



Software Design

Team Teaching Software Engineering Course
Department of Information Technology
State Polytechnic of Malang

Purpose

- Understand software design

Preface

Requirement	Analisis
<p>The output from the requirements stage becomes input at the analysis stage.</p> <p>This section is important because it is the stage of understanding the problem domain to be solved by the software.</p>	<p>Output of the Stages Analysis is used at the design stage.</p> <p>The results of this process are used as a reference for creating software implementations.</p>

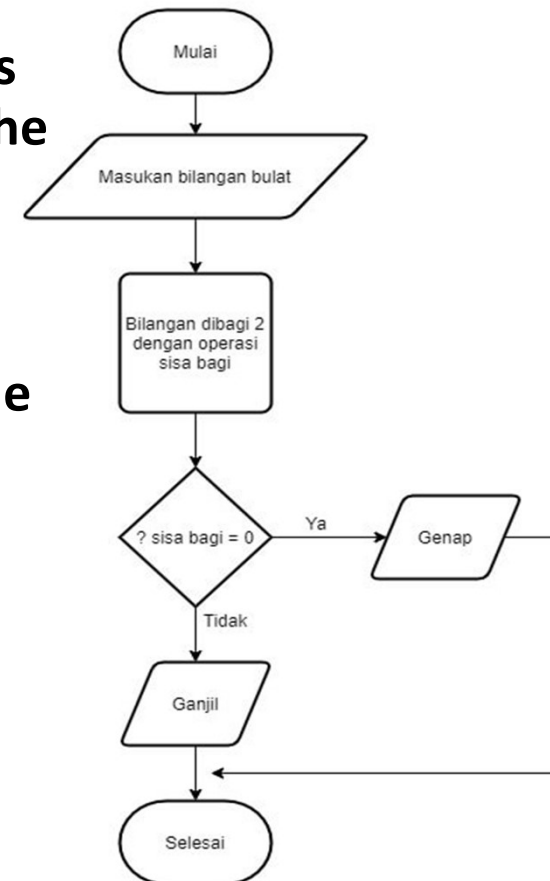
The design stages translate software requirements into a model that software developers can understand

Preface

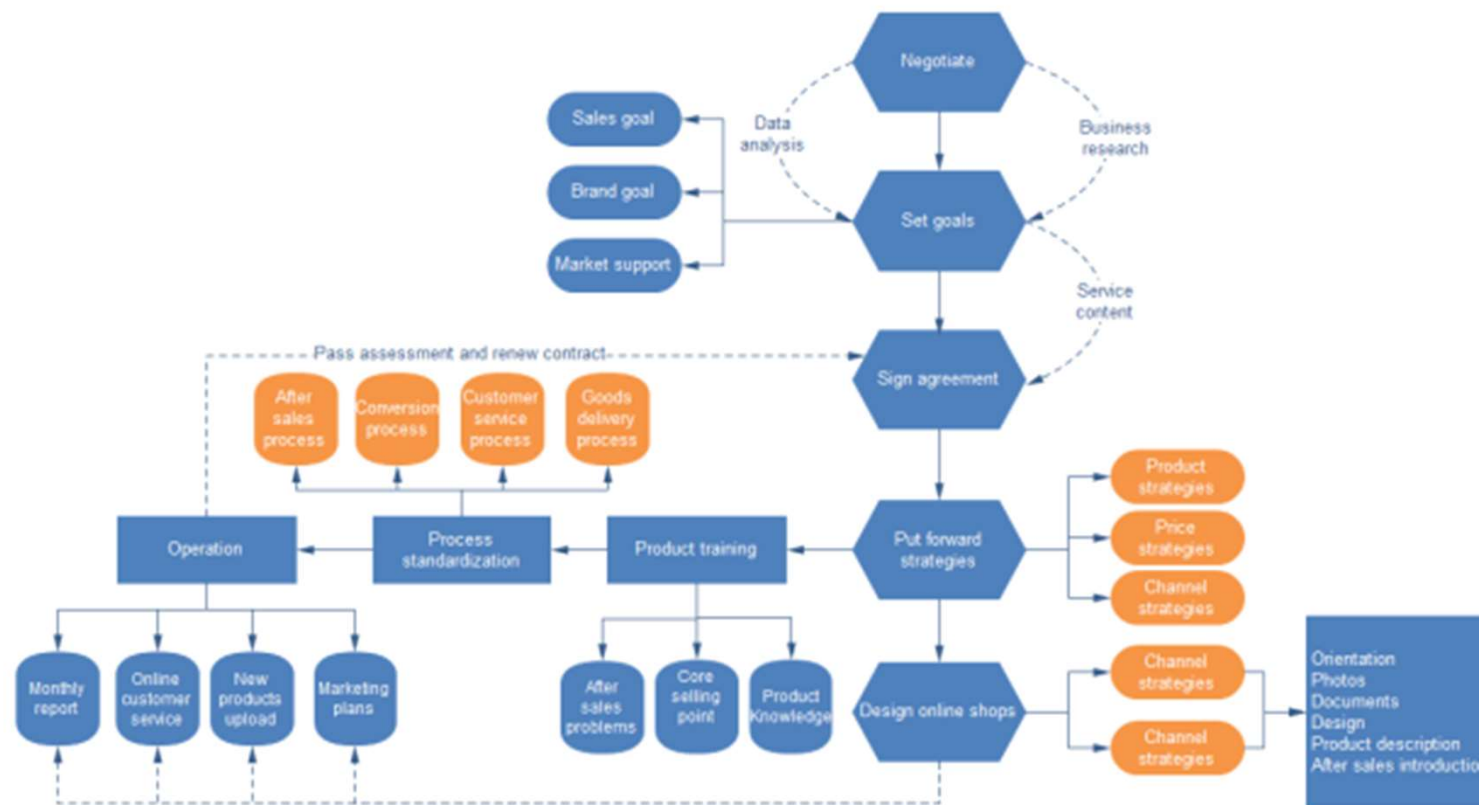
	Paradigm	Diagrams
1	Process-oriented Paradigm	Flowchart
2	Data-oriented Paradigm	DFD
3	Object-oriented Paradigm (data + process)	UML

