Phân tích thiết kế hướng đối tượng

Bài 11: Thiết kế

TS. Nguyễn Hiếu Cường
Bộ môn CNPM, Khoa CNTT, Trường ĐH GTVT
Email: cuonggt@gmail.com

Hà Nội, 2025

Vị trí của thiết kế trong OOAD

- Khảo sát yêu cầu → Phân tích → **Thiết kế** → Cài đặt → Test
- Thiết kế gồm:
  - Thiết kế tổng thể (kiến trúc, các tầng, các lớp BCE)
  - Thiết kế chi tiết (lớp, phương thức, quan hệ giữa các lớp)
- Sơ đồ lớp thiết kế (Design Class Diagram) = phần trung tâm của thiết kế chi tiết

Khác biệt giữa Phân tích và Thiết kế

- Phân tích (Analysis) → Hệ thống cần làm gì?
- Thiết kế (Design) → Hệ thống sẽ làm như thế nào?
- Vai trò của các Sơ đồ phân tích → Thiết kế
  - Use Case: Mô tả hành vi nghiệp vụ → Thiết kế các phương thức
  - Class: Tên & vai trò nghiệp vụ → Thêm các thuộc tính, kiểu dữ liệu, các phương thức
  - Sequence: Mô tả luồng thông điệp → Tạo các phương thức thiết kế

Các chiến lược thiết kế

- Tự phát triển hoàn toàn (Custom development)
- Mua một phần (Packaged software)
- Thuê ngoài (Outsourcing)

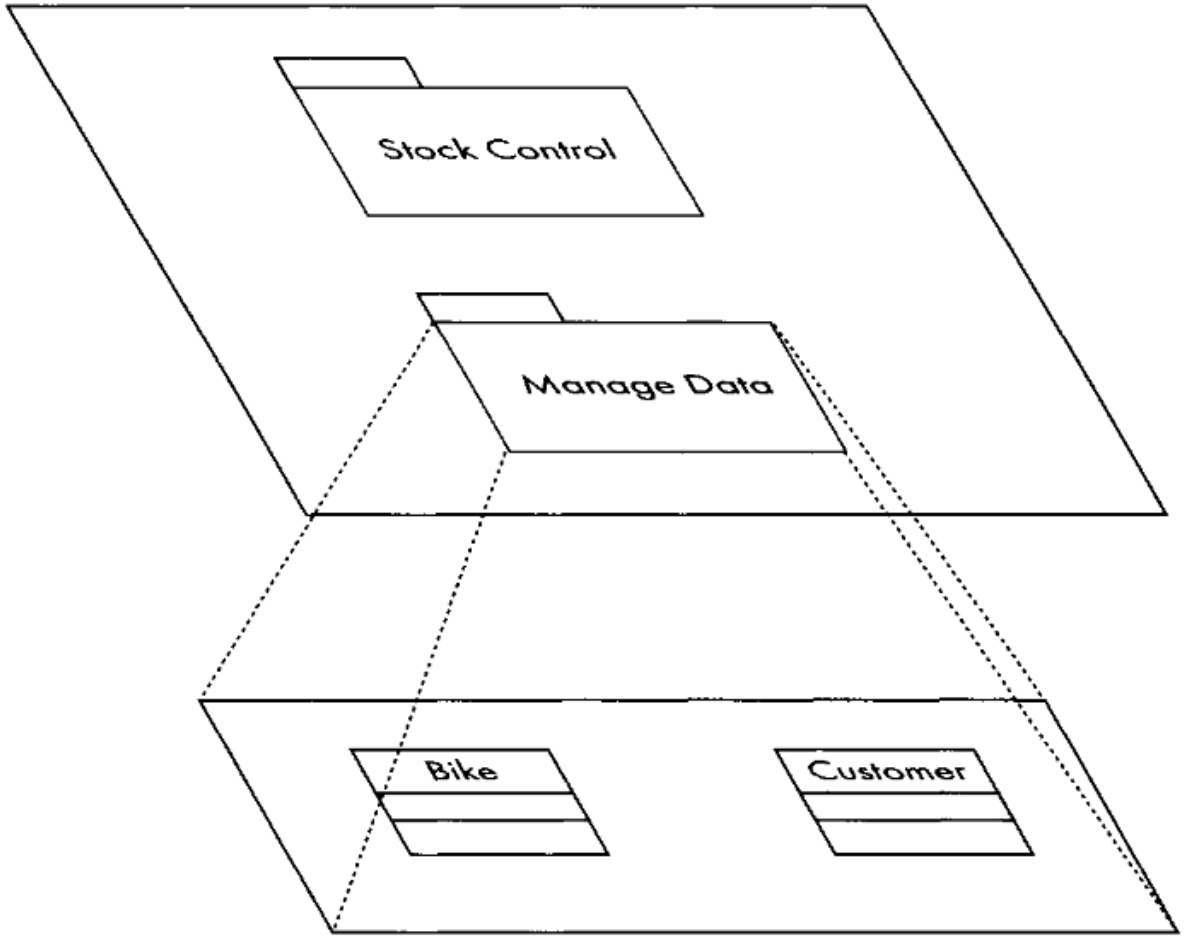
	Use Custom Development When…	Use a Packaged System When…	Use Outsourcing When…
Business Need	The business need is unique.	The business need is common.	The business need is not core to the business.
In-house Experience	In-house functional and technical experience exists.	In-house functional experience exists.	In-house functional or technical experience does not exist.
Project Skills	There is a desire to build in-house skills.	The skills are not strategic.	The decision to outsource is a strategic decision.
Project Management	The project has a highly skilled project manager and a proven methodology.	The project has a project manager who can coordinate the vendor's efforts.	The project has a highly skilled project manager at the level of the organization that matches the scope of the outsourcing deal.
Time frame	The time frame is flexible.	The time frame is short.	The time frame is short or flexible.

Phân hoạch hệ thống (partition)

- Hệ thống có thể phân hoạch thành các hệ con/ gói (package)
- Gói dùng để nhóm các phần tử mô hình
  - Các ca sử dụng (trong sơ đồ ca sử dụng)
  - Các lớp (trong sơ đồ lớp)
  - Các đối tượng (trong sơ đồ cộng tác)
- Ví dụ:
  - Phân tách hệ thống Wheels thành 2 gói, mỗi gói có các lớp liên quan:
    - Gói Stock Control gồm các lớp: Hire, Payment...
    - Gói Manage Data gồm các lớp: Bike, Customer...

Ví dụ (phân hoạch thành các gói)

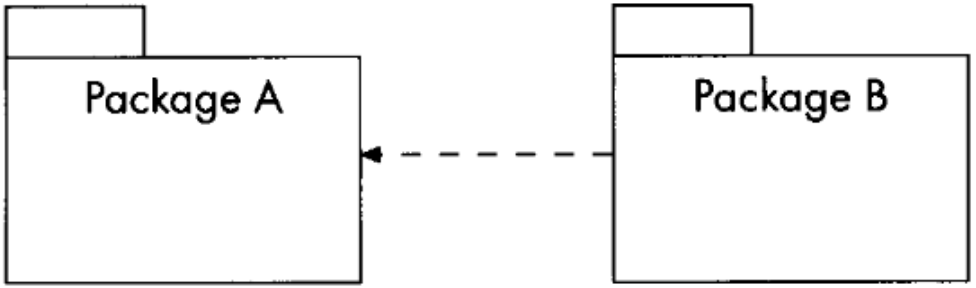
Level 1 Package diagram of subsystem



Level 2 class diagram for Manage Data subsystem

Phụ thuộc giữa các gói

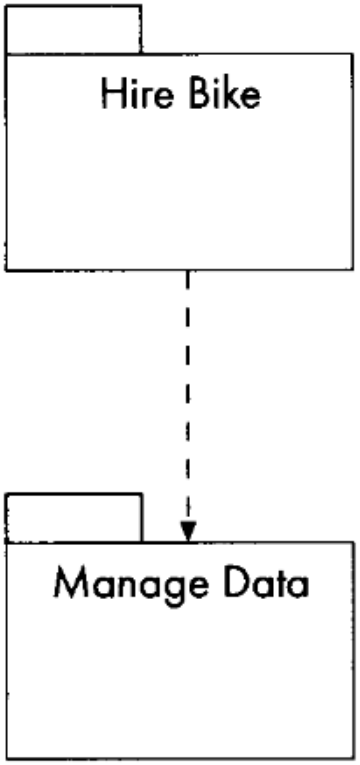
- Phụ thuộc giữa hai gói (dependency) nếu sự thay đổi ở gói này có ảnh hưởng đến gói kia
- Gói B phụ thuộc vào gói A
  - Nếu sự thay đổi ở gói A ảnh hưởng đến gói B
  - Ví dụ: Nếu có một lớp của gói B phụ thuộc vào một lớp của gói A thì gói B phụ thuộc gói A



Ví dụ (phụ thuộc)

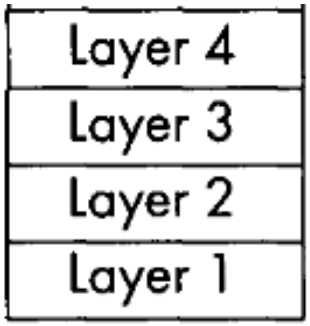
- Gói Hire Bike có các lớp: Hire và Payment
- Gói Manage Data có lớp Bike

- Nếu một số lớp trong gói Hire Bike cần sử dụng các dịch vụ của lớp Bike thì: gói Hike Bike phụ thuộc gói Manage Data



Phân tầng hệ thống (layering)

- Khi chuyển đổi sang mô hình thiết kế cần bổ sung thêm thông tin về môi trường hệ thống (data management, user interface…)
- Có thể phân chia các thành phần kiến trúc thành các tầng
- Tầng trên chỉ sử dụng các dịch vụ tầng ngay dưới cung cấp → Nếu tầng dưới thay đổi chỉ thì ảnh hưởng đến tầng trên liền kề



Ví dụ: Kiến trúc 3 tầng

Tác dụng: Phân chia trách nhiệm: UI, Logic, Data

1. **Tầng trình diễn** (Presentation layer)
   - Giao tiếp với người dùng, hiển thị dữ liệu và nhận đầu vào

     (Form đăng nhập, Trang hiển thị danh sách, ...)
2. **Tầng nghiệp vụ** (Business Logic layer/ Applicaton layer)
   - Xử lý nghiệp vụ, quyết định dữ liệu nào cần xử lý ra sao

     (Kiểm tra tài khoản, mật khẩu có đúng không, Tính chiết khấu, ...)
3. **Tầng dữ liệu** (Data Access layer/ Data layer)
   - Làm việc với CSDL, truy xuất, lưu trữ và cập nhật dữ liệu

Các loại lớp

1. **Lớp biên** (boundary class)

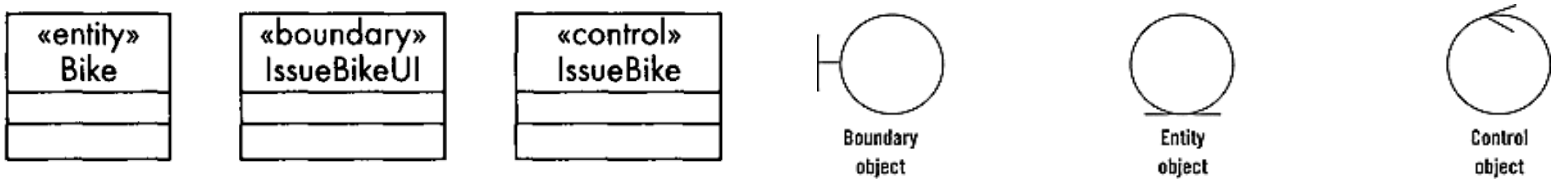
   Giao diện của hệ thống (màn hình, form, …)

2. **Lớp điều khiển** (control class)

   Thực hiện thao tác kết nối giữa các lớp biên và các lớp thực thể

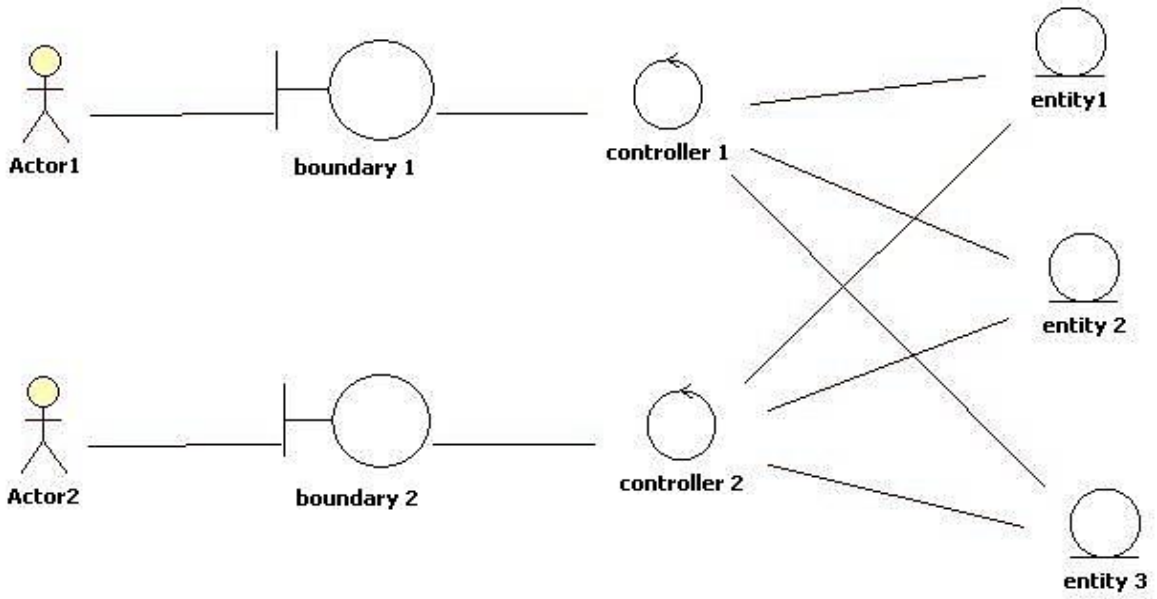
3. **Lớp thực thể** (entity class)

   Thực thể của hệ thống (Customer, Bike, Hire, …)



Quy tắc giao tiếp giữa các đối tượng

- Tác nhân <--> Đối tượng lớp biên
- Đối tượng lớp biên <--> Đối tượng lớp điều khiển
- Đối tượng lớp điều khiển <--> Đối tượng lớp thực thể



Ví dụ (Library)

- Thiết kế tổng thể
  - Boundary: BorrowForm, ReturnForm
  - Control: LoanController, ReservationController
  - Entity: Member, Book, Copy, Loan, Reservation
- Thiết kế chi tiết các lớp thì cần xác định:
  - Các thuộc tính
  - Các phương thức/ hành vi
  - Các liên kết (bao gồm cả multiplicity, navigability)

Lớp phân tích và lớp thiết kế

- **Lớp phân tích**: nhấn mạnh các khái niệm nghiệp vụ (**what**), ít hoặc không có kiểu dữ liệu, visibility
- **Lớp thiết kế**: thêm các chi tiết kỹ thuật (**how**): kiểu dữ liệu, visibility, exceptions, interfaces, ...
- Ví dụ:

  Loan (analysis) →

  Loan {loanId: Long, borrowDate: Date, dueDate: Date,

  createLoan(member, copy): Loan} (design)

Các lớp phân tích

- Mục tiêu chính của phân tích: hiểu domain và yêu cầu nghiệp vụ

  → Các lớp thực thể (Entity classes) → Sơ đồ lớp phân tích

  - Ví dụ (Library System): Member, Book, Copy, Loan, Reservation
- Trong khi phân tích, có thể phác thảo lớp Boundary và Control, nhưng chưa cần chi tiết
  - Boundary: BorrowForm (đại diện cho tương tác người dùng)
  - Control: LoanController (chỉ để nhắc đến ai điều phối)

Boundary ở phân tích → thiết kế

Phân tích:

- Boundary chỉ là khái niệm giao tiếp giữa actor và hệ thống
  - Ví dụ: BorrowForm, ReturnForm, LoginForm
- Không cần có chi tiết kỹ thuật, chỉ biết: "Người dùng nhập vào đây, hệ thống phản hồi ra đó."

Thiết kế:

- Boundary được cụ thể hóa thành loại giao diện / API thực tế:
  - Form GUI (dialog box, web page, mobile screen)
  - REST API endpoint (/borrow, /return)

Ví dụ (Library System)

- Phân tích: <<boundary>> BorrowForm
- Thiết kế: <<boundary>> BorrowForm
  - Attributes:

    - inputFields: List<String>
  - Methods:

    + showForm(),

    + getInput(): MemberId,

    + displayError(msg: String)

Control ở phân tích → thiết kế

- **Phân tích**:
  - Control chỉ là "lớp trung gian" để điều phối use case
  - Chưa nói chi tiết về method, transaction, exception
- **Thiết kế**:
  - Control trở thành service/controller class thực sự:
  - Có method tương ứng với các bước trong use case
  - Có xử lý logic: gọi Entity, kiểm tra business rules, ...

Ví dụ (Library System)

- Phân tích: <<control>> LoanController (khái niệm)
- Thiết kế: <<control>>LoanController
  - Methods:

    +borrowBook(memberId: int, copyId: int): Loan

    +returnBook(loanId: int): void
  - Control gọi:
    - Loan.createLoan(),
    - Copy.checkAvailability(),
    - Loan.closeLoan()

Tóm tắt

- Yêu cầu → **Use Case Diagram** (mô tả hành vi)
- Phân tích → **Analysis Class Diagram** (khái niệm nghiệp vụ)
- **Sequence Diagram** → Xác định thông điệp giữa các đối tượng
- Thiết kế → **Design Class Diagram** (có thuộc tính, method, kiểu dữ liệu)

- Cài đặt → Code (C++/C#/Java...) dựa trên các lớp thiết kế

Thiết kế giao diện

- User interface (UI): Giao diện người dùng là tập các màn hình và các phần tử thể hiện khác (nút lệnh, biểu tượng…) mà người dùng tương tác (với các thiết bị, phần mềm…)
- User experience (UX): Trải nghiệm người dùng là những cảm nhận và kinh nghiệm mà người dùng có được khi tương tác

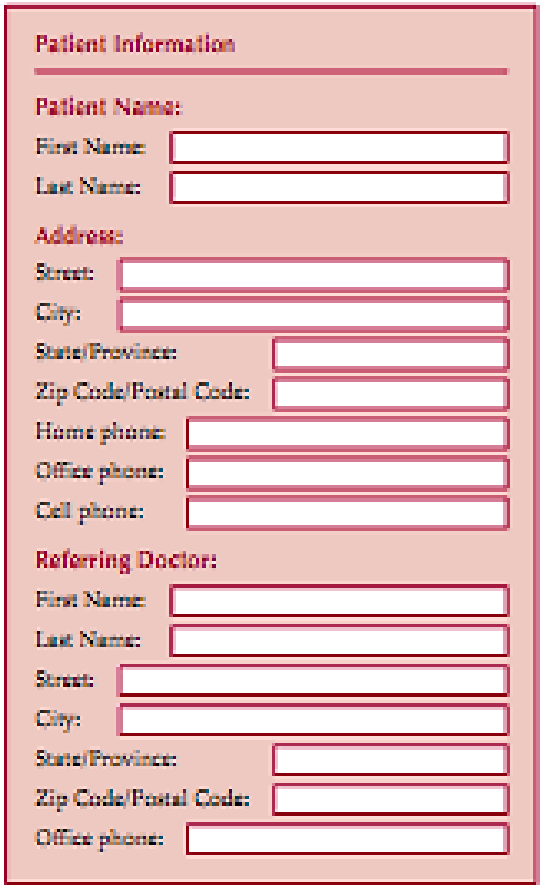
Một số nguyên tắc chung về giao diện

- Bố cục (Layout): Giao diện có thể gồm các vùng, sử dụng với các mục đích khác nhau
- Nhận thức (Content awareness): Người dùng luôn phải biết được mình đang ở đâu và thông tin gì đang hiển thị
- Tính thẩm mỹ (Aesthetics): Font chữ, màu sắc, khoảng trống...
- Trải nghiệm người dùng (User experience): Dễ học, dễ dùng, thuận tiện, nỗ lực tối thiểu (minimal user effort)
- Nhất quán (Consistence)

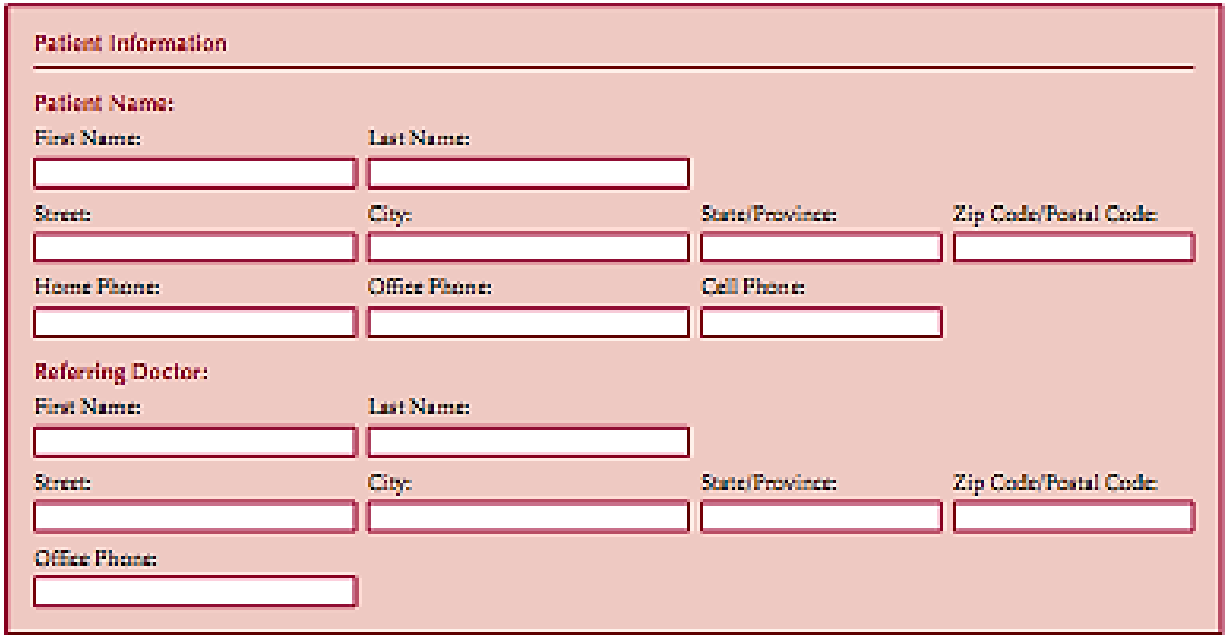
Tóm tắt các nguyên tắc về giao diện

Principle	Description
Layout	The interface should be a series of areas on the screen that are used consistently for different purposes—for example, a top area for commands and navigation, a middle area for information to be input or output, and a bottom area for status information.
Content Awareness	Users should always be aware of where they are in the system and what information is being displayed.
Aesthetics	Interfaces should be functional and inviting to users through careful use of white space, colors, and fonts. There is often a trade-off between including enough white space to make the interface look pleasing without losing so much space that important information does not fit on the screen.
User Experience	Although ease of use and ease of learning often lead to similar design decisions, sometimes there is a trade-off between the two. Novice or infrequent users of software prefer ease of learning, whereas frequent users prefer ease of use.
Consistency	Consistency in interface design enables users to predict what will happen before they perform a function. It is one of the most important elements in ease of learning, ease of use, and aesthetics.
Minimal User Effort	The interface should be simple to use. Most designers plan on having no more than three mouse clicks from the starting menu until users perform work.

Bố cục (luồng dọc và luồng ngang)

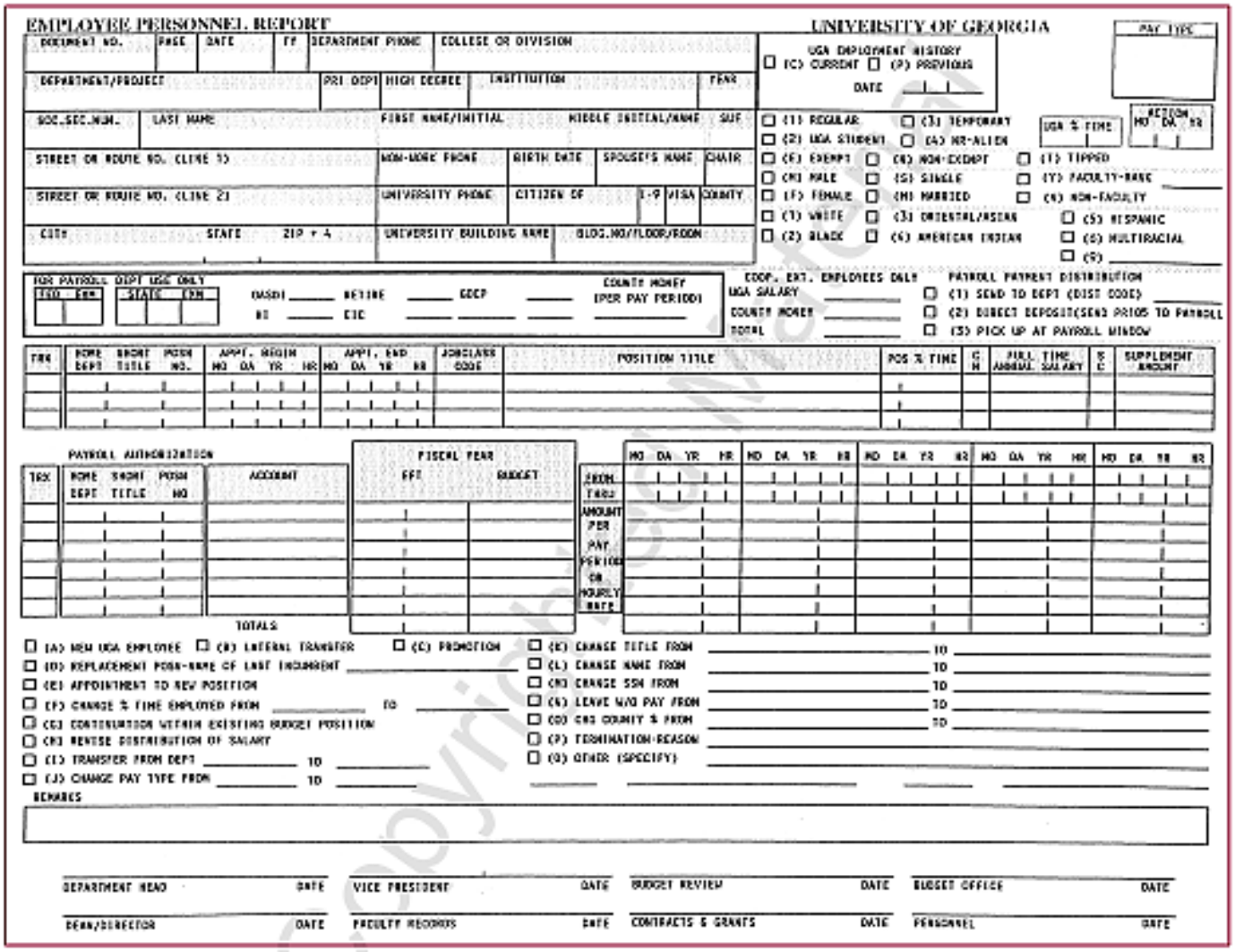


(A) Vertical Flow



(B) Horizontal Flow

Nhận xét gì về giao diện sau?



EMPLOYEE PERSONNEL REPORT

UNIVERSITY OF GEORGIA

DOCUMENT NO. | PAGE | DATE | FY | DEPARTMENT PHONE | COLLEGE OR DIVISION

UGA DEPLOYMENT HISTORY
(C) CURRENT (P) PREVIOUS
DATE

PAY TYPE

DEPARTMENT/PROJECT | PRI DEPT | HIGH DEGREE | INSTITUTION | YEAR

SOC.SEC.NUM. | LAST NAME | FIRST NAME/INITIAL | MIDDLE INITIAL/NAME | SUF

(1) REGULAR (3) TEMPORARY
(2) UGA STUDENT (4) NR-ALIEN

UGA % TIME

ACTION MO DA YR

STREET OR ROUTE NO. (LINE 1) | NON-WORK PHONE | BIRTH DATE | SPOUSE'S NAME | CHAIR

(E) EXEMPT (6) NON-EXEMPT (T) TIPPED

STREET OR ROUTE NO. (LINE 2) | UNIVERSITY PHONE | CITIZEN OF | I-9 | VISA | COUNTY

(M) MALE (S) SINGLE (Y) FACULTY-RANK
(F) FEMALE (M) MARRIED (N) NON-FACULTY

CITY | STATE | ZIP + 4 | UNIVERSITY BUILDING NAME | BLDG.NO/FLOOR/ROOM

(1) WHITE (3) ORIENTAL/ASIAN (5) HISPANIC
(2) BLACK (4) AMERICAN INDIAN (6) MULTIRACIAL
(9)

FOR PAYROLL DEPT USE ONLY
FED | FICA | STATE | FICA

OASDI ____ RETIRE ____ GDCP ____ COUNTY MONEY (PER PAY PERIOD)
HI ____ EIC ____ ____ ____ ____

COOP. EXT. EMPLOYEES ONLY
UGA SALARY ____
COUNTY MONEY ____
TOTAL ____

PAYROLL PAYMENT DISTRIBUTION
(1) SEND TO DEPT (DIST CODE) ____
(2) DIRECT DEPOSIT(SEND PRIOR TO PAYROLL
(3) PICK UP AT PAYROLL WINDOW

TRX	HOME DEPT	SHORT TITLE	POSN NO.	APPT. BEGIN MO DA YR HR	APPT. END MO DA YR HR	JOBCLASS CODE	POSITION TITLE	POS % TIME	G H	FULL TIME ANNUAL SALARY	S C	SUPPLEMENT AMOUNT

PAYROLL AUTHORIZATION

TRX	HOME DEPT	SHORT TITLE	POSN NO	ACCOUNT	FISCAL YEAR EFF	FISCAL YEAR BUDGET	FROM / THRU	MO DA YR HR	MO DA YR HR	MO DA YR HR	MO DA YR HR	MO DA YR HR
							AMOUNT PER PAY PERIOD OR HOURLY RATE					
			TOTALS									

(A) NEW UGA EMPLOYEE (B) LATERAL TRANSFER (C) PROMOTION
(D) REPLACEMENT POSN-NAME OF LAST INCUMBENT ____
(E) APPOINTMENT TO NEW POSITION
(F) CHANGE % TIME EMPLOYED FROM ____ TO ____
(G) CONTINUATION WITHIN EXISTING BUDGET POSITION
(H) REVISE DISTRIBUTION OF SALARY
(I) TRANSFER FROM DEPT ____ TO ____
(J) CHANGE PAY TYPE FROM ____ TO ____

(K) CHANGE TITLE FROM ____ TO ____
(L) CHANGE NAME FROM ____ TO ____
(M) CHANGE SSN FROM ____ TO ____
(N) LEAVE W/O PAY FROM ____ TO ____
(O) CHG COUNTY % FROM ____ TO ____
(P) TERMINATION-REASON ____
(Q) OTHER (SPECIFY) ____

REMARKS

DEPARTMENT HEAD DATE | VICE PRESIDENT DATE | BUDGET REVIEW DATE | BUDGET OFFICE DATE

DEAN/DIRECTOR DATE | FACULTY RECORDS DATE | CONTRACTS & GRANTS DATE | PERSONNEL DATE

Độ dày

- Mật độ thông tin
  - Tỷ lệ thông tin trên một diện tích
  - Mật độ cao (như ví dụ trên): nhiều thông tin, khó nhìn
  - Mật độ thấp: dễ nhìn, nhưng mất nhiều thao tác dịch chuyển
- Tùy thuộc trình độ của người dùng
  - Người dùng mới: ít thông tin, mật độ không nên quá 50%
  - Người dùng có kinh nghiệm: mật độ thông tin có thể chiếm đến 90%

Câu hỏi

1. Trong Design Class Diagram, một lớp được biểu diễn như thế nào (gồm những gì)?
2. Hãy phân biệt ngắn gọn Analysis Class Diagram và Design Class Diagram.
3. Trong BCE, vai trò của Control class khác gì so với Entity class?
4. Khi nào nên dùng lớp kết hợp (Association Class) thay vì chỉ dùng association thông thường? Cho ví dụ trong Library System.
5. Giả sử trong Library System có use case "Reserve Book". Hãy mô tả cách đi từ Sequence Diagram sang Design Class Diagram.

Bài tập

1. Có hai gói Stock và Ordering, trong đó mỗi Product có giá, mỗi OrderLine có thuộc tính số lượng được liên kết với một Product. Mỗi OrderLine phải thực hiện được bằng cách hỏi giá từ Product tương ứng để nhân với số lượng. Hãy xác định sự phụ thuộc giữa hai gói.

Stock package	*Ordering package*
Product	Customer
Supplier	Order
	OrderLine

Bài tập

2. Khi khách hàng trả xe, Annie cần tìm chi tiết của việc thuê xe đó. Hãy thiết kế màn hình hiển thị: tên khách hàng, số xe, nhà sản xuất, model, loại xe (nam, nữ) và kích thước; ngày bắt đầu thuê, số ngày thuê, số tiền đặt cọc và phí thuê mỗi ngày.