

Lecture 9: Object-Oriented Analysis and Design

UML – Analysis

Agenda

- OO Analysis and Design
- Modelling
- Unified Modelling Language (UML)
- Views of UML
- Use-Case Modelling



OO Analysis and Design Approach

- ▶ **Object-Oriented Analysis (OOA)** is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain. [Booch]
- ▶ **Object-Oriented Design (OOD)** is a method of design that encompasses the process of object-oriented decomposition and a notation for depicting both logical (class and object structure) and physical (process and data flow) models of a system as well as static and dynamic aspects of a system. [Booch]

Modelling (1)

- ▶ A model is an abstraction describing system or a subset of a system
- ▶ A view depicts selected aspects of a model
- ▶ A notation is a set of graphical or textual rules for representing views
- ▶ Views and models of a single system may overlap each other

Modelling (2)

- ▶ **Modelling languages form an important part of software development since:**
 - ▶ Provides an exact specification for developers
 - ▶ The blueprint allows the project managers to estimate cost with greater accuracy
 - ▶ A good method of communication between technical and non-technical users
 - ▶ Allows developers to understand the user requirement

Unified Modelling Language (UML)

- ▶ **The Unified Modeling Language (UML) – is the most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure.**
- ▶ UML is a nonproprietary language.
- ▶ It addresses the needs of various kinds of software development.
- ▶ The goals of the UML are:
 - ▶ To provide a simple and ready-to-use expressive visual modeling language
 - ▶ To be process independent
 - ▶ To be language independent

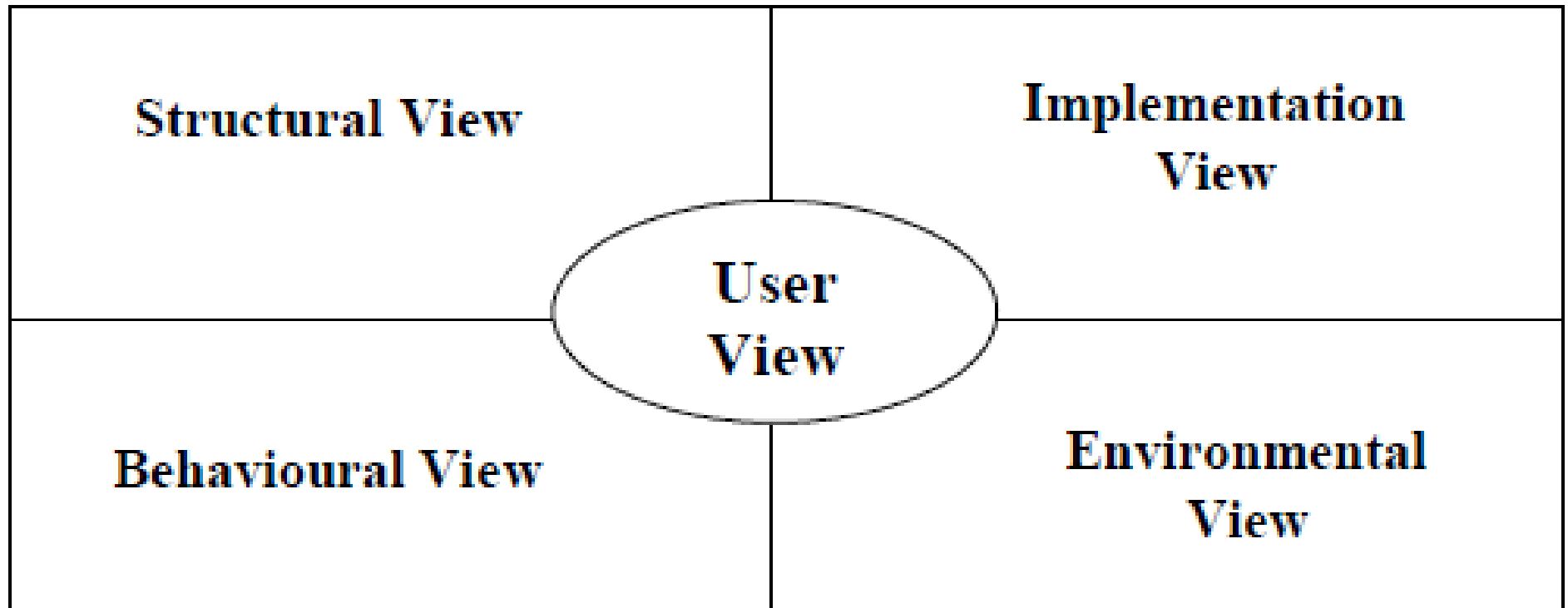
What is UML used to represent?

- ▶ Visualize the software system with well-defined symbols.
- ▶ Specify the software system and help in building precise, unambiguous and complete models
- ▶ Construct models of the software system that can directly communicate with a variety of programming languages
- ▶ Document models of the software system by expressing requirements of the system during its development and deployment stages

Views in UML

- ▶ The **user view** – represents the goals and objectives of the system from a user's viewpoint
- ▶ The **structural view** – represents the static or idle state of the system
- ▶ The **behavioral view** – represents the dynamic or changing state of the system
- ▶ The **implementation view** – represents the distribution of the logical elements of the system
- ▶ The **environment view** – represents the distribution of the physical elements of the system

Views in UML



Types of UML Diagrams

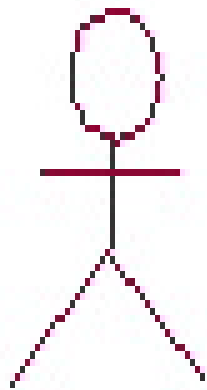
- ▶ **Use case diagram-** for gaining a system overview from a user perspective
- ▶ **Class diagram-** for defining system objects and their relationships
- ▶ **Sequence diagram-** for illustrating the sequence of events and changes over time
- ▶ **Component diagram-** for defining software system components e.g files, tables
- ▶ **Deployment diagram-** show the physical layout of components on hardware nodes.
- ▶ **Collaboration diagrams-** show how the elements of the system being developed work together to achieve the system's purpose

Use Case (1)

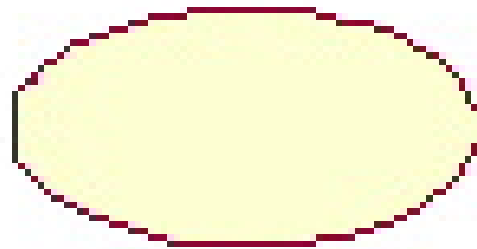
- ▶ A use case is a set of scenarios that describing an interaction between a user and a system.
- ▶ Offer a dynamic view of the system
- ▶ A use case diagram displays the relationship among actors and use cases.
- ▶ The use-case diagram is used to depict the functionality of a system as it appears to external users of the system.

