**+-`**

**Synopsis**

**MCSP-060**

**e-MEDICAL INFROMATION MANAGEMENT SYSTEM**

**(e-MIMS)**

**SUBMITTED BY**

**NAME: ARJUN KUMAR**

**ENROLMENT NO: 175596952**

**Study center code: 2702**

**Under Guidance**

**of**

**Mr. Amresh Kumar Amar**

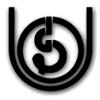
**Submitted to the School of Computer and Information Sciences, IGNOU**

**in partial fulfilment of the requirements**

**for the award of the degree**

**Master of Computer Applications (MCA)**

**YEAR OF SUBMISSION: 2022**



**Indira Gandhi National Open University**

**Maidan Garhi**

**New Delhi – 110068.**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | *Contents* | *Page No.* |
|  | *Title of the Project* | *3* |
|  | *Introduction and Objective* | *4* |
|  | *Project Category* | *6* |
|  | *Tools / Platform, Hardware & Software requirement* | *7* |
|  | *Problem Definition* | *9* |
|  | *Requirements Specification:* | *11* |
|  | *Functional Requirements* | *11* |
|  | *Technical Requirements* | *12* |
|  | *Project Planning and Scheduling* | *14* |
|  | *Scope of the solution* | *15* |
|  | *Data Flow Diagram* | *16* |
|  | *ER Diagram* | *21* |
|  | *Module Description* | *22* |
|  | *Data Structure Or Data Modeling* | *24* |
|  | *Process Logic & Implementation Methodology* | *34* |
|  | *Report Generation* | *35* |
|  | *Overall network architecture* | *36* |
|  | *Security and Validations Check* | *37* |
|  | *Future Scope of the Project* | *38* |
|  | *Bibliography* | *39* |

**TITLE OF THE PROJECT**

**e-MEDICAL INFORMATION MANAGEMENT SYSTEM**

**INTRODUCTION AND OBJECTIVE**

This project “e-Medical Information Management System” is a web based organized computerized system designed and programmed to deal with day to day operations and management of the hospital activities. The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs. It also maintains hospital information such as ward id, doctors in charge and department administering.

This project includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id. The e-MIMS can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

This project is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. It is designed to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

This project is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes

This project modulates protocols and techniques to suit the requirement. Presently the whole functioning of Hospital Management is manual, which is very time consuming and inefficient. This project aims at eliminating all these manual works, which will be replaced by e-form based fully computerized system requiring very less manpower and adding transparency to the system. This gives a flexible way to perform all the task efficiently and at the same time assuring the following:-

* Time Efficient.
* Cost Effective.
* Paperless Office.
* Fast data access.
* No duplication of job.
* Fully integrated system.
* User friendly environment.
* Data security and reliability.
* Added Accountability for staffs.
* Added transparency to the system.

**PROJECT CATEGORY**

**RELATIONAL DATA BASE MANAGEMENT SYSTEM (RDBMS)**

The RDBMS model is based on the structure of a database. A database is simply a collection of one or more relations or tables with columns and rows. The use of set theory allows for data to be structured in a series of tables that has both columns and rows. Each column corresponds to an attribute of that relation, while each row corresponds to a record that contains data values for an entity.

The main elements of RDBMS are based on Ted Codd’s 13 rules for a relational system, the concept of relational integrity, and normalization. The three fundamentals of a relational database are that all information must be held in the form of a table, where all data are described using data values. The second fundamental is that each value found in the table columns does not repeat. The final fundamental is the use of Standard Query Language (SQL).

RDBMS makes the system is simple, flexible, and productive. Because the tables are simple, data is easier to understand and communicate with others. RDBMS are flexible because users do not have to use predefined keys to input information. Also, RDBMS are more productive because SQL is easier to learn. This allows users to spend more time inputting instead of learning. More importantly, RDBMS’s biggest advantage is the ease with which users can create and access data and extend it if needed. After the original database is created, new data categories can be added without the existing application being changed.

**The benefits of this approach are:**

1**. Reduce redundancy of information**:- Data’s are being made simple and complex, we don’t have to input as many data’s which can only result in overloaded space in computer hard disk/ memory.

2**. Consistent data flow**:- Once the data’s are being analyzed in sequence, once can easily identify the flow of information and hence, results will come in reliable output.

3**. Integration of data**:- Incorporating of data’s in tables should result in a highly assimilation of information.

4. **Security and User privileges**:- Protection of data’s from unwanted users and giving user’s rights to what level they are to use the application.

5. **Ease of application development**:- It is a more comfortable workspace for the relevance of its factual purpose.

**TOOLS / PLATFORM, HARDWARE & SOFTWARE REQUIREMENT:**

To access this web application on client side, it only needs a PC/Laptop/Mobile with an integrated and updated web browser and having Local area network connection.

**Desktop Browser** : Google Chrome, Mozilla Firefox, Opera, Safari, IE, etc.

**Mobile Browser** : Android, Chrome, iOS Safari, etc.

On the server side, a PC with minimum these specifications-

**HARDWARE:**

|  |  |  |
| --- | --- | --- |
| Processor | : | Intel core 2 duo 2.20 GHZ or above |
| Memory | : | 2GB or above |
| Cache memory | : | 128 KB or above |
| Hard Disk | : | 500 GB |
| Others | : | Keyboard, Mouse, Monitor, redundant power supply |

**SOFTWARE:**

|  |  |  |
| --- | --- | --- |
| Operating System | : | Windows, Linux, Mac |
| Front-End Tool | : | HTML 5, PHP 8, CSS3, Java script, jQuery, Ajax |
| Back End Tool | : | MySQL 5.0 or above |
| Server | : | XAMPP version 8.0 or above |
| Web Browser | : | Chrome/ Safari/ Internet Explorer/ Edge |

**MySQL 5.X**

MySQL is the world’s most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications. Free-software-open source projects that require a full-featured database management system often use MySQL.

The new capabilities introduced in MySQL 5.7 are packaged to customers in different ways: as part of the base product, as new features are added and limitation are improved. A comprehensive list of new capabilities is available in the MySQL 5.7 documentation.

MySQL Database’s Management console : phpmyadmin is free and open source administration tool for MYSQL and MariaDB. As a portable web application written primarily in PHP, it has become one of the most popular MySQL administration tools, especially for web hosting services.

**Scripting Language: PHP**

PHP is a dynamic and interactive server-side, object-oriented, open source scripting language. PHP is a language used to develop interactive and dynamic content on the web and it is often used together with the Apache web server. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

**HTML**

Hyper Text Markup Language and Extensible markup Language are the predominant markup languages for web pages.It provides a means to describe the structure of text-based information in a document and to supplement that text with interactive forms, embedded images, and other objects

**CASCADING STYLE SHEETS** (**CSS**):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

**PROBLEM DEFINITION:**

Presently the Hospital Management of the Hospital works as follows:

* There is manual system for Hospital Management.
* Patient has to report in reception for appointment where receptionist will make entry in register for OPD.
* Hospital staff will also maintain many register for patient, Doctor, Nurse and staff Management for different department.
* Doctor will prescribe medicine to OPD patient and admit patient as IPD patient in hospital for further investigation.
* Nurse will care of IPD patients and allot the room of wards and keep maintain records of individual patient till discharge.
* After patient discharge all patient records are transferred to record room for future references.
* Maintaining records of various level at different departments of hospital are cumbersome task.

The following difficulties are found in existing system.

**Lack of immediate retrievals: -**

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in convenience and wastage of time.

**Lack of immediate information storage: -**

The information generated by various transactions takes time and efforts to be stored at right place.

**Lack of prompt updating: -**

Various changes to information like patient details are difficult to make as paper work is involved.

**Error prone manual calculation: -**

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments.

Hence using “e-Medical Information Management System” web application will digitize the problem area. As a result, The availability of right type of system plays an important role in saving the time of many individual every day Which works as follows:

* Hospital addresses the problems of its record management using an e-Medical Information Management system.
* The system allows receptionist to register patient in hospital records and make their appointment with concern doctor online.
* Doctor has facility to treat OPD as well as IPD and suggest prescription and update record in online respective module.
* Nurse has facility to record medical information of IPD patients and update records in online respective module.
* Hospital staff will make billing of discharged patient and doctor fees and provides various discount to different kinds of patients.
* The report module will generate of various types of reports based on the data feed by various staff.
* The blood bank management module will provide prompt requirement of the blood to needy patients.
* The chatbot will be based on intelligence system which answer of users question based on the reply feed by admin in the system which will provide beneficial information to users.

**REQUIREMENTS SPECIFICATIONS**

**FUNCTIONAL REQUIREMENTS:**

Presently the whole functioning of e-Medical Information Management System is manual, which is very time consuming and inefficient. Bulky files and a lot of old-fashioned documents are kept to maintain records. Lots of queries also keep arising that needs cumbersome searching from these bulky records. The following functions will be performed by the system:

* There is user role based accessed in e-MIMS. Hospital Admin has full privilege to control all activities. They can create login ID and Password for receptionist, Doctor, Nurse and billing and report generation staff. They can control all other module and update and add department details.
* Patient will report to receptionist in hospital and the receptionist will register the patient details and patient will get the unique ID which can be access by the whole system for future reference.
* Doctor will get access by his individual login and will treat the OPD and IPD patient and access patient details by his unique ID and prescribe medicine and also admit OPD patient as IPD patient as per his condition.
* Nurse will get access by his individual login will governs all patient room of the hospital and allot room to admit in patient and keep records of medical information of patient.
* Hospital staff will make billing of discharge patient, update doctor fees and discount for different patient in billing.
* Report generation module will generate all types of reports like daily billing, patient’s record.
* The blood bank management system will be access by the admin and it will keep records of all the donors and blood available in blood bank which will be provide available to the needy patient.
* The chatbot will interact with the user using message box. Admin has all control for the Chatbot giving answer to unanswered questions.
* The whole process is to be made paperless. Thus saving lots of papers and contributing towards the cleaning of the environment.

