# Test Description (Group 3 Milestone 3)

**Test Name or ID**: Test assignPackage (by Bilal)

**Test Type**: Black box

**Description**: Tests the assignPackage function to ensure that all packages are assigned to the correct trucks, as well as which route the truck will follow.

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_assignPackage

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests to see if a shipment is assinged to a correct truck which shares the same route on the package | PARAMETERS:  Truck:  [1000,50.0, "Blue"], [10, 20.00, "Red"], [10, 10.0, "Green"]  Shipment:  [400, 5, Point[H,7] | 0 | 0 | PASS |
| 1. Tests to see if a shipment that is valid is unable to ship from all trucks being full | PARAMETERS:  Truck:  {2500, 50.0, green, {}},  {2500, 50.0, blueRoute, {}},  {2500, 50.0, yellow, {}}  };  Shipment:  [200, 5, Point[K,2] | -1 | -1 | PASS |
| 1. Tests to see if a shipment that is invalid (not a home) is correctly dealt with | PARAMETERS:  Truck:  [5000, 250.00, "Blue"], [90, 250.00, "Red"], [10, 250.00, "Green"]  Shipment:  [-1, 5, Point[X,1] | -1 | -1 | PASS |
| 1. Tests to see if a shipment that must be diverted is correctly assigned to truck that is within range | PARAMETERS:  Truck:  [5000, 250.00, "Blue"], [0, 250.00, "Green"], [10, 250.00, “Yellow” Point[G,12]  Shipment:  [3000,1, 2, Point[V,3] | 1 | 1 | PASS |

**Bugs Found**:

Bug was found that checkSpace/Valid function would sometimes return false under a valid truck/package, checkspace and valid were adjusted.

**Test Name or ID**: Test valid (by Bilal)

**Test Type**: Black box

**Description**: Tests the valid function to ensure that all packages/shipment are handled correctly.

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_checkValid

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests to see if a valid shipment is accepted (size within range, weight and an existing truck id) | PARAMETERS:  Shipment:  [20, 2, "Blue"] | TRUE | TRUE | PASS |
| 1. Tests to see if a valid weight, odd size and non accepted size, and an extremly large, non-coresponding ID, is valid | PARAMETERS:  Shipment:  [4000, 7, "242103131221321332131"] | FALSE | FALSSE | PASS |
| 1. Tests to see if a valid size, and id is met, but not weight | PARAMETERS:  Shipment:  [-1, 5, "Red"] | FALSE | FALSE | PASS |
| 1. Tests to see if a valid size and id is met, but maxInt for weight | PARAMETERS:  Shipment:  [MAXINT+1, 2, "Green"] | FALSE | FALSE | PASS |

**Bugs Found**:

N/A (Not yet implemented/executed)

**Test Name or ID**: Test checkSpace (by Bilal)

**Test Type**: Black box

**Description**: Tests the checkSpace function to ensure that trucks handle storage space correctly depending on the packages/shipment.

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_checkSpace

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests to see if a valid shipment fit's on a truck with enough space | PARAMETERS:  Truck:  [5000, 250.00, "Blue"]  Shipment:  [20, 2, "Blue"] | TRUE | TRUE | PASS |
| 1. Tests to see if a wrong size and non id fits on the truck | PARAMETERS:  Truck:  [5000, 250.00, "Blue"]  Shipment:  [4000, 7, ""] