Technical Report

Patient Management System

Submitted By

Eng. Aseel Al-Shadfan

Submitted To

Eng. Ahmad Batayneh

January 2023

Abstract

The Patient Management System (PMS) provides the benefits of streamlined operations, enhanced administration and control, superior patient care, and improved profitability. PMS is powerful, flexible, and easy to use, and it is designed and developed to deliver real, conceivable benefits to the hospital.

The project "Patient Management System" is based on database techniques. There are many areas where we keep records in the database, for which we use MS SQL software, which is one of the best and easiest to use. This project uses MS SQL Server Management Studio as the back-end software.

The Patient Management System is built to meet the specific requirements of small and mid-size hospitals across the globe. It contains all of the required modules, such as the patient treatment location, patient registration, clinics, patient diseases history, medicine details, diagnosis details, clinic doctors' schedule, users, user types, patient visit details, patient visit type, staff registration, bill payment details, patient visit rate, record modification, discharge details, and so on.

Keywords:
Patient Management System (PMS).
Microsoft Structured Query Language (MS SQL).

Acknowledgment

I would like to thank all those who are involved in this endeavor for their kind cooperation in its successful completion. At the outset, I express my gratitude for my capstone project to those who helped me efficiently complete this project.

I offer my special thanks to my project supervisor, ***Eng. Ahmad Batayneh***; without his help and support throughout this project, it would not have been this successful. His advice was invaluable throughout my project and the writing of this project report.

Also, I would like to thank my family members for their warmth, support, kindness, and patience. I'm thankful to all my friends who always advised and motivated me throughout the course.

Table of Contents

Contents

List of Figures

List of Tables

Chapter One

Introduction

1.1 Introduction

In the present era of globalization and advanced technology, the need for efficient record-keeping is very essential. As technology advances, manually maintaining information in different organizations becomes more difficult and inadequate.

When a patient visits a hospital and is admitted, he or she is termed an "inpatient." On the other hand, outpatients are patients who are not admitted but rather visit the hospital from their homes. The patient visits the hospital for diagnosis, treatment, or therapy and leaves as an outpatient without staying overnight; an inpatient may stay overnight or for several days, weeks, months, or even more than a year in some cases. The hospital is usually distinguished from other types of medical facilities by their ability to admit and care for inpatients; others are often described as clinics [3].

Hospitals are very important to society. Healthcare providers need to do their job effectively and efficiently. The number of investments in computers and hospital equipment has increased, but medical records are still on paper, which is cumbersome, bulky, and difficult to manage, especially when searching for information. A DBMS will tremendously outperform the manual method.

1.2 Project description.

The patient management system will manage inpatient hospital emergency visits and outpatient hospital clinic visits, for which they will arrange an appointment to visit the hospital clinics and access the system to book an appointment and see the doctor's schedule. The system will store outpatient and inpatient visit details. Each patient has a profile for his information and history of diseases. As for the emergency visits, the receptionists will insert the patient data into the system.

1.3 Problem statement.

Since hospitals are associated with the lives of common people and their day-to-day routines, I decided to work on this project.

The manual handling of the record is time-consuming and highly prone to error. The purpose of this project is to automate or make online the process of day-to-day activities like the admission of new patients, the discharge of patients, assigning a doctor, computing the bill, etc. I have tried my best to make the complicated process of the patient management system as simple as possible using the ER Model Relationship Technique. I have tried to design the database in such a way that users may not have any difficulty using it without much effort. The main goal of my project is to perform each hospital's activity digitally rather than manually, which is time-consuming, and to avoid issues such as more paper, more complexity, data confusion, no way to look into the patient's history, bad handwriting that may cause side effects, wrong medicine that may cause death, and dis-connectivity between (pharmacies, doctors, patients, and receptionists).

I am confident that this database (PMS) can be readily used by non-programming personnel to avoid human-handled chances of error.

Keyword: Entity Relationship (ER).

1.4 Objective

Hospitals are an essential part of our lives, providing the best medical facilities to people suffering from various ailments, which may be due to changes in climatic conditions, increased workload, emotional stress, etc. Hospitals must keep track of the day-to-day activities and records of their patients, doctors, nurses, and other staff personnel to keep the hospital running smoothly and successfully.

But keeping track of all the activities and their records on paper is very difficult and error-prone. It is also a very inefficient and time-consuming process, given the continuous increase in population and the number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient, and error-prone. It is also not economically or technically feasible to maintain these records on paper. Thus, I am keeping the workings of the manual system as the basis of my project.

The primary goal of my project is to provide a paperless hospital for up to 90% of documents. The system also provides excellent security of data at every level of user-system interaction and provides robust and reliable storage and backup facilities [15].

1.5 Scope

The Patient Management System (PMS) will be used in any hospital or clinic. They will get information from the patients and then store that data for future usage. The current system for hospital use is paper-based. It is too slow and cannot provide updated lists of patients within a reasonable time frame, increase the number of patients that can be treated accurately, make it easy for doctors to check and diagnose their patients correctly, let patients use the system to show their history of illness and all the patient's history in one place, increase the connectivity between all the members of the healthcare system, and generate reports accurately.

1.6 Discussion

The patient management system will manage inpatient hospital emergency visits, and outpatients can arrange an appointment to visit the hospital clinics and access the system to see the doctor's scheduled time and the available time slot, or they can call the receptionist in the clinic to book a visit. The system will store outpatient and inpatient visit details. Each patient has a profile for his information and history of diseases. As for the emergency visits, the receptionists will insert the patient data into the system. A patient will have a unique patient ID with a full description of his details, the disease, and what treatment is going on.

The doctor will handle patients; one doctor can treat more than one patient. Also, each doctor will have a unique ID. Doctors and patients will be related. There are some nurses for the maintenance of the hospital and patient care in the emergency room and clinic. Based on the visit details, a treatment bill will be generated, and the patient can pay with cash or a credit card. The visit type describes the type of patient, which is inpatient or outpatient. Inpatients go to the hospital emergency room, and outpatients go to the clinic.

- Inpatient visit procedure:

In hospital emergency, inpatients can visit the hospital without making an appointment to see any emergency doctor. The receptionist employees immediately insert their information and disease history. Doctors and nurses can diagnose their diseases, and doctors insert their diagnoses into the system. Based on the visit details and medication, the treatment bill will be generated.

- Outpatient visit procedure:

In the clinic, outpatients can book or cancel their visit, register their information, and also have permission to view the doctor's available time slots. They reserve a day and time at any clinic of their choice. Outpatients get treatment and medicine depending on their diagnosis and their disease history. Doctors can diagnose their diseases and insert their diagnoses into the system. Based on the visit details and medication, the treatment bill will be generated.

Hospital clinics are dentistry, otolaryngology, orthopedics, dermatology, pediatrics, and obstetrics. The schedule represents the scheduled doctor days, start and end times, as well as the average consultation time for each doctor. A doctor has a set of start and end timeslots which depend on his start and end times of work and his average consultation time.

Types of users are:

1. Administrator: Insert and manage system permissions and user accounts.
2. Receptionist: Insert inpatient and outpatient information details and their histories of diseases.
3. Patient: Book a visit date and time, insert their information, and their history of diseases.
4. Doctor: Insert diagnoses for inpatients and outpatients, and insert doctor information details.
5. Pharmacists: Insert patient medicine details.
6. Accountant: Insert patient treatment bills.

Notes:

- The main users are administrators, receptionists, doctors, pharmacists, and accountants.
- A sub-user is a patient.
- Users have permission to insert, delete, and update data.
- Patients can rate their visits to track staff performance.
- The system has several jobs that automatically execute them, such as:
  - ✓ The system will automatically delete all the canceled outpatient visits from the visit details table every Thursday after 4:00 PM.

1.7 Functional requirements.

The system should add the details, along with the automatically generated registration number of newly admitted patients, to the database. The system should search for existing patient details for a given registration number. The outpatient should arrange an appointment to visit the hospital clinics and should access the system to book an appointment and see the doctor's schedule.

