

The Food Ordering System Example

Resources

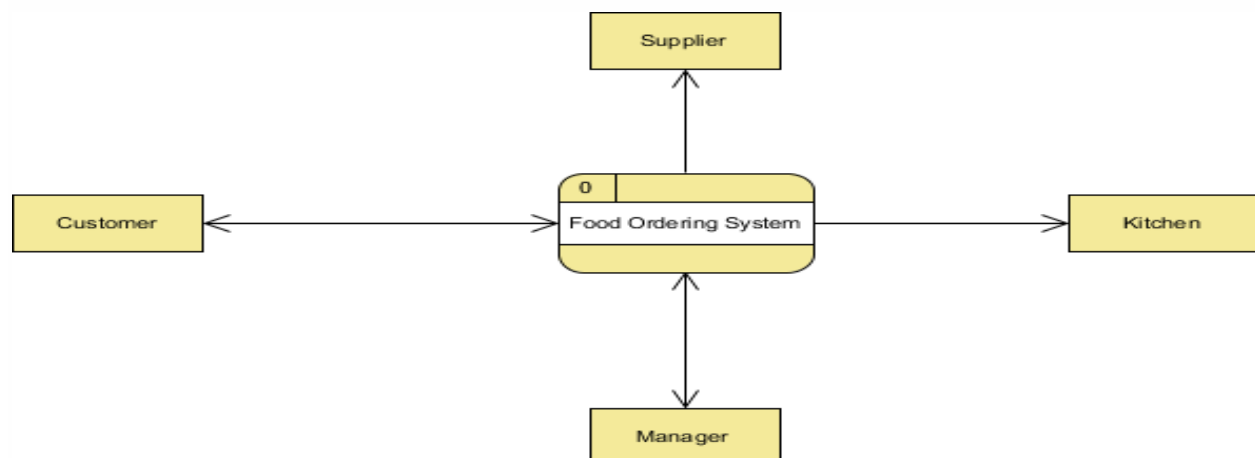
1. [Food-Ordering-System.vpp](#)

Context DFD

A context diagram is a data flow diagram that only shows the top level, otherwise known as Level 0. At this level, there is only one visible process node that represents the functions of a complete system in regard to how it interacts with external entities. Some of the benefits of a Context Diagram are:

1. Shows the overview of the boundaries of a system
2. No technical knowledge is required to understand with the simple notation
3. Simple to draw, amend and elaborate as its limited notation

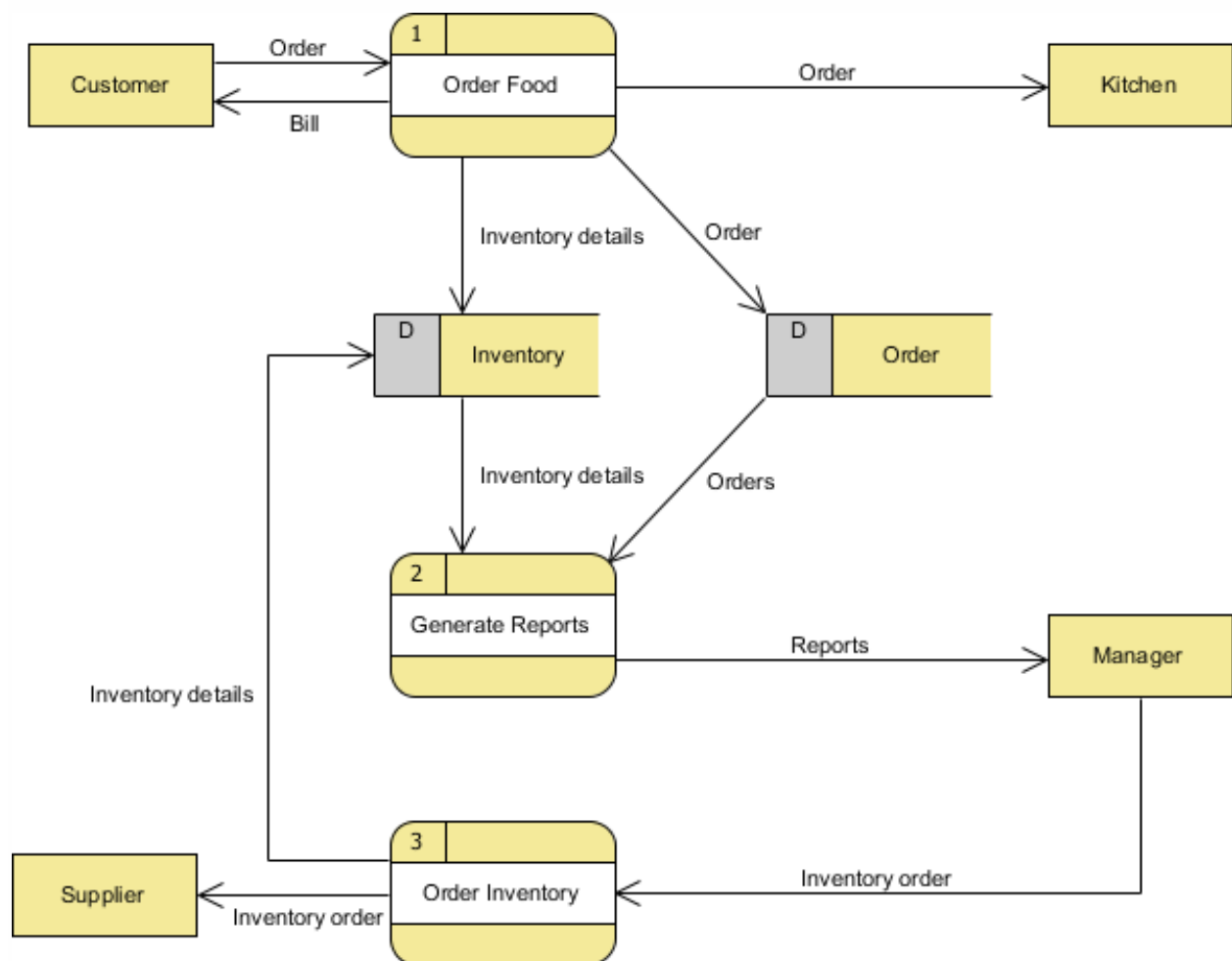
The figure below shows a context Data Flow Diagram that is drawn for a Food Ordering System. It contains a process (shape) that represents the system to model, in this case, the "*Food Ordering System*". It also shows the participants who will interact with the system, called the external entities. In this example, *Supplier*, *Kitchen*, *Manager* and *Customer* are the entities who will interact with the system. In between the process and the external entities, there are data flow (connectors) that indicate the existence of information exchange between the entities and the system.



Context DFD is the entrance of a data flow model. It contains one and only one process and does not show any data store.

Data Flow Diagram:

The figure below shows the level 1 DFD, which is the decomposition (i.e. break down) of the Food Ordering System process shown in the context DFD. Read through the diagram and then we will introduce some of the key concepts based on this diagram.



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*.

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.

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.

Data Flow Diagram Tips and Cautions

Tips

1. Process labels should be verb phrases; data stores are represented by nouns
2. A data store must be associated to at least a process
3. An external entity must be associated to at least a process
4. Don't let it get too complex; normally 5 - 7 average people can manage processes
5. DFD is non-deterministic - The numbering does not necessarily indicate sequence, it's useful in identifying the processes when discussing with users
6. Data stores should not be connected to an external entity, otherwise, it would mean that you're giving an external entity direct access to your data files
7. Data flows should not exist between 2 external entities without going through a process
8. A process that has inputs, but no outputs is considered to be a black-hole process