****

**Trường Đại học Bách khoa Hà Nội**

**Viện Điện tử- Viễn thông**

**\*\*\*\*\*\*\*\*\*\*\*\*\***



**Báo cáo giữa kỳ**

**Đề tài: Phân tích, thiết kế hệ thống quản lý dịch vụ phòng game**

**Gvhd:Ths Vũ Song Tùng**

**Sinh viên**

|  |  |
| --- | --- |
| Nguyễn Công Kiên | 20142406 |
| Lê Thanh Sơn | 20143805 |
| Vũ Bảo Ngọc | 20143242 |

**Mục lục**

[**1.** **Tính khả thi của đề tài** 2](#_Toc514760036)

[*1.1* *Khả thi về mặt kỹ thuật* 2](#_Toc514760037)

[*1.2* *Khả thi về mặt kinh tế* 2](#_Toc514760038)

[*1.3* *Khả thi về mặt tổ chức* 3](#_Toc514760039)

[**2.** **Thu thập dữ liệu và xác định yêu cầu** 3](#_Toc514760040)

[*2.1* *Kế hoạc phỏng vấn* 3](#_Toc514760041)

[*2.2* *Kết quả thu được* 7](#_Toc514760042)

[*2.3 Class diagram* 8](#_Toc514760043)

[*2.4* *Sequence diagram* 8](#_Toc514760044)

[**3** **Xây dựng hệ thống** 9](#_Toc514760045)

[3.1 .User( client) 9](#_Toc514760046)

[*a, Các chức năng* 9](#_Toc514760047)

[b, Mô tả source code 9](#_Toc514760048)

[3.2 . GIAO DIỆN ADMIN 14](#_Toc514760049)

[a, Các chức năng và giao diện tương ứng 14](#_Toc514760050)

[- Mô hình hoạt động 21](#_Toc514760051)

[} 22](#_Toc514760052)

1. **Tính khả thi của đề tài**
   1. *Khả thi về mặt kỹ thuật*

* Dự trên yêu cầu của đề tài: Hệ thống quản lý phòng game tương đối phổ biến và dễ dàng tiếp cận . Độ phức tạp trung bình và có nhiều nguồn phục vụ cho việc tìm kiếm và thu thập thông tin.
* Dựa trên khả năng của từng thành viên: Các thành viên trong nhóm đều đã từng học và làm về phân tích thiết kế trong học phần Kỹ thuật phần mềm ứng dụng . Các thành việc đều có kiến thức cơ bản về lập trình C, C++ và song song đó là môn lập trình nâng cao với C# . Nhóm có thành viện có kinh nghiệm lập trình ứng dụng Window bằng C++ .
  1. *Khả thi về mặt kinh tế*

Với các chức năng và ưu điểm của sử dụng ứng dụng thay vì sử dụng sổ sách ghi chép ta có thể tối ưu được các mặt sau:

* Nhân công : Phần mềm với 1 máy chủ + 1 nhân viên có thể quản lý được tổ hợp nhiều máy (khoảng từ 80-100 máy – phụ thuộc vào quy mô của phòng net) , trong khi đó mỗi nhân viện chỉ có thể quản lý được từ 15-20 máy khi sử dụng sổ ghi chép . -> lợi ích kinh tế về mặt nhân công .
* Độ chính xác : Việc sử dụng ghi chép sẽ phát sinh nhiều sai sót khi mà 1 nhân viện sẽ phải quản lý quá nhiều máy , lợi nhuận sẽ dựa vào thời gian nên việc sai sót khi tính toán sẽ gấy mất lợi nhuận . Sử dụng phần mềm sẽ hầu như loại bỏ toàn bộ sai sót vì sử dụng việc nạp giờ chơi, và khi hết giờ máy tính sẽ được phần mềm điều khiển tắt .
* Tiết kiệm : Sử dụng phần mềm sẽ cho chúng ta tối thiểu hóa thời gian các máy thành việc mở mà không được sử dụng . Một phần làm giảm tiền điện phát sinh , một mặt nữa kéo dài tuổi thọ sử dụng của các máy tính thành viên . - > lợi ích về mặt hỏng hóc , bảo dưỡng và giảm tiền điện .
  1. *Khả thi về mặt tổ chức*

Như đã nói ở trên , đề tài có đọ phức tạp ở mức trung bình và các ứng dụng tương tự khá phổ biến và dễ dàng tiếp cận vậy nên việc tổ chức phân tích và tổ chức thiết kế sẽ khá dễ dàng vì ta có thể dựa và đó để phân tích các chức năng và sửa lỗi.