1.8 Nonfunctional requirements.

- Usability: The system should be user-friendly and self-explanatory.
- Availability: The system should be available to the user 24 hours a day, seven days a week. The system shall be operational 24 hours a day, 7 days a week.
- Accuracy: The accuracy of the system should be limited by the speed at which its users use it.
- Access Reliability: The system shall provide 100% access reliability.

1.9 Summary

The patient management system is built for:

1. Ensure file security and record accuracy in the hospital emergency, and clinics.
2. Speed up the processing, sorting, and retrieval of information in the hospital emergency, and clinics.
3. Eliminate the stress encountered in the manual method.
4. Reduce time wasted when the patient is waiting for a staff member to search, sort, and retrieve their file.

Chapter Two
Project Management

2.1 Project planning and scheduling.

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined, and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables the identification of the critical path [1].

2.1.1 Methodology

I have used the iterative and incremental development model (IID) for my project development in Agile. The Agile model encompasses clusters of processes that are run in sequence over a set period of iteration. The customer is sent a feedback loop to validate the solution. It can be found, especially in the software development industry. The context has been adopted successfully. Likewise, this method can be used for patient management system development projects in healthcare [10].

Keyword: Iterative and Incremental Development (IID).

Agile methodology was selected for this project for the reasons listed below:

- The majority of the solution has been identified. The goal needs to be specific and measurable.
- There may be several deviations from the unidentified ranges of the solution.
- Allow stakeholders to be involved in developing the solution.
- Evaluating and validating the influence of alternatives.

For managing the impact of changing the system, an incremental and iterative approach is a good one. Incremental prototypes paired with agile tools help cover changing requirements, benefits and outputs more rapidly.

Fig.1: Agile Model.

2.1.2 Project management life cycle.

The project management life cycle has four phases. Each project life cycle phase is described alongwith the tasks needed to complete it [11].

The four phases are:

1. Inception Phase: It includes a higher-level look at the project scope, tangible andintangible requirements, and risks.
2. The Elaboration Phase: It takes the risks you identify in the Inception Phase and starts with the working architecture that meets non-functional requirements.
3. Construction Phase: Takes the architecture you put in place finished code that passes theiterative model already.
4. Transition Phase: It pushes the final product to production.

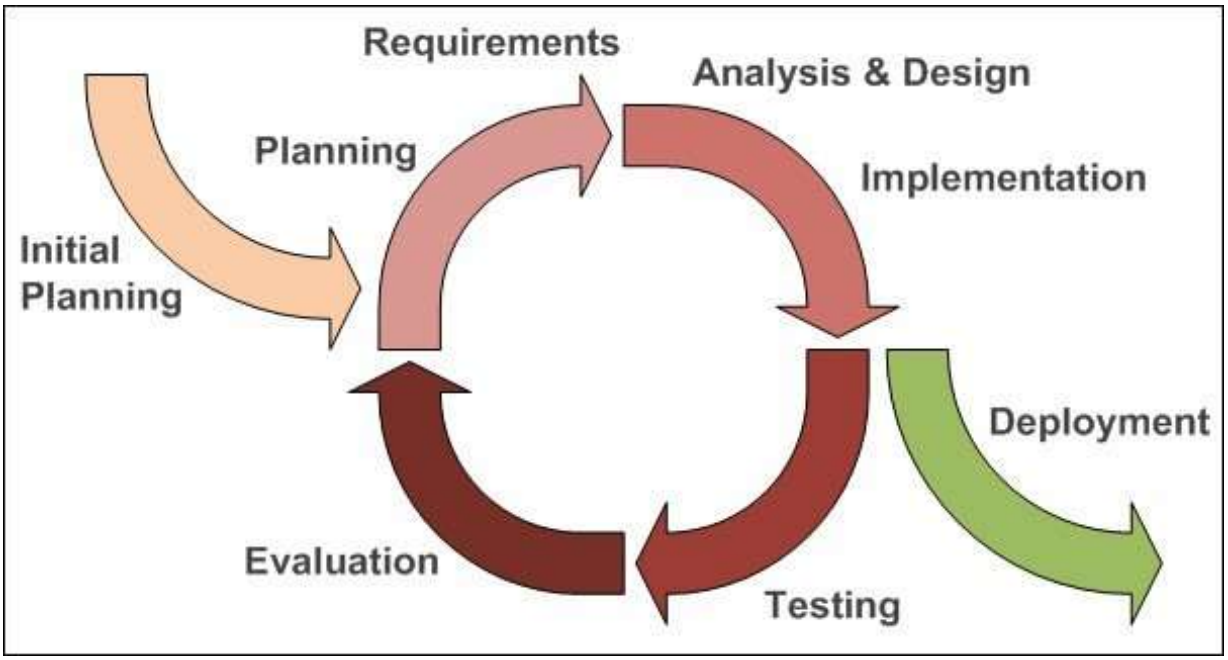


Fig.2: Iterative and Incremental Life Cycle.

2.1.3 Project plan.

Once I determine that the project is feasible, I undertake project planning. The table below describes how I planned my project.

	Task Name	Duration	Start	Finish
1	Planning	3 days	23/12/2022	26/12/2022
2	Design	2 days	27/12/2022	28/12/2022
3	Coding	2 days	29/12/2022	31/12/2022
4	Testing	5 day	01/01/2023	01/01/2023
5	Delivery	4 days	06/01/2023	10/01/2023

Table.1: Project Plan.

2.1.4 Schedule representation.

Scheduling the project tasks is an important project planning activity. It involves deciding which tasks will be taken on and when to schedule the project activities.

- Gantt chart:

Days	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Planning															
Design															
Coding															
Testing															
Delivery															

Table.2: Gantt chart.

Chapter Three

System Analysis

3.1 Background study.

System analysis is the separation of a substance into parts for study, implementation, and detailed examination.

Before designing any system, the nature of the business and the way it currently operates must be clearly understood. The detailed examination provides the specific data required during design to ensure that all the requirements are fulfilled. The investigation or study conducted during the analysis phase is largely based on the feasibility study. Rather, it would not be wrong to say that the analysis and feasibility phases overlap. The high-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. The analysis begins with system initialization and continues until system maintenance. Even after the successful implementation of the system, the analysis may play a role in the periodic maintenance and evolution of the system. One of the main causes of project failure is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is poor planning and system analysis [16].

3.2 System attributes.

3.1.1 Reliability: This system is a reliable product that produces fast and verified outputfrom all its processes.

3.1.2 Availability: This system will be available to use and help them carry out theiroperations conveniently.

3.1.3 Security: This system will be designed in a maintainable manner. It will be easy to incorporate new requirements into the individual modules.

4.3 Scope of working.

The Patient Management System (PMS) will be used in any hospital or clinic. They will get information from the patients and then store that data for future usage. The current system in use is paper-based. It is too slow and cannot provide updated lists of patients within a reasonable timeframe. The system intends to reduce overtime pay, increase the number of patients that can be treated accurately, make it easy for doctors to check and diagnose their patients correctly, let patients use the system to show their history of illness, keep all the patient's history in one place, increase connectivity between all the members of the healthcare system, and generate reports accurately.

Chapter Four

System Design

4.1 Database design.

Database design is the process of producing a detailed data model of the database. This data modelcontains all the needed logical and physical design choices and physical storage parameters neededto generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term "database design" can be used to describe many different parts of the design of an overalldatabase system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In the relational model, these are the tables and views. In an object database, the entities and relationships map directly to object classes and named relationships. However, the term "database design" could also be used to refer to the overall process of designing not just the base data structure but also the forms and queries used as part ofthe overall database application within the database management system [14].

4.2 E-R diagram of the patient management system.

An entity-relationship diagram (ERD) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion [5].

