**DBMS-2006 (254354) Final Project Milestone One**

**Project Proposal**

Developer Name: Ma Crizza Lynne Regacho

Project Name: NutriSenior

Date: 2024-11-17

1. Project Description

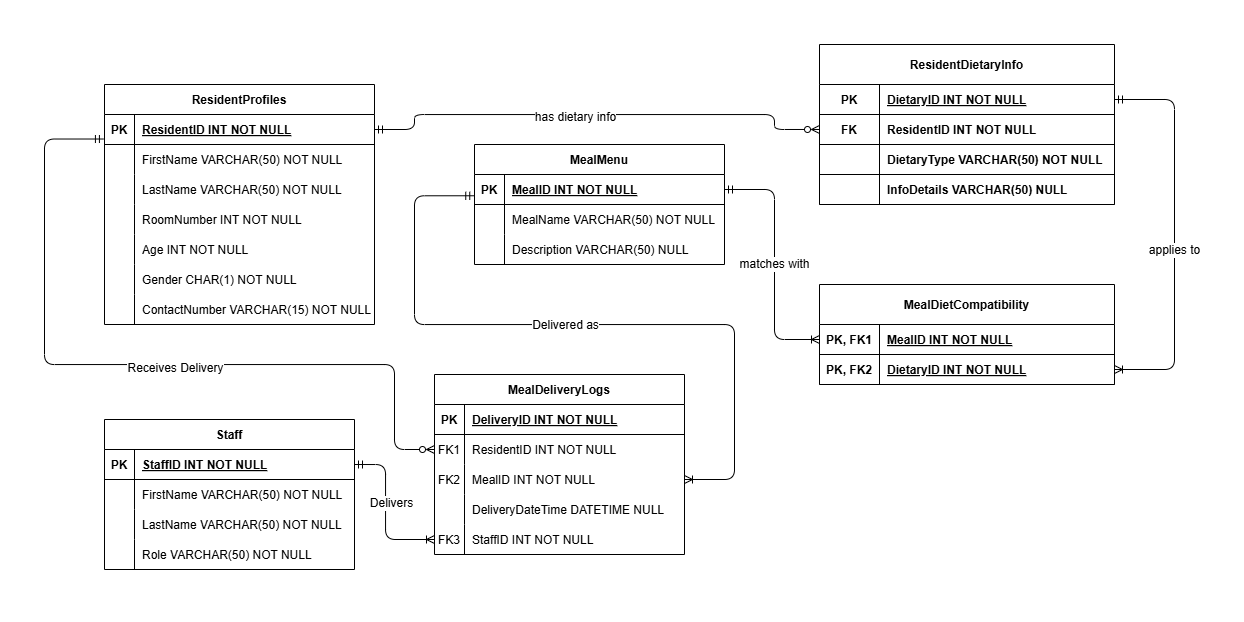
NutriSenior is a dedicated project designed to efficiently manage the dietary needs of senior residents in an assisted living home for independent seniors. Utilizing database, this project focuses on tracking and organizing key information such as residents' dietary preferences, allergies, meal deliveries, and the staff responsible for these services. The database includes essential details like resident information such as age, gender, room number, and contact information, as well as thorough records on food preferences, allergies, and meal deliveries. The primary goal of NutriSenior is to ensure that each resident's dietary requirements are met accurately and promptly, thereby enhancing their overall health and satisfaction. This project was chosen because I work in an assisted living home and have personally witnessed the importance of personalized nutrition for senior residents. My empathy for their needs and my desire to improve their quality of life inspired me to create a solution that addresses these critical aspects of their daily care.

1. Business Case

The NutriSenior project is required to effectively manage the dietary requirements of elderly residents in an assisted living home. Accurate documentation of dietary preferences, allergies, and meal deliveries is essential in ensuring that senior residents receive meals tailored to their specific needs, which improves their health and satisfaction. This project addresses common challenges faced by facility workers, such as handling complex dietary requirements and guaranteeing timely food delivery, with the goal of reducing the risk of errors and boosting operational efficiency.

This project's target audience consists of assisted living home personnel and administrators who manage residents' dietary needs, as well as residents who benefit from personalized nutrition and improved quality of life. NutriSenior ensures that dietary requirements are satisfied consistently and accurately resulting in a healthier and more comfortable living environment for senior residents.

1. Business Rules / Assumptions
2. Resident may have zero to many dietary preferences recorded in the ResidentFoodPreferences table.
3. Each dietary preference in the ResidentFoodPreferences table must link to one specific preference in the DietaryPreferences table
4. Each meal in the MealMenu table can support **one to many** dietary preferences, recorded in the MealDietCompatibility table
5. Each compatibility record in MealDietCompatibility links a specific meal to one dietary preference in the DietaryPreferences table.
6. A resident may have **zero to many** allergy records listed in the ResidentAllergyRecords table.
7. Each allergy record in ResidentAllergyRecords must link to one specific allergy type in the AllergyTypes table.
8. A resident can have zero to many meal deliveries recorded in the MealDeliveryLogs table.
9. Each meal in the MealMenu table can appear in multiple delivery logs in the MealDeliveryLogs table.
10. Each staff member in the StaffDirectory table can be responsible for **one to many** meal deliveries, recorded in the MealDeliveryLogs table.
11. A resident can provide **zero to many** feedback records for meals they have received, recorded in the ResidentMealFeedback table.
12. Each meal in the MealMenu table can receive feedback from **zero to many** residents, recorded in the ResidentMealFeedback table.
13. ENTITY RELATIONSHIP DIAGRAM



**Describe each entity and how they relate to each other in both directions**

1. Staff

* Description: Stores staff member details such as First name, last name, role, and contact information.
* Relation to MealDeliveryLogs:
  + Direction 1 (Staff to MealDeliveryLogs): Each staff member can be responsible for many meal deliveries.
  + Direction 2 (MealDeliveryLogs to Staff): Each meal delivery is associated with one specific staff member.

2. MealDeliveryLogs

* Description: Tracks the meals delivered, the resident who received them, and the staff member who delivered them.
* Relation to ResidentProfiles:
  + Direction 1 (ResidentProfiles to MealDelivery): Each resident can receive multiple meal deliveries.
  + Direction 2 (MealDelivery to Resident): Each meal delivery is associated with one specific resident.
* Relation to MealMenu:
  + Direction 1 (MealMenu to MealDelivery): Each meal from the MealMenu can be delivered multiple times.
  + Direction 2 (MealDeliveryLogs to MealMenu): Each meal delivery corresponds to one specific meal from the MealMenu.

3. MealMenu

* Description: Stores the details of meals available in the system (e.g., name, description).
* Relation to MealDietCompatibility:
  + Direction 1 (MealMenu to MealDietCompatibility): Each meal can support multiple dietary preferences.
  + Direction 2 (MealDietCompatibility to MealMenu): Each dietary compatibility record links to one specific meal.
* Relation to MealDeliveryLogs:
  + Direction 1 (MealMenu to MealDelivery): Each meal in the MealMenu can appear in multiple delivery logs.
  + Direction 2 (MealDelivery to MealMenu): Each meal delivery is linked to one specific meal from the MealMenu.

4. ResidentProfiles

* Description: Contains personal information about each resident, such as name, age, and contact details.
* Relation to ResidentFoodPreferences:
  + Direction 1 (Resident to ResidentFoodPreferences): A resident can have zero or more food preferences.
  + Direction 2 (ResidentFoodPreferences to Resident): Each food preference belongs to a specific resident.
* Relation to ResidentAllergyRecords:
  + Direction 1 (Resident to ResidentAllergyRecords): A resident can have zero or more allergy records.
  + Direction 2 (ResidentAllergyRecords to Resident): Each allergy record is linked to a specific resident.
* Relation to MealDeliveryLogs:
  + Direction 1 (Resident to MealDeliveryLogs): A resident can receive zero or multiple meals, tracked in the MealDeliveryLogs.
  + Direction 2 (MealDeliveryLogs to Resident): Each meal delivery corresponds to one specific resident.

