Information Systems and Data Modeling – IT1090

Assignment

Title: Diet Plan & Health Checkup System

Batch Number: Y1.S2.WD.15.1	Group Number: MLB_15.01_07

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

IT22575562	WIJEKOON W.M.M.G.K.P	[illegible] ....
IT22567642	THILAKARATHNA P.L.B.H	[illegible] ....
IT22591098	JAYALATH J.P.R.J	[illegible] ....
IT22566034	PERERA K.C.D	[illegible] ....
IT22217622	HETTIARACHCHI A.N.M	[illegible] ....

Submitted on: 28/05/2023

Contents

1.Introduction:

Diet planning and health check-up system is a system that contributes to maintaining our health based on our needs and wants. This is very helpful in protecting our health as well as looking for weaknesses.

The health check-up and diet planning system plays a vital role in promoting overall health and well-being. Regular health check-ups are essential for early detection of health issues, leading to increased chances of effective treatment. Factors like age, health condition, family history, and lifestyle choices influence the frequency of check-ups. On the other hand, diet planning focuses on determining the optimal nutrient intake for individuals or groups, aiming for nutritionally adequate diets with minimal risk of nutrient imbalances.

To efficiently manage this system, a robust database management system is necessary. It enables the storage, organization, and retrieval of crucial information related to patient health records, screening results, dietary guidelines, and individual preferences. The database system facilitates seamless coordination between health check-ups and diet planning, ensuring accurate record-keeping, personalized recommendations, and ongoing monitoring. By leveraging a well-designed database system, healthcare professionals can streamline operations, enhance data accessibility, and ultimately deliver effective healthcare services tailored to individual needs.

2. Hypothetical Scenario:

A user who has the need related to food planning and health protection system can connect with the system. Accordingly, we can identify two main systems in terms of food planning and health care.

There will be a doctor to conduct the relevant tests and there will be patients to contact for that. There is a nutritionist to carry out tests on food plans and there is a recipe system under a nutritionist. A report will be issued by the doctor and nutritionist related to those tests and the report will be made available to the examiner or patient. In relation to a report that is issued, the food planning and health care system will monitor the odd report. There is a method of payment related to each of those tests and reports and related to each payment, the patient or examiner must make the corresponding payment.

In connection with the relevant hypothesis:

There is at least one test method for a user or patient connected to the system and there is a mandatory payment method for that. There is a doctor and a nutritionist related to the relevant tests and they are directly connected with the system. There may be several food planning methods related to a nutritionist and one nutritionist related to a food plan is led by one. also under a nutritionist there are related food plans and there is a nutritionist related to each food plan. Likewise, there is a doctor related to a medical examination and there may be several medical examinations related to one doctor and there are reports related to each of those tests and there will be a patient related to a report and reports according to the relevant tests related to one patient. There may be several.

3.Requirement Analysis:

3.1 Main Requirements:

3.1.1 Functional Requirements:

- ❖ Main Functions of the website and Events that take place between the users and the system is described by the Functional requirements.
- ❖ Eight users are using this Diet Plan and Health Checkup System.
  Namely: Client-User, Registered User, Administrator, Manager, Doctor, Nutritionist and Fitness Expert and the Developer.
- ❖ They access this system in different ways where it is related to them.

1.User Requirements for Client-User and Registered User (Front-end Access):

User Requirements:

-User can check FAQ.

-User views the available Feedbacks.

- Create an account with personal information (name, email, etc.).

- Registered User can do health checkup or diet plan service

- Log in to the system using credentials.

- View their profile information.

- Set personal goals (weight loss, muscle gain, etc.).

- Access and select from available diet plans.

- View and track their daily meal plans.

- Access and track their health checkup records.

- View and download reports generated by health checkups.

- Make payments for services rendered.

- Communicate with nutritionists/fitness experts for guidance and support.