1. **Thu thập dữ liệu và xác định yêu cầu**
   1. *Kế hoạc phỏng vấn*

Sử dụng bảng câu hỏi trắc nghiệm:

a, Các câu hỏi dành cho chủ quán:

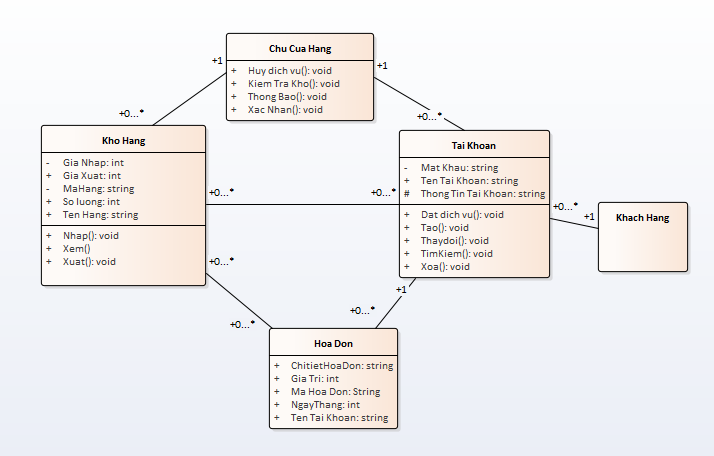
|  |  |
| --- | --- |
| Câu hỏi | Câu trả lời |
| Quy mô của quán |  |
| Các sản phẩm kinh doanh phụ? | a, nước giải khát  b, thuốc lá  c, Đồ ăn nhanh  d, thẻ game, thẻ điện thoại  e, ý kiến khác:…  f, không có |
| Quán có sử dụng nhân viên không? | a, Có  b, Không |
| Các danh mục muốn hệ thống quản lý? | a, Khách hàng (tài khoản; thông tin cá nhân; )  b, Các dịch vụ phụ khác (đồ ăn, nước uống …)  c, Nhân viên  d, Các máy tính  e, ý kiến khác:… |
| Các thông tin khách hàng muốn quản lý? | a, Tài khoản sư dụng dịch vụ  b, thông tin cá nhân (họ tên, năm sinh, số điện thoại, emai, CMND)  c, ý kiến khác:… |
| Quản lý các dịch vụ phụ như thế nào?(nếu có) | a, số lượng  b, đơn giá  c, trạng thái hiện tại của dịch vụ ( còn hay hết,…)  d, ý kiến khác:…. |
| Quản lý nhân viên?( nếu có) | a, Thông tin cá nhân  b, Thời gian làm việc  c, Hiệu suất làm việc  d, ý kiến khác:… |
| Hệ thống có cần lưu lại dữ liệu và lịch sử giao dịch hay không? Nếu có thì thời gian lưu là bao lâu? | a, có 1 tuần  b, có, 1 tháng  c, có 1 năm  d, không  e, lựa chọn khác:………. |
| Hệ thống có cần chức năng thống kê hay không | a, Có  b, Không |
| Các thông số cần thống kê nếu có? | a, Thời gian khách hang sử dụng  b, Doanh thu từ dịch vụ internet  c, Doanh thu từ các dịch vụ khác(nếu có)  d, số lượng hang hóa, dịch vụ (nếu có)  e, Khách hang sử dụng nhiều nhất, ít nhất,…  f, ý kiến bổ sung:….. |
| Hạn mức thống kê | a, hằng ngày  b, hằng tuần  c, hằng năm  d, ý kiến khác:… |
| Tính an toàn , bảo mật của hệ thống | a, không yêu cầu  b, yêu cầu bình thường  c, Yêu cầu tối đa |
| Có cần phân quyền truy quán lý hay không | a, có  b, không |
| Bảo trì, nâng cấp hệ thống khi cần | a, không  b, có  c, ý kiến khác:… |

b,Câu hỏi dành cho khách hàng:

|  |  |
| --- | --- |
| Câu hỏi | Trả lời |
| Những thông tin khách hàng có thể cung cấp khi tạo tài khoản để sử dụng dịch vụ? | a, Họ tên, ngày sinh  b, Số điện thoại  c, Email  d, Số CMND  e, ý kiến khác:… |
| Độ bảo mật tài khoản? | a, Không yêu cầu  b, Yêu cầu bình thường  c, Yêu cầu tối đa |
| Xác minh tài khoản khi cần: quên tài khoản, tranh chấp tài khoản,…? | a, Có  b, Không |
| Khách hang có muốn xem lịch sử sử dụng dịch vụ của mình hay không? | a, Có  b, Không |
| Các lịch sử muốn xem? | a, Lịch sử đăng nhập  b, Lịch sử nạp tiền |
| Thời gian muốn lưu lịch sử ? | a, 1 tuần  b, 1 tháng  c, 1 năm  d, càng lâu càng tốt |
| Xem các thông tin thống kê của tài khoản? | a, Có  b, Không |
| Các thông tin cần thống kê? | a, Số giờ sử dụng dịch vụ internet  b, Số tiền nạp  c, Số tiền sử dụng các dịch vụ khác |
| Mức thời gian muốn xem các thống kê? | a, Theo ngày  b, Theo tháng  c, Theo năm |
| Đặt đồ ăn, nước uống,… trực tiếp trên hệ thống? | a, có  b, không |

