- Chapter 1: Project Overview
  - Company Profile
  - Importance of the Study
  - Statement of the Problem
  - Objectives of the Study
  - Definition of Terms

- Chapter 2: Methodology
  - Software Development Methodology
  - Scope and Delimitation
  - Data Gathering Techniques
  - Sources of Data
- Chapter 3: Discussion of Finding
  - SOP1
  - SOP2
  - SOP3

//with supporting UML Diagrams

UML DIAGRAM

Unified Modeling Language

- Used in object oriented software engineering.
- Rich language that can be used to model an application

  structures, behavior and even business processes.
- Two types of UML
  - struct ure diagrams
  - behavioral diagrams.

UML Diagrams

- 1. Class Diagram
- 2. Component Diagram
- 3. Deployment Diagram
- 4. Object Diagram
- 5. Package Diagram
- 6. Profile Diagram
- 7. Composite Structure Diagram

- 8. Use Case Diagram
- 9. Activity Diagram
- 10. State Machine Diagram
- 11. Sequence Diagram
- 12. Communication Diagram
- 13. Interact ion Overview Diagram
- 14. Timing Diagram

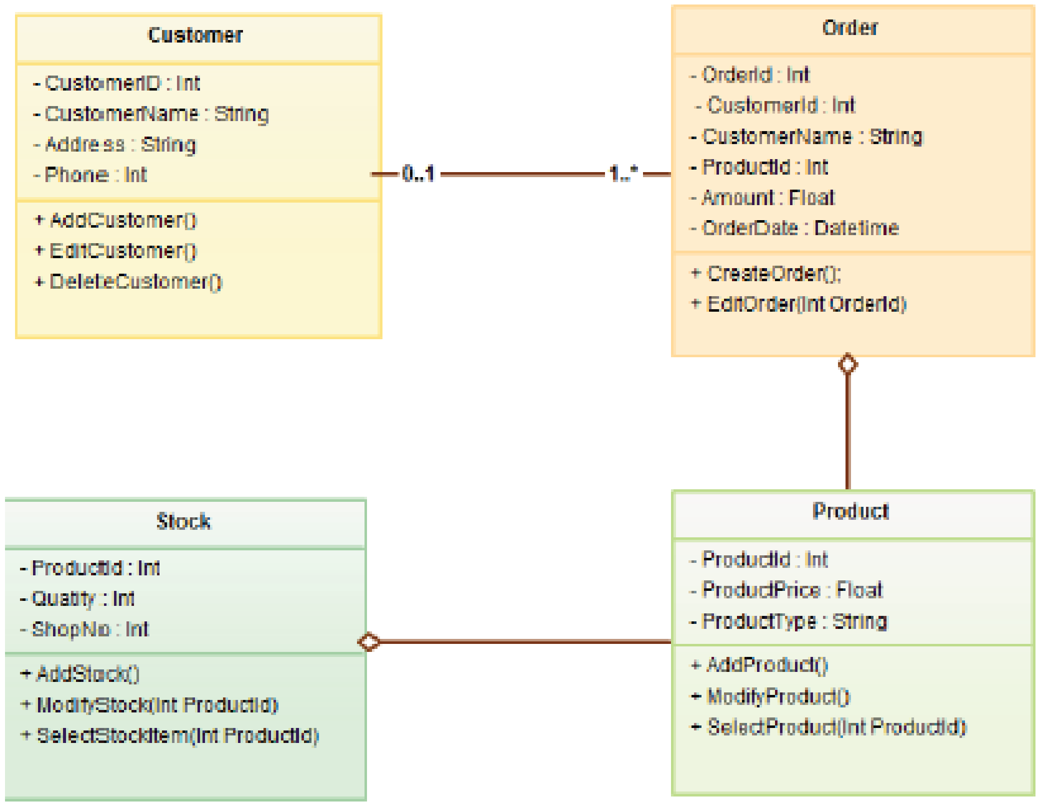
Structure Diagrams

- Used to present different object s in a system
  - 1. Class Diagram
  - 2. Component Diagram
  - 3. Deployment Diagram
  - 4. Object Diagram
  - 5. Package Diagram
  - 6. Profile Diagram
  - 7. Composite Structure Diagram

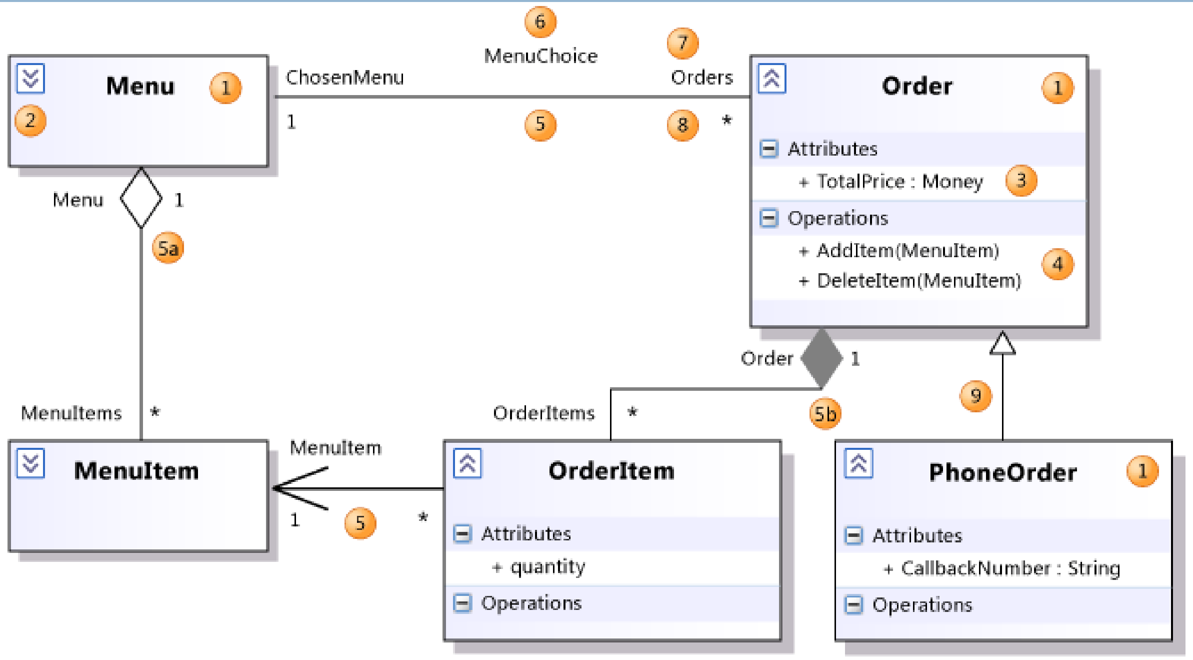
Class Diagram

- It shows the classes in a system, attributes and operations of each class and the relationship between each class.
- In large systems with many classes, related classes are grouped together to create class diagrams.

