**LỜI NÓI ĐẦU**

Ngày nay việc ứng dụng công nghệ thông tin trong tất cả các công việc đã không còn xa lạ. Đối với ngành hàng không, việc có một hay nhiều phần mềm giúp hỗ trợ các hoạt động quản lí là không thể thiếu. Vì vậy, nhóm chúng em đã quyết định thực hiện “ Phần mềm quản lí bán vé chuyển bay – SE Airline” là phần mềm có thể giúp các nhân viên, đại lí bán vé có thể thực hiện tốt hơn công việc của mình một cách nhanh chóng và chính xác nhất.

Chúng em chân thành cảm ơn cô Đỗ Thị Thanh Tuyền đã nhiệt tình giúp đỡ và cho chúng em những kiến thức quý giá về môn Nhập môn Công nghệ phần mềm để chúng em có thể hoàn thành đồ án một cách tốt nhất. Bên cạnh đó, tất nhiên đồ án vẫn còn một vài thiếu sót mong cô thông cảm và có thể góp ý thêm cho chúng em.

**MỤC LỤC**

**CHƯƠNG 1**

**TỔNG QUAN**

**1.1 Giới thiệu bài toán cần giải quyết**

Ứng dụng công nghệ thông tin vào việc hỗ trợ hoạt động quản lí bán vé cho các doanh nghiệp hàng không, giúp người dùng làm việc một cách dễ dàng và chính xác nhất. Từ đó, thiết kế và hoàn chỉnh phần mềm “ Quản lí bán vé chuyến bay – SE Airline”

**1.2 Quy trình thực hiện các công việc chính**

- Thiết kế cơ sở dữ liệu.

- Thiết kế giao diện phần mềm.

- Thiết kế hệ thống phần mềm.

- Cài đặt và kiểm thử.

- Bảo trì, nâng cấp và sửa chữa.

**CHƯƠNG 2**

**XÁC ĐỊNH VÀ MÔ HÌNH HÓA YÊU CẦU PHẦN MỀM**

2.1

**CHƯƠNG 3**

**THIẾT KẾ HỆ THỐNG**

**3.1 Kiến trúc hệ thống**

**3.1.1 Mô hình ERD**

**3.1.1.1 Mô hình thực thể ERD**

Thực thể 1: SANBAY

Các thuộc tính:

* Mã sân bay(MASANBAY): Đây là thuộc tính khóa, nhờ thuộc tính này mà ta phân biệt được các sân bay khác nhau.
* Tên sân bay(TENSANBAY): Ứng với mã sân bay
* Quốc gia(QUOCGIA): Sân bay thuộc quốc gia nào

Thực thể 2: CHUYENBAY

Các thuộc tính:

* Mã chuyến bay(MACHUYENBAY): Thuộc tính khóa.
* Giá vé hạng 1(GIAVE): Giá vé ghế hạng 2\*1.05
* Giá vé hạng 2: Giá vé hạng 1
* Sân bay đi(SANBAYDI): Vị trí bay
* Sân bay đến(SANBAYDEN): Vị trí đáp
* Ngày(NGAY): Ngày xuất phát chuyến bay
* Giờ(GIO): Thời gian xuất phát chuyến bay
* Thời gian bay(THOIGIANBAY): Thời gian bay dự kiến
* Số lượng vé hạng 1(HANG1):
* Số lượng vé hạng 2(HANG2):
* Số lượng vé hạng 1 còn lại(HANG1CONLAI):
* Số lượng vé hạng 2 còn lại(HANG2CONLAI):

Thực thể 3: SANBAYTRUNGGIAN

* Mã trung gian(MATRUNGGIAN): Thuộc tính khóa.
* Mã sân bay(MASANBAY): Mã sân bay trung gian
* Mã chuyến bay(MACHUYENBAY): Mã chuyến bay khi đáp xuống sân bay trung gian
* Thời gian dừng(THOIGIANDUNG): Thời gian dừng dự kiến ở sân bay trung gian

Thực thể 4: KHACHHANG

* Mã khách hang(MAKHACHHANG): Thuộc tính khóa
* Họ tên(HOTEN):
* Giới tính(GIOITINH):
* Chứng minh nhân dân(CMND):
* Số điện thoại(SDT):
* Địa chỉ(DIACHI):

Thực thể 5: VE

* Mã chuyến bay(MACHUYENBAY): Thuộc tính khóa
* Mã khách hang(MAKHACHHANG):
* Hạng vé(HANGVE):

Thực thể 6: QUYDINH6(Thay đổi các quy định)

