



**FPT POLYTECHNIC**

**CHUYÊN NGÀNH NHẬP MÔN KỸ THUẬT PHẦN MỀM**

## **Bài 4: User Interface - Object Oriented Analysis and Design**

[www.poly.edu.vn](http://www.poly.edu.vn)

## Nội dung bài học

1. User Interface
2. Object-Oriented Analysis & Design

- Class Modeling
- Dynamic Modeling
- Functional Modeling



# **1. Giao diện người dùng (User Interface)**

## User Interface (UI)

- User Interface (giao diện người dùng) là một khái niệm để nói tới nơi mà con người và máy móc cùng làm việc với nhau.
- Với sự ra đời của máy tính, UI có thể coi là những gì chúng ta nhìn thấy trên màn hình và tương tác với máy tính thông qua những câu lệnh được mã hóa.
- Đối với lĩnh vực thiết kế, UI có thể coi là những tác phẩm mà các nhà thiết kế thông qua đó truyền tải thông điệp tới người sử dụng.
- Mỗi người thiết kế như những nhà lập trình phải tìm mọi cách để bất cứ ai cũng có thể hiểu và sử dụng được sản phẩm của mình.

## User Interface (UI) - Prototype

- Thực hiện cụ thể như một phần của một thiết kế hệ thống
- Để tìm hiểu vấn đề liên quan đến người dùng:
  - ✓ Để tinh chỉnh các yêu cầu người dùng
  - ✓ Để khám phá những ý tưởng thiết kế
  - ✓ Chia sẻ hoặc đồng phát triển thiết kế với những người tham gia sử dụng
  - ✓ Thực hiện một bài kiểm tra chính xác các vấn đề cụ thể
  - ✓ Thu thập dữ liệu thử nghiệm khả năng sử dụng

# User Interface (UI) - Prototype

- Screen Flow Diagram (Storyboard): (kịch bản, sơ đồ giao diện người dùng, sơ đồ dòng chảy, sơ đồ các cửa sổ chuyển hướng, bản đồ ngữ cảnh menu) được sử dụng để mô hình hóa các mối quan hệ cấp cao giữa các thành phần giao diện người dùng, từ đó đặt câu hỏi khả năng sử dụng cơ bản.

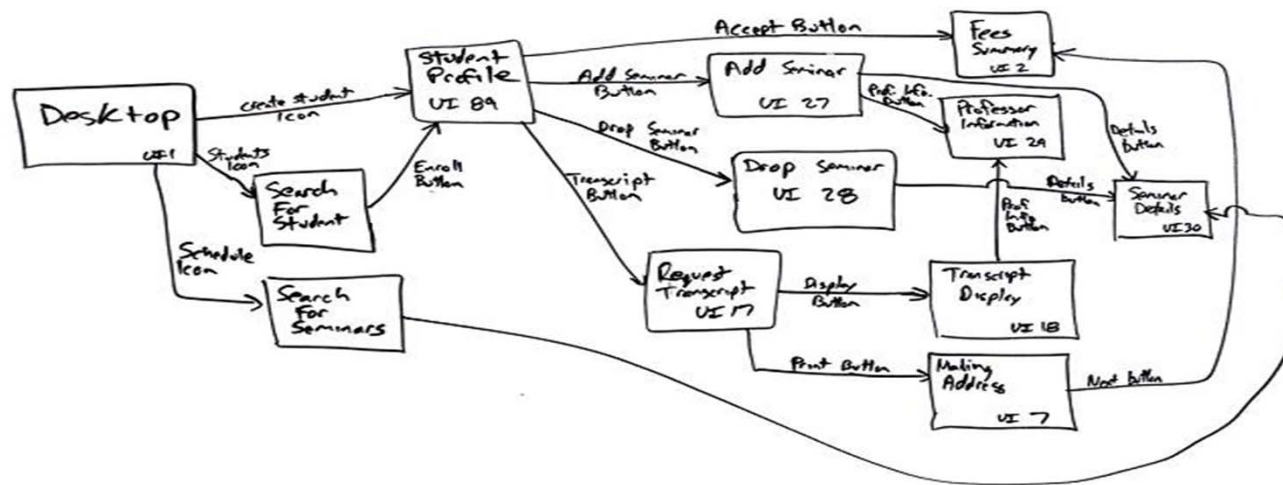


Image: <http://www.agilemodeling.com>

# User Interface (UI) - Prototype

- Screen Flow Diagram (State Machine Diagram)

