**SOFTWARE ENGINEERING**

**&**

**PROJECT DEVELOPMENT PROCESS**

**SECTION – 1 LECTURE : 1-2**

**Software Engineering & Development Models**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SECTION – 2 LECTURE : 3-5**

**Software Analysis & Design Tools**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SECTION – 3 LECTURE : 6**

**Object Oriented S/w Analysis & Designing Tool**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SECTION – 4 LECTURE : 7-8**

**DFD & ER Examples**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SECTION – 5 LECTURE 9-10**

**Final Year Project, Internships & Software Engineering Related**

**Interview Questions & Answers**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION - 1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Software Engineering & Development Models**

Software Engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

“A Software Development Project is a complex undertaking by two or more persons within the boundaries of time, budget, and staff resources that produces new or enhanced computer code that adds significant business value to a new or existing business process. “.

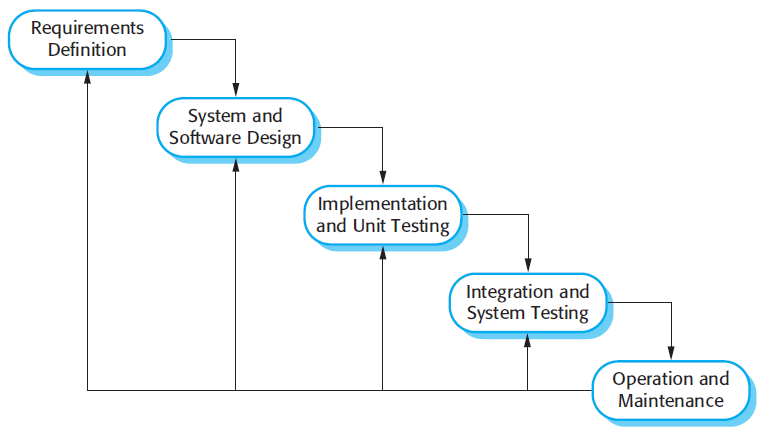
In Software Engineering, a Software Development Process  is the process of dividing Software Development work into distinct phases to improve design, product management, and project management.

Software Developing Life Cycle (SDLC) Models  is a conceptual framework describing all activities in a software development project from planning to maintenance. This process is associated with several models, each including a variety of tasks and activities.

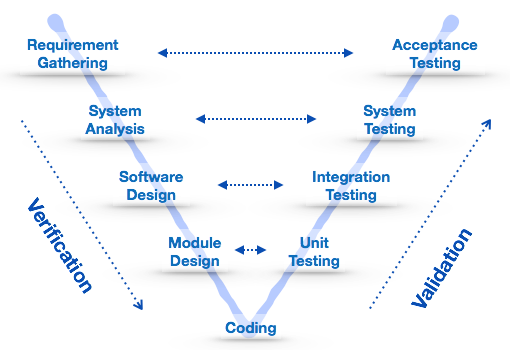
Models are:

* Waterfall Model.
* V-Shaped Model.
* Evolutionary Prototyping Model.
* Spiral Method (SDM)
* Iterative and Incremental Method.
* Agile development.