System Requirements:

- User registration functionality.
- User authentication and login system.
- Profile management functionality.
- Diet plan selection and display.
- Meal plan tracking and management.
- Health checkup record management.
- Report generation and retrieval.
- Payment integration.
- Communication features (messaging, notifications, etc.).

2.User Requirements for Nutritionist and Fitness Expert (Back-end Access):

User Requirements:

- Register as a nutritionist or fitness expert.
- Log in to the system using credentials.
- Manage their profile information and expertise.
- Access patient/client information.
- Create personalized diet plans for patients/clients.
- Track and monitor patient/client progress.
- Generate reports based on patient/client data.
- Communicate with patients/clients for support and guidance.

System Requirements:

- User registration and authentication system for nutritionists/fitness experts.
- Profile management functionality.
- Access control for patient/client information.
- Diet plan creation and customization features.
- Progress tracking and monitoring tools.
- Report generation capabilities.
- Communication features with patients/clients.

3.User Requirements for Doctor (Back-end Access):

User Requirements:

- Register as a doctor.
- Log in to the system using credentials.
- Manage their profile information.
- Access patient information and health checkup records.
- Review and analyze health checkup results.
- Provide recommendations and prescriptions.
- Communicate with patients for further consultations.

System Requirements:

- User registration and authentication system for doctors.
- Profile management functionality.
- Access control for patient information and health checkup records.
- Health checkup result viewing and analysis features.
- Recommendation and prescription management tools.
- Communication features with patients.

4.User Requirements for Administrator and Manager (Back-end Access):

User Requirements:

- Register as an administrator or manager.
- Log in to the system using credentials.
- Manage user accounts and access permissions.
- Monitor system performance and security.
- Generate reports on system usage and performance.
- Handle system configurations and settings.

System Requirements:

- User registration and authentication system for administrators and managers.
- User account management functionality.
- System performance monitoring and security features.
- Reporting and analytics capabilities.
- System configuration and settings management tools.

3.1.2 Non-Functional Requirements

Non-functional requirements are simply known as quality attributes. It describes the characteristics of the system that are not directly concerned with specific functionality. Non-functional requirements may be more critical than functional requirements. If these are not achieved, the system may be useless.

Speed

- ❖ The system must have good speed.
- ❖ The system can access more users at the same time without any failures.

Availability

- ❖ The system should be available 24/7.

User friendly

- ❖ The system should be accessible to users with low IT literacy.

Reliability

- ❖ The system must have the ability to detect invalid user credentials.

Security

- ❖ The system should have the ability to prevent unauthorized access, misuse, forgery, and secure user data.
- ❖ Also, by providing unique user ID and password, no one can access the system by using any other's user ID and password.

Scalability

- ❖ The system should be able to handle a higher workload on-demand.

Performance

- ❖ Admin can add, edit, remove, update properties.
- ❖ Any number of users can be able to access the system at the same time and the response of the system regarding to the user requests will be very high.

