

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CNTT I

BỘ MÔN HỆ ĐIỀU HÀNH



BÁO CÁO BÀI TẬP LỚN

Quản lý thư viện

NHÓM THỰC HIỆN: 10

Giảng viên hướng dẫn	: VŨ HOÀI NAM
Họ và tên sinh viên	: Nguyễn Đức Hải Nguyễn Chí Khanh Đào Duy Mạnh

Lớp	: D23 – 130
Nhóm	: 15

Hà Nội – 2023

MỤC LỤC

Lời mở đầu

I. Bối cảnh và động lực của dự án

Trong bối cảnh công nghệ thông tin phát triển, việc tìm hiểu và nắm vững các nền tảng là vô cùng quan trọng. Đề tài này tập trung vào việc tái hiện và xây dựng một ứng dụng Desktop hoàn chỉnh bằng cách sử dụng các công nghệ “kinh điển” đã đặt nền móng cho ngành phát triển phần mềm: Visual Basic 6.0 và Microsoft Access

Việc cài đặt và vận hành thành công một môi trường làm việc cũ trên máy ảo PCem (giả lập Window 98) không chỉ là 1 thử thách kỹ thuật mà còn là cơ hội để hiểu rõ cách các ứng dụng Client-Server 2-tier nguyên thủy tương tác trực tiếp với cơ sở dữ liệu

II. Mục tiêu của dự án

Mục tiêu của đề tài này là xây dựng 1 hệ thống “Quản lý thư viện” có khả năng chạy độc lập trên Window 98, áp dụng các nguyên lý lập trình sự kiện (Event – Driven Programming) của VB6 để giúp cho sinh viên hiểu hơn về các hệ điều hành cũ

Các chức năng cốt lõi bao gồm:

- Hệ thống đăng nhập (formLogin) và điều hướng (formMain).
- Một Form quản lý chính (formLibra) với đầy đủ các chức năng CRUD (Thêm, sửa, xóa, hiển thị) và khả năng mở file PDF

III. Tổng quan về hệ thống

1. Kiến trúc hệ thống

Hệ thống được thiết kế theo kiến trúc 2-Tier (2 lớp) cơ bản - là kiến trúc phổ biến trên các ứng dụng VB6/Access

- Lớp 1 (Client – giao diện): Là file QuanLyThuVien.exe (được build từ VB6). Lớp này bao gồm các Form ('formLogin', 'formMain', 'formLibra') chịu trách nhiệm hiển thị giao diện và xử lý toàn bộ logic nghiệp vụ (business logic).
- Lớp 2 (Data – Dữ liệu): Là file "ThuVien.mdb" (MB Access). Lớp này chỉ chịu trách nhiệm lưu trữ dữ liệu thô

2. Công nghệ sử dụng

- **Môi trường giả lập:** PCem (phiên bản v17) với bộ xử lý Petinum II
- **Hệ điều hành:** Microsoft Window 98
- **Ngôn ngữ lập trình:** Visual Basic 6.0
- **Hệ quản trị Cơ sở dữ liệu:** Microsoft Access (phiên bản 97/2000)
- **Thư viện kết nối:** ADO (Microsoft ActiveX Data Objects 2.x)

IV. Thành viên nhóm và phân công nhiệm vụ

Chương 1

Quá trình cấu hình PCem cho máy ảo Window98

I. Chuẩn bị tài nguyên và các công cụ

Để bắt đầu, ta cần thu thập các tài nguyên sau:

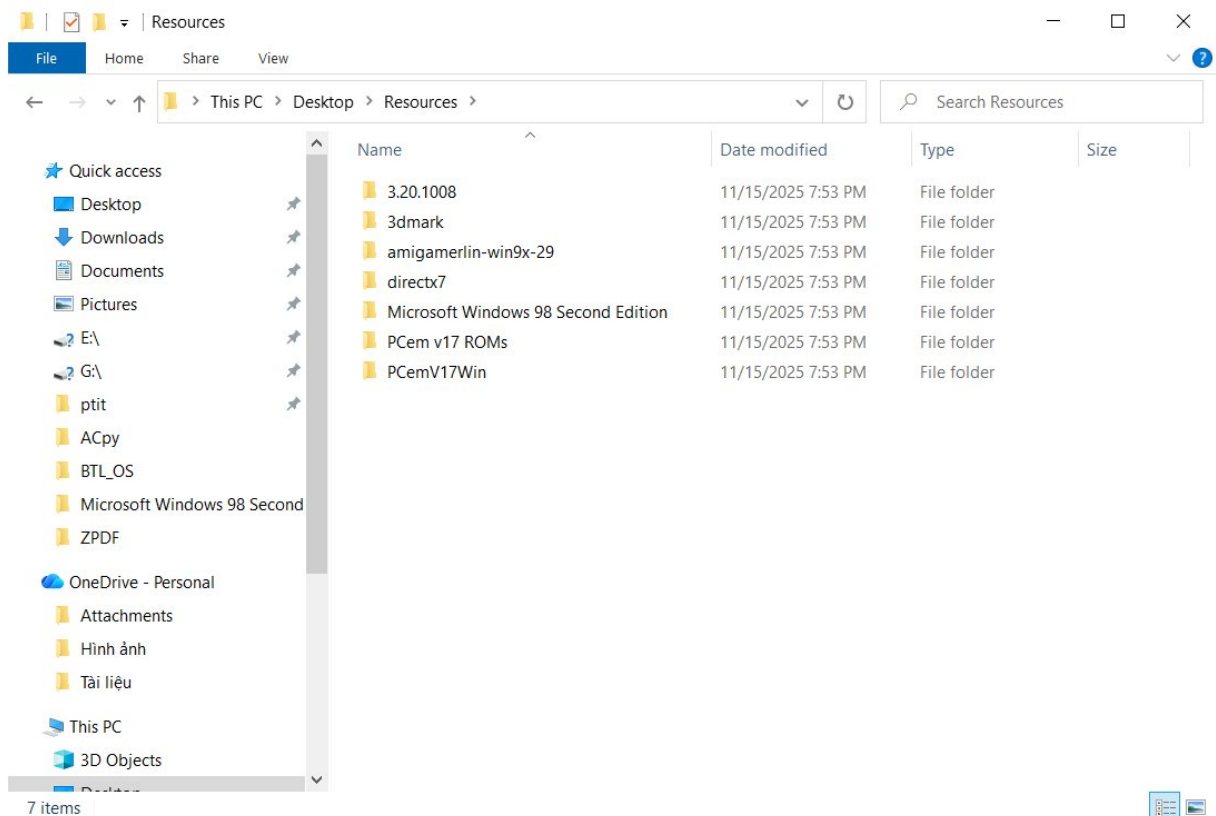
1. PCem Emulator (v17): <https://pcem-emulator.co.uk/downloads.html>
2. Roms File (bản zip): [pcem-v-17-roms directory listing](#)
3. ISO của Window 98 SE (OEM Full): [WinWorld: Windows 98 Second Edition](#)
4. Intel Chipset Driver (3.20.1008) : [Intel Chipset Drivers - PHILSCOMPUTERLAB.COM](#)
5. Amigamerlin-win9x-29.zip (card đồ họa): [3dfx graphics drivers - PHILSCOMPUTERLAB.COM](#)
6. Sound Blaster PCI 128 drivers (card âm thanh): [Sound Blaster PCI 128 - PHILSCOMPUTERLAB.COM](#) (sbpci128setupus_w9x)
7. DirectX 7: [DirectX - PHILSCOMPUTERLAB.COM](#)

Sau khi có tài nguyên, để có thể triển khai ứng dụng, ta cần các phần mềm với định dạng ISO

1. Visual Studio 6.0: [Visual Studio 6.0 Professional MSDN ISO 1999 : Microsoft : Free Download, Borrow, and Streaming : Internet Archive](#)
2. Microsoft Office 2000: [WinWorld: MSDN - Microsoft Office 2000 Developer](#)

3. Adobe Acrobat 5.0: [WinWorld: Adobe Acrobat 5.0 for Windows \(2001\) \(ISO\)](#)
4. ImgBurn: [The Official ImgBurn Website](#)

Ⓟ Tải các file tài nguyên kia về rồi giải nén sau đó cho hết vào 1 folder



Hình 1.1 Các folder cần chuẩn bị

Sau đó thì lấy tất cả các tập tin ROM rồi sao chép vào thư mục ROMs bên trong thư mục cài đặt trình giả lập PCem. Quá trình chuẩn bị tài nguyên hoàn tất.

