Coding Standards:

- Variable declaration/definitions should follow the declaration comments.
- No UPPERCASE or non-camelcased variable names.
- No global variables that can cause memory leaks.
- Good programming standards and variable/function names with readability.

Team Rules:

- Using QT as our main IDE, C++ as the language.
- Code should work on a Mac and a PC
- Team meeting 3 times a week
- Standup Meetings (approx 10 mins during class times)
- Doxygen for automatic documentation.
- Using private Discord server to communicate
- Don't struggle with a problem for more than a day
- 24-hour meeting cancellation notice with reason
- GitHub used for the agile management tool
- Keep calendar of important group member dates on scrum log
- Prioritize main requirements over minor details
- When someone completes a story, the scrum master and project owner should REVIEW AND APPROVE IT BEFORE MERGING to the master branch.

Data Structures Used:

- We used multiple vectors to store the different cities, foods, and what not throughout the program.

Agile Stories:

Story 1:
As a prospecting traveler, I would like a main window that allows me to navigate the program.

Description:
The prospecting traveler will have access to the main window UI which will be the baseline for every other function. This window will be the backbone for all the other features.

Assumptions:
- [x] There are no assumptions

Tasks:
- [x] Create a main window UI for the program to run.

Estimation:
1

Definition of Done:
The main window of the program will be completed and a GUI will be created for a traveler to utilize the program.

Story 2:
As a prospecting traveler, I would want the option to display the traditional food items listed from each city that I choose.

Description:
A button/drop down to display the traditional food items from each city will be created. Upon clicking on the drop down and selecting a city, the food for that specific city will then be shown.

Assumptions:
- [x] Main window will be created.
- [x] The database would have been created.

Tasks:
- [x] Create a drop down menu in a new window.
- [x] Create a list of foods within a data structure.
- [x] Display the foods for one city at a time.

Estimation:
2

Definition of Done:

A prospecting traveler will be able to utilize a drop down menu in order to display a food list from an individual city.

Story 3:
As an administrator, I would like a database created storing the European cities and the food they have to offer.

Description:
The database will be created holding the different city names and other necessary data.

Assumptions:
- [x] There are no assumptions.

Tasks:
- [x] Create a database containing the city information and the food information.

Estimation:
1

Definition of Done:
The administrator can see the database that was created for the cities and food.

Story 4:
As a prospecting traveler, I would like the option to display the European Cities I can travel to for my travel plan, as well as look at their distances from Berlin once displayed in order to plan accordingly.

Description:
The prospecting traveler will be greeted with a button in order to display the list of cities, alongside the city names, a distance measurement will be present alongside the list of cities.

Assumptions:
- [x] Main window will be created.
- [x] The database would have been created.

Tasks:
- [x] Add a button to call the method to display those cities.

Estimation:
2

Definition of Done:
The prospecting traveler will be able to click on a button to display cities and the distance from Berlin for each city in order to plan out their trip.

Story 5:
As an administrator I'd like to have a secure way to create and update data during runtime.

Description:
The administrator should have a secure way to add and remove European cities; change the prices of traditional food items; create new traditional food items; and delete traditional food items all during runtime of the program.

Assumptions:
- [x] A secure location to read in information.

Tasks:
- [x] Read in data from a secure place.
- [x] Update locations available.
- [x] Creating new traditional food items.
- [x] Deleting old traditional food items.
- [x] Editing pre-existing traditional food items.

Tests:
- [x] Look at the database to confirm the changed information.

Estimation:
5

Definition of Done:
Administrator is able to update data from a secure file and normal users cannot edit it during runtime.

Story 6:
As an administrator, I'd like to have a login so I can access the program as an administrator.

Description:
The administrator should be presented with a login page and login via a username and password to access the program.

Assumptions:
- [x] A login window is created.

Tasks:
- [x] Create a space to type in username and password.

Tests:
- [x] Make sure you can login as an administrator.

Estimation:
1

Definition of Done:
Administrator is able to login to the program as an administrator.

Story 7:
As a traveler, I'd like to have a login so I can access the program as an traveler.

Description:
The traveler should be presented with a login page and login via a username and password to access the program.

Assumptions:
- [x] A login window is created.

Tasks:
- [x] Create a space to type in username and password.

Tests:
- [x] Make sure you can login as an traveler.

Estimation:
1

Definition of Done:
Traveler is able to login to the program as an traveler.

Story 8:
As an administrator, I'd like to be able to upload an input file in order to append new data to the database.

Description:
The administrator will have the functionality to select a text file and upload the data from that file to the database.

