



**Department of Information and Communication Technology  
Faculty of Technology  
University of Ruhuna**

**Database Management Systems Practicum  
ICT 1222  
Assignment 02**

Group 04

Submitted to: Ms. W.A.M. Prabuddhi

Submitted by: TG/2020/672 – G.W.M.D. Rupasinghe  
TG/2020/700 – J.M.N.R.D. Jayaweera  
TG/2020/703 – K.A.D.P.M. Siriwardana  
TG/2020/711 – H. M. M. A. Herath  
TG/2020/723 – W.V.P.V. Karunasinghe  
TG/2020/726 – M.K.K. Jithma  
TG/2020/749 – H.P.G.L.P. Jayathilake  
TG/2020/751 – N.W.V.B.S.B. Weragoda

## Table of Contents

1	Brief introduction about the problem/group project.....	1
2	Brief introduction to the solution .....	1
3	Proposed ER/EER diagram .....	2
4	Proposed Relational mapping diagram.....	3
5	Table structure of your solution .....	4
5.1	Double click on the icon to view Table structure.....	6
6	Architecture of your solution.....	7
7	Tools and technologies that you have used.....	7
8	Security measures that you have taken to protect your DB.....	7
9	Brief description about DB Accounts/Users and the reasons for creating such Accounts/Users ..	7
10	Code snippets to support your work .....	8
10.1	Query 01: (To view all details of doctors) .....	8
10.2	Query 02: (To view all details of nurses).....	8
10.3	Query 03: (To view all details of ward boys).....	8
10.4	Query 04: (To view all details of patient).....	8
10.5	Query 05: (To view details about medicine) .....	9
10.6	Query 06: (To view Given medication details) .....	9
10.7	Query 07: (To view patient contact numbers).....	9
10.8	Query 08: (To view Inpatient records with all details).....	10
10.9	Query 09: (To view Outpatient records with all details).....	10
10.10	Query 10: (To view inpatient who are still in the hospital) .....	10
10.11	Query 11:(To view Bill details) .....	11
10.12	Query 12: (To view Unique patient bill details by input patient_ID) .....	11
10.13	Query 13:(To view available Medicine in pharmacies) .....	11
10.14	Query 14:(To view details about treatment) .....	12
11	Problems that you faced during the development of the solution .....	12
12	Solutions/how you have overcome the above identified problems .....	12
13	New database technologies/trends that you have used to develop the backend .....	13
14	If you are going to host your backend, where are you going to host it and reasons for the Selection.....	14
15	If you are going to host your backend in a cloud environment what are the things/changes that you must do in your backend.....	14
16	Individual contribution to the backend development.....	14
17	References .....	15

## 1 Brief introduction about the problem/group project

In this hospital used manual data management system. In that case there are some problems happened when they used that manual system. To do all data entering and updating humans should involve directly.

In manual system cannot identify if there any data redundancies. Because of that same data can be duplicated. When they want emergency data, they have to check all files manually. All the files should check to find data. So, there can be delay and very slow.

In that hospital there should be so many data files for store all the data about patients, staff and treatments. So that need more physical space to store data files. If a data file is lost due to any reason more time can be west to recover it. Very difficult to keep data backups.

In manual system all the process are happed in manually. Because of that there can be some data entering and processing errors or some data can be missed while entering data. Cannot ensure accuracy of data.

Any people can access and can do any kind of changes. Lack of security was a problem on that system.

## 2 Brief introduction to the solution

To avoid these problems, we developed and introduced an automated data base management system. In here data entering and updating and retrieving is very easy when comparing these two systems.

In that system we use primary keys to avoid data redundancies. No data duplications and no space wasting happen. When they want emergency data they can search and find by using key words. It is very quickly. No need to wait so long.

