## Data Flow Diagram

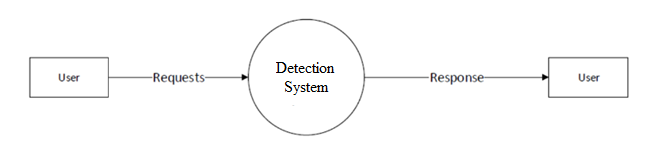
Data flows are data structures in motion, while data stores are data structures. Data flows are paths or ‘pipe lines’, along which data structures travel, whereas the data stores are place where data structures are kept until needed.

Data flows are data structures in motion, while data stores are data structures at rest. Hence it is possible that the data flow and the data store would be made up of the same data structure. Data flow diagrams is a very handy tool for the system analyst because it gives the analyst the overall picture of the system, it is a diagrammatic approach.

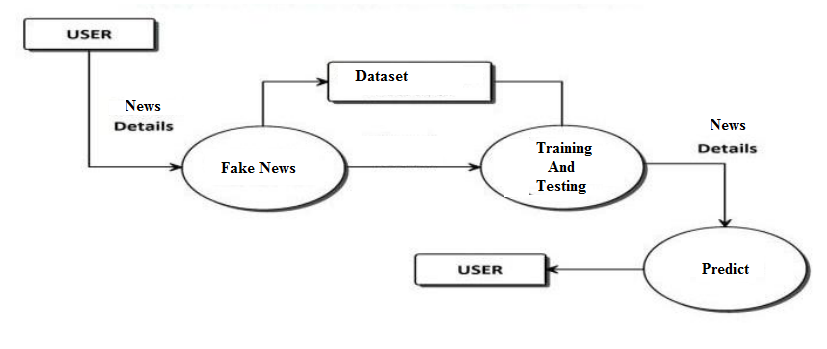
A DFD is a pictorial representation of the path which data takes from its initial interaction with the existing system until it completes any interaction. The diagram will describe the logical data flows dealing the movements of any physical items.

The DFD also gives the insight into the data that is used in the system i.e., who actually uses it is temporarily stored. A DFD does not show a sequence of steps. A DFD only shows what the different process in a system is and what data flows between them.

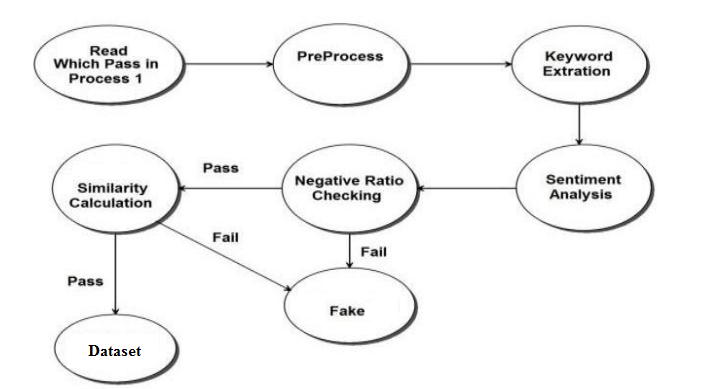
**Level 1.0**



**Level 1.1**



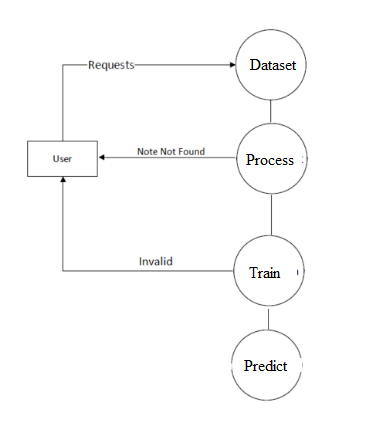
**Level 1.1**

****

## Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The figure shows the use case diagram for the system.

The following figure shows the use case diagram:



## System Sequence Diagram

A system sequence diagram is, as the name suggests, a type of sequence diagram in UML. These charts show the details of events that are generated by actors from outside the system. Standard sequence diagrams show the progression of events over a certain amount of time, while system sequence diagrams go a step further and present sequences for specific use cases. Use case diagrams are simply another diagram type which represents a user's interaction with the system. An SSD shows – for one particular scenario of a use case –

* The events that external actors generate
* Their order
* Inter-system events

SSDs are derived from use cases; SSDs are often drawn for the main success scenarios of each use case and frequent or complex alternative scenarios. System sequence diagrams are used as input for object design.