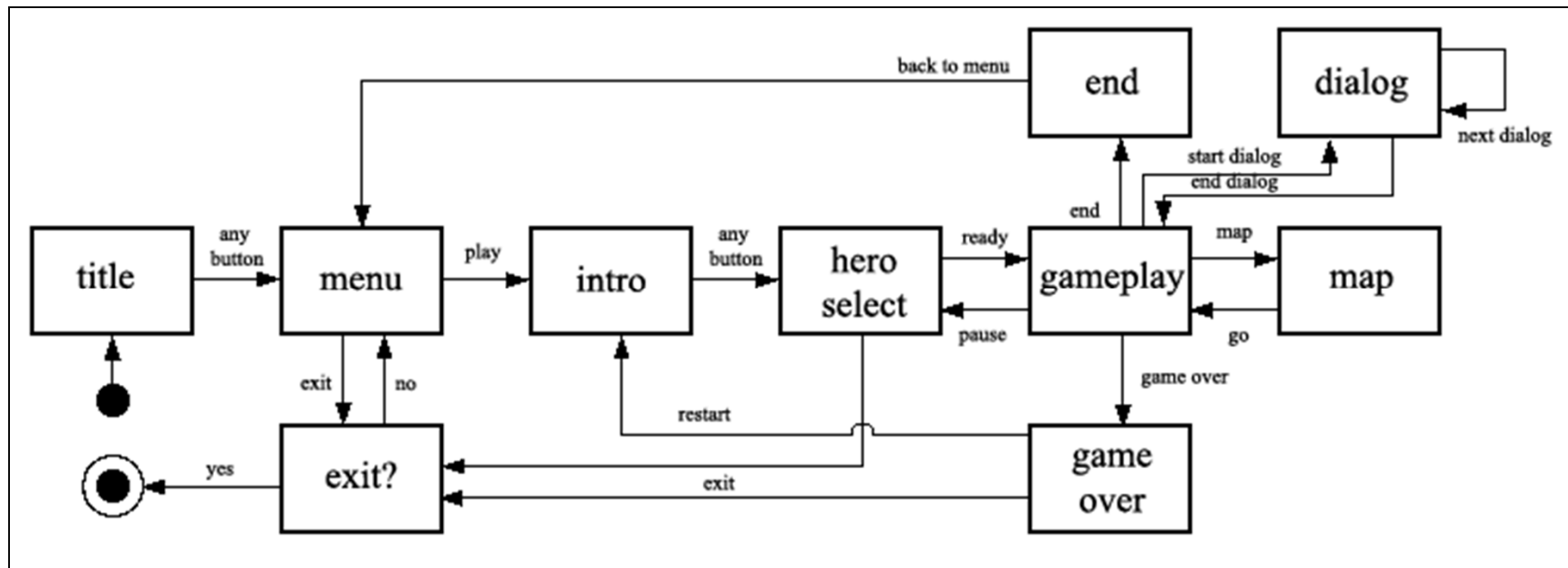


Image: <http://emr.deviantart.com>

# User Interface (UI) - Prototype

- The Interaction Flow Modeling Language (IFML)