**TECHNICAL SPECIFICATIONS**

The software will require following modules to be developed:

|  |  |  |
| --- | --- | --- |
| Reception Module | : | It caters the initial registration of patients and make appointment with doctors. |
| Doctor Module | : | It caters for prescription of medicine for OPD and IPD patients. |
| Nurse Module | : | It provides the facility to the nurse to patient medication and its input/output records, patient’s history. |
| Staff Module | : | It provide the facility to hospital staff for generating bill of patient, print invoice, print receipt. |
| Report Module | : | It provides report of patient masterlist, individual patient, output patient, admit and discharge patient. It also provide daily sales and doctor fees report. |
| Blood Bank Module | : | It provides the blood bank management for needy patient it is controlled by the admin. |
| Chatbot Module | : | It interacts with user for informative purpose only. All reply are controlled by the admin. |
| Admin Module | : | It has overall control over the all modules and its has additional features for hospital management. |

The following Database Tables may be used to capture data of the project:

|  |  |  |
| --- | --- | --- |
| USERS | : | It stores records of hospital management authority. |
| USERS\_ROLE | : | It stores roles of users and provides role based access. |
| COMPANY\_INFO | : | It stores hospital information. |
| DEPARTMENT | : | It stores department information. |
| DESIGNATION | : | It stores designation information. |
| PATIENT\_PERSONAL\_INFO | : | It stores data of patient basic information and provide unique ID for future reference. |
| PATIENT\_APPOINTMENT | : | It stores data of patient appointment with doctor in hospital with date and time. |
| PATIENT\_DETAILS\_IOP | : | It stores data of admitted patient in the hospital |
| MEDICINE\_CATEGORY | : | It stores data of medicine category. |
| MEDICINE\_DRUG\_NAME | : | It stores data of medicine drug name with other details. |
| DIAGNOSIS | : | It stores data of diagnosis information. |
| FLOOR | : | It stores data of floor information. |
| ROOM MASTER | : | It stores data of room information. |
| ROOM CATEGORY | : | It stores data of room category. |
| ROOM BEDS | : | It stores data of Beds available in particular room. |
| DOCTOR\_FEE | : | It stores data of Doctor fees. |
| BILL\_GROUP\_NAME | : | It stores data of bill group name. |
| BILL\_PARTICULAR | : | It stores data of particular patient bill. |
| PATIENT\_DETAILS\_IOP | : | It keeps records of admitted patient details. |
| IOP\_BILLING | : | It stores of discharged patient biiling details. |
| IOP\_BED\_SIDE\_PROCEDURE | : | It stores data of bed side procedure feed by nurse |
| IOP\_COMPLAINTS | : | It stores data of admitted patient complaint |
| IOP\_DIAGNOSIS | : | It stores data of admitted patient diagnosis. |
| IOP\_DISCHARGE\_SUMMARY | : | It stores data of patient discharge summary. |
| IOP\_INTAKE\_RECORD | : | It stores data of patient intake records. |
| IOP\_MEDICATION | : | It stores data of patient medication. |
| IOP\_LABORATORY | : | It stores data of patient laboratory test. |
| IOP\_NURSE\_NOTES | : | It stores data of nurse notes for patient condition. |
| IOP\_OPERATION\_THEATER | : | It stores data of Operation Theater available. |
| IOP\_OUTPUT\_RECORDS | : | It stores data of output patient records. |
| IOP\_PROGRESS\_NOTE | : | It stores data of patient progress note. |
| IOP\_ROOM\_TRANSFER | : | It stores data of patient room transfer. |
| IOP\_VITAL\_PARAMETERS | : | It stores data of patient vital parameters. |
| BB\_BLOOD\_INVENTORY | : | It stores data of blood bank inventory |
| BB\_DONORS | : | It stores list of donors available for Blood donation. |
| BB\_REQUESTS | : | It stores list of users request for blood. |
| BB\_HANDEDOVER\_REQUEST | : | It stores list of users to whom blood was given. |
| CB\_QUESTIONS | : | It stores list of questions asked by the users. |
| CB\_RESPONSES | : | It stores list of answer for each questions. |
| CB\_UNANSWERED | : | It stores list of unanswered questions |

**PROJECT PLANNING AND SCHEDULING:**

The project of “**e-MEDICAL INFORMATION MANAGEMENT SYSTEM”** will be completed in a phased manner. We plan to complete the project into the following few steps:-

1. Requirements Gathering.

2. Design the look and feel of various modules.

3. Design Database tables and other structures.

4. Coding the system.

5. Test the system unit-wise & also the complete Software.

**Gantt chart for the project** (Milestones are displayed using ♦ symbol) **:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Week 1** | | | | **Week 2** | | | | **Week 3** | | | | | **Week 4** | | | | **Week 5** | | | | **Week 6** | | | | **Week 7** | | | | **Week 8** | | | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Gathering**  **Requirements** |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |  | ♦ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Design various**  **modules** |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | ♦ |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Design database** |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  | ♦ |  |  |  |  |  |  |  |  |  |  |  |  | |
| **Coding** |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ♦ |  |  |  |  | |
| **testing** |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ♦ | |

**SCOPE OF THE SOLUTION:**

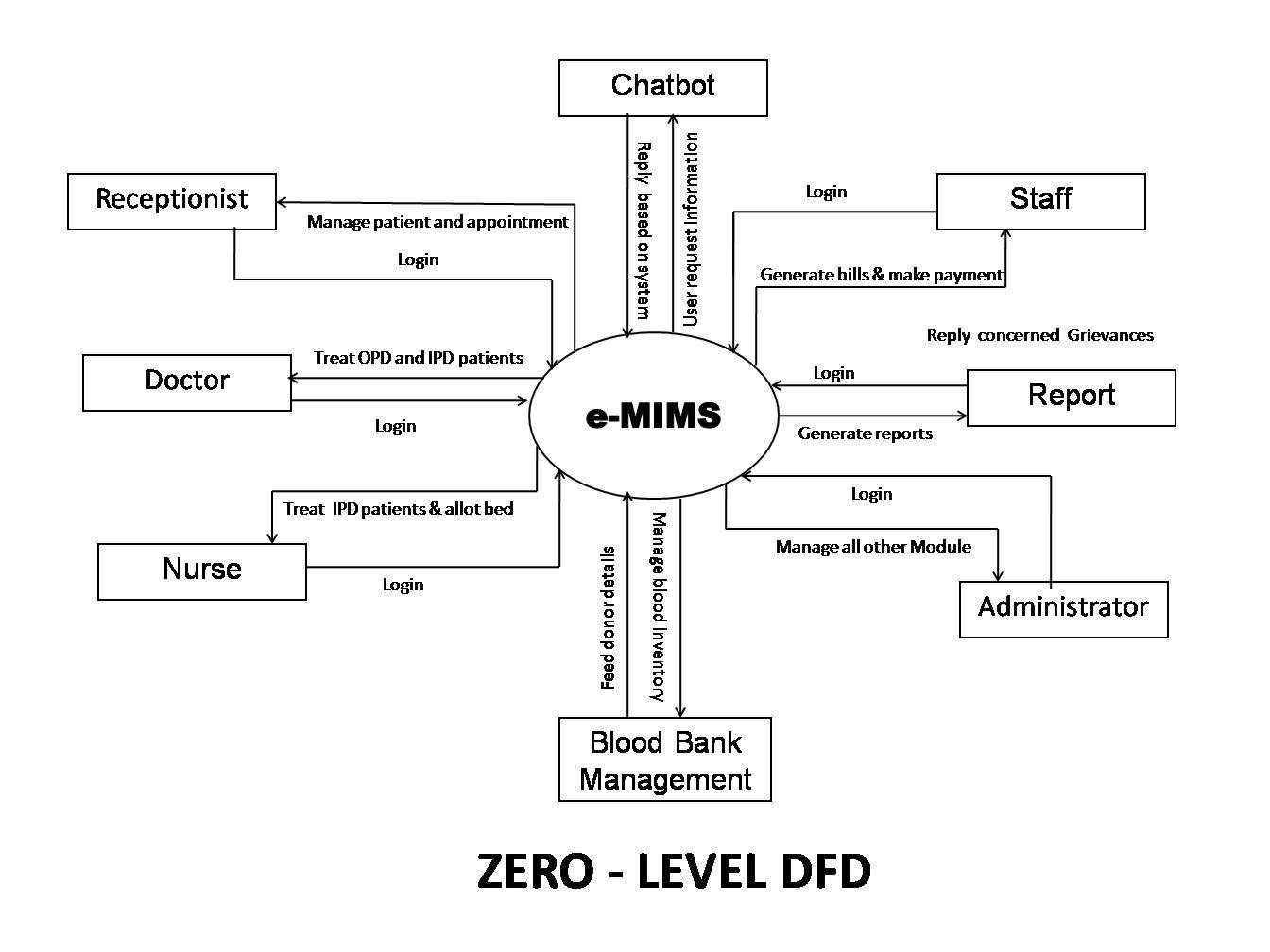
It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Hospital Management . It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

