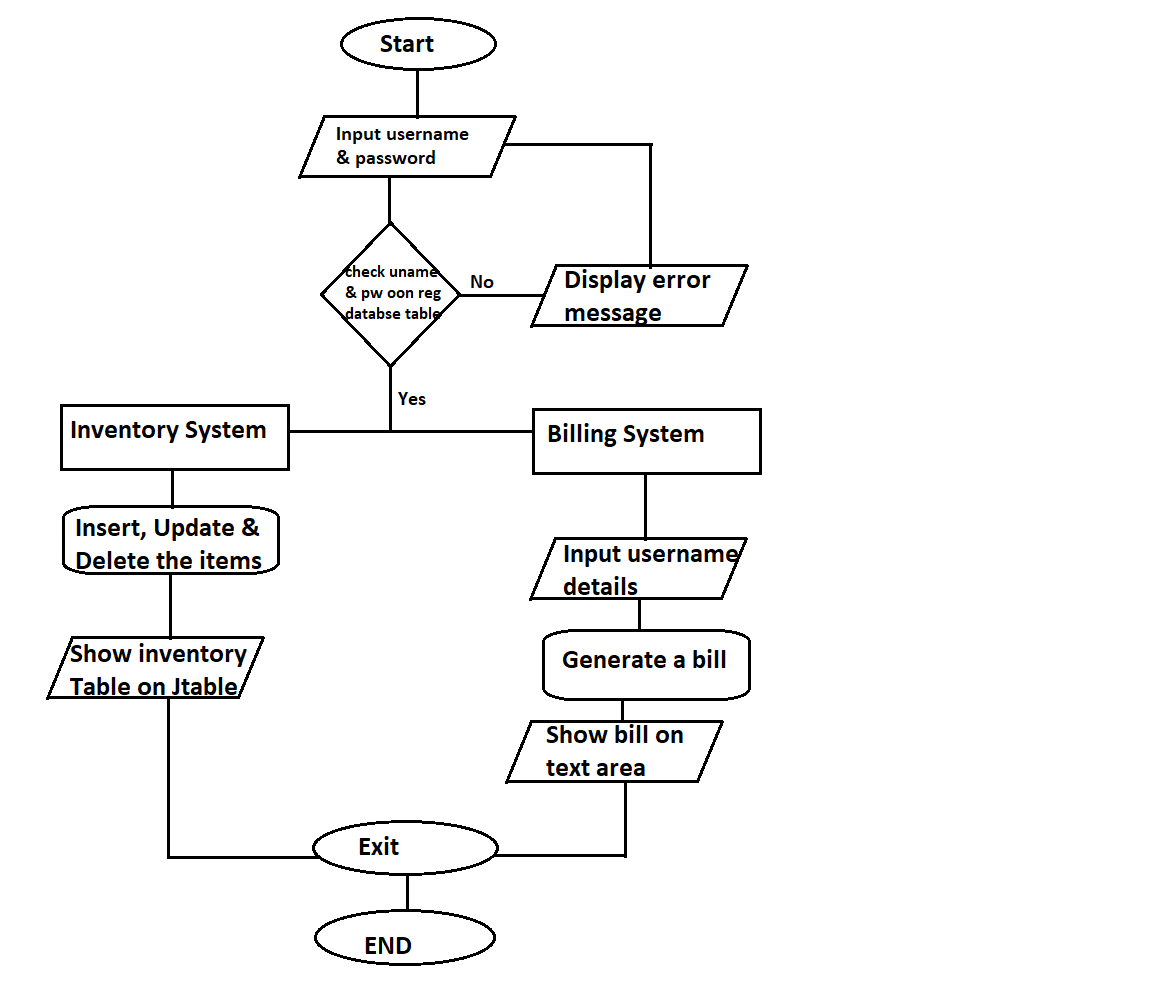
**CHAPTER 3**

**SYSTEM DESIGN**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

**3.1 Flow Chart**

A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. With proper design and construction, it communicates the steps in a process very effectively and efficiently. Fig 3.1 depicts the flow chart of Shop Inventory and Billing Systemwhich shows various processing steps based on different events, actions and conditions.



**Fig 3.1:** *Flowchart of Shop Inventory and Billing System*

**3.2 Database Table Design**

The basic database unit is the table. A table is a unit consisting of rows of related information. Each row consists of fields of information where data is stored. Field attributes include information and rules that govern the data stored in the field. The field attributes and rules may limit the type of data stored in the field.