II. Cấu hình chi tiết

1. Cấu hình máy ảo (PCem ConfigurationManager)

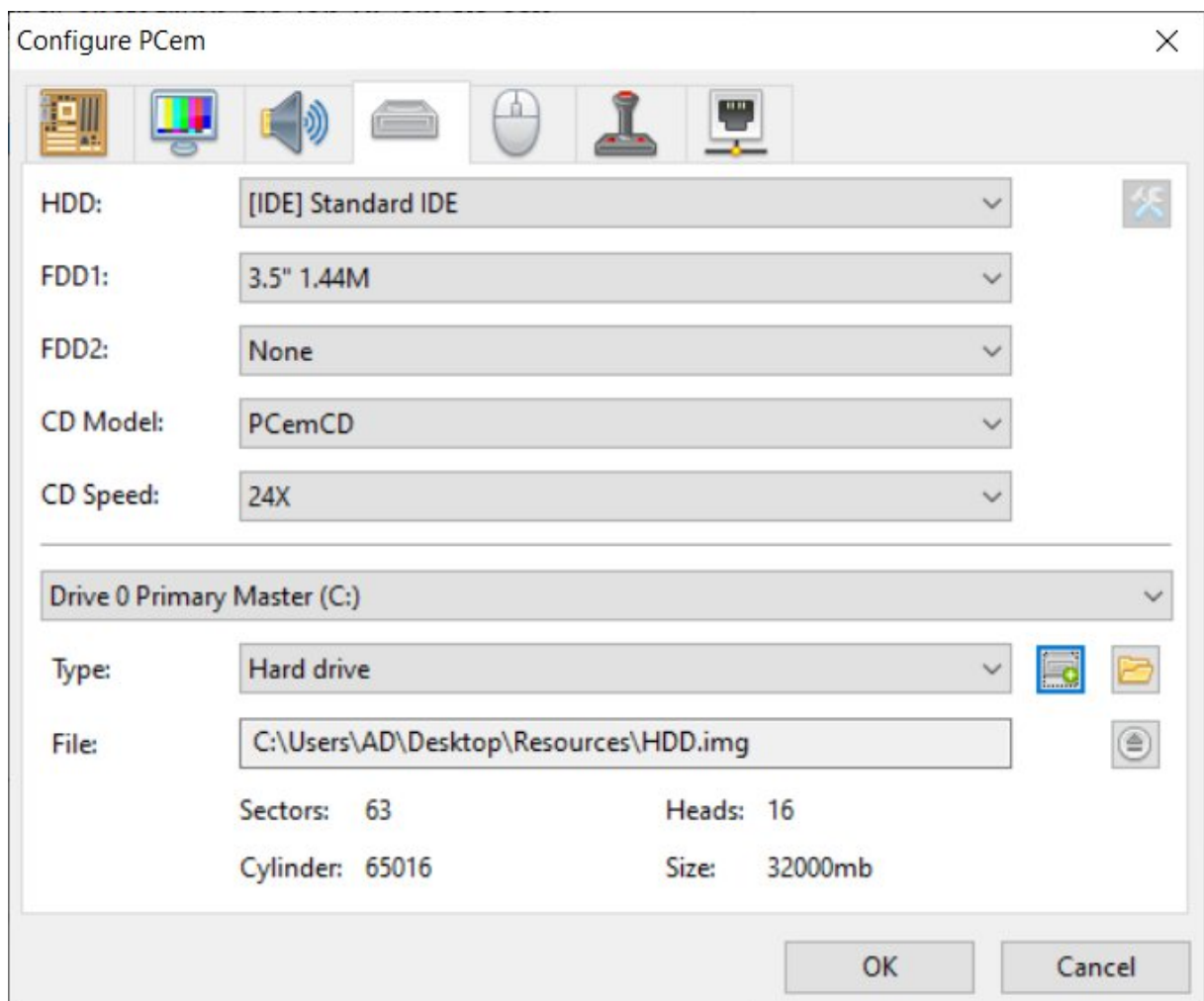
Sau khi chuẩn bị tài nguyên xong, khởi chạy trình giả lập PCem và cấu hình máy mới

- Nhấp vào trình quản lý cấu hình PCem và nhập tên mới cho cấu hình

- Chọn máy (Machine): [Slot 1] Gigabyte GA-686BX
- CPU: Intel / Pentium II/300
- Memory: 256 MB
- Card đồ họa: 3DFX Voodoo 3 3000
- Speed: Fast VLB/PCI
- Âm thanh: Sound Blaster PCI 128

2. Cấu hình lưu trữ

Thiết lập ổ đĩa mềm, ổ đĩa quang, ổ cứng như sau



Hình 1.2 Cấu hình lưu trữ

3. Cấu hình thiết bị ngoại vi

Để có thể sử dụng các thiết bị ngoại vi tương tác với máy tính như chuột thì cần phải cấu hình nó

- Chuột: Microsoft Intellimouse (PS/2)
- Mạng: Không cần thiết

Ⓢ Sau đó ấn Ok

4. Thiết lập BIOS

Việc thiết lập hệ thống vào ra cơ bản BIOS trong quá trình cấu hình máy ảo PCem là 1 bước thiết yếu nhằm điều chỉnh các thiết lập cơ bản của máy ảo để đảm bảo quá trình cài đặt hệ điều hành (Window 98 SE) và vận hành máy diễn ra suôn sẻ, hiệu quả. Bởi 2 lý do sau

- Thiết lập thứ tự khởi động (Boot Order):
à Đây là thay đổi quan trọng nhất cần thực hiện, mục đích của nó là đảm bảo máy ảo khởi động cho đến khi CD – ROM được ưu tiên đầu tiên, sau đó mới đến ổ cứng (hard drive)
- Tối ưu hóa và Vô hiệu hóa các thiết bị ngoại vi không cần thiết:
à Tải cấu hình Mặc định để thiết lập các thông số cơ bản cho hiệu suất tốt
à Vô hiệu hóa các tài nguyên không cần thiết chẳng hạn như cổng serial và cổng parallel giúp cho đơn giản hóa cấu hình và tránh xung đột tài nguyên không cần thiết.

Tóm lại thì việc thiết lập BIOS giống như việc cài đặt các thiết lập cơ bản của bo mạch chủ trên 1 máy tính. Nó đảm bảo máy ảo biết phải khởi động từ đâu và chỉ bật các thành phần cần thiết cho hoạt động đã định

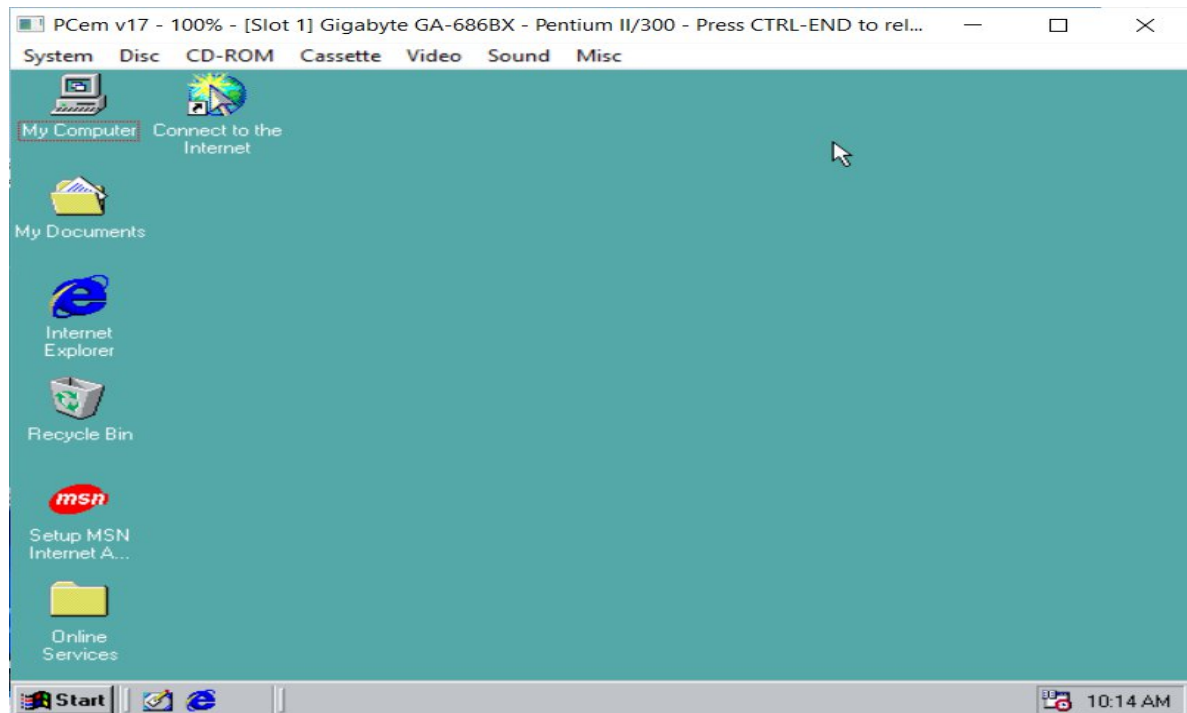
Quá trình thực hiện diễn ra như sau

- Sau khi cấu hình phần cứng thì khởi chạy trình giả lập. Sau khi màn hình hiện lên thì nhanh tay ấn “**delete**” để truy cập BIOS
- Sau khi vào được BIOS, ta bắt đầu set up
- Load Performance Defaults à Yes

- Vào BIOS Feature Setup: Boot Sequence à CDROM, C, A
- Vào Intergrated PeriPherals: Tắt các cổng Serial và Parallel
- Trước khi save & exit, ta sẽ load image (ấn vào CDRom ở thanh menu) để đẩy Window 98 SE vào. Sau đó thì màn hình hiển thị các lựa chọn để cài đặt Window 98 SE thì ta chấp nhận hết rồi save & exit để bắt đầu quá trình cài đặt từ CD

5. Cài đặt Window 98 SE

- Khởi động CD-ROM và bắt đầu quá trình cài đặt Window 98 SE
- Chấp nhận tùy chọn phân vùng ổ đĩa cứng, bật hỗ trợ ổ đĩa lớn (large disk support)
- Cho phép hệ thống khởi động lại và tiếp tục khởi động từ CD-ROM để thực hiện định dạng ổ cứng
- Trong quá trình cài đặt, tùy chỉnh các thành phần sẽ được cài đặt, chỉ giữ lại các mục đa phương tiện cần thiết như CD Player và Volume Control. Bước này nhằm tạo ra 1 hệ điều hành tinh gọn, giảm thiểu các tiến trình nền và tài nguyên tiêu thụ để tối đa hóa hiệu suất.
- Hoàn thành giai đoạn sao chép tệp ban đầu và khởi động lại
- Truy cập lại BIOS (nhấn “delete”) và thay đổi thứ tự khởi động trở lại Hard Dive làm ưu tiên hàng đầu
- Sau đó cung cấp các thông tin người dùng và nhập mã bản quyền (license key) khi được yêu cầu
à Window Product Key: **TYG73-R4XDG-9XBJ4-YJGT9-738BY**
- Hoàn tất các bước cuối cùng cho đến khi màn hình desktop của Windows 98 SE xuất hiện



6. Cài đặt trình điều khiển

Thứ tự cài đặt trong quá trình này rất quan trọng để đảm bảo sự ổn định của hệ thống

- Chuẩn bị dữ liệu: Tắt hoàn toàn máy ảo. Trên máy chủ, gắn (chuột phải + mount) tệp VHD đã tạo. Sao chép các tệp trình điều khiển vào 1 thư mục trên ổ đĩa ảo. Ngắt kết nối (chuột phải + eject) VHD một cách an toàn
- Trình điều khiển Chipset: Khởi động lại máy ảo. Cài đặt trình điều khiển chipset Intel trước tiên và khởi động lại máy khi được yêu cầu.
- Kích hoạt DMA: Truy cập Device Manager, tìm ổ cứng IDE trong Disk drives, vào phần Settings và chọn hộp kiểm DMA. Điều này cải thiện đáng kể hiệu suất truy xuất đĩa. Khởi động lại máy.
- Trình điều khiển đồ họa: Cài đặt trình điều khiển Voodoo 3 và khởi động lại máy.
- Trình điều khiển âm thanh: Cài đặt trình điều khiển Sound Blaster PCI 128.

