Foodi: Food Allergy system

By Group 11

Who are we

Rami Yassein Eltayeb

Ammar Abdulrahman Anaam Mudhsh

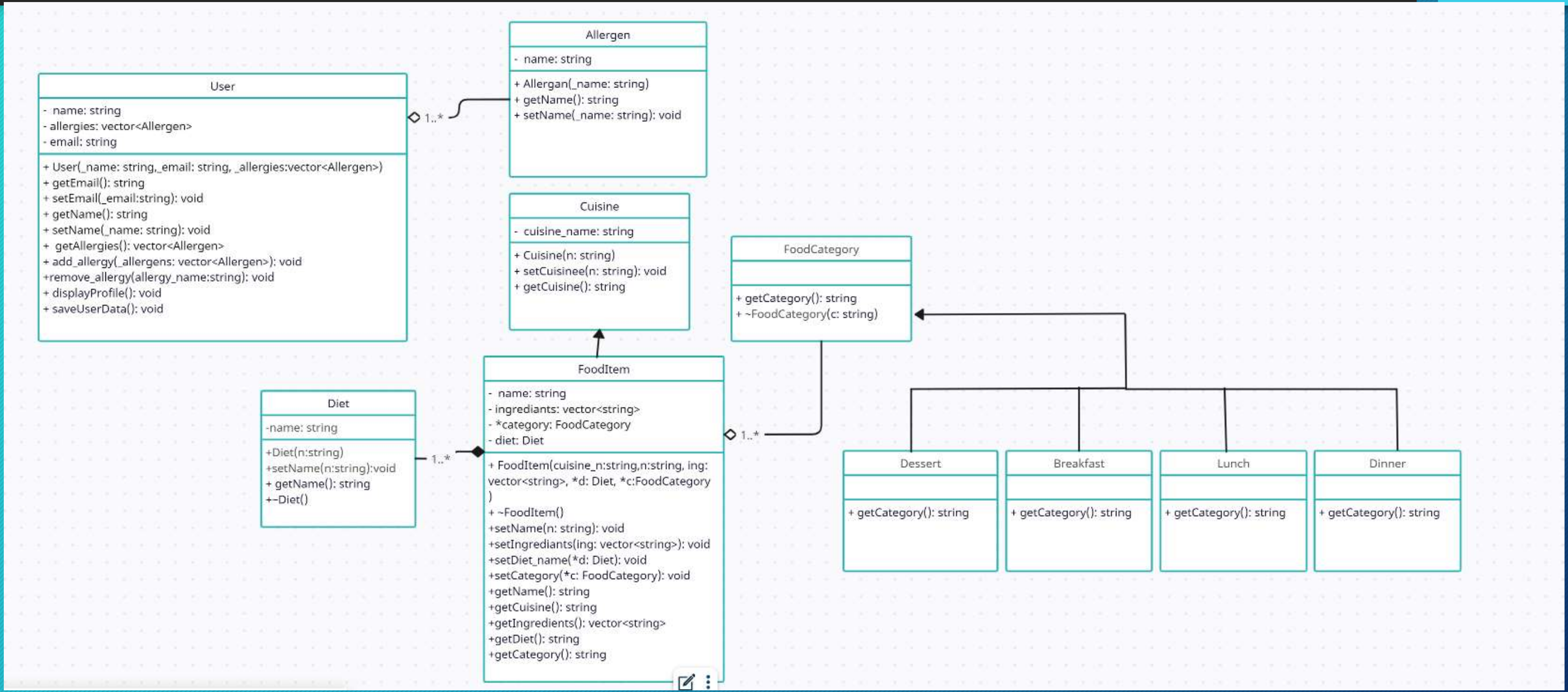
Mohamed Ali Mohamed Ali

About Foodi.

- Foodi is a Food allergy Alert system that enables the user to compare the ingredients of food items to their list of allergens and alert them if the allergens are present. Our objectives for creating this system are:

- To Help users Avoid Any food that may triggers their Allergies.
- To Help keeping people safe because some Allergies can lead to death.
- To provide a fast and easy to use Tool that gives specific and concise information about Allergies based on each Unique case.

