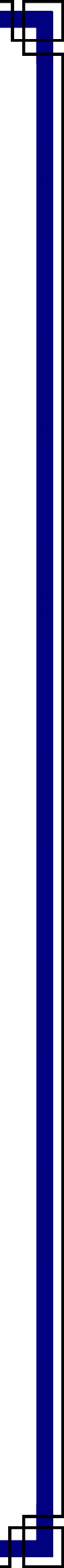
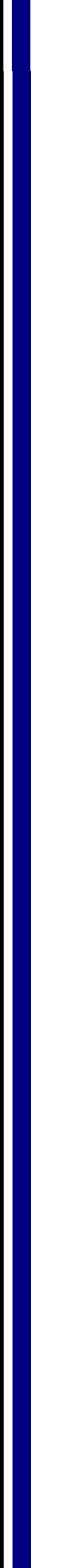
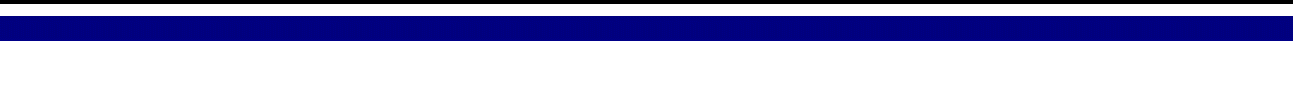
**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**



**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**Course name: IT PROJECT**

**CLOSED VEHICLES CARE SYSTEM**

**Lecturer: PhD.Le Van Vinh**

**Class: 221PROJ215879E**

**Members:**

Nguyen Minh Tri – 20110422

Nguyễn Thanh Toan – 20110012

***Thu Duc city, December 4th, 2022***

**CONTENTS**

[INTRODUCTION 1](#_Toc120899154)

[TASK ASSIGNMENT 3](#_Toc120899155)

[I. INTRODUCTION TO THE PROBLEM 5](#_Toc120899156)

[1. Survey and analysis of the current situation 5](#_Toc120899157)

[2. Problem 5](#_Toc120899158)

[II. DATABASE DESIGN 7](#_Toc120899159)

[1. ERD 7](#_Toc120899160)

[2. Overview 7](#_Toc120899161)

[2.1. NHANVIEN Table 8](#_Toc120899162)

[2.2. VE table 9](#_Toc120899163)

[2.3. BAIXE table 9](#_Toc120899164)

[2.4. XE table 10](#_Toc120899165)

[2.5. THONGKE table 11](#_Toc120899166)

[2.6. CHIACA table 12](#_Toc120899167)

[2.7. THO table 13](#_Toc120899168)

[2.8. TIMELAMVIEC table 14](#_Toc120899169)

[III. DESIGN PROGRAM 15](#_Toc120899170)

[1. Overview of the program 15](#_Toc120899171)

[1.1. GUI layer: 15](#_Toc120899172)

[1.2. DAO layer: 16](#_Toc120899173)

[1.3. DTO layer 18](#_Toc120899174)

[2. Program interface and functions 18](#_Toc120899175)

[2.1. Login function 18](#_Toc120899176)

[2.2. Forget password fuction 19](#_Toc120899177)

[2.3. Account categories 21](#_Toc120899178)

[2.4. Management Directory 26](#_Toc120899179)

[2.5. Statistical List 33](#_Toc120899180)

[2.6. Help List 36](#_Toc120899181)

[IV. ASSESSMENT OF THE PROGRAM 37](#_Toc120899182)

[1. Advantages 37](#_Toc120899183)

[2. Disadvantages 37](#_Toc120899184)

[OVERALL 38](#_Toc120899185)

INTRODUCTION

Today, with the rapid development of information technology, it has become an indispensable part of human life. Information technology is widely applied in all fields of social life. With the trend of development and integration of the country, investment in IT is very necessary and important. The application of computer software in the fields helps to improve the efficiency and quality of work, in addition to saving time and reducing human fatigue.

The actual needs of society require people to always be active and creative to create products to meet the increasingly diverse needs of society. Even in the field of parking at schools, residential areas, shopping centers... people have been using software for management for a long time. form, organization, management and towards a scientific system. Applying IT and automation in the system to operate and manage to improve the quality of operations, service, simplify management, search, import and export vehicles in and out of the station... is what this topic is about.

With the guidance of Mr. Le Van Vinh and the knowledge imparted, the group chose the topic "Management of the toll collection system, keeping motorbikes and cars" to build software, know how to analyze design a system, build a database for the software, design the interface, etc. Currently, there are many programming languages ​​used such as C, C++, C#, VB.NET, JAVA, PHP, Javascript, ASP.Net.... Along with JAVA, C# is the most popular language. present, thanks to its flexibility.

We have researched and fully completed the requirements of the big assignment, although it is inevitable that there will be shortcomings and not good, but it is my effort in the initial integration into the working environment outside. outside and have a toddler exposure to Information Technology, so we are looking forward to receiving suggestions from the teacher and all students in the class so that the big assignment can be completed well.

We sincerely thank you.

TASK ASSIGNMENT

|  |  |  |
| --- | --- | --- |
| **Student’s name** | **Evaluate contribution** | **Taskwork** |
| Nguyễn Minh Trí | 100% | Database + Code |
| Nguyễn Thanh Toàn | 100% | Code + Report |

**Note:** %: The percentage of each student participating.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Goal** | | **Schedule** | | | | | | |
| 1 | Understand & decrible the requirements of the project. | | o | o |  |  |  |  |  |
| 2 | Learn & study technology, language programming to do project | | o | o | o |  |  |  |  |
| 3 | Identify the functions to be used in the software. | | o | o | o |  |  | o |  |
| 4 | Building & design front-end  architecture. | | o | o | o |  |  |  |  |
| 5 | Building & design back-end  architecture. | |  |  | o | o | o |  |  |
| 6 | Testing the software. | |  |  |  |  | o | o | o |
| 7 | Write report. | |  |  |  |  | o | o | o |
| Week | | | 2 | 4 | 6 | 8 | 10 | 12 | 14 |
| Note | | o-Begin | | O – Complete 50% | | | O – Complete 100% | | |

**Remark of teacher:**

...............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

|  |
| --- |
| Thu Duc city, December 4th, 2022 |
| **Signature and full name of teacher** |

I. INTRODUCTION TO THE PROBLEM

1. Survey and analysis of the current situation

Currently, many parking areas in our country are still managed according to the traditional method. That's how to manage and store papers and books. This method has many limitations. In order to manage fully, in detail, and accurately – the owner of the parking area has to spend a large amount of money on buying materials (papers, books, documents). Moreover, due to the characteristics of paper materials are very quickly damaged, must be regularly replaced, upgraded, each change is a time to copy books, preserve documents ... costly in terms of investment costs. , it takes a lot of time and effort. Besides, the management and statistics of the number of vehicles entering and leaving is also a difficult problem. Management and maintenance activities in the traditional way take up a lot of time and effort. Aware of that importance, we make the management system software for motorbikes and cars, for parking and managing vehicles in collective areas, hospitals, schools...

Aware of that importance, we make the management system software for motorbikes and cars, for parking and managing vehicles in collective areas, hospitals, schools...

2. Problem

The management system for parking motorbikes and cars has functions such as managing vehicles entering and leaving the station, managing employee accounts, managing workers, dividing shifts for workers, managing tickets, managing vehicles in the station, statistics on the number of tickets issued, statistics on tickets, statistics on vehicles sent, statistics on the amount of money collected. There are two types of parking: parking:

There are two types of parking: parking:

* Send vehicles in turn.
* Send vehicles in month.

For customers with turn-by-turn parking, each turn is counted as one time entering and leaving the station. Customers must pay when the car leaves the station at the price of 50000 VND/trip for cars and 5000 VND/way for motorbikes and 1000 VND/trip for bicycles.

For customers who send cars by month, customers must register for tickets sent by month, 1 month is equal to 30 days (calculated from the time of registration). Customers have to pay when registering for 2000000 VND/month for cars, 120000 VND/month for motorbikes and 45,000 VND for bicycles.

When the customer leaves the car, the parking attendant will enter the license plate number and the ticket number (the ticket number printed on the ticket) into the system. Then send the customer the ticket with the code just entered. Tickets are plastic or metal card, each ticket has a unique code that matches the code in the system's list of ticket numbers.

Khi When customers leave the station, they must return the ticket to the parking attendant, who will enter the ticket and number plate (or enter the number plate and search):

* If it matches, the vehicle information and the amount to be paid will be displayed, the staff will confirm that the car will leave the station successfully, if the staff does not confirm, the car will still be in the list of vehicles in the station (not allowed to leave the station). Wharf).
* If it does not match, a message will be displayed on the screen, the vehicle is still in the list of vehicles in the station (not allowed to leave the station).

II. DATABASE DESIGN

1. ERD

|  |
| --- |
|  |
| *Figure 2.1 Diagrams Model* |

2. Overview

The database is designed in 8 tables:

* NHANVIEN
* VE
* XE
* BAIXE
* THONGKE
* CHIACA
* THO
* TIMELAMVIEC

2.1. NHANVIEN Table

|  |  |
| --- | --- |
| Capture.PNG | Capture.PNG |
| *Figure 2.2: Employee table* | |

The staff table includes:

* *maNV*: is the primary key of the table, containing the employee's code and also the account name to log in to the system, not allowed to be empty.
* *tenHienThi*: contains the employee's first and last name, not allowed to be blank.
* *loaiNV*: contains the type of employee (regular employee or manager), not allowed to leave blank.
* *ngaySinh*: contains the employee's date of birth, not allowed to be blank.
* *gt*: contains the employee's gender, not allowed to be blank.
* *diaChi*: Chicontains the employee's address, not allowed to be empty.
* *sdt*: contains the employee's phone number, not allowed to be blank.
* *matKhau*: contains the employee's password to log in to the system, not allowed to leave it blank.

2.2. VE table

|  |  |
| --- | --- |
|  |  |
| *Figure 2.3: VE table* | |

Ticket table includes:

* *loaiXe*: là xe máy hoặc ô tô, cho phép để trống.
* *maSoVe*: là khóa chính của bảng, chứa mã số của vé xe, không cho phép để trống.
* *giave*: giá tiền của vé xe, cho phép để trống.
* *bienSo*: chứa biển số của xe đang mang vé, cho phép để trống.

2.3. BAIXE table

|  |  |
| --- | --- |
| Capture.PNG | Capture.PNG |
| *Figure 2.4: BAIXE table* | |

Parking included:

* *khuVuc*: name of the parking area, not allowed to be empty.
* *loaiXe*: is a motorbike or a car, not allowed to be empty.
* *tongSoCho*: the total number of positions in the parking lot, not allowed to be empty.
* *soChoTrong*: contains the number of empty positions, not allowed to be empty.
* *soTienThuVe*: total amount collected, not allowed to leave blank.

2.4. XE table

|  |  |
| --- | --- |
|  |  |
| *Figure 2.5: XE table* | |

XE table includes:

* *maSoVe*: is the ticket of the vehicle, pointed to the maSoVe of the VE table, not allowed to be empty.
* *bienSo*: is the primary key of the table, containing the license plate of the vehicle, not allowed to be empty.
* *loaiXe*: vehicle type refers to a motorcycle or a car, pointed to the type of vehicle of the BAIXE table, which cannot be left blank.
* *tenxe*: name of the vehicle, leave it blank.
* *mauxe*: color of the car, leave it blank.
* *loaiVe*: type of bus ticket is a one-way ticket or a monthly ticket that cannot be left blank.
* *tgvaoben*: the time the vehicle enters the station, it is allowed to leave it blank.
* *tgxuatben*: bus departure time, allowed to leave.
* *ngayDangKy*: registration period for monthly tickets (For vehicles registered to send monthly) is allowed to be left blank.
* *ngayHetHan*: The expiry time of the monthly ticket (For vehicles registered to send monthly) is allowed to be left blank.

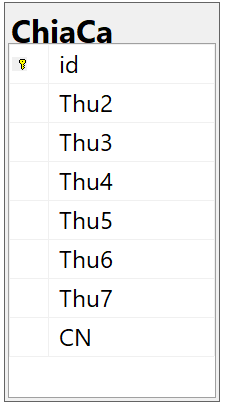
2.5. THONGKE table

|  |  |
| --- | --- |
|  |  |
| *Figure 2.6: THONGKE table* | |

THONGKE table includes:

* *id*: ordinal number of the vehicle list is statistic, is the primary key, is automatically incremented, not allowed to be empty.
* *maSoVe*: contains the bus ticket number, which is pointed to the maSoVe of the VE table, not allowed to be empty.
* *bienSo*: contains license plate number, not allowed to leave blank.
* *loaiXe*: are motorbikes or cars, not allowed to be empty.
* *tenXe*: name of the vehicle, leave it blank.
* *mauxe*: color of the car, leave it blank.
* *tgvaoben*: the time the car enters the station, not allowed to be empty.
* *tgxuatben*: time the car leaves the station, not allowed to be empty.
* *loaiVe*: The type of ticket is a one-way ticket or a monthly ticket that cannot be left blank.
* *ngayDangKy*: registration period for monthly tickets (For vehicles registered to send monthly) is allowed to be left blank.
* *ngayHetHan*: the expiration time of the monthly ticket (For vehicles registered to send monthly) is allowed to be left blank.

2.6. CHIACA table



*Figure 2.7: ChiaCa table*

ChiaCa table includes:

* *id*: is the primary key of the table, containing the employee's code and also the account name to log in to the system, not allowed to be empty.
* *Thu2 – CN*: Working time from monday to Sunday.

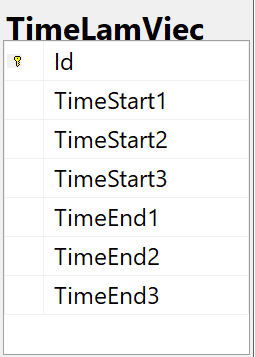
2.7. THO table

|  |  |
| --- | --- |
|  |  |
| *Figure 2.8: Tho table* | |

Tho table includes:

* *IdTho*: is the primary key of the table, containing the employee's code and also the account name to log in to the system, not allowed to be empty.
* *ten*: contains the employee's name, not allowed to be blank.
* *ngaysinh*: contains the employee's date of birth, not allowed to be blank.
* *gioitinh*: contains the employee's gender, not allowed to be blank.
* *phone*: the employee's phone number, not allowed to be blank.
* *diachi*: contains the employee's address, not allowed to be empty.
* *vaitro*: contains the employee’s role, not allowed to be empty.
* *luong*: contains the employee’ salary, not allowed to be empty.

2.8. TIMELAMVIEC table



*Figure 2.9: TimeLamViec table*

TimeLamViec table:

* *Id*: is the primary key of the table, containing the employee's code and also the account name to log in to the system.
* *TimeStart1 – TimeStart3*: starting time of shifts from 1 to 3.
* *TimeEnd1 – TimeEnd3*: ending time of shifts from 1 to 3.

III. DESIGN PROGRAM

1. Overview of the program

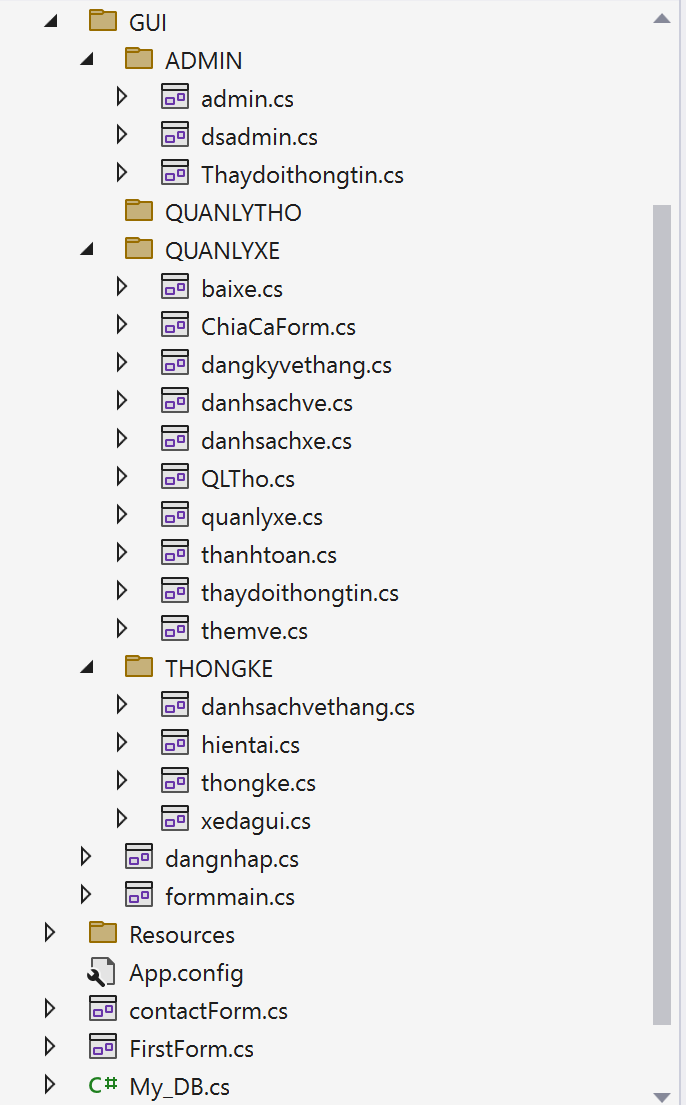
**Our parking management system** is designed based on a 3-layer model, so it can be easily changed and upgraded, easily distributed data and divided work for each person in the group.

Program divided into 3 layers: **GUI**, **DAO** and **DTO**

|  |  |
| --- | --- |
|  |  |
| *Figure 3.1: Solution Explorer of the project* | |

1.1. GUI layer:

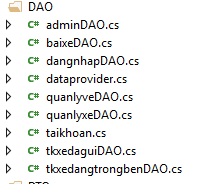
This is the interface layer of the program including forms.



*Figure 3.2: GUI layer*

1.2. DAO layer:

The DAO layer is the layer that processes data taken from the database to the interface and vice versa, this layer contains the query file and connects to the database.



*Figure 3.3: DAO layer*

In this layer we also incorporate both normalization and data binding.

Dataprovider.cs is used to connect to the database, the remaining files correspond to each of the program's interface forms and each file is responsible for retrieving data from the database and the interface and taking back input data from the interface to add to the database, such as Add, Edit, Delete, search data...

For example some functions like:

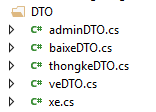
* ExecuteQuery(String query): The Data Connection contains the Query query.
* ExecuteNonQuery(String query) : check the number of successful fields when inserting and updating data.

|  |
| --- |
| Capture.PNG |
| *Examples of functions in veDAO.cs* |

|  |
| --- |
| Capture.PNG |
| *Examples of functions in adminDAO.cs* |

1.3. DTO layer

The DTO layer is responsible for mapping data from the database to the DAO layer



*Figure 3.4: DTO layer*

2. Program interface and functions

2.1. Login function

When launching the program, the employee must log in to the system. If the user enters the wrong account and password in the database, the system will notify the login failed and not allow to use the software with the employee's rights. or administrator (depending on the account), otherwise, when the user successfully logs in, the program will open the main interface to work.

|  |
| --- |
|  |
| *Figure 3.5: Sign in* |

* *Successful login will open the main interface.*

2.2. Forget password fuction

We create a new form to get the password if user forgot their password.

