CANCER PATIENT DATA MANAGEMENT CASE STUDY REPORT

Group 4

Student Names: Anjana Sajan Mini Kumari and Neelima Cherukuri

Executive summary

This use case study examines the development of a Cancer Patient Data Management System designed to tackle the intricate challenges of managing the lifecycle of cancer patient care. To address the complex issues involved in managing the lifetime of cancer patient care, this use case study looks at the creation of a cancer patient data management system. The lack of integrated systems in the current healthcare system causes inefficient care coordination, treatment delays, and less than ideal health outcomes. Through the integration of patient medical histories, treatment plans, and outcomes into a single platform, this project seeks to offer a comprehensive solution. The system's features include better patient monitoring and follow-up, enhanced provider coordination, and comprehensive reporting and analytics for treatment efficacy.

Through constructing a centralized database that unifies patient medical histories, treatment plans, and outcomes into a single platform, this study seeks to offer a comprehensive solution. To obtain practical insights and help healthcare providers monitor treatment effectiveness and improve patient care, this study also entails writing analytical queries.

I. Introduction

Coordinating the care of cancer patients involves many challenges, such as maximizing the efficacy of treatment and fostering collaboration among healthcare providers. To avoid inefficiencies in patient care, postponed treatments, and less than ideal health outcomes, there is a critical need for integrated systems to manage these intricate requirements.

Problem Statement: Managing the complete lifecycle of cancer patient care presents numerous challenges, including the coordination of care among various healthcare providers, tracking, and optimizing treatment effectiveness, and maintaining comprehensive patient records. Currently, there is a lack of integrated systems capable of handling these complex requirements effectively, which often leads to inefficiencies in patient care, delayed treatments, and suboptimal health outcomes.

Goal

Integrate patient medical histories, treatment plans, and outcomes.

- Improve coordination among healthcare providers.
- Enhance patient follow-up and monitoring.
- Provide robust reporting and analytics for treatment efficacy.
- Ensure compliance with healthcare regulations and data privacy standards.

Requirements: The system must integrate patient medical histories, treatment plans, outcomes, and support robust data analytics.

II. Conceptual Data Modeling

The conceptual modeling phase of this study involved designing a comprehensive data model using Enhanced Entity-Relationship (EER) and Unified Modeling Language (UML) diagrams. Using Unified Modeling Language (UML) and Enhanced Entity-Relationship (EER) diagrams, a comprehensive data model was designed during the conceptual modeling phase of this study. The key entities in the cancer patient data management system—patients, oncologists, nurses, treatment plans, medications, appointments, and medical records—were identified using these tools. Common features were retrieved where appropriate to streamline the overall structure and produce a more streamlined and effective model that appropriately captures the complexity of managing cancer patient data.

Entities and Attributes:

Patients - Attributes: PatientID (PK), Name, Age, Gender, ContactInfo, DiagnosisDetails, TreatmentHistory

