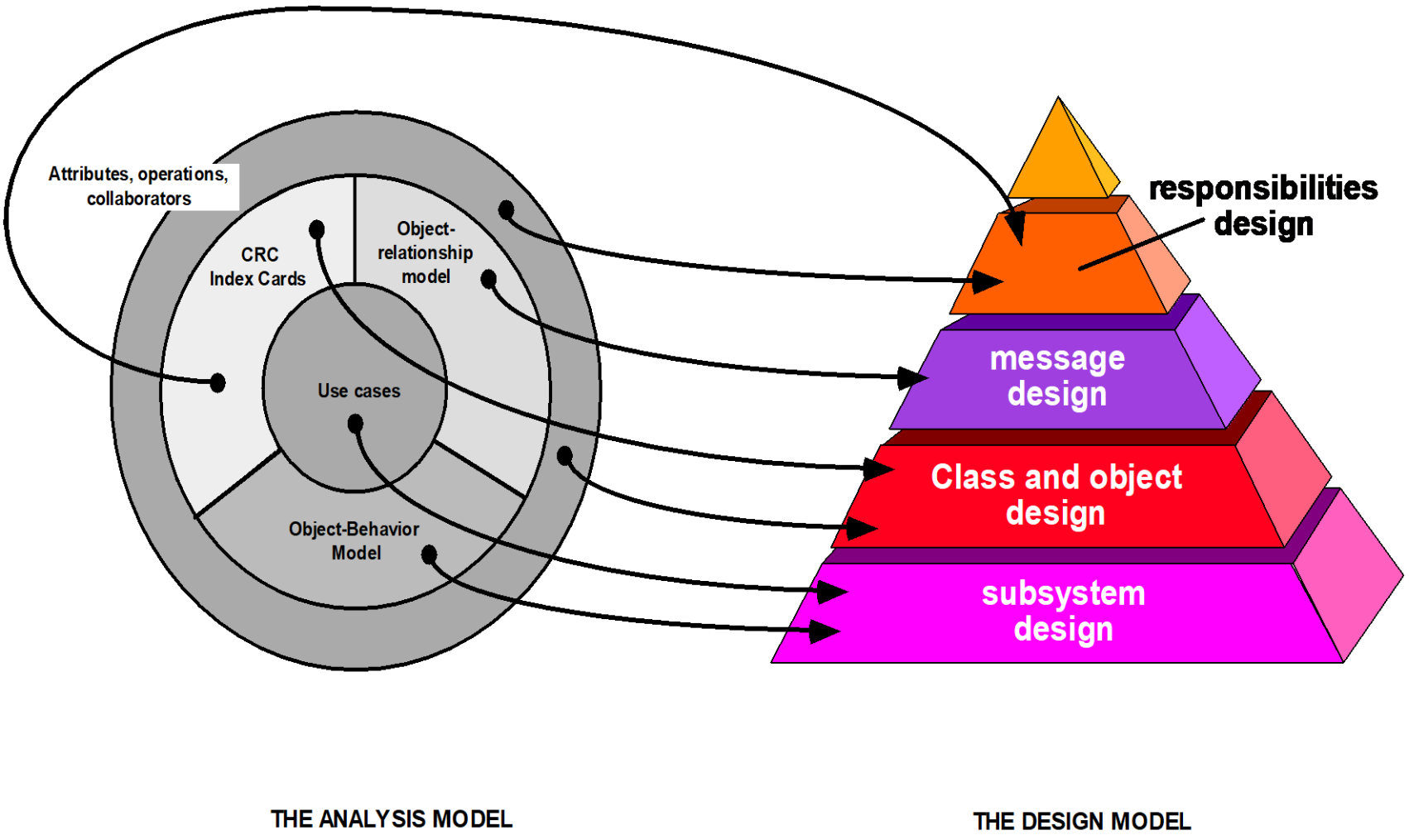
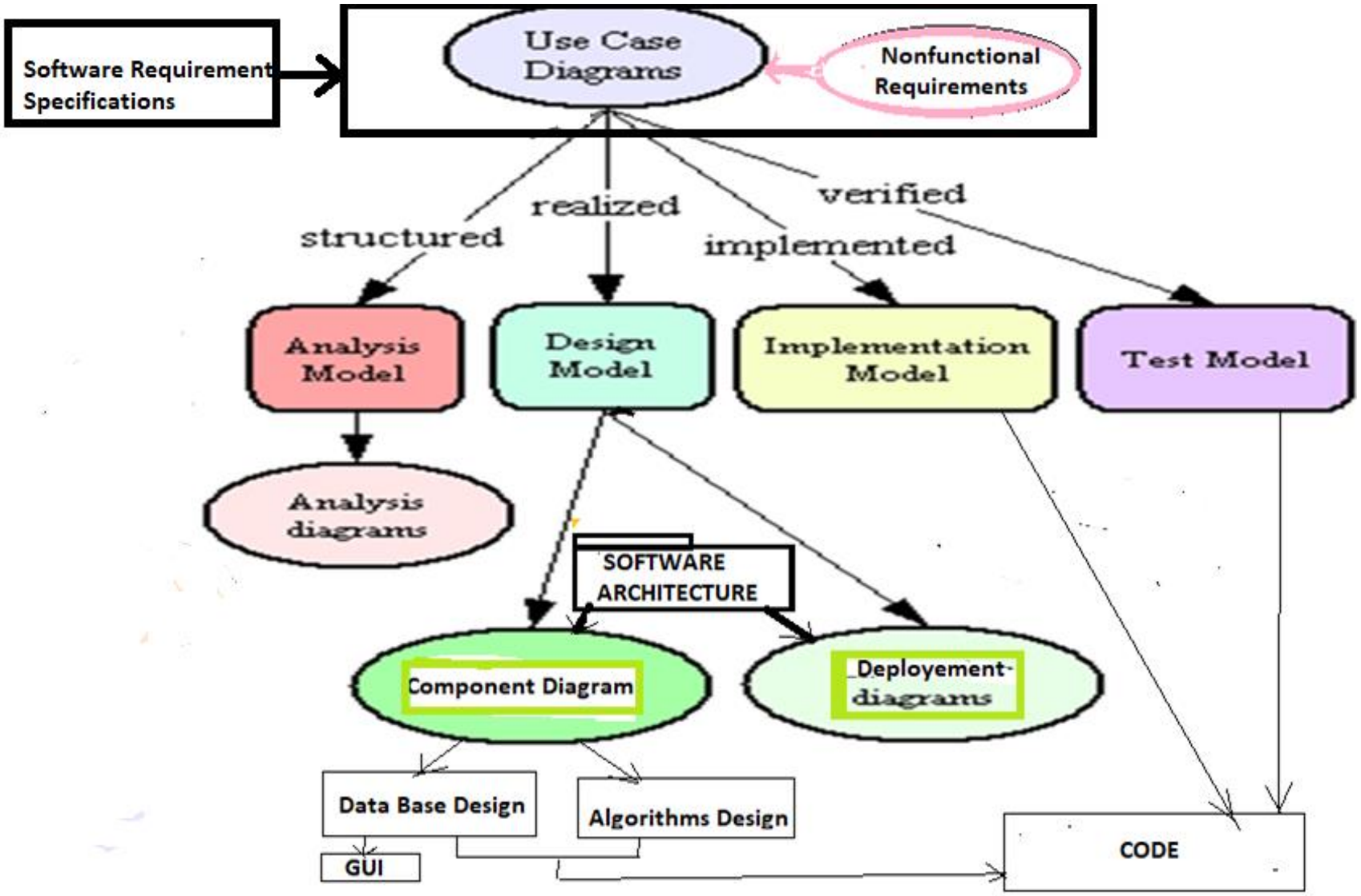
Program= Data Structure+ Algorithms

"The ideal architect should be a man of letters, a skillful draftsman, a mathematician, familiar with historical studies, a diligent student of philosophy, acquainted with music, not ignorant of medicine, learned in the responses of jurisconsults, familiar with astronomy and astronomical calculations."
— **Vitruvius**

OOA and OOD


Attributes, operations, collaborators
CRC Index Cards
Object-relationship model
Use cases
Object-Behavior Model
responsibilities design
message design
Class and object design
subsystem design
THE ANALYSIS MODEL
THE DESIGN MODEL

SOFTWARE ARCHITECTURE DESIGN



Non functional requirement : Information Security : No scope here

SOFTWARE ARCHITECTURE DESIGN

Software Architecture = Component Diagram + Deployement Diagram

Component Diagram = Data Structure+ Algorithms

Data Structure+ Algorithms => CODE

Program= Data Structure+ Algorithms

OO Design

1.Software ARCHITECTURE (Component Diagram)

2.DATA STRUCTURE DESIGN

Name Of The Table And Fields/Data Dictionary.

3.ALGORITHMS DESIGN: from use case and activity diagram

4.GUI DESIGN

Software Architecture

- Software Architecture show static, implementation view of system using source code structure and runtime implementation structure
- Software Architecture has two forms: Component Diagram and deployment Diagram
- Component diagrams shows structure of the software elements like packages and subsystems and subroutines and their interaction with data structures or data base tables.
- The component diagram known as Logical software architecture can be organized into units such as layers known as presentation layer, application layer and persistent layer.
- The user interacts with presentation layer known as graphical user interface. The presentation layer makes request to the application layer which contains application logic or business logic. Application layer loads and access databases from the persistent layer.

Where to use Component Diagrams?

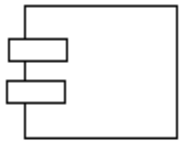
- The component diagram is a special purpose diagram, which is used to visualize the static implementation view of a system. It represents the physical components of a system, or we can say it portrays the organization of the components inside a system.
- The components, such as libraries, files, executables, etc. are first needed to be organized before the implementation.

The component diagram can be used for the followings:

- To model the components of the system.
- To model the schemas of a database.
- To model the applications/ALGORITHMS of an application.
- To model the system's source code.

COMPONENT DIAGRAM 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 <component> written above the name of the component to help distinguish it from a class.

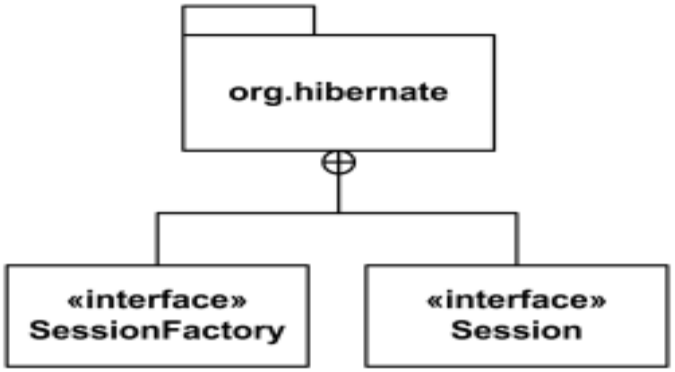
Package

A package is used to group elements, and provides a namespace for the grouped elements. A package is a namespace for its members, and may contain other packages.

