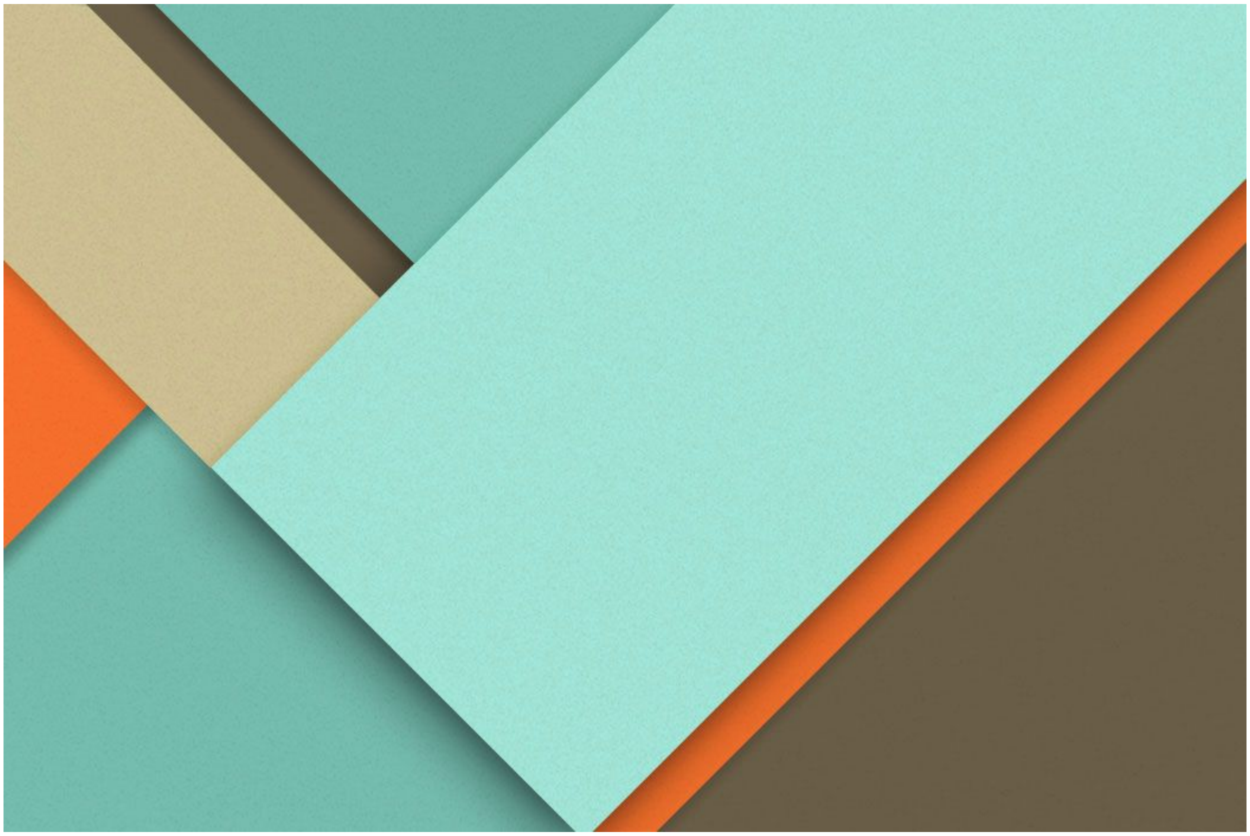

SQUAD

Team Members:

William Scott

Jake Sichley

Jose Cardenas

Erick Vickers

Thea Knasiak

Story One.

As a <traveler>, I want < a display of the European cities & their distances from Berlin on my screen > so that < I can see which cities are in my spectrum>.

A. Assumption:
   1. Database is Connected.
   2. Database holds the European Cities and their corresponding distances from Berlin.