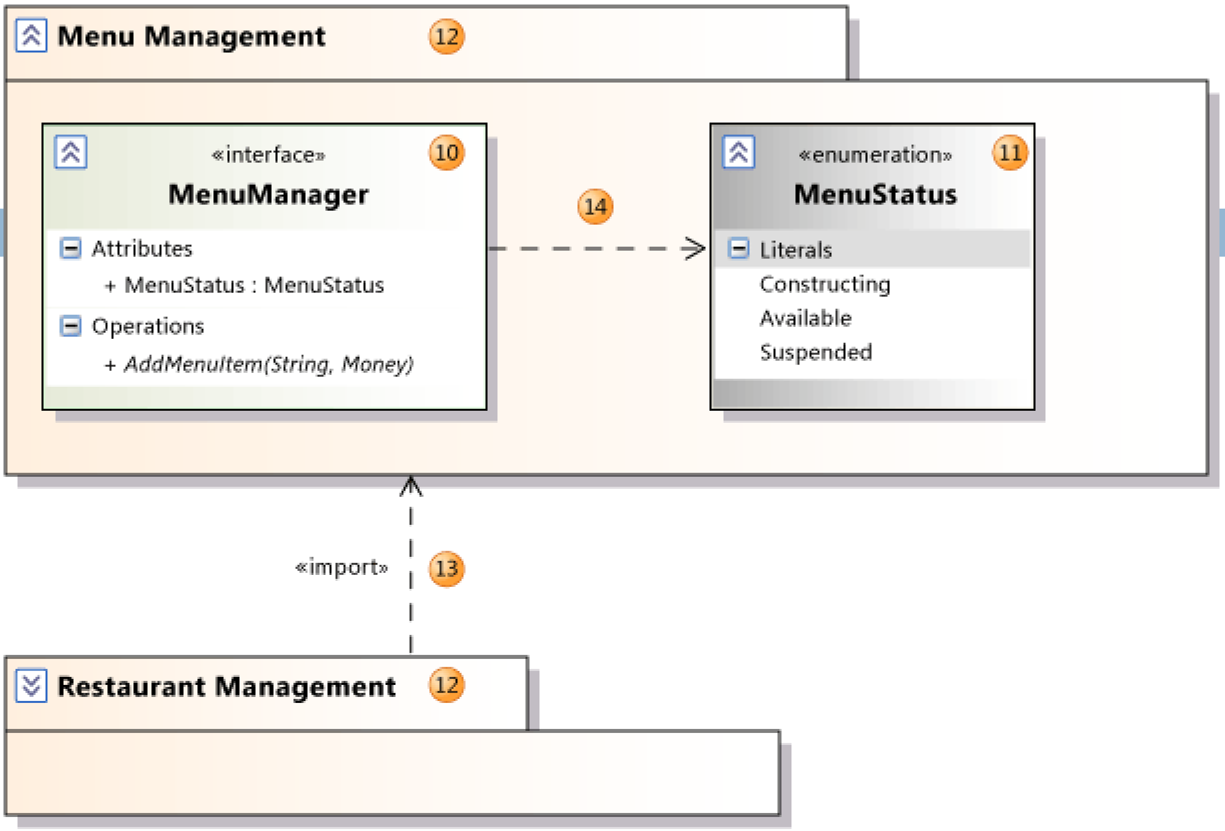
Class Diagram for Order Processing System



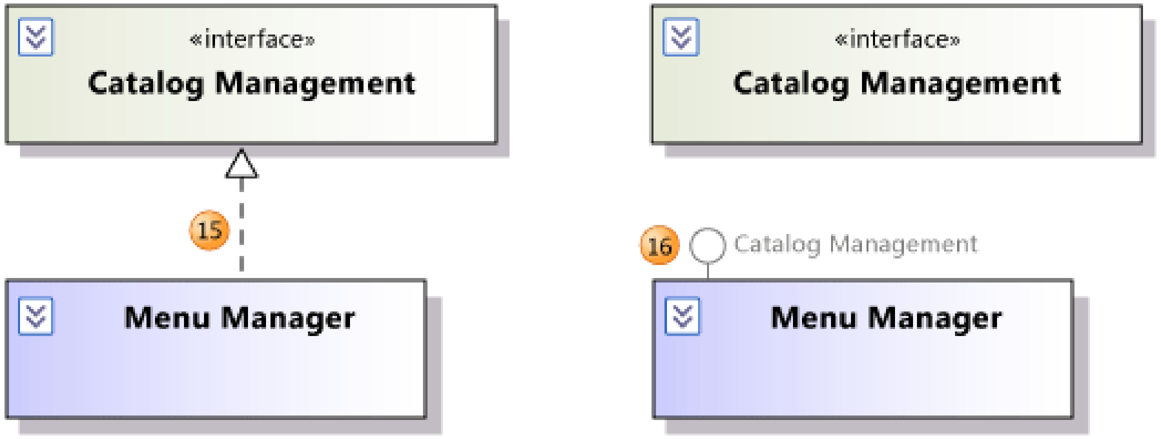
Relationship Arrows



- **5: Association:** A relationship between the members of two classifiers.
- **5a: Aggregation:** An association representing a shared ownership relationship. The **Aggregation property** of the owner role is set to **Shared**.
- **5b: Composition:** An association representing a whole-part relationship. The **Aggregation** property of the owner role is set to **Composite**.
- **9: Generalization:** The specific classifier inherits part of its definition from the general classifier. The general classifier is at the arrow end of the connector. Attributes, associations, and operations are inherited by the specific classifier. Use the **Inheritance** tool to create a generalization between two classifiers.