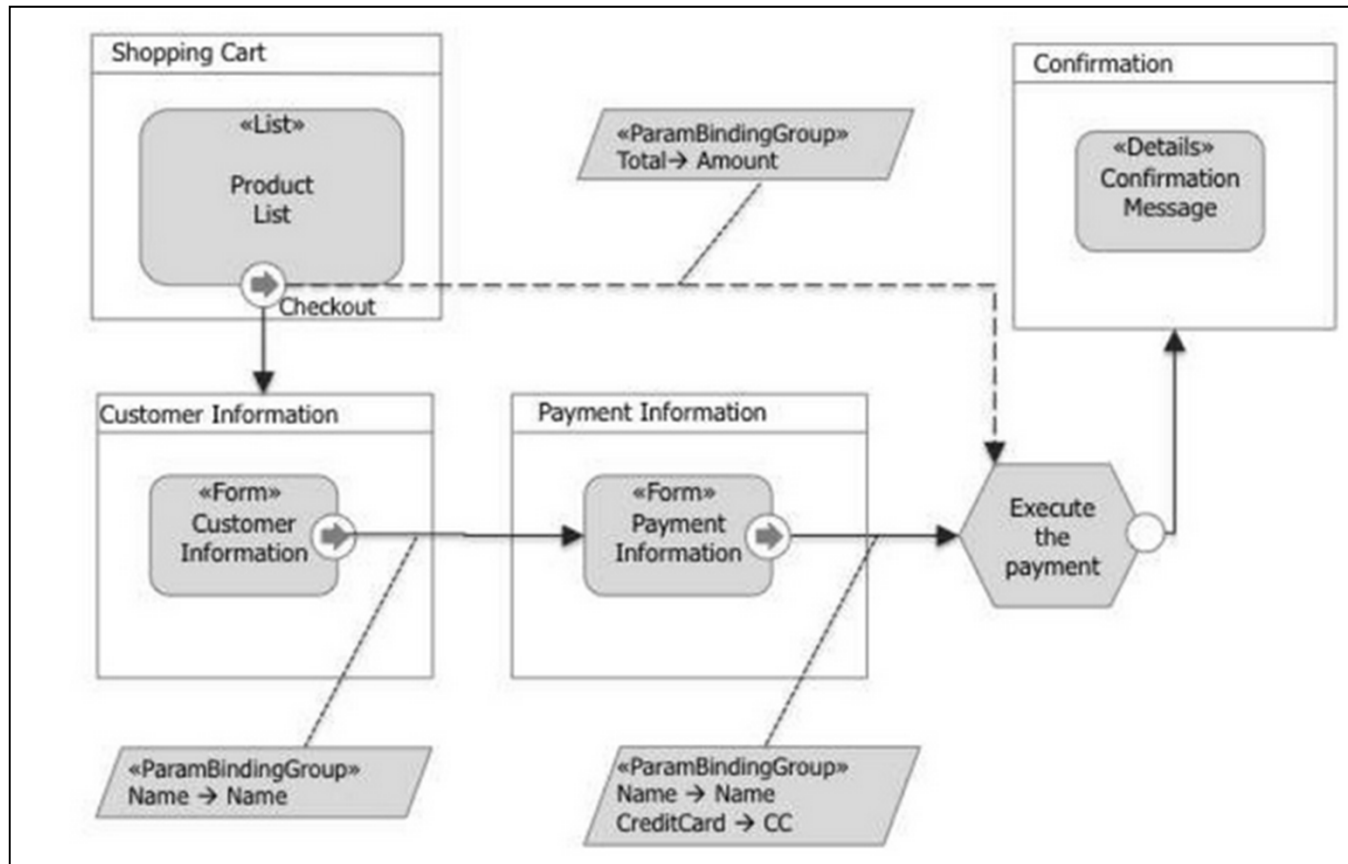
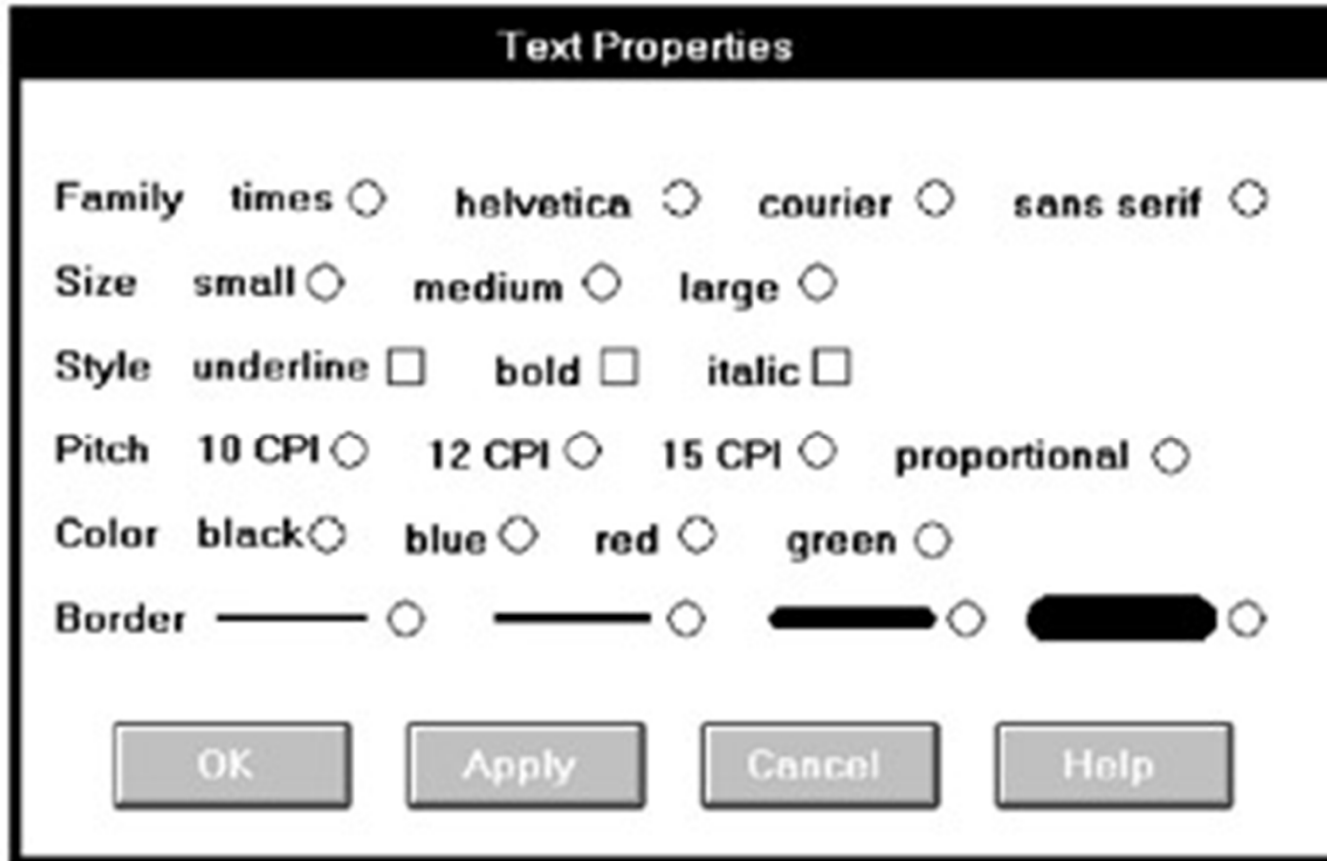


