

MODULE 2 & LAB 2

LANGUAGE FOR MODELING SOFTWARE

This material (provided by tdque@yahoo.com) is offered to students D18CNTT.PTIT. Module 2 focuses on the following issues:

- Brief introduction to Software Engineering
- Software process or software development methodologies
- UML for modeling software

We only focus on some diagrams that are foundations for modeling software in analysis and design phases later.

2.1 BRIEF INTRODUCTION TO SE

Selecting a software process for your project is important. You always follow strategies of activities “short or long” and these *activities* are classified into: *requirement, design, implementation, testing and maintenance*. And each such phase is again a series of activities or process.

Software requirement is a process including a sequence of activities:

- Elicitation
- Analysis
- Specification
- Validation

And the principle purpose of this phase is to produce an artifact called “SRS: Software Requirement Specification”. SRS is an artifact in the form of document which describes what the stakeholders/customer/users need to build. These descriptions may be of natural languages and diagrams as well. This module focuses on presenting diagrams and description of functionality in *use case/scenarios and user stories* which are often utilized for specifying requirements:

- Data flow diagram (DFD)
- Use case – Scenario
- User stories and acceptance criteria
- Activity diagram and swimlane diagram
- Class diagram (in analysis)
- Logical data model (ERD)

Software Design is a process including a sequence of activities of two stages

- Architecture design: describe overall design of the software system w.r.t. hardware systems
 - Package diagram
 - Deployment diagram
- Detailed design
 - Class design (class diagram in design)
 - Database design
 - Interface design

2.2 METHODOLOGY OF SOFTWARE DEVELOPMENT

Methodology

Selecting the right software development methodology for organizing your product development depends largely on your project size, goals, and other factors. Here is an overview of the most widely utilized and recognized software development methodologies to help you decide which is right for your team (2019)

<https://blog.planview.com/top-6-software-development-methodologies/>

Top 4 software development methodologies

<https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/>

2.3 LANGUAGE FOR MODELING SOFTWARE

2.3.1 Introduction to UML diagrams

Any language from natural languages to mathematical language, notations such as “a”, “b”... or $\sum, \sqrt{\quad}$ contain in themselves the meanings we need to understand to describe or express statements, ideas, formulas...UML (Unified Modeling Language) is a language we need to learn for describing/modeling the software systems that we need to develop. UML and some other diagrams are accepted in software engineering to represent or model the systems including both software and hardware in phases:

- Requirement
- Design
- Implementation

The principle webpage of organization UML

<https://www.uml-diagrams.org/>

includes all things you need to discover for you future jobs.

The tool VP

<https://www.visual-paradigm.com/download/>

with tutorials

<https://www.visual-paradigm.com/tutorials/>

and guides

<https://www.visual-paradigm.com/guide/>

are very useful for students in studying and pursuing the interesting jobs in information technology.