- **13: Import:** A relationship between packages, indicating that one package includes all the definitions of another.
- **14: Dependency:** The definition or implementation of the dependent classifier might change if the classifier at the arrowhead end is changed.



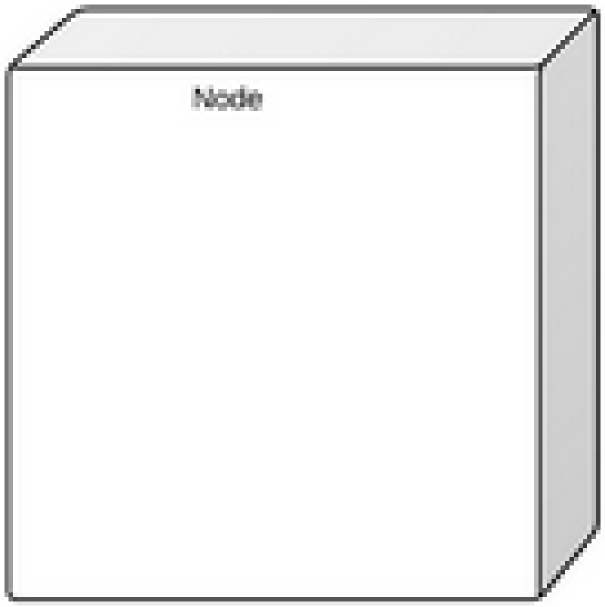
- **15: Realization:** The class implements the operations and attributes defined by the interface. Use the Inheritance tool to create a realization between a class and an interface.
- **16: Realization:** An alternative presentation of the same relationship. The label on the lollipop symbol identifies the interface.

Deployment Diagram

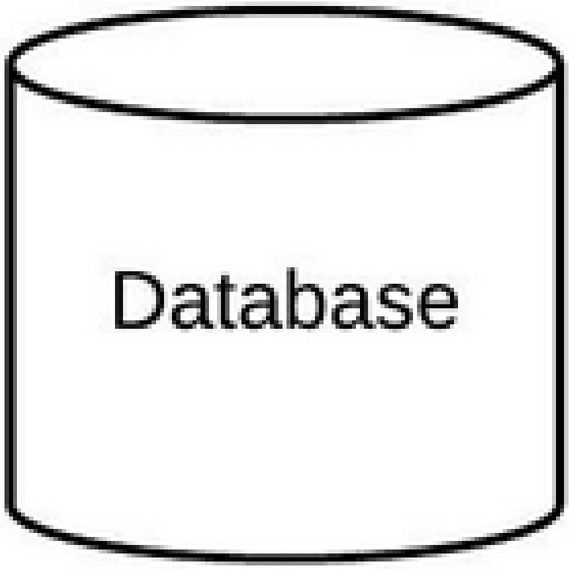
- Shows the hardware of your system and the software in those hardware.
- Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration.