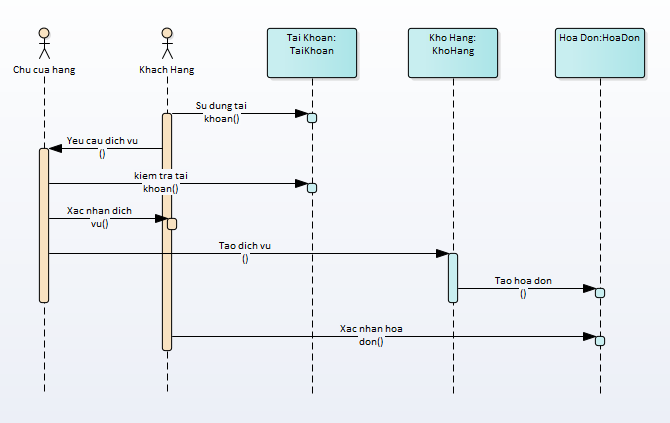
* 1. *Kết quả thu được*
* Quy mô khoảng 60 máy
* Các sản phẩm kinh doanh : Đồ ăn , nước uống , thuốc lá , thẻ nạp.
* Hệ thống quản lý : Tài khoản khách hàng , Các dịch vụ phụ , Các máy tính thành viên .
* Quản lý thông tin khách hàng gồm : Tài khoản, mật khẩu; thong tin cá nhân của khách hàng.
* Quản lý dịch vụ phụ dựa trên : Số lượng , Đơn giá , Trạng thái hiện tại của dịch vụ .
* Không sử dụng nhân viên .
* Cần lưu lại lịch sử giao dịch trong vòng 1 tháng .
* Hệ thống cần có chức năng thống kê Thời gian khách hàng sử dụng , doanh thu tổng thể hàng tháng . thống kê theo từng tuần
* Tính bảo mật của hệ thống cao .
* Yêu cầu có bảo trì và nâng cấp hệ thông định kỳ

*2.3 Class diagram*



Hinh 4. 7 Class Diagram

* 1. *Sequence diagram*



Hình 4. 8 Sequence diagram

1. **Xây dựng hệ thống**

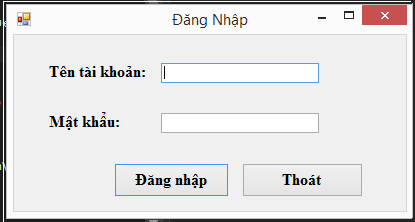
Phần mềm dùng để quản lý hệ thống dịch vụ và kho hàng của các phòng game .

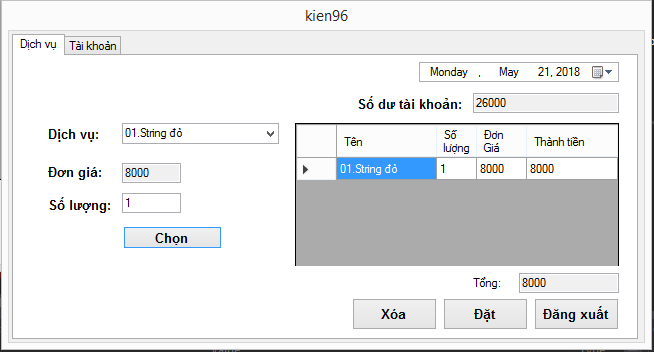
# 3.1 .User( client)

## *a, Các chức năng*

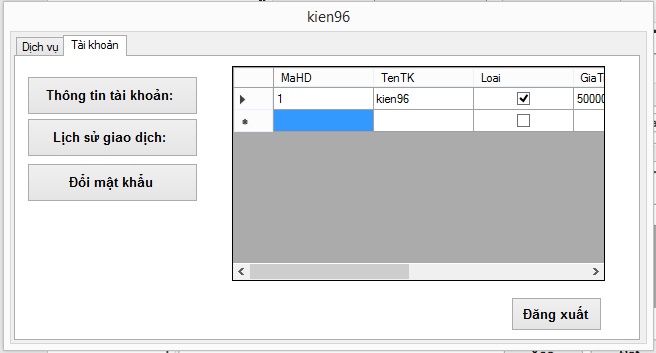
* Đăng nhập
* Đặt dịch vụ
* Tra cứu thông tin tài khoản, lịch sử giao dịch
* Đổi thông tin tài khoản

## b, Mô tả source code

* sử dụng mô hình 3 lớp :
  + Lớp giao diện:
    - 



Khi ấn nút đặt hàng bên giao diện admin sẽ có thông báo. Sử dụng liên kết client- server