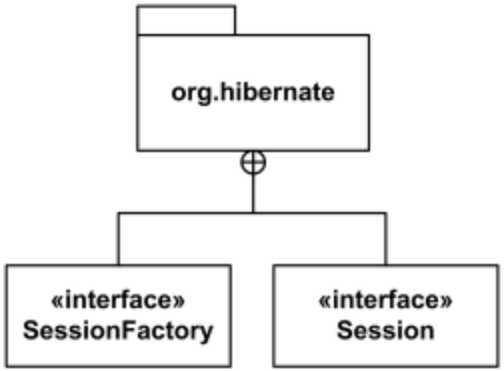
Package

A package is rendered as a tabbed folder - a rectangle with a small tab attached to the left side of the top of the rectangle. If the members of the package are not shown inside the package rectangle, then the name of the package should be placed inside.

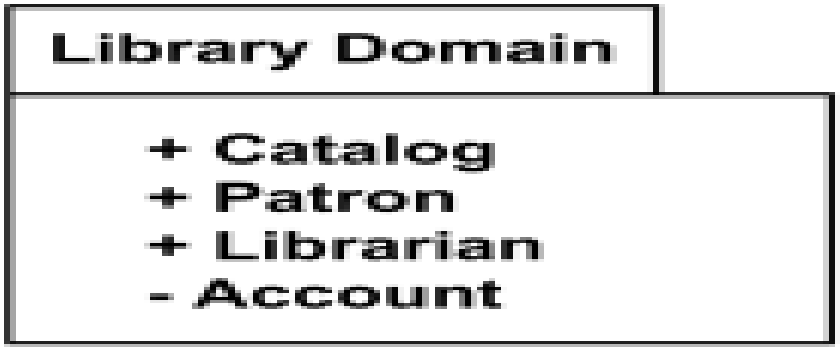
A diagram showing a package with content is allowed to show only a subset of the contained elements according to some criterion.



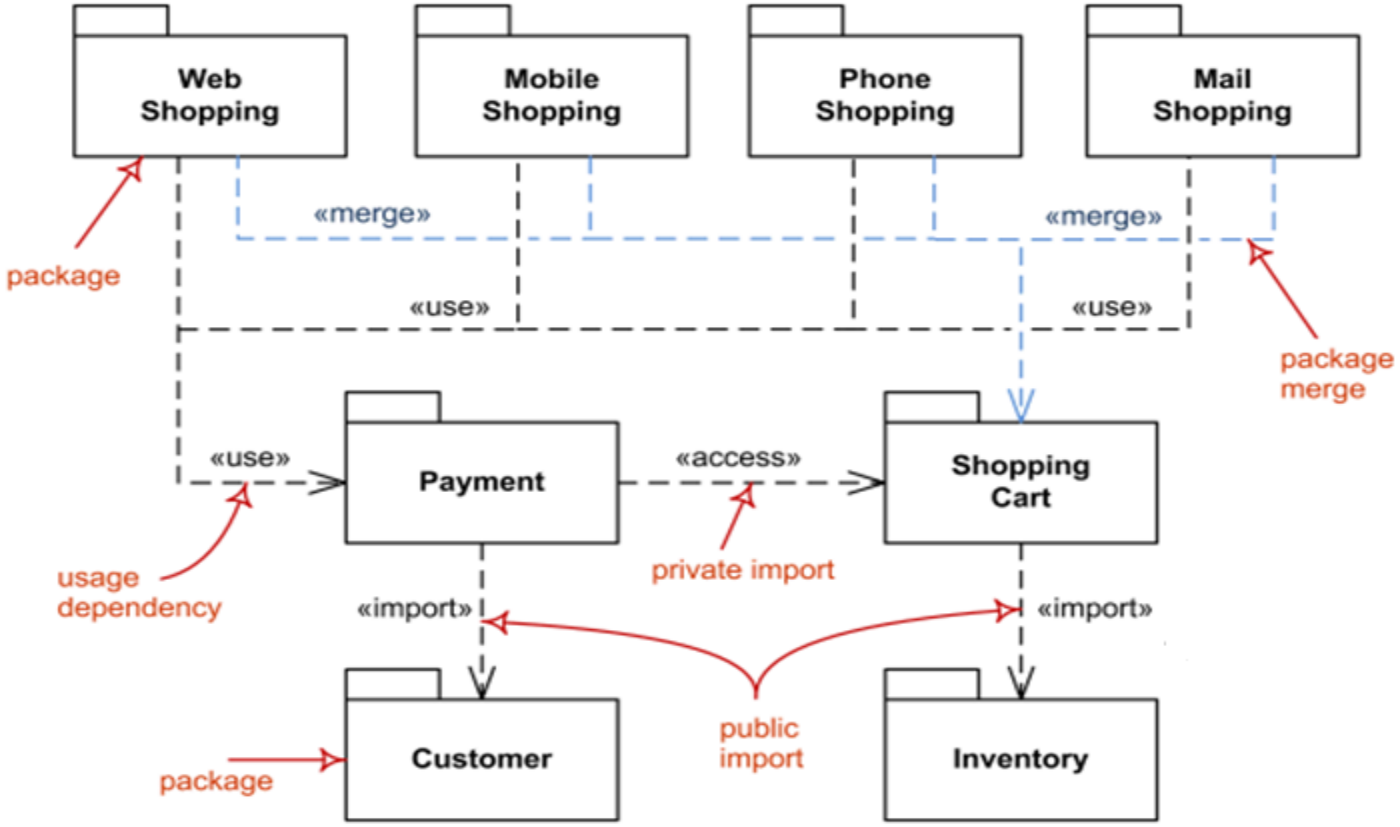
The members of the package may be shown within the boundaries of the package. In this case the name of the package should be placed on the tab.



- *All elements of Library Domain package are public except for Account*
- The public elements of a package are always accessible outside the package through the use of **qualified** names.



Some major elements of the package diagram are shown on the drawing below. Web Shopping, Mobile Shopping, Phone Shopping, and Mail Shopping **packages** **merge** Shopping Cart package. The same 4 packages **use** Payment package. Both Payment and Shopping Cart packages **import** other packages.



COMPONENT DIAGRAM NOTATIONS

Component

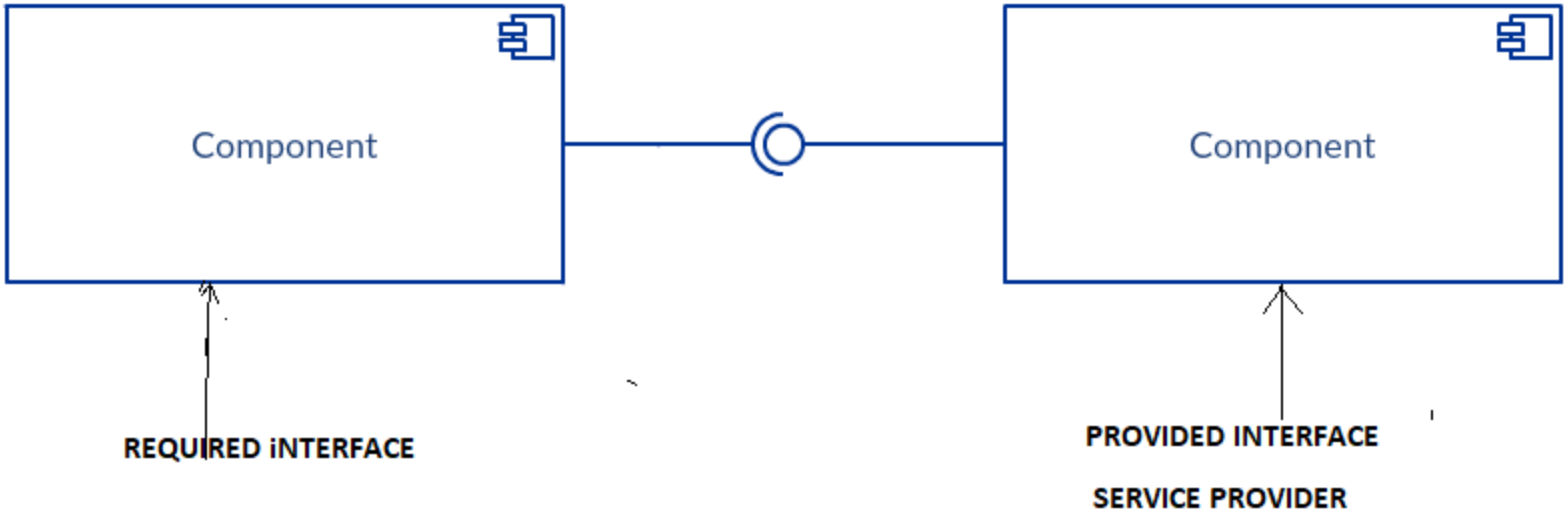
There are three ways the component symbol can be used.

1) Rectangle with the component stereotype (the text <<component>>). The component stereotype is usually used above the component name to avoid confusing the shape with a class icon.

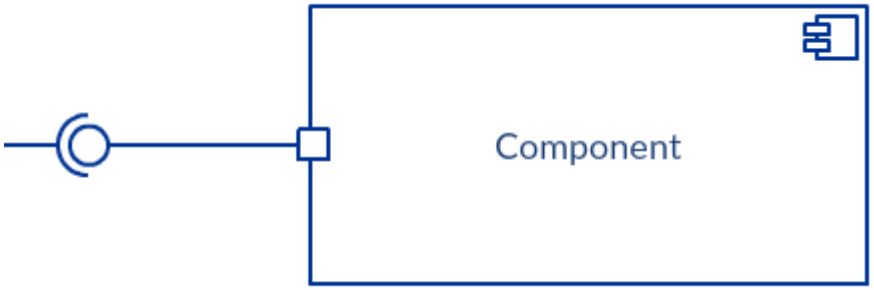
2) Rectangle with the component icon in the top right corner and the name of the component.

3) Rectangle with the component icon and the component stereotype.

Provided Interface and the Required Interface



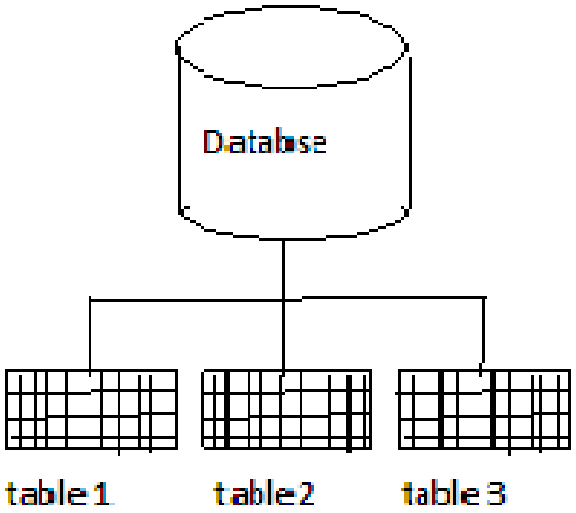
Port



Dependencies

Model Dependencies From Left To Right (Horizontal)
Place Child Components Below Parent Components (vertical)

Although you can show more detail about the relationship between two components using the ball-and-socket notation (provided interface and required interface), you can just as well use a dependency arrow to show the relationship between two components.

