

SYSTEM DESIGN

The most creative and challenging phase of the system development is system design, is a solution to how to approach to the creation of the proposed system. It refers to the technical specification that will be applied. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development. At an early stage in designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfil. The first step is to determine how the output is to be produced and in what format. Second input data and master files (database) have to be designed to meet the requirements of the proposed output. The operational (processing) phases are handled through program construction and testing. The system design includes:

- INPUT DESIGN
- OUTPUT DESIGN
- DATABASE DESIGN
- DATA FLOW DIAGRAM
- ER DIAGRAM
- USE CASE DIAGRAM

INPUT DESIGN

Input design is the process of converting user-oriented input to a based format. Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operators can be controlled by input design. The goal of designing input data is to make data entry as easy, logical and free from errors. When we approach input data design; we design the data source documents that capture the data and then select the media used to enter them into computer.

User-friendly screen format can reduce the burden on end users, who are not highly proficient in computers. An important step in input design stage is a design of source document. Source document is the form in which the data can initially capture. The next step is the design of the document layout. In the layout organizes the document by placing information, where it will be noticed and establishes the appropriate sequence of items.

OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Efficient and intelligent output design improves the system's relationship and helps user decision making. In the

output design it is determine how the implementation is to be played for immediate need and also the hardcopy output. A major form of input is a hardcopy from the printer. Printouts should be designed around the output requirement of the user. Printers, CRT screen display are the examples for providing computer based output. The output design associated with the system includes the various reports of the table generations and query executions.

Output design is one of the, most important features of the information system. The logical design of an information system is analogous to an engineering blue print of an automobile. It shows the major features and how they are related to one another. The outputs, inputs and databases are designed are in this phase.

DATABASE DESIGN

Database design, A most important part of the system design phase. In a database environment, data available are used by several users instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity.

Primary key is one of the candidate keys that are chosen to be the identifying key for the entire table.

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Here are the most commonly used normal forms:

- First normal form(1NF)
- Second normal form(2NF)
- Third normal form(3NF)
- Boyce - Codd normal form (BCNF)

FIRST NORMAL FORM (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

SECOND NORMAL FORM (2NF)

A table is said to be in 2NF if both the following conditions hold:

- **Table is in 1NF (First normal form)**
- No non-prime attribute is dependent on the proper subset of any candidate key of table.

THIRD NORMAL FORM (3NF)

Dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as non-prime attribute.

In other words, 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency $X \rightarrow Y$ at least one of the following conditions hold:

X is a super key of table

Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

BOYCE CODD NORMAL FORM (BCNF)

It is an advance version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the super key of the table..

DATA FLOW DIAGRAM (DFD)

The DFD also known as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data and the output data generated by the system. The main reason why this DFD technique is so popular is probably because of the fact that DFD is a very simple formalism- it is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among these systems. Starting with a set of high-level functions that a system performance of DFD model in hierarchically it represents various sub functions. The Data Flow Diagramming technique also follows a simple set of intuitive concepts and rules.

Data flow diagram (DFD) is used to show how data flows through the system and the processes that transform the input data into output. Data flow diagrams are a way of expressing system requirements in a graphical manner. DFD represents one of the most ingenious tools used for structured analysis

BASIC DFD SYMBOLS

Function Symbol:

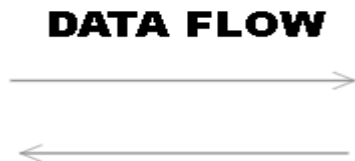


A function is represented using a circle or round-edged rectangle. This symbol is called a process or a bubble. Bubbles are annotated with the names of corresponding functions.

External Entity Symbol:

An external entity such as a user, project manager etc. is represented by a rectangle. The external entities are essentially those physical entities external to the application system, which interact with the system by inputting data to the system or by consuming the data produced by the system. In addition to the human users the external entity symbols can be used to represent external hardware and software such as application software.

Data Flow Symbol:



A directed arc or an arrow is used as a Data Flow Symbol. This represents the data flow occurring between two processes or between an external entity and a process; in direction of the Data Flow Arrow. Data flow Symbols are annotated with corresponding data names.

