

Practical 5

Data Flow Diagram (DFD)

- A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system.
- A neat and clear DFD can depict the right amount of the system requirement graphically.
- Data Flow Diagram is a high-level diagram that is drawn with a high degree of abstraction.
- It can be manual, automated, or a combination of both.
- It shows how data enters and leaves the system, what changes the information, and where data is stored.
- The objective of a DFD is to show the scope and boundaries of a system as a whole.
- It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.
- The DFD is also called a data flow graph.
- A data flow diagram (DFD) represents graphically a flow of data within a system. It illustrates how data is input and output from the system.
- It also shows destinations, storage, and sources of the information in the system.
- In other words, DFD represents the information flow as well as where data comes from, where data goes and how it is stored.

Data flow diagram has 4 major elements:

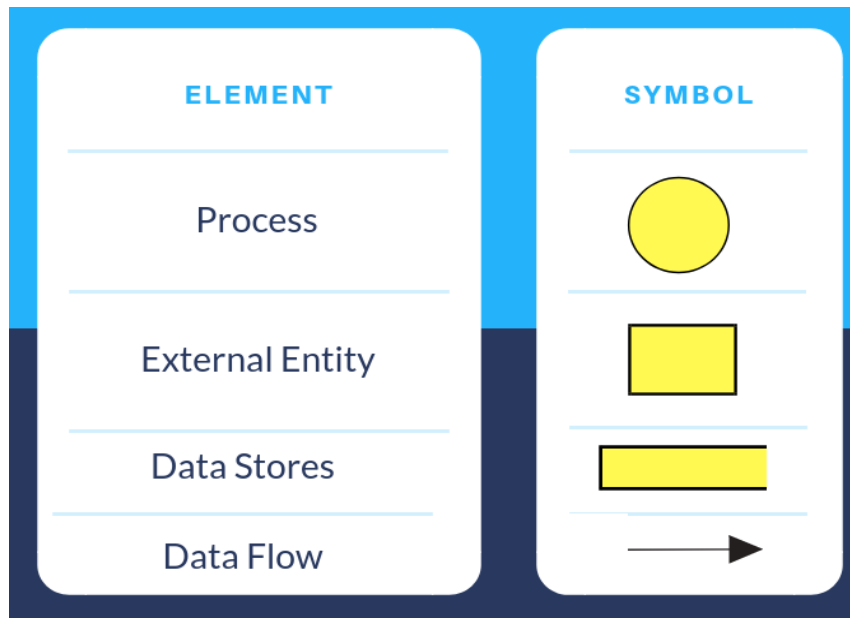
- **Processes [verb phrase - system/subsystem name]**– the main activities that are happening within the system boundary. The process can be as simple as collecting customer data and storing it in the company database. Also, it can be a very complicated process such as creating a report containing bank contracts with customers of all bank clones in a region.
- **External entities [nouns]**– the sources of information coming to or leaving the system. External entities are outside systems such as people (customers, stakeholders, managers), organizations, computers and other systems that send or receive data from our system.
- **Data stores [nouns]** – places where data is held such as files or repositories. Data stores show information that is not moving.
- **Data flows [name of data]**– illustrate the movements that data have between the external entities, data stores, and the processes.

Some Important Points about DFD's

- All names should be unique. This makes it easier to refer to elements in the DFD.
- Remember that DFD is not a flow chart. Arrows is a flowchart that represents the order of events; arrows in DFD represent flowing data. A DFD does not involve any order of events.
- Suppress logical decisions. [ordering of events makes no sense in a DFD]
- Do not become bogged down with details.