Oncologists - Attributes: OncologistID (PK), Name, Specialization, ContactInfo

Nurses - Attributes: NurseID (PK), Name, Qualifications, AssignedWard

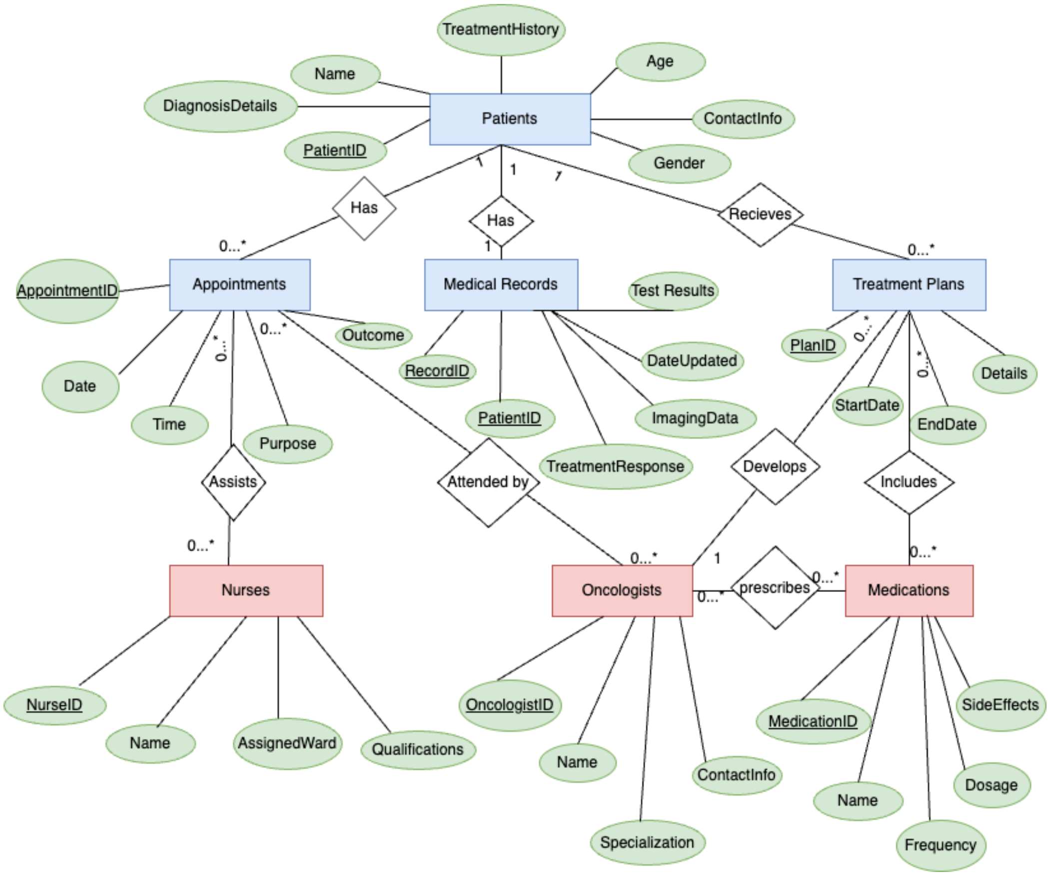
Treatment Plans - Attributes: PlanID (PK), StartDate, EndDate, Details

Medications - Attributes: MedicationID (PK), Name, Dosage, Frequency, SideEffects

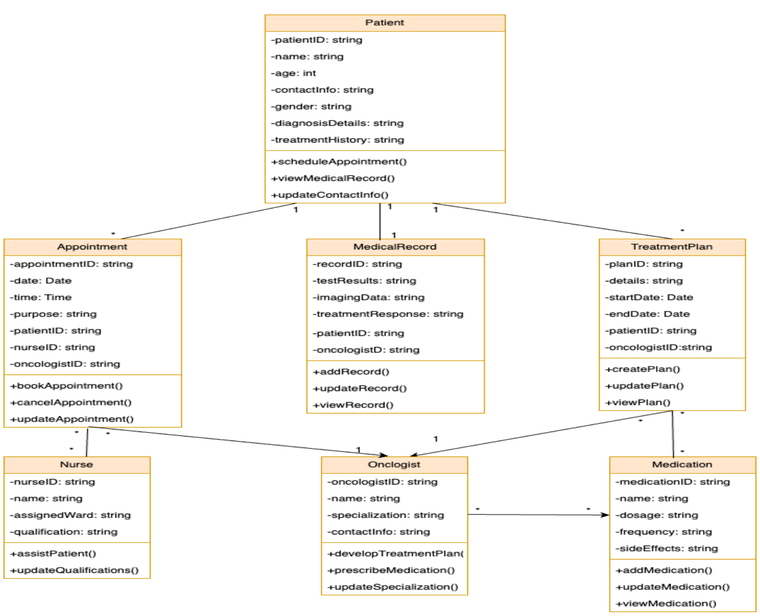
Appointments - Attributes: AppointmentID (PK), Date, Time, Purpose, Outcome

Medical Records - Attributes: RecordID (PK), PatientID (FK), DateUpdated, TestResults, ImagingData, TreatmentResponse