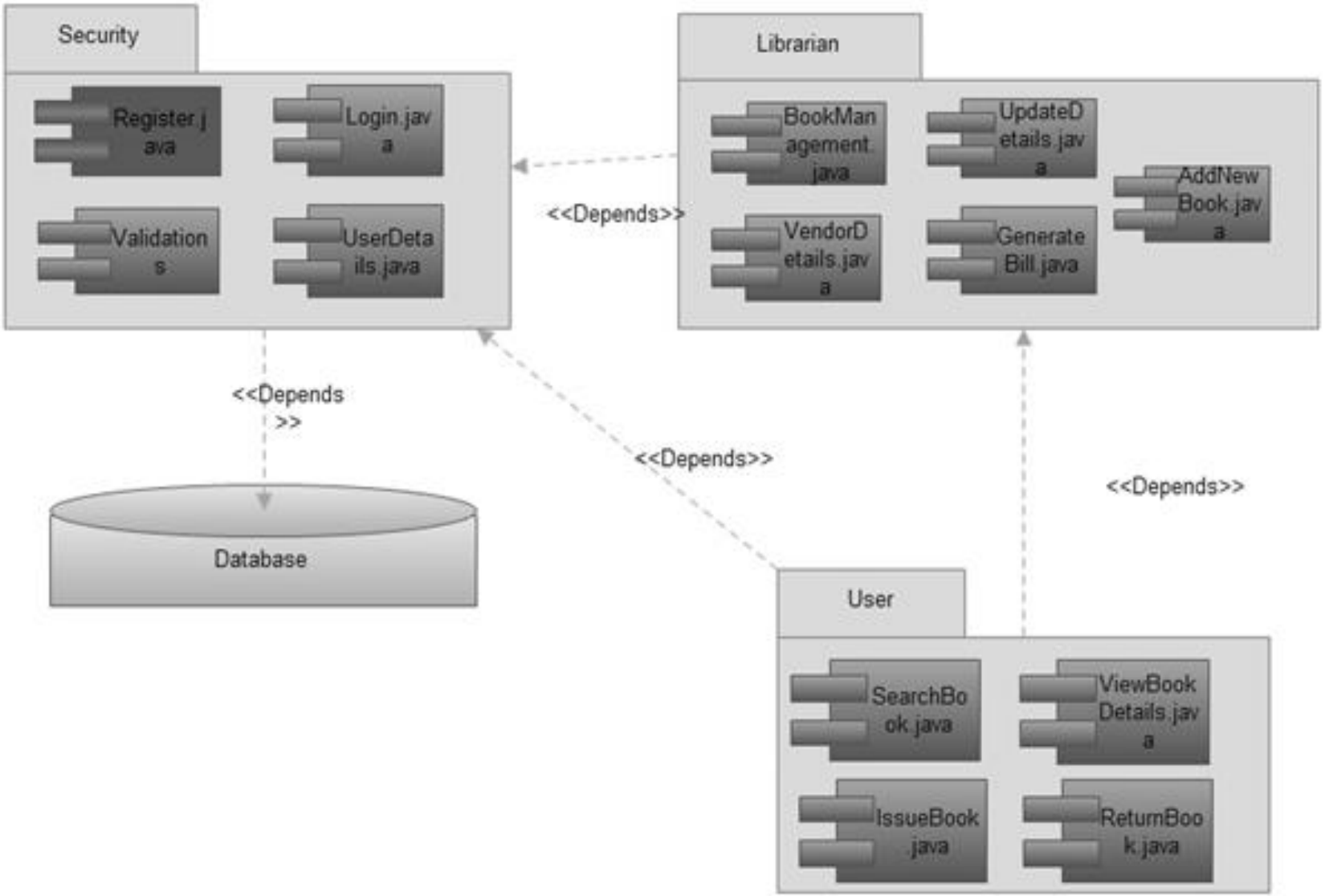


Data base represent the data structure requirement or table required for the implementation of the software.

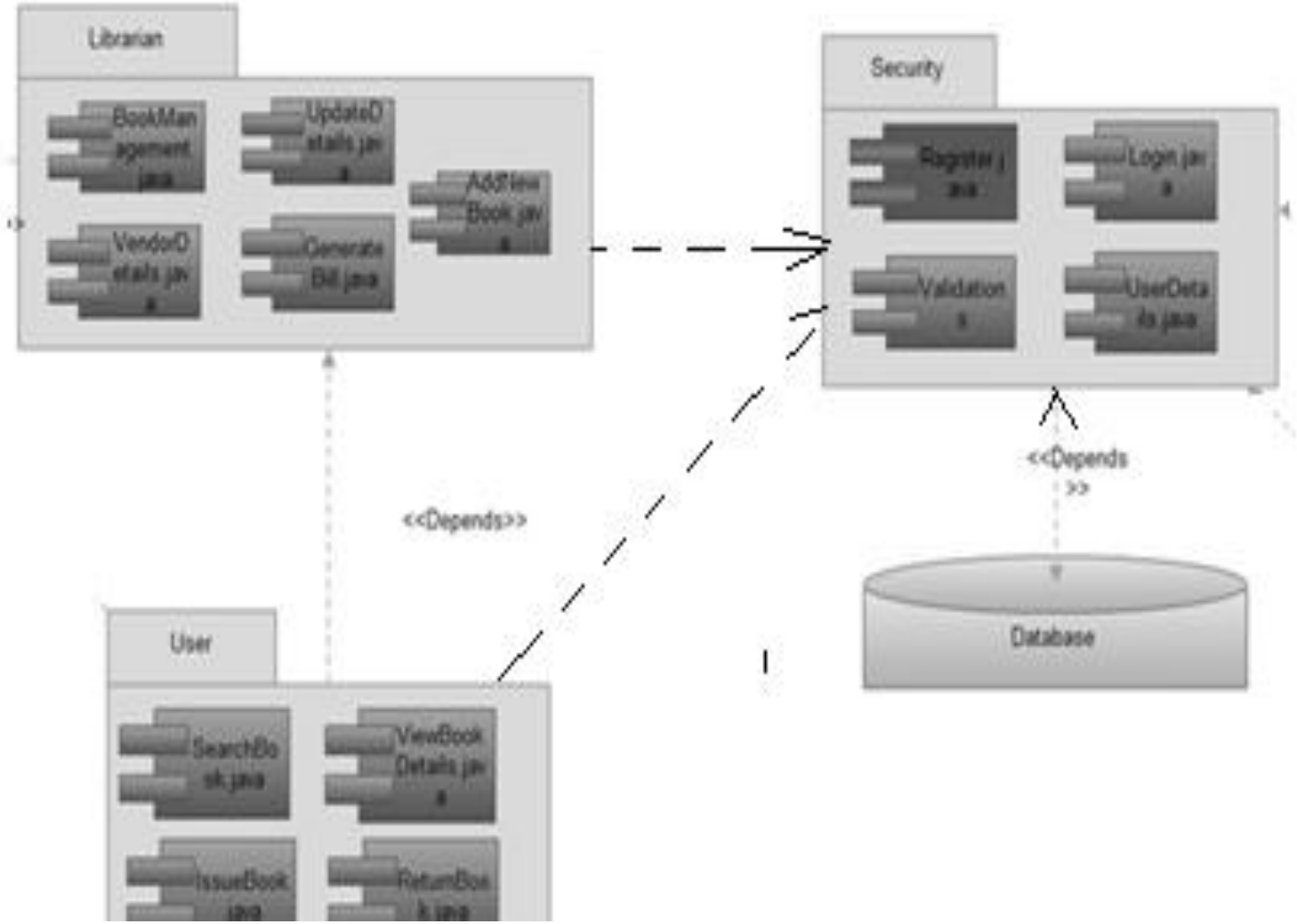
```
/*import java.awt.event.*; // * signifies all classes in this package are imported import javax.swing.JFrame // here only the JFrame class is imported */
/*Include in c++ */
package p1;
 class c1()
{
 public void m1()
{
System.out.println("m1 of c1");
 }
public static void main(string args[])
{
c1 obj = new c1(); obj.m1();
}
}
```

```
package p2;
 import p1.*; //imports classes only in package p1
{
 public void m3()
{
System.out.println("Method m3 of Class c3");
 }
 public static void main(String args[])
{
c1 obj1 = new c1(); obj1.m1();
 }
}
```

:Component Diagram for Online Bharatratna Dr Babasaheb Ambedkar Library Management System



CORRECT


Librarian
BookMan agement java
UpdateD etails jav a
AddNew Book jav a
VendorD etails jav a
Generate Bill java
Security
Register java
Login jav a
Validation s
UserData ils java
<<Depends>>
<<Depends>>
Database
User
SearchBo ok java
ViewBook Details jav a
IssueBook java
ReturnBoo k java

Design of the required cpp files

- The browser will display the user interface files:
- Login.cpp
- LibrarianUI.html
- UserUI.cpp
- Client node is connected to the web server via HTTP. Web server hosts all servlet files:
- searchBook.cpp
- issueBook.cpp
- returnBook.cpp
- calculateFine.cpp

- The web server accesses data from the Library Database server via JDBC connection or file handling.

map component diagram with code

1. How to Map Software Architecture (component diagram) with code ?

1.1.Map Data Structure with code-GUI and TABLES

1.2 Map Algorithms with code

Deployment Implementation

1. Then Deploy the code on designed deployment diagram.
2. Make the real life deployment environment for loading the systems programs and developed software.
3. Then Test It .

Object Oriented Design

- This section describes data structure design theory and software Architecture Design, algorithm design and deployment environment for the object oriented applications.

- **Class Design Theory**

- Object oriented Data Structures are intended to capture behaviour. To this end, the relevant institution is that classes have attached to them a set of messages, which are specified in the object oriented schema via signatures, and which are implemented as methods. In addition, behaviour can be inherited by subclasses and message names can be overloaded (polymorphism) ie. reused in various context. Therefore Class can be defined as
- $C = (C_{STRUCTURE} , C_{BEHAVIOR})$
- Where structural schema is a named quadruple of the form :
- $CLASS_{STRUCTURE} = (C_C , T_C , IS\text{-}A)$ where
- C_C is a finite set of class names of the application to be implemented,
- T_C is a finite set of data types attributes in the class.
- IS-A is a partial order on C with respect to sub typing or other relationships.
- Notice that implementations typically add a number of additional features not included in $C_{STRUCTURE}$, like attributes , a distinction of class attributes from instance attributes, a distinction between private , and public attributes, a different set of constructors , an explicit inclusion of distinct types of relationships between classes and their objects, ,Integrity constraints like object constraints, class constraints and data structure constraints.

Behavior schema

- **Behavior schema** is a named five tuple of the form:
- $C_{BEHAVIOR}$ = (C_C , M_e , M , message, implementation)
- C_C is a set of class names
- M_e is a set of message names,
- M is a finite set of methods or programs.
- Message: message passing mechanism through collaboration among objects.
- Implementation: {(m,c) | m ∈ message(c)} → M is a partial function.
- Notice that m ∈ M and c ∈ C .