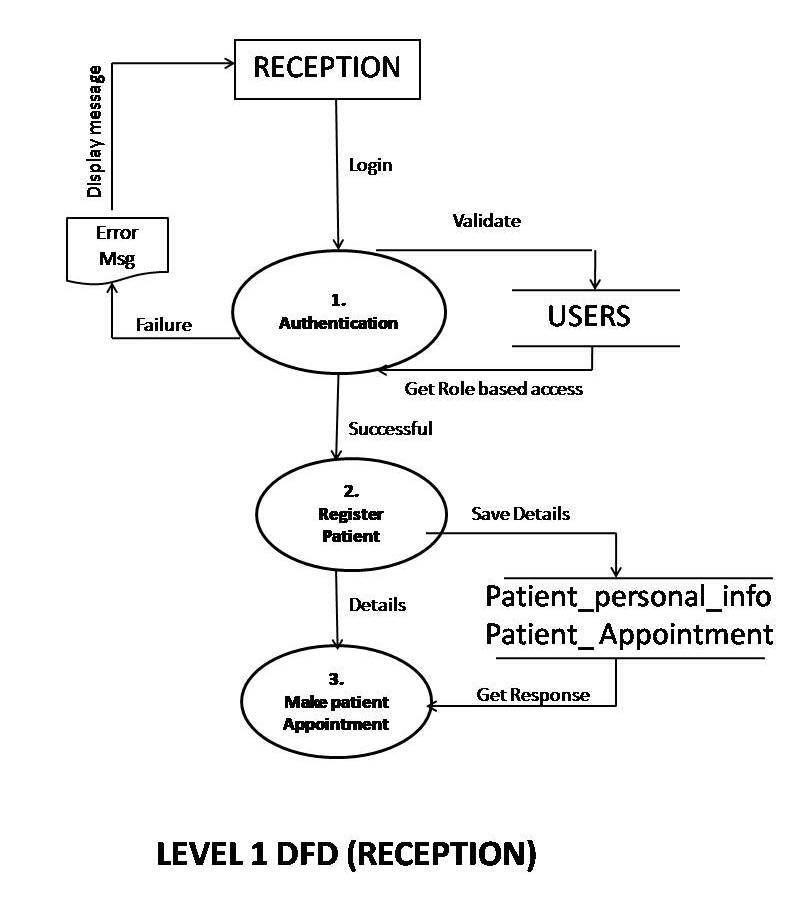
* Our project aims at Business process automation, i.e. we have tried to computerize various processes of Hospital Management.
* In computer system the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
* In computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
* To assist the staff in capturing the effort spent on their respective working areas.
* To utilize resources in an efficient manner by increasing their productivity through automation.
* The system generates types of information that can be used for various purposes.
* It satisfy the user requirement.
* Be easy to understand by the user and operator.
* Be easy to operate
* Have a good user interface.
* Be expandable
* Delivered on schedule within the budget.

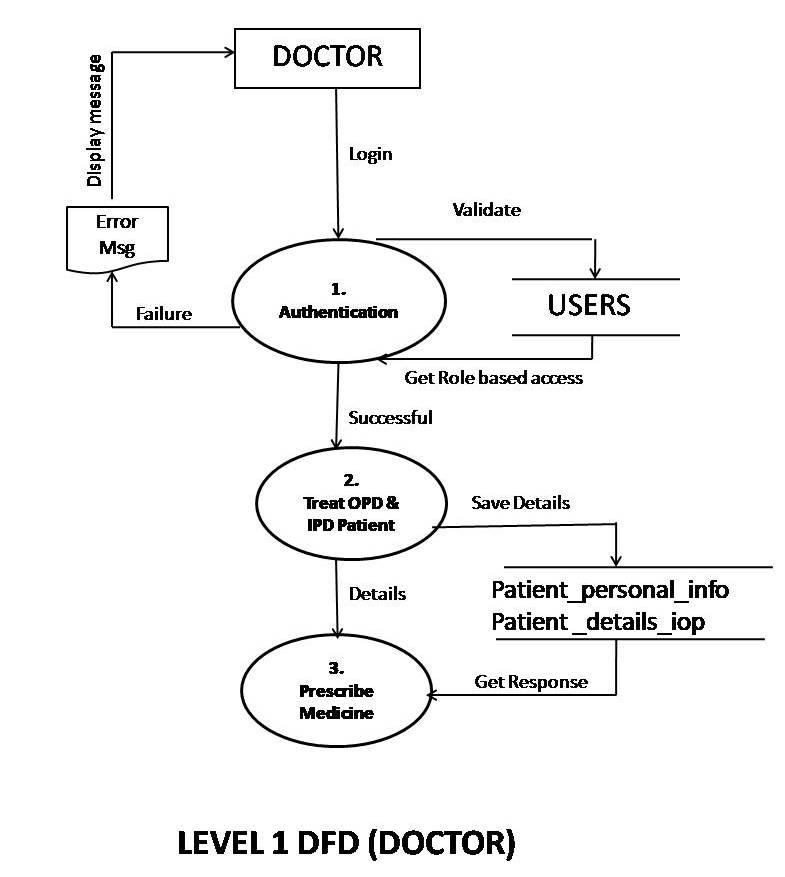
The scope of the project is to fully automate the Hospital management of the Hospital. Since the Hospital is hosting website on intranet for informative purpose. Hence the same hardware specification can be utilise to host the e-MIMS And Hospital staff can use this platform.

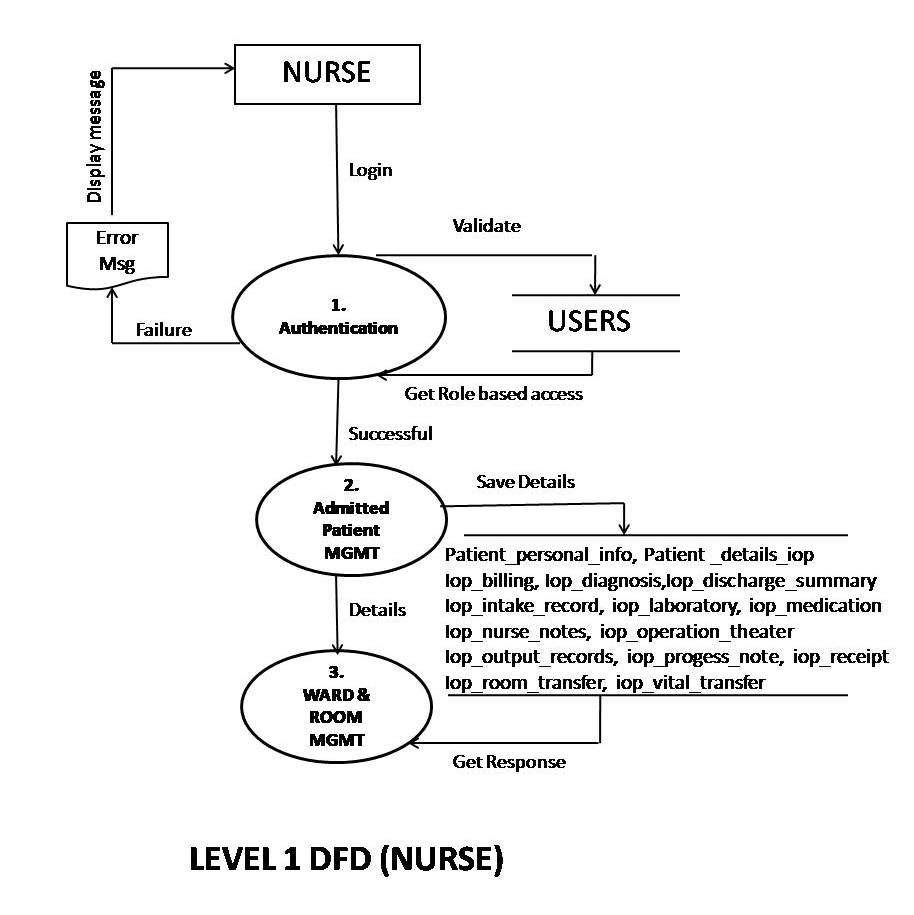
As a result, the availability of the right type of systems plays an important role in saving the time of many individual every day. So having online e-Medical Information Management System. it will save the Hospital staff as well as patient precious time and will be more convenient way to get the best service from the system. This software can be of great help for all of them to tackle difficulties.