B. Assignee: William

C. Story Point Estimation: 4

D. Priority: 1

E. List of Task and Tests:
   1. Task:
      a) Read from the database and Display the European Cities in a bordered container.
      b) Read from the database and Display the corresponding distance from Berlin next to the bordered container
   2. Test:
      a) Check that the correct European Cities are displayed.
      b) Check that the correct corresponding distances are displayed next to the European Cities.

F. Definition of Done:
   1. Acceptance criteria is verified during Testing.
   2. Unit testing - passed and completed.
   3. Story is accepted by the Product Owner
   4. No memory Leaks.

Story Three.

As a <traveler>, I want < a display of the food items for any given city> so that < I can see what they are selling >.

A. Assumption:
   1. Database is Connected.
   2. Database holds the European traditional food items.
   3. The number of traditional food items is known.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
   1. Task:
      a) Read in from the database.
      b) Create a label generator that populates a box with the traditional food items.
      c) Space the traditional food items evenly.

d) Display the name of the traditional food items and its price.

2. Test:
    a) Check that the label generator populates the correct number of traditional food items.
    b) Check that the label generator populates the correct traditional items for a given European City
    c) Check that no label overlaps one-another.

F. Definition of Done:
1. Acceptance criteria is verified during Testing.
2. Unit testing - passed and completed.
3. Story is accepted by the Product Owner
4. No memory Leaks.

Story Three.

As a <traveler>, I want < to visit the initial 11 European Cities starting at Paris> so that <I can see the most efficient trip starting at Paris >.

A. Assumptions:
1. Database is Connected.
2. The initial 11 European Cities are in the database
3. The corresponding distances of each European city is inside the Database.
4. The Starting location is Paris.
5. The recursive functionality that takes the next closest city is completed.
6. The Label generator functionality that populates the traditional food items for a given city is completed.
7. The functionality to make a purchase is completed.
8. The functionality that takes account the number of purchases made is completed.
9. The functionality that takes into account the distance traveled is completed.

B. Assignee: Erich

C. Story Point Estimation: 10

D. Priority: 2

E. List of Task and Tests:
1. Task:
    a) Read in the European Cities from the Database.
    b) Create a button for the “Initial 11 European Cities Trip”.
    c) Implement the Recursive function.
    d) Load the trip destinations.
    e) Show the next stop(city) on the trip.
    f) Show the distance traveled.
    g) Show the traditional food items at each city.

            h) Show the total purchases made the traveler at that location of the visit.
    2. Test:
        a) Check the that correct European Cities were read in.
        b) Check that the trip destination is in the most efficient order.
        c) Check that the next stop shows the correct city from the trip destination
        d) Check that the purchases made are accounted for in the database.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner
    4. No memory Leaks.

Story Four.

As a <traveler>, I want < the next closest city to be my next stop and so on > so that < my trip is very efficient >.

A. Assumptions:
    1. The Database is connected.
    2. The Starting location is set.
    3. The European Cities on the trip are allocated.
    4. The corresponding distance to each cities is in the database.
    5. The functionality that tracks the distance is completed.

B. Assignee: Thea

C. Story Point Estimation: 10

D. Priority: 2

E. List of Task and Tests:
    1. Task:
        a) Read in the specified European Cities on the trip
        b) Select the starting point (explain)
        c) Find the next closest city, this will become the next starting point and the previous starting point will be excluded from the list.
    2. Test:
        a) Check that the correct cities are chosen.
        b) Check the starting point has been confirmed (boolean).
        c) Check that the next closest city to the starting location is correct, and so on.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner

4. No memory Leaks.

Story Five.

As a <traveler>, I want < an option to plan the shortest trip starting at London> so that < I can select the number of cities to visit and experience the shortest trip >.