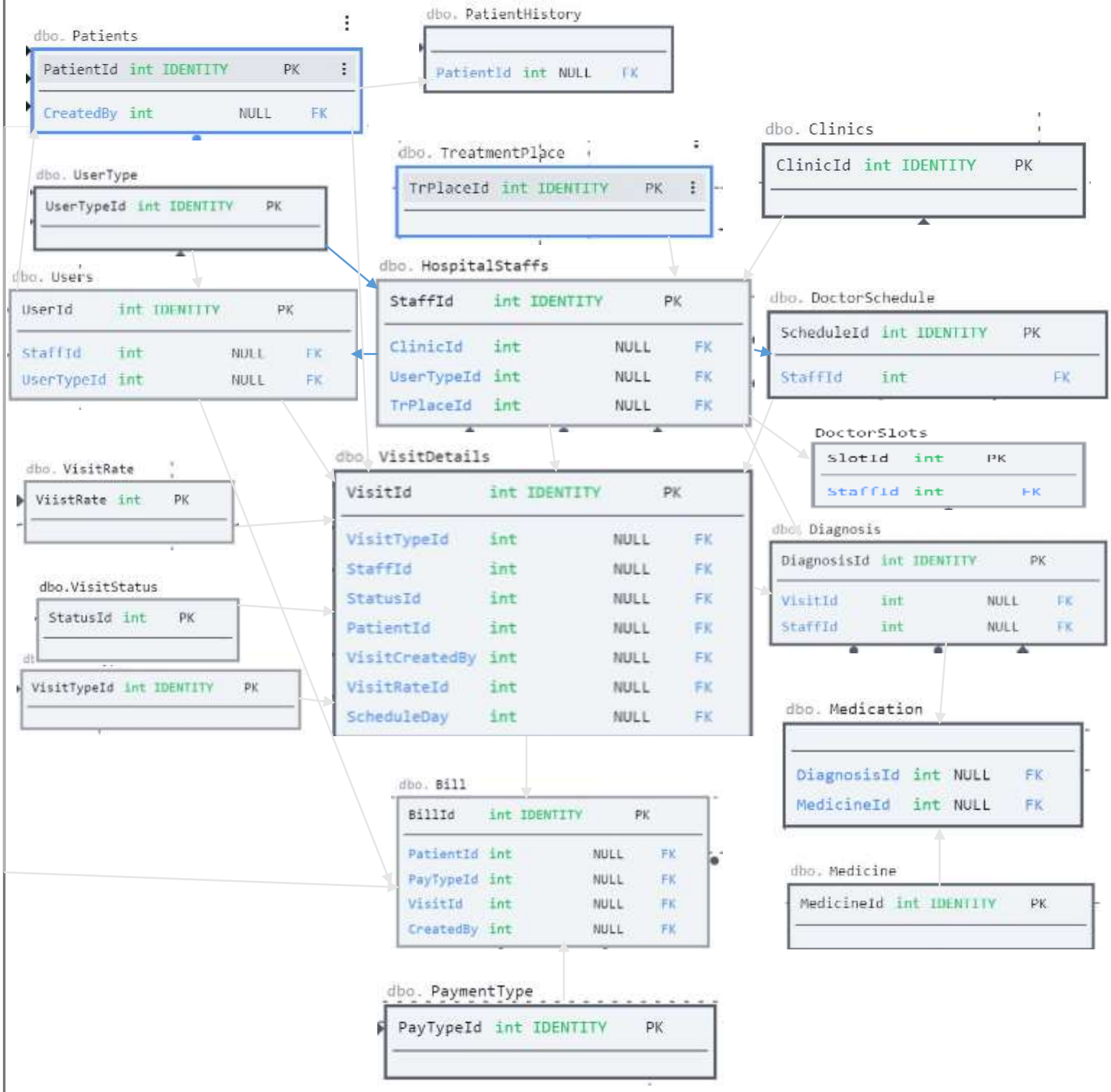


Fig. 3: E-R Diagram of the Patient Management System.

4.3 Database schema of the patient management system.

A database schema is the skeleton structure that represents the logical view of the entire database. It specifies how data is organized and how relationships between them are defined. It formulates all the constraints that are to be applied to the data.

A database schema can be broadly divided into two categories:

- Physical Database Schema: This schema pertains to the actual storage of data and its form of storage, like files, indices, etc. It defines how the data will be stored in secondary storage.
- Logical Database Schema: This schema defines all the logical constraints that need to be applied to the stored data. It defines tables, views, and integrity constraints.

4.4 Database relation types.

There are essentially three types of relationships [18]:

- **One-to-one (1-1)**: For every record in the primary table, there is one and only one record in the foreign table.

- **One-to-many (1-M):** For every record in the primary table, there are one or more related records in the foreign table.

- **Many-to-many (M-M):** For every record in the primary table, there are many related records in the foreign table, and for every record in the foreign table, there are many related records in the primary table.

When I designed the ER diagram, I used one-to-many and many-to-one relationships. I avoided using the many-to-many relationships because they generate uncertainty and data duplication.

4.5 List of entities (tables) of the patient management system.

1. Users table:

DESKTOP-TG0DVCL.P...ystem - dbo.Users

Column Name	Data Type	Allow Nulls
UserId	int	☐
UserName	varchar(50)	☐
Password	varchar(50)	☐
UserAFullName	nvarchar(200)	☐
UserEFullName	varchar(200)	☐
StaffId	int	☑
UserTypeId	int	☑
IsActive	bit	☐
		☐

Table.3: Users

- **Primary key:** User Id**. Foreign keys:** Staff Id, and User Type Id**.**

2. User type table:

DESKTOP-TG0DVCL....em - dbo.UserType

Column Name	Data Type	Allow Nulls
UserTypeId	int	☐
UserTypeDesc	varchar(50)	☐
		☐

Table.4: User Type.

- **Primary key:** User Type Id**.**

3. Hospital staff table:

DESKTOP-TG0DVCL.P...bo.HospitalStaffs

Column Name	Data Type	Allow Nulls
StaffId	int	☐
StaffAFullName	nvarchar(200)	☐
StaffEFullName	varchar(200)	☐
Email	varchar(100)	☐
ContactNo	char(12)	☐
Address	nvarchar(200)	☐
DOB	date	☐
Age	int	☐
Gender	char(1)	☐
SSN	char(9)	☐
Salary	decimal(15, 3)	☐
HireDate	date	☐
ClinicId	int	☑
IsActive	bit	☐
UserTypeId	int	☑
TrPlaceId	int	☑

Table.5: Hospital Staff.

- **Primary key:** Staff Id**. Foreign keys:** Staff Id, User Type Id, Treatment Place Id, and Clinic Id**.**

4. Visit details table:

ESKTOP-TG0DVCL.Pa...- dbo.VisitDetails | view visit.sql - not connect

Column Name	Data Type	Allow Nulls
VisitId	int	☐
VisitTypeId	int	☑
VisitCreated	date	☐
StartTime	time(7)	☐
EndTime	time(7)	☑
PatientId	int	☑
VisitCreatedBy	int	☑
StaffId	int	☑
VisitRateId	int	☑
ScheduleDay	int	☑
VisitStatus	int	☑
IsCome	char(1)	☑

Table.6: Visit Details.

Primary key: Visit Id**. Foreign keys:** Staff Id, Patient Id, Visit Created By, Visit Rate Id, Visit Status, and Schedule Day**.**

5. Visit type table:

DESKTOP-TG0DVCL.P...m - dbo.VisitRate

Column Name	Data Type	Allow Nulls
ViistRate	int	☐
VisitRateDesc	nchar(10)	☐
		☐

Table.7: Visit Type.

- **Primary key:** Visit Id.