Flowchart

- A flowchart is a diagram that displays the steps and decisions to perform a process of a program. Each step is depicted in the form of a diagram and connected by a line or direction of an arrow.
- Flowcharts play an important role in deciding a step or functionality of creating a program that involves many people at once.
- The use of flowcharts in the world of programming is also a great way to make connections between technical and non-technical needs, and programs will be clearer, concise, and reduce the possibility of misinterpretation.

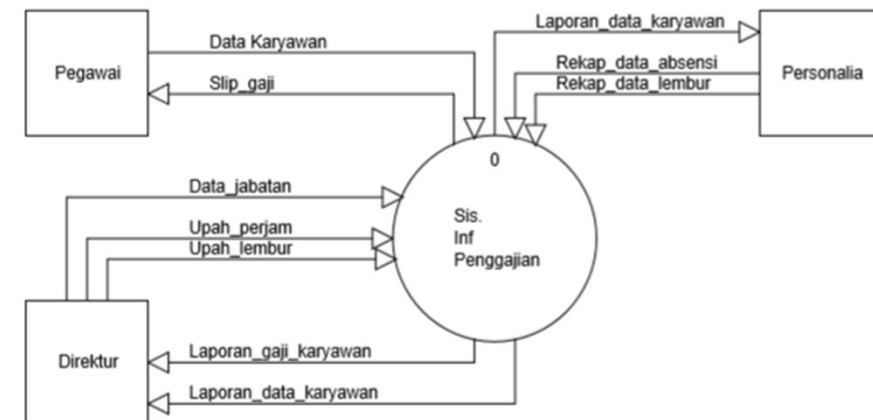


E-commerce Trusteeship Service Flowchart

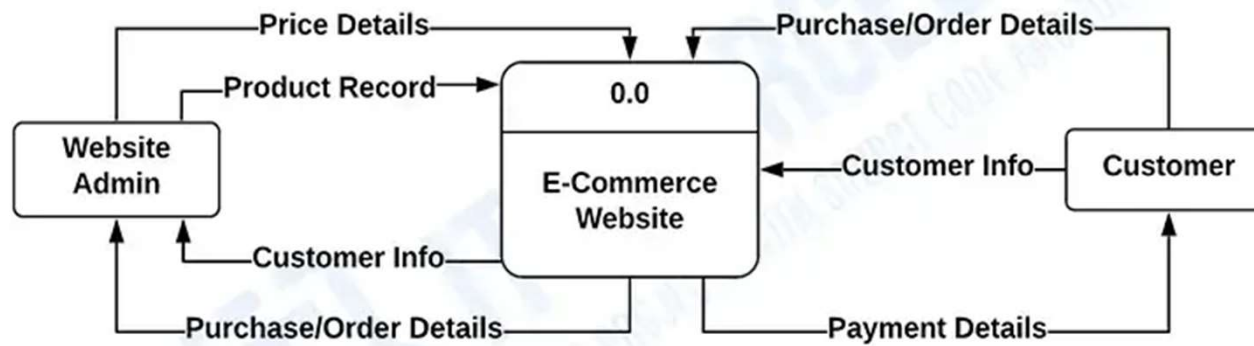


Data Flow Diagram (DFD)