Image: <http://webratio.com>



## User Interface (UI) – Ví dụ 1



Chú thích không thể hiển rõ, từ mô tả sự lựa chọn, và các mô tả sự lựa chọn ban đầu không có, sắp xếp không logic.

Radiobutton và checkbox không được mạnh mẽ, rõ ràng dễ gây ra lựa chọn nhầm lẫn.

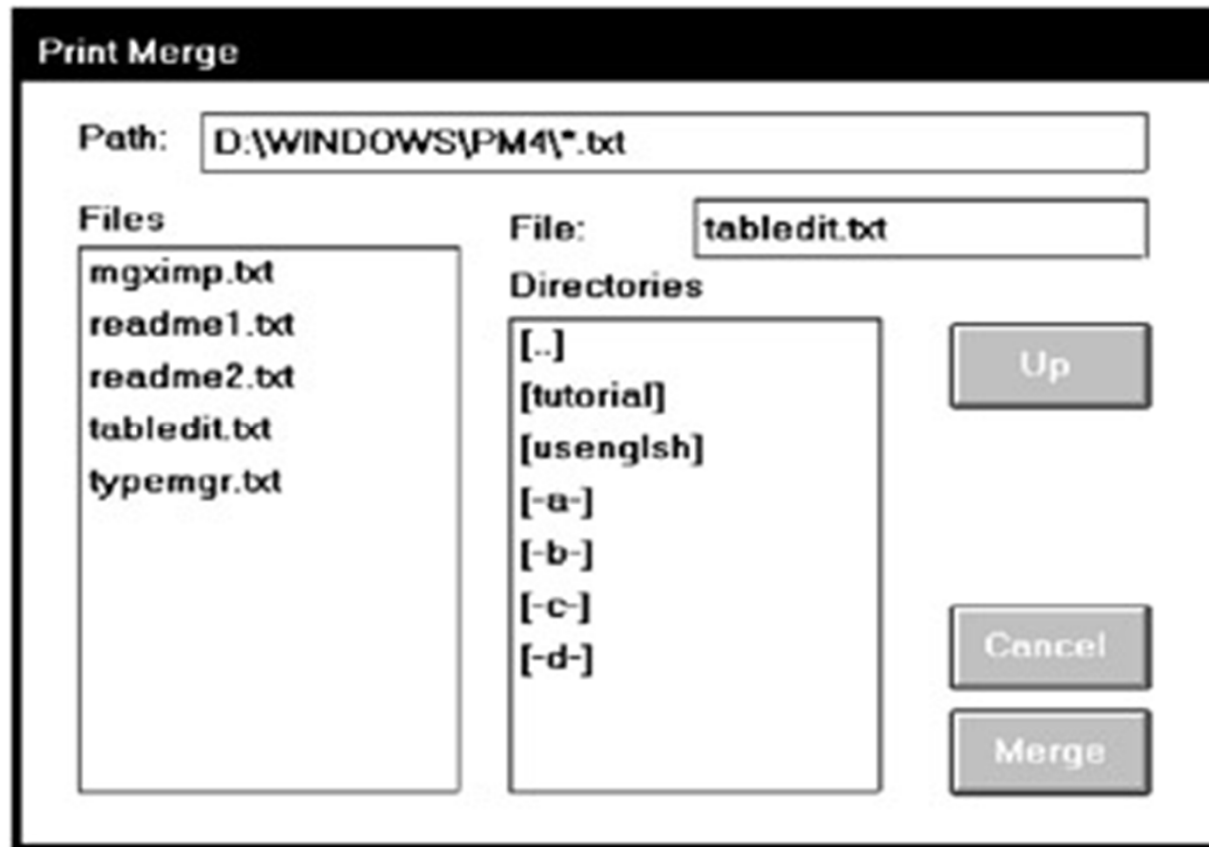
# User Interface (UI) – Ví dụ 1

The image shows a classic Windows-style dialog box titled "Text properties" with a blue title bar and a close button (X) in the top right corner. The dialog is organized into six sections, each with a title and a list of options:

- Family:** Radio buttons for Courier, Helvetica, Sans serif, and Times. "Times" is selected.
- Pitch:** Radio buttons for 10 CPI, 12 CPI, 15 CPI, and Proportional. "10 CPI" is selected.
- Border:** Radio buttons for four different line styles (thin, medium, thick, and double). The first style is selected.
- Size:** Radio buttons for Small, Medium, and Large. "Small" is selected.
- Style:** Check boxes for Bold, Italic, and Underline. All are currently unchecked.
- Color:** Radio buttons for Black, Blue, Green, and Red. "Black" is selected.

