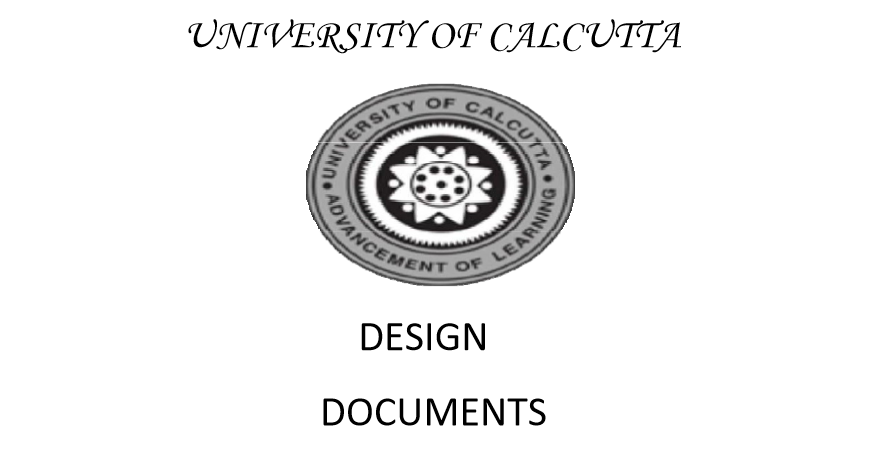
****

**GROUP NO 🡪 09**

**NAME : -**

1. Jyoti kumari (19)
2. Manohar kumar verma(40)
3. Ayush verma(10)

**Design Document-“Canteen Automation System”**

**Data Flow Diagram:**

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

The data-flow diagram is part of the structured-analysis modelling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data-flow plan.

Data flow diagram is the starting point of the design phase that functionally decomposes the requirements specification. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flow rather than how they are processed, so it does not hardware, software and data structure.

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

The data flow diagram is a graphical description of a system’s data and how to

Process transform the data is known as Data Flow Diagram (DFD).

Unlike details flow chart, DFDs don’t supply detail descriptions of modules that graphically describe a system’s data and how the data interact with the system.

Following Symbol are used in our Dataflow diagram. The significance of different Shapes are as follows :

process

Data store

Source/sink

Data flow

(DeMarco & Yourdon symbols)

There are seven rules for construction of a data flow diagram.

];

1. Arrows should not cross each other.
2. Squares, circles and files must wear names.
3. Decomposed data flows must be balanced.
4. No two data flows, squares or circles can be the same names.
5. Draw all data flows around the outside of the diagram.
6. Choose meaningful names for data flows, processes & data stores.
7. Control information such as record units, password and validation requirements are not penitent to a data flow diagram.

Additionally, a DFD can be utilized to visualize data processing or a structured design.

This basic DFD can be then disintegrated to a lower level diagram demonstrating smaller steps exhibiting details of the system that is being modeled.

Dataflow Diagram have different Levels. As Level of DFD increases more details are explained or stated in DFD.

Here is a Level 0 and Level 1 DFD of our Project on Canteen Automation System.

Level 0 :