- Cài đặt DirectX: Cài đặt DirectX 7. Thực hiện lần khởi động lại cuối cùng của giai đoạn này để hoàn tất việc cài đặt trình điều khiển.

Ⓟ Công việc trên đã vạch ra một lộ trình chi tiết và có hệ thống để triển khai thành công một môi trường Windows 98 giả lập, được tối ưu hóa cao trên nền tảng PCem. Bằng cách tuân thủ nghiêm ngặt các giai đoạn từ chuẩn bị tài nguyên, cấu hình phần cứng ảo, cài đặt hệ điều hành và trình điều khiển theo đúng thứ tự, cho đến tinh chỉnh và kiểm tra nghiêm ngặt, dự án đã đáp ứng tất cả các mục tiêu ban đầu.

III. Triển khai hệ thống

Bây giờ, sau khi đã có tài nguyên. Ta sẽ cài đặt các công cụ cần thiết trên máy ảo để có thể triển khai hệ thống.

1. Thiết kế Cơ sở dữ liệu

Hệ thống sử dụng kiến trúc 2-Tier (2 Lớp) cổ điển, trong đó Lớp Dữ liệu (Data Layer) được triển khai bằng một file cơ sở dữ liệu duy nhất.

1.1 Lựa chọn Công nghệ

- Hệ quản trị CSDL: Microsoft Access (phiên bản 2000)
- File Dữ Liệu: ThuVien.mdb
- Lý do lựa chọn: MS Access được chọn vì tính tương thích "nguyên bản" (native) với môi trường Windows 98 và Visual Basic 6.0. Nó cho phép kết nối trực tiếp, nhanh chóng thông qua thư viện ADO (ActiveX Data Objects) mà không cần cài đặt thêm server CSDL phức tạp.

1.2 Kiến trúc bảng

Để đáp ứng các yêu cầu của project, một bảng dữ liệu trung tâm đã được thiết kế: BangSach

Bảng này lưu trữ toàn bộ thông tin "metadata" (siêu dữ liệu) của các quyển sách trong thư viện, cũng như đường dẫn tới file tài liệu đính kèm.

1.2 Chi tiết các cột trong bảng BangSach

Dưới đây là mô tả chi tiết từng cột trong bảng BangSach, bao gồm các quyết định thiết kế quan trọng để đảm bảo tính ổn định khi tương tác với VB6:

+ **id** : Kiểu dữ liệu AutoNumber à Khóa chính của bảng. Dùng AutoNumber để Access tự động tạo ra một mã số định danh duy nhất (unique) cho mỗi cuốn sách mới. Điều này đảm bảo tính toàn vẹn dữ liệu và giúp phân biệt các bản ghi, ngay cả khi đã xóa các bản ghi cũ (ID không "tái sử dụng").

+ **nameBook**: Kiểu dữ liệu Text à Lưu tên của quyển sách. Được thiết lập Required = Yes (Bắt buộc) và Allow Zero Length = Yes để đảm bảo tránh lỗi khi dùng thao tác bỏ qua nếu không có dữ liệu đầu.

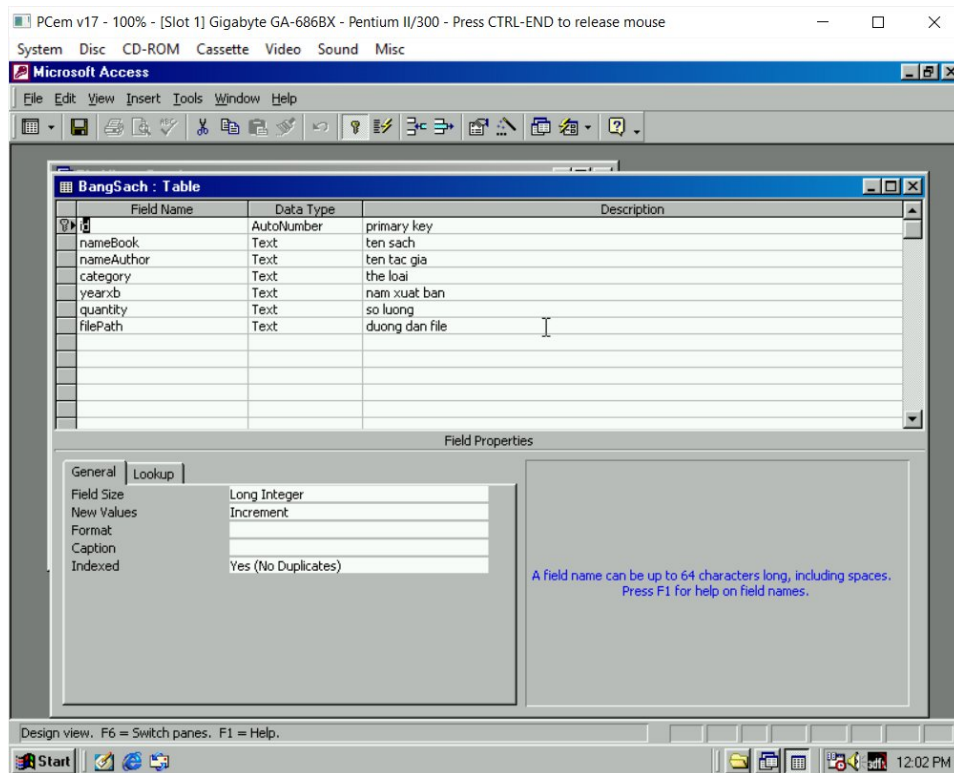
+ **nameAuthor**: Kiểu dữ liệu Text à Lưu tên tác giả. Được thiết lập Allow Zero Length = Yes (Cho phép chuỗi rỗng) để linh hoạt, cho phép người dùng bỏ trống nếu không biết.

+ **category**: Kiểu dữ liệu Text à Lưu thể loại sách (ví dụ: "Truyện tranh", "Lập trình", "Hệ điều hành"...).

+ **yearxb**: Kiểu dữ liệu Text à Lưu năm xuất bản

+ **quantity**: Kiểu dữ liệu Text à Lưu số lượng sách. Cột này được thiết kế kiểu dữ liệu Text thay vì Number là để tránh lỗi Data type Mismatch (khi gán Text vào Number) và lỗi Validation Rule (khi gán "" tức số 0 vào 1 cột bị cấm số 0)

+ **filePath**: Kiểu dữ liệu Text à Lưu đường dẫn đầy đủ tới file đính kèm (Ví dụ: C:\Ebooks\LapTrinhCsharp.pdf)



2. Thiết kế giao diện

Lớp trình diễn (Presentation Layer) của ứng dụng được xây dựng bằng **Windows Forms** của **Visual Basic 6.0**. Kiến trúc này tuân thủ mô hình **Lập trình Hướng sự kiện (Event-Driven Programming)**, nơi các hành động của người dùng (như nhấn nút, click chuột) sẽ kích hoạt các đoạn code (Sub) tương ứng.

Hệ thống bao gồm 3 Form chính, mỗi Form đảm nhận một vai trò riêng biệt.

2.1 Form đăng nhập (formLogin)

Đây là "cửa ngõ" của ứng dụng, chịu trách nhiệm xác thực người dùng trước khi cho phép truy cập vào các chức năng chính.

- **Mục đích:** Ngăn chặn truy cập trái phép và xác định người dùng.

- **Các control chính:**

- + 2 label: Hiển thị tiêu đề “Tên đăng nhập” và “Mật khẩu”

- + 2 TextBox: txtUser (cho tên đăng nhập) và txtPass (cho mật khẩu)

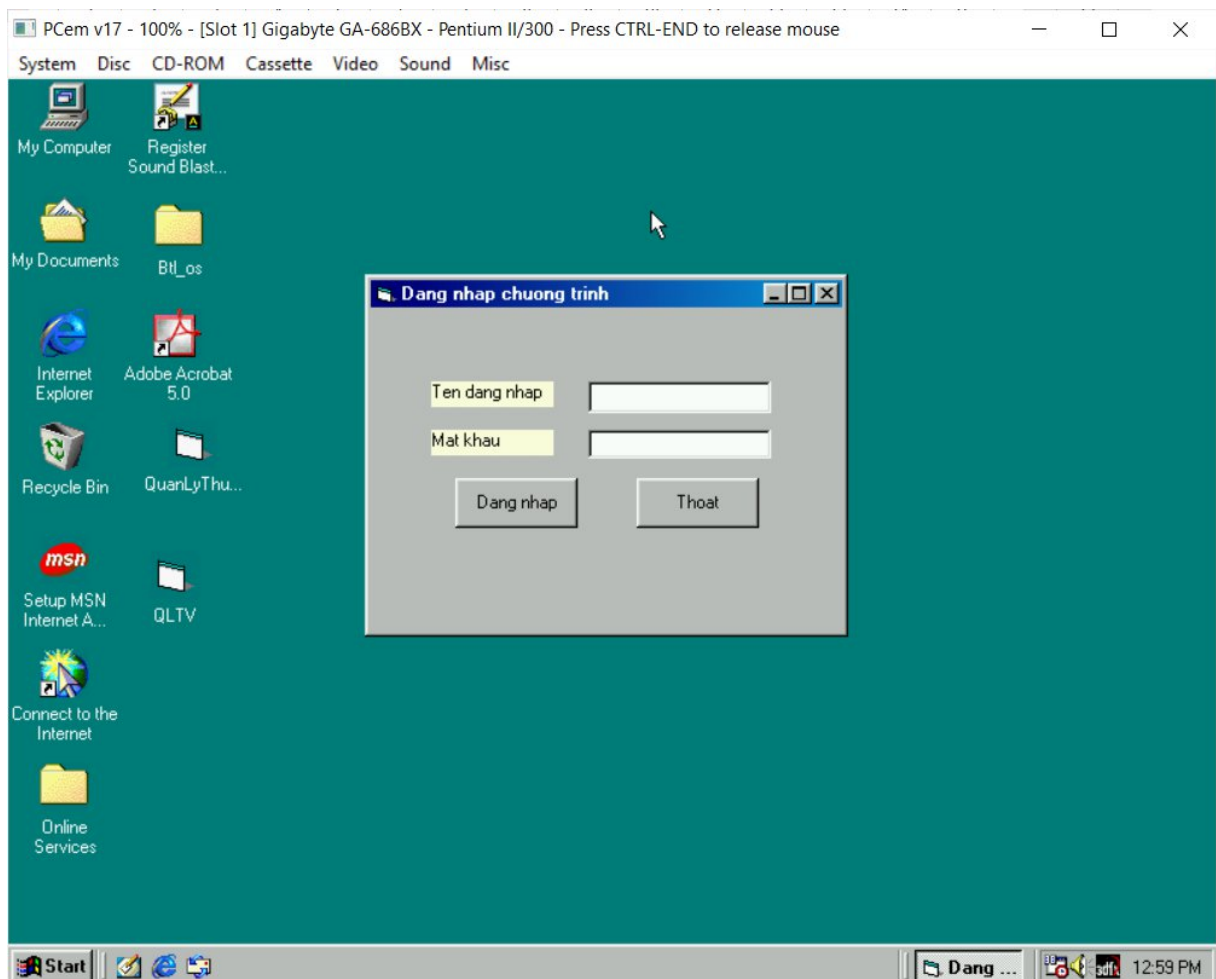
- + 2 Command Button: cmdLogin (nút “Đồng ý”) và cmdExit (nút “Thoát”)

