https://meeraacademy.com/dfd-diagram-for-online-food-ordering-system/

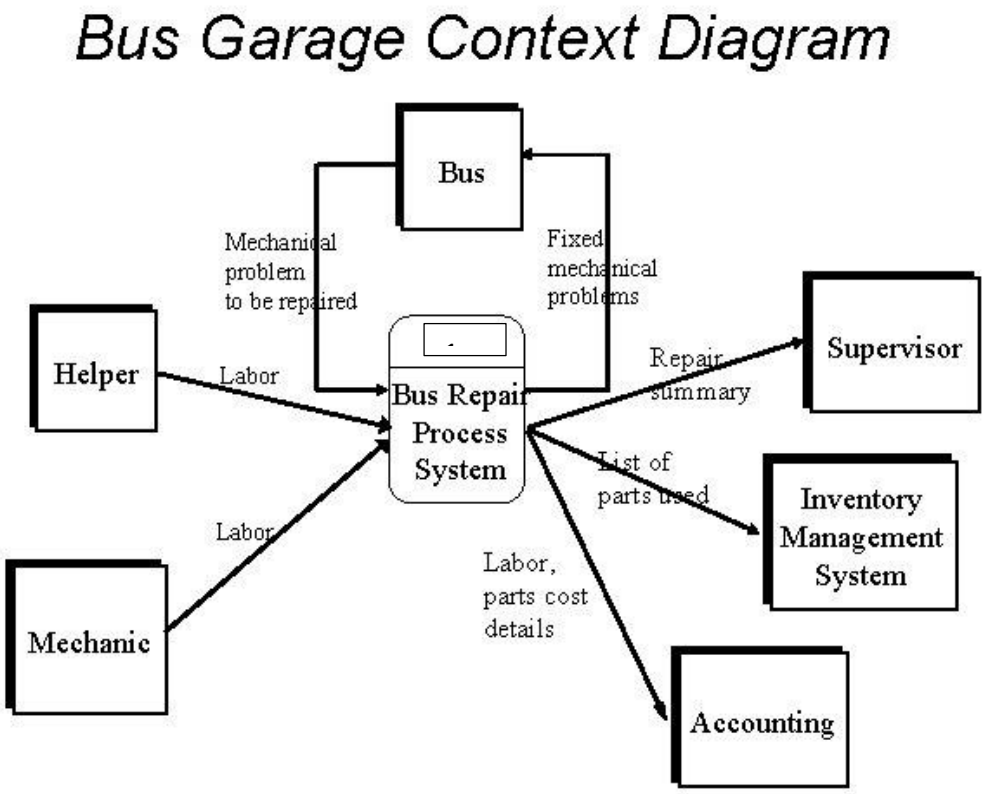
DFD Example: Bus Garage Repairs

- Buses come to a garage for repairs.
- A mechanic and helper perform the repair, record the reason for the repair and record the total cost of all parts used on a Shop Repair Order.
- Information on labor, parts and repair outcome is used for billing by the Accounting Department, parts monitoring by the inventory management computer system and a performance review by the supervisor.