We using SMPT Protocol to connect to the sever and send the verify code to the user’s email on SQL Server if user want to get a new password.

|  |
| --- |
|  |
| *Figure 3.6: Send the CODE through user’s email.* |



*Code has been sent!*

2.3. Account categories

The account directory is where all employee accounts are displayed and some functions such as:

* Management staff can view the list of employees, search by employee code, add, delete, change information of all other employees.

|  |
| --- |
|  |
| *Figure 3.7: List of accounts for management staff.* |

|  |
| --- |
|  |
| *Figure 3.8: Button to update information of managers* |

* Employees can usually only view the list of employees and change their own information

|  |
| --- |
|  |
| *Figure 3.9: List of accounts for regular employees* |

|  |
| --- |
| Capture.PNG |
| *Figure 3.10: Button to update information of regular employees* |

**Add staff:**

* All textboxes cannot be empty
* Employee code must not exceed 20 characters.
* Employee name should not exceed 100 characters.
* Date of birth enter in mm/dd/yyy format or select.
* Enter gender as “Male” or “Female” or select.
* Type of employee enter “as a manager” or “a regular employee” or select.
* Phone number enter whole number or enter according to international phone number standard.
* Password must not exceed 20 characters.
* If the input is incorrect when pressing the “Add” button, there will be an error message.
* If it is correct, there will be a success message.

|  |
| --- |
|  |
| *Figure 3.11: Error message when not entering correctly at add button* |

**Find button:**

The administrator enters the characters to be searched into the search textbox and then presses the "search by code" button. If a code containing the characters just entered is displayed, the information will be displayed on the board.

**Delete button:**

The administrator enters the employee code to be deleted into the employee code textbox. If the employee has the code just entered, the message will be deleted successfully, if it does not exist, an error message will be reported.

**Information change button:**

* All textboxes cannot be empty
* The multiplier must already exist in the employee table.
* Employee name should not exceed 100 characters.
* Date of birth enter in mm/dd/yyy format or select.
* Enter gender as “Male” or “Female” or select.
* Employee type enter “as a manager” or “a regular employee” or select.
* Phone number enter whole number or enter according to international phone number standard.
* Password must not exceed 20 characters.
* If the input is incorrect when pressing the “Add” button, there will be an error message.
* If it is correct, there will be a success message.

|  |
| --- |
| Capture.PNG |
| *Figure 3.12: Error message when not entering correctly at edit button* |

2.4. Management Directory

The management portfolio includes:

* Employee mangement
* Vehicle management
* Ticket management
* Parking management

***2.4.1. Employee management***

|  |
| --- |
|  |
| *Figure 3.13: Employee management* |

**“Thêm” button:** When enter full of infomations, we can add a new woker.

**“Xóa” button:** We can delete a woker by the way click on the datagridview and information will be show in the information textboxs, then click on the “Xóa” button.

**“Thay đổi” button:** we can change the information of the worker by clicking on the employee to be deleted in the datagridview and the information will be displayed on the textbox above, then we click the button "Thay đổi".

**“Tải lại” button:** this button only refesh the information again.

**“Chia ca” button:** This button is responsible for dividing shifts among existing workers

**“Reset time” button:** reload time.

***2.4.2. Vehicle management***

|  |
| --- |
|  |
| *Figure 3.14: Vehicle management* |

**Search button:** The employee enters the ticket number in the search textbox and then presses the "search by code" button. If a code containing the characters you just entered appears, the information will be displayed on the board.

**Button to dock**: staff enter information into texboxes

* Number plate: cannot be blank, must not exceed 20 characters and must not be in the berth.
* Ticket number: must not be empty, must match the list of ticket numbers in the station, must not exceed 10 characters and must not be issued to vehicles already in the station.
* Vehicle type: choose a vehicle type that is a motorbike or a car.
* Vehicle name: vehicle name, vehicle name must not exceed 100 characters, can be left blank.
* Vehicle color: vehicle color must not exceed 20 kg and can be left blank.
* The arrival date and time are automatically selected and the staff presses the “Enter” button.
* If the station is full, the vehicle cannot be added to the station and an error message will be displayed.
* If the input is incorrect and press the “Enter” button, there will be an error message.
* If you enter it correctly and press “Enter” button, a success message will be displayed.