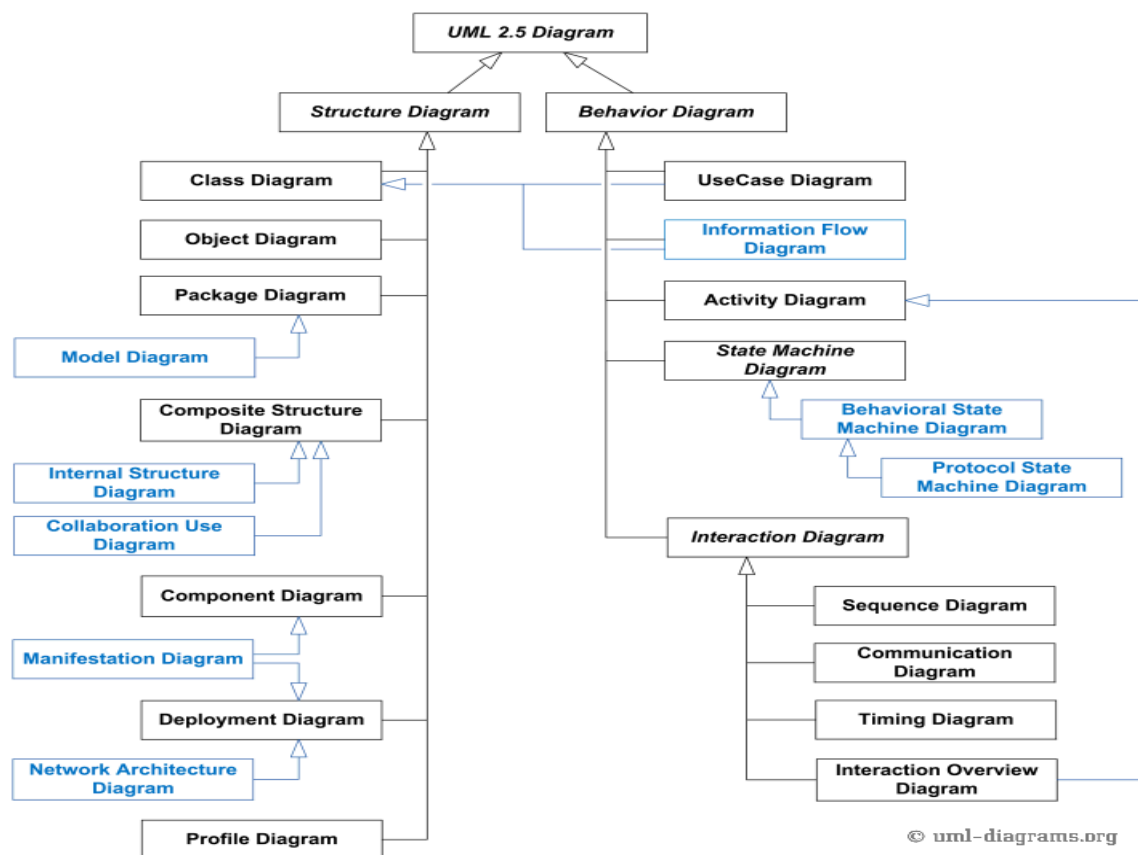
2.3.2 Data flow diagram (DFD) //not UML

2.3.3 Sequence diagram

2.3.4 Activity diagram

2.3.5 Class diagram (in analysis)

2.3.6 Entity relation diagram (ERD) //not UML



LAB 2

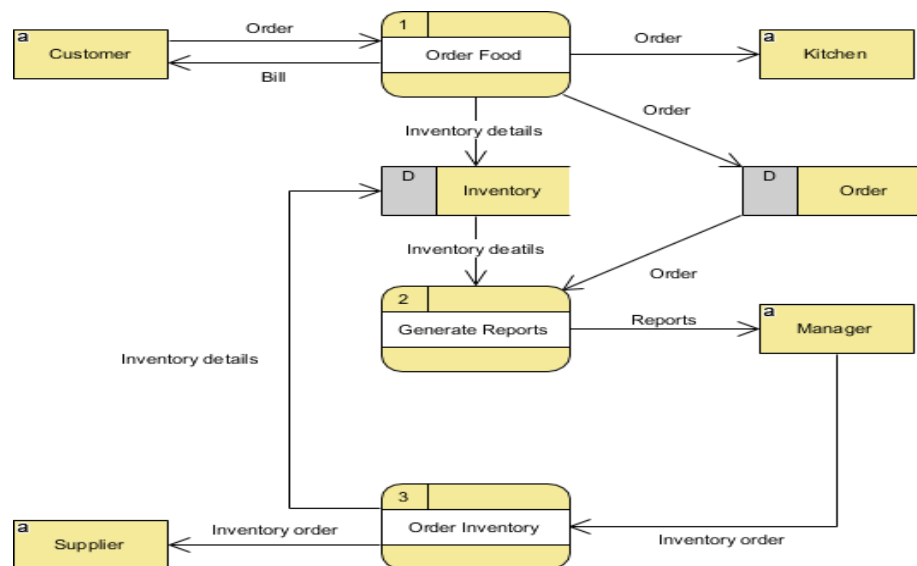
Exercise 1: (1 mark)

- Investigate Figure 1
Explain in words the components and relationships of the following DFD level1.
- Draw it with VP tool

What is DFD and when we use it?

