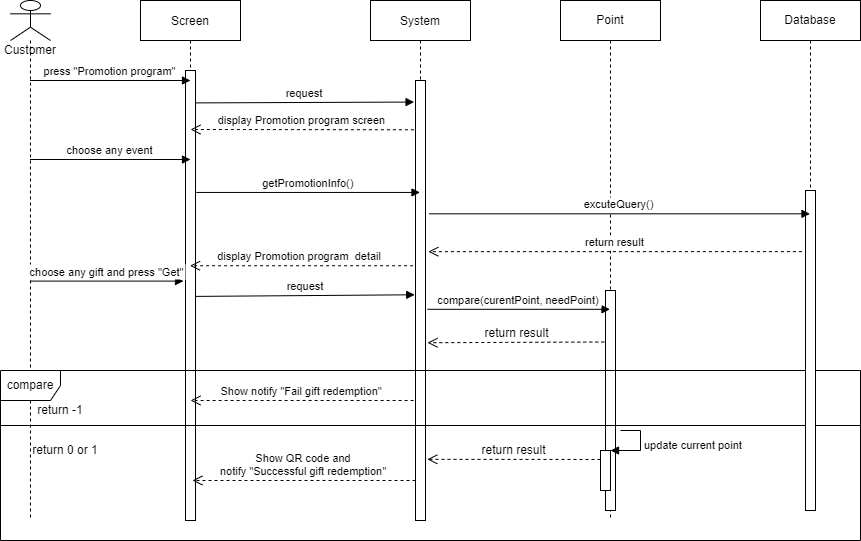
* 1. **Prototype**



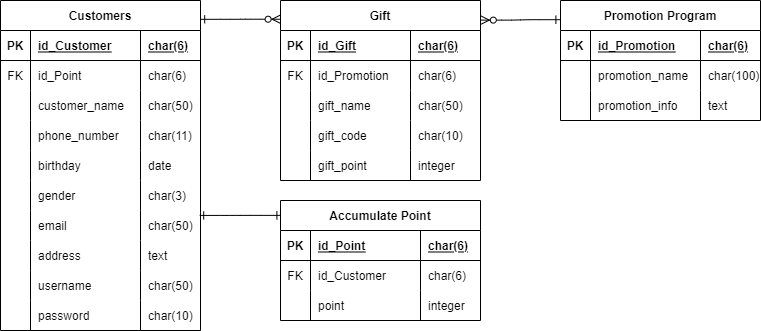
* 1. **Activity**



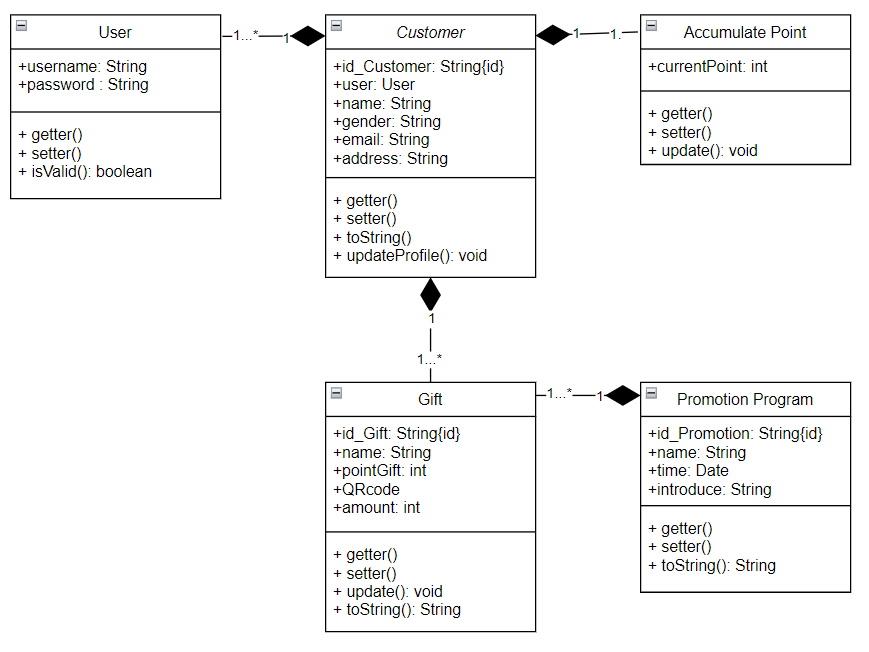
* 1. **Sequence**

****

* 1. **Database**

****

* 1. **Class Diagram**

****

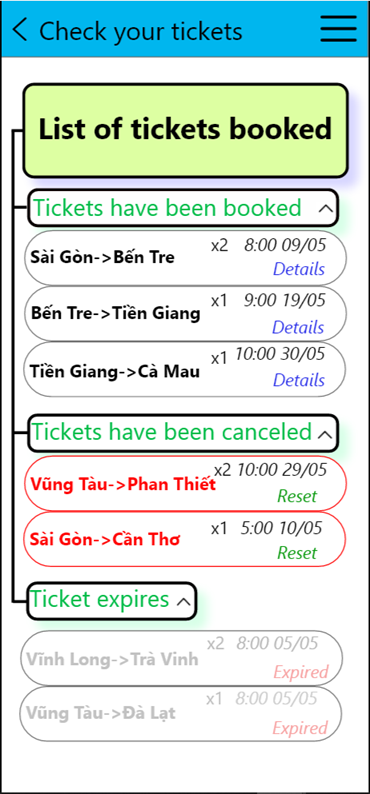
1. **Check ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Check ticket | Nguyen Hoang Vinh |  |  |  |

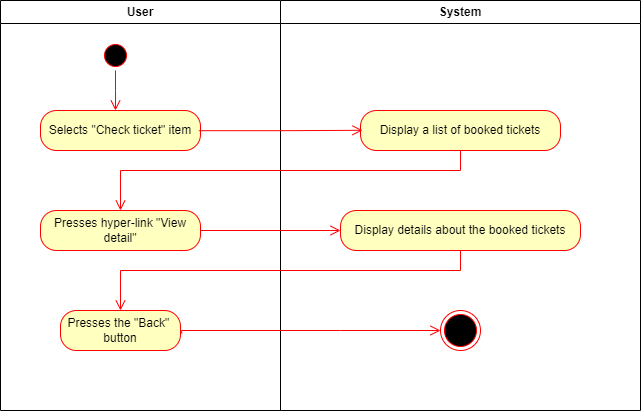
* 1. **Use case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Check ticket | |
| **Actor(s)** | | User | |
| **Description** | | Users want to check the information of the tickets booked. | |
| **Trigger event** | | User selects “Check ticket” on the main screen. | |
| **Trigger type** | |  | |
| **Precondition** | | -User successfully logged into the system.  - Make sure the device is connected to the Internet. | |
| Actor actions | | | System actions |
| **1** | User selects the “Check ticket” item. | |  |
| **2** |  | | The system displays a list of booked tickets with the following information: Ticket code, number of tickets, price of money. And the View Details button at the end of the line. |
| **3** | The user presses the View details. | |  |
| **4** |  | | The system will display detailed trip information: Ticket number, ticket\_id, price, vehicle number, date of departure\_time of departure, point of departure\_ destination, seat\_number. |
|  |  | |  |
| **Post condition** | | The user will see the booking information and can cancel the ticket within the specified time. | |
| **Assumptions** | |  | |
| **Requirement met** | | User has booked ticket to view tickets. | |
| **Outstanding issues** | |  | |
| **Priority** | |  | |
| **Risk** | |  | |

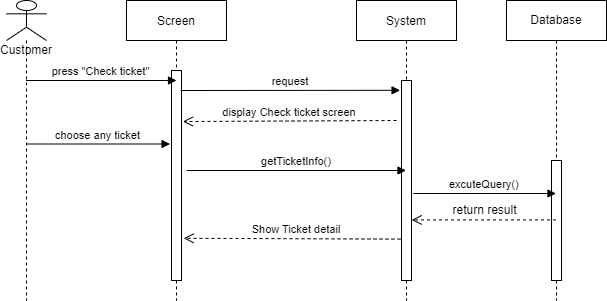
* 1. **Prototype**



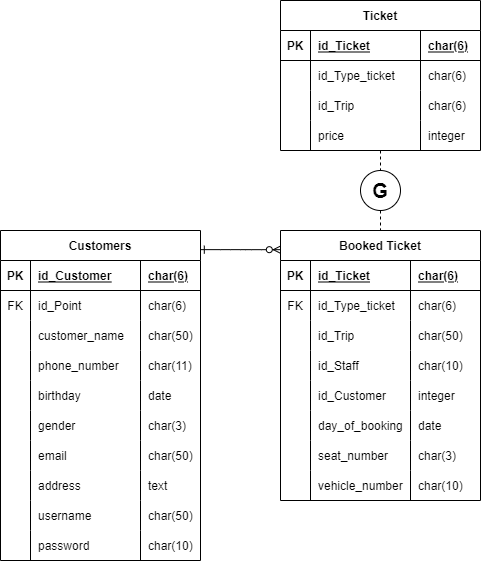
* 1. **Activity**



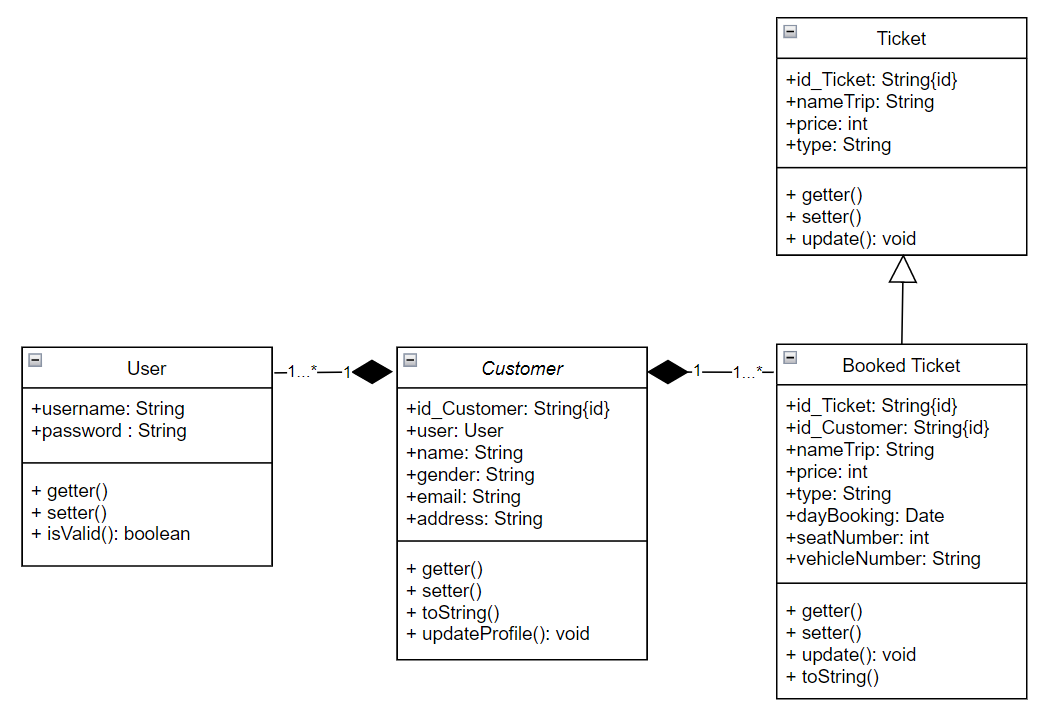
* 1. **Sequence**

****

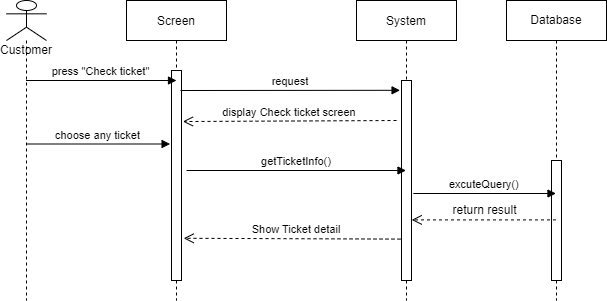
* 1. **Database**

****

* 1. **Class Diagram**

****

**9.5. Sequence**



1. **Use case scenario - Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Use case scenario - Payment | Nguyen Tran Minh Hieu |  |  |  |
|

* 1. **Use case**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case name** | | Payment | | ID | | | UC-05 |
| **Actor(s)** | | Customer, Staff at the ticket counter | | | | | |
| **Description:** | | Payment use case allows for users and systems to produce payments and checks. | | | | | |
| **Trigger event:** | | Customer click the “Checkout” button on the last screen of the booking ticket function  Customer click the checkout button at ticket cart | | | | | |
| **Trigger type:** | | x | External | |  | Temporal (Timing) | |
| **Precondition:** | | Customers need to have an account in the system and maintain login status and have selected tickets  To pay online at an online payment gateway, customers need a bank account  Customer need to produce payment process in 10 minutes after choosing ticket. | | | | | |
| **Step performed:** | | | | Information per steps | | | |
| 1 | The system gets information about the tickets to be processed | | | The system gets all the information of ticket which customer chose. | | | |
| 2 | The system create the invoices | | | The system creates the invoice for customer  The system adds the tickets into the invoice. | | | |
| 3 | The system to calculate the amount to be paid | | | The system will calculate the total amount to be paid by the customer including ticket tax and passenger insurance | | | |
|  |  | | | The system displays the invoice screens. | | | |
| 4 | Customers have to enter necessary information for each ticket.  (such as name of the ticket holder, email, phone, identity card number.) | | | The system will check the validity of data | | | |
| 5 | The system displays a dialog for the customer to confirm that the information provided is completely correct. | | | A dialog has content: “Please, confirm your information is completely correct. You may not be able to get on the bus if the ticket is incorrect.” | | | |
| 6 | Customer check information of the person who booked the ticket again. | | | The system will check the validity of the data if the customer corrects it | | | |
| 7 | Customer choose the method of payment | | | If the customer chooses to pay by cash, the customer must pay the fare at the ticket counter within 24 hours of booking and go through **step 9**, otherwise the ticket will be resold after the time is up. | | | |
|  |  | | | If the customer chooses to pay online,the customer needs to choose the appropriate method and go to **step 8** | | | |
| 8 | Customers need to enter correct information and pay **at the online payment gateway** to complete the booking. | | |  | | | |
| 9 | The system will display result | | | If the customer pays by cash, the system will output the message "Successfully booked, waiting for payment" | | | |
|  |  | | | If the customer pays online: | | | |
|  |  | | | If there is no error in the checkout process, the message "Successfully booked" will be output. | | | |
|  |  | | | If there is an error during checkout, the message "Error has occurred, please check again" | | | |
| 10 | The system saves invoice information for revenue statistics if the payment is successful. | | |  | | | |
| **Post condition** | | Invoices are made, with invoice codes to look up booking information | | | | | |
| **Assumptions** | | Whenever the customer logs out of the system or exits the browser or loses network connection, the payment is canceled. | | | | | |
| **Requirement met** | |  | | | | | |
| **Outstanding issues** | | If the customer account is not enough, the payment will fail | | | | | |
| **Priority** | |  | | | | | |
| **Risk** | |  | | | | | |

