**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY FOR HIGH-QUALITY TRAINING**

**COURSE NAME: Database Management System**

**🙡🙠✵🙢🙣**



**FINAL PROJECT REPORT**

**Project name:**

BUS TICKET BOOKING MANAGEMENT SYSTEM

**Lecturer:** Prof. Nguyễn Thành Sơn

**Course ID:** DBMS330284E\_22\_2\_01FIE

**Group:** 6

**Date:** 2nd Sem/2022-2023

*Ho Chi Minh city, May, 2023*

**LIST OF STUDENTS – GROUP 6**

**Project: Bus ticket booking management system**

|  |  |
| --- | --- |
| ***ID*** | ***Full name*** |
| 21110758 | Lê Xuân Cường |
| 21110092 | Bùi Quốc Thông |
| 21110066 | Phạm Vũ Bảo Nhân |
| 21110785 | Mai Nguyễn Nhật Nam |

**Professor’s comment**

Ho Chi Minh city, May …, 2023

Grading

**Contents**

[Prologue 4](#_heading=h.gjdgxs)

[INTRODUCTION 5](#_heading=h.30j0zll)

[CHAPTER I: SYSTEM OVERVIEW 6](#_heading=h.1fob9te)

[1.](#_heading=h.3znysh7) Specifications 6

[1.1.](#_heading=h.1y810tw) Problem statement 6

[1.2.](#_heading=h.2et92p0) Overview 6

[1.3.](#_heading=h.tyjcwt) Problem process 8

[1.4.](#_heading=h.3dy6vkm) Main functions 9

[1.5.](#_heading=h.1t3h5sf) Attributes and operations reference 10

[CHAPTER 2: SYSTEM ANALYSIS AND DESIGN 14](#_heading=h.4d34og8)

[1.](#_heading=h.2s8eyo1) Relational database design 14

[2.](#_heading=h.17dp8vu) Logical level database design 14

[3.](#_heading=h.3rdcrjn) Required constraints 14

[4.](#_heading=h.26in1rg) Database settings and constraints 14

[5.](#_heading=h.lnxbz9) Other constraints 14

[6.](#_heading=h.35nkun2) Trigger to check for constraints 14

[CHAPTER 3: DESIGNING FUNCTIONS 15](#_heading=h.1ksv4uv)

[1.](#_heading=h.ihv636) Connect to database 15

# Prologue

Firstly, we would like to express our gratitude to Prof. Nguyễn Thành Sơn for his whole-hearted instructions that helped us finish our final project for the Database Management System course. Thanks to the knowledge the professor has provided us, we were able to firmly grasp the basic knowledge and foundation for building a database management system. And through this project, our group would like to present the development process of a database management system and demonstrate by programming a related project once again.

During the process of executing this project, it will be hard to avoid mistakes. Because of that, we would love to get the professor’s suggestion on improving our work so it would be more functional and complete. We wish you good health and the best of luck pursuing the path of teaching.

Finally, we would like to thank all the teachers and classmates who studied with us on this course and offered us support while we carried out our final project.

# INTRODUCTION

In recent years, the Information and Technology (IT) area has been integrated into our society and daily lives, regardless of any field and/or occupations. It also plays an important part of booking management in Vietnam and especially in almost every country as there are many applications made to help fix problems that big organizations frequently face.

The creation of the bus ticket booking management system is the result of many developers’ creativity and hard work with the aim of aiding companies in managing their businesses.

With that in mind, to better understand the application and role of Information and Technology (IT) in Database Management, we have decided on the **“Bus ticket booking management system”** as our final project.

# CHAPTER I: SYSTEM OVERVIEW

## Specifications

### Problem statement

The bus ticket booking management system will:

* Manage the employees, passengers, bus, trips, routes easier.
* More convenient for users to check, book the trip
* Check the state and location of the trip more clearly through the map.
* More convenient for the car brand to statistic: revenue, number of passengers, number of trips, salary of employee, outcome, etc. per day, per month, per year.

Vehicle management: Managing travel vehicles including their location, date and time of arrival/leave, pricing, etc.

System management: Managing employees, drivers, customers, travel curriculum.

Statistics: Employee statistics, vehicle statistics, daily sales, etc.

### Overview

A car branch needs to have a bus ticket reservation system. The bus ticket reservation system should contain the following data:

The car branch manages a lot of agents. Each agent has: agent ID,cash reserve ID, address, agent names.

Each agent has only one cash reserve. Cash reserve include: cash reserve ID, counter.