UML Class Diagram.



User

- name: string
- allergies: vector<Allergen>
- email: string

+ User(_name: string,_email: string, _allergies:vector<Allergen>)
+ getEmail(): string
+ setEmail(_email:string): void
+ getName(): string
+ setName(_name: string): void
+ getAllergies(): vector<Allergen>
+ add_allergy(_allergens: vector<Allergen>): void
+remove_allergy(allergy_name:string): void
+ displayProfile(): void
+ saveUserData(): void

Allergen

- name: string

+ Allergan(_name: string)
+ getName(): string
+ setName(_name: string): void

Cuisine

- cuisine_name: string

+ Cuisine(n: string)
+ setCuisinee(n: string): void
+ getCuisine(): string

FoodCategory

+ getCategory(): string
+ ~FoodCategory(c: string)

Diet

-name: string

+Diet(n:string)
+setName(n:string):void
+ getName(): string
+~Diet()

FoodItem

- name: string
- ingrediants: vector<string>
- *category: FoodCategory
- diet: Diet

+ FoodItem(cuisine_n:string,n:string, ing: vector<string>, *d: Diet, *c:FoodCategory)
+ ~FoodItem()
+setName(n: string): void
+setIngrediants(ing: vector<string>): void
+setDiet_name(*d: Diet): void
+setCategory(*c: FoodCategory): void
+getName(): string
+getCuisine(): string
+getIngredients(): vector<string>
+getDiet(): string
+getCategory(): string