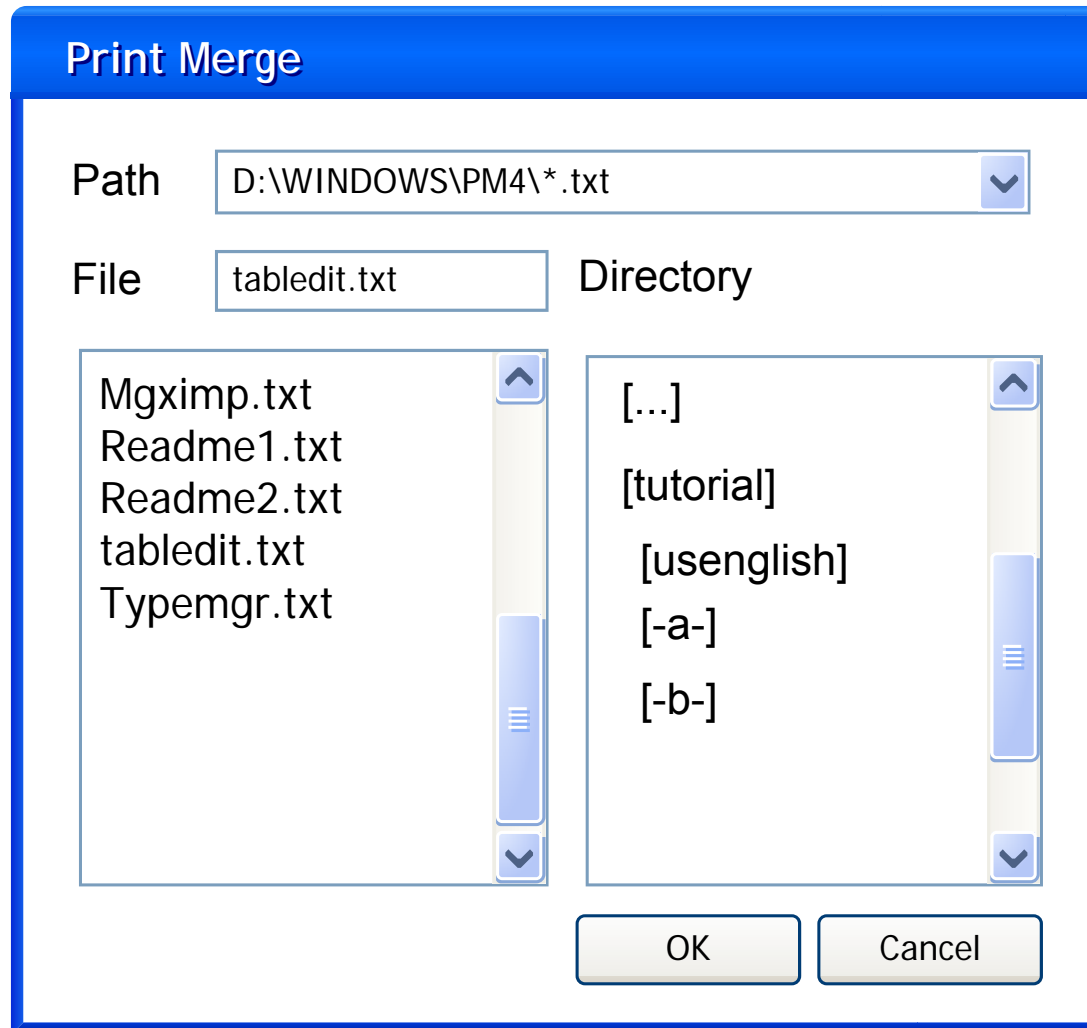
At the bottom of the dialog, there are four buttons: "OK", "Apply", "Cancel", and "Help".

## User Interface (UI) – Ví dụ 2



Các điều khiển trên màn hình Print Merge đang rất kém phù hợp. Các hộp văn bản tập tin nằm khá xa từ hộp danh sách liên quan. Nút Up làm gì? Điều này chắc chắn là không rõ ràng.

