Practical no: 7

Name: Saloni Vishwakarma
Batch-Roll no: C1-13
Subject: Software Engineering and Project Management Lab
Date of execution: 30 October 2023

Aim: 1. Component diagram for an ATM system
2. Component diagram for Hospital Management System.

Theory: A component diagram is a type of UML (Unified Modeling Language) diagram used in software engineering and systems design to illustrate the structural organization of a system or application. Component diagrams are particularly useful for visualizing the high-level architecture of a system, showing how various components or modules interact with each other. Here are some key elements and concepts related to component diagrams:

Component: In a component diagram, a component represents a modular part of a system, which can be a software module, a physical component, or any other building block of the system. Components can have well-defined interfaces and may encapsulate functionality.

Interface: Interfaces define the way components interact with each other. They specify the methods, attributes, and services that a component provides or requires from other components. Interfaces are represented as labeled connectors in the component diagram.

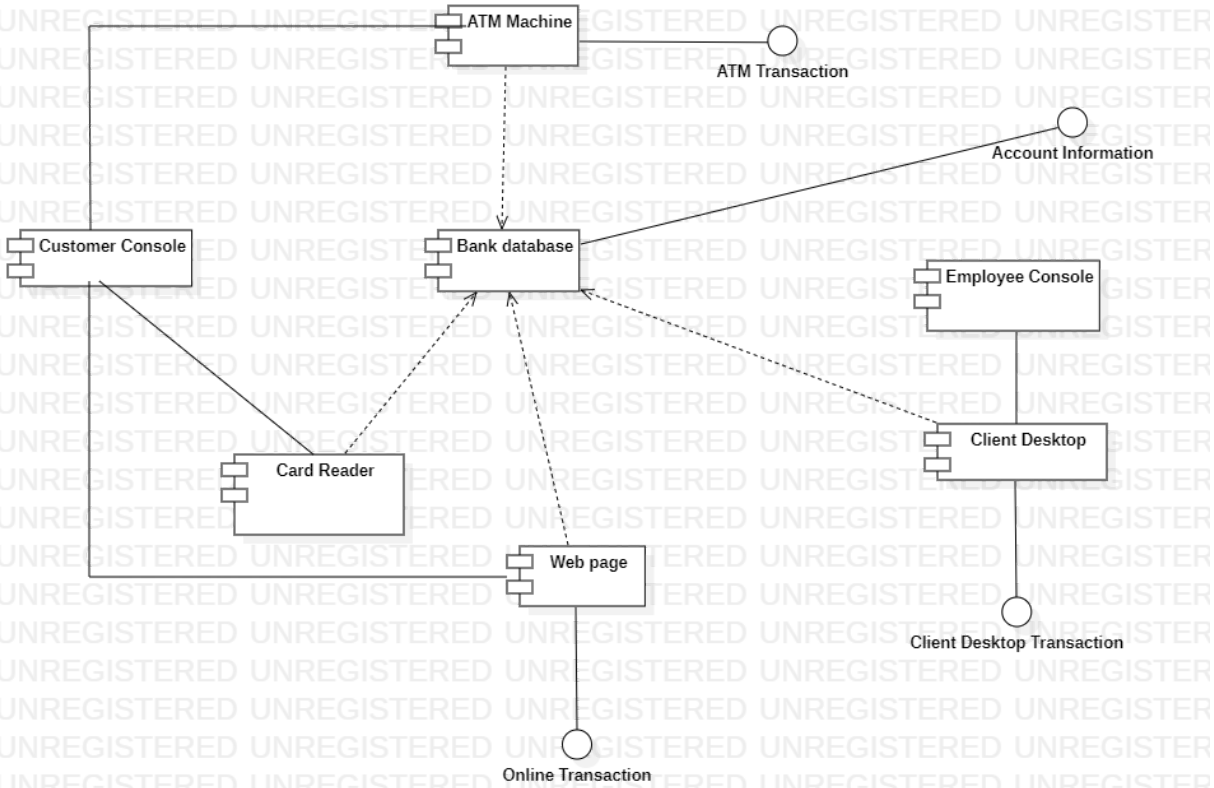
Dependency: Dependencies between components are indicated by connecting lines with arrows. A dependency relationship signifies that one component depends on another. For example, if Component A uses or relies on Component B, there will be a dependency arrow from A to B.