Data Structure Design for case study

The database(data structure) contains the following classes or files or data base tables:

- User(user_id, username, password)
- Book(bookid, bookname, authorname, publication, number of copies)
- BookIssue(user_id, book_id, issue date, due date, return date, fine)

- **Algorithms Design**
- Now we can map the software architecture with the algorithms used for the implementation of the applications.
- Some of the sample algorithms are given below:

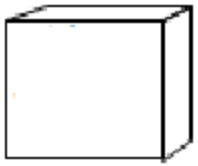
Algorithmn Calculate fine: librarian calculates fine for the book returned
Precondition: book is returned by user
Post condition: fine amount is calculated
Algorithm:

1. Start
2. Userid←enter user id
3. Bookid←enter id of book to be returned
4. Duedate←fetch due date of book
5. If(return date> due date)
6. Fine=(return date – due date)*rate
7. Else
8. Fine=0
9. End if
10. Return fine
11. Stop

Deployment Diagram DESIGN

Deployment diagrams shows structure of the run-time system having hardware and Systems software requirements to execute the software architecture (Application Program written by the programmer). The deployment environment consists of client machine, web server, application server and database server with their hardware and systems software requirement depending on the requirements of the user program. You can specify the hardware requirements of each machines and the systems software such as databases, compilers. Operating systems loaded on each machine.

Processor/machine



Processor indicate the machine required for the execution of the program. You can specify the hardware and software requirements of this machine

Device

Device indicates the devices attached to the machine.

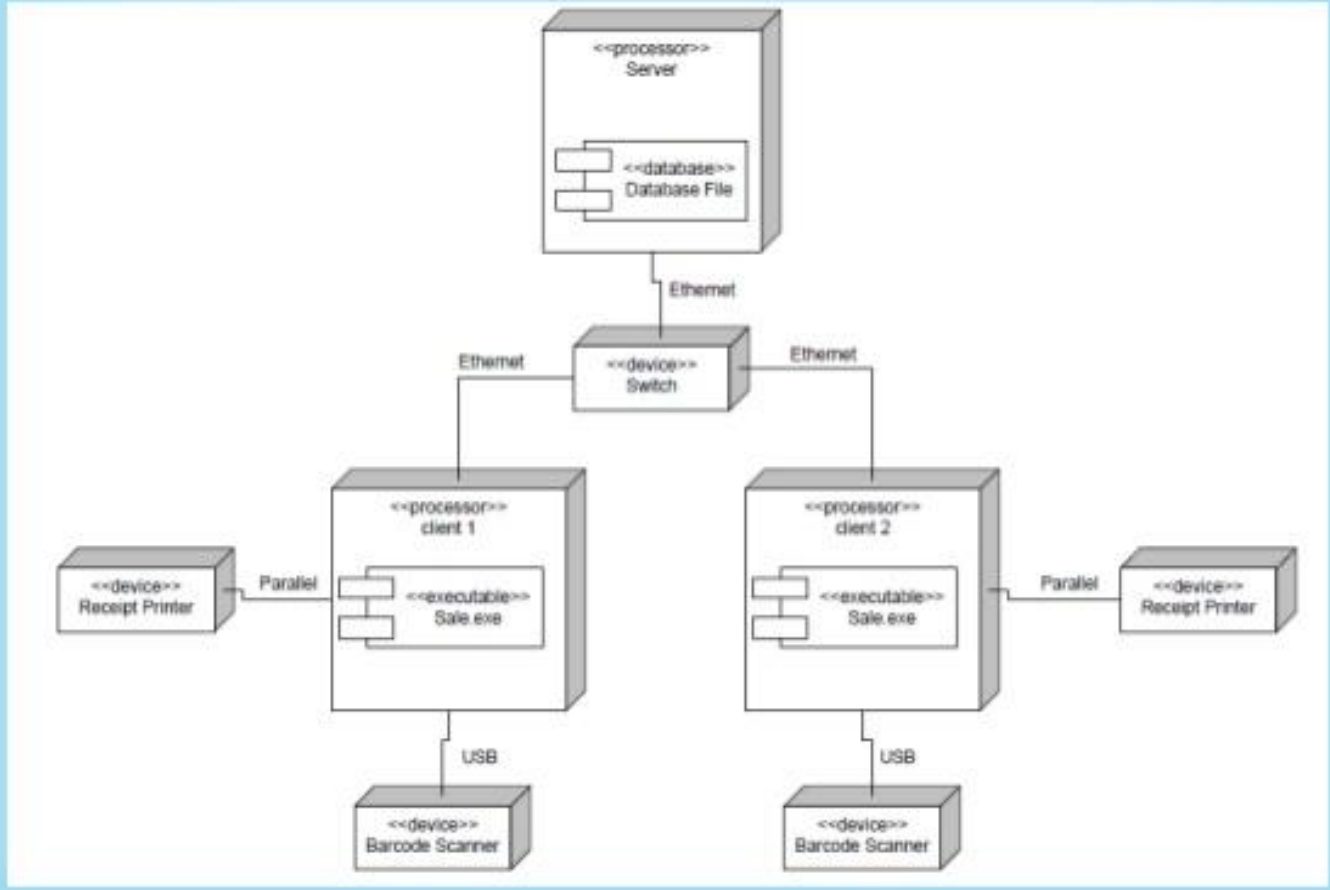
Communication Line

———

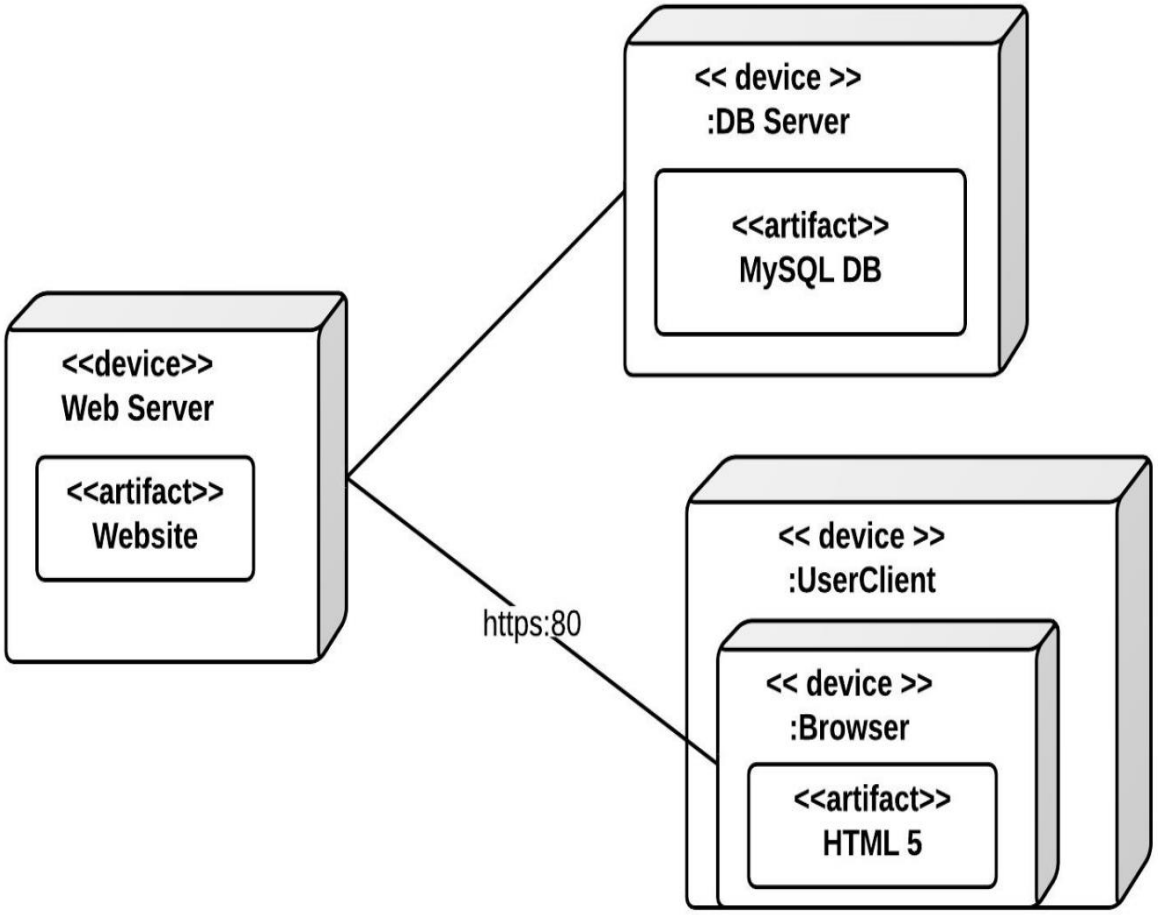
The communication line is useful for communication among the machines and devices.

Client server Deployment

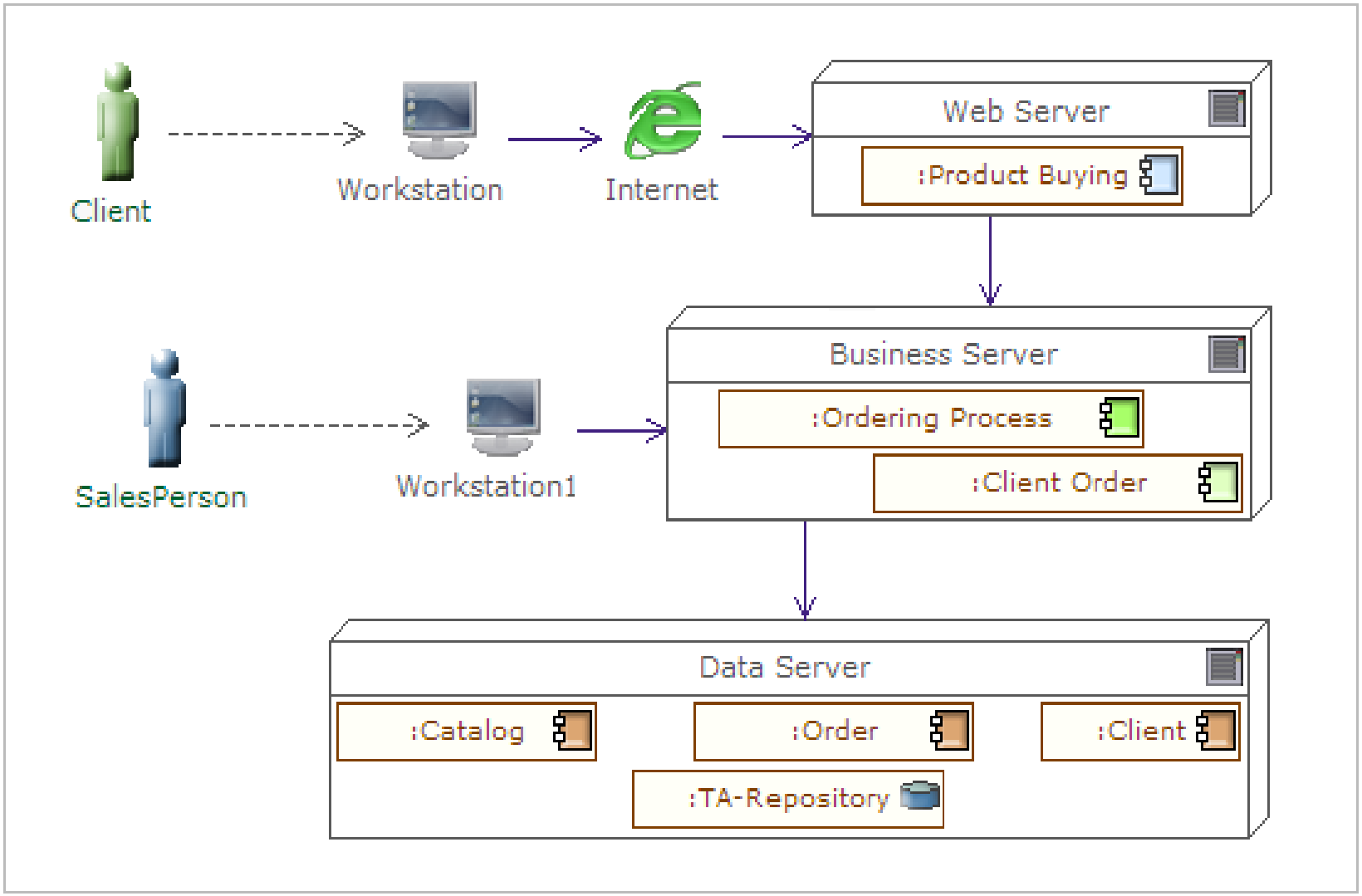
Deployment diagram
(An Example)



