Đa số người dùng  đánh giá chất lượng của 1 hệ thống thông qua giao diện hơn là thông qua chức năng. Giao diện không tốt là nguyên nhân dẫn người dùng đến lỗi.

Thiết kế giao diện người dùng không tốt là nguyên nhân dẫn đến nhiều phần mềm không được sử dụng

* Giới thiệu về thiết kế giao diện người dùng.
* Các loại giao diện người dùng.
* Nguyên lý thiết kế giao diện người dùng.
* Phương pháp thiết kế giao diện.
* Một số lưu ý khi thiết kế giao diện.
* Các thành phần thiết kế giao diện

## Giới thiệu về thiết kế giao diện người dùng

#### Khả năng nhớ hữu hạn

 Con người có thể nhớ được khoảng 7 thuật ngữ về thông tin ngay lập tức, nhưng nếu bạn đưa ra nhiều hơn thì có thể dẫn đến sai phạm.

#### Con người tạo ra lỗi

 Khi con người gây ra lỗi, hệ thống chạy sai, nếu các thông báo lỗi ko thích hợp > stress >gây nhiều lỗi hơn.

#### Khả năng của từng người khác nhau

 Khả năng của mỗi người khác nhau người thiết kế nên thiết kế theo khả năng của họ.

#### Sở thích khác nhau

Mỗi người thích tương tác với giao diện khác nhau (một số thích hình ảnh, một số thích text).

Các loại giao diện người dùng

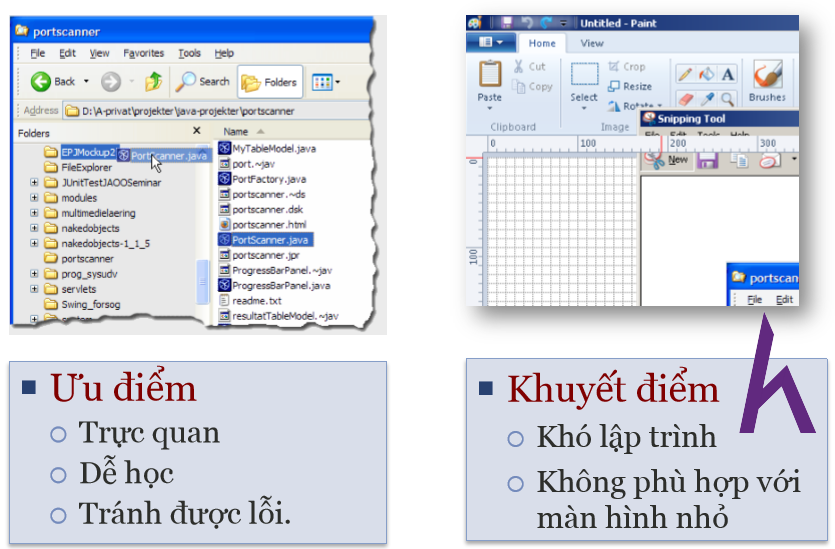
Theo hình thức tương tác

* Tương tác trực tiếp bằng tay
* Chọn menu
* Điền vào mẫu cho sẳn
* Dùng ngôn ngữ lệnh
* Dùng ngôn ngữ tự nhiên

Theo chức năng

* Giao diện nhập
* Giao diện cập nhật
* Giao diện xuất

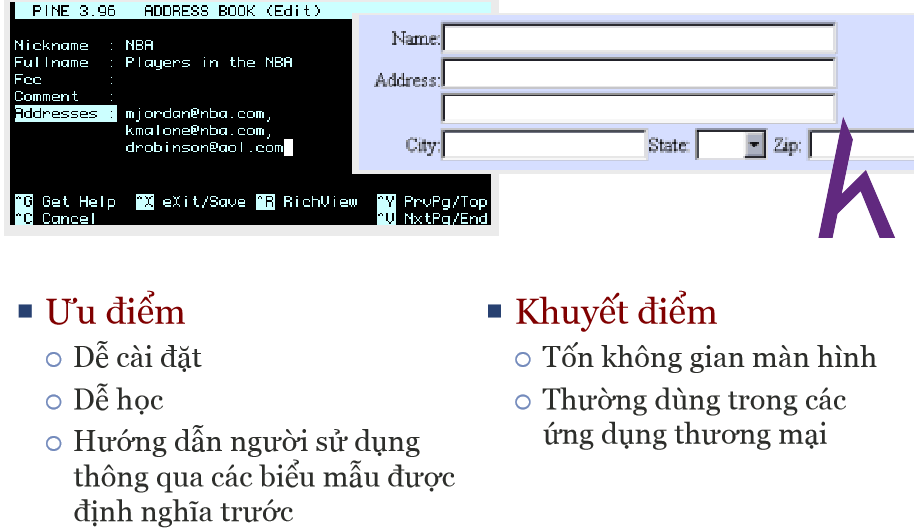
#### Tương tác trực tiếp bằng tay



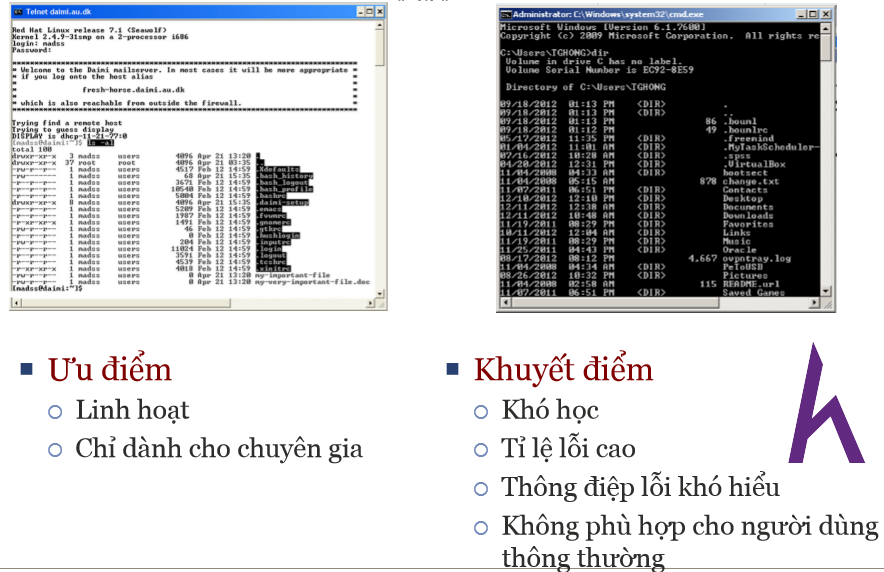
#### Chọn menu

#### kteam, howkteam, how kteam, howkteam.com, phân tích thiết kế phần mềm, cơ sở dữ liệu, mô hình thực thể, mô hình quan hệ, định danh, ER diagram, ERD,

#### Điền vào mẫu cho sẵn



#### Dùng ngôn ngữ lệnh



Dùng ngôn ngữ tự nhiên

**Ưu điểm**

* Dễ mở rộng
* Dễ sử dụng

**Khuyết điểm**

* Yêu cầu gõ nhiều
* Không đáng tin cậy

Nguyên lý thiết kế giao diện người dùng

Cần xem xét nhu cầu, kinh nghiệm và khả năng sử dụng của người dùng.

Xem xét giới hạn của người dùng có thể nhận dạng lỗi xảy ra.

 Quen thuộc với người dùng

Các thuật ngữ và khái niệm trên giao diện nên lấy từ kinh nghiệm từ những người dùng hệ thống thường xuyên

 Nhất quán

Giao diện nên được thiết kế nhất quán về thao tác trên cả hệ thống

Tối thiểu hóa sự bất ngờ

Người dùng không nên bị bất ngờ trước hành vi của hệ thống

Có khả năng khôi phục

Giao diện nên có các cơ chế nhắc nhở - khôi phục khi người dùng làm sai

Hướng dẫn người dùng

Giao diện nên có phản hồi cho người dùng khi có lỗi và hỗ trợ người dùng trong các ngữ cảnh khác nhau

