**CHAPTER 3**

**DESIGN**

**3 SYSTEM DESIGN**

Design is a meaningful engineering representation of something that is to be built. It is an iterative process through which requirements are translated in to a blueprint for constructing the software. The goal of the design phase is to plan a solution of the problem specified by the requirements document. Major activities during the design phase are:

* Data Base Design
* Architectural Design
* Interface Design
* Modular Design

**3.1 DATABASE DESIGN**

A database is storing some data. All entries and attributes have been identified while creating the database. The collections of inter related data stored with minimum redundancy to serve many users quickly and efficiently. In database design data independence, accuracy, privacy, and security are given higher priority. Database design is an integrated approach to file design. This activity deals with the design of the physical e database design deals with the grouping of data into number of tables so as to reduce the duplication of data, minimize storage space, and retrieve the data efficiently.

Guidelines for designing a database:

1. Design a relational schema so that it is easy to explain its meaning. Do not combine attributed from multiple entity and relationship types into a single relation.

2. Design the database schema so that no insertion, deletion or modification anomalies are present in the relation.

3. As far as possible, avoid placing attributes in a base relation whose values may frequently be null.

**Advantages**

1. Data independence

2. Accuracy and integrity

3. Avoiding inordinate delays

4. Recovery from failure

5. Privacy and security.

**3.1.1** **E-R DIAGRAM**

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and relationship between entities, within an information system.

**There are three basic elements in ER models:**

* **Entities** are the “things” aboutwhich we seek information
* **Attributes** are the data we collect about entities.
* **Relationships** provided the structure needed to draw information from multiple entities.

**E-R Diagram Symbols:**

Entity

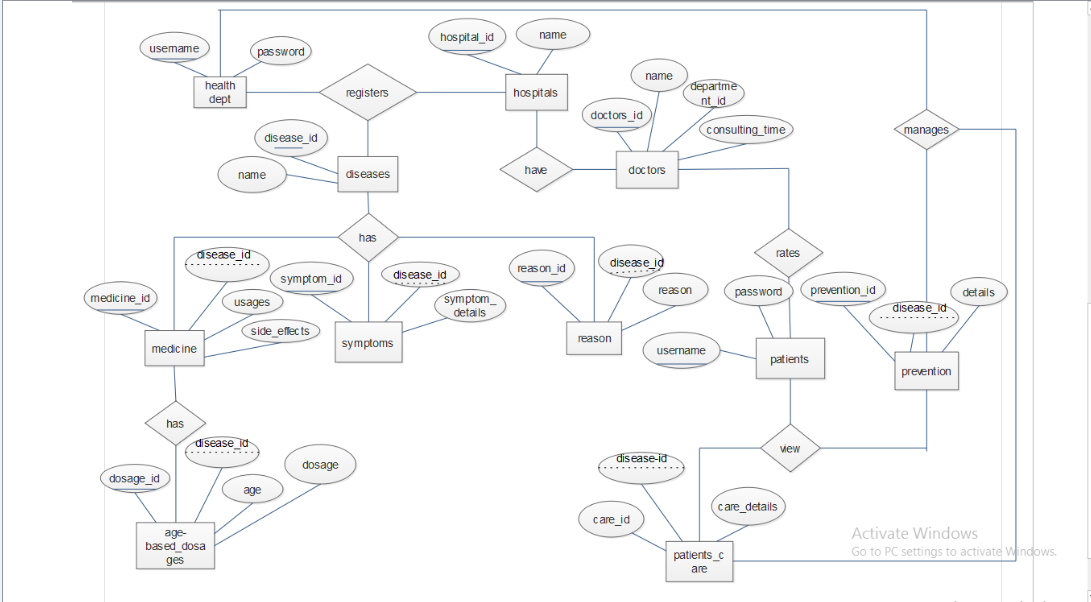
Attributes

Relation

Key Attribute

Composite Attribute

**E –R DIAGRAM**

****

**3.1.2** **TABLE DESIGN**

In the database all the information are stored in the form of tables. A table is simply a way storing data in rows and columns. In the system data is stored in many tables. The table structures are shown below with sample data.