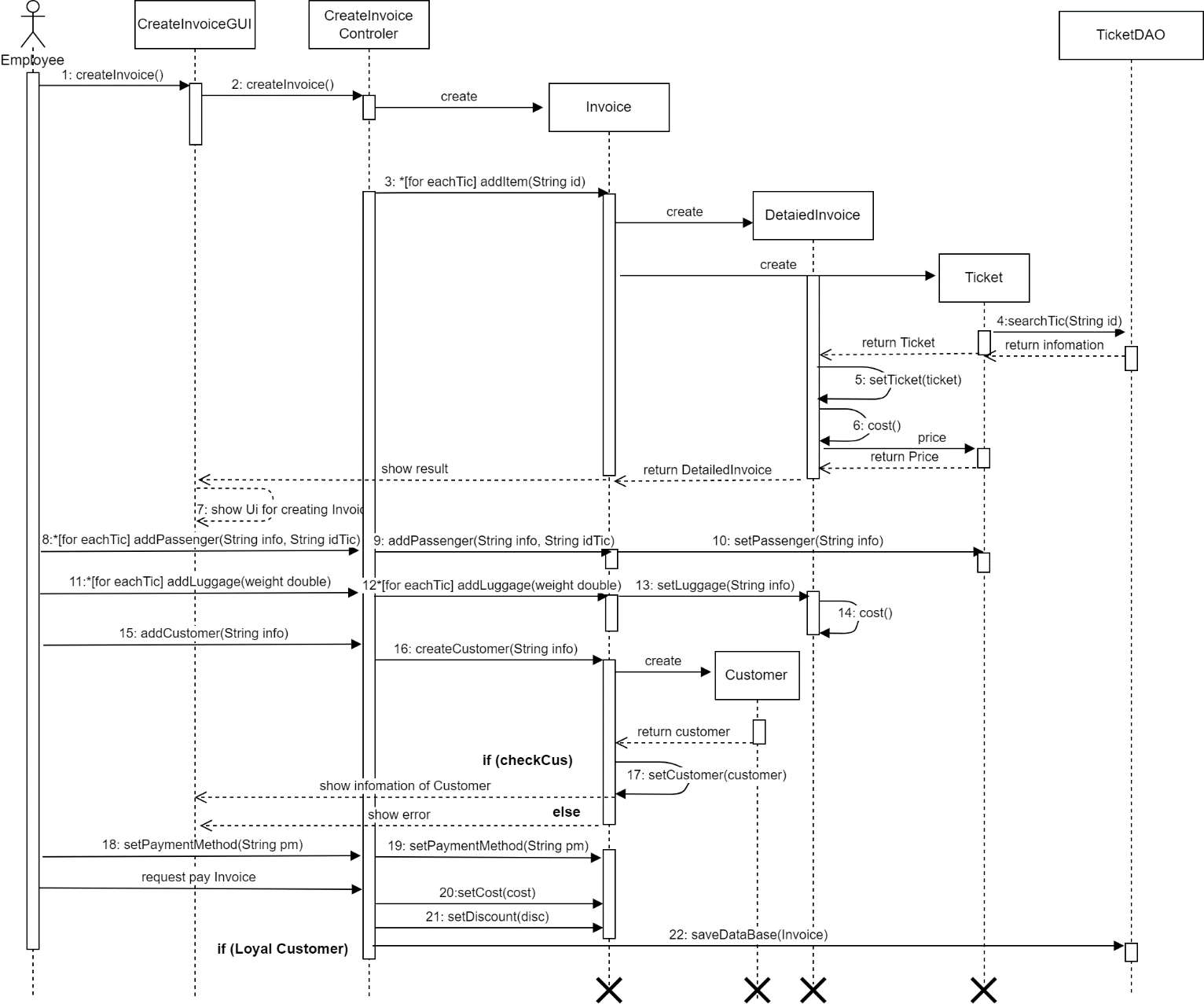
* 1. **Prototype**

|  |  |
| --- | --- |
| C:\Users\84981\Desktop\ticketAgain\paymentEng3.jpg  Hình 1: Nhập thông tin hành khách vào hóa đơn -1 | C:\Users\84981\Desktop\ticketAgain\paymentEng2.jpg  Hình 2: Nhập thông tin hành khách vào hóa đơn-2 |
| C:\Users\84981\Desktop\ticketAgain\paymentEng1.jpg  Hình 3: Lựa chọn hình thức thanh toán | C:\Users\84981\Desktop\ticketAgain\4.png  Hình 4: Kết quả thanh toán trả sau |

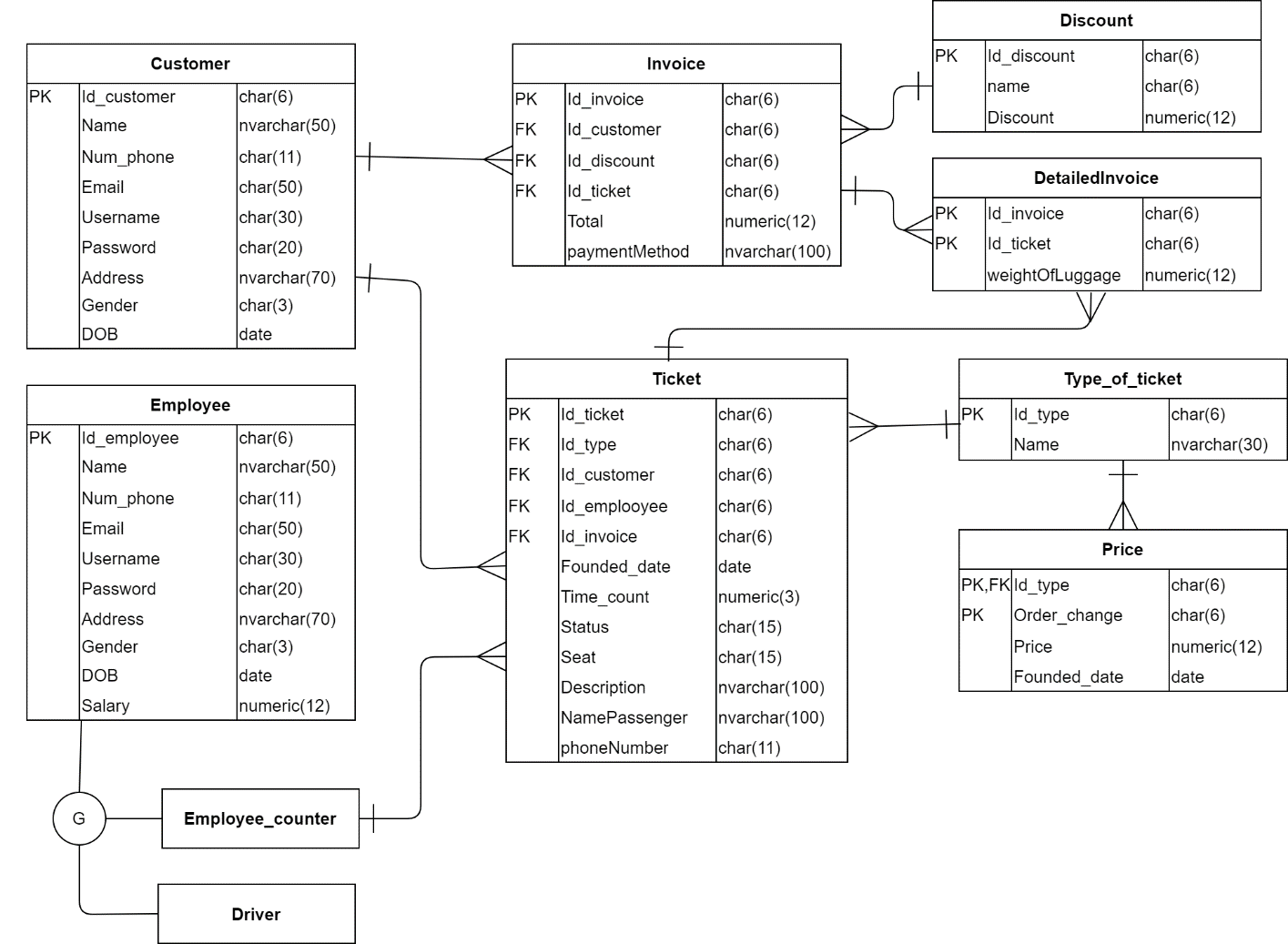
* 1. **Activity**



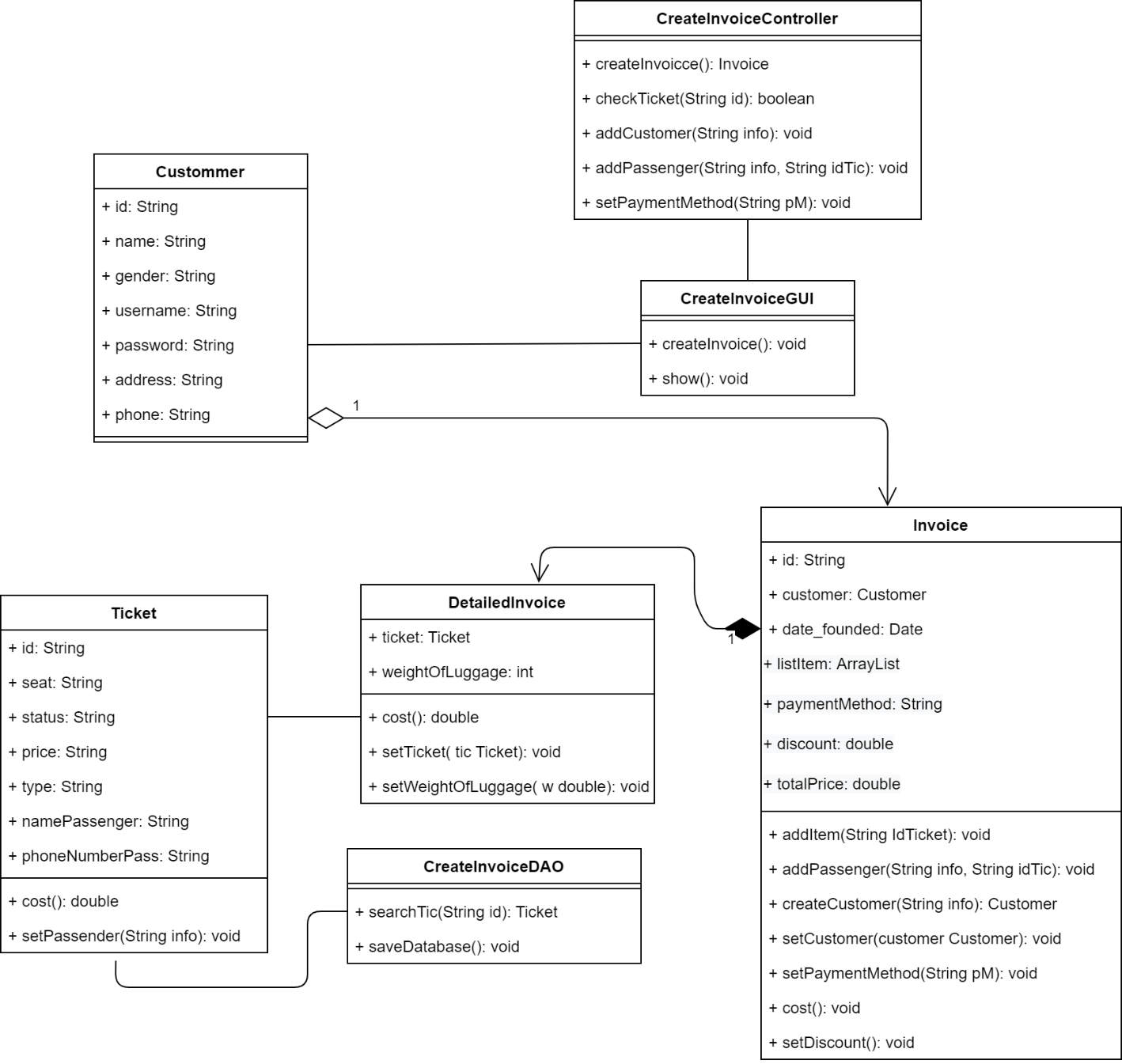
* 1. **Sequence**



* 1. **Database**



* 1. **Class Diagram**



1. **Employee Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Use case scenario - Payment | Nguyen Tran Minh Hieu |  |  |  |
|

* 1. **Use case**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case name** | | Employee Management | | | | ID | UC-001 |
| **Actor(s)** | | Administrator | | | | | |
| **Description:** | | Employee Management use case allows admin to add, edit, delete. | | | | | |
| **Trigger event:** | | Administrator click on “User Management” Items on the menu bar and choose “Employee” | | | | | |
| **Trigger type:** | | x | External | |  | Temporal (Timing) | |
| **Precondition:** | | Administrator need to have an account in the system with admin role and maintain login status | | | | | |
| **Step performed:** | | | | | | Information per steps | |
| To manage an administrator user account, there are four functions as follows: add, edit, delete employee accounts. | | | | | | | |
| **Additional functions:** | | | | | | | |
| 1 | Administrator selects the “ADD” button on the menu bar | | | The system displays a form that allows the administrator to enter new employee information | | | |
| 2 | Administrator enter employee information in the form | | | The system check validate of data | | | |
| 3 | Administrator press the “CREATE” button to add the user | | | The system will get information from the form entered by the administrator and save it in the database | | | |
| Editing functions: | | | | | | | |
| 1 | **C:\Users\84981\Desktop\imageP1\Web 1920 – 5.png**Administrator selects an employee account to edit  (by click on the pencil at the end of the record containing the username) | | | The system displays a form that allows the administrator to correct employee information | | | |
| 2 | Administrator correct employee information by entering a new value in the form | | | The system check validate of data | | | |
| 3 | Administrator click the “UPDATE” button to complete the employee information editing process | | | The system will get information from the form entered by the administrator and save it in the database | | | |
| Deleting functions: | | | | | | | |
| 1 | Administrator selects an account to delete | | |  | | | |
| 2 | **C:\Users\84981\Desktop\imageP1\Web 1920 – 5.png**  Administrator click button to delete the user | | | The system will display a message for the administrator to confirm the deletion of the user | | | |
| 3 | Administrator choose Yes or No in the dialog on the screen | | | If the administrator chooses YES, the system will delete the employee information from the database  If the administrator chooses NO, the deletion will be canceled | | | |
| 4 | The system displays the results | | | In step 3 if:  The administrator selects YES. The system will delete employee information in the database and display a “Success” message  The administrator selects NO. The system will cancel the operation and the system displays a “Failed” message | | | |
| **Post condition** | | Administrator can add/ edit/ delete employee accounts successfully | | | | | |
| **Assumptions** | | Whenever the administrator logs out of the system or exits the browser or loses network connection, the management will not be performed. | | | | | |
| **Requirement met** | |  | | | | | |
| **Outstanding issues** | |  | | | | | |
| **Priority** | |  | | | | | |
| **Risk** | | Administrators need to consider carefully before deleting, because it may not be possible to recover data | | | | | |