Tính đa dạng cho người dùng

Giao diện nên cung cấp các tương tác thích hợp cho các người dùng khác nhau

Thiết kế trực quan

Giao diện có thể sử dụng ngay lần đầu tiên, không cần đào tạo

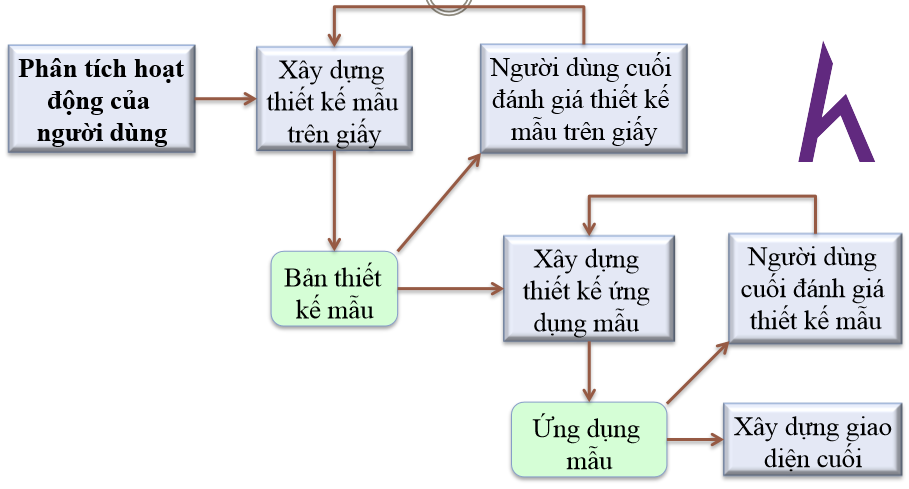
3 clicks

Hệ thống menu 3 clicks để truy cập thông tin mong muốn

Sự sáng tạo & Hội chứng gà con

Người dùng thường có khuynh hướng từ chối những hệ thống không tương tự

* Sơ đồ tổng

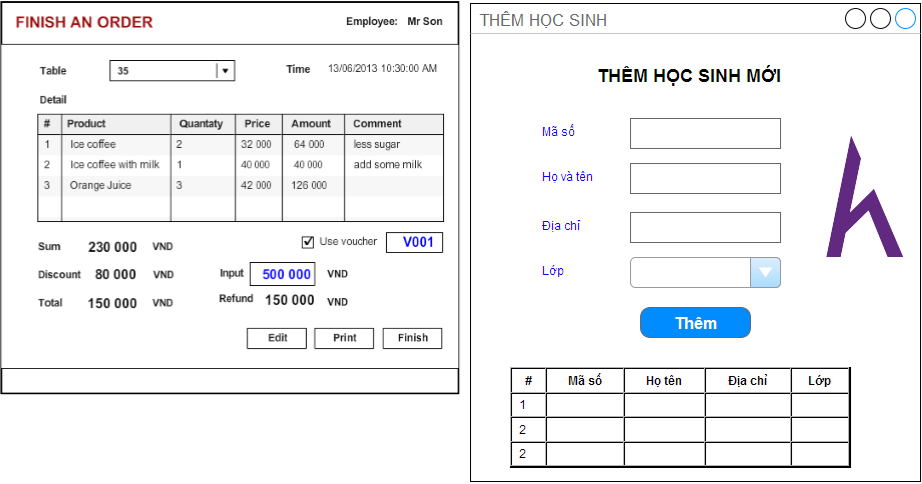


Thiết kế giao diện

Duyệt qua các UC, với mỗi UC phác thảo các giao diện của UC sử dụng một công cụ thiết kế giao diện nào đó.



## Phương pháp thiết kế giao diện



#### Bước 1: Xác định thông tin

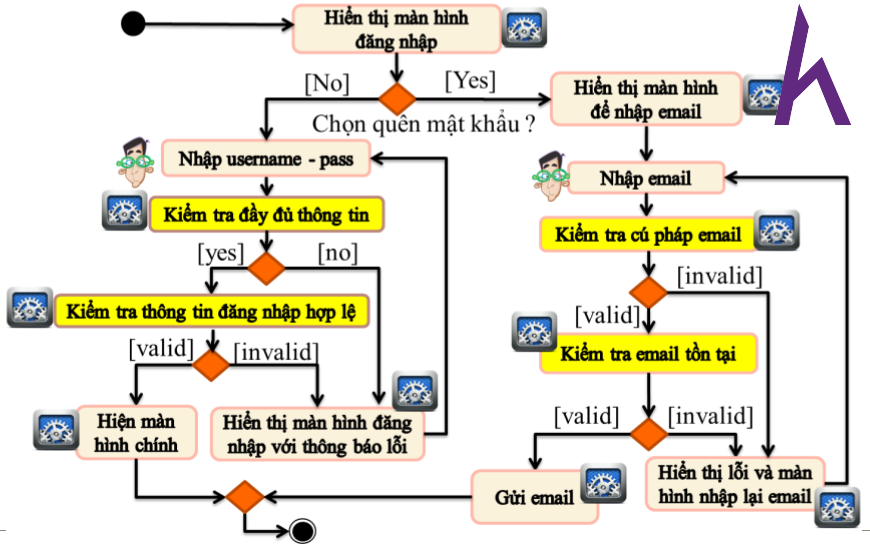
Nếu đã có các mô hình dữ liệu thì dựa vào đó để xác định thông tin (mô hình ER, mô hình quan hệ)

