PROJECT REPORT



**ACE ENGINEERING COLLEGE, GATHKESAR**

***(An Autonomous Institute under JNTU, Hyderabad)***

**INFORMATION TECHNOLOGY**

EBox Project:Online Therapy Platform

**BY TEAM NO 40:** P.MANSI(21AG1A1241) G.VARALAXMI(21AG1A1218)

N.NIKHITHA(21AG1A1239)

**CONTENTS :**

* Abstract
* Introduction
* Functional Requirement
* ER Diagram
* Queries To Create DataBase
* Class/UML Attributes
* All Class Java File Codes
* Challenges List

**Abstract**

The Online Therapy System is a web-based platform designed to provide mental health support and counseling services remotely. With the increasing demand for accessible and convenient mental health care, especially in light of recent global challenges, such as the COVID-19 pandemic, online therapy systems have become crucial tools for connecting individuals with qualified therapists from the comfort of their own homes.

This project aims to develop a comprehensive online therapy system that offers a range of features to facilitate therapy sessions, including scheduling appointments, conducting video sessions, exchanging secure messages, accessing resources, and tracking progress. The system caters to both therapists and patients, providing each user with personalized experiences tailored to their needs.

Key functionalities of the Online Therapy System include user authentication and authorization, profile management, session scheduling, secure messaging, video conferencing capabilities, session assessments, prescription management, payment processing, and review and rating systems. The system prioritizes user privacy and security by implementing robust encryption protocols and compliance with healthcare data protection regulations.

By leveraging the power of technology, the Online Therapy System aims to break down barriers to mental health care by offering convenient, affordable, and accessible therapy services to individuals worldwide. Through its user-friendly interface and comprehensive features, the system seeks to improve the overall mental well-being of its users and promote a more inclusive approach to mental health care.

**Introduction**

In recent years, there has been a significant shift in the way mental healthcare services are delivered, with a growing emphasis on accessibility, convenience, and flexibility. Traditional barriers to accessing therapy, such as geographical constraints, scheduling conflicts, and stigma, have prompted the development of innovative solutions aimed at democratizing mental health support. One such solution is the Online Therapy Platform (OTP), a digital platform designed to connect individuals with qualified therapists and counselors through secure and convenient virtual channels.

The OTP project represents a timely and essential endeavor aimed at harnessing the power of technology to address the evolving needs of individuals seeking mental health support. By leveraging the ubiquity of the internet and advancements in telecommunication technologies, the OTP aims to overcome traditional barriers to therapy, making quality mental healthcare accessible to a broader demographic, regardless of geographical location or time constraints.

The primary objective of the OTP project is to develop a comprehensive and user-centric online therapy platform that offers a range of features and functionalities to facilitate therapeutic interactions between therapists and clients. Through the seamless integration of communication tools, scheduling mechanisms, session documentation, and payment processing systems, the OTP seeks to streamline the therapy process, enhance the therapeutic experience, and ultimately improve mental well-being outcomes for its users.

**Key Components of the OTP Project:**

1. User Interface and Experience Design: The OTP project prioritizes user experience design, ensuring that the platform is intuitive, user-friendly, and accessible to individuals of all ages and backgrounds. A clean and visually appealing interface, coupled with responsive design principles, enhances engagement and usability across different devices and screen sizes.

2. Therapist Matching and Onboarding: The OTP incorporates sophisticated algorithms and matching algorithms to pair clients with therapists based on factors such as expertise, specialization, availability, and therapeutic approach. Therapist onboarding processes are streamlined to ensure that only licensed and qualified professionals are registered on the platform, fostering trust and credibility among users.

3. Secure Communication Channels: Privacy and confidentiality are paramount in therapy settings, and the OTP project prioritizes the implementation of robust security protocols to safeguard sensitive information exchanged between therapists and clients. Secure messaging, video conferencing, and data encryption mechanisms ensure the integrity and confidentiality of therapeutic interactions.