* Số lượng sân bay(SOLUONGSANBAY):
* Thời gian bay tối thiểu(THOIGIANBAYTOITHIEU):
* Số sân bay trung gian tối đá(SOSANBAYTRUNGGIANTOIDA):
* Thời gian dừng tối đa(THOIGIANDUNGTOIDA)
* Thời gian dừng tối thiểu(THOIGIANDUNGTOITHIEU):
* Số lượng vé hạng 1 (SOLUONG VEHANG1)
* Số lượng vé hạng 2(SOLUONGVEHANG2)

**3.1.1.2 Mô hình ERD**

**3.1.2 Chuyển mô hình ERD thành mô hình quan hệ**

SANBAY(MaSanBay,TenSanBay,QuocGia)

CHUYENBAY(MaChuyenBay,GiaVeHang1,GiaVeHang2,SanBayDi,SanBayDen,Ngay,Gio,ThoiGianBay,Hang1,Hang2,Hang1ConLai,Hang2ConLai)

SANBAYTRUNGGIAN(MaTrungGian,MaSanBay,MaChuyenBay,ThoiGianBay,ThoiGianDung)

KHACHHANG(MaKhachHang,HoTen,GioiTinh,CMND,SDT,DiaChi)

VE(MaChuyenBay,MaKhachHang,HangVe)

QUYDINH6(SoLuongSanBay,ThoiGianBayToiThieu,SoSanBayTrungGianToiDa,ThoiGianDungToiDa,ThoiGianDungToiThieu,SoLuongVeHang1,SoLuongVeHang2,ThoiGianDatVeChamNhat,ThoiGianHuyDatVe)

**CHƯƠNG 4**

**THIẾT KẾ DỮ LIỆU**

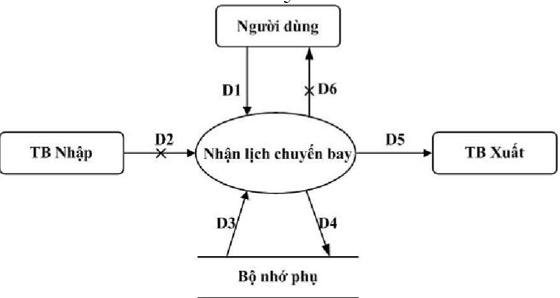
**2.4.1 Xét yêu cầu phần mềm “Flight Search”**

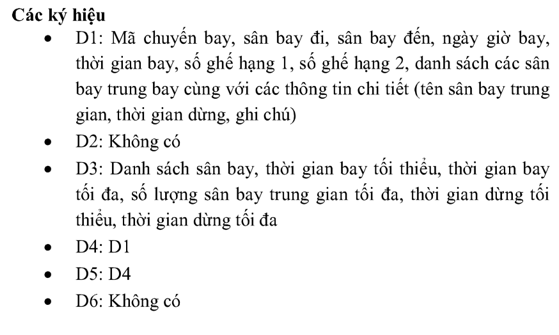
**2.4.1.1 Thiết kế dữ liệu với tính đúng đắn**

**2.4.1.1.1 Biểu mẫu liên quan: BM1**

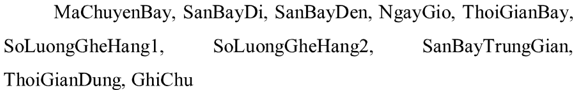


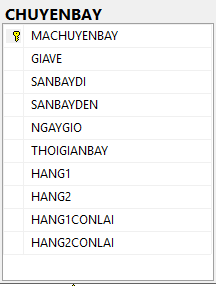
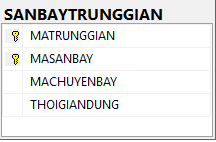
2.4.1.1.2 Sơ đồ luồng dữ liệu





