**DIET ROUTINE**

1. An appropriate software engineering methodology for my system would be incremental development. The incremental methodology is well suited for projects that require flexibility and adaptability, as it emphasizes collaboration between cross-functional teams, continuous delivery, and rapid prototyping.

a. Context Models:

- Use Case Model:

- Use Case: Request Personalized Diet Plan

- Actors: User, Healthcare Provider

- Description: User requests a personalized diet plan based on their health status and dietary restrictions/recommendations provided by healthcare providers.

Request personalized Diet plan

Provide personalized Diet plan

User/Patient

Health Provider

**Diet-Routine System**

- System Context Diagram:

- External Entities: User, Healthcare Provider

- Relationships: User interacts with the system to request a personalized diet plan, and the system interacts with the healthcare provider to gather dietary restrictions/recommendations.

Request personilazed Diet plan

**Diet-Routine**

**system**

Dietry restriction

Health Provider

- Stakeholder Map:

- Stakeholders: Users, Healthcare Providers, Development Team

- Relationships: Users interact with the system, healthcare providers provide dietary recommendations, and the development team builds and maintains the system.

b. Interaction Models:

- Sequence Diagram:

- Description: This diagram illustrates the sequence of interactions between the user, healthcare provider, and the system when a user requests a personalized diet plan.

- Participants: User, Healthcare Provider, System

- Actions: User provides health symptoms and dietary restrictions, healthcare provider provides recommendations, and the system generates a personalized diet plan.

Provide health sysmptoms

Provide dietry restriction/recomendation

- Use Case Diagram:

- Use Cases: User Registration, Symptom Checking, Dietary Recommendations, Personalized Diet Plan

- Relationships: User interacts with each use case as needed, and the system coordinates the interactions between the use cases.

c. Structural Models:

- Class Diagram:

- Classes: User, Healthcare Provider, Symptoms, Dietary Restrictions, Food Items

- Relationships: User has health information and dietary preferences, healthcare provider provides recommendations, symptoms guide dietary decisions, and food items contribute to the diet plan.

User/Patient

-Health information

-Dietry prefences

Food Items

-Contribute to diet plan

Symptoms

-Guide dietry decision

- Entity-Relationship Diagram:

- Entities: User, Healthcare Provider, Symptoms, Dietary Restrictions, Food Items, Database

- Relationships: User has health information and dietary preferences, healthcare provider provides recommendations, symptoms affect dietary decisions, food items are stored in the database.

d. Behavioral Models:

- Activity Diagram:

- Description: This diagram illustrates the flow of activities when a user requests a personalized diet plan, including symptom checking, dietary recommendation, and diet plan generation.

- States: User Registration, Symptom Checking, Dietary Recommendation, Diet Plan Generation

- Transitions: User progresses through the states based on their actions and system responses.

Symptom checking

User registration

No symptom

Symptom

discoverd

Dietry recommendation

- State Diagram:

- Description: This diagram represents the different states of the system and how it transitions between them based on user interactions.

- States: User Registration, Symptom Checking, Dietary Recommendation, Diet Plan Generation

- Transitions: User actions trigger transitions between the states.