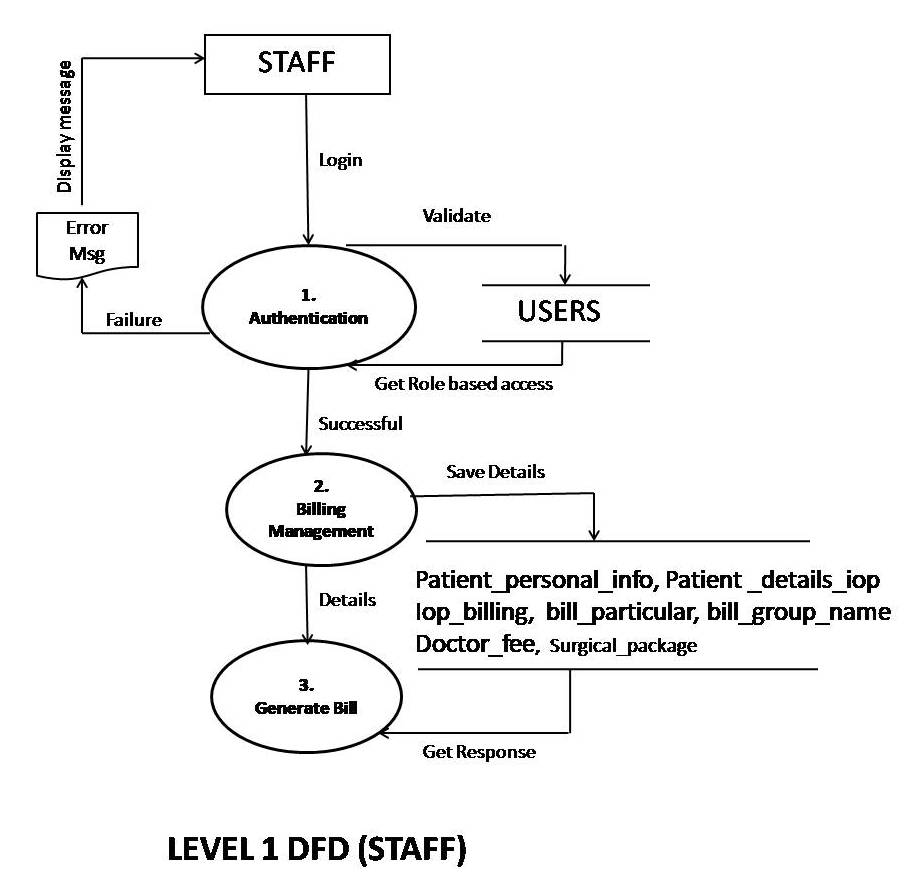
**DATA FLOW DIAGRAM:-**

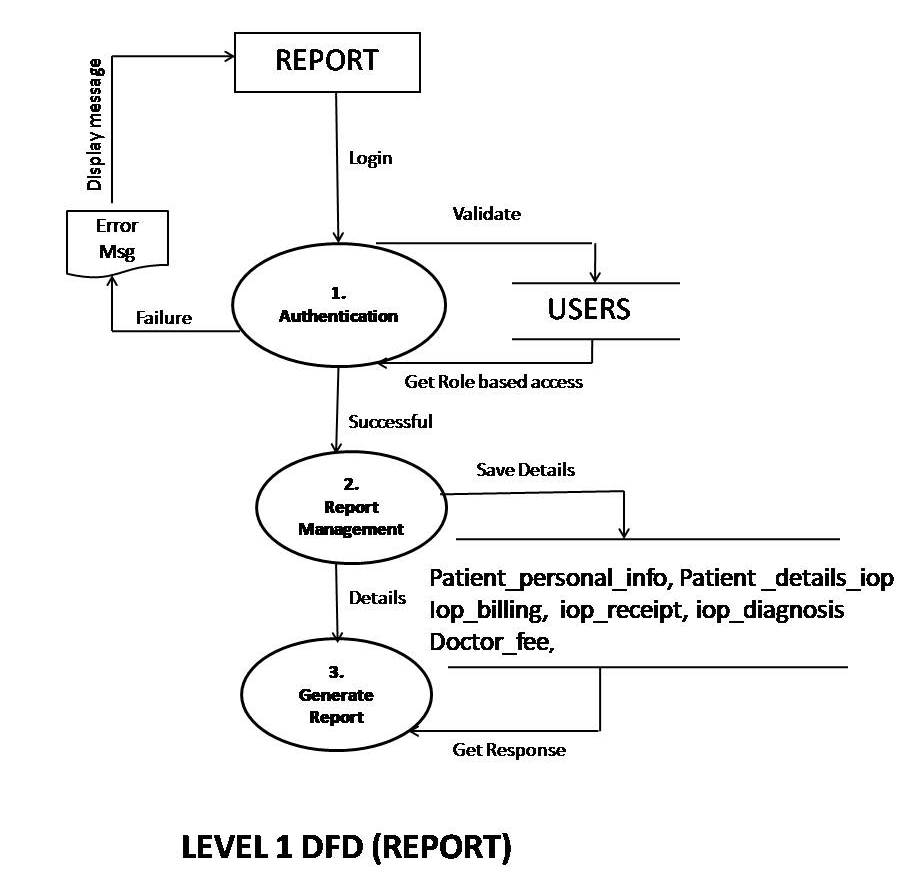


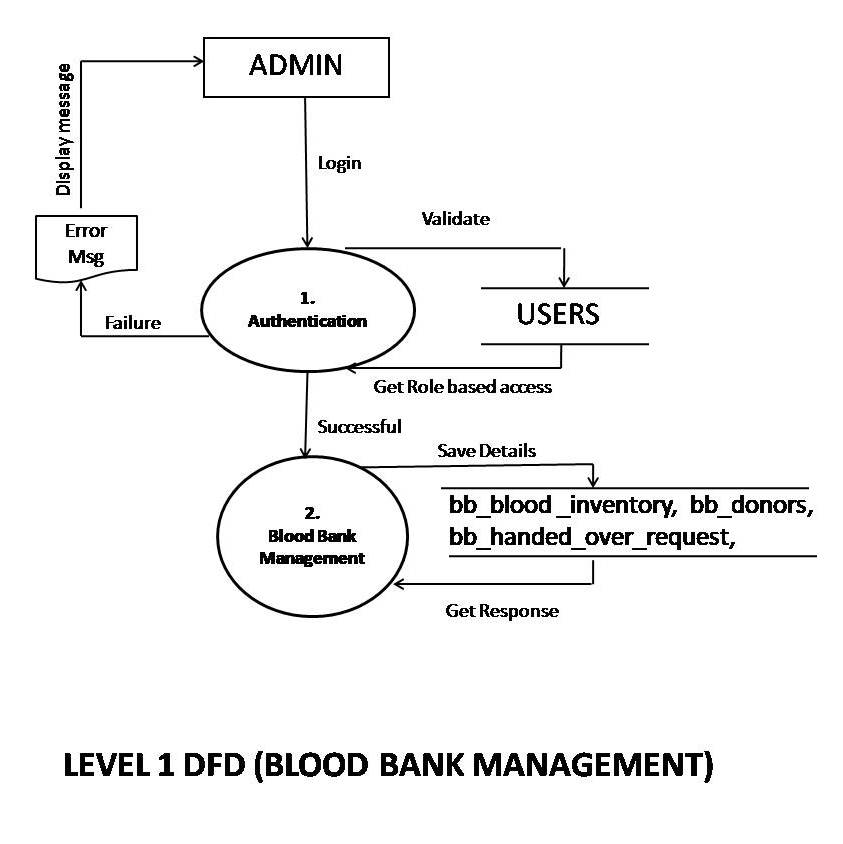


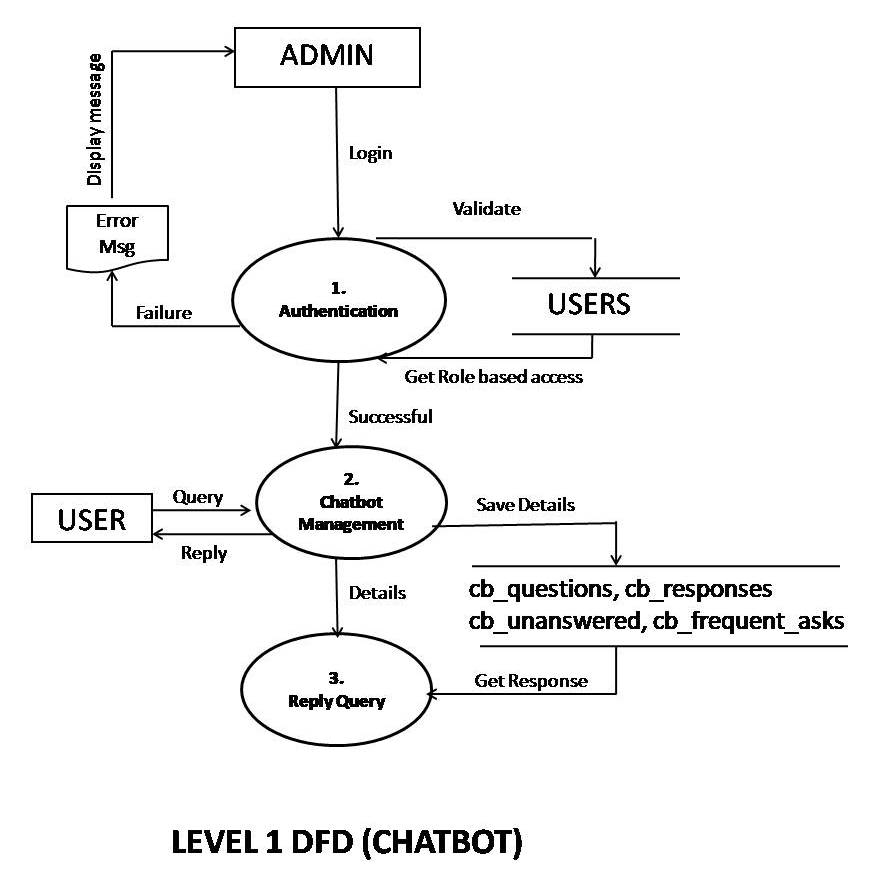


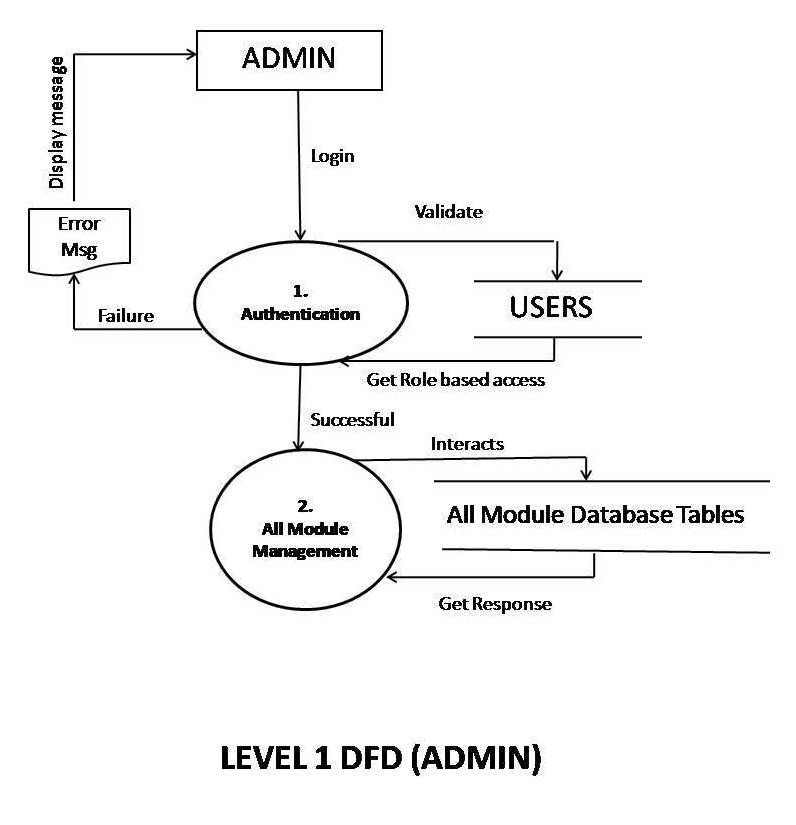




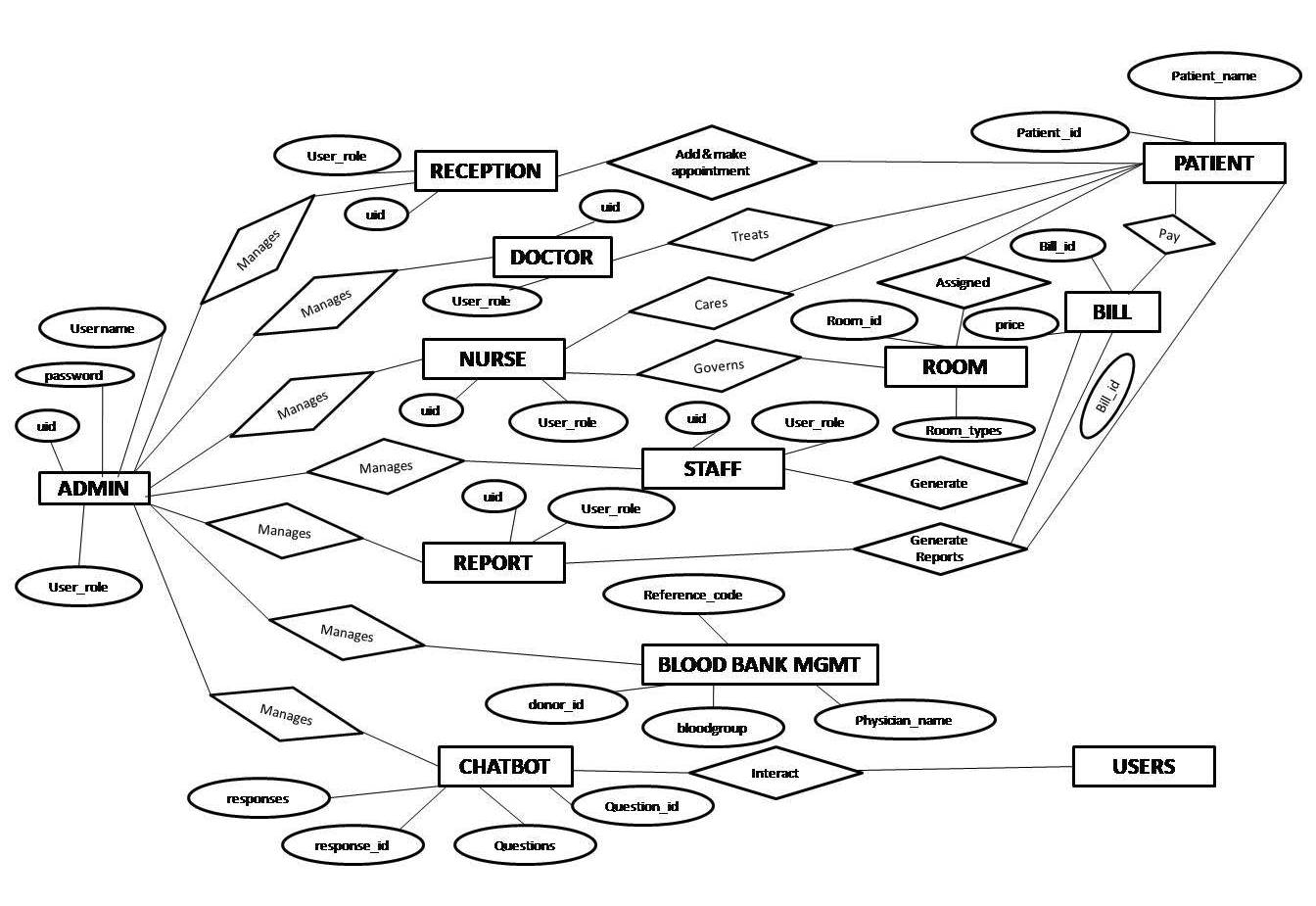








**ER DIAGRAM:-**

****

**VARIOUS MODULES AND THEIR DESCRIPTIONS:-**

1. **Reception:-**

This module caters registration of the patients who wants to visit the Hospital. Basic information are filled up through an online form in this module and on successful submission of the form an unique ID is generated for each patient. Whenever there is any change, the data needs to be updated using edit forms by receptionist in his individual login. This data entry and updation process goes through various validation checks. After registration, patient appointment with doctor can be booked by the receptionist. The receptionist can also book the in patient appointment. The receptionist Login ID and password will be given by the Admin of Hospital after entering his details in new users and selecting roles receptionist and this module having following sub modules.

(i) Dashboard:- show Today’s Patient, New Patient and visited patient

(ii) Patient Appointment:- Add and Manage Patient appointment.

(iii) Patient Management:- Add and Manage Patient of OPD as well as IPD

(iv) User Profile:- update his profile and change the password.

1. **Doctor:-**

The Doctor Module is having the role based access. These IDs are created by the Admin of The Hospital. This module provide the doctor to login the web application by providing the user ID and Password and prescribe medicine to OPD and IPD Patient and this module having following sub modules.

(i) Dashboard:- show Today’s Patient, New Patient and visited patient

(ii) Doctor Module:- view Out and In patient of the Hospital and prescribe medicine.

(iii) User Profile:- update his profile and change the password.

1. **Nurse:-**

The Nurse Module is having the role based access. These IDs are created by the Admin of The Hospital. This module provide the Nurse to login the web application by providing the user ID and Password and allow ward management and record in-patient medical records and intake records And this module having following sub modules.

(i) Dashboard:- show Today’s Patient, New Patient and visited patient

(ii) Nurse Module:- ward management and record In patient of the Hospital medical and intakes record

(iii) User Profile:- update his profile and change the password.