EER : Cancer Patient Data Management



UML : Cancer Patient Data Management



III. Mapping Conceptual Model to Relational Model

Relational Schema:

1. **Patients Table**

PatientID (PK), Name, Age, Gender, ContactInfo, DiagnosisDetails, TreatmentHistory

2. **Oncologists Table**

OncologistID (PK), Name, Specialization, ContactInfo

3. **Nurses Table**

NurseID (PK), Name, Qualifications, AssignedWard

4. **Treatment Plans Table**

PlanID (PK), StartDate, EndDate, Details, PatientID (FK), OncologistID (FK)

5. **Medications Table**

MedicationID (PK), Name, Dosage, Frequency, SideEffects

6. **Appointments Table**

AppointmentID (PK), Date, Time, Purpose, Outcome, PatientID (FK), OncologistID (FK), NurseID (FK)

7. **Medical Records Table**

RecordID (PK), PatientID (FK), DateUpdated, TestResults, ImagingData, TreatmentResponse

IV. Implementation of Relational Model via MySQL and NoSQL

MySQL Implementation:

The implementation of the relational model for this study was carried out using MySQL, a robust and widely-used relational database management system.. Using a series of relational tables, the conceptual model was translated to represent the various key entities, including patients, oncologists, nurses, treatment plans, medications, appointments, and medical records. To create relationships between these tables and guarantee referential integrity

Tables created as per the relational schema.

```sql
use cancer_patient_management;
CREATE TABLE Patients (
    PatientID INT PRIMARY KEY,
    Name VARCHAR(255),
    Age INT,
    Gender VARCHAR(50),
    ContactInfo VARCHAR(255),
    DiagnosisDetails TEXT,
    TreatmentHistory TEXT
);
CREATE TABLE MedicalRecords (
    RecordID INT PRIMARY KEY,
    PatientID INT,
    DateUpdated DATE,
    TestResults TEXT,
    ImagingData BLOB,
    TreatmentResponse TEXT,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
);
CREATE TABLE Medications (
    MedicationID INT PRIMARY KEY,
    PatientID INT,
    Name VARCHAR(255),
    Dosage VARCHAR(100),
    Frequency VARCHAR(100),
    SideEffects TEXT,
    FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)
    );
```

Dataset creation

To generate the data for this study, we utilized the Faker library in Python, a powerful tool for creating synthetic data that mimics real-world information. We created realistic datasets for all the important entities—patients, oncologists, nurses, treatment plans, prescription drugs, appointments, and medical records—using Faker. The characteristics and connections outlined in the conceptual model were carefully incorporated into each dataset, guaranteeing that the information was complete and indicative of real-world healthcare situations. Following generation, the data was exported into CSV files that corresponded to each relational table. The data was then seamlessly integrated into the database by using the Import Wizard in MySQL Workbench to import these CSV files into MySQL. By ensuring that the relational model was filled with reliable, consistent data, this procedure allowed for thorough testing and analysis of the system's functionalities.