Use Case (2)

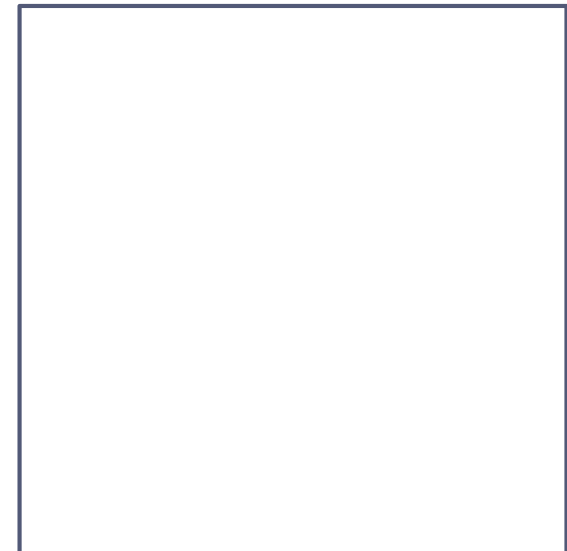
- Components of a Use case diagram



Actor



Use Case



System
Boundary

Use Case (3)

▶ **Actors**

- ▶ Actors are elements that lie outside the system and thus are called external entities of a system.
- ▶ Actors interacts directly with the system, exchange messages, and
- ▶ request for some action to be performed.

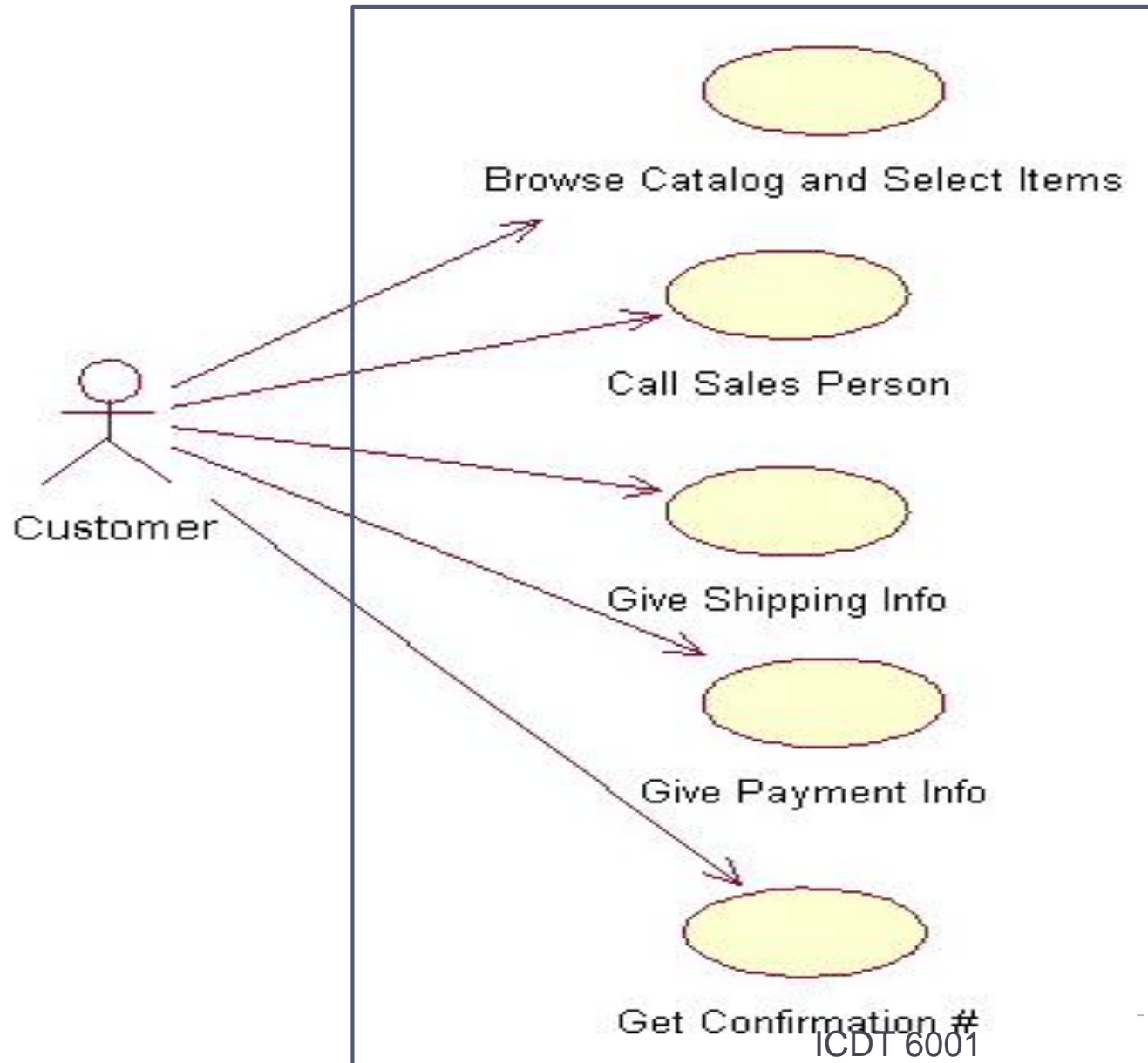
▶ **Use Cases**

- ▶ Use cases consist of elements that lie inside the system and are responsible for the working, that is, the functionality and behavior of the system.

Example 1: Customer Order System

- ▶ A user placing an order with a sales company might follow these steps.
 - ▶ Browse catalog and select items.
 - ▶ Call sales representative.
 - ▶ Supply shipping information.
 - ▶ Supply payment information.
 - ▶ Receive conformation number from salesperson.

