CAN THO UNIVERSITY

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY

PROJECT - FUNDAMENTAL TOPICS

IN INFORMATION TECHNOLOGY

(HIGH-QUALITY PROGRAM)

BOOKING BUS TICKET WINDOWS-BASED APPLICATION

Student: Duong Minh Khang

Student ID: B2105670

Class: 2023-2024 (K47)

Advisor: Dr. Lam Nhut Khang

Can Tho, 04/2024

CAN THO UNIVERSITY

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

PROJECT - FUNDAMENTAL TOPICS IN INFORMATION TECHNOLOGY (HIGH-QUALITY PROGRAM)

BOOKING BUS TICKET WINDOWS-BASED APPLICATION

Student: Duong Minh Khang

Student ID: B2105670

Class: 2023-2024 (Cohort K47)

Advisor: Dr. Lam Nhut Khang

Can Tho, 04/2024

ACKNOWLEDGEMENTS

The project acknowledgement is a tribute to all those who have supported can contributed to the completion of this project.

First and foremost, I would like to thank my supervisor for their guidance, expertise and continuous encouragement throughout the research process. Additionally, I would like to express my gratitude to my friends and colleagues for their assistances and discussions. In conclusion, the project would not have been possible without the supports and contributions of the aforementioned individuals.

ABSTRACT

The process of booking bus tickets at tickets counters is heavily burdened, often leading to long queues, delays and inconvenience for customers. Moreover, the lacking of administrator privileges can impede the overall operation. In this research, a comprehensive analysis of bus booking ticket system is conducted, along with gathering user requirements, designing the system and implementing it by using the Windows Forms framework. These digital solutions offer convenient features such as seat selection, secure payment options and user-friendly interfaces. In conclusion, the digital booking ticket system presents as an alternative solution for addressing the burden associated with booking bus ticket.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF DEFINITIONS, ACRONYMS AND ABBREVIATIONS

No.	Term / Abbreviation	Definition
1	CDM	Conceptual Data Model
2	LDM	Logical Data Model
3	PDM	Physical Data Model
4	char(n)	A data type: Fixed-length character string of length n
5	varchar(n)	A data type: Variable-length character string of length up to maximum of n characters
6	byte(n)	A data type: Fixed-length byte of length n
7	id	identifier
8	SHA-256	A cryptographic hash function computed with eight 32-bit words
9	salt / salt of the password	A random string appending or prepending to the password before hashing

CHAPTER 1: INTRODUCTION

1. Problem Statement

Currently, Information Technology is being widely applied in business fields, especially in the field of online booking bus ticket. There are many booking bus ticket websites used by private enterprise. However, the situation of booking tickets directly at the ticket counters still happens a lot during holidays, which has caused congestion at ticket counters. Moreover, websites are difficult to maintain during high traffic leads to customers not being able to book tickets in time. Additionally, the lack of administrator privileges in the bus ticket booking system leads to various challenge and limitations such as performing some administrative tasks, making reports, etc. To reduce these burdens, and to satisfy the increasing demand for efficient and user-friendly booking platforms, the "Booking bus ticket Windows-based application" is necessary to solve these problems effectively and precisely.

2. Purpose

The purpose of the study "Booking bus ticket Windows-based application" is to build a windows-based application that can manage the sale of bus tickets to customers and bus ticket managers in the fastest, simplest and most convenient way.

3. Scope and Approach

3.1. Scope

The subjects use the application including: Bus ticket managers (Admin) and Customers.

3.2. Approach

The requirements for booking bus ticket system can be described as followed. For customers, the system should provides user-friendly interfaces that are capable of performing fast and conveniently for customers to search and book tickets. Customer should be able to easily navigate through the system.

For managers (admin), the system should offer features which enables them to manage customers effectively. Additionally, these admin interfaces should be straightforward and handle jobs easily.

For design, we will analyze and design the system using CDM, LDM and PDM.

For theory, we will combine the theory of database, system analysis and design, a programming language (C#) and Window Forms graphical class library.

4. Research Contents

The contents of the project consist of: Understanding the book bus ticket business process, Analyze and design the system, Building the system and Testing the system at the end of the process.

5. Outline

The outline of this project consists of 7 chapters. Firstly, Chapter 1 states the problem, the purpose, scope and approach, and research contents of the project. Secondly, we present the literature review, system requirements specification as well as system design specification in Chapter 2, Chapter 3 and Chapter 4 respectively. In Chapter 5 and 6, we describe in details the implementations of the system together with the system testing. Finally, we make a conclusion with future work for this project in Chapter 7.

CHAPTER 2: LITERATURE REVIEW

1. Visual Studio

Visual Studio is an integrated development environment (IDE) developed by Microsoft. It is mainly used to develop desktop applications as well as web applications and its services.

It also supports cross-platform development, which enables developers to build applications for various platform such as Windows, macOS, Android and iOS.

2. Windows Form

Windows Form (WinForms) is a free and open-source graphical user interface GUI framework for building Windows desktop application. It is a part of the .NET framework and is built using C# or Visual Basic programming languages. [1]

Windows Form enables the developers to build the desktop applications based on the visual designer provided in Visual Studio. It allows developers to create interactive and user-friendly interfaces.

Windows Form applications also support the synchronization of data between the user interfaces and databases. This simplifies the process of displaying and manipulating data in the application.

3. Guna Framework

Guna Framework is a user interface (UI) framework specifically designed for Windows Form applications. It provides a variety of pre-designed of UI controls and customization options. It also offers additional features such as animation, transition, etc. Therefore, it can significantly enhance the development speed as well as the user experience.

4. MySQL

MySQL is an open-source relational database management system (RDBMS). It is one of the most popular and common adopted database systems in the world.

MySQL provides MySQL Connectors and APIs, which are the drivers and libraries that are used to connect applications in different programming languages such as C#, Java, etc. to MySQL database servers.

MySQL integrates well with various programming languages, frameworks, and tools. It provides connectors and APIs for popular programming languages like Java, C#, PHP, etc. [2]

5. Apache

The Apache HTTP Server, commonly known as Apache, is developed and maintained by The Apache Software Foundation [3]. It is designed as a software running on web server to serve web pages and handle HTTP requests from clients. This allows user to sends requests to the server from the web browsers.

6. VNPAY-QR

VNPAY-QR, also known as VNPAY, is a QR code payment solution developed by VNPAY, a leading Fintech company in the field of electronic payment in Vietnam [3]. It acts as intermediary portal that connects business units with banks. Customers are able to make simple and convenient payment through various methods such as Mobile Banking, ATM card/Domestic account, Visa card, etc.

CHAPTER 3: SYSTEM REQUIREMENTS SPECIFICATION

1. Functional Requirements

The booking bus ticket system offers a range of functionalities to both customers and admins that can be summarized as follows.

1.1. Customer

For customers, the system allow customers to **register account** if the customers do not have any existing account; and then they can **log in** and **sign out** using the registered accounts.

After the customers have logged in using the registered accounts, they can perform various functions such as **change account password, view or modify the account information.** They can also access the **search bus trip** menu which searching the bus trip by using some filters such as origin, destination, departure date, number of tickets or including already booked trips or not. Then they can select one of the bus trips to **choose their desired seat number** or **view the bus trip schedule information.**

Once the customers have chosen the seat numbers of a bus trip, they can confirm the selection and **book the tickets,** with two options for payment. The first option is made physically at ticket counters. The other one is **online payment** through VNPAY-QR. Additionally, they can **view their booked tickets**, including the general information, ticket's payment status, bus trip details and seat numbers.

1.2. Admin

On the other hand, admins have their own set of functionalities. They can **log in** and **sign out** using only the existing accounts in the system.

After the admins have logged in using the existing accounts, they can **search for the customer account** by using the phone number. Firstly, admins have abilities to **view or change the customer account information, reset customer account password** or **delete customer account.** Secondly, they can **manage customer's booked tickets**, including reviewing the ticket details, **changing the seat numbers** or **deleting the booked tickets** as needed. Finally, they can access various **statistics of booked tickets** by customers. This allows them to observe the fluctuation of number of tickets as well as the total selling tickets and revenue.

2. Non-functional Requirements

For timing requirements, the time response is no more than 5 seconds, processing and calculating time are less than 10 seconds.

For security, the system must ensure the safety of users' personal information and data, and the user privileges.

For reliability, the system is expected to work 24/7 except when updating new versions. Each update is spaced at least 2 months apart and the update time should not exceed 1 hour. Furthermore, the system can recover data from 1 hour before technical problems such as power failure, memory overflow or software conflicts.

3. Operating System

The application runs on Personal Computer (Desktop Computer, Laptop) with Windows Operating System (Windows 10 version 1703 or higher, Windows 11 version 21H2 or higher).

4. Design and Implementation Constraints

Constraint	Description
Programming Language	C#
Windows Forms	.NET Framework 4.7.2 Guna Framework
Database Management System	MySQL version 8.0.34
Integrated Development Environment	Microsoft Visual Studio Community 2022 (64-bit) Version 17.8.3
Language	English

Table 1. Design and Implementation Constraints

5. Assumption and Dependencies

Firstly, the system may experience high memory usage and potentially overflow RAM. Additionally, the system may not compatible with the operating systems that are not included in the constraints. Finally, the system may be vulnerable to computer viruses or malware attacks.