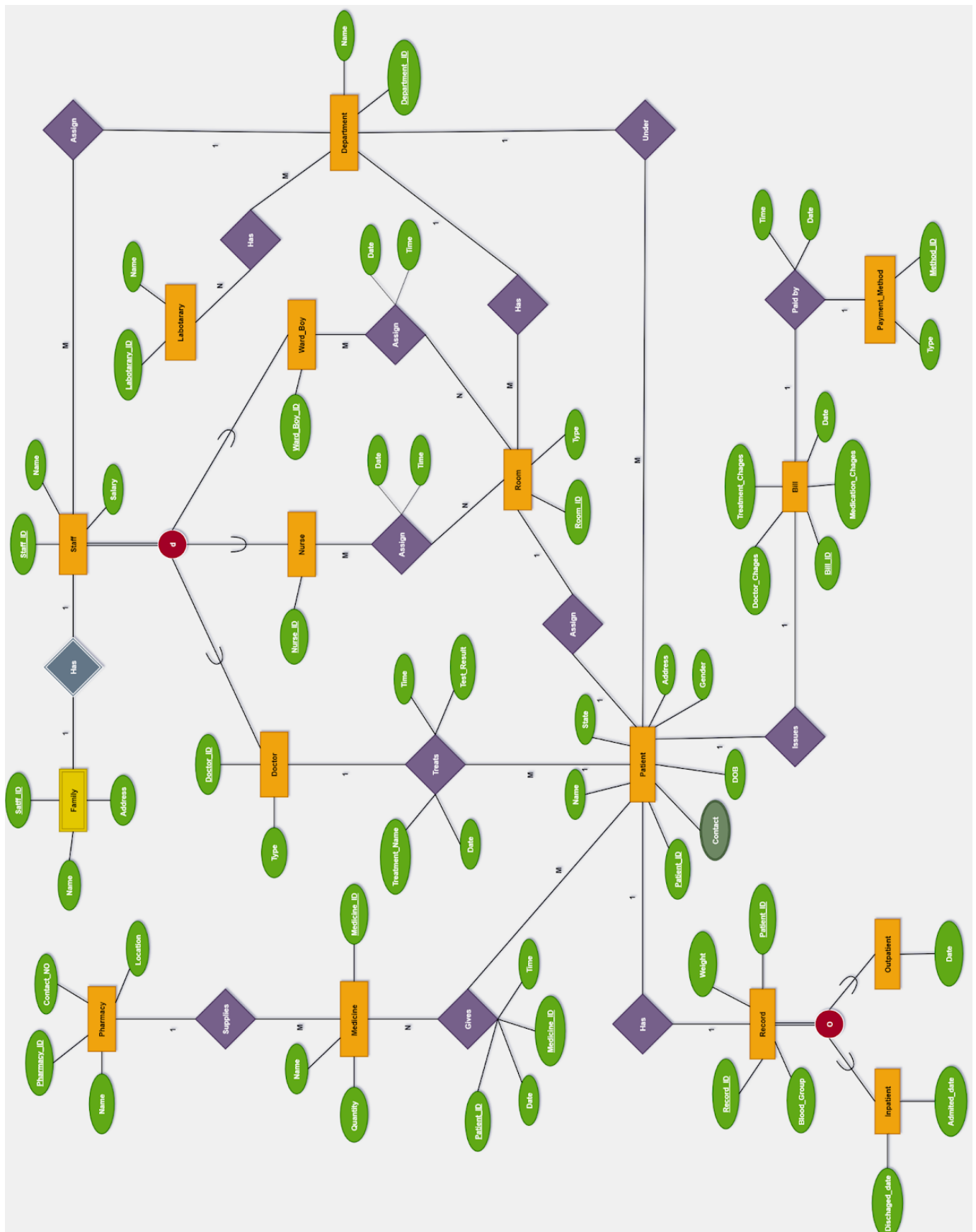
To store all kind of data there should be only computer system with enough storage. No need more physical space to store like a manual system. If any data is lost easily can recover it because of backup facilities.

In that data base management system does not allow to enter invalid, irrelevant data or incorrect data. System can identify those anomalies. If there is any essential data system not allowed to miss out it. All the data processing is happed in automatable way. There is no need to involve humans for that. Data accuracy is high.