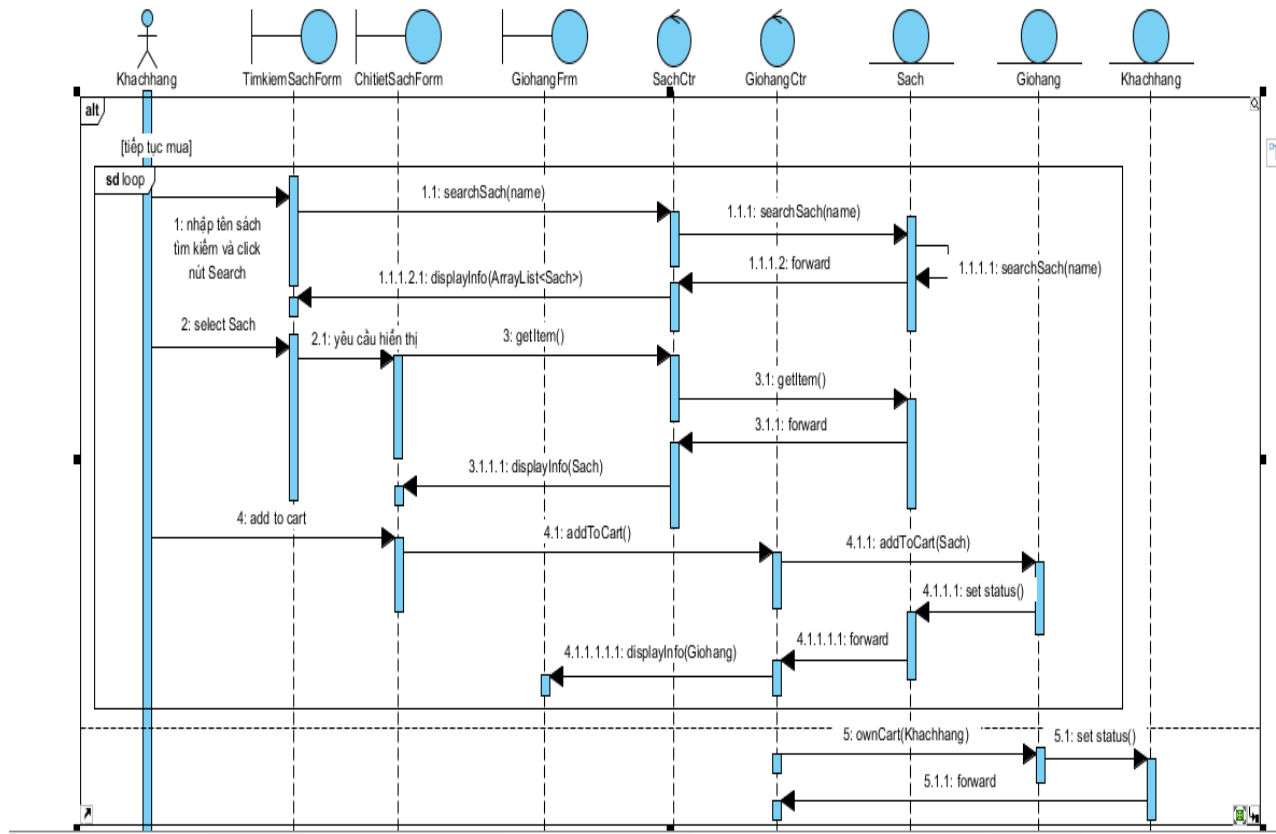
- Data flow diagram (DFD) is used to represent *the flow of data throughout processes in a given system*.
- DFD is often used in requirement phase and divided into levels details 0, 1, 2. DFD level 0 is also called **context diagram**.
- DFD shows the kind of information that will be input to and output from processes as well as where the data will be stored.
- Data flow diagrams (DFD) are categorized in logical or physical ones. A logical DFD focuses on the business and how the business operates. It describes the business events that take place and the data required and produced by each event. Whereas, a physical DFD shows how the system will be implemented.