CHAPTER 4: SYSTEM DESIGN SPECIFICATION

1. Application Architecture

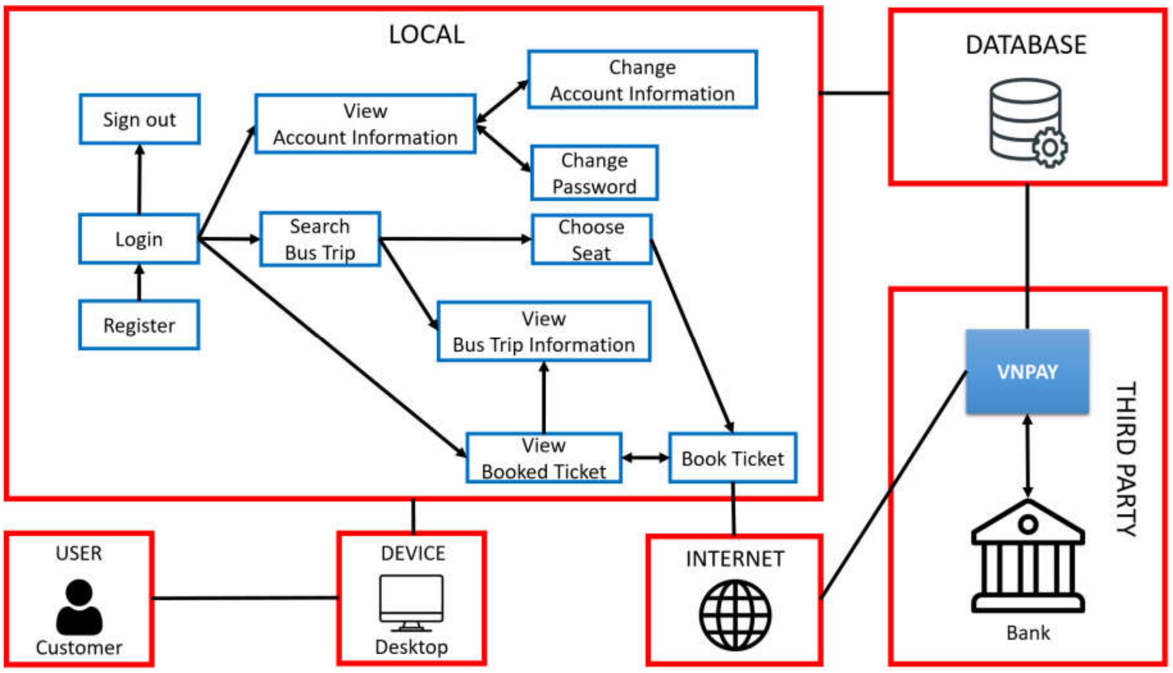


Figure 1. Application Architecture (Customer)

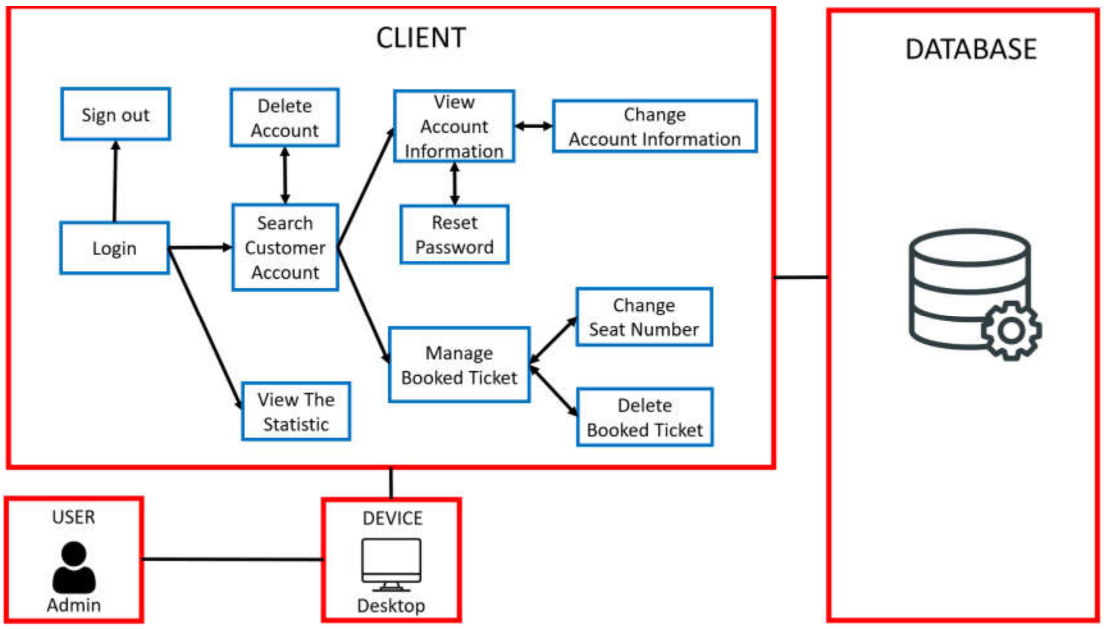


Figure 2. Application Architecture (Admin)

2. Data Design

2.1. Data Description

2.1.1. CDM

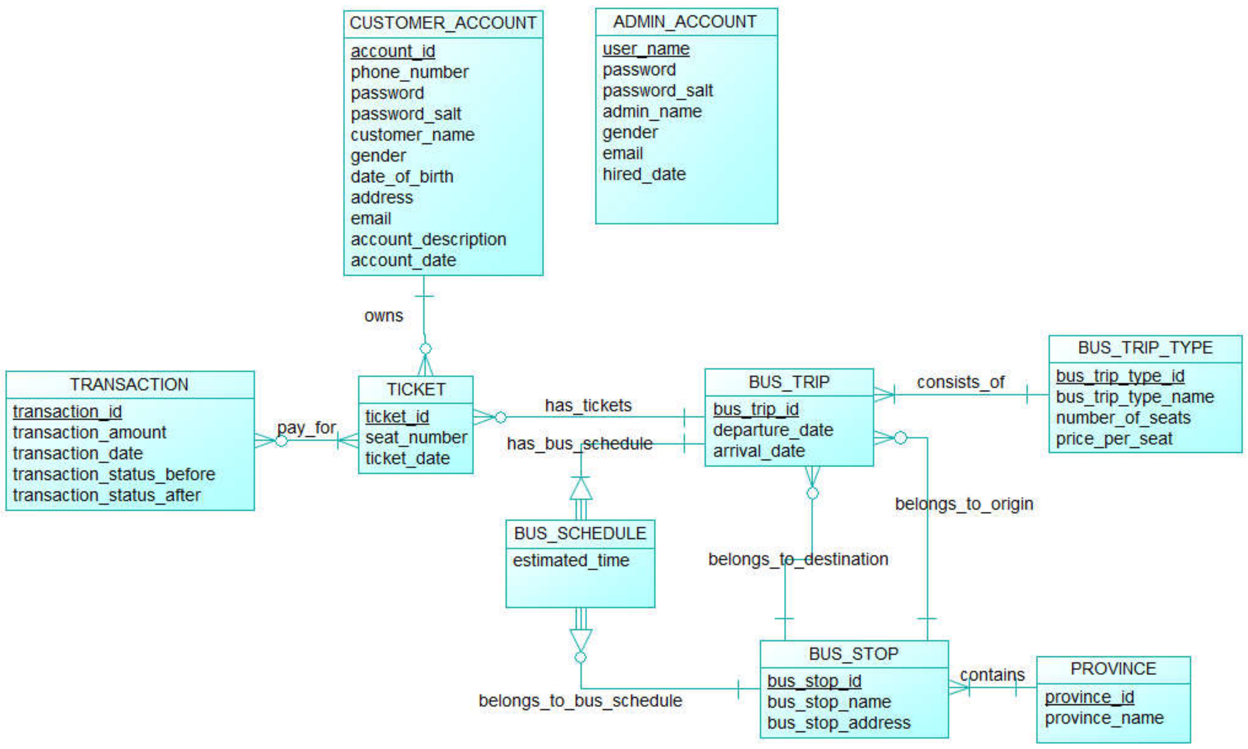


Figure 3. Conceptual Data Model

2.1.2. LDM

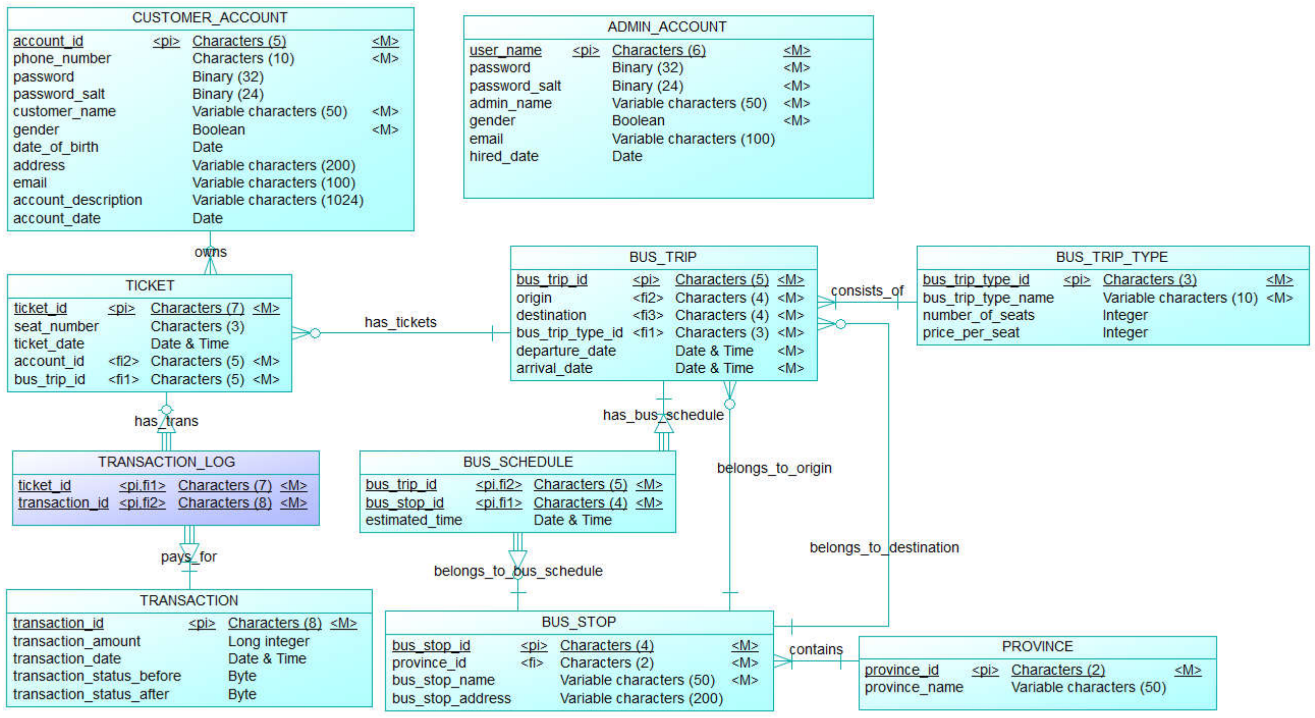

CUSTOMER_ACCOUNT
account_id <pi> Characters (5) <M>
phone_number Characters (10) <M>
password Binary (32)
password_salt Binary (24)
customer_name Variable characters (50) <M>
gender Boolean <M>
date_of_birth Date
address Variable characters (200)
email Variable characters (100)
account_description Variable characters (1024)
account_date Date
ADMIN_ACCOUNT
user_name <pi> Characters (6) <M>
password Binary (32) <M>
password_salt Binary (24) <M>
admin_name Variable characters (50) <M>
gender Boolean <M>
email Variable characters (100)
hired_date Date
owns
TICKET
ticket_id <pi> Characters (7) <M>
seat_number Characters (3)
ticket_date Date & Time
account_id <fi2> Characters (5) <M>
bus_trip_id <fi1> Characters (5) <M>
has_tickets
BUS_TRIP
bus_trip_id <pi> Characters (5) <M>
origin <fi2> Characters (4) <M>
destination <fi3> Characters (4) <M>
bus_trip_type_id <fi1> Characters (3) <M>
departure_date Date & Time <M>
arrival_date Date & Time <M>
consists_of
BUS_TRIP_TYPE
bus_trip_type_id <pi> Characters (3) <M>
bus_trip_type_name Variable characters (10) <M>
number_of_seats Integer
price_per_seat Integer
has_trans
has_bus_schedule
TRANSACTION_LOG
ticket_id <pi,fi1> Characters (7) <M>
transaction_id <pi,fi2> Characters (8) <M>
BUS_SCHEDULE
bus_trip_id <pi,fi2> Characters (5) <M>
bus_stop_id <pi,fi1> Characters (4) <M>
estimated_time Date & Time
belongs_to_origin
belongs_to_destination
pays_for
belongs_to_bus_schedule
TRANSACTION
transaction_id <pi> Characters (8) <M>
transaction_amount Long integer
transaction_date Date & Time
transaction_status_before Byte
transaction_status_after Byte
BUS_STOP
bus_stop_id <pi> Characters (4) <M>
province_id <fi> Characters (2) <M>
bus_stop_name Variable characters (50) <M>
bus_stop_address Variable characters (200)
contains
PROVINCE
province_id <pi> Characters (2) <M>
province_name Variable characters (50)