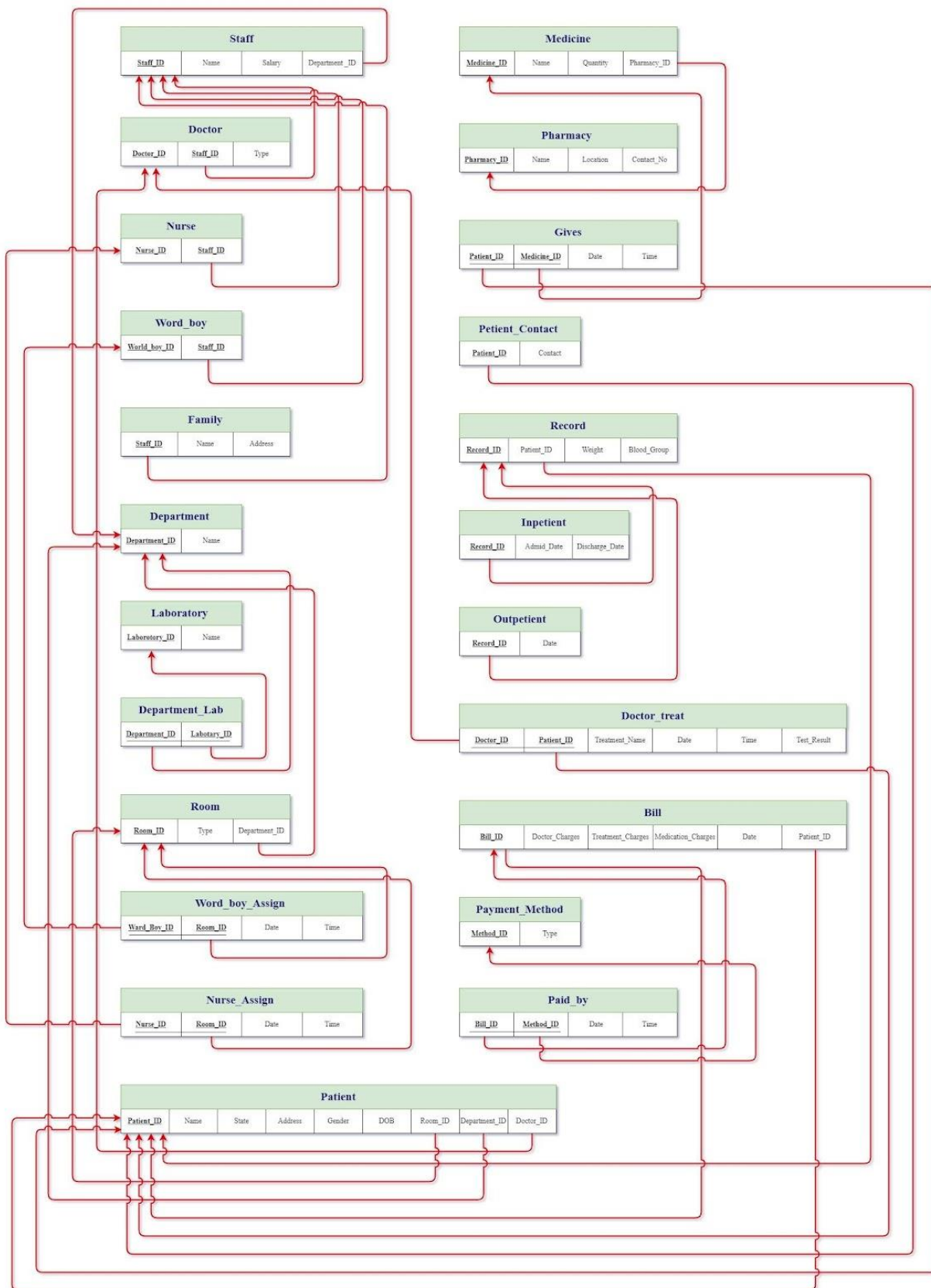
Automated systems can limit access authority. Only authorized people can access and operate the system. So, in this kind of systems security is high. The risk of data destruction is minimal.

Few people can handle this system very easily. Lower operating cost and labor is low.

