EXPERIMENT NO. 2

Aim: To perform the function oriented diagram:DFD and Structured chart.

Theory: Data Flow Diagram (DFD) for Car Showroom Management System

A DFD uses external entities, processes, data flows, and data stores to model the system.

Types of DFD

Data Flow Diagrams are either Logical or Physical.

• **Logical DFD -** This type of DFD concentrates on the system process, and flow of data in the System. For example in a Banking software system, how data is moved between different entities.

• **Physical DFD -** This type of DFD shows how the data flow is actually implemented in The system. It is more specific and close to the implementation.

DFD Components

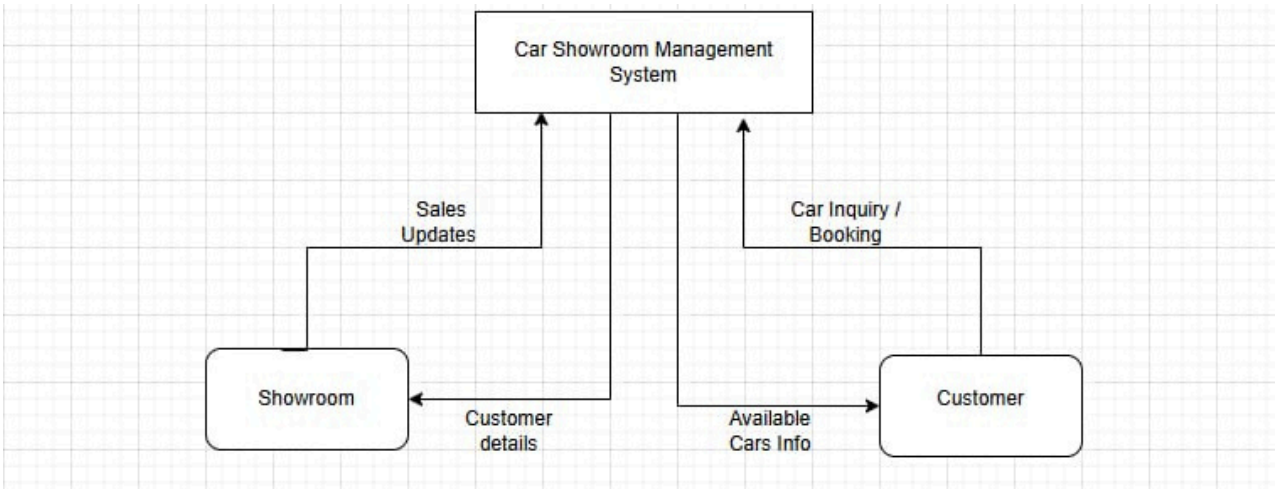
DFD can represent Source, destination, storage and flow of data using the following set of components –

- **External Entity (Rectangle)**: Customer provides a Car Order Request to the system.
- **Process (Rounded Rectangle)**: Process Sale/Booking transforms the order into a Sales Invoice.
- **Data Flow (Arrow):** Inventory Update flows from the process to the Car Inventory store.
- **Data Storage (Open-Ended Rectangle)**: Car Inventory holds details like model, VIN, and stock quantity.

Levels of DFD

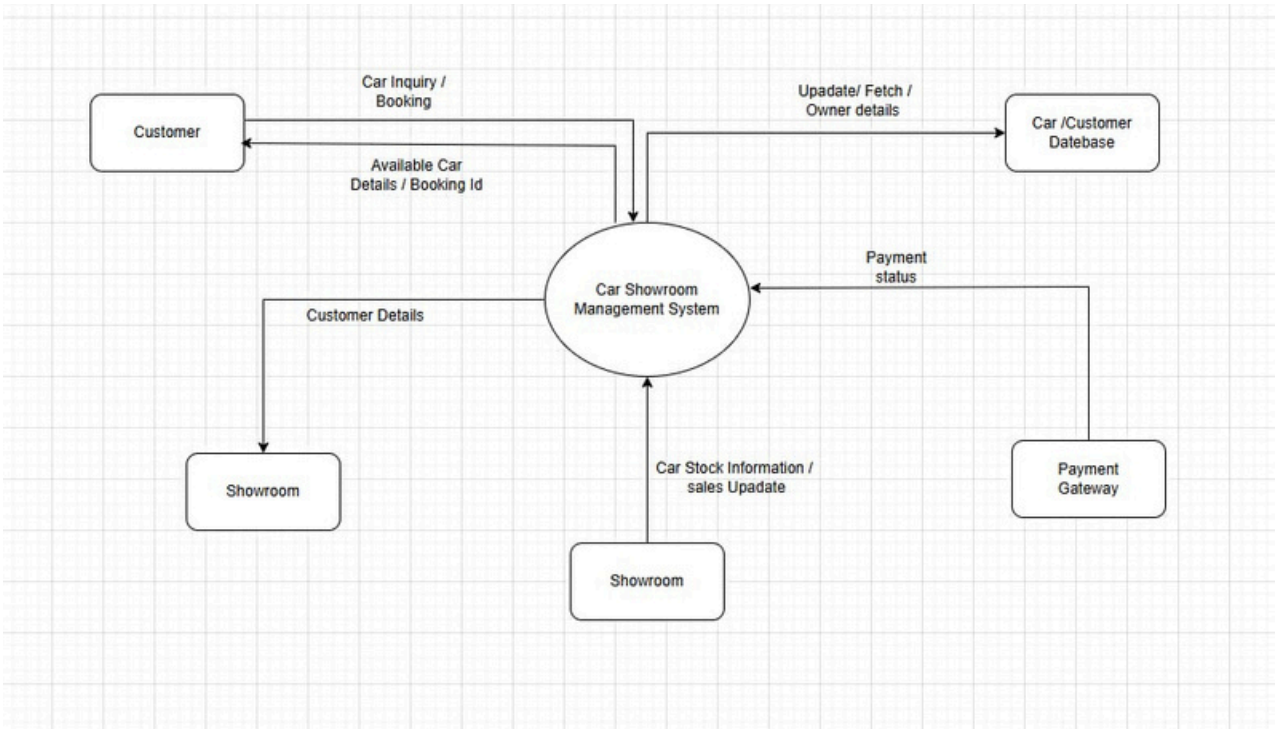
Level 0:

Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.



Level 1:

The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.



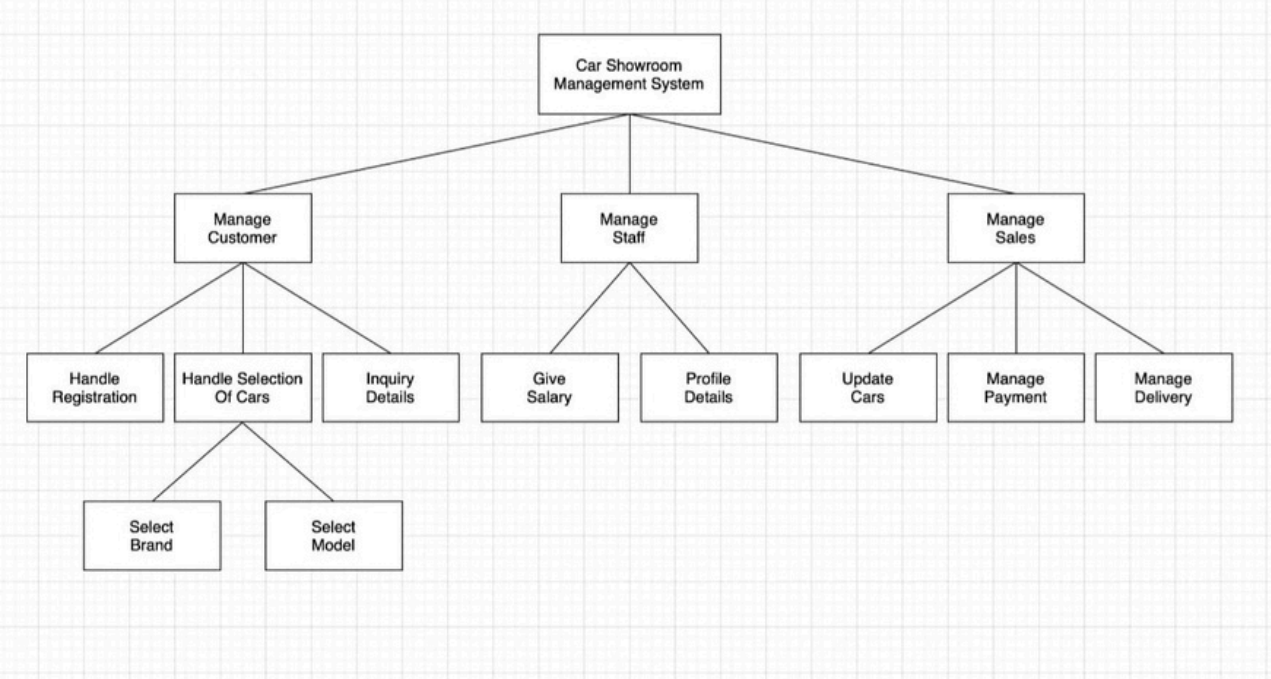
Level 2:

At this level, DFD shows how data flows inside the modules mentioned in Level 1. Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.

Structure Charts

The **Structured Chart** (or Structure Chart) illustrates the **hierarchical structure** of the system modules, showing the sequence of execution and the data/control information passed between them. It's used for **structured design** and maps well to a hierarchical programming structure. Here are the symbols used in construction of structure charts –

- **Module:** Represents a specific function or subroutine **(e.g., Validate Login, Search by Title).**
- **Control Flow:** Indicates one module calls another.
- **Data Flow:** Represents data passed as parameters from the calling module to the called module.
- **Control Flow:** Represents a flag or signal passed **(e.g., an error code or a status signal).**



Structure Charts of Car Showroom Management System