**(d.) Staff:-**

Hospital staff are generally used for the admin work i.e: Billing, sanitization. In E-MIMS used the Staff for billing purpose only. Login ID and password will be created by admin and passed to concerned staff confidentially. After Login in staff module, staff name will be displayed on right top corner and. It is having following sub modules.

(i) Dashboard:- show Today’s Patient, New Patient and visited patient

(ii) Billing:- Billing of discharged patient, Surgical quotation.

(iii) User Profile:- update his profile and change the password.

**(e.) Report:-**

This module has the interface that allows the hospital statistic staff to generate report of patient master list, individual patient record, out patient report, admitted patient report, Discharged patient report, Daily sales report, Docter’s fee report and acknowledge receipt. This module is having following sub modules.

(i) Dashboard:- show Today’s Patient, New Patient and visited patient

(ii) Report Generation:- Generation of various reports as mentioned above.

(iii) User Profile:- update his profile and change the password.

**(f.) Blood bank Management:-**

This Module interface was given in admin mode which helps the Hospital admin to maintain the data of blood donor and request made by the patient and blood donation records. This module is having following sub modules.

(i) Donor:- records of voluntary person for blood donations

(ii) Blood Donations:- records of blood donated.

(iii) Requests:- Records of blood is needed by the patient.

(iv) Handed Over:- records of blood is handed over to whom

(v) Users:- for Blood Bank User Management

**(g.) Chatbot:-**

This Module interface was given on login page and chatbot control interface was given in admin mode. Chatbot interact with users via chatting box for information based query that is stored in the system by the admin. This control module is having following sub modules.

(i) Dashboard:- Display of frequently asked questions by the user

(ii) setting:- Change the Chatbot name and Welcome message of the Chatbot.

(iii) Responses List:- Responses given to question asked by the users.

(iv) Unanswered List:- Unanswered questions which are not replied by Chatbot.

**(h.) Admin:-**

This module has the interface that allows the admin to search, add, update and delete various records like patient management, staff management, appointment management. This module can control the reception, doctor, nurse, staff, report, bloodbank, chatbot module. In addition this module are having the following sub module.

(i) Room Management:- records of the Hospital room for the patients.

(ii) EMR sheet:- records of Out & In Patient Electronic Medical sheet records.

(iii) User Management:- Create role based access for the users.

(iv) Administrator:- Allow admin management of the Hospital.