- **Logic đặc biệt:**

- + Thuộc tính PasswordChar của txtPass được thiết lập là * để che giấu mật khẩu khi gõ

- + Logic xác thực được “hard core” (gán cứng) trong cmdLogin_Click() để kiểm tra admin/123

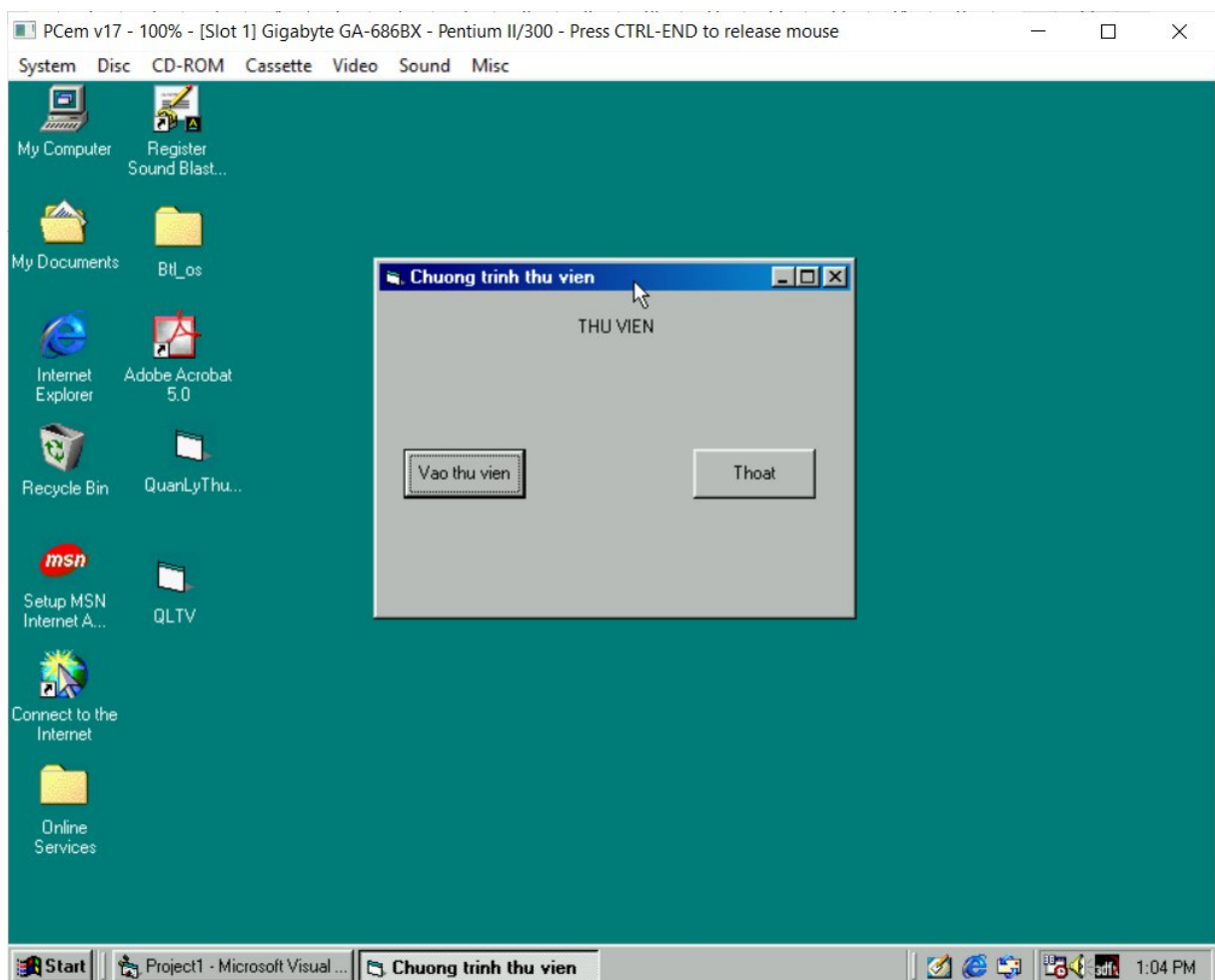
- + Khi xác thực thành công, Form này tự ẩn đi (Me.Hide) và gọi Form chính formMain.show



2.2 Form chính

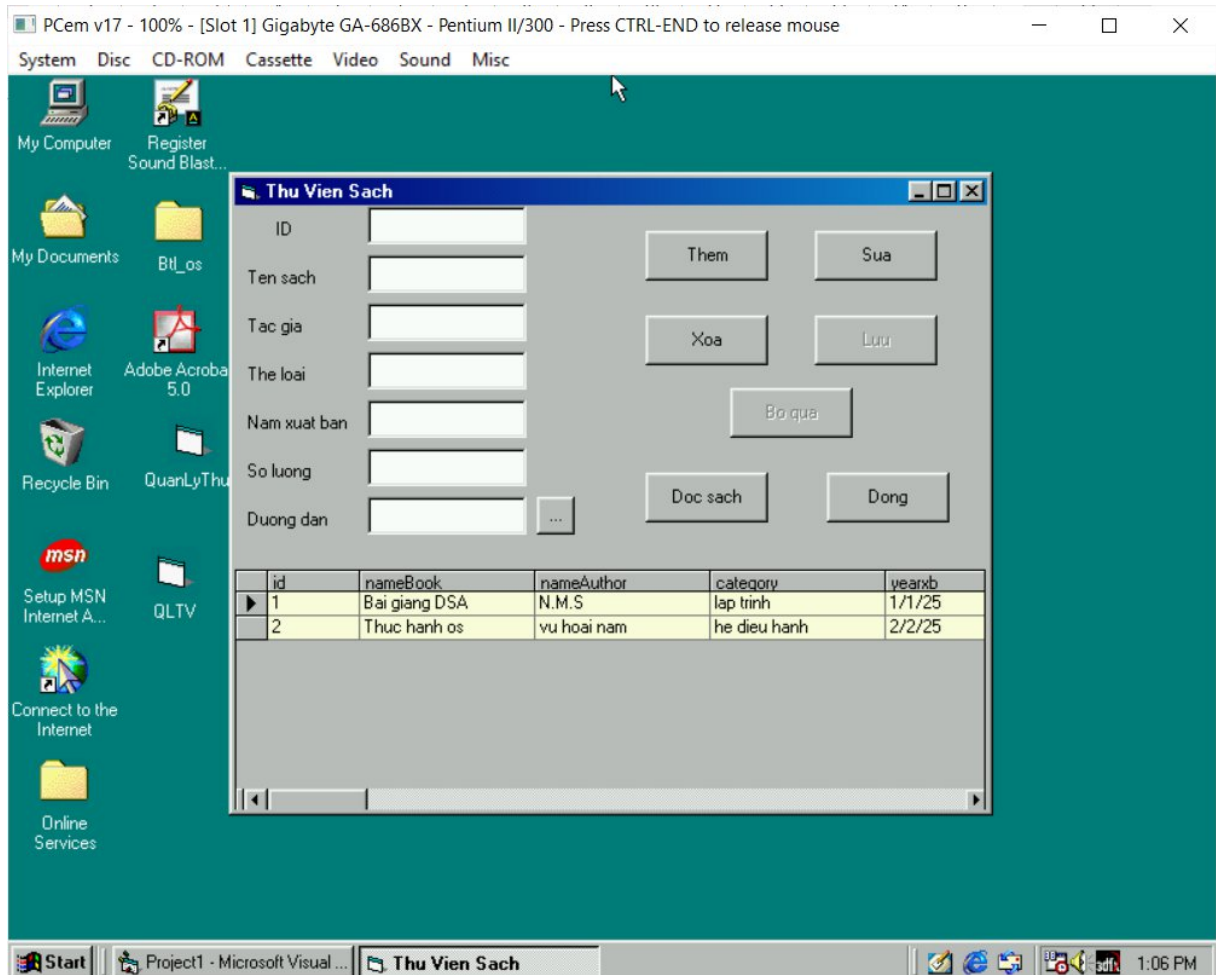
Đây là Form điều hướng (menu) chính, xuất hiện sau khi người dùng đăng nhập thành công.

- **Mục đích:** Đóng vai trò là bảng điều khiển trung tâm (main dashboard), cho phép người dùng chọn các chức năng.
- **Các Control chính:**
 - + cmdVaoThuVien: Nút điều hướng chính, dùng để mở Form Quản lý thư viện
 - + cmdRoiThuVien: Để thoát khỏi formMain
- **Logic:** Khi ấn cmdVaoThuVien, formMain sẽ tự ẩn đi (Me.Hide) và gọi formLibra (formLibra.Show)



2.3 Form quản lý thư viện (formLibra)

Đây là Form trung tâm, phức tạp nhất, chứa toàn bộ logic nghiệp vụ (business logic) của project. Giao diện được thiết kế theo 3 khu vực chức năng rõ rệt.



2.3.1 Khu vực nhập liệu bên trái

+ Bao gồm 7 Label và 7 TextBox (txtID, txtTenSach, txtTacGia, txtTheLoai, txtNamXB, txtSoLuong, txtDuongDanFile).

+ Các ô này dùng để hiển thị dữ liệu khi bấm vào bảng, và cũng là nơi người dùng nhập liệu khi "Thêm" hoặc "Sửa".

+ Ô txtID được thiết lập Enabled = False để khóa, không cho người dùng sửa Khóa chính (ID).

2.3.2 Khu vực Chức năng (Bên phải)

- + Bao gồm 8 CommandButton (cmdThem, cmdSua, cmdXoa, cmdLuu, cmdBoQua, cmdDocSach, cmdDong, và cmdTimFile).
- + Các nút này điều khiển toàn bộ logic CRUD và các tính năng nâng cao.
- + Logic SetButtons (Boolean) được lập trình để Bật/Tắt (Enable/Disable) các nút "Lưu", "Bỏ qua" tùy theo trạng thái (đang "chỉnh sửa" hay "xem").

2.3.3 Khu vực hiển thị (bảng bên dưới)

- + Sử dụng control "trùm cuối" **DataGrid** (tên là dgSach).
- + Control này được "ràng buộc dữ liệu" (data-bound) trực tiếp với "thùng hàng" ADODB.Recordset (rs) của chúng ta.
- + Nó chịu trách nhiệm hiển thị toàn bộ dữ liệu từ CSDL.
- + Code dgSach.Columns(0).Visible = False được sử dụng để **ẩn cột id** (đã sắp xếp) khỏi tầm nhìn của người dùng, tăng tính thẩm mỹ và chỉ hiển thị các thông tin cần thiết.

3. Kết nối VB6 với Access

3.1 Khai báo thư viện và linh kiện

Trước khi đi vào chi tiết code nghiệp vụ, bước đầu tiên là phải "nhập khẩu" (import) các "vũ khí" (thư viện và linh kiện) cần thiết vào môi trường VB6.

- **Project -> References...:** Thư viện "then chốt" (key) đã được thêm vào là **Microsoft ActiveX Data Objects 2.x Library** (ADO). Thư viện này cung cấp các đối tượng ADODB.Connection ("xe tải" cnn) và ADODB.Recordset ("thùng hàng" rs) để kết nối và truy vấn CSDL Access. Nếu không có bước này, VB6 sẽ báo lỗi "User-defined type not defined" (kiểu chưa định nghĩa).

[Chèn ảnh chụp màn hình lúc cậu chọn "Microsoft ActiveX Data Objects" trong References]

- **Project -> Components...:** Để xây dựng giao diện, project đã nhập khẩu 2 linh kiện:
 1. **Microsoft DataGrid Control 6.0:** Cung cấp control DataGrid (tên dgSach), dùng để hiển thị dữ liệu dạng bảng.
 2. **Microsoft Common Dialog Control 6.0:** Cung cấp control "vô hình" CommonDialog1, dùng để gọi hộp thoại "Mở File" (ShowOpen).

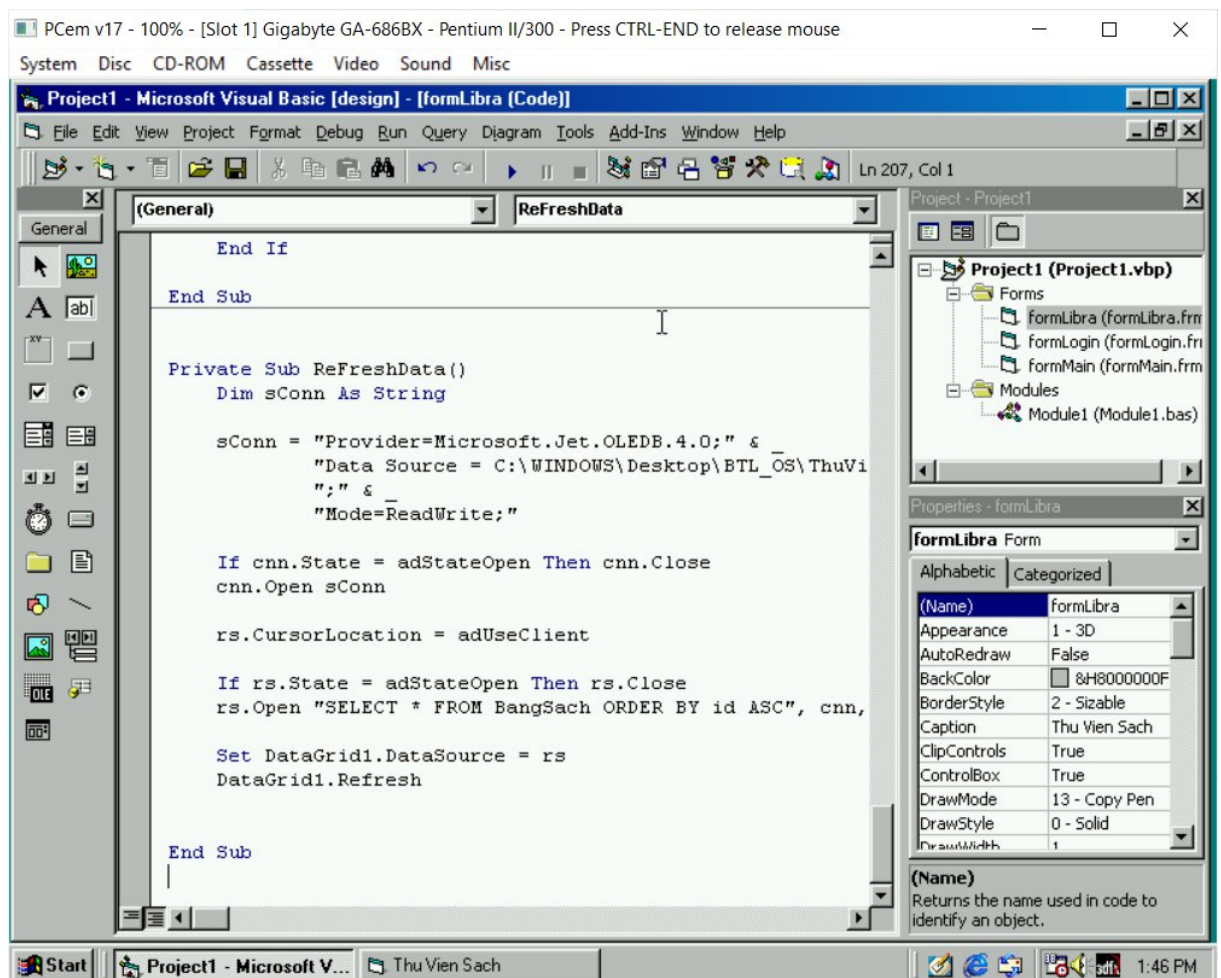
3.2 Module kết nối CSDL (Hàm RefreshData)

Hàm RefreshData là "trái tim" của Form FormLibra. Toàn bộ việc tải dữ liệu từ Access lên DataGrid và "làm mới" (refresh) sau khi Thêm/Sửa/Xóa đều do hàm này đảm nhiệm.

Trước hết cần khai báo biến toàn cục

```
Dim cnn As New ADODB.Connection  
Dim rs As New ADODB.Recordset
```

Sau đó đi vào hàm ReFreshData



3.3. Module Hiển thị Chi tiết (Đọc - Read)

Sau khi dữ liệu đã được tải (Load) thành công lên DataGrid (bằng hàm RefreshData), bước tiếp theo là cho phép người dùng tương tác với dữ liệu đó.

- **Mục tiêu:** Khi người dùng bấm (click) vào bất kỳ dòng nào trên DataGrid (DataGrid1), toàn bộ thông tin của dòng đó phải được "bơm" (load) ngược lên các ô TextBox (txtTenSach, txtTacGia...) tương ứng ở trên.
- **Sự kiện (Event):** Chúng ta sử dụng sự kiện dgSach_Click().

3.3.1 Tái cấu trúc Code với NapDuLieuLenTextBox

Ban đầu, code "bơm" dữ liệu được viết thẳng vào DataGrid1_Click(). Tuy nhiên, trong quá trình gỡ lỗi (debug) nút "Bỏ qua" (cmdBoQua_Click), chúng ta phát hiện ra rằng chúng ta cũng cần "tải lại" dữ liệu của dòng đang chọn sau khi Hủy.

Để tránh lặp code (Don't Repeat Yourself - DRY), chúng ta đã "nâng cấp" (refactor) logic "bơm" dữ liệu thành một hàm phụ (helper function) riêng biệt: Private Sub NapDuLieuLenTextBox().

```
Private Sub NapDuLieuLenTextBox()  
    If rs.RecordCount > 0 Then  
        rs.Bookmark = DataGrid1.Bookmark  
        txtID.Text = rs.Fields("id") & ""  
        txtTenSach.Text = rs.Fields("nameBook") & ""  
        txtTacGia.Text = rs.Fields("nameAuthor") & ""  
        txtTheLoai.Text = rs.Fields("category") & ""  
        txtNamXB.Text = rs.Fields("yearxb") & ""  
        txtSoLuong.Text = rs.Fields("quantity") & ""  
        txtDuongDanFile.Text = rs.Fields("filePath") & ""  
        SetButtons (False)  
    End If  
  
End Sub
```

Code Sự kiện DataGrid1_Click:

Sau khi "tái cấu trúc", sự kiện Click của DataGrid trở nên cực kỳ gọn gàng:

```
Private Sub DataGrid1_Click()
    NapDuLieuLenTextBox
End Sub
```

4. Code các chức năng

4.1 Module Quản lý Sách (Thêm, Sửa, Xóa - CRUD)

Đây là module nghiệp vụ (business logic) chính, nơi project gặp nhiều thử thách nhất. Quá trình gỡ lỗi (debug) đã phát hiện và sửa chữa hàng loạt các "lỗi kinh điển" của VB6/ADO.

4.1.1 Chức năng "Thêm" (cmdThem_Click) và "Sửa" (cmdSua_Click)

Hai nút này đóng vai trò "khởi động" trạng thái.