<https://www.visual-paradigm.com/guide/data-flow-diagram/logical-vs-physical-data-flow-diagrams/>



Exercise 2 (1 mark): Investigate the sequence diagram of creating cart in Book Store

- Explain *entity, controller, boundary classes and methods* in the diagram
- What is still not suitable to UML convention? Hint: <-----
- Redraw it in VP



Exercise 3 (1.5 marks): Investigate two activity diagrams: Order process and process for student enrollment in a university

- Explain notations, flow...
- Draw them in VP

Order Process as follows

Once the order is received, the activities split into two parallel sets of activities. One side fills and sends the order while the other handles the billing.

On the Fill Order side, the method of delivery is decided conditionally. Depending on the condition either the Overnight Delivery activity or the Regular Delivery activity is performed.

Finally the parallel activities combine to close the order.

A process for student enrollment in a university as follows:

- An applicant wants to enroll in the university.
- The applicant hands a filled out copy of Enrollment Form.
- The registrar inspects the forms.
- The registrar determines that the forms have been filled out properly.
- The registrar informs student to attend in university overview presentation.
- The registrar helps the student to enroll in seminars
- The registrar asks the student to pay for the initial tuition.

Visual Paradigm

What's New Features Tutorials Support Pricing Try Now Request Demo

What is Activity Diagram?
When to Use Activity Diagram
Learn by Examples
Word Processor
Processor Order
Student Enrollment
Swimlane
Activity Diagram Notations

The activity diagram example below visualize the flow in graphical form.

```

graph TD
    Start(( )) --> ReceiveOrder([Receive Order])
    ReceiveOrder --> Fork1[ ]
    Fork1 --> FillOrder([Fill Order])
    Fork1 --> SendInvoice([Send Invoice])
    FillOrder --> Decision1{ }
    Decision1 -- "[rush order]" --> ArrangeOvernight([Arrange Overnight Delivery])
    Decision1 -- "[else]" --> ArrangeRegular([Arrange Regular Delivery])
    ArrangeOvernight --> Join1{ }
    ArrangeRegular --> Join1
    SendInvoice --> ReceivePayment([Receive Payment])
    Join1 --> Fork2[ ]
    ReceivePayment --> Fork2
    Fork2 --> End((( )))
  
```

We use cookies to offer you a better experience. By visiting our website, you agree to the use of cookies as described in our [Cookie Policy](#). OK

Visual Paradigm

What's New Features Tutorials Support Pricing Try Now Request Demo

What is Activity Diagram?
When to Use Activity Diagram
Learn by Examples
Word Processor
Processor Order
Student Enrollment
Swimlane
Activity Diagram Notations

- The registrar informs student to attend in university overview presentation.
- The registrar helps the student to enroll in seminars
- The registrar asks the student to pay for the initial tuition.

```

graph TD
    Start(( )) --> FillOutForms([Fill-out Enrollment Forms])
    FillOutForms -- "[correct]" --> EnrollInUniversity([Enroll in University])
    FillOutForms -- "[incorrect]" --> Decision1{ }
    Decision1 -- "[help available]" --> ObtainHelp([Obtain Help to Fill-out Forms])
    Decision1 -- "[trivial problems]" --> EnrollInUniversity
    ObtainHelp --> EnrollInUniversity
    EnrollInUniversity --> Fork1[ ]
    Fork1 --> AttendPresentation([Attend University Overview Presentation])
    Fork1 --> EnrollInSeminar([Enroll in Seminar(s)])
    AttendPresentation --> Fork2[ ]
    EnrollInSeminar --> Fork2
    Fork2 --> MakePayment([Make Initial Tuition Payment])
    MakePayment --> Fork3[ ]
    Fork3 --> End((( )))
  