3.2 Data Requirements:

Patient

- PID
- Fname
- Lname
- Gender
- DOB
- NIC
- Email

Doctor

- DID
- Dname
- Speciality
- Phone_no

Health Check-up

- Date
- Check-upID
- Blood_Sugar_level
- Pressure_level
- Cholesterol_level

Report

- RID
- Report_Type
- Recommendation

Nutritionist

- NID
- Experience
- Nname
- Phone_no

Diet Plan

- PlanID
- Plan_name
- Calorie_goal

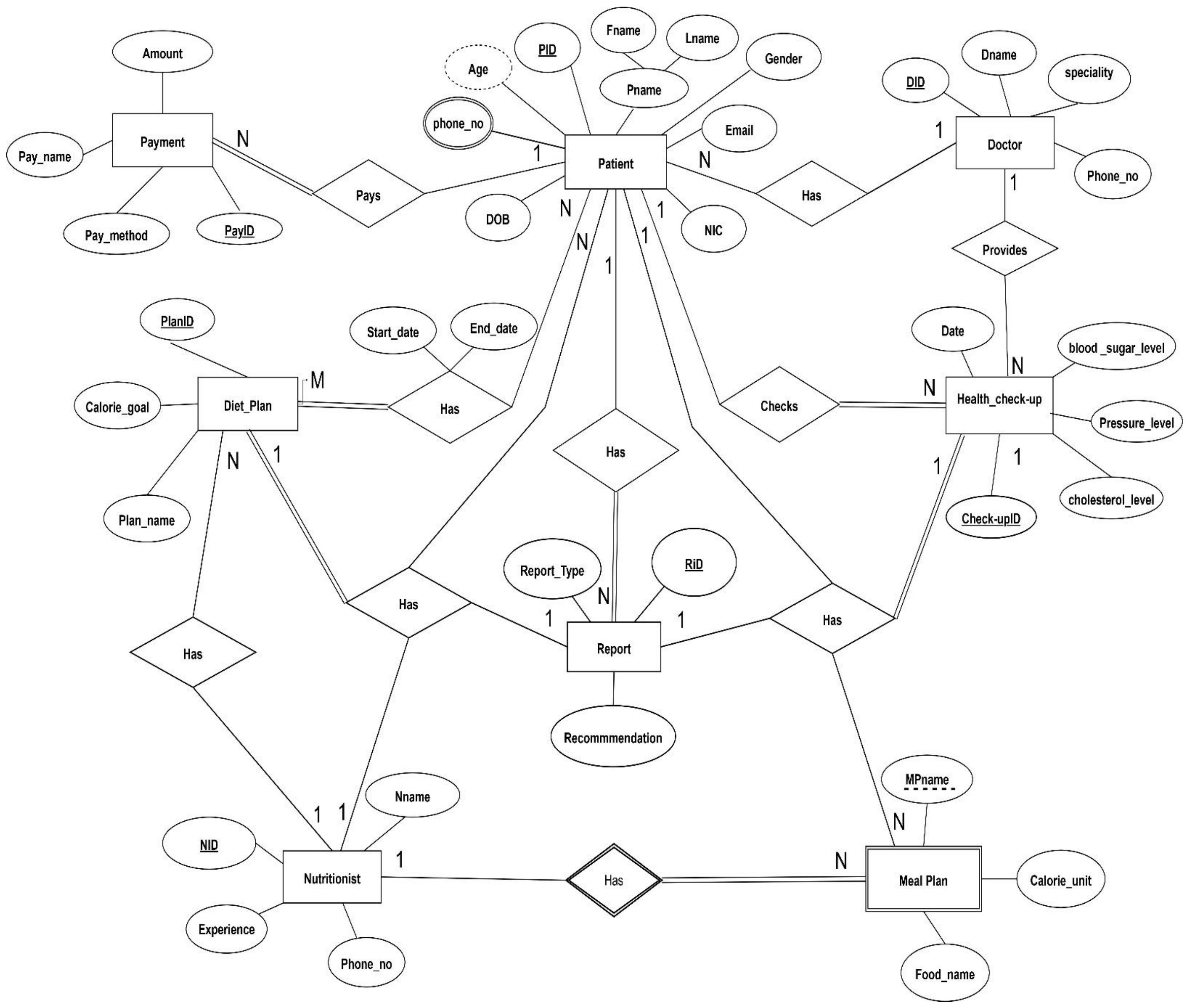
Payment

- PayID
- Pay_name
- Amount
- Pay_method

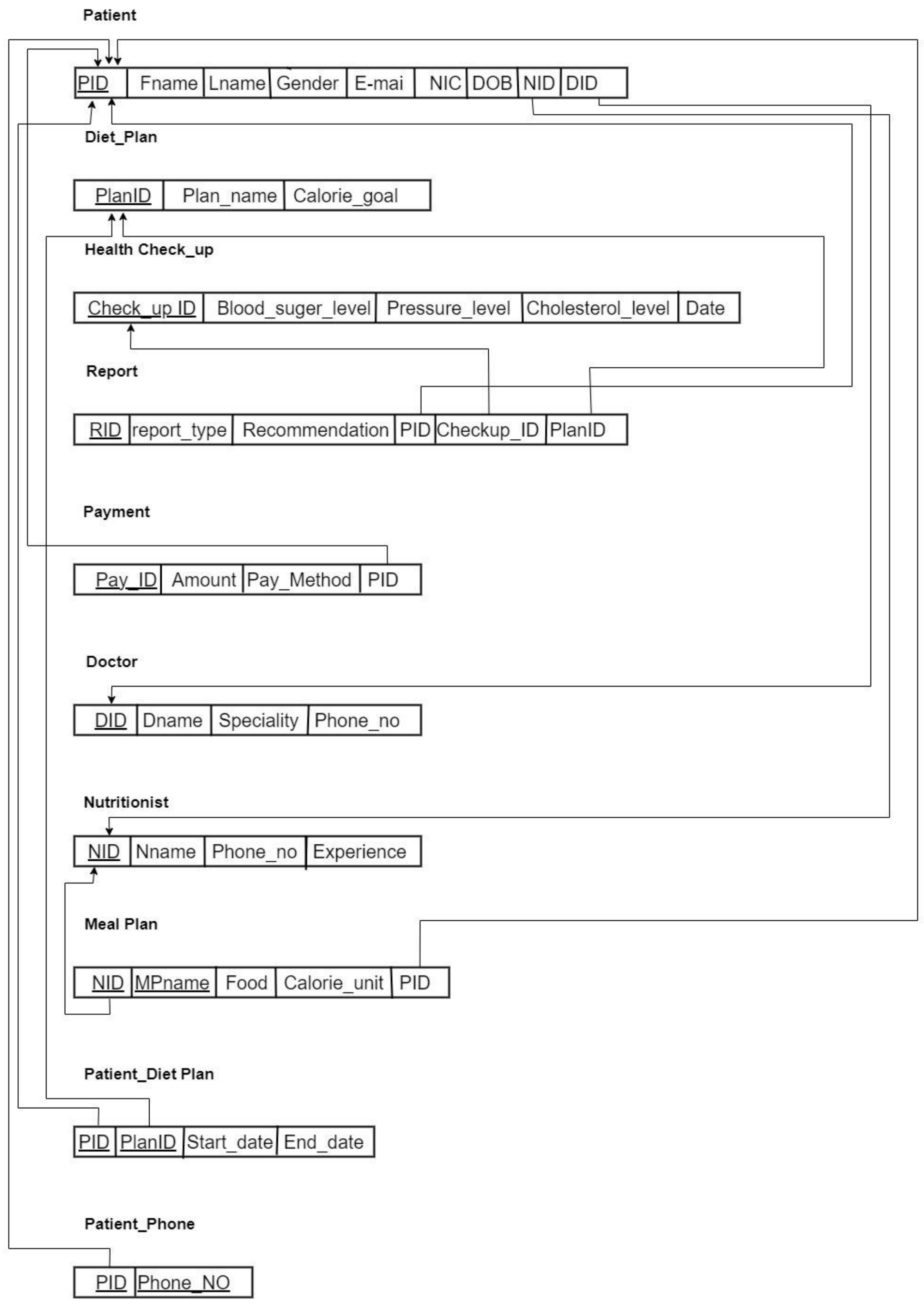
Meal Plan

- MPname
- Food_name
- Calorie_unit

4. Entity Relationship (ER Diagram):



5. Relational Schema :


Patient
PID | Fname | Lname | Gender | E-mai | NIC | DOB | NID | DID
Diet_Plan
PlanID | Plan_name | Calorie_goal
Health Check_up
Check_up ID | Blood_suger_level | Pressure_level | Cholesterol_level | Date
Report
RID | report_type | Recommendation | PID | Checkup_ID | PlanID
Payment
Pay_ID | Amount | Pay_Method | PID
Doctor
DID | Dname | Speciality | Phone_no
Nutritionist
NID | Nname | Phone_no | Experience
Meal Plan
NID | MPname | Food | Calorie_unit | PID
Patient_Diet Plan
PID | PlanID | Start_date | End_date
Patient_Phone
PID | Phone_NO

