**EXPERIMENT NO:- 4**

**Aim: Study and design DFD Diagrams (Level-0, Level-1, Level-2) for allotted projects.**

**Explanation:**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information](http://en.wikipedia.org/wiki/Information_system) [system](http://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data](http://en.wikipedia.org/wiki/Data_processing) [processing](http://en.wikipedia.org/wiki/Data_processing)

(structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

**Diagram Notations in Software Tools.**

Now we'd like to briefly introduce to you a few diagram notations which you'll see in the tutorial below.

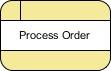
# External Entity

An external entity can represent a human, system or subsystem. It is where certain data comes from or goes to. It is external to the system we study, in terms of the business process. For this reason, people use to draw external entities on the edge of a diagram.



# Process

A process is a business activity or function where the manipulation and transformation of data takes place. A process can be decomposed to finer level of details, for representing how data is being processed within the process.



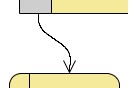
# Data Store

A data store represents the storage of persistent data required and/or produced by the process. Here are some examples of data stores: membership forms, database table, etc.



# Data Flow

A data flow represents the flow of information, with its direction represented by an arrow head that shows at the end(s) of flow connector.



# Levels of Data Flow Diagrams

DFDs can range from simple overviews to complex, granular representations of a system or process with multiple levels, starting with level 0. The most common and intuitive DFDs are level 0 DFDs, also called context diagrams. They’re digestible, high-level overviews of the flow of information through a system or process, so almost anyone can understand it.

**Level 0: Context Diagram**

This DFD level focuses on high-level system processes or functions and the data sources that flow to or from them. Level 0 diagrams are designed to be simple, straightforward overviews of a process or system.

**Level 1: Process Decomposition**

While level 1 DFDs are still broad overviews of a system or process, they’re also more detailed — they break down the system’s single process node into subprocesses.

**Level 2: Deeper Dives**

The next level of DFDs dive even deeper into detail by breaking down each level 1 process into granular subprocesses.

