

Tasks:

1. SDLC
2. DFD
3. ER diagram
4. Use Case
5. Documentation of Project

SDLC

What is Software Development Life Cycle (SDLC)?

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.

Purpose:

Purpose of SDLC is to deliver a high-quality product which is as per the customer's requirement.

For Example, A software has to be developed and a team is divided to work on a feature of the product and is allowed to work as they want. One of the developers decides to design first whereas the other decides to code first and the other on the documentation part.

This will lead to project failure because of which it is necessary to have a good knowledge and understanding among the team members to deliver an expected product.

SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

1) Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

For Example, A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

2) Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

3) Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

4) Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

5) Deployment

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

6) Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

REFERENCE:

1. <https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>
2. <https://stackify.com/what-is-sdlc/>

DED

Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.


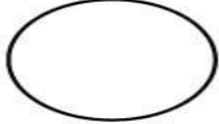

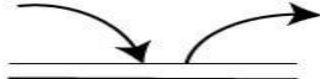
The following observations about DFDs are essential:

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision

points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.

4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:

Symbol	Name	Function
	Data flow	Used to Connect Processes to each , other , to sources or Sinks; te arrow head indicates direction of data flow.
	Process	Performs Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

Symbols for Data Flow Diagrams

REFERENCE:

1. <https://www.javatpoint.com/software-engineering-data-flow-diagrams>
2. https://en.wikipedia.org/wiki/Data-flow_diagram

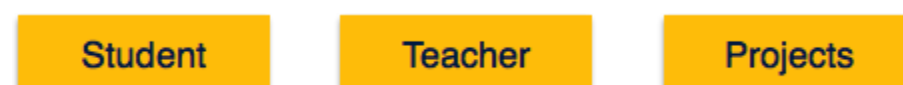
ER Diagram

What is an Entity Relationship Diagram (ERD)?

ERD stands for entity relationship diagram. People also call these types of diagrams ER diagrams and Entity Relationship Models. An ERD visualizes the relationships between entities like people, things, or concepts in a database. An ERD will also often visualize the attributes of these entities.

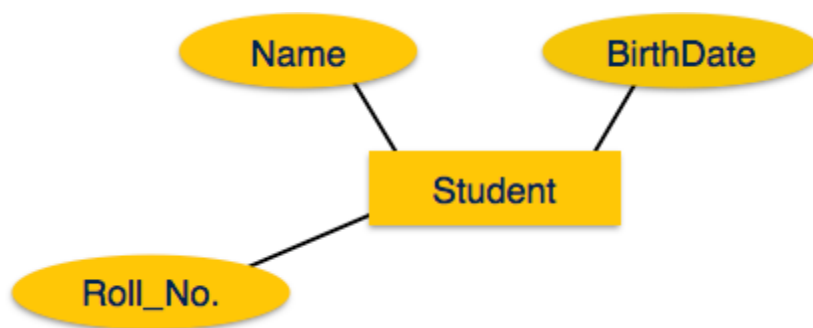
Entity

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.



Attributes

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



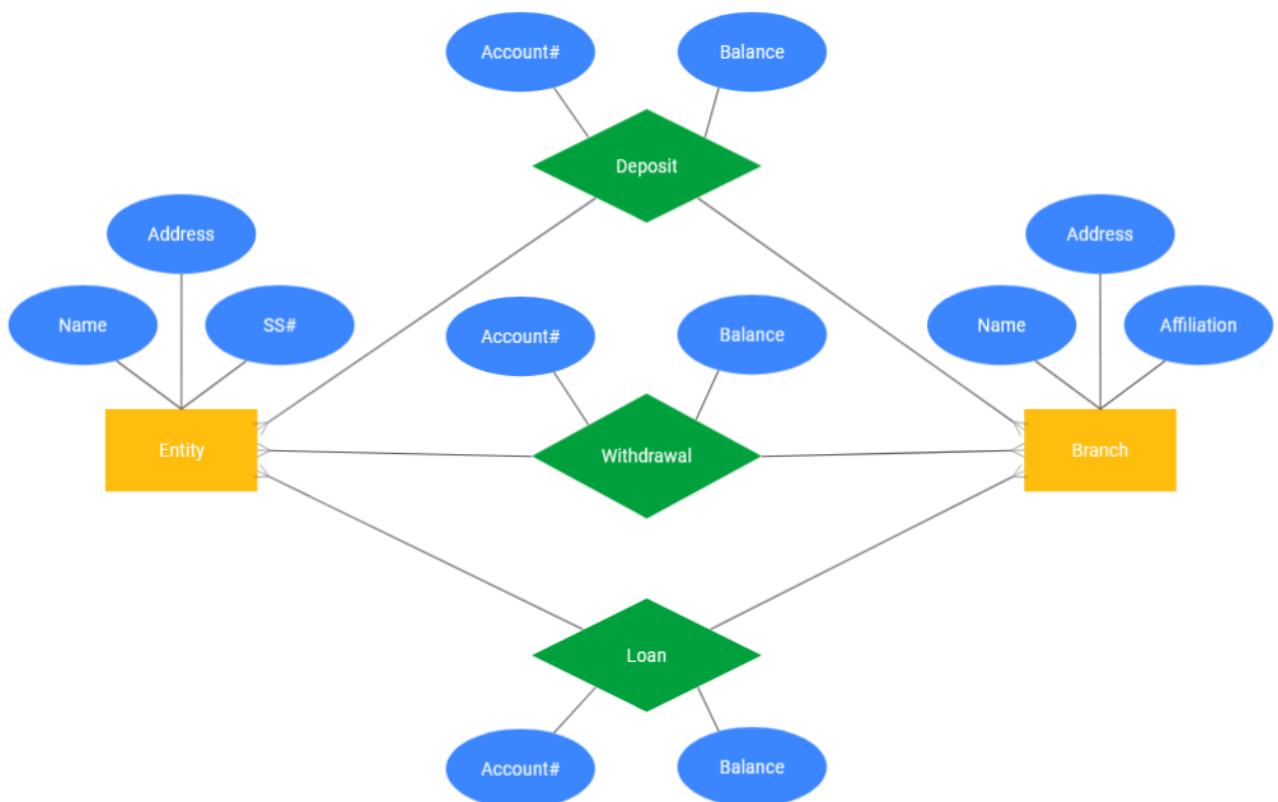
Relationship:

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line

1. One-to-one – When only one instance of an entity is associated with the relationship, it is marked as '1:1'.
2. One-to-many – When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.
3. Many-to-one – When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.

Many-to-many – The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.

Example of ER diagram:



Reference:

1. <https://www.smartdraw.com/entity-relationship-diagram/>
2. https://www.tutorialspoint.com/dbms/er_diagram_representation.htm

USE CASE

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Purpose of Use Case Diagrams

The main purpose of a use case diagram is to portray the dynamic aspect of a system.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

Example of a Use Case Diagram

A use case diagram depicting the Online Shopping website is given below.

Here the Web Customer actor makes use of any online shopping website to purchase online. The top-level uses are as follows; View Items, Make Purchase, Checkout, Client Register. The View Items use case is utilized by the customer who searches and view products. The Client Register use case allows the customer to register itself with the website for availing gift vouchers, coupons, or getting a private sale invitation. It is to be noted that the Checkout is an included use case, which is part of Making Purchase, and it is not available by itself.

Reference:

1. <https://www.javatpoint.com/uml-use-case-diagram>
2. https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm

Documentation of Project

What is project documentation?

Project documentation is the process of recording the key project details and producing the documents that are required to implement it successfully.

some basic documents are needed in most cases. Here are five project document examples that your project will likely require.

Project proposal

A project proposal is written to initiate a project – it's the first step in the project management process. The goal of this document is to convince the decision-makers and stakeholders that the

idea behind the project is worth pursuing. A project proposal needs to outline the project's core value proposition, which is often done in the form of a business case.

Project Documentation consist of:

1. Title and Cover Page:

It contains Title of Project

2. Declaration:

Declaration is a statement that project is done by X Person sincerely.

It concludes with name and signature.

3. Approval:

The approval page is a confirmation of Head about acceptance of Project.

It endorsed with signature of head and confirming their approval of the project.

4. Acknowledgment:

It depicts the gratitude, respect and thankfulness towards the people who helped in pursuing the project successfully.

5. Abstract :

Abstract represents a summarized report of the complete project in a very concise and informative format covering the main objective and aim of the project, the background information, processes and methods used, and methodologies implemented, followed with a brief conclusion of two to three lines talking about the results and scope of the project.

6. Index:

It contains all the key words in project in their page number.

7. Body of the Project and chapters :

The main body of the project should comprise several chapters with the corresponding titles, and each page within these chapters must be numbered in numerals as page numbers. The usual way of presenting these chapters is given below.

Chapter 1: Introduction chapter.

Chapter 2: Chapter of Literature Review.

Chapter 3-4 or 5: These chapters describe the overall in-depth information about the project.

8. Experiments and results :

9. Conclusion and Recommendations

The conclusion and recommendations part summarizes the whole report by highlighting all the chapters and their significance and the importance of the project and the achievements.

The Recommendations are interlinked with the conclusion. The conclusion drawn from the project report can be further implemented in the recommendation section to overcome the constraints of the project.

10. Referencing and Appendices

The project report must be considered as a very standard report, and therefore, it should follow all rules, guidelines, and protocols of gathering and presenting information, and implementing that, and drawing conclusions out of it.

Reference ;

1. <https://www.elprocus.com/final-project-report-format-for-electronics-engineering-students/>