A field may be defined as a key or may be limited by rules requiring specific masks, such as a field may limited to dates, formatted numbers like telephone numbers, or be limited to a specific number of characters. The database schema contains these rules. Database tables used in Shop Inventory and Billing System are shown below.

***Table 3.2.1:*** *Registration*

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Constraints** |
| Fname | Varchar(45) | Not Null |
| Lname | Varchar(45) | Not Null |
| Uname | Varchar(45) | Primary Key |
| Pass | Varchar(45) | Not Null |
| Repass | Varchar(45) | Not Null |
| Cnum | Varchar(45) | Not Null |

The above registration table is used to store user related information like first name, last name, user name, password, Re-password, Contact number when new user Registered to the Database.

***Table 3.2.2:*** *Customer*

|  |  |  |
| --- | --- | --- |
| ***Name*** | ***Datatype*** | ***Constraints*** |
| Cust\_id | Varchar(45) | Primary Key, Auto Increment |
| Cust\_name | Varchar(45) | Not Null |
| Cust\_number | Varchar(45) | Not Null |
| Cust\_address | Varchar(45) | Not Null |
| Bill\_number | Varchar(45) | Not Null |

The above Customer table used to store the customer related information like Customer id, Customer name, Customer number, Customer Address, Bill number when new customer bills a product.

***Table 3.2.3:*** *Inventory*

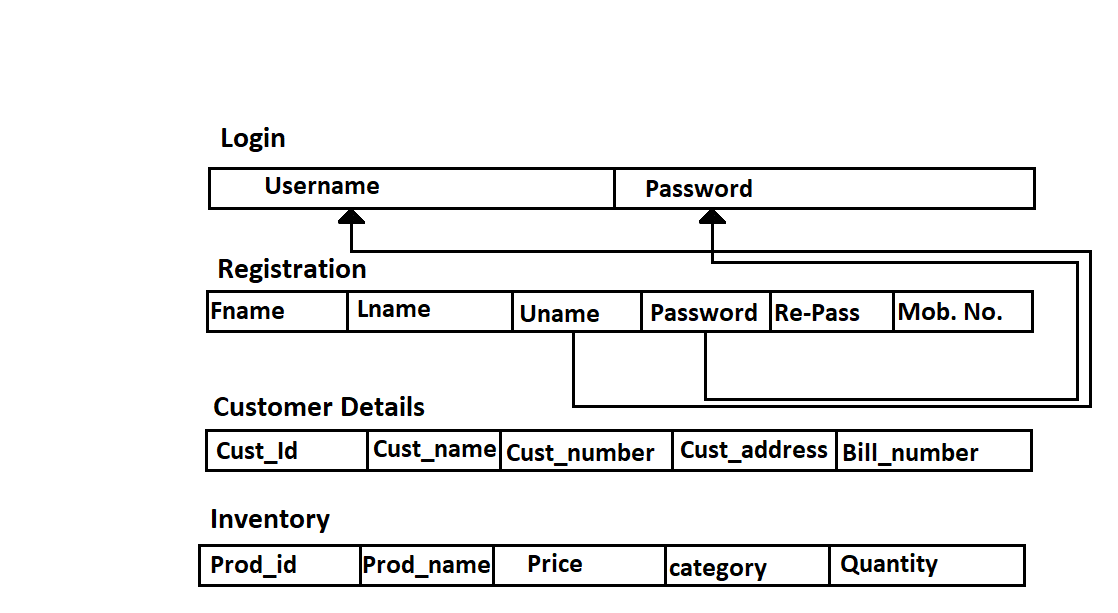
|  |  |  |
| --- | --- | --- |
| ***Name*** | ***Datatype*** | ***Constraints*** |
| Prod\_id | Varchar(45) | Primary Key, Auto Increment |
| Prod\_name | Varchar(45) | Not Null |
| Prod\_price | Varchar(45) | Not Null |
| Category | Varchar(45) | Not Null |
| Quantity | Varchar(45) | Not Null |

The above Inventory table used to store the product related information like Product id, Product name, Product price, Category, Quantity when new product stored in the Inventory.

**3.3 Schema Diagram**

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema’s underlying mission business rules for which the database is created. In a schema diagram, all database tables are designated with unique columns and special features, e.g., primary/foreign keys or not null, etc. Formats and symbols for expression are universally understood, eliminating the possibility of confusion. The table relationships also are expressed via a parent table’s primary key lines when joined with the child table’s corresponding foreign keys.