Python code:

```python
]

# Generate Patients data
patients = []
for i in range(1, num_records + 1):
    patients.append({
        "PatientID": i,
        "Name": fake.name(),
        "Age": random.randint(18, 90),
        "Gender": random.choice(["Male", "Female", "Non-binary"]),
        "ContactInfo": fake.phone_number(),
        "DiagnosisDetails": random.choice(cancer_diagnoses),
        "TreatmentHistory": random.choice(treatments)
    })

# Generate Oncologists data
oncologists = []
for i in range(1, num_records + 1):
    oncologists.append({
        "OncologistID": i,
        "Name": fake.name(),
        "Specialization": random.choice(["Medical Oncology", "Surgical Oncology", "Radiation Oncology","Pediatric Oncology","Gynecologic Oncology"
        "ContactInfo": fake.phone_number()
    })

# Generate Nurses data
nurses = []
for i in range(1, num_records + 1):
    nurses.append({
        "NurseID": i,
        "Name": fake.name(),
        "Qualifications": random.choice(["Registered Nurse (RN)", "Bachelor of Science in Nursing (BSN)", "Master of Science in Nursing (MSN)",
    "Oncology Certified Nurse (OCN)", "Advanced Oncology Certified Nurse (AOCN)",
    "Certified Pediatric Hematology Oncology Nurse (CPHON)", "Certified Registered Nurse Anesthetist (CRNA)",
    "Nurse Practitioner (NP)", "Clinical Nurse Specialist (CNS)", "Certified Nurse Midwife (CNM)",
    "Advanced Cardiovascular Life Support (ACLS) Certified", "Basic Life Support (BLS) Certified",
    "Palliative Care Certification", "Chemotherapy/Biotherapy Certification", "Geriatric Nurse Certification"]),
        "AssignedWard": random.choice(["Ward 1", "Ward 2", "Ward 3", "Ward 4", "Ward 5",
    "Ward 6", "Ward 7", "Ward 8", "Ward 9", "Ward 10",
    "Ward 11", "Ward 12", "Ward 13", "Ward 14", "Ward 15"])
    })

# Generate Medications data
medications_data = []
for i in range(1, num_records + 1):
    medications_data.append({
        "MedicationID": i,
        "PatientID": random.randint(1, num_records),
        "Name": random.choice(medications),
        "Dosage": f"{random.randint(1, 500)} mg",
        "Frequency": random.choice([ "Once Daily", "Twice Daily", "Three Times Daily", "Four Times Daily",
    "Every 4 Hours", "Every 6 Hours", "Every 8 Hours", "Every 12 Hours",
    "Once Weekly", "Twice Weekly", "Every Other Day", "As Needed (PRN)",
    "Before Meals", "After Meals", "At Bedtime"]),
        "SideEffects": random.choice(side_effects)
    })

# Generate TreatmentPlans data
treatment_plans = []
for i in range(1, num_records + 1):
    treatment_plans.append({
        "PlanID": i,
        "StartDate": fake.date_this_decade(),
        "EndDate": fake.date_between(start_date="today", end_date="+6m"),
        "Details": f"{random.randint(3, 8)} cycles of {random.choice(treatments)}",
        "PatientID": random.randint(1, num_records),
        "OncologistID": random.randint(1, num_records)
    })
```

Analytical Queries

1.Ranking Cancer Diagnoses by Prevalence: This query ranks cancer diagnoses by the number of cases, helping to identify the most prevalent types of cancer within the patient population. By understanding which cancer types are most common, healthcare providers can prioritize resources, research, and treatment plans accordingly.

Query

```
#1
SELECT
    DiagnosisDetails,
    COUNT(*) AS CaseCount,
    RANK() OVER (ORDER BY COUNT(*) DESC) AS PrevalenceRank
FROM
    Patients
GROUP BY
    DiagnosisDetails;
```

Output

DiagnosisDetails	CaseCount	PrevalenceRank
Skin Cancer	428	1
Prostate Cancer	396	2
Stage IV Pancreatic Cancer	394	3
Brain Tumor	394	3
Lymphoma	391	5
Leukemia	388	6
Colorectal Cancer	384	7
Stage III Colon Cancer	383	8
Stage II Lung Cancer	383	8
Melanoma	378	10
Ovarian Cancer	376	11
Stage I Breast Cancer	373	12
Pancreatic Cancer	332	13

2. Analyzing Oncologist Performance Based on Treatment Outcomes: This query evaluates the performance of oncologists by counting the number of patients they have treated and the corresponding treatment outcomes (successful vs. failed treatments).