* + Lớp business được chứa trong thư mục DAO
    - Lớp Dataprovider() chứa các hàm kết nối , truy vấn với cơ sở dữ liệu :
      * *ExcuteQuery(): trả về bảng dữ liệu*
* private string connectionSTR = @"Data Source=.\SQLexpress;Initial Catalog=QuanLyPhongGame;Integrated Security=True";
* // tra ve bang du lieu theo tham so
* public DataTable ExcuteQuery (string query, object[] parameter = null)
* {
* DataTable data = new DataTable();
* using (SqlConnection connection = new SqlConnection(connectionSTR))
* {
* connection.Open();
* SqlCommand command = new SqlCommand(query, connection);
* if(parameter != null)
* {
* string[] listPara = query.Split(' ');
* int i = 0;
* foreach (string item in listPara)
* {
* if(item.Contains('@'))
* {
* command.Parameters.AddWithValue(item, parameter[i]);
* i++;
* }
* }
* }
* SqlDataAdapter Adapter = new SqlDataAdapter(command);
* Adapter.Fill(data);
* connection.Close();
* return data;
* }
* }
  + - * ExcuteNonQuery(): trả về số trường thành công
* public int ExcuteNonQuery(string query, object[] parameter = null)
* {
* int data = 0;
* using (SqlConnection connection = new SqlConnection(connectionSTR))
* {
* connection.Open();
* SqlCommand command = new SqlCommand(query, connection);
* if (parameter != null)
* {
* string[] listPara = query.Split(' ');
* int i = 0;
* foreach (string item in listPara)
* {
* if (item.Contains('@'))
* {
* command.Parameters.AddWithValue(item, parameter[i]);
* i++;
* }
* }
* }
* data = command.ExecuteNonQuery();
* connection.Close();
* return data;
* }
* }
  + - * *ExcuteScalarQuery(): trả về giá trị ô đầu tiên*
* public object ExcuteScalar(string query, object[] parameter = null)
* {
* object data = 0;
* using (SqlConnection connection = new SqlConnection(connectionSTR))
* {
* connection.Open();
* SqlCommand command = new SqlCommand(query, connection);
* if (parameter != null)
* {
* string[] listPara = query.Split(' ');
* int i = 0;
* foreach (string item in listPara)
* {
* if (item.Contains('@'))
* {
* command.Parameters.AddWithValue(item, parameter[i]);
* i++;
* }
* }
* }
* data = command.ExecuteScalar();
* connection.Close();
* return data;
* }
* }
  + - *Lớp AcountDAO() chứa các hàm trả về các kết quả truy vấn với cơ sở dữ liệu liên quan đến tài khoản.*
* public bool Login(string UserName, string Password)
* {
* string query = "USP\_Login @userName , @passWord";
* DataTable result = DataProvider.Instance.ExcuteQuery(query,new object[] { UserName,Password});
* return result.Rows.Count>0;
* }
* // tao mot store process trong sql đẻ hạn chế lỗi sql injection
* public bool updatePassWordAcount(string password, string newpassword)
* {
* int result = DataProvider.Instance.ExcuteNonQuery("exec USP\_UpdateMK @tenTk , @matkhau , @matkhaumoi",new object[] {frmLogin.tenTK, password,newpassword});
* return result>0;
* }
* public string LoadSoDuTaiKhoan(string tenTK)
* {
* string query = "SELECT SoDuTaiKhoan FROM dbo.TaiKhoan WHERE TenTK='" +tenTK + "'";
* DataTable data = DataProvider.Instance.ExcuteQuery(query);
* return data.Rows[0][0].ToString();
* }
* public bool updateSoDuTaiKhoan()
* {
* string query = "";
* int result = DataProvider.Instance.ExcuteNonQuery(query);
* return result > 0;
* }
* public DataTable LoadThongTinTk( string tenTK)
* {
* string query = "SELECT TenTk, ChuTK, GioiTinh, NgaySinh, SoDienThoai, SoDuTaiKhoan FROM dbo.TaiKhoan Where TenTK='"+tenTK+"'";
* DataTable data = DataProvider.Instance.ExcuteQuery(query);
* return data;
* }
  + - *Lớp LoadMatHangDAO chứa các hàm trả về các kết quả truy vấn liên quan đến KhoHang*
* public List<string> TenHang()
* {
* string query = "select \* from KhoHang";
* DataTable data = DataProvider.Instance.ExcuteQuery(query);
* List<string> List = new List<string>();
* foreach (DataRow item in data.Rows)
* {
* List.Add(item["MaHang"].ToString() + "." + item["TenHang"]);
* }
* return List;
* }
* // ghép tên hàng + mã hàng
* public string Giaban(string tenhang)
* {
* int i = 0;
* // ComboBox cb = sender as ComboBox;
* if (tenhang == null)
* return null;
* i = tenhang.LastIndexOf(".");
* string ma = tenhang.Remove(i);
* string query = "SELECT GiaBan FROM dbo.KhoHang where MaHang='" + ma + "'";
* DataTable data = DataProvider.Instance.ExcuteQuery(query);
* return data.Rows[0][0].ToString();
* }
  + Lớp DataLayer
    - Cơ sở dữ liệu SQL server

Source code :

CREATE DATABASE QuanLyPhongGame

GO

CREATE TABLE TaiKhoan

(

TenTk VARCHAR(100) NOT NULL PRIMARY KEY,

MatKhau VARCHAR(100)NOT NULL,

ChuTK NVARCHAR(100),

GioiTinh NVARCHAR(15),

NgaySinh DATETIME,

SoDienThoai VARCHAR(15),

SoDuTaiKhoan INT

)

GO

CREATE TABLE KhoHang

(

MaHang VARCHAR(50) NOT NULL PRIMARY KEY,

TenHang NVARCHAR(100)NOT NULL,

SoLuong INT,

GiaNhap INT,

GiaBan INT,

)

GO

CREATE TABLE HoaDon

(

MaHD VARCHAR(50) NOT NULL PRIMARY KEY,

TenTK VARCHAR(100) FOREIGN KEY REFERENCES dbo.TaiKhoan(TenTk),

Loai BIT,

GiaTri INT,

Ngay DATE,

Gio TIME,

ChiTiet NVARCHAR(500)

)