* 1. **Prototype**

|  |
| --- |
| **C:\Users\84981\Desktop\imageP1\Web 1920 – 5.png**  **Manager Employee** |
| **C:\Users\84981\Desktop\imageP1\Web 1920 – 6.png**  **Add employee** |
| **C:\Users\84981\Desktop\imageP1\Web 1920 – 8.png**  **Edit Infomation Employee** |
| **C:\Users\84981\Desktop\imageP1\Web 1920 – 7.png**  **Delete Employee** |

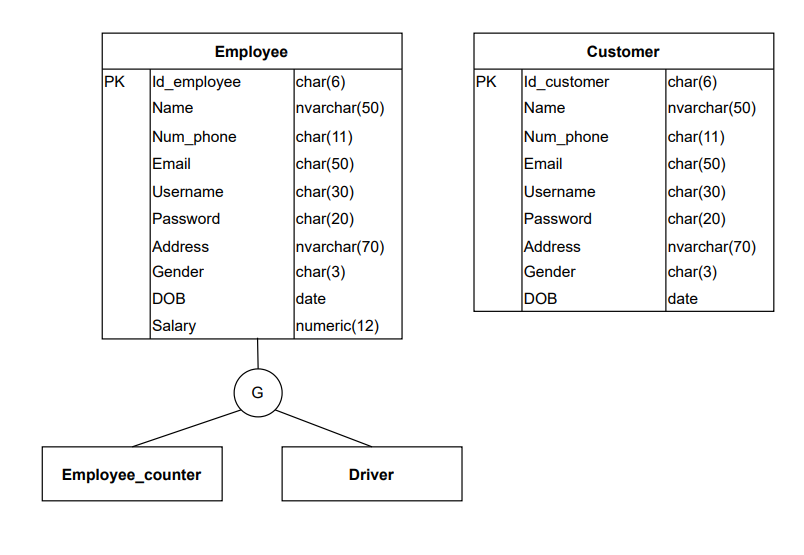
* 1. **Activity**

|  |
| --- |
| C:\Users\84981\Desktop\ticketAgain\actiAdd200.png  **Activity Diagram of Add Employee** |
| C:\Users\84981\Desktop\ticketAgain\acedit200.png  **Activity Diagram of Edit Employee** |
| C:\Users\84981\Desktop\ticketAgain\acDel200.png  **Activity Diagram of Delete Employee** |

* 1. **Sequence**

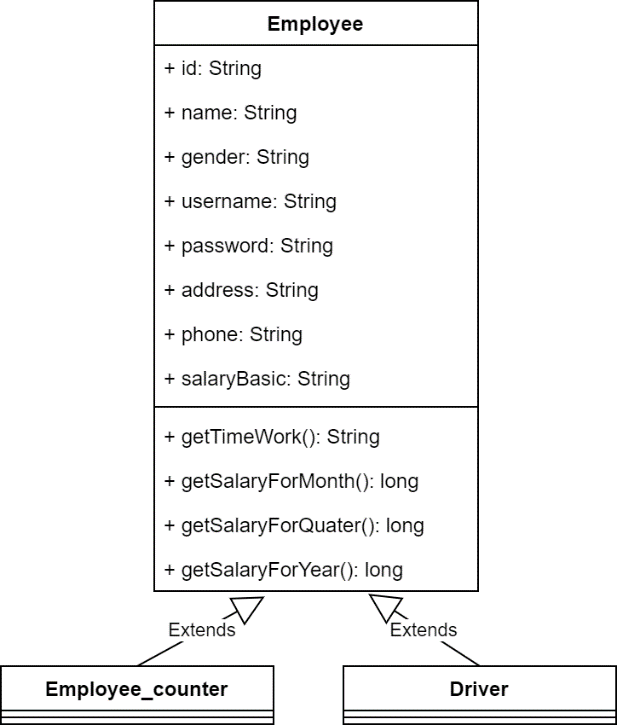
|  |
| --- |
| C:\Users\84981\Desktop\ticketAgain\add200.png  **Sequence Diagram of Add Employee** |
| C:\Users\84981\Desktop\ticketAgain\del200.png  **Sequence Diagram of Edit Employee** |
| C:\Users\84981\Desktop\ticketAgain\del200.png  **Sequence Diagram of Delete Employee** |

* 1. **Database**



**Entity Relationship Diagram**

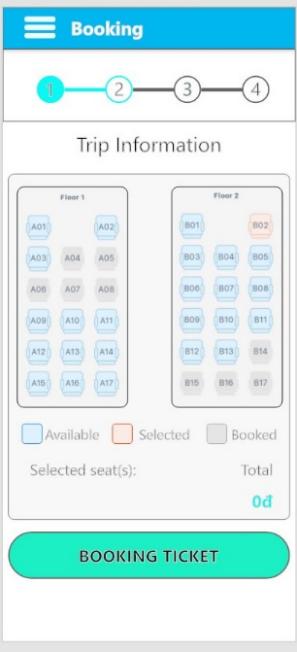
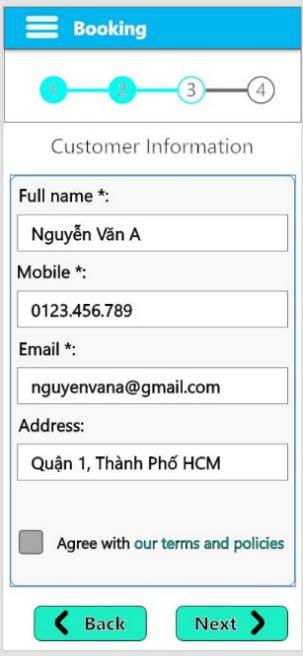
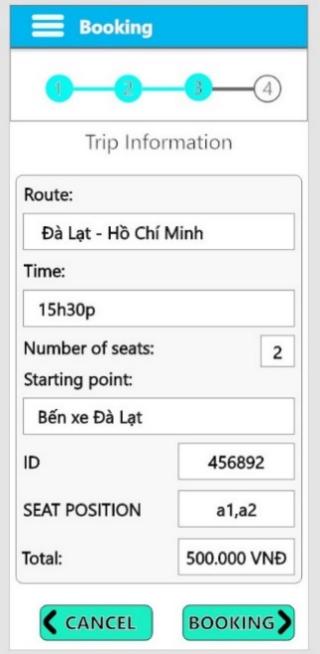
* 1. **Class Diagram**



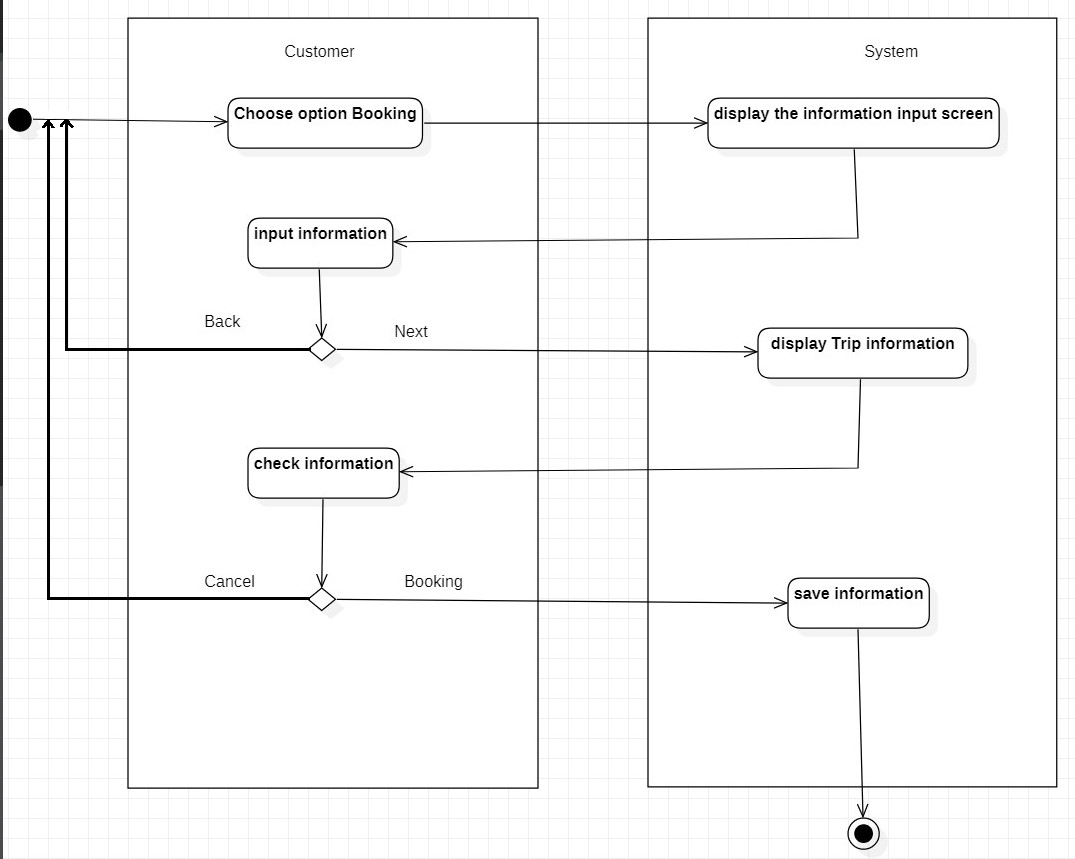
1. **Booking Ticketing**
   1. **Use case Statistics of frequency.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Booking tickets ID: OBS-BookingTickets | |
| **Actor(s)** | | Customer | |
| **Description** | | 1. This use case allows the customer to manipulate the ticket booking function on the system interface | |
| **Trigger event** | | 1. Users choose the option Booking tickets. | |
| **Trigger type** | | x External | Temporal (Timing) |
| **Precondition** | | 1. The main screen is displayed with options for the user to choose, in which there is an option to book tickets. | |
| **Step performed** | | | Information per steps. |
| **1** | The customer selects the booking icon after selecting the seat position. | |  |
| **2** | The device displays the user information panel. | | Information includes full name, contact phone number, email address, current residential address. |
| **3** | Customers fill in personal information according to the request form. | |  |
| **4** | Customer selects the icon below the form. | | If the cancel icon is selected, the user will return to the main screen. |
| If a customer clicks on the next icon, the user will go to the next step. |
| **5** | The screen displays trip information again for customers to check again. | | The information allows itinerary, time, number of seats, departure location, ticket number, seat location, total payment. |
| **6** | Customer selects the icon below the form. | | If the customer chooses an icon cancel, the user will return to the main screen of the application. |
| If the user chooses bt booking, the ticket will be booked and saved in the system |
| **7** | The screen displays the information that has been saved to the customer | |  |
| **Postcondition** | | 1. Tickets are booked if customers choose option Booking in step 6. | |
| **Assumptions** | | 1. At any time, if a customer presses the Cancel button, the operation is canceled and returned to the Main Menu. | |
| **Requirement met** | | 1. Customers need to log in to the system and fill in the information requested by the site in order to find a trip and conduct a booking. | |
| **Outstanding issues** | | 1. When a customer incorrectly enters the travel information resulting in incorrect ticket booking and the customer may lose money. | |
| **Priority** | | 1. No priority case. | |
| **Risk** | | 1. When incorrect tickets are made, the system shall not have to refund money to the customers. | |

