User Stories

Team C - Memory Leak

Nathan Tran, Lina Kang, Michael Duenas, Jeff Purdy, Nathan Kim

Team Name: Memory Leak
Team Members: Nathan Kim, Nathan Tran, Michael Duenas, Lina Kang, Jeff Purdy

Product Owner: Nathan Tran
Scrum Master: Nathan Kim

Team Rules:

1. Code should be thoroughly tested before presenting to scrum master and product owner.
2. When someone finishes a story the product owner must verify that it meets the standard of the user story.
3. Be on time to team meetings to ensure everybody is on the same page.
4. Collaborate and ask for help when needed.
5. Suggestions and improvements to the software should be discussed during team meetings where every group member could have a say.
6. Conflicts within the team will be discussed as a team.
7. If an assigned story is not finished on time, priority will be set high till complete and another team member will be assigned to help you.
8. If an assigned story is finished early, you will be reassigned to another active story (will be assisting another member of the team).

Coding Standards:

1. For every block of code (if...else.., for(), while(), etc.), put a block comment above that generally describes what that block of code is supposed to do.
2. Every function should have a general description of what it achieves along with specifying the outcome after it runs.
3. Variable names should be specific and should be significant to the purpose that they serve.
4. Constant variables should be all caps and should also be significant to the purpose that they serve.
5. Global variables should not be used to avoid confusion during debugging.

Definition of Done for Sprint 1

1. There is a data structure that contains cities, their corresponding food items, and their prices.
2. There is an interface in place that allows ease of usage
3. Admin will be able to manipulate the data in the data structures

Which stories will be assigned to Sprint 1

- Story 1, Story 2, Story 10
- Design stories: UML class diagrams, use cases, activity diagrams.

Which stories will be assigned to Sprint 2

- Story 3, Story 4, Story 6, Story 7

Which stories will be assigned to Final Sprint

- Story 5, Story 8, Story 9
- Doxygen

User Story #1:

A. Statement:
   a. As an administrator, I would like to create a data structure that can store at least 30 cities, along with their respective local food items.
B. Description:
   a. The data structure will be able to contain 30 objects or more. The rest of the project will be able to access this data structure.
C. Tasks:
   a. Declare the data structure (QVector)
   b. Set up an input function.
   c. Use this input function to store the data into the data structure
D. Tests:
   a. Verify the input function works by output statements which verify whether the data got transferred properly.
   b. Verify that the data structure (QVector) stored the data properly by outputting the data elements within the QVector.
E. Assignee:
   a. Nathan Kim

F. Story Point: Baseline
G. Priority: High
H. Done:
   a. 30 (or more) cities and their corresponding foods will be displayed on the GUI
   b. Segment of code will be shown to Product Owner and Scrum Master
   c. All tests are passed

d. Coding Standards are met
e. Merged with the main branch on GitHub

User Story #2:

A. Statement:
   a. As a developer, I want to create a class (City) that stores the data for each city.
B. Description:
   a. The objects within the QList data structure will be instances of this class.
C. Tasks:
   a. Create a class (City) with the necessary data members (location, local food items).
   b. Write member functions (getters, setters)
D. Tests:
   a. Verify that the class functions are working.
E. Assignee:
   a. Lina Kang
F. Story Point: 1 point
G. Priority: Medium
H. Done:
   a. A city will have its corresponding data (local foods, location) displayed on the GUI
   b. Segment of code will be shown to Scrum Master
   c. All tests are passed
   d. Coding Standards are met
   e. Merged with the main branch on GitHub

User Story #3 //no trip only display

A. Statement:
   a. As a prospective traveler, I want to be able to see all the cities' distances from Berlin.
B. Description:
   a. The traveler will be able to see the cities they wish to travel to (from Berlin) in a list.
B. Tasks:
   a. Display a list of cities to select from
   b. Store the list of cities selected by user
   c. Output the cities.
C. Tests:
   a. Output the list of selects cities to ensure they are saved

D. Assignee:
   a. Nathan Kim

E. Story Point: 3 points
F. Priority: Medium
G. Done:
    a. The cities will be displayed to the GUI
    b. Code adhered to coding standard
    c. Product Owner verified
    d. All tests passed
    e. Merged on main branch on Github

User Story #4:

I. Statement: As a Travel planner I would like to offer an option to plan the shortest trip starting at Paris.
J. Description:
    a. The traveler will be able to select any amount of cities they would like to visit and be given the total distance traveled as well was the list of cities to be traveled to in order
    b. Traveler will be able to purchase traditional foods at any given city when visited
    c. Recursively calculate shortest distance from one city to another
K. Tasks:
    a. Display a list of cities for the traveler to select any
    b. Plan an efficient trip based off those cities and display the order and distance traveled
L. Tests:
    a. Output the distance traveled and the list of cities being traveled to in the order of efficiency
    b. Ensure the list of cities selected by traveler are maintained via output
    c. Manually check distances and verify with output
M. Assignee:
    a. Michael Duenas
N. Story Point: 3 points
O. Priority: Medium
P. Done:
    a. Variety of plan options are shown on the UI sorted by distances from shortest to longest
    b. Code adhered to coding standard
    c. Product Owner verified
    d. All tests passed
    e. Merged to the main branch on Github

User Story #5

A. Statement
    a. As a prospective traveler, I want to be able to keep track of the food items I have purchased along with a receipt of the money I have spent.

B. Description:
    a. The user can select the desired food items.
    b. The user can see a receipt for what they have purchased.