4. Appointment Scheduling and Management: The OTP facilitates seamless appointment scheduling and management, allowing clients to book sessions at their convenience and providing therapists with tools to efficiently manage their calendars. Automated reminders and notifications help minimize no-shows and ensure session adherence.

5. Session Documentation and Progress Tracking: Comprehensive session documentation features enable therapists to record session notes, treatment plans, and progress updates within the platform. Clients have access to their session history and progress reports, empowering them to track their therapeutic journey and collaborate actively in their treatment process.

6. Payment Processing and Billing: The OTP streamlines payment processing and billing procedures, offering multiple payment options and secure transaction processing mechanisms. Automated invoicing, billing reminders, and reimbursement support simplify financial transactions for both therapists and clients.

Through the development and implementation of the Online Therapy Platform project, we aim to revolutionize the delivery of mental healthcare services, making therapy more accessible, convenient, and effective for individuals worldwide. By leveraging the power of technology to bridge gaps in traditional therapy models, the OTP project seeks to promote mental well-being, reduce stigma, and empower individuals to lead healthier and more fulfilling lives.

**Functional Requirements**

1. User Registration and Authentication:

- Clients and therapists should be able to register accounts on the platform.

- User authentication mechanisms, such as email verification and password hashing, should be implemented to ensure security.

2. User Profiles:

- Users should be able to create and manage their profiles, including personal information, therapy preferences, and contact details.

- Therapist profiles should include qualifications, areas of expertise, and availability.

3. Therapist Matching:

- The platform should provide algorithms for matching clients with therapists based on criteria such as specialization, availability, and therapeutic approach.

4. Appointment Scheduling:

- Clients should be able to view therapist availability and schedule appointments based on their preferences.

- Therapists should have access to a calendar system for managing their schedules and accepting or rejecting appointment requests.

5. Communication Tools:

- Secure messaging functionality should allow clients and therapists to communicate asynchronously between sessions.

- Video conferencing capabilities should enable real-time virtual therapy sessions with high-quality audio and video.

6. Session Documentation:

- Therapists should be able to document session notes, treatment plans, and progress updates securely within the platform.

- Clients should have access to their session history and treatment information for reference.

7. Payment Processing:

- The platform should support secure payment processing for therapy sessions, including multiple payment methods and automated invoicing.

- Billing features should provide therapists with tools for tracking payments and managing client invoices.

8. Prescription Management (if applicable):

- Therapists should have the ability to generate electronic prescriptions for medication management within the platform.

- Prescription records should be securely stored and accessible to authorized users.

9. Progress Tracking and Reporting:

- The platform should facilitate tracking of client progress and treatment outcomes over time.

- Therapists should be able to generate progress reports and treatment summaries for clients.

10. Review and Rating System:

- Clients should have the option to provide feedback and ratings for therapy sessions, enabling continuous quality improvement.

- Therapists should be able to view and respond to client reviews, fostering accountability and transparency.

11. Privacy and Security:

- The platform should adhere to strict privacy and security standards, including data encryption, user confidentiality, and compliance with healthcare regulations (e.g., HIPAA).

- Measures should be in place to protect sensitive information and prevent unauthorized access.