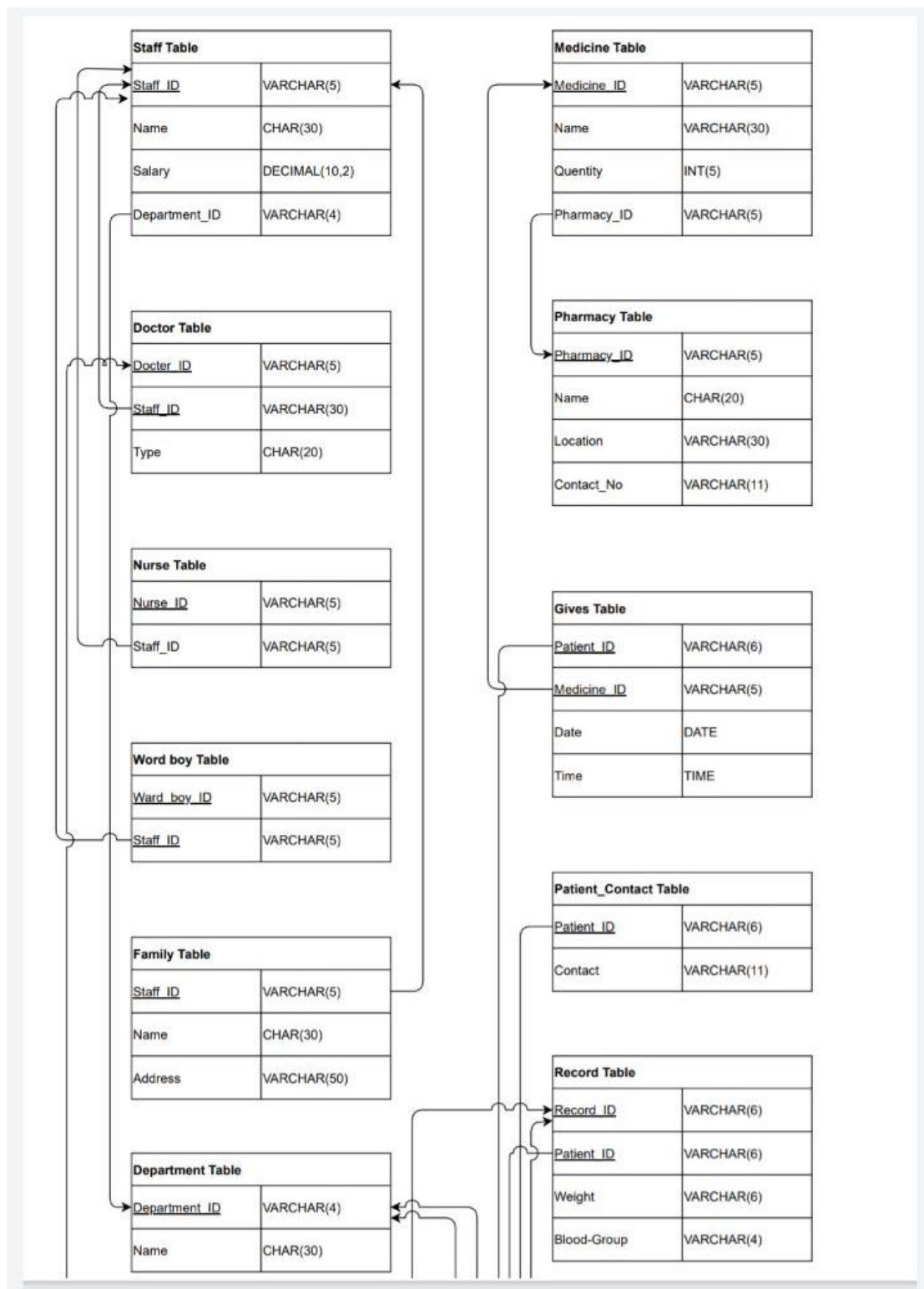
### 3 Proposed ER/EER diagram

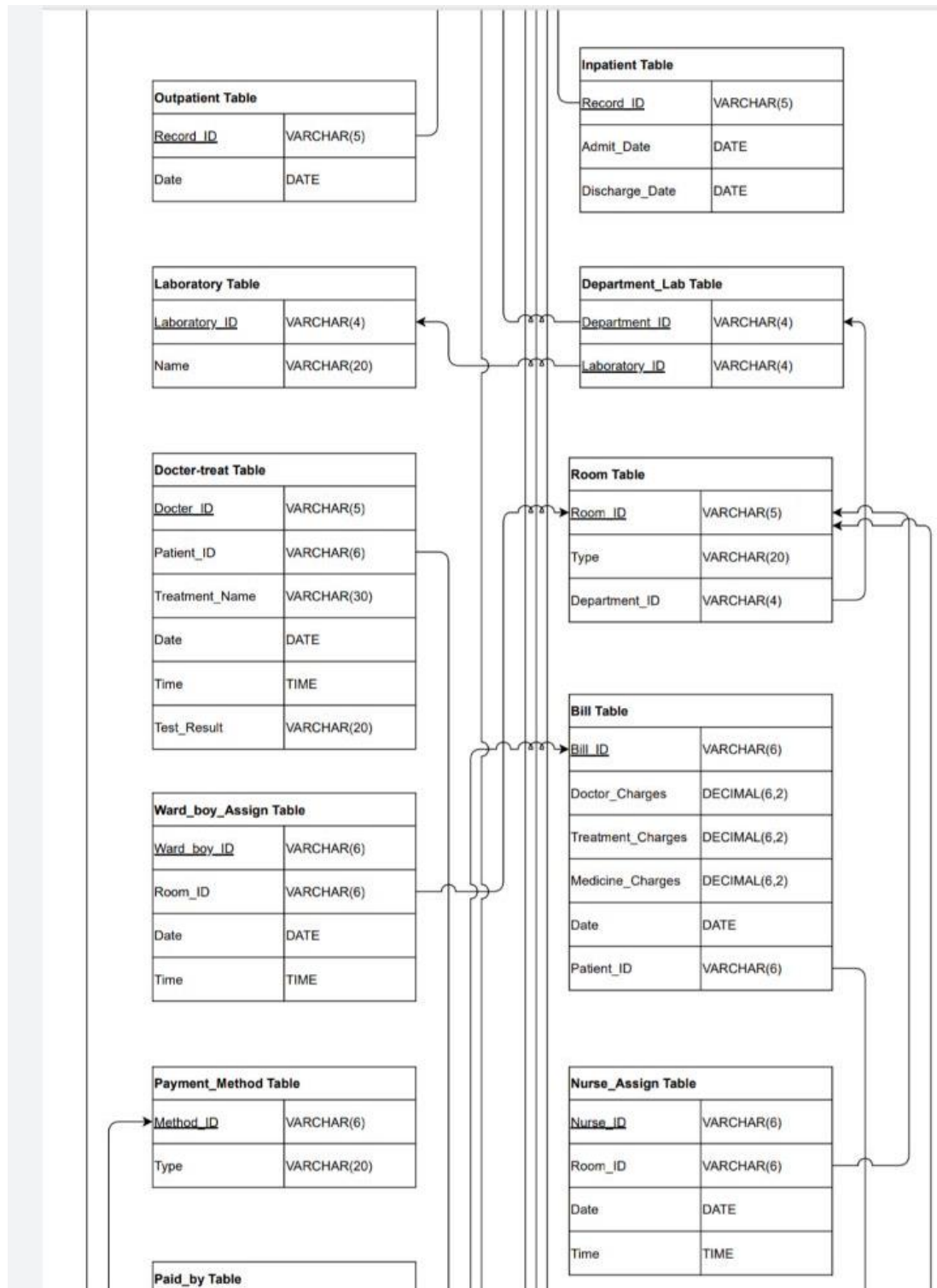