**2.4.1.1.3 Các thuộc tính mới**





2.4.1.1.4 Các thuộc tính trừu tượng

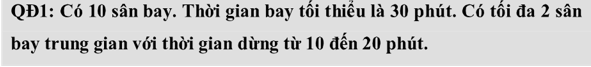
**MACHUYENBAY, MATRUNGGIAN,MASANBAY**

2.4.1.1.5 Sơ đồ logic

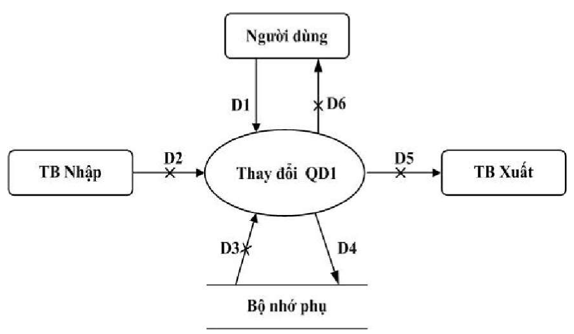


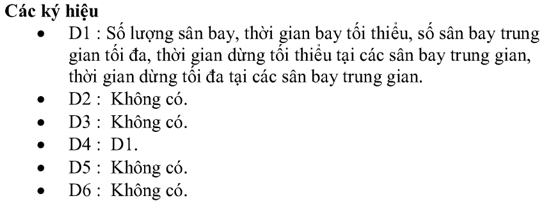
2.4.1.2 Thiết kế dữ liệu với tính tiến hóa

2.4.1.2.1 Quy định liên quan: QĐ1



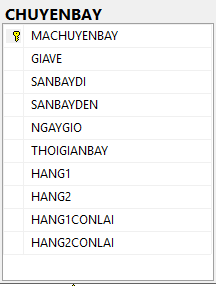
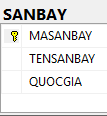
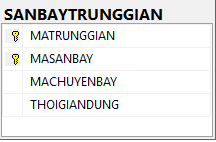
2.4.1.2.2 Sơ đồ luồng dữ liệu về việc thay đổi quy định

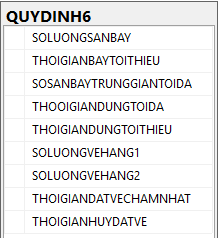




2.4.1.2.3 Các thuộc tính mới







2.4.1.2.4 Các thuộc tính trừu tượng

MASANBAY

2.3.1.2.5 Sơ đồ logic

|  |
| --- |
| CHUYENBAY |

|  |
| --- |
| SANBAYTRUNGGIAN |

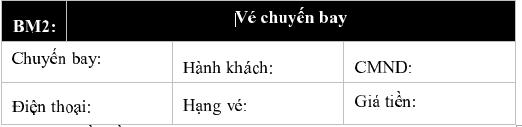
|  |
| --- |
| SANBAY |

|  |
| --- |
| QUYDINH6 |

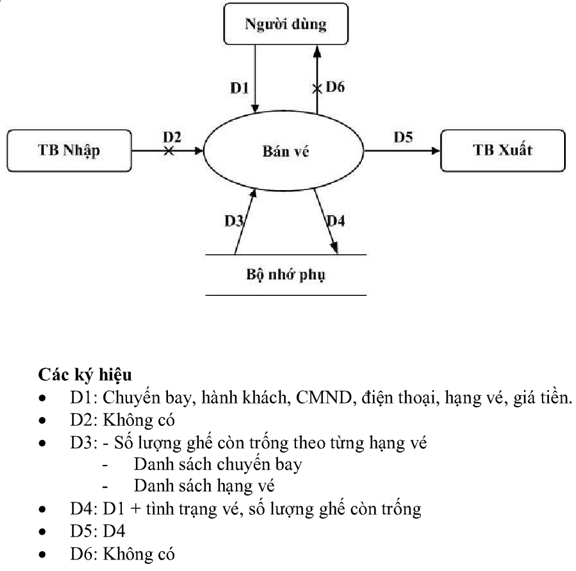
**2.4.2 Xét yêu cầu phần mềm “Book”**

*2.4.2.1 Thiết kế dữ liệu với tinh đúng đắn*

2.4.2.1.1 Biểu mẫu liên quan BM2

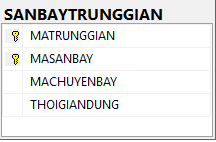
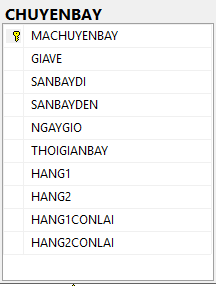