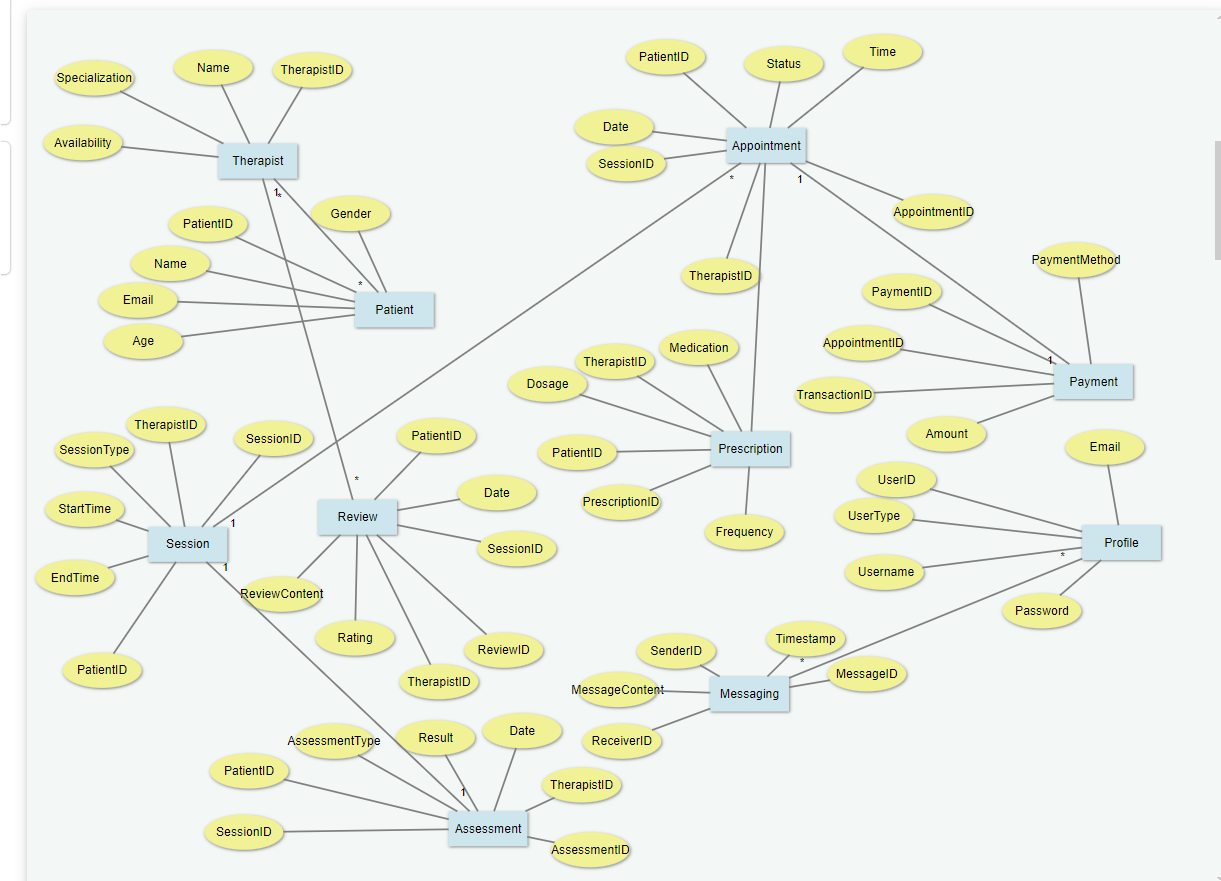
12. Accessibility and Usability:

- The platform should be accessible to users with diverse abilities and needs, complying with accessibility standards (e.g., WCAG).

- User interfaces should be intuitive, responsive, and optimized for usability across different devices and screen sizes.

These functional requirements form the foundation of the Online Therapy Platform project, ensuring that the platform effectively addresses the needs of both clients and therapists while prioritizing security, usability, and privacy

**ER Diagram**

****

**Queries To Create DataBase**

CREATE TABLE Profile (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

UserType VARCHAR(50),

Username VARCHAR(100),

Email VARCHAR(100),

Password VARCHAR(100)

);

CREATE TABLE Therapist (

TherapistID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Specialization VARCHAR(100),

Availability TEXT

);

CREATE TABLE Patient (

PatientID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100),

Email VARCHAR(100),

Age INT,

Gender VARCHAR(10)

);

CREATE TABLE Session (

SessionID INT PRIMARY KEY AUTO\_INCREMENT,

TherapistID INT,

PatientID INT,

StartTime DATETIME,

EndTime DATETIME,

SessionType VARCHAR(50),

FOREIGN KEY (TherapistID) REFERENCES Therapist(TherapistID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE Appointment (