**TABLES**

**1. register**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| User\_id | int | Primary Key | Id of user |
| Name | varchar(15) | Not null | Name of user |
| Place | varchar(15) | Not null | Place of user |
| Mobile | numeric(12,0) | Not null | Mobile number |
| Email | varchar(15) | Not null | Email of user |

**2. login**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| log\_id | int | Primary key | Login id of user |
| User-id | varchar(15) | Foreign key | id of user |
| password | varchar(15) | Not null | Password |
| User\_role | varchar(15) | Not null | Role of user |

**3. disease**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Datatype** | **Key** | **Description** |
| disease\_id | int | Primary key | Id of disease |
| Name | Varchar(25) | Not null | Name of disease |

**4. disease\_reason**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Datatype** | **Key** | **Description** |
| disease\_reason\_id | int | Primary key | Disease reason id |
| disease\_id | int | Foreign key | Id of disease |
| Reason | Varchar(450) | Not null | Reason for disease |

**5. age\_based\_dosage**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| dosage\_id | int | Primary key | Id of dosage |
| medicine\_name | varchar(25) | Not null | Name of medicine |
| Age | int | Not null | Age |
| dosage \_details | varchar(450) | Not null | Dosage details |

**6. hospitals**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| hospital \_id | int | Primary key | Id of hospital |
| hospital\_name | varchar(25) | Not null | Name of hospital |
| disease\_id | int | Not null | Id of disease |
| Addressline | varchar(25) | Not null | Address of hospital |
| Place | varchar(450) | Not null | Place |
| Phone | varchar(45) | Not null | Hospital Phone number |
| Latitude | float | Not null | Direction |
| Longitude | float | Not null | Direction |

**7. doctors**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| doctor\_id | int | Primary key | Id of doctor |
| hospital\_id | int | Foreign key | Id of hospital |
| Name | varchar(25) | Not null | Name of doctor |
| department \_id | int | Not null | Department of doctor |
| consulting\_time | varchar(45) | Not null | Consulting time |

**8.doctors\_rating**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| rate\_id | int | Primary key | Doctors rating id |
| user\_id | int | Foreign key | Id of user |
| doctor\_id | int | Foreign key | Id of doctor |
| Date | datetime | Not null | Date |
| rate\_details | varchar(45) | Not null | Details of rating |

**9.food-details**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| Food\_detail\_id | int | Primary key | Food id |
| disease\_id | int | Foreign key | Id of disease |
| Details | varchar(45) | Not null | Food details |

**10.medicine**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| medicine\_id | int | Primary key | Id of medicine |
| Name | varchar(25) | Not null | Name of medicine |
| side\_effect | varchar(450) | Not null | Side effect of medicine |
| Usages | varchar(450) | Not null | Medicine usages |

**11.Patient\_care**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| patient\_care\_id | int | Primary key | Id of patients care |
| disease\_id | int | Foreign key | Id of disease |
| care\_details | varchar(450) | Not null | Patient care details |

**12.Preventions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| prevention\_id | int | Primary key | Id of prevention |
| disease\_id | int | Foreign key | Id of disease |
| prevention\_details | varchar(450) | Not null | Prevention details |

**13.Symptoms**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| symptom\_id | int | Primary key | Id of symptoms |
| disease\_id | int | Foreign key | Id of disease |
| symptom\_details | varchar(450) | Not null | Details of symptom |

**14.user\_question**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Datatype** | **Key** | **Description** |
| question\_id | int | Primary key | Id of question |
| user\_id | int | Foreign key | Id of user |
| Doubt | varchar(45) | Not null | User query |
| Reply | varchar(45) | Not null | Reply of query |
| Date | datetime | Not null | Date |

**3.2 ARCHITECTURAL DESIGN**

The architectural design develops a modular program structure and represents the control relationships between modules. It also defines interfaces that enable data to flow throughout the program.