**DATA STRUCTURE:-**

Following are the Database tables used in this project. The Description of each of the tables is mentioned against them.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **USER TABLE** |  |  |  |
| id | BIGINT(11) | Primary Key | Id |
| User\_id | VARCHAR(15) | Not Null | User\_ID |
| department | INT(11) | Not Null | Department name |
| Designation | INT(11) | Not Null | Designation name |
| title | INT(11) | Not Null | User title |
| lastname | VARCHAR(50) | Not Null | User last name |
| firstname | VARCHAR(50) | Not Null | User first name |
| middlename | VARCHAR(50) | Not Null | User middle name |
| age | INT(2) | Not Null | Age of users |
| Street | VARCHAR(50) | Not Null | Address details |
| Province | CHAR(50) | Not Null | State |
| Phone\_no | VARCHAR(15) | Not Null | Office number |
| Mobile\_no | VARCHAR(15) | Not Null | Mobile number |
| Gender | INT(1) | Not Null | Gender status |
| Civil\_status | INT(1) | Not Null | Married status |
| birthday | DATE | Not Null | Date of birth |
| birthplace | VARCHAR(100) | Not Null | Birthplace name |
| Email\_address | VARCHAR(75) | Not Null | User email Id |
| Username | VARCHAR(25) | Not Null | User id |
| password | VARCHAR(50) | Not Null | User password |
| picture | VARCHAR(100) | Not Null | Pic information |
| doctorIsIn | VARCHAR(10) | Not Null | Date & Time |
| doctorLastIn | VARCHAR(25) | Not Null | Date & Time |
| doctorLastOut | VARCHAR(25) | Not Null | Date & Time |
| InActive | INT(1) | Not Null | 1 for inactive  0 for active |
|  |  |  |  |
| **USER \_ROLE TABLE** |  |  |  |
| Role\_id | INT(11) | Primary Key | Role Id |
| module | VARCHAR(50) | Not Null | Module Name |
| Role\_name | VARCHAR(50) | Foreign Key,Not Null | Role based access |
| Role\_description | TEXT | Not Null | Role description |
| InActive | INT(1) | Primary Key | 1 for inactive  0 for active |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **COMPANY\_INFO TABLE** |  |  |  |
| Company\_name | VARCHAR(150) | NOT NULL | HOSPITAL NAME |
| Company\_address | TEXT | NOT NULL | HOSPITAL ADDRESS |
| Company\_contactNO | VARCHAR(50) | NOT NULL | HOSPITAL CONTACT |
| TIN | VARCHAR(50) | NOT NULL | HOSPITAL TIN |
| logo | VARCHAR(900) | NOT NULL | HOSPITAL LOGO |
|  |  |  |  |
| **DEPARTMENT TABLE** |  |  |  |
| Department\_id | INT(11) | PRIMARY KEY | DEPARTMENT ID |
| Dept\_code | VARCHAR(100) | FORIGEN KEY | DEPARTMENT CODE |
| Dept\_name | VARCHAR(150) | NOT NULL | DEPARTMENT NAME |
| InActive | INT(1) | NOT NULL | STATUS |
|  |  |  |  |
| **DEPARTMENT TABLE** |  |  |  |
| Designation\_id | INT(11) | PRIMARY KEY | DESIGNATION ID |
| Designation | VARCHAR(50) | NOT NULL | DESIGNATION NAME |
| Description | VARCHAR(150) | NOT NULL | DESIGNATION DESC |
| InActive | INT(1) | NOT NULL | STATUS |
|  |  |  |  |
| **PATIENT\_PERSONAL\_INFO** |  |  |  |
| Reg\_no | BIGINT(11) | PRIMARY KEY | REGISTRATION NO |
| Patient\_no | VARCHAR(15) | FORIEGN KEY | PATIENT ID |
| Title | INT(11) | NOT NULL | TITLE |
| Lastname | VARCHAR(35) | NOT NULL | LAST NAME |
| Firstname | VARCHAR(35) | NOT NULL | FIRST NAME |
| Middlename | VARCHAR(35) | NOT NULL | MIDDLE NAME |
| Gender | INT(11) | NOT NULL | GENDER |
| Civil\_status | INT(11) | NOT NULL | MARRIED |
| Birthday | DATE | NOT NULL | DATE OF BIRTH |
| Birthplace | VARCHAR(150) | NOT NULL | BIRTH PLACE |
| Fathers\_name | VARCHAR(150) | NOT NULL | FATHERNAME |
| Address1 | TEXT | NOT NULL | ADDRES LINE |
| Address2 | TEXT | NOT NULL | ADDRESS LINE |
| Age | INT(3) | NOT NULL | AGE |
| Religion | INT(11) | NOT NULL | RELIGION |
| Street | VARCHAR(50) | NOT NULL | STREET |
| Province | VARCHAR(90) | NOT NULL | STATE |
| Phone\_no | VARCHAR(25) | NOT NULL | OFFICE NO |
| Mobile\_no | VARCHAR(25) | NOT NULL | MOBILE NO |
| Email\_address | VARCHAR(50) | NOT NULL | EMAIL ADDRESS |
| Picture | VARCHAR(100) | NOT NULL | PICTURE |
| Date\_entry | DATETIME | NOT NULL | DATE ENTRY |
| Blood\_group | INT(11) | NOT NULL | BLOOD GROUP |
| Insurance\_comp | INT(11) | NOT NULL | INSURANCE COMP |
| Insurance\_no | VARCHAR(25) | FORIEGN KEY | INSURANCE NO |
| InActive | INT(1) | NOT NULL | STATUS |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **PATIENT\_APPOINTMENT** |  |  |  |
| appID | BIGINT(20) | PRIMARY KEY | APPLICATION ID |
| Patient\_no | VARCHAR(25) | FORIEGN KEY | PATIENT NO |
| appointmentDate | DATE | NOT NULL | APPOINTMENT |
| appHour | TINYINT(2) | NOT NULL | APPOINT HOUR |
| appMinutes | TINYINT(2) | NOT NULL | APPOINT MINT |
| appAMPM | VARCHAR(10) | NOT NULL |  |
| appointmentTime | TIME | NOT NULL | APPOINT TIME |
| AppointmentReason | VARCHAR(10) | NOT NULL | APPOINT REASON |
| consultantDoctor | VARCHAR(10) | NOT NULL | DOCTOR NAME |
| dateVisit | DATETIME | NOT NULL | DATE VISIT |
| appointmentStatus | CHAR(1) | NOT NULL | APPOINT STATUS |
| dateEntry | DATETIME | NOT NULL | DATE ENTRY |
|  |  |  |  |
| **PATIENT\_DETAILS\_IOP** |  |  |  |
| Id | BIGINT(11) | PRIMARY KEY | ID |
| IO\_ID | VARCHAR(15) | FOREIGN KEY | IPD PATIENT |
| Patient\_no | VARCHAR(15) | FOREIGN KEY | PATIENT\_NO |
| Patient\_type | VARCHAR(5) | NOT NULL | PATIENT\_TYPE |
| Date\_visit | DATE | NOT NULL | DATE\_VISIT |
| Time\_visit | TIME | NOT NULL | TIME\_VISIT |
| Doctor\_id | VARCHAR(15) | FOREIGN KEY | DOCTOR\_ID |
| Referral\_doctor | INT(11) | NOT NULL | REFERRAL\_DOCTOR |
| Room\_id | INT(11) | FOREIGN KEY | ROOM\_ID |
| Department\_id | INT(11) | FOREIGN KEY | DEPARTMENT\_ID |
| Provisional\_diagnosis | TEXT | NOT NULL | PROV DIAGNOSIS |
| Complaints | TEXT | NOT NULL | COMPLAINTS |
| Allergies | TEXT | NOT NULL | ALLERGIES |
| Warnings | TEXT | NOT NULL | WARNINGS |
| Social\_history | TEXT | NOT NULL | SOCIAL\_HISTORY |
| Family\_history | TEXT | NOT NULL | FAMILY\_HISTORY |
| Personal\_history | TEXT | NOT NULL | PERSONAL\_HISTORY |
| Past\_medical\_history | TEXT | NOT NULL | PAST\_MED\_HISTORY |
| Pulse\_rate | VARCHAR(150) | NOT NULL | PULSE\_RATE |
| Temperature | VARCHAR(150) | NOT NULL | TEMPERATURE |
| Height | VARCHAR(150) | NOT NULL | HEIGHT |
| Bp | VARCHAR(150) | NOT NULL | BP |
| Respiration | VARCHAR(150) | NOT NULL | RESPIRATION |
| Weight | VARCHAR(150) | NOT NULL | WEIGHT |
| Nstatus | VARCHAR(15) | NOT NULL | NSTATUS |
| isPaid | INT(1) | NOT NULL | ISPAID |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **MEDICINE\_CATEGORY** |  |  |  |
| Cat\_id | INT(11) | PRIMARY KEY | CAT\_ID |
| Med\_category\_name | VARCHAR(500) | NOT NULL | MED\_CATE\_NAME |
| Med\_category\_desc | TEXT | NOT NULL | MED\_CATE\_DESC |
| InActive | INT(11) | NOT NULL | INACTIVE |
|  |  |  |  |
| **MEDICINE\_DRUG\_NAME** |  |  |  |
| Drug\_id | INT(11) | PRIMARY KEY | DRUG\_ID |
| Med\_cat\_id | INT(11) | FOREIGN KEY | MED\_CAT\_ID |
| Drug\_desc | TEXT | NOT NULL | DRUG\_DESC |
| Uom | INT(11) | NOT NULL | UOM |
| Re\_order\_level | FLOAT(11,2) | NOT NULL | RE\_ORDER\_LEVEL |
| nPrice | FLOAT(11,2) | NOT NULL | NPRICE |
| nStock | FLOAT(11,2) | NOT NULL | NSTOCK |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **DIAGNOSIS** |  |  |  |
| Diagnosis\_id | INT(11) | PRIMARY KEY | DIAGNOSIS\_ID |
| Diagnosis\_name | VARCHAR(500) | NOT NULL | DIAGNOSIS\_NAME |
| Diagnosis\_desc | TEXT | NOT NULL | DIAGNOSIS\_DESC |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **FLOOR** |  |  |  |
| Floor\_id | INT(11) | PRIMARY KEY | FLOOR\_ID |
| Floor\_name | VARCHAR(25) | NOT NULL | FLOOR\_NAME |
| Floor\_description | TEXT | NOT NULL | FLOOR\_DESC |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **ROOM\_MASTER** |  |  |  |
| Room\_master\_id | INT(11) | PRIMARY KEY | ROOM\_MASTER\_ID |
| Category\_id | INT(11) | FOREIGN KEY | CATEGORY\_ID |
| Room\_name | VARCHAR(50) | NOT NULL | ROOM\_NAME |
| Floor | INT(11) | NOT NULL | FLOOR |
| Room\_rates | FLOAT(11,2) | NOT NULL | ROOM\_RATES |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **ROOM\_CATEGORY** |  |  |  |
| Category\_id | INT(11) | PRIMARY KEY | CATEGORY\_ID |
| Category\_name | VARCHAR(50) | NOT NULL | CATEGORY\_NAME |
| Category\_desc | TEXT | NOT NULL | CATEGORY\_DESC |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **ROOM\_BEDS** |  |  |  |
| Room\_bed\_id | INT(11) | PRIMARY KEY | ROOM\_BED\_ID |
| Room\_master\_id | INT(11) | FOREIGN KEY | ROOM\_MASTER\_ID |
| Bed\_name | VARCHAR(25) | NOT NULL | BED\_NAME |
| nStatus | VARCHAR(15) | NOT NULL | NSTATUS |
| Patient\_no | VARCHAR(25) | NOT NULL | PATIENT\_NO |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **DOCTOR\_FEE** |  |  |  |
| doctorfeeID | INT(11) | PRIMARY KEY | DOCTORFEEID |
| User\_id | VARCHAR(11) | FOREIGN KEY | USER\_ID |
| Invoice\_no | VARCHAR(25) | NOT NULL | INVOICE\_NO |
| Date | DATE | NOT NULL | DATE |
| CompleteDate | VARCHAR(50) | NOT NULL | COMPLETEDATE |
| feeType | VARCHAR(25) | NOT NULL | FEETYPE |
| Value | FLOAT(11,2) | NOT NULL | VALUE |
| TotalFee | FLOAT(11,2) | NOT NULL | TOTALFEE |
| notes | TEXT | NOT NULL | NOTES |
|  |  |  |  |
| **BILL\_GROUP\_NAME** |  |  |  |
| Group\_id | INT(11) | PRIMARY KEY | GROUP\_ID |
| Group\_name | VARCHAR(150) | NOT NULL | GROUP\_NAME |