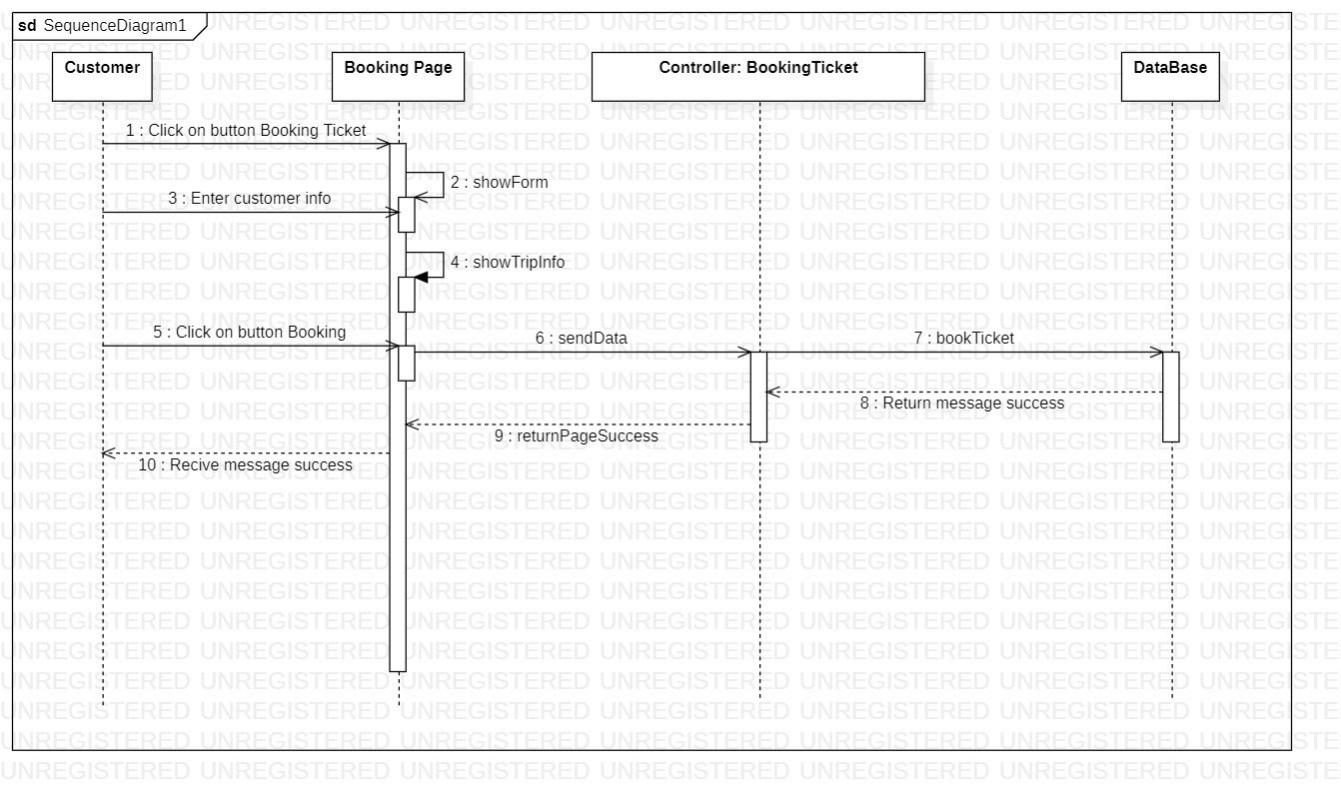
* 1. **Prototype**

* 1. **Activity**



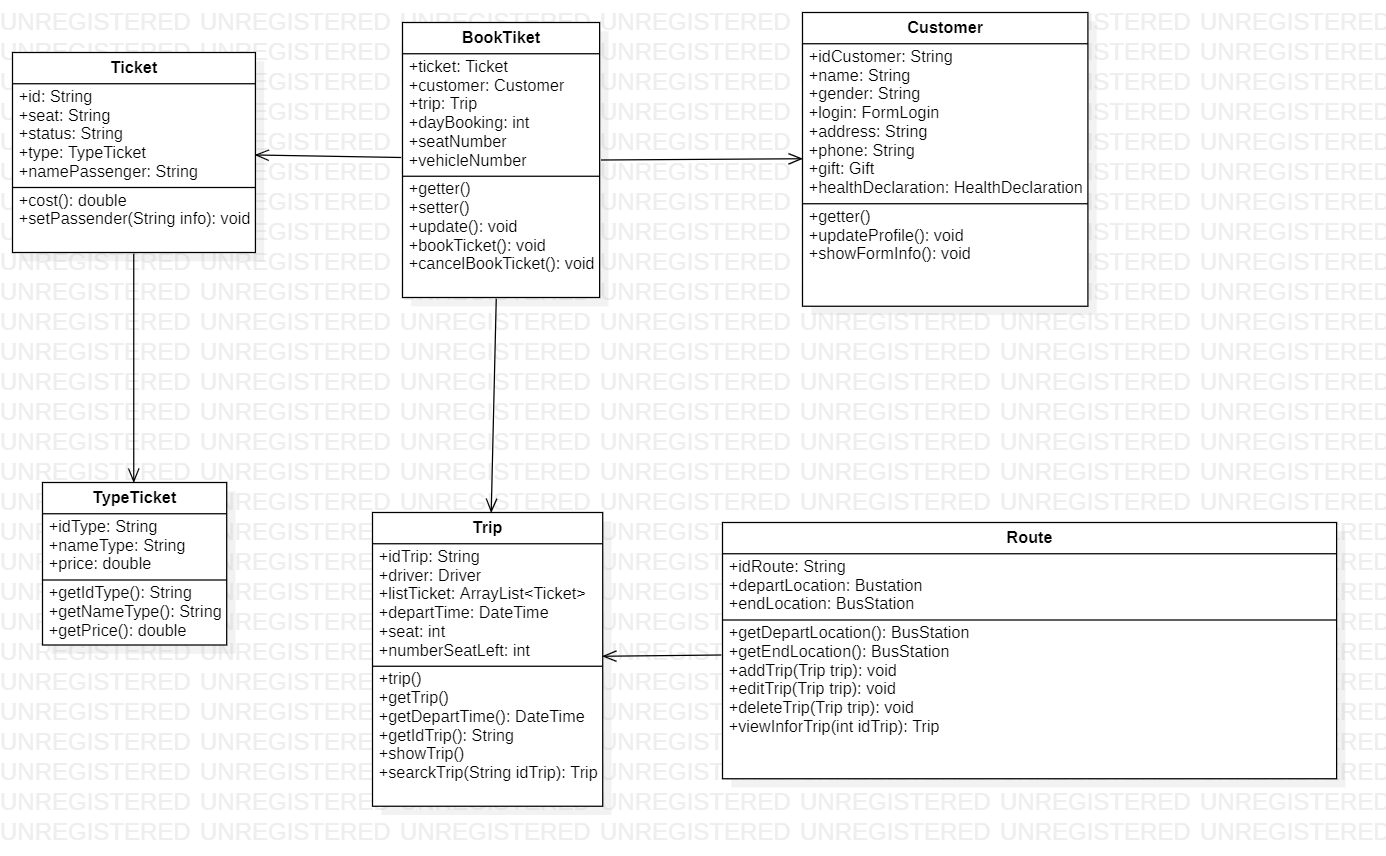
* 1. **Sequence**



* 1. **Database**



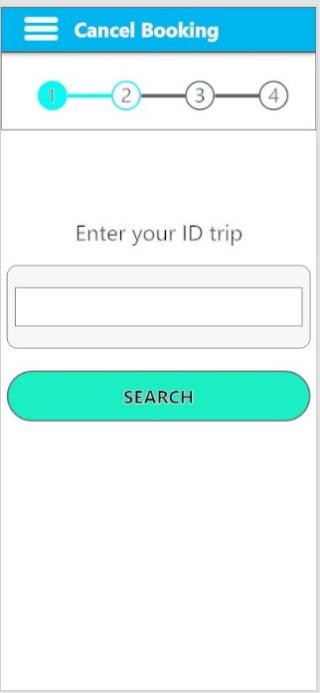
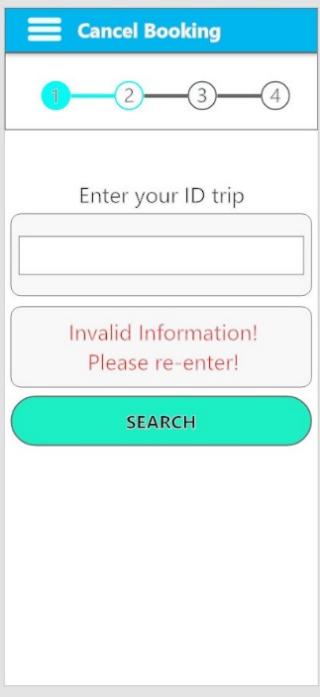
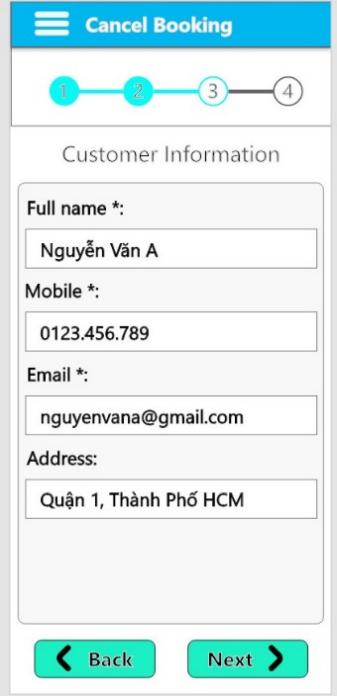
* 1. **Class Diagram**

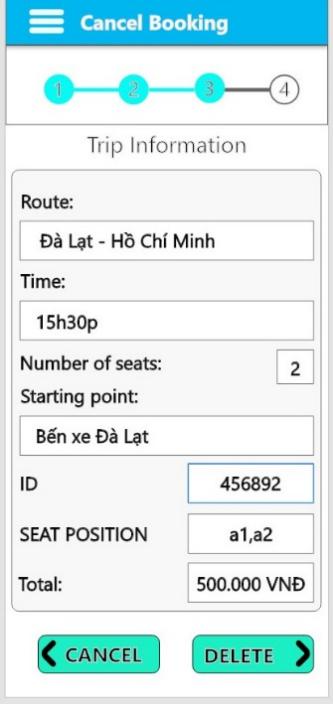


1. **Cancel booking**
   1. **Use case**

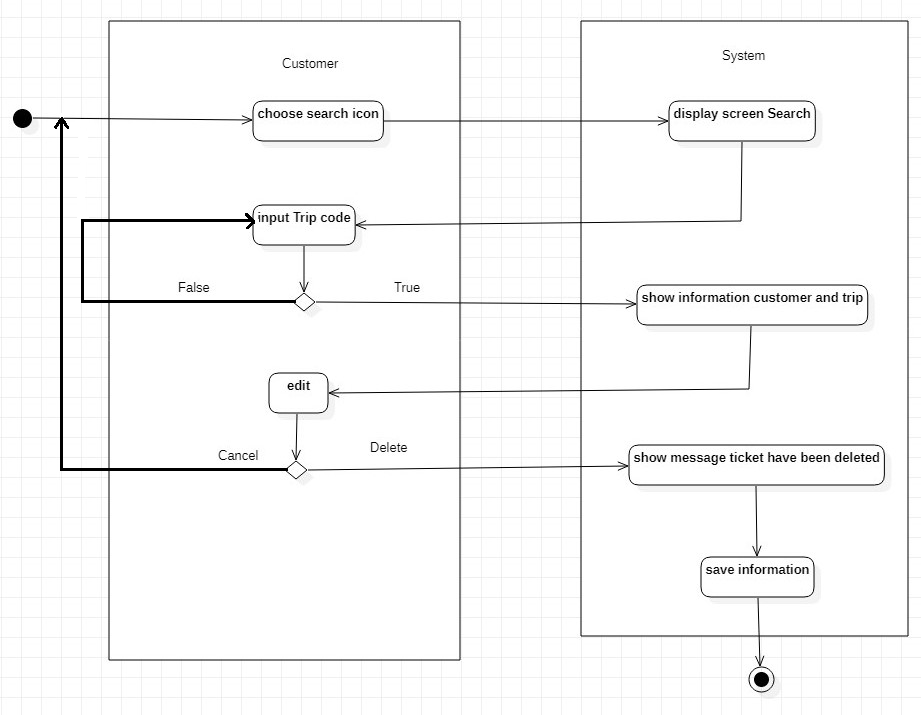
|  |  |  |  |
| --- | --- | --- | --- |
| **Use case name** | | Cancel Booking tickets ID: OBC-CancelBookingTickets | |
| **Actor(s)** | | Customer | |
| **Description** | | * This use case allows customers to manipulate the ticket booking function on the system interface | |
| **Trigger event** | | * Users choose the option Delete tickets in My tickets. | |
| **Trigger type** | | x External | Temporal (Timing) |
| **Precondition** | | * The user has logged in into the system * The main screen is displayed with options for the user to choose, in which there is an option to Delete tickets. | |
| **Step performed** | | | Information per steps. |
| **1** | The device displays the screen requesting customers to input tripcode. | | The tripcode will be sent after the customer confirms the booking and payment include 6 digits. |
| **2** | Customer input 6 digits. | |  |
| **3** | The device displays compare code. If tripcodes are identical, the device displays the screen trip information. | |  |
|  | If tripcodes are not identical, the device requesting the customer inputs the trip code again. | | (Come back to step 2) |
| **4** | The device displays customer and trip information. | | If you select cancel, return main menu. |
| If you select delete, the ticket have been deleted. |
| **5** | The screen displays the information that has been saved to the system. | |  |
| **Postcondition** | | * Tickets are delegated if customer chooses the option to delete in step 4. | |
| **Assumptions** | | * At any time, if customer press button Cancel, the operation is canceled and return to Main Menu | |
| **Requirement met** | | * The customer needs to correctly enter a ticket code so that they can view or cancel booking tickets | |
| **Outstanding issues** | | * When a customer loses a ticket code, it is impossible to look up the information and modify the tickets. | |
| **Priority** | | * No priority case. | |
| **Risk** | | * Customers who lose a confirmation code don't get tickets for any reason. | |