WEBSIDE DEPLOYEMENT



ORDER PROCESSING SOFTWARE DEPLOYEMENT



Deployment Environment-Hardware and Software Requirement

1 Hardware Configuration,

2.Operating Systems

3.Data base Server (hardware & software)

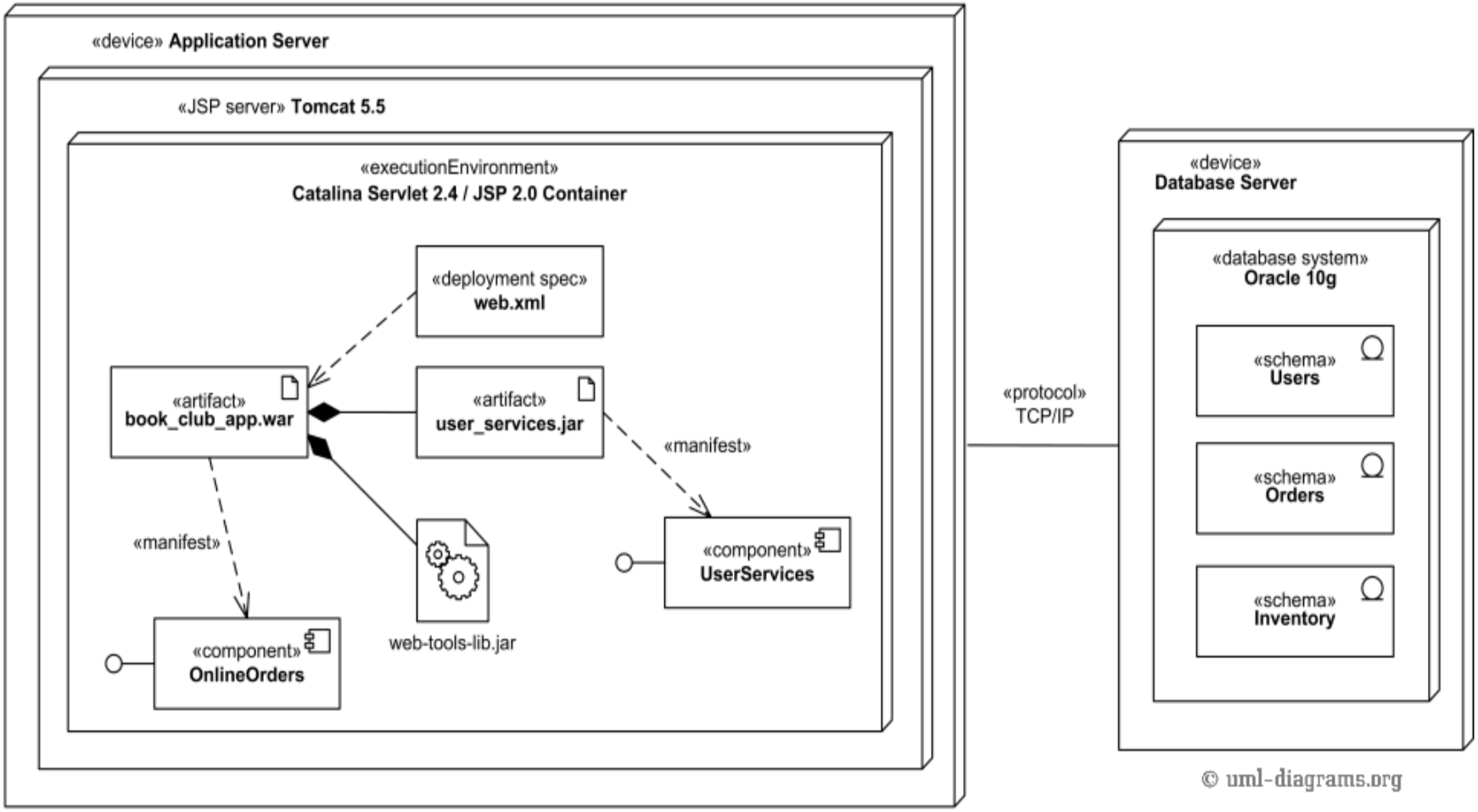
4.Web server (hardware & software)

5.Compiler or programming language used (development environment),

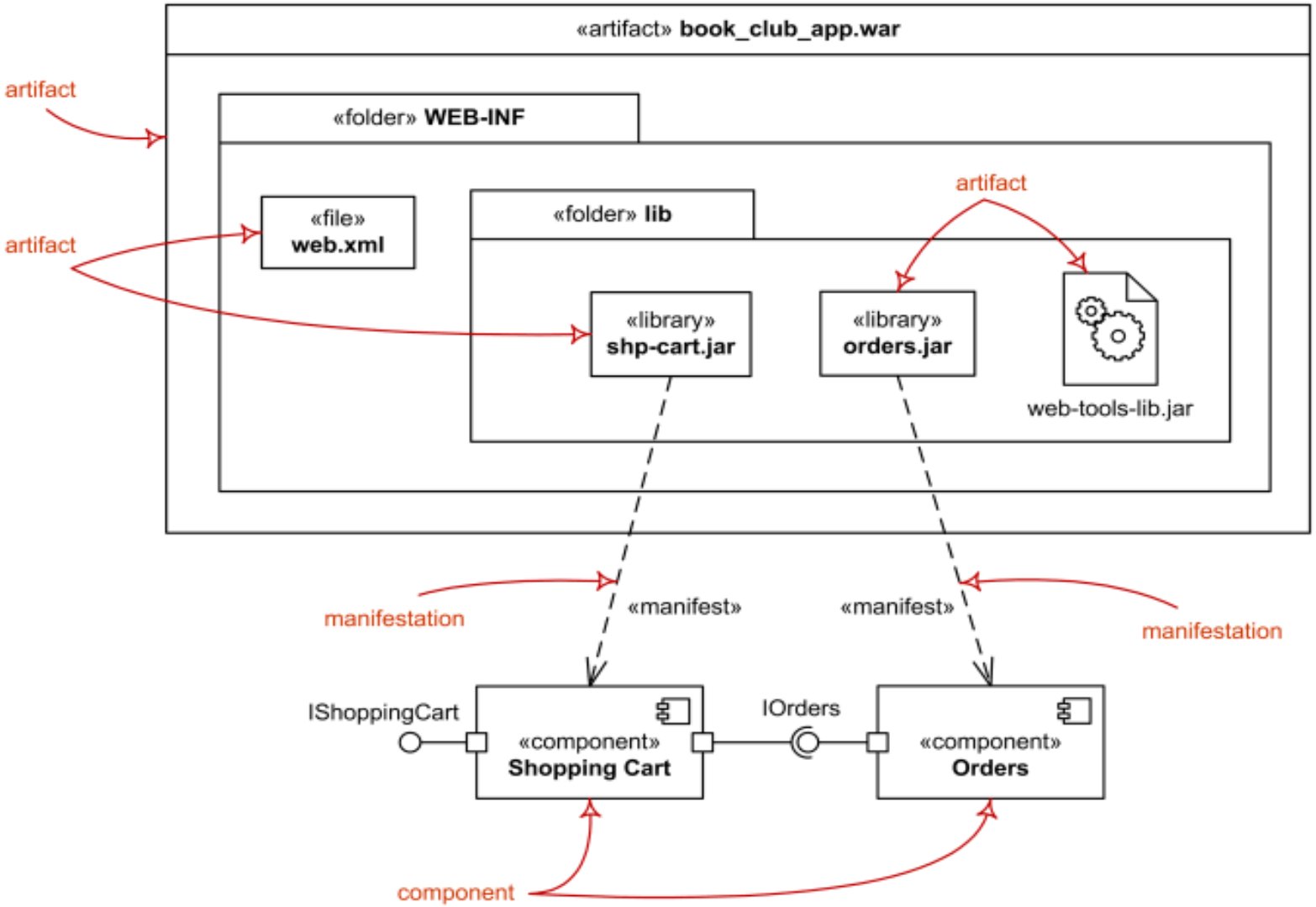
6.Web browser

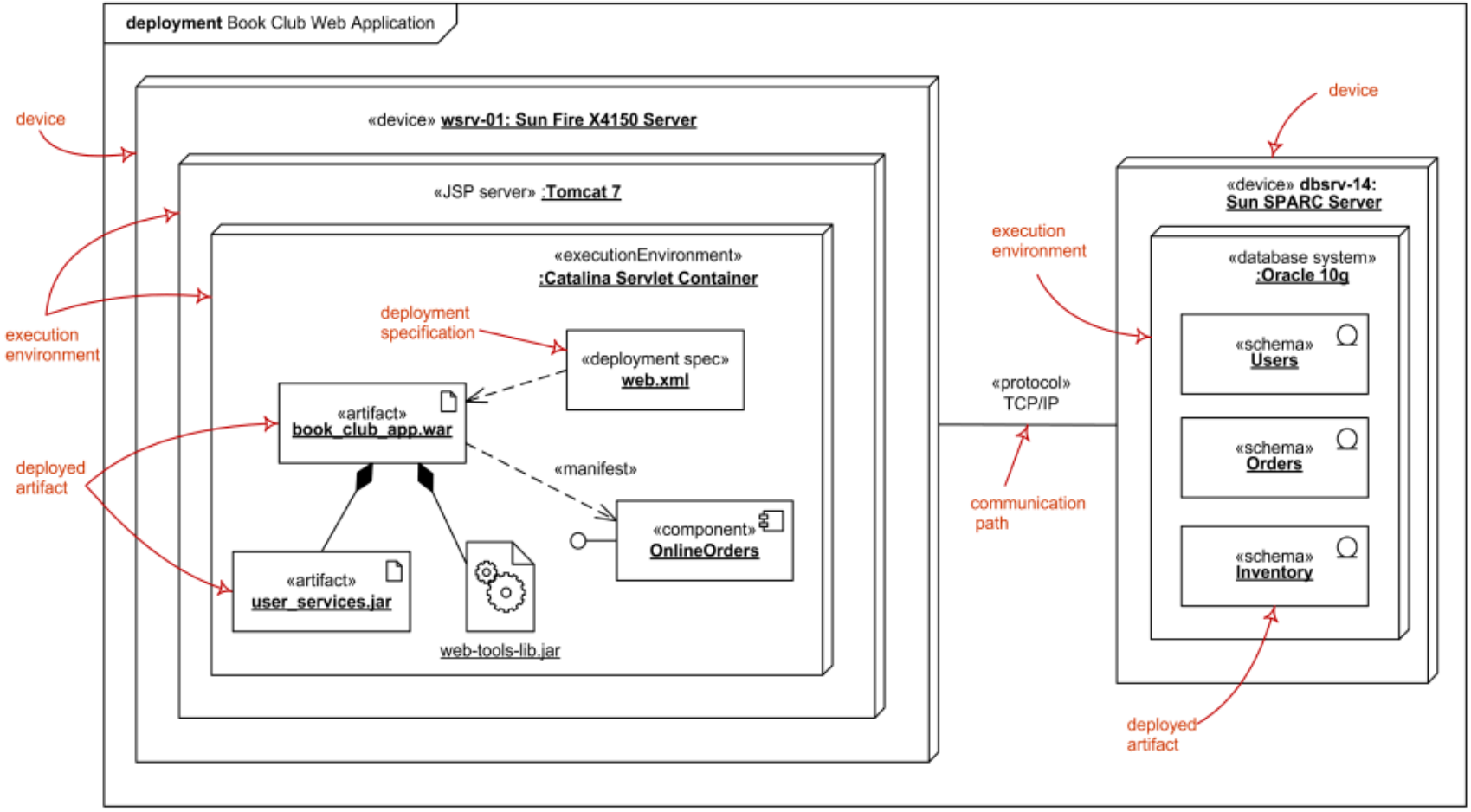
7. software Tools used

UML *Deployment Diagram for* web application .