| Group\_desc | TEXT | NOT NULL | GROUP\_DESC |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **BILL\_PARTICULAR** | INT(11) | PRIMARY KEY |  |
| Particular\_id | INT(11) | NOT NULL | PARTICULAR\_ID |
| Group\_id | VARCHAR(150) | NOT NULL | GROUP\_ID |
| Particular\_desc | TEXT | NOT NULL | PARTICULAR\_DESC |
| Charge\_amount | FLOAT(11,2) | NOT NULL | CHARGE\_AMOUNT |
| InActive | INT(11) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_BILLING** |  |  |  |
| Bill\_id | INT(11) | PRIMARY KEY | BILL\_ID |
| Receipt\_no | VARCHAR(25) | NOT NULL | RECEIPT\_NO |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| Patient\_no | VARCHAR(25) | FOREIGN KEY | PATIENT\_NO |
| Payment\_type | VARCHAR(25) | NOT NULL | PAYMENT\_TYPE |
| Invoice\_no | VARCHAR(50) | FOREIGN KEY | INVOICE\_NO |
| dDate | DATE | NOT NULL | DDATE |
| Discount | FLOAT(11,2) | NOT NULL | DISCOUNT |
| Reason\_discount | INT(5) | NOT NULL | REASON\_DISCOUNT |
| Sub\_total | FLOAT(11,2) | NOT NULL | SUB\_TOTAL |
| Total\_amount | FLOAT(11,2) | NOT NULL | TOTAL\_AMOUNT |
| Total\_purchased | FLOAT(11,2) | NOT NULL | TOTAL\_PURCHASED |
| creditCardNo | INT(11) | NOT NULL | CREDITCARDNO |
| creditCardHolder | INT(11) | NOT NULL | CREDITCARDHOLDER |
| Insurance\_company | INT(11) | NOT NULL | INSURAN\_COMPANY |
| Remarks | TEXT | NOT NULL | REMARKS |
| inActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **IOP\_BED\_SIDE\_PROCEDURE** |  |  |  |
| Bed\_pro\_id | INT(11) | PRIMARY KEY | BED\_PRO\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| Qty | INT(5) | NOT NULL | QTY |
| Notes | TEXT | NOT NULL | NOTES |
| cPreparedBy | VARCHAR(25) | NOT NULL | CPREPAREDBY |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_COMPLAINTS** |  |  |  |
| Iop\_com\_id | INT(11) | PRIMARY KEY | IOP\_COM\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| Complain\_id | INT(11) | FOREIGN KEY | COMPLAIN\_ID |
| Remarks | TEXT | NOT NULL | REMARKS |
| dDate | DATE | NOT NULL | DDATE |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_DIAGNOSIS** |  |  |  |
| Iop\_diag\_id | INT(11) | PRIMARY KEY | IOP\_DIAG\_ID |
| Iop\_id | VARCHAR(11) | FOREIGN KEY | IOP\_ID |
| Diagnosis\_id | INT(11) | NOT NULL | DIAGNOSIS\_ID |
| Remarks | TEXT | NOT NULL | REMARKS |
| dDate | DATETIME | NOT NULL | DDATE |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_DISCHARGE\_SUMMARY** |  |  |  |
| Dis\_id | INT(11) | PRIMARY KEY | DIS\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | DATETIME | NOT NULL | DDATETIME |
| Reason\_admission | TEXT | NOT NULL | REASON\_ADMISSION |
| Condition\_upon\_discharge | INT(11) | NOT NULL | COND\_UPON\_DISC |
| Admitting\_impression | TEXT | NOT NULL | ADMITTING\_IMP |
| Final\_diagnosis | TEXT | NOT NULL | FINAL\_DIAGNOSIS |
| Physical\_exam\_findings | TEXT | NOT NULL | PHYS\_EXAM\_FINDIS |
| Course\_ward | TEXT | NOT NULL | COURSE\_WARD |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **IOP\_INTAKE\_RECORD** |  |  |  |
| Intake\_id | INT(11) | PRIMARY KEY | INTAKE\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| particulars | VARCHAR(500) | NOT NULL | PARTICULARS |
| IV\_fluids | VARCHAR(25) | NOT NULL | IV\_FLUIDS |
| oral | VARCHAR(25) | NOT NULL | ORAL |
| No\_stool | VARCHAR(25) | NOT NULL | NO\_STOOL |
| No\_urine | VARCHAR(25) | NOT NULL | NO\_URINE |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_MEDICINE** |  |  |  |
| Iop\_med\_id | INT(11) | PRIMARY KEY | IOP\_MED\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| Medicine\_id | INT(11) | FOREIGN KEY | MEDICINE\_ID |
| instruction | TEXT | NOT NULL | INSTRUCTION |
| advice | TEXT | NOT NULL | ADVICE |
| days | INT(2) | NOT NULL | DAYS |
| Total\_qty | INT(5) | NOT NULL | TOTAL\_QTY |
| InActive | INT(1) | NOT NULL | INACTIVE |
| dDate | DATETIME | NOT NULL | DDATE |
|  |  |  |  |
| **IOP\_LABORATORY** |  |  |  |
| Io\_lab\_id | INT(11) | PRIMARY KEY | IO\_LAB\_ID |
| Iop\_id | VARCHAR(11) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(100) | NOT NULL | DDATETIME |
| Category\_id | INT(11) | NOT NULL | CATEGORY\_ID |
| Laboratory\_id | INT(11) | NOT NULL | LABORATORY\_ID |
| Findings | TEXT | NOT NULL | FINDINGS |
| Result | TEXT | NOT NULL | RESULT |
| Doctor | INT(11) | NOT NULL | DOCTOR |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_NURSE\_NOTES** |  |  |  |
| Nurse\_notes\_id | INT(11) | PRIMARY KEY | NURSE\_NOTES\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| Focus | VARCHAR(50) | NOT NULL | FOCUS |
| Notes | TEXT | NOT NULL | NOTES |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **IOP\_OPERATION\_THEATER** |  |  |  |
| Operation\_id | INT(11) | PRIMARY KEY | OPERATION\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate\_from | DATE | NOT NULL | DDATE\_FROM |
| dTime\_from | TIME | NOT NULL | DTIME\_FROM |
| dDate\_to | DATE | NOT NULL | DDATE\_TO |
| dTime\_to | TIME | NOT NULL | DTIME\_TO |
| Operation\_name | VARCHAR(250) | NOT NULL | OPERATION\_NAME |
| Bed\_id | INT(11) | FOREIGN KEY | BED\_ID |
| Diagnosis | TEXT | NOT NULL | DIAGNOSIS |
| Name\_of\_surgeon | VARCHAR(25) | NOT NULL | NAME\_OF\_SURG |
| Name\_of\_anesthesia | VARCHAR(25) | NOT NULL | NAME\_OF\_ANESIA |
| Assistant\_name | VARCHAR(150) | NOT NULL | ASSISTANT\_NAME |
| Operation\_procedure | TEXT | NOT NULL | OPATION\_PRORE |
| Notes | TEXT | NOT NULL | NOTES |
| Date\_created | TIMESTAMP | NOT NUL | DATE\_CREATED |
|  |  |  |  |
| **IOP\_OUTPUT\_RECORD** |  |  |  |
| Output\_id | INT(11) | PRIMARY KEY | OUTPUT\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| Urine | VARCHAR(25) | NOT NULL | URINE |
| Feaces | VARCHAR(25) | NOT NULL | FEACES |
| Respitation | VARCHAR(25) | NOT NULL | RESPITATION |
| Skin | VARCHAR(25) | NOT NULL | SKIN |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| InActive | INT(1) | NOT NULL | INACTIVE |
|  |  |  |  |
| **IOP\_PROGRESS\_NOTE** |  |  |  |
| Progess\_id | INT(11) | PRIMARY KEY | PROGESS\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| Progess | TEXT | NOT NULL | PROGESS |
| Treatment | TEXT | NOT NULL | TREATMENT |
| Remarks | TEXT | NOT NULL | REMARKS |
|  |  |  |  |
| **IOP\_ROOM\_TRNASFER** |  |  |  |
| Transfer\_id | INT(11) | PRIMARY KEY | TRANSFER\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| Room\_category\_id | INT(3) | FOREIGN KEY | ROOM\_CATRY\_ID |
| Room\_master\_id | INT(3) | FOREIGN KEY | ROOM\_MASTER\_ID |
| Bed\_id | INT(11) | NOT NULL | BED\_ID |
| reason | TEXT | NOT NULL | REASON |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **IOP\_VITAL\_PARAMETERS** |  |  |  |
| Vital\_id | INT(11) | PRIMARY KEY | VITAL\_ID |
| Iop\_id | VARCHAR(25) | FOREIGN KEY | IOP\_ID |
| dDate | DATE | NOT NULL | DDATE |
| dDateTime | VARCHAR(50) | NOT NULL | DDATETIME |
| Pulse\_rate | VARCHAR(25) | NOT NULL | PULSE\_RATE |
| Temperature | VARCHAR(25) | NOT NULL | TEMPERATURE |
| Height | VARCHAR(25) | NOT NULL | HEIGHT |
| Bp | VARCHAR(25) | NOT NULL | BP |
| Respiration | VARCHAR(25) | NOT NULL | RESPIRATION |
| Weight | VARCHAR(25) | NOT NULL | WEIGHT |
|  |  |  |  |
| **BB\_BLOOD\_INVENTORY** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Blood\_group | VARCHAR(10) | NOT NULL | BLOOD\_GROUP |
| Volume | FLOAT | NOT NULL | VOLUME |
| Status | TINYINT(1) | NOT NULL | STATUS |
| Donor\_id | INT(30) | FOREIGN KEY | DONOR\_ID |
| Request\_id | INT(30) | FOREIGN KEY | REQUEST\_ID |
| Date\_created | DATETIME | NOT NULL | DATE\_CREATED |
|  |  |  |  |
| **BB\_DONORS** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Blood\_group | VARCHAR(10) | NOT NULL | BLOOD\_GROUP |
| Name | TEXT | NOT NULL | NAME |
| Address | TEXT | NOT NULL | ADDRESS |
| Contact | VARCHAR(20) | NOT NULL | CONTACT |
| Email | VARCHAR(50) | NOT NULL | EMAIL |
| Date\_created | DATETIME | NOT NULL | DATE\_CREATED |
|  |  |  |  |
| **BB\_REQUESTS** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Ref\_code | VARCHAR(20) | FOREIGN KEY | REF\_CODE |
| Patient | TEXT | NOT NULL | PATIENT |
| Blood\_group | VARCHAR(10) | NOT NULL | BLOOD\_GROUP |
| Volume | FLOAT | NOT NULL | VOLUME |
| Physician\_name | TEXT | NOT NULL | PHYSICIAN\_NAME |
| Status | TINYINT(1) | NOT NULL | STATUS |
| Date\_created | DATETIME | NOT NULL | DATE\_CREATED |
|  |  |  |  |
|  |  |  |  |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
|  |  |  |  |
| **BB\_HANEDOVER\_REQUEST** |  |  |  |
| id | INT(30) | PRIMARY KEY | ID |
| Request\_id | INT(30) | FOREIGN KEY | REQUEST\_ID |
| Picked\_up\_by | TEXT | NOT NULL | PICKED\_UP\_BY |
| Date\_created | DATETIME | NOT NULL | DATE\_CREATED |
|  |  |  |  |
| **CB\_QUESTIONS** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Question | TEXT | NOT NULL | QUESTION |
| Response\_id | INT(30) | FOREIGN KEY | RESPONSE\_ID |
|  |  |  |  |
| **CB\_RESPONSES** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Response\_message | TEXT | NOT NULL | RESP\_MESAGE |
|  |  |  |  |
| **CB\_UNANSWERED** |  |  |  |
| Id | INT(30) | PRIMARY KEY | ID |
| Question | TEXT | NOT NULL | QUESTION |
| No\_asks | INT(30) | NOT NULL | NO\_ASKS |