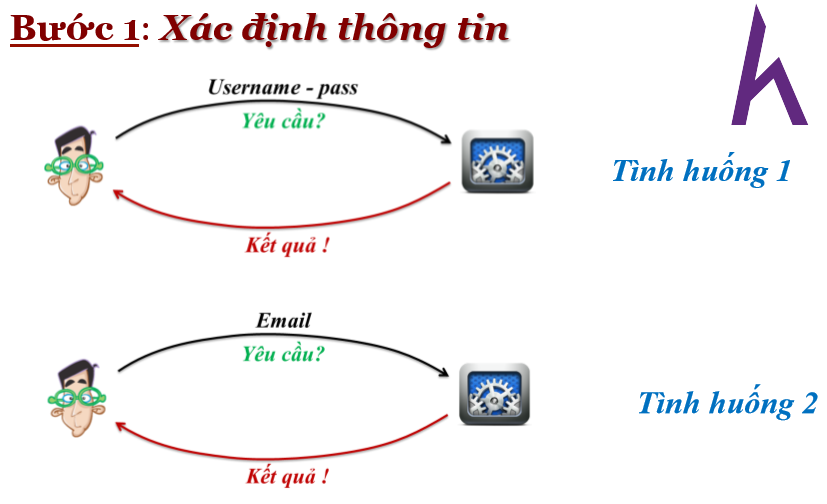
#### Bước 2: Thiết kế giao diện đúng

Chỉ quan tâm tới tính đúng đắn. Không quan tâm tới tính tiện dụng

#### Bước 3: Tinh chỉnh giao diện

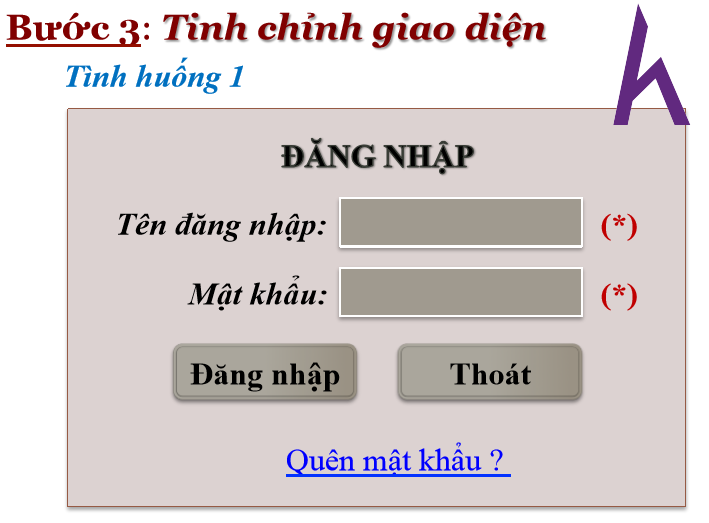
Quan tâm tới tính tiện dụng dựa vào  phân tích hoạt động của người dùng