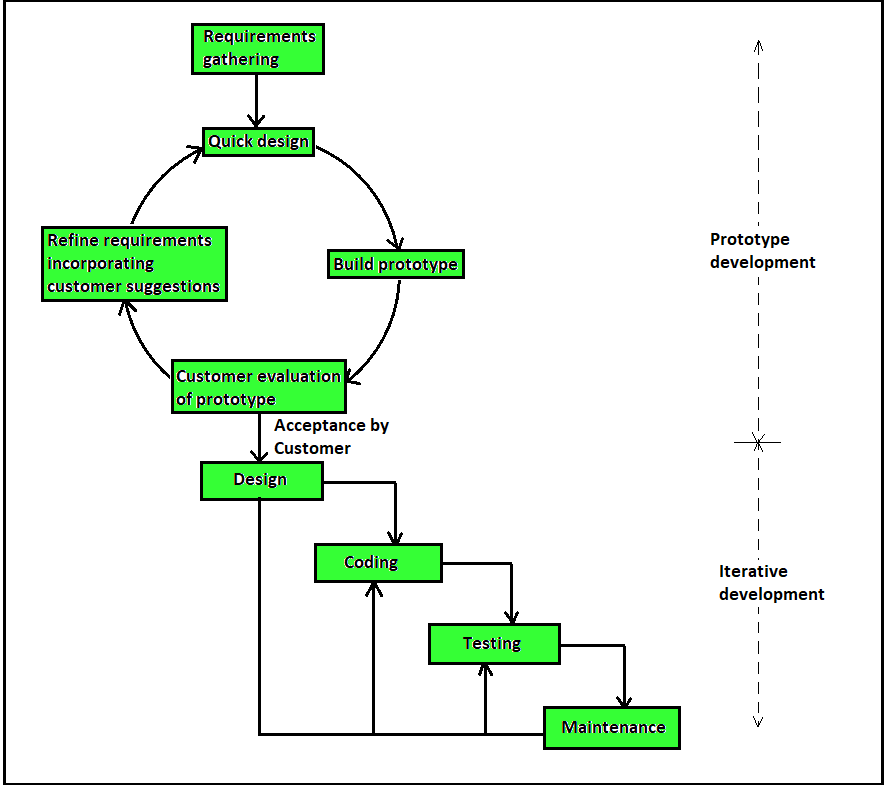
**WATERFALL MODEL**



**V-SHAPED MODEL**



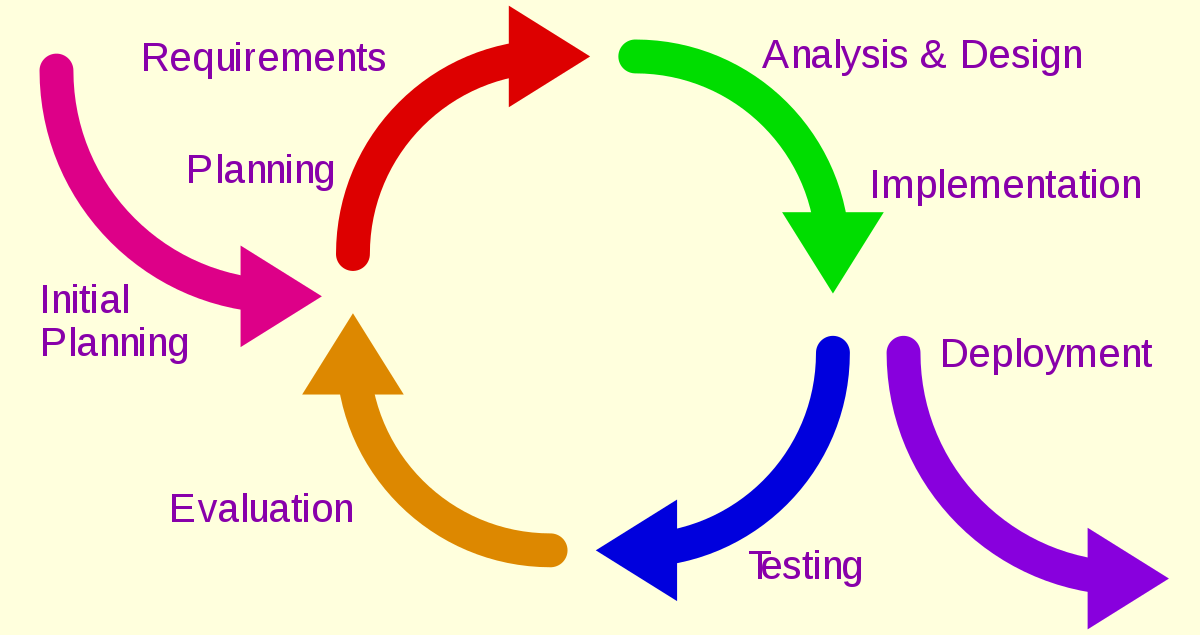
**EVOLUTIONARY PROTOTYPING MODEL**



**SPIRAL METHOD (SDM)**

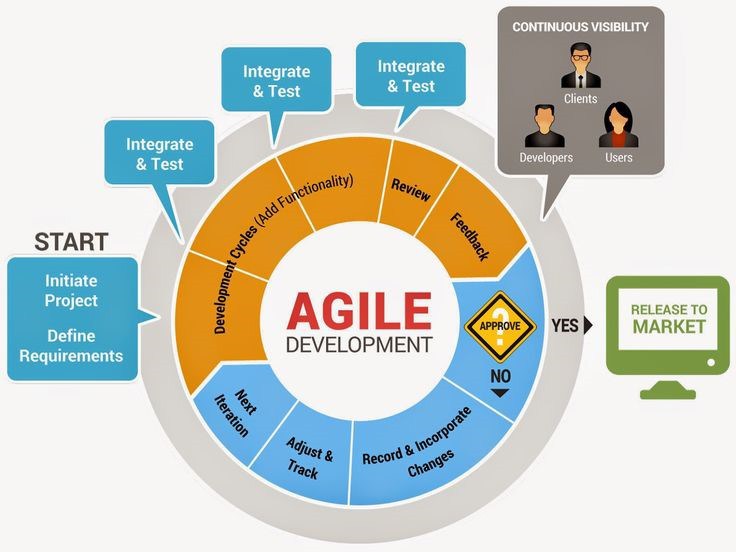


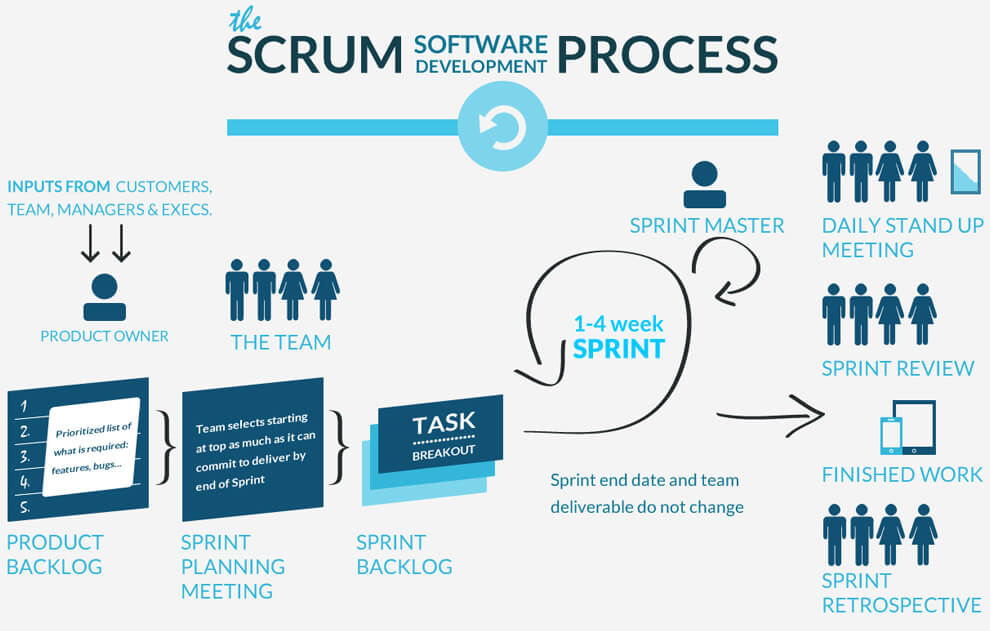
**ITERATIVE AND INCREMENTAL METHOD**



**AGILE & SCRUM DEVELOPMENT**

**Agile** is a development **methodology** based on iterative and incremental **approach**. **Scrum** is one of the implementations of **agile methodology**. In which incremental builds are delivered to the customer in every two to three weeks' time. **Scrum** is ideally used in the project where the requirement is rapidly changing.





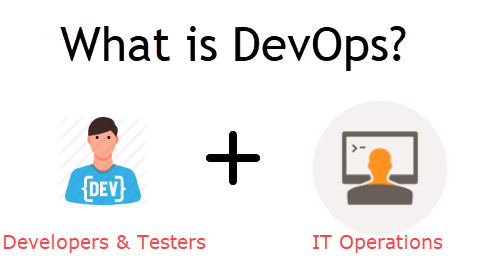
**DevOps**

DevOps is a set of practices that combines software development and information-technology operations which aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

DevOps is a culture which promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way. The word 'DevOps' is a combination of two words 'development' and 'operations.'

DevOps helps to increases an organization's speed to deliver applications and services. It allows organizations to serve their customers better and compete more strongly in the market.

In simple words, DevOps can be defined as an alignment of development and IT operations with better communication and collaboration.