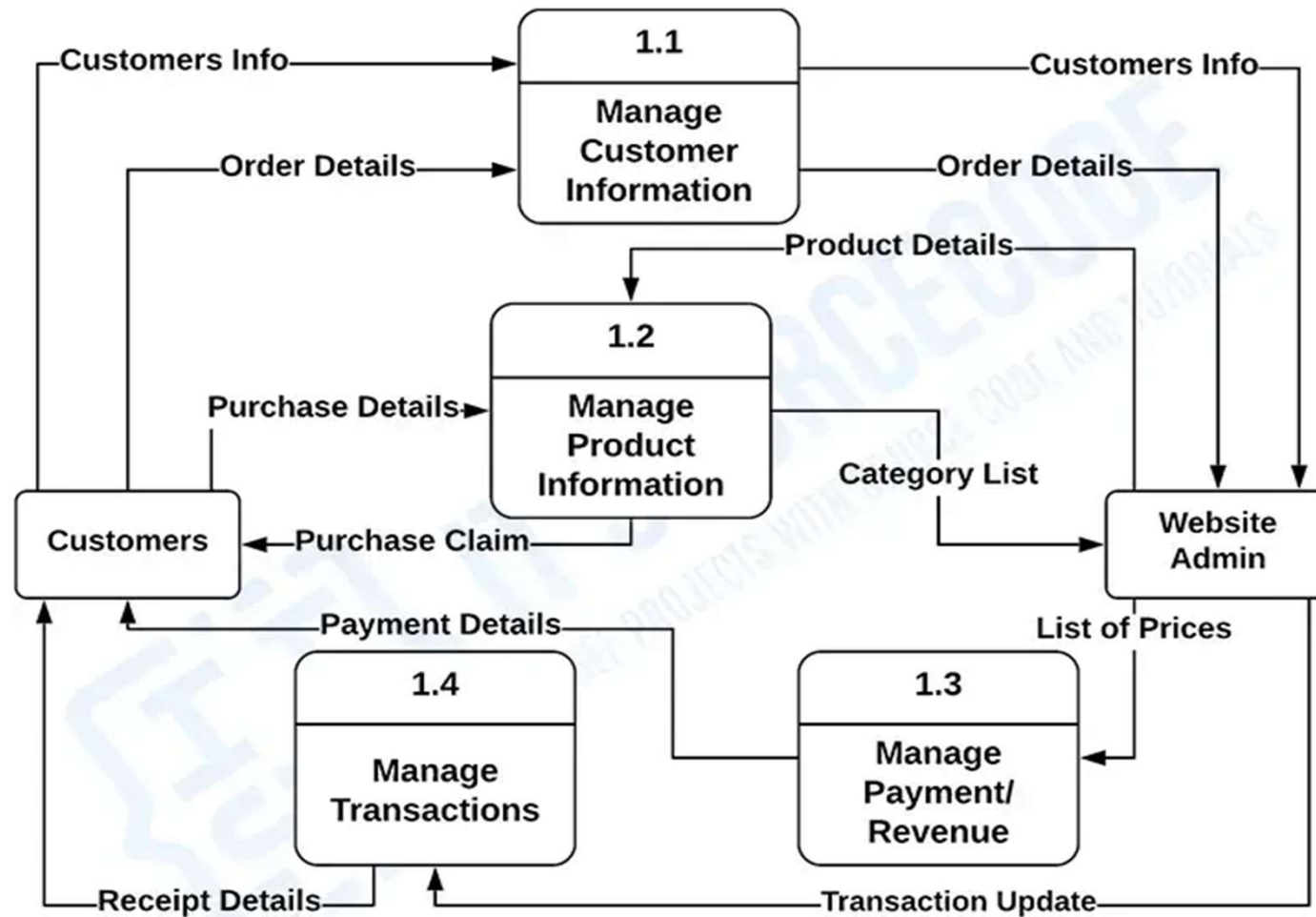
- Data Flow Diagram (DFD) is a modelling tool that allows systems professionals to describe systems as being connected to each other by data flow, either computerized. DFD is often referred to as bubble chart, bubble diagram, process model, workflow diagram, or function model.
- DFD is a model creation tool that emphasizes only system functions and is oriented to data flow with the concept of decomposition can be used for the description of analysis and system design that is easily communicated by system professionals to users and program makers.