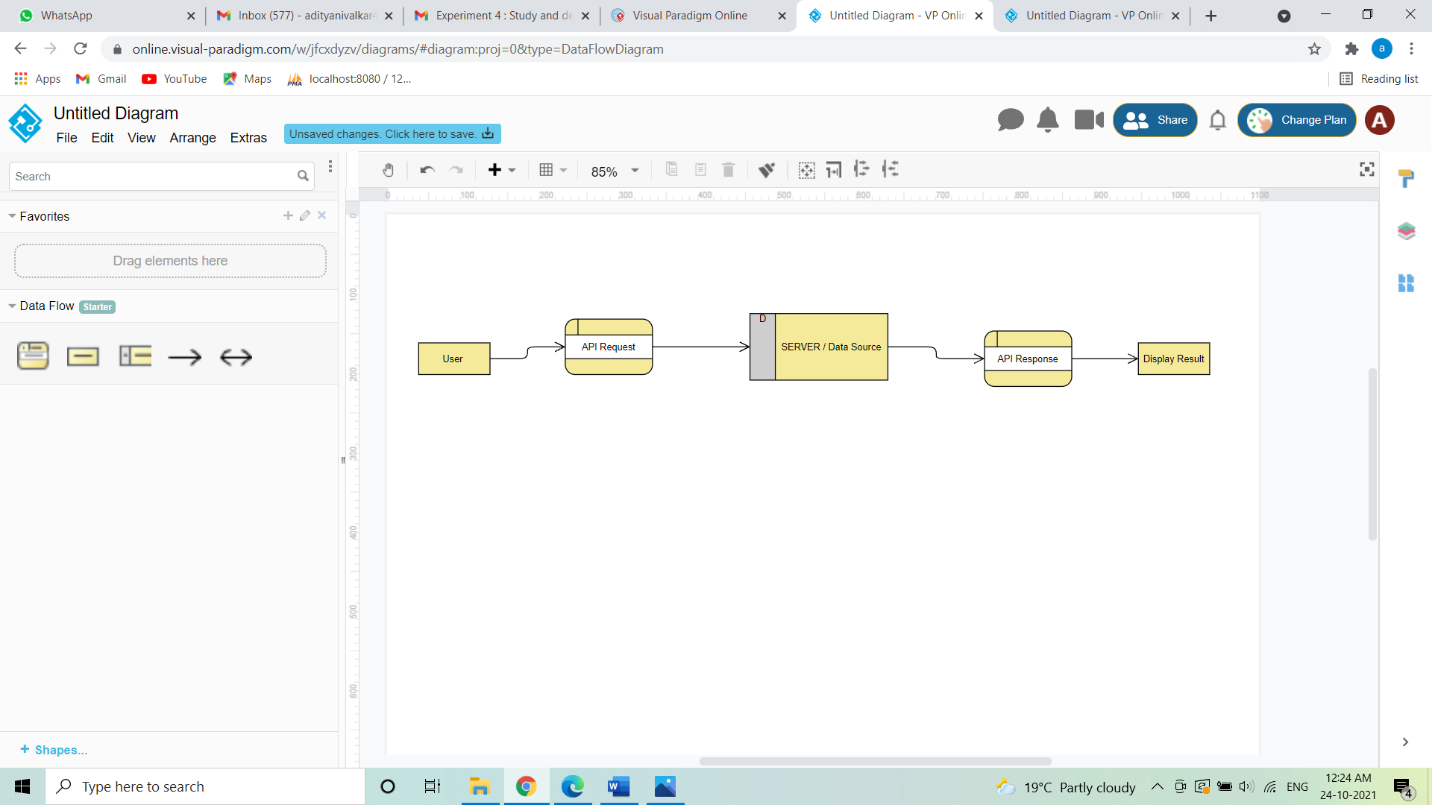
**Level3: Increasing Complexity**

Level 3 and higher-numbered DFDs are uncommon. This is largely due to the amount of detail required, which defeats its original purpose of being easy to understand.

# Data Flow Diagram Of Online Book Store

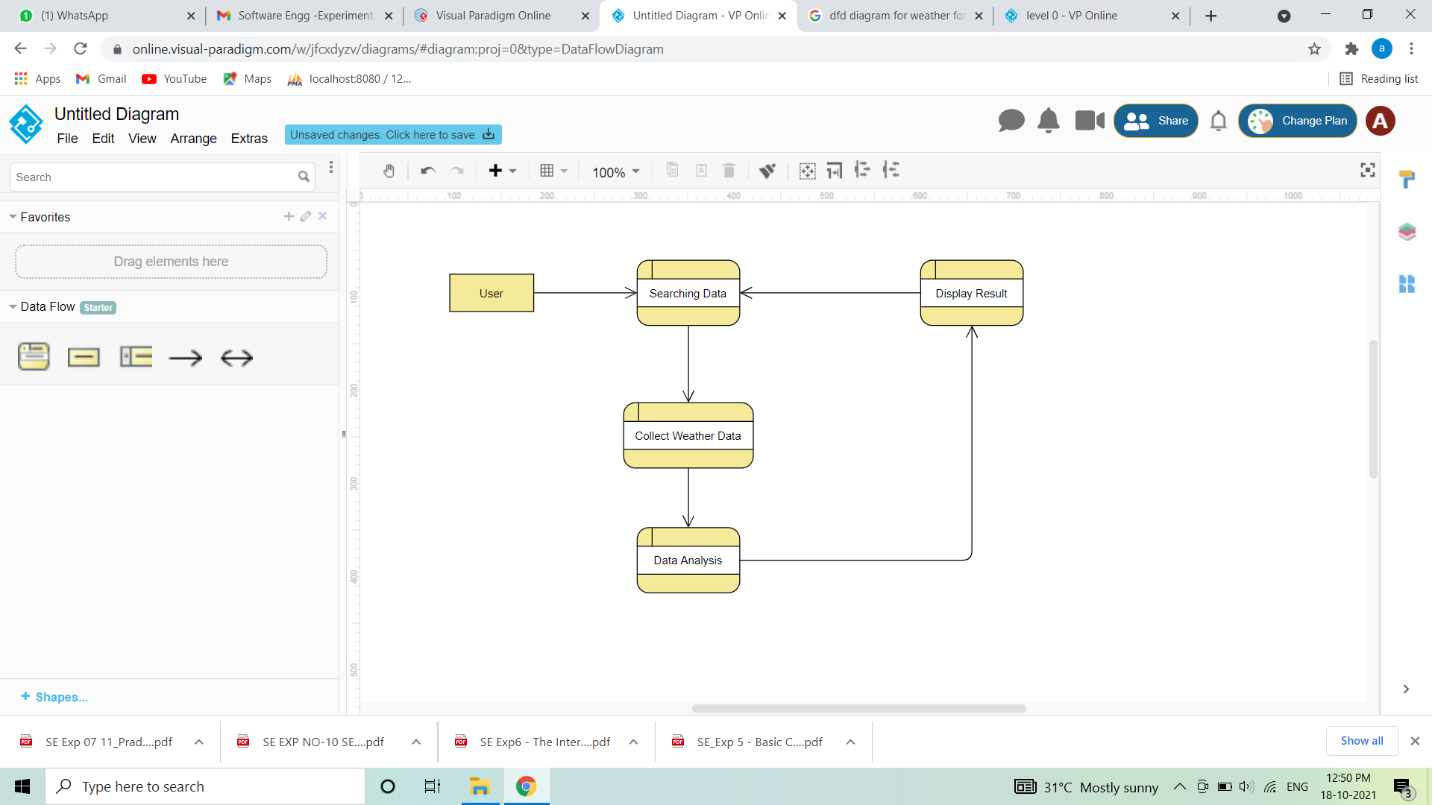
Professionals in various industries, like software engineering, IT, ecommerce, and product management & design, can use DFDs to better understand, refine, or implement a new system or process.