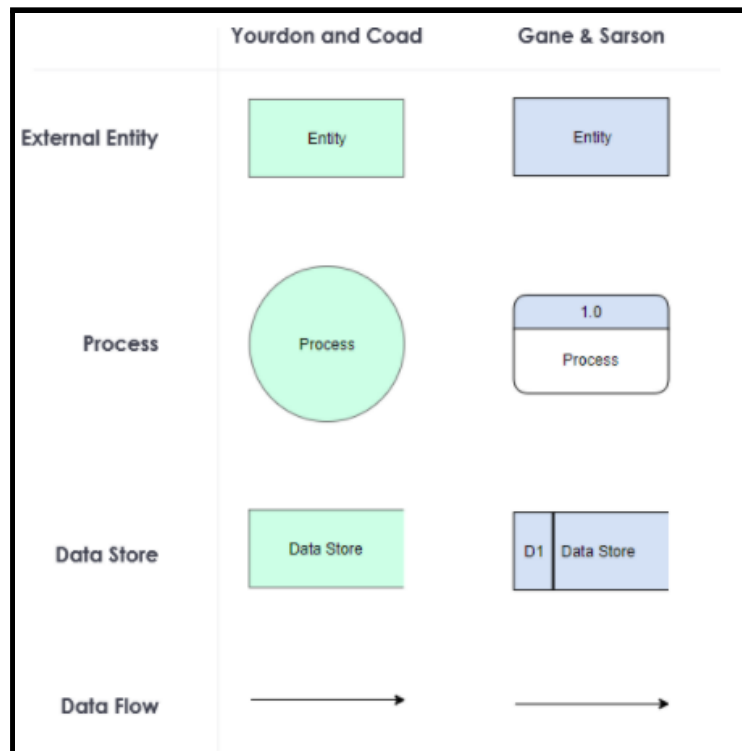
Symbols used in data flow diagrams

Each of the above elements has a symbol that represents it. Typically, data flow diagram uses the following symbols:



The above ones are so-called symbols of Yourdon and Coad.

There is also the symbol system of Gane and Sarson, but in our data flow diagram examples, we will use Yourdon and Coad symbols as they are easier for drawing and remembering.



DFD rules, guidelines, and tips:

Creating data flow diagrams requires some guidelines and rules that should be followed. These guidelines make DFD easily understandable and lucid.

Here are some of the key rules and tips.

1. Each process has at least one outgoing data flow and at least one ingoing data flow.



2. Each process can go to any other symbol (other processes, data store, and entities).



3. Each data store should have at least one incoming and at least one outgoing data flow.



4. Entities must be connected to a process by a data flow.



5. Data flows cannot cross with each other.

6. Data stores cannot be connected to external entities. Otherwise, it means you're allowing an external entity access to your data files and stores.

7. The labels of processes can be verb phrases. Data stores are displayed by nouns.

8. Data flows cannot run between two external entities without going through a process (as you will see in the data flow diagram examples below).

Advantages and disadvantages of data flow diagrams:

Before going further to data flow diagram examples, let's see what are some key benefits and cons of DFD.

Advantages:

- A graphical technique that is relatively easy to understand for stakeholders and other users.
- Provides a detailed view of the system components and boundaries.
- Provide clear and detailed information about the processes within a system.
- Shows the logic of the data flow.
- Presents a functional breakdown of the system.
- Used as a part of the system documentation.

Disadvantages:

- Takes a long time to create.
- Does not give any information about the timing, sequence, and synchronization of processes i.e. data flow diagrams do not specify when the processes are performed. Therefore it should not be confused with a process or flowchart diagram which can illustrate these things.
- Sometimes it might be difficult for non-technical users to understand the diagram.

Data Flow Diagram Examples

1. Context data flow diagram: definition and example with explanation

When it comes to simple data flow diagram examples, context one has the top place.

Context data flow diagram (also called Level 0 diagram) uses only one process to represent the functions of the entire system.

It does not go into details as marking all the processes.

The purpose is to express the system scope at a high level as well as to prevent users from deep down into complex details.

The major advantage of context DFD is simplicity.