An agent can have many employees. Each employee has: employee ID, position ID, account ID, agent ID, name, address, phone number, identity number, salary, email, birth. Each employee is provided an account (username and password) . Each employee has a different position and power.

The information of the position group contains: position ID, type.

There are several types:

* + Travel planner
  + Travel supervisor
  + Driver
  + Ticket seller
  + Service guide
  + Security guard
  + Porter

Each position group has separate privileges. The information of the privileges group includes: privilege ID, name.

The agent manages passengers. Each passenger has: passenger ID, name, phone number, address, identity number, gender, email.

The gender attribute of passenger above has two options:

* Male
* Female

Easily manage and filter the address of stations in the general local area, there is information of places: place ID, region.

Each passenger can choose a pickup station and drop-off station. Each station has: station ID, detail address, name, capacity, parked bus number.

The bus of each branch has: bus ID, registration number, model, capacity.

Routes involving the journey have: route ID, start bus station ID, final bus station ID, travel distance.

Each trip is set up by the travel planner which includes: trip ID, drivers ID, bus ID, route ID, departure time, duration, number of booked seat, direction, state.

The direction attribute of trip above has two options:

* Go
* Return

The state attribute of trip above has three options:

* Waiting
* Going
* Finish

The drivers ID in the trip relation is an attribute of trip\_drivers relation: trip ID, driver ID. In those, driver ID is a multivalued attribute.

The agent distributes tickets to the passenger. Each ticket has: ticket ID, trip ID, passenger ID, fare, type, seat number.

The type of ticket has two options:

* Seat ticket
* Sleeper ticket

The agent manages the booking transaction. Each booking transaction includes: transaction ID, ticket ID, passenger ID, employee ID, booking time.

Each employee can take on more than 1 position.

Each passenger can book more than 1 ticket.

Each trip can have more than 1 driver.

Each route is bi-directional. Determine the direction of a route based on the direction attribute of the trip.

### Problem process

Booking period:

The service guide records the passenger’s full field information including: their ID number, phone number, address, gender, email. Then, the ticket seller checks again to guarantee all the required fields are correctly fielded.

Then, the passenger picks a trip by choosing from multiple options: destination, pickup station, drop-off station, departure time, the available seat, ticket type. Options will be planned by the travel planner, so the passenger must follow this template.

Then, the ticket seller verifies the customer’s selection. If valid, the ticket seller informs the passenger and waits for their confirmation. If they confirm, the ticket seller prints the ticket, gives it to them and reminds them to arrive at the correct time on the ticket. Else if they refuse, the customer needs to modify the information.

Departure period:

The passengers wait in the agent. 15 minutes before departure time of the trip, the vehicle will take the passengers to the bus station.

At the bus station, the porter put the passengers’ luggage into the trunk.

When it’s time, the service guide instructs passengers to the vehicle, and provides water and tissues to them.

Drop-off period:

When the bus arrives at the last bus station, the porter takes passengers’ luggage of the bus and gives it to the passenger.

### Main functions

Administrator (global):

* Add, modify, delete, authorize for positions
* Add, delete the position
* Add, modify, delete employees of the position

Travel planner:

* Add, modify, delete trips
* Add, modify, delete routes
* Add (distribute the ticket of the trip), modify, delete tickets

Travel supervisor:

* Add, delete passenger of the trip
* Report errors (trip, route, passenger, booking)

Ticket-selling:

* Add, modify, delete passenger
* Export bill
* Export ticket
* Change the state attribute of trip

\*Authorization:

* admin: full control on the whole system - global privilege
* […] (other privileges): local privilege

### Attributes and operations reference

Bus relation:

\*ID attribute:

*Format:*

bu[model]\_[number]

Example: **bu5272f29s28\_5**

* model: 5272f29s28
* number: 5

In those:

* model: Model of the bus
* number: Order of the bus

\*Capacity attribute of bus relation is fixed (default of model).

Trip relation:

\*Trip attribute:

*Format:*

tr[id\_route]\_[departure\_time]

Example: **trr78\_20230918**

* id\_route: r78
* departure\_time: 2023/09/18

In those:

* id\_route: ID of route in route relation
* departure\_time: Written in continuous form (No special characters)

\*Duration attribute is calculated using the following formula:

Duration = distance / average\_speed

In those:

* distance: The distance attribute of route relation
* average\_speed: The moving value is statistic by system

\*booked\_seat attribute: Updated after every insert or delete statement of the passenger of the trip. It must satisfy this rule:

booked\_seat ≤ capacity – inherent\_seat

In those:

* capacity: Get from the bus relation
* inherent\_seat: This value is set by the planner or ticket seller

Ticket relation:

\*ID attribute:

*Format:*

ti[type]\_[seat\_number]

Example: **tiseat\_a14**

* type: seat
* seat\_number: a14

In those:

* type: Ticket type
* seat\_number: Number of tickets

\*Fare attribute: Set by the travel planner.