Dessert

+ getCategory(): string

Breakfast

+ getCategory(): string

Lunch

+ getCategory(): string

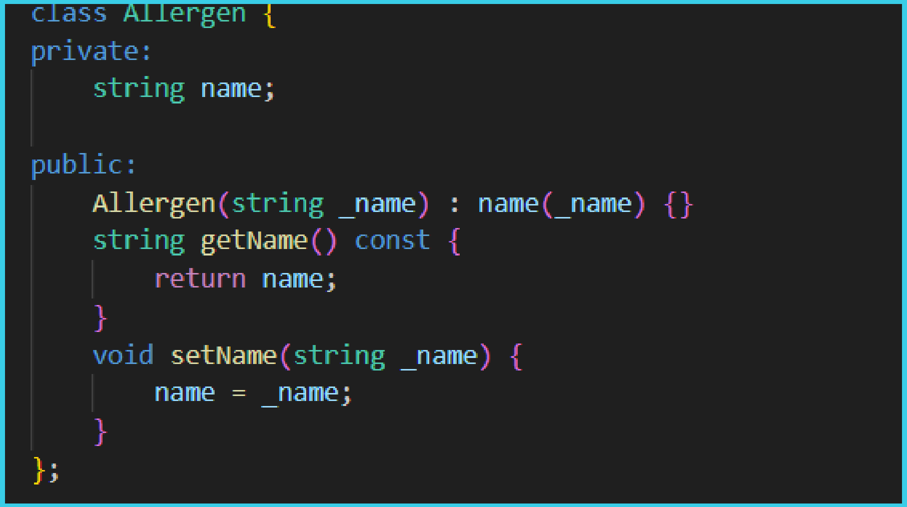
Dinner

+ getCategory(): string

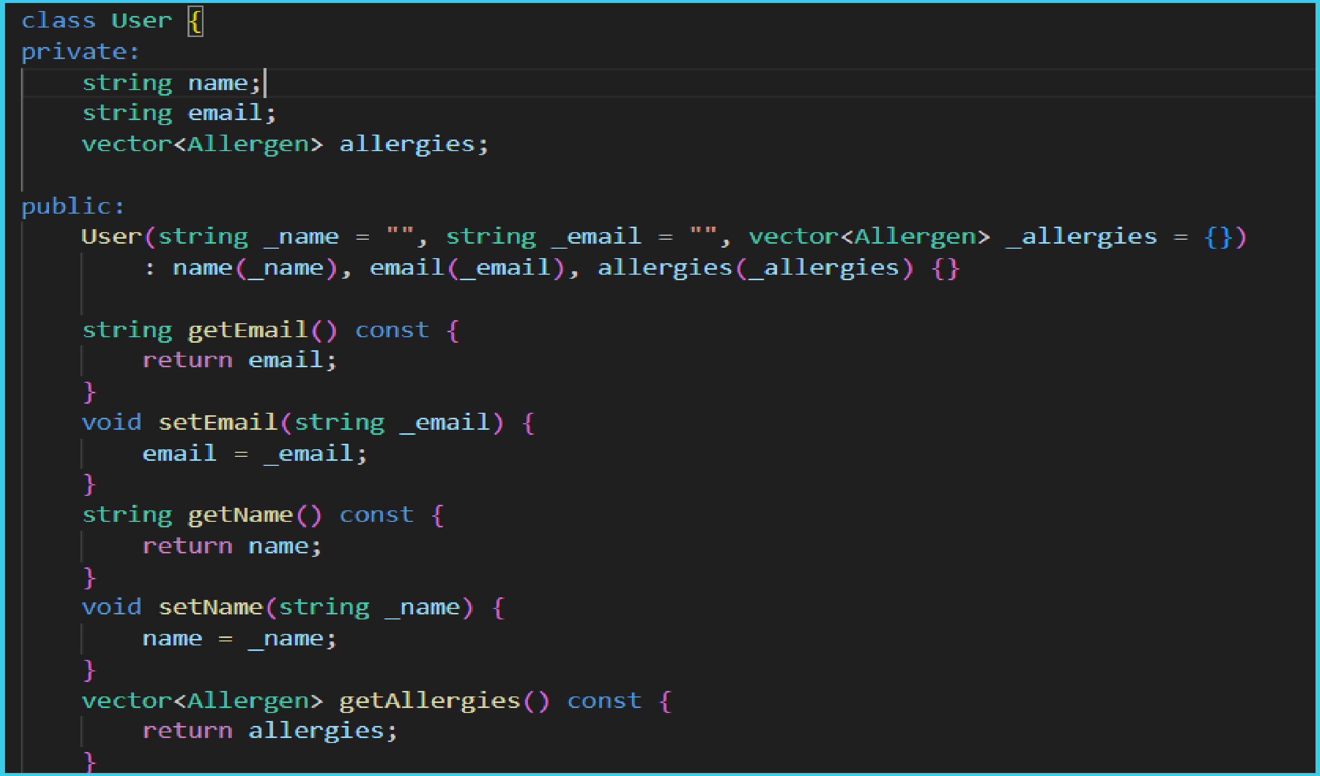
OOP Concepts: Aggregation

- Aggregation relationship between User and Allergens.

- “Has a” relationship.



```
class Allergen {
private:
    string name;

public:
    Allergen(string _name) : name(_name) {}
    string getName() const {
        return name;
    }
    void setName(string _name) {
        name = _name;
    }
};
```