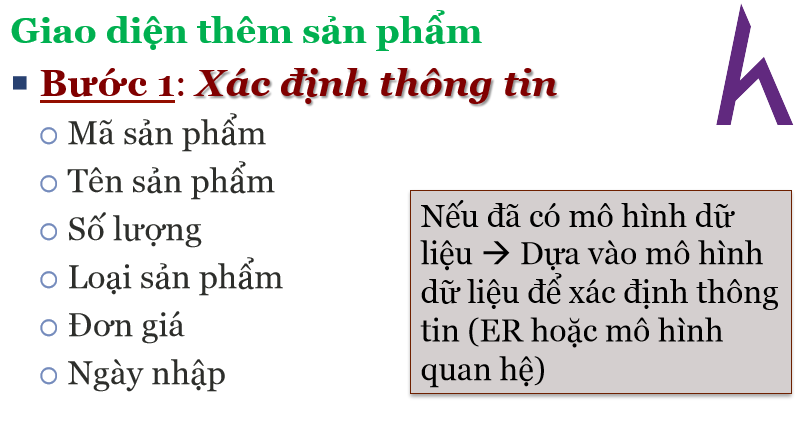


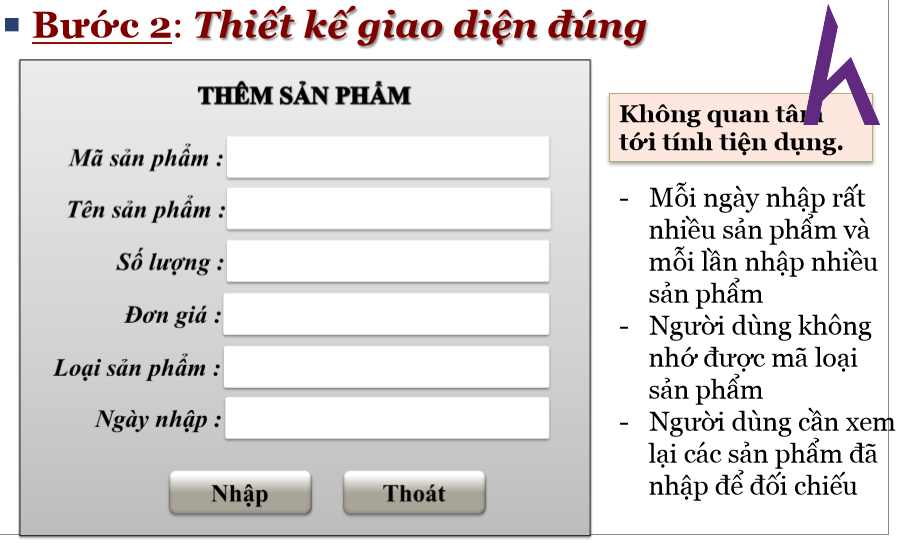
















## Một số lưu ý khi thiết kế giao diện

### Phân tích người dùng

Nếu không nắm được những gì người dùng muốn làm với hệ thống > không thiết kế được giao diện hiệu quả.

Người dùng và người thiết kế phải hiểu nhau.

Ngữ cảnh > mô tả kịch bản sử dụng. Đây là một cách để mô tả các phân tích.

### Phân tích kĩ thuật

#### Phân tích công việc

Mô tả các bước bên trong một công việc.

#### Sử dụng phỏng vấn và bảng câu hỏi

Hỏi người dùng về công việc mà họ làm

#### Quan sát

Quan sát người dùng tại nơi họ làm việc

### Nguyên tắc sử dụng màu sắc

Hạn chế số lượng màu sử dụng và thận trọng khi dùng chúng.

Dùng  sự thay đổi của màu sắc để trình bày một thay đổi trạng thái của hệ thống.

Dùng màu sắc để mã hóa các công việc mà người dùng cần thực hiện.

Cần thận trọng và nhất quán khi sử dụng màu sắc

Cẩn thận khi kết hợp màu sắc

### Thông báo lỗi

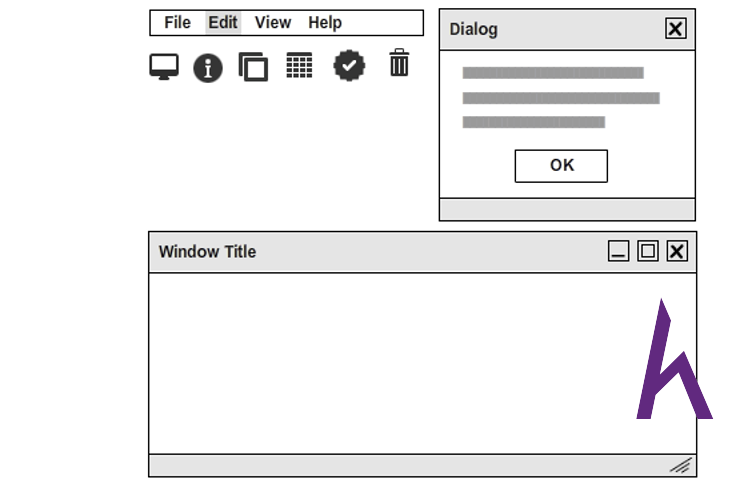
Thông báo lỗi rất quan trọng, thông báo lỗi tồi sẽ dễ làm người dùng ko muốn sử dụng hệ thống.

Thông báo lỗi nên: lịch sự , súc tích, thống nhất và có tính xây dựng.

Nền tảng và kinh nghiệm của người dùng nên là nhân tố chính được quan tâm khi thiết kế thông báo lỗi.