6.SQL Queries:

6.1 Data Base Create:

```
--Create Database–

CREATE DATABASE Diet_plan_Health_checkup;


--Create Doctor table--

CREATE TABLE Doctor (
  DID VARCHAR(20) NOT NULL,
  Dname VARCHAR(20) NOT NULL,
  Speciality VARCHAR(20),
  Phone_no VARCHAR(20) NOT NULL,

 CONSTRAINT Doctor_PK PRIMARY KEY (DID)

);


--Create Nutritionist table--

CREATE TABLE Nutritionist (

  NID VARCHAR(20) NOT NULL,
  Nname VARCHAR(20) NOT NULL,
  Phone_no INT,
  Experience CHAR(10),

  CONSTRAINT Nutritionist_PK PRIMARY KEY(NID)
);



--Creating Patient table--

CREATE TABLE Patient
(
  PID VARCHAR(20) NOT NULL,
  Fname VARCHAR(20) NOT NULL,
  Lname VARCHAR(20) NOT NULL,
  Gender CHAR(10) NOT NULL,
  DOB DATE NOT NULL,
  NIC CHAR(20) NOT NULL,
  Email VARCHAR(50),
  DID VARCHAR(20) NOT NULL,
  NID VARCHAR(20) NOT NULL,

  CONSTRAINT Patient_PK PRIMARY KEY(PID),
  CONSTRAINT Patient_FK1 FOREIGN KEY(DID) REFERENCES Doctor(DID),
  CONSTRAINT Patient_FK2 FOREIGN KEY(NID) REFERENCES Nutritionist(NID)
);
```

```
--Creating Patient_phone table--

CREATE TABLE Patient_Phone
(
       PID VARCHAR(20) NOT NULL,
       Phone_no VARCHAR(20) NOT NULL,

       CONSTRAINT Patient_Phone_PK PRIMARY KEY(PID,Phone_no),
       CONSTRAINT Patient_Phone_FK FOREIGN KEY(PID) REFERENCES  Patient(PID)

);


--Creating Payment table--

CREATE TABLE Payment (

  Pay_ID VARCHAR(20) NOT NULL,
  Amount DECIMAL(10,2) NOT NULL,
  Pay_method CHAR(20),
  PID VARCHAR(20) NOT NULL,
  CONSTRAINT payment_PK PRIMARY KEY(Pay_ID),
  CONSTRAINT Payment_FK FOREIGN KEY(PID) REFERENCES  Patient(PID)

);


--Creating Diet_Plan table--

CREATE TABLE Diet_Plan (
  PlanID VARCHAR(20) NOT NULL,
  Plan_name VARCHAR(50) NOT NULL,
  Calorie_goal INT NOT NULL,
  NID VARCHAR(20) NOT NULL,

  CONSTRAINT Diet_Plan_PK PRIMARY KEY (PlanID),
  CONSTRAINT Diet_Plan_FK FOREIGN KEY (NID) REFERENCES Nutritionist(NID)
);


--Creating Patient_Diet_Plan table--

CREATE TABLE Patient_Diet_Plan(

  PID VARCHAR(20) NOT NULL,
  PlanID VARCHAR(20) NOT NULL,
  Start_Dates DATE NOT NULL,
  End_Dates DATE NOT NULL,

  CONSTRAINT Patient_Diet_Plan_PK PRIMARY KEY (PID,PlanID),
  CONSTRAINT Patient_Diet_Plan_FK1 FOREIGN KEY (PID) REFERENCES Patient(PID),
  CONSTRAINT Patient_Diet_Plan_FK2 FOREIGN KEY (PlanID) REFERENCES Diet_Plan(PlanID)

);
```

```
--Create Health_Checkup table--


CREATE TABLE Health_Checkup (

  Checkup_ID VARCHAR(20) NOT NULL,
  Dates DATE NOT NULL,
  Pressure_level CHAR(20) NOT NULL,
  blood_sugar_level CHAR(20) NOT NULL,
  PID VARCHAR(20) NOT NULL,
  DID VARCHAR(20) NOT NULL,

  CONSTRAINT Health_Checkup_PK PRIMARY KEY(Checkup_ID),
  CONSTRAINT Health_Checkup_FK1 FOREIGN KEY(PID) REFERENCES  Patient(PID),
  CONSTRAINT Health_Checkup_FK2 FOREIGN KEY(DID) REFERENCES  Doctor(DID)
);



--Create Report table--

CREATE TABLE Report (

  RID VARCHAR(20) NOT NULL,
  Report_Type CHAR(20),
  Recommendation VARCHAR(100),
  PID VARCHAR(20) NOT NULL,
  Checkup_ID VARCHAR(20) NOT NULL,
  PlanID VARCHAR(20) NOT NULL,

  CONSTRAINT Report_PK PRIMARY KEY(RID),
  CONSTRAINT Report_FK1 FOREIGN KEY(PID) REFERENCES Patient(PID),
  CONSTRAINT Report_FK2 FOREIGN KEY(Checkup_ID) REFERENCES Health_Checkup(Checkup_ID),
  CONSTRAINT Report_FK3 FOREIGN KEY(PlanID) REFERENCES Diet_Plan(PlanID)

);



--Create Meal_Plan table--

CREATE TABLE Meal_Plan (

  NID VARCHAR(20) NOT NULL,
  MPname CHAR(50) NOT NULL,
  Food VARCHAR(30) NOT NULL,
  Calorie_Unit INT NOT NULL,
  PID VARCHAR(20) NOT NULL,

  CONSTRAINT Meal_Plan_PK PRIMARY KEY (NID, MPname),
  CONSTRAINT Meal_Plan_FK1 FOREIGN KEY (NID) REFERENCES Nutritionist(NID),
  CONSTRAINT Meal_Plan_FK2 FOREIGN KEY (PID) REFERENCES Patient(PID)
);
```