Query

```sql
#2
SELECT
    o.Name AS Oncologist,
    o.Specialization,
    COUNT(m.RecordID) AS NumberOfPatients,
    SUM(CASE WHEN m.TreatmentResponse = 'Positive Response' THEN 1 ELSE 0 END) AS SuccessfulTreatments,
    SUM(CASE WHEN m.TreatmentResponse = 'No Response' THEN 1 ELSE 0 END) AS FailedTreatments
FROM
    Oncologists o
JOIN
    MedicalRecords m ON m.PatientID IN (
        SELECT
            PatientID
        FROM
            TreatmentPlans
        WHERE
            OncologistID = o.OncologistID
    )
GROUP BY
    o.OncologistID;
```

__Output__

Oncologist	Specialization	NumberOfPatients	SuccessfulTreatme...	FailedTreatments
Jason Adams	Thoracic Oncology	2	1	1
David Ruiz	Urologic Oncology	3	1	1
Sara Lopez	Breast Oncology	2	1	0
Brandon Hall	Breast Oncology	4	0	3
Derek Ramirez	Surgical Oncology	2	0	2
Morgan Miller	Urologic Oncology	2	0	1
Elizabeth Shaffer	Surgical Oncology	5	1	2
Bryan Sandoval	Dermatologic Oncology	4	2	0
Noah Cooke	Surgical Oncology	6	2	0
Michael Fowler	Medical Oncology	3	0	1
Jared Miller	Dermatologic Oncology	1	0	1
Hannah Reyes	Neuro-Oncology	2	0	0
Ann Jackson	Gynecologic Oncology	3	1	2
Kimberly Mend…	Dermatologic Oncology	4	1	2
Travis Jennings	Dermatologic Oncology	3	1	1
Curtis Vaughn	Urologic Oncology	2	1	0
Kimberly Williams	Gynecologic Oncology	2	1	0
Suzanne Robbins	Pediatric Oncology	2	1	0
Melissa Wright	Neuro-Oncology	3	2	0
Richard Rice	Gynecologic Oncology	5	2	3
Dennis Zavala	Surgical Oncology	1	0	0
Christine Hall	Pediatric Oncology	1	1	0
Jeffrey Benson	Surgical Oncology	7	1	4
Matthew Stone…	Dermatologic Oncology	2	0	1
Mr. Scott Jeffer…	Radiation Oncology	5	0	3
Jennifer Hill	Thoracic Oncology	6	1	1
Andrew Young	Thoracic Oncology	1	0	0

3. Evaluating Patient Care Coordination Across Multiple Oncologists and Nurses: This query evaluates the level of care coordination by counting the number of distinct oncologists and nurses involved in treating a patient and correlating that with the treatment success rate.

__Query__

```sql
#3
SELECT
    p.PatientID,
    p.Name AS PatientName,
    COUNT(DISTINCT t.OncologistID) AS OncologistsInvolved,
    COUNT(DISTINCT a.NurseID) AS NursesInvolved,
    AVG(CASE WHEN m.TreatmentResponse = 'Positive Response' THEN 1.0 ELSE 0.0 END) AS SuccessRate
FROM
    Patients p
JOIN
    TreatmentPlans t ON p.PatientID = t.PatientID
JOIN
    MedicalRecords m ON p.PatientID = m.PatientID
LEFT JOIN
    Appointments a ON p.PatientID = a.PatientID
GROUP BY
    p.PatientID, p.Name
HAVING
    COUNT(DISTINCT t.PlanID) > 1
ORDER BY
    SuccessRate DESC;
```

__Output__

PatientID	PatientName	OncologistsInvolved	NursesInvolved	SuccessRate
20	Jordan Wagner	3	3	1.00000
32	Laura Brown	3	3	1.00000
44	Leah Davis	3	3	1.00000
79	David Benton	4	0	1.00000
86	Jennifer Ruiz	3	0	1.00000
109	Tyler Little	2	2	1.00000
126	Sandra Singh	2	1	1.00000
162	Mark Wright	2	1	1.00000
172	Anna Smith	2	3	1.00000
190	Amanda Taylor	3	2	1.00000
198	Mr. Daniel Haas	5	1	1.00000
201	John Fernandez	4	1	1.00000
214	Diana Knight	2	1	1.00000
221	Mrs. Tracy Shea MD	4	1	1.00000
241	Adam Smith	2	0	1.00000
286	Brittany Gomez	2	2	1.00000
295	Benjamin Lindsey	2	2	1.00000
304	Larry Brewer	2	1	1.00000
320	Matthew Martinez	2	0	1.00000

