Omkar Gurav
BE IT
8048

SDM Theory assignment 3

Q.1. Explain component diagram & draw diagram for your CL-VIII problem statement.

→

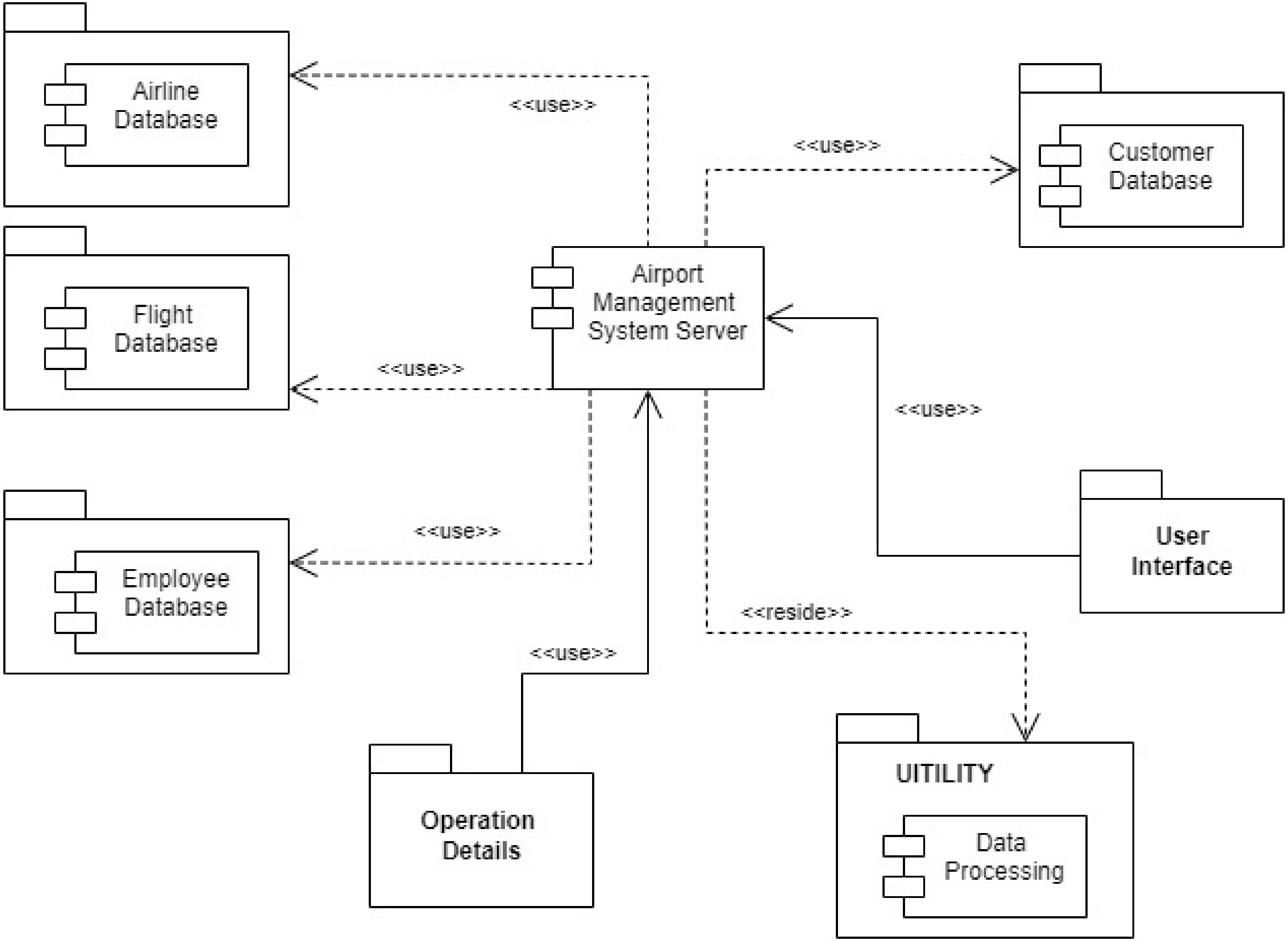
- A component diagram is basically meant for exhibiting the physical characteristics of the object oriented system.
- It is also used for modeling the static implementation view of the software system.
- A component diagram provides a framework for constructing the executable software systems & applications by means of forward & reverse engineering.
- A component diagram depicts a set of components & relationships among those components.
- Graphically, it is represented as a rectangle with tabs.
- Each component involved within a system should be given a name in order to identify a particular component uniquely from other component.
- There are 3 types of components:-