6. Visit rate table:

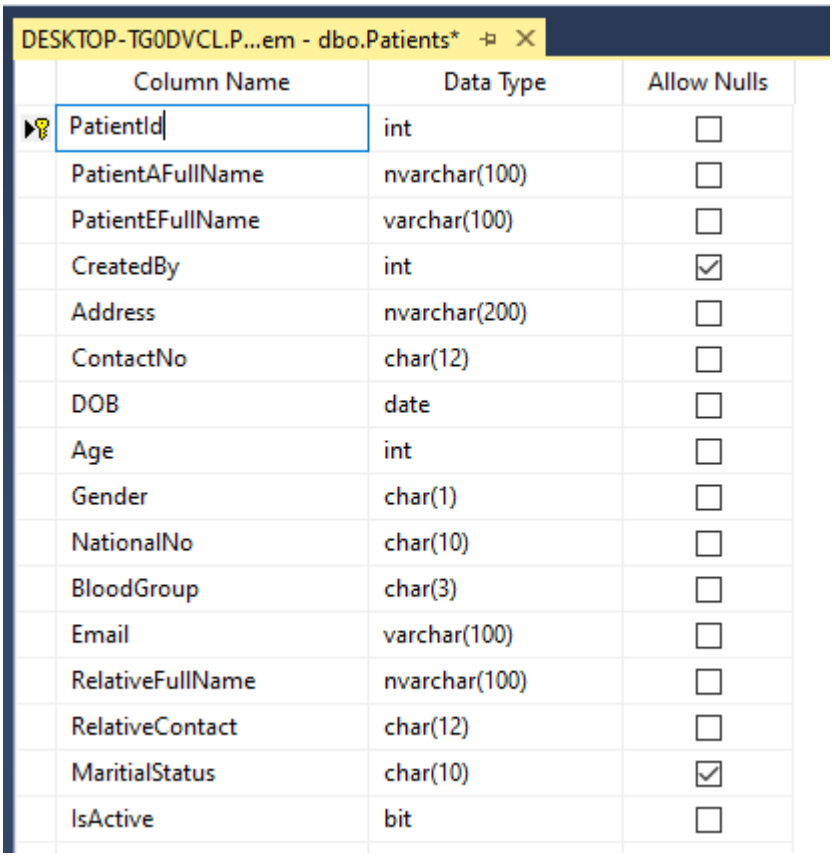
DESKTOP-TG0DVCL.P...m - dbo.VisitType

Column Name	Data Type	Allow Nulls
VisitTypeId	int	☐
VisitTypeDesc	varchar(20)	☐
		☐

Table.8: Visit Rate.

- **Primary key:** Visit Type Id.

7. Patients table:

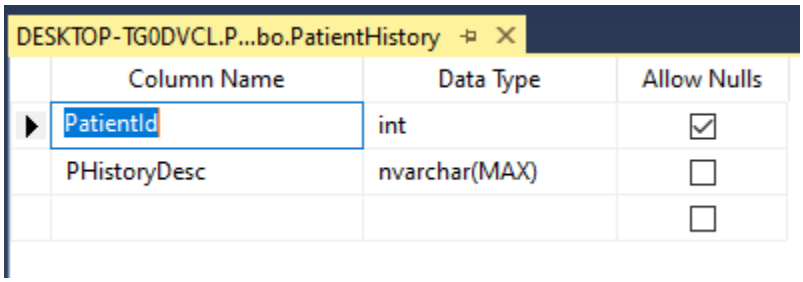


Column Name	Data Type	Allow Nulls
PatientId	int	☐
PatientAFullName	nvarchar(100)	☐
PatientEFullName	varchar(100)	☐
CreatedBy	int	☑
Address	nvarchar(200)	☐
ContactNo	char(12)	☐
DOB	date	☐
Age	int	☐
Gender	char(1)	☐
NationalNo	char(10)	☐
BloodGroup	char(3)	☐
Email	varchar(100)	☐
RelativeFullName	nvarchar(100)	☐
RelativeContact	char(12)	☐
MaritialStatus	char(10)	☑
IsActive	bit	☐

Table.9: Patients

- **Primary key:** Patient Id**. Foreign key:** Created By.

8. Patient history table:



Column Name	Data Type	Allow Nulls
PatientId	int	☑
PHistoryDesc	nvarchar(MAX)	☐
		☐

Table.10: Patient History

- **Foreign key:** Patient Id**.**

9. Clinics table:

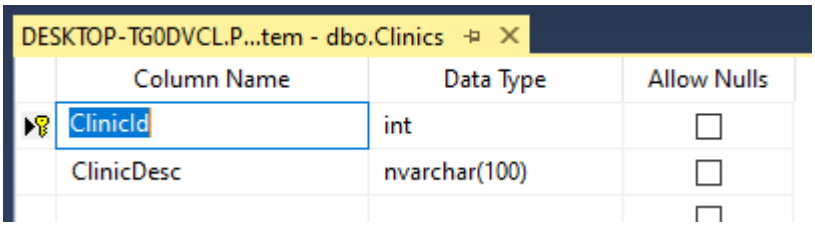


Table.11: Clinics

- **Primary key:** Clinic Id**.**

10. Bill table:

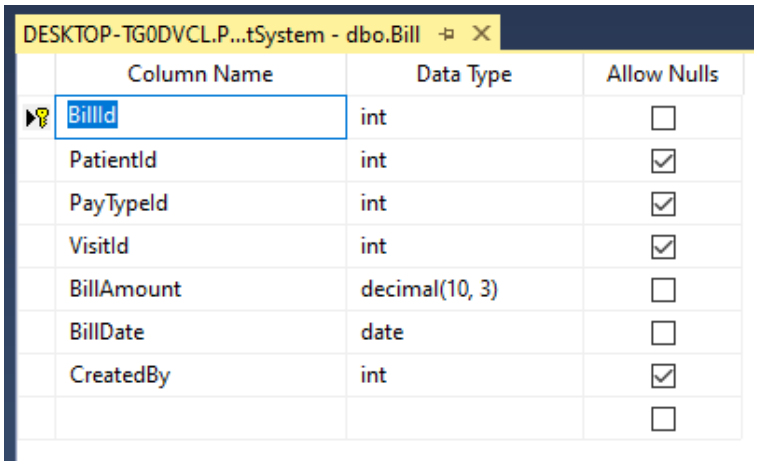


Table.12: Bill

- **Primary key:** Bill Id**. Foreign key:** Created By, Patient Id, Visit Id, and Pay Type Id.

11. Payment type table:

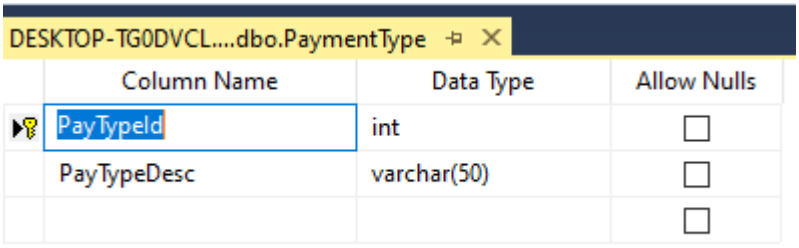


Table.13: Payment Type.

- **Primary key:** Pay Type Id**.**

12. Doctor schedule table:

DESKTOP-TG0DVCL....bo.DoctorSchedule | DESKTOP-TG0DVCL....bo.

Column Name	Data Type	Allow Nulls
ScheduleId	int	☐
DoctorSDate	date	☐
DocSDays	varchar(50)	☐
DocSStartTime	time(7)	☐
DocSEndTime	time(7)	☐
AvgCounsultingTime	int	☐
StaffId	int	☑
IsActive	bit	☐
		☐

Table.14: Doctor Schedule.

- **Primary key:** Schedule Id**. Foreign key:** Staff Id.

13. Doctor slots table:

DESKTOP-TG0DVCL.P...- dbo.DoctorSlot | SQLQuery7.sql - DES...(patie

Column Name	Data Type	Allow Nulls
SlotId	int	☐
StaffId	int	☐
StartSlot	time(7)	☐
EndSlot	time(7)	☐
IsActive	bit	☐

Table.15: Doctor Time Slots.

- **Primary key:** Slot Id**. Foreign key:** Staff Id.

14. Treatment place table:

DESKTOP-TG0DVCL....bo.TreatmentPlace* | DESKTOP-TG0DVCL.P...y

Column Name	Data Type	Allow Nulls
TrPlaceId	int	☐
TrPlaceDesc	varchar(50)	☐
PhoneNo	char(12)	☐
Email	varchar(50)	☐

Table.16: Treatment Place.