NoSQL Implementation (MongoDB)

After installing MongoDB on the local server, the system was integrated with Studio 3T, a MongoDB management tool, to facilitate easier management and querying of data. CSV files containing the data for entities such as Patients, Oncologists, Nurses, Treatment Plans, Medications, Appointments, and Medical Records were directly imported into MongoDB through Studio 3T. This setup allowed for the creation of a dynamic and scalable NoSQL database environment

Analytical Queries

1. **Identifying Patients Requiring Immediate Attention:** This query finds patients with a negative treatment response and sorts them by the most recent test results. This helps healthcare providers identify patients needing immediate attention.

```
db.MedicalRecords.aggregate([
    { $match: { TreatmentResponse: "Negative Response" } },
    {
        $lookup: {
            from: "Patients",
            localField: "PatientID",
            foreignField: "PatientID",
            as: "PatientDetails"
        }
    },
    { $unwind: "$PatientDetails" },
    { $project: { _id: 0, PatientID: 1, PatientName:
"$PatientDetails.Name", DiagnosisDetails:
"$PatientDetails.DiagnosisDetails", TestResults: 1,
TreatmentResponse: 1 } },
    { $sort: { DateUpdated: -1 } }
])
```

MedicalRecords > TestResults

PatientID	TestResults	TreatmentResponse	PatientName	DiagnosisDetails
3218	Inconclusive	Negative Respor	Emily White	Stage IV Pancreatic Cancer
13	Negative	Negative Respor	Gary Juarez	Stage I Breast Cancer
1828	Severe Abnorma	Negative Respor	Lindsey Oconno	Ovarian CancerColorectal CancerSkin CancerPancreatic CancerBrain Tumor
2683	Worsening	Negative Respor	Matthew Anders	Stage II Lung Cancer
713	Negative	Negative Respor	Sierra Peterson	Stage I Breast Cancer
4719	Severe Abnorma	Negative Respor	Mathew Gibson	Stage I Breast Cancer
1060	MRI: Progressior	Negative Respor	Daniel Bolton	Lymphoma
2016	Elevated	Negative Respor	Brad Goodman	Melanoma
845	Elevated	Negative Respor	Ross Neal	Stage III Colon Cancer
462	Pending	Negative Respor	Arthur Brown	Leukemia
124	Improved	Negative Respor	Theresa Collins	Stage III Colon Cancer

2. **Analyzing the Success Rates of Treatment Plans by Oncologist:** This query calculates the success rate of treatment plans managed by each oncologist. It helps in assessing the effectiveness of oncologists and can be used for performance evaluations and quality control.

```
db.TreatmentPlans.aggregate([
    {
        $lookup: {
            from: "MedicalRecords",
            localField: "PatientID",
            foreignField: "PatientID",
            as: "TreatmentOutcomes"
        }
    },
    { $unwind: "$TreatmentOutcomes" },
    { $match: { "TreatmentOutcomes.TreatmentResponse": "Positive Response" } },
    {
        $group: {
            _id: "$OncologistID",
            successfulTreatments: { $sum: 1 },
            totalTreatments: { $count: {} }
        }
    },
    {
        $lookup: {
            from: "Oncologists",
            localField: "_id",
            foreignField: "OncologistID",
            as: "OncologistDetails"
        }
    },
    { $unwind: "$OncologistDetails" },
    {
        $project: {
            OncologistName: "$OncologistDetails.Name",
            Specialization: "$OncologistDetails.Specialization",
            ContactInfo: "$OncologistDetails.ContactInfo",
            successRate: { $multiply: [{ $divide: ["$successfulTreatments", "$totalTreatments"] }, 100] }
        }
    },
    { $sort: { successRate: -1 } }
])
```