* 1. **Prototype**

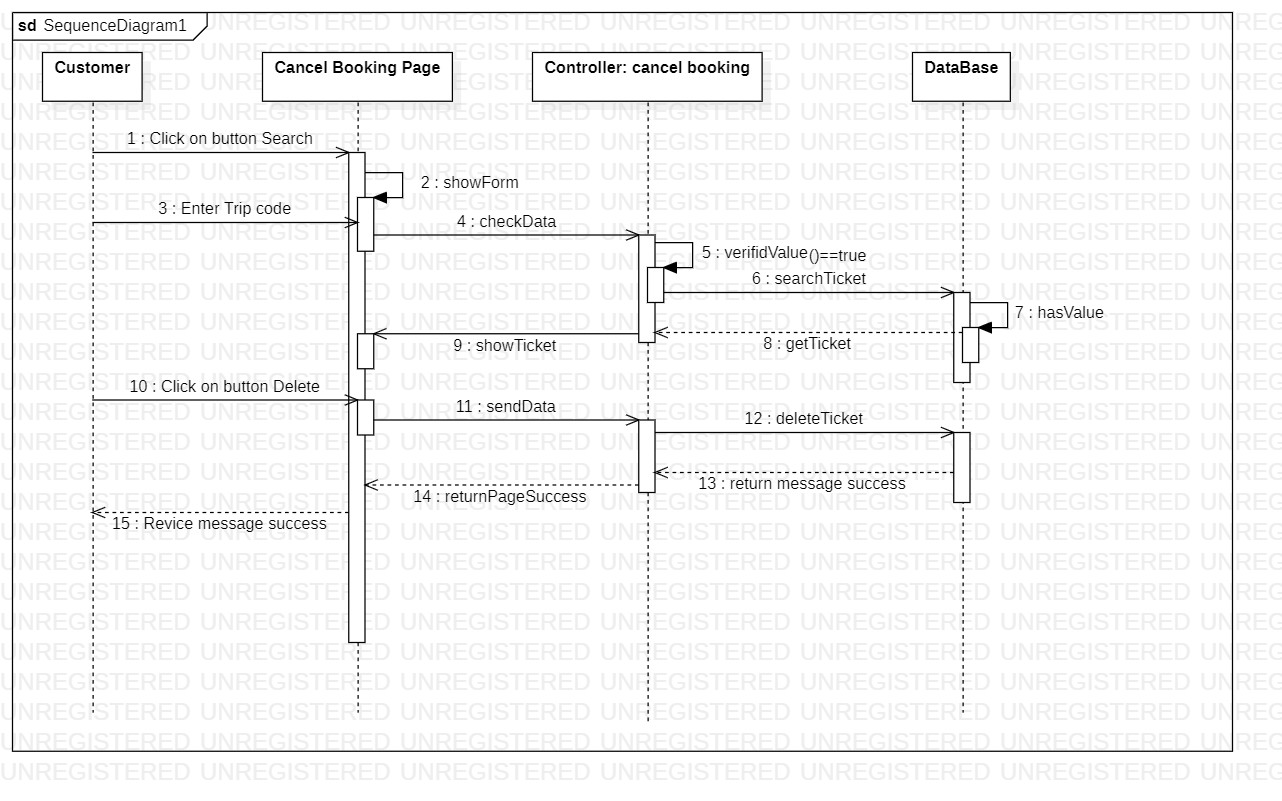
  

* 1. **Activity**



* 1. **Sequence**



* 1. **Database**



* 1. **Class Diagram**



# Manage trips.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Manage trips | Nguyen Ngoc Thanh Xuan |  |  |  |

## Use case

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case name** | | | Manage trips | | | ID | | UC-ManageTrips |
| **Actor(s)** | | | Administrator | | | | | |
| **Description:** | | | Manage trips use case allows administrators to “Add”, “Edit”, “View” and “Delete” trips. | | | | | |
| **Trigger event:** | | | Administrator click on Trip Management Items on the menu bar | | | | | |
| **Trigger type:** | | | x | External |  | Temporal (Timing) | | |
| **Precondition:** | | | Administrator need to have an account in the system and maintain login status  Administrator must choose “Trip Management”. | | | | | |
| **Step performed:** | | | | | | **Information per steps** | | |
| To manage trips, there are four functions as follows: “Add”, “Edit”, “View” and “Delete” trips. | | | | | | | | |
| **Additional functions:** | | | | | | | | |
| 1 | | Admin selects the “ADD” button on the table. | | | | | The system displays a form that allows the administrator to enter new trip information. | |
| 2 | | Admin enters trip information in the form. | | | | |  | |
| 3 | | Admin choose the “Save” button to add the trip into the database or “Cancel” button. | | | | | - If Admin chooses the “Save” button: => The system will get trip information from the form entered by the admin and save it in the database.  - If Admin choose “Cancel” button:  => The system will go back to the trip list screen. | |
| **Editing functions:** | | | | | | | | |
| 1 | | Admin selects a trip that wanna edit by clicking on the pencil at the end of the record containing the trip. | | | | | The system displays a form that allows the admin to correct trip information. | |
| 2 | | The admin corrects the trip information by entering a new value in the form. | | | | |  | |
| 3 | | Admin click the “Save” button to complete the trip information editing process or “Cancel” button. | | | | | - If Admin chooses the “Save” button: => The system will get trip information from the form entered by the admin and save it in the database.  - If Admin choose “Cancel” button:  => The system will go back to the trip list screen. | |
| **Deleting functions:** | | | | | | | | |
| 1 | | Admin selects a trip to delete by clicking on the trash can at the bottom of the record containing the trip. | | | | |  | |
| 2 | | Admin click the delete button to delete the trip. | | | | | The system will display a message for the administrator to confirm the deletion of the trip. | |
| 3 | | Admin proceed to confirm the information by selecting “Yes” or “No”. | | | | | - If the admin chooses YES:  => The system will delete the user information from the database.  - If the admin chooses NO:  => The deletion will be canceled. | |
| **View functions:** | | | | | | | | |
| 1 | Administrator chooses a trip that want view | | | | | | The system will display information about trip for the administrator to view a trip. | |
| 2 | Administrator chooses an OK button. | | | | | |  | |
| **Post condition** | | | Administrator can add, edit, delete, view trips successfully | | | | | |
| **Assumptions** | | | Whenever the administrator logs out of the system or exits the browser or loses network connection, the management will not be performed. | | | | | |
| **Requirement met** | | |  | | | | | |
| **Outstanding issues** | | |  | | | | | |
| **Priority** | | |  | | | | | |
| **Risk** | | | Administrators need to consider carefully before deleting, because it may not be possible to recover data. | | | | | |