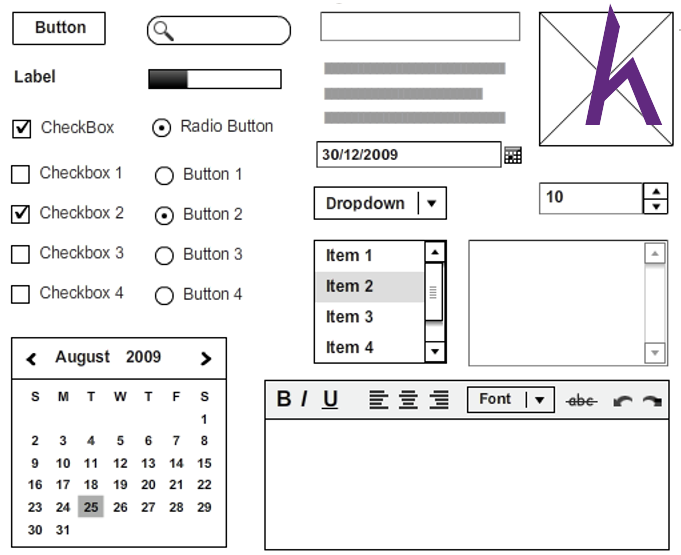
## Các thành phần thiết kế giao diện

WIMPs (Windows, Icons, Menus, Pointers)



Common Controls

* Button
* Label
* TextBox
* ListBox / ComboBox / SelectBox\* / CheckListBox
* RadioButton / CheckBox
* DatetimePicker, NumericUpDown
* Image, ProgressBar

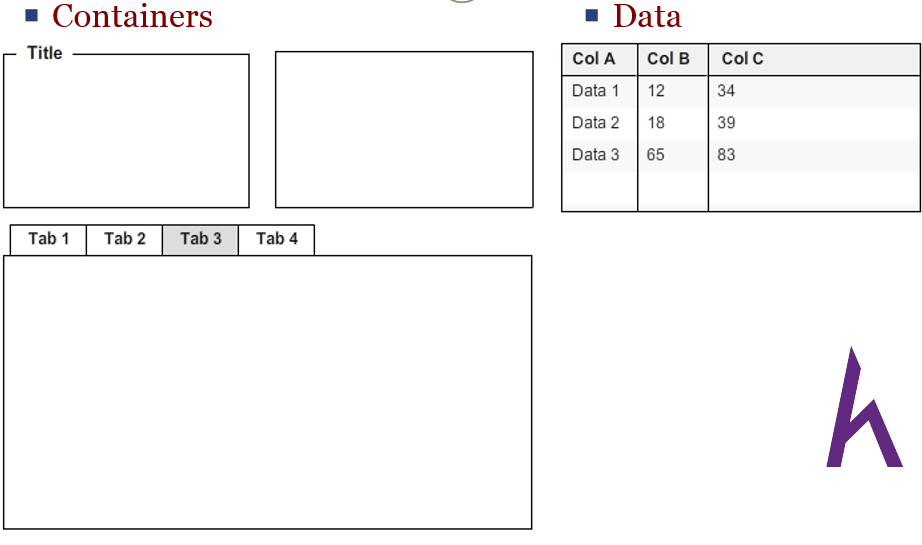


 Containers

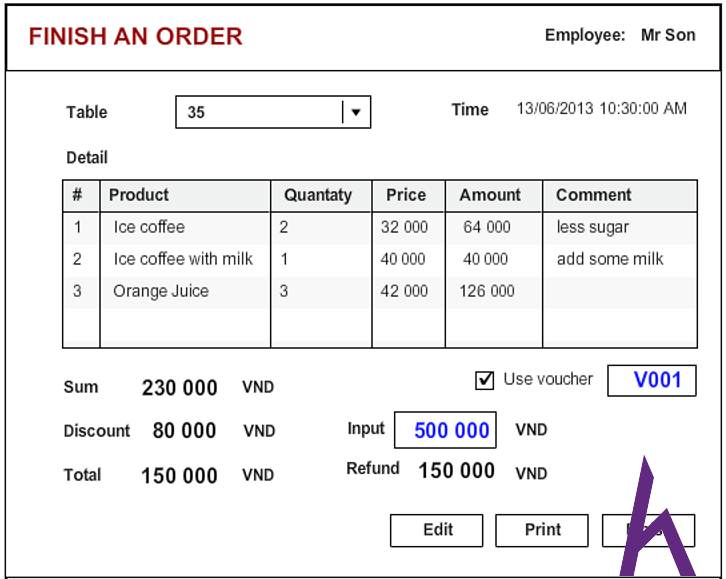
* Groupbox / Panel / Tabs

Data

* GridView / Table



**Ví dụ:**



adobe xd or balsamiq mockups -> UI

<https://www.lucidchart.com/pages/> vẽ ERD

<https://sketchboard.io> vẽ Usercase