## DevOps Lifecycle

# https://www.guru99.com/images/2-2017/092917_0812_DevOpsTrain2.png

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# KEY POINTS

* DevOps is a culture which promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way
* Before DevOps operation and Development team working in completed isolation.
* Manual code deployment leads to human errors in production
* In the Old process, Operations team has no clue on the progress of the Development team. So, operations team developed a IT infrastructure purchase and monitoring plan as per their understanding.
* In the DevOps process operation team fully aware of the progress of developer. The purchase and monitoring planning is accurate.
* DevOps offers Maintainability, Predictability, Greater quality cost efficiency and time to market.
* Agile process focuses on functional and non-functional readiness while DevOps focuses on that IT infrastructure aspects.
* DevOps life cycle includes Development, Testing, Integration, Deployment, and Monitoring.
* DevOps engineer will work with development team staff to tackle the coding and scripting needs.
* DevOps engineer should have the soft skill of a problem-solver and be a quick-learner
* DevOps Certifications are available from Amazon web services, Red Hat, Microsoft Academy, DevOps Institute.
* DevOps helps organizations in shifting their code deployment cycles to weeks and months instead of years.

# Software Requirement Specification

# A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

# Contents-in-Software-Requirements-Specification-document.jpg

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION -2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Software Analysis & Design Tools**

oftware analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

Let us see few analysis and design tools used by software designers:

## Data Flow Diagram

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. DFD does not contain any control or branch elements.

### Types of DFD

Data Flow Diagrams are either Logical or Physical.

* Logical DFD - This type of DFD concentrates on the system process, and flow of data in the system.For example in a Banking software system, how data is moved between different entities.
* Physical DFD - This type of DFD shows how the data flow is actually implemented in the system. It is more specific and close to the implementation.

### DFD Components

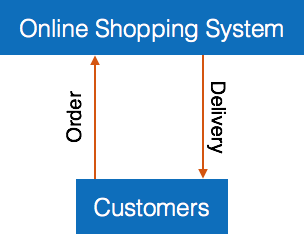
DFD can represent Source, destination, storage and flow of data using the following set of components -



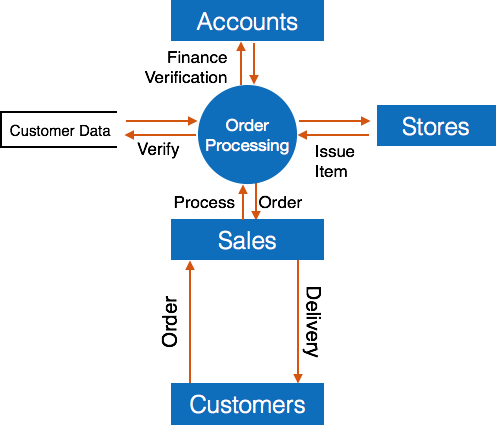
* Entities - Entities are source and destination of information data. Entities are represented by a rectangles with their respective names.
* Process - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
* Data Storage - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
* Data Flow - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

### Levels of DFD

* Level 0 - Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.



* Level 1 - The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.



* Level 2 - At this level, DFD shows how data flows inside the modules mentioned in Level 1.

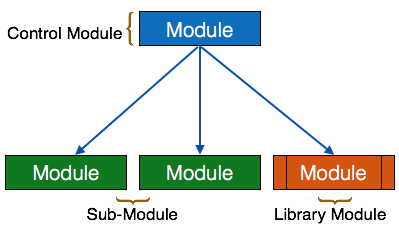
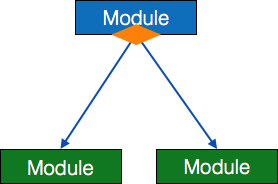
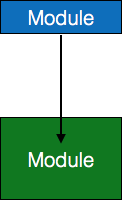
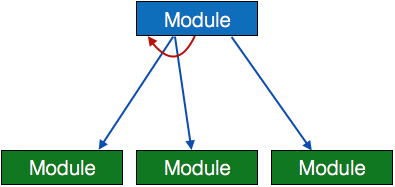
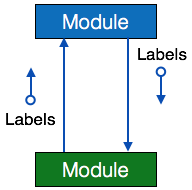
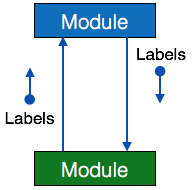
Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.

## Structure Charts

Structure chart is a chart derived from Data Flow Diagram. It represents the system in more detail than DFD. It breaks down the entire system into lowest functional modules, describes functions and sub-functions of each module of the system to a greater detail than DFD.

Structure chart represents hierarchical structure of modules. At each layer a specific task is performed.

Here are the symbols used in construction of structure charts -

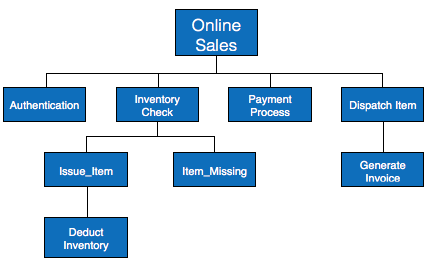
* Module - It represents process or subroutine or task. A control module branches to more than one sub-module. Library Modules are re-usable and invokable from any module.
* Condition - It is represented by small diamond at the base of module. It depicts that control module can select any of sub-routine based on some condition.
* Jump - An arrow is shown pointing inside the module to depict that the control will jump in the middle of the sub-module.
* Loop - A curved arrow represents loop in the module. All sub-modules covered by loop repeat execution of module.
* Data flow - A directed arrow with empty circle at the end represents data flow.
* Control flow - A directed arrow with filled circle at the end represents control flow.

## HIPO Diagram

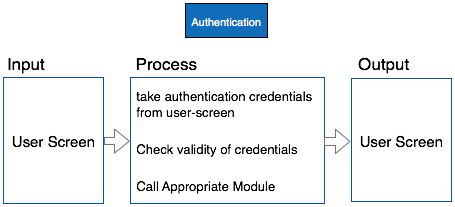
HIPO (Hierarchical Input Process Output) diagram is a combination of two organized method to analyze the system and provide the means of documentation. HIPO model was developed by IBM in year 1970.

HIPO diagram represents the hierarchy of modules in the software system. Analyst uses HIPO diagram in order to obtain high-level view of system functions. It decomposes functions into sub-functions in a hierarchical manner. It depicts the functions performed by system.