```

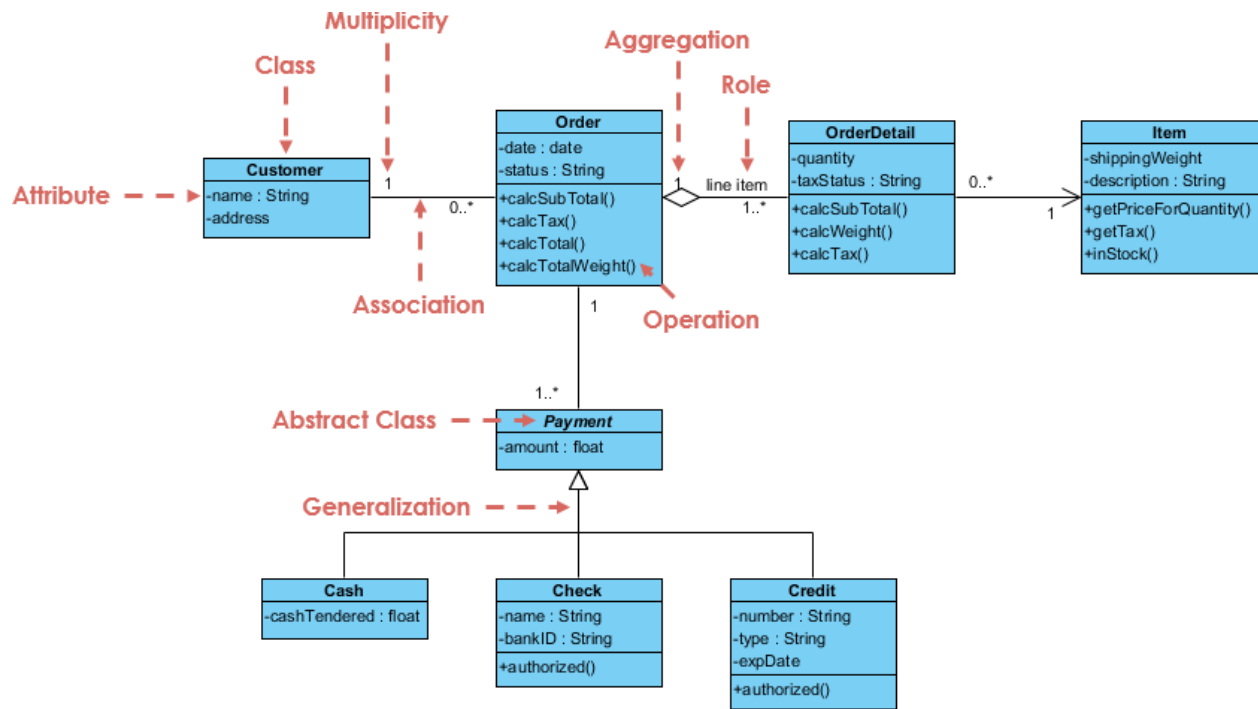
Activity Diagram - Swimlane

A swimlane is a way to group activities performed by the same actor on an activity diagram or activity diagram or to group activities in a single thread. Here is an example of a swimlane activity diagram for modeling Staff Expenses Submission:

We use cookies to offer you a better experience. By visiting our website, you agree to the use of cookies as described in our [Cookie Policy](#). OK

