**Tìm hiểu về Controls**

## **1. Input Design (Thiết kế đầu vào)**

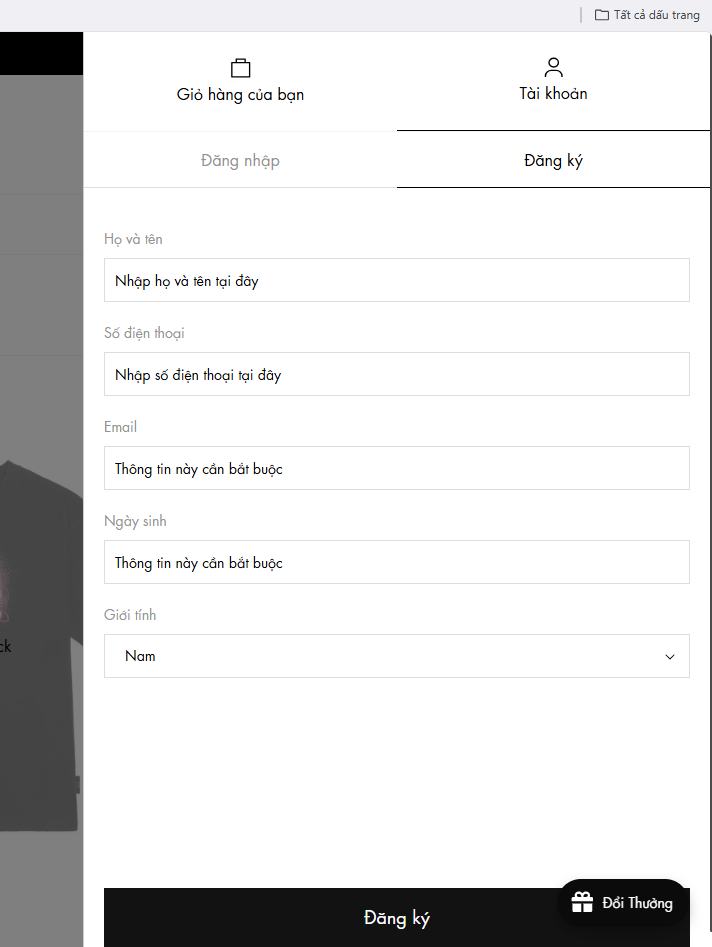
### **\*\*Khái niệm**

**Input Design** là quá trình thiết kế cách hệ thống nhận dữ liệu từ người dùng hoặc từ các nguồn khác để xử lý. Thiết kế đầu vào quyết định **tính chính xác, hiệu quả và tính khả dụng của hệ thống**.

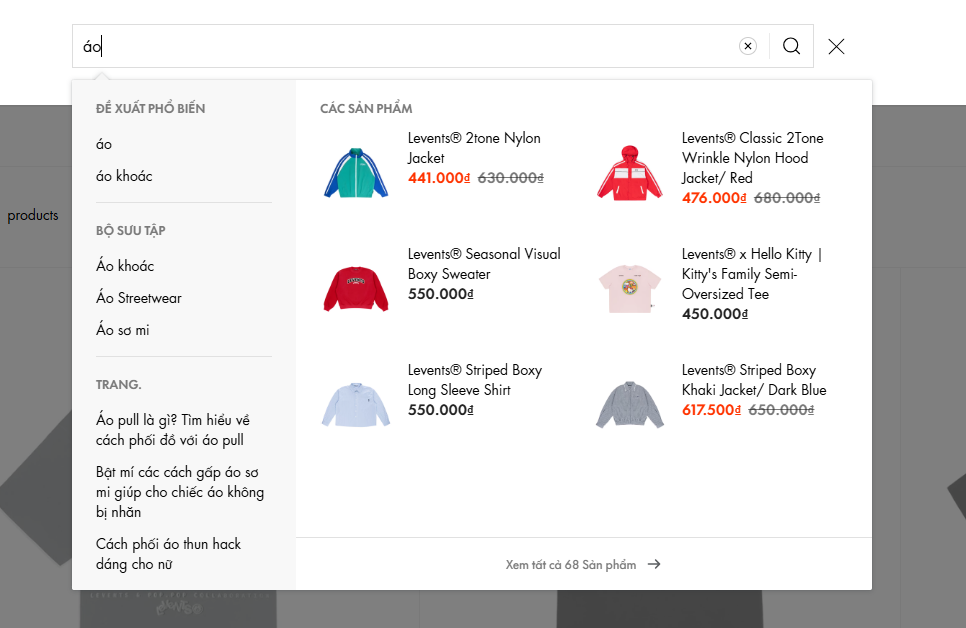
**Mục tiêu chính của Input Design:**✅ Đảm bảo dữ liệu đầu vào chính xác, đầy đủ, không bị lỗi  
✅ Giúp người dùng nhập dữ liệu nhanh chóng và dễ dàng  
✅ Hạn chế sai sót bằng cách kiểm tra và xác thực dữ liệu