Key context DFD characteristics:

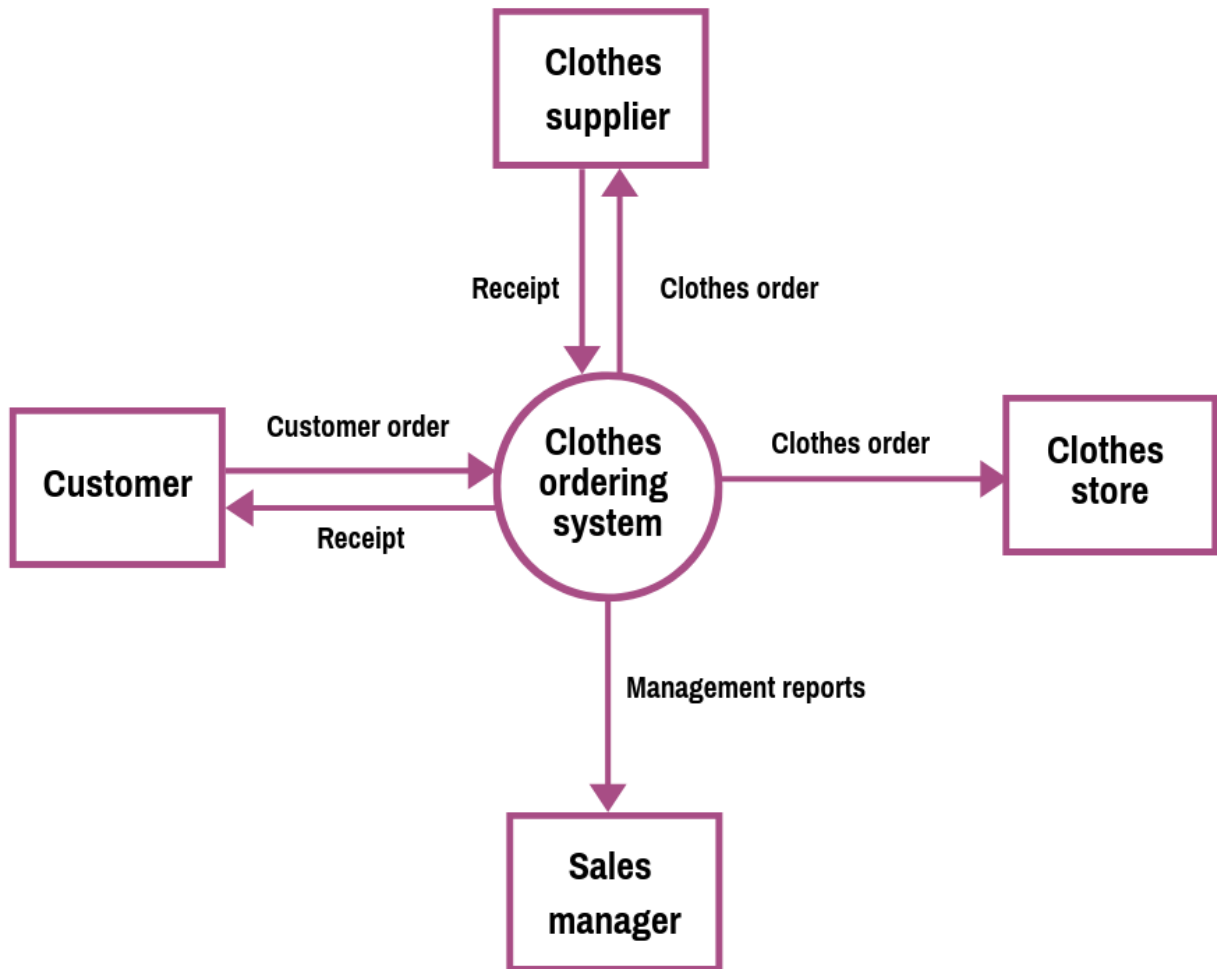
- Simple to draw.
- No need for technical knowledge to understand it.
- Shows the system boundaries.