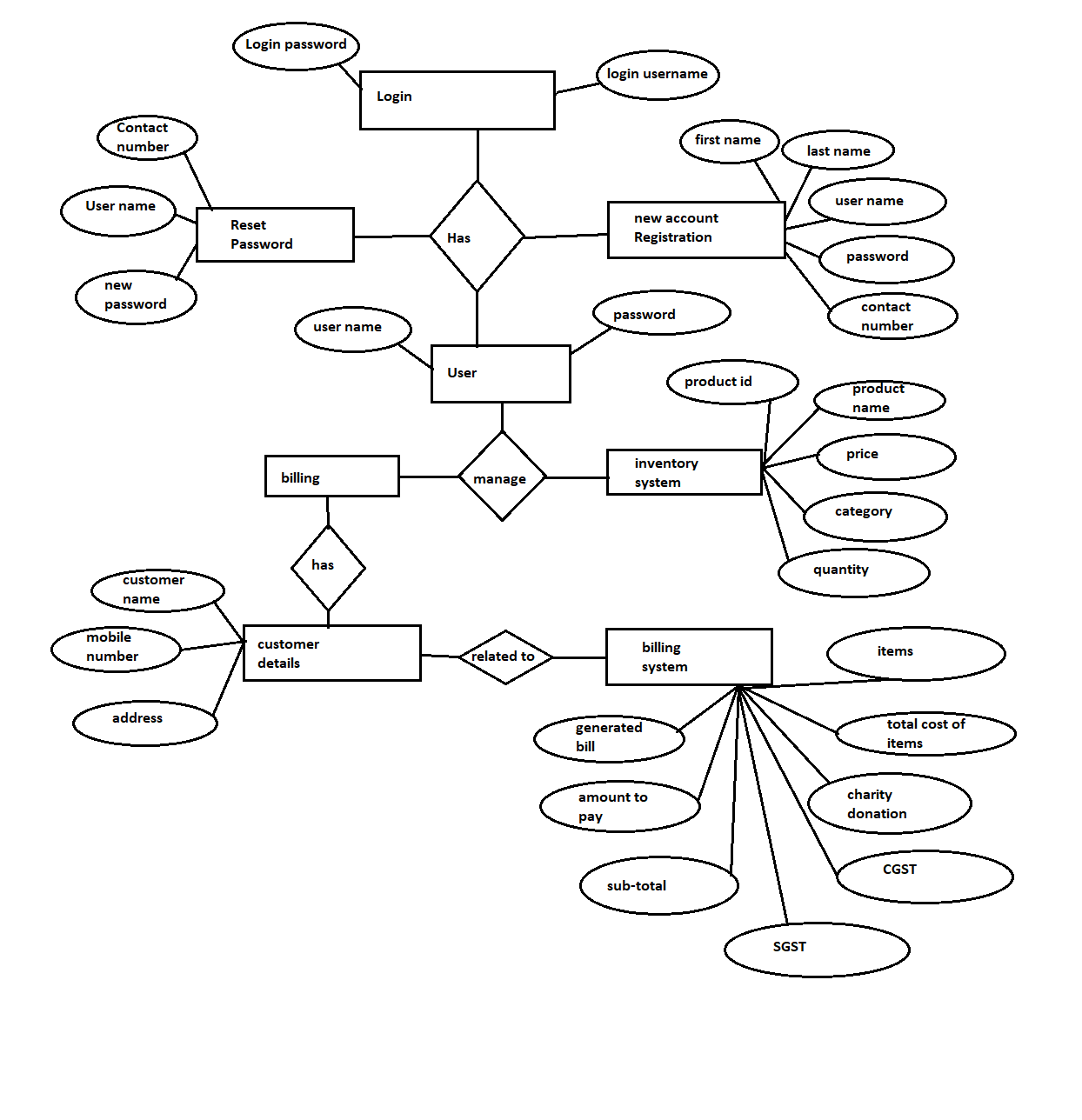
Schema diagrams have an important function because they force database developers to transpose ideas to paper. This provides an overview of the entire database, while facilitating future database administrator work. Fig 3.2 shows the schema diagram of Shop Inventory and Billing System.