## User Interface (UI) – Ví dụ 2



## **2. Phân tích và thiết kế hướng đối tượng (Object-Oriented Analysis & Design - OOAD)**

# OOAD

- Object-Oriented Methodology (PP hướng đối tượng):
  - ✓ Cách tiếp cận phát triển hệ thống phức tạp sử dụng các khái niệm về đối tượng, lớp, đa hình, và thừa kế với tầm nhìn hướng tới việc tái sử dụng
  - ✓ Khuyến khích các kỹ sư phần mềm nghĩ đến những vấn đề về lĩnh vực ứng dụng sớm và áp dụng một cách tiếp cận phù hợp trong suốt toàn bộ vòng đời

# OOAD

- Object-Oriented Analysis & Design :
  - ✓ Mô hình phân tích yêu cầu của "thế giới thực", không phụ thuộc vào môi trường thực hiện.
  - ✓ Thiết kế áp dụng các khái niệm hướng đối tượng để phát triển, truyền tải kiến trúc và chi tiết cách làm để đáp ứng yêu cầu.

# OOAD

Các bước phân tích và thiết kế hướng đối tượng:

1. Class Modeling
2. Dynamic Modeling
3. Functional Modeling



## OOAD - Class Modeling

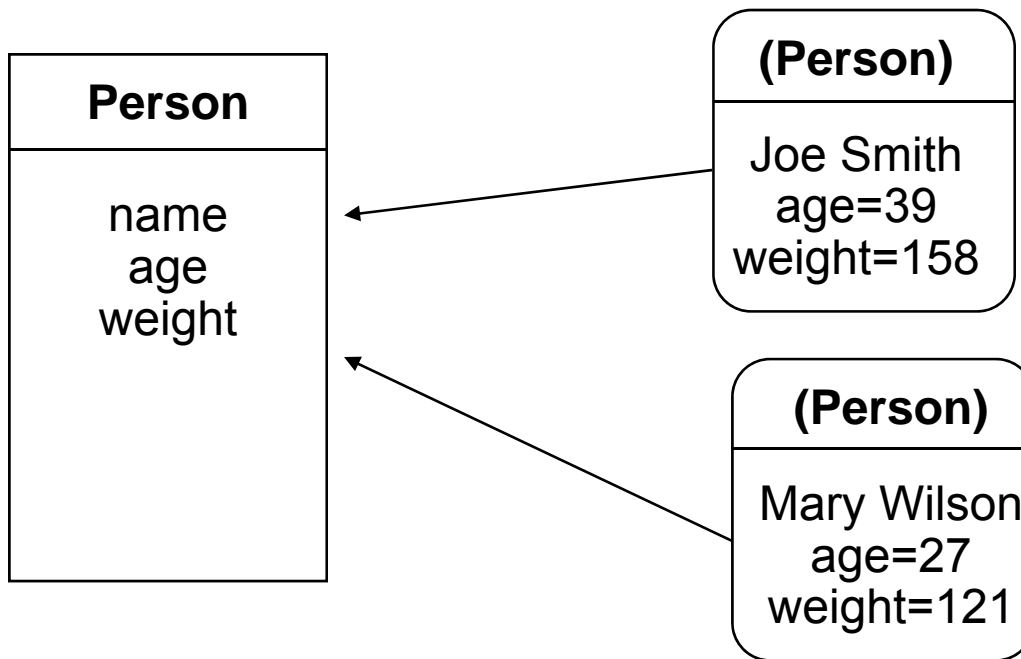
- Xác định đối tượng và các Class
- Chuẩn bị dữ liệu mẫu
- Xác định các thuộc tính đối tượng và thiết lập ban đầu
- Xác định mối liên quan giữa các đối tượng
- Tổ chức các lớp đối tượng kế thừa

## OOAD - Class Modeling

- Attribute: xác định các thuộc tính của đối tượng
  - ✓ Mỗi thể hiện của lớp có các thuộc tính giống nhau
  - ✓ Một thuộc tính có một kiểu dữ liệu
  - ✓ Các giá trị của các thuộc tính có thể khác nhau trong trường hợp
- Operation: xác định hành vi của các đối tượng
  - ✓ Hành động thực hiện trên một đối tượng
  - ✓ Có sẵn cho tất cả các thể hiện của lớp
  - ✓ Không cần phải là duy nhất trong lớp