① Work product components: Examples are data files, source code files etc.

② Deployment components: This type of components are essential for creation of execution components.

③ Execution components: Executable, Dynamic libraries etc are the typical examples.

Component Diagram



Q.2. Explain Deployment Diagram & draw diagram for your CL-VIII statement.

→

- A decision is made about what component instances are essential, during application deployment process.
- Following activities are involved in the process of application deployment:
  - Define instances of the component
  - Interconnect component instances
  - Map the component instances to physical nodes
- In UML, deployment diagram depicts the physical aspect of an object oriented system.
- A deployment is basically used for modelling the static deployment view of a system.
- A deployment diagram may contain components, subsystems & packages as per the necessity of a system.
- Nodes are used to design the topology of the hardware on which a proposed software system executes.
- Association relationship can be furthermore used to define indirect connections.
- Association relationship shows a physical connection between nodes for instance, shared bus, Ethernet line etc.
- Dependency is the other kind of relationship that can be used for showing the interdependency amongst nodes involved within a system scenario.

Q.3. Explain Client/server software architectures with types.

→

- The client/serve software architecture is a multipurpose, message-based & modular infrastructure that is envisioned to increase usability, flexibility, interpreability & scalability as compared to centralized, mainframe, time sharing computing.
- A client is defined as a requester of services & a server is defined as the provider of services.
- Several client-server software architectural structure patterns are:-
  - Multiple Client-Single service Architectural pattern,
  - Multiple Client-Multiple Services Architectural pattern
  - Multi-tier Client Server Architectures

ⓐ Multiple Client-Single service Architectural Pattern:-

The Multiple Client-Single Service architectural pattern comprises of multiple clients that request a service & a service that fulfills requests from clients.

The banking system is the classic example of this type where, it contains many ATMs & one Banking Service System.

ⓑ Multiple Client-Multiple Service Architectural Pattern:-

The multiple Client-Multiple service architectural pattern supports multiple services.

The multiple Client/Multiple Service architectural pattern can be represented with the help of a deployment

diagram.
In ATM example, one ATM client subsystem may access multiple bank services.

© Multi-tier Client service Architectural Pattern:-

The Multi-tier Client/Service Pattern has a midway tier that is, layer which offers a role of both a client & a service.
A midway tier is a client of its service tier & also offers a service for its clients.
It is possible for us to have more than 1 Midway tier.

Q4. Explain Designing Service Oriented Software Architectures & broker.

→

- A service-oriented architecture (SOA) is a distributed software architecture which involves mainflod self-governing service.
- These services are distributed in such a way that they can execute on different nodes with different service provider.
- Standard protocols are provided to allow services to communicate with each other & to exchange information.
- So, an important objective of service-oriented architecture is to design services as self-governing reusable software components.

- Software Architectural Broker Patterns:-

  - Object brokers act as a mediators amongst clients & servers in service-oriented architectures.
  - Broker patterns are also known as object Broker patterns or object Request Broker patterns.
  - Servers register with the broker. Clients locate services through the broker.
  - After the broker has brokered the connection amongst client & server, communication amongst client & server may be direct or by means of the broker.
  - The broker provides both location transparency & platform transparency.
  - Location transparency discusses the fact that, users cannot tell where a resource is physically located in the system scenario.