### **Các loại Input (Dữ liệu đầu vào)**

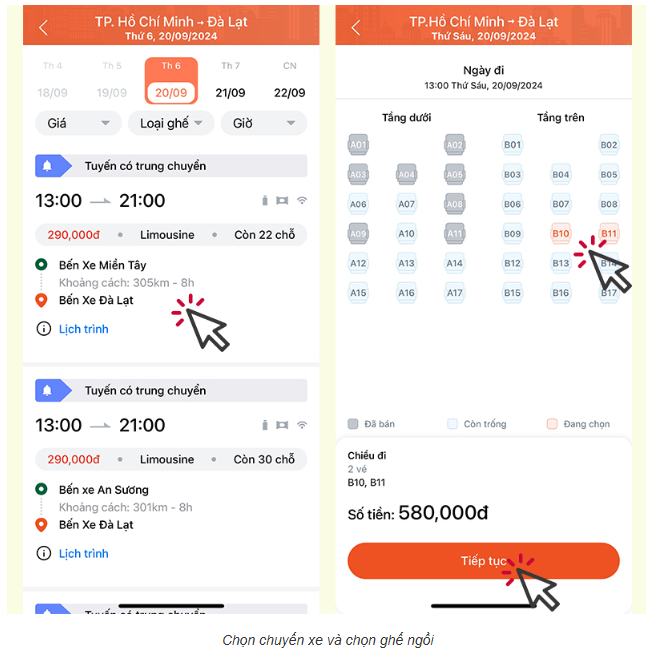
1. **Nhập dữ liệu đơn vị**: Dữ liệu riêng lẻ như thông tin khách hàng, sách, sản phẩm...  
   *Ví dụ:* Nhập họ tên, số điện thoại, địa chỉ email vào hệ thống.



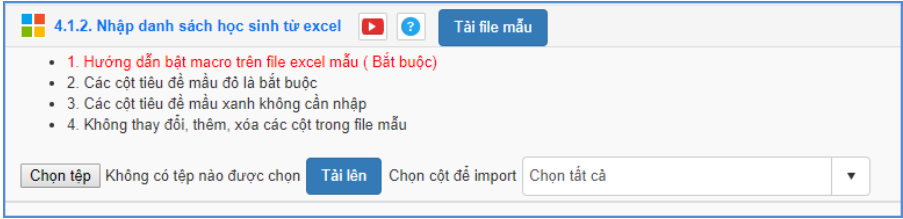
1. **Nhập dữ liệu quan hệ**: Dữ liệu có mối quan hệ với nhau như đơn hàng, mượn sách...  
   *Ví dụ:* Khi nhập một đơn hàng, cần nhập ID khách hàng, danh sách sản phẩm, phương thức thanh toán.
2. **Nhập tìm kiếm**: Dữ liệu để tra cứu thông tin.  
   *Ví dụ:* Người dùng nhập từ khóa vào ô tìm kiếm để tìm sản phẩm trên website.



1. **Nhập theo thời gian thực**: Dữ liệu được nhập và xử lý ngay lập tức, sử dụng biểu mẫu và controls.  
   *Ví dụ:* Hệ thống bán vé xe hiển thị ngay số ghế còn trống khi người dùng nhập thông tin.



1. **Nhập hàng loạt (Batch input)**: Nhập một lượng lớn dữ liệu theo nhóm, thường là file dữ liệu có cấu trúc.  
   *Ví dụ:* Nhập danh sách sinh viên từ file Excel vào hệ thống quản lý sinh viên.



### **Nguyên tắc thiết kế Input Design**

✔ **Đơn giản, tự nhiên**: Giao diện nhập liệu cần thân thiện, dễ hiểu.  
✔ **Chỉ nhập thông tin cần thiết**: Tránh bắt buộc nhập quá nhiều dữ liệu không cần thiết.  
✔ **Cung cấp giá trị mặc định**: Giúp giảm thao tác nhập thủ công.  
✔ **Sử dụng danh sách lựa chọn (Dropdown, ComboBox)**: Giảm sai sót khi nhập dữ liệu.  
✔ **Có kiểm tra lỗi (Validation)**: Ví dụ: Email phải đúng định dạng, số điện thoại chỉ chứa số.  
✔ **Có hướng dẫn và ví dụ**: Hiển thị format yêu cầu (VD: Ngày sinh: dd/mm/yyyy).  
✔ **Cho phép thoát khỏi màn hình nhập liệu mà không mất dữ liệu**.

1. Output design

Trong thiết kế giao diện (UI - User Interface), "output design" thường đề cập đến quá trình thiết kế cách thông tin hoặc kết quả được hiển thị cho người dùng sau khi họ tương tác với hệ thống. Đây là một phần quan trọng để đảm bảo trải nghiệm người dùng (UX) mượt mà, dễ hiểu và trực quan. Dưới đây là một số khía cạnh chính của output design trong thiết kế giao diện:

### **1. Mục đích của Output Design**

* **Truyền tải thông tin rõ ràng**: Đảm bảo người dùng hiểu được dữ liệu hoặc phản hồi từ hệ thống (ví dụ: thông báo thành công, lỗi, hoặc kết quả tìm kiếm).
* **Hỗ trợ quyết định**: Giúp người dùng đưa ra hành động tiếp theo dựa trên thông tin được trình bày.
* **Tăng tính thẩm mỹ**: Kết hợp giữa chức năng và hình thức để giao diện đẹp mắt và chuyên nghiệp.