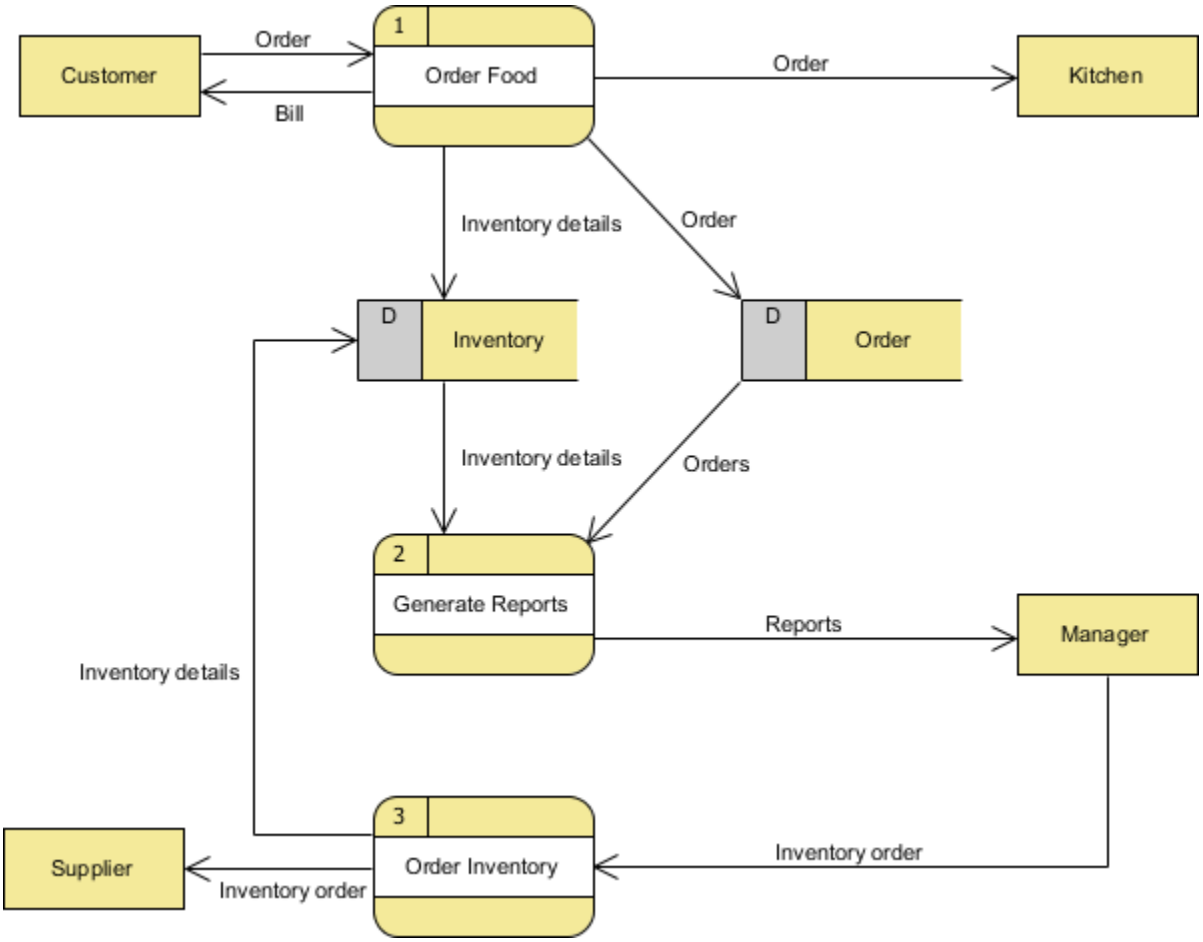
<u> Identify the entity, process, data store and data flow</u>**

- *External Entities*: Bus, Mechanic, Helper, Supervisor, Inventory Management System, Accounting Department, etc.
- *Key process* ("the system"): performing repairs and storing information related to repairs
- ***Processes*****:**
  - Record Bus ID and reason for repair
  - Determine parts needed
  - Perform repair
  - Calculate parts extended and total cost
  - Record labor hours, cost
- ***Data stores*****:**
  - Personnel file
  - Repairs file
  - Bus master list
  - Parts list
- ***Data flows*****:**
  - Repair order
  - Bus record