Figure 4. Logical Data Model

2.1.3. PDM

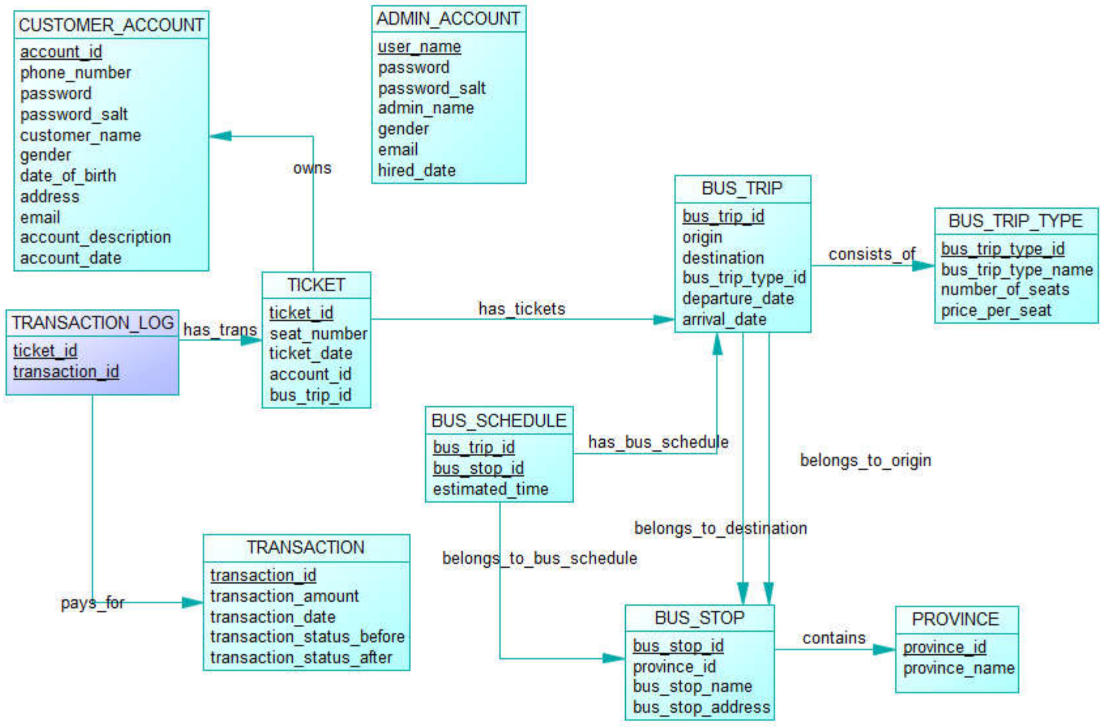


Figure 5. Physical Data Model

2.2. Data Dictionary

2.2.1. ADMIN_ACCOUNT entity

Description: To store information related to admin accounts including: A unique user name to identify each admin account, the hashed password of customer account using hash function SHA-256, the salt of the password; admin account information such as full name, gender email and hired date of admin.

No.	Attribute Name	Type	Description
1	**user_name**	char(8)	Admin account's user name
2	password	binary(32)	Admin account's encrypted password
3	password_salt	binary(24)	Admin account's password salt
4	admin_name	varchar(50)	Full name of admin
5	gender	boolean	Gender of admin, male is true(1) and female is false(0)
6	email	varchar(100)	Email of admin
7	hired_date	date	The hired date of admin

Table 2. ADMIN_ACCOUNT entity

2.2.2. CUSTOMER_ACCOUNT entity

Description: To store information related to customer accounts including: An account id for management, an unique phone number to identify each customer account, the hashed password of customer account using hash function SHA-256, the salt of the password; customer account information such as full name, gender, address, email, account description and the date that the account was created.

No.	Attribute Name	Type	Description
1	**account_id**	char(5)	Id of customer account
2	phone_number	char(10)	Customer's phone number
3	password	binary(32)	Customer account's encrypted password
4	password_salt	binary(24)	Customer account's password salt
5	customer_name	varchar(50)	Full name of customer

6	gender	boolean	Gender of customer
7	date_of_birth	date	Date of birth of customer
8	address	varchar(200)	Address of customer
9	email	varchar(100)	Email of customer
10	acccount_description	varchar(1024)	Description of customer account
11	account_date	date	The creation date of customer account

Table 3. CUSTOMER_ACCOUNT entity

2.2.3. PROVINCE entity

Description: To store information of provinces. Each province is assigned to a unique id for management.

No.	Attribute Name	Type	Description
1	**province_id**	byte	Id of province
2	province_name	varchar(50)	Name of province

Table 4. PROVINCE entity

2.2.4. BUS_STOP entity

Description: A unique id is used for managing the bus stop name with its address. Moreover, the id of the province where the bus stop located in is also included.

No.	Attribute Name	Type	Description
1	**bus_stop_id**	char(4)	Id of bus stop
2	bus_stop_name	varchar(50)	Name of bus stop
3	bus_stop_address	varchar(200)	Address of bus stop

Table 5. BUS_STOP entity

2.2.5. BUS_TRIP_TYPE entity

Description: To store information of the types of a bus trips. A bus trip type is made up of the id (for management), name, the capacity (the total number of seats) it can contain and the price of each seat.

No.	Attribute Name	Type	Description
1	**bus_trip_type_id**	char(3)	Id of bus trip type
2	bus_trip_type_name	varchar(10)	Name of the bus trip type
3	number_of_seats	integer	The total number of seats
4	price_per_seat	integer	Price per seat (currency is Vietnamese Dong)

2.2.6. BUS_TRIP entity

Description: To store information related to bus trips. A bus trip is assigned to a unique id for management, as well as the departure date and arrival date are recorded. A bus trip must have an id for its type. In addition, the departure location id (origin) and arrival location id (destination) must also be included in a bus trip.

No.	Attribute Name	Type	Description
1	**bus_trip_id**	char(5)	Id of bus trip
2	departure_date	datetime	Departure date of bus trip
3	arrival_date	datetime	Arrival date of bus trip

Table 6. BUS_TRIP entity

2.2.7. BUS_SCHEDULE entity

Description: To store information of bus schedules. A bus schedule is only defined when a bus trip is expected to arrive at a bus stop. It is necessary to store the estimated time for the time when the bus trip to arrive at that bus stop in a specific bus schedule.

No.	Attribute Name	Type	Description
1	estimated_time	datetime	Estimated time to arrive at a bus stop of a bus trip

Table 7. BUS_SCHEDULE entity

2.2.8. TICKET entity

Description: To store information of a ticket. A unique id is used for management tickets, the seat number of the ticket is composed of one letter A or B at the beginning followed by two digits, e.g. A02, B14, and the date when ticket was booked. The ticket is created at the time when a customer choose to book a seat number in a bus trip and it is owned by that customer's account.

No.	Attribute Name	Type	Description
1	**ticket_id**	char(7)	Id of ticket
2	seat_number	char(3)	Seat number of the ticket
3	ticket_date	date	Booked date

Table 8. TICKET entity

2.2.9. TRANSACTION entity

Description: To store information of a transaction. A transaction consists of a unique id created at the time when that transaction is created through VNPAY, some relevant information to the transaction, such as the amount of money being transferred and the date. Additionally, the transaction status before and after the commit must also be recorded.

No.	Attribute Name	Type	Description
1	**transaction_id**	char(8)	Id of transaction
2	transaction_amount	long	The amount of money
3	transaction_date	datetime	The created date
4	transaction_status_before	byte	The status (the respond code) before the commit
5	transaction_status_after	byte	The status (the respond code) after the commit

Table 9. TRANSACTION entity

3. Function Design

The following sections will describe in details the functionalities of booking bus ticket system for both customers and admins.

3.1. Login

Function: Login	ID: FN-01
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to login to the system using the registered account	

Normal flow: **1.** The system requests the actor to enter user name/phone number and password. **2.** The actor enters User name/Phone number and Password. **3.** The actor chooses the **Login** function **4.** The system checks the login information. **Sub4a:** Verify the login information **5.** The system announces that the actor logins successfully and displays the system menu according to privilege of the actor.
Sub flows: **Sub 4a:** Verify the login information **4.1.** The system checks if any input fields is invalid (e.g. empty, maximum limit). If yes then the system describes which field is invalid, prompts the actor to re-enter and then go to Step 3; else go to Step 4.2. **4.2.** The system checks if the user name/phone number and password are in the database. If yes then go to Step 5; else the system announces login failed, prompts the actor to re-enter those fields and go to Step 3.
Alternate/Exceptional flows: If the system can't login, it will display error message.

3.2. Sign out

Function: Sign out	ID: FN-02
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to sign out from the system	
Normal flow: **1.** The actor chooses the **Sign out** function. **2.** The system requests the confirmation of sign out from the actor. **3.** The actor confirms to sign out from the system **4.** The system signs the actor out and announces that he/she signs out successfully.	
Sub flows: (None)	

Alternate/Exceptional flows: If the actor does not confirm to sign out from the system, then the system will remain the state before signing out. If the system can't sign out, it will display error message.

