Mục Lục

[**1.** **Mô tả** 2](#_Toc453760492)

[**2.** **Phân mảnh** 3](#_Toc453760493)

[**2.1.** **Khác Site (phân mảnh rồi chia ra cho mỗi chi nhánh)** 3](#_Toc453760494)

[**2.2.** **Theo từng Site (ở mỗi chi nhánh)** 4](#_Toc453760495)

[**3.** **Lược đồ** 7](#_Toc453760496)

[**3.1.** **Lược đồ ERD toàn cục** 7](#_Toc453760497)

[**3.2.** **Lược đồ ERD data dictionary** 7](#_Toc453760498)

[**3.3.** **Database diagram chi nhánh Hà Nội** 8](#_Toc453760499)

[**3.4.** **Database diagram chi nhánh Hồ Chí Minh** 9](#_Toc453760500)

[**3.5.** **Database diagram data dictionary** 10](#_Toc453760501)

[**4.** **Các chức năng và luồng xử lý** 11](#_Toc453760502)

[**4.1.** **Đăng nhập** 11](#_Toc453760503)

[**4.1.1.** **Đăng nhập Server** 11](#_Toc453760504)

[**4.1.2.** **Đăng nhập chi nhánh** 13](#_Toc453760505)

[**4.2.** **Quản lý Sách** 14](#_Toc453760506)

[**4.2.1.** **Load dữ liệu Sach** 15](#_Toc453760507)

[**4.2.2.** **Thêm, Sửa sách.** 17](#_Toc453760508)

[**4.2.3.** **Xóa Sách** 20](#_Toc453760509)

[**4.2.4.** **Tìm kiếm** 21](#_Toc453760510)

[**4.3.** **Quản lý Nhân viên** 24](#_Toc453760511)

[**4.3.1.** **Chuyển nhân viên** 25](#_Toc453760512)

[**4.4.** **Bán hàng (bán sách)** 28](#_Toc453760513)

[**5.** **Kết luận** 29](#_Toc453760514)

[**5.1.** **Kết quả đạt được** 29](#_Toc453760515)

[**5.2.** **Phần chưa làm được** 29](#_Toc453760516)

1. **Mô tả**

Nhà sách có 2 chi nhánh. Mỗi chi nhánh đều có 1 quản lý và 10 nhân viên sẽ có các công việc khác nhau ứng với từng chức vụ của nhân viên đó, và được quản lý bởi Giám đốc nhà sách.   
Đối tượng phục vụ của nhà sách chủ yếu là học sinh, sinh viên và 1 số ít những khách hàng có nhu cầu đọc các loại sách tham khảo.

Các hoạt động chính của nhà sách là: Bán sách, lưu trữ các dữ liệu liên quan, quản lý sách, nhân viên,nhà cung cấp, nhà xuất bản, khu, kệ để sách, tác giả, thể loại.

Việc quản lý Nhà sách được phân cấp theo từng bộ phận:

* Giám đốc nhà sách:

+ Hoạch định kế hoạch, đưa ra những mục tiêu phát triển cho nhà sách trong một giai đoạn nhất định; lãnh đạo toàn hệ thống nhà sách.

+ Chịu trách nhiệm trước pháp luật về hoạt động của nhà sách, giải quyết các vấn đề mâu thuẩn nội bộ nhà sách (nếu có).

* Đối với từng chi nhánh:
* Quản lý :

+ Quản lý tài chính, sổ sách, nhân sự của từng chi nhánh.

* Nhân viên kho:

+ Kiểm tra đơn hàng khi nhập hàng, báo cáo các vấn đề liên quan đến nhập xuất, hàng bị lỗi.

+ Thống kê sách tồn kho, sách cần thanh lý.

* Nhân viên bán hàng:

+ Kiểm tra và quét mã vạch sách.

+ Thống kê doanh thu của chi nhánh.

+ Xuất hóa đơn, thu tiền và trả tiền cho khách hàng.

* Nhân viên chăm sóc khách hàng.

+ Tìm kiếm, khai thác khách hàng cần cung cấp những loại sách nào.

+ Sắp xếp lại những cuốn sách khách hàng đã đọc và để sai chỗ.

* Quy trình nhập sách:

Nhân viên kho sẽ kiểm tra đơn hàng xem có bị lỗi hay không, nếu bị lỗi sẽ bàn bạc với nhà cung cấp để giải quyết vấn đề, nếu đơn hàng không bị lỗi thì nhân viên kho tiến hành nhận đơn hàng, lập phiếu nhập hàng và làm thủ tục nhận hàng. Sau đó thanh toán hóa đơn và đưa hàng vào kho rồi nhập thông tin hoặc bổ sung sách vào phần mềm.

* Quy trình bán sách:

Khi khách hàng tới quầy thanh toán, nhân viên bán hàng sẽ quét mã vạch đối với từng cuốn sách, in hóa đơn cho khách hàng và thu nhận tiền từ khách hàng.

Các thể loại sách nhà sách kinh doanh chủ yếu là:

* Sách giáo khoa, tham khảo.
* Sách khoa học - kỹ thuật.
* Truyện thiếu nhi.
* Sách học ngoại ngữ.
* Sách kiến thức tổng hợp
* Sách kinh tế, chính trị.
* Từ điển.
* Văn học trong nước.
* Mỗi thể loại sách được bày bán ở mỗi khu khác nhau, trong mỗi khu sẽ có nhiều kệ để sách và có nhân viên chăm sóc, tư vấn khách hàng ở mỗi khu.

Nhà sách cũng có nhiều quy định và chính sách:

+ Giờ mở cửa của nhà sách từ 7h30 đến 21h30.

+ Nhân viên chỉ được đi muộn so với giờ quy định 10 phút.

+ Mỗi nhân viên được đăng kí ca làm, tối thiểu phải làm 6 ca 1 tuần.

* Ca 1: 7h30 đến 12h
* Ca 2: 11h30 21h30

+ Trong 1 quý nhân viên có thời gian làm nhiều nhất sẽ được thưởng 1 ngày lương.

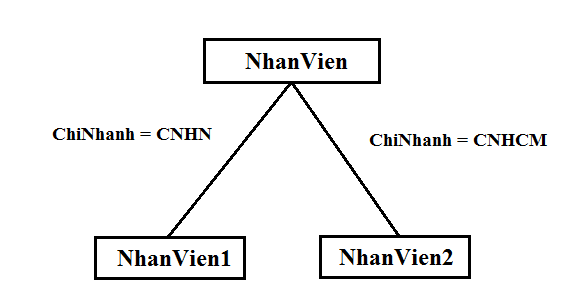
+ Nhân viên sẽ được thưởng tết theo quy định là 1 phần quà do giám đốc phê chuẩn.

+ Đối với những loại sách chỉ bán được < 10% doanh số nhập vào trong 1 năm thì sẽ giảm giá.

+ Đối với những cuốn sách đã cũ thì tiến hành thanh lý.

1. **Phân mảnh**
   1. **Khác Site (phân mảnh rồi chia ra cho mỗi chi nhánh)**

* Quan hệ NhanVien(nhân viên) được phân ra làm 2 mảnh ngang chính theo từng chi nhánh bao gồm:
  + Nhân viên chi nhánh Hà Nội.
  + Nhân viên chi nhánh Hồ Chí Minh.