- **cmdThem_Click:** Gọi hàm rs.AddNew để tạo một "dòng ảo" (ghost row) trong Recordset (rs), sẵn sàng nhận dữ liệu mới.

```
Private Sub cmdThem_Click()
    SetButtons (True)
    ClearTextBoxes
    rs.AddNew
    txtTenSach.SetFocus
End Sub
```

- **cmdSua_Click:** Không cần gọi hàm gì, nó chỉ "mở khóa" các nút bấm.
- Cả hai nút này đều gọi hàm phụ SetButtons(True) để "mở" (Enable) 2 nút "Luu" và "Bỏ qua", đồng thời "khóa" (Disable) các nút "Thêm", "Sửa", "Xóa" để tránh người dùng thao tác nhầm.

```
Private Sub cmdSua_Click()
    If rs.RecordCount <= 0 Then
        MsgBox "Chua co du lieu de sua!", vbExclamation
        Exit Sub
    End If
    SetButtons (True)
End Sub
```

4.1.2 Chức năng "Xóa" (cmdXoa_Click)

Chức năng này đơn giản hơn. Nó chỉ cần lấy id của dòng đang chọn và gọi lệnh rs.Delete, sau đó gọi RefreshData để tải lại bảng.

```

Private Sub cmdXoa_Click()
    If rs.RecordCount <= 0 Then
        MsgBox "Chua co du lieu de xoa!", vbExclamation
        Exit Sub
    End If

    If MsgBox("Ban co chac chan muon xoa sach: " & txtTenSach.Text & "?", _
        vbQuestion + vbYesNo, "Xac nhan xoa") = vbYes Then
        rs.Delete
        DataGrid1.Refresh
        MsgBox "Da xoa thanh cong", vbInformation
    End If
End Sub

```

4.1.3. Chức năng "Luu" (cmdLuu_Click)

Đây là chức năng phức tạp nhất. Nó phải xử lý 2 logic (Thêm mới và Sửa), "Validate" (kiểm tra) dữ liệu đầu vào, và "Refresh" (tải lại) DataGrid để hiển thị ID mới.

```

Private Sub cmdLuu_Click()
    On Error GoTo LoiLuu_Debug
    If Trim(txtTenSach.Text) = "" Then
        MsgBox "loi ten sach ko dc de trong", vbCritical, "loi du lieu"
        txtTenSach.SetFocus
        Exit Sub
    End If
    rs.Fields("nameBook") = txtTenSach.Text
    rs.Fields("nameAuthor") = txtTacGia.Text
    rs.Fields("category") = txtTheLoai.Text
    rs.Fields("yearxb") = txtNamXB.Text
    rs.Fields("quantity") = txtSoLuong.Text
    rs.Fields("filePath") = txtDuongDanFile.Text
    rs.Update
    rs.Requery
    If rs.RecordCount > 0 Then
        rs.MoveLast
    End If
    Set DataGrid1.DataSource = Nothing
    Set DataGrid1.DataSource = rs
    DataGrid1.Refresh
    If rs.RecordCount > 0 Then DataGrid1.Bookmark = rs.Bookmark
    SetButtons (False)
    MsgBox "Luu thanh cong", vbInformation
    Exit Sub

```

4.1.4 Chức năng "Bỏ qua" (cmdBoQua_Click)

Được dùng để bỏ qua khi người dùng không muốn tiếp tục thêm sách vào database

4.2 Module Tính năng Nâng cao (UX)

Ngoài các chức năng CRUD cơ bản, project còn triển khai 2 tính năng nâng cao để cải thiện trải nghiệm người dùng và tính "chuyên nghiệp" của ứng dụng.

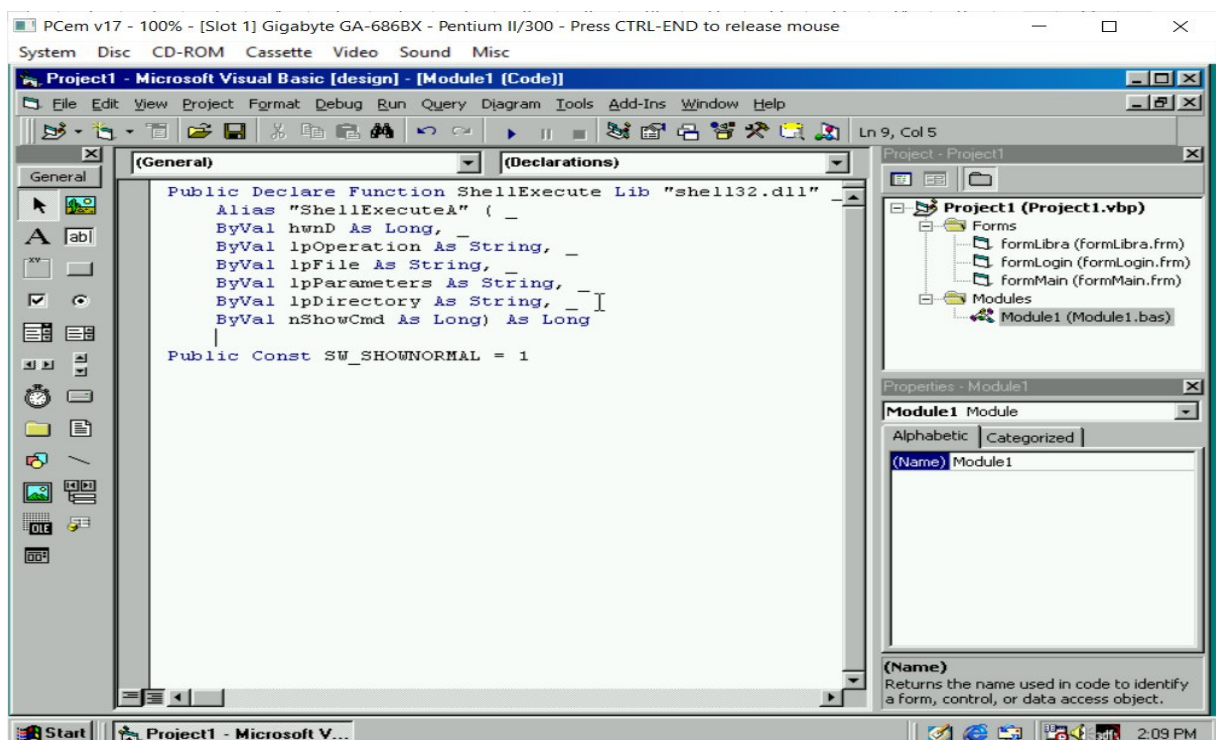
4.2.1 Chức năng "Đọc Sách" (Gọi API của Windows)

Thay vì chỉ quản lý tên sách, ứng dụng cho phép người dùng mở file tài liệu (PDF) đính kèm bằng trình đọc mặc định của Windows (ví dụ: Adobe Acrobat Reader).

- **Thách thức:** Visual Basic 6.0 không thể tự "đọc" file PDF. Nó phải "nhờ" hệ điều hành Windows làm việc này.
- **Giải pháp:** Sử dụng hàm **ShellExecute** của Windows API. Đây là một hàm "thần thánh" nằm trong thư viện shell32.dll của Windows, cho phép một ứng dụng "ra lệnh" cho Windows mở bất kỳ file hoặc đường dẫn nào.

Triển khai (Gồm 2 bước):

Bước 1: Khai báo (Declare) hàm API trong Module Chúng ta không thể gọi ShellExecute trực tiếp. Trước tiên, chúng ta phải "giới thiệu" hàm này cho VB6 biết bằng cách "Khai báo" (Declare) nó trong một Module (Module1.bas) riêng biệt.



Bước 2: Viết code cho Nút "Đọc Sách" (cmdDocSach_Click) Sau khi đã "khai báo", nút "Đọc Sách" trong FormLibra giờ đã có thể "triệu hồi" hàm ShellExecute.

- **Logic:** Code sẽ lấy đường dẫn file từ ô txtDuongDanFile, kiểm tra xem đường dẫn có rỗng không, và (quan trọng nhất) kiểm tra xem file có *thực sự tồn tại* trên ổ cứng không (dùng hàm Dir()). Nếu mọi thứ hợp lệ, nó sẽ gọi ShellExecute với lệnh "open".

```
Private Sub cmdDocSach_Click()  
    Dim sDuongDan As String  
    sDuongDan = txtDuongDanFile.Text  
    MsgBox "Duong dan: [" & sDuongDan & "]", vbInformation, "kiem tra"  
    If sDuongDan = "" Then  
        MsgBox "Sach nay khong co file dinh kem", vbExclamation, "thong bao"  
  
        Exit Sub  
    End If  
  
    If Dir(sDuongDan) = "" Then  
        MsgBox "Loi khong tim thay file tai duong dan" & vbCrLf & sDuongDan, vbCritical  
        Exit Sub  
    End If  
    |  
    ShellExecute Me.hwnd, "open", sDuongDan, "", "", SW_SHOWNORMAL  
  
End Sub
```