Steps for creating a context DFD:

- **Step1:** Define the process.
- **Step2:** Create a list of all external entities (all people and systems).
- **Step3:** Create a list of the data flows.
- **Step4:** Draw the diagram.

Let's illustrate the things with a context data flow diagram example.

Below is shown a simple context DFD drawn for a Clothes Ordering System and explanation.



Now, let's explain how we create the diagram.

Step 1: Define the process.

As it is a context data flow diagram, the process is only one. In our case, it is *Clothes Ordering System*. Draw a rectangle for the process.

Step 2: Create the list of all external entities.

In our example, the external entities are: *Customer*, *Clothes Store*, *Clothes Supplier*, and *the Sales Manager*. These are all entities who are involved with our system. Also, now you can draw a rectangle for each of the entities.

Step 3: Create a list of the data flows.

In between our process and the external entities, there are data flows that show a brief description of the type of information exchanged between the entities and the system.

In our example, the list of data flows includes: *Customer Order*, *Receipt*, *Clothes Order*, *Receipt*, *Clothes Order*, and *Management Report*.

Now, connect the rectangles with arrows signifying the data flows. If data flows both ways between any two rectangles, create two individual arrows.

Step4: This is our diagram.

2. Level 1 data flow diagram: definition and example with explanation

As you saw above, the DFD contains only one process and does not illustrate any data store.

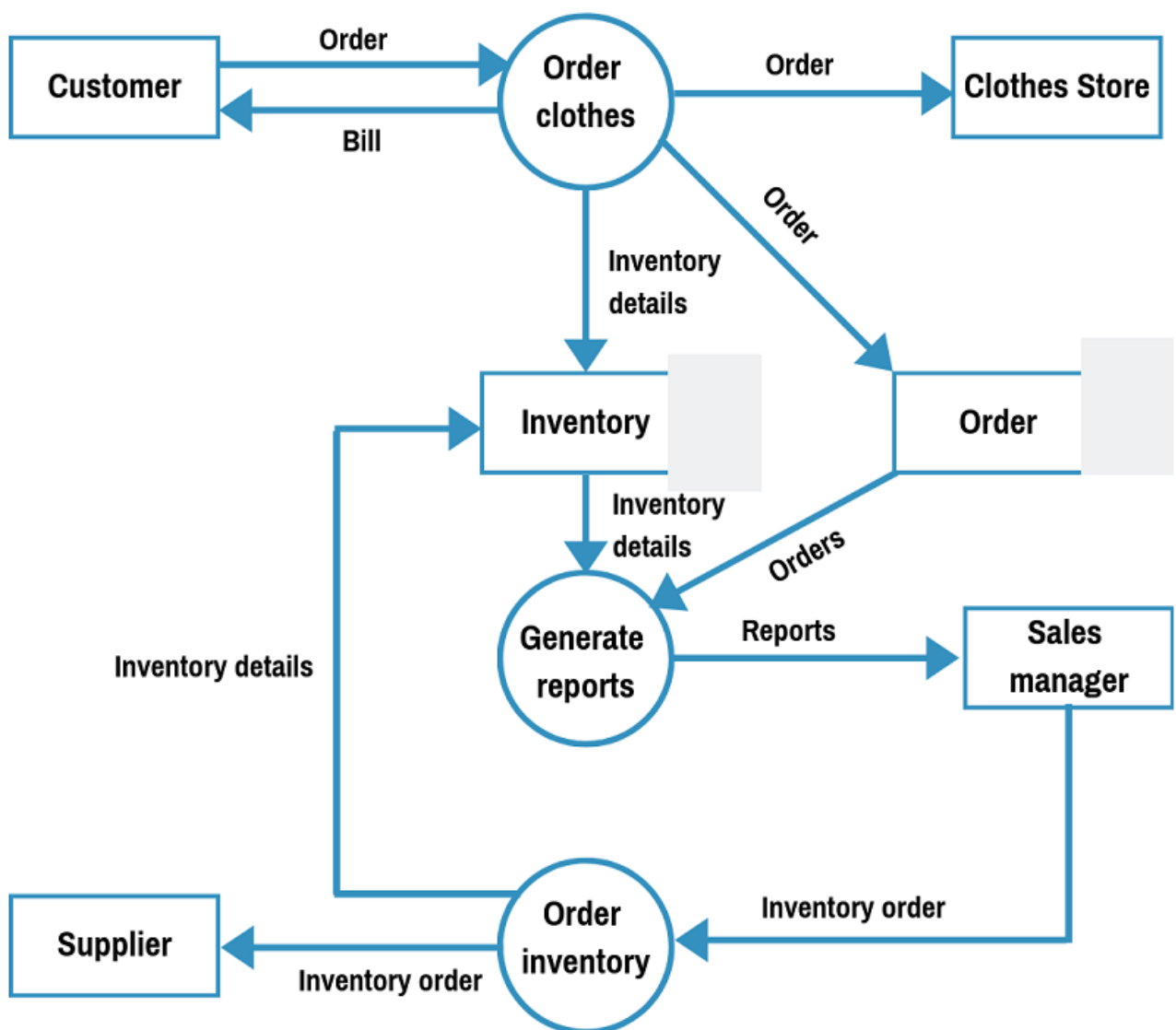
This is the main difference with level 1 DFD.