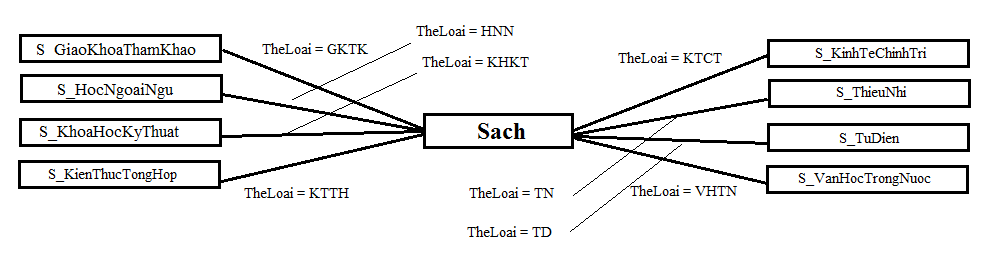


NhanVien1 = бChiNhanh = CNHN NhanVien

NhanVien2 = бChiNhanh = CNHCM NhanVien

NhanVien = NhanVien1 Ս NhanVien2

* 1. **Theo từng Site (ở mỗi chi nhánh)**
* Quan hệ Sach(sách) được phân ra làm 8 mảnh ngang chính theo từng thể loại bao gồm:
  + Mảnh sách giáo khoa tham khảo
  + Mảnh sách học ngoại ngữ.
  + Mảnh sách khoa học kỹ thuật.
  + Mảnh sách kiến thức tổng hợp.
  + Mảnh sách kinh tế chính trị.
  + Mảnh sách thiếu nhi.
  + Mảnh sách từ điển.
  + Mảnh sách văn học trong nước.



S\_GiaoKhoaThamKhao = бTheLoai = GKTK Sach

S\_HocNgoaiNgu = бTheLoai = HNN Sach

S\_KhoaHocKyThuat = бTheLoai = KHKT Sach

S\_KienThucTongHop = бTheLoai = KTTH Sach

S\_KinhTeChinhTri = бTheLoai = KTCT Sach

S\_ThieuNhi = бTheLoai = TN Sach

S\_TuDien = бTheLoai = TD Sach

S\_VanHocTrongNuoc = бTheLoai = VHTN Sach

Sach = S\_GiaoKhoaThamKhao **Ս** S\_HocNgoaiNgu **Ս** S\_KhoaHocKyThuat **Ս** S\_KienThucTongHop **Ս** S\_KinhTeChinhTri **Ս** S\_ThieuNhi **Ս** S\_TuDien **Ս** S\_VanHocTrongNuoc

* Quan hệ CTHDBanSach(chi tiết hóa đơn bán sách) được phân mảnh ngang dẫn xuất từ quan hệ Sach(sách) bao gồm:
  + Chi tiết hóa đơn bán sách sách giáo khoa tham khảo
  + Chi tiết hóa đơn bán sách sách học ngoại ngữ.
  + Chi tiết hóa đơn bán sách sách khoa học kỹ thuật.
  + Chi tiết hóa đơn bán sách sách kiến thức tổng hợp.
  + Chi tiết hóa đơn bán sách sách kinh tế chính trị.
  + Chi tiết hóa đơn bán sách sách thiếu nhi.
  + Chi tiết hóa đơn bán sách sách từ điển.
  + Chi tiết hóa đơn bán sách sách văn học trong nước.

CTHDBanSachGKTK = CTHDBanSach ǁ˃< MaSach = MaSach S\_GiaoKhoaThamKhao

CTHDBanSachHNN = CTHDBanSach ǁ˃< MaSach = MaSach S\_HocNgoaiNgu

CTHDBanSachKHKT = CTHDBanSach ǁ˃< MaSach = MaSach S\_KhoaHocKyThuat

CTHDBanSachKTTH = CTHDBanSach ǁ˃< MaSach = MaSach S\_KienThucTongHop

CTHDBanSachKTCT = CTHDBanSach ǁ˃< MaSach = MaSach S\_KinhTeChinhTri

CTHDBanSachTN = CTHDBanSach ǁ˃< MaSach = MaSach S\_ThieuNhi

CTHDBanSachTD = CTHDBanSach ǁ˃< MaSach = MaSach S\_TuDien

CTHDBanSachVHTN = CTHDBanSach ǁ˃< MaSach = MaSach S\_VanHocTrongNuoc

CTHDBanSach = CTHDBanSachGKTK **Ս** CTHDBanSachHNN **Ս** CTHDBanSachKHKT **Ս** CTHDBanSachKTTH **Ս** CTHDBanSachKTCT **Ս** CTHDBanSachTN **Ս** CTHDBanSachTD **Ս** CTHDBanSachVHTN

* Quan hệ CTHDNhapSach(chi tiết hóa đơn nhập sách) được phân mảnh ngang dẫn xuất từ quan hệ Sach(sách) bao gồm:
  + Chi tiết hóa đơn nhập sách sách giáo khoa tham khảo
  + Chi tiết hóa đơn nhập sách sách học ngoại ngữ.
  + Chi tiết hóa đơn nhập sách sách khoa học kỹ thuật.
  + Chi tiết hóa đơn nhập sách sách kiến thức tổng hợp.
  + Chi tiết hóa đơn nhập sách sách kinh tế chính trị.
  + Chi tiết hóa đơn nhập sách sách thiếu nhi.
  + Chi tiết hóa đơn nhập sách sách từ điển.
  + Chi tiết hóa đơn nhập sách sách văn học trong nước.