**Edit button:** does not allow to edit the number plate.

* Ticket number: must not be empty, must match the list of ticket numbers in the station, must not exceed 10 characters and must not be issued to vehicles already in the station.
* Vehicle type: automatically selected according to the ticket.
* Vehicle name: vehicle name, vehicle name must not exceed 100 characters, can be left blank.
* Vehicle color: vehicle color must not exceed 20 kg and can be left blank.
* The arrival date and time are automatically selected and the staff presses the “Enter” button.
* If the input is incorrect and the “correct” button is pressed, an error message will appear.
* If you enter it correctly and press the “edit” button, there will be a success message.

**Departure button:** staff enter number plate and ticket number.

* If it matches, the vehicle information and the amount to be paid will be displayed, the staff will confirm that the car will no longer be in the list of vehicles in the station, if the staff press cancel, the car is still in the list of vehicles in the station and has notification.
* If it doesn't match, there will be a failure message

**Button for monthly ticket registration**: to register for a monthly ticket, you need to fill in all the textboxes and enter the correct encoding like the button to enter the station.

**2.4.3. Ticket management**

|  |
| --- |
|  |
| *Figure 3.15: Ticker management* |

Total ticket list button: displays a list of total tickets.

Ticket list button in use: display a list of tickets with cars.

Empty ticket list button: displays the list of tickets that have not been exported.

Button to create more tickets: used when the bus station makes some more ticket cards. This button is only for admin staff.

|  |
| --- |
|  |
| *Figure 3.16: Add ticker* |

Note:

* *Mã số vé* Ticket number cannot be empty and must not exceed 10 characters.
* Type of vehicle to enter or select is “Motorcycle” or “Car”.

*2.4.4. Parking management*

Displays the entire parking table of the database, besides it also displays the total number of positions and empty positions of the car and motorbike parking lot and allows you to change the total number of parking positions.

|  |
| --- |
|  |
| *Figure 3.17: Parking management* |

Button to change station information: only for management staff, it allows to change the total number of parking positions.

|  |
| --- |
| Capture.PNG |
| *Figure 3.18: Change parking information* |

Employees enter the number of positions for cars and the number of positions for motorbikes so that the number of positions must be larger than the number of vehicles currently in the station and not more than 9999999.

2.5. Statistical List

2.5.1. The list of vehicles sent at the station

Shows the list of vehicles, the amount collected from the vehicles that have been deposited at the station since the last reset.

|  |
| --- |
|  |
| *Figure 3.19: Statistics of the list sent at the wharf* |

**Search button:** search for vehicles with codes containing the characters entered in the search textbox.

**Reset button:** clears the entire list of sent vehicle statistics, returns the amount collected to 0.

**2.5.2. Current directory**

Display the list of vehicles in the station, the number of empty positions.

|  |
| --- |
|  |
| *Figure 3.20: statistics at the present time* |

**2.5.3. List of vehicles registered for monthly tickets.**

Display the list of vehicles registered for monthly tickets.

|  |
| --- |
|  |
| *Figure 3.21: List of vehicles registered for monthly tickets* |

2.6. Help List

When the user clicks on the help button, the system automatically opens the default browser and goes to the support website from the software developer.

IV. ASSESSMENT OF THE PROGRAM

1. Advantages

* Easy to upgrade, update and change due to its 3-layer design.
* The system ensures basic security, specifically some important functions, affecting the system only management staff can use.
* Easy to use intuitive interface.
* No more basic errors.
* Easily list the vehicle list, and the amount collected.

2. Disadvantages

* Simple database.
* Takes a while for the import and export process.
* Some features are not close to reality.

OVERALL

Through this course, by designing a program, we have understood well how through the process to make a complete software, also learned how to work in a team, improve programming thinking in addition.

The software is sketchy and has many shortcomings. We would like to thank our teachers for their guidance, comments and evaluations to help us complete this software!

**THE END.**