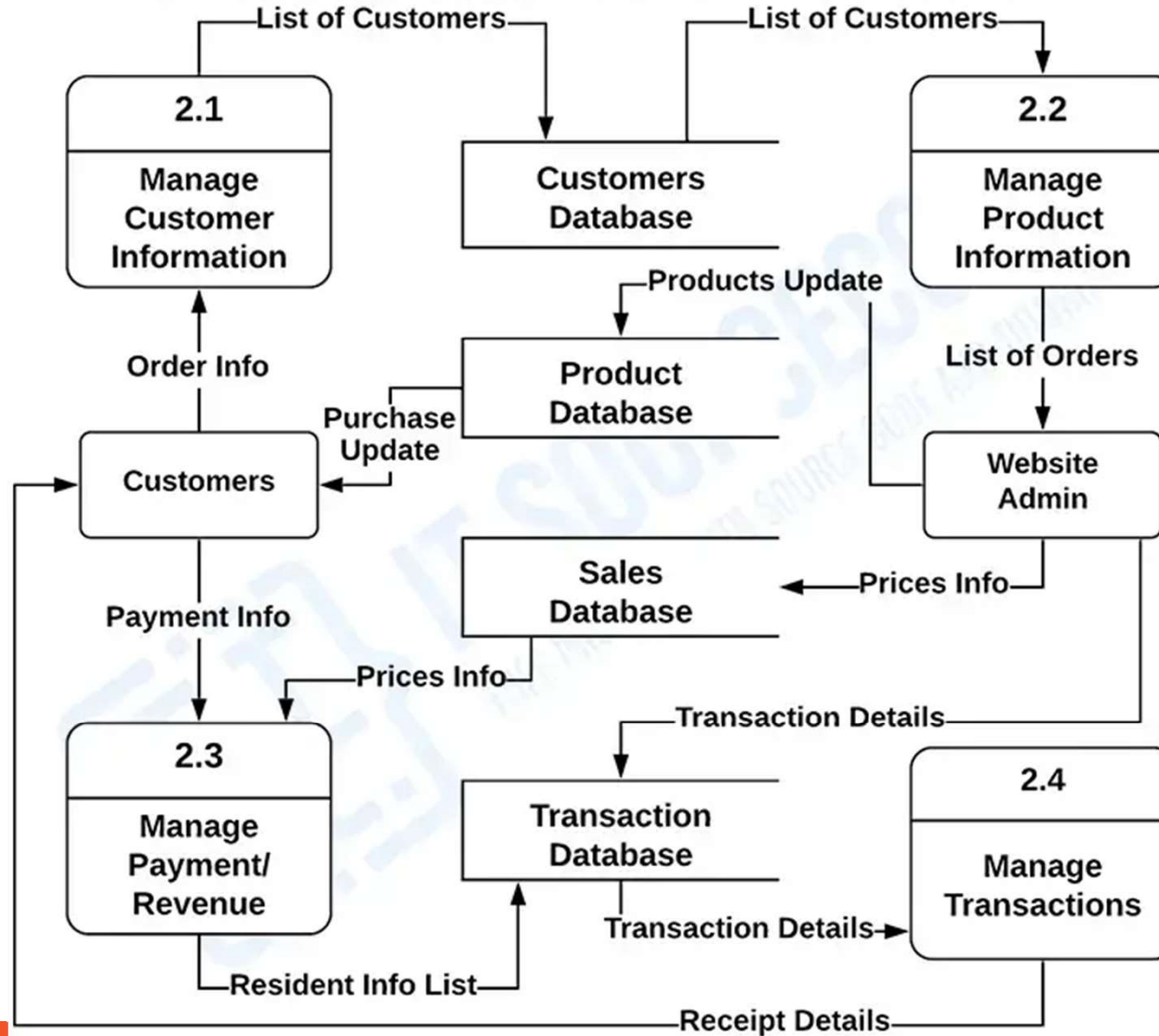
E-COMMERCE WEBSITE SYSTEM



E-COMMERCE WEBSITE SYSTEM

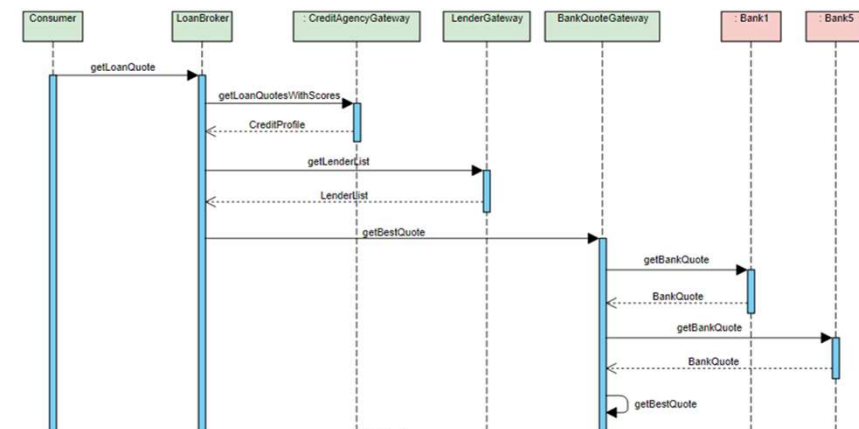
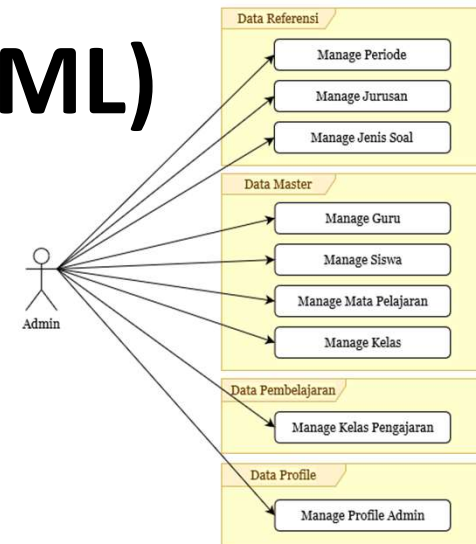