CTHDNhapSachGKTK = CTHDNhapSach ǁ˃< MaSach = MaSach S\_GiaoKhoaThamKhao

CTHDNhapSachHNN = CTHDNhapSach ǁ˃< MaSach = MaSach S\_HocNgoaiNgu

CTHDNhapSachKHKT = CTHDNhapSach ǁ˃< MaSach = MaSach S\_KhoaHocKyThuat

CTHDNhapSachKTTH = CTHDNhapSach ǁ˃< MaSach = MaSach S\_KienThucTongHop

CTHDNhapSachKTCT = CTHDNhapSach ǁ˃< MaSach = MaSach S\_KinhTeChinhTri

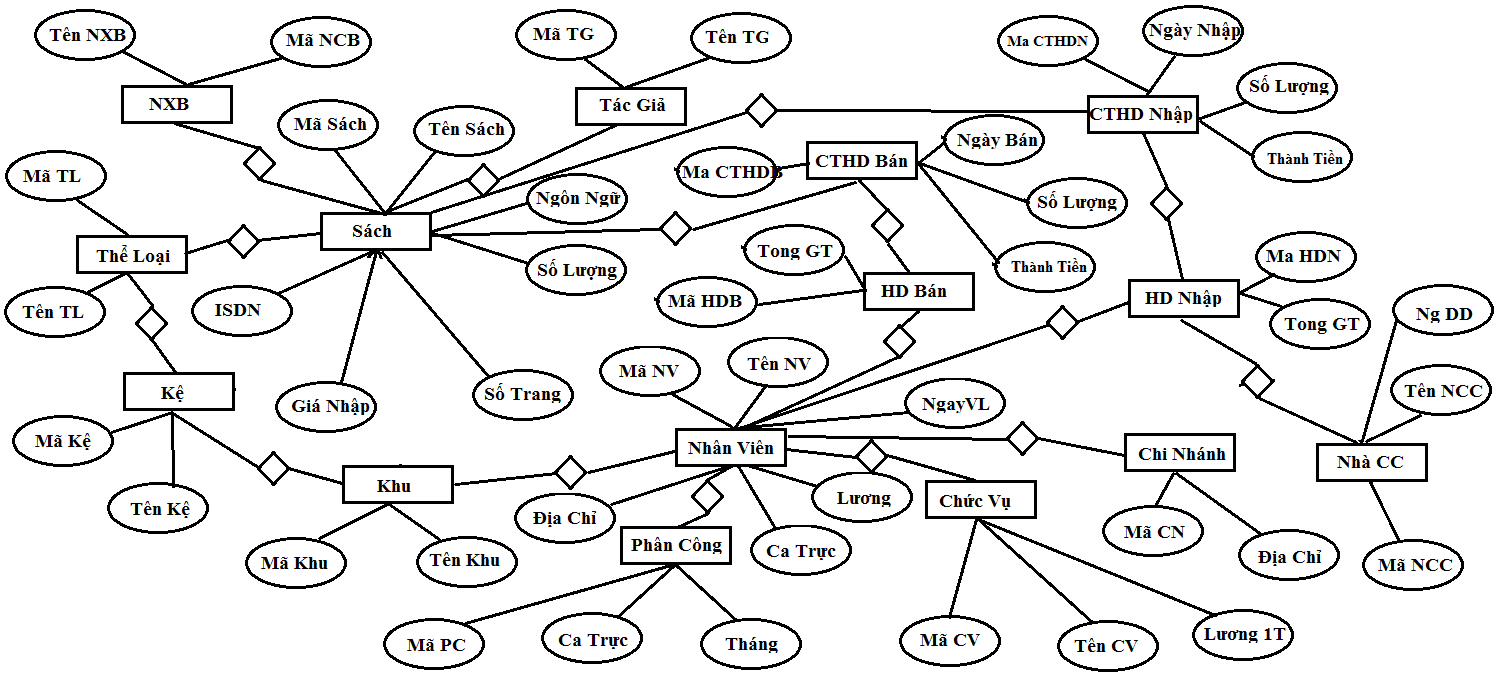
CTHDNhapSachTN = CTHDNhapSach ǁ˃< MaSach = MaSach S\_ThieuNhi

CTHDNhapSachTD = CTHDNhapSach ǁ˃< MaSach = MaSach S\_TuDien

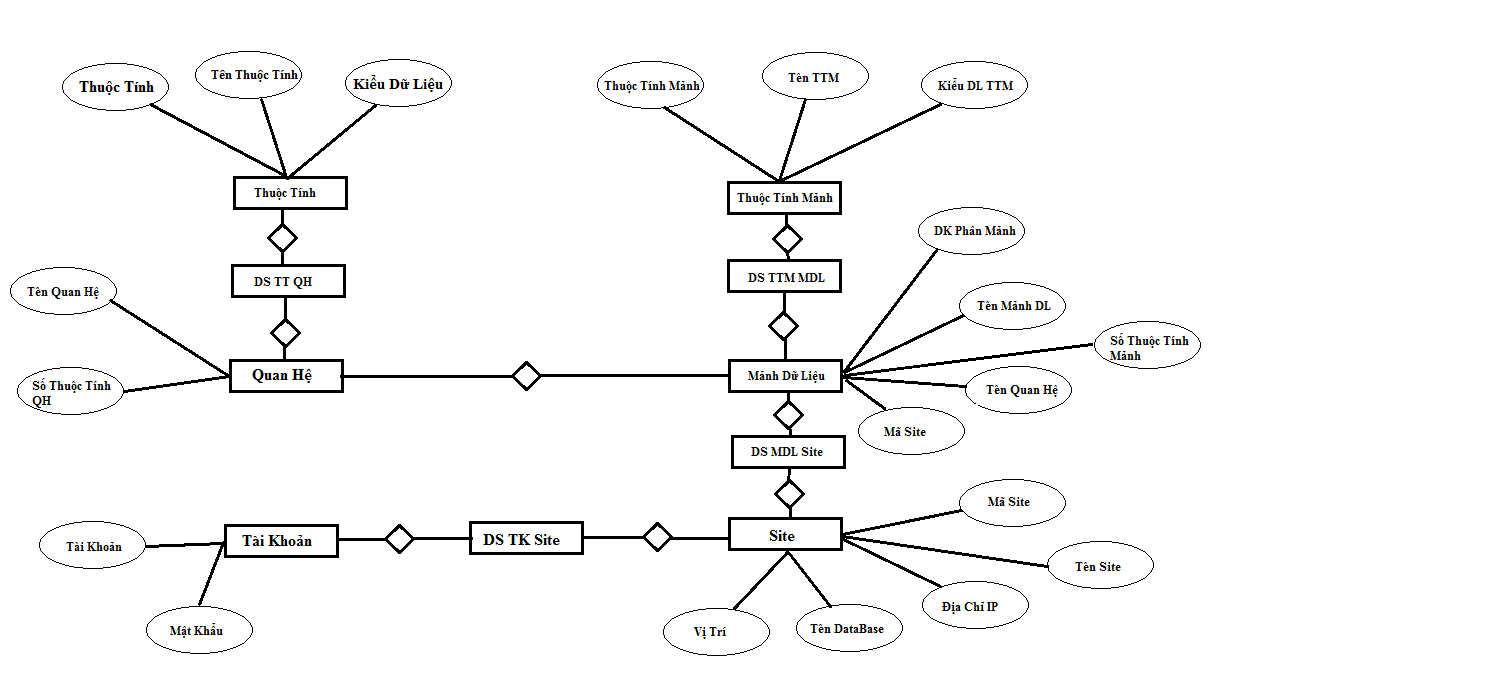
CTHDNhapSachVHTN = CTHDNhapSach ǁ˃< MaSach = MaSach S\_VanHocTrongNuoc

CTHDNhapSach = CTHDNhapSachGKTK **Ս** CTHDNhapSachHNN **Ս** CTHDNhapSachKHKT **Ս** CTHDNhapSachKTTH **Ս** CTHDNhapSachKTCT **Ս** CTHDNhapSachTN **Ս** CTHDNhapSachTD **Ս** CTHDNhapSachVHTN

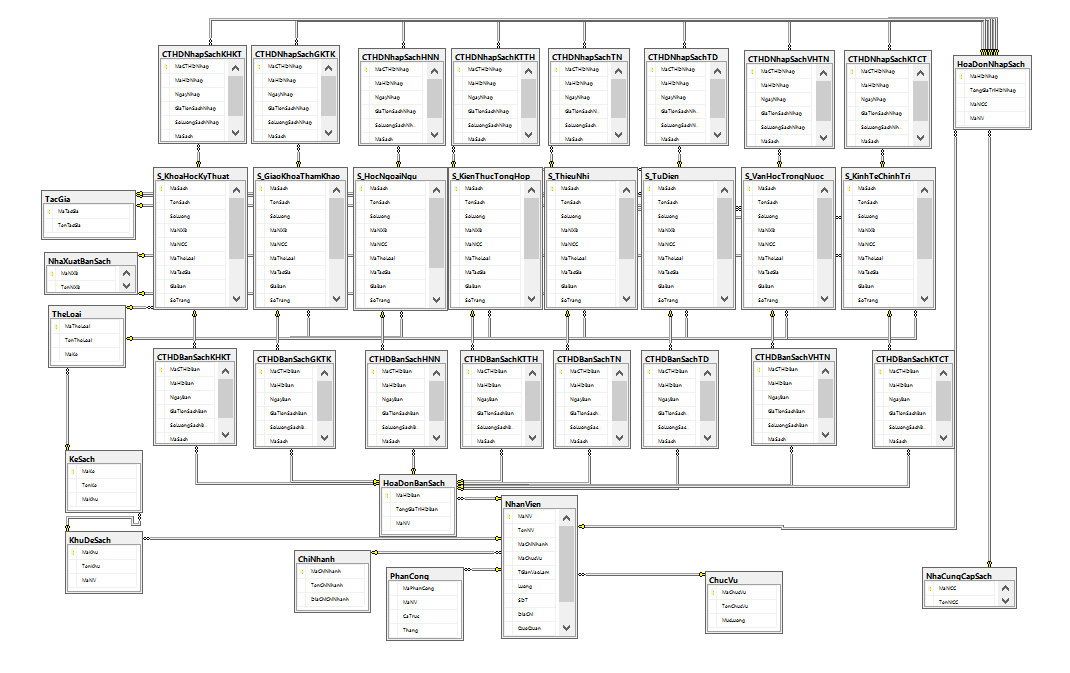
1. **Lược đồ**
   1. **Lược đồ ERD toàn cục**