- Parts record
- Employee timecard
- Invoices



The Food Ordering System Example



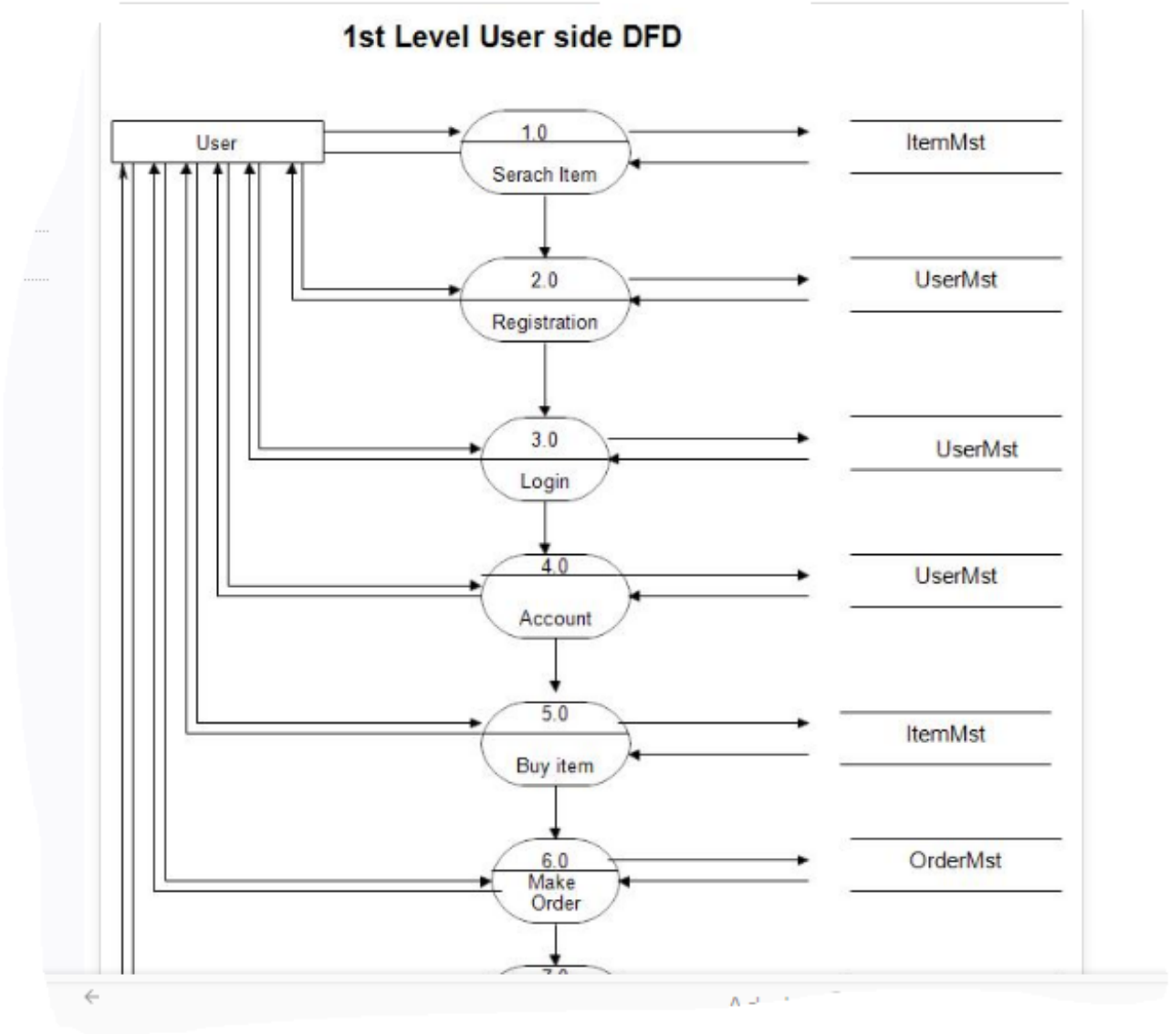
The Food Order System Data Flow Diagram example contains three processes, four external entities, and two data stores.

Based on the diagram, we know that a *Customer* can place an *Order*. The *Order Food* process receives the *Order*, forwards it to the *Kitchen*, store it in the *Order* data store, and store the updated *Inventory details* in the *Inventory* data store. The process also delivers a *Bill* to the *Customer*.