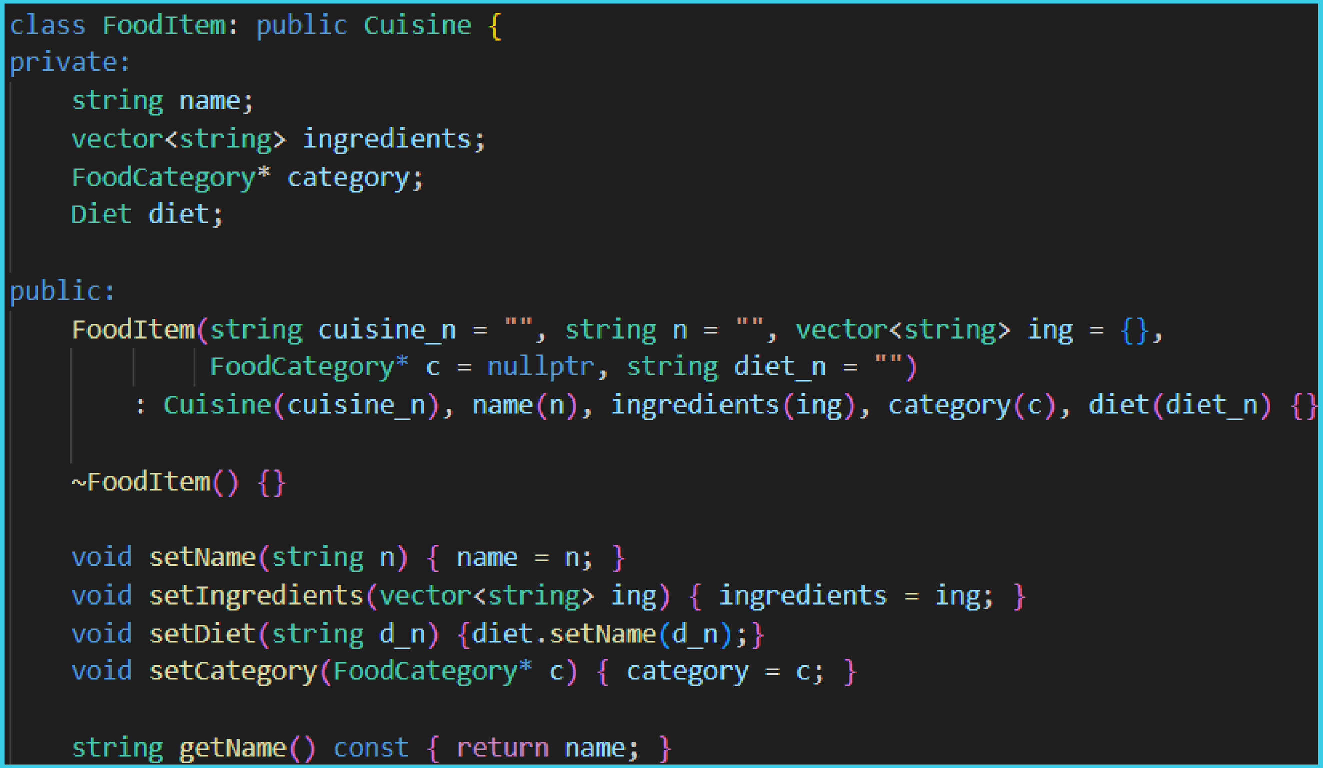
```
class User {
private:
    string name;
    string email;
    vector<Allergen> allergies;

public:
    User(string _name = "", string _email = "", vector<Allergen> _allergies = {})
        : name(_name), email(_email), allergies(_allergies) {}

    string getEmail() const {
        return email;
    }
    void setEmail(string _email) {
        email = _email;
    }
    string getName() const {
        return name;
    }
    void setName(string _name) {
        name = _name;
    }
    vector<Allergen> getAllergies() const {
        return allergies;
    }
```

OOP Concepts: Aggregation, Composition and Inheritance.

- Aggregation relationship between FoodItem and FoodCategory, It is a “has a” relationship.

- Composition relationship between FoodItem and Diet.