TreatmentPlans > OncologistName

_id	OncologistName	Specialization	ContactInfo	successRate
1163	Tracy Wolfe	Thoracic Oncolo	+1-646-489-70	100.0
3055	Stephanie Olson	Breast Oncology	224.298.4553	100.0
3778	Lisa Barron	Dermatologic Or	731.390.7993x8	100.0
3537	Gregory Harris	Hematologic On	+1-739-687-64	100.0
2011	Michael Lawrenc	Gynecologic Onc	5685716448	100.0
565	Eric Cobb	Dermatologic Or	+1-749-735-42	100.0
1910	Patricia Dickersc	Surgical Oncolog	234-840-2188x	100.0
373	Christopher Jarv	Neuro-Oncology	001-281-464-0	100.0
1320	James Ramirez	Hematologic On	246-884-3525	100.0
3231	Willie Gardner	Surgical Oncolog	992.395.7255	100.0
437	Jennifer King	Pediatric Oncolo	(370)419-2681x	100.0
1468	Angela Cabrera	Gynecologic Onc	6607615986	100.0
3260	Mark Sanders	Breast Oncology	240-889-6472	100.0
3519	Jacob Powell	Surgical Oncolog	001-245-601-18	100.0
3364	Dana Clark	Dermatologic Or	932.423.2495	100.0

V. Database Access via R or Python

To demonstrate access to the MySQL database, we connected the database to RStudio using the DBI and RMySQL packages. The dbConnect function was used to establish the connection, which gave us the ability to communicate with the database directly from the R environment. After connecting, we ran SQL queries to get datasets out of the database with an emphasis on obtaining crucial data pertinent to the treatment of cancer patients. Following the generation of visualizations using these datasets, data-driven decision-making was made with new insights.

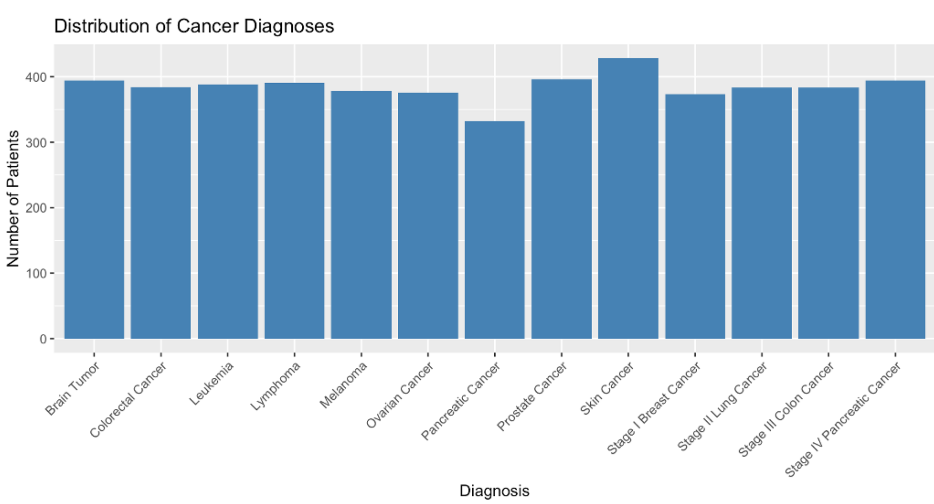
```r
on <- dbConnect(RMySQL::MySQL(),
                dbname = "cancer_patient_management",
                host = "127.0.0.1",
                port = 3306,
                user = "root",
                password = "*************")

# Test the connection by listing tables
dbListTables(con)
library(ggplot2)

# Load the dataset
Patients <- read.csv("/Users/anjanasajanminikumari/Documents/Patients.csv")
Appointments <- read.csv("/Users/anjanasajanminikumari/Documents/Appointments.csv")
MedicalRecords <-read.csv("/Users/anjanasajanminikumari/Documents/MedicalRecords.csv")
Oncologists <- read.csv("/Users/anjanasajanminikumari/Documents/Oncologists.csv")
Medications <- read.csv("/Users/anjanasajanminikumari/Documents/Medications.csv")
Nurses <- read.csv("/Users/anjanasajanminikumari/Documents/Nurses.csv")
TreatmentPlans <- read.csv("/Users/anjanasajanminikumari/Documents/TreatmentPlans.csv")
```