A. Assumptions:
   1. Database is Connected.
   2. All the European cities and their corresponding distances to each other city are in the database.
   3. The Starting location is London.
   4. The recursive functionality that takes the next closest city is completed.
   5. The Label generator functionality that populates the traditional food items for a given city is completed.
   6. The functionality to make a purchase is completed.
   7. The functionality that takes account the number of purchases made is completed.
   8. The functionality that takes into account the distance traveled is completed.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
   1. Task:
      a) Make a button for the "The shortest Trip"
      b) Make the user specify the amount of cities they wish to visit on their trip.
      c) Read in the specified European Cities on the trip including Paris.
      d) Implement the Recursive function.
      e) Load the trip destinations.
      f) Show the next stop(city) on the trip.
      g) Show the distance traveled.
      h) Show the traditional food items at each city.
      i) Show the total purchases made the traveler at that location of the visit.
   2. Test:
      a) Check the that correct number of European Cities were read in.
      b) Check that the trip destination is in the most efficient order.
      c) Check that the next stop shows the correct city from the trip destination
      d) Check that the purchases made are accounted for in the database.

F. Definition of Done:
   1. Acceptance criteria is verified during Testing.
   2. Unit testing - passed and completed.
   3. Story is accepted by the Product Owner
   4. No memory Leaks.

Story Six.

As a <traveler>, I want < an option to plan a custom trip> so that < I can decide the starting point, and the cities I wish to visit >.

A. Assumptions:
   1. Database is Connected.
   2. All the European cities and their corresponding distances to each other city are in the database.
   3. The recursive functionality that takes the next closest city is completed.
   4. The Label generator functionality that populates the traditional food items for a given city is completed.
   5. The functionality to make a purchase is completed.
   6. The functionality that takes account the number of purchases made is completed.
   7. The functionality that takes into account the distance traveled is completed.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
   1. Task:
      a) Make a button for the “Custom Trip”
      b) Input: let the user specify the starting European City.
      c) Input: let the traveler specify the European Cities on the trip.
      d) Read in the specified European Cities on the trip.
      e) Implement the Recursive function.
      f) Load the trip destinations.
      g) Show the next stop(city) on the trip.
      h) Show the distance traveled.
      i) Show the traditional food items at each city.
      j) Show the total purchases made the traveler at that location of the visit.
   2. Test:
      a) Check that the correct number of European Cities were read in.
      b) Check that the correct European Cities were read in.
      c) Check that the correct starting location is the one specified by the traveler.

d) Check that the trip destination is in the most efficient order.
e) Check that the next stop shows the correct city from the trip destination
f) Check that the purchases made are accounted for in the database.

F. Definition of Done:
1. Acceptance criteria is verified during Testing.
2. Unit testing - passed and completed.
3. Story is accepted by the Product Owner
4. No memory Leaks.

Story Seven.

As a <traveler>, I want < the amount of purchased items to be tracked at each city > so that < I can see how many items were bought at each location >.

A. Assumptions:
1. Database is Connected.
2. The database is setup to handle the amount of purchases for each city and their given tradition food item.
3. The Label generator functionality that populates the traditional food items for a given city is completed.
4. The functionality to make a purchase is completed.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
1. Task:
a) Increment the purchase count for each European traditional food item.
b) Create a total purchase count for each European city based on their traditional food items purchase count.
2. Test:
a) Check that the correct traditional food item is incremented after a purchase is made.
b) Check that the total purchase count is accurate with the total purchases made for each traditional food items for a given city.

F. Definition of Done:
1. Acceptance criteria is verified during Testing.
2. Unit testing - passed and completed.
3. Story is accepted by the Product Owner
4. No memory Leaks.

Story Eight.

As a <traveler>, I want < the total distance traveled to be displayed > so that < I can see how far we have traveled> .

A. Assumptions:
   1. Database is Connected.
   2. All the European cities and their corresponding distances to each other city are in the database.
   3. The recursive functionality that takes the next closest city is completed.
B. Assignee: Jose
C. Story Point Estimation: 2
D. Priority: 3
E. List of Task and Tests:
   1. Task:
      a) The recursive functionality should have created the most efficient order in which to travel.
      b) Set a total distance variable, increment it with the distances to the next city from the recursive functionality list.
   2. Test:
      a) Check that the total distance covered is accurate by adding the distances from the recursive functionality list.
F. Definition of Done:
   1. Acceptance criteria is verified during Testing.
   2. Unit testing - passed and completed.
   3. Story is accepted by the Product Owner
   4. No memory Leaks.