**3.2.1** **DATA FLOW DIAGRAM**

A data flow diagram is a graphical technique that depicts data flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A Level 0 DFD also called a fundamental system model or context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in next level i.e., Level 1 DFD. Each of the processes represented at level 1 are sub functions of overall system depicted in the context model.

**Data flow diagram symbol:**

Source/Destination of Data

Data flow

Process

Storage

**Level 0 DFD**

User details

username, password

**user**

User

health\_ enquiry

**Level 1 DFD**

Username& password

login

Administrator

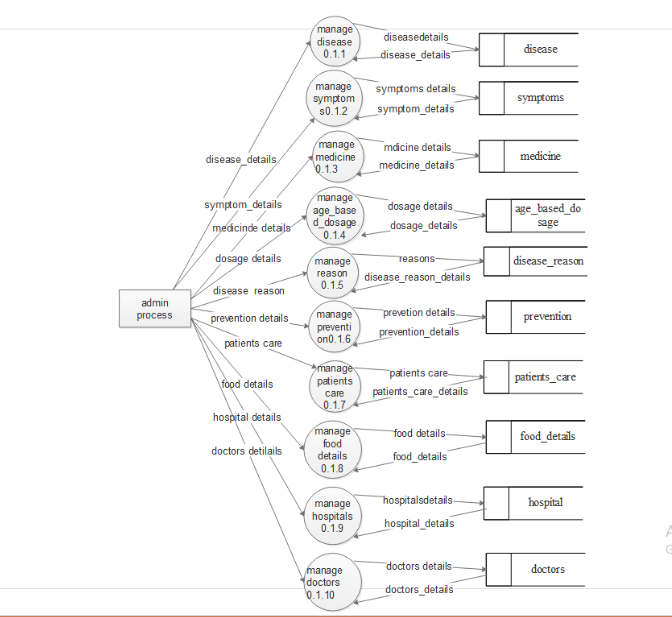
health\_ enquiry

login

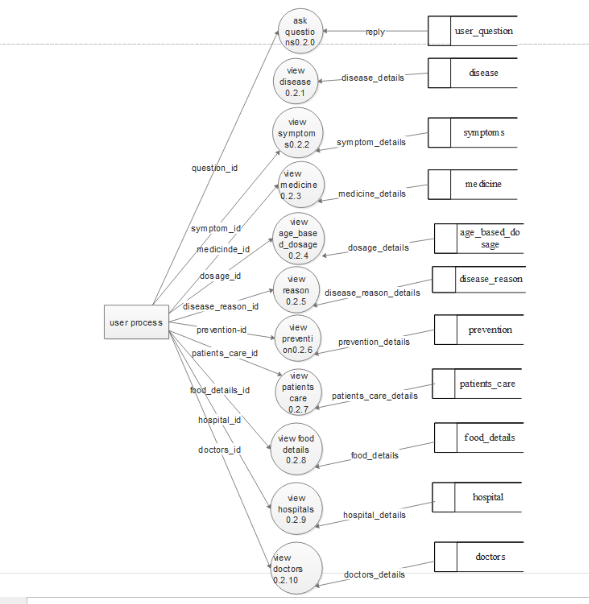
Username& password

User

**Level 2-admin**

****

**Level 2 user**

****

**3.2.2** **HIERARCHICAL DIAGRAM**

The hierarchical diagram is a technique for representing the modules of a system as a hierarchy and for documenting each module. It was used to develop requirements, construct the design, and support implementation of an expert system to demonstrate and verify the system. Structure charts can be used to display several types of information.