3.3. Register account

Function: Register account	ID: FN-03
Main actor: Customer	Priority: Essential
Brief description: Allow the actor to register new account in order to log in to the system	
Normal flow: **1.** The actor chooses the **Register account** function. **2.** The system displays a form to input the following fields: Full name, Email, Gender, Phone number, Password, Confirm password. **3.** The actor confirms the register information. **4.** The system checks the register information. **Sub4a:** Verify the register information. **5.** The system registers a new account for the actor and announces that it has registered successfully.	
Sub flows: **Sub 4a:** Verify the login information **4.1.** The system checks if any input fields is invalid (e.g. empty, maximum limit). If yes then the system describes which field is invalid, prompts the actor to re-enter and go to Step 3; else go to Step 4.2. **4.2.** The system checks if the phone number is in the database. If yes then the system asks the actor to re-enter another phone number and go to Step 3; else go to Step 5.	
Alternate/Exceptional flows: If the actor does not confirm to register a new account, then the system will remain the state before that. If the system can't register a new account, it will display error message.	

3.4. Change account password

Function: Change account password	ID: FN-04
Main actor: Customer	Priority: Essential
Brief description: Allow the actor to change the account password	
Normal flow: **1.** Include <<View account information>> **2**. The actor chooses **Change account password** function **3.** The system asks the actor to enter the following fields: Old password, New password, Confirm password. **4**. The actor confirms the inputs. **5.** The system checks if any input fields is invalid (e.g. empty, maximum limit, not match). If yes then the system describes which field is invalid, prompts the actor to re-enter and go to Step 4; else go to Step 6. **6.** The system changes the old password to new one and announces that it has changed password successfully.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm the inputs, then the system will remain the state before that. If the system can't change old password to the new one, it will display error message.	

3.5. View account information

Function: View account information	ID: FN-05
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to view the account information if the actor is Customer or customer account information selected from <<Select customer account>> if the actor is Admin.	
Normal flow: **1**. Include <<Select customer account>> if the actor is Admin.	

2. The actor chooses the **View account information** function. **3.** The system displays the account information including: Full name, Address, Email, Gender, Phone number, Date of birth, Account date, Account description.
Sub flows: (None)
Alternate/Exceptional flows: If the actor chooses Close/Cancel, the system will cancel the display of account information. If the system can't get the account information from the database, it will display error message.

3.6. Change account information

Function: Change account information	ID: FN-06
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to change the account information if the actor is Customer or customer account information selected from <<Select customer account>> if the actor is Admin.	
Normal flow: **1.** Include <<Select customer account>> if the actor is Admin. **2.** Include <<View account information>> **3.** The actor can modify the following fields: Full name, Address, Email, Gender, Date of birth, Account description or *Phone number* (if the actor is Admin). **4.** The actor confirms to save changes. **5.** The system checks the account information. **Sub5a:** Verify the account information. **6.** The system changes the account information and announces that it has changed account information successfully.	
Sub flows: **Sub5a:** Verify the account information **5.1.** The system checks if any input fields is invalid (e.g. empty, maximum limit). If yes then the system describes which field is invalid, prompts the actor	

to re-enter and go to Step 4; else go to Step 5.2 (if the actor is Admin) or Step 6 (if the actor is Customer). **5.2.** The system checks if the phone number is in the database. If yes then the system asks the actor to re-enter phone number and go to Step 4; else go to Step 6.
Alternate/Exceptional flows: If the actor does not confirm to save changes, the system will cancel all the temporary changes. If the system can't change the account information, it will display error.

3.7. Search bus trip

Function: Search bus trip	ID: FN-07
Main actor: Customer	Priority: Essential
Brief description: Allow the actor to search bus trip	
Normal flow: **1**. The actor chooses **Search bus trip** function. **2.** The system asks the actor to enter the following fields: Origin, Destination, Departure date and Number of tickets to book or Including booked trip or not. **3.** The actor confirms the inputs. **4.** The system checks if any input fields is invalid (e.g. empty, maximum limit). If yes then the system describes which field is invalid, prompts the actor to re-enter and go to Step 3; else go to Step 5. **5.** The system displays the list of bus trips that match the inputs. Each bus trip includes the following information: Origin bus stop, Destination bus stop, Departure date, Arrival date, Bus trip type, Number of available seats, Price per seat and *Chosen seat numbers* (more details in <<Choose seat number>>).	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm the inputs, the system will remain the state before that. If the system can't get the bus trip information from database, it will display error. If the system does not find any bus trip that matches the inputs, it will announce that no result found.	

3.8. View bus trip schedule

Function: View bus trip schedule	ID: FN-08
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to view	
Normal flow: **1**. Include <<Search bus trip>> or <<View booked ticket>> **2.** The actor chooses **View bus trip schedule** function. **3.** The system displays the list of bus stops such that a bus trip goes through. Each bus stop includes the following information: Bus stop name, Bus stop address and Estimated time to arrive at that bus stop.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor chooses Close/Cancel, the system will cancel the display of bus trip schedule. If the system can't get the bus trip schedule from the database, it will display error message. If the bus trip schedule is empty, the system will announce that no result found.	

3.9. Choose seat number

Function: Choose seat number	ID: FN-09
Main actor: Customer	Priority: Essential
Brief description: Allow the actor to choose seat number in a bus trip	
Normal flow: **1**. Include <<Search bus trip>> **2.** The actor chooses **Choose seat number** function. **3.** The system displays the seat number layout of a bus trip and includes the following information: Seat number and its state (booked or not booked) **4.** The actor chooses the seat numbers that are not booked. **5.** The actor confirms the inputs. **6.** The system updates *chosen seat numbers* according to the inputs.	

Sub flows: (None)
Alternate/Exceptional flows: If the actor chooses the seat numbers that have already been booked, the system will denied. If the actor does not confirm the inputs, the system will remain the state before that.

3.10. Book ticket

Function: Book ticket	ID: FN-10
Main actor: Customer	Priority: Essential
Brief description: Allow the actor to book the chosen tickets	
Normal flow: **1**. Include <<Search bus trip>> and <<Choose seat number>> **2.** The actor chooses **Book ticket** function. **3.** The system checks if the actor has chosen any seat numbers from a bus trip. If yes then go to Step 4; else display error message. **4.** The system displays the detail for the tickets including the following information: Bus trip information (Bus trip origin and destination, Departure date, Number of chosen seats, Chosen seat number list and Total price) and Actor information (Full name, Phone number and Email). **5.** The actor confirms the bill detail. **6.** The system books the tickets for the actor and announces that it has booked successfully.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm the bill detail, the system will remain the state before that. If the system can't book the tickets for the actor or can't get bus trip information & actor information from the database, it will display error message. The actor can also enter online payment function (<<Online payment>>) for the booked tickets after Step 6.	

3.11. View booked ticket

Function: View booked ticket	ID: FN-11
Main actor: Customer, Admin	Priority: Essential
Brief description: Allow the actor to view booked ticket if the actor is Customer or customer's booked ticket selected from <<Select customer account>> if the actor is Admin.	
Normal flow: 1. Include <<Select customer account>> if the actor is Admin. **2.** The actor chooses **View booked ticket** function. **3.** The system displays the list of booked tickets. Each booked ticket includes the following information: Ticket id, origin and destination of that bus trip, seat number, ticket booked date, some relevant bus trip information like bus trip type, departure date, arrival date and some functions such as <<View bus trip schedule>> , <<Online payment>>.Moreover, if the actor is Admin, the system will display the following addition functions: <<Change seat number of booked ticket>> and <<Delete booked ticket>>.	
Sub flows: (None)	
Alternate/Exceptional flows: If the list of booked tickets is empty, the system will announce that no result found. If the system can't get information related to booked ticket, it will display error message.	

3.12. Online payment

Function: Online payment	ID: FN-12
Main actor: Customer	Priority: Essential
Brief description: Allow the actor make online payment for the tickets through VNPAY.	
Normal flow: **1.** The actor inputs the data of the transaction for the booked tickets including the amount of money transferred and payment method. **2.** The system starts and processes the transaction through VNPAY.	

3. The actor provides some involved information for the transaction such as card number, card holder, etc. and confirms it. **4.** When the transaction is done, VNPAY informs the system to update the payment status for those tickets.
Sub flows: (None)
Alternate/Exceptional flows: If the actor inputs invalid inputs at Step 1 or the system can't start the transaction through VNPAY at Step 2, the system will display error message. If the transaction is not finished at Step 4, no changes will be made in the system.

3.13. Search customer account

Function: Search customer account	ID: FN-13
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to search customer account by phone number	
Normal flow: **1**. The actor chooses **Search customer account** function. **2.** The system asks the actor to enter customer phone number. **3.** The actor confirms the inputs. **4.** The system checks if the phone number is exists in the database. If yes then go to Step 5; else announce that no result found. **5.** The system displays list of functions for the actor to choose including: <<View account information>>, <<Reset password>>, <<View booked ticket>> and <<Delete account>>.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm the inputs, the system will remain the state before that If the actor provides an empty input, the system will display error. If the system can't get the phone number in the database, it will display error.	

3.14. Reset password

Function: Reset password	ID: FN-14
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to reset customer account password	
Normal flow: **1**. Include <<Search customer account>> **2.** The actor chooses **Reset password** function. **3.** The system asks the actor to confirm the action. **4.** The actor confirms. **5.** The system generates a new random password, replaces old password with the new one and displays it to the actor.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm, the system will not generate new password. If the system fails to do any task in Step 5, it will display error message.	

3.15. Delete account

Function: Delete account	ID: FN-15
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to delete customer account	
Normal flow: **1**. Include <<Search customer account>> **2.** The actor chooses **Delete account** function. **3.** The system asks the actor to confirm the action. **4.** The actor confirms. **5.** The system deletes all information related to the account including: Account information and Booked tickets by that account. **6.** The system announces that it has deleted account successfully.	
Sub flows: (None)	

Alternate/Exceptional flows: If the actor does not confirm, the system will not delete account. If the system fails to delete any information related to that account, it will display error message.

3.16. Change seat number

Function: Change seat number	ID: FN-16
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to change seat number of the specific booked ticket.	
Normal flow: **1**. Include <<View booked ticket>> **2.** The actor chooses **Change seat number** function of a specific ticket. **3.** The system asks the actor to enter the new seat number. **4.** The actor confirms the input. **5.** The system changes the seat number of that booked ticket to the new one and announces that it has changed seat number successfully.	
Sub flows: (None)	
Alternate/Exceptional flows: If the actor does not confirm, the system will remain the state before that. If system can't change the seat number to the new one, it will display error message.	