**Figure 3.2:** *Schema Diagram of Shop Inventory and Billing System*

**3.4 ER Diagram**

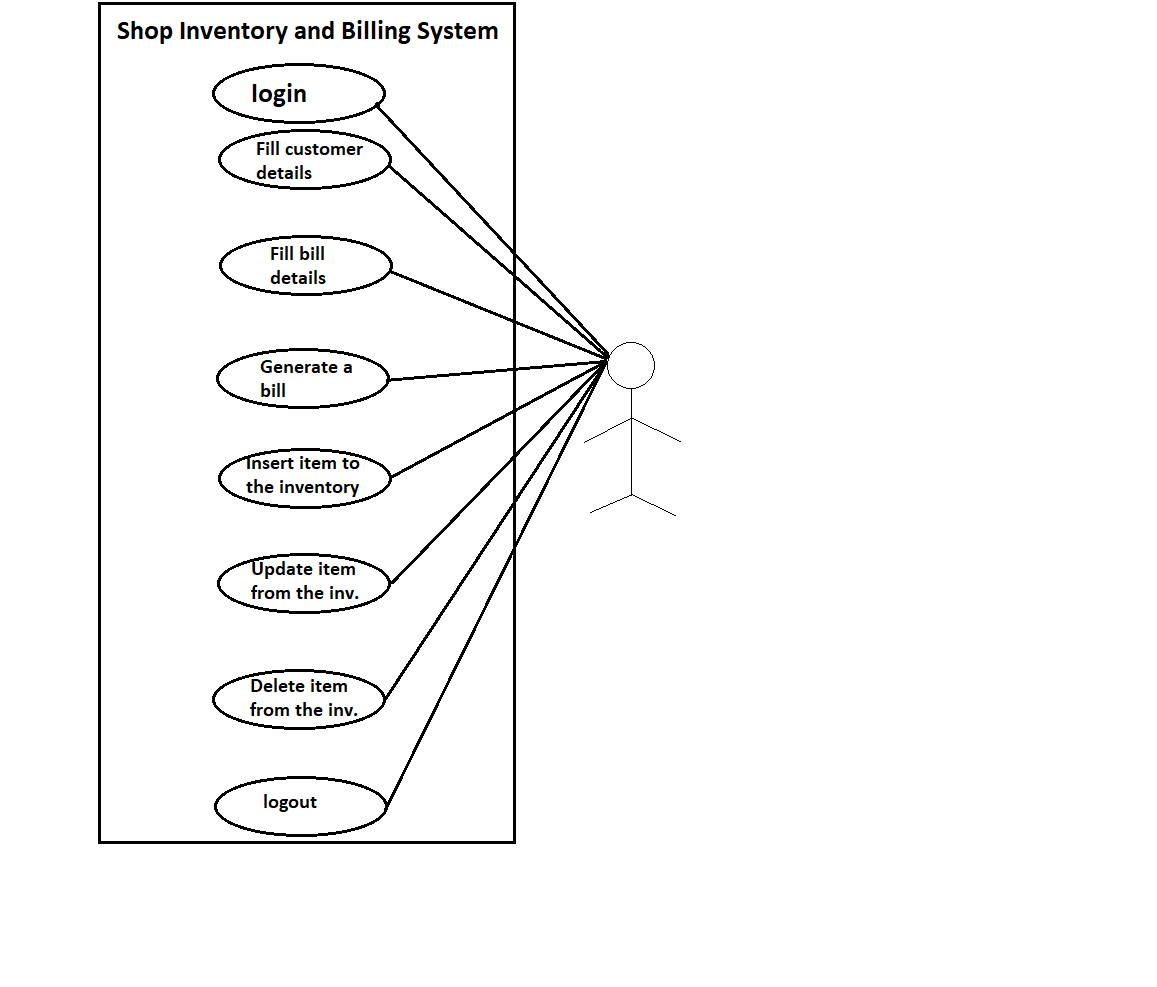
An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database. Fig 3.3 depicts the ER diagram of Shop Inventory and Billing System. It shows various entities, their attributes, relationships with other entities, cardinality ratios between entities and participation constraints used in designing Shop Inventory and Billing Systemdatabase.



**Fig 3.3**: *ER diagram of Shop Inventory and Billing System*

**3.5 Use case Diagram**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Users interacting with application are shown outside with stickman symbol.



**Fig 3.4*:*** *Use case diagram of Shop Inventory and Billing System*