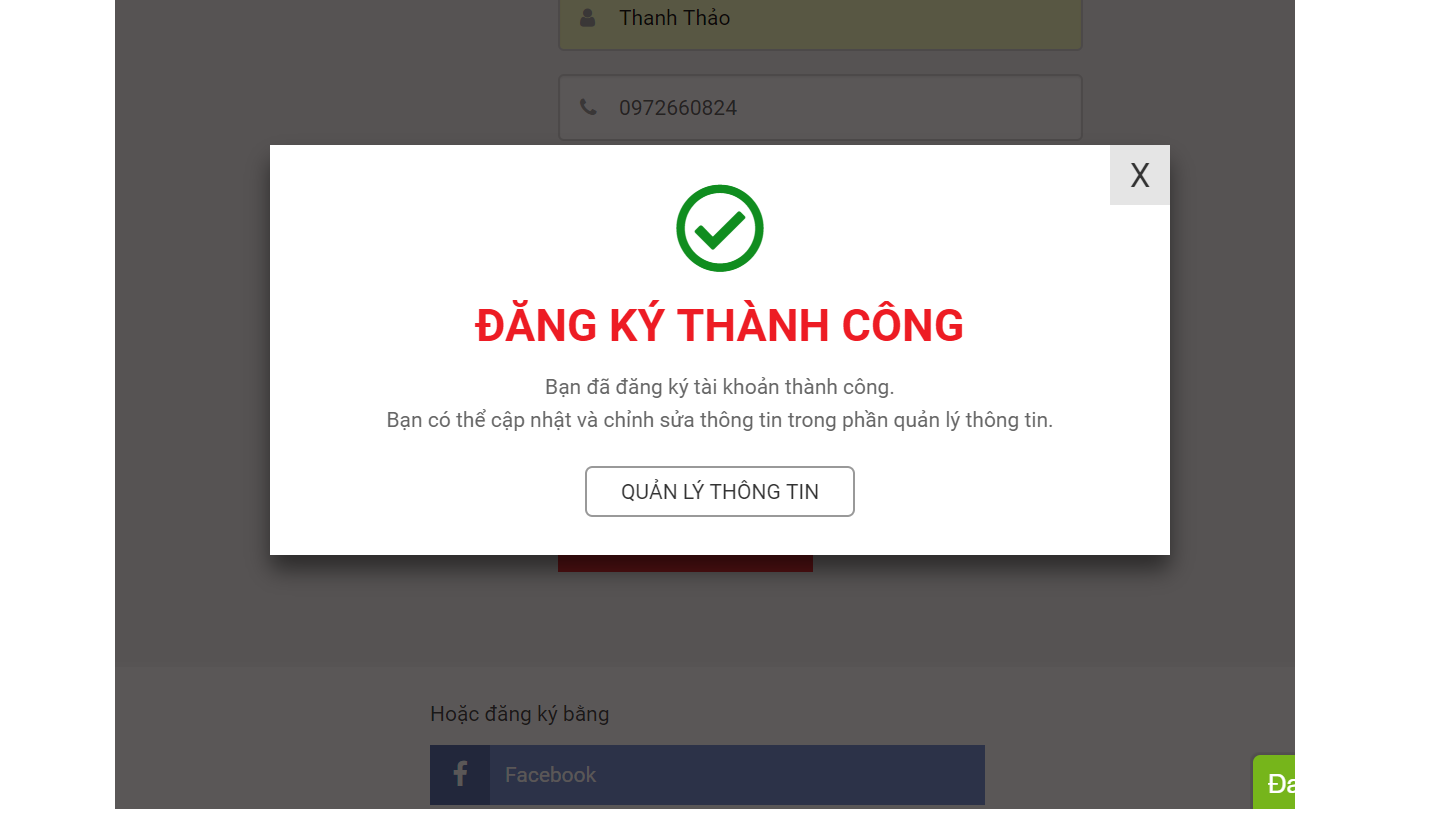
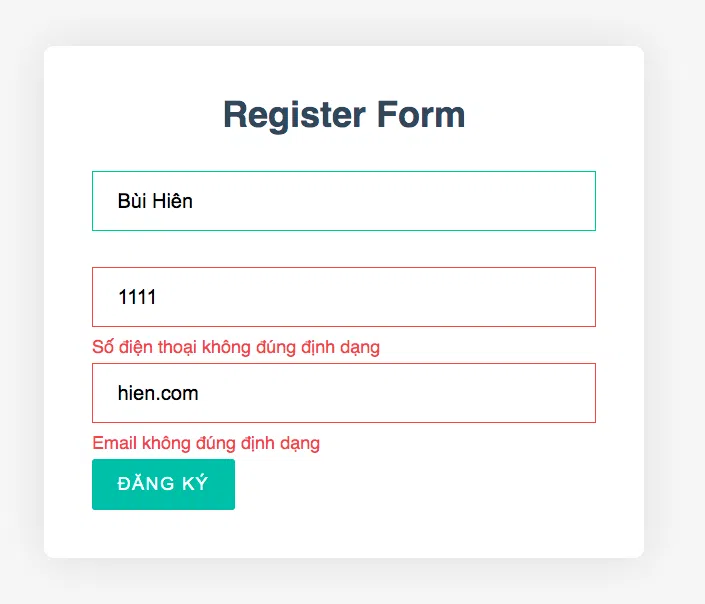
### **2. Các yếu tố chính trong Output Design**