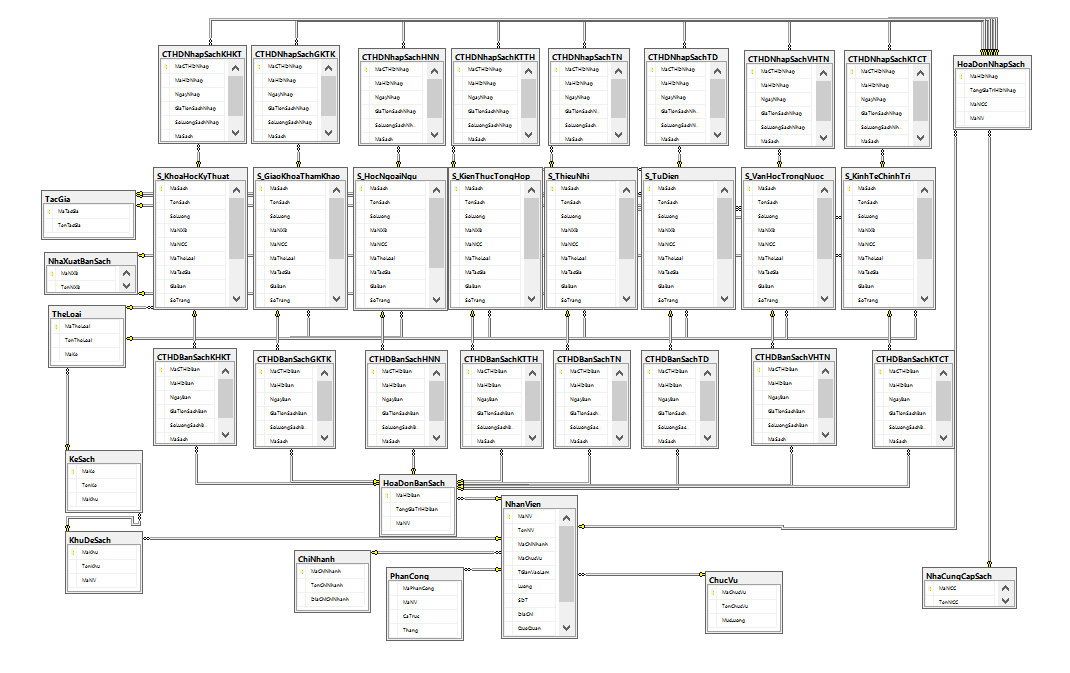
* 1. **Lược đồ ERD data dictionary**



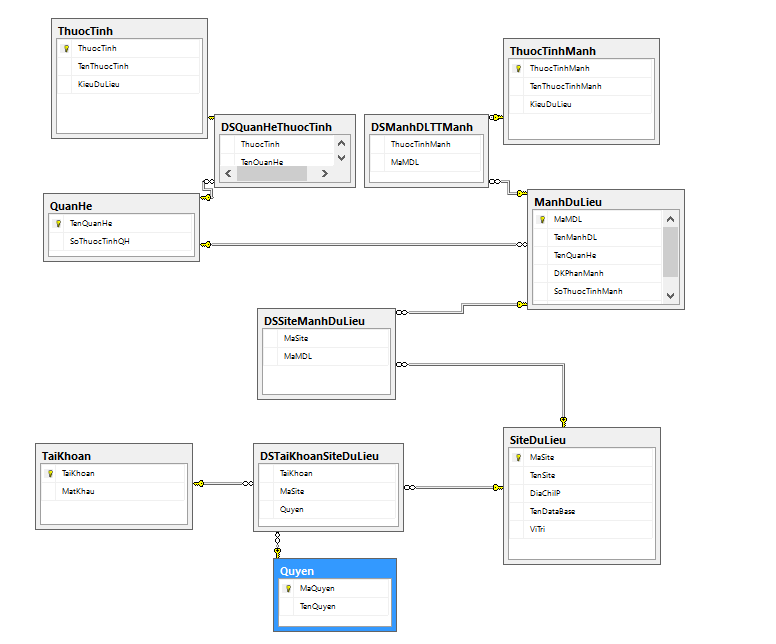
* 1. **Database diagram chi nhánh Hà Nội**

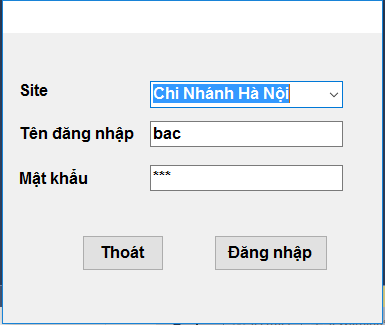


* 1. **Database diagram chi nhánh Hồ Chí Minh**



* 1. **Database diagram data dictionary**



1. **Các chức năng và luồng xử lý**
   1. **Đăng nhập**
      1. **Đăng nhập Server**

* Thao tác đăng nhập

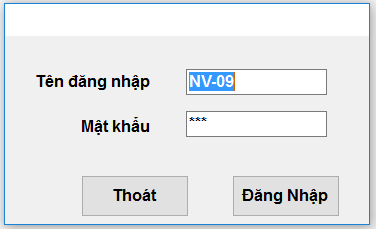
|  |
| --- |
| private void btnLogin\_Click(object sender, EventArgs e)  {  User us = new User();  String PassCSDL = us.LayPass(txtUser.Text);  String SiteCSDL = us.LayMaSite(txtUser.Text);  if (txtPass.Text.Trim() == PassCSDL.Trim() &&  cbxSite.SelectedValue.ToString().Trim() == SiteCSDL.Trim())  {  //lấy pass + tên user  TenDangNhap = txtUser.Text.Trim();  PassWord = txtPass.Text.Trim();  MaSite = cbxSite.SelectedValue.ToString().Trim();  //  MessageBox.Show("Đăng nhập thành công !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close();  }  else  {  MessageBox.Show("Đăng nhập thất bại !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* Hàm *LayPass*

|  |
| --- |
| public String LayPass(string TenUs)  {  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL.stringConnection = DAL.connStringDic;  DataSet dsUs = db.ExecuteQueryDataSet(  "select MatKhau from TaiKhoan where TaiKhoan='" + TenUs + "'",  CommandType.Text);  DataTable dtUs = dsUs.Tables[0];  if (dtUs == null)  return "!!!";  return dtUs.Rows[0][0].ToString();  }  catch (Exception)  {  return "!!!";  }  } |

* Hàm *LayMaSite*

|  |
| --- |
| public String LayMaSite(string TenUs)  {  string QuerySQl = "select MaSite from DSTaiKhoanSiteDuLieu where TaiKhoan='" + TenUs + "'";  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL.stringConnection = DAL.connStringDic;  DataSet dsUs = db.ExecuteQueryDataSet(QuerySQl,  CommandType.Text  );  DataTable dtUs = dsUs.Tables[0];  if (dtUs == null)  return "!!!";  return dtUs.Rows[0][0].ToString();  }  catch (Exception)  {  return "!!!";  }  } |