Story Nine.

As a <traveler>, I want < the ability to purchase multiple items at each city > so that < I can try out the food at each city >.

A. Assumptions:
   1. Database is Connected.
   2. The database is setup to handle the amount of purchases for each city and their given tradition food item.
B. Assignee:
C. Story Point Estimation:
D. Priority:
E. List of Task and Tests:

1. Task:
    a) Create a label generator functionality that populates the traditional food items for a given city based on their amount of food items.
    b) Allow items to be added to a cart once clicked. if clicked more than once increase the quantity.
    c) Create a delete button next to it do delete the item off the cart.
    d) Once ready, create a purchase button to finalize the purchase of the items on the cart.
    e) When “purchase” is hit, all the quantities of the items in the purchase are recorded and sent to the database.
2. Test:
    a) Check that the label generator functionality populates the right amount of labels.
    b) Check that the quantity increases once hit more than once.
    c) Check that the delete button works in the right manner, decrementing the quantity by one.
    d) Check that the quantities of the items are being recorded, as well as accumulating the total amount spent for that visit.

F. Definition of Done:
  1. Acceptance criteria is verified during Testing.
  2. Unit testing - passed and completed.
  3. Story is accepted by the Product Owner
  4. No memory Leaks.

Story Ten.

As an <admin>, I want < the European Cities & their food items to be inputted from a text file > so that < It can be loaded into the database >.

A. Assumptions:
  1. Database is connected.
  2. The database is set and ready to take the input from the text file.

B. Assignee: Jake

C. Story Point Estimation: 1

D. Priority: 1

E. List of Task and Tests:
  1. Task:
      a) Specified which text file is being read in.
      b) Create an algorithm to take in the data from the text file.
      c) Read the data from the text file to the database.
  2. Test:
      a) Check that the correct text file is used.
      b) Check that the data inside the database is the correct data from the text file.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner
    4. No memory Leaks.

Story Eleven.

As an <admin>, I want < the ability to edit traditional food item prices > so that < prices can be changed from time to time >.

A. Assumptions:
    1. Database is connected.
    2. The database is set and ready for changes to be made.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
    1. Task:
        a) Create a UI for the user to edit and change data of a specified traditional food item.
    2. Test:
        a) Check that the traditional food item has also been changed inside the database.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner
    4. No memory Leaks.

Story Twelve.

As an <admin>, I want < the ability to Add/Delete traditional food items > so that < traditional food items can be added or discarded in later times >.

A. Assumptions:
    1. Database is connected.
    2. The database is set and ready for changes to be made.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
    1. Task:
        a) Create a UI for the user to add and delete an traditional food items from any given European city.

2. Test:
    a) Check that the changes made to the traditional food items are displayed in the database.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner
    4. No memory Leaks.

Story Thirteen.

As an <admin>, I want < the ability to login and logout of my account> so that < I can only make the changes to the database>.

A. Assumptions:
    1. Database is connected.
    2. The database holds the names and passwords for those able to edit that data.

B. Assignee:

C. Story Point Estimation:

D. Priority:

E. List of Task and Tests:
    1. Task:
        a) Create a login UI for the admin.
        b) Take in the username and the password.
        c) Bind the values to the database to check if it matches. If yes, give access to the data, else give an error.
    2. Test:
        a) Check that the username and password match a user.
        b) Check for unknown values to be imputed for the username and password that would cause a red flag in the program.

F. Definition of Done:
    1. Acceptance criteria is verified during Testing.
    2. Unit testing - passed and completed.
    3. Story is accepted by the Product Owner
    4. No memory Leaks.