**System Design**

##### DATA FLOW DIAGRAMS

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts. The Basic Notation used to create a DFD’s are as follows:

**Dataflow:** Data move in a specific direction from an origin to a destination.

**Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**3. Source:** External sources or destination of data, which may be People, programs, organizations or other entities.

**Data Store:** Here data are stored or referenced by a process in the System.

**Context Level Data Flow Diagram**



**AUTHENTICATION DFD:**



**Administrator:**

Level-1 Data Flow Diagram for Admin:



**LEVEL-2:**

**ADMINISTRATOR:**











**LEVEL-3 DIAGRAM:**

**ADMINISTRATOR:**















**LEVEl-4:**

**Administrator:**



Figure 1. Level 4 Data Flow Diagram for Add Employee Details

****

Figure 2. Level 4 Data Flow Diagram for Add New Category Details

**Customer:**



**Leve-1 Data Flow Diagram for Customer:**



**LEVEL-2 Data Flow Diagram for Customer:**











**LEVEL-3 Data Flow Diagram for Customer**

















**LEVEL-4**

**Customer:**



Level 4 Data Flow Diagram for Add New Item Details



Level 4 Data Flow Diagram for Add New Brand Details

Employee Context Level Data Flow Diagram:



**Level-1 Data Flow Diagram for Employee:**



**LEVEL-2 Data Flow Diagram for Employee:**





**Level -3 Data Flow Diagram for Employee:**



**E-R Diagram**

**E-R DIAGRAM**

##### Entity

An entity is an object or concept about which you want to store information.

Entity

##### Weak Entity

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

Weak Entity

##### Key attribute

A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

Key attribute

##### Multivalued attribute

A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

Multivalued attribute

##### Derived attribute

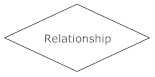
A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

Derived attribute

##### Relationships

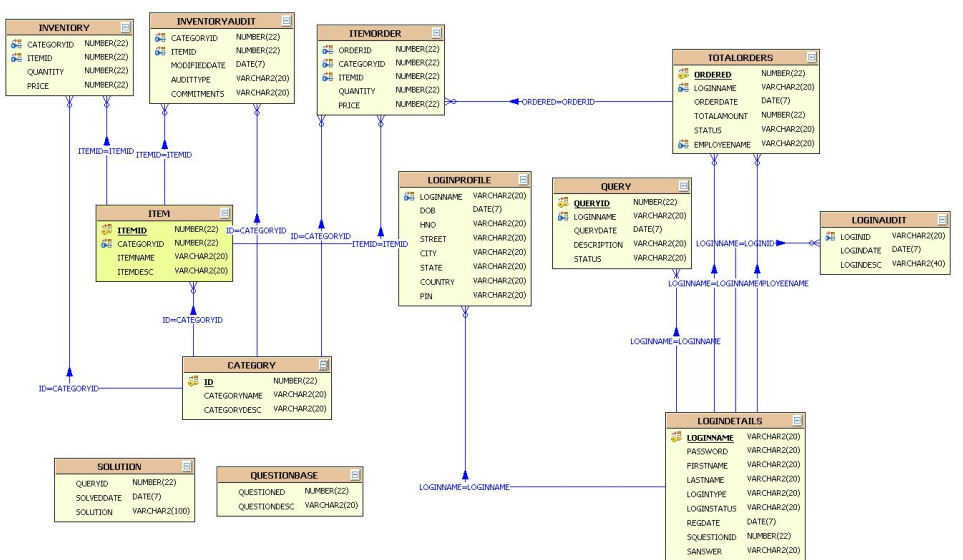
Relationships illustrate how two entities share information in the database structure.

Learn how to draw relationships:  
First, [connect the two entities](http://www.smartdraw.com/resources/tutorials/Drawing-ER-Diagrams-1), then [drop the relationship notation on the line](http://www.smartdraw.com/resources/tutorials/Drawing-ER-Diagrams-2).



ER-Diagram:

 It also called an entity-relationship (ER) diagram, a graphical representation of entities and their relationships to each other, typically used in the organization of [data](http://www.webopedia.com/TERM/D/data.html) within [databases](http://www.webopedia.com/TERM/D/database.html) or information systems.

****

**UML Diagrams**

**(Unified Modeling Language**)

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* **User Model View:**
  1. This view represents the system from the users perspective.
  2. The analysis representation describes a usage scenario from the end-users perspective.
* **Structural model view:**
  1. In this model the data and functionality are arrived from inside the system.
  2. This model view models the static structures.
* **Behavioral Model View:**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View:**

In this the structural and behavioral as parts of the system are represented as they are to be built.