4.2.2 Chức năng "Tìm File" (Dùng CommonDialog)

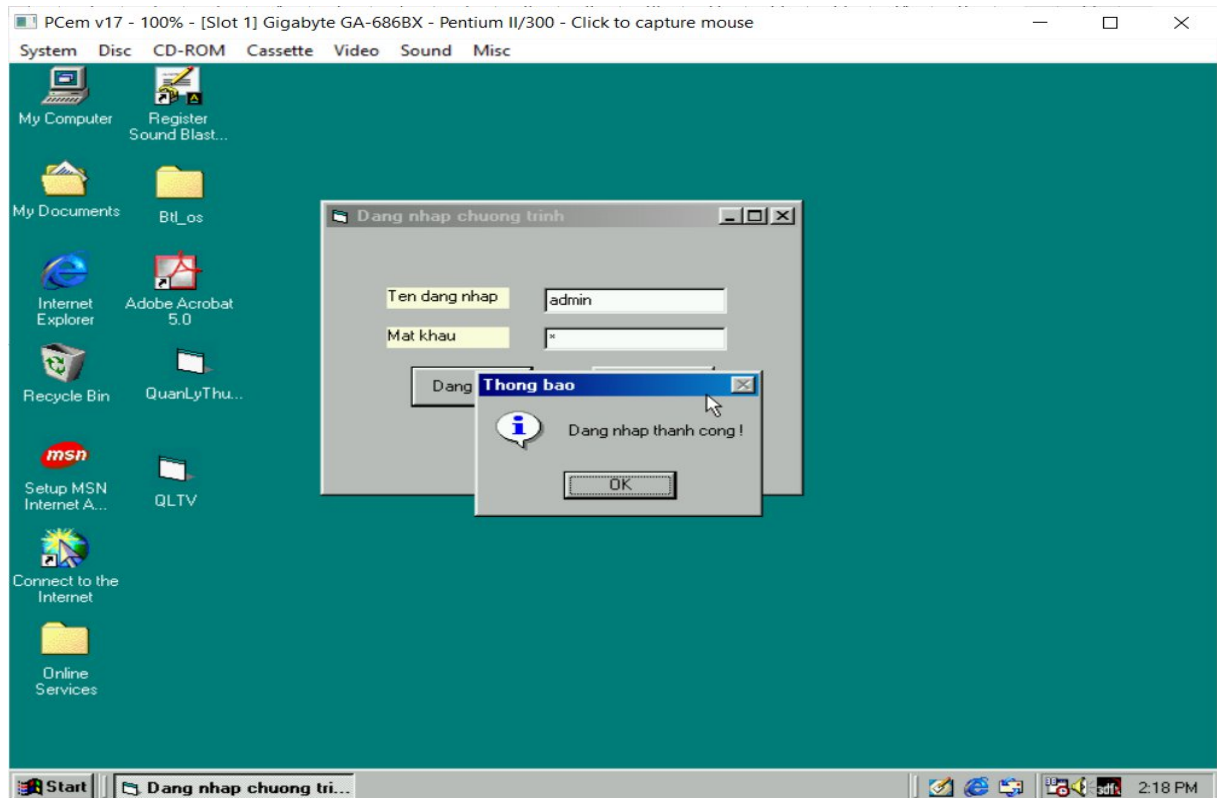
Để tránh lỗi gõ sai chính tả khi nhập đường dẫn file PDF, ứng dụng đã tích hợp hộp thoại "Mở File" chuẩn của Windows.

- **Kỹ thuật:** Sử dụng control **CommonDialog** (đã được nhập khẩu ở Project -> Components...).
- **Triển khai:** Một nút "..." (cmdTimFile) được thêm vào giao diện. Khi nhấn, nó sẽ gọi control CommonDialog1 (đã được kéo "vô hình" vào Form) và cấu hình "Bộ lọc" (Filter) để chỉ cho phép chọn file .pdf

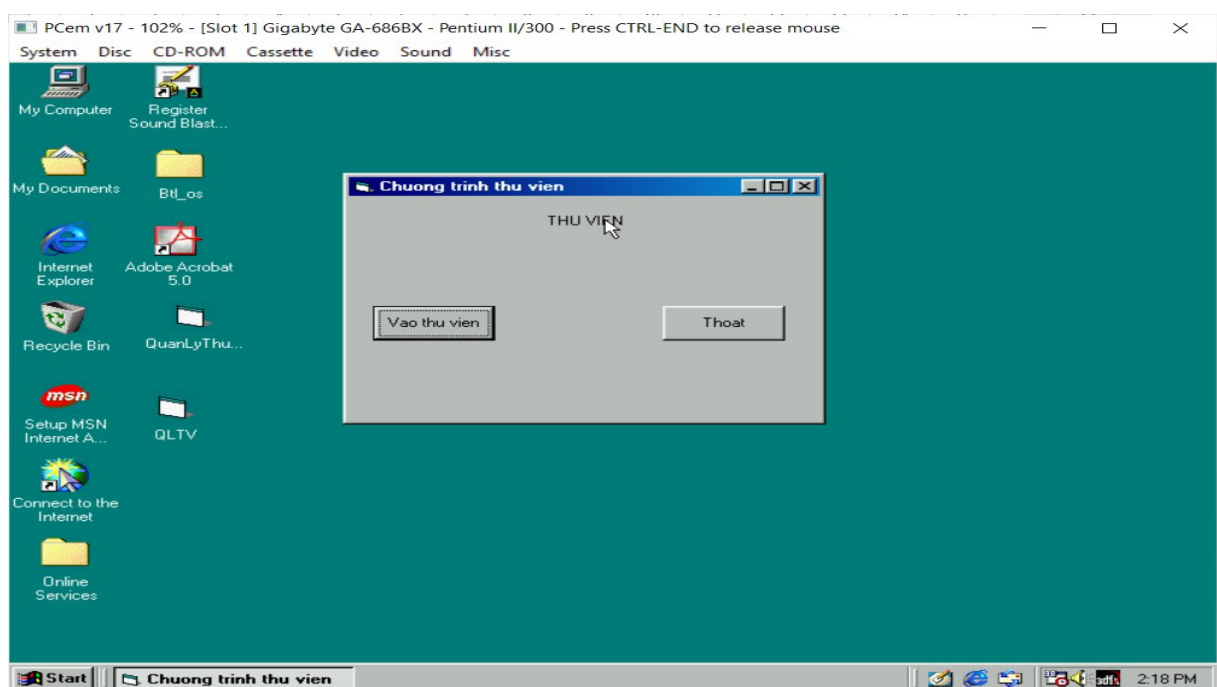
```
Private Sub cmdTimFile_Click()  
    CommonDialog1.Filter = "PDF File (*.pdf)|*.pdf"  
    CommonDialog1.InitDir = "C:\SACH_PDF"  
    CommonDialog1.ShowOpen  
    If CommonDialog1.FileName <> "" Then  
        txtDuongDanFile.Text = CommonDialog1.FileName  
    End If  
  
End Sub
```

5. DEMO

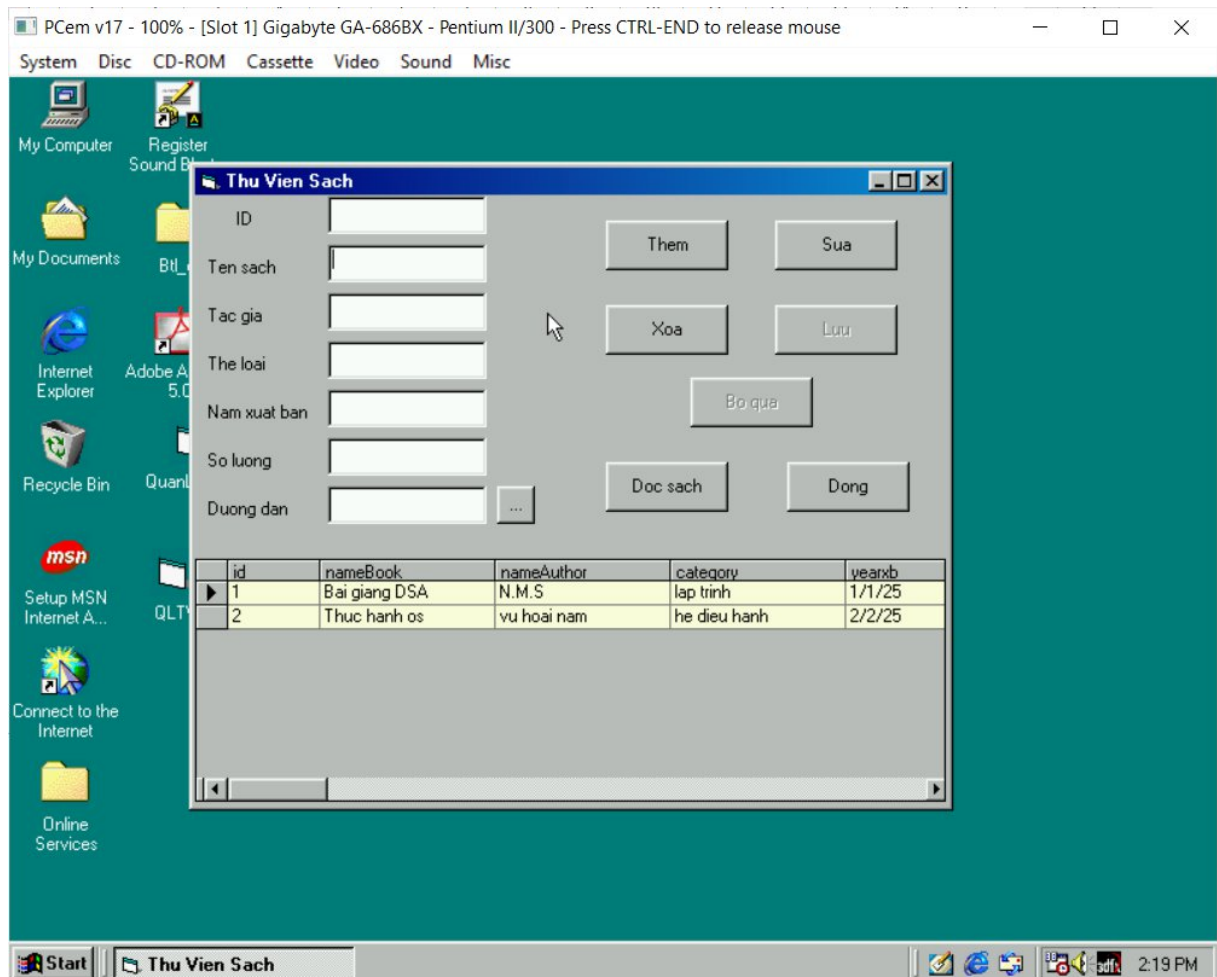
- Đăng nhập (formLogin)