E-COMMERCE WEBSITE SYSTEM

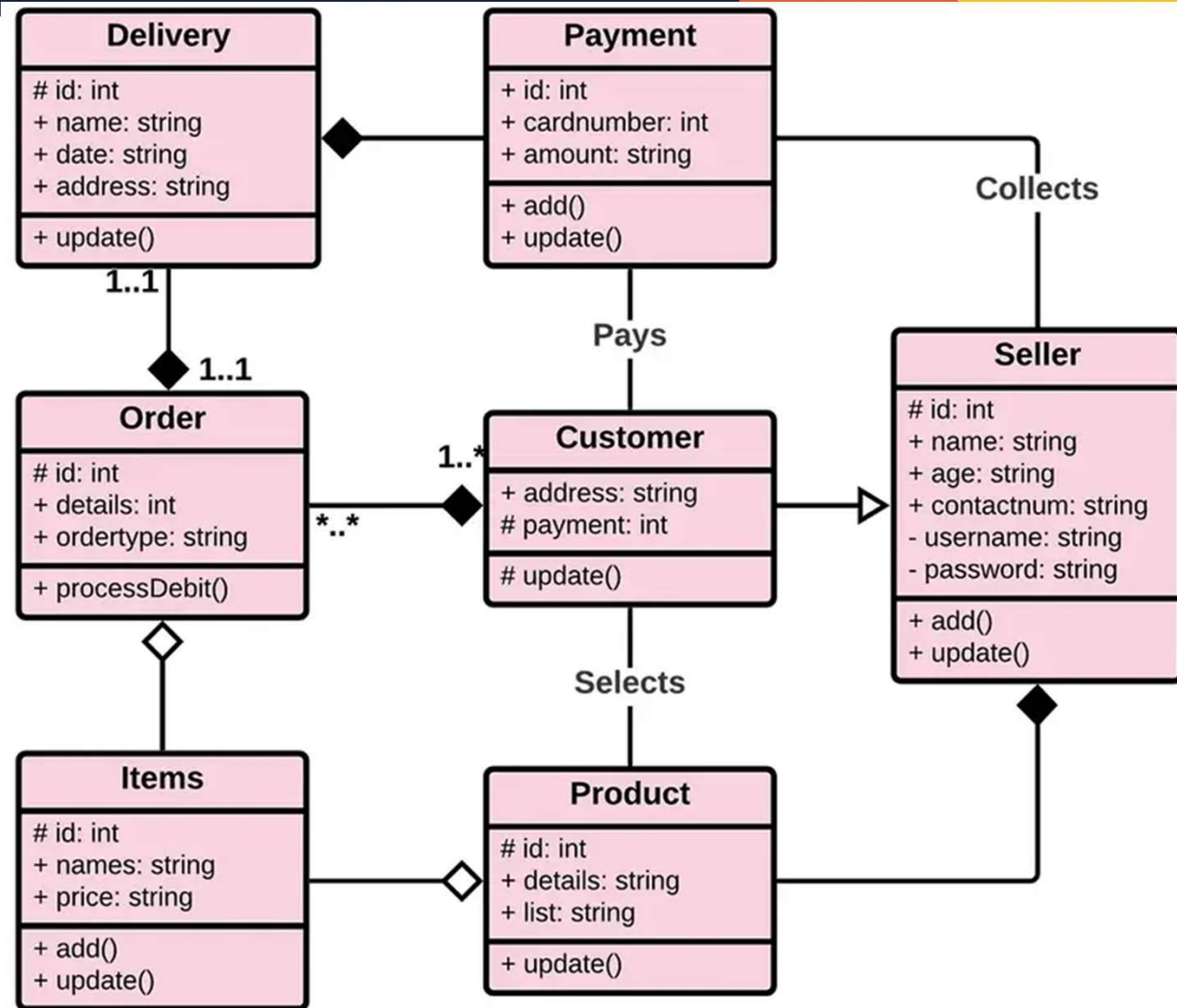


Unified Modeling Language (UML)

- Unified Modelling Language (UML) is a method of visual modeling used as a means of designing object-oriented systems.
- UML can also be defined as a standard language for visualizing, designing, and documenting systems, or also known as the standard language for writing blueprints for software.