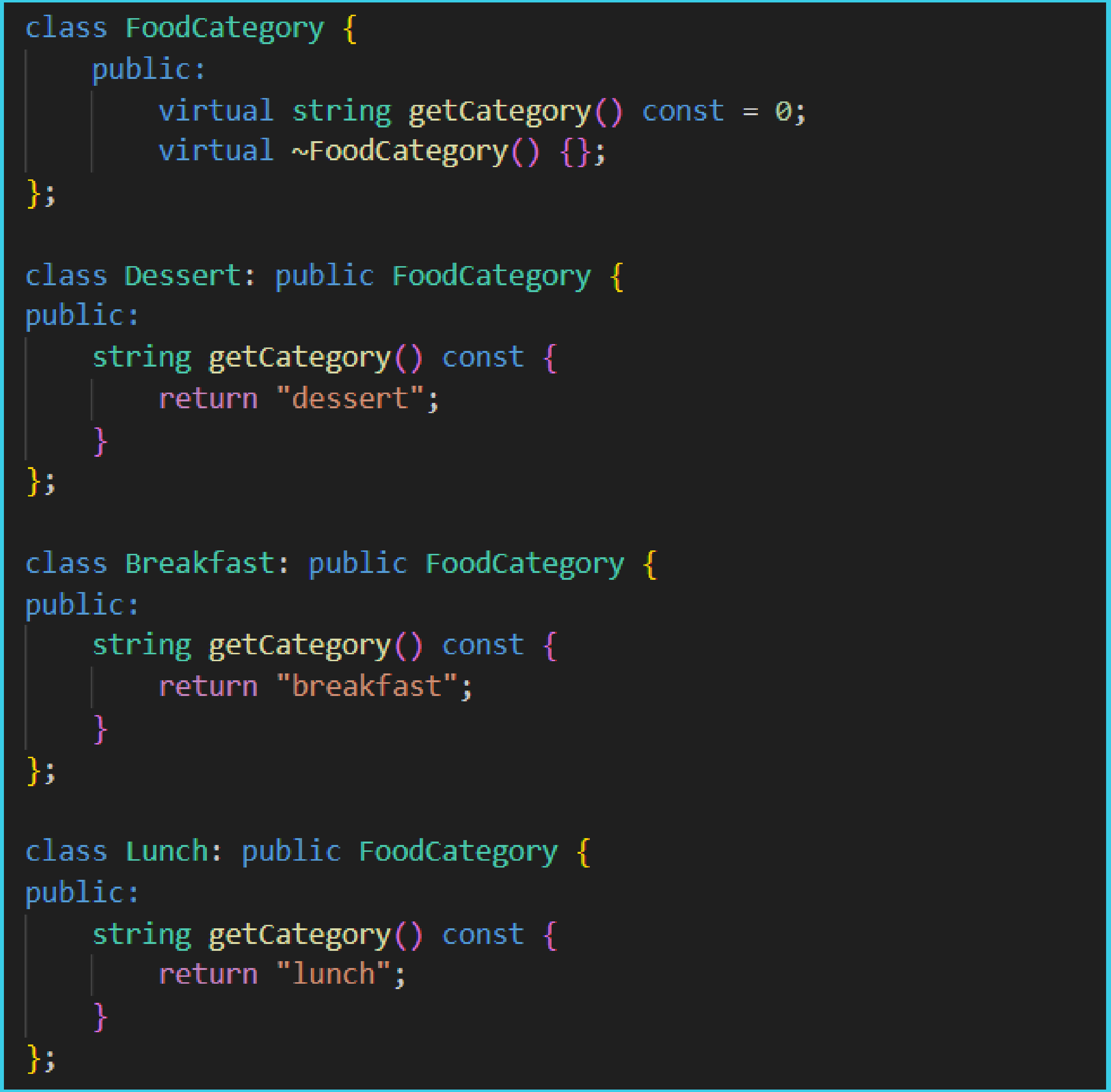
- FoodItem Inherits from Cuisine. This creates an Inheritance relationship.



```
class FoodItem: public Cuisine {
private:
    string name;
    vector<string> ingredients;
    FoodCategory* category;
    Diet diet;

public:
    FoodItem(string cuisine_n = "", string n = "", vector<string> ing = {},
             FoodCategory* c = nullptr, string diet_n = "")
        : Cuisine(cuisine_n), name(n), ingredients(ing), category(c), diet(diet_n) {}

    ~FoodItem() {}

    void setName(string n) { name = n; }
    void setIngredients(vector<string> ing) { ingredients = ing; }
    void setDiet(string d_n) {diet.setName(d_n);}
    void setCategory(FoodCategory* c) { category = c; }

    string getName() const { return name; }
```

OOP Concepts: Polymorphism

- Dessert, Breakfast, Lunch, and Dinner all Inherit the method getCategory() from the class Food Category to implement it in a different way.
- This type of Inheritance creates polymorphism.



```
class FoodCategory {
    public:
        virtual string getCategory() const = 0;
        virtual ~FoodCategory() {};
};

class Dessert: public FoodCategory {
public:
    string getCategory() const {
        return "dessert";
    }
};

class Breakfast: public FoodCategory {
public:
    string getCategory() const {
        return "breakfast";
    }
};

class Lunch: public FoodCategory {
public:
    string getCategory() const {
        return "lunch";
    }
};
```

VS code and program Demo

Thank you