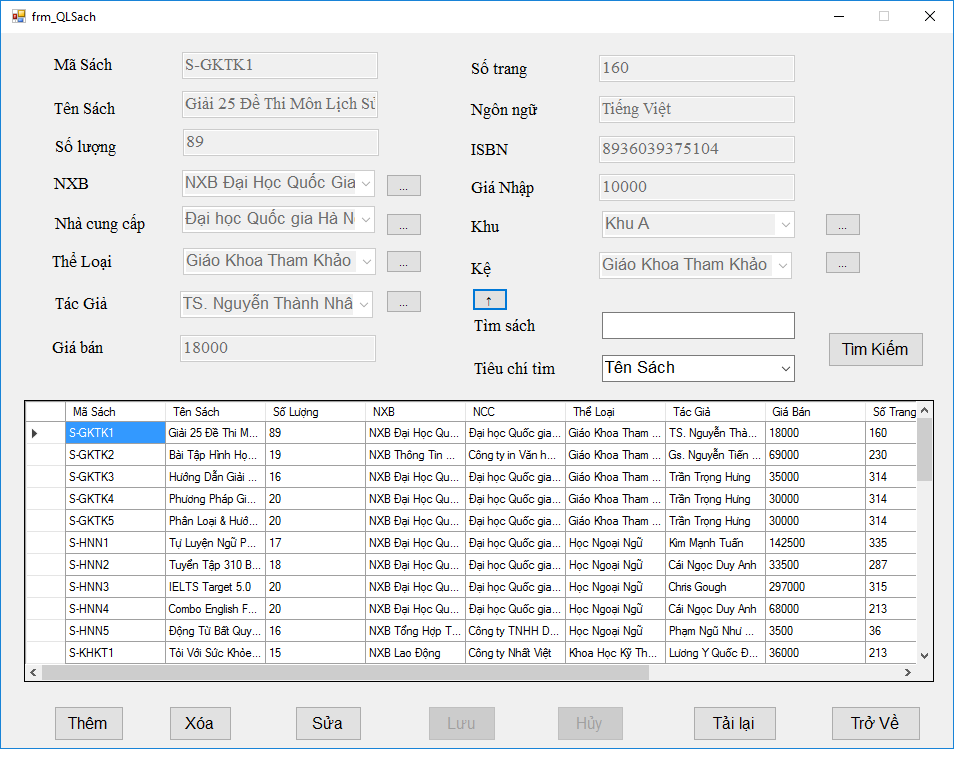
* + 1. **Đăng nhập chi nhánh**
* Thao tác đăng nhập
* Thao tác đăng nhập

|  |
| --- |
| private void btnLogin\_Click(object sender, EventArgs e)  {  User us = new User();  User.TenDangNhap = TenDangNhap;  User.MatKhau = PassWord;  string PassCSDL= us.LayPassData(MaSite, txtUser.Text.Trim());  if (txtPass.Text.Trim() == PassCSDL.Trim())  {  //  MessageBox.Show("Đăng nhập thành công !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  MaNV = txtUser.Text.ToString().Trim();  this.Close();  }  else  {  MessageBox.Show("Đăng nhập thất bại !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* Hàm *LayPassData*

|  |
| --- |
| public String LayPassData(string MaSite,string MaNV)  {  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSite);  String Query = "select MatKhau from NhanVien where MaNV='" + MaNV + "'";  //chuỗi két nối khác  DAL.stringConnection = conn;  DataSet dsPa = db.ExecuteQueryDataSet(Query, CommandType.Text);  //  DataTable dtPa = dsPa.Tables[0];  //  return dtPa.Rows[0][0].ToString();  }  catch (Exception)  {  return "!!!";  }  } |

* 1. **Quản lý Sách**



* + 1. **Load dữ liệu Sách**
* Hàm *LoadDuLieu*

|  |
| --- |
| void LoadDuLieu()  {  try  {  Sach sach = new Sach();  Sach.TenDangNhap = TenDangNhap;  Sach.MatKhau = PassWord;  Sach.MaSite = MaSite;  //  DataTable dtSach = sach.LayDLBangSach();  //  dgvSach.DataSource = dtSach;  }  catch (Exception)  { }  } |

* Hàm *LayDLBangSach* (Nằm trong lớp thư viện BAL)

|  |
| --- |
| //lấy toàn bộ Dl của sách  public DataTable LayDLBangSach()  {  DAL da = new DAL();  DataTable dtKQ = new DataTable();  //các mảnh chứa DL  DataTable DSManhQH = da.LayDSManhQuanHe("Sach", MaSite);  //thuộc tính của bảng Sách  string ThuocTinh = da.LayThuocTinhStringManhMaSite ("Sach", MaSite);  //câu try vấn  string Query = "";  for (int i=0;i<DSManhQH.Rows.Count;i++)  {  if (i != DSManhQH.Rows.Count - 1)  Query += "Select " + ThuocTinh + " from " + DSManhQH.Rows[i][0].ToString().Trim() + " where TenSach not like '!!!%'" + " union ";  else  Query += "Select " + ThuocTinh + " from " + DSManhQH.Rows[i][0].ToString().Trim() + " where TenSach not like '!!!%'";  }  //chuỗi kết nối  String conn = da.getConnectStrSite(MaSite);  DAL.stringConnection = conn;  //thực thi  DataSet dsSach = da.ExecuteQueryDataSet(Query, CommandType.Text);  //table  dtKQ = dsSach.Tables[0];  return dtKQ;  } |

* Hàm *LayDSManhQuanHe* (Trong lớp thư viện DAL)

|  |
| --- |
| //lấy các mảnh của quan hệ  public DataTable LayDSManhQuanHe(string QuanHe, string MaSite)  {  try  {  //  DAL da = new DAL();  DAL.stringConnection = DAL.connStringDic;  //câu truy vấn  string QueRy = "select TenManhDL,DKPhanManh" +  " from ManhDuLieu " +  " where TenQuanHe = '" + QuanHe + "' and MaSite= '" + MaSite + "' ";  DataTable dtKQ = da.ExecuteQueryDataSet(QueRy, CommandType.Text).Tables[0];    return dtKQ;  }  catch(Exception)  {  return null;  }  } |

* Hàm *LayDSManhQuanHe* (Trong lớp thư viện DAL)

|  |
| --- |
| public String LayThuocTinhStringManhMaSite(String TenBang, string maSite)  {  DAL db = new DAL();  DAL.stringConnection = connStringDic;  String querySQL = "select ThuocTinhManh from DSManhDLTTManh where MaMDL= (Select MaMDL From ManhDuLieu where TenManhDL = '" + TenBang + "'and MaSite='"+maSite+"')";  DataTable dtSach = db.ExecuteQueryDataSet(querySQL, CommandType.Text).Tables[0];  //  String SeZone = "";  String SelectZone = "";  //  foreach (DataRow sa in dtSach.Rows)  {  SeZone += sa[0].ToString().Trim() + ",";  }  //bỏ dấu phẩy  int length = SeZone.Length;  for (int i = 0; i < length - 1; i++)  {  SelectZone += SeZone[i];  }  //  return SelectZone;  } |