# OOAD - Class Modeling

- Ví dụ về đối tượng Person với các thuộc tính



## OOAD - Dynamic Modeling

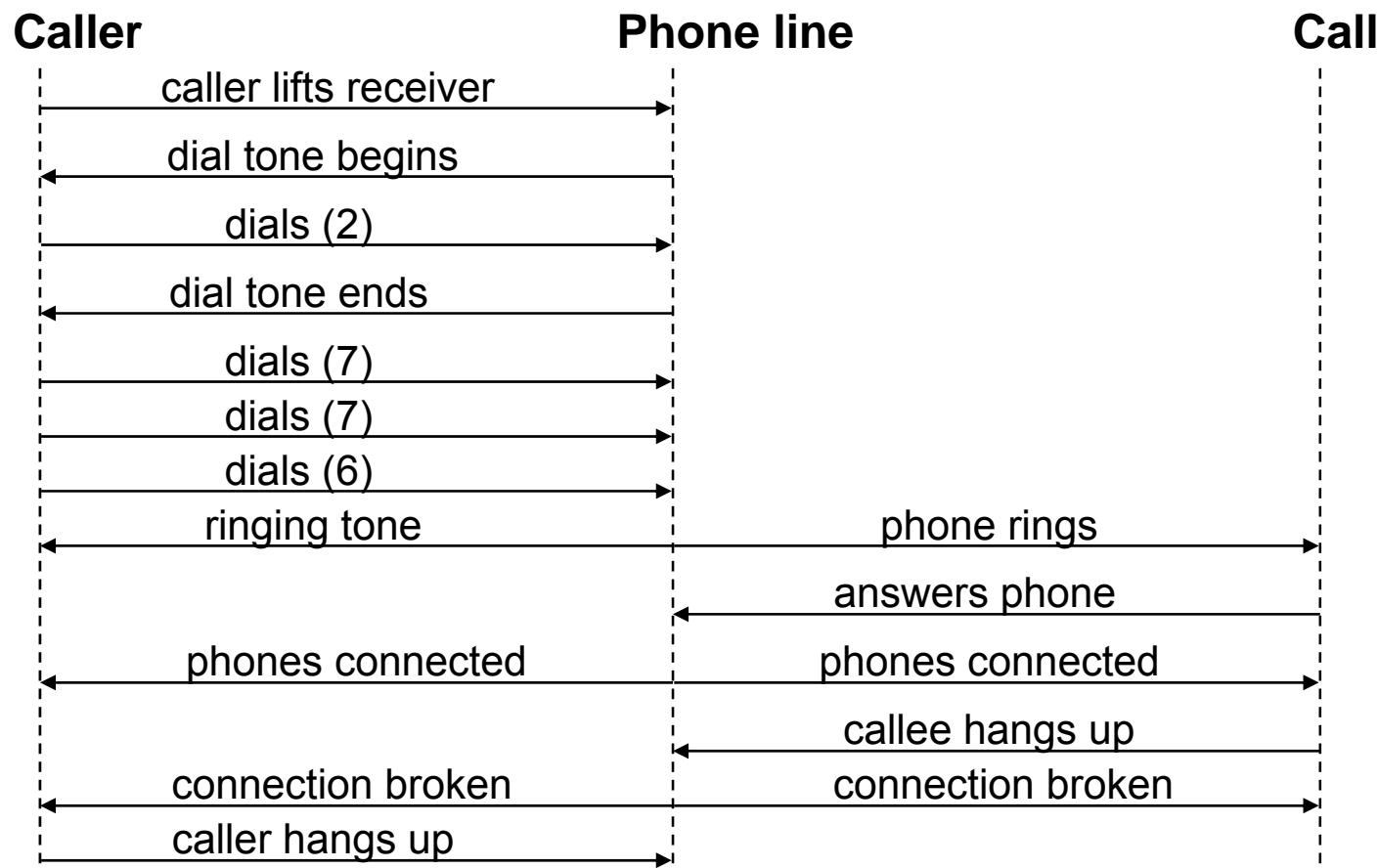
- Chuẩn bị kịch bản
- Xác định sự kiện giữa các đối tượng
- Chuẩn bị dữ liệu cho mỗi kịch bản
- Xây dựng một sơ đồ trạng thái
- Dữ liệu mẫu cho các đối tượng để xác minh tính nhất quán.

## OOAD - Dynamic Modeling

- Ví dụ về kịch bản cho **phone call**
  - ✓ caller lifts receiver
  - ✓ dial tone begins
  - ✓ caller dials digit (2)
  - ✓ caller dials digit (7)
  - ✓ caller dials digit (7)
  - ✓ caller dials digit (6)
  - ✓ specified phone rings
  - ✓ etc.

# OOAD - Dynamic Modeling

- đối tượng là những đường thẳng đứng
- sự kiện là những đường ngang
- mũi tên chỉ người gửi và người nhận
- thời gian đi từ trên xuống dưới



## OOAD - Functional Modeling - DFD

- Chức năng cấp cao nên được phân thành các đơn vị chức năng nhỏ hơn:
  - ✓ Mỗi quá trình cấp cao có thể được mở rộng thành một DFD riêng
  - ✓ Quy trình cấp cao tương ứng với các hoạt động trên các đối tượng phức tạp, trong khi quá trình cấp thấp hơn là các hoạt động trên các đối tượng cơ bản
- Phân tích sâu vào chức năng ứng dụng:
  - ✓ Ngừng lại với chức năng đơn giản
  - ✓ Mỗi cấp phải được gắn kết.