GO

CREATE PROC USP\_Login

@userName varchar(100),

@passWord varchar(100)

AS

BEGIN

SELECT \* FROM dbo.TaiKhoan WHERE TenTk =@userName AND MatKhau = @passWord

END

CREATE PROC USP\_UpdateMK

@tenTk varchar(100),

@matkhau varchar(100),

@matkhaumoi varchar(100)

AS

BEGIN

DECLARE @dungMk INT = 0

SELECT @dungMk = COUNT(\*) FROM dbo.TaiKhoan WHERE MatKhau = @matkhau AND TenTk =@tenTk

IF(@dungMk=1)

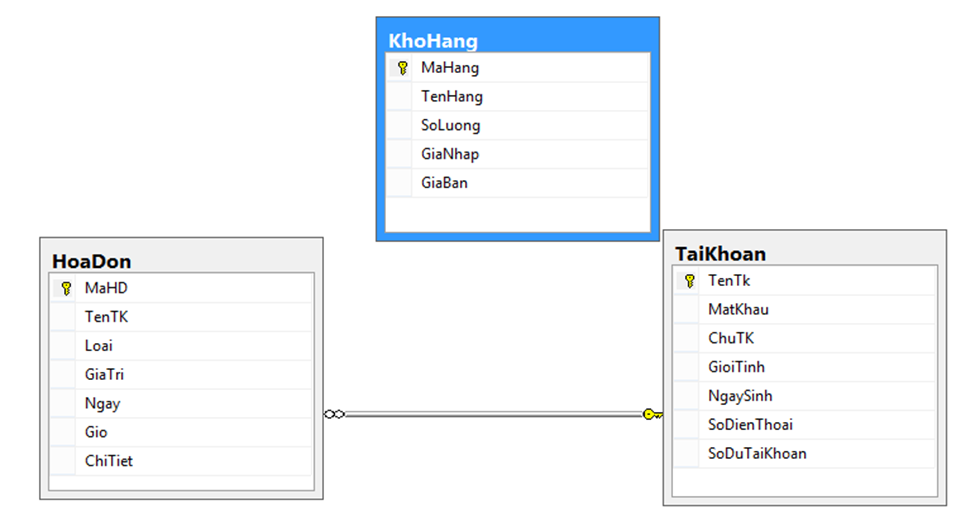
BEGIN

UPDATE dbo.TaiKhoan SET MatKhau=@matkhaumoi WHERE TenTk=@tenTK

end

END

GO

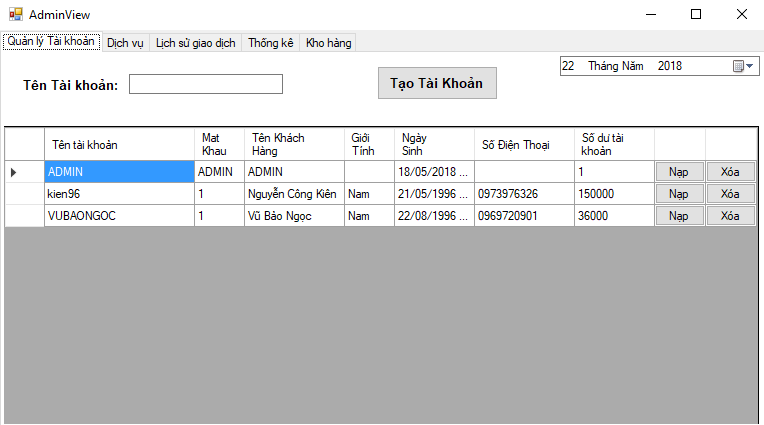


* + - Tạo các stored Procedure trong sql để thực hiện chức năng đăng nhập và sửa mật khẩu

# . GIAO DIỆN ADMIN

## a, Các chức năng và giao diện tương ứng

* **Quản lý tài khoản** . Bao gồm : Tạo mới tài khoản , chỉnh sửa thông tin tài khoản , Tìm kiếm tài khoản theo tên .



Source cho Tạo tài khoản:

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

{

if (TxtMK.Text == TxtMK2.Text)

{

tk.TenTk = TxtTenTK.Text;

tk.MatKhau = TxtMK.Text;

tk.ChuTK = TxtTenKH.Text;

tk.GioiTinh = cbGioiTinh.Text;

tk.NgaySinh = dTPNgaySinh.Value;

tk.SoDienThoai = TxtSoDT.Text;

tk.SoDuTaiKhoan = Convert.ToInt32(TxtNap.Text);

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

db.Entry(tk).State = EntityState.Added;

db.SaveChanges();

}

Clear();

MessageBox.Show("Tạo Tài khoản thành công");

}

else TxtChuThich.Text = "Xác nhận mật khẩu không đúng!";

} private void BtnTaoTK\_Click(object sender, EventArgs e)

{

if (TxtMK.Text == TxtMK2.Text)

{

tk.TenTk = TxtTenTK.Text;

tk.MatKhau = TxtMK.Text;

tk.ChuTK = TxtTenKH.Text;

tk.GioiTinh = cbGioiTinh.Text;

tk.NgaySinh = dTPNgaySinh.Value;

tk.SoDienThoai = TxtSoDT.Text;

tk.SoDuTaiKhoan = Convert.ToInt32(TxtNap.Text);