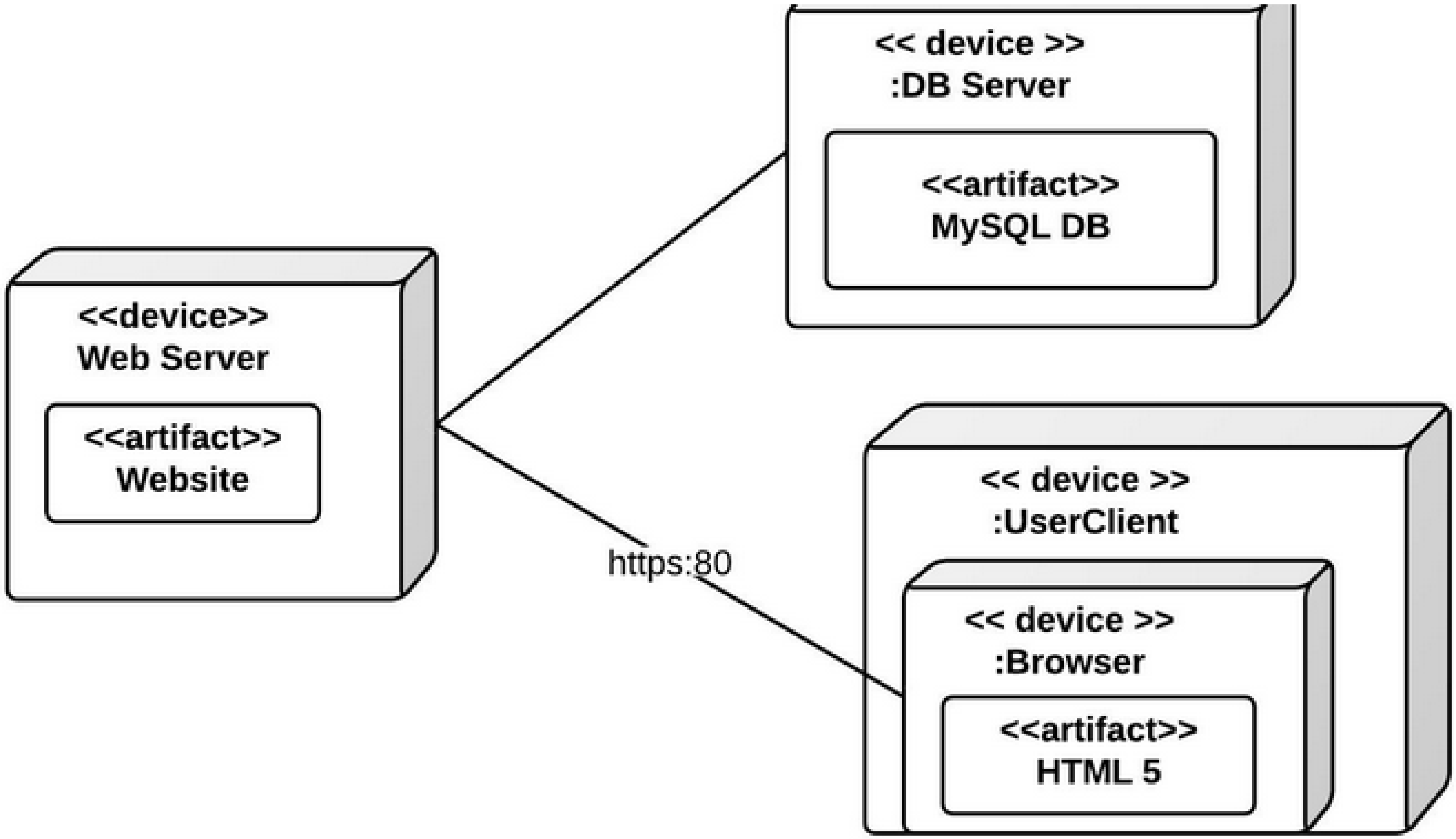
Deployment Diagram Notations

Nodes



Database

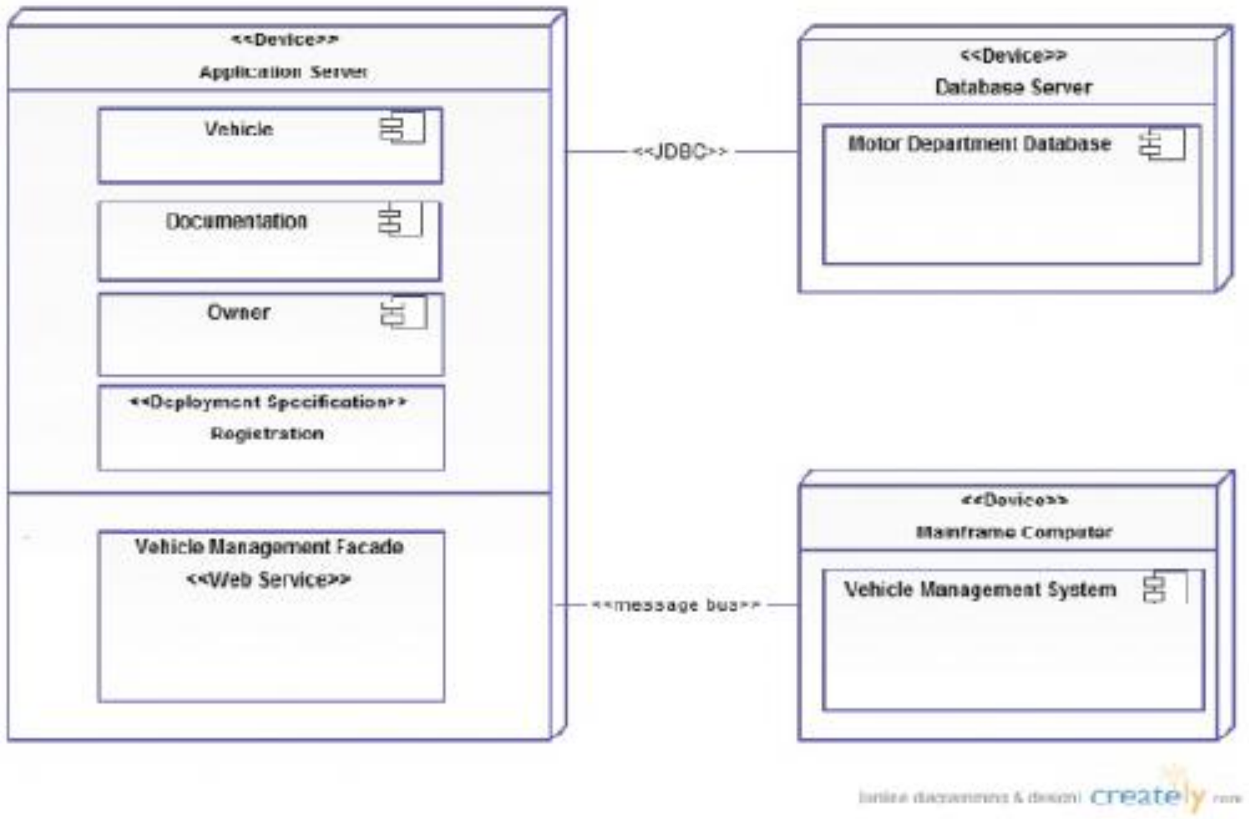




<< device >>
:DB Server
<<artifact>>
MySQL DB
<<device>>
Web Server
<<artifact>>
Website
https:80
<< device >>
:UserClient
<< device >>
:Browser
<<artifact>>
HTML 5