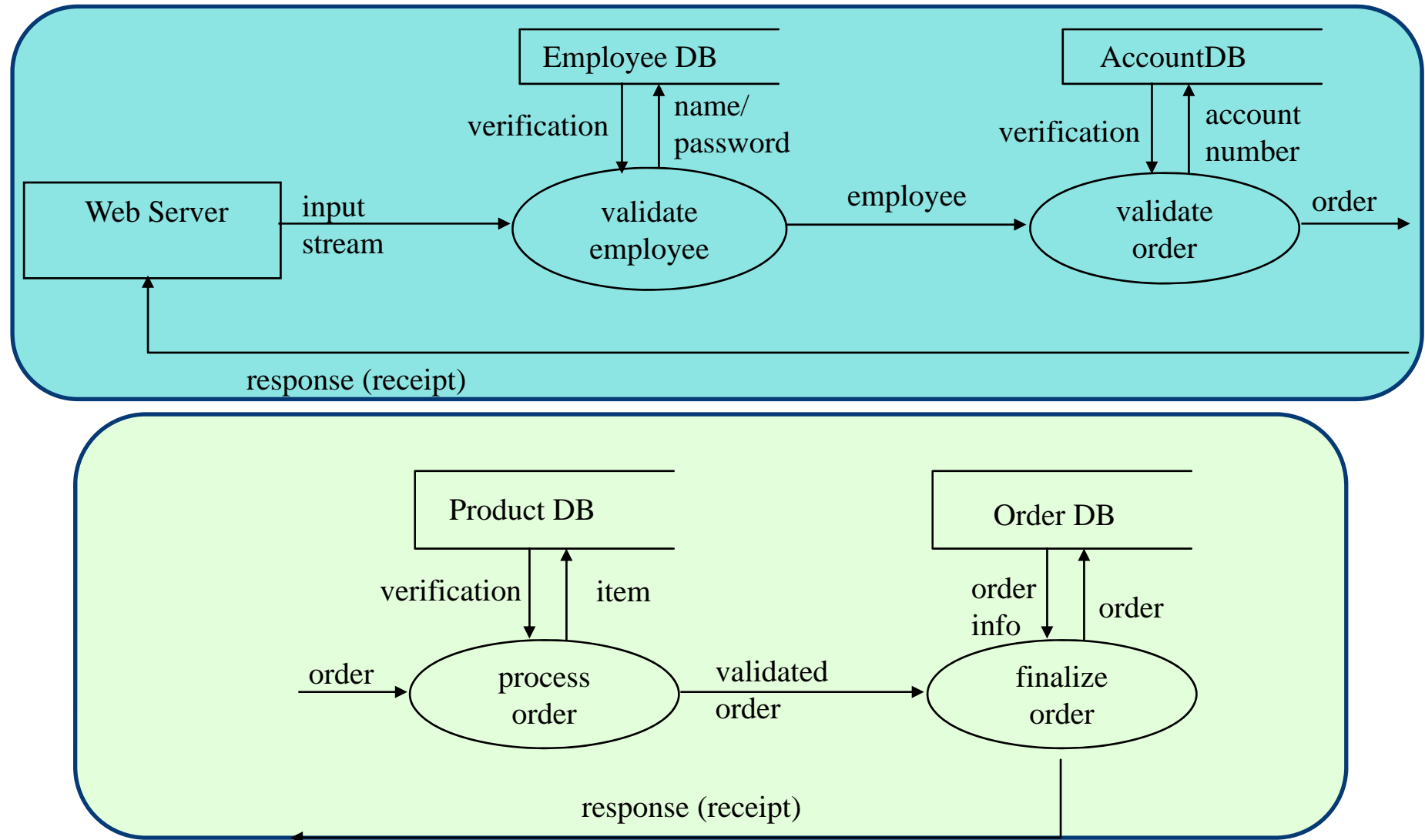
## OOAD - Functional Modeling - DFD

- Cấu trúc DFD tương ứng với những điều sau đây:
  - ✓ Sơ đồ ngữ cảnh cho thấy ranh giới của hệ thống
  - ✓ Mid – level DFD phân chia nhỏ cho thấy bối cảnh chức năng.
  - ✓ DFD nguyên thủy là những chức năng đơn giản mà không cần phải được mở rộng thêm.



# OOAD - Functional Modeling - DFD

- Ví dụ: DFD của Office supply



## Chuẩn bị Workshop 3

- Sử dụng công cụ Mockup thiết kế giao diện phần mềm dựa theo Requirement của WS1 và WS2.



# Tổng kết nội dung bài học

1. User Interface

2. Object-Oriented Analysis  
& Design

- Class Modeling
- Dynamic Modeling
- Functional Modeling





**KẾT THÚC**