* **Bố cục (Layout)**:  
  + Sắp xếp thông tin hợp lý, tránh lộn xộn.
  + Sử dụng lưới (grid system) để tạo sự nhất quán.
* **Typography**:  
  + Chọn font chữ dễ đọc, kích thước phù hợp.
  + Phân cấp tiêu đề, nội dung chính, và chú thích rõ ràng.
* **Màu sắc**:  
  + Dùng màu để nhấn mạnh (ví dụ: màu đỏ cho lỗi, xanh lá cho thành công).
  + Đảm bảo độ tương phản giữa văn bản và nền.
* **Biểu tượng và hình ảnh**:  
  + Sử dụng icon hoặc hình minh họa để trực quan hóa thông tin.
  + Đảm bảo hình ảnh không làm rối giao diện.
* **Phản hồi (Feedback)**:  
  + Hiển thị thông báo hoặc kết quả ngay sau hành động của người dùng (ví dụ: "Đã lưu thành công").
  + Sử dụng animation nhẹ để tăng tính tương tác.

### **3. Nguyên tắc thiết kế Output**

* **Đơn giản hóa**: Trình bày chỉ những thông tin cần thiết, tránh quá tải dữ liệu.
* **Nhất quán**: Giữ phong cách thiết kế đồng bộ trên toàn bộ giao diện (màu sắc, font, cách hiển thị).
* **Dễ tiếp cận**: Đảm bảo output phù hợp với mọi đối tượng người dùng, bao gồm cả người khuyết tật (ví dụ: hỗ trợ screen reader).
* **Thời gian thực**: Nếu có thể, cung cấp kết quả ngay lập tức để tăng tính hiệu quả.

### **4. Ví dụ thực tế**

* **Form đăng ký**: Sau khi người dùng nhấn "Gửi", output có thể là một thông báo "Đăng ký thành công" với nút dẫn đến bước tiếp theo.
* 
* **Dashboard**: Hiển thị số liệu bằng biểu đồ hoặc bảng, với màu sắc phân biệt rõ các loại dữ liệu.
* 
* **Thông báo lỗi**: Một dòng thông báo lỗi nhập email và số điện thoại không đúng định dạng
* 

II. Control :

1. Introduction

**Controls (hay còn gọi là Widgets)** là các thành phần giao diện người dùng có thể **tái sử dụng**, giúp người dùng **tương tác với hệ thống** thông qua nhập dữ liệu (input) và nhận kết quả (output).

**Mục đích của Controls:**✅ Giúp giao diện dễ sử dụng, trực quan hơn  
✅ Cho phép người dùng nhập, chọn, điều hướng và thực hiện các thao tác trên hệ thống  
✅ Tối ưu hiệu suất làm việc bằng cách chuẩn hóa các thành phần giao diện

**Các đặc điểm của Controls:**🔹 Có thể sử dụng lại trong nhiều ứng dụng  
🔹 Hỗ trợ xử lý sự kiện (event handling) như nhấn nút, chọn danh sách  
🔹 Có nhiều phong cách hiển thị khác nhau theo nền tảng (Windows, MacOS, Web...)

2. Some typical control types

#### **A. Controls dùng để chứa nội dung (Container Controls)**

Những control này giúp tổ chức và quản lý các thành phần khác bên trong nó.

1️⃣ **Dialog (Hộp thoại)**

* **Modal dialog**: Hộp thoại chặn tất cả các thao tác khác cho đến khi đóng (Ví dụ: Hộp thoại xác nhận "Bạn có chắc chắn muốn thoát không?")
* **Modeless dialog**: Hộp thoại không chặn, cho phép tiếp tục thao tác với cửa sổ chính (Ví dụ: Hộp thoại Tìm kiếm trong Microsoft Word)

2️⃣ **Panel (Bảng chứa nội dung)**

* Dùng để nhóm nhiều controls khác nhau
* **FlowLayoutPanel**: Sắp xếp các control theo dòng chảy
* **TableLayoutPanel**: Chia layout theo bảng

3️⃣ **TabControl**

* Tạo giao diện nhiều tab (giống trình duyệt web)

#### **B. Controls dùng để tổ chức nội dung (Organizing Controls)**

Giúp hiển thị dữ liệu một cách có hệ thống và trực quan.