AppointmentID INT PRIMARY KEY AUTO\_INCREMENT,

TherapistID INT,

PatientID INT,

SessionID INT,

Date DATE,

Time TIME,

Status VARCHAR(50),

FOREIGN KEY (TherapistID) REFERENCES Therapist(TherapistID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),

FOREIGN KEY (SessionID) REFERENCES Session(SessionID)

);

CREATE TABLE Prescription (

PrescriptionID INT PRIMARY KEY AUTO\_INCREMENT,

TherapistID INT,

PatientID INT,

Medication VARCHAR(100),

Dosage VARCHAR(50),

Frequency VARCHAR(50),

FOREIGN KEY (TherapistID) REFERENCES Therapist(TherapistID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

AppointmentID INT,

Amount DECIMAL(10, 2),

PaymentMethod VARCHAR(50),

TransactionID VARCHAR(100),

FOREIGN KEY (AppointmentID) REFERENCES Appointment(AppointmentID)

);

CREATE TABLE Assessment (

AssessmentID INT PRIMARY KEY AUTO\_INCREMENT,

SessionID INT,

TherapistID INT,

PatientID INT,

AssessmentType VARCHAR(50),

Result TEXT,

Date DATE,

FOREIGN KEY (SessionID) REFERENCES Session(SessionID),

FOREIGN KEY (TherapistID) REFERENCES Therapist(TherapistID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE Messaging (

MessageID INT PRIMARY KEY AUTO\_INCREMENT,

SenderID INT,

ReceiverID INT,

MessageContent TEXT,

Timestamp DATETIME,

FOREIGN KEY (SenderID) REFERENCES Profile(UserID),

FOREIGN KEY (ReceiverID) REFERENCES Profile(UserID)

);

CREATE TABLE Review (

ReviewID INT PRIMARY KEY AUTO\_INCREMENT,

TherapistID INT,

PatientID INT,

SessionID INT,

Rating INT,

ReviewContent TEXT,

Date DATE,

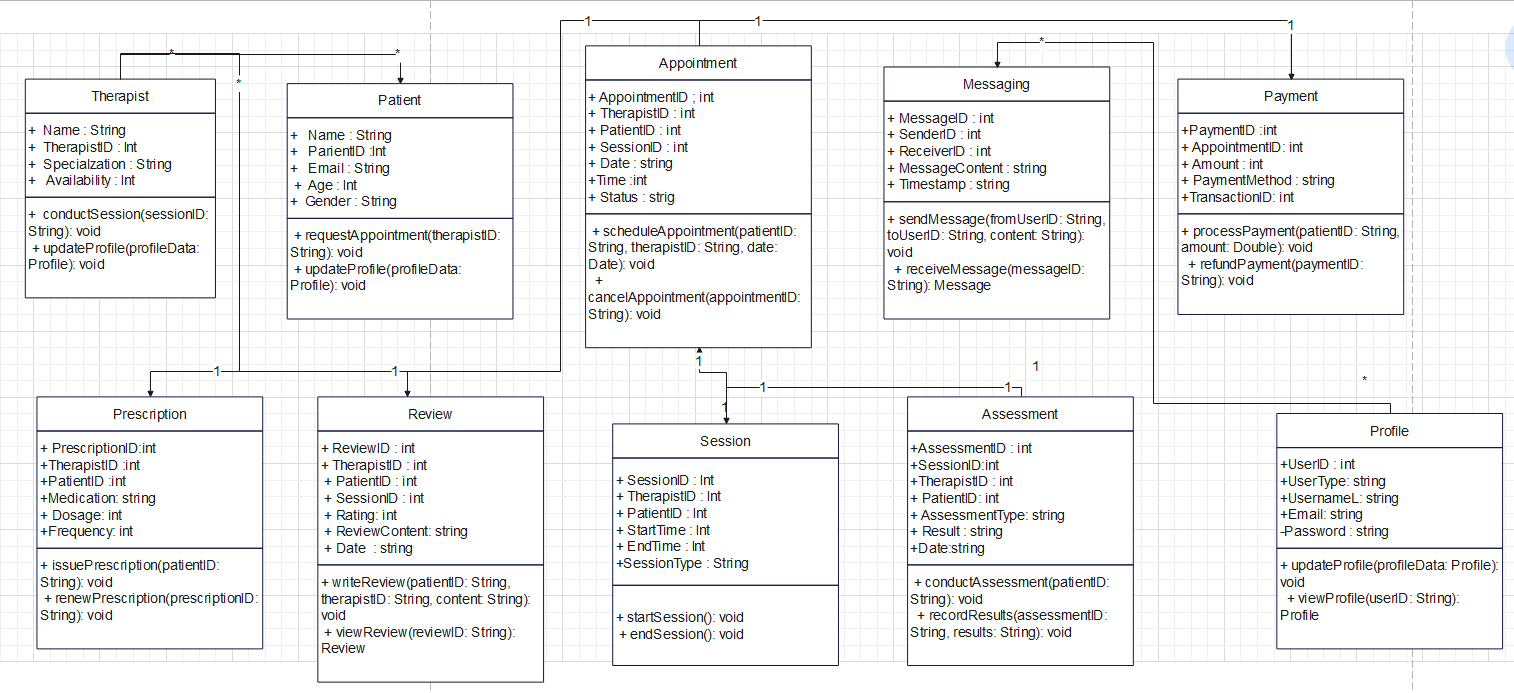
FOREIGN KEY (TherapistID) REFERENCES Therapist(TherapistID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),