5. ResidentFoodPreferences

* Description: Links residents to their dietary preferences (e.g., vegetarian, gluten-free).
* Relation to ResidentProfiles:
  + Direction 1 (Resident to ResidentFoodPreferences): A resident can zero or multiple food preferences.
  + Direction 2 (ResidentFoodPreferences to Resident): Each food preference is linked to one specific resident.
* Relation to DietaryPreferences:
  + Direction 1 (ResidentFoodPreferences to DietaryPreferences): A dietary preference for a resident corresponds to one specific preference in the DietaryPreferences table.
  + Direction 2 (DietaryPreferences to ResidentFoodPreferences): A dietary preference can be applied to zero to multiple residents.

6. DietaryPreferences

* Description: Lists all possible dietary preferences (e.g., vegetarian, gluten-free).
* Relation to ResidentFoodPreferences:
  + Direction 1 (ResidentFoodPreferences to DietaryPreferences): Each dietary preference record in ResidentFoodPreferences must link to a specific dietary preference in the DietaryPreferences table.
  + Direction 2 (DietaryPreferences to ResidentFoodPreferences): Each dietary preference can apply to zero to multiple residents.
* Relation to MealDietCompatibility:
  + Direction 1 (DietaryPreferences to MealDietCompatibility): Each dietary preference can be associated with one to many meals.
  + Direction 2 (MealDietCompatibility to DietaryPreferences): Each compatibility record links a specific dietary preference to one meal.

7. MealDietCompatibility

* Description: Links meals from the MealMenu table with dietary preferences from the DietaryPreferences table.
* Relation to MealMenu:
  + Direction 1 (MealMenu to MealDietCompatibility): Each meal can be supports to one to multiple dietary preferences.
  + Direction 2 (MealDietCompatibility to MealMenu): Each compatibility record links to one specific meal.
* Relation to DietaryPreferences:
  + Direction 1 (MealDietCompatibility to DietaryPreferences): Each compatibility record links one specific meal to a dietary preference.
  + Direction 2 (DietaryPreferences to MealDietCompatibility): Each dietary preference can be applied to multiple meals.

8. ResidentAllergyRecords

* Description: Stores the allergies of each resident.
* Relation to ResidentProfiles:
  + Direction 1 (Resident to ResidentAllergyRecords): A resident can have zero or more allergy records.
  + Direction 2 (ResidentAllergyRecords to Resident): Each allergy record belongs to a specific resident.
* Relation to AllergyTypes:
  + Direction 1 (ResidentAllergyRecords to AllergyTypes): Each allergy record links to one specific allergy type.
  + Direction 2 (AllergyTypes to ResidentAllergyRecords): Each allergy type can be linked to multiple allergy records.

9. AllergyTypes

* Description: Lists the types of allergies (e.g., peanuts, shellfish).
* Relation to ResidentAllergyRecords:
  + Direction 1 (AllergyTypes to ResidentAllergyRecords): An allergy type can be associated with multiple allergy records.
  + Direction 2 (ResidentAllergyRecords to AllergyTypes): Each allergy record links to one specific allergy type.

10. ResidentMealFeedback

* Description: Stores feedback on meals provided to residents, which includes comments.
* Relation to ResidentProfiles:
  + Direction 1 (Resident to ResidentMealFeedback): A resident can provide feedback on multiple meals.
  + Direction 2 (ResidentMealFeedback to Resident): Each feedback record is associated with one specific resident.
* Relation to MealMenu:
  + Direction 1 (MealMenu to ResidentMealFeedback): Each meal can receive feedback from multiple residents.
  + Direction 2 (ResidentMealFeedback to MealMenu): Each feedback record is linked to one specific meal

**Explain why the table was created**

1. StaffDirectory

* Reason for Creation: To track and manage staff involved in meal deliveries.

