**Case Study: Hospital Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for a Hospital Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage patient records and doctor assignments.

**Requirements:**

1. **Patient Management**:
   * Implement the functionality to add, update, and delete patient records.
   * Each patient should have attributes such as patient\_id, name, age, gender, disease, and doctor\_id.
2. **Doctor Assignment**:
   * Implement the functionality to assign doctors to patients.
   * Each doctor should have attributes such as doctor\_id, name, and specialization.
3. **Reporting**:
   * Implement the functionality to generate a report of patients assigned to a specific doctor.

**Business Functionalities:**

1. **Add/Update/Delete Patients**:
   * Create a class Patient with attributes patient\_id, name, age, gender, disease, and doctor\_id.
   * Implement methods to add a new patient, update existing patient details, and delete a patient from the system.
2. **Assign/Update Doctors**:
   * Create a class Doctor with attributes doctor\_id, name, and specialization.
   * Implement methods to assign a doctor to a patient and update the doctor’s details.
3. **Patient Report by Doctor**:
   * Implement a method to generate a list of patients assigned to a specific doctor by doctor\_id.

**Section 2: MySQL Database Management**

Design a MySQL database schema to support the Hospital Management System and provide solutions for the problem statements.

**Table Structures:**

1. **Patients Table**:
   * patient\_id: INT, Primary Key
   * name: VARCHAR(100)
   * age: INT
   * gender: VARCHAR(10)
   * disease: VARCHAR(100)
   * doctor\_id: INT, Foreign Key References Doctors(doctor\_id)
2. **Doctors Table**:
   * doctor\_id: INT, Primary Key
   * name: VARCHAR(100)
   * specialization: VARCHAR(100)

**Problem Statements:**

1. Write a query to find the total number of patients assigned to each doctor.
2. Write a query to find the names of doctors and the total number of patients assigned to them.
3. Write a query to find the names of patients who have not been assigned a doctor.
4. Write a query to find the specializations of doctors who have more than 10 patients assigned.
5. Write a query to find the patient names and their corresponding diseases for patients assigned to a specific doctor.