FOREIGN KEY (SessionID) REFERENCES Session(SessionID)

);

**UML Diagram**

****

**All Class Java File Codes**

**//Therapist**

public class Therapist {

private int therapistID; // Primary Key

private String name;

private String specialization;

private String availability; // You might want to use a more complex type for availability, such as a list of available time slots or a custom object

// Constructor

public Therapist(int therapistID, String name, String specialization, String availability) {

this.therapistID = therapistID;

this.name = name;

this.specialization = specialization;

this.availability = availability;

}

// Getter and Setter methods

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSpecialization() {

return specialization;

}

public void setSpecialization(String specialization) {

this.specialization = specialization;

}

public String getAvailability() {

return availability;

}

public void setAvailability(String availability) {

this.availability = availability;

}

@Override

public String toString() {

return "Therapist{" +

"therapistID=" + therapistID +

", name='" + name + '\'' +

", specialization='" + specialization + '\'' +

", availability='" + availability + '\'' +

'}';

}

}

**//Patient**

public class Patient {

private int patientID; // Primary Key

private String name;

private String email;

private int age;

private String gender;

// Constructorpublic Patient(int patientID, String name, String email, int age, String gender) {

this.patientID = patientID;

this.name = name;

this.email = email;

this.age = age;

this.gender = gender;

}

// Getter and Setter methods

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public String getGender() {

return gender;

}

public void setGender(String gender) {

this.gender = gender;

}

@Override

public String toString() {

return "Patient{" +

"patientID=" + patientID +

", name='" + name + '\'' +

", email='" + email + '\'' +

", age=" + age +

", gender='" + gender + '\'' +

'}';

}

}

**//Session**

import java.util.Date;

public class Session {

private int sessionID; // Primary Key

private int therapistID; // Foreign Key

private int patientID; // Foreign Key

private Date startTime;

private Date endTime;

private String sessionType;

// Constructor

public Session(int sessionID, int therapistID, int patientID, Date startTime, Date endTime, String sessionType) {

this.sessionID = sessionID;

this.therapistID = therapistID;

this.patientID = patientID;

this.startTime = startTime;

this.endTime = endTime;

this.sessionType = sessionType;

}

// Getter and Setter methods

public int getSessionID() {

return sessionID;

}

public void setSessionID(int sessionID) {

this.sessionID = sessionID;

}

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public Date getStartTime() {

return startTime;