Level 1 DFD breaks down the main process into subprocesses that can then be seen on a more deep level. Also, level 1 DFD contains data stores that are used by the main process.

Steps for creating a context DFD:

- **Step1:** Define the processes (the main process and the subprocesses).
- **Step2:** Create a list of all external entities (all people and systems).
- **Step3:** Create a list of the data stores.
- **Step4:** Create a list of the data flows.
- **Step5:** Draw the diagram.

Here is our level 1 data flow example – a decomposition of the Clothes Ordering System illustrated in the context DFD.



As you see, the above Clothes Order System Data Flow Diagram Example shows three processes, four external entities, and also two data stores.

Here are the steps for creating the level 1 DFD:

Step 1: Define the processes.

The three processes are: *Order Clothes, Generate Reports, and Order Inventory.*

Step 2: Create the list of all external entities.

The external entities are: *Customer, Clothes Store, Sales Manager, and Supplier*

Step 3: Create the list of the data stores.

These are: *Order and Inventory*

Step 4: Create the list of the data flows

Data flows are: *Order, Bill, Order, Order, Inventory details, Inventory details, Orders, Reports, Inventory Order, Inventory Order, Inventory details.*

Step5: Create the diagram.

Extra Example [DFD for Lemonade Stand]:

Example

Create a context level diagram identifying the sources and sinks (users).

Customer Order
Serve Product
Collect Payment

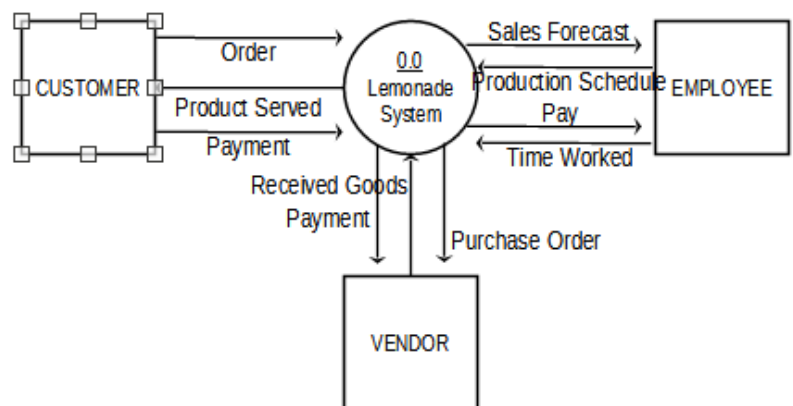
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

Construct Context Level DFD
(identifies sources and sink)

Context Level DFD



Example

Create a level 1 diagram identifying the logical subsystems that may exist.