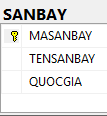
2.4.2.1.2 Sơ đồ luồng dữ liệu

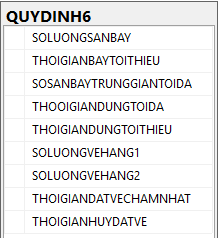


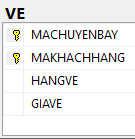
2.4.2.1.3 Các thuộc tính mới

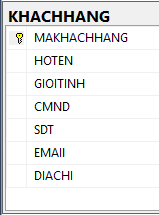












2.4.2.1.4 Các thuộc tính trừu tượng



2.4.2.1.5 Sơ đồ logic

|  |
| --- |
| CHUYENBAY |

|  |
| --- |
| CHITIETCHUYENBAY |

|  |
| --- |
| SANBAY |

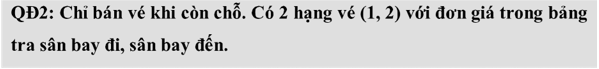
|  |
| --- |
| QUYDINH6 |

|  |
| --- |
| VECHUYENBAY |

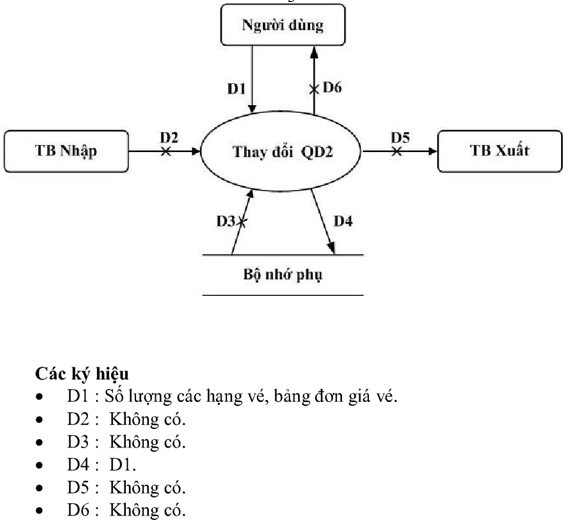
|  |
| --- |
| HANHKHACH |

*2.4.2.2 Thiết kế dữ liệu với tính tiến hóa*

2.4.2.2.1 Quy định liên quan : QĐ2



2.4.2.2.2 Sơ đồ luồng dữ liệu ( về việc thay đổi quy định)



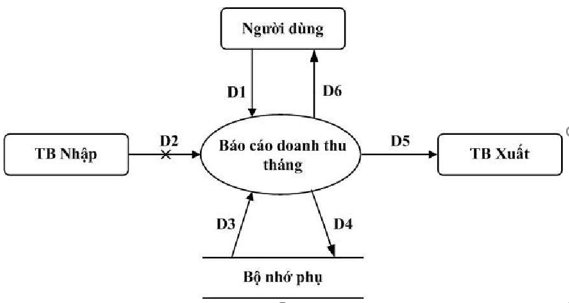
**2.4.3 Xét yêu cầu “ Revenue report”**

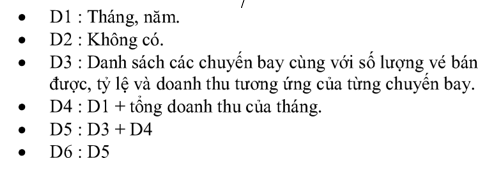
*2.4.3.1 Thiết kế dữ liệu với tính đúng đắn*

2.4.3.2.1 Biểu mẫu liên quan BM5.1



2.4.3.2.2 Sơ đồ luông dữ liệu

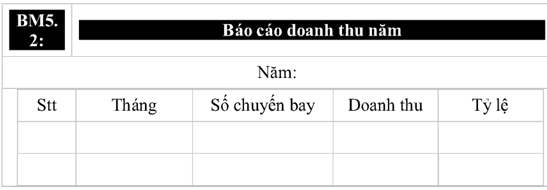




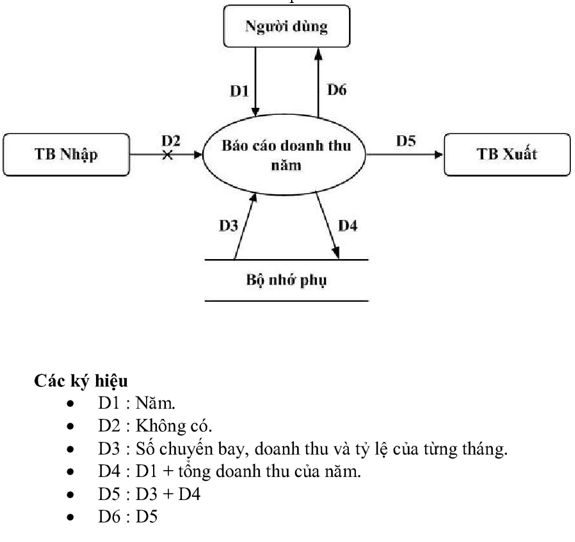
**2.4.4 Xét yêu cầu “xét yêu cầu lập báo cáo doanh thu năm”**