Authentication, Verification and Control

Admin

i/p o/p

Maintaining and Managing Canteen

Seller

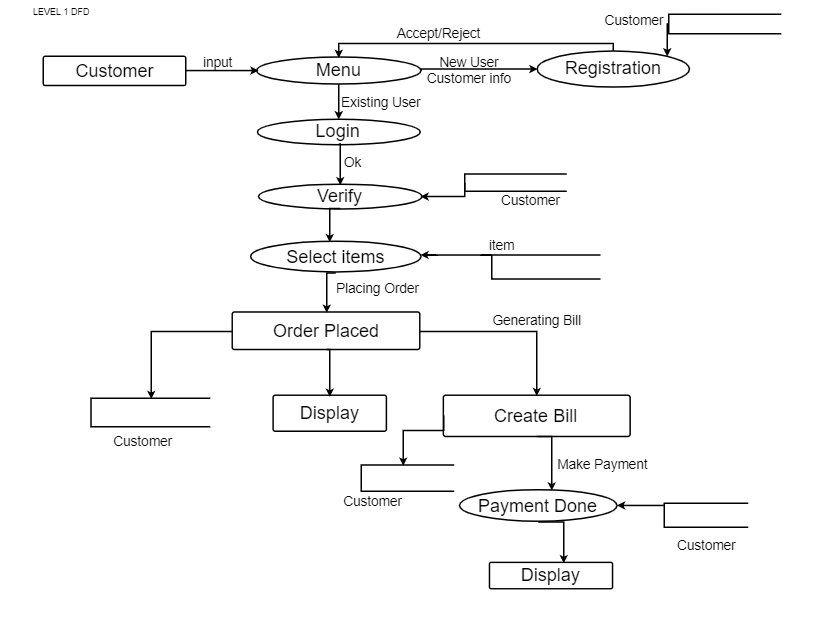
i/p o/p

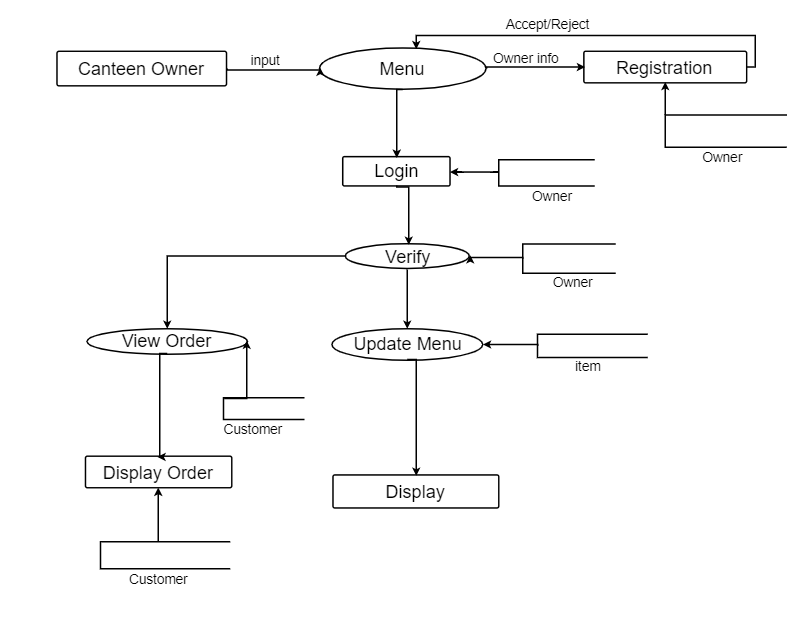
View menu & Place Order

Customer

i/p o/p

Level 1:





**ER Diagram:**

**Entity Relationship Diagram**

E-R Model is a popular high level conceptual data model. This model and its variations are frequently used for the conceptual design of database application and many database design tools employ its concept.

A database that confirms to an E-R diagram can be represented by a collecton of tables in the relational system. The mapping of E-R diagram to the entities are:

* Attributes
* Relations

o Many-to-many

o Many-to-one

o One-to-many

o One-to-one

* Weak entities
* Sub-type and super-type

*Symbols used in Entity-Relationship Diagram are as follows:*

* Represent Data Entity.
* Represent connection Administrator.
* Connect two Entities /

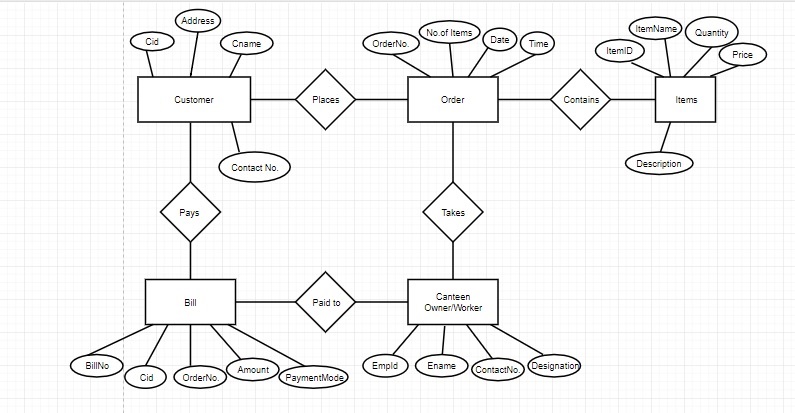
One to one relationship.