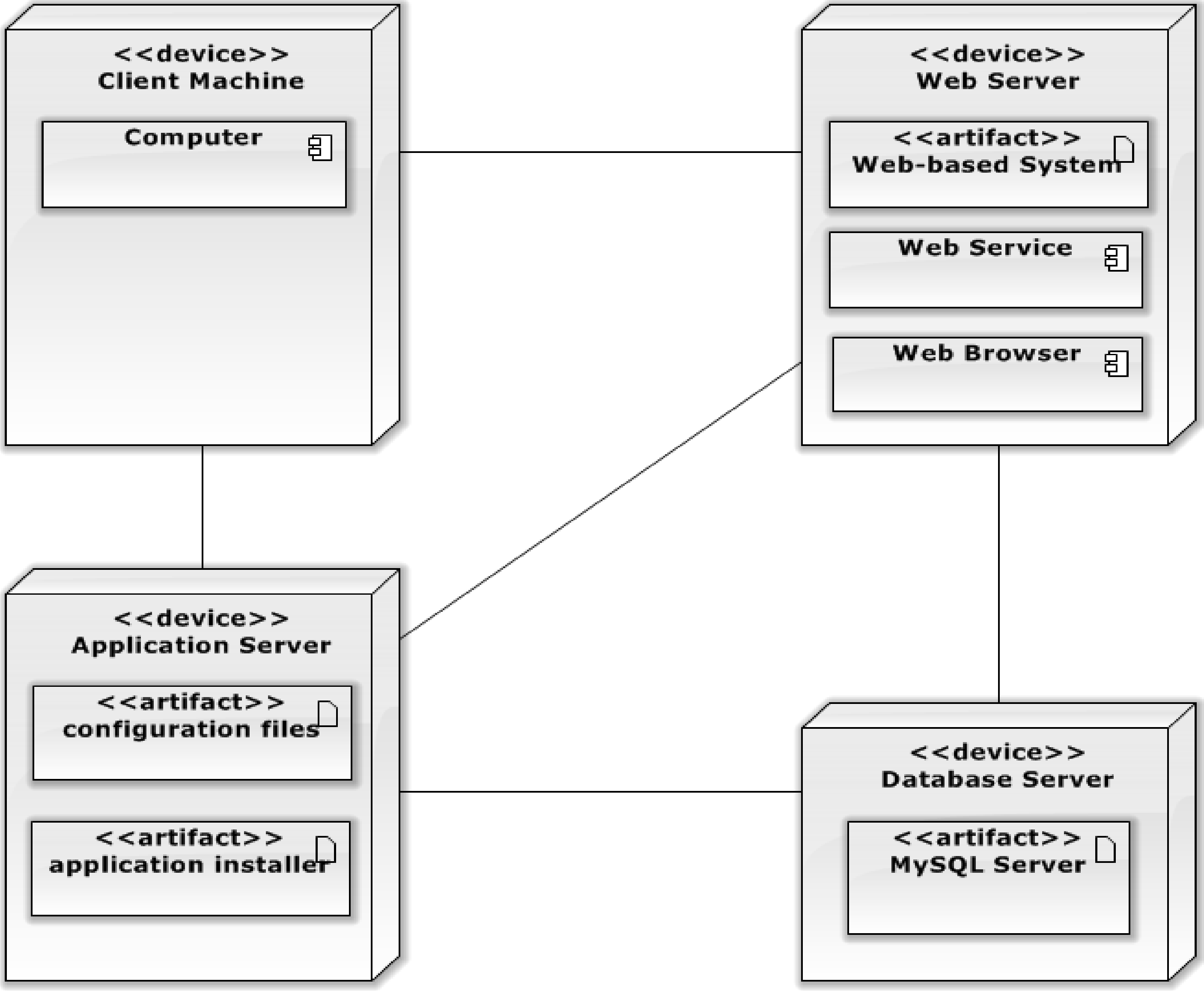
- **Communication path.**A straight line that represents communication between two device nodes.
- **Artifacts.** A box with the header "<<artifact>>" and then the name of the file.
- **Package.** A package is a file shaped box that groups together all the device nodes to encapsulate the entire deployment.
- **Component**

- **Have you identified the scope of your system?**
- **Make sure you have considered the limitations of your physical hardware.**
- **Which distribution architecture are you using?**
  - How many tiers will your application have?
  - What is the application you will be deploying to?
- **Do you have all the nodes you need? Do you know how they are all connected?**
- **Do you know which components are going to be on which nodes?**



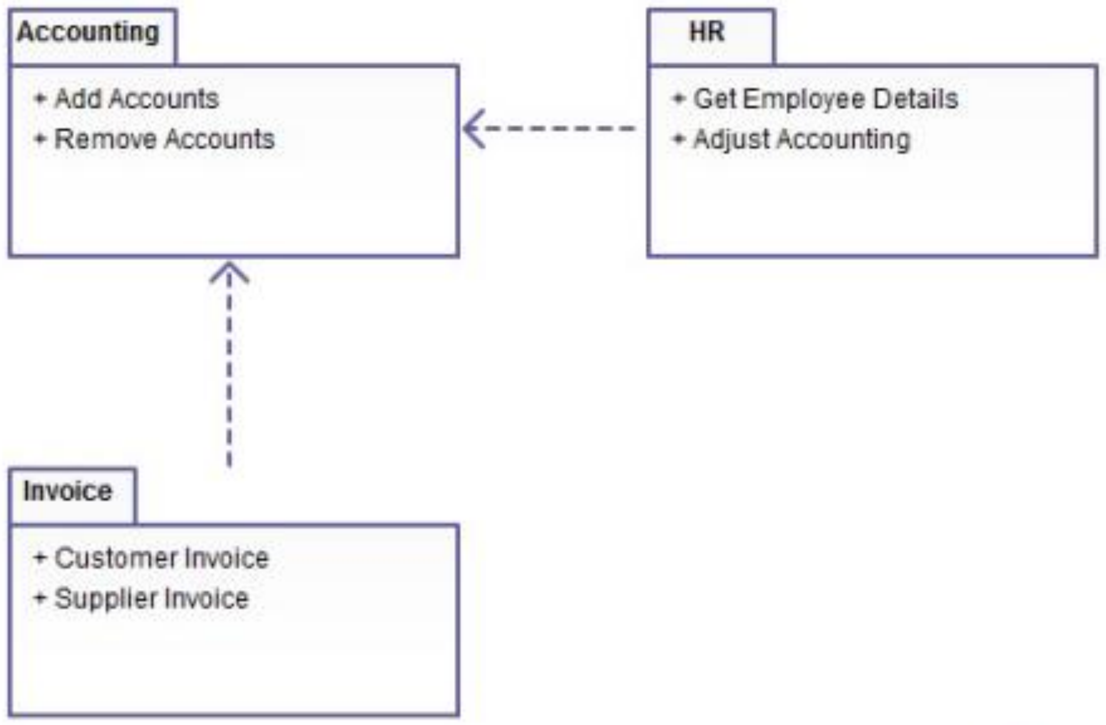
UML Deployment Diagram (Click on the image to use it as a template)