*2.4.4.1 Thiết kế dữ liệu với tinh đúng đắn*

2.4.4.1.1 Biểu mẫu liên quan: BM5.2



2.4.4.1.2 Sơ đồ luồng dữ liệu



**CHƯƠNG 5**

**THIẾT KẾ GIAO DIỆN**

1. Sơ đồ liên kết các màn hình.
2. Danh sách các màn hình:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Màn hình** | **Loại màn hình** | **Chức năng** |
| 1 | Flight Info | User Control | Tra cứu thông tin chuyến bay |
| 2 | Book trip | User Control | Đặt vé chuyến bay |
| 3 | Database Update | User Control | Cập nhật cơ sở dữ liệu mới |
| 4 | Income Report | User Control | Báo cáo doanh thu |
| 5 | Rule Change | User Control | Thay đổi các quy định |

1. Mô tả các màn hình:
2. Flight Info
   * + - 1. Giao diện
         2. Mô tả các đối tượng trên màn hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | From | ComboBox |  | Chọn điểm đi |
| 2 | To | ComboBox |  | Chọn điểm đến |
| 3 | Departure Date | Datepicker |  | Chọn ngày đi |
| 4 | Search | Button |  | Thực hiện truy vấn |
| 5 | Chuyến bay | Data Gird |  | Hiển thị các chuyến bay đã truy vấn |

* + - * 1. Danh sách biến cố và xử lý tương ứng trên màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 |  |  |
| 2 |  |  |

1. Book trip
2. Giao diện
3. Mô tả các đối tượng trên màn hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Full name | Textbox |  | Nhập tên |
| 2 | Phone number | Textbox |  | Nhập SĐT |
| 3 | Address | Textbox |  | Nhập địa chỉ |
| 4 | ID | Textbox |  | Nhập số CMND |
| 5 | Email | Textbox |  | Nhập email |
| 6 | Book | Button |  | Lưu dữ liệu đặt vé |

1. Danh sách biến cố và xử lý tương ứng trên màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 |  |  |
| 2 |  |  |

1. Database Upload
2. Giao diện
3. Mô tả các đối tượng trên màn hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Upload | Button |  | Mở Choose file dialog |
| 2 | Choose file | Dialog |  | Chọn file database |

1. Danh sách biến cố và xử lý tương ứng trên màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 |  |  |
| 2 |  |  |

1. Income Report
2. Giao diện
3. Mô tả các đối tượng trên màn hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Month | Combobox |  | Chọn tháng |
| 2 | Year | Combobox |  | Chọn năm |
| 3 | Export | Button |  | In báo cáo doanh thu |

1. Danh sách biến cố và xử lý tương ứng trên màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 |  |  |
| 2 |  |  |

1. Rules change
2. Giao diện
3. Mô tả các đối tượng trên màn hình:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Rules | Combobox |  | Chọn quy định cần thay đổi |
| 2 | Value | Textbox |  | Nhập giá trị mới |
| 3 | Update | Button |  | Cập nhật giá trị mới |

1. Danh sách biến cố và xử lý tương ứng trên màn hình:

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 |  |  |
| 2 |  |  |

**CHƯƠNG 6**

**CÀI ĐẶT VÀ KIỂM THỬ**

## 6.1. Các công nghệ sử dụng

### 6.1.1. Ngôn ngữ lập trình c#

C# là một ngôn ngữ cấp cao và thuần hướng đối tượng, hỗ trợ rất mạnh trong việc xây dựng các ứng dụng trên môi trường Windows. Đối với những lập trình viên đã từng sử dụng qua C, C++, hoặc Java, việc nắm bắt và sử dụng C# một cách hiệu quả là khá dễ dàng và ít tốn thời gian.

### 6.1.2. Ngôn ngữ truy vấn cơ sở dữ liệu SQL

SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn mang tính cấu trúc. Nó được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS).

### 6.1.3. Bunifu framework

Bunifu framework giúp người lập trình dễ dàng thiết kế giao diện bắt mắt

Bunifu framework là một thư viện hỗ trợ việc thiết kế cho windows form đơn giản hơn. Với bunifu framework .NET, thiết kế winform chưa bao giờ dễ như thế.