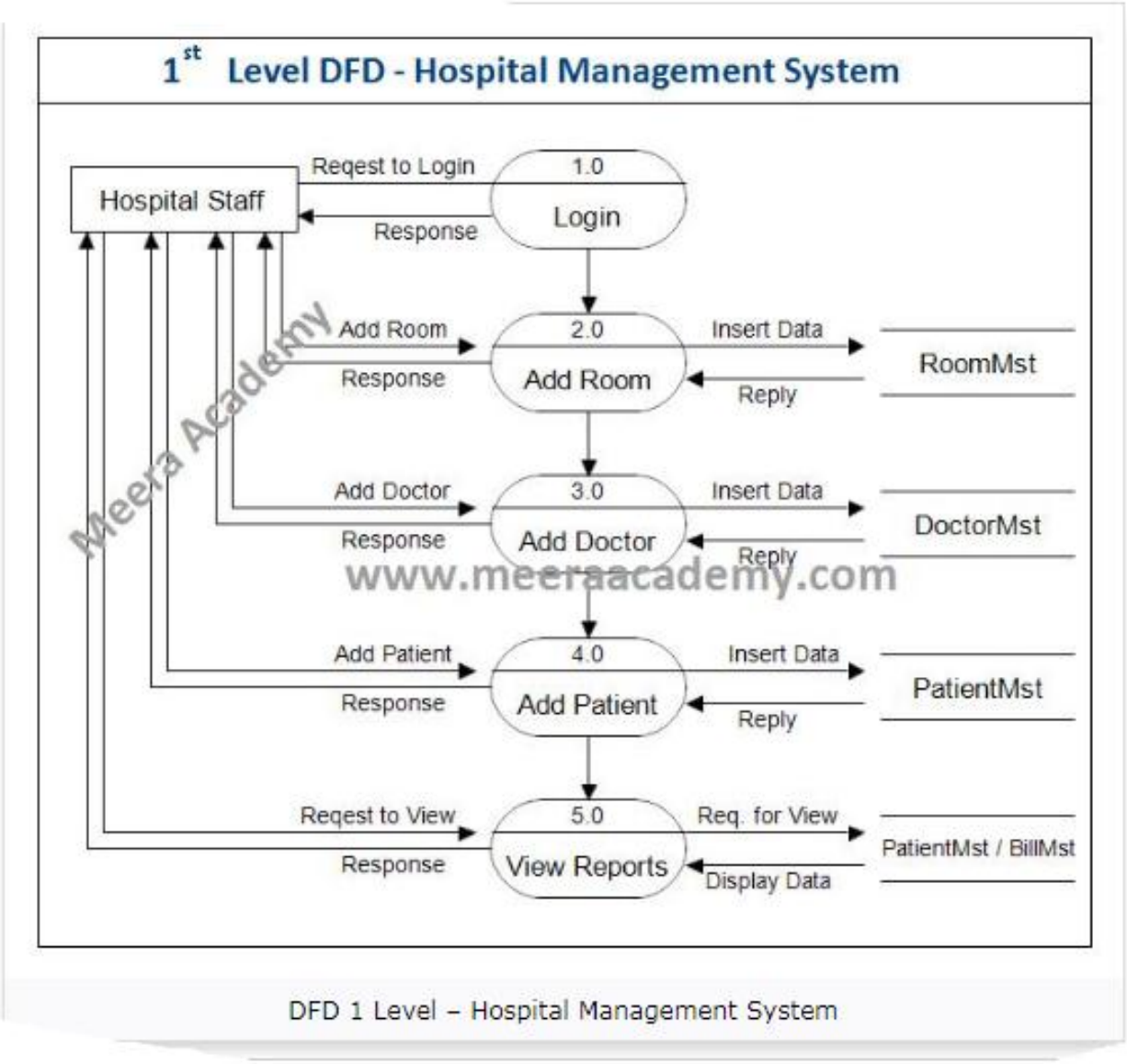
The *Manager* can receive *Reports* through the *Generate Reports* process, which takes *Inventory details* and *Orders* as input from the *Inventory* and *Order* data store respectively.