\*Seat number attribute has a limit: 0 < seat\_number < 15.

*Format:*

{[A | B | C]} {[1 | 2 | ...]}

In those:

* A: Floor 1
* B: Floor 2
* C: Rear seats
* 1, 2, …, 5: column 1 (after driver’s seat)
* 6, 7, …,10: column 2
* 11, 12, …, 15: column 3
* For C (rear seat): 1,2, …, 5 (left to right)

Route relation:

\*ID attribute:

*Format:*

r[route\_number]

Example: **r78**

* route\_number: 78

In those:

* route\_number: The route number

Booking relation:

\*ID attribute:

*Format:*

b[booking\_time]

Example: **b20230719**

* booking\_time: 2023/07/19

In those:

* booking\_time: The booking time attribute of this relation. Written in continuous form (No special characters)

\*Booking\_time attribute: Written in GMT+7 format.

Bus\_station relation:

\*Cur\_bus attribute: Updated after the related trip state change.

Passenger relation:

\*Phone\_number attribute: 10-11-digits length.

System\_account relation:

\*User\_name attribute:

*Format:*

admin: sysad\_[optional\_name]  
system employees: sys\_[optional\_name]

[ ] : Ignore this notation

Cash\_reserve relation:

\*Counter: Updated after a transaction (booking) occurs.

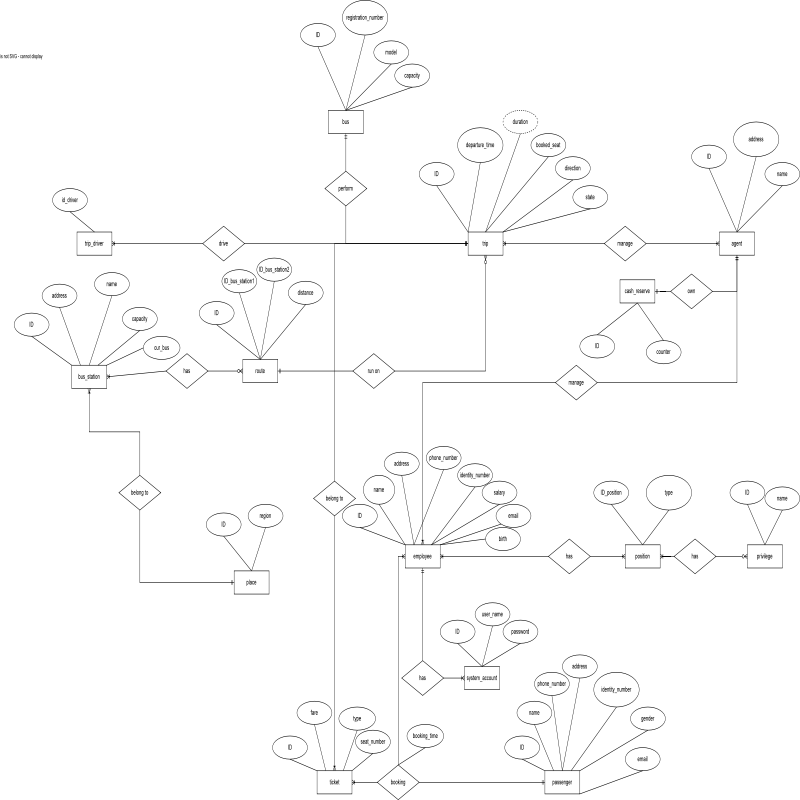
Trip\_driver relation:

\*Id\_driver attribute is reference to the id\_employee of employee relation where position is driver.

# CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

## Conceptual level database design

From the needed data in description of problem, the following Entity Relationship Model (ERD) is formed.



Sharp image: [ERD sharp image (busticketbookingerd.netlify.app)](https://busticketbookingerd.netlify.app/)

## Logical level database design

## Required constraints

|  |  |  |
| --- | --- | --- |
| No | Attribute name | Constraint |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Database settings and constraints

## Other constraints

## Trigger to check for constraints

# CHAPTER 3: DESIGNING FUNCTIONS

## Connect to database