* **Environmental Model View:**

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

* **UML is specifically constructed through two different domains they are:**
  1. UML Analysis modeling, this focuses on the user model and structural model views of the system.
  2. UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.
* **Use case Diagrams** represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.
* **Actors** are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

**Class Collaboration Diagram**

**Class Diagram:**

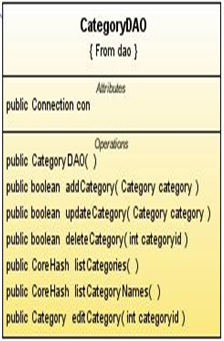
**Abstract Data Access Object:**



**Query:**



**Category:**

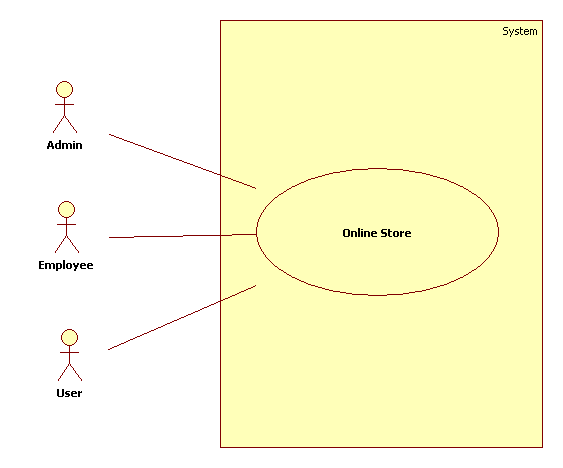


**Brand:**



**Use Case Diagrams**

**System Use Case Diagram:**



**Administrator Use Case Diagram:**

**Admin**

**Products**

**Inventory**

**Employees**

**Queries**

**Orders**

**Item**

**Categery**

**View Categery**

**Add Employe**

**View Employe**

**View Queries**

**View Orders**

**Customer Use Case Diagram:**



**Employee Use Case Diagram**



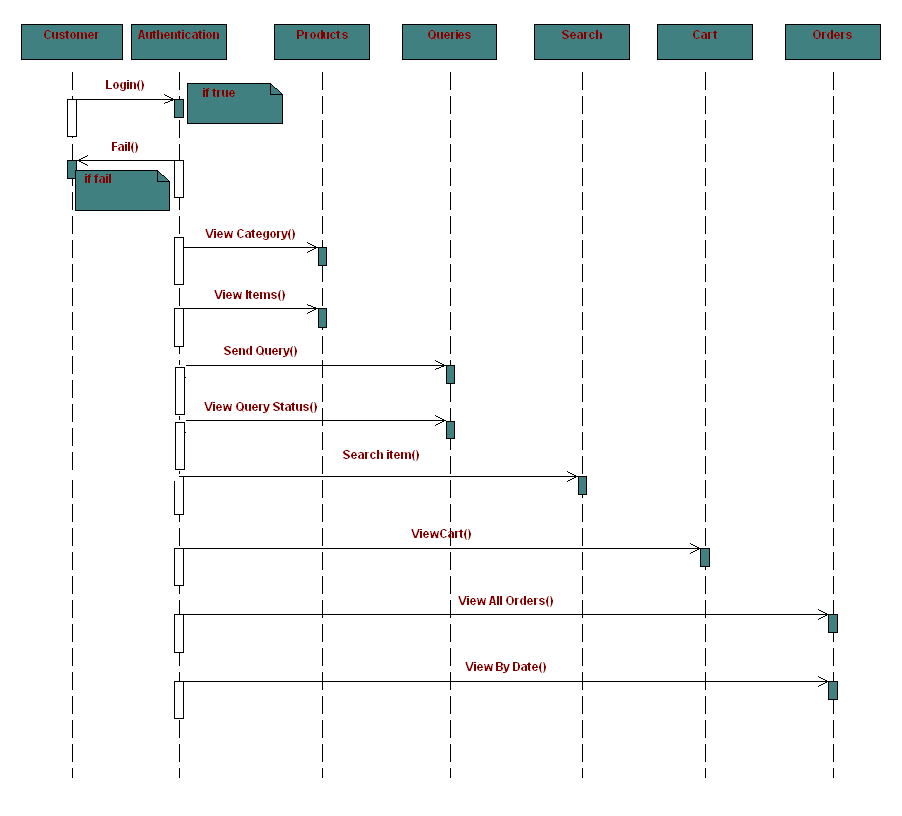
**Sequence Diagrams**

**Administrator Sequence Diagram:** 

**Employee Sequence Diagram:**



**Customer Sequence Diagram:**



**Collaboration Diagrams**

**Admin collaboration Diagram:**



**Employee Collaboration Diagram:**



**Customer Collaboration Diagram:**



**Activity Diagram**

**Administrator Activity Diagram:**



**Customer Activity Diagram:**

Customer

Authentication

Check Authentication

Enter UserName,Password

if Success

if fail

Check

Home

Products

Queries

Search

Cart

Orders

Logout

Item

Category

SendQueries

ViewStatus

ViewCart

ViewOrders

ViewItems

ViewItems

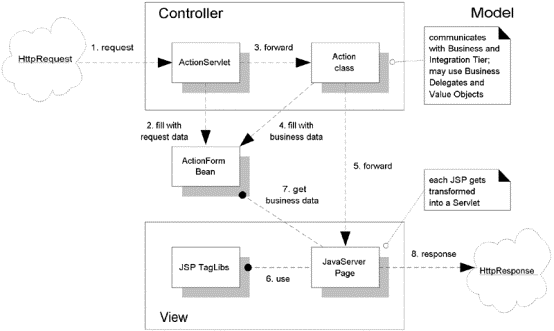
Itemsearch

Vie By Date

**Employee Activity Diagram:**



**Component Diagram**





**Deployment Diagram**