1. MealDeliveryLogs

* Reason for Creation: To log who delivered what meal to which resident and when.

1. MealMenuPurpose:

* Reason for Creation: To provide a catalog of meals that can be linked to deliveries, and meal diet compatibility.

1. ResidentProfiles

* Reason for Creation: To keep track of residents, their preferences, allergies, and meal deliveries.

1. ResidentFoodPreferences

* Reason for Creation: To ensure residents receive meals matching their dietary needs.

1. DietaryPreferences

* Reason for Creation: To define and standardize dietary preferences for residents.

1. MealDietCompatibility

* Reason for Creation: To manage the many-to-many relationship between meals and dietary preferences.

1. ResidentAllergyRecords

* Reason for Creation: To track allergens and prevent residents from receiving harmful meals.

1. AllergyTypes

* Reason for Creation: To standardize the types of allergens for consistency in tracking allergies.

1. ResidentMealFeedback

* Reason for Creation: To collect and analyze feedback for improving meal quality.

**If there were any bridge or joiner tables created explain why the table was created?**

**1. ResidentFoodPreferences**

This is a **joiner table** that connects the **ResidentProfiles** table to the **DietaryPreferences** table.

* A resident can have multiple food preferences, and each dietary preference can apply to multiple residents. Therefore, this many-to-many relationship is managed by this joiner table.

**2. MealDietCompatibility**

This is a **bridge table** that links the **MealMenu** table to the **DietaryPreferences** table.

* Each meal may be compatible with multiple dietary preferences (e.g., vegetarian, gluten-free), and each dietary preference can apply to multiple meals. This many-to-many relationship is handled by the **MealDietCompatibility** table.

**3. ResidentAllergyRecords**

This is another **joiner table** that connects the **ResidentProfiles** table to the **AllergyTypes** table.

* A resident can have multiple allergies, and each allergy can apply to multiple residents. This many-to-many relationship requires a bridge table to properly link the two entities.

**Explain what PK was created for the joiner/bridge table and what makes it a good primary key**

1. ResidentFoodPreferences:

* Primary Key: ResidentID + PreferenceID (composite key)
  + Why? Combination of ResidentID and PreferenceID uniquely identifies each resident-food preference pairing, ensuring that each resident can only have one instance of a given dietary preference

2. MealDietCompatibility:

* Primary Key: MealID + PreferenceID (composite key)
  + Why? The combination of MealID and PreferenceID ensures that each meal is only linked to a specific dietary preference once, and no duplicate pairs are recorded in the table.

3. ResidentAllergyRecords:

* Primary Key: ResidentID + AllergyID (composite key)
  + Why? The combination of ResidentID and AllergyID ensures that each resident is linked to a specific allergy only once. It avoids the possibility of a resident being assigned the same allergy multiple times

**Explain FK’s in the bridge table and their purpose**

1. ResidentFoodPreferences:

* Foreign Keys:
  + ResidentID: References the ResidentID in the ResidentProfiles table.
  + PreferenceID: References the PreferenceID in the DietaryPreferences table.

Purpose of FKs:

* ResidentID: Links each resident to the specific dietary preference they have.
* PreferenceID: Links each dietary preference to a specific resident.

2. MealDietCompatibility:

* Foreign Keys:
  + MealID: References the MealID in the MealMenu table.
  + PreferenceID: References the PreferenceID in the DietaryPreferences table.

Purpose of FKs:

* MealID: Links a specific meal to its dietary compatibility.
* PreferenceID: Links a specific dietary preference to the meals that are compatible with it.

3. ResidentAllergyRecords:

* Foreign Keys:
  + ResidentID: References the ResidentID in the ResidentProfiles table.
  + AllergyID: References the AllergyID in the AllergyTypes table.

Purpose of FKs:

* ResidentID: Links a resident to their allergy records.
* AllergyID: Links an allergy to a specific resident.