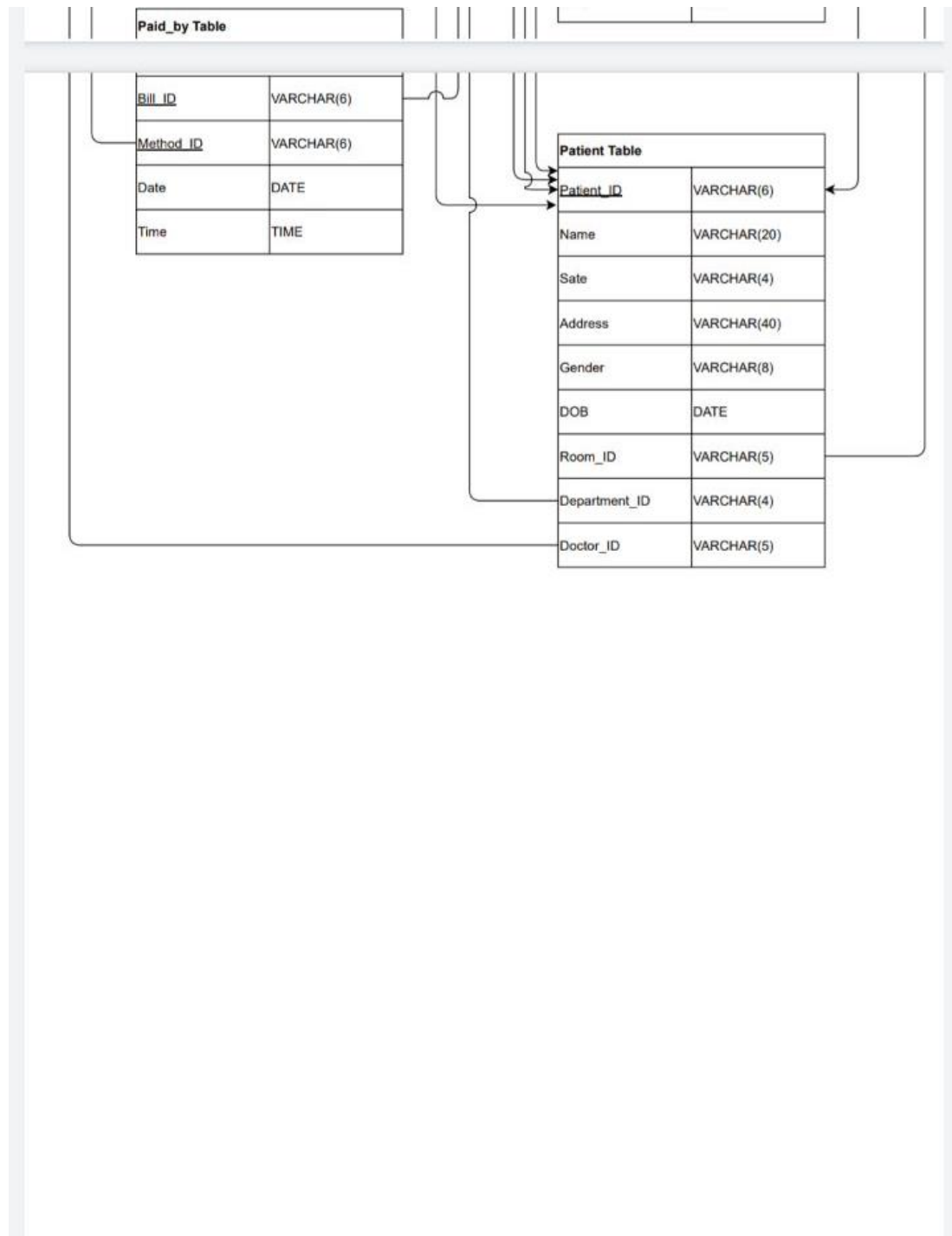
## 4 Proposed Relational mapping diagram