3.17. Delete booked ticket

Function: Delete booked ticket	ID: FN-17
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to delete customer booked ticket.	
Normal flow: **1**. Include <<View booked ticket>> **2.** The actor chooses **Delete booked ticket** function of a specific ticket.	

3. The system asks the actor to confirm the action. **4.** The actor confirms. **5.** The system deletes the booked ticket from the database and announces that it has deleted booked ticket successfully.
Sub flows: (None)
Alternate/Exceptional flows: If the actor does not confirm, the system will remain the state before that. If system can't delete booked ticket, it will display error message.

3.18. View the statistics

Function: View the statistics	ID: FN-18
Main actor: Admin	Priority: Essential
Brief description: Allow the actor to view the statistics of booked tickets	
Normal flow: **1**. The actor chooses **View the statistics** function. **2.** The system asks the actor which statistic will be made. **3.** The actor chooses a statistic option as well as the filters for the statistic. **4.** The system displays some graphs along with some numeric data related to number of booked tickets and total revenue.	
Sub flows: (None)	
Alternate/Exceptional flow: If the system can't display the requested statistic, it will display error message.	

CHAPTER 5: SYSTEM IMPLEMENTATION

In this chapter, we will describe in details how we implement the system by using available tools, software and services as we mentioned in Chapter 2.

When first opening the application, the users are required to log in so that they can use the booked ticket system as showed in Figure 6. If the users are *customers,* they must enter the phone number that has been registered to the system (if not have one, they can create a new account as showed in Figure 6). In the case of *admins*, they must enter the user name given by their managers. Moreover, there is an option for concealing or revealing the password characters. Once they have gained access to the system, the system will display the main menu interface according to their privilege.

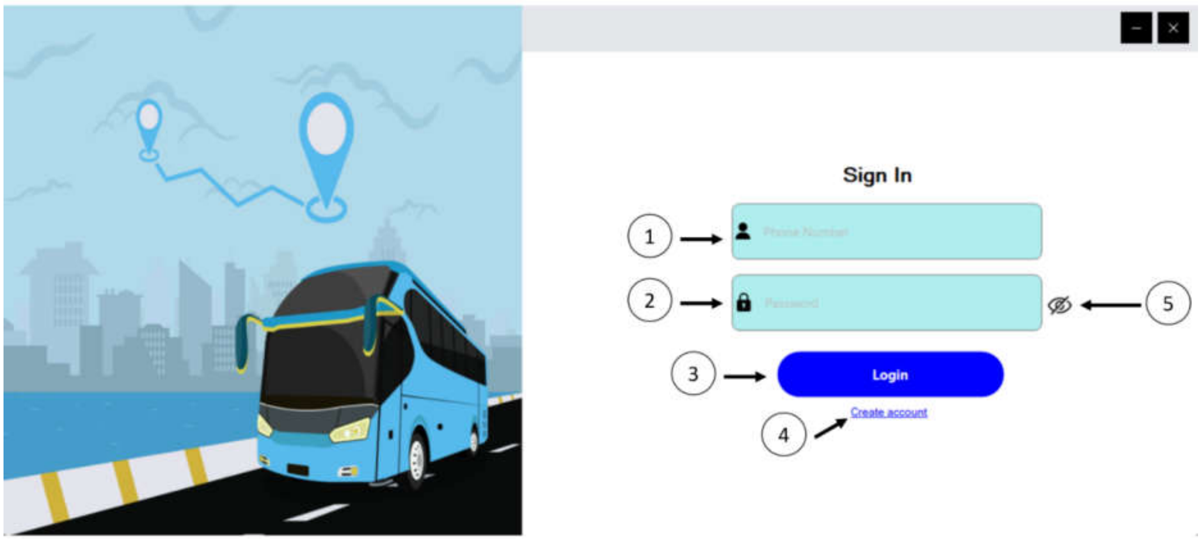


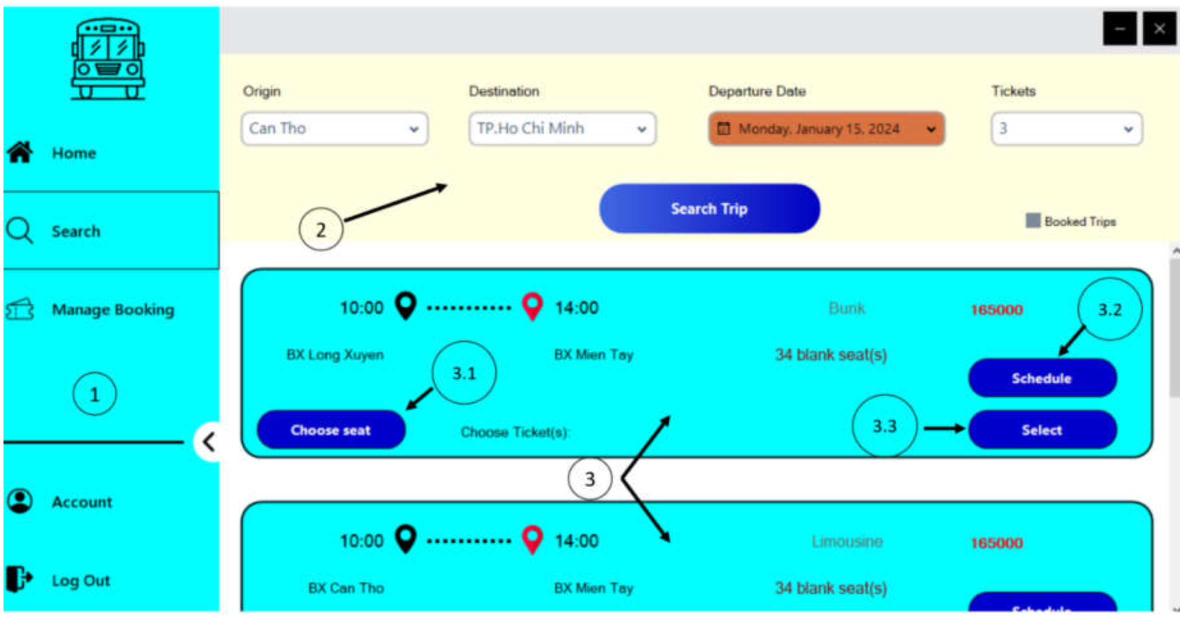
Figure 6. Login interface

1. User name, 2. Password, 3. Login Button, 4. Create Account, 5. Hide/Unhide password option

1. Customer

The customers can access various function of the system from the main menu including: search bus trip, manage booking, manage account, etc. Now we will demonstrate the scenario of booking tickets and managing them.

From the main menu, the customers can select the option Search from side menu bar on the left to open the search bus trip function (Figure 7). Then they are obligatory to fill the origin, destination, departure date as well as the number of tickets they want to book. Then the system will display list of bus trips that match the input condition (Figure 7).


Origin: Can Tho

Destination: TP.Ho Chi Minh

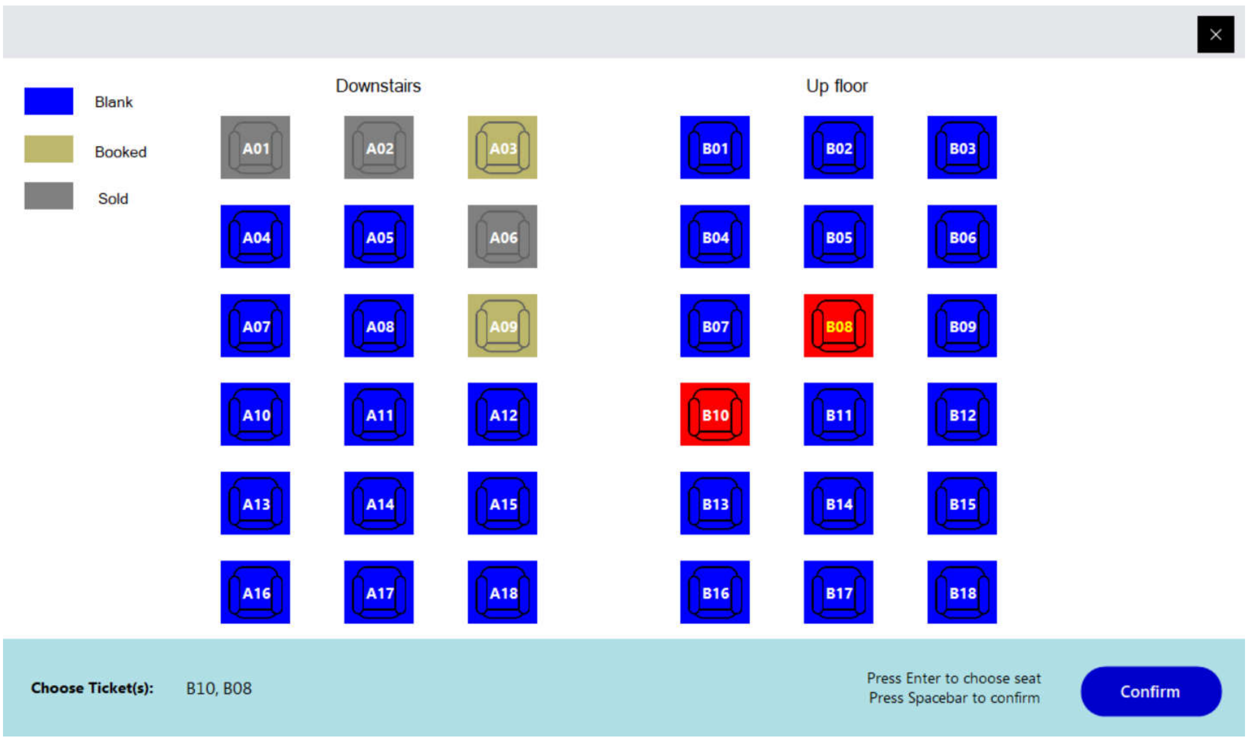
Departure Date: Monday, January 15, 2024

Tickets: 3

Figure 7. Search bus trip interface

1. Side menu bar with options such as Home, Search Bus Trip, Ticket Management, Account Management and Log Out; 2. The filter for searching bus trip including origin location, destination location, departure date, number of tickets and the option for including already booked trips or not; 3. Example records of bus trip found consists of: 3.1. Choose Seat button, 3.2. Schedule button, 3.3. Select button and other information such as the expected departure and arrival time, name of bus stops, available blank seats, etc.