Customer Order
Serve Product
Collect Payment

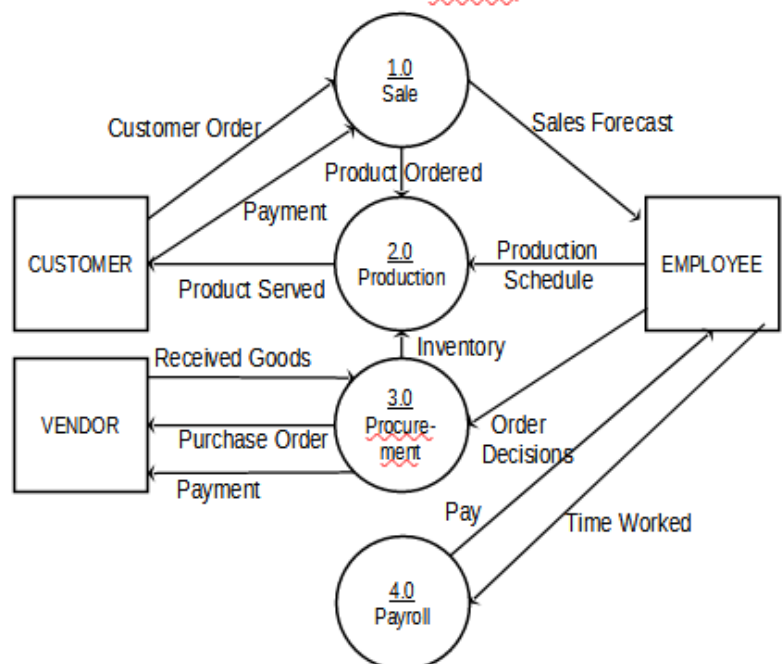
Produce Product
Store Product

Order Raw Materials
Pay for Raw Materials

Pay for Labor

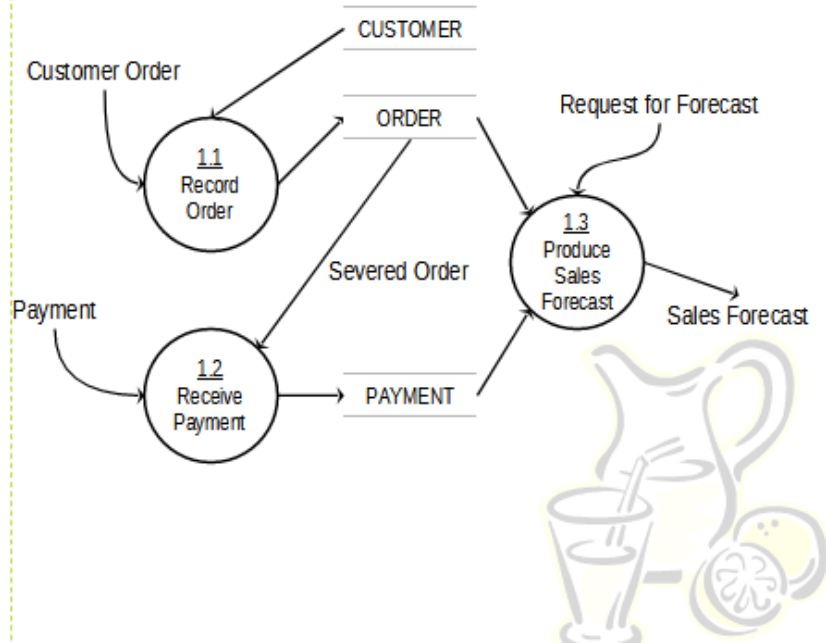
3. Construct Level 1 DFD
(identifies manageable sub processes)