## Prototype

### 

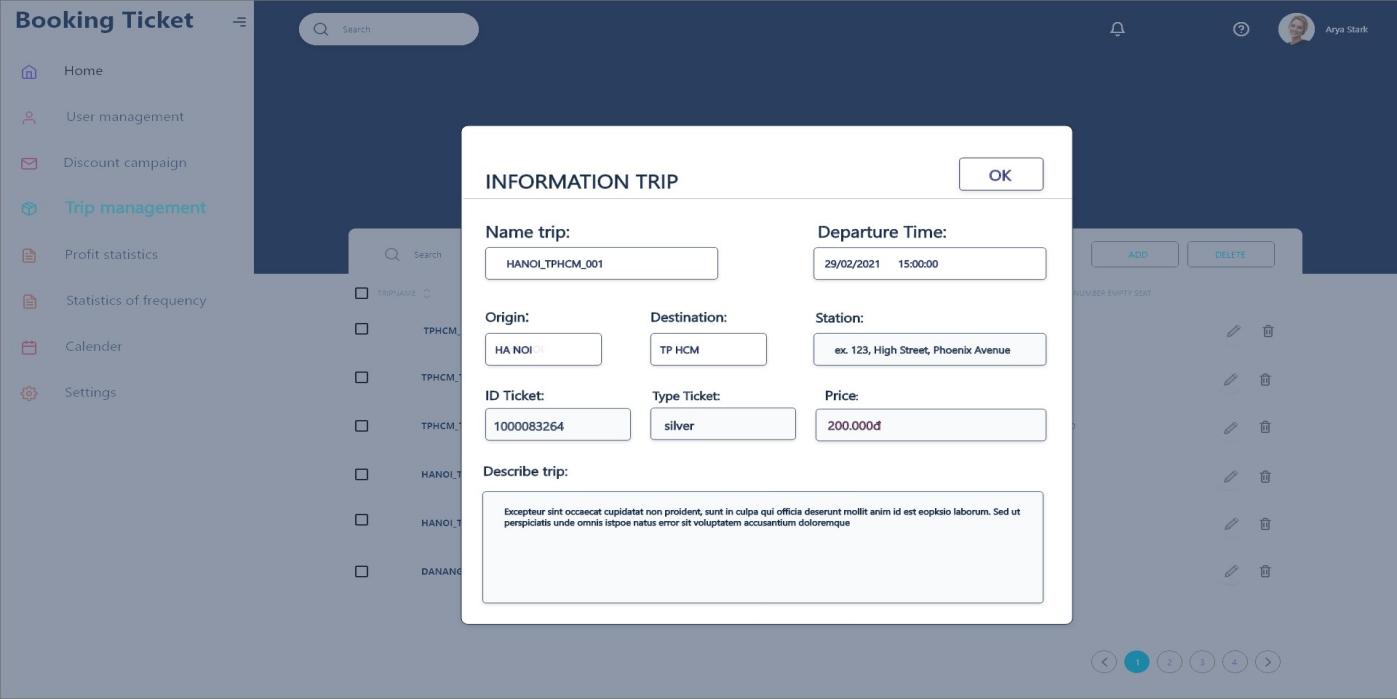
**Add trips**

### 

**Edit trips**

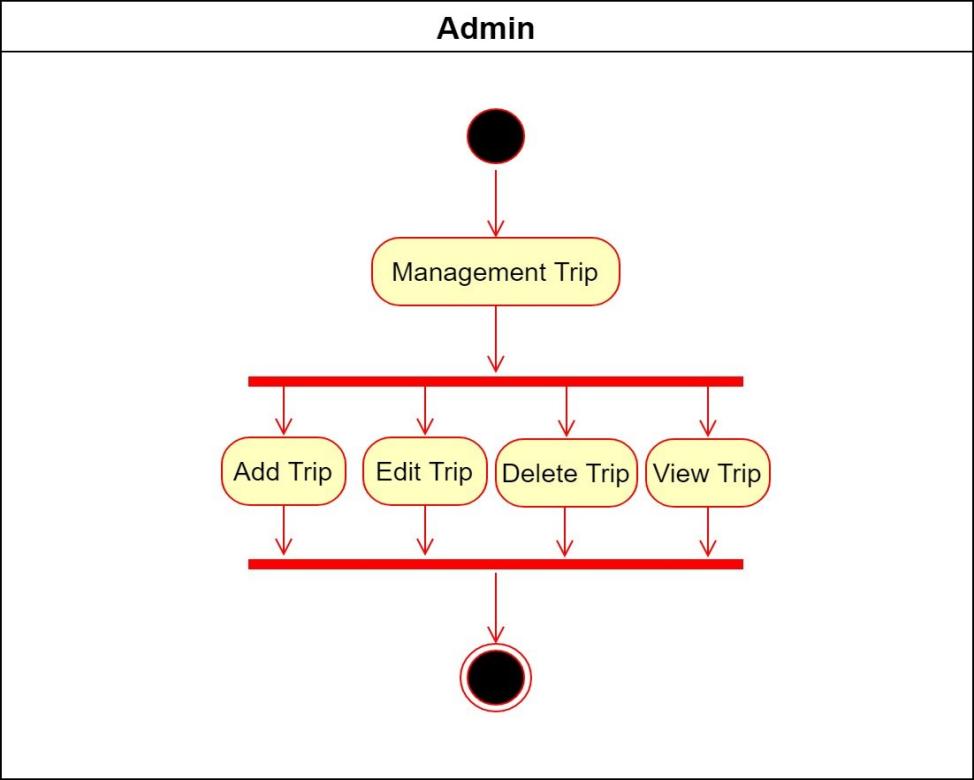
### 

**Delete trips**

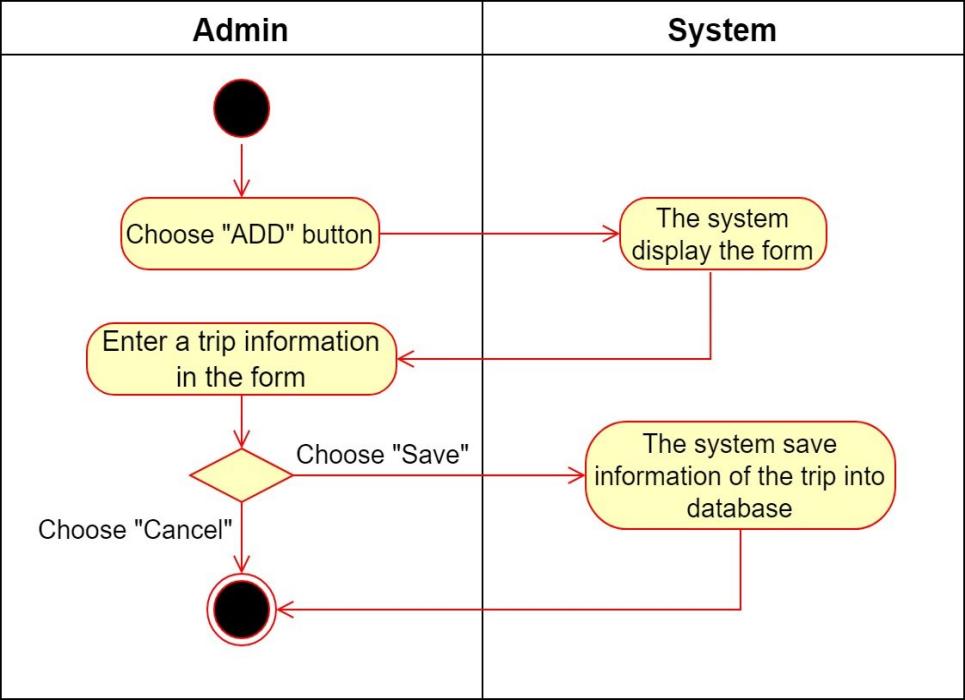


**View trips**

* 1. **Activity**



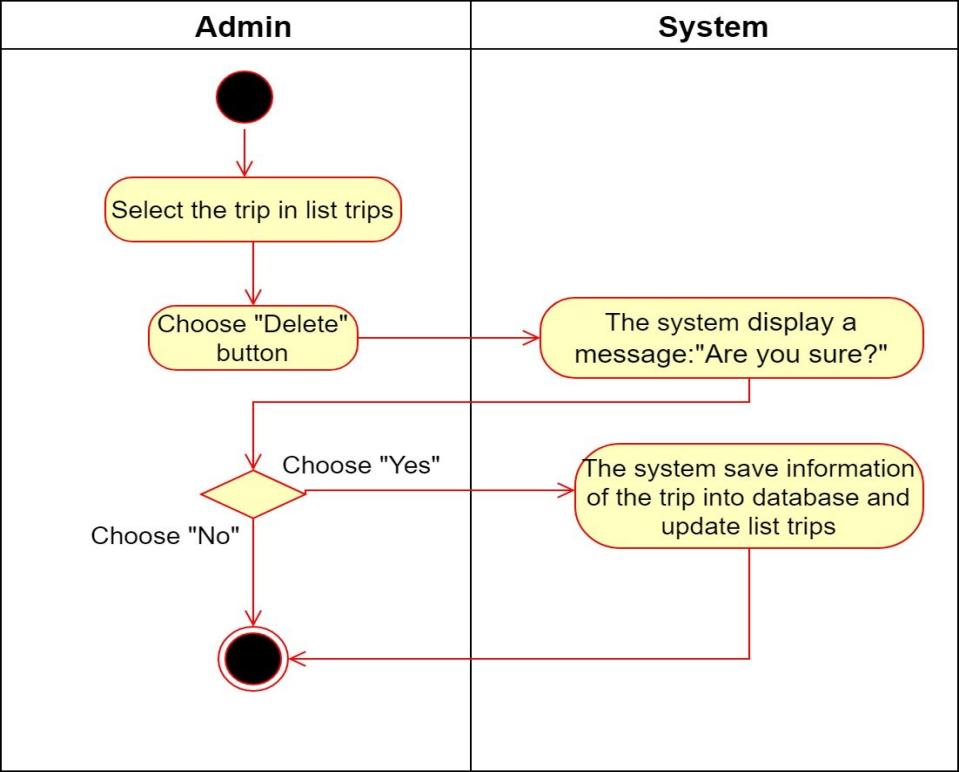
**Activity Admin**



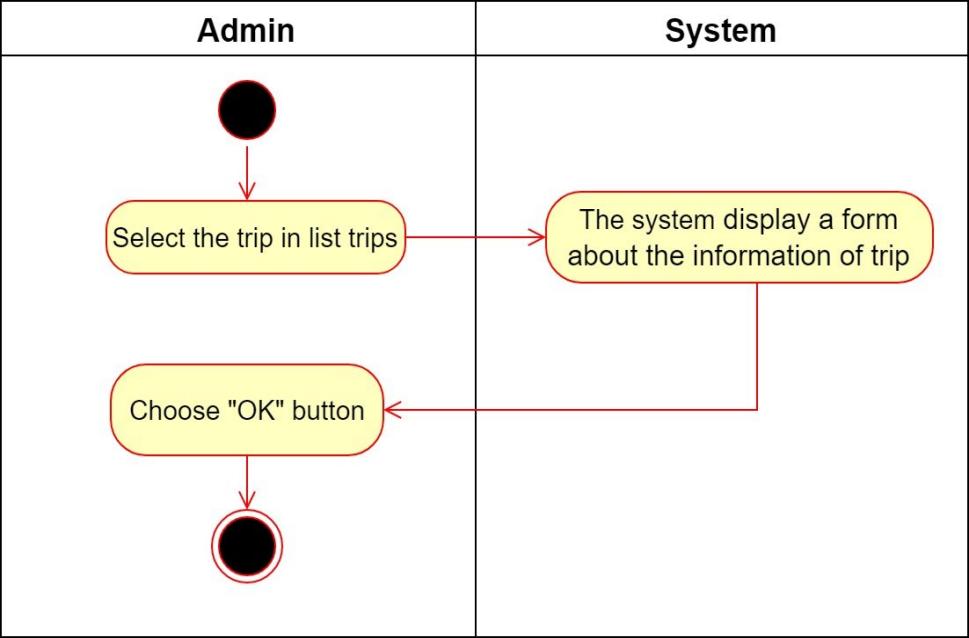
**Activity Add trips**

### 

**Activity Edit trips**

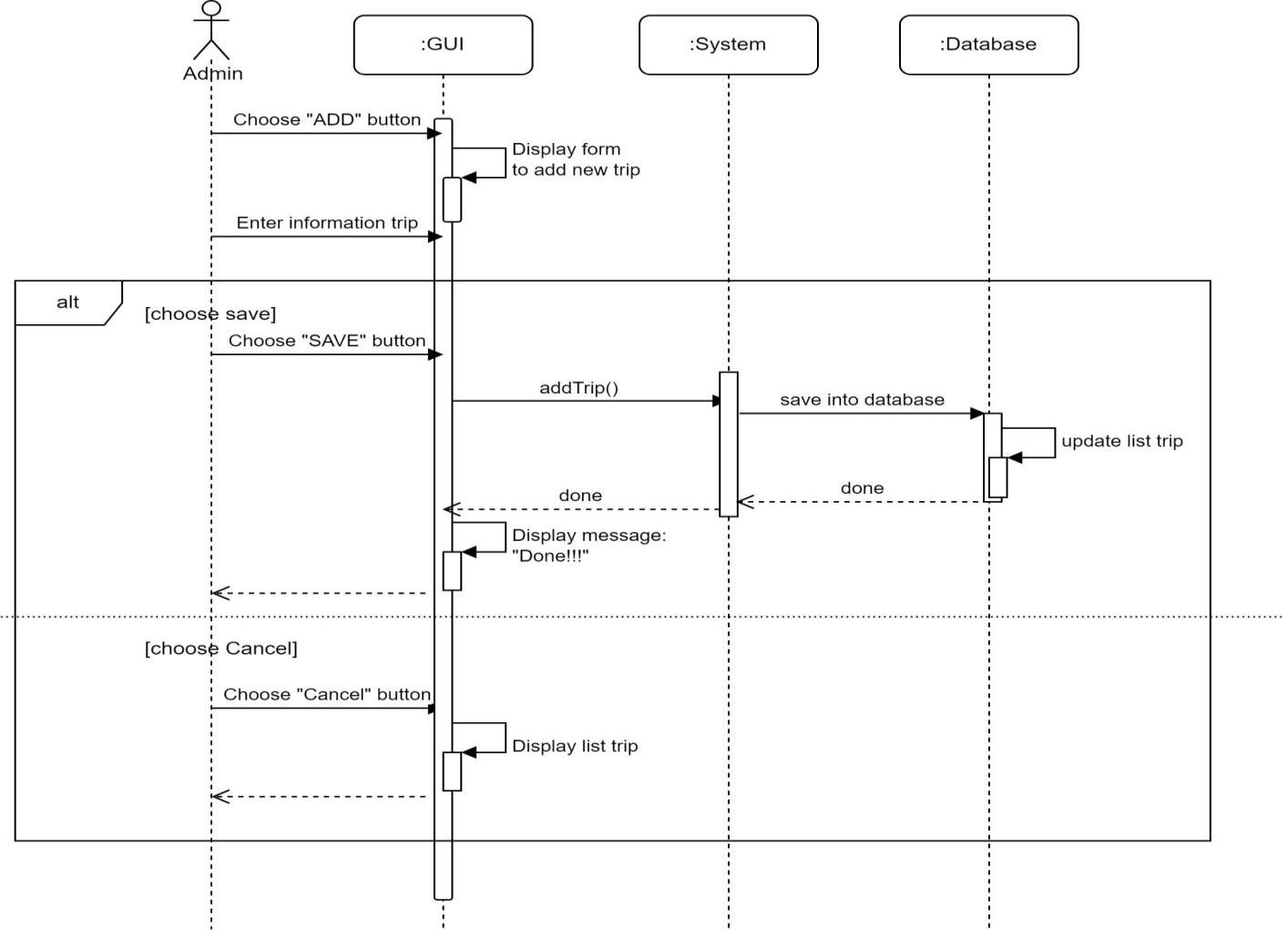


**Activity Delete trips**

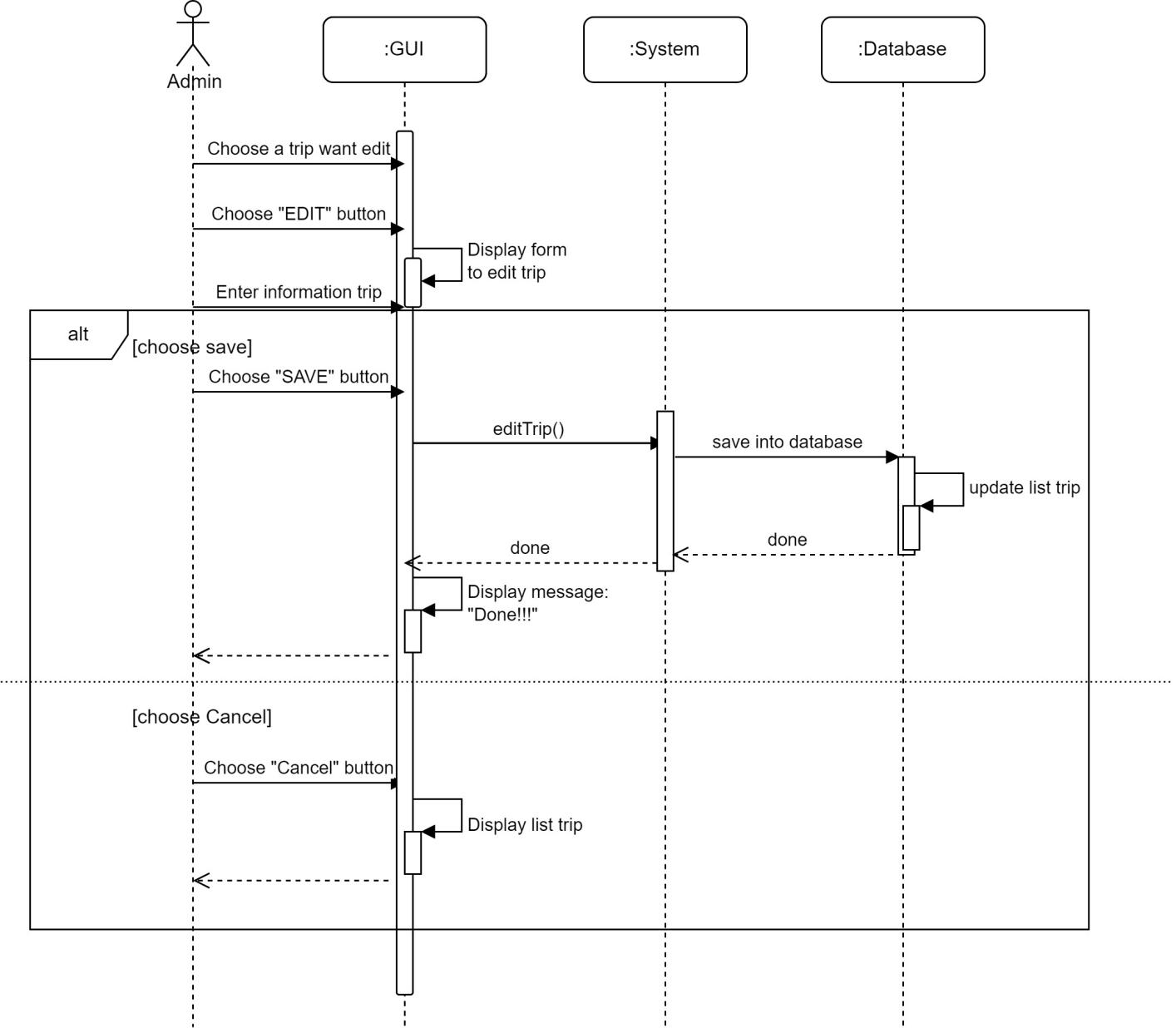


**Activity View trips**

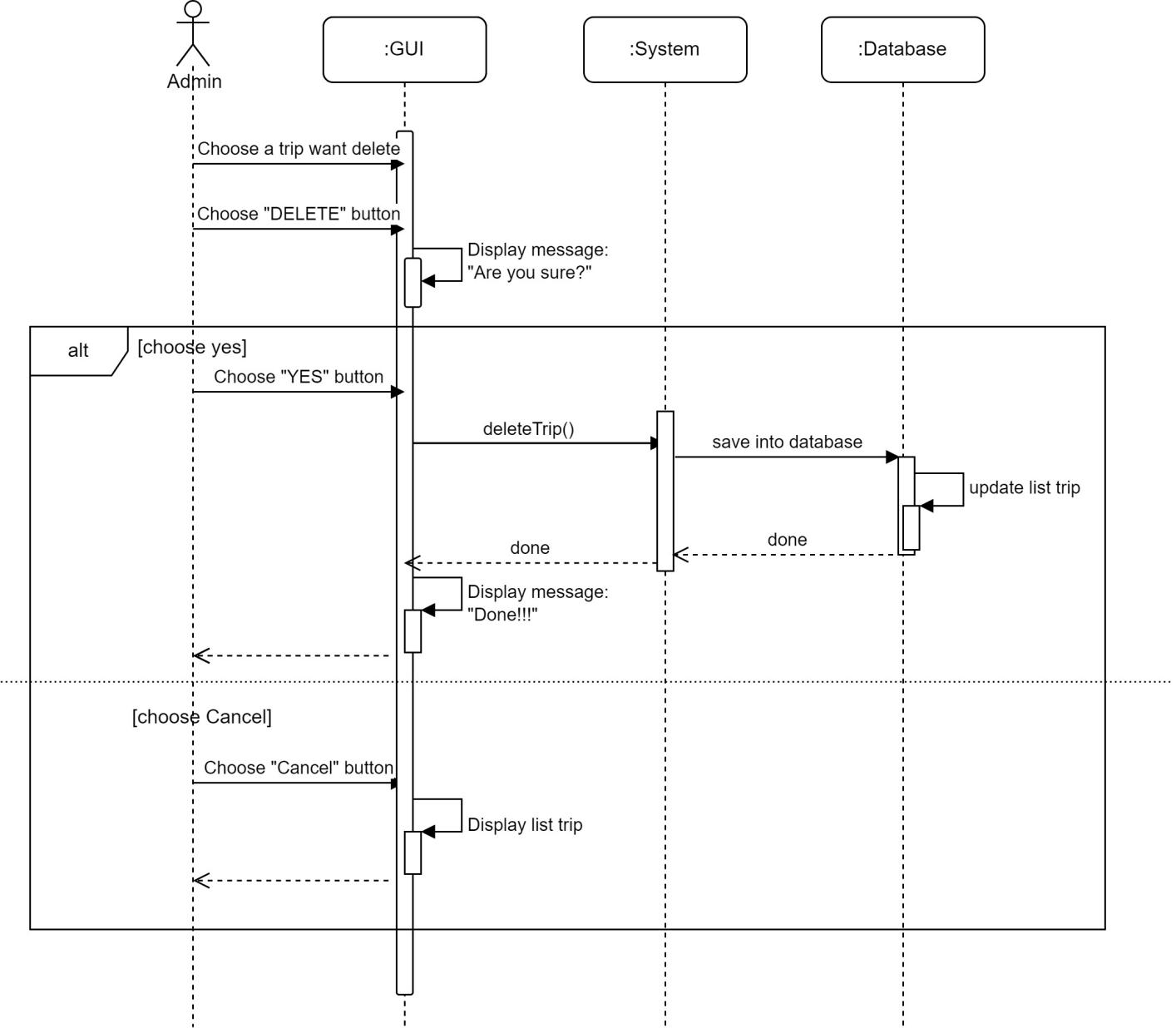
* 1. **Sequence**

****

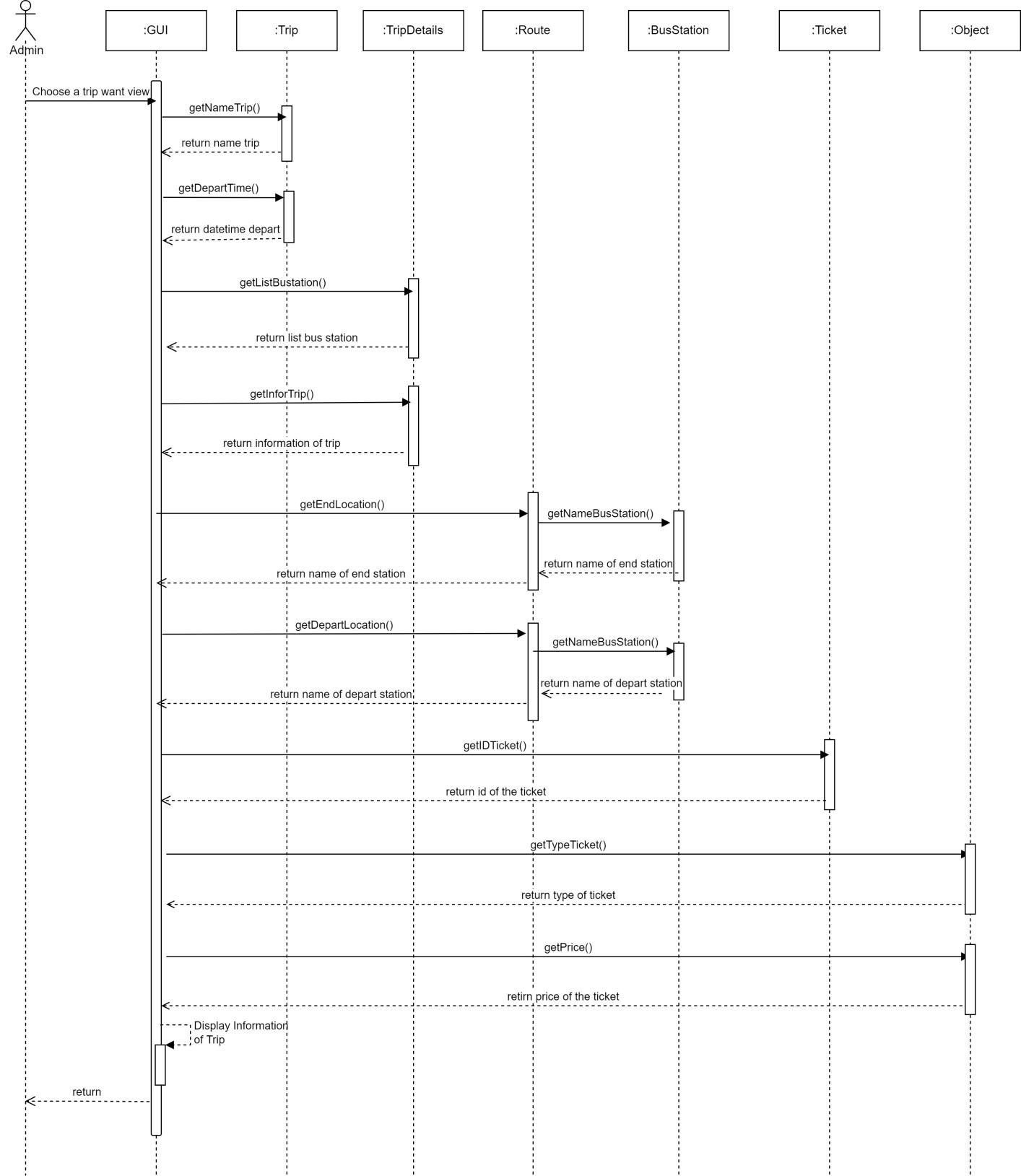
**Sequence Add trips**

****

**Sequence Edit trips**

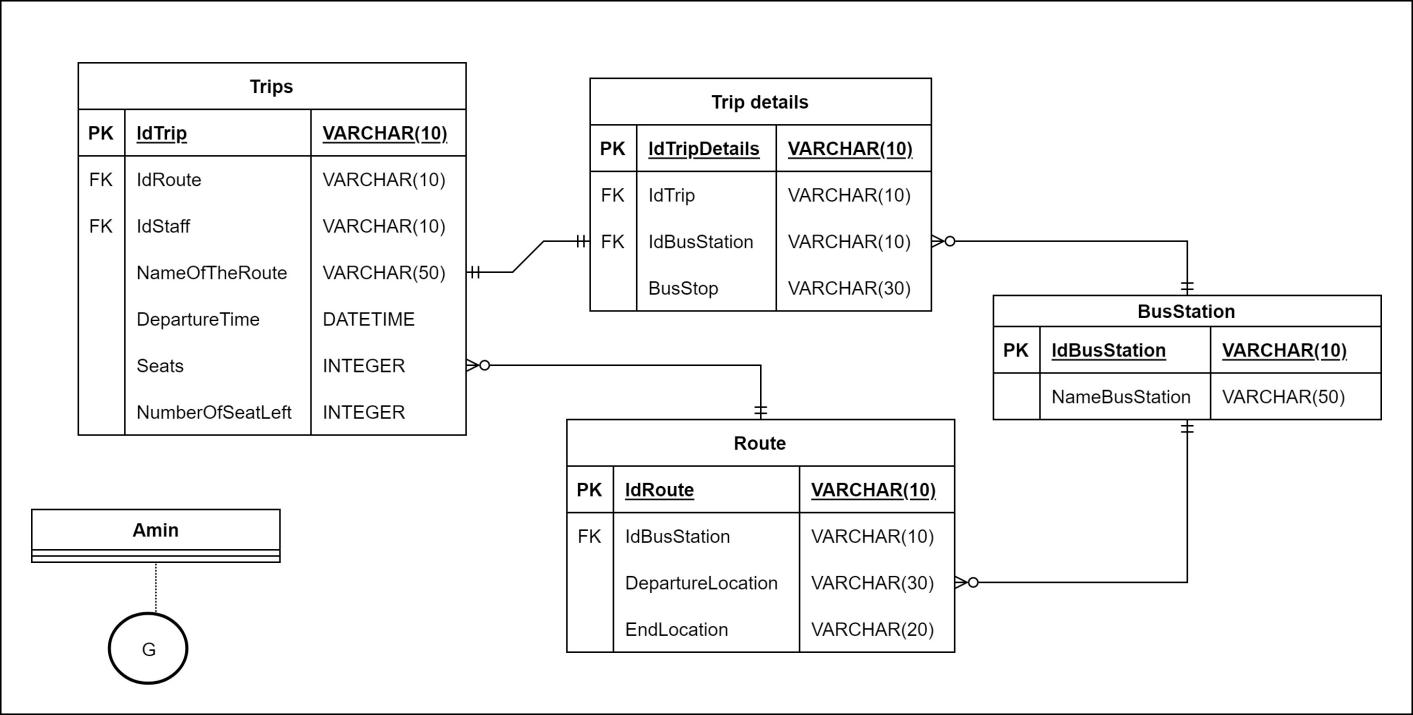
****

**Sequence Delete trips**

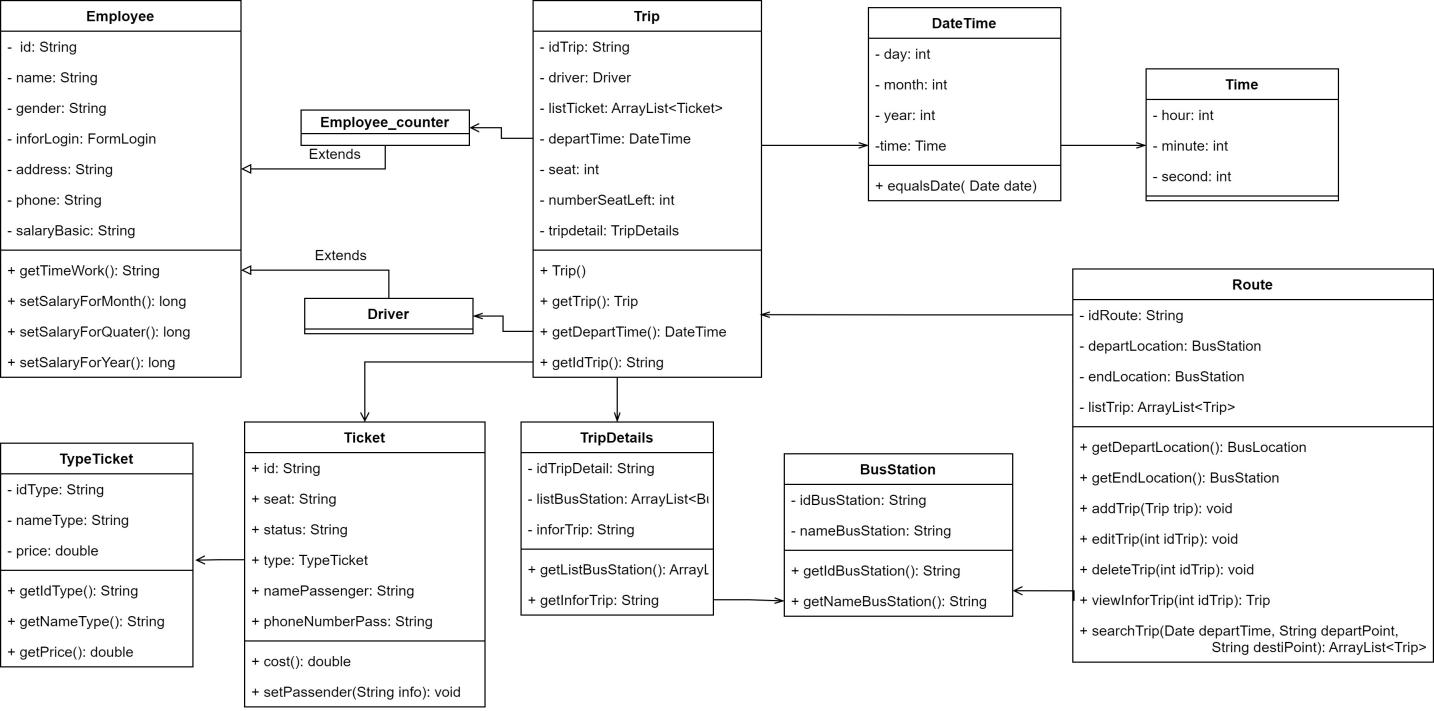
****

**Sequence View trips**

* 1. **Database**

****

* 1. **Class Diagram**

****

# Search Trip.

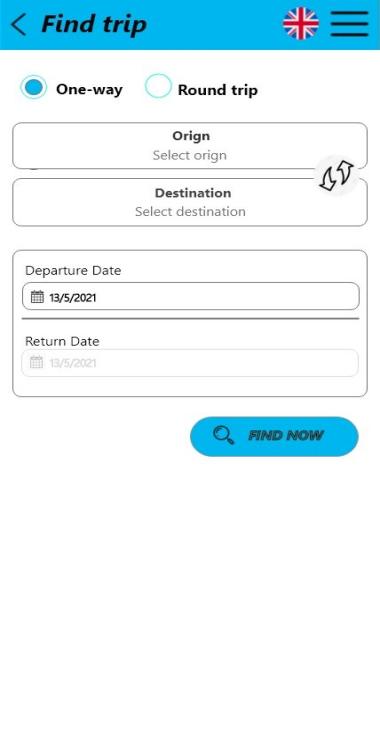
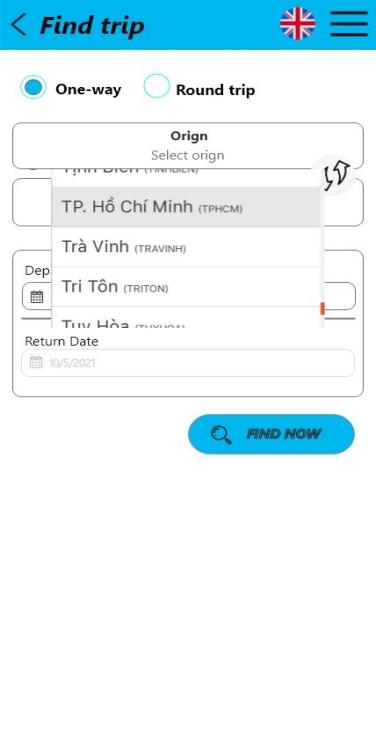
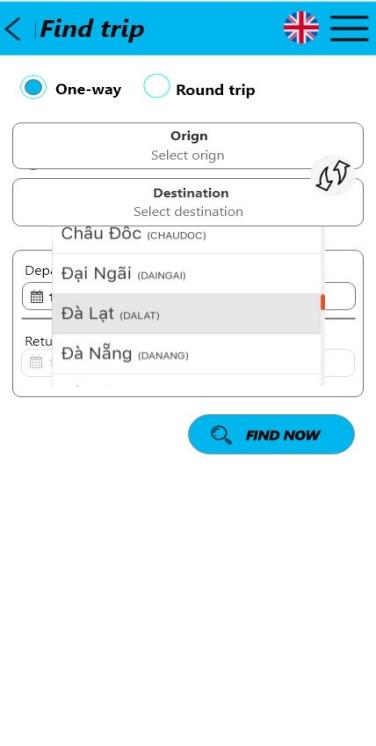
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticketing System** | **Requirement** | **Created** | **Approved** | **Updated** | **Approved** |
| Trip Search | Nguyen Ngoc Thanh Xuan |  |  |  |

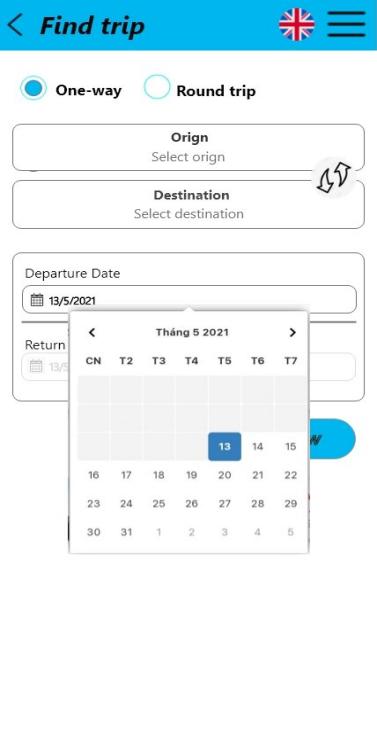
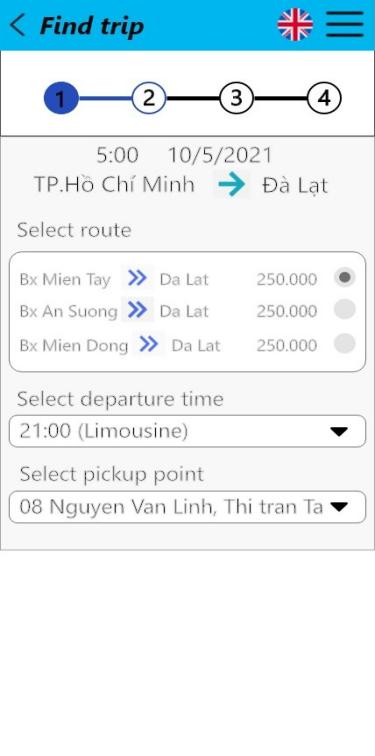
## 

## Use case Trip Search

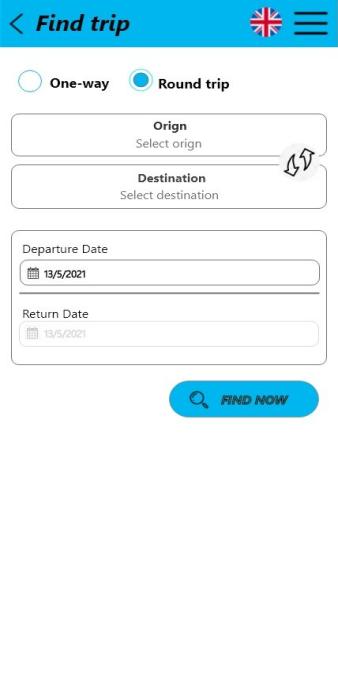
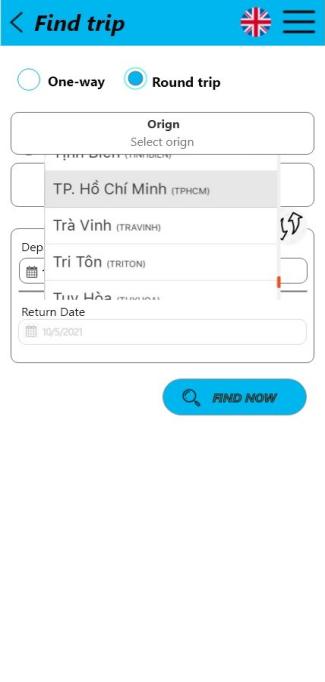
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case name** | | Search Trip | | | ID | | UC-SearchTrip |
| **Actor(s)** | | Customer | | | | | |
| **Description:** | | Search Trip use cases allow customer views information about your trip, such as upcoming trips, booking requests, and your searches. | | | | | |
| **Trigger event:** | | Customers access the ticketing system and choose to “Search for trips”. | | | | | |
| **Trigger type:** | | x | External |  | Temporal (Timing) | | |
| **Precondition:** | | Customers must access the ticketing system. | | | | | |
| **Step performed:** | | | | | **Information per steps:** | | |