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

db.Entry(tk).State = EntityState.Added;

db.SaveChanges();

}

Clear();

MessageBox.Show("Tạo Tài khoản thành công");

}

else TxtChuThich.Text = "Xác nhận mật khẩu không đúng!";

}

Source cho Tìm kiếm tài khoản:

private void GetTaiKhoan()

{

dataGridViewAd1.Rows.Clear();

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

for (int i = 0; i < db.TaiKhoans.Count(); i++)

{

int check = 0;

tk = db.TaiKhoans.ToArray<TaiKhoan>()[i];

for (int j = 0; j < txtName.Text.Length; j++)

{

if (tk.TenTk[j] == txtName.Text[j]) check++;

else break;

}

if (check == txtName.Text.Length)

{

dataGridViewAd1.Rows.Add(tk.TenTk, tk.MatKhau, tk.ChuTK, tk.GioiTinh, tk.NgaySinh, tk.SoDienThoai, tk.SoDuTaiKhoan, "Nạp", "Xóa");

}

}

}

}

Source cho Xóa tài khoản:

if (e.ColumnIndex == 8)

{

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

// Tạo messagebox xác nhận chắc chắn việc xóa

if (MessageBox.Show("Bạn chắc chắn muốn xóa tài khoản này ?","Xóa Tài Khoản",MessageBoxButtons.YesNo) == DialogResult.Yes)

{

for (int i = 0; i < db.TaiKhoans.Count(); i++)

{

tk = db.TaiKhoans.ToArray<TaiKhoan>()[i];

if (tk.TenTk == dataGridViewAd1.SelectedCells[0].ToString()) break;

}

// Xóa Tài khoản đang được chọn trong DATABASE

/\*var entry = db.Entry(tk);

if (entry.State == EntityState.Detached)

db.TaiKhoans.Attach(tk);

db.TaiKhoans.Remove(tk);\*/

db.Entry(tk).State = EntityState.Deleted;

db.SaveChanges();

// Xóa tài khoản được chọn trong datagridvView

if (e.RowIndex >= 0) dataGridViewAd1.Rows.RemoveAt(e.RowIndex);

// Tạo messagebox thông báo việc xóa thành công

MessageBox.Show("Đã Xóa Tài Khoản Được Chọn ");

}

}

}

Source cho nạp tiền:

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

{

// Update số tiền mới của Tài Khoản vào DATABASE

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

for (int i = 0; i < db.TaiKhoans.Count(); i ++)

{

tk = db.TaiKhoans.ToArray<TaiKhoan>()[i];

if (tk.TenTk == this.Text) break;

}

tk.SoDuTaiKhoan += Convert.ToInt32(TxtSoLuong.Text);

db.Entry(tk).State = EntityState.Modified;

// Tạo Hóa Đơn và Nhập và DataBase

hd.MaHD = Convert.ToString(db.HoaDons.Count() + 1);

hd.TenTK = "ADMIN";

hd.Loai = true;

hd.GiaTri = Convert.ToInt32(TxtSoLuong.Text);

hd.ChiTiet = "Nạp tiền vào TK :" + this.Text;

DateTimePicker dtp = new DateTimePicker();

hd.Ngaygio = dtp.Value;

db.Entry(hd).State = EntityState.Added;

db.SaveChanges();

}

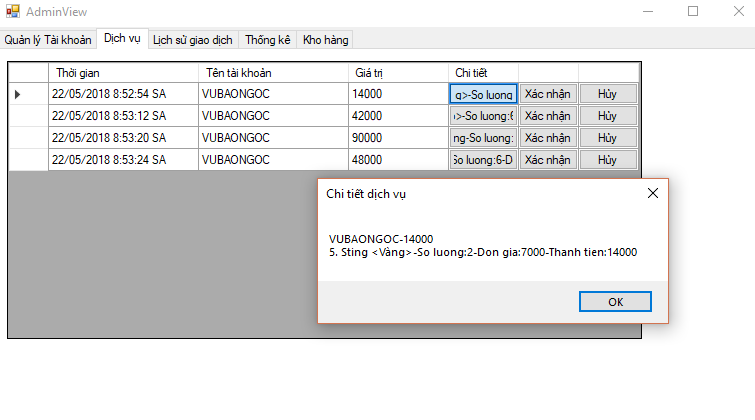
TxtSoLuong.Text = "";

// Tạo MessageBox thông báo xác nhận nạp thành công

MessageBox.Show("Nạp tiền thành công !");

}

* **Quản lý dịch vụ** : Xác nhận dịch vụ , hủy dịch vụ , xem chi tiết dịch vụ



Source code cho xác nhân dịch vụ:

Source code cho hủy dịch vụ:

if (e.ColumnIndex == 5)

{

server.Broadcast(thongbao);

if (e.RowIndex >= 0) dataGridViewAd2.Rows.RemoveAt(e.RowIndex);

}

Source code cho xem chi tiết dịch vụ:

if (e.ColumnIndex == 3)

{

//frmCTHD frm\_CTHD = new frmCTHD();