1️⃣ **Scrollbar (Thanh cuộn)**

* Dùng để cuộn nội dung khi bị giới hạn không gian

2️⃣ **Groupbox (Hộp nhóm)**

* Nhóm các controls liên quan lại với nhau

3️⃣ **Listbox (Danh sách lựa chọn)**

* Hiển thị danh sách các mục để chọn
* *Ví dụ:* Danh sách các tỉnh thành trong form đăng ký

4️⃣ **Combobox (Hộp chọn thả xuống)**

* Cho phép chọn một mục từ danh sách thả xuống
* *Ví dụ:* Chọn quốc gia từ danh sách

#### **C. Controls dùng để nhập văn bản (Text Controls)**

Dùng để người dùng nhập dữ liệu dạng văn bản.

1️⃣ **Textbox (Ô nhập liệu)**

* Cho phép nhập một dòng văn bản
* Hỗ trợ nhập liệu có kiểm tra định dạng

#### **D. Controls dùng để lựa chọn (Selection Controls)**

Dùng để chọn các tùy chọn trong danh sách.

1️⃣ **Checkbox**

* Cho phép chọn nhiều tùy chọn
* *Ví dụ:* Chọn các sở thích trong form đăng ký

2️⃣ **Radio Button**

* Cho phép chọn một tùy chọn duy nhất trong một nhóm
* *Ví dụ:* Chọn giới tính Nam/Nữ

3️⃣ **Toggle Button (Nút bật/tắt)**

* Chuyển đổi giữa hai trạng thái
* *Ví dụ:* Chuyển đổi chế độ sáng/tối trên điện thoại

#### **E. Controls dùng để thực hiện lệnh (Command Controls)**

Giúp người dùng thực hiện các thao tác.

1️⃣ **Button (Nút bấm)**

* Kích hoạt một hành động khi bấm vào
* *Ví dụ:* Nút "Đăng nhập" trong form đăng nhập

2️⃣ **Menu (Thanh menu)**

* Hiển thị danh sách tùy chọn theo dạng menu kéo xuống
* *Ví dụ:* Menu "File" trong Microsoft Word

3️⃣ **Context Menu**

* Menu xuất hiện khi bấm chuột phải
* *Ví dụ:* Menu chuột phải trên màn hình desktop

4️⃣ **Toolbar (Thanh công cụ)**

* Tập hợp các nút tắt giúp truy cập nhanh tính năng
* *Ví dụ:* Thanh công cụ trong trình duyệt web

5️⃣ **Hyperlink (Liên kết)**

* Dùng để mở trang web hoặc điều hướng trong ứng dụng

6️⃣ **Ribbon**

* Thanh công cụ kiểu mới thay thế menu truyền thống
* *Ví dụ:* Ribbon trong Microsoft Office

#### **F. Controls đặc biệt (Other Controls)**

1️⃣ **DateTimePicker**

* Cho phép chọn ngày tháng

2️⃣ **TreeView**

* Hiển thị dữ liệu dạng cây phân cấp
* *Ví dụ:* Trình quản lý file của Windows

3️⃣ **Spinner (DomainUpDown)**

* Cho phép chọn giá trị bằng cách bấm nút tăng/giảm

#### **G. Controls để hiển thị dữ liệu (Data Controls)**

1️⃣ **DataGrid (Bảng dữ liệu)**

* Hiển thị dữ liệu theo dạng bảng
* *Ví dụ:* Danh sách đơn hàng trong phần mềm quản lý

3. User Controls

**Khái niệm**

* Là các control được tạo ra bằng cách **kế thừa các control có sẵn** và bổ sung tính năng mới.
* Giúp **tăng khả năng tái sử dụng** và tạo giao diện chuyên biệt cho ứng dụng.

**Ví dụ:**

* Tạo **User Control nhập địa chỉ** gồm Textbox (Nhập số nhà, tên đường), Combobox (Chọn tỉnh/thành phố), Button (Mở Google Maps).
* Tạo **User Control chọn màu sắc** gồm danh sách màu phổ biến, bộ chọn màu nâng cao.

**Các bộ điều khiển thương mại phổ biến:**🔹 **Java**: Chủ yếu là widget miễn phí  
🔹 **.NET**: Các thư viện chuyên nghiệp như:

* **DevExpress WinForm** (~799$)
* **Telerik RadControls** (~799$)
* **Infragistics NetAdvantage** (~995$)

4. kết luận

* **Controls** là thành phần không thể thiếu trong thiết kế giao diện.
* **Hiểu rõ từng loại control** giúp chọn lựa phù hợp để tăng tính tiện dụng và hiệu suất làm việc.
* **Tùy chỉnh User Control** giúp tạo giao diện đặc biệt, chuyên nghiệp hơn.
* **Sử dụng bộ control chuyên nghiệp** giúp tăng tốc độ phát triển phần mềm.