Data Store Symbol:



A Data Store represents a logical file; it is represented using two parallel lines. A logical file can represent either Data Store Symbol, which can represent either data structure or a physical file on disk. Each data store is connected to a process by means of a Data Flow Symbol. The direction of the Data Flow Arrow shows whether data is being read from or written into a Data Store. An arrow flowing in or out of a data store implicitly represents the entire area of the Data Store and hence arrows connecting to a data store need not be annotated with the names of the corresponding data items.

ER DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

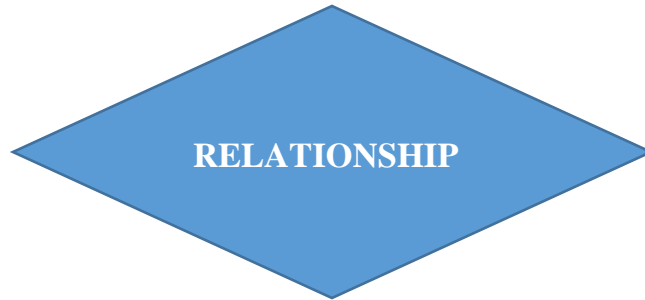
Attributes:

Attributes are the **properties which define the entity type**. In ER diagram, attribute is represented by an oval.



Relationship:

A relationship type represents the **association between entity types**. In ER diagram, relationship type is represented by a diamond.



Entity

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

Actor:

The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.



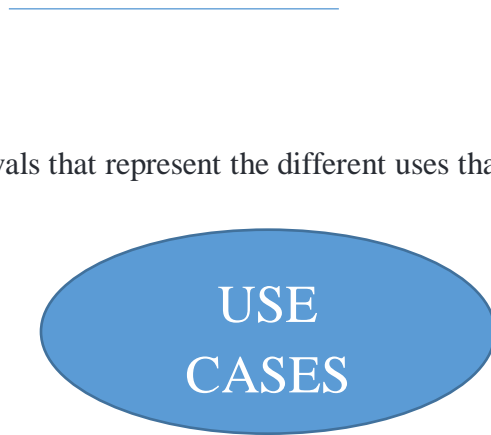
System:

A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system.

SYSTEM

Association:

A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.



Use cases:

Horizontally shaped ovals that represent the different uses that a user might have.

TABLE DESIGN

The Table Designer is a **visual tool where you design and visualizes database tables**.

Fields:

A field is the smallest unit of stored data. Each field consist of data of specific type.

Constraints:

SQL constraints are **used to specify rules for the data in a table**. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

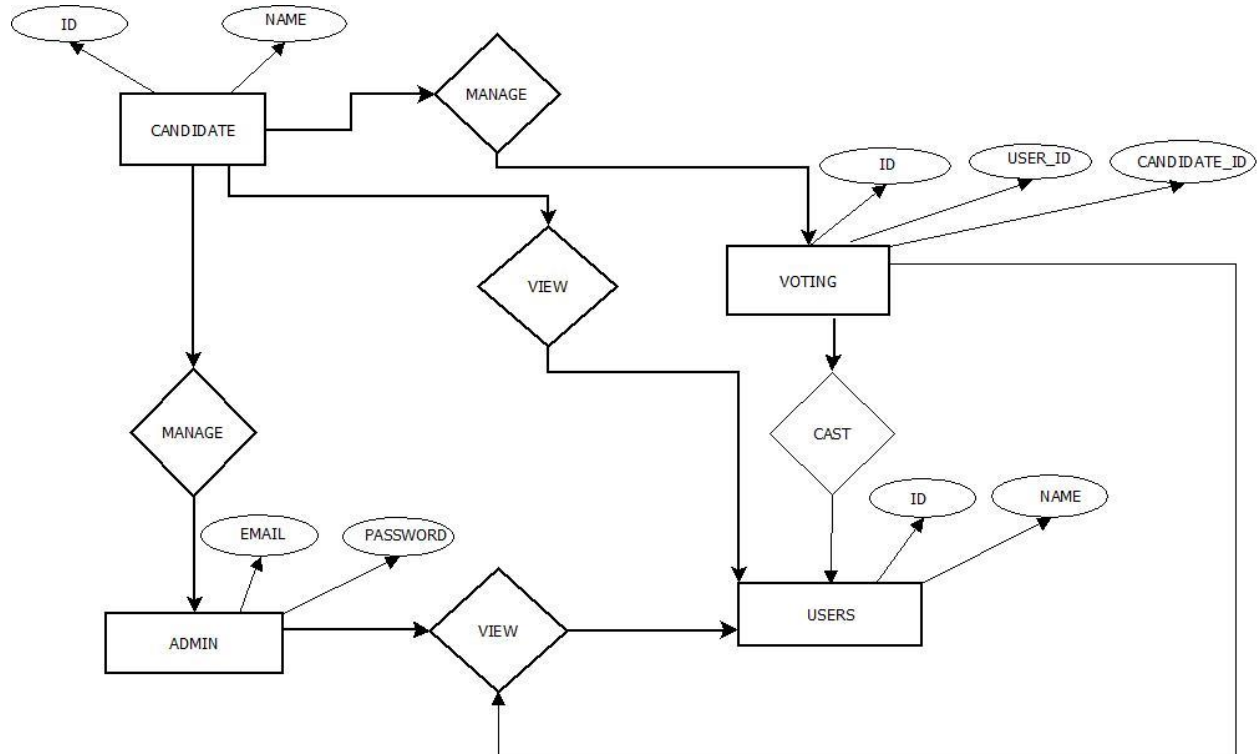
- Primary key:

The PRIMARY KEY constraint **uniquely identifies each record in a table**. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

- Foreign key:

A foreign key is a **column or columns of data in one table that connects to the primary key data in the original table.**

ER DIAGRAM



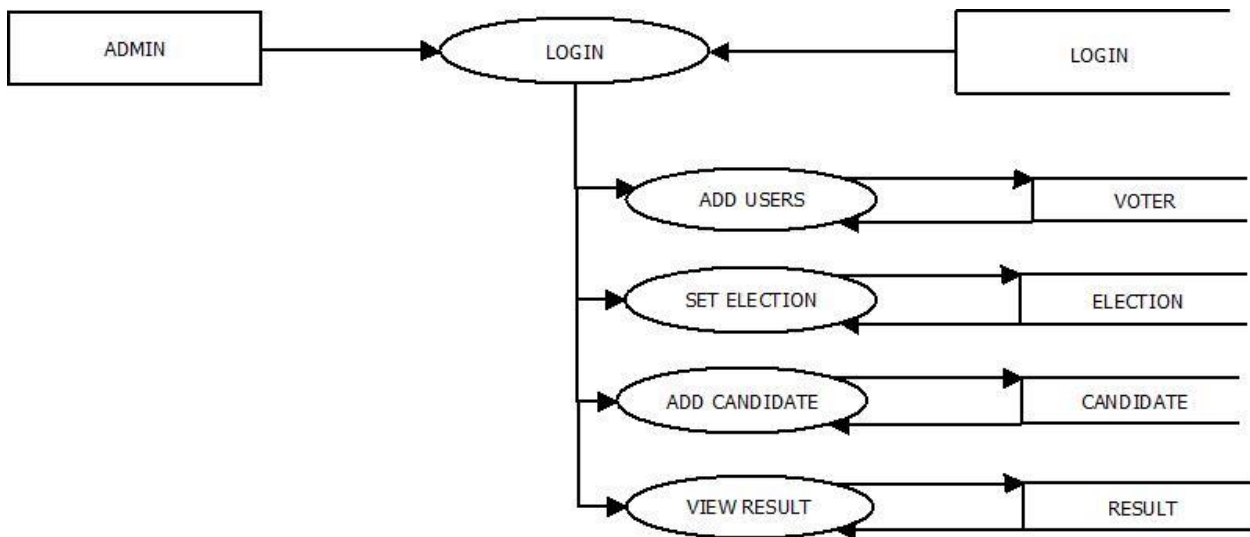
DATA FLOW DIAGRAM [DFD]