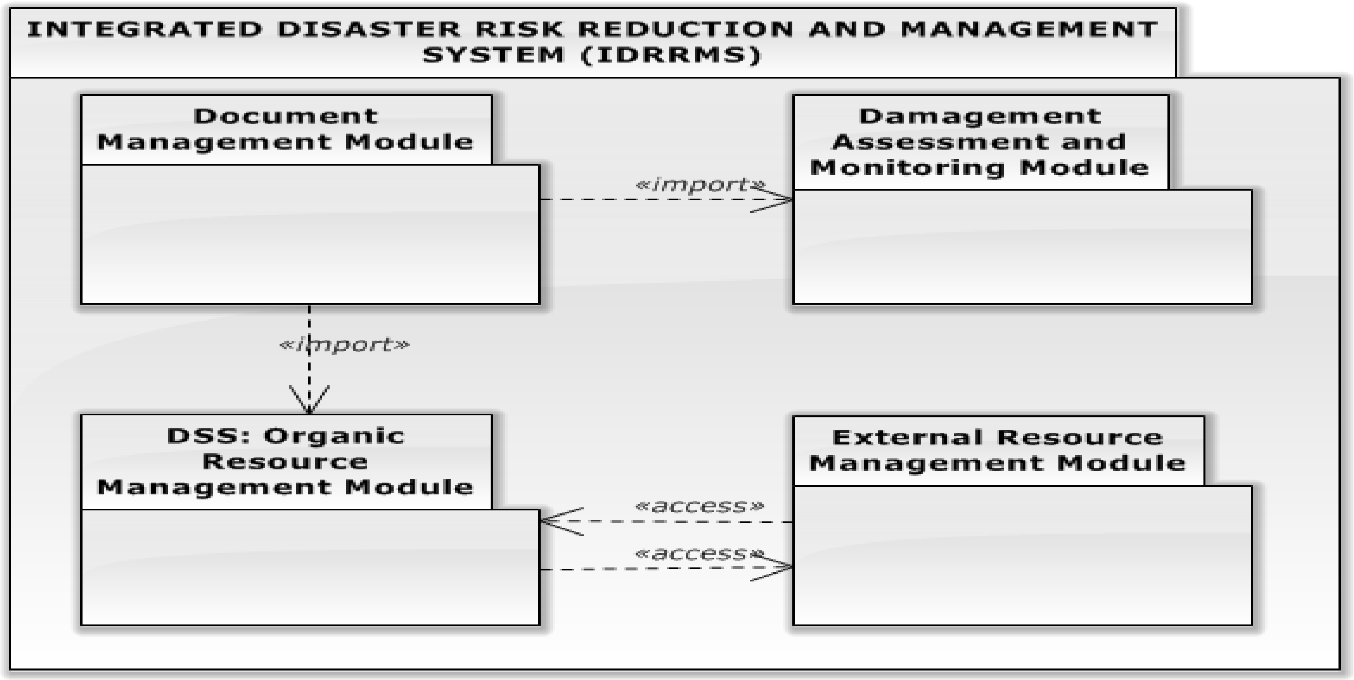
[illegible]


<<device>>
Client Machine
Computer
<<device>>
Web Server
<<artifact>>
Web-based System
Web Service
Web Browser
<<device>>
Application Server
<<artifact>>
configuration files
<<artifact>>
application installer
<<device>>
Database Server
<<artifact>>
MySQL Server

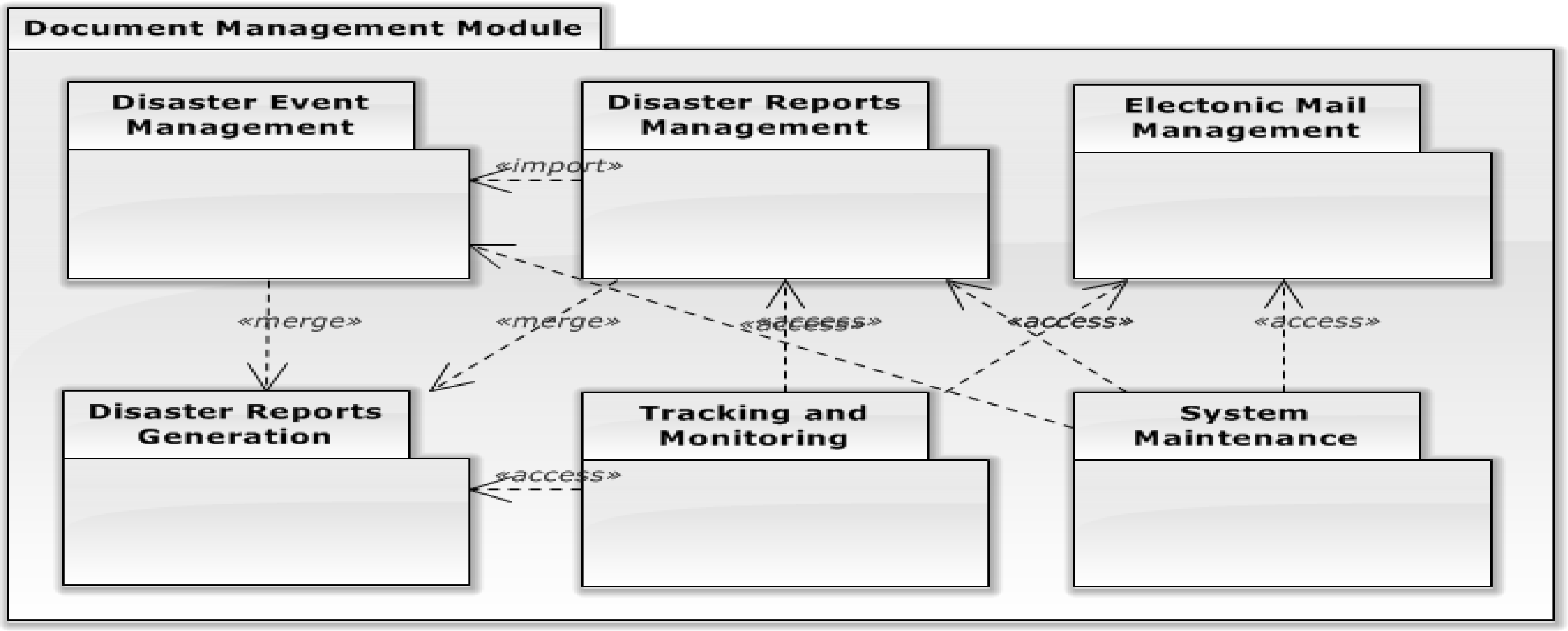
Package Diagram

- Shows the dependencies between different packages in a system





INTEGRATED DISASTER RISK REDUCTION AND MANAGEMENT SYSTEM (IDRRMS)
Document Management Module
Damagement Assessment and Monitoring Module
«import»
«import»
DSS: Organic Resource Management Module
External Resource Management Module
«access»
«access»



Document Management Module
Disaster Event Management
Disaster Reports Management
Electonic Mail Management
«import»
«merge»
«merge»
«access»
«access»
«access»
«access»
Disaster Reports Generation
Tracking and Monitoring
System Maintenance
«access»