HIPO diagrams are good for documentation purpose. Their graphical representation makes it easier for designers and managers to get the pictorial idea of the system structure.



In contrast to IPO (Input Process Output) diagram, which depicts the flow of control and data in a module, HIPO does not provide any information about data flow or control flow.



### Example

Both parts of HIPO diagram, Hierarchical presentation and IPO Chart are used for structure design of software program as well as documentation of the same.

## Pseudo-Code

Pseudo code is written more close to programming language. It may be considered as augmented programming language, full of comments and descriptions.

Pseudo code avoids variable declaration but they are written using some actual programming language’s constructs, like C, Fortran, Pascal etc.

Pseudo code contains more programming details than Structured English. It provides a method to perform the task, as if a computer is executing the code.

### Example

Program to print Fibonacci up to n numbers.

void function Fibonacci

Get value of n;

Set value of a to 1;

Set value of b to 1;

Initialize I to 0

for (i=0; i< n; i++)

{

if a greater than b

{

Increase b by a;

Print b;

}

else if b greater than a

{

increase a by b;

print a;

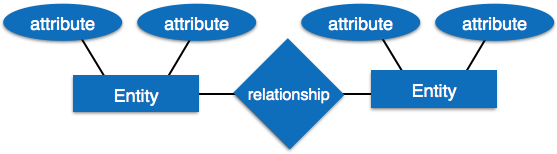
}

}

## Entity-Relationship Model

Entity-Relationship model is a type of database model based on the notion of real world entities and relationship among them. We can map real world scenario onto ER database model. ER Model creates a set of entities with their attributes, a set of constraints and relation among them.

ER Model is best used for the conceptual design of database. ER Model can be represented as follows :



* Entity - An entity in ER Model is a real world being, which has some properties called *attributes*. Every attribute is defined by its corresponding set of values, called *domain*.

For example, Consider a school database. Here, a student is an entity. Student has various attributes like name, id, age and class etc.

* Relationship - The logical association among entities is called *relationship*. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of associations between two entities.

Mapping cardinalities:

* + one to one
  + one to many
  + many to one
  + many to many

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION -3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Object Oriented S/W analysis & Designing Tool**

**UML: Unified Modeling Language**

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.

Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

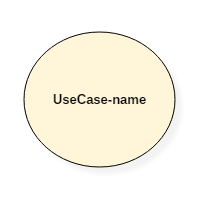
These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

## Use-case diagram notations

Following are the common notations used in a use case diagram:

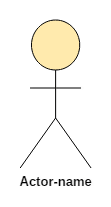
**Use-case:**

Use cases are used to represent high-level functionalities and how the user will handle the system. A use case represents a distinct functionality of a system, a component, a package, or a class. It is denoted by an oval shape with the name of a use case written inside the oval shape. The notation of a use case in UML is given below:

**UML UseCase Notation** 

**Actor:**

It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is the best example of an actor. An actor is an entity that initiates the use case from outside the scope of a use case. It can be any element that can trigger an interaction with the use case. One actor can be associated with multiple use cases in the system. The actor notation in UML is given below.

**UML Actor Notation** 

## How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

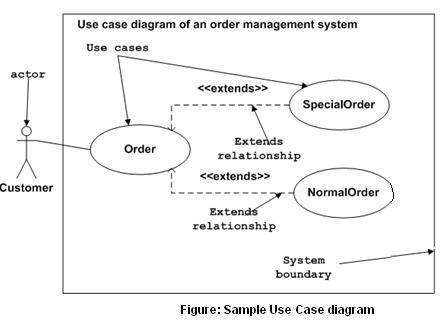
* Functionalities to be represented as use case
* Actors
* Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

* The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.
* Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
* Use notes whenever required to clarify some important points.

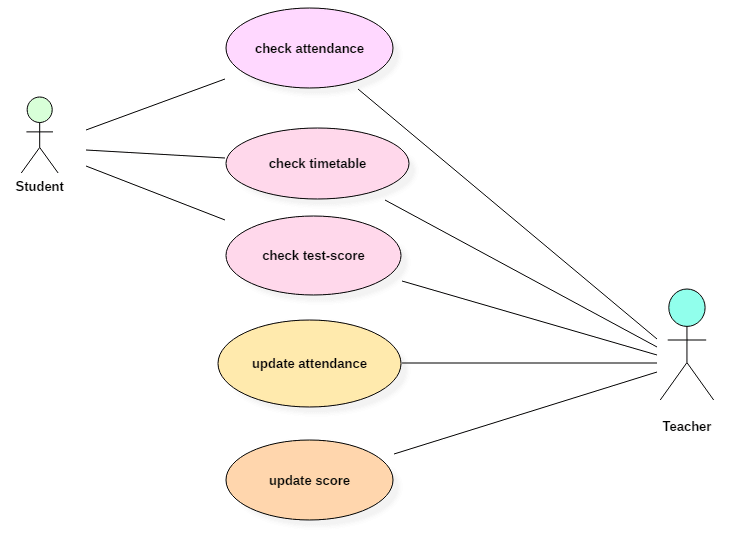
Following is a sample use case diagram representing the order management system. Hence, if we look into the diagram then we will find three use cases **(Order, SpecialOrder, and NormalOrder)** and one actor which is the customer.

The SpecialOrder and NormalOrder use cases are extended from *Order* use case. Hence, they have extended relationship. Another important point is to identify the system boundary, which is shown in the picture. The actor Customer lies outside the system as it is an external user of the system.