LEVEL 0



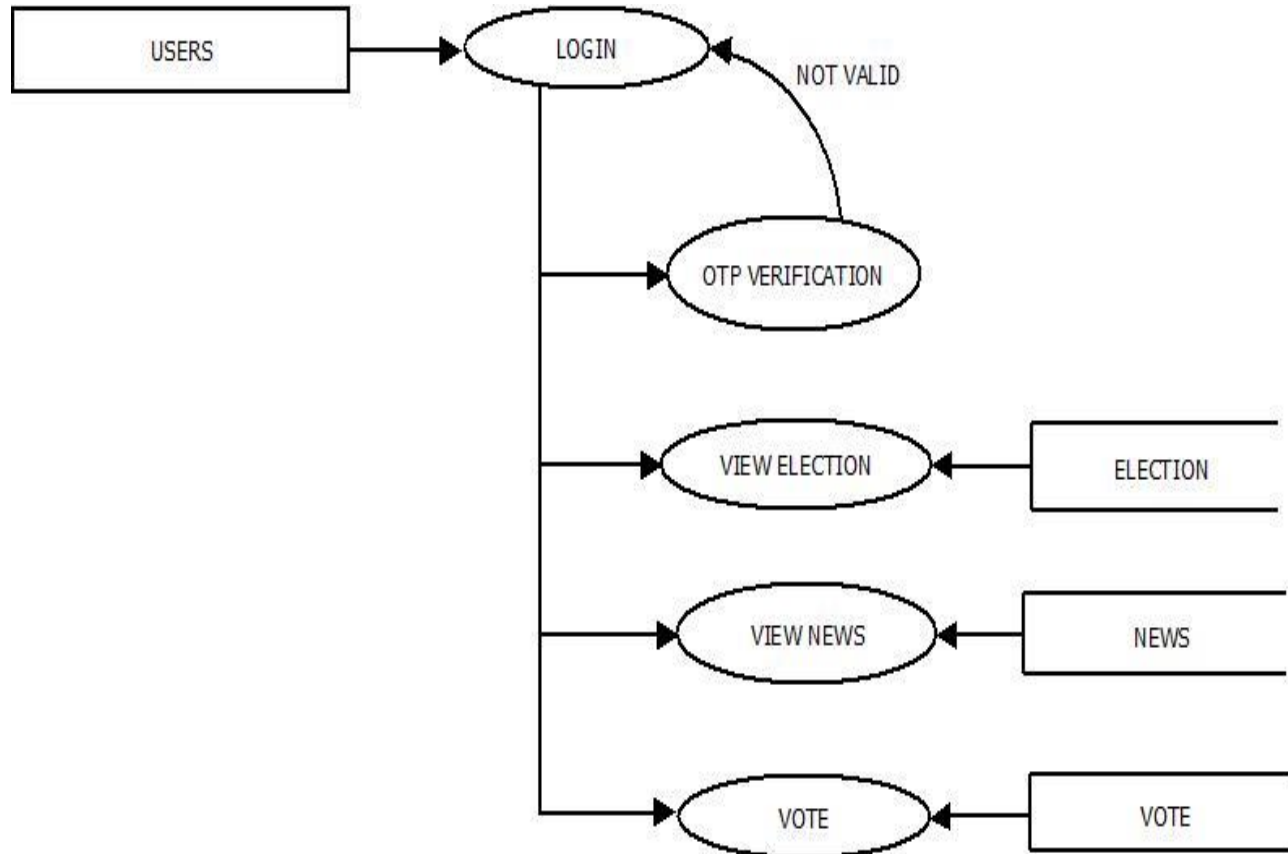
LEVEL 0

LEVEL 1 FOR ADMIN



LEVEL 1 FOR ADMIN

LEVEL 2 FOR USERS



LEVEL 2 FOR USER

USE CASE DIAGRAM

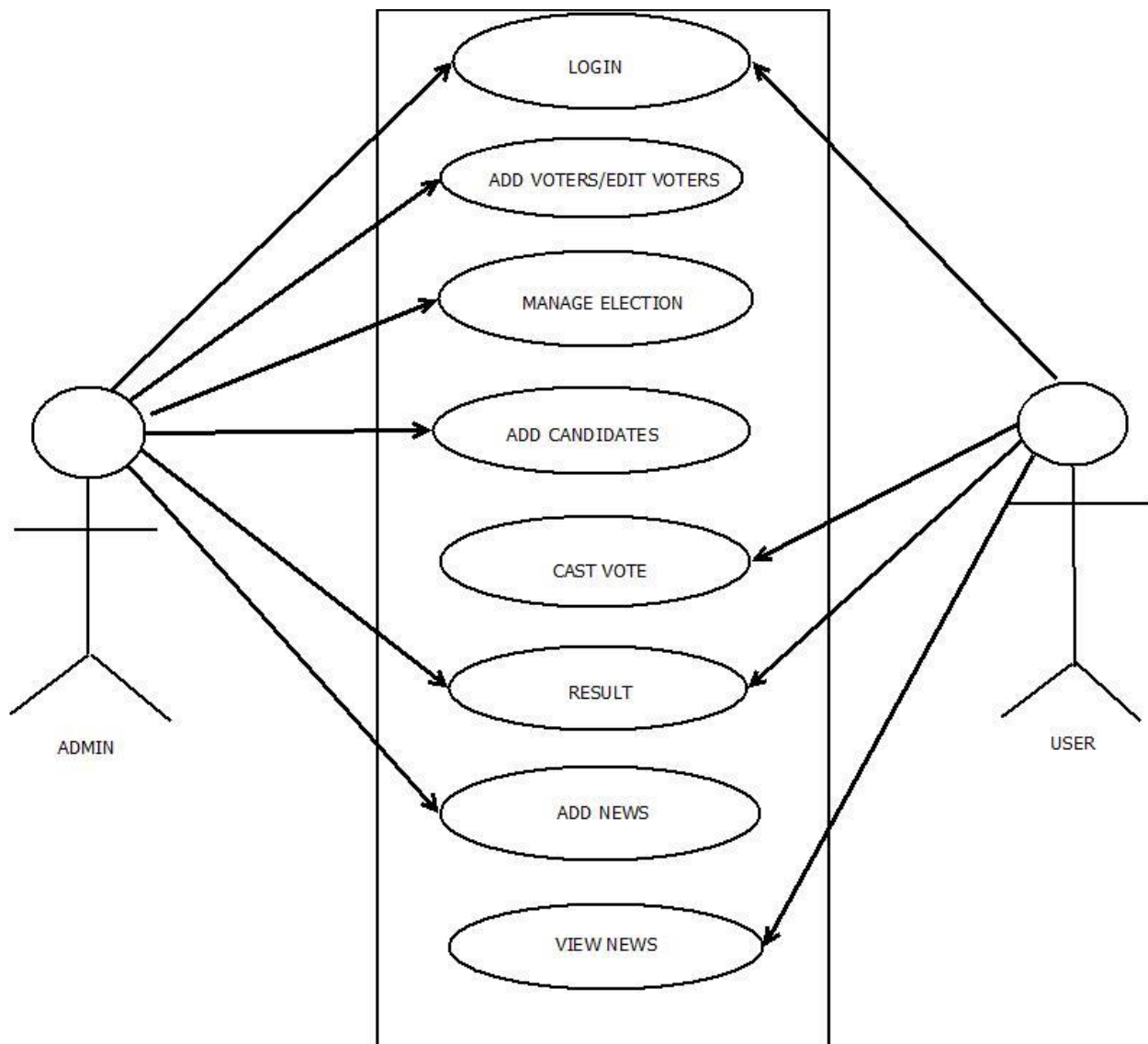


TABLE DESIGN

1. CANDIDATE TABLE

FIELD	DATA TYPE	SIZE	DESCRIPTION
ID	Int	10	Used to store id
NAME	Varchar	20	Used to store name
EMAIL	Varchar	30	Used to store email
ADDRESS	Varchar	50	Used to store address
PHONE	Varchar	20	Used to store phone number
PARTY NAME	Varchar	10	Used to store party name
SYMBOL	Varchar	20	Used to store symbol
IMAGE	Varchar	20	Used to store image
GENDER	Varchar	7	Used to store gender

2. USER TABLE

FIELD	DATA TYPE	SIZE	DESCRIPTION
ID	Int	10	Used for id(primary key)
NAME	Varchar	20	Used to store name
EMAIL	Varchar	30	Used to store email
PASSWORD	Varchar	15	Used to store password
ADDRESS	Varchar	50	Used to store address

PHONE	Varchar	20	Used to store phone number
GENDER	varchar	10	Used to store gender

3. VOTING TABLE

FIELD	DATA TYPE	SIZE	DESCRIPTION
ID	Int	10	Used to store id
USER ID	Int	10	Used to store user id
CANDIDATE ID	Int	10	Used to store candidate id
DATE & TIME	Date	20	Used to store date and time
SYMBOL	Varchar	30	Used to store symbol