* Hàm *getConnectStrSite* (Trong lớp thư viện DAL)

|  |
| --- |
| //lấy chuỗi kết nối dựa trên site và server  public string getConnectStrSite(String MaSite )  {  string str = "";  DAL.stringConnection = connStringDic;  string IP = "";  string NameDB = "";  //  String querySQL="select MaSite,TenSite,DiaChiIP,TenDataBase from SiteDuLieu where MaSite= '"+MaSite+"'";  DAL db = new DAL();  DataTable dt = new DataTable();  dt = db.ExecuteQueryDataSet(querySQL,CommandType.Text).Tables[0];  foreach (DataRow t in dt.Rows)  {  IP = t[2].ToString().Trim();  NameDB = t[3].ToString().Trim();  }  str = String.Format("Server = {0}; Database = {1} ; uid = {2}; pwd = {3}; Integrated Security = False", IP, NameDB, User, Pass);  //  return str;  } |

* + 1. **Thêm, Sửa sách.**
* Thao tác lưu dữ liệu khi thêm sửa

|  |
| --- |
| private void btnLuu\_Click(object sender, EventArgs e)  {  try  {  //lớp sách  Sach sach = new Sach();  Sach.TenDangNhap = TenDangNhap;  Sach.MatKhau = PassWord;  Sach.MaSite = MaSite;  //  if (Them)  {  // //  bool tl = sach.ThemSach(MaSach,TenSach,SoLuong,NXB,NCC,TheLoai,  TacGia,GiaBan,SoTrang,NgonNgu,ISBN,GiaNhap);  if (tl)  {  MessageBox.Show("Thêm thành công !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  MessageBox.Show("Đã xảy ra lỗi!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  else//sửa  {  //xử lý  //DL nhập vào    if (TenSachVao == TenSachCSDL && SoLuongVao ==  SoLuongCSDL && NXBVao == NXBCSDL && TheLoaiVao == TheLoaiCSDL  && TacGiaVao == TacGiaCSDL && GiaBanVao == GiaBanCSDL  && SoTrangVao == SoTrangCSDL  && NgonNguVao == NgonNguCSDL &&  ISBNVao == ISBNCSDL && GiaNhapVao == GiaNhapCSDL)  {  MessageBox.Show("Cập nhật thành công !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  LoadDuLieu();  }  else  {  bool tl= sach.CapNhatSach(MaSachVao, TenSachVao  ,SoLuongVao,NXBVao, NCCVao, TheLoaiVao, TacGiaVao, GiaBanVao,  SoTrangVao, NgonNguVao, ISBNVao, GiaNhapVao);  if (tl)  {  MessageBox.Show("Cập nhật thành công !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  MessageBox.Show("Đã xảy ra lỗi!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  catch (Exception)  {  MessageBox.Show("Đã xảy ra lỗi trong quá trình thêm !!!", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  }  } |

* Hàm bổ trợ *ThemSach*

|  |
| --- |
| //thêm sách mới  public bool ThemSach(string MaSach, string TenSach, string SoLuong,  string NXB, string NCC, string TheLoai,  string TacGia, string GiaBan, string SoTrang, string NgonNgu, string ISBN, string GiaNhap)  {  string err="";  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSite);  //thuộc tính của Sách  string ThuocTinhSach = da.LayThuocTinhStringManhMaSite ("Sach", MaSite);  //các mảnh chứa DL  DataTable DSManhQH = da.LayDSManhQuanHe("Sach", MaSite);  //tên mảnh thêm vào  string TenManh = "";  //tìm tên mảnh  for (int i = 0; i < DSManhQH.Rows.Count; i++)  {  string LoaiTrongDK = LayLoaiTrongDK(DSManhQH.Rows[i][0].ToString().Trim()).ToString().Trim();  //  if(LoaiTrongDK==TheLoai.Trim())  {  TenManh = DSManhQH.Rows[i][0].ToString().Trim();  i = DSManhQH.Rows.Count;  }  }  //  string Query = "INSERT INTO "+TenManh+" VALUES ('" + MaSach +  "', N'" + TenSach + "' , " + SoLuong + " , '" + NXB + "' , '" + NCC + "' , '" + TheLoai + "' , '"  + TacGia + "' , '" + GiaBan + "' , " + SoTrang + " , N'" + NgonNgu + "' , '" + ISBN + "' , " + GiaNhap + " )";  DAL.stringConnection = conn;  return da.MyExecuteNonQuery(Query, CommandType.Text,ref err);  }  catch (Exception)  {  return true;  }  } |

* Hàm bổ trợ *SuaSach*

|  |
| --- |
| //cập nhật sách mới  public bool CapNhatSach(string MaSach, string TenSach, string SoLuong, string NXB, string NCC, string TheLoai,  string TacGia, string GiaBan, string SoTrang, string NgonNgu, string ISBN, string GiaNhap)  {  string err = "";  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSite);  //  string TTTSachach = da.LayThuocTinhStringManhMaSite ("Sach", MaSite);  string TenManh = ManhChuaDuLieu(MaSach);  //  string Query = "update " + TenManh + " " +  " set TenSach = N'"+TenSach+"',SoLuong="+SoLuong+" ," +  " MaNXB = '"+NXB+"',MaNCC = '"+NCC+"',MaTheLoai = '"+TheLoai+"',MaTacGia = '"+TacGia+"',GiaBan = '"+GiaBan+"'," +  " SoTrang = "+SoTrang+",NgonNgu = N'"+NgonNgu+"',ISBN = '"+ISBN+"',GiaNhap = "+GiaNhap+"" +  " where MaSach = '"+MaSach+"'";  DAL.stringConnection = conn;  return da.MyExecuteNonQuery(Query, CommandType.Text, ref err);  }  catch (Exception)  {  return false;  }  } |

* + 1. **Xóa Sách**
* Thao tác lưu dữ liệu khi xóa

|  |
| --- |
| private void btnXoa\_Click(object sender, EventArgs e)  {  //lớp sach  Sach sach = new Sach();  Sach.TenDangNhap = TenDangNhap;  Sach.MatKhau = PassWord;  Sach.MaSite = MaSite;  //  //xử lý  //DL  string MaSachVao = txtMaSach.Text.Trim();  string TenSachVao = txtTenSach.Text.Trim();  bool tl = sach.XoaSach(MaSachVao, TenSachVao);  if (tl)  {  MessageBox.Show("Xóa thành công !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  }  else  MessageBox.Show("Đã xảy ra lỗi trong quá trình xóa !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);  LoadDuLieu();  } |

* Hàm *XoaSach*

|  |
| --- |
| //xóa một cuốn sách  public bool XoaSach(string MaSach, string TenSach)  {  string err = "";  try{  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSite);  string TTTSachach = da.LayThuocTinhStringManhMaSite ("Sach", MaSite);  Sach sach=new Sach();  string ManhChuaDL=sach.ManhChuaDuLieu(MaSach);  string Query = "update "+ManhChuaDL+  " set TenSach = N'!!!" + TenSach + "'"+  " where MaSach = '" + MaSach + "'";  DAL.stringConnection = conn;  bool b= da.MyExecuteNonQuery(Query, CommandType.Text, ref err);  return b;  }  catch (Exception)  {  return false;  }  } |