# III. User control:

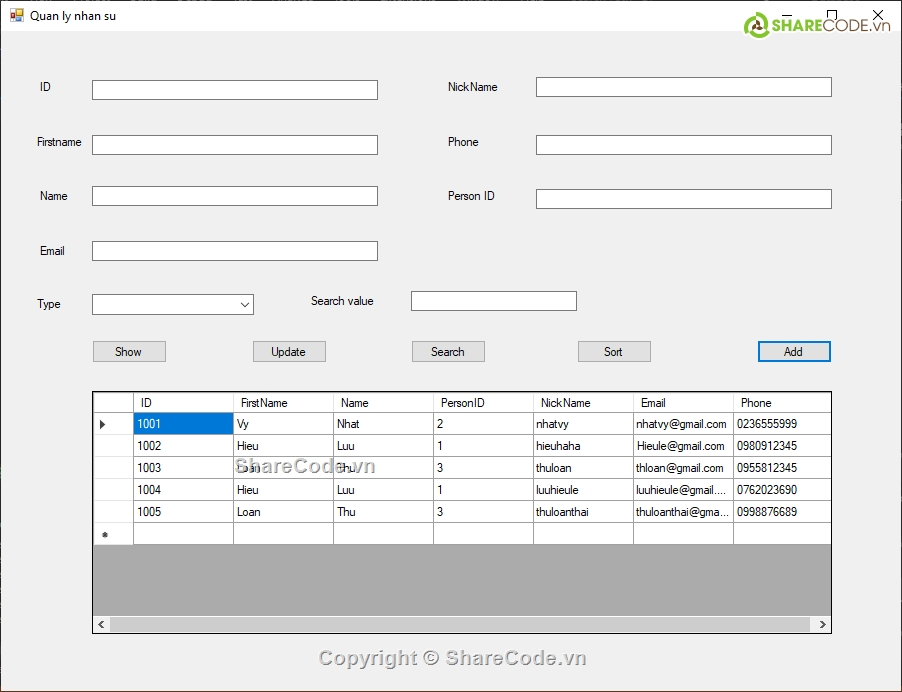
### **1. Thiết kế giao diện & Input/Output**

* **Input Design**: Nhập dữ liệu theo hai dạng:
  + **Real-time input**: Sử dụng form và các control để nhập dữ liệu trực tiếp.
  + **Batch input**: Nhập dữ liệu hàng loạt từ nguồn có cấu trúc (e.g. import file CSV).

VD : **Input Design trong C#**

Giả sử chúng ta cần nhập thông tin khách hàng với form đơn giản.

#### **C# WinForms**



* **Output Design**: Xuất dữ liệu ra màn hình, máy in, file, hoặc tìm kiếm.

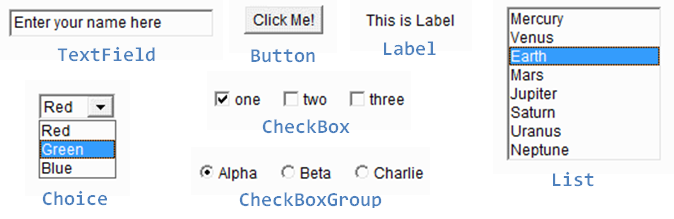
VD : **Output Design - Tìm kiếm thông tin sinh viên**

****

### **2. Controls trong lập trình**

* **Controls là gì?**
  + Các thành phần UI có thể tái sử dụng (widgets).
  + Xử lý cả dữ liệu đầu vào và đầu ra.
  + Có thể kích hoạt sự kiện thông qua các **callback function** như **delegate, listener**.
* **Các kiểu control phổ biến**:
  + **Tổ chức nội dung**: Scrollbar, Groupbox, Listbox, Combobox.
  + **Nhập văn bản**: Textbox.
  + **Lựa chọn**: Checkbox, Radio Button, Toggle Button.
  + **Thao tác lệnh**: Button, Menu, Context Menu, Toolbar.
  + **Hiển thị dữ liệu**: DataGrid, TreeView, DateTimePicker.
* **Kiến trúc UI**
  + **SDI (Single Document Interface)**: Một cửa sổ chính điều khiển giao diện, ví dụ Notepad.
  + **MDI (Multiple Document Interface)**: Cửa sổ chính chứa nhiều cửa sổ con, ví dụ MS Word.

VD :



### **3. Lập trình với Controls trong các ngôn ngữ khác nhau**

* **C#**: System.Windows.Forms
* **Java**: awt, swing, swt (Eclipse)
* **Mô hình kế thừa (Inheritance)**: Có thể kế thừa từ các control chuẩn để tạo custom controls.

VD : Hiển thị danh sách sản phẩm với DataGridView trong C#

dataGridView1.DataSource = new List<Product>

{

new Product { Name = "Laptop", Price = 1000 },

new Product { Name = "Phone", Price = 500 }

};

VD2 : Combobox chọn danh mục trong Java Swing

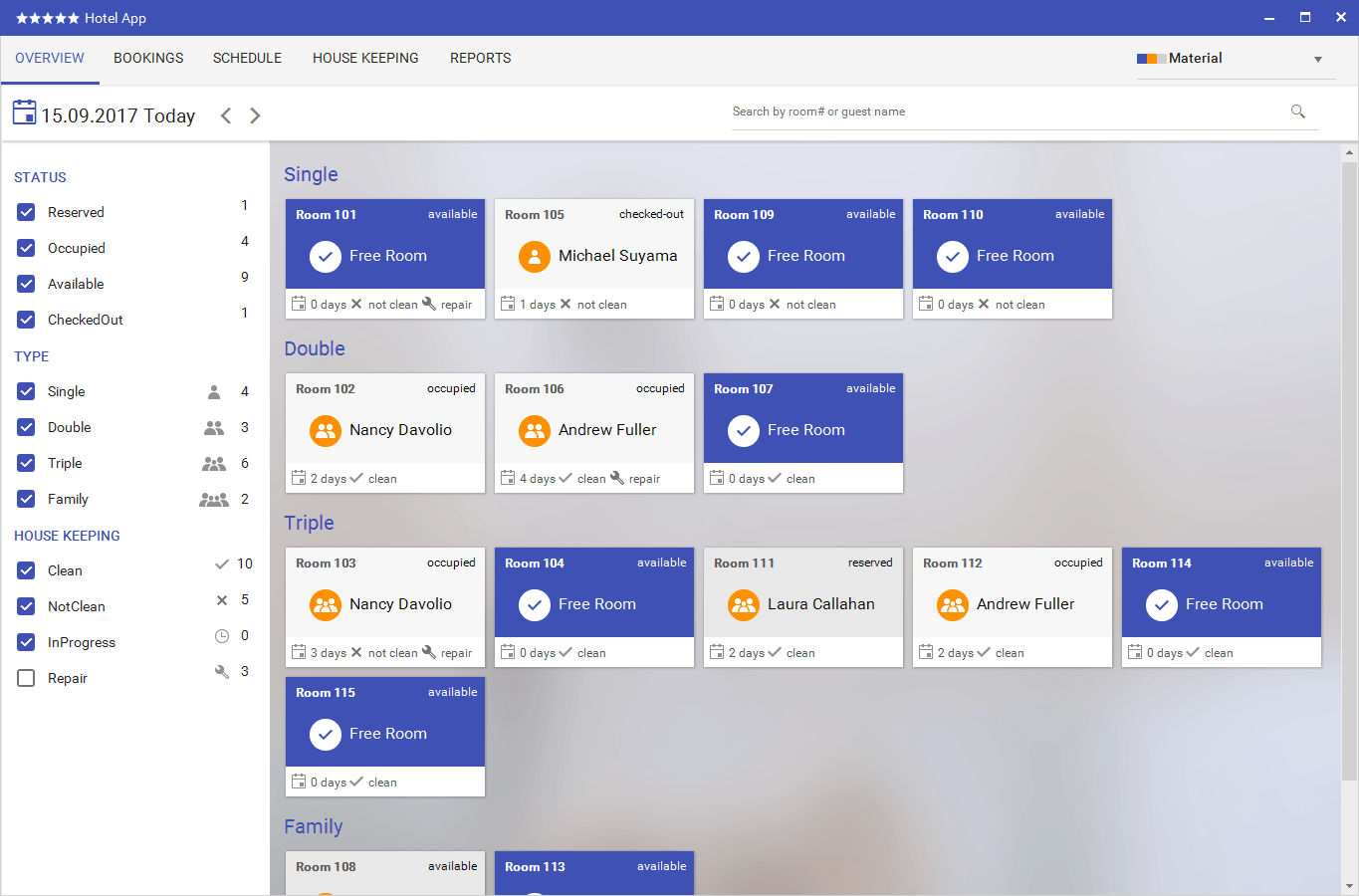
String[] items = { "Laptop", "Điện thoại", "Tablet" };

JComboBox<String> comboBox = new JComboBox<>(items);

### **4. Các thư viện UI hỗ trợ**

* **C# .NET**:
  + DevExpress WinForm ($799)
  + Telerik RadControls ($799)
  + Infragistics NetAdvantage ($995)
* **Java**: Chủ yếu là các widget miễn phí.

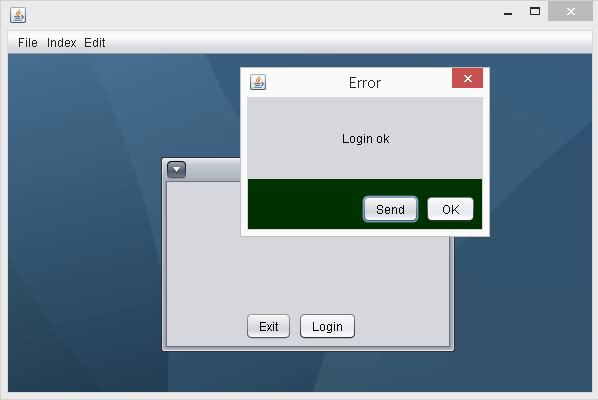
VD : Telerik UI trong C#



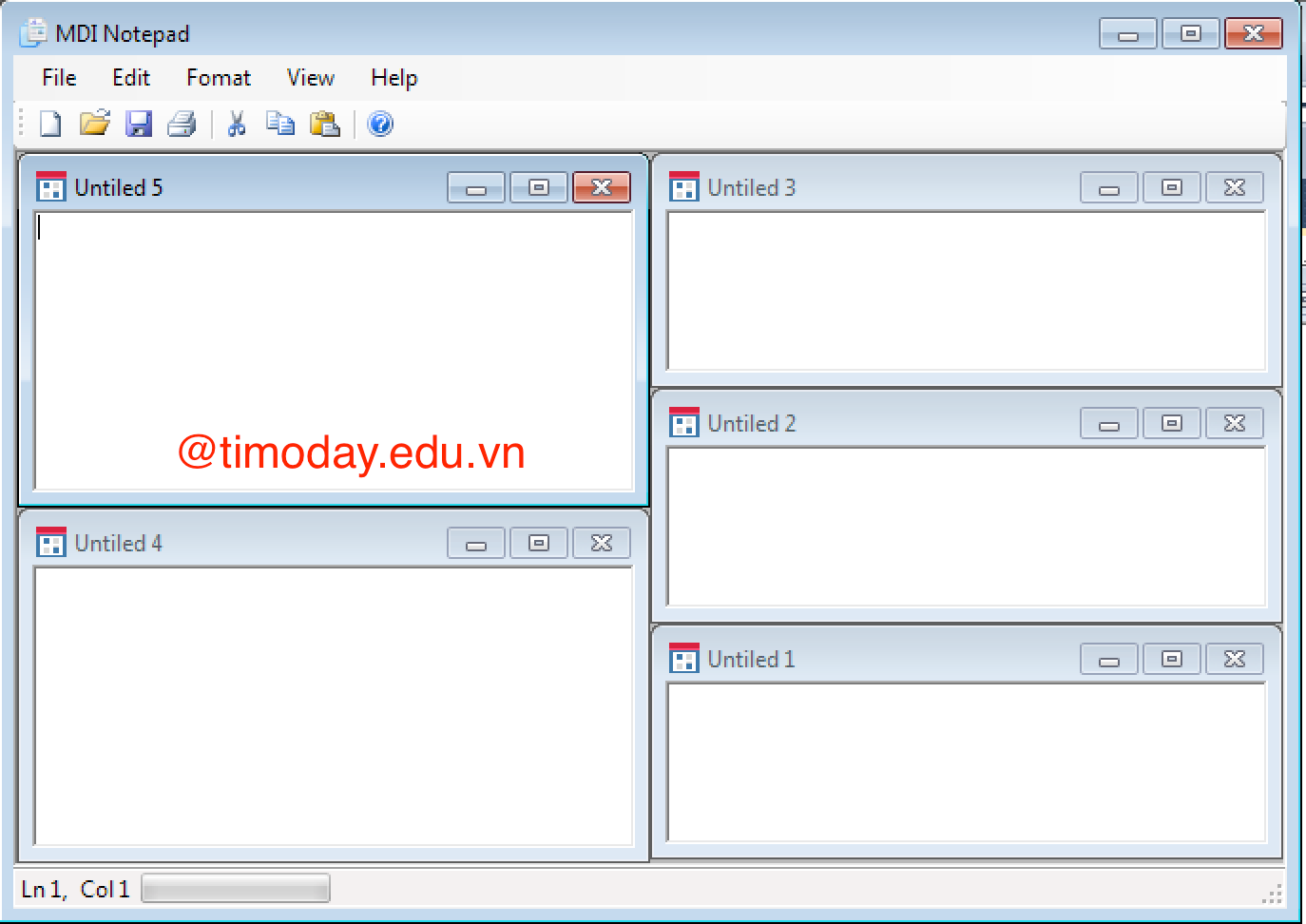
### **5. Lập trình GUI**

* **Chất lượng ứng dụng phụ thuộc vào việc sử dụng thành phần UI hiệu quả**.
* Các giải pháp kỹ thuật giúp tăng tính khả dụng (usability).

### **VD: Tạo cửa sổ chính SDI trong C#**



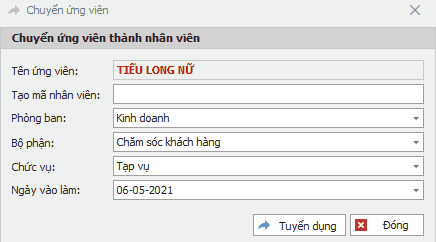
### **VD: Tạo giao diện MDI trong Java Swing**

* 

### **6. Thực hành với User Control**

* Tạo **User Control** tùy chỉnh, ví dụ nhập địa chỉ Việt Nam với C#.
* Liên kết với Google Maps, nhập số nhà, chọn quận/huyện, tỉnh/thành.

### **VD : Custom Control nhập địa chỉ trong C#**



}