After that, they can click on **Choose Seat** button (Figure 7) to open the layout of seat numbers within a bus trip. From then, they can choose a certain of seat numbers that are not currently pending or sold (Figure 8). Once they have done with the seat numbers selection, they can proceed to book the required seat numbers by clicking on **Select** button in Search bus trip interface (Figure 7). Each seat number is assigned to each unique ticket generated by the system. Additionally, after booking the tickets, the customers have an option to make payment for booked tickets. The payment implementation scenario will be explained in depth in Section 3 Chapter 5. When they have finished the booking, they can go to tickets managements to access some functions for booked tickets as shown in Figure 9.

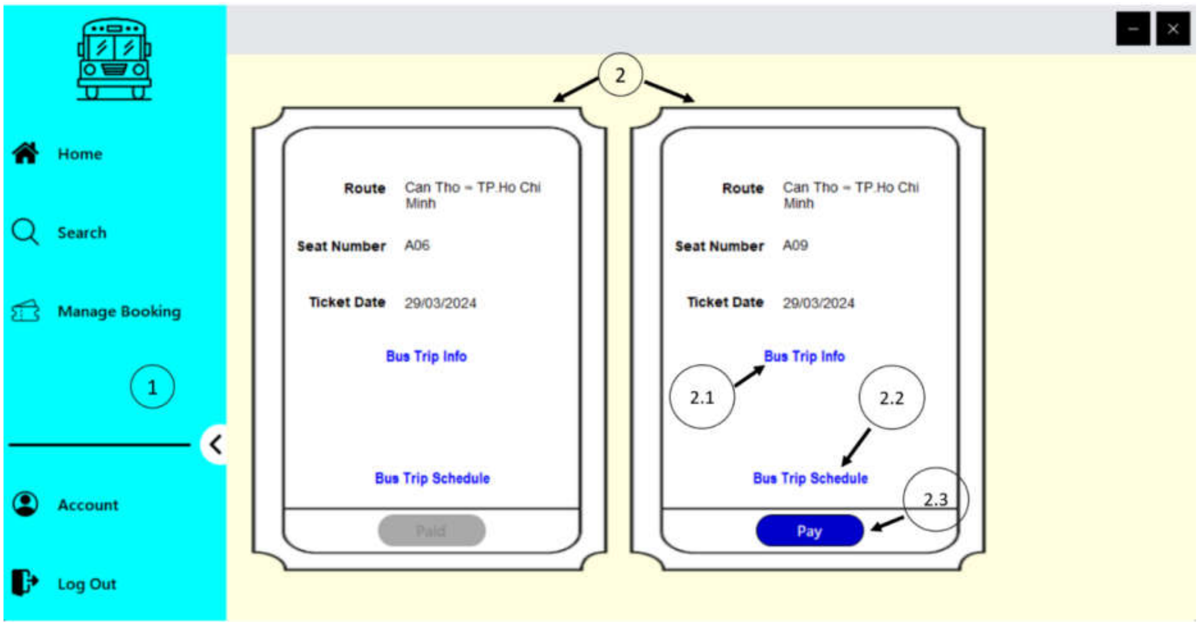


Choose Ticket(s): B10, B08

Press Enter to choose seat
Press Spacebar to confirm

Figure 8. Seat layout interface of a bus trip

An example seat layout of bus trip type "Bunk" with sold seats (A01, A02 and A06) indicated by the gray boxes, booked/pending seats (A03 and A09) illustrated by the dark khaki boxes, current chosen seats marked by red boxes, and the empty seats as blue ones.



Route	Can Tho – TP.Ho Chi Minh
Seat Number	A06
Ticket Date	29/03/2024

Route	Can Tho – TP.Ho Chi Minh
Seat Number	A09
Ticket Date	29/03/2024

Figure 9. Ticket management interface

1. Side menu bar; 2. An example of booked tickets with the left ticket has been paid, the right one has not been paid yet; booked tickets contains the following information: 2.1. Bus trip information, 2.2 Bus trip schedule, 2.3. Pay button.

2. Admin

Once the users have log on to the system using admin privileges, there are two basic tools that they can use. The first one is Customer Service function, which helps them to manage customer accounts in some aspects as demonstrated in Figure 10. The other one is Statistics, which enables them to make some statistical results related to bus tickets. In the rest of this section, we will go through the scenarios of changing seat number of a ticket and making statistics.

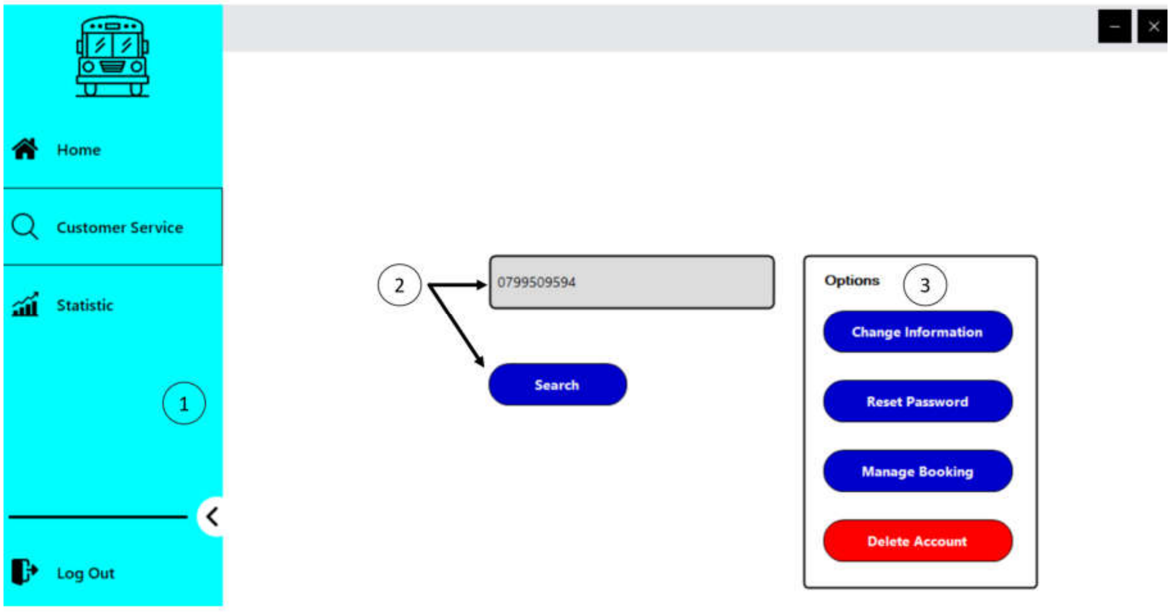


Figure 10. Customer Service interface

1. Side menu bar with options: Home, Customer Service, Statistics and Log Out; 2. A text box and a Search button for admins to search customers' phone number; 3. Multiple available options for admins such as change customer account information, reset customer account password, manage booked tickets and delete customer account.

After opening the Customer Service interface, admins can enter customer phone number, which has been registered to the system. From then, they can click on Manage Booking button in Figure 10 to open the management booked tickets interface as shown in Figure 11.

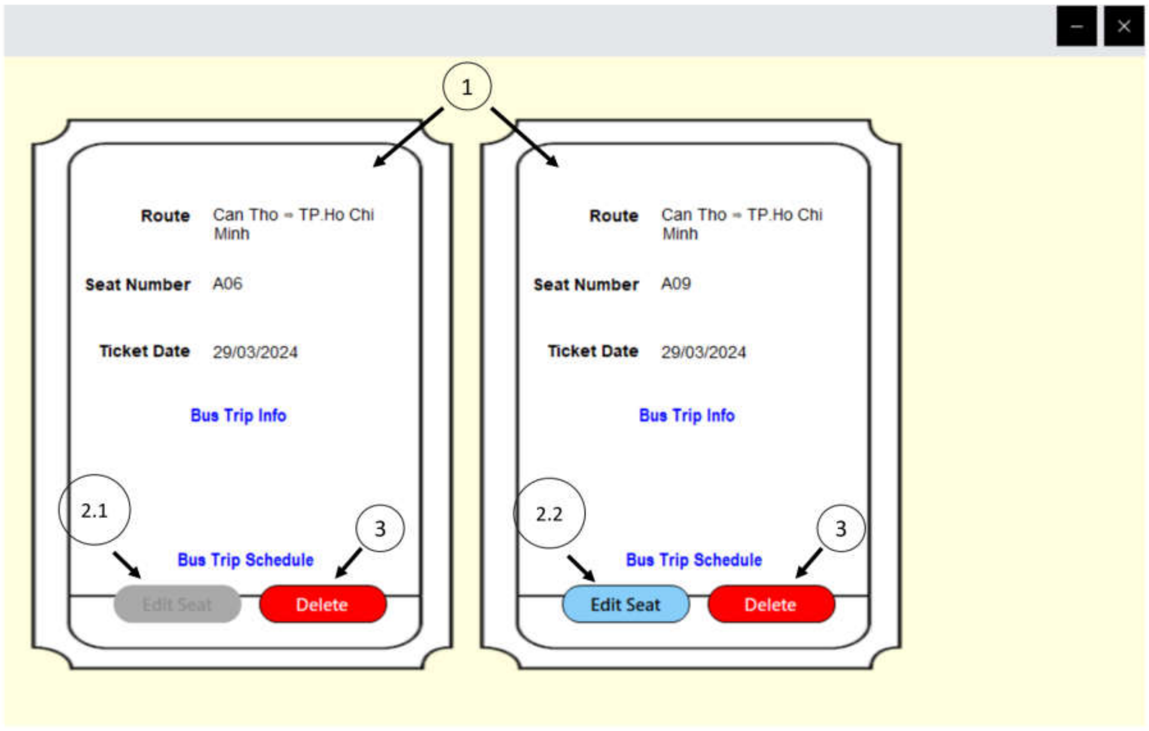


Figure 11. Customer Booked Tickets Management interface

1. An example booked tickets of customer with the left one has already paid by the customer and the right one is booked but not paid yet; 2.1. Edit Seat button in disable state; 2.2. Edit Seat button; 3. Delete Ticket button.

In this interface, beside some basic ticket information mentioned in the scenario in Section 1 Chapter 5, admins are able to delete tickets if the customer wants to cancel their booking tickets and change seat number using the Edit Seat button (paid tickets are not allowed to change). When clicking on that button, a new interface appears (Figure 12) for the admins to modify the seat number to the new one if available.