* + 1. **Tìm kiếm**
* Các thao tác dữ liệu khi tìm kiếm

|  |
| --- |
| private void btnTKSach\_Click(object sender, EventArgs e)  {  //lớp sách  Sach sach = new Sach();  Sach.MatKhau = PassWord;  Sach.MaSite = MaSite;  Sach.TenDangNhap = TenDangNhap;  //  DataSet dsTK;  DataTable dtTK = null;  if (cbxTieuChiTim.Text != "Thể Loại")  {  try  {  string TenThuocTinh = "";  //lấy tên thuộc tính  for (int i = 0; i < dgvSach.Columns.Count; i++)  {  if (dgvSach.Columns[i].HeaderText == cbxTieuChiTim.Text)  TenThuocTinh = dgvSach.Columns[i].Name;  }  //lấy tên bảng  string TenBang = sach.TimTenBang(TenThuocTinh);  //  //  dsTK = sach.TimKiem(txtTimKiem.Text.Trim(), TenBang.Trim(), TenThuocTinh.Trim());  dgvSach.Refresh();  dgvSach.DataSource = dsTK.Tables[0];  }  catch (Exception)  {  dgvSach.DataSource = null;  }  }  else  {  if (cbxTheLoaiDuocTim.Text != "Chọn thể loại")  {  dtTK = sach.TimTheoTheLoai(cbxTheLoaiDuocTim.SelectedValue.ToString());  dgvSach.Refresh();  dgvSach.DataSource = dtTK;  //chuyển đổi DL  ChuyenDoiDL();  }  }  //chuyển đổi DL  ChuyenDoiDL();  } |

* Hàm *TimKiem*

|  |
| --- |
| //tìm kiếm sách  public DataSet TimKiem(string chuoiTim,string Bang,string ThuocTinh)  {  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSite);  //thuộc tính của bảng sách  string ThuocTinhSachThem = da.LayThuocTinhStringThemManhMaSite("Sach", MaSite);    //thuộc tính chung  string ThuocTinhChung = da.LayThuocTinhChung("Sach", Bang, MaSite);  //thuộc tính của sách  string ThuocTinhSach = da.LayThuocTinhStringManhMaSite("Sach", MaSite);  //các mảnh chứa DL  DataTable DSManhQH = da.LayDSManhQuanHe("Sach", MaSite);  //câu try vấn  string QuerySach = "";  for (int i = 0; i < DSManhQH.Rows.Count; i++)  {  if (i != DSManhQH.Rows.Count - 1)  QuerySach += "Select " + ThuocTinhSach + " from " + DSManhQH.Rows[i][0].ToString().Trim() + " union ";  else  QuerySach += "Select " + ThuocTinhSach + " from " + DSManhQH.Rows[i][0].ToString().Trim();  }  //  //câu try vấn  string Query = "";  //  if (Bang != "Sach")  {  if (ThuocTinhChung == ThuocTinh)  Query = "select distinct " + ThuocTinhSachThem +  " from " + Bang + ", ("+QuerySach+") as Sach where Sach." + ThuocTinhChung + " = " + Bang + "." + ThuocTinhChung +  " and " + " Sach." + ThuocTinh + " like N'%" + chuoiTim + "%' and TenSach not like '!!!%'";  else  {  Query = "select distinct " + ThuocTinhSachThem +  " from " + Bang + ",(" + QuerySach + ") as Sach where Sach." + ThuocTinhChung + " = " + Bang + "." + ThuocTinhChung +  " and " + ThuocTinh + " like N'%" + chuoiTim + "%' and TenSach not like '!!!%'";  }  }  else  {  Query = "select distinct " + ThuocTinhSachThem +  " from (" + QuerySach + ") as Sach where Sach." + ThuocTinh + " like N'%" + chuoiTim + "%' and TenSach not like '!!!%'";  }  DAL.stringConnection = conn;  DataSet dsTK = da.ExecuteQueryDataSet(Query, CommandType.Text);  //  return dsTK;  }  catch (Exception)  {  return null;  }  } |

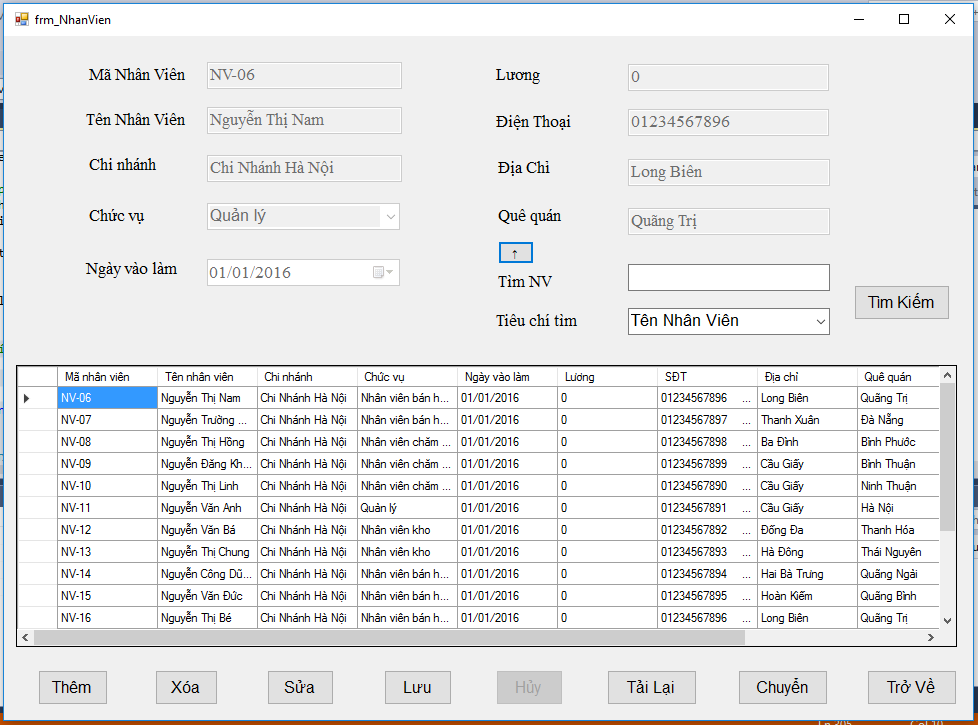
* Hàm *LayThuocTinhStringThemManhMaSite*

|  |
| --- |
| public String LayThuocTinhStringThemManhMaSite(String TenBang, string MaSite)  {  DAL db = new DAL();  DAL.stringConnection = connStringDic;  String querySQL = "select ThuocTinhManh from DSManhDLTTManh where MaMDL=(Select MaMDL from ManhDuLieu where TenManhDL = '" + TenBang + "' and MaSite='"+MaSite+"')";  DataTable dtSach = db.ExecuteQueryDataSet(querySQL, CommandType.Text).Tables[0];  //  String SeZone = "";  String SelectZone = "";  //  foreach (DataRow sa in dtSach.Rows)  {  SeZone += TenBang + "." + sa[0].ToString().Trim() + ",";  }  //bỏ dấu phẩy  int length = SeZone.Length;  for (int i = 0; i < length - 1; i++)  {  SelectZone += SeZone[i];  }  //  return SelectZone;  } |

* Hàm *LayThuocTinhChung*

|  |
| --- |
| //lấy thuộc tính chung của 2 bảng  public string LayThuocTinhChung(String TenBang1, String TenBang2, string MaSite)  {  DAL da = new DAL();  DataTable dtBang1 = da.LayThuocTinhDataSet(TenBang1, MaSite).Tables[0];  DataTable dtBang2 = da.LayThuocTinhDataSet(TenBang2, MaSite).Tables[0];  foreach (DataRow b1 in dtBang1.Rows)  {  foreach (DataRow b2 in dtBang2.Rows)  {  if (b1[0].ToString().Trim() == b2[0].ToString().Trim())  return b1[0].ToString().Trim();  }  }  return "!!!";  } |

* 1. **Quản lý Nhân viên**



Phần Thêm, Sửa, Xóa, Tìm kiếm giống phần Thêm, Sửa, Xóa,Tìm kiếm bên Quản lý Sách.