## 5 Table structure of your solution







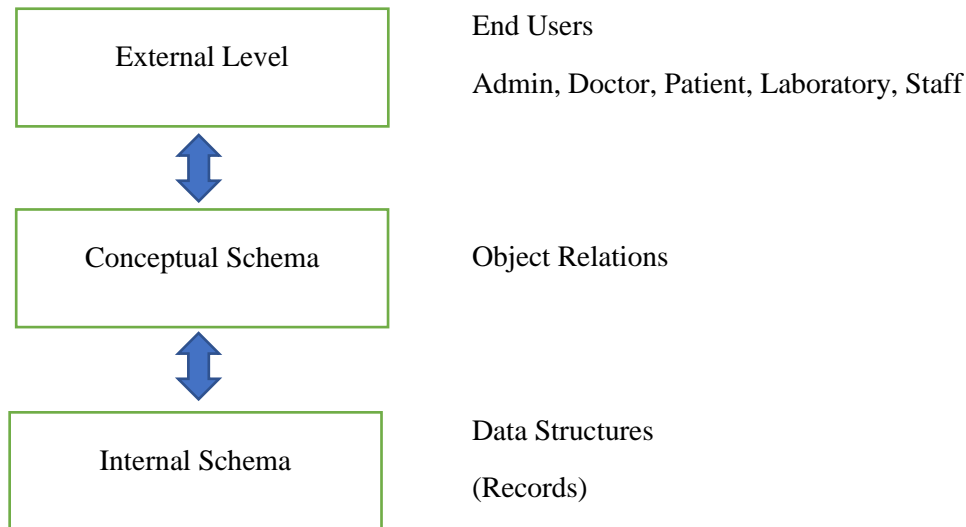
### 5.1 Double click on the icon to view Table structure.



Table Structure.jpg



## 6 Architecture of your solution



## 7 Tools and technologies that you have used

- Draw.io:
  - Used to draw ER diagram, relational schema, and table structure
- MySQL Workbench, VS code, Notepad, WAMP server:
  - Used to create database

## 8 Security measures that you have taken to protect your DB

- Applying user privileges
  - Admin: With All privileges with Grant
  - Doctor: With All privileges without Grant
  - Patients: Read permissions all tables/views
  - Laboratory Staff: All privileges without Grant and user creation

## 9 Brief description about DB Accounts/Users and the reasons for creating such Accounts/Users

- Our Hospital Data Management System contain below user accounts:
  - Admin
    - Create and maintain user profile.
  - Doctor
    - Update records
    - Create new records
    - View data and information
  - Patient
    - View data and information only
  - Laboratory

- Update records
- Create new records

## 10 Code snippets to support your work

### 10.1 Query 01: (To view all details of doctors)

```
DELIMITER //
CREATE PROCEDURE view_doc_details()
BEGIN
    SELECT s.Staff_ID,d.Doctor_ID,s.Name,dep.Name AS "Department Name",s.salary
    FROM staff AS s,Doctor AS d,department AS dep
    WHERE s.staff_ID=d.staff_ID AND dep.Department_ID=s.Department_ID;
END//
DELIMITER ;
```

**CALL view\_doc\_details();**

### 10.2 Query 02: (To view all details of nurses)

```
DELIMITER //
CREATE PROCEDURE view_nurse_details()
BEGIN
    SELECT s.Staff_ID,n.Nurse_ID,s.Name,dep.Name AS "Department Name",s.salary
    FROM staff AS s,nurse AS n,department AS dep
    WHERE s.staff_ID=n.staff_ID AND dep.Department_ID=s.Department_ID;
END//
DELIMITER ;
```

**CALL view\_nurse\_details();**

### 10.3 Query 03: (To view all details of ward boys)

```
DELIMITER //
CREATE PROCEDURE view_ward_boy_details()
BEGIN
    SELECT s.Staff_ID,w.ward_boy_ID,s.Name,dep.Name AS "Department
    Name",s.salary
    FROM staff AS s,ward_boy AS w,department AS dep
    WHERE s.staff_ID=w.staff_ID AND dep.Department_ID=s.Department_ID;
END//
DELIMITER ;
```