C. Tasks:
    a. The food items must be stored as well as their prices
    b. The price will be calculated
    c. The receipt will be displayed

D. Tests:
    a. Verify that the selected food items are stored.
    b. Verify that the price calculated is correct.
    c. Verify that the receipt is appropriately displayed in a clean format.

E. Assignee: Lina Kang

F. Story Point: 2 points

G. Priority: Medium

H. Done:
    a. Purchased food items and their corresponding records will be shown on the GUI
    b. Code adhered to coding standard
    c. Product Owner and Scrum master verified
    d. All tests passed
    e. Merged to the main branch on Github

User Story #6

A. Statement:
    a. As an administrator, I want to be able to enter a password to access any administrator function to ensure that the application is secure.

B. Description:
    a. There will be a login that is password protected that will enable access to administrator functions.

C. Backlog:
    a. Select a log-in from the file dropdown that brings up a log-in screen that allows the administrator to log-in if necessary.
    b. Create a database for approved administrator usernames and passwords.
    c. Once the admin has successfully logged in, display the administrator functions on the UI.

D. Tests:
    a. Verify that the log-in button brings up a log-in screen.
    b. Verify only username and password from the database allows for a successful log-in.
    c. Verify that once the admin has successfully logged in, display the administrator functions on the UI.

E. Assignee:
    a. Jeff, develop the UI features as well as database for usernames and passwords.

F. Story Point: 3
G. Priority: Low
H. Done:
    a. User will be able to see a separate login screen that will allow for administrator access using a username and a password
    b. The code is well commented.
    c. All tests are passed.
    d. Coding Standards are met and confirmed by the scrum master and product owner
    e. Merged with the main branch on GitHub

User Story #7

C. Statement:
    a. As a prospective traveler, I want to be able to plan a trip starting at Berlin to any of 13 european cities and have it be planned in the most efficient way.
D. Description:
    b. The traveler will be able to select any cities they wish to travel to from a list and have the trip be planned out for them in the most efficient way to travel the least distance.

B. Tasks:
    d. Display a list of cities to select from
    e. Store the list of cities selected by user
    f. Calculate the combined distance
    g. Output the order in which cities will be traveled to, as well as the calculated distance.

C. Tests:
    b. Output the list of selects cities to ensure they are saved
    c. Output the sorted list based on travel efficiency
    d. Output calculated distance

D. Assignee:
    a. Michael Duenas

E. Story Point: 3 points
F. Priority: Low
G. Done:
    a. From Berlin, the total distance traveled, this distance being calculated with respect to Berlin, will be displayed to the GUI
    b. Code adhered to coding standard
    c. Product Owner verified
    d. All tests passed
    e. Merged on Github

User Story #8

A. Statement:
    a. As a prospective traveler, I want to be able to customize the starting city of my itinerary.

B. Description:
    a. The user can select the starting city of their trip.
C. Tasks:
    a. Implement a simple interface to allow for easy selection of starting city and final city.
    b. Develop back end functions to rearrange the itinerary after starting city and final city have been changed.
D. Tests:
    a. Select a starting city and then output the new itinerary.
    b. Select a final city and then output the new itinerary.
E. Assignee:
    a. Nathan Kim

F. Story Point: 2 points
G. Priority: Low
H. Done:
    a. Verify that a user can change the starting and ending cities for their trip (changes will be displayed to the screen)
    b. The code is well commented.
    c. All tests are passed.
    d. Coding Standards are met and confirmed by the scrum master and product owner
    e. Merged with the main branch on GitHub

User Story #9

A. Statement:
    a. As a prospective traveler, I want to have a clean and easy to use interface, so that I can navigate the application and use all it's features.
B. Description:
    a. A traveler can clearly navigate the interface.
    b. The traveler will be able to easily create a custom travel plan and purchase foods easily.
C. Tasks:
    a. As a prospective traveler, I will be able to easily navigate the application.
    b. As a prospective traveler, it will be easy to create a custom travel plan and purchase foods from European cities.
D. Tests:
    a. Verify that the application can be navigated easily without confusion.
    b. Verify that the custom plan interface is simple and easy to use.
E. Assignee:
    a. Nathan Tran - Implement a basic menu that allows the user to navigate the many features of the app.
F. Story Point: 5 points
G. Priority: Medium
H. Done:

a. The GUI will have a menu of buttons to navigate between the list of cities and their foods.
b. The interface will allow the traveler to navigate to their custom plan and edit it.
c. The code is well commented
d. The user interface loads up and the traveler is able to navigate through all the features of the application without the application crashing.
e. All features display their corresponding menus correctly.

User Story #10

A. Statement:
   a. As an administrator, I want to be able to access and maintain the data so that I can add new European cities and their corresponding food items.
B. Description:
   a. The administrator can access the data structure and add new European cities and their respective foods.
   b.
C. Tasks:
   a. As an administrator, I can access the data structure and add new European cities.
   b. As an administrator, I can add/edit the foods in each European city and their corresponding prices.
D. Tests:
   a. Verify that European cities are actually added in the data structure.
   b. Verify that Each European city can have their list of foods edited.
E. Assignee:
   a. Jeff Purdy - Implement a feature that allows the administrator to easily add another European city and their corresponding food item in the database.
F. Story Point: 3 points
G. Priority: Low
H. Done:
   a. Changes to cities, or their child data (food, locations, etc.) can be visually seen on the GUI
   b. Code is verified by the scrum master and product owner.
   c. The code is well documented.
   d. The administrator will be allowed to edit the data structure after logging in with a password
   e. The administrator is able to add in European cities and their corresponding foods into the data structure.
   f. Newly added European cities are saved and can be accessed by the traveler when they create a new custom travel plan.