Smart health enquiry

View disease

Login

Login

User

Administrator

Ask questions

Add/remove/update disease disease

View hospitals

View symptoms

Add/remove/update medicine

View reasons

Add/remove/update age based dosage

View prevention

Add/remove/update reasons

View patients care

Add/remove/update preventions

View medicines

search

reply

View questions

Add/remove/update patients care

View age based dosage

Add/remove/update hospitals

View hospitals

Add/remove/update doctors

View doctors list

**3.3 INTERFACE DESIGN**

An interface design elements for the software tell how information flows into and out of the system and how it is communicated among the components as part of the architecture.

**3.3.1** **INPUT DESIGN**

Input design is the link between the information system and users and those steps that are necessary to put transaction data into a usable form for processing data entry. Instructing the computer to read data from a written printed document can active the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the errors, avoid delay, and keeping the process simple. System analyst decides the following input design details.

* What data to input?
* What medium to use?
* How the data is arranged and coded?

In my project named Smart health enquiry, I tried to include the following design constrains provided in the software engineering.

**1: Avoid scattering of fields in the forms**

In all forms of the software the textboxes (which provided to input some data), label (which label the text boxes), combo box (list a set of values) etc all are arranged in a neat and well format. It provides a simple look to the pages. The buttons are placed at the bottom of the page and easily accessible to the user. The menus are arranged below the heading and at a minimum level of menus are arranged with pages. Menu provides the continuity to the pages.

**2: User only needs to enter a minimum amount of data**

All forms contain a minimum amount data, but most essentials. No page provides or wanted bulky of data. It provides more easiness to the user. It creates more the software to the end user. Also the operation continues by single click.

**3: Avoid confusion in the forms**

All forms have a well defined menus and each menu name indicate its purpose. So the user can easily access various forms without confusion. Each form and its sub forms are well labeled. So the user can easily identify the forms and work on that.

**The following are the input forms present in this project:**

* Login form
* Registration form
* Add age based medical dosage
* Add symptoms
* Add disease form
* Add hospital form
* Add doctors form
* Add medicine form
* Add prevention form
* Add reason form
* Add patients care form

**3.3.2** **OUTPUT DESIGN**

Designing computer should proceed in well thought out manner. The term output means any information produced by the information system weather printed or displayed. Output design is a process that involves designing necessary output that have to be used by various users according to requirement. The efficient intelligent output design should remove the system relationship with the users and help in decision making.

When designing the output, system analyst must accomplish the following:

* Determine the information present
* Decide whether to print, display the information and select output medium
* Arrange information in acceptable format.

In my project, the outputs are in the form of reports. They are well format and it provides the output in a correct and neat format.

**The following are the output forms present in this project:**

* Form for viewing disease details
* Form for viewing symptoms details
* Form for viewing doctors details
* Form for viewing hospital details

**3.4 PROCEDURAL DESIGN**

The procedural design determines the modules included in the whole project which help us to identify the major functions.

**MODULE SPECIFICATIONS**

The following are the modules in this application.

1. Disease
2. Symptoms
3. Medicine and care
4. Hospital information

**Disease Module**

This module handles all the information about disease:

* Add details: Add disease details, add reason for disease details.
* View details : view disease details, view reason for disease details
* Update details: update disease details, update reason for disease details
* Delete details: delete disease details, delete reason for disease details

**2. Symptoms module**

* Add details: add symptoms details.
* View details: view symptoms
* Update details: update symptoms
* Delete details: delete symptoms

**3. Medicine and care**

* Add details: add medicine details, add age based dosage details, add side effects of medicine, add prevention methods, and add patients care details.
* View details: view medicine details, view age based dosage details, view side effects of medicine, view prevention methods, view patients care details
* Update details: update medicine details, update age based dosage details, update side effects of medicine, update prevention methods, update patients care details
* Delete details: delete medicine details, delete age based dosage details, delete side effects of medicine, delete prevention methods, delete patients care details

**4. Hospital Module**

* Add details: add hospitals, add doctor’s details, and add consulting time.
* View details: view hospitals, view doctor’s details, and view consulting time.
* Update details: update hospitals, update doctor’s details, and update consulting time.
* Delete details: delete hospitals, delete doctor’s details, and delete consulting time