[](https://1.bp.blogspot.com/-0UqslEEk08Y/WFUQukWcubI/AAAAAAAAAIA/mmymK2MAB60nE7z3mD9w8_Zj40MxwRPLwCLcB/s1600/Bunifu+UI++banneffr.png)

Với bunifu  bạn có thể làm nhiều thứ dễ dàng hơn như :

* Slide menu hoạt hình (đóng mở)
* Nền winform đẹp hơn với Gradient background - Tô màu cầu vồng cho form (trộn 3 màu)
* Hỗ trợ Flat UI - Thiết kế giao diện phẳng như Metro Design
* Nhiều chức năng khác như progressbar quay tròn,biểu đồ ,rất tiện lợi.

## 6.2. Các hàm xử lý chính

### 6.2.1. Hàm Search

Công dụng : Tìm kiếm các chuyến bay theo địa điểm khởi hành , địa điểm đến , ngày bay , giờ khởi hành.

private void searchBt\_Click(object sender, EventArgs e)

{

if ((fromStation.selectedIndex == 0) || (toStation.selectedIndex == 0))

return;

string sql = "SELECT MACHUYENBAY, SB1.TENSANBAY AS SANBAYDI, SB2.TENSANBAY AS SANBAYDEN, GIO, HANG1CONLAI, GIAVEHANG1, HANG2CONLAI, GIAVEHANG2 " +

"FROM CHUYENBAY, SANBAY SB1, SANBAY SB2 " +

"WHERE CHUYENBAY.SANBAYDEN=SB2.MASANBAY AND CHUYENBAY.SANBAYDI= SB1.MASANBAY" +

" AND SB1.TINH=N'" + fromStation.selectedValue +

"' AND SB2.TINH=N'" + toStation.selectedValue +

"' AND NGAY='" + datePicker.Value.ToString() + "'";

if (gioKhoiHanh.selectedValue != "Any Time")

sql += "AND GIO='" + gioKhoiHanh.selectedValue + "'";

SqlCommand cmd = new SqlCommand(sql, LoginForm.Connection.Connection);

cmd.CommandType = CommandType.Text;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

flightInfo.DataSource = dt;

}

### 6.2.2. Hàm Book

Công dụng : Đặt vé dựa theo thông tin mã chuyến bay , hạng vé và thông tin của khách hàng . Lưu thông tin khách hàng và vé vào database.

### 6.2.3. Hàm Upload

Công dụng : Chuyển dữ liệu từ file excel mẫu vào database

private void dataBt\_Click(object sender, EventArgs e)

{

openFileDialog1.Filter = "Excel | \*.xlsx; \*.xls| All File (\*.\*)|\*.\*";

openFileDialog1.ShowDialog();

if (openFileDialog1.FileName != "")

{

// tạo đối tượng excel

Excel.Application excel = new Excel.Application();

LoginForm.Connection.OpenConn();

// mở tệp

try

{

Excel.Workbook workbook = excel.Workbooks.Open(openFileDialog1.FileName);

// mở sheet

for (int i = 1; i < 4; i++)

{

Excel.Worksheet sheet = workbook.Sheets[i];

Excel.Range range = sheet.UsedRange;

object[,] valueArray = (object[,])range.get\_Value(Excel.XlRangeValueDataType.xlRangeValueDefault);

int row = valueArray.GetLength(0);

string command = "";

switch (i)

{

case 1: // SANBAY

for (int j = 2; j <= row; j++)

{

string maSanBay = valueArray[j, 1].ToString();

string tenSanBay = valueArray[j, 2].ToString();

string tinh = valueArray[j, 3].ToString();

string quocGia = valueArray[j, 4].ToString();

command = "INSERT INTO SANBAY VALUES("

+ "'" + maSanBay + "'" + ", "

+ "N'" + tenSanBay + "'" + ", "

+ "N'" + tinh + "'" +", " + "N'" + quocGia + "'" + ")";

ExcuteCommand(command);

}

break;

case 2: // CHUYENBAY

for (int j = 2; j <= row; j++)

{

if (valueArray[j, 1] == null)

{

break;

}

string maChuyenBay = valueArray[j, 1].ToString();

long giaVeHang1 = long.Parse(valueArray[j, 2].ToString());

long giaVeHang2 = long.Parse(valueArray[j, 3].ToString());

string sanBayDi = valueArray[j, 4].ToString();

string sanBayDen = valueArray[j, 5].ToString();

string ngayBay = valueArray[j, 6].ToString();

string gioBay = valueArray[j, 13].ToString();

int thoiGianBay = int.Parse(valueArray[j, 8].ToString());

int hang1 = int.Parse(valueArray[j, 9].ToString());

int hang2 = int.Parse(valueArray[j, 10].ToString());

int hang1conlai = int.Parse(valueArray[j, 11].ToString());

int hang2conlai = int.Parse(valueArray[j, 12].ToString());

command = "INSERT INTO CHUYENBAY VALUES("