* Represent the relationship.

The entities and their relationships between them are shown using the following conventions. :-

* A data model is a mechanism that provides this abstraction for database application.
* Data modelling is used for representing entities and their relationship in the database.
* Entities are the basic units used in modelling database entities can have concrete existence or constitute ideas or concepts.
* Entity type or entity set is a group of similar objects concern to an organization for which it maintains data,
* Properties are characteristics of an entity also called as attributes.
* A key is a single attribute or combination of 2 or more attributes of an entity set is used to identify one or more instances of the set.
* In relational model we represent the entity by a relation and use tuples to represent an instance of the entity.
* Relationship is used in data modelling to represent in association between an entity set.

Following is the ER Diagram of our Project – “Canteen Automation System”



ER Diagram, Canteen Automation System

In this ER diagram we have considered Five Entity set namely Customer, Order, Item, Bill, and Canteen Owner. Here we have considered five relations among the entity sets. They are:” Places, Contains, Pays, Takes, and Paid to”. Customer Places Order and pays bill and Bill is paid to canteen owner and Canteen owner takes order which contains some items. This scenario is explained in the ER diagram.

**Assumption:** Canteen owner refers to the worker working in the canteen who are entitled to take order.

Each entity set have some set of attribute which is maintained in the database.

**Customer:** Attributes of Customer Entity set are Cid, Cname, Address, ContactNo. Cid is primary key.

**Order:** Attributes of Order Entity set are Order No., No. of items, Date and Time. Order No. is primary key.

**Item:** Attribute of items are ItemID, ItemName, Quantity, Price, Description. ItemID is primary key.

**Bill:** Attribute of Bill are BillNo, Cid, OrderNo., Amount, PaymentMode. Here BillNo. Is the primary Key.

**Canteen Owner/Worker :** Attributes of Canteen owner/Worker are Empid, Ename, ContactNo., And Designation. Here Empid is the Primary Key.

**Database Tables :**

|  |  |  |  |
| --- | --- | --- | --- |
| Customer | | | |
| Cid | Cname | Address | Contact No. |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Order | | | |
| OrderNo. | No. of Items | Date | Time |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | | | | |
| ItemID | ItemName | Quantity | Price | Description |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bill | | | | |
| BillNo. | Cid | Order no. | Amount | PaymentMode |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Canteen Owner/Worker | | | |
| Empid | Ename | ContactNo. | Designation |
|  |  |  |  |

|  |  |
| --- | --- |
| Places | |
| Cid | OrderNo. |
|  |  |

|  |  |  |
| --- | --- | --- |
| Contains | | |
| OrderNo. | No. of items | Item id(Multivalued) |
|  | |  |

|  |  |
| --- | --- |
| Item id | |
| Order No. | Item id |
|  |  |

|  |  |
| --- | --- |
| Bill | |
| BillNo | Cid |
|  |  |

|  |  |
| --- | --- |
| Paid to | |
| Empid | BillNo. |
|  |  |

|  |  |
| --- | --- |
| Takes | |
| Empid | OrderNo. |
|  |  |

**Interface :**

**System Design of Company Canteen Management System**

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the clients’s requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. ***Primary Design Phase***:

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

1. ***Secondary Design Phase:***

In the secondary phase the detailed design of every block is performed.

**The general tasks involved in the design process are the following:**

1. Design various blocks for overall system processes.

2. Design smaller, compact and workable modules in each block.

3. Design various database structures.

4. Specify details of programs to achieve desired functionality.

5. Design the form of inputs, and outputs of the system.

6. Perform documentation of the design.

7. System reviews.

**User Interface Design**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the

system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

***The following steps are various guidelines for User Interface Design:***

1. The system user should always be aware of what to do next.

2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.

