Object Oriented Software Engineering (SE313)
Lab Session # 8

Objective: Working with the Component Diagram

Component Diagram

A component diagram, also known as a UML component diagram, describes the organization of the physical components in a system. Component diagrams are often drawn to help model implementation details and verify every aspect of the system's required functions is covered by planned development. In the first version of UML, components included in these diagrams were physical: documents, database table, files, and executables, all physical elements with a location.

In the world of UML 2, these components are less physical and more conceptual stand-alone design elements such as a business process that provides or requires interfaces to interact with other constructs in the system.

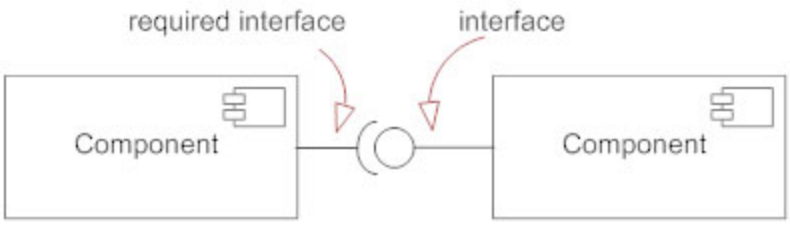
Symbols and Notations

Component

A component is a logical unit block of the system, a slightly higher abstraction than classes. It is represented as a rectangle with a smaller rectangle in the upper right corner with tabs or the word written above the name of the component to help distinguish it from a class.

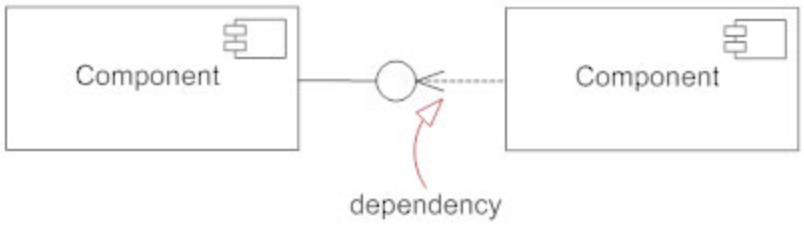
Interface

An interface (small circle or semi-circle on a stick) describes a group of operations used (required) or created (provided) by components. A full circle represents an interface created or provided by the component. A semi-circle represents a required interface, like a person's input.



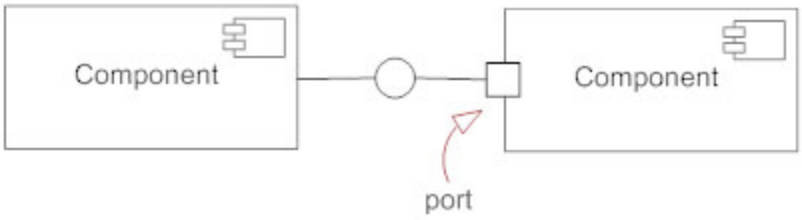
Dependencies

Draw dependencies among components using dashed arrows.

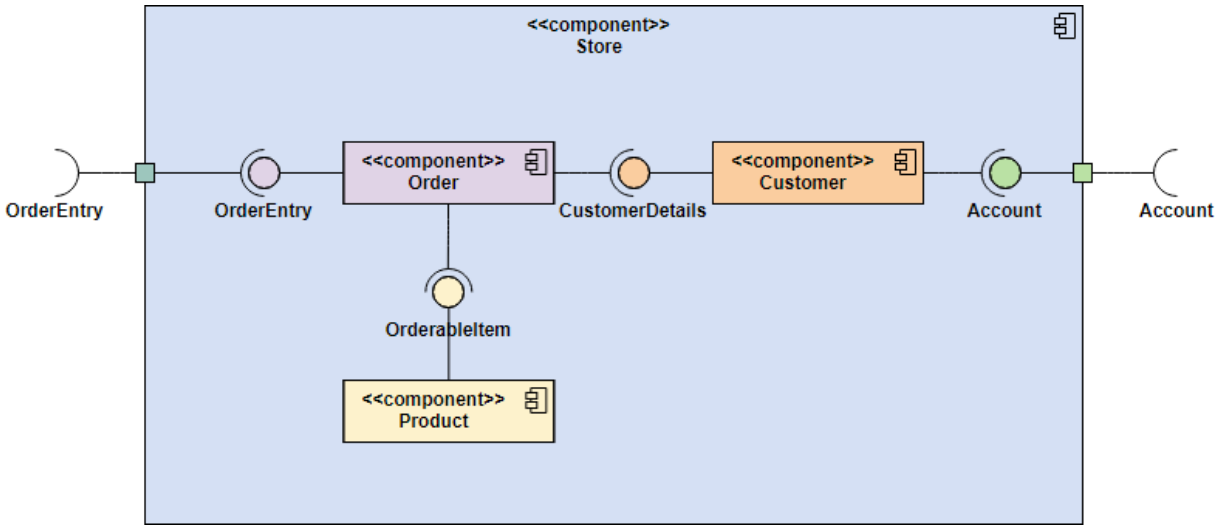


Port

Ports are represented using a square along the edge of the system or a component. A port is often used to help expose required and provided interfaces of a component.