The *Manager* can also initiate the *Order Inventory* process by providing *Inventory order*. The process forwards the *Inventory order* to the *Supplier* and stores the updated *Inventory details* in the *Inventory* data store.

dfd diagram for online shopping website



DFD for Hospital Management System Project / Patient Monitoring System



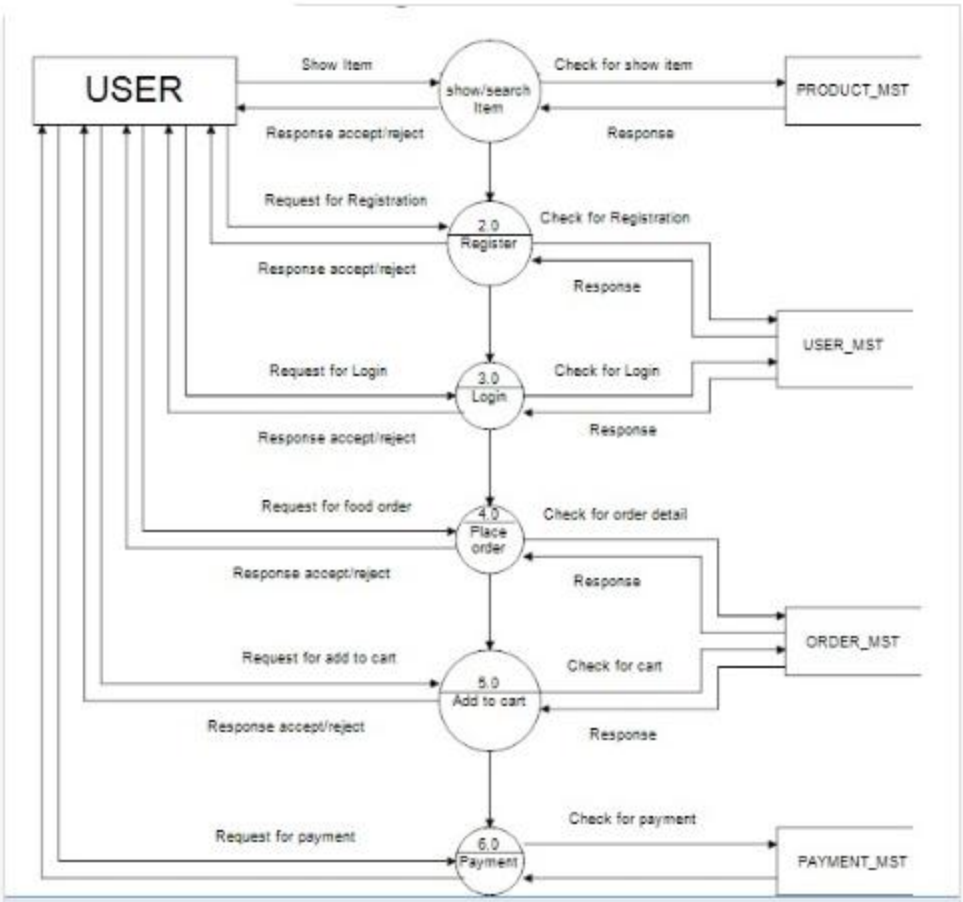
DFD 1 Level – Hospital Management System