6.2 Data Store in Data Base:

```
-- Inserting values into the Doctor table –

INSERT INTO Doctor VALUES ('D123', 'Dr.Kumar', 'Cardiology', '0712345678');
INSERT INTO Doctor VALUES ('D258', 'Dr.Perera', 'Pediatrics', '0776543210');
INSERT INTO Doctor VALUES ('D963', 'Dr.Silva', 'Dermatology', '0767890123');
INSERT INTO Doctor VALUES ('D741', 'Dr.Gunaratne', 'Orthopedics', '0723456789');
INSERT INTO Doctor VALUES ('D652', 'Dr.Fernando', 'Internal Medicine', '0756789012');


-- Inserting values into the Nutritionist table –

INSERT INTO Nutritionist VALUES ('N425', 'Ms.Perera', '0712345678', '5 Years');
INSERT INTO Nutritionist VALUES ('N356', 'Mr.Silva', '0776543210', '8 Years');
INSERT INTO Nutritionist VALUES ('N147', 'Dr.Fernando', '0767890123','10 Years');
INSERT INTO Nutritionist VALUES ('N352', 'Ms.Gunawardena', '0723456789', '3 Years');
INSERT INTO Nutritionist VALUES ('N852', 'Mr.Rajapakse', '0756789012', '6 Years');


-- Inserting values into the Patient table --

INSERT INTO Patient  VALUES ('P123','Kamal','Silva','Male','1990-01-
01','900101123V','kamalsilva@gmail.com','D123','N425');
INSERT INTO Patient  VALUES ('P212','Nimali','Fernando','Female','1992-05-
15','920515456V','nimalifernando@gmail.com','D258','N356');
INSERT INTO Patient  VALUES ('P324','Ravi','Perera','Male','1985-09-
22','850922789V','raviperera@gmail.com','D963','N147');
INSERT INTO Patient  VALUES ('P412','Asha','Gunawardena','Female','1988-07-
10','880710890V','ashagunawardena@gmail.com','D741','N352');
INSERT INTO Patient  VALUES ('P521','Saman','Rajapakse','Male','1995-03-
03','950303654V','samanrajapakse@gmail.com','D652','N852');


-- Inserting values into the Meal_Plan table --


INSERT INTO Meal_Plan  VALUES ('N425', 'Breakfast', 'Rice and curry', 100,'P123');
INSERT INTO Meal_Plan  VALUES ('N356', 'Dinner', 'Chicken Breast', 200,'P212');
INSERT INTO Meal_Plan  VALUES ('N147', 'Lunch', 'Brown Rice', 150,'P324');
INSERT INTO Meal_Plan  VALUES ('N352', 'Desserts', 'Fruit salad ', 50,'P412');
INSERT INTO Meal_Plan  VALUES ('N852', 'Snacks', 'Fresh fruit', 120,'P521');


-- Inserting values into the Patient_Phone table --

INSERT INTO Patient_Phone  VALUES ('P123', '0712345678');
INSERT INTO Patient_Phone  VALUES ('P123', '0767221436');
INSERT INTO Patient_Phone  VALUES ('P212', '0776543210');
INSERT INTO Patient_Phone  VALUES ('P324', '0767890123');
INSERT INTO Patient_Phone  VALUES ('P324', '0702067686');
INSERT INTO Patient_Phone  VALUES ('P412', '0723456789');
INSERT INTO Patient_Phone  VALUES ('P521', '0756789012');