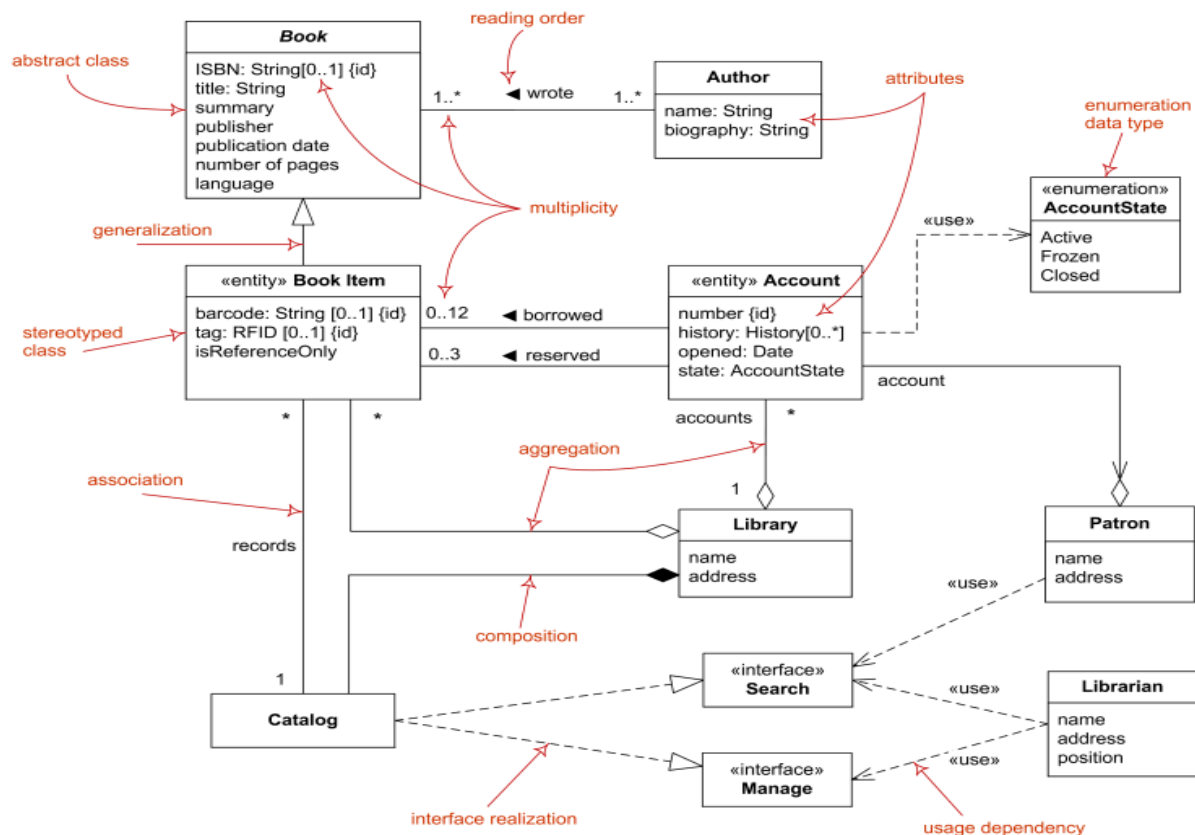
Exercise 4 (1.5 marks): Investigate the class diagram of an order system of items

- Construct a class diagram in analysis and a class diagram in design from the diagram
- Add attributes/methods into classes
- Create packages and relationships
- Draw the diagram in VP