* + 1. **Chuyển nhân viên**



* Thao tác

|  |
| --- |
| private void btnChuyen\_Click(object sender, EventArgs e)  {  //  NhanVien nhanvien = new NhanVien();  NhanVien.TenDangNhap = TenDangNhap;  NhanVien.MatKhau = PassWord;  NhanVien.MaSite = MaSite;  //  String maSiteDich = cbxChiNhanh.SelectedValue.ToString().Trim();  //  string NgayVaolam = System.DateTime.Now.Date.ToString("MM/dd/yyyy");  string maChiNhanh = nhanvien.LayMaChiNhanhODic(maSiteDich, cbxChiNhanh.Text.Trim()).Trim();  double luong = 0;  int SoCaTruc = 0;  String matKhau = "123";  int SoNV = nhanvien.DemSoNhanVien(maSiteDich);  SoNV = SoNV + 1;  MaNVF = "NV-" + SoNV;  bool tl = nhanvien.ChuyenNV(MaNVF, TenNV, maChiNhanh, cbxChucVu.SelectedValue.ToString().Trim(), NgayVaolam, luong, SDT, DiaChi, QueQuan, SoCaTruc, matKhau, maSiteDich);  if (tl)  {  MessageBox.Show("Chuyen thành công !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Information);  nhanvien.XoaNVF(MaNV, MaSite);  this.Close();  }  else  MessageBox.Show("Đã xảy ra lỗi trong quá trình Chuyen !", "Thông báo",  MessageBoxButtons.OK, MessageBoxIcon.Error);    } |

* Hàm *LayMaChiNhanhODic*

|  |
| --- |
| //Lấy mã Chi Nhánh  public string LayMaChiNhanhODic(string MaSiteDic, string tenChiNhanh)  {  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSiteDic);  //  string Query = "Select MaChiNhanh from ChiNhanh where TenChiNhanh = N'" + tenChiNhanh + "'";  DAL.stringConnection = conn;  DataSet dsTK = da.ExecuteQueryDataSet(Query, CommandType.Text);  DataTable st = dsTK.Tables[0];  //  return st.Rows[0][0].ToString();  }  catch (Exception)  {  return "CN-HN";  }  } |

* Hàm *DemSoNhanVien*

|  |
| --- |
| public int DemSoNhanVien(String MaSiteDich)  {  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(MaSiteDich);  //  string Query = "Select MAX(A) From (SELECT A = SUBSTRING(MaNV, pos, LEN(MaNV)) FROM (SELECT MaNV, pos = PATINDEX('%[0-9]%', MaNV) FROM NhanVien) d) c";  DAL.stringConnection = conn;  DataSet dsTK = da.ExecuteQueryDataSet(Query, CommandType.Text);  DataTable st = dsTK.Tables[0];  //  return int.Parse(st.Rows[0][0].ToString());  }  catch (Exception)  {  return 0;  }  } |

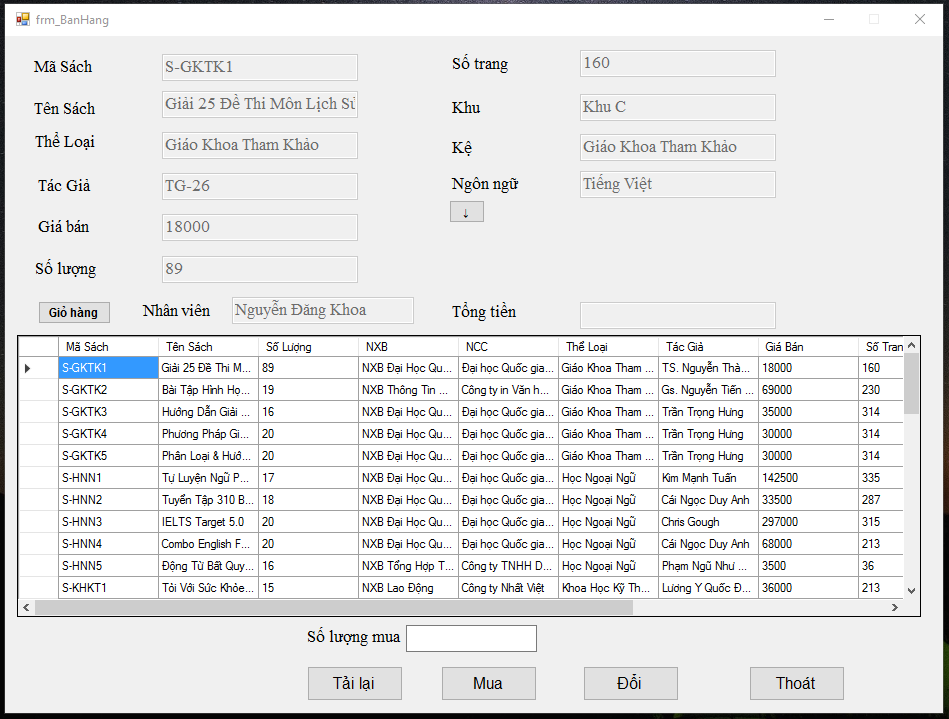
* Hàm *ChuyenNV*

|  |
| --- |
| //Chyen NhV  public bool ChuyenNV(string MaNV, string TenNV, string MaChiNhanh, string MaChucVu, string TGVaoLam, double Luong,  string SDT, string DiaChi, string QueQuan, int SoCaTruc, string MatKhau, String maSiteDich)  {  string err = "";  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(maSiteDich);  //  string TTTNV = da.LayThuocTinhStringManhMaSite("NhanVien", maSiteDich);  //  string Query = "INSERT INTO NhanVien (" + TTTNV + ") VALUES ('" + MaNV +  "', N'" + TenNV + "' , '" + MaChiNhanh + "' , '" + MaChucVu + "' , '" + TGVaoLam + "' , '" + Luong + "' , '"  + SDT + "' , '" + DiaChi + "' , '" + QueQuan + "' , " + SoCaTruc + " , '" + MatKhau + "' )";  DAL.stringConnection = conn;  return da.MyExecuteNonQuery(Query, CommandType.Text, ref err);  }  catch (Exception)  {  return false;  }  } |

* Hàm *XoaNVF*

|  |
| --- |
| public void XoaNVF(string MaNV, string maSite)  {  string err = "";  try  {  DAL.User = TenDangNhap;  DAL.Pass = MatKhau;  DAL da = new DAL();  String conn = da.getConnectStrSite(maSite);  string Query = "Delete From NhanVien where MaNV='"+MaNV+"'";  DAL.stringConnection = conn;  da.MyExecuteNonQuery(Query, CommandType.Text, ref err);  }  catch (Exception)  {    }  } |

* 1. **Bán hàng (bán sách)**



1. **Kết luận**
   1. **Kết quả đạt được**

* Xây dựng, thiết kế được cơ sở dữ liệu toàn cục, từ CSDL toàn cục đã phân tán ra các mảnh chia ra cho các chi nhánh(các site).
* Xây dựng, thiết kế được từ điển dữ liệu.
* Xậy dựng, thiết kế được phần mềm với 1 số chức năng cơ bản truy xuất dữ liệu từ các site, các mảnh dữ liệu như : Thêm, Sửa, Xóa, Chuyển nhân viên, tìm kiếm, bán hàng.
  1. **Phần chưa làm được**
* Phần mềm chưa hoàn thiện về chức năng, chưa đáp ứng được yêu cầu thầy giao.
* Giao diện chưa được bắt mắt.