**PROCESS LOGIC AND IMPLEMENTATION METHODOLOTY:**

The complete Process Logic of various module and their implementation methodology is mentioned below:

**(a) Reception:-**

This module is logged in by the receptionist by his user id and password which will be provided by the system administrator. The receptionist can make patient management and they can add and edit patient details in the e-MIMS. Based on the registration patient can get unique ID for future reference. And receptionist can make registered patient appointment with doctor. All appointment will be displayed on the dashboard.

**(b) Doctor:-**

The Doctor will get the role based access on e-MIMS Web application. The doctor will treat the OPD & IPD patient and prescribe the medicine to patient and serious condition patient will be given admit advice by the doctor.

**(c) Nurse:-**

On successful login Nurse will come to dashboard of Nurse moduel. Where all IPD patient management will be done . where patient medicine records, intake/Output Records, Nursing progress notes, vital notes, Bed side Procedure, IP Room Transfer, patient History, Discharge Summary will be managed by the Nurse.

**(d) Staff**

For Hospital management staff portal login id and password is generated by the admin staff. After login in e-MIMS staff dashboard will open. Where all patient appointments will be displayed. He generate the patient bill and provides the surgical quotation. He can also update his profile and change password which is updated in staff database table.

**(e) Report:-**

For report generation staff portal login id and password is generated by the admin staff. After login in e-MIMS staff dashboard will open. Where all patient appointments will be displayed. He generate the patient master list report, individual patient report, output patient report, admitted patient report discharged patient report, doctor’s fee, daily sales reports. patient bill and provides the surgical quotation. He can also update his profile and change password which is updated in concerned database table.

**(f) Blood Bank Management:-**

Blood bank management module is managed and control by admin staff of The Hospital. After login by the Admin staff he can keep record of donor list and blood handed over request and blood request. Timely blood can be provided to the needy patient.

**(g) Chatbot:-**

Chatbot moduleis managed and controlled by the admin staff of the Hospital. On main login page Chatbot can interact with user for the information which will be fed into the e-MIMS application by Admin staff. All responses to the questions will be given by the admin staff. And unanswered question list will be displayed in admin control blood bank dash board.

**(h) Admin:-**

Admin is having whole control of this e-Medical Information Management system. His login id and password is initial created by web application development team Which can be change in change password page after login. And the data is updated in user database table. On login e-MIMS dashboard will open. Where all others modules with additional module will be managed.

Admin module having access staff module, patient management module, nurse module and doctor module, report module, and blood bank management module and chatbot module and some addition module room management, EMR sheet records for in and Out patient, user management are done by the admin.

Hence Admin can control the whole activites of this application.

**LIST OF REPORTS LIKELY TO BE GENERATED:-**

|  |  |  |
| --- | --- | --- |
| Patient Masterlist report | : | view details of patient masterlist |
| Individual Patient report | : | View details of individual patient report |
| Out patient report | : | View details of out patient report |
| Admitted patient report | : | View details of admitted patient report |
| Discharged patient report | : | View details of discharged patient report |
| Daily sales report | : | View details of daily sales reports |
| Doctor’s fee report | : | View details of Docter’s fee report |
| Acknowledge receipt report | : | Vies details of acknowledge receipt report |

**OVERALL NETWORK ARCHITECTURE:-**

As the Hospital is hosting its website and all users are having Local area network connected computers can access e-Medical Information Management system Web application. So All are connected through intranet.

**Computer networks can be used for a variety of purposes**:

* Facilitating communications:- Using a network, people can communicate efficiently and easily via email, instant messaging, telephone, video telephone calls, and video conferencing
* Sharing hardware:-In a networked environment, each computer on a network may access and use hardware resources on the network, such as printing a document on a shared network printer.
* Sharing files, data, and information:- In a network environment, authorized user may access data and information stored on other computers on the network. The capability of providing access to data and information on shared storage devices is an important feature of many networks
* Sharing software:- Users connected to a network may run application programmes on remote computers.
* Information preservation.
* Security.
* Speed up.

**Security and Validations Check:-**

**Module Level Security and Validations:-**

In this project different validation checks have been used. Once the form is being filled up, then it has to be validated properly after checking whether it satisfies all stipulated rules or not. Database constraints are used to stop data corruption. All the Tables are linked together using Primary key – Foreign key relationship.

Security is one of the one of the main component of the project. The concept of Secure Login has been implemented. Using Secure Login, all individuals will login to the software and do all their activities. All these activities will be monitored by capturing logs in various Audit tables.

The user password is kept secure using a very complex encryption algorithm. The password is to be at least 10 characters long, must contain a mix of alphabets, numbers and special characters.

In addition to module level security and validation checks, PHP and MYSQL has the edge in providing Strict Security and System integrity.

**PHP provides following securities:-**

(i) Authentication  
(ii) Authorization  
(iii) Data integrity  
(iv) Data confidentiality.

**MYSql 5.X provides following securities:-**

(i) MySQL offers encryption using Secure Sockets Layer (SSL) protocol.

(ii) provides data masking..

(iii) authentication plugin, other layers of security to protect data integrity

(iv) MySQL supports multi-threading that makes it easily scalable.

(v) MySQL allows transactions to be rolled back, commit, and crash recovery.

(vi) Its efficiency is high because it has a very low memory leakage problem.

(vii) MySQL is faster, more reliable and, cheaper because of its unique storage engine architecture.

(viii) MySQL uses triggers, Stored procedures, and views that allow the developer to give higher productivity.

(ix) it is platform independent can be install and execute on most of the available operating systems.

(x) Partitioning features improves the performance and provides fast management of the large database.

**FUTURE SCOPE OF THE PROJECT:-**

The e-Medical Information Management System is aimed to provide the relief to the staff and patients of the Hospital from manual channel and to search through bulky files of reply of queries. If this system is implemented successfully, it will provide a very good communication channel within the Hospital. With some modifications depending on the feedback gained in run-time, more fruitful results can be obtained.

Once the Software is successfully completed and runs smoothly, it can be of great help for all other Hospital to automate their respective Hospital management system. Also, if the e-Medical Information management system gets automated, it will enhance the scope for further automation for by adding other module which will reduce the Hospital unnecessary paper work.

Here we can maintain the records of Hospital and Doctors. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the Hospital Management System. Enhancements can be done to maintain all the Hospital, Doctors, Patient, staff, report. We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

**BIBLIOGRAPHY:-**

1. IGNOU:- Sytem and analysis Design (MCS-014).
2. IGNOU:- Introduction to Database Management Systems(MCS-023).
3. IGNOU:- Software Engineering (MCS-034).
4. IGNOU:- Advanced Database Management Systems (MCS-043).
5. Lynn Beighley & Micchael Morrison :- The head first **PHP & MySQL** by O’REILLY.

Referred Web Pages:-

1. Google for problem solving
2. [www.w3schools.com](http://www.w3schools.com).
3. www.codecademy.com
4. [www.](http://www.)udemy.com.
5. www.tutorialspoint.com
6. www.stackoverflow.com
7. www.tutorialspoint.com/mysql/