**CALL view\_ward\_boy\_details();**

### 10.4 Query 04: (To view all details of patient)

```
DELIMITER //
CREATE PROCEDURE view_patient_details()
BEGIN
```

```
SELECT
p.Patient_ID,p.Name,p.State,p.Address,p.Gender,p.DOB,p.Room_ID,s.Name AS
"Doctor Name",dep.Name AS "Department Name"
FROM patient AS p,record AS r,doctor AS d,department AS dep,staff AS s
WHERE p.patient_ID=r.patient_ID AND p.Doctor_ID=d.Doctor_ID AND
p.Department_ID=dep.Department_ID AND s.staff_ID=d.Staff_ID;
END//
DELIMITER ;

CALL view_patient_details();
```

### 10.5 Query 05: (To view details about medicine)

```
DELIMITER //
CREATE PROCEDURE view_med_details()
BEGIN
    SELECT M.Medicine_ID,M.MeName AS "Medicine Name",M.Quantity,P.PhName
    AS "Pharmacy Name",P.Location,P.Contact_No
    FROM Medicine AS M,pharmacy AS P
    WHERE M.Pharmacy_ID=P.Pharmacy_ID;
END//
DELIMITER ;

CALL view_med_details();
```

### 10.6 Query 06: (To view Given medication details)

```
DELIMITER //
CREATE PROCEDURE view_gmed_details()
BEGIN
    SELECT G.patient_ID,M.MeName AS "Medicine Name",G_Date AS "Given
    Date",G_Time AS "Given Time"
    FROM Gives AS G,Medicine AS M
    WHERE G.Medicine_ID=M.Medicine_ID;
END//
DELIMITER ;

CALL view_gmed_details();
```

### 10.7 Query 07: (To view patient contact numbers)

```
DELIMITER //
CREATE PROCEDURE view_pcontact_details()
BEGIN
    SELECT PC.patient_ID,P.Name,PC.Contact
    FROM patient_contact AS PC,Patient AS P
    WHERE PC.patient_ID=P.patient_ID;
END//
DELIMITER ;

CALL view_pcontact_details();
```

**10.8 Query 08: (To view Inpatient records with all details)**

```
DELIMITER //
CREATE PROCEDURE view_inpatient_details()
BEGIN
    SELECT R.record_ID,R.patient_ID,P.Name,R.weight,R.Blood_Group,dep.Name AS
    "Department Name",P.Room_ID,IP.Admit_Date,IP.Discharge_Date
    FROM record AS R,patient AS P,Inpatient AS IP,Department AS dep
    WHERE R.patient_ID=P.patient_ID AND p.Room_ID IS NOT NULL AND
    IP.record_ID=R.record_ID AND p.department_ID=dep.department_ID;
END//
DELIMITER ;

CALL view_inpatient_details();
```

**10.9 Query 09: (To view Outpatient records with all details)**

```
DELIMITER //
CREATE PROCEDURE view_outpatient_details()
BEGIN
    SELECT R.record_ID,R.patient_ID,P.Name,R.weight,R.Blood_Group,OP.ODate AS
    "OPD DATE"
    FROM record AS R,patient AS P,Outpatient AS OP
    WHERE R.patient_ID=P.patient_ID AND p.Room_ID IS NULL AND
    OP.record_ID=R.record_ID;
END//
DELIMITER ;

CALL view_outpatient_details();
```

**10.10 Query 10: (To view inpatient who are still in the hospital)**

```
DELIMITER //
CREATE PROCEDURE view_still_inpatient_details()
BEGIN
    SELECT R.record_ID,R.patient_ID,P.Name,R.weight,R.Blood_Group,dep.Name AS
    "Department Name",P.Room_ID,IP.Admit_Date
    FROM record AS R,patient AS P,Inpatient AS IP,Department AS dep
    WHERE R.patient_ID=P.patient_ID AND p.Room_ID IS NOT NULL AND
    IP.record_ID=R.record_ID AND p.department_ID=dep.department_ID AND
    IP.Discharge_Date IS NULL;
END//
DELIMITER ;
```

**CALL view\_still\_inpatient\_details();**

### **10.11 Query 11:(To view Bill details)**

```
DELIMITER //
CREATE PROCEDURE view_bill_details()
BEGIN
    SELECT B.Bill_ID,B.patient_ID,P.Name AS "Patient
    Name",B.Doctor_Charges,B.Treatment_Charges,B.Medicine_Charges,B.BDate AS
    "BILL Date",(B.Doctor_Charges + B.Treatment_Charges + B.Medicine_Charges) AS
    "Total Amount"
    FROM bill AS B,patient AS P
    WHERE B.patient_ID=P.patient_ID;
END//
DELIMITER ;
```