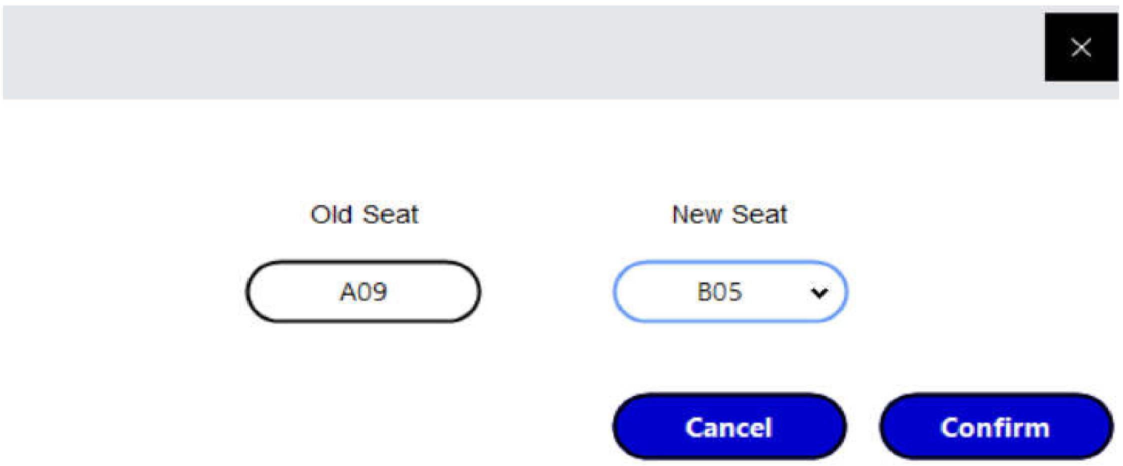


Figure 12. Change Seat Number interface

On the other hand, admins can have access to statistics function to choose which statistics to be made. They can choose some statistical options such as booked tickets by province and by bus stop (Figure 13). After that, they may choose some filters for the statistics. The system will display graphs as well as some statistical results (tickets sales, revenue, etc.) dynamically according to the filters they chosen.

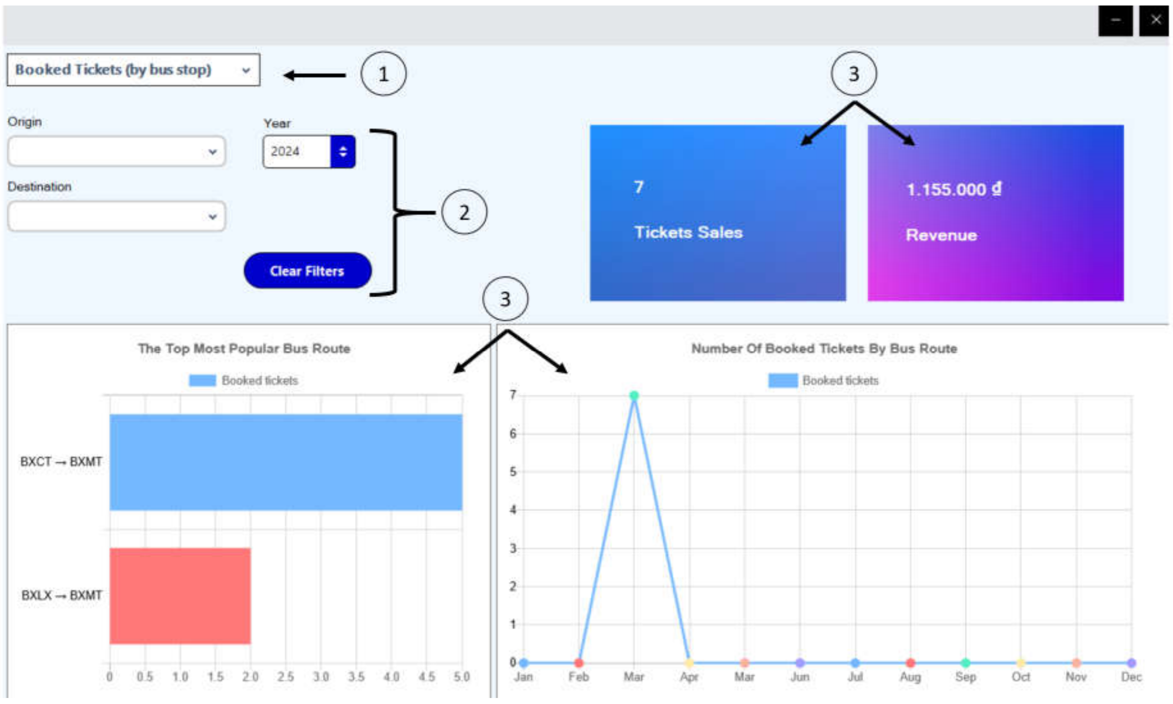


Figure 13. Statistics interface

1. Combo box to choose statistics options; 2. Filters for the statistics; 3. Some graphs and numeric results for the statistics.

3. Payment

In this section, we will describe the scenario of making a payment for one ticket (this is also applied for paying multiple tickets). From the ticket management interface (Figure 14), the customers can click on the Pay button to make payment for that ticket (e.g. ticket A05).

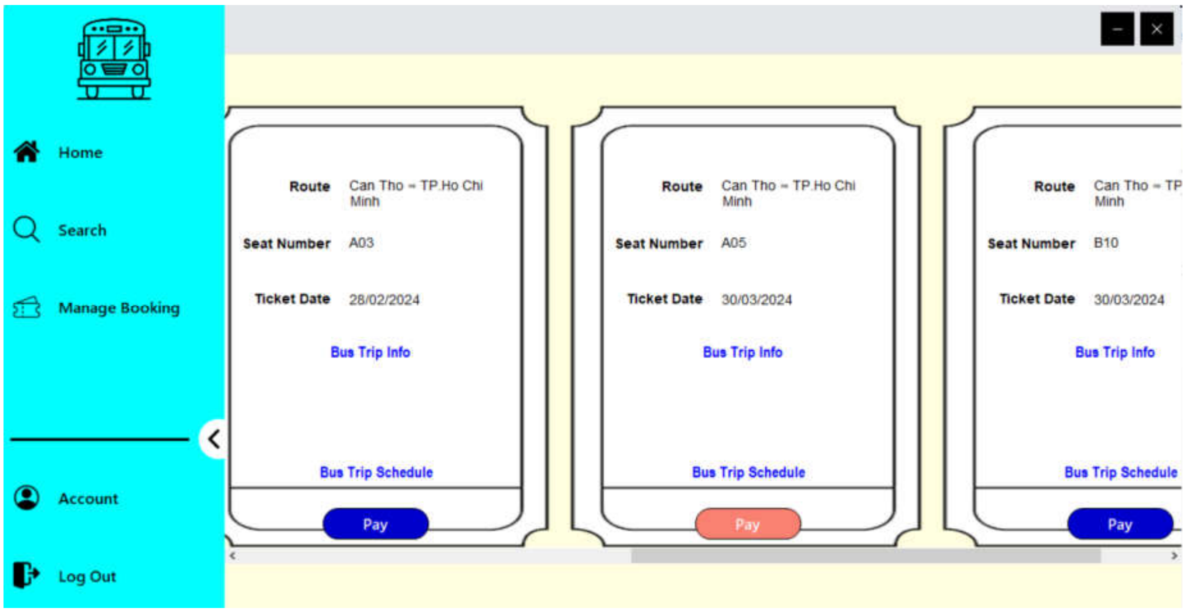
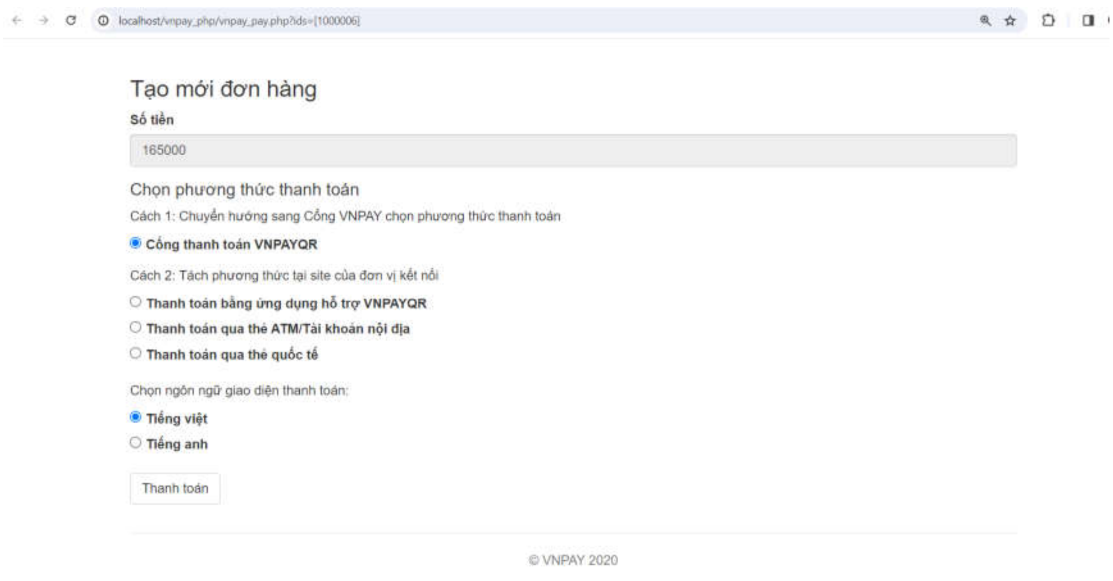


Figure 14. Ticket Management interface (before payment)

Then the system will open a local webserver using Apache. The customers can choose which payment method will be used. In this example, we will choose the method of directing to VNPAYQR payment gateway.



localhost/vnpay_php/vnpay_pay.php?ids=[1000006]

Tạo mới đơn hàng

Số tiền

165000

Chọn phương thức thanh toán

Cách 1: Chuyển hướng sang Cổng VNPAY chọn phương thức thanh toán

[✓ **Cổng thanh toán VNPAYQR**]