- **Primary key:** Treatment Place Id**.**

15. Diagnosis table:

DESKTOP-TG0DVCL....m - dbo.Diagnosis

Column Name	Data Type	Allow Nulls
DiagnosisId	int	☐
DiagDesc	nvarchar(MAX)	☐
DiagDate	date	☐
DiagAmount	decimal(5, 3)	☑
VisitId	int	☑
StaffId	int	☑
History	varchar(MAX)	☑
		☐

Table.17: Diagnosis

- **Primary key:** Diagnosis Id**. Foreign key:** Visit Id, and Staff Id.

16. Medication table:

DESKTOP-TG0DVCL.P...- dbo.Medication

Column Name	Data Type	Allow Nulls
DiagnosisId	int	☑
MedicineId	int	☑
Quantity	int	☐
		☐

Table.18: Medication

- **Foreign key:** Diagnosis Id, and Medicine Id.

17. Medicine table:

DESKTOP-TG0DVCL....em - dbo.Medicine

Column Name	Data Type	Allow Nulls
MedicineId	int	☐
MedicineDesc	varchar(100)	☐
UnitPrice	decimal(10, 3)	☐
IsActive	bit	☐
		☐

Table.19: Medicine

- **Primary key:** Medicine Id**.**

18. Visit Status:

DESKTOP-TG0DVCL.Pa...- dbo.VisitStatus | Report.sql - DESKTO...(pa

Column Name	Data Type	Allow Nulls
StatusId	int	☐
StatusDesc	varchar(50)	☑
		☐

Table.20: Visit Status.

- **Primary Key**: Status Id.

19. Data Log table:

DESKTOP-TG0DVCL....tem - dbo.DataLog

Column Name	Data Type	Allow Nulls
Id	int	☐
TableName	varchar(100)	☐
RowId	int	☐
OldValue	varchar(MAX)	☑
NewValue	varchar(MAX)	☑
ActionDate	datetime	☐
ActionBy	int	☐
ActionTypeId	int	☑
		☐

Table.21: Data Log.

- **Primary key:** Id**. Foreign key:** Action Type Id.

20. Action type table:

DESKTOP-TG0DVCL.P...- dbo.ActionType

Column Name	Data Type	Allow Nulls
ActionTypeId	int	☐
ActionTypeDesc	varchar(50)	☐
		☐

Table.22: Action type.

- **Primary key:** Action Type Id**.**

21. Audit trail log table:

DESKTOP-TG0DVCL.P...dbo.AuditTrailLog

Column Name	Data Type	Allow Nulls
Id	int	☐
AuditDescription	varchar(MAX)	☑
UserId	int	☐
AuditDate	datetime	☐
		☐

Table.23: Audit Trail Log.

- **Primary key:** Id**.**

22. Error log table:

DESKTOP-TG0DVCL....em - dbo.ErrorLog

Column Name	Data Type	Allow Nulls
ErrorId	int	☐
ErrorMessage	varchar(MAX)	☑
ErrorLocation	varchar(MAX)	☐
ErrorDate	datetime	☐
ErrorUser	int	☐
		☐

Table.24: Error Log.

- **Primary key:** Error Id**.**

Chapter Five

Reports

5.1 SQL Server Reporting Services (SSRS).

I have used the SQL Server reporting services for building the patient management system reports, which provide a set of on-premises tools and services that create, deploy, and manage paginated reports. A database report is the formatted result of database queries and contains useful data for decision-making and analysis. Most good business applications contain a built-in reporting tool; this is simply a front-end interface that calls or runs back-end database queries that are formatted for easy application usage. Formal reports may be used to convey information, analyze an issue or problem, and provide a recommended course of action [13].

5.1.1 Get all patient information.

```
-- Get all patient information

Create procedure SP_GetAllPatientsInfo

@LangIndicator char(2)
 as
 begin

 select p.PatientId ,
 case when @LangIndicator = 'AR' then p.PatientAFullName when @LangIndicator = 'EN' then p.PatientEFullName END as PatientName,
 case when @LangIndicator = 'AR' then u.UserAFullName when @LangIndicator = 'EN' then u.UserEFullName END as UserName  ,
 ut.UserTypeDesc,
 Address,ContactNo,DOB,Age,IIf(Gender = 'M','Male','Female') Gender,NationalNo,BloodGroup,Email,RelativeFullName,
 RelativeContact,MaritialStatus,p.IsActive
 from Patients p
 inner join Users u on p.CreatedBy = u.UserId
 inner join UserType ut on u.UserTypeId = ut.UserTypeId

 End

 exec SP_GetAllPatientsInfo 'AR'
```

Fig.4: Procedure: Get all patient information.

All Patient Information

ID	Patient Name	Created By	User Type	Address	Contact No	DOB	Age	Gender	National No	Blood Group	Email	Relative Name	Relative Contact	Marital Status	Is Active
1	Sarah Malkawi	Sarah Malkawi	Patient	Amman	962111111111	1/1/1994	28	F	111111111	A	Sarah@Patient.com	Ahamd Malkawi	962969962961	Single	True
2	Shahed Ahmad	Shahed Ahmad	Patient	Amman-Marka	962222222222	7/5/1982	40	F	111111112	O	Ahmad	Ahmad	962969962964	Married	True
3	Mohammad Kamal	Mohammad Kamal	Patient	Jarash	962333333333	10/15/1992	30	M	111111113	A+	Mohammed@Patient.com	Marwa Kamal	962969962966	Married	True
4	Saleem Alfaqeeh	Saleem Alfaqeeh	Patient	Irbid	962444444444	1/3/2002	20	M	111111114	B	Saleem@Patient.com	Amneh Alfqeeh	962969962968	Single	True
5	Israa Naeem	Israa Naeem	Patient	Amman-Jubiha	962555555555	12/31/2021	1	F	111111115	B	Israa@Patient.com	Essam Naeem	962969962910	Single	True
6	Fatmeh Ahmad	Amjad Saleem	Receptionist	Amman-Marka	962666666666	12/2/1977	45	F	111111116	B	Fatmeh@Patient.com	Faris Ahmad	962969962912	Married	True
7	Saleh Alsyouf	Ahmad Mohammed	Receptionist	Amman-Jubiha	962777777777	6/3/1967	55	M	111111117	O	Saleh@Patient.com	Hamza Alsyouf	962969962914	Married	True
8	Qusai Aljamal	Ahmad Mohammed	Receptionist	Amman	96288888888	12/5/2005	17	M	111111118	AB+	Qusai@Patient.com	Usaid Aljamal	962969962916	Single	True
9	Jaber Odeh	Amjad Saleem	Receptionist	Irbid	962999999999	4/13/2012	10	M	111111119	AB+	Jaber@Patient.com	Shadi Odeh	962969962918	Single	True
10	Samah Alissa	Ahmad Mohammed	Receptionist	Amman-Jubiha	962111111112	8/15/1947	75	F	111111110	A	Samah@Patient.com	Tuqa Alissa	962969962920	Married	True

Table.25: All patient info.

5.1.2 Get all patient history by patient ID.

```
create Procedure SP_GetPatientHistoryByPatId
@patientid int,
@LangIndicator char(2)
as
begin

 select case when @LangIndicator = 'AR' then p.PatientAFullName when @LangIndicator = 'EN' then p.PatientEFullName END as PatientName,
        case when @LangIndicator = 'AR' then u.UserAFullName when @LangIndicator = 'EN' then u.UserEFullName END as UserName  ,
        case when @LangIndicator = 'AR' then h.StaffAFullName when @LangIndicator = 'EN' then h.StaffEFullName END as DoctorName,
 v.VisitId,p.patientid,c.ClinicDesc,isnull('-',s.DocSDays) DocSDays,
 vt.VisitTypeDesc,vr.VisitRateDesc,isnull('-',vs.StatusDesc) StatusDesc ,p.Age,d.DiagDesc,tr.TrPlaceDesc , ut.UserTypeDesc
 from VisitDetails v
left join VisitStatus vs on v.VisitStatus = vs.StatusId
left join VisitType vt on v.VisitTypeId = vt.VisitTypeId
left join VisitRate vr on v.VisitRateId = vr.ViistRate
left join patients p on v.PatientId = p.PatientId
left join HospitalStaffs h on v.StaffId = h.StaffId
left join Diagnosis d on d.VisitId = v.VisitId
left join TreatmentPlace tr on tr.TrPlaceId = h.TrPlaceId
left join Users u on u.UserId = v.VisitCreatedBy
left join UserType ut on u.UserTypeId = ut.UserTypeId
left join DoctorSchedule s on v.ScheduleDay = s.ScheduleId
left join Clinics c on c.ClinicId = h.ClinicId
where v.PatientId = @patientid
 order by p.PatientId
 end
exec SP_GetPatientHistoryByPatId 1,'en'
```

Fig.5: Procedure: Get all the patient history by patient Id.