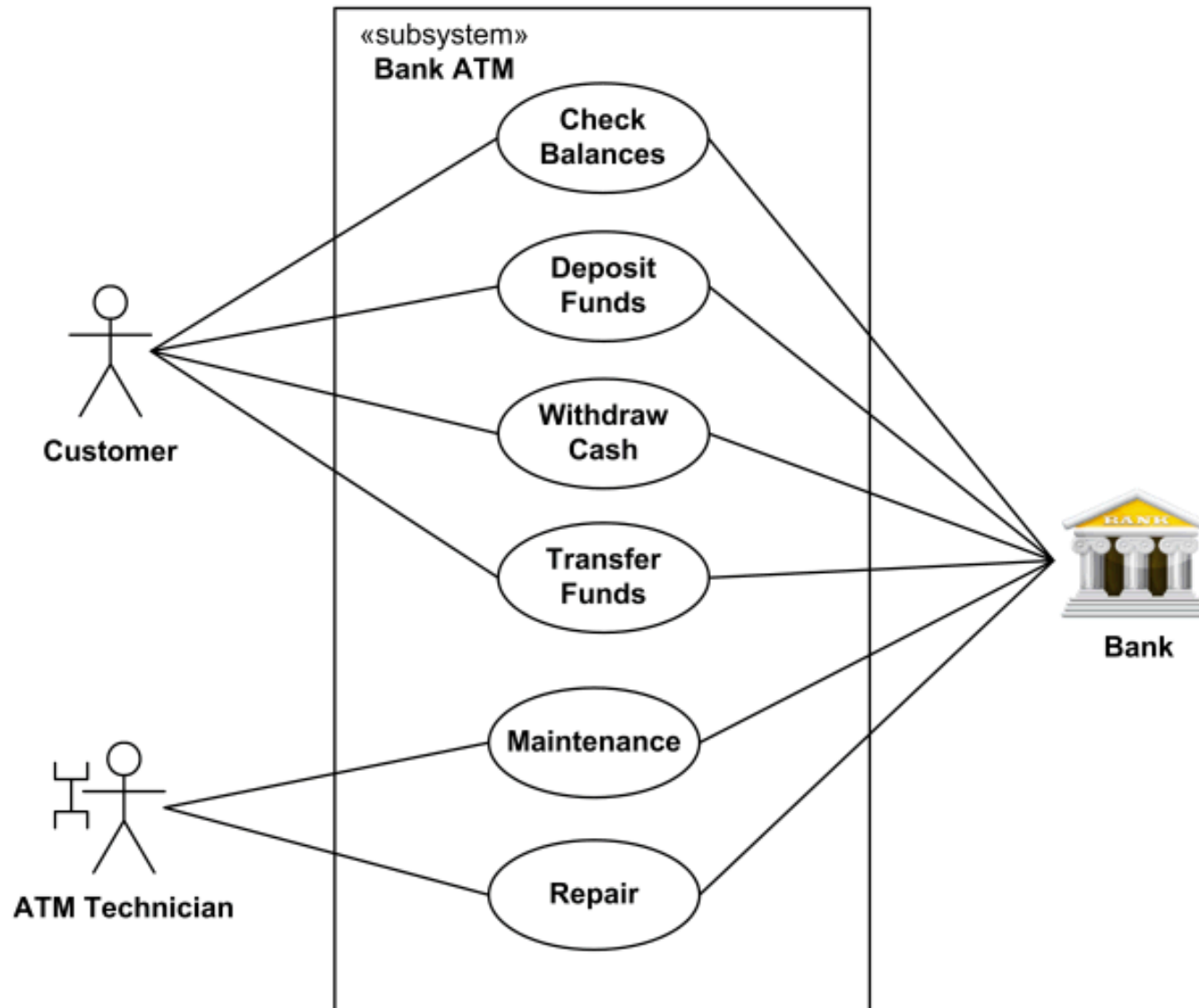
Example 1: Customer Order System



Example 2: Bank ATM

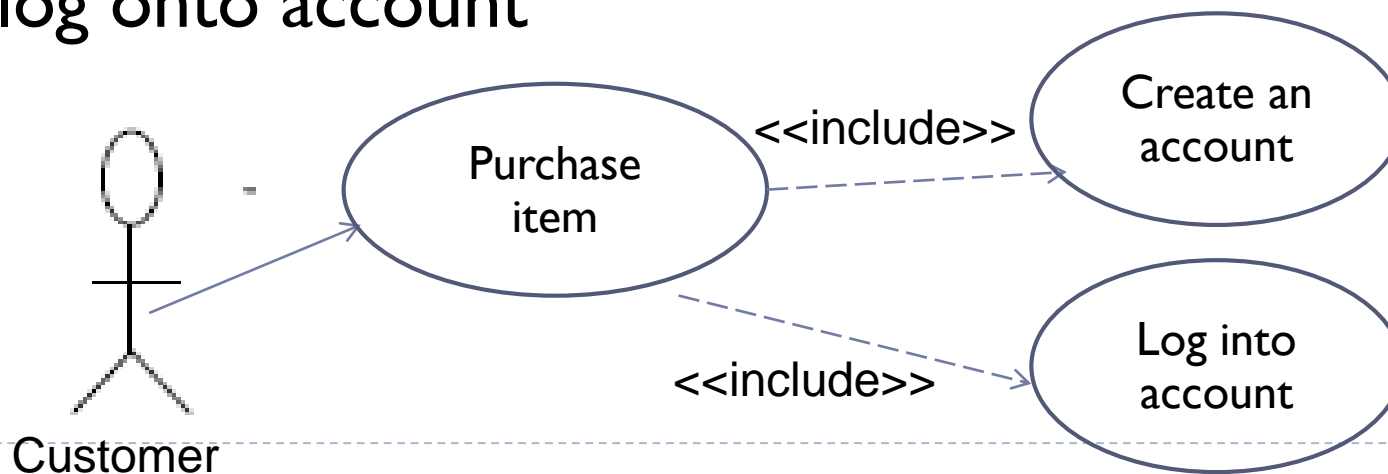
- ▶ **Purpose:** Describe use cases that an automated teller machine (ATM) or the automatic banking machine (ABM) provides to the bank customers.
- ▶ **Summary:** Customer uses a bank ATM to check balances of his/her bank accounts, deposit funds, withdraw cash and/or transfer funds (use cases). ATM Technician provides maintenance and repairs to the ATM.