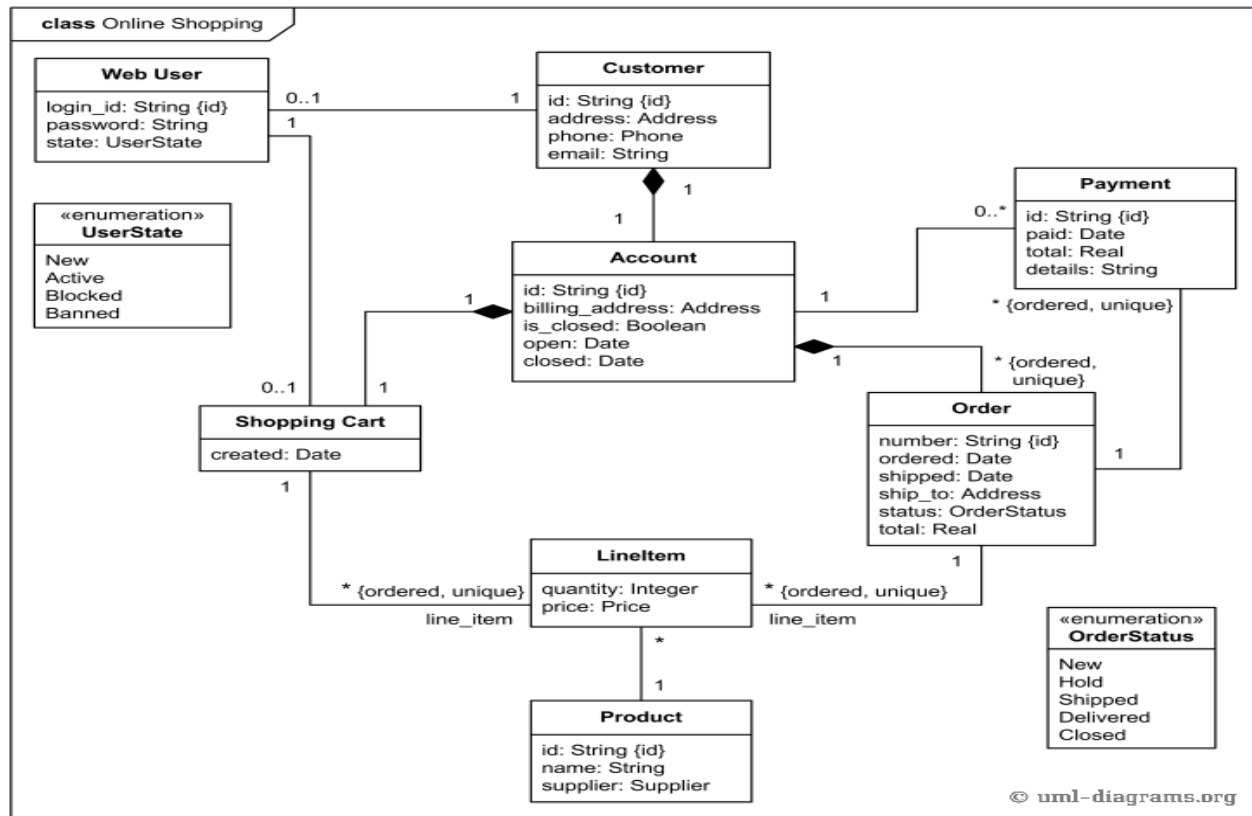
Exercise 5 (1.5 marks): Investigate the class diagram of a library management system

- Construct a class diagram in analysis and a class diagram in design from the diagram
- Add attributes/methods into classes
- Add packages and relationships
- Draw the diagram in VP



Exercise 6 (1.5 marks): Investigate the class diagram of a shopping online system

- Construct a class diagram in analysis and a class diagram in design from the diagram
- Add attributes/methods into classes
- Add packages and relationships
- Draw the diagram in VP



Exercise 7 (2 marks): Given three diagrams: conceptual model, logical data model and physical data model.

- Compare three models and give your comments
- Draw them in on a sheet o paper and rewrite your comments
- Draw them in VP