Patient visit history by ID

Patient Name	User Name	Doctor Name	Patient Id	Age	Created By	Visit Id	Clinic	Booked Day	Visit Type	Visit Rate	Status	Diagnosis	Treatment Place
Sarah Malkawi	Amjad Saleem	Shadi Alzgoul	1	28	Receptionist	18	Emergency	-	IN	Good	-	Check blood pressure and diabetes	Emergency
	Sarah Malkawi	Manal Khaled			Patient	1	Dentistry	Saturday	OUT	Good	Booked	Gingivitis	Clinic
		Musaab Naji				16	Dermatology	Thursday		Very Good		Iritic	
		Sultan Alsyouf				17	Obstetrics and Gynecology	Wednesday		Helpful		Sudden, severe abdominal or pelvic pain	
		Usaid Jamal				11	Otolaryngology	Monday		Very Good		Inflammation of the middle ear	

Table.26: Patient with Id = 1 visit history.

Patient visit history by ID

Patient Name	User Name	Doctor Name	Patient Id	Age	Visit Id	Created By	Clinic	Booked Day	Visit Type	Visit Rate	Status	Diagnosis	Treatment Place
سماح العيسى	أحمد محمد	شادي الزغول	10	75	10	Receptionist	Emergency	-	IN	Good	-	shortness of breath – dyspnea	Emergency
		مجاهد العمري			19					Bad		Paranasal sinuses	
	أمجد سليم	شادي الزغول			20					Helpful		Severe hypoglycemia	
		مجاهد العمري			21					Good		Dry eyes; increased tearing	
					22					Good		Reduction of Blood pressure	

Table.27: Patient with Id = 10 visits history.

5.1.3 Get the most frequent patients grouped by emergency and outpatient clinics.

```
--  Get Most FrequenT Patient CNT

create procedure SP_MostFrequenTPatientCNT
as
begin
select  count (*) CNT ,v.PatientId,p.PatientEFullName,vt.VisitTypeDesc
                from VisitDetails v
                inner join VisitType vt on v.VisitTypeId = vt.VisitTypeId
                inner join Patients p on  p.PatientId = v.PatientId
                where v.VisitTypeId =1
                group by v.PatientId,p.PatientEFullName,vt.VisitTypeDesc
       having  count (*) =
        (select top 1 count (*)
                from VisitDetails
                where VisitTypeId =1
                group by PatientId
                order by 1 desc)
union
select  count (*) CNT ,v.PatientId,p.PatientEFullName,vt.VisitTypeDesc
                from VisitDetails v
                inner join VisitType vt on v.VisitTypeId = vt.VisitTypeId
                inner join Patients p on  p.PatientId = v.PatientId
                where v.VisitTypeId =2
                group by v.PatientId,p.PatientEFullName,vt.VisitTypeDesc
       having  count (*) =
        (select top 1 count (*)
                from VisitDetails
                where VisitTypeId =2
                group by PatientId
                order by 1 desc)
end
```

Fig.6: Procedure: The most frequently seen patients.

Most Frequent Patient

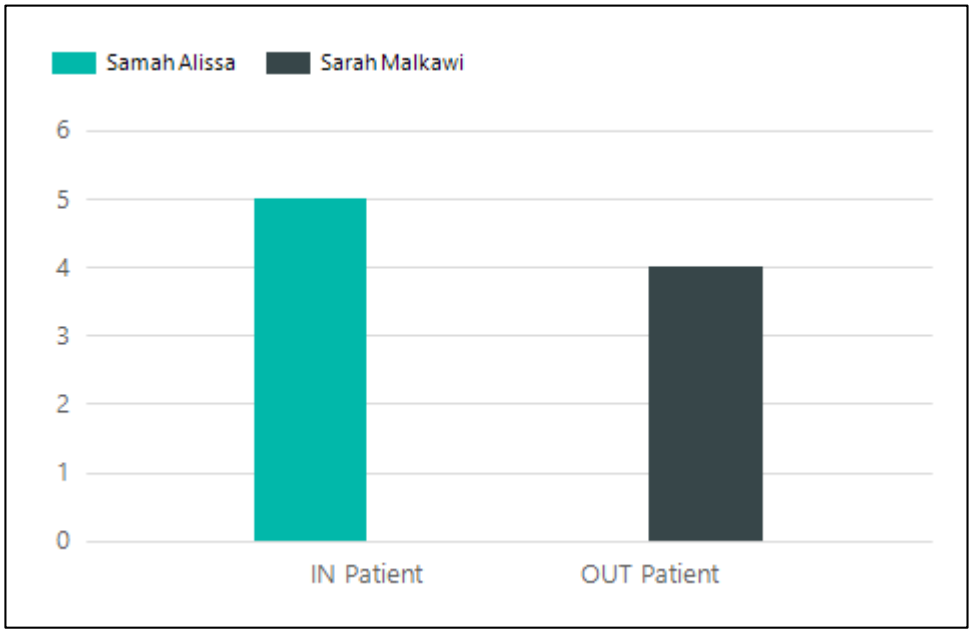


Fig.7: The most frequent inpatient and outpatient chart.

Visit Count	Patient Id	Patient Name	Visit Type
4	1	Sarah Malkawi	**OUT Patient**
5	**10**	**Samah Alissa**	**IN Patient**

Table.28: The most frequent inpatient and outpatient.

5.1.4 The total number of patients is divided by age group as shown below:
[1-10], [11-18], [19-25], [26-36], [37-48], [49-60], [>60)

```
-- Total patient by age

Create Procedure SP_GetPatientCntByRange
as
begin
 SELECT count(*) PatientCNT, *
 FROM
(
select
  case
  when age  between 1 and 10 then '1-10'
  when age  between 11 and 18 then '11-18'
  when age  between 19 and 25 then '19-25'
  when  age  between 26 and 36 then '26-36'
  when  age  between 37 and 48 then '37-48'
  when  age  between 49 and 60 then '49-60'
  when  age > 60 then 'Upper 60'

 END as Age_range
 from Patients
) t
group by Age_range
order by Age_range

end

exec SP_GetPatientCntByRange
```

Fig.8: Procedure: Total number of patients, separated by age.

Number of Patients by Age Group

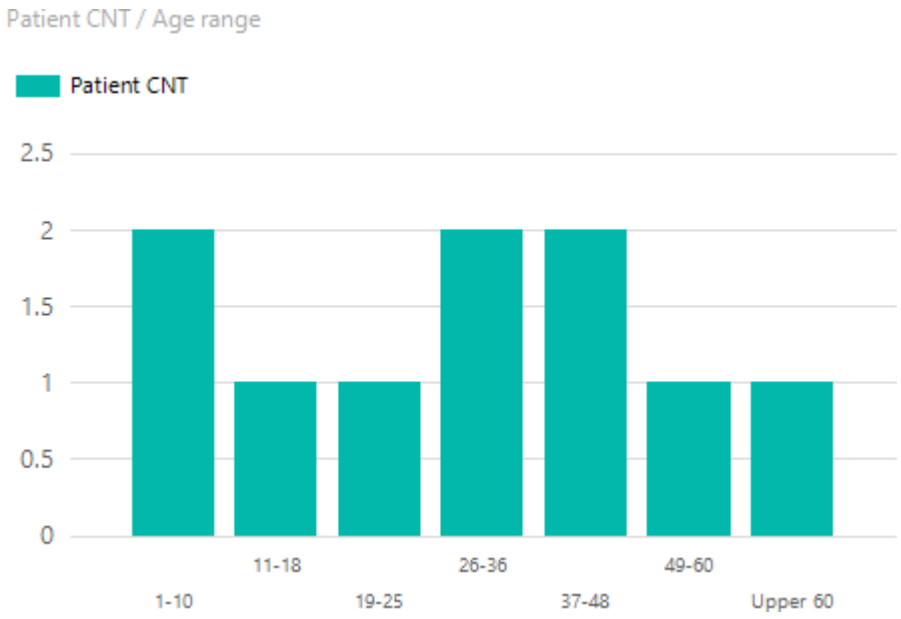


Fig.9: Chart: The total number of patients is divided by their age.