**CALL view\_bill\_details();**

### **10.12 Query 12: (To view Unique patient bill details by input patient\_ID)**

```
DELIMITER //
CREATE PROCEDURE view_patient_bill_details(IN x varchar(6))
BEGIN
    SELECT B.Bill_ID,B.patient_ID,P.Name AS "Patient
    Name",B.Doctor_Charges,B.Treatment_Charges,B.Medicine_Charges,B.BDate AS
    "BILL Date",(B.Doctor_Charges + B.Treatment_Charges + B.Medicine_Charges) AS
    "Total Amount"
    FROM bill AS B,patient AS P
    WHERE B.patient_ID=P.patient_ID AND B.patient_ID=x;
END//
DELIMITER ;
```

**CALL view\_patient\_bill\_details('PT0004');**

### **10.13 Query 13:(To view available Medicine in pharmacies)**

```
DELIMITER //
CREATE PROCEDURE view_available_medicine_details()
BEGIN
    SELECT Medicine_ID,MeName AS "Medicine Name",Quantity,PhName AS
    "Pharmacy Name",Location,Contact_No
    FROM medicine
    NATURAL JOIN pharmacy;
END//
DELIMITER ;
```

**CALL view\_available\_medicine\_details();**

#### **10.14 Query 14:(To view details about treatment)**

**DELIMITER //**

**CREATE PROCEDURE view\_treatment\_details()**

**BEGIN**

```
SELECT dt.patient_ID,p.Name AS "Patient Name",d.Doctor_ID,s.Name AS "Doctor
Name",dt.Treatment_Name,dt.D_Date AS "Treatment Given Date",D_Time AS
"Treatment Given Time",dt.Test_Result
FROM patient AS p,doctor AS d,doctor_treat AS dt,staff AS s
WHERE p.patient_ID=dt.patient_ID AND s.staff_ID=d.staff_ID AND
d.doctor_ID=dt.doctor_ID;
```

**END//**

**DELIMITER ;**

**CALL view\_treatment\_details();**

### **11 Problems that you faced during the development of the solution**

- When we assign nurses and ward boys to rooms, we need to consider a primary key at the very first time we take primary key as nurseID and RoomID & wardboyID and RoomID for both tables separately but when we think about more records we understood if we get only those two columns, we can't get each record uniquely.
- When we add data to tables there are some errors of data types.
- When we draw ER, we must decide in which place we need to put inpatient and outpatient.
- After all those things we add Bill entity for ER then we must decide how we can add payment method to this because we have three payment methods.

### **12 Solutions/how you have overcome the above identified problems**

- We decided to get three columns for primary key of those (Nurse Assign, Ward boy Assign) tables. Finally, we get primary key including date also.
- We use another suitable data types and ranges for fix them.

- We decided to put another record entity for each patient and at then put that attribute in that table after that according to patient type we specialize the table and divided those in to another two tables as Inpatient and outpatient.
- We decided to add another attribute as payment method and add those payment methods to it. After that we assign to payment method to Bill by relationship as paid by.

### **13 New database technologies/trends that you have used to develop the backend**

- MySQL
- MySQL Workbench
- Visual Studio Code
- Notepad ++
- WAMP Server

## **14 If you are going to host your backend, where are you going to host it and reasons for the Selection**

- In a cloud environment
  - Because when we use cloud environments, we can access our database any time anywhere with proper internet connection.
- We expect to store our database in cloud.  
Because,
  - Cloud can support relational databases very well.
  - Enables users to host databases without buying dedicated hardware.
  - We can be accessed through web interface.
  - We can expand our database capacity on run time.
  - In the event of a natural disaster, equipment failure or power outage we can secure our data through backups on remote servers.
  - We didn't want to pay another hosting services.

## **15 If you are going to host your backend in a cloud environment what are the things/changes that you must do in your backend**

We have to mod our current database structure. In the current database structure, there are some little bit issues in our database. Also, we need to add more security features to our database. In current database system we have added privileges for our database security.

## **16 Individual contribution to the backend development**

- TG/2020/672 – G.W.M.D. Rupasinghe – Data insert queries and Table Create.
- TG/2020/700 – J.M.N.R.D. Jayaweera – Queries (Store Procedure), Report, Relation Schema.
- TG/2020/703 – K.A.D.P.M. Siriwardana - Data insert queries and Table Create.
- TG/2020/711 – H. M. M. A. Herath – Update Queries, Table Create.
- TG/2020/723 – W.V.P.V. Karunasinghe – Join Queries, Table Create, Data insert, Data Dictionary.
- TG/2020/726 – M.K.K. Jithma – ER Diagram, Requirement Document, Table Create.
- TG/2020/749 – H.P.G.L.P. Jayathilake – Table Create and Insert data
- TG/2020/751 – N.W.V.B.S.B. Weragoda – Requirements document and Table Create.



## 17 **References**

- Lecture Notes.