Specification Level Deployment Diagram




deployment Book Club Web Application
device
«device» wsrv-01: Sun Fire X4150 Server
«JSP server» :Tomcat 7
«executionEnvironment»
:Catalina Servlet Container
execution environment
deployment specification
«deployment spec»
web.xml
«artifact»
book_club_app.war
«manifest»
deployed artifact
«component»
OnlineOrders
«artifact»
user_services.jar
web-tools-lib.jar
«protocol»
TCP/IP
communication path
device
«device» dbsrv-14:
Sun SPARC Server
execution environment
«database system»
:Oracle 10g
«schema»
Users
«schema»
Orders
«schema»
Inventory
deployed artifact

DEPLOYEMENT DIGRAM DESIGN

MACHINES	SOFTWARE	H/W OR CONFIGURATION	OPERATING SYSTEMS	COMPILER	ANY OTHE SOFTWARES	SOFTWARE MODULES AND PATH
CLIENT			WINDOW 10	JAVA	Google Chrome 67	Gui;
WEB SERVER	Apache Web Server/ (IIServices Web Server)	**Sun Fire X4150 Server**	**Red Hat Enterprise Linux 5** Windows, and Mac OS X, Apple OS X (Microsoft Windows)	**JAVA** PHP, Perl, and Python/ (ASP.NET)	with IIS	PATH D:/>Product buying

DEPLOYEMENT

MACHINES		CONFIGURATION/MACHINE NAME/ MAKE	OPERATING SYSTEMS	COMPILER	ANY OTHE SOFTWARES	SOFTWARE MODULES AND PATH
APPLICATION SERVER)	Apache Tomcat 9 - 64 bit		WINDOW, LINUX	HTML,JAVA, JSP, JAVA SCRIPT/Oracle Application Server	Mozilla Firefox 61	Path: c:> CLIENT ORDER

MACHINES	CONFIGURATION/MACHINE NAME/MAKE	OPERATING SYSTEMS	COMPILER	ANY OTHE SOFTWARES	SOFTWARE MODULES AND PATH
DATA BASE SERVER	Sun SPARC Enterprise servers OR Sun SPARC Enterprise M9000	LINUX	ORACLE 10 DBMS		tables
DEVICES-PRINTER, SCANNER , CAMERAAND THE LIKE					

Sun SPARC Enterprise M9000-64 server sets a new standard for performance and configuration flexibility in large symmetric multiprocessing platforms

Hardware Requirements for Web and Database Servers

ITEM	Web server (minimal)	Web server (recommended)	Combined Web & Database Server (minimal)	Combined Web & Database Server (recommended)
Processor	1,6 GHz CPU	2 x 1,6 GHz CPU	2 x 1,6 GHz CPU	4 x 1,6 GHz CPU
RAM	1,75 GB RAM	3,5 GB RAM	3,5 GB RAM	7 GB RAM
HDD	1x 40 GB of free space or more is recommended for the webshop data (non-system drive is preferred) 1x 40 GB of free space or more is recommended for the software that is listed in the software requirements (system drive)			
Recommended Microsoft Azure Virtual Machine Configuration	Basic Small VM	Basic Medium VM	Basic Medium VM	Basic Large VM

Sun SPARC Enterprise servers

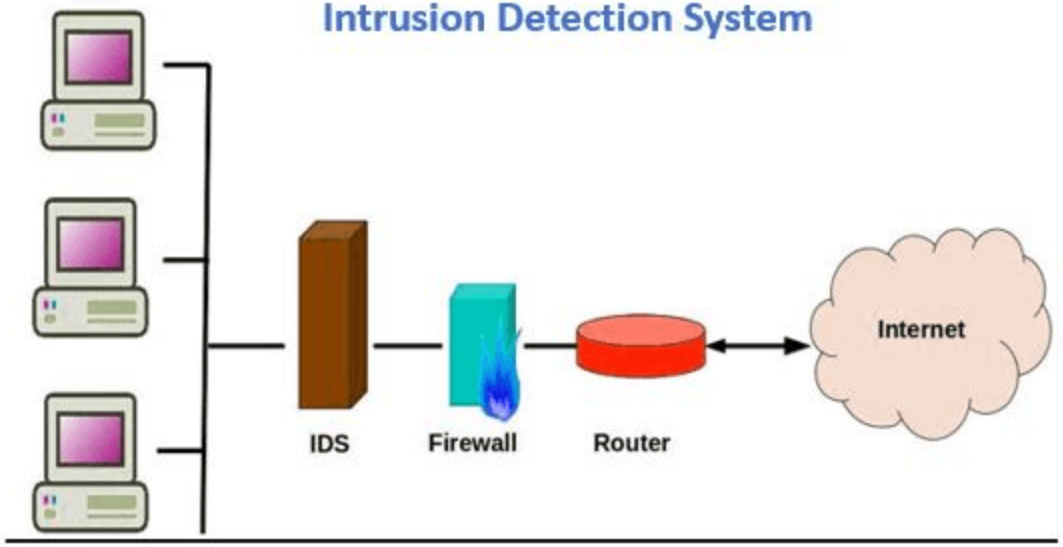
Oracle’s extensive server product line. It supports up to 64 dualcore SPARC64 VI 64-bit processors (up to 128 cores), or up to 64 quad-core SPARC64 VII 64-bit processors (up to 256 cores), or combinations of SPARC64 VI and SPARC64 VII processors. It also features 2 TB of memory; 128 hot-swappable PCI Express (PCIe) I/O slots; up to 64 internal disks; a new high-performance interconnect capable of up to 737 GB/sec; and support for external I/O, up to 24 domains, and thousands of software applications for the Oracle Solaris operating system (OS). Exhibiting 7.5 times greater system bandwidth than previous generations and scalability in every dimension,

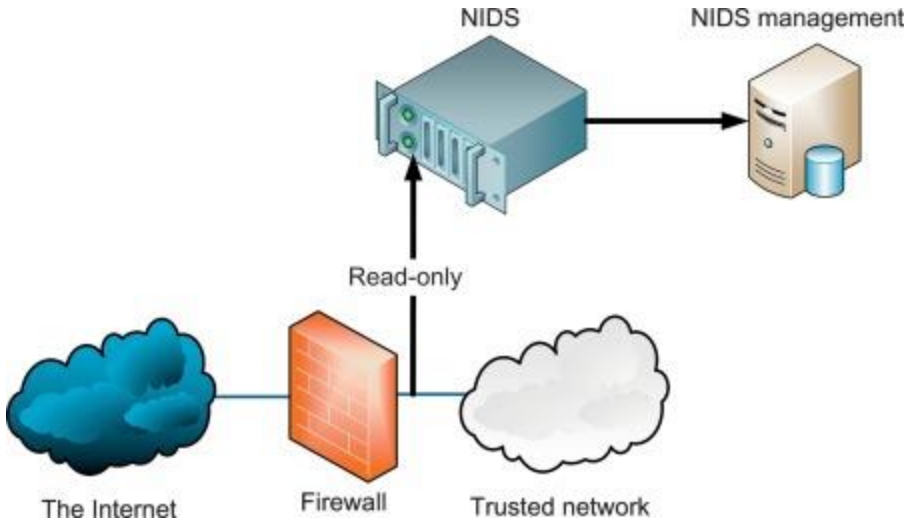
https://help.sana-commerce.com/sana-commerce-83/installation/setup-web-and-database-server/hardware-requirements-for-web-and-database-servers