## An example of a use-case diagram

Following use case diagram represents the working of the student management system:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION -4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**DFD & ER EXAMPLES**

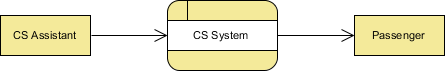
**1. Data Flow Diagram with Examples - Customer Service System**

The data flow diagram is a hierarchy of diagram consist of:

1. Context Diagram (conceptually level zero)
2. The Level-1 DFD
3. And possible Level-2 DFD and further levels of functional decomposition depending on the complexity of your system

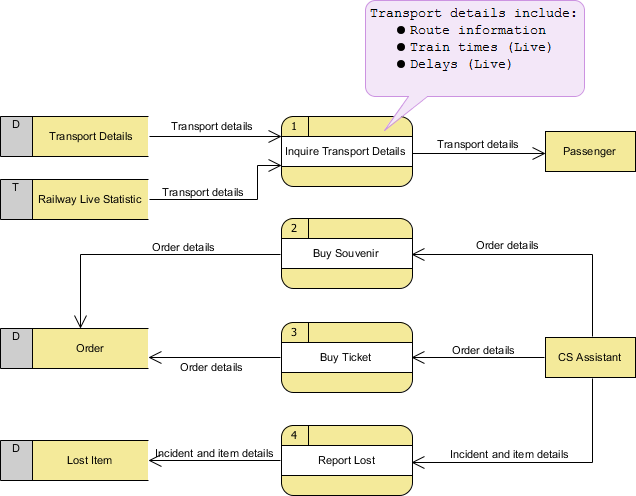
### Context DFD

The figure below shows a context Data Flow Diagram that is drawn for a railway company's Customer Service System. It contains a process (shape) that represents the system to model, in this case, the "*CS System*". It also shows the participants who will interact with the system, called the external entities. In this example, *CS Assistant* and *Passenger* are the two 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. **Also mark necessary data flows between entities and process.**



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

### Level 1 DFD

The figure below shows the level 1 DFD, which is the decomposition (i.e. break down) of the CS 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 CS System Data Flow Diagram example contains four processes, two external entities and four data stores. Although there is no design guidelines that governs the positioning of shapes in a Data Flow Diagram, we tend to put the processes in the middle and data stores and external entities on the sides to make it easier to comprehend.

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*. While data stored in *Transport Details* are persistent data (indicated by the label "D"), data stored in *Railway Live Statistic* are transient data that are held for a short time (indicated by the label "T"). A callout shape is used to list out the kind of details that can be inquired by passenger.

*CS Assistant* can initiate the *Buy Souvenir* process, which will result in having the *Order details* stored in the *Order* data store. Although customer is the real person who buy souvenir, it is the *CS Assistant* who accesses the system for storing the order details. Therefore, we make the data flow from *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. Data Flow Diagram is a high level diagram that is drawn with a high degree of abstraction. The data store Order which is drawn here does not necessarily imply a real order database or order table in a database. The way how order details are stored physically is to be decided later on when implementing the system.

Finally, *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.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

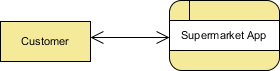
**Supermarket App Example**

The data flow diagram is a hierarchy of diagram consist of:

1. Context Diagram (conceptually level zero)
2. The Level-1 DFD
3. And possible Level-2 DFD and further levels of functional decomposition depending on the complexity of your system.

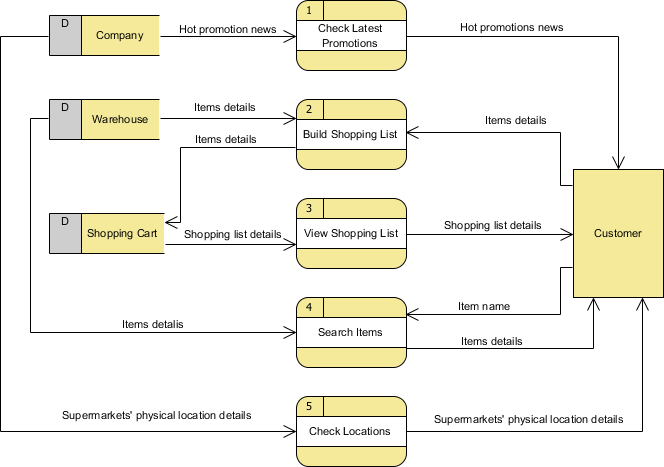
### Context DFD

The figure below shows a context Data Flow Diagram that is drawn for an Android supermarket app. It contains a process (shape) that represents the system to model, in this case, the "*Supermarket App*". It also shows the participants who will interact with the system, called the external entities. In this example, there is only one external entity, which is the *Customer*. In between the process and the external entity, there is a bi-directional connector, which indicates the existence of information exchange between customer and the app, and the information flow is bi-directional.



Context DFD is the entrance of a data flow model. It contains one and only one process and does not show any data store, which makes the diagram simple. **Also mark necessary data flows between entities and process.**

### Level 1 DFD

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

The Supermarket App Data Flow Diagram example contains five processes, one external entity and three data stores. Although there is no design guideline that governs the positioning of shapes in a Data Flow Diagram, we tend to put the processes in the middle and data stores and external entities on the sides to make it easier to comprehend.

Based on the diagram, we know that a *Customer* can receive *Hot promotion news* from the *Check Latest Promotions* process and the news is provided by the *Company* database. Note that by common sense we know that *Check Latest Promotions* is likely to be a feature of the app but the Data Flow Diagram itself implies no such thing. Theoretically speaking, a process in Data Flow Diagram may correspond to a feature or a set of features.

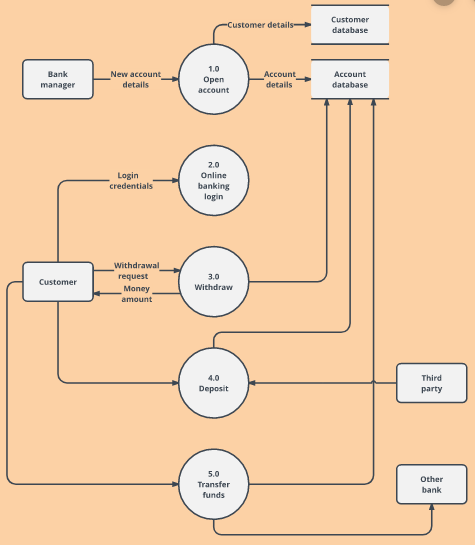
A *Customer* can *Build Shopping List* by providing *Items details* and the details will be stored in the *Shopping Cart* database. The *Warehouse* database will also provide the *Items details* required to complete the process.

A *Customer* can receive *Shopping list details* from the *View Shopping List* process and such details is provided by the *Shopping Cart* database.

A *Customer* can receive *Items details* by performing the *Search Items* process. He/she must provide an *Item name* for searching and the *item details* are returned from the *Warehouse* database after searched. Although we said that the search result is returned after searching, the Data Flow Diagram, again, implies no such thing. It is our common sense that leads us to interpret the diagram in the way that we understand it naturally. Keep in mind that Data Flow Diagram only tells you where information exchange takes place. It does not answer in what way and in what order the information is being used throughout a system. If this information is important and worth mentioning, consider to model it with diagrams like [BPMN Business Process Diagram](https://www.visual-paradigm.com/features/bpmn-diagram-and-tools/#business-process-diagram) or [UML Activity Diagram](https://www.visual-paradigm.com/features/uml-tool/#activity-diagram).

