Fast Food Fantasy

Your team is to write a program that reads data and stores it in a data structure of your choice. The data structure must have the ability to hold information for at least 20 fast food restaurants (the initial data contains fewer than that) where each fast food restaurant may have a maximum of 8 different menu items. This data structure will be used to allow foodies at Saddleback College to plan various day trips. You must be able to randomly access the various fast food restaurants. You will need to provide the ability to modify information about a fast food restaurant (such as adding menu items to the existing list), print a list of the fast food restaurants, plan a trip, and take a trip.

Displaying the initial information:

1. Display the list of fast food restaurants and their distances from Saddleback (the initial 10 restaurants).

2. Display all the menu items for any given fast food restaurants.

Planning trips:

3. Provide the capability to visit the initial 10 fast food restaurants starting at Saddleback.
   a. Plan the trip starting at Saddleback and then visit the closest fast food restaurant.
   b. Then visit each of the other fast food restaurants in the most efficient order (recursively choosing the next fast food restaurant closest to the previous fast food restaurant).
   c. Display the total distance traveled

4. Offer the option to plan the **shortest** trip starting at Domino’s Pizza
   a. Obtain the <u>number</u> of fast food restaurants to visit
   b. Visit the number of fast food restaurants specified (including Domino’s Pizza)
   c. Plan the trip starting at Domino’s Pizza then visit each of the other fast food restaurants in the most efficient order

(recursively choosing the next fast food restaurant closest to the previous fast food restaurant).

d. Display the total distance traveled
e. Allow the foodie to purchase multiple menu items when visiting the fast food restaurants

5. Offer the option to plan a custom trip
    a. Allow the foodie to select the starting fast food restaurant they wish to visit
    b. Then allow the foodie to select all fast food restaurants they wish to visit
    c. Plan the trip starting with the selected fast food restaurant then visit each of the other fast food restaurants in the most efficient order (recursively choosing the next fast food restaurant closest to the previous fast food restaurant).
    d. Display the total distance traveled
    e. Allow the foodie to purchase multiple menu items when visiting the fast food restaurants

6. When taking any trip:
    a. A user can purchase multiple menu items
    b. Your team must keep track of the number of menu items purchased and the total spent at each fast food restaurant (plus a grand total)
    c. Display the total revenue for each fast food restaurant including a grand total for all the fast food restaurants

7. Maintenance (administrator only - requires a password to gain access)
    a. Provide the capability to add fast food restaurants and their corresponding menu items by having your program read from an input file given to the class
    b. Provide the capability to delete a fast food restaurant and their corresponding menu items
    c. Provide the capability to change the prices of the menu items
    d. Provide the capability to add new menu items

    e. Provide the capability to delete menu items

8. Provide the capability to visit the12 fast food restaurants starting at Saddleback.
    a. Plan the trip starting at Saddleback and then visit the closest fast food restaurant.
    b. Then visit each of the other fast food restaurants in the most efficient order (recursively choosing the next fast food restaurant closest to the previous fast food restaurant).
    c. Display the total distance traveled

Please let me know your partners by January 29th (three points will be deducted from your score if you do not meet this deadline). All projects are due by March 12th. **No late projects will be accepted.** Your team must demonstrate your project to me before it will be graded. Each teammate must identify their accomplishments on the project. Not all team members will necessarily earn the same score.

1. Design a very readable, easy to use interface to demonstrate your program.
2. Contingency handling should include addressing invalid input.
3. Write at least 10 agile stories (including description, tasks, test scenarios, and story points) before any software is developed. The team must follow the Scrum process (the Scrum master **must** document all meetings and the product owner must document the backlog).
4. Submit your coding standards and team rules
5. Submit the following UML class diagrams
    a. Three use cases
    b. One activity diagram
    c. Three state diagrams with your project.
6. Submit a test plan.
7. All changes must be persistent between executions.
8. Submit a discussing the **Big-Oh** of your project for at least **three** methods.
9. Identify all the data structures used.
10. Each team must use a version control system (only team members should have access to their repository), graphical

user interface tool, automated documentation tool, and an Agile management tool. (GITHUB, DOXYGEN, WAFFIO.IO, graphical user interface using software such as QT, etc.).

Schedule:
First checkpoint – February 12th – 5 points
Second checkpoint – February 26th – 5 points
Final checkpoint – March 12th or – March 14th - 30 points

The project will be graded using the following scale:

Description	Value
First checkpoint	5
Second checkpoint	5
Meet requirements	8
Coding Style	3
Use of C++ data structures	4
User interface	3
Contingency handling	3
Test Plan	3
Adherence to Scrum	3
Identification of Big Oh	3
Total	40

Final demonstration meeting:

1. Submit all your project artifacts (code, test plan, agile stories, UML diagrams, traceability between agile stories and requirements, identify data structures used, identify accomplishments, etc.)
2. Be prepared to demonstrate all project's requirements within the 20 minute timeframe.
3. All team members must be present.

4. Each individual team member must clearly identify their accomplishments.