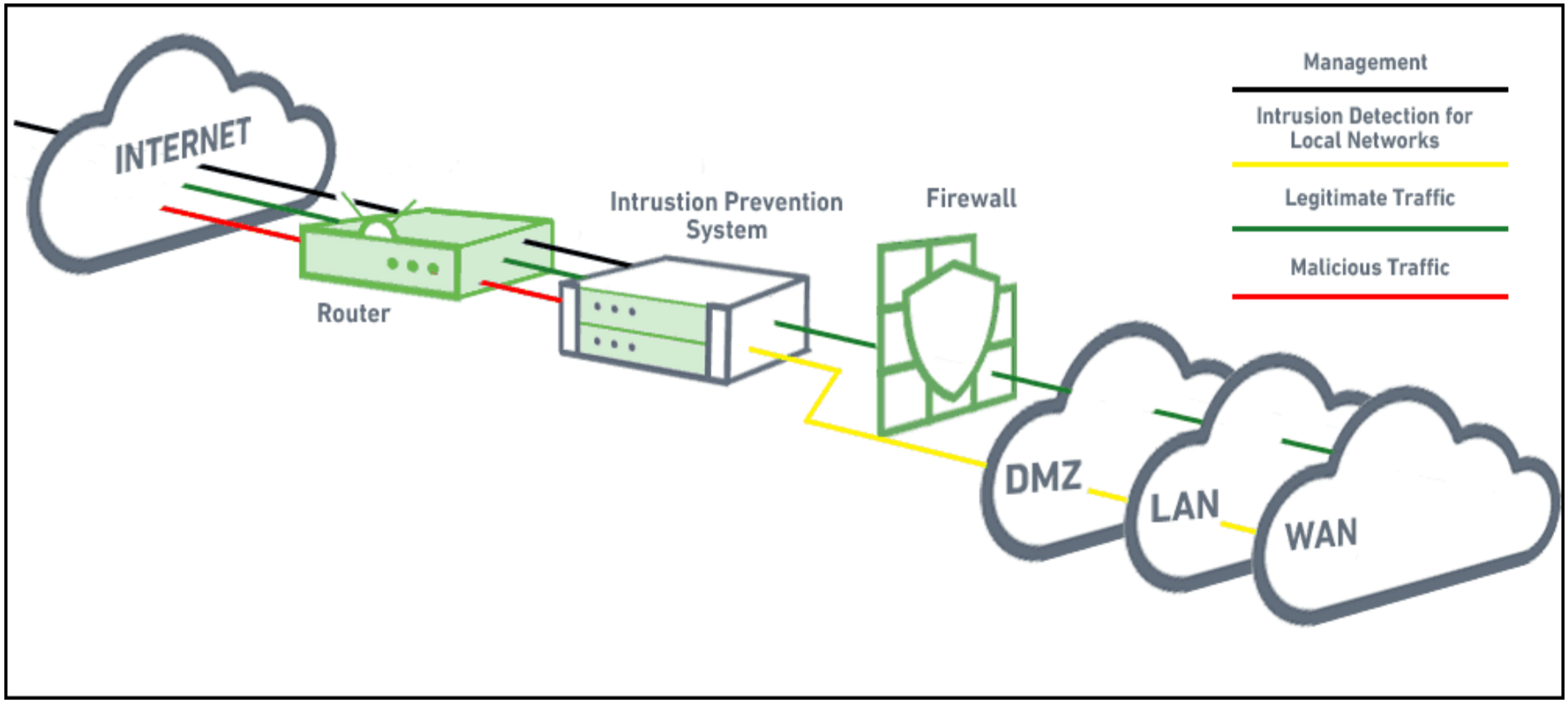
Network Architecture Diagrams

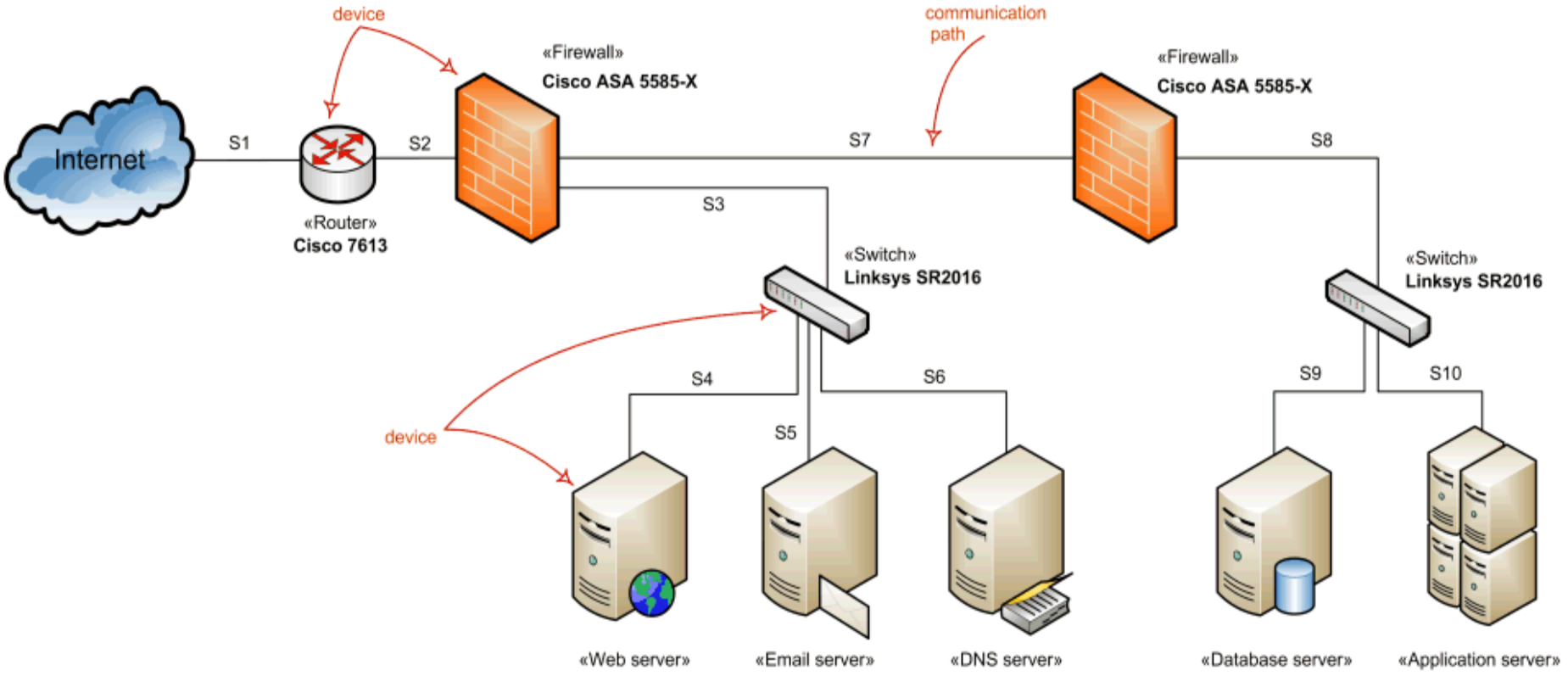
Note that this diagram uses networking icons that are **not** part of the UML standard. UML's standard for the node or **device** is a 3-dimensional view of a cube

Intrusion Detection System






INTERNET
Router
Intrustion Prevention System
Firewall
DMZ
LAN
WAN
Management
Intrusion Detection for Local Networks
Legitimate Traffic
Malicious Traffic



Windows Server System Reference Architecture (WSSRA) (see **Microsoft Network Architecture Blueprint** uses the following networking devices to show the overall network architecture:

Switch devices

Router devices

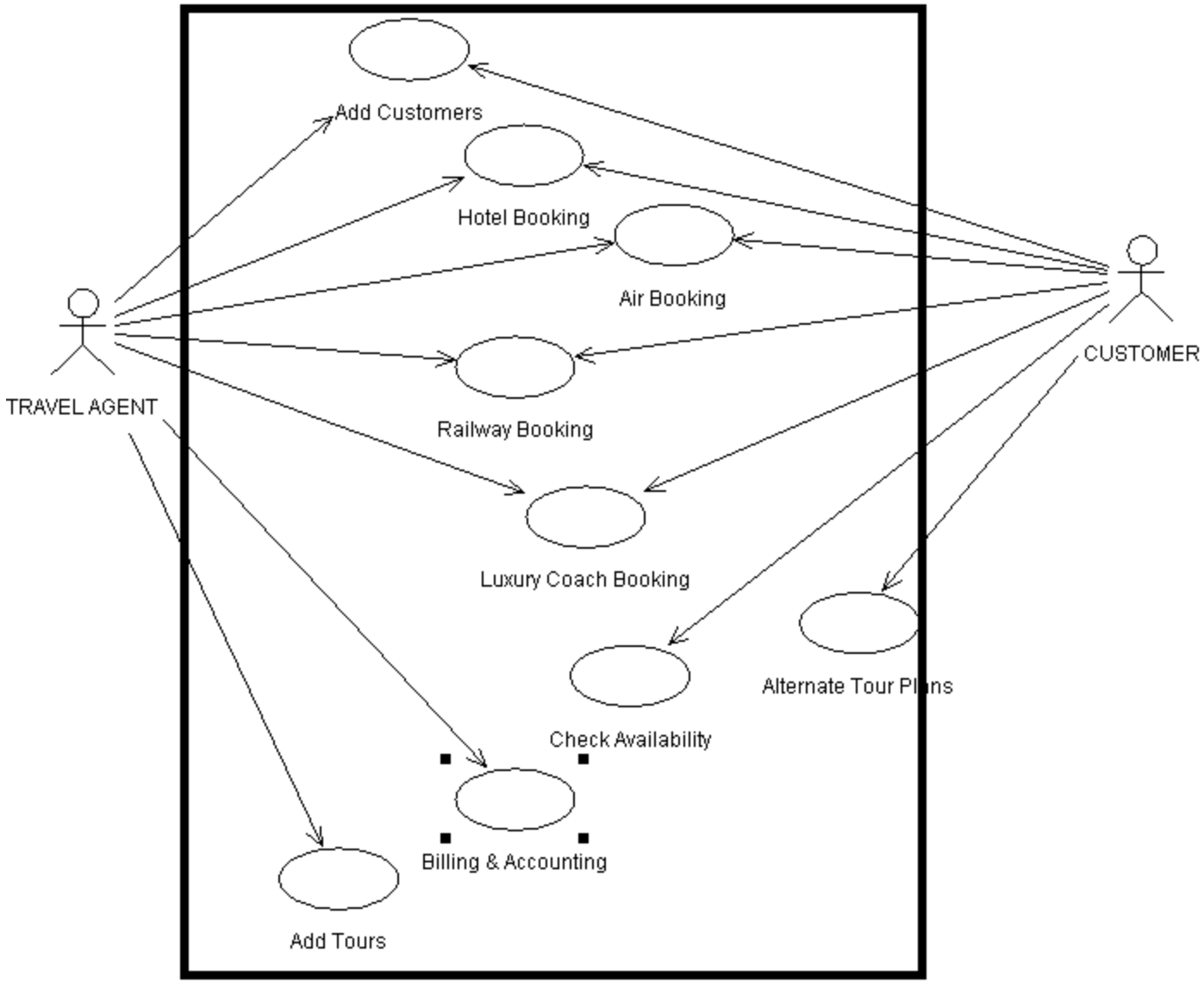
Load balancing devices

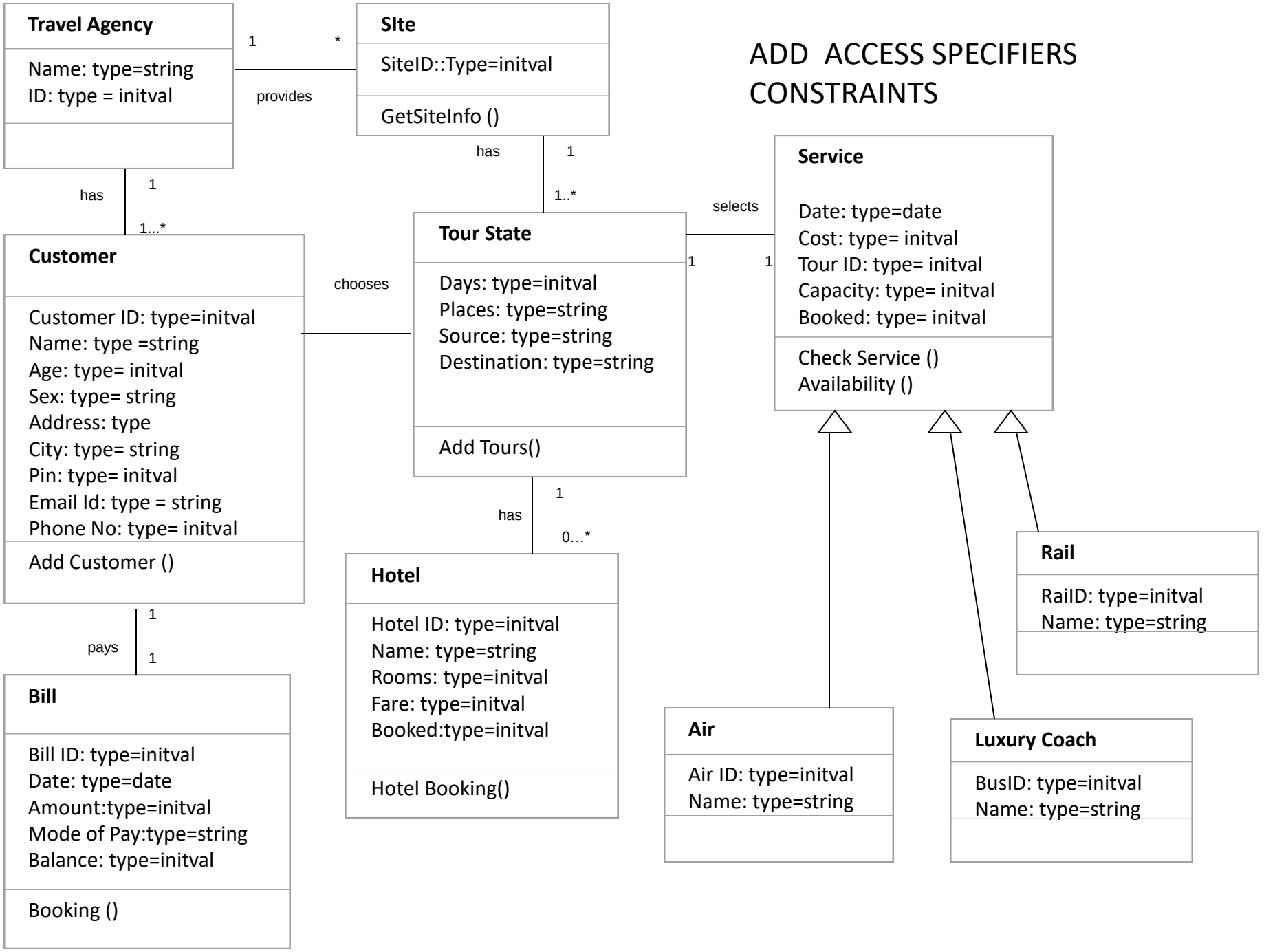
Firewall devices

Virtual Private Network (VPN) devices

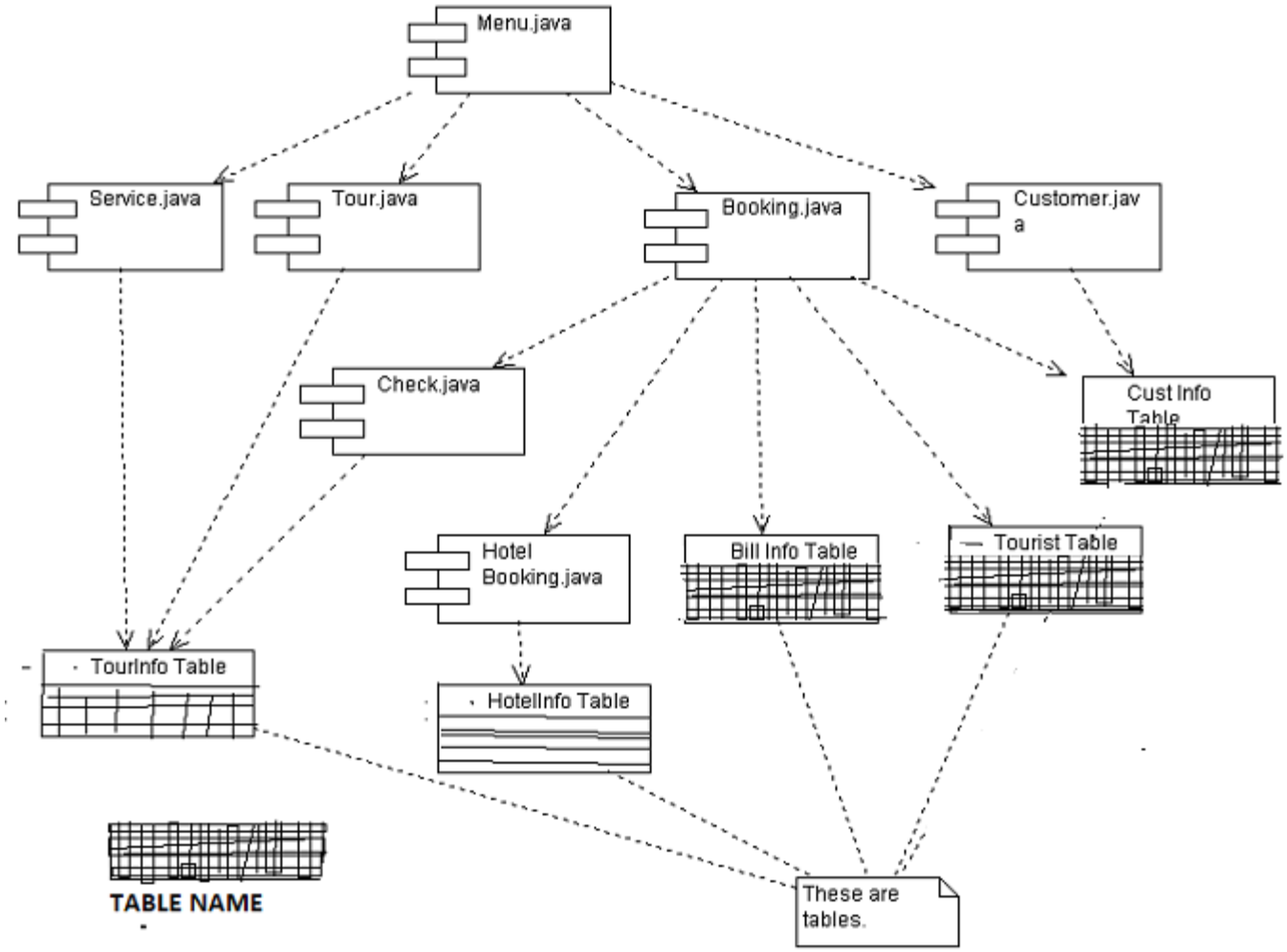
CASE STUDY

A leading travel agency has decided to develop application package to help its customers in planning tours.The agency provides services like air, railway, luxury, coach, hotel booking etc. Many a times customers do not have idea of availability of transport services to a particular destination. The agency also gives advice regarding economical planning of vacation/tour.Given the tour constraints like number of days, affordable cost & places to visit the software should present alternative tour plans .Alternatively the software may be just used for querying to know availability of transport services, hotels etc. Besides these main objectives the software should also have facilities for billing & accounting for the agency.You are appointed as a consultant to develop implementation strategy for the ***automated tourist system.***

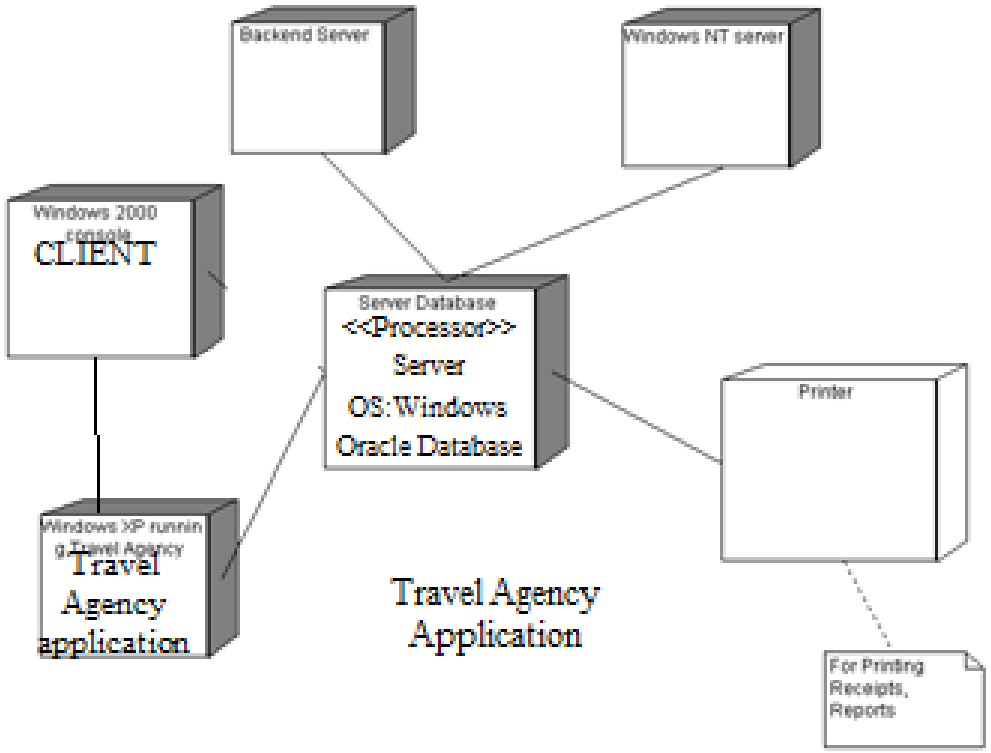

Add Customers
Hotel Booking
Air Booking
Railway Booking
Luxury Coach Booking
Alternate Tour Plans
Check Availability
Billing & Accounting
Add Tours
TRAVEL AGENT
CUSTOMER


Travel Agency
Name: type=string
ID: type = initval
1
*
SIte
SiteID::Type=initval
GetSiteInfo ()
provides
has
1
1...*
has
1
1..*
Customer
Customer ID: type=initval
Name: type =string
Age: type= initval
Sex: type= string
Address: type
City: type= string
Pin: type= initval
Email Id: type = string
Phone No: type= initval
Add Customer ()
chooses
Tour State
Days: type=initval
Places: type=string
Source: type=string
Destination: type=string
Add Tours()
selects
1
1
Service
Date: type=date
Cost: type= initval
Tour ID: type= initval
Capacity: type= initval
Booked: type= initval
Check Service ()
Availability ()
1
has
0...*
Hotel
Hotel ID: type=initval
Name: type=string
Rooms: type=initval
Fare: type=initval
Booked:type=initval
Hotel Booking()
pays
1
1
Bill
Bill ID: type=initval
Date: type=date
Amount:type=initval
Mode of Pay:type=string
Balance: type=initval
Booking ()
Air
Air ID: type=initval
Name: type=string
Luxury Coach
BusID: type=initval
Name: type=string
Rail
RailID: type=initval
Name: type=string
ADD ACCESS SPECIFIERS
CONSTRAINTS

Implementation :COMPONENT DIAGRAM



DEPLOYEMENT



COMPLETE THE FOLLOWING TABLE FOR THIS APPLICATION.
USE THIS TABLE FOR YOUR TA AND LABORATORY WORK.

Software requirements for this project are VB Visual Basic programming, Oracle Database Server and works on Windows XP. Hardware requirements are system should have 2 GB Ram, 80 GB hard disk Space. Find below added images to konw more details about Abstract and SRS of the system.

Software & Hardware Specifications for Tourism Management System

Hardware specification

Processor	: Celtron 500MHZ or any PENTIUM Processor
Ram	: 2GB or above
Hard disk capacity	: Minimum requirement is 80 GB
Display type	: Standard VGA or SVGA card

Software specification

Operating system	: Windows 95 or 98, XP
Backend	: ORACLE 8.0
Frontend	: VISUAL BASIC 6.0

DEPLOYEMENT DIAGRAM DESIGN

MACHINES	SOFTWARE	H/W OR CONFIGURATION	OPERATING SYSTEMS	COMPILER	ANY OTHE SOFTWARES	SOFTWARE MODULES AND PATH
CLIENT						
WEB SERVER						
APPLICATION SERVER						
DATABASE SERVER						