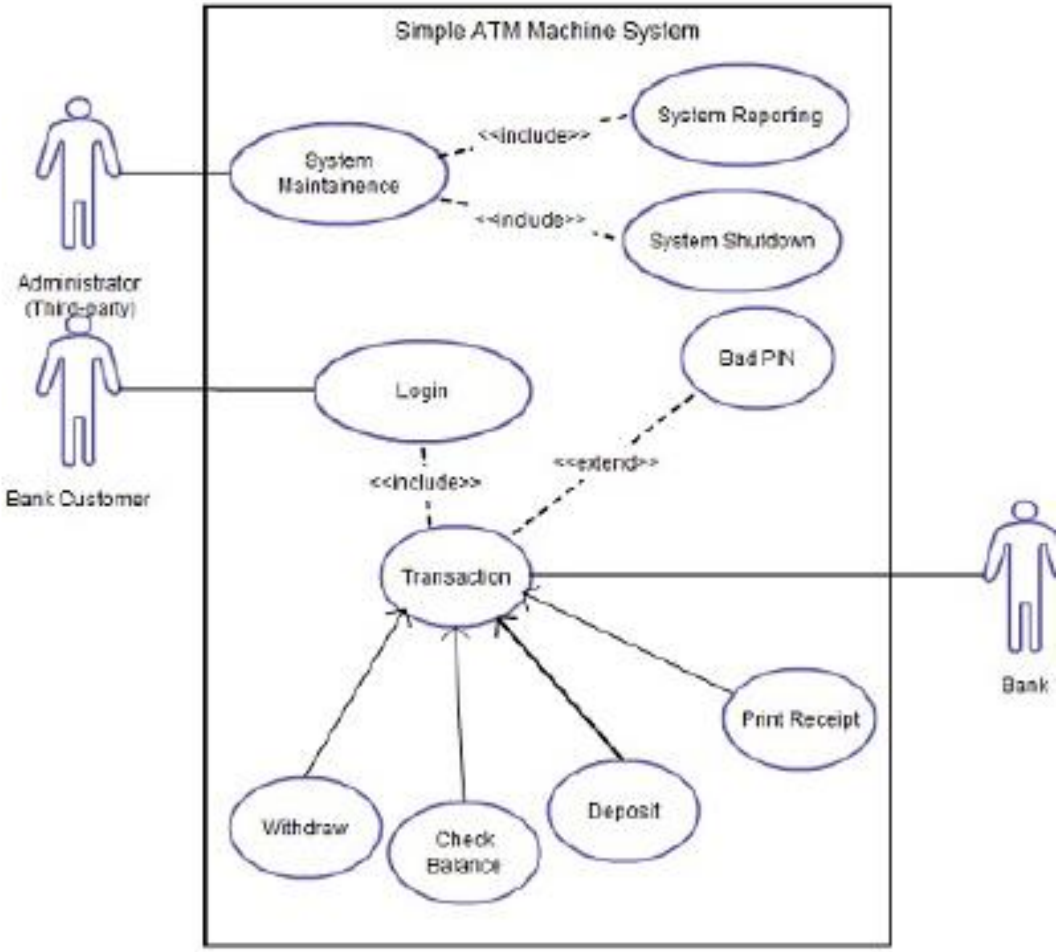
Use Case Diagram

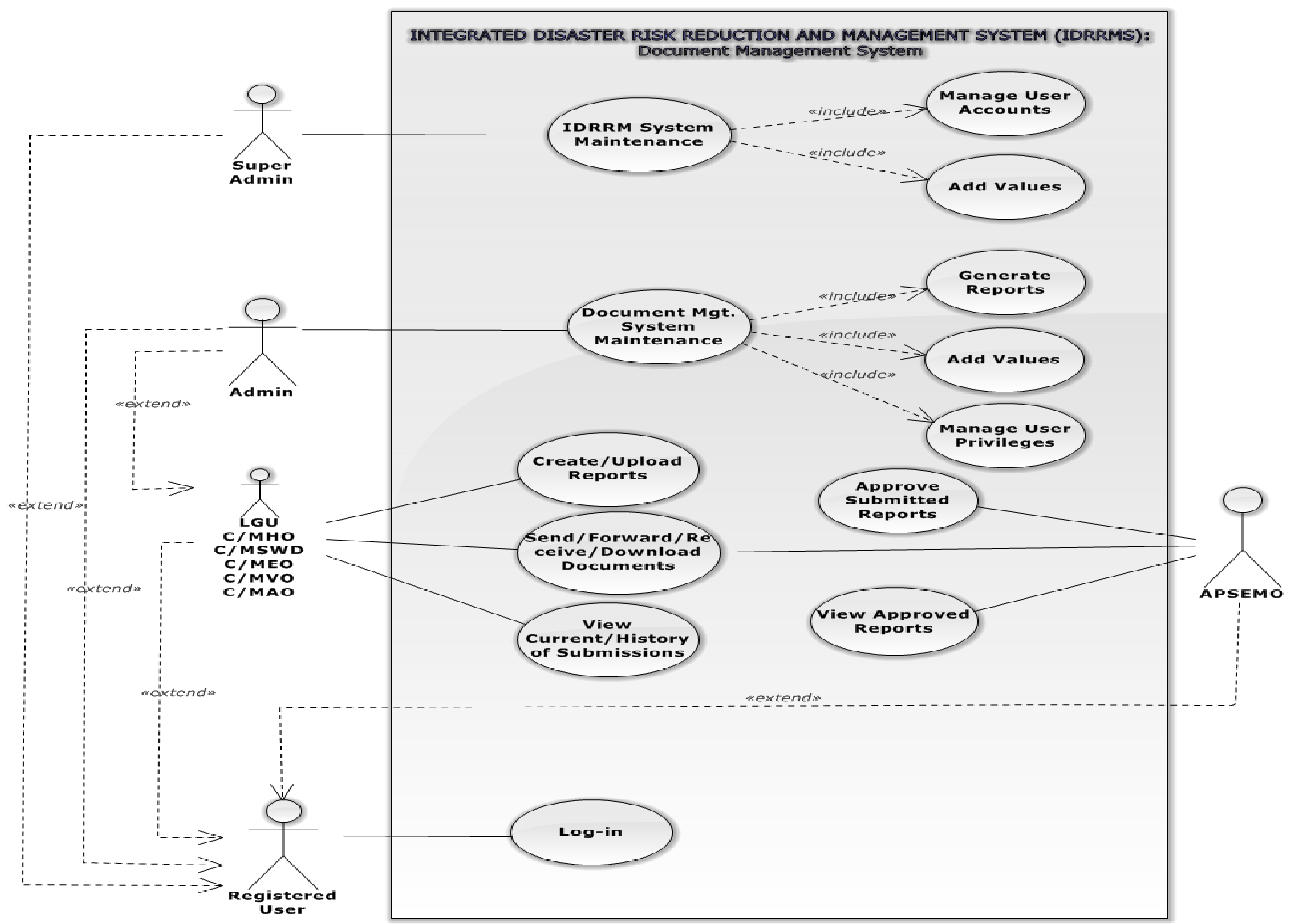
- Use case diagrams gives a graphic overview of:
  - the actors involved in a system
  - the different functions needed by the actors
  - how the different functions interact