Finally, a *Customer* can receive *Supermarkets' physical location details* by performing *Check Locations* and the details is provided by the *Company* database.

**Example 3: DFD for Banking Transaction System**



**CASE STUDIES**

**Draw the DFD of following scenarios:**

**SCENARIO 1 :**

Video-Rental LTD case study Video-Rental LTD is a small video rental store. The store lends videos to customers for a fee, and purchases its videos from a local supplier. A customer wishing to borrow a video provides the empty box of the video they desire, their membership card, and payment – payment is always with the credit card used to open the customer account. The customer then returns the video to the store after watching it. If a loaned video is overdue by a day the customer's credit card is charged, and a reminder letter is sent to them. Each day after that a further card is made, and each week a reminder letter is sent. This continues until either the customer returns the video, or the charges are equal to the cost of replacing the video.

New customers fill out a form with their personal details and credit card details, and the counter staff give the new customer a membership card. Each new customer's form is added to the customer file.

The local video supplier sends a list of available titles to Video-Rental LTD, who decide whether to send them an order and payment. If an order is sent then the supplier sends the requested videos to the store. For each new video a new stock form is completed and placed in the stock file.

**SCENARIO 2 :**

Estate Agency case study Clients wishing to put their property on the market visit the estate agent, who will take details of their house, flat or bungalow and enter them on a card which is filed according to the area, price range and type of property. Potential buyers complete a similar type of card which is filed by buyer name in an A4 binder.

Weekly, the estate agent matches the potential buyer's requirements with the available properties and sends them the details of selected properties. When a sale is completed, the buyer confirms that the contracts have been exchanged, client details are removed from the property file, and an invoice is sent to the client.

The client receives the top copy of a three part set, with the other two copies being filed. On receipt of the payment the invoice copies are stamped and archived. Invoices are checked on a monthly basis and for those accounts not settled within two months a reminder (the third copy of the invoice) is sent to the client.

# ER Diagram

**EXAMPLE 1 :**

*address*

*model*

*driver-id*

*name*

*license*

*year*

*owns*

*location*

*report-number*

*date*

*driver*

*participated*

*damage-amount*

*accident*

*car*

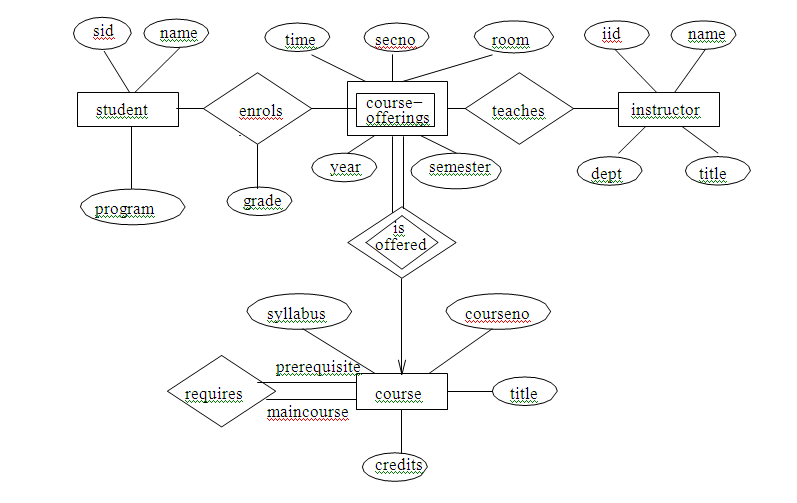
*person*

**E-R diagram for a Car-insurance company**

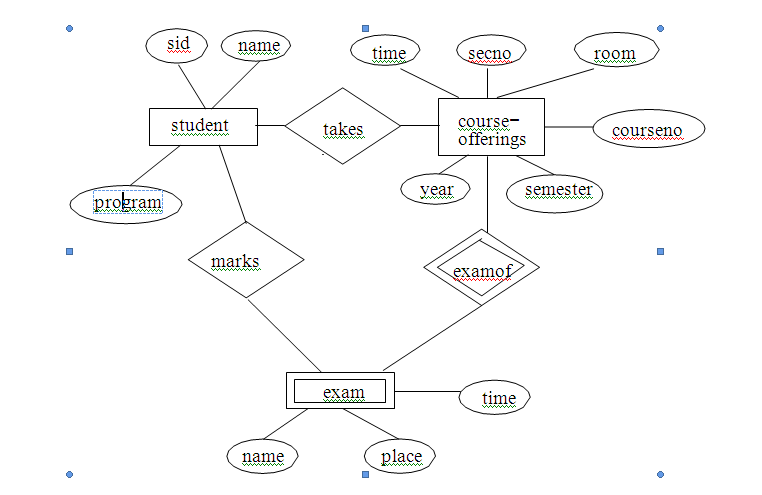
# EXAMPLE 2 :

* 1. A university registrar’s office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and ti- tle. Further, the enrollment of students in courses and grades awarded to stu- dents in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar’s office. Document all assumptions that you make about the mapping constraints.

****

**E-R diagram for a university**

****

**E-R diagram for marks database**

**PROBLEM 1 :**

Assume we have the following application that models soccer teams, the games they play, and the players in each team. In the design, we want to capture the following:

* We have a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs.
* Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses.
* Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
* For each match we need to keep track of the following:
  + The date on which the game is played
  + The final result of the match
  + The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card.
  + During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
* Each match has exactly three referees. For each referee we have an ID (unique identifier), name, DoB, years of experience. One referee is the main referee and the other two are assistant referee.

## Design an ER diagram to capture the above requirements. State any assumptions you have that affects your design (use the back of the page if needed). Make sure cardinalities and primary keys are clear.