+ "'" + maChuyenBay + "'" + ", "

+ giaVeHang2 + ", " + giaVeHang1 + ", " + "'" + sanBayDi + "'" + ", "

+ "'" + sanBayDen + "'" + ", " + "'" + ngayBay + "'" + ", "

+ "'" + gioBay + "'" + ", " + thoiGianBay + ", " + hang1 + ", "

+ hang2 + ", " + hang1conlai + ", " + hang2conlai + ")";

ExcuteCommand(command);

}

break;

case 3: // SANBAYTRUNGGIAN

for (int j = 2; j <= row; j++)

{

string chuyenBay = valueArray[j, 1].ToString();

string sanbay = valueArray[j, 2].ToString();

int thoiGianDung = int.Parse(valueArray[j, 3].ToString());

command = "INSERT INTO SANBAYTRUNGGIAN VALUES("

+ "'" + chuyenBay + "'" + ", "

+ "'" + sanbay + "'" + ", "

+ thoiGianDung + ")";

ExcuteCommand(command);

}

break;

}

}

MessageBox.Show("Data update successful !");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Thông báo !", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Bạn chưa chọn tệp tin nào !", "Thông báo !", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

### 6.2.4. Hàm Report

Công dụng : Xuất ra file excel dữ liệu doanh thu theo từng tháng và tính tổng doanh thu

private void reportBt\_Click(object sender, EventArgs e)

{

if(countMoney()=="0 VNĐ")

{

MessageBox.Show("Tháng này không có thu nhập !", "Thông báo");

return;

}

string sql = "SELECT VE.MAVE, VE.MACHUYENBAY, VE.HANGVE, VE.GIAVE FROM VE , CHUYENBAY WHERE VE.MACHUYENBAY=CHUYENBAY.MACHUYENBAY AND month(CHUYENBAY.NGAY)=" +

monthRp.selectedValue + " AND YEAR(NGAY)=" + yearRp.selectedValue;

SqlCommand cmd = new SqlCommand(sql, LoginForm.Connection.Connection);

cmd.CommandType = CommandType.Text;

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

listView.DataSource = dt;

//excel

string title = "Báo cáo tháng " + monthRp.selectedValue + "-" + yearRp.selectedValue;

SaveFileDialog save = new SaveFileDialog();

save.Filter = "(Tất cả các tệp)|\*.\*|(Các tệp excel)|\*.xlsx";

save.ShowDialog();

if (save.FileName != "")

{

Excel.Application app = new Excel.Application();

Excel.Workbook wb = app.Workbooks.Add(Type.Missing);

Excel.Worksheet sheet = null;

try

{

sheet = wb.ActiveSheet;

sheet.Name = title;

sheet.Range[sheet.Cells[1, 1], sheet.Cells[1, listView.Columns.Count]].Merge();

sheet.Cells[1, 1].Value = title;

sheet.Cells[1, 1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

sheet.Cells[1, 1].Font.Size = 20;

sheet.Cells[1, 1].Borders.Weight = Excel.XlBorderWeight.xlThin;

//tính tổng

for (int i = 1; i <= 10; i++)

{

sheet.Columns[i].ColumnWidth = 18;

}

sheet.Cells[2, 1].Value = "STT";

sheet.Cells[2, 1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

sheet.Cells[2, 1].Font.Bold = true;

sheet.Cells[2, 1].Borders.Weight = Excel.XlBorderWeight.xlMedium;

for (int i = 1; i <= listView.Columns.Count; i++)

{

sheet.Cells[2, i+1].Value = listView.Columns[i - 1].Name.ToString();

sheet.Cells[2, i+1].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

sheet.Cells[2, i+1].Font.Bold = true;

sheet.Cells[2, i+1].Borders.Weight = Excel.XlBorderWeight.xlMedium;

}

for (int i = 1; i < listView.Rows.Count; i++)

{

sheet.Cells[i + 2, 1] = i.ToString();

sheet.Cells[i + 2, 1].Borders.Weight = Excel.XlBorderWeight.xlThin;

sheet.Cells[i + 2, 2] = listView[0, i - 1].Value.ToString();

sheet.Cells[i + 2, 2].Borders.Weight = Excel.XlBorderWeight.xlThin;

for (int j = 1; j <= listView.Columns.Count; j++)