Banking system DFD

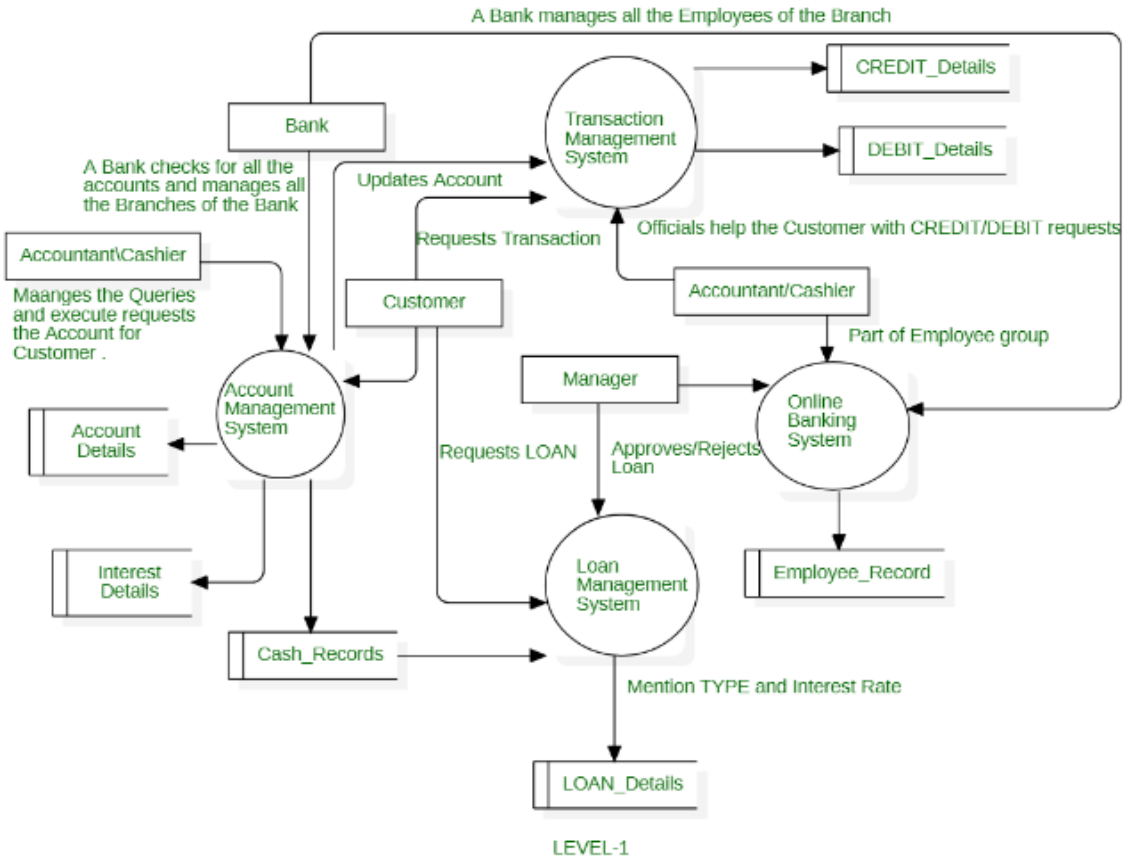
A **Bank Manager** provides **new account details** to the **Open Account** process which results in **Customer details** being persisted in the **Customer Database** data store and **Account details** being persisted in the **Account Database** data store. Although we have used the phrase 'results in' as part of this explanation, the DFD implies no such cause and effect; all it shows is that the **Open Account** process can read in data from the **Bank Manager** interface and write out data to the **Customer Database** and **Account Database** data stores in no particular order.

A **Customer** actor using the **Online Banking** Login process must provide some data in the form of a set of **Login credentials** such as a user name and password.
A **Customer** actor can receive a **Money amount** from the **Withdraw** process and can supply a **Money amount** to the **Deposit** process; in either case causing (although this causation cannot be explicitly modeled) an **Account balance update** to the **Account Database** data store.
A **Customer** actor can initiate the **Transfer Funds** process, to which he or she must provide an **Account destination and money amount**. The **Transfer Funds** process can send a **Money amount** to another bank via the **Other Bank** interface.
Just like the **Customer** actor, a **Third Party** actor can make use of the **Deposit** process (but obviously not the **Withdraw** process) by supplying a **Money amount**.

dfd diagram for online food ordering system



dfd for online food ordering system in asp.net project



Level-1 DFD – Online Banking System