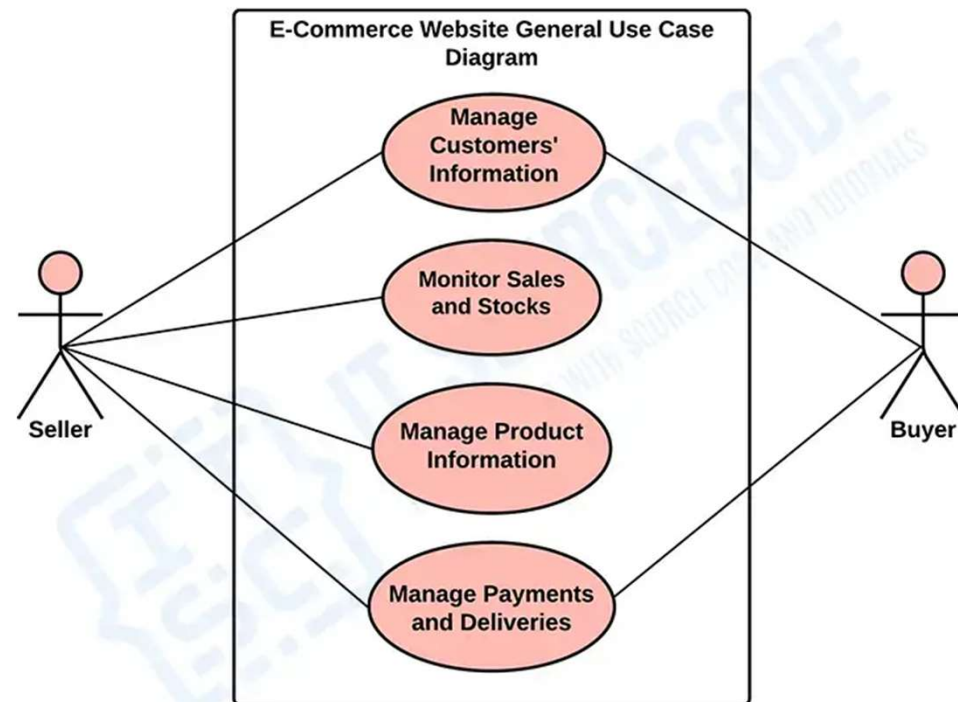


Class Diagram



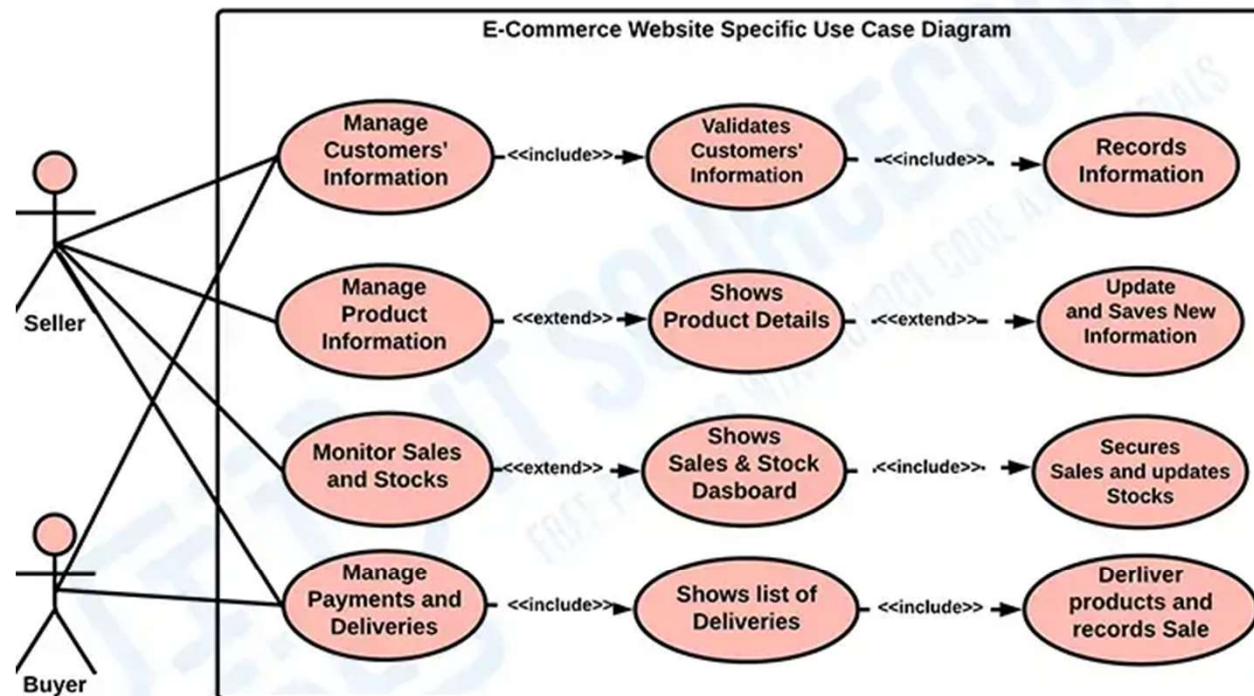
Use Case Diagram

E-COMMERCE WEBSITE

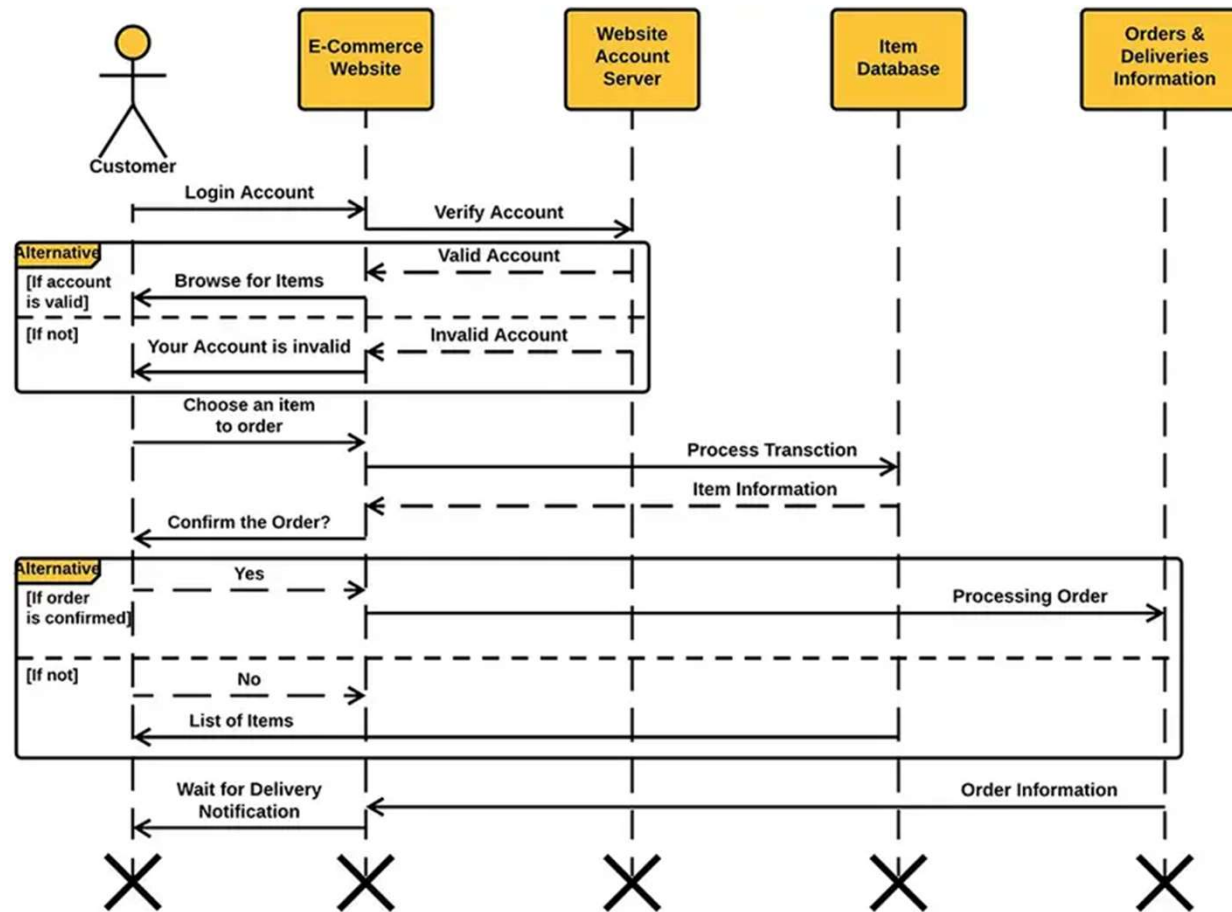


GENERAL USE CASE DIAGRAM

E-COMMERCE WEBSITE

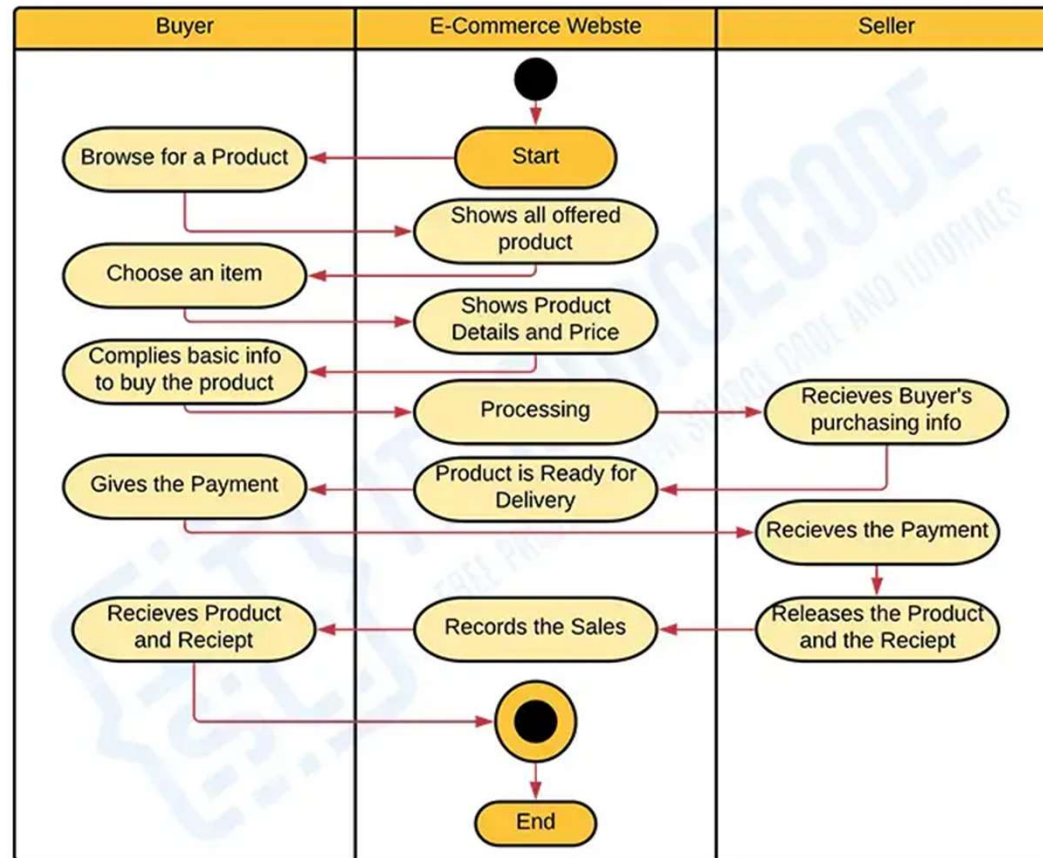


Sequence Diagram



Activity Diagram

E-COMMERCE WEBSITE



ACTIVITY DIAGRAM

Overview

- Software design is a stage of software development whose results can be used by software developers to create programs.
- Input software design: Requirements analysis model and specification documentation
- Design software output: Design model and design specification documentation

Design Principles

Software design, both conventional/procedural and object-oriented design, is carried out by referring to certain principles or guidelines to facilitate the design process itself and to produce high-quality designs.

These principles are general principles that can be applied to any project.

Design Process

Design aims to form a model that is ready to be implemented into the program. In forming a design model, there are a series of processes that need to be carried out while adhering to design principles. All design activities need to be documented and the documentation process needs good management.

In object-oriented design, several other activities are outlined at the design stage

- Define design objectives.

- Define subsystems.

- Mapping subsystems into the platform used

- Persistent data management.

- Define access control.

- Define flow control.

- Defines boundary conditions.

Types of software design

- **Data design:** transforming an information domain model into a data structure
- **Architectural design:** describes the relationships between the main structural elements of the program
- **Interface design:** describes how the software communicates with users
- **Design procedure:** transforms structural elements of program architecture into a procedure description of a software component.

Desain Data

Data design is the first of four design activities performed during software engineering. The process of selecting structures in determining the most efficient design as needed.

Purpose

- To get a good data structure so that more effective programs are obtained and reduce the complexity of software development

