**LEGAL CASE MANAGEMENT SYSTEM**

# **PROJECT REPORT**

**ABSTRACT**

A Legal case management system is a software application that makes easier to manage cases and give every member of your law firm insight into what’s happening. This makes collaborating on difficult cases and streamlining the day to day running of your practice a breeze.

The development of the legal case management system utilizes MySQL and java to streamline and enhance the management of legal cases within law firms and legal departments. It aims to tackle the inefficiencies and complexities associated with the traditional case management by providing an integrated, user friendly solution.

**INTRODUCTION**

A **Legal Case Management System (LCMS)** is a crucial tool for organizing and efficiently managing legal cases. It provides a structured foundation to handle the complexities of legal proceedings. In this system, we’ll focus on the following core entities:

1. **Clients**:
   * A **client** is an individual seeking legal assistance or representation from a lawyer or law firm.
   * Key attributes:
     + **Client ID (Primary Key):** A unique identifier for each client.
     + **Name:** The full name of the client.
     + **Address:** The residential address of the client.
     + **Email:** The client’s email address.
     + **Phone Number:** The client’s phone number.
2. **Lawyers**:
   * **Lawyers** are professionals who provide legal advice, representation, and assistance to clients in legal matters.
   * Key attributes:
     + **Lawyer ID (Primary Key):** A unique identifier for each lawyer.
     + **Name:** The full name of the lawyer.
     + **Specialization:** The area of law in which the lawyer practices.
3. **Cases**:
   * A **case** represents a legal matter being handled by the law firm.
   * Key attributes:
     + **Case ID (Primary Key):** A unique identifier for each case.
     + **Title:** A brief description of the case.
     + **Status:** The current status of the case (e.g., pending, closed, in progress).
4. **Documents**:
   * **Documents** include legal paperwork related to cases, such as pleadings, motions, and court filings.
   * Key attributes:
     + **Document ID (Primary Key):** A unique identifier for each document.
     + **Title:** A descriptive title for the document.
     + **Content:** The actual content of the document.
5. **Billing**:
   * **Billing** tracks financial aspects related to legal services provided.
   * Key attributes:
     + **Invoice ID (Primary Key):** A unique identifier for each invoice.
     + **Amount:** The billed amount for legal services.
     + **Due Date:** The deadline for payment.
6. **Case Assignments**:
   * **Case assignments** link lawyers to specific cases.
   * Key attributes:
     + **Assignment ID (Primary Key):** A unique identifier for each assignment.
     + **Lawyer ID:** The lawyer assigned to the case.
     + **Case ID:** The case to which the lawyer is assigned.

**Features of a Legal Case Management System:**

1. **Case Management:**
   * Track case details, parties involved, case status, and key dates.
   * Manage case documents.
2. **Client Management:**
   * Maintain client information and communication history.
   * Track client billing.
3. **Calendar and Task Management:**
   * Schedule court dates, deadlines, and appointments.
   * Assign and track tasks.
4. **Document Management:**
   * Store and organize legal documents securely.
   * Implement version control.
5. **Time and Billing:**
   * Track billable hours.
   * Generate invoices.
6. **Compliance and Security:**
   * Ensure legal compliance and data protection.

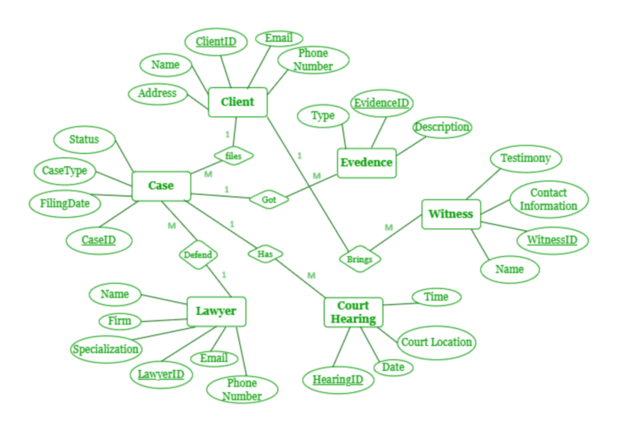
**FUNCTIONAL REQUIREMENTS**

1. **Case Management**:
   * **Track Case Details**:
     + Maintain case numbers, parties involved, case status, and key dates.
   * **Manage Case Documents**:
     + Store and organize legal documents related to cases (e.g., pleadings, motions, court filings).
   * **Client Communication**:
     + Record communication history with clients (emails, phone calls, meetings).
2. **Client Management**:
   * **Client Information**:
     + Maintain client details, including contact information (name, address, email, phone number).
   * **Billing Information**:
     + Track billing details for each client.
3. **Lawyer Management**:
   * **Lawyer Profiles**:
     + Store lawyer information (name, specialization).
   * **Assign Lawyers to Cases**:
     + Associate lawyers with specific cases.
4. **Document Management**:
   * **Document Storage**:
     + Securely store legal documents.
   * **Version Control**:
     + Implement version tracking for documents.
5. **Billing and Invoicing**:
   * **Billable Hours**:
     + Track billable hours for lawyers and staff.
   * **Generate Invoices**:
     + Create invoices for legal services provided.
6. **Security and Compliance**:
   * **Data Protection**:
     + Ensure compliance with legal regulations (e.g., GDPR) to protect sensitive client information.
   * **Access Control**:
     + Restrict access to authorized users.
7. **Mobile Access**:
   * **Responsive Interface**:
     + Provide mobile apps or responsive web interfaces for on-the-go access.
   * **Case Updates**:
     + Allow lawyers and staff to manage cases remotely.

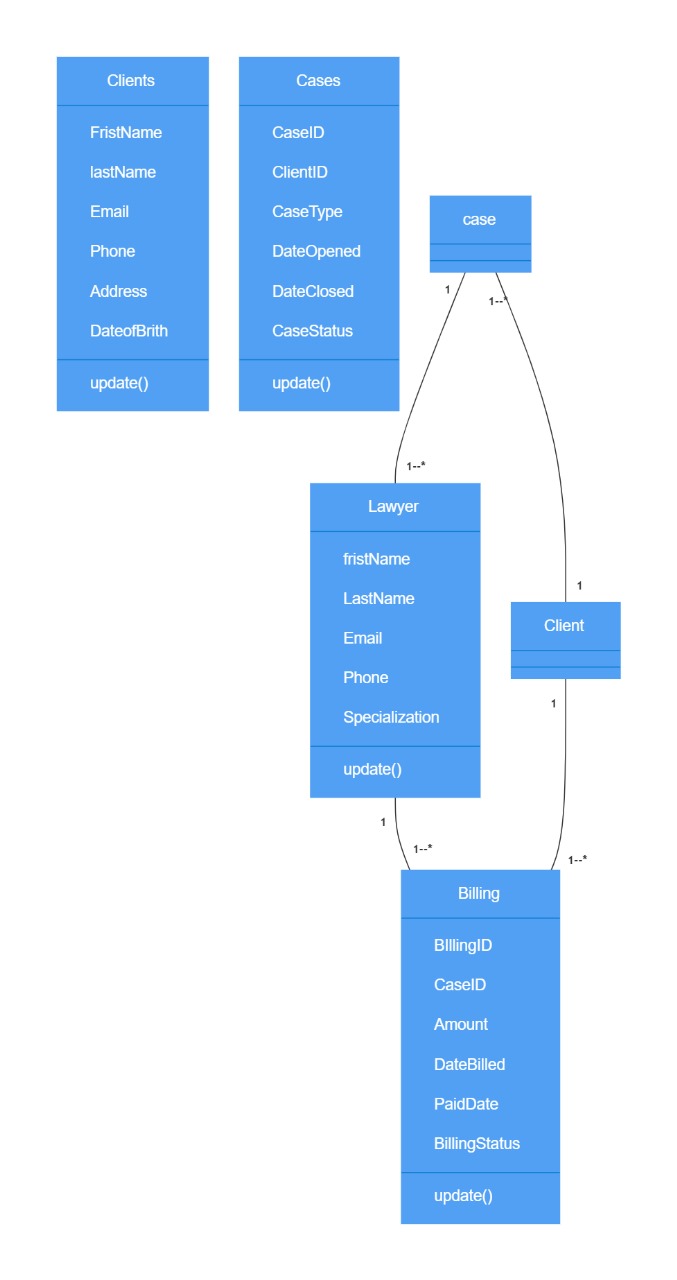
**ER DIAGRAM**

An Entity-Relationship (ER) diagram for Legal case management system. The ER diagram captures the relationships between various entities in the system.

We’ll outline the core entities, their attributes, and the relationships between them.

1. **Clients**:
   * **Client ID (Primary Key):** Unique identifier for each client.
   * **Name:** Full name of the client.
   * **Address:** Residential address.
   * **Email:** Client’s email address.
   * **Phone Number:** Client’s phone number.
2. **Lawyers**:
   * **Lawyer ID (Primary Key):** Unique identifier for each lawyer.
   * **Name:** Full name of the lawyer.
   * **Specialization:** Area of law in which the lawyer practices.
3. **Cases**:
   * **Case ID (Primary Key):** Unique identifier for each case.
   * **Title:** Brief description of the case.
   * **Status:** Current case status (e.g., pending, closed, in progress).
4. **Documents**:
   * **Document ID (Primary Key):** Unique identifier for each document.
   * **Title:** Descriptive title for the document.
   * **Content:** Actual content of the document.
5. **Billing**:
   * **Invoice ID (Primary Key):** Unique identifier for each invoice.
   * **Amount:** Billed amount for legal services.
   * **Due Date:** Payment deadline.
6. **Case Assignments**:
   * **Assignment ID (Primary Key):** Unique identifier for each assignment.
   * **Lawyer ID:** Lawyer assigned to the case.
   * **Case ID:** Case to which the lawyer is assigned.

**UML DIAGRAM**



**Queries to create the tables in the Database:**

* create database legalcase;

use legalcase;

* CREATE TABLE Clients (

ClientID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(15),

Address TEXT,

DateOf Birth DATE,

DateCreateTIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Notes TEXT

);

alter table Clients drop column Notes;

* CREATE TABLE Cases (

CaseID INT PRIMARY KEY AUTO\_INCREMENT,

ClientID INT,

CaseTitle VARCHAR(255),

CaseDescription TEXT,

CaseType VARCHAR(50),

CaseStatus VARCHAR(50),

DateOpened DATE,

DateClosed DATE,

Notes TEXT,

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

* CREATE TABLE Lawyers (

LawyerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

Phone VARCHAR(15),

Specialization VARCHAR(100),

BarNumber VARCHAR(50),

DateHired DATE,

Notes TEXT

);

Alter table Lawyers add column DateOfBirth DATE;

* TABLE CaseAssignments (

AssignmentID INT PRIMARY KEY AUTO\_INCREMENT,

CaseID INT,

LawyerID INT,

AssignmentDate DATE,

Role VARCHAR(50),

Notes TEXT, CREATE

FOREIGN KEY (CaseID) REFERENCES Cases(CaseID),

FOREIGN KEY (LawyerID) REFERENCES Lawyers(LawyerID)

);

Alter table CaseAssignments add column status VARCHAR(50);

* CREATE TABLE Documents (

DocumentID INT PRIMARY KEY AUTO\_INCREMENT,

CaseID INT,

DocumentName VARCHAR(255),

DocumentType VARCHAR(50),

UploadDate TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FilePath VARCHAR(255), -- or use a BLOB for file storage

Notes TEXT,

FOREIGN KEY (CaseID) REFERENCES Cases(CaseID)

);

* CREATE TABLE Billing (

BillingID INT PRIMARY KEY AUTO\_INCREMENT,

CaseID INT,

ClientID INT,

LawyerID INT,

DateBilled DATE,

Amount DECIMAL(10, 2),

BillingStatus VARCHAR(50),

Notes TEXT,

FOREIGN KEY (CaseID) REFERENCES Cases(CaseID),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),

FOREIGN KEY (LawyerID) REFERENCES Lawyers(LawyerID)

);

Alter table Billing add column PaidDate DATE;

**All Classes Java file codes:**

**Clients.java:**

import java.util.Date;

public class Client {

    private int clientID;

    private String firstName;

    private String lastName;

    private Date dateOfBirth;

    private String address;

    private String phone;

    private String email;

    public Client(int clientID, String firstName, String lastName) {

        this.clientID = clientID;

        this.firstName = firstName;

        this.lastName = lastName;

    }

    public int getClientID() {

        return clientID;

    }

    public void setClientID(int clientID) {

        this.clientID = clientID;

    }

    public String getFirstName() {

        return firstName;

    }

    public void setFirstName(String firstName) {

        this.firstName = firstName;

    }

    public String getLastName() {

        return lastName;

    }

    public void setLastName(String lastName) {

        this.lastName = lastName;

    }

    public Date getDateOfBirth() {

        return dateOfBirth;

    }

    public void setDateOfBirth(Date dateOfBirth) {

        this.dateOfBirth = dateOfBirth;

    }

    public String getAddress() {

        return address;

    }

    public void setAddress(String address) {

        this.address = address;

    }

    public String getPhone() {

        return phone;

    }

    public void setPhone(String phone) {

        this.phone = phone;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

}

**Cases.java:**

import java.util.Date;

public class LegalCase {

    private int caseID;

    private int clientID;

    private String caseType;

    private Date dateOpened;

    private Date dateClosed;

    private String caseStatus;

    public LegalCase(int caseID, int clientID, String caseType, Date dateOpened, Date dateClosed, String caseStatus) {

        this.caseID = caseID;

        this.clientID = clientID;

        this.caseType = caseType;

        this.dateOpened = dateOpened;

        this.dateClosed = dateClosed;

        this.caseStatus = caseStatus;

    }

    public int getCaseID() {

        return caseID;

    }

    public int getClientID() {

        return clientID;

    }

    public String getCaseType() {

        return caseType;

    }

    public Date getDateOpened() {

        return dateOpened;

    }

    public Date getDateClosed() {

        return dateClosed;

    }

    public String getCaseStatus() {

        return caseStatus;

    }

    public void setCaseID(int caseID) {

        this.caseID = caseID;

    }

    public void setClientID(int clientID) {

        this.clientID = clientID;

    }

    public void setCaseType(String caseType) {

        this.caseType = caseType;

    }

    public void setDateOpened(Date dateOpened) {

        this.dateOpened = dateOpened;

    }

    public void setDateClosed(Date dateClosed) {

        this.dateClosed = dateClosed;

    }

    public void setCaseStatus(String caseStatus) {

        this.caseStatus = caseStatus;

    }

}

**Lawyers.java:**

import java.util.Date;

public class Lawyer {

    private int lawyerID;

    private String firstName;

    private String lastName;

    private Date dateOfBirth;

    private Date dateHired;

    private String specialization;

    public Lawyer(int lawyerID, String firstName, String lastName, Date dateOfBirth, Date dateHired,

            String specialization) {

        this.lawyerID = lawyerID;

        this.firstName = firstName;

        this.lastName = lastName;

        this.dateOfBirth = dateOfBirth;

        this.dateHired = dateHired;

        this.specialization = specialization;

    }

    public int getLawyerID() {

        return lawyerID;

    }

    public String getFirstName() {

        return firstName;

    }

    public String getLastName() {

        return lastName;

    }

    public Date getDateOfBirth() {

        return dateOfBirth;

    }

    public Date getDateHired() {

        return dateHired;

    }

    public String getSpecialization() {

        return specialization;

    }

    public void setLawyerID(int lawyerID) {

        this.lawyerID = lawyerID;

    }

    public void setFirstName(String firstName) {

        this.firstName = firstName;

    }

    public void setLastName(String lastName) {

        this.lastName = lastName;

    }

    public void setDateOfBirth(Date dateOfBirth) {

        this.dateOfBirth = dateOfBirth;

    }

    public void setDateHired(Date dateHired) {

        this.dateHired = dateHired;

    }

    public void setSpecialization(String specialization) {

        this.specialization = specialization;

    }

}

**Case assignments.java:**

import java.util.Date;

public class CaseAssignment {

    private int assignmentID;

    private int caseID;

    private int lawyerID;

    private Date assignmentDate;

    private String status;

    public CaseAssignment(int assignmentID, int caseID, int lawyerID, Date assignmentDate, String status) {

        this.assignmentID = assignmentID;

        this.caseID = caseID;

        this.lawyerID = lawyerID;

        this.assignmentDate = assignmentDate;

        this.status = status;

    }

    public int getAssignmentID() {

        return assignmentID;

    }

    public int getCaseID() {

        return caseID;

    }

    public int getLawyerID() {

        return lawyerID;

    }

    public Date getAssignmentDate() {

        return assignmentDate;

    }

    public String getStatus() {

        return status;

    }

    public void setAssignmentID(int assignmentID) {

        this.assignmentID = assignmentID;

    }

    public void setCaseID(int caseID) {

        this.caseID = caseID;

    }

    public void setLawyerID(int lawyerID) {

        this.lawyerID = lawyerID;

    }

    public void setAssignmentDate(Date assignmentDate) {

        this.assignmentDate = assignmentDate;

    }

    public void setStatus(String status) {

        this.status = status;

    }

}

**Documents.java**

import java.util.Date;

public class Document {

    private int documentID;

    private int caseID;

    private String documentType;

    private Date uploadDate;

    private String filePath;

    private String notes;

    public Document(int documentID, int caseID, String documentType, Date uploadDate, String filePath, String notes) {

        this.documentID = documentID;

        this.caseID = caseID;

        this.documentType = documentType;

        this.uploadDate = uploadDate;

        this.filePath = filePath;

        this.notes = notes;

    }

    public int getDocumentID() {

        return documentID;

    }

    public int getCaseID() {

        return caseID;

    }

    public String getDocumentType() {

        return documentType;

    }

    public Date getUploadDate() {

        return uploadDate;

    }

    public String getFilePath() {

        return filePath;

    }

    public String getNotes() {

        return notes;

    }

    public void setDocumentID(int documentID) {

        this.documentID = documentID;

    }

    public void setCaseID(int caseID) {

        this.caseID = caseID;

    }

    public void setDocumentType(String documentType) {

        this.documentType = documentType;

    }

    public void setUploadDate(Date uploadDate) {

        this.uploadDate = uploadDate;

    }

    public void setFilePath(String filePath) {

        this.filePath = filePath;

    }

    public void setNotes(String notes) {

        this.notes = notes;

    }

**Billing.java:**

import java.util.Date;

public class Billing {

    private int billingID;

    private int caseID;

    private double amount;

    private Date dateBilled;

    private Date paidDate;

    private String billingStatus;

    public Billing(int billingID, int caseID, double amount, Date dateBilled, Date paidDate, String billingStatus) {

        this.billingID = billingID;

        this.caseID = caseID;

        this.amount = amount;

        this.dateBilled = dateBilled;

        this.paidDate = paidDate;

        this.billingStatus = billingStatus;

    }

    public int getBillingID() {

        return billingID;

    }

    public void setBillingID(int billingID) {

        this.billingID = billingID;

    }

    public int getCaseID() {

        return caseID;

    }

    public void setCaseID(int caseID) {

        this.caseID = caseID;

    }

    public double getAmount() {

        return amount;

    }

    public void setAmount(double amount) {

        this.amount = amount;

    }

    public Date getDateBilled() {

        return dateBilled;

    }

    public void setDateBilled(Date dateBilled) {

        this.dateBilled = dateBilled;

    }

    public Date getPaidDate() {

        return paidDate;

    }

    public void setPaidDate(Date paidDate) {

        this.paidDate = paidDate;

    }

    public String getBillingStatus() {

        return billingStatus;

    }

    public void setBillingStatus(String billingStatus) {

        this.billingStatus = billingStatus;

    }

}

**Challenges list:**

* Designing a normalised and efficient database schema to handle complex relationships like many-to-many .
* Creating the tables in a specific order and inserting the values accordingly .
* Implementing and managing the various one-to-many and many-to-many relationships in Java objects and ensuring data integrity.
* Ensuring that the data entered into the system is valid and consistent