INSERT INTO Patient_Phone  VALUES ('P521', '0726689052');
```

```sql
-- Inserting values into the Payment table --

INSERT INTO Payment  VALUES ('R123', 50.00,'Cash','P123');
INSERT INTO Payment  VALUES ('R245', 100.00,'Credit Card','P212');
INSERT INTO Payment  VALUES ('R367', 75.00,'Cash','P324');
INSERT INTO Payment  VALUES ('R432', 120.00,'Credit Card','P412');
INSERT INTO Payment  VALUES ('R521', 80.00,'Cash','P521');



-- Inserting values into the Diet_Plan table --

INSERT INTO Diet_Plan  VALUES ('DP421', 'Weight Loss Plan',2000, 'N425');
INSERT INTO Diet_Plan  VALUES ('DP258', 'Muscle Building Plan', 2500 ,'N356');
INSERT INTO Diet_Plan  VALUES ('DP254', 'Healthy Eating Plan', 1800, 'N147');
INSERT INTO Diet_Plan  VALUES ('DP631', 'Low Carb Plan', 2200,'N352');
INSERT INTO Diet_Plan  VALUES ('DP592', 'Vegetarian Plan', 1600,'N852');




-- Inserting values into the Patient_Diet_Plan table --

INSERT INTO Patient_Diet_Plan VALUES ('P123','DP421','2023-01-01','2023-02-28');
INSERT INTO Patient_Diet_Plan VALUES ('P212','DP258','2023-03-01','2023-04-30');
INSERT INTO Patient_Diet_Plan VALUES ('P324','DP254','2023-05-01','2023-06-30');
INSERT INTO Patient_Diet_Plan VALUES ('P412','DP631','2023-07-01','2023-08-31');
INSERT INTO Patient_Diet_Plan VALUES ('P521','DP592','2023-09-01','2023-10-31');



-- Inserting values into the Health_Checkup table --

INSERT INTO Health_Checkup VALUES ('CHK15625', '2023-01-15', 'Normal', '120
mg/dL','P123', 'D123');
INSERT INTO Health_Checkup VALUES ('CHK25625', '2023-03-20', 'High', '150 mg/dL','P212',
'D258');
INSERT INTO Health_Checkup VALUES ('CHK15422', '2023-05-25', 'Normal', '110
mg/dL','P324', 'D963');
INSERT INTO Health_Checkup VALUES ('CHK35628', '2023-07-30', 'Low', '80 mg/dL','P412',
'D741');
INSERT INTO Health_Checkup VALUES ('CHK45625', '2023-09-05', 'Normal', '130
mg/dL','P521', 'D652');



-- Inserting values into the Report table --

INSERT INTO Report  VALUES ('R001', 'Cardiac', 'Maintain a healthy diet and exercise
regularly.', 'P123','CHK15625','DP421');
INSERT INTO Report  VALUES ('R002', 'Pediatric', 'Ensure proper nutrition and
vaccinations.', 'P212','CHK25625','DP258');
INSERT INTO Report  VALUES ('R003', 'Dermatology', 'Follow skincare routine and avoid
allergens.','P324','CHK15422','DP254');
INSERT INTO Report  VALUES ('R004', 'Orthopedic', 'Physical therapy and rest for faster
recovery.','P412','CHK35628','DP631');
INSERT INTO Report  VALUES ('R005', 'Internal Medicine', 'Monitor blood sugar levels and
maintain a balanced diet.','P521','CHK45625','DP592');
```