}

public void setStartTime(Date startTime) {

this.startTime = startTime;

}

public Date getEndTime() {

return endTime;

}

public void setEndTime(Date endTime) {

this.endTime = endTime;

}

public String getSessionType() {

return sessionType;

}

public void setSessionType(String sessionType) {

this.sessionType = sessionType;

}

@Override

public String toString() {

return "Session{" +

"sessionID=" + sessionID +

", therapistID=" + therapistID +

", patientID=" + patientID +

", startTime=" + startTime +

", endTime=" + endTime +

", sessionType='" + sessionType + '\'' +

'}';

}

}

**//Appointment**

import java.util.Date;

public class Appointment {

private int appointmentID; // Primary Key

private int therapistID; // Foreign Key

private int patientID; // Foreign Key

private int sessionID; // Foreign Key

private Date date;

private String time;

private String status;

// Constructor

public Appointment(int appointmentID, int therapistID, int patientID, int sessionID, Date date, String time, String status) {

this.appointmentID = appointmentID;

this.therapistID = therapistID;

this.patientID = patientID;

this.sessionID = sessionID;

this.date = date;

this.time = time;

this.status = status;

}

// Getter and Setter methods

public int getAppointmentID() {

return appointmentID;

}

public void setAppointmentID(int appointmentID) {

this.appointmentID = appointmentID;

}

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public int getSessionID() {

return sessionID;

}

public void setSessionID(int sessionID) {

this.sessionID = sessionID;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public String getTime() {

return time;

}

public void setTime(String time) {

this.time = time;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

@Override

public String toString() {

return "Appointment{" +

"appointmentID=" + appointmentID +

", therapistID=" + therapistID +

", patientID=" + patientID +

", sessionID=" + sessionID +

", date=" + date +

", time='" + time + '\'' +

", status='" + status + '\'' +

'}';

}

}

**//Prescription**

public class Prescription {

private int prescriptionID; // Primary Key

private int therapistID; // Foreign Key

private int patientID; // Foreign Key

private String medication;

private String dosage;

private String frequency;

// Constructor

public Prescription(int prescriptionID, int therapistID, int patientID, String medication, String dosage, String frequency) {

this.prescriptionID = prescriptionID;

this.therapistID = therapistID;

this.patientID = patientID;

this.medication = medication;

this.dosage = dosage;

this.frequency = frequency;

}

// Getter and Setter methods

public int getPrescriptionID() {

return prescriptionID;

}

public void setPrescriptionID(int prescriptionID) {

this.prescriptionID = prescriptionID;

}

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public String getMedication() {

return medication;

}

public void setMedication(String medication) {

this.medication = medication;

}

public String getDosage() {

return dosage;

}

public void setDosage(String dosage) {

this.dosage = dosage;

}

public String getFrequency() {

return frequency; }

public void setFrequency(String frequency) {

this.frequency = frequency;

}

@Override

public String toString() {

return "Prescription{" +

"prescriptionID=" + prescriptionID +

", therapistID=" + therapistID +

", patientID=" + patientID +

", medication='" + medication + '\'' +

", dosage='" + dosage + '\'' +

", frequency='" + frequency + '\'' +

'}';

}

**//Payment**

public class Payment {

private int paymentID; // Primary Key

private int appointmentID; // Foreign Key

private double amount;

private String paymentMethod;

private String transactionID;

// Constructor

public Payment(int paymentID, int appointmentID, double amount, String paymentMethod, String transactionID) {

this.paymentID = paymentID;

this.appointmentID = appointmentID;

this.amount = amount;

this.paymentMethod = paymentMethod;

this.transactionID = transactionID;

}

// Getter and Setter methods

public int getPaymentID() {

return paymentID;

}

public void setPaymentID(int paymentID) {

this.paymentID = paymentID;

}

public int getAppointmentID() {

return appointmentID;

}

public void setAppointmentID(int appointmentID) {

this.appointmentID = appointmentID;

}

public double getAmount() {

return amount;