Use Diagram Notations

- Use cases - Use cases are the horizontally shaped ovals. This represents the different uses that a user might need.
- Actors - represented by stick figure people and are the people actually employing the use cases
- Associations - represented by a line between actors and use cases. In a more complex diagram, it is important to know which actors are associated with which use cases.

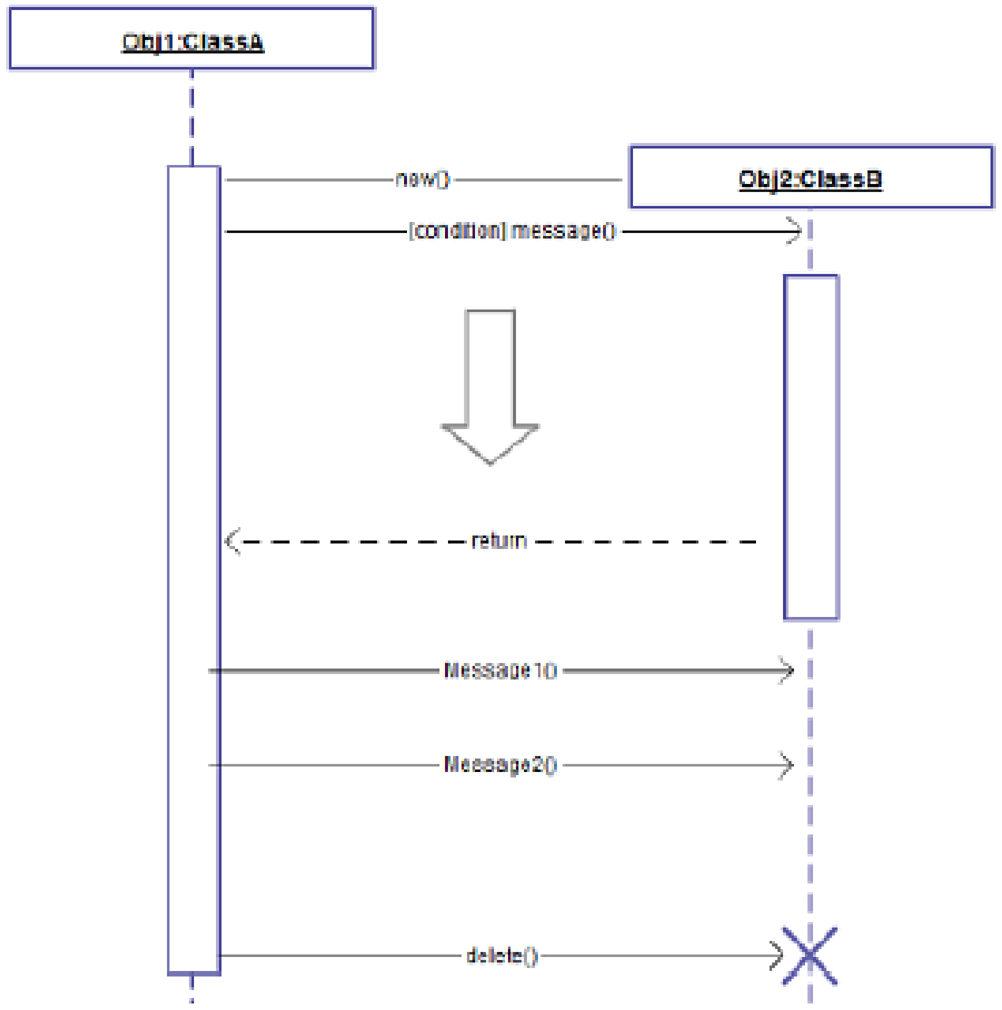
- System boundary boxes - a box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system.
- Packages - A UML shape that allows you to put different elements into groups.


Simple ATM Machine System
Administrator (Third-party)
Bank Customer
Bank
System Maintainence
System Reporting
System Shutdown
<<include>>
<<include>>
Bad PIN
Login
<<include>>
<<extend>>
Transaction
Print Receipt
Withdraw
Check Balance
Deposit


INTEGRATED DISASTER RISK REDUCTION AND MANAGEMENT SYSTEM (IDRRMS): Document Management System
Super Admin
Admin
LGU C/MHO C/MSWD C/MEO C/MVO C/MAO
APSEMO
Registered User
IDRRM System Maintenance
Manage User Accounts
Add Values
«include»
«include»
Document Mgt. System Maintenance
Generate Reports
Add Values
Manage User Privileges
«include»
«include»
«include»
Create/Upload Reports
Send/Forward/Receive/Download Documents
View Current/History of Submissions
Approve Submitted Reports
View Approved Reports
Log-in
«extend»
«extend»
«extend»
«extend»
«extend»

Sequence Diagram

- Shows how object interact with each other and the order those interact ions occur.
- It 's important to note that they show the interactions for a particular scenario.
- The processes are represented vertically and interact ions are shown as arrows.


Obj1:ClassA
Obj2:ClassB
new()
[condition] message()
return
Message1()
Message2()
delete()