Example 2: Bank ATM



<<include>> Relationship

- ▶ A base use case is dependent on the included use case(s); without it/them the base use case is incomplete as the included use case(s) represent sub-sequences of the interaction
- ▶ An <<include>> relationship represents behavior that is factored out of the use case.
- ▶ E.g. Customer purchase an item after creating an account and log onto account

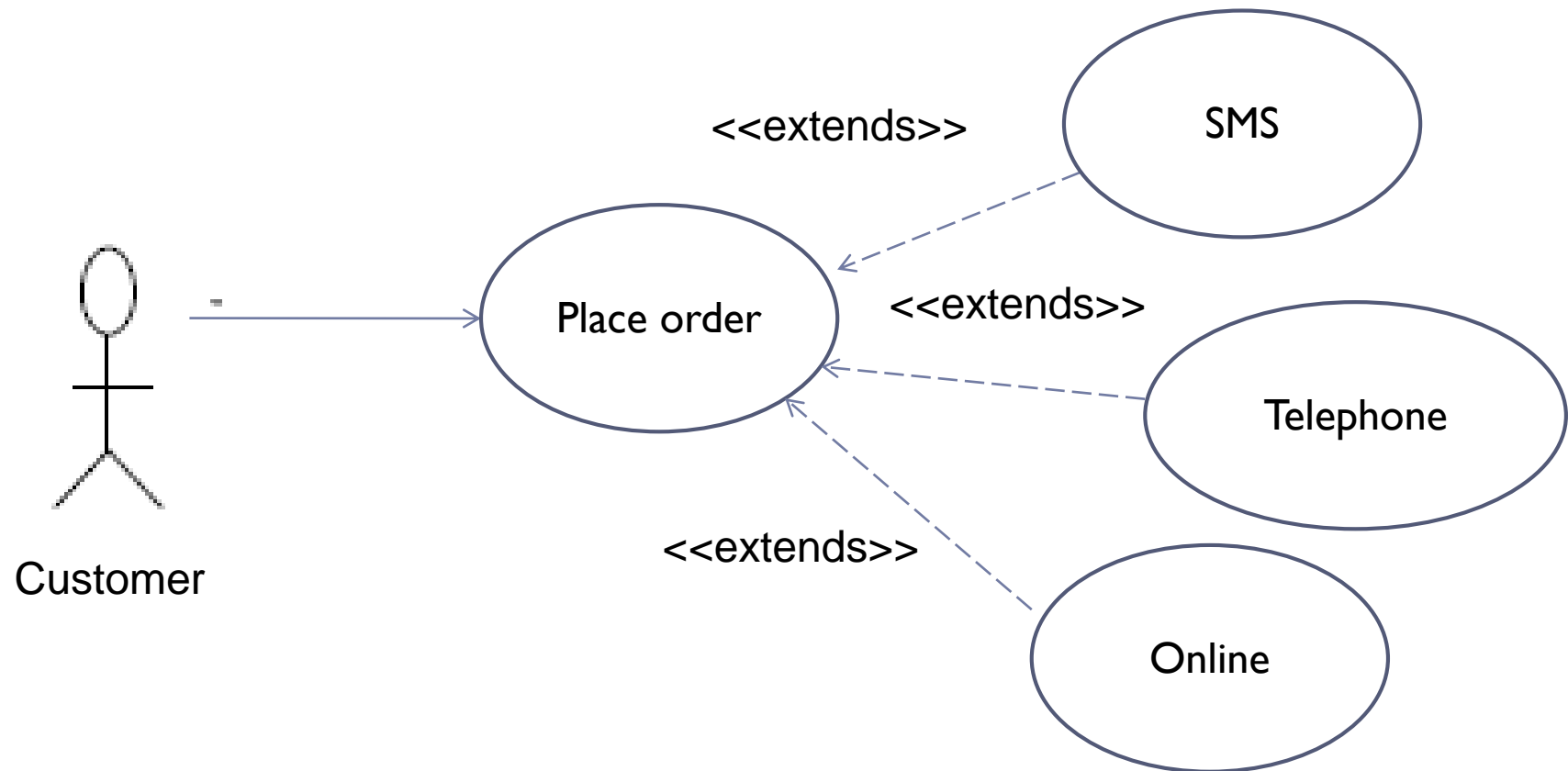


<<Extends>>(1)

- ▶ <<extend>> relationships represent exceptional or seldom invoked cases.
- ▶ You can add extend relationships to a model to show the following situations:
- ▶ A part of a use case that is optional system behavior
- ▶ A subflow is executed only under certain conditions

<<Extends>>(2)

- ▶ E.g. A customer place order by telephone, sms, online.