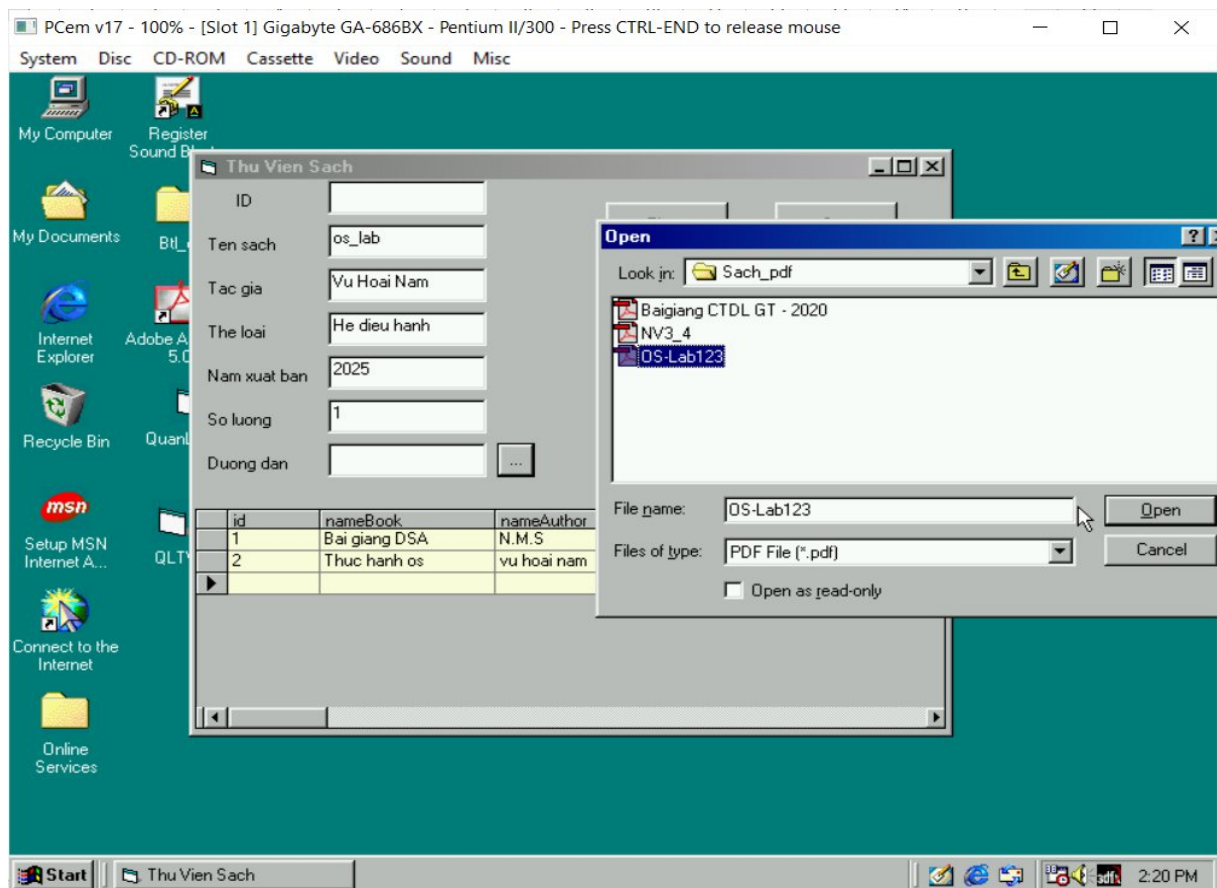
- formMain



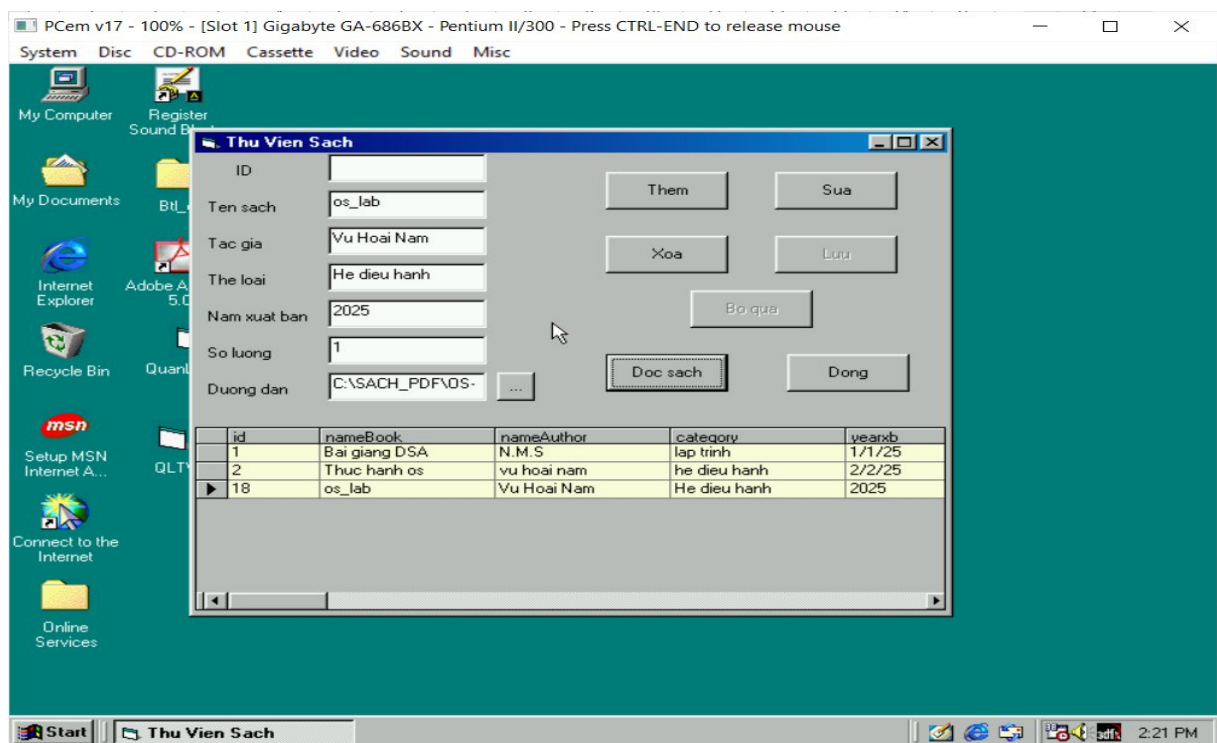
- formLibra (sau khi vào thư viện)



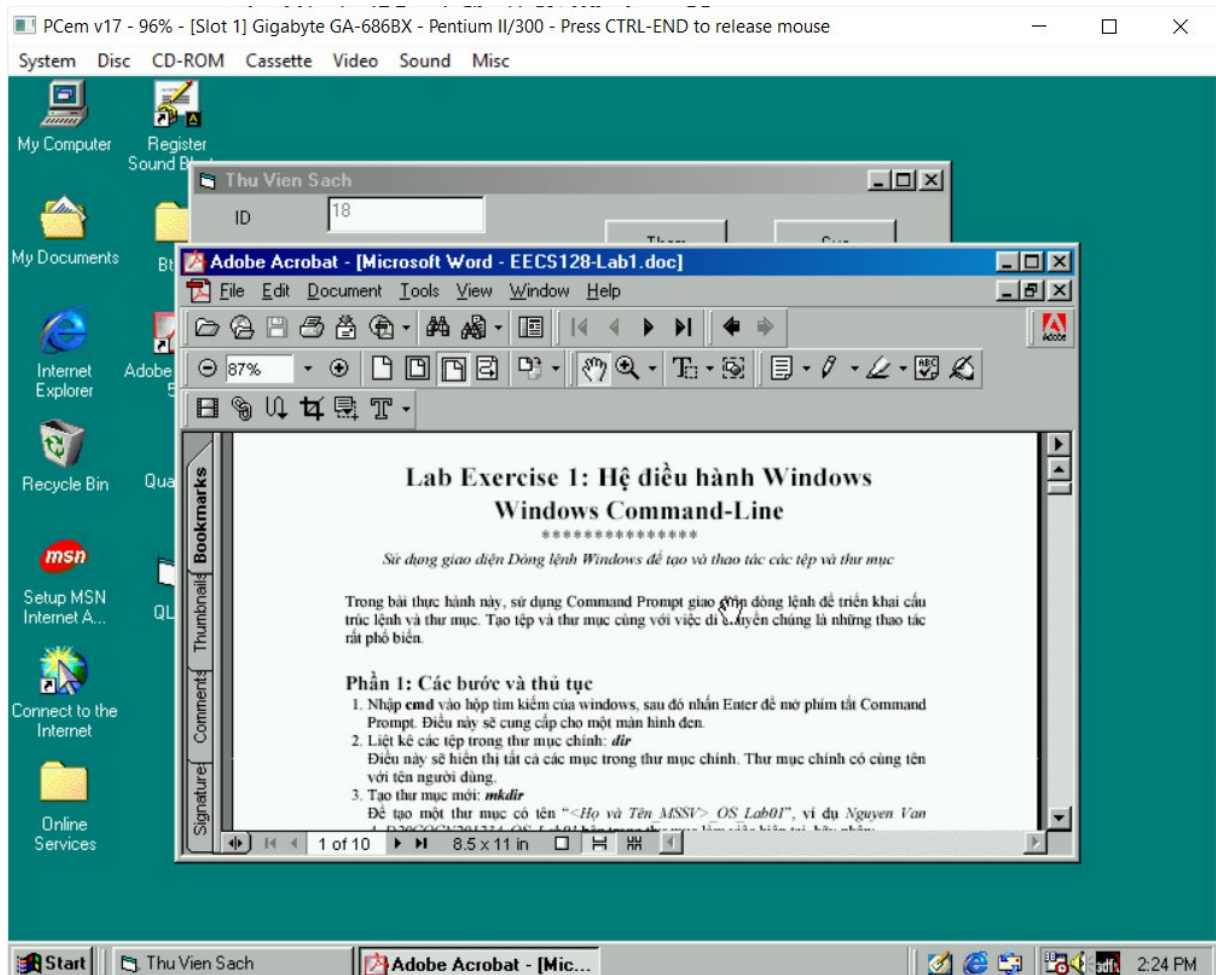
- Thêm sách (ấn vào nút “Them”). Sau đó điền thông tin vào. Trong đó mục đường dẫn thì ấn dấu 3 chấm bên cạnh để tìm file trong máy tính. Sau đó ấn nút “Luu” để thêm sách



Sau khi thêm thì bảng bên dưới sẽ hiển thị thêm 1 dòng



- Khi xóa thì chỉ cần ấn vào dòng muốn xóa trong DataGrid1 rồi ấn nút xóa. Lập tức, dòng (quyển sách) đó sẽ bị xóa
- Chức năng đọc sách: Người dùng ấn vào 1 dòng sách muốn đọc rồi ấn nút đọc sách, lúc này thì máy tính sẽ khởi động phần mềm adobe acrobat(pdf) để mở file



6. Kết luận