**PROBLEM 2 :**

Read the following case study, which describes the data requirements for a video rental company. The

video rental company has several branches throughout the USA. The data held on each branch is the

branch address made up of street, city, state, and zip code, and the telephone number. Each branch is

given a branch number, which is unique throughout the company. Each branch is allocated sta;, which

includes a Manager. The Manager is responsible for the day-today running of a given branch. The data

held on a member of sta; is his or her name, posion, and salary. Each member of sta; is given a sta;

number, which is unique throughout the company. Each branch has a stock of videos. The data held on a

video is the catalog number, video number, tle, category, daily rental, cost, status, and the names of

the main actors, and the director. The catalog number uniquely iden<es each video. However, in most

cases, there are several copies of each video at a branch, and the individual copies are iden<ed using

the video number. A video is given a category such as Acon, Adult, Children, Drama, Horror, or Sci-Fi.

The status indicates whether a speci<c copy of a video is available for rent. Before hiring a video from

the company, a customer must <rst register as a member of a local branch. The data held on a member

is the <rst and last name, address, and the date that the member registered at a branch. Each member is

given a member number, which is unique throughout all branches of the company. Once registered, a

member is free to rent videos, up to maximum of ten at any one me. The data held on each video

rented is the rental number, the full name and number of the member, the video number, tle, and

daily rental, and the date the video is rented out and date returned. The rental number is unique

throughout the company.

a) Idenfy the main enty types of the video rental company.

b) Idenfy the main relaonship types between the enty types described in (a) and represent

each relaonship as an ER diagram.

c) Determine the mulplicity constraints for each relaonship described in (b). Represent the

mulplicity for each relaonship in the ER diagrams created in (b).

d) Idenfy a1ributes and associate them with enty or relaonship types. Represent each a1ribute

in the ER diagrams created in (c).

e) Determine candidate and primary key a1ributes for each (strong) enty type.

f) Using your answers (a) to (e) a1empt to represent the data requirements of the video rental

company as a single ER diagram. State any assumpons necessary to support your design

Read the following case study, which describes the data requirements for a video rental company. The

video rental company has several branches throughout the USA. The data held on each branch is the

branch address made up of street, city, state, and zip code, and the telephone number. Each branch is

given a branch number, which is unique throughout the company. Each branch is allocated sta;, which

includes a Manager. The Manager is responsible for the day-today running of a given branch. The data

held on a member of sta; is his or her name, posion, and salary. Each member of sta; is given a sta;

number, which is unique throughout the company. Each branch has a stock of videos. The data held on a

video is the catalog number, video number, tle, category, daily rental, cost, status, and the names of

the main actors, and the director. The catalog number uniquely iden<es each video. However, in most

cases, there are several copies of each video at a branch, and the individual copies are iden<ed using

the video number. A video is given a category such as Acon, Adult, Children, Drama, Horror, or Sci-Fi.

The status indicates whether a speci<c copy of a video is available for rent. Before hiring a video from

the company, a customer must <rst register as a member of a local branch. The data held on a member

is the <rst and last name, address, and the date that the member registered at a branch. Each member is

given a member number, which is unique throughout all branches of the company. Once registered, a

member is free to rent videos, up to maximum of ten at any one me. The data held on each video

rented is the rental number, the full name and number of the member, the video number, tle, and

daily rental, and the date the video is rented out and date returned. The rental number is unique

throughout the company.

a) Idenfy the main enty types of the video rental company.

b) Idenfy the main relaonship types between the enty types described in (a) and represent

each relaonship as an ER diagram.

c) Determine the mulplicity constraints for each relaonship described in (b). Represent the

mulplicity for each relaonship in the ER diagrams created in (b).

d) Idenfy a1ributes and associate them with enty or relaonship types. Represent each a1ribute

in the ER diagrams created in (c).

e) Determine candidate and primary key a1ributes for each (strong) enty type.

f) Using your answers (a) to (e) a1empt to represent the data requirements of the video rental

company as a single ER diagram. State any assumpons necessary to support your design

**QUESTION 1: THE DATA DESIGN PROCESS .**

Imagine that you have been assigned to a team that will be developing an inventory tracking system. As part of the project startup, your manager has asked each team leader to bring a basic work plan to the next meeting. At that meeting, these work plans will be analyzed to determine the overall project timeframe, costs, personnel requirements and software requirements. For now, as the team leader for the data design team, you have been asked to bring a work plan that identifies the phases of data design and includes the following information for each phase: a). a description of the data design phase, b). the inputs of the phase, c). the outputs of the phase, d). a key issue addressed in the phase e). a challenge that you can anticipate would occur in the phase. Please prepare the response you will bring to the meeting.

**QUESTION 2: CREATING AN ENTITY-RELATIONSHP DIAGRAM**

UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute. Please create an Entity Relationship diagram that captures this information about the UPS system. Be certain to indicate identifiers and cardinality constraints.

**QUESTION 3: CREATING A RELATIONAL DATABASE SCHEMA**

Production tracking is important in many manufacturing environments (e.g., the pharmaceuticals industry, children’s toys, etc.). The following ER diagram captures important information in the tracking of production. Specifically, the ER diagram captures relationships between production lots (or batches), individual production units, and raw materials. Prepare & convert the ER diagram into a relational database schema. Be certain to indicate primary keys and referential integrity constraints.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SECTION - 5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**FINAL YEAR PROJECT, INTERNSHIPS & SOFTWARE ENGINEERING RELATED**

**INTERVIEW QUESTIONS & ANSWERS**

# 1. Explain in brief about your project. [INFOSYS, TCS,WIPRO, CONGNIZANT, LIDO, WEBKUL]

## Answer will be in following sequence:

## Title of the project

## Team members

## Your role

## About the project – what is project all about, including objective & scope

## Module Descriptions

## Frontend & Backend (H/W & S/W requirements)

## Current status of the project

# 2.What was your role in the project? [INFOSYS, TCS]

# Mention your role in terms of S/W engineering:

## - I am a group leader.

## I am involve in Analysis of a project. It includes existing

## SYSTEM/ALGORITHM study and requirements collection. Also define

## “Modules” in a project.

## - I am involve in designing. Designing includes “DATABASE DESIGN &

## INTERFACE DESIGN”

## - CODING/TESTING should be mentioned by every member in a group.

## 3.What is a Software Project Scope? [INFOSYS, TCS,WIPRO]

## A scope is utilized to outline the activities performed to design, develop and deliver a software product. In other words, scope contains information on what project is intended to deliver and what it does not intend to. The scope also outlines information on what software product developed contains and what it does not contain.

## 4.What are Software Metrics? [INFOSYS]

## Metrics are utilized to guide the software product delivery as per the business standards. Metrics can also be used to measure few features of software product delivery. Metrics are divided into requirement metrics, product metrics, performance metrics, and process metrics.