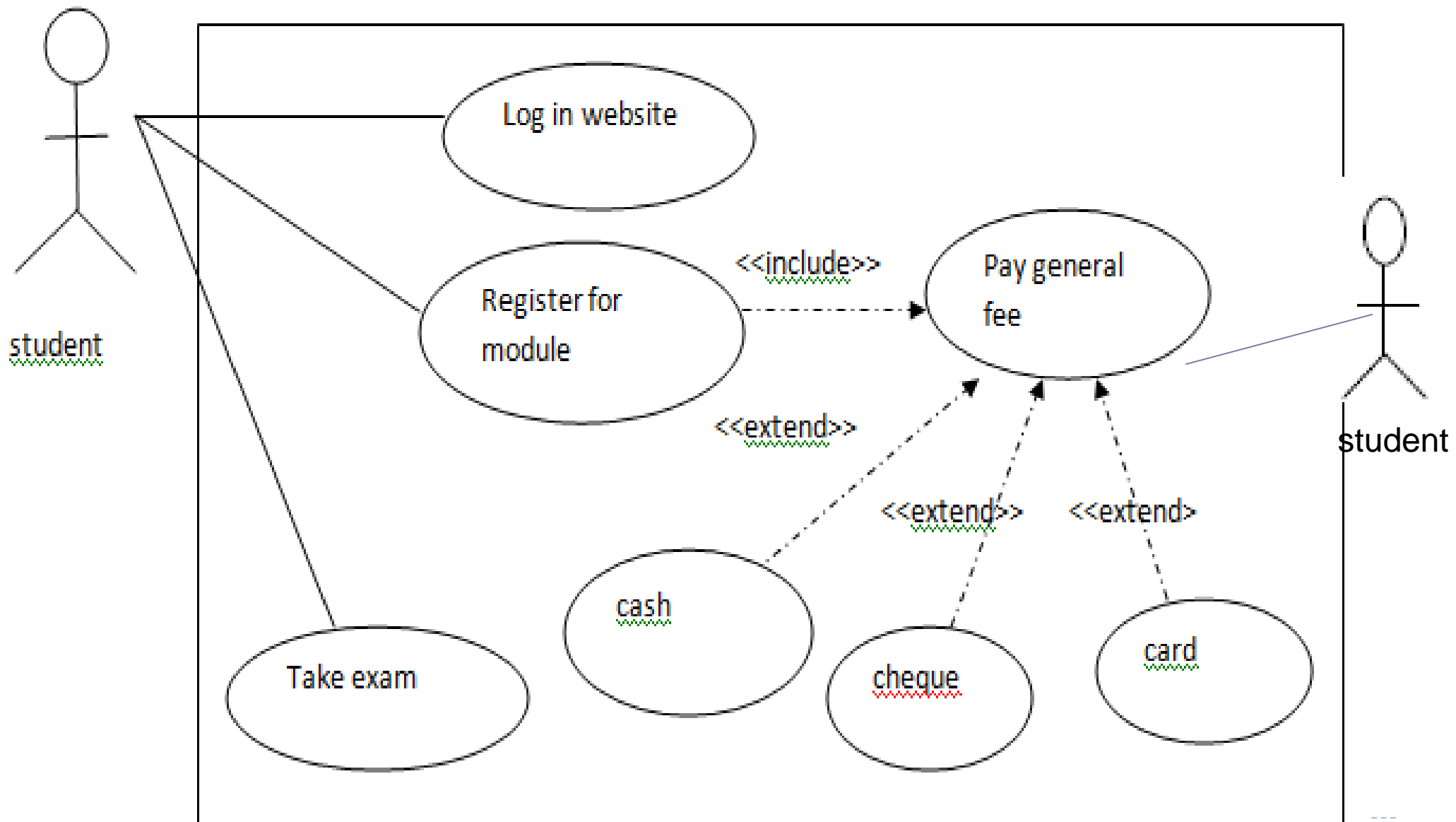


Activity

Registration of Modules to take exam at the University of Mauritius

A student logs on the University website to select the modules that are on offer. The student can register for the modules he wants to take provided that he has paid the “General fees” claimed by the university. This fee can be paid by cash, card or cheque. A receipt is given to the student. The latter can then take the exam at the end of the semester.