| 1 | Customers choose the departure point. | | | | | When selecting the departure point, the customer clicks on "Select origin".  At this point, the system will display a list of departure points.  Customers can choose one of the departure points in the list or can also manually enter the name of the departure point. | |
| 2 | Customers choose a destination. | | | | | To select a destination, the customer clicks on "Select destination".  Customers choose the same destination as the above-mentioned operation. | |
| 3 | Customers choose the departure day. | | | | | When choosing a departure date, customers click on the date selection box below "Departure date".  At this time, the system will display a calendar frame for users to choose. | |
| 4 | Customer chooses the type of ticket. | | | | | Customer can choose “One way” or “Round trip”. | |
| 5 | If Customer choose “Round Trip”, Customer must choose “Return Date” | | | | | When choosing a return date, the Customer selects "Return date" and does the same to select the departure date. | |
| 6 | Customers select the button "Find trip". | | | | | The system displays a 1 page list of trips and returns (if any). Includes basic information of the trip such as departure time, specific address of the departure point,.. | |
| **Post condition** | | The system displays a list of trips and return trips (if any) on the day that the customer has selected. | | | | | |
| **Assumptions** | | If the system doesn’t find a trip that matches the customer's request, the system will display the message:  "Not found!!! Please choose another route or date". | | | | | |
| **Requirement met** | | Customers must provide complete information about departure date and departure point. | | | | | |
| **Outstanding issues** | | Customers cannot search without a departure point and departure date. | | | | | |
| **Priority** | |  | | | | | |
| **Risk** | | The trip may start a few hours later than the scheduled time.  The customer may not be able to find a suitable trip. | | | | | |