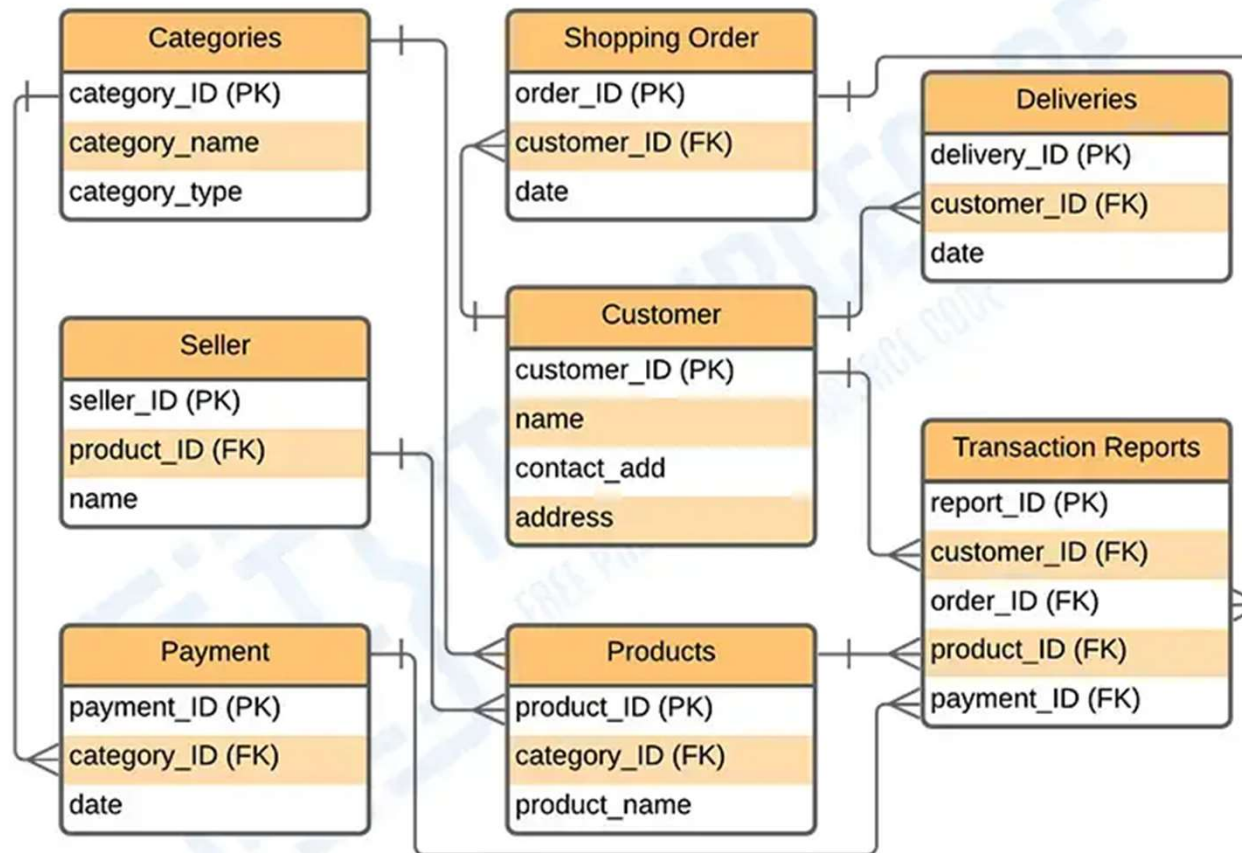
Tools

- Basisdata : ERD (Entity Relationship Diagram)

E-COMMERCE WEBSITE SYSTEM



ERD



Architectural Design

The main goal of architectural design is to develop a modular program structure and represent control relationships between modules. Architectural design is a macro/structural design that reflects the quality as well as functionality of the software. The activity of architectural formation is a decomposition activity, which is dividing software into elements / modules.

Purpose

- To obtain a good architecture and interconnection between modules in the software

Tools

- For object-based software design, you can use Unified Modeling Language (UML) Diagrams

Interface Design (interface)

Interface design provides an overview of the program structure to software engineers (programmers), making it easier to see how the appearance and interaction that occurs in the software.

Interface design focuses on 3 areas:

Design interfaces between modules in software

Design of interfaces between software and external entities

Design of human interface with computer

Tools

- For object-based software design, you can use UML Component Diagrams and UI Diagram tools such as Figma/Canva etc.

Procedure Design

Design procedures are carried out after the completion of the design of data design, architecture, and software interface. This design is used to design how the behavior of a software.

Purpose

- To obtain procedural specifications of existing behavior and; AM Software

Tools

- For object-based software design, you can use Unified Modeling Language (UML) Diagrams

System Modeling

- ❖ Systems modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system.
- ❖ System modeling generally means representing systems using graphical notation (diagrams), such as Unified Modeling Language (UML).
- ❖ Models used during the engineering process need to help get system requirements, to describe the system to the engineer implementing the system, to document the structure and operation of the system

Systems Modeling Perspectives

There are several views (perspectives) in system modeling

-
- External perspective, modeling the context or environment of the system.
- Interaction perspective, modeling the interaction between a system and its environment or between system components.
- Structural perspective, modeling the organization of the system or the structure of the data processed by the system.
- Behavioral perspective, modeling the dynamic behavior of a system and how it responds to a process.

System Modeling with UML

- ❖ UML is widely used in modeling a software system.
- ❖ In its implementation, there are five types of diagrams (from many diagrams ± 13 types of diagrams) that can represent the essence of a system, namely
 - ❖ ***Activity Diagram, a diagram that shows the activities involved in a process or in data processing.***
 - ❖ ***Use Case Diagram, a diagram that shows the interaction between a system and its environment/actors.***
 - ❖ ***Sequence Diagram, a diagram that shows the interaction between actors with the system and between system components.***
 - ❖ ***Class diagram, a diagram that shows the classes of objects in the system and the associations between those classes.***
 - ❖ ***State Diagram, which shows how the system reacts to internal and external events.***

Reference

- Sumber : Sommerville, I., (2011) 9th. *Software Engineering*. Pearson
- Romi Satria Wahono, *Systems Analysis and Design*.

Group Homework

Find DFD, ERD, class diagram, use case diagram, and sequence diagram based on the topic of the assignment you worked on last week. Ensure that the diagrams have features that are almost identical to your assignment from the previous week