}

public void setAmount(double amount) {

this.amount = amount;

}

public String getPaymentMethod() {

return paymentMethod;

}

public void setPaymentMethod(String paymentMethod) {

this.paymentMethod = paymentMethod;

}

public String getTransactionID() {

return transactionID;

}

public void setTransactionID(String transactionID) {

this.transactionID = transactionID; }

@Override

public String toString() {

return "Payment{" +

"paymentID=" + paymentID +

", appointmentID=" + appointmentID +

", amount=" + amount +

", paymentMethod='" + paymentMethod + '\'' +

", transactionID='" + transactionID + '\'' +

'}';

}

}

**//Assessment**

import java.util.Date;

public class Assessment {

private int assessmentID; // Primary Key

private int sessionID; // Foreign Key

private int therapistID; // Foreign Key

private int patientID; // Foreign Key

private String assessmentType;

private String result;

private Date date;

// Constructor

public Assessment(int assessmentID, int sessionID, int therapistID, int patientID, String assessmentType, String result, Date date) {

this.assessmentID = assessmentID;

this.sessionID = sessionID;

this.therapistID = therapistID;

this.patientID = patientID;

this.assessmentType = assessmentType;

this.result = result;

this.date = date;

}

// Getter and Setter methods

public int getAssessmentID() {

return assessmentID;

}

public void setAssessmentID(int assessmentID) {

this.assessmentID = assessmentID;

}

public int getSessionID() {

return sessionID;

}

public void setSessionID(int sessionID) {

this.sessionID = sessionID;

}

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public String getAssessmentType() {

return assessmentType;

}

public void setAssessmentType(String assessmentType) {

this.assessmentType = assessmentType;

}

public String getResult() {

return result;

}

public void setResult(String result) {

this.result = result;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

@Override

public String toString() {

return "Assessment{" +

"assessmentID=" + assessmentID +

", sessionID=" + sessionID +

", therapistID=" + therapistID +

", patientID=" + patientID +

", assessmentType='" + assessmentType + '\'' +

", result='" + result + '\'' +

", date=" + date +

'}';

}

}

**//Profile**

public class Profile {

private int userID; // Primary Key

private String userType;

private String username;

private String email;

private String password;

// Constructor

public Profile(int userID, String userType, String username, String email, String password) {

this.userID = userID;

this.userType = userType;

this.username = username;

this.email = email;

this.password = password;

}

// Getter and Setter methods

public int getUserID() {

return userID;

}

public void setUserID(int userID) {

this.userID = userID;

}

public String getUserType() {

return userType;

}

public void setUserType(String userType) {

this.userType = userType;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

@Override

public String toString() {

return "Profile{" +

"userID=" + userID +

", userType='" + userType + '\'' +

", username='" + username + '\'' +

", email='" + email + '\'' +

", password='" + password + '\'' +

'}';

}

}

**//Messaging**

import java.util.Date;

public class Messaging {

private int messageID; // Primary Key

private int senderID; // Foreign Key

private int receiverID; // Foreign Key

private String messageContent;

private Date timestamp;

// Constructor

public Messaging(int messageID, int senderID, int receiverID, String messageContent, Date timestamp) {

this.messageID = messageID;

this.senderID = senderID;

this.receiverID = receiverID;

this.messageContent = messageContent;

this.timestamp = timestamp;

}

// Getter and Setter methods

public int getMessageID() {

return messageID;

}

public void setMessageID(int messageID) {

this.messageID = messageID;

}

public int getSenderID() {

return senderID;

}

public void setSenderID(int senderID) {

this.senderID = senderID;

}

public int getReceiverID() {

return receiverID;

}

public void setReceiverID(int receiverID) {

this.receiverID = receiverID;

}

public String getMessageContent() {

return messageContent;

}

public void setMessageContent(String messageContent) {

this.messageContent = messageContent;

}

public Date getTimestamp() {

return timestamp;

}

public void setTimestamp(Date timestamp) {

this.timestamp = timestamp;

}

@Override

public String toString() {

return "Messaging{" +

"messageID=" + messageID +

", senderID=" + senderID +

", receiverID=" + receiverID +

", messageContent='" + messageContent + '\'' +

", timestamp=" + timestamp +

'}';

}}

**//Review**

import java.util.Date;

public class Review {

private int reviewID; // Primary Key

private int therapistID; // Foreign Key

private int patientID; // Foreign Key

private int sessionID; // Foreign Key

private int rating;

private String reviewContent;

private Date date;

// Constructor

public Review(int reviewID, int therapistID, int patientID, int sessionID, int rating, String reviewContent, Date date) {

this.reviewID = reviewID;

this.therapistID = therapistID;

this.patientID = patientID;

this.sessionID = sessionID;

this.rating = rating;

this.reviewContent = reviewContent;

this.date = date;

}

// Getter and Setter methods

public int getReviewID() {

return reviewID;

}

public void setReviewID(int reviewID) {

this.reviewID = reviewID;

}

public int getTherapistID() {

return therapistID;

}

public void setTherapistID(int therapistID) {

this.therapistID = therapistID;

}

public int getPatientID() {

return patientID;

}

public void setPatientID(int patientID) {

this.patientID = patientID;

}

public int getSessionID() {

return sessionID;

}

public void setSessionID(int sessionID) {

this.sessionID = sessionID;

}

public int getRating() {

return rating;

}

public void setRating(int rating) {

this.rating = rating;

}

public String getReviewContent() {

return reviewContent;

}

public void setReviewContent(String reviewContent) {

this.reviewContent = reviewContent;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

@Override

public String toString() {

return "Review{" +

"reviewID=" + reviewID +

", therapistID=" + therapistID +

", patientID=" + patientID +

", sessionID=" + sessionID +

", rating=" + rating +

", reviewContent='" + reviewContent + '\'' +

", date=" + date +

'}';

}

}

**//Main**

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

public class Main {

public static void main(String[] args) {

// Creating instances of entities

Therapist therapist1 = new Therapist(1, "John Doe", "Psychologist", "Monday, Wednesday, Friday");

Patient patient1 = new Patient(1, "Alice Smith", "alice@example.com", 30, "Female");

Session session1 = new Session(1, 1, 1, new Date(), new Date(), "Individual");

Appointment appointment1 = new Appointment(1, 1, 1, 1, new Date(), "10:00 AM", "Confirmed");

Prescription prescription1 = new Prescription(1, 1, 1, "Medicine X", "10mg", "Twice daily");

Profile profile1 = new Profile(1, "Therapist", "john\_doe", "john@example.com", "password123");

Review review1 = new Review(1, 1, 1, 1, 5, "Great session!", new Date());

// Creating a list of appointments

List<Appointment> appointments = new ArrayList<>();

appointments.add(appointment1);

// Creating a list of prescriptions

List<Prescription> prescriptions = new ArrayList<>();

prescriptions.add(prescription1);

// Creating a list of reviews

List<Review> reviews = new ArrayList<>();

reviews.add(review1);

// Printing details

System.out.println("Therapist: " + therapist1);

System.out.println("Patient: " + patient1);

System.out.println("Session: " + session1);

System.out.println("Appointments: ");

for (Appointment appointment : appointments) {

System.out.println(appointment);

}

System.out.println("Prescriptions: ");

for (Prescription prescription : prescriptions) {

System.out.println(prescription);

}

System.out.println("Profile: " + profile1);

System.out.println("Reviews: ");

for (Review review : reviews) {

System.out.println(review);

}

}

**Challenges List**

* Representing complex relationships, especially **many-to-many** relationships, can be challenging. These relationships involve multiple entities and require careful modeling to avoid ambiguity.
* Creating the tables in a specific order and inserting the values accordingly
* Handling sensitive information such as patient health records requires strict adherence to data security and privacy regulations. Ensuring data encryption, access controls, and secure storage mechanisms is crucial.