Example: Component diagram of an online store



Class Exercise 1: CHAT SERVER

Draw a component diagram of the below chat server application

Chat server is a standalone application that is made up the combination of two applications namely, Server application which runs on server side and Client application which runs on client side. This application is used for exchanging information in the form of messages. The network for the chat application could be easily setup on LAN.

To initiate a network, the administrator should first start the Server application. User login through the Client application as soon as the Server validates them. The clients thus join the chat room and can now start communicating between each other by sending messages via the Server. A client when logouts leaves the chat room. When all the clients leave the chat room, the administrator may stop the Server application thus closing the network.

Class Exercise 2: New Programming Language

Draw a component diagram of the below application and identify it's all possible connectors

Suppose we have designed a new programming language called *EyeJay*. Our task is to develop a compiler that receives a program and generates machine instructions for a specific processor. The compilation process consists of different steps. Flexible error handling in these steps is of utmost importance; we aim to have error handling strategies that are specific to different stages of the compilation process. To this end, we have decided to split the compilation process into three stages: front end, middle end, and back end.

The front end reads the source file, parses it, and converts it into an abstract syntax tree. The front end performs semantic analysis on the constructed abstract tree, augments it with additional information and passes the augmented tree to the middle end.

The purpose of the middle end is to generate code in an intermediate language and to perform optimization on the generated code. The optimization phase consists of different steps. The architecture should offer design flexibility in the optimization phase such that we can add, remove, or reorder the optimization steps in different releases of the compiler.

The back end receives the optimized code in the intermediate language and translates it to machine instructions for a specific processor.

Draw UML Component diagrams of the below student exercise (Identify the components and its connectors)

Students Exercise 1 "HOTEL MANAGEMENT SYSTEM"

Introduction

The customer needs in hotel are enquiry about rooms, reservation process, vacating process, canceling the reservation, and the restaurant for food items.

Description:

1. In this system the first module gives the customer view for lodging, The customer needs enquiry for first process.
2. The next process of customer is reservation. This reservation may be for today or future, for this the reservation form can work more efficient than the manual process.
3. This form also checks the rooms for reservation dates. It is interactive to the customer, if the rooms are not available it gives message and not allows reserving for that date.
4. Vacating, Billing and other services like restaurants.
   The next process for the customer is vacating the room. For this he needs the total bills. Also he uses the restaurant for food, that information also we need in the bill. The vacating form works for this information effectively. This form calculates the user bills of room and restaurant; it is minus by the advance and shows the total bill he need to pay.
5. The cancellation process is the next process of the customer. For this the cancellation form works effectively. The customers reserved for future needs the cancellation process. That time we need to calculate the refund amount from the advance, For this, this form works efficiently.
6. The next module gives the restaurant information for the customer. The customer gives order from room and also spot order. For the room order and the spot order the customer having same process, only one difference is for the room order he need to select the id, and for the customer order he need to enter the name.
7. The last module is reports of the hotel management system. It is having all details of the database.

.

Students Exercise 2 "HOSPITAL MANAGEMENT SYSTEM"

Admissions:

This Module helps in registering information about patients and handles patient's query. A unique ID Is generated for each patient after registration This helps in implementing customer relationship management and also maintains medical history of the patient.

Doctor Appointments:

This module deals with, when the ID is generated the patient receives the Appointment time and number from the Receptionist and accordingly visit the doctor.

Tests Appointments:

This module deals with, when the ID is generated the patient receives the appointment time and number from the receptionist and accordingly undergoes the tests.

Bed Allotment:

This module handles with allotting the Bed to various patients by checking their ID.

Undergo Operation:

This module handling with undergoes the various operations by diagnosing the patients.

Authentication:

This module checks whether the person is a valid registered Doctor/Staff who can handles various activities such as other modules operations.

Draw Salary:

This module checks whether the person is a Doctor/Staff and draws salary based on the information

Manage Doctor/Staff:

This module handles the activities such as adding/deleting/editing Doctor/Staff Information into the database.

Prescribe Tests:

This module handles various activities such as Doctor Diagnoses the patient, gives treatment and gives suggestions to the patients, and prescribes laboratory tests and medicines.

Wardwise Bed Status:

This module takes care of medical equipment, doctor visit, vitals recording, patient case sheet, diet ordering, blood requisition, transfer intimation and discharge intimation etc. It also deals with the maintenance of the wards, inter-and intra-ward transfers.

Admission/Discharge Reports:

This module helps in generating patient's discharge summary, which includes patient's health at the time of discharge, medical history, various diagnosis and drug prescriptions, history of present illness and course in hospitals.

Patient Information:

This module helps in generating the patient information which is provided by doctor.