Assumptions:
- [x] On prompt, the admin will be able to select a text file
- [x] When selected, the program will automatically read in the data from the file and write it to the database.
- [x] The program will automatically refresh, showing the newly appended data
- [x] Program is contiguous on close

Tasks:
- [x] Select a text file
- [x] Read in data from .txt
- [x] Write data from .txt to .db
- [x] Refresh tables, showing new data

Tests:
- [x] Look at the database to confirm the changed information

Estimation:
3

Definition of Done:
Administrator is able to update data from a secure file and normal users cannot edit it

Story 9:
As a frequent traveler, I want to be able to contact the developers of the website/software and suggest features, or report a bug I encountered.

Description:
The traveler will be able to visit a "contact us" page and send a message including their email, a topic and the message they wish to send.

Assumptions:
- [x] The main window is complete.
- [x] There is a "Contact Us" button to link to this page.

Tasks:
- [x] Create Contact Us page
- [x] Include fields for email, topic and message
- [x] Clear page upon submission in case they wish to send another message

Estimation:
1

Definition of Done:
Traveler can open the contact us page through a UI or button somewhere on the main page, put the subject and their issue in a text box, and submit the contact form to the developers.

Story 10:
As a prospecting traveler, I want the option to be able to visit the initial 11 European cities. I want to be able to plan my trip starting from Paris and have it plan out my route in the most efficient order.

Description:
The traveler will have the option to visit the 11 initial European cities. The traveler will have it displayed in the most efficient order, from Paris, when planning a trip. This will be done with recursion and will choose the city that is closest to the previous city before it.

Assumptions:
- [x] Main window UI created.

Tasks:
- [x] Create a list/database of the 11 Cities including Paris
- [x] Develop a recursive function to find the shortest distance between cities
- [x] Create a planning window for a trip originating form Paris.

Estimation:
8

Definition of Done:
The prospecting traveler will be able to plan a trip starting from Paris to the 11 European Cities and have it in the most efficient order

Story 11:
As a prospecting traveler, I would like to plan the shortest trip starting from London.

Description:
Allow the traveler to input how many cities they want to visit starting from London.

Assumptions:
- [x] Main window UI created.
- [x] City database created.

Tasks:
- [x] Obtain the "number" of European cities to visit
- [x] Visit the "number" of cities specified (including London)
- [x] Plan the trip starting at London then visit the closest city to Paris, then visit the closest city to that city and so on (recursively choose the city closest to the previous city)

Tests
- [x] Desk check the number of European cities that are to be visited.
- [x] Desk check to make sure the recursive function has chosen the most efficient path.

Estimation:
8

Definition of Done:

The traveler is able to input how many cities they want to visit.

Story 12:
As a traveler with not a lot of time I want to be able to plan a quick and efficient route.

Description:
Starting from any city there should be a function to calculate the most efficient path to visit the cities that the traveler wants to visit.

Assumptions:
- [x] Database is already created and accessible
- [x] Information on locations and starting city is given by traveler

Tasks:
- [x] Create a recursive function that plans the most optimal route using distances between each location obtained from the database

Tests:
- [x] Desk check that the recursive function did indeed make the most optimal path.

Estimation:
8

Definition of Done:
The user should be able to put in their locations and the starting location and the route will be made in the most efficient manner.

Story 13:
As a traveler who enjoys foreign foods, I want to be able to be able to purchase multiple traditional food items, so I can order dinner for my whole family.

Description:
The user will be able to purchase items at each city they travel to, see how much they've spent at each city and see the total they spent across the trip.

Assumptions:
- [x] A database for the food items has been created.
- [x] A method of tracking purchases is available.
- [x] A database of all cities has been created.

Tasks:
- [x] Create a way for user to purchase food items at each city
- [x] Track purchases at each individual city, and total it.
- [x] Track purchases and total across entire trip.

Estimation:
5

Definition of Done:
Traveler can see menus from restaurants, put in an order of multiple items of food via an order UI, track their spending for both that individual city and the entire trip through a statistics page fed from a database.

Story 14:
As a prospecting traveler, I would like to plan a custom trip.

Description:
Allow the traveler to input the starting city and all other cities they want to visit.

Assumptions:
- [x] Main window UI created.
- [x] City database created.

Tasks:
- [x] Allow a traveler to select the starting European city they wish to visit
- [x] Then allow a traveler to select all other European cities
they wish to visit
- [x] Plan the trip starting with the selected city then visit each of the other European cities in the most efficient order (recursively choose the city closest to the previous city).

Tests
- [x] Desk check to make sure the recursive function has chosen the most efficient path.

Estimation:
8

Definition of Done:
The traveler is able to plan a custom trip that is efficiently planned.

Story 15:
As a prospecting traveler, I would like to see the total distance traveled of my trip.

Description:
Have an option for the traveler to request the total distance traveled of their trip.

Assumptions:
- [x] Main UI is created.
- [x] City database is created.

Tasks:
- [x] Display the total distance traveled of a trip.

Tests
- [x] Desk check the total distance using google maps.

Estimation:
1

Definition of Done:
The traveler is able to see the distance traveled of their trip.