{

sheet.Cells[i + 2, j+1] = listView[j - 1, i - 1].Value.ToString();

sheet.Cells[i + 2, j+1].Borders.Weight = Excel.XlBorderWeight.xlThin;

}

}

sheet.Cells[listView.Rows.Count+3, 3].Value = "Tổng doanh thu :" + countMoney();

sheet.Cells[listView.Rows.Count + 3, 3].HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

sheet.Cells[listView.Rows.Count + 3, 3].Font.Size = 20;

sheet.Cells[listView.Rows.Count + 3, 3].Borders.Weight = Excel.XlBorderWeight.xlThin;

wb.SaveAs(save.FileName);

MessageBox.Show("Xuất thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally

{

app.Quit();

wb = null;

}

}

}

### 6.2.5. Các hàm update quy định

#### 6.2.5.1. Update số lượng sân bay tối đa

void UpdateNumberAirport()

{

if (index == 0 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set SOLUONGSANBAY = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update số lượng sân bay thành công");

}

}

#### 6.2.5.2. Update thời gian bay tối thiểu

void UpdateMinimunFlying()

{

if (index == 1 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set THOIGIANBAYTOITHIEU = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update thời gian bay tối thiểu thành công");

}

}

#### 6.2.5.3. Update số lượng sân bay trung gian tối đa

void UpdateImmeAirport()

{

if (index == 2 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set SOSANBAYTRUNGGIANTOIDA = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update số lượng sân bay trung gian tối đa thành công");

}

}

#### 6.2.5.4. Update thời gian dừng tối thiểu

void UpdateMinStopTime()

{

if (index == 3 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set THOIGIANDUNGTOITHIEU = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update thời gian dừng tối thiểu thành công");

}

}

#### 6.2.5.5. Update thời gian dừng tối đa

void UpdateMaxStopTime()

{

if (index == 4 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set THOOIGIANDUNGTOIDA = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update thời gian dừng tối đa thành công");

}

}

#### 6.2.5.6. Update số lượng vé hạng 1 tối đa

void Update1stTickets()

{

if (index == 5 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set SOLUONGVEHANG1 = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update số lượng vé hạng 1 thành công");

}

}

#### 6.2.5.7. Update số lượng vé hạng 2 tối đa

void Update2ndTickets()

{

if (index == 6 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set SOLUONGVEHANG2 = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update số lượng vé hạng 2 thành công");

}

}

#### 6.2.5.8. Update thời gian đặt vé trễ nhất

void UpdateSlowestBooking()

{

if (index == 7 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set THOIGIANDATVECHAMNHAT = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update thời gian đặt vé chậm nhất thành công");

}

}

#### 6.2.5.9. Update thời gian huỷ đặt vé

void UpdateCancellationBooking()

{

if (index == 8 && newValue.Text != null)

{

cmd.CommandText = "Update dbo.QUYDINH6 set THOIGIANHUYDATVE = " + value;

cmd.CommandType = CommandType.Text;

cmd.Connection = LoginForm.Connection.Connection;

try

{

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

string mes = ex.Message;

cmd.Dispose();

}

MessageBox.Show("Update thời gian hủy đặt vé thành công");

}

}

## 6.3. Kiểm thử

### 6.3.1. Mức độ hoàn thành

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mức độ hoàn thành (%)** | **Ghi chú** |
| 1 | Flight Info (Tìm kiếm chuyến bay) | 100% |  |
| 2 | Book Trip (Đặt vé chuyến bay) | 100% |  |
| 3 | Update Database (Cập nhật dữ liệu) | 100% |  |
| 4 | Revenue Report (Báo cáo doanh thu theo tháng/năm) | 100% |  |
| 5 | Rules Change (Thay đổi quy định cần thiết) | 100% |  |

### 6.3.2. Unit Test

Để đảm bảo chất lượng của chương trình, từng lớp sẽ được kiểm tra một cách riêng rẽ bằng phương pháp hộp trắng. Người tiến hành sẽ là những người trực tiếp cài đặt các lớp và phương thức ấy.

### 6.3.3. Intergration Test

Kiểm tra sự liên kết và giao diện giữa các lớp trong phần mềm.

### 6.3.4. System Test

Hệ thống sẽ được kiểm tra theo phương pháp hộp đen. Tức là dựa vào những đặc tả và yêu cầu của hệ thống đã thu thập được từ đầu để kiểm tra output chung của hệ thống mà không cần quan tâm đến việc thực hiện bên trong của chúng.

**CHƯƠNG 7**

**TỔNG KẾT**