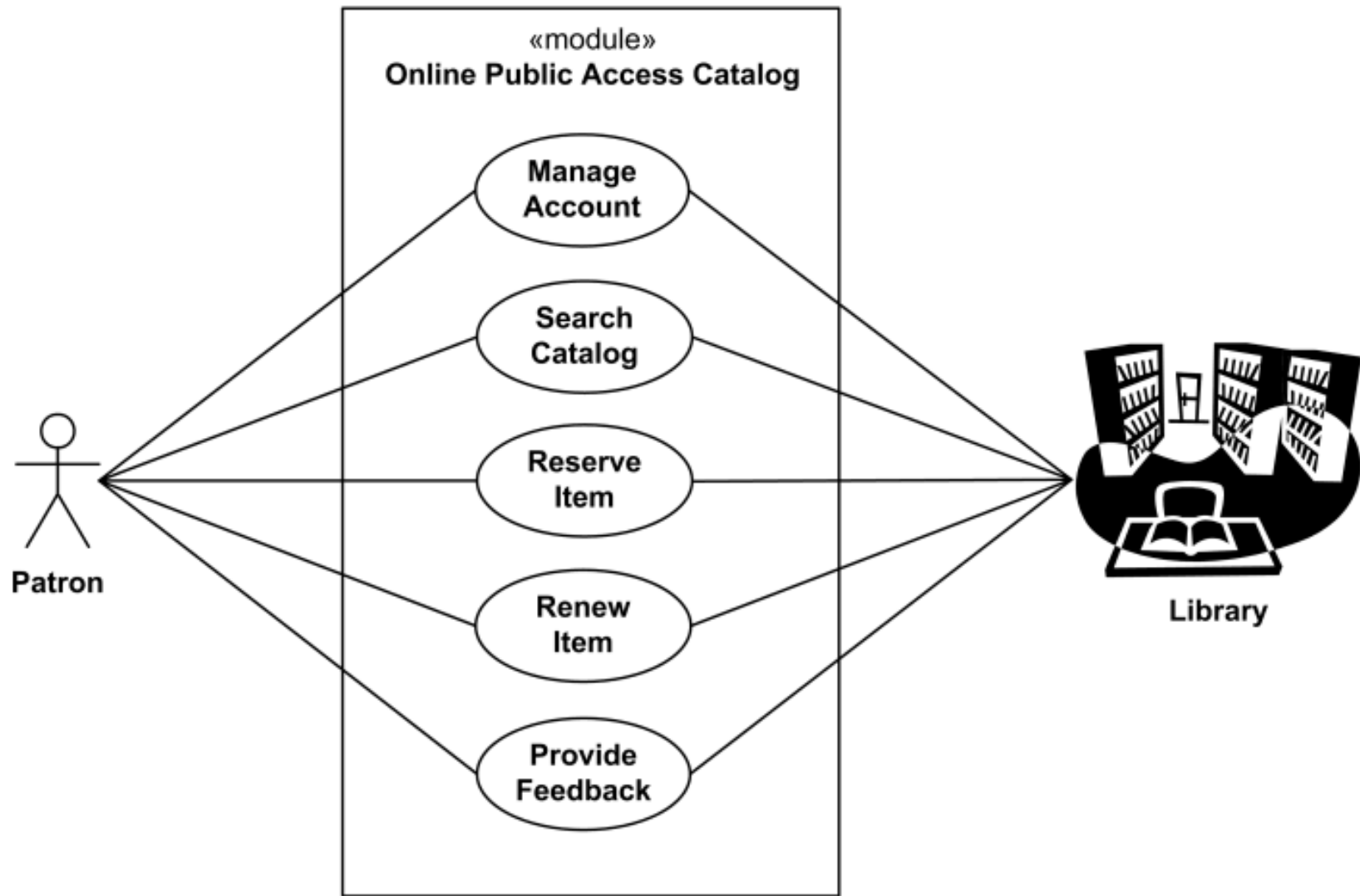
Answer



Example 3: Online Public Access Catalog

- ▶ An **Online Public Access Catalog (OPAC)** is e-Library website which is part of **Integrated Library System (ILS)**, also known as a **Library Management System (LMS)**, and managed by a library or group of libraries.
- ▶ Patrons of the library can search library catalog online to locate various resources - books, periodicals, audio and visual materials, or other items under control of the library. Patrons may reserve or renew item, provide feedback, and manage their account.

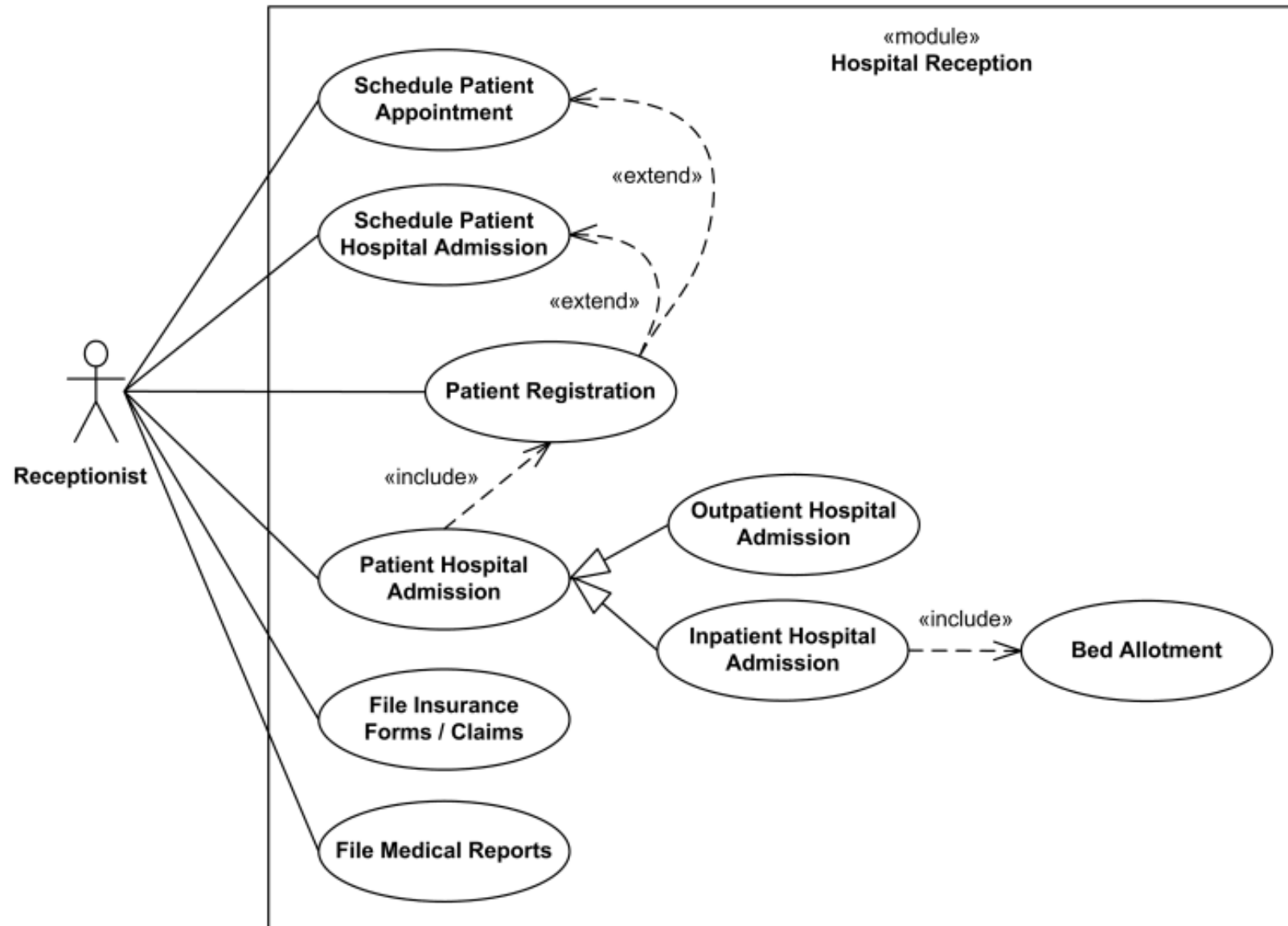
Example 3: Online Public Access Catalog



Example 4: Hospital Reception

- ▶ **Purpose:** Describe major services (functionality) provided by a hospital's reception.
- ▶ **Summary: Hospital Reception** subsystem or module supports some of the many job duties of hospital receptionist. Receptionist schedules patient's appointments and admission to the hospital, collects information from patient upon patient's arrival and/or by phone. For the patient that will stay in the hospital ("inpatient") she or he should have a bed allotted in a ward. Receptionists might also receive patient's payments, record them in a database and provide receipts, file insurance claims and medical reports.