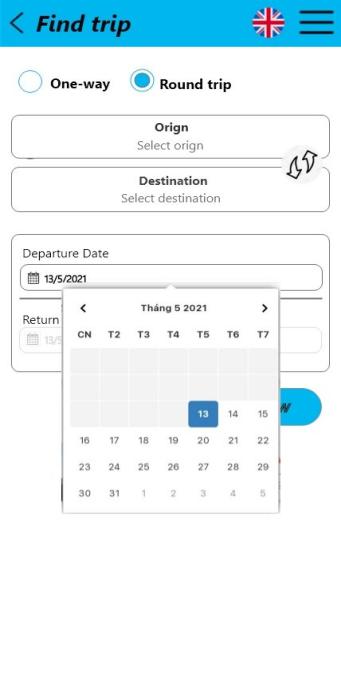
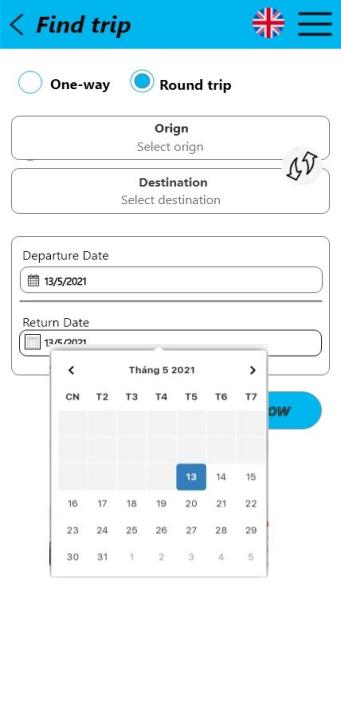
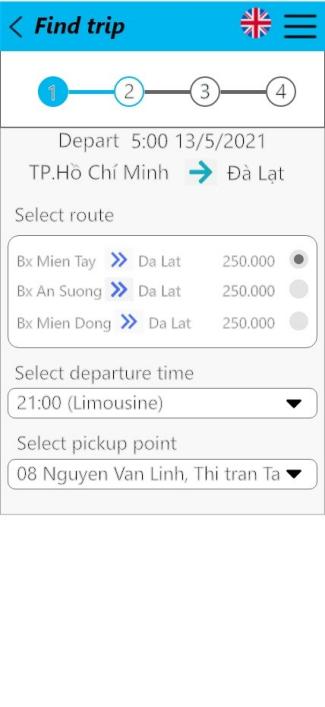
* 1. **Prototype**

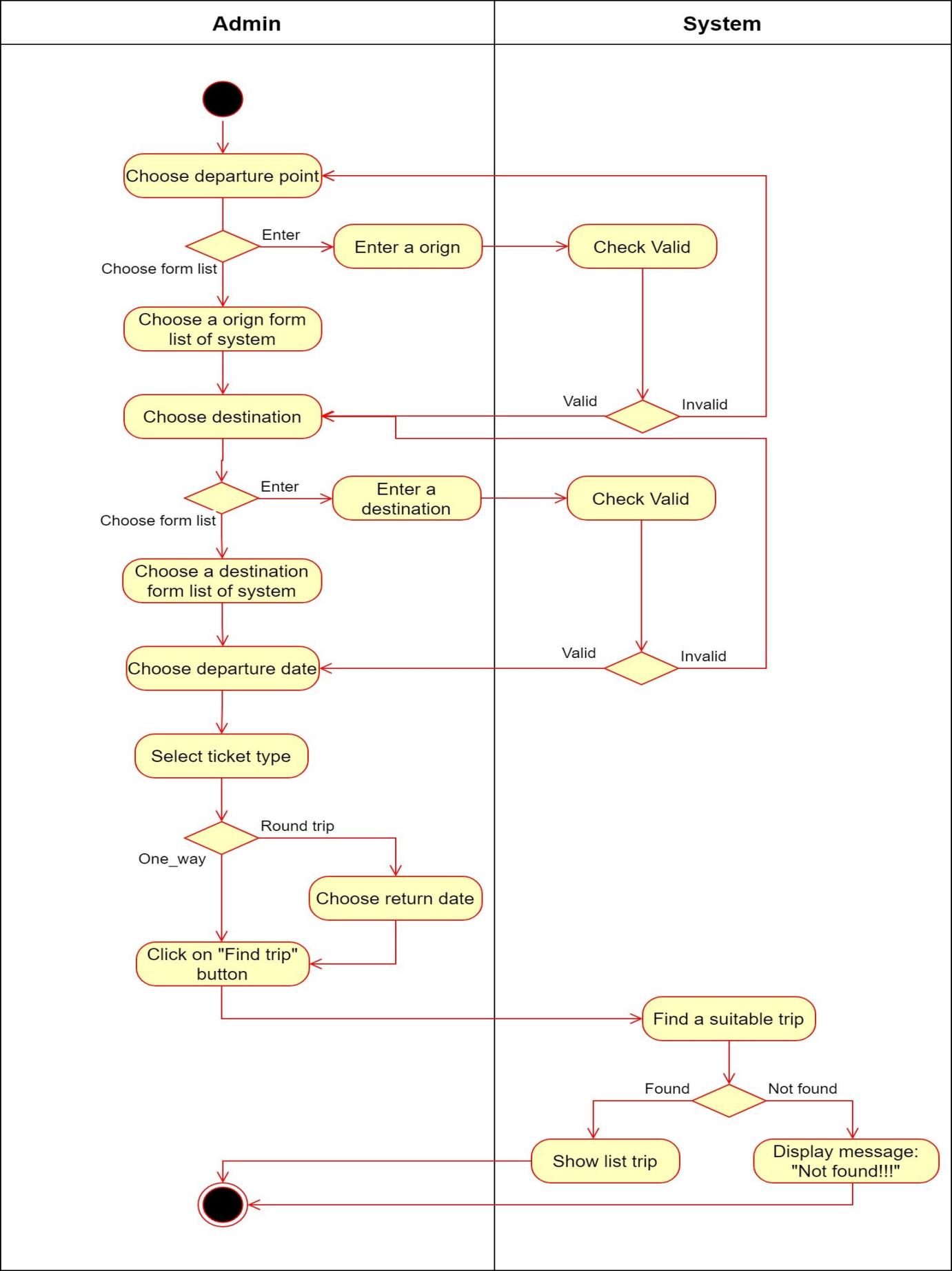
**Prototype One-way**

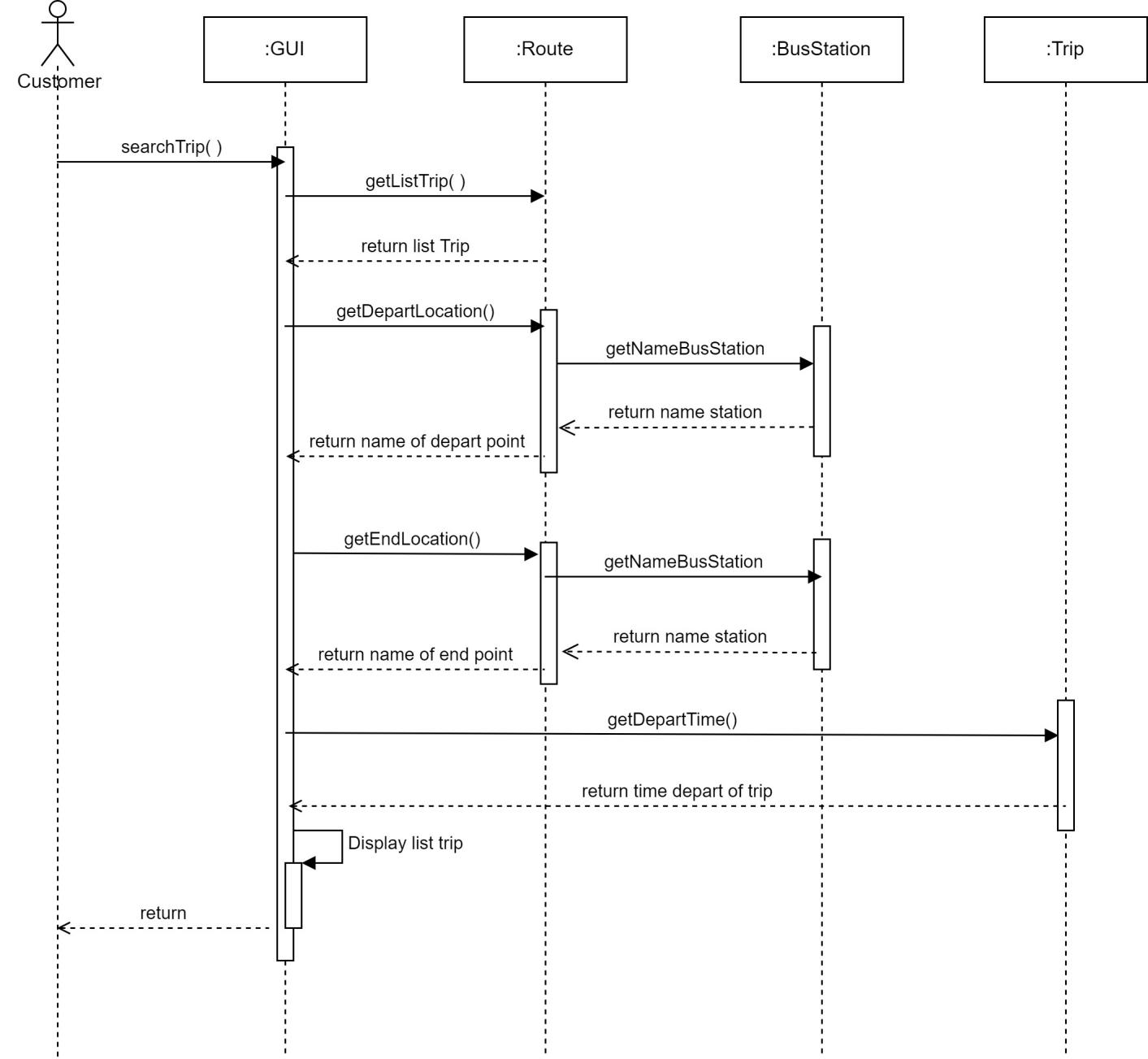
   

**Prototype Round Trips**

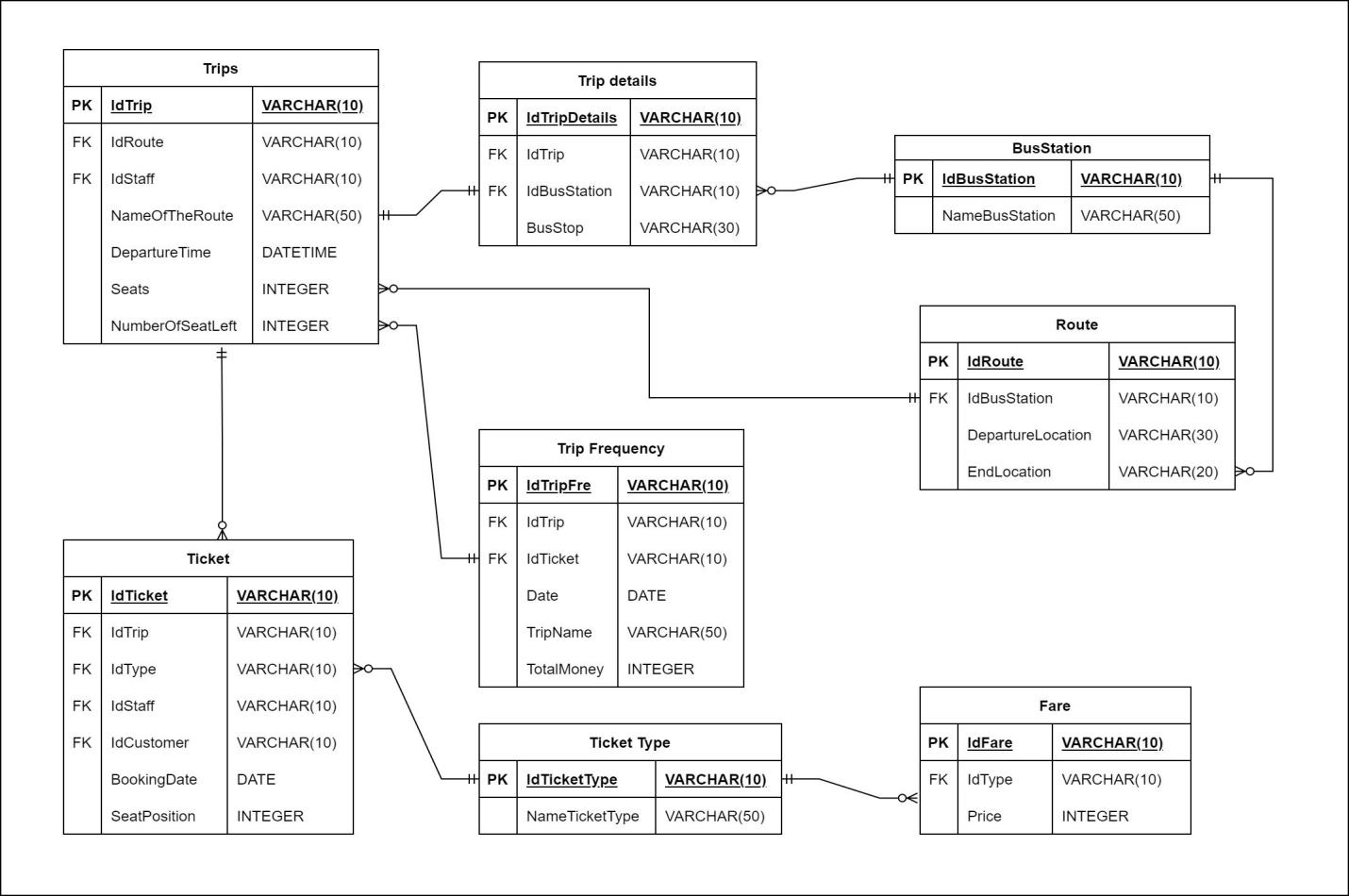
* 1. **Activity**

****

* 1. **Seqence**

****

* 1. **Database**



* 1. **Class Diagram**

