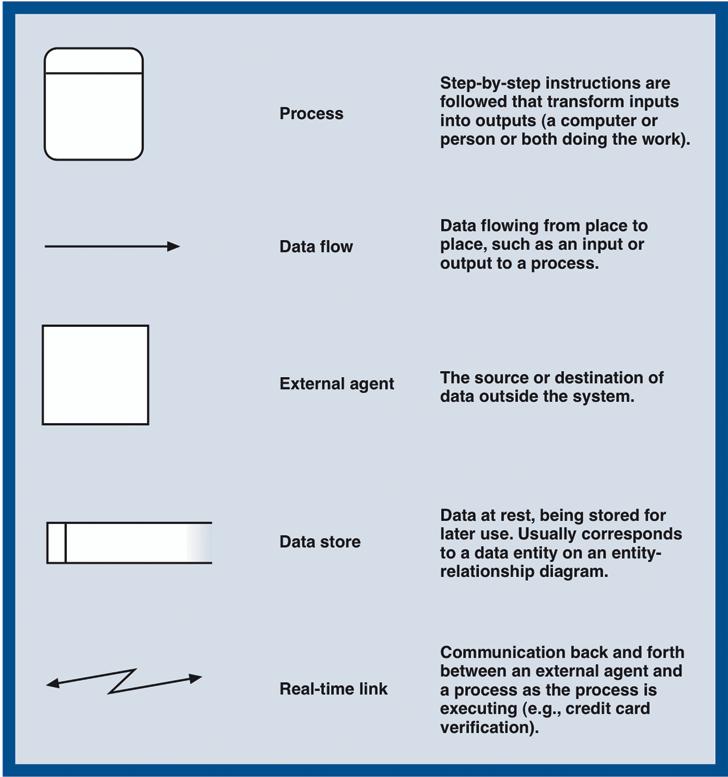
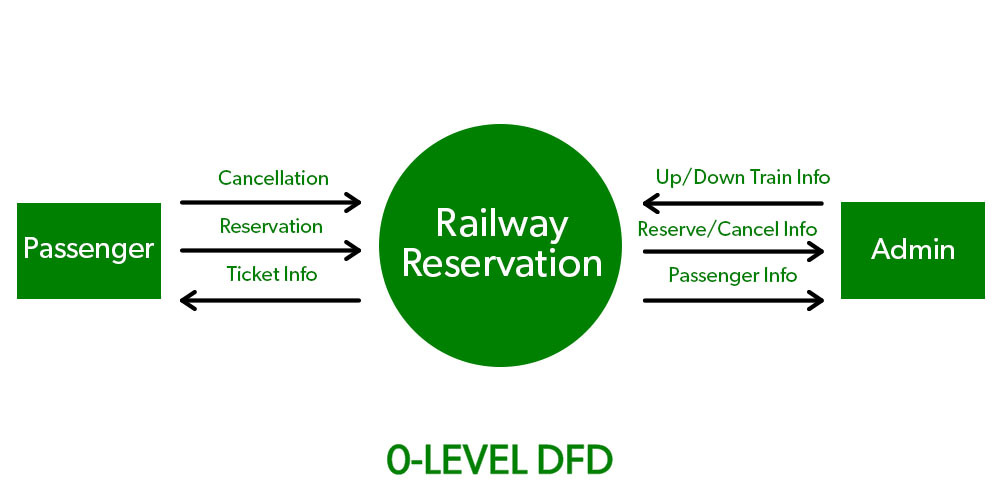
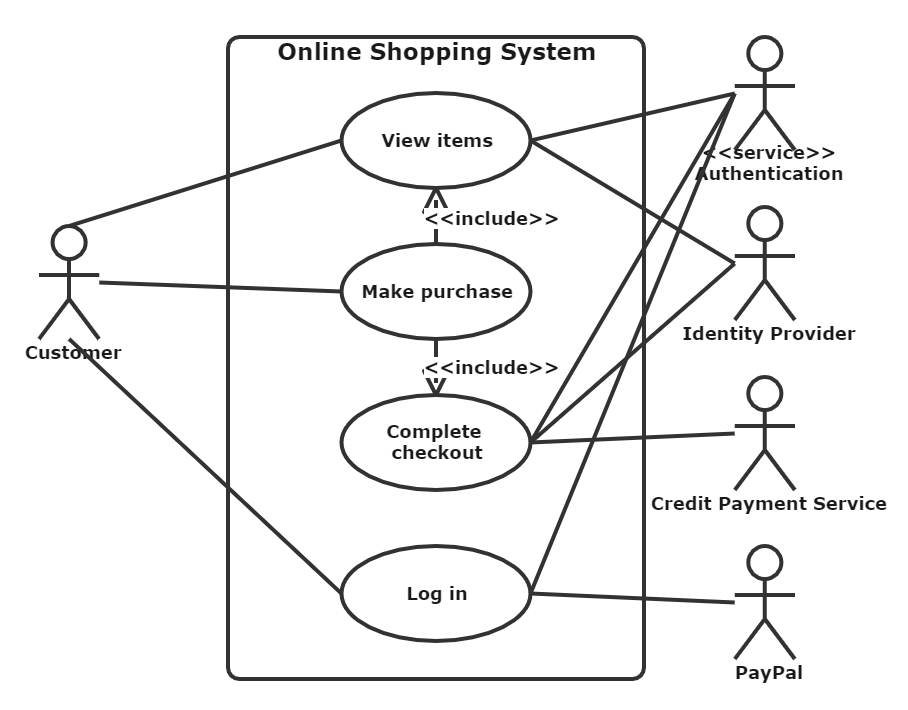
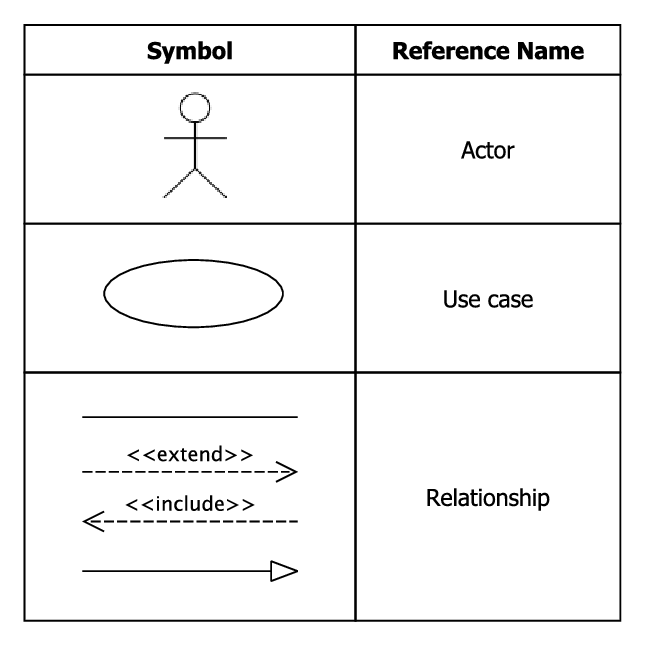
This article explores the similarity and difference between Data Flow Diagram and Use Case Diagram used in software design and development.