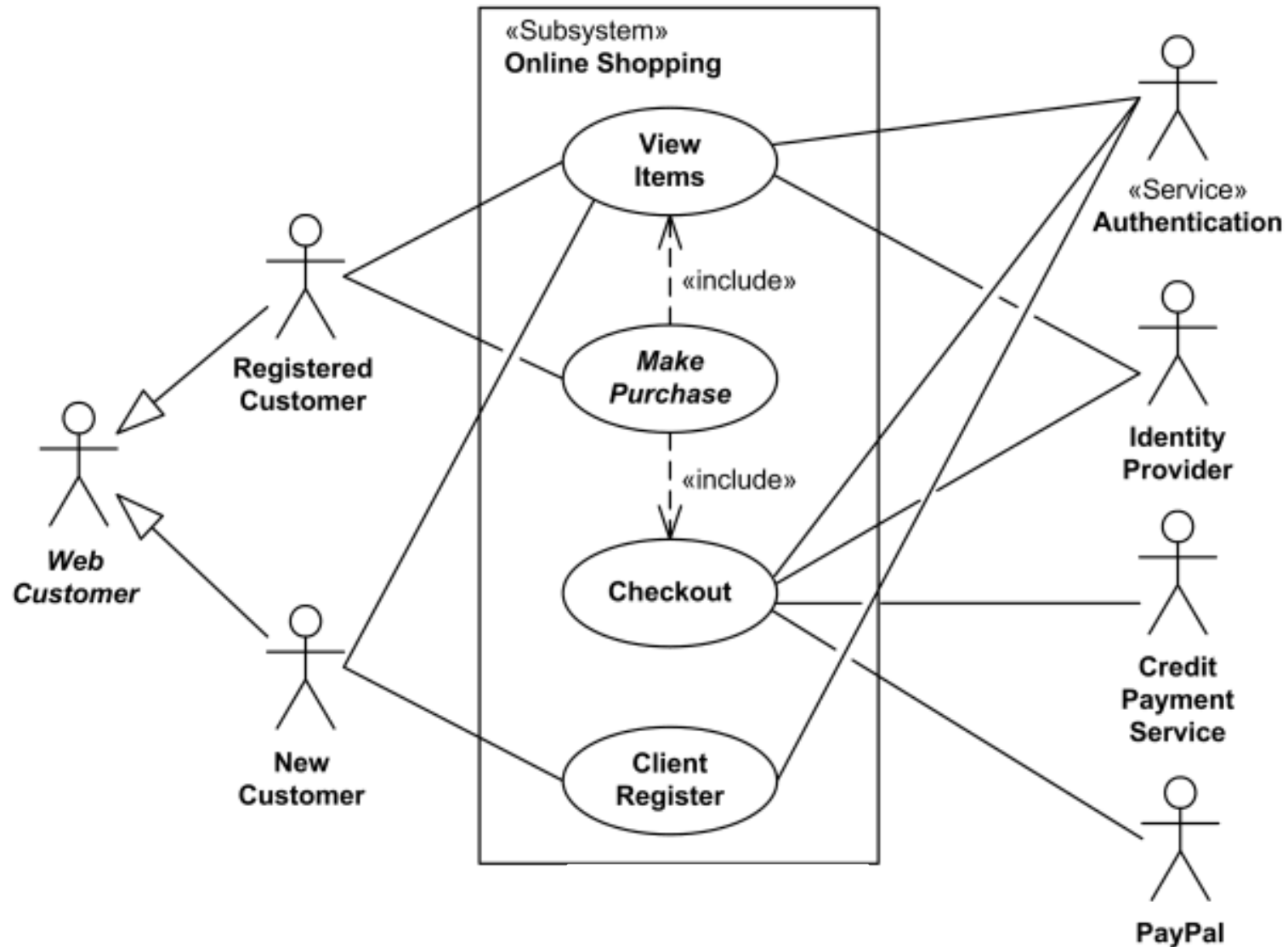
Example 4: Hospital Reception



Example 5: Shopping Online

- ▶ **Purpose:** Provide top level use cases for a web customer making purchases online.
- ▶ **Summary:** Web customer actor uses some web site to make purchases online. Top level use cases are **View Items**, **Make Purchase** and **Client Register**.

Example 5: Shopping Online



Example 5: Shopping Online

- ▶ **View Items** use case is extended by several optional use cases - customer may search for items, browse catalog, view items recommended for him/her, add items to shopping cart or wish list. All these use cases are extending use cases because they provide some optional functions allowing customer to find item.
- ▶ **Customer Authentication** use case is included in **View Recommended Items** and **Add to Wish List** because both require customer to be authenticated. At the same time, item could be added to the shopping cart without user authentication.