Patient CNT	Age Range
2	1-10
1	11-18
1	19-25
2	26-36
2	37-48
1	49-60
1	Upper 60

Table.29: The total number of patients is divided by their age.

5.2 PMS Stored Procedures.

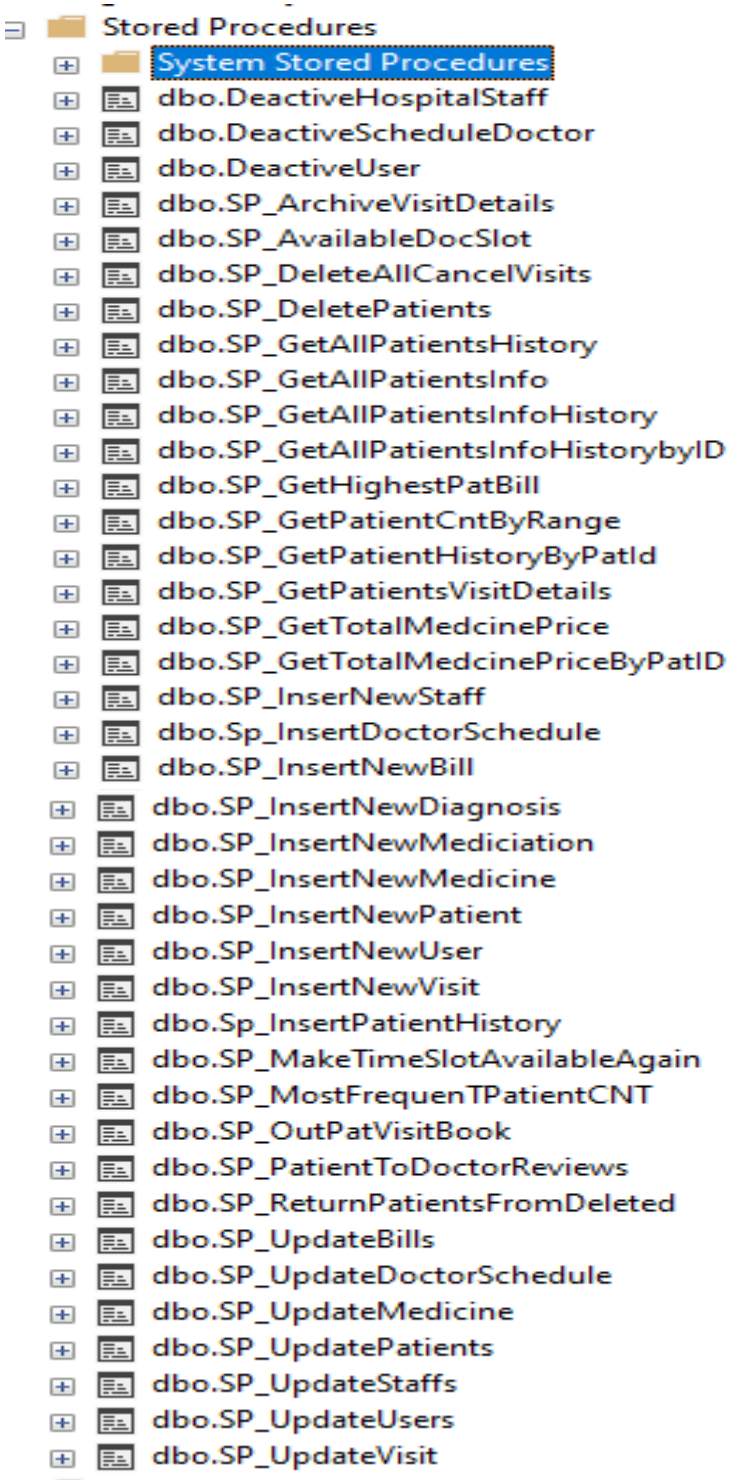


Fig.10: Stored procedures.

5.3 PMS Functions.

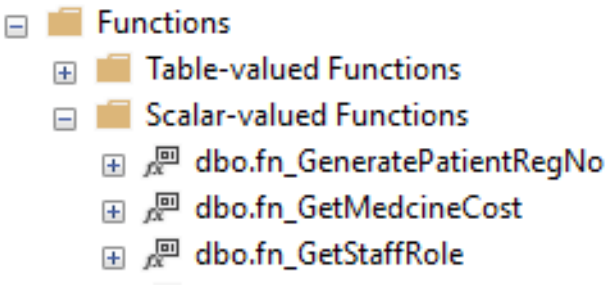


Fig.11: Functions

5.4 PMS Trigger.

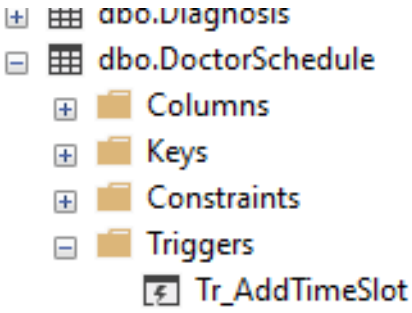


Fig.12: Trigger

5.5 PMS Views.

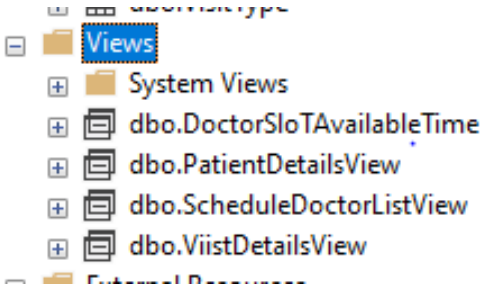


Fig.13: Views

5.6 PMS Jobs.

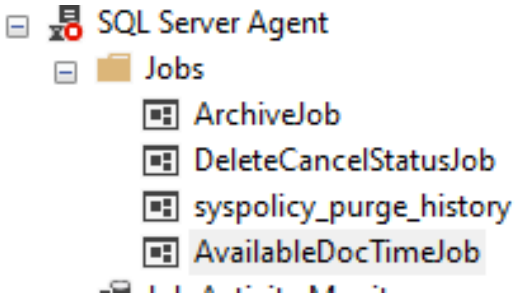


Fig.14: Jobs

Chapter Six

Conclusion

6.1 Conclusion

The Patient Management System (PMS) project is for digitizing the work in a hospital. It is a great improvement over the manual system. The digitization of the system has speeded up the process. The system takes care of all the requirements of an average hospital and is capable of providing easy and effective storage of information related to the patients that come up to the hospital. It generates reports and also allows you to search for the patient's information.

6.2 Recommendations

Since the data in the medical center are so essential that the management cannot do without them, I do suggest that any hospital should, as much as possible, ensure that:

- There are backups for all files for security reasons. Each office has a computer with large memory to run the system.
- Each of the offices should be networked to reduce the need to move files from one office to another and also to save time.
- An uninterruptible power supply (UPS) with an inverter should be provided for every computer to reduce the rate of hard disk or system failures.

6.3 Further enhancements.

The proposed system is the Patient Management System. I can improve this system by adding more features, such as a pharmacy system for medication stock details. Having such features allows users to enter more comments into the system. Add different types of rooms and wards that inpatients will stay in.

6.4 Limitation

- The size of the database increases day by day, increasing the load on the database and data maintenance activities.
- Training for simple computer operations is necessary for the users working on the system.

References

[1] The Project planning scheduling. [Online] available at https://www.educba.com/project-planning-and-scheduling/ [Accessed 23 Dec. 2022].