Cách 2: Tách phương thức tại site của đơn vị kết nối

Thanh toán bằng ứng dụng hỗ trợ VNPAYQR

Thanh toán qua thẻ ATM/Tài khoản nội địa

Thanh toán qua thẻ quốc tế

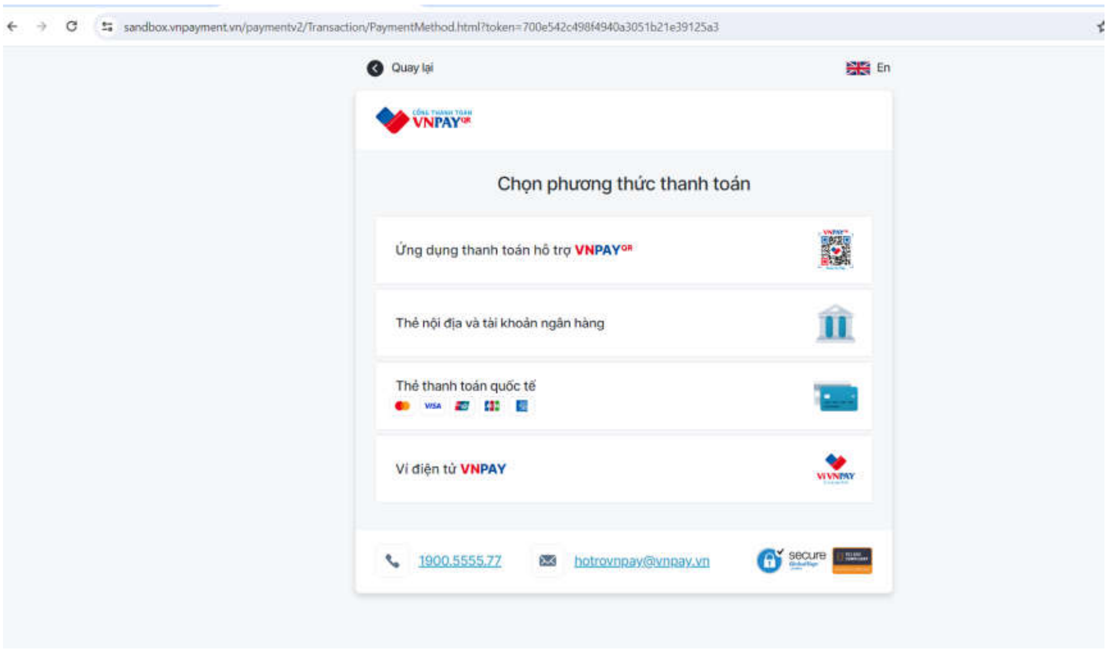
Chọn ngôn ngữ giao diện thanh toán:

[✓ **Tiếng việt**]

Tiếng anh

Figure 15. Payment method options in local webserver

After directing to VNPAYQR payment gateway, customers can proceed to make payment by using various methods consisting of VNPAYQR scan method, domestic card and bank account, international payment cards or VNPAY E-Wallet as shown in Figure 16.



sandbox.vnpayment.vn/paymentv2/Transaction/PaymentMethod.html?token=700e542c498f4940a3051b21e39125a3

Chọn phương thức thanh toán

Ứng dụng thanh toán hỗ trợ **VNPAY**QR

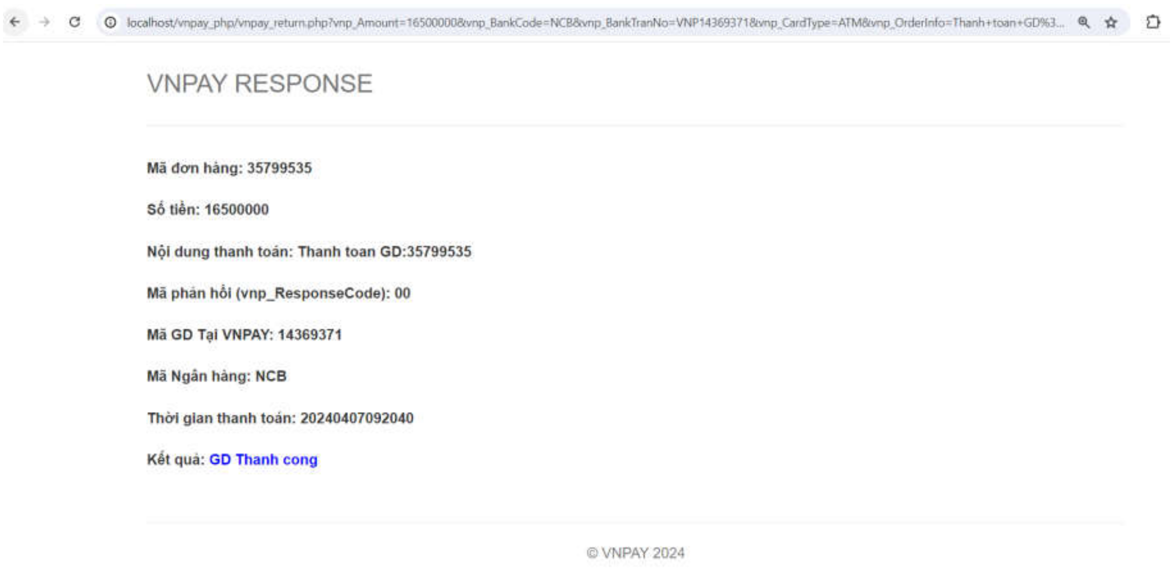
Thẻ nội địa và tài khoản ngân hàng

Thẻ thanh toán quốc tế

Ví điện tử **VNPAY**

Figure 16. VNPAYQR payment gateway

When completing the payment, an announcement will appear at the local webserver to notify the customers that the payment is successful or not (Figure 17). Additionally, they can also refresh the ticket management interface of the system to update the payment status of that ticket (Figure 18). At the time when the transaction is completed, an Instant Payment Notification, also known as IPN [4], is made to update the transaction response code to the database so that the system can adjust the ticket's payment status according to that transaction. We need to use third party software such as Ngrok [5] to expose our local webserver to the Internet in order to make server-to-server communication with the VNPAYQR.



localhost/vnpay_php/vnpay_return.php?vnp_Amount=1650000&vnp_BankCode=NCB&vnp_BankTranNo=VNP14369371&vnp_CardType=ATM&vnp_OrderInfo=Thanh+toan+GD%3...

VNPAY RESPONSE

Mã đơn hàng: 35799535

Số tiền: 16500000

Nội dung thanh toán: Thanh toan GD:35799535

Mã phản hồi (vnp_ResponseCode): 00

Mã GD Tại VNPAY: 14369371

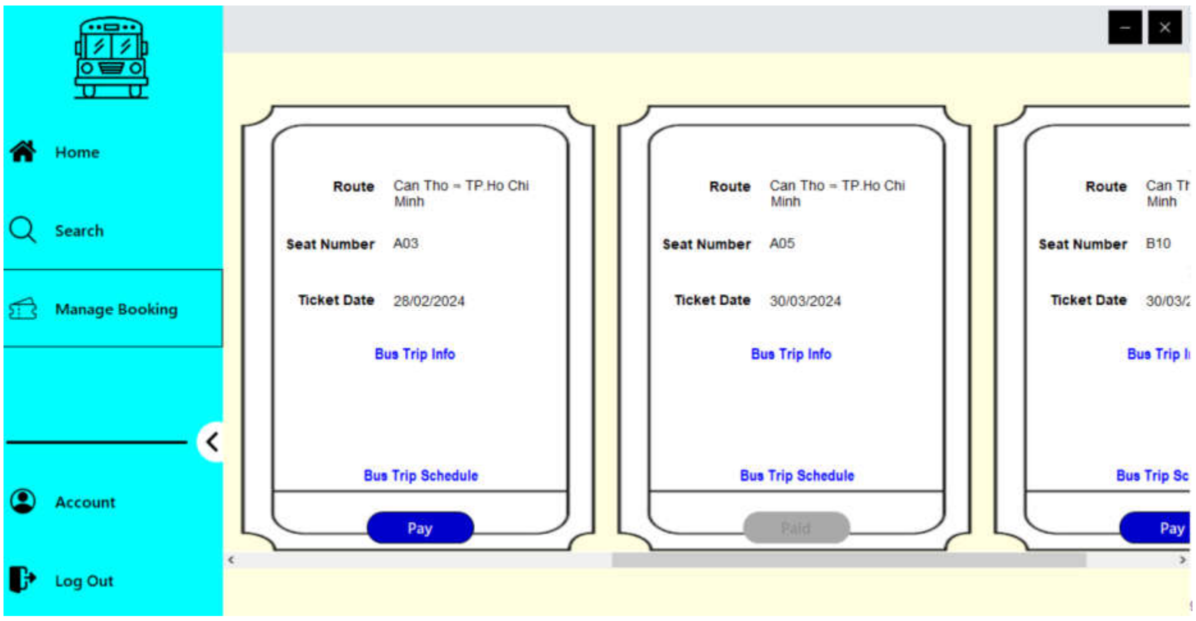
Mã Ngân hàng: NCB

Thời gian thanh toán: 20240407092040

Kết quả: GD Thanh cong

© VNPAY 2024

Figure 17. A successful payment announced in local webserver



Route Can Tho - TP.Ho Chi Minh
Seat Number A03
Ticket Date 28/02/2024
Bus Trip Info
Bus Trip Schedule
Pay

Route Can Tho - TP.Ho Chi Minh
Seat Number A05
Ticket Date 30/03/2024
Bus Trip Info
Bus Trip Schedule
Paid

Route Can Th
Minh
Seat Number B10
Ticket Date 30/03/2
Bus Trip I
Bus Trip Sc
Pay

Figure 18. Ticket Management interface (after payment)

CHAPTER 6: SYSTEM TESTING

1. Purpose

To ensure the validation and verification of functionalities as well as effectiveness of the developed booking bus ticket system, various testing and evaluation are conducted.

2. Scope

The test includes unit testing, integration testing and system testing. The test covers multiple scenarios of booking bus ticket such as bus trip searching, seat selection, payment processing and administrative functions. Moreover, usability testing is also performed to ensure the user-friendliness and ease of navigation throughout the interfaces.

3. Evaluation

System testing confirms the functionalities of booking bus ticket system, with all essential features performed as intended.

CHAPTER 7: CONCLUSION

1. Results

After researching and conducting the project, we have implemented the system with several functionalities related to customers and administrators for booking bus tickets. However, there are some minor features that we have not fulfill yet such as printing function (tickets, banking invoice, etc.), responsive interface (adjusting layout according to the device and different size), custom settings (theme, font, language, etc.) and so on.

2. Future Work

Firstly, to address these disadvantages and further enhance the booking bus ticket system, we need to make an effort in designing and improving compatibility and responsiveness of the system. Secondly, it is necessary to implement printing function supporting wide range of format options. Besides, gathering continuous user feedbacks and conducting regular usability testing to customize the user experience. Last but not least, exploring the integration of emerging technologies such as artificial intelligence and block chain can offer enhanced security and convenient for customers and administrators.

REFERENCES

[1] "Introduction to Windows Forms" (Visual Studio 2003 documentation). Microsoft 2003.

[2] https://dev.mysql.com/doc/refman/8.3/en/what-is-mysql.html

[3] https://httpd.apache.org/docs/

[4] https://en.wikipedia.org/wiki/Instant_payment_notification

[5] https://ngrok.com/docs/