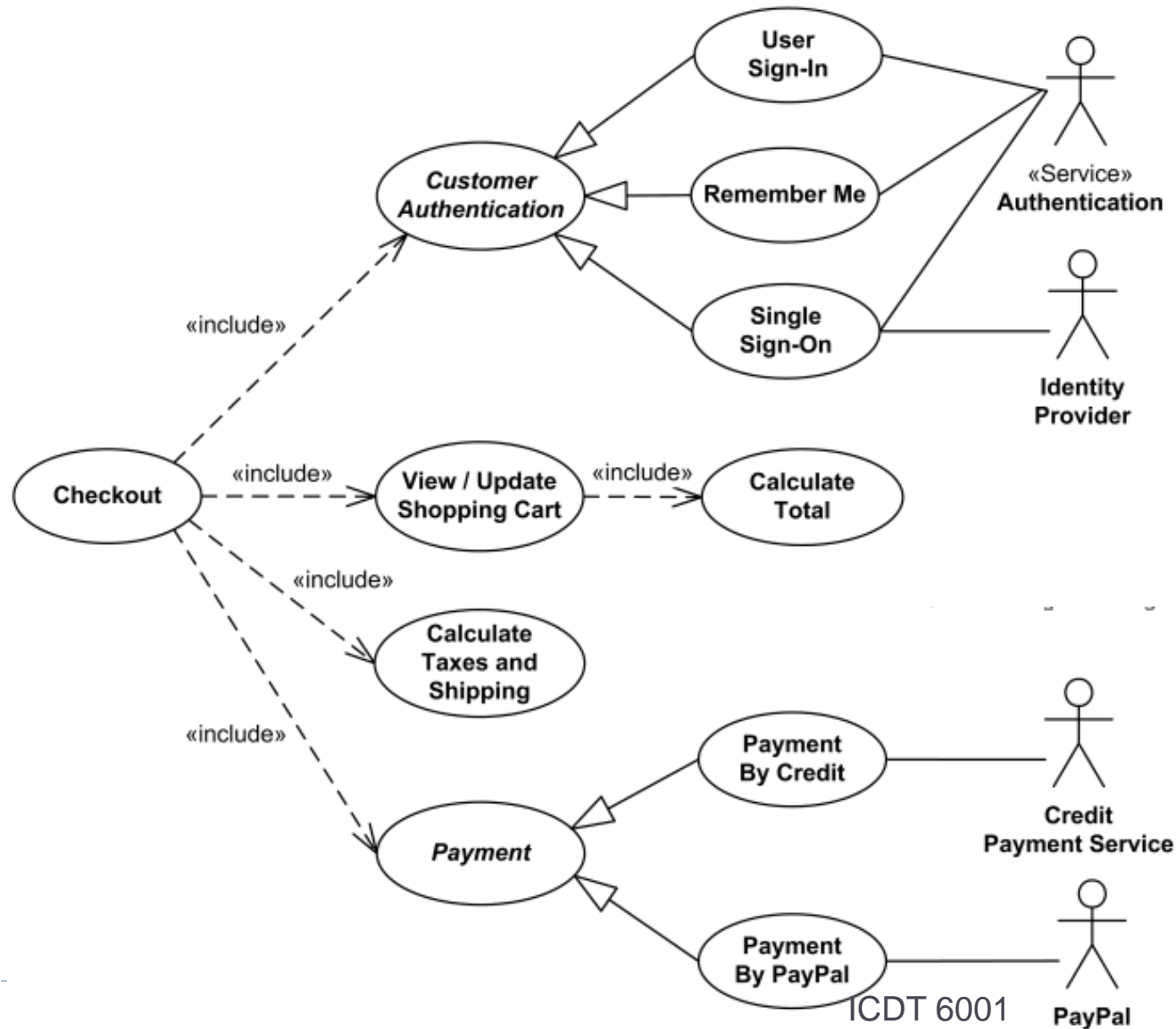
Example 5: Shopping Online



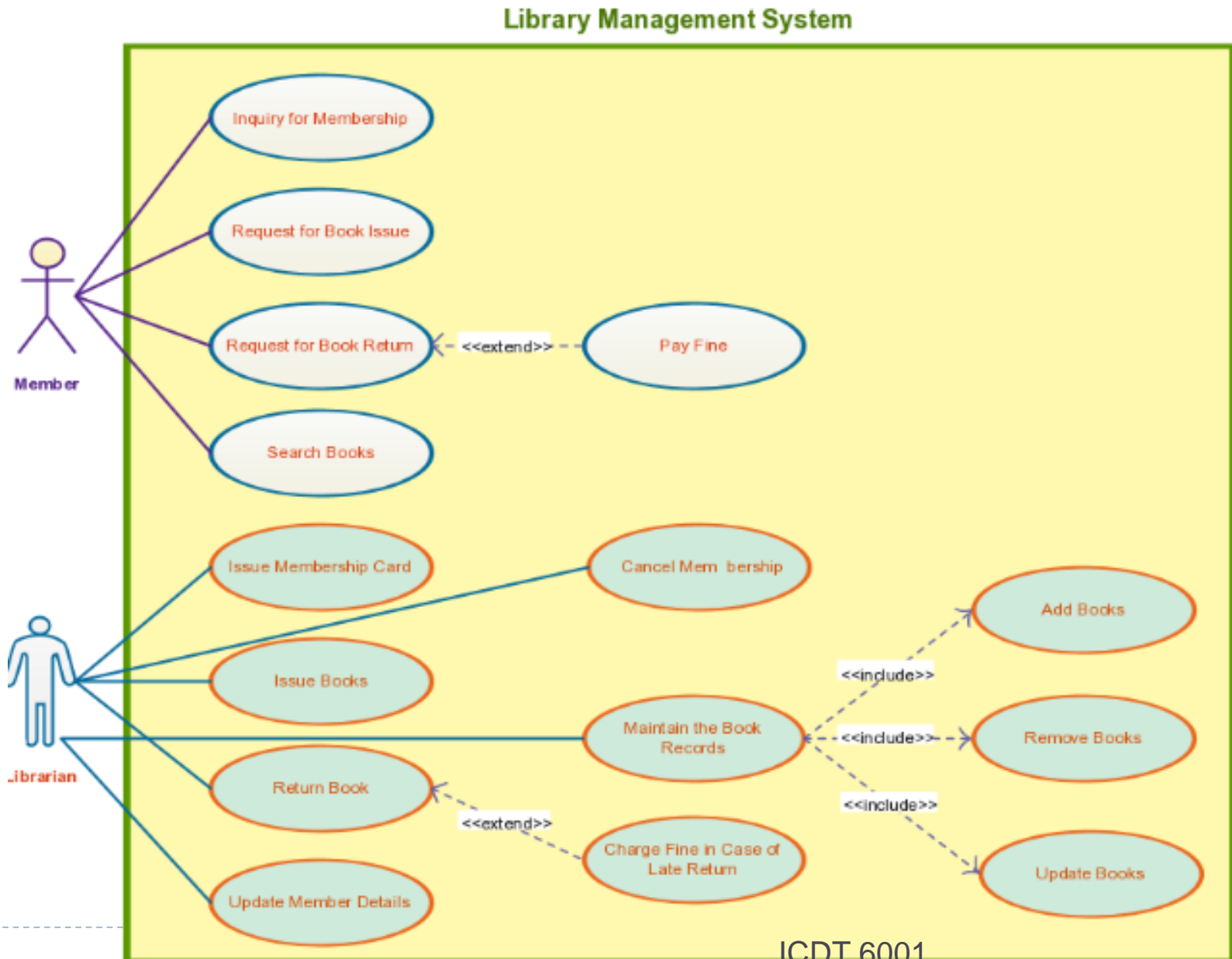
Example 5: Shopping Online

- ▶ **Checkout** use case includes several required uses cases. Web customer should be authenticated. It could be done through user login page, user authentication cookie ("Remember me") or Single Sign-On (SSO). Web site authentication service is used in all these use cases, while SSO also requires participation of external identity provider.
- ▶ **Checkout** use case also includes **Payment** use case which could be done either by using credit card and external credit payment service or with PayPal.

Example 5: Shopping Online



Example 6: Library Management System



Example 7: Trading System