Assembly Connector: An assembly connector is used to show how components are connected or composed to form a larger system. It represents the relationship between a composite component (the one that contains other components) and the components it contains.

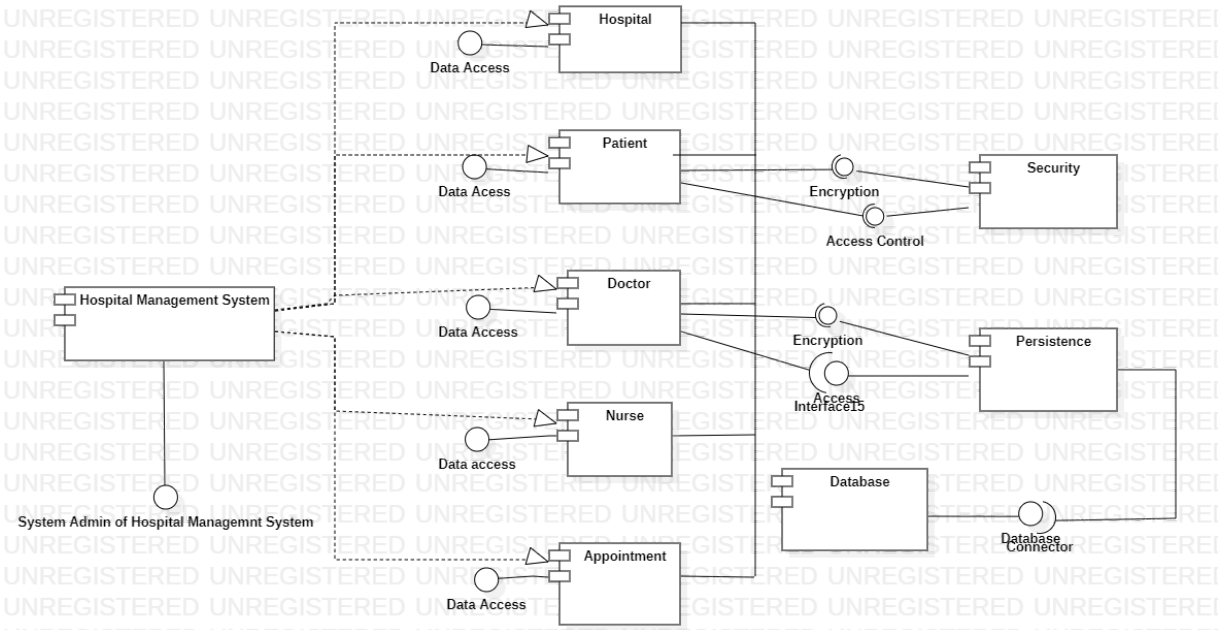
Provided and Required Interfaces: Components can provide interfaces (services they offer to other components) and require interfaces (services they need from other components). These are often depicted using stereotypes like "<<provides>>" and "<<requires>>."

Notation: Component diagrams use a simple notation, typically with a rectangle representing the component, and connectors (lines) representing relationships between components. The component name is written inside the rectangle, and interfaces are labeled on the connectors.

1) ATM


ATM Machine
ATM Transaction
Account Information
Customer Console
Bank database
Employee Console
Client Desktop
Card Reader
Web page
Client Desktop Transaction
Online Transaction

2) Hospital Management System



Conclusion: We have successfully studied and implemented Component diagrams which are valuable for high-level system design and can help teams understand the overall structure and dependencies within a complex software system or any system with multiple interconnected components.