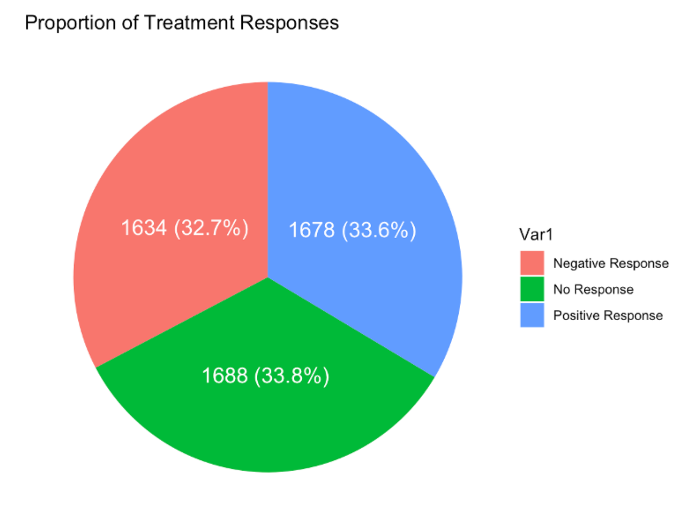
Visualizations using R-Studio

1) Using the ggplot2 package in RStudio, a visualization was made to examine the distribution of cancer diagnoses among patients. An important summary of the distribution of cancer diagnoses among patients is given by this, which also offers insightful information about the frequency of different cancer types in the patient population. Healthcare professionals can better allocate resources, prioritize research, and customize treatment plans to meet the unique needs of the patient population by identifying the most prevalent cancer types by analyzing this distribution.



2. This pie-chart depicts the percentage of different treatment responses among cancer patients, offering valuable information about how patients are responding to various treatment regimens. Healthcare professionals can assess the efficacy of current treatment approaches, spot trends in

patient reactions, and modify treatments as necessary to enhance results by examining the distribution of treatment responses.



VII. Summary and Recommendations

Summary

This case study explores the creation of a Cancer Patient Data Management System, which unifies treatment plans, appointments, medications, and medical histories into a single database to simplify the complex process of managing cancer patient care. The system uses MySQL for strong database management and the Faker library in Python for realistic data generation. This ensures data integrity and complies with healthcare regulations. The study also shows how the system can be used practically using RStudio, which retrieves and visualizes data to give useful insights into patient diagnoses and treatment outcomes. By enabling healthcare professionals to make well-informed, data-driven decisions, these visual analytics improve patient outcomes and maximize treatment plans.

Advantages

1) Improved Coordination of Care: The system guarantees that all healthcare providers have access to current and comprehensive information by centralizing patient data, including medical histories, treatment plans, and appointments. This improves coordination and results in more efficient patient care.

2) Enhanced Treatment Monitoring: By continuously monitoring treatment responses, the system helps healthcare professionals to promptly determine the efficacy of various treatments and make necessary modifications to improve patient outcomes.

3) Data-Driven Decision Making: The system offers actionable insights that assist healthcare professionals in making decisions based on real-time data, improving patient care and resource allocation. It does this by integrating analytical tools and visualizations.

4) Flexibility and Scalability: The utilisation of RStudio and MySQL enables scalability, which enables the system to accommodate expanding datasets and be modified over time to accommodate evolving healthcare requirements.

Shortcomings

1) Complexity of Implementation: Smaller healthcare facilities with limited IT resources may find it difficult to develop and maintain such a comprehensive system due to the elevated level of technical expertise and resources required.

2) Prospect of Data Overload: As a result of the massive amounts of data being produced and stored, there is a chance that there will be data overload, which makes it impossible to effectively manage and analyze the information and may even mask important insights.

Recommendations

1) Frequent Data Audits: To guarantee the timeliness and accuracy of the data in the system, conduct regular data audits. This will facilitate more trustworthy decision-making and help preserve data integrity.

2) Advanced Analytics and AI Integration: Combined with artificial intelligence (AI) capabilities, advanced analytics and AI can help manage massive data sets, spot trends, and produce predictive insights that can improve patient care and treatment results.