**Level 0 DFD**



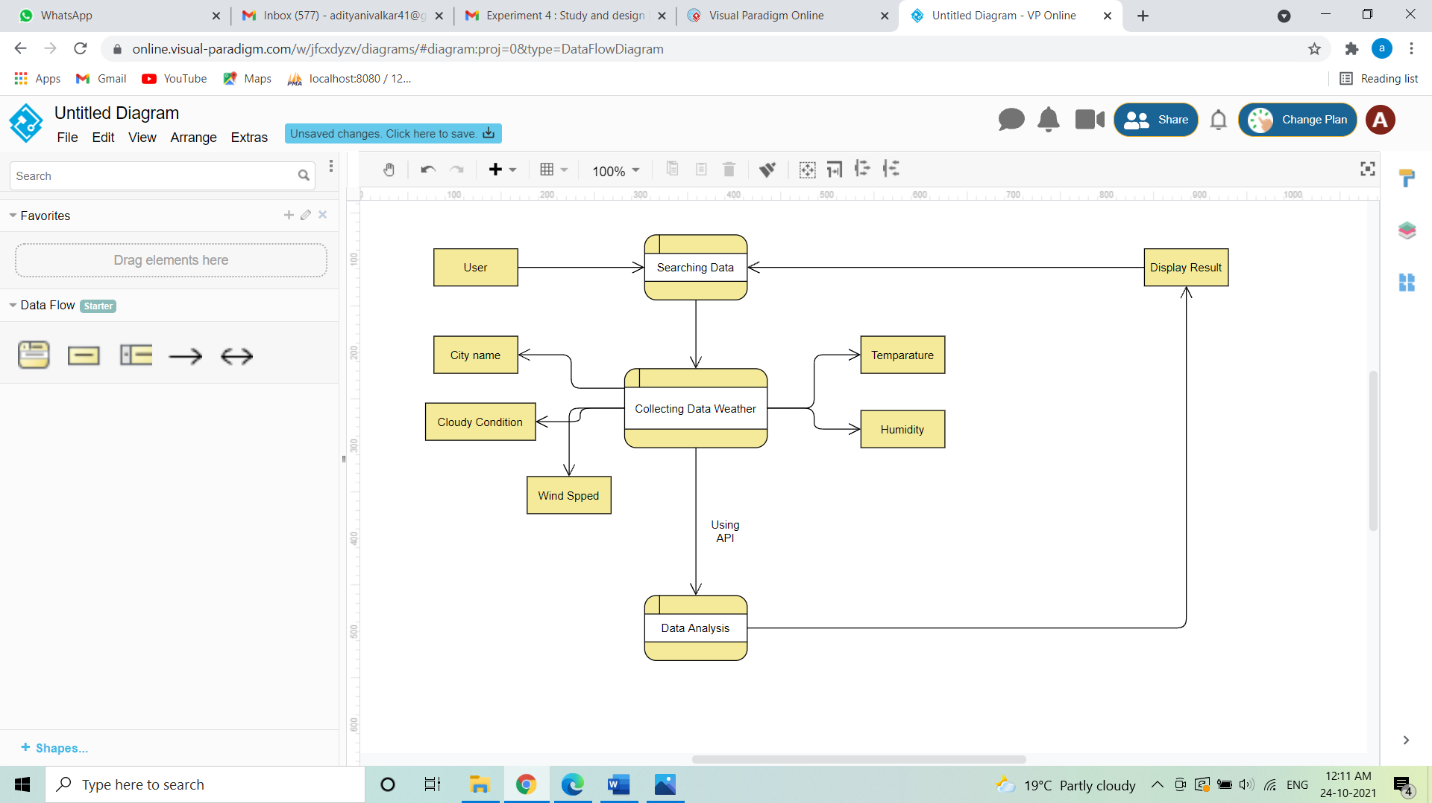
This Level 0 DFD provides a contextual map of a securities trading platform. Data flows in one direction from the User and the admin to Student Management System, and in two directions from Student to the System.

**Level 1 DFD**



This Level 1 DFD breaks down the Student process in more detail, expanding it to include Login, add student details, View or edit Information and register student.

**Level 2 DFD**



This Level 2 DFD decomposes the “Student” process to contextualize the steps required to add details of student.