Programming Project / C++

Restaurant/Food Management System

Student 1: Kerekes Denis

Student 2: Miklo Ruben

I. Task Description

Denis is responsible for:

- Managing restaurant profiles (add/remove)
- Managing food items in restaurant menus (add/remove)
- Implementing the modifier command-line interface

Ruben is responsible for:

- Viewing all the restaurants and their menus
- Persistent data storage in foodapp_data.txt
- Modifying restaurant and food details

II. Data Classes

The following classes are implemented using structs (public members):

FoodItem Class:

```
struct FoodItem {
   std::string name;
   double price;
};
```

Restaurant Class:

```
struct Restaurant {

   std::string name;

   std::vector<FoodItem> menu;

};
```

Main Application Class:

```
class FoodDeliveryApp {

private:

   std::map<std::string, Restaurant> restaurants;

   const std::string dataFile = "foodapp_data.txt";

   // Helper functions for validation, saving, loading

public:

   // Management functions

};
```

III. File Structure

Data Storage File (foodapp_data.txt)

Format: RESTAURANT:Restaurant Name

ITEM:Food Item 1,12.99

ITEM:Food Item 2,8.50

ENDRESTAURANT

Executable Files

- Built executable: test.exe (from CMake configuration)
- Main Application: FoodApp2.exe

- Modifier CLI: ModifierCLI.exe
- Command reference: Commands.txt

IV. Command Interface

A. FoodApp2.exe Commands:

```
./FoodApp2.exe addrestaurant <name>
```

⇔ Add new restaurant

```
./FoodApp2.exe removerestaurant <name>
```

⇔ Delete restaurant

```
./FoodApp2.exe addfood <restaurant> <food> <price>
```

⇔ Add food to restaurant

```
./FoodApp2.exe removefood <restaurant> <food>
```

⇔ Remove food from restaurant

```
./FoodApp2.exe list
```

⇔ Show restaurants and menus

```
./FoodApp2.exe exit
```

⇔ Exit application

B. ModifierCLI.exe Commands:

```
./ModifierCLI.exe modify_restaurant <old_name> <new_name>
```

⇔ Rename a restaurant

```
./ModifierCLI.exe modify_food_name <restaurant> <old_food> <new_food>
```

⇔ Rename a food item

```
./ModifierCLI.exe modify_food_price <restaurant> <food> <new_price>
```

⇔ Change food price

```
./ModifierCLI.exe list
```

⇔ Show restaurants and menus

./ModifierCLI.exe exit

⇔ Exit modifier

V. Validation Rules

1. Restaurant/Food Names:

    o Must be non-empty

    o Only allow: alphanumeric, spaces, hyphens, apostrophes bool isValidName(const string& name);

2. Price Validation:

    o Must be positive numbers

    o No negative values allowed

bool isValidPrice(const string& priceStr);