Data Flow Diagrams (DFDs) are primarily used to model the flow of data within a system. They depict the processes that manipulate the data, the data stores where information is held, and the data flows that represent the movement of data between these components. DFDs are hierarchical, with different levels (context level, level 0, level 1, etc.) used to represent increasing levels of detail about the system's data flow.

On the other hand, Use Case Diagrams are part of the Unified Modeling Language (UML) and are used to depict interactions between actors (users or external systems) and the system being modeled. They focus on the functional requirements of the system, showing how users interact with the system to achieve specific goals or tasks. Use Case Diagrams include actors, use cases (representing system functionalities), and the relationships between them.

Aleryani's study likely discusses the key characteristics, strengths, and limitations of both DFDs and Use Case Diagrams. For example, DFDs are excellent for modeling data flow and processes but may not capture user interactions and system functionalities as explicitly as Use Case Diagrams. Use Case Diagrams, on the other hand, are great for representing user-system interactions and functional requirements but may not provide as much detail about data flow and processing logic.

The study may also explore scenarios where one diagram type is more suitable than the other based on the specific requirements of the system being modeled. For complex systems, a combination of both DFDs and Use Case Diagrams, along with other modeling techniques, may be necessary to capture all aspects comprehensively. Overall, Aleryani's comparative study likely provides insights into the complementary nature of DFDs and Use Case Diagrams in system analysis and software development, highlighting their respective roles and contributions to understanding and modeling systems.