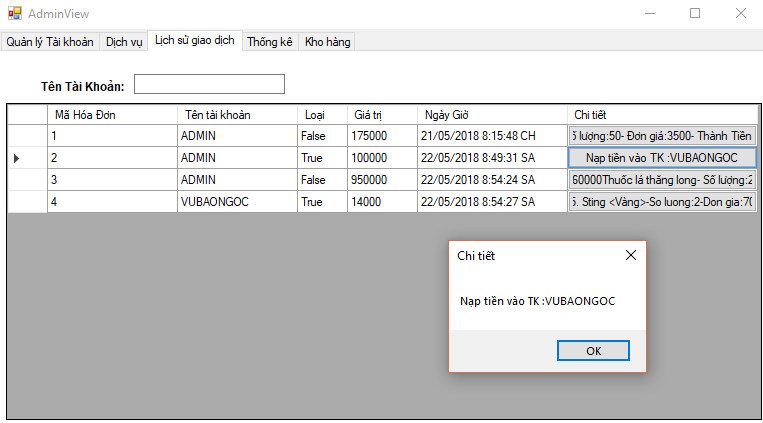
//frm\_CTHD.Show();

string chitiet = dataGridViewAd2.Rows[e.RowIndex].Cells[e.ColumnIndex].Value.ToString();

MessageBox.Show(chitiet, "Chi tiết dịch vụ");

}

* **Lịch sử giao dịch** : Xem các lịch sử giao dịch dưới dạng các hóa đơn



Source code:

private void GetHoaDon()

{

dataGridViewAd3.Rows.Clear();

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

for (int i = 0; i < db.HoaDons.Count(); i++)

{

int check = 0;

hd = db.HoaDons.ToArray<HoaDon>()[i];

for (int j = 0; j < TxtMaHD.Text.Length; j++)

{

if (hd.MaHD[j] == TxtMaHD.Text[j]) check++;

else break;

}

if (check == TxtMaHD.Text.Length)

{

dataGridViewAd3.Rows.Add(hd.MaHD, hd.TenTK, hd.Loai, hd.GiaTri, hd.Ngaygio, hd.ChiTiet);

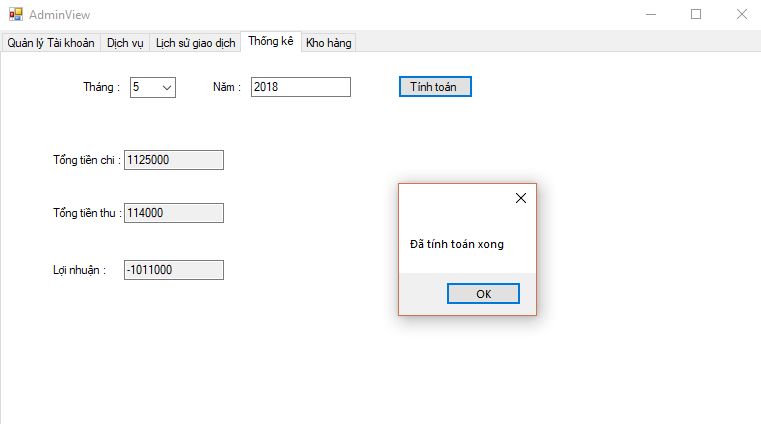
}

}

}

}

* **Thống kê doanh thu theo tháng**

****

Source code:

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

{

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

for (int i = 0; i < db.HoaDons.Count(); i ++ )

{

hdtt = db.HoaDons.ToArray < HoaDon >()[i];

DateTime dt = Convert.ToDateTime(hdtt.Ngaygio.ToString());

if ((dt.Month == Convert.ToInt32(cbThang.Text)) && (dt.Year == Convert.ToInt32(TxtNam.Text)) )

{

if (hdtt.Loai == false)

TxtTongChi.Text = Convert.ToString(Convert.ToInt32(TxtTongChi.Text) + hdtt.GiaTri);

else

TxtTongthu.Text = Convert.ToString(Convert.ToInt32(TxtTongthu.Text) + hdtt.GiaTri);

}

}

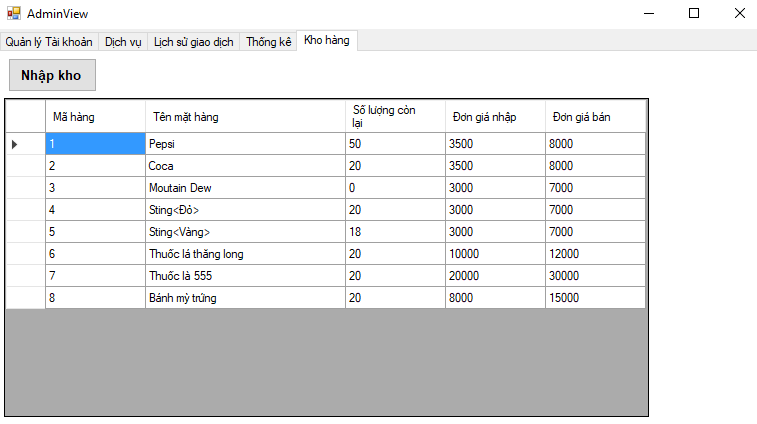
TxtLoiNhuan.Text = Convert.ToString(Convert.ToInt32(TxtTongthu.Text) - Convert.ToInt32(TxtTongChi.Text));