## 5.Mention a few software analysis & Design tools? [INFOSYS, TCS,WIPRO]

## Some of the key software analysis & design tools are Data flow Diagrams (DFD), Structured Charts, Data Dictionary, UML (Unified Modeling Languages) diagrams, ER (Entity Relationship) Diagrams etc.

## 6.What is black box testing and white box testing? [INFOSYS, TCS,WIPRO, WEBKUL]

## Black Box Testing: is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester.

## White Box Testing: is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

## 7.What do you mean by Change Control?

## Change control tracks the changes made in a software to ensure consistency and updates are incorporated as per the enterprise standards.

## 8.Where did you get the idea from? [INFOSYS, TCS, MPHESIS]

## Answer may:

## Based on your area of interest(may be algorithm implementation of any core subject)

## Find some real life manual system which needs to be automated

## Live project

## Internet

# 9.Explain Functional and Non Functional Requirements of your project. [TCS]

# Functional requirements of a s/w :

## Functional requirements specifies a function that a system or system component must be able to perform.

## Functional requirements is what a system is supposed to accomplish. It may be

### Calculations

### Technical details

### Data manipulation

### Data processing

### Other specific functionality

# Non functional requirements of a s/w :Non-functional requirements are any other requirement than functional requirements.These are the requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

## Non-functional requirements - can be divided into two main categories:

### Execution qualities, such as security and usability, which are observable at run time.

### Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

## Non Functional Requirements are:

## Performance requirements

## Interface requirements

## Operational requirements

## Resource requirements

## Verification requirements

## Acceptance requirements

## Documentation requirements

## Security requirements

## Portability requirements

## Quality requirements

## Reliability requirements

## Maintainability requirements

## Safety requirements

# 10.Did you face any challenges and how did you settle them?[ [INFOSYS, TCS,WIPRO]

## Answer will vary form project to project. Common headings may :

## Challenges related with work distribution

## Challenges related with team members

## Challenges related with time line

## Challenges related with disputes between members

## - Challenges related with new technology & platforms

# 11.Did you lead the project? How good team player you would consider yourself?[INFOSYS,WIPRO]

## -As a leader matter will be same as previous slide.

## As a team player, you can mention that :

### Always tried to complete my module on allotted time.

### If is there any clash , it should be settled with mutual discussion or in consultation with team leader.

### If I too feel some problem I put it in front of every one and resolve.

## 12.Suppose budget decided for the project is exceeded and 30% of work is pending. What will be your approach to complete the project and how you will manage the cost? [[INFOSYS]

## First I would try to find out the reasons and activities which are consuming more cost.

## Try to minimize the resources(manpower etc.) and increase the time and effort to cover the remaining tasks with in time.

## 13.What is your group size? How did you divide the tasks between team members? [INFOSYS, TCS,WIPRO]

## -Mentioned your group size.

## We have conducted the meeting after the Synopsis approval.

## We shared our area of interest and expertise.

## Leader (if you are leader use I)then divided the initial tasks.

## All have decided to learn Technology & Platform on which we are going to develop our project.

## 

## 14.What strategy did you follow to integrate daily developments of your project?[INFOSYS] - We have decided to share our daily progress through mail or

## Whatsapp.

## If possible we can meet and share.

## Our Guide have given a fix day in a week to meet and share progress. Area of improvements & suggestions also shared by him.

## 15.What do you think about the future scope of the project? [[INFOSYS, TCS,WIPRO, WEBKUL]

## Answer will depend on type of project. Explain in terms of your project’s future & scope mentioned in Synopsis.

## 16.Share some challenging tasks you faced while executing your project and what was your approach in solving them? [INFOSYS, TCS]

## Answer will be based on project to project.

## Common answer may be:

## When we integrate modules we face so many errors. We examine each and every module in detail.

## We face few problems related with Machine & OS like 32 bit m/c and 64 bit s/w and vice-versa.

## 17.Can your project be commercialized? [INFOSYS, TCS,WIPRO]

## -If you are developing Live project or a kind of project which is

## implementable then say YES otherwise not.

## If you will say “YES” then next question will be, how you will calculate

## 18. How we can calculate the cost of project? [ TCS]

## - Cost estimation tools are:

#### a. Cocomo Model

#### b. Effort Estimation (LOC,KLOC)

#### LOC(Line of Code), KLOC(Thousand Line of Code)

## 19.Given a chance now, do you think you can perform well in the project? If yes, then please explain how. [INFOSYS, TCS,WIPRO]

## Yes I can perform in project development process because:

## I am good team player.

## I have enough knowledge of Software Engineering Methodologies.

## I always try to update myself with latest technologies & platforms.

## 19.Mention three most important references that helped you in development of the your project. [INFOSYS]

## Mention website related with your project.

## Mention website related with S/W & OS used in your project.

## Mention any organization from where you have collected data(specially in case of Live project).

## Mention name of Books(if you are going to implement some specific algorithm)

## 20.Tell me something about your summer Internships and Trainings, If any.

**[INFOSYS, TCS,WIPRO,LIDO,WEBKUL]**

## Explain your Internships & Training Programs:

## What S/W, Language or Technology you have learnt.

## Duration

## From where

## Outcome

## 21.What are your learnings from the Internships & Trainings programs? How you will use them and where? [INFOSYS, TCS,WIPRO,LIDO, WEBKUL]

## State your learnings and mention that you will implement it in final year project.

## You can also mentioned that its for up gradation and will be useful for Job purposes.

**22.** **What is Validation and Verification? [INFOSYS, TCS,WIPRO,AMAZON]**

**Validation**

**Validation** is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e. it checks what we are developing is the right product. it is validation of actual and expected product. Validation is the dynamic testing.

Validation means **Are we building the right product?**

**Verification:**

## Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we have. Verification is static testing.

## Verification means ****Are we building the product right?****

## 22.Do you know latest Software Development tools used in S/W Industry?]INFOSYS]

## A comprehensive list of software development tools and integrated development environments (IDE), both hosted and downloadable tools, used for software programming .

## Atom. ...

## Cloud9 IDE. ...

## CodeCharge Studio. ...

## Code Envy. ...

## CodeLobster (Win) (Free) ...

## CodePen. ...

## Comparison of IDEs. ...

## Crimson Editor (Win) (Free)

## LINX