7.Performance Requirement:

both diet planning and health check-up systems can have performance requirements. Performance requirements are criteria or specifications that define how well a system should perform in terms of speed, efficiency, responsiveness, and other relevant factors. In the context of diet planning and health check-up systems, some common performance requirements may include:

- **Responsiveness:** The system should respond quickly to user input and provide timely results. For example, when a user enters their health information or dietary preferences, the system should process the data and generate personalized recommendations within a reasonable time frame.

- **Scalability:** The system should be able to handle a growing number of users and data without significant performance degradation. As more users join the system or the amount of health data increases, the system should be able to accommodate the increased workload without becoming slow or unresponsive.

- **Accuracy:** The system's recommendations and health check-up results should be accurate and reliable. Users rely on these systems for making informed decisions about their diet and health, so it is crucial that the information provided is trustworthy.

- **Efficiency:** The system should utilize computational resources efficiently. This includes optimizing algorithms and data structures to minimize processing time and reduce resource consumption, such as CPU and memory usage.

- **Availability:** The system should be available and accessible to users whenever they need it. It should have minimal downtime or scheduled maintenance periods that might interrupt users' access to the system.

These performance requirements ensure that the diet planning and health check-up systems deliver a fast, accurate, and reliable service to users, enabling them to make informed decisions about their diet and monitor their health effectively.

8.Security Requirements:

Database system security is essential for preventing unwanted access to sensitive data and preserving the availability and integrity of the data. Access control establishes the degree of permissions for users and guarantees that only those with authorization can access the database. By making the data unintelligible without the right decryption key, data encryption protects the data and ensures its secrecy. Database activity is tracked via auditing, allowing for the identification of questionable activity and preserving data integrity. Data backup makes duplicates of the database, enabling data recovery in the event of a breach and ensuring data accessibility.

Security of the database system is crucial for avoiding unauthorized access to private information and maintaining the availability and integrity of the data. The degree of user permissions is determined by access control, which also ensures that only those with authorization can access the database. Data encryption protects the data and guarantees its confidentiality by rendering it unreadable without the proper decryption key. By monitoring database activity through auditing, it is possible to spot suspicious activity and protect the integrity of the data. Data backup creates copies of the database, maintaining data accessibility and enabling data recovery in the event of a breach.

Organizations may create a strong database security framework that safeguards their important data from unwanted access, loss, or compromise by combining these security systems and best practices.