3. Message, instructions or information should be displayed long enough to allow the system user to read them.

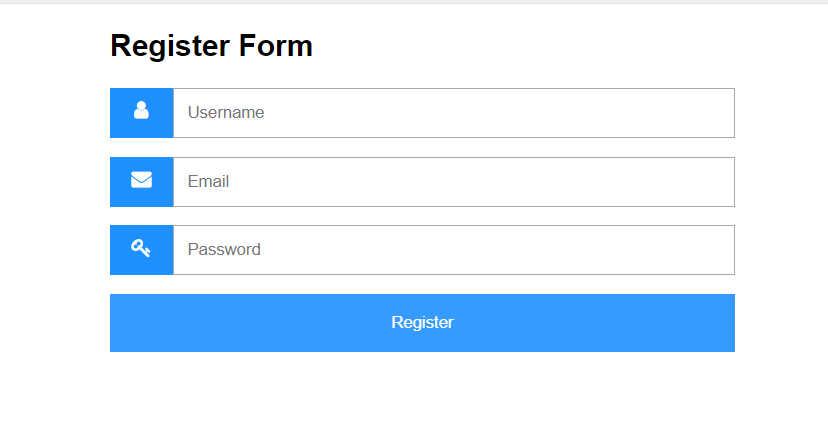
4. Use display attributes sparingly.

5. Default values for fields and answers to be entered by the user should be specified.

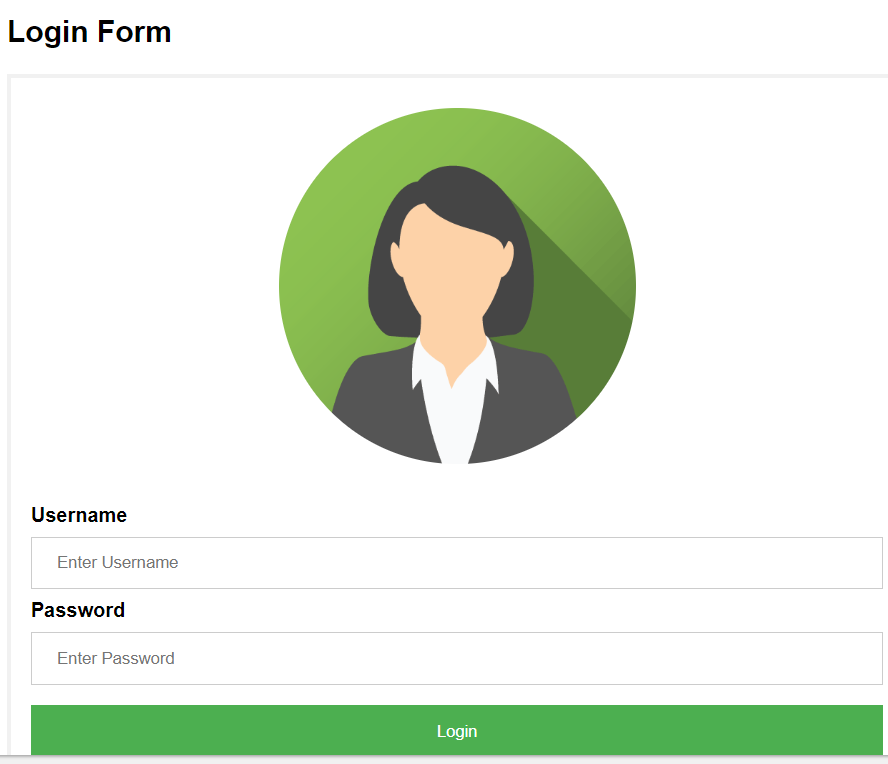
6. A user should not be allowed to proceed without correcting an error.

7. The system user should never get an operating system message or fatal error.

USER INTERFACE REGISTRATION PAGE



USER INTERFACE LOGIN PAGE



And so on…..