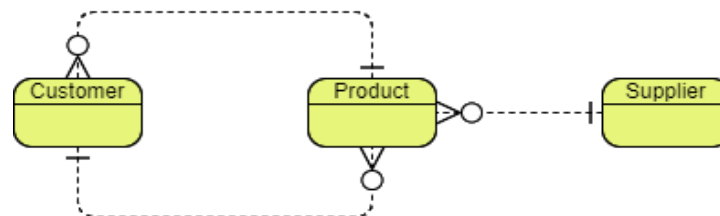
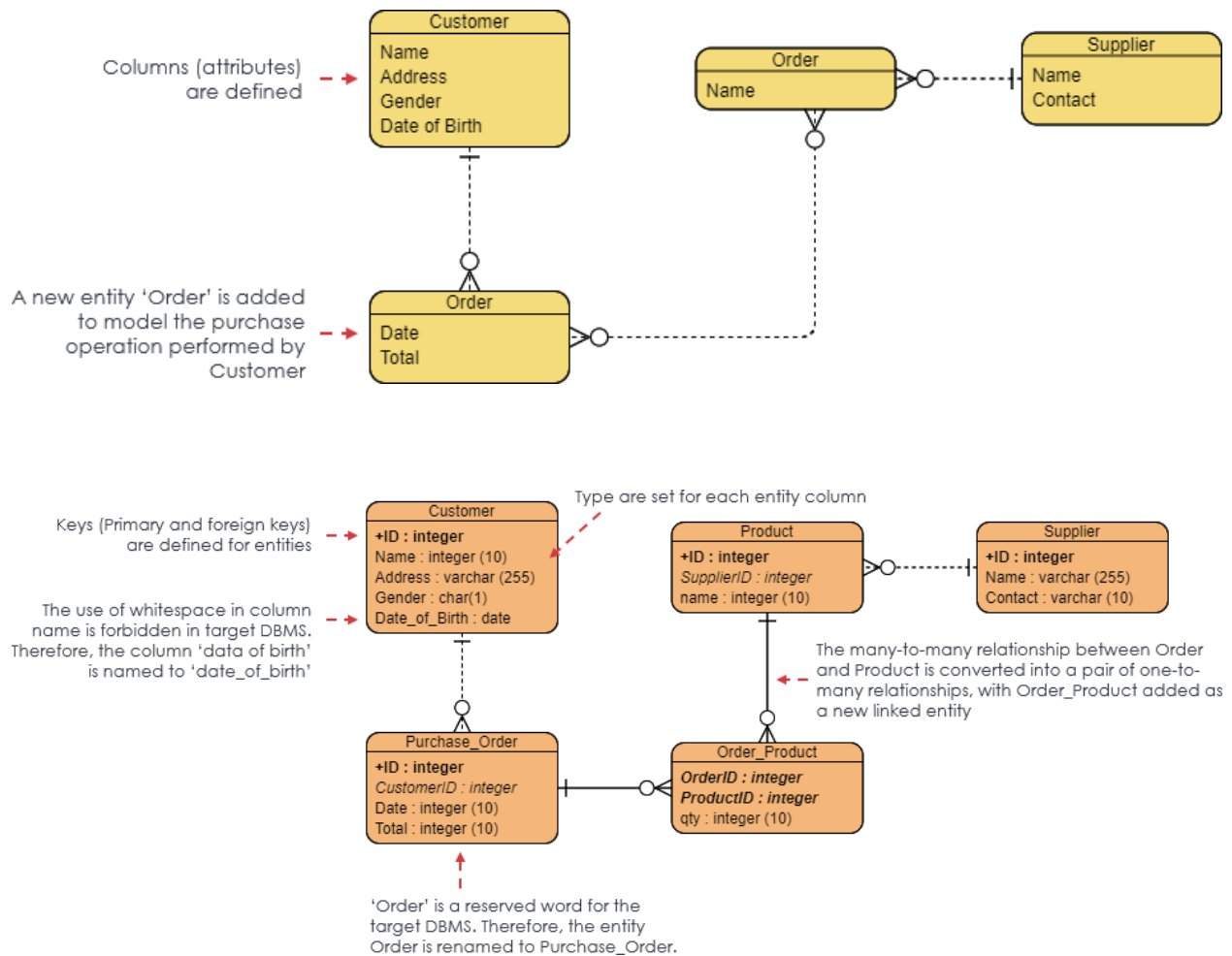


Figure: conceptual model



- The **Conceptual Model** is To Establish The Entities, Their Attributes, And Their Relationships.
- The **Logical Data Model** defines The Structure Of The Data Elements and Set the relationships between them.
- The **Physical Data Model** describes The Database-Specific Implementation Of The Data Model.

Conceptual ERD models the business objects that should exist in a system and the relationships between them. A *conceptual model is developed to present an overall picture of the system* by recognizing the business objects involved. It defines what entities exist, NOT which tables.

Logical data model

Logical ERD is a detailed version of a Conceptual ERD. A logical ER model is developed to enrich a conceptual model by *defining explicitly the columns in each entity and introducing operational and transactional entities*. Although a logical data model is still independent of the actual database system in which the database will be created, you can still consider that if it affects the design.

Physical data model

Physical ERD represents the actual design blueprint of a relational database. A physical data model elaborates on the logical data model by assigning each column with type, length, nullable, etc. Since a physical ERD represents how data should be structured and related in a specific DBMS. It is important to consider the convention and restriction of the actual database system in which the database will be created. Make sure the column types are supported by the DBMS and reserved words are not used in naming entities and columns.