Level 1 DFD



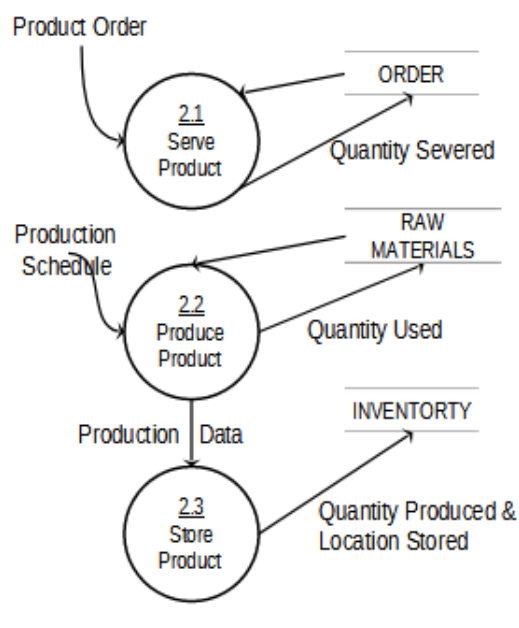
Customer Order
Serve Product
Collect Payment

Produce Product
Store Product
Order Raw Materials
Pay for Raw Materials
Pay for Labor

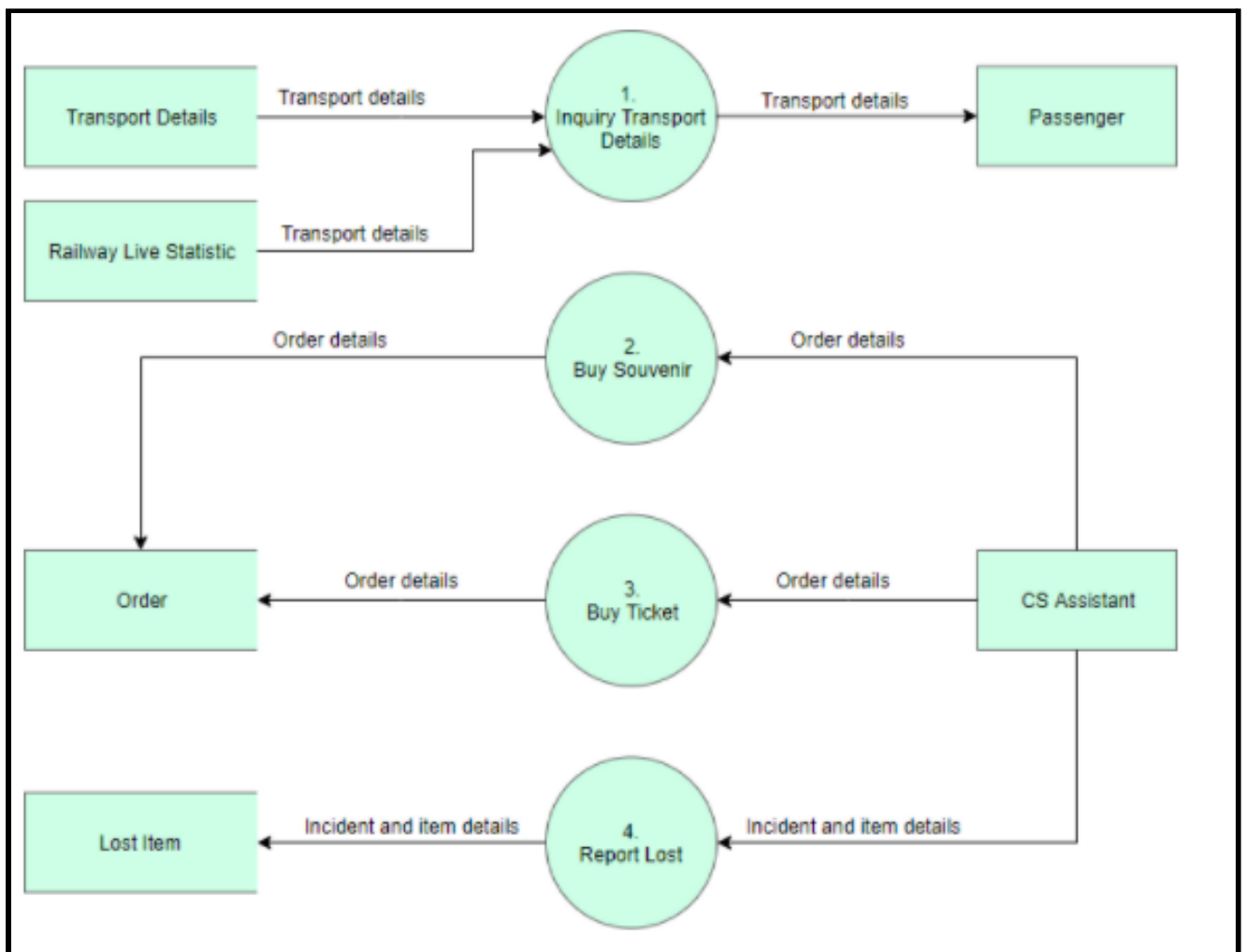
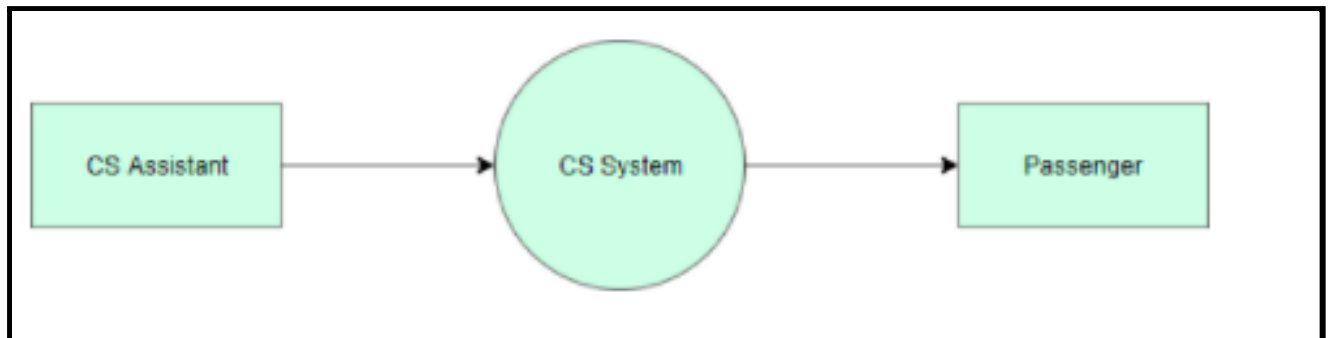


Customer Order
Serve Product
Collect Payment

Produce Product
Store Product
Order Raw Materials
Pay for Raw Materials
Pay for Labor



Extra Example [DFD for Customer Service-Railway Service]:



- Based on the diagram, we know that a Passenger can receive Transport details from the Inquiry Transport Details process, and the details are provided by the data stores Transport Details and Railway Live Statistic.
- CS Assistant can initiate the Buy Souvenir process, which will result in having the Order details stored in the Order data store.

- Although the customer is the real person who buys souvenirs, it is the CS Assistant who accesses the system for storing the order details. Therefore, we make the data flow from the CS Assistant to the Buy Souvenir process.
- CS Assistant can also initiate the Buy Ticket process by providing Order details and the details will be stored again in the Order data store.
- The data store Order which is drawn here does not necessarily imply a real order database or order table in a database.
- Finally, the CS Assistant can initiate the Report Lost process by providing the Incident and item details and the information will be stored in the Lost Item database.