[2] Hospital management system. [Online] available at https://www.simplexhimes.com/hospital-management/?gclid=CjwKCAiAqt-dBhBcEiwATw-ggFsrJNtLoDsJP5GtuRKFYRy7GFlXC6zVo0s4-PJDkWv_TAvR5wDnWhoCIwUQAvD_BwE [Accessed 24 Dec. 2022].

[3] Outpatient and inpatient meaning. [Online] available at https://www.google.com/search?q=why+we+need+report+in+database%3F&oq=why+we+need+report+in+database%3F&aqs=chrome..69i57j33i160l3j33i15i22i29i30j33i22i29i30l2.14884j0j 15&sourceid=chrome&ie=UTF-8 [Accessed 25 Dec. 2022].

[4] Problems faced by patients in hospitals. [Online] available at

https://www.excel-medical.com/problems-faced-by-patients-in-hospitals/ [Accessed 27 Dec.2022].

[5] ER diagram. [Online] available at https://app.sqldbm.com/OpenSample/# [Accessed 27 Dec.2022].

[6] SQL coding. [Online] available at https://www.w3schools.com/sql/ [Accessed 28 Dec. 2022].

[7] MS SQL software. [Online] available at https://www.microsoft.com/en-us/sql-server/sql-server-downloads [Accessed 28 Dec. 2022].

[8] Online appointment booking. [Online] available at https://www.youtube.com/watch?v=npdKtvKnuZs

[Accessed 31 Dec. 2022].

[9] Types of clinics. [Online] available at https://www.healthline.com/health/types-of-health-clinics [Accessed 01 Jan. 2023].

[10] 5 Keys for Implementing Agile Database Methodologies. [Online] available at https://www.dbmaestro.com/blog/database-devops/implementing-agile-database-methodologies [Accessed 02 Jan. 2023].

[11] Incremental and iterative development strategy. [Online] available at https://colaninfotech.com/blog/agile-software-development-is-based-on-both-incremental- iterative/ [Accessed 03 Jan. 2023].

[12] Patient's disease and treatment. [Online] available at https://www.mayoclinic.org/ar/diseases-conditions [Accessed 04 Jan. 2023].

[13] Reporting database. [Online] available at https://martinfowler.com/bliki/ReportingDatabase.html [Accessed 05 Jan. 2023].

[14] Database Design. [Online] available at

https://en.wikipedia.org/wiki/Database_design[Accessed 06 Jan. 2023].

[15] Hospitals. [Online] available at https://www.ipl.org/essay/Essay-On-Importance-Of-Hospital-PKB4CJ22FCED6 [Accessed 26 Dec. 2022].

[16] System analysis. [Online] available at

https://en.wikipedia.org/wiki/Systems_analysis[Accessed 26 Dec. 2022].

[17] Report builder. [Online] available at https://learn.microsoft.com/en-us/sql/reporting- services/install-windows/install-report-builder?view=sql-server-ver16 [Accessed 06 Jan. 2023].

[18] Database relations. [Online] available at https://learn.microsoft.com/en-us/office/vba/access/concepts/structured-query-language/define-relationships-between-tables-using-access-sql [Accessed 08 Jan. 2023].

[19]Agile model. https://colaninfotech.com/blog/agile-software-development-is-based-on-both-incremental- iterative/

[20] Iterative and incremental model. https://www.google.com/search?q=iterative+and+incremental+life+cycle&sxsrf=ALiCzsZN708 kO7-At43gwSQ4EfOohkJ4g:1673012202071&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjvlY mQiLP8AhUgTaQEHc79BTEQ_AUoAXoECAEQAw&biw=1707&bih=821&dpr=0.8

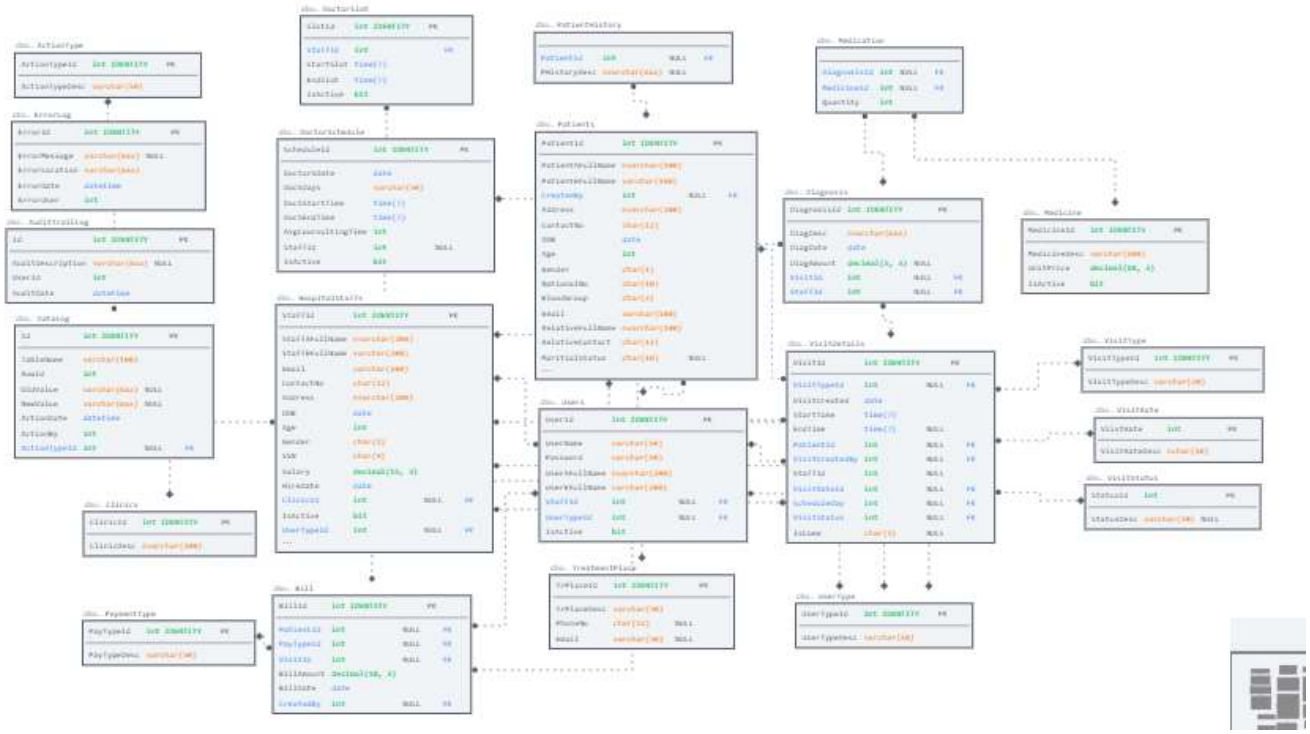
[21] Gantt chart model.
https://www.google.com/search?q=gant+chart&tbm=isch&ved=2ahUKEwihxsiRiLP8AhX1UqQ EHbusDoQQ2-cCegQIABAA&oq=gant+chart&gs_lcp=CgNpbWcQAzIHCCMQ6gIQJzIHCCMQ6gIQJzIHCC
MQ6gIQJzIHCCMQ6gIQJzIHCCMQ6gIQJzIHCCMQ6gIQJzIHCCMQ6gIQJzIHCCMQ6gIQJz
IHCCMQ6gIQJzIHCCMQ6gIQJ1DpEFjBPWDTQGgBcAB4AIABAIgBAJIBAJgBAKABAao BC2d3cy13aXotaW1nsAEKwAEB&sclient=img&ei=7SO4Y-GbDPWlkdUPu9m6oAg&bih=821&biw=1707#imgrc=10Cw1KNyrEpprM

[22] Tables by using MS SQL Management Studio. https://www.microsoft.com/en-us/sql-server/sql-server-downloads

[23] Project plan table.

http://www.eiminstitute.org/resource-portals/meta-data-management/meta-data-repository- project-plan-the-orientation-phase/

Appendix



- ER diagram with attributes:

https://app.sqldbm.com/SQLServer/Edit/p245992/#