}

MessageBox.Show("Đã tính toán xong");

}

* **Quản lý kho hàng gồm** : Xem thông tin kho hàng , nhập thêm vào kho hàng



Source code xem thông tin kho:

private void GetKhoHang()

{

dataGridViewAd4.AutoGenerateColumns = false;

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

dataGridViewAd4.DataSource = db.KhoHangs.ToList<KhoHang>();

}

}

Source nhập kho:

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

{

// Nhập đơn hàng vào kho

using (QuanLyPhongGameEntities1 db = new QuanLyPhongGameEntities1())

{

for (int i = 0; i < dataGridViewNhap1.Rows.Count; i++)

{

for (int j = 0; j < db.KhoHangs.Count(); j++)

{

kh = db.KhoHangs.ToArray<KhoHang>()[j];

if (kh.TenHang == dataGridViewNhap1.Rows[i].Cells[0].Value.ToString())

{

kh.SoLuong += Convert.ToInt32(dataGridViewNhap1.Rows[i].Cells[1].Value.ToString());

db.Entry(kh).State = EntityState.Modified;

db.SaveChanges();

}

}

// Tạo hóa đơn và đưa và Database

hd.MaHD = Convert.ToString(db.HoaDons.Count() + 1);

hd.TenTK = this.Text;

hd.Loai = false;

hd.GiaTri = Convert.ToInt32(TxtTong.Text);

DateTimePicker dtp = new DateTimePicker();

hd.Ngaygio = dtp.Value;

string tmp = "";

for (int k = dataGridViewNhap1.Rows.Count - 1; k >= 0; k--)

{

tmp = tmp.Insert(0, "\r\n");

tmp = tmp.Insert(0, dataGridViewNhap1.Rows[k].Cells[3].Value.ToString());

tmp = tmp.Insert(0, "- Thành Tiền:");

tmp = tmp.Insert(0, dataGridViewNhap1.Rows[k].Cells[2].Value.ToString());

tmp = tmp.Insert(0, "- Đơn giá:");

tmp = tmp.Insert(0, dataGridViewNhap1.Rows[k].Cells[1].Value.ToString());

tmp = tmp.Insert(0, "- Số lượng:");

tmp = tmp.Insert(0, Convert.ToString(dataGridViewNhap1.Rows[k].Cells[0].Value.ToString()));

}

hd.ChiTiet = tmp;

}

db.Entry(hd).State = EntityState.Added;

db.SaveChanges();

MessageBox.Show("Nhập vào kho hàng thành công");

}

// Clear

dataGridViewNhap1.Rows.Clear();

BtnNhap.Enabled = false;

BtnClear.Enabled = false;

}

## Mô hình hoạt động

Sử dụng Entity Framework . Load csdl lên một class và thực hiện thay đổi , nhập xóa , chọn thông tin của csdl bằng việc tác động tương ứng lên class đó .

Dưới đây là dòng config cho việc sử dụng Entity Framework :

<add name="QuanLyPhongGameEntities1" connectionString="metadata=res://\*/DataBase.csdl|res://\*/DataBase.ssdl|res://\*/DataBase.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=KIENPC\SQLEXPRESS;initial catalog=QuanLyPhongGame;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" /></connectionStrings>

Sử dụng SimpleTcp Class để thực hiện việc trao đổi thông tin giữa máy client và máy chủ . Dưới đây là code cho việc kết nối server với client và trao đổi dữ liêu :

private void Admin\_Load(object sender, EventArgs e)

{

server = new SimpleTcpServer();

server.Delimiter = 0x13;

server.StringEncoder = Encoding.UTF8;

System.Net.IPAddress ip = System.Net.IPAddress.Parse("127.0.0.1");

server.Start(ip, 8088);

server.DataReceived += server\_DataReceived;

}

private void server\_DataReceived(object sender, SimpleTCP.Message e)

{

DateTimePicker dtp = new DateTimePicker();

string Time = Convert.ToString(dtp.Value) ;

string Name = "";

for (int i = 0; e.MessageString[i] != '-'; i++)

Name = Name.Insert(Name.Length, Convert.ToString(e.MessageString[i]));

string giatri = "";

for (int j = e.MessageString.IndexOf('-') + 1; e.MessageString[j] != '\r'; j++)

giatri = giatri.Insert(giatri.Length, Convert.ToString(e.MessageString[j]));

dataGridViewAd2.Invoke((MethodInvoker)delegate ()

{

dataGridViewAd2.AllowUserToAddRows = false;

dataGridViewAd2.Rows.Add(Time, Name, giatri, e.MessageString, "Xác nhận", "Hủy");

});

MessageBox.Show("Có yêu cầu đặt hàng mới", "Thông báo");

}

private void frmCustomer\_Load(object sender, EventArgs e)

{

client1 = new SimpleTcpClient();

client1.Connect("127.0.0.1",8088);

client1.StringEncoder = Encoding.UTF8;

client1.DataReceived += data\_Received;

}

private void data\_Received(object sender, SimpleTCP.Message e)

{

MessageBox.Show(e.MessageString,"Phản hồi dịch vụ");

## }