Aswan University
Aswan Faculty of Engineering,
Electrical Engineering Department

Health Care (Clinic Management System)

Edited by:

Maha Mamdouh Ahmed

Third year Computer and systems Engineer

Presented to:

Dr/ Mahmoud Ali Saber

Supervised by:

Eng / Rowan Omar

Overview:

The system is built as a web-based platform with a structured SQL database to manage user-specific details efficiently. This architecture ensures:

- Secure and centralized data storage.
- Easy accessibility for authorized users (doctor and receptionist).
- Consistent and seamless communication between the two administrative roles.

This workflow fosters an efficient clinic management process, where patient data is accurately recorded, updated, and utilized to provide better healthcare services.

System Objective:

The application simplifies the operational workflow of clinics by providing an intuitive interface for storing, retrieving, and managing patient data. It ensures accuracy, reduces manual effort, and supports effective appointment scheduling.

User Roles and Responsibilities:

The Clinic Management System is designed to be operated by two primary administrative roles: **Doctor** and **Receptionist**.

1) Receptionist Responsibilities

- Patient Data Management**:**

- Assign patient details for individuals visiting the clinic.
- Save these details securely in the system's database.
- Transmit patient information to the doctor for review and consultation.

2) Doctor Responsibilities

- Patient Record Review**:**

- Access patient details submitted by the receptionist.
- Examine the patient and determine appropriate treatments or medications.

- Prescription Management**:**

- Enter prescribed medications and treatment notes into the system for each patient.
- Forward this information to the receptionist for further communication with the patient.

3) Shared Access

- Patient History**:**

- Both doctors and receptionists can access and review a comprehensive history of the patient’s medical records, ensuring continuity of care.

Technologies Used:

- Back-End

- **Database**: MySQL
- **Server-Side**: Servlets

- Front-End

- **Languages and Tools**: JSP, HTML, CSS, JavaScript

- Server

- **Application Server**: Apache Tomcat v9.0.89

Features:

- **Appointment Scheduling**:

  - Integrated calendar to schedule and manage patient appointments efficiently, with visibility into available time slots.

Proposed Features for Enhancement:

- **Appointment Calendar Integration**:

  - Sync with the doctor's email calendar to automatically update and send reminders for appointments.

- **Advanced History Features**:

  - Enhanced functionality for viewing and filtering patient medical histories by date, treatment type, or condition.

- **Online Payment Integration**:

  - Support for secure online payment options, reducing reliance on cash transactions and enhancing patient convenience.

Code examples:

Log-In Functionality**:**

Log-In Servlet

```
package com.example.yaraaaaab;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class LoginServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        try (Connection con = new Database().connection()) {
            String query = "SELECT * FROM users WHERE username = ? AND
password = ?";
            PreparedStatement ps = con.prepareStatement(query);
            ps.setString(1, username);
            ps.setString(2, password);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                HttpSession session = request.getSession();
                session.setAttribute("username", username);
                response.sendRedirect("index.jsp");
            } else {
                request.setAttribute("error", "Invalid username or
password");
                request.getRequestDispatcher("index.jsp").forward(request,
response);
            }
        } catch (SQLException e) {
            throw new RuntimeException(e);
        } catch (ServletException e) {
            throw new RuntimeException(e);
        }
    }
}

  return go(f, seed, [])
}
```

Log-In JSP Page

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sign Up</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: Arial, sans-serif;
    }

    body {
      background-color: #f4f4f9;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    .container {
      background: white;
      width: 500px;
      padding: 70px 20px;
      border-radius: 10px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }

    .container h1 {
      text-align: center;
      margin-bottom: 20px;
      color: #007e85;
    }

    .form-group {
      margin-bottom: 15px;
    }

    label {
      display: block;
      margin-bottom: 5px;
      color: #333;
    }

    input {
      width: 100%;
      padding: 10px;
      border: 1px solid #ccc;
      border-radius: 5px;
      font-size: 14px;
    }

    input:focus {
      border-color: #007e85;
      outline: none;
    }

    .btn {
      width: 100%;
      padding: 10px;
      background: #007e85;
      color: white;
      border: none;
      border-radius: 5px;
      font-size: 16px;
      cursor: pointer;
    }

    .btn:hover {
      background: #007e85;
    }

    .link {
      text-align: center;
      margin-top: 10px;
    }

    .link a {
      color: #007e85;
      text-decoration: none;
      font-size: 14px;
    }

    .link a:hover {
      text-decoration: underline;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Sign Up</h1>
    <form action="SignUpServlet" method="post">
      <div class="form-group">
        <label for="fullName">Full Name</label>
        <input type="text" id="fullName" name="fullName" placeholder="Enter
your full name" required>
      </div>
      <div class="form-group">
        <label for="email">Email</label>
        <input type="email" id="email" name="email" placeholder="Enter your
email" required>
      </div>
      <div class="form-group">
        <label for="password">Password</label>
        <input type="password" id="password" name="password"
placeholder="Create a password" required>
      </div>
      <button type="submit" class="btn">Sign Up</button>
      <div class="link">
        <p>Already have an account? <a href="signIn.jsp">Sign In</a></p>
      </div>
    </form>

  </div>
</body>
</html>
```

Database:

```java
package com.example.yaraaaaab;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Database {

    // Database credentials
    private static final String URL = "jdbc:mysql://localhost:3306/mydb";
 // Replace with your database URL
    private static final String USER = "root";  // Replace with your
database username
    private static final String PASSWORD = "password";  // Replace with
your database password

    // Method to establish a database connection
    public Connection connection() {
        try {
            // Load the MySQL JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Create and return the connection
            Connection con = DriverManager.getConnection(URL, USER,
PASSWORD);
            System.out.println("Connection Successful");

            return con;  // Return the connection object

        } catch (ClassNotFoundException | SQLException e) {
            // Handle and print any errors that occur
            System.out.println("Error: " + e.getMessage());
            e.printStackTrace();
        }
        return null;  // Return null if connection fails
    }

    public static void main(String[] args) {
        Database db = new Database();
        db.connection();  // Test the connection method
    }
}
```

Conclusion

The Clinic Management System serves as an efficient tool to enhance communication and administrative efficiency within healthcare facilities. With additional proposed features, it aims to provide a comprehensive solution for modern clinic operations.

Thank you!
For any questions, please reach out.