Vietnam National University of HCMC
International University
School of Computer Science and Engineering

Object – Oriented Analysis and Design

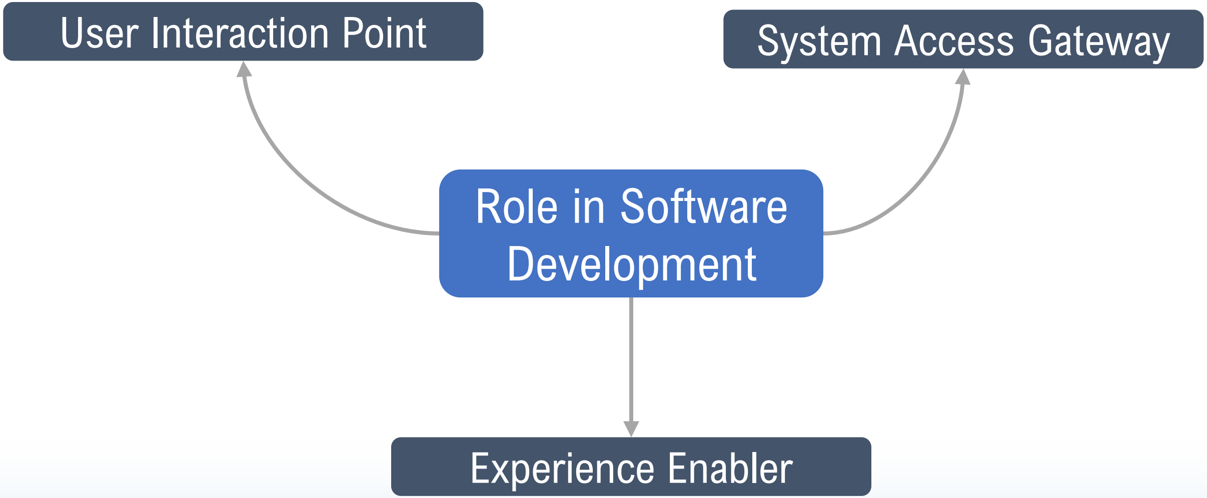
User Interface

Instructor: Le Thi Ngoc Hanh, Ph.D

ltnhanh@hcmiu.edu.vn

User Interface (UI) Design

- UI is the space where users interact with a system through visual, auditory, and interactive components.



Why UI is important?

- Architectural layer integration
- Separation of concerns
- Design impact

Principles of UI Design

User Interface design principles ensure that systems are intuitive, efficient, and enjoyable to use.

- Consistency between internal and external
- Feedback and visibility
- Error prevention and recovery
- User control and freedom
- Aesthetic and minimalistic design

Types of UIs

User interfaces come in various types, each designed to facilitate interaction between users and systems in specific contexts.

- Command Line Interface (CLI)
- Graphical User Interface (GUI)
- Touch-based Interfaces
- Voice-based Interfaces

UI Design Process

- Requirements gathering (user personas, tasks)
- Prototyping and wireframing
- UI design tools overview (Figma, Adobe XD, etc.)
- Usability testing

UI Design in OOAD

- How UI connects with system architecture components.
- Mapping UI elements to system components using UML diagrams.

Role of UI in OOAD

- **Presentation Layer:** Represents the user interface where data is displayed and inputs are captured.
- **Application Layer:** Contains business logic that processes user inputs.
- **Data Layer:** Manages persistent storage and data retrieval.

Mapping UI Elements to System Components

- **Use Case Diagrams:** Show how users interact with the system.
- **Class Diagrams:** Define classes responsible for handling UI elements.
- **Sequence Diagrams:** Illustrate the flow of interactions between UI components and backend systems.

Example

Online Food Ordering System:

- **UI layer:** Login form, menu screen, and checkout page.
- **Application layer:**
  - OrderManager class processes orders.
  - PaymentProcessor class handles transactions.
- **Data layer:** Stores user details, orders, and payment history.

Evaluate UI based on OOAD principles

- **Check architectural compliance:** Ensure the UI design adheres to layered system architecture.
- **Evaluate design patterns:** Look for relevant design patterns applied in UI development.
- **Assess reusability and modularity:** Ensure the UI components are reusable and maintainable.
- **Consider user experience:** Focus on usability, accessibility, and responsiveness.
- **Review scalability:** Ensure the system supports future expansion with minimal changes.

Sample judgment criteria

Goal: modularity and separation of concerns:

- The UI layer only handles presentation.
- Business logic resides in the application layer.
- Data storage is managed by the data layer.

Sample judgment criteria

Goal: encapsulation and information hiding:

- UI elements expose only relevant features.
- Backend implementation changes should not affect UI design.

Sample judgment criteria

Goal: Reusability

- Reusable components like buttons, forms, and menus.
- Consistent styles applied using design patterns.

Sample judgment criteria

Goal: Abstraction

- Abstract UI components reduce complexity.
- Developers can create new features by extending existing components.

Sample judgment criteria

Goal: Usability and user-center design

- Simple and consistent navigation.
- Accessible features and readable text.
- Proper use of feedback and error messages.

Sample judgment criteria

Goal: Scalability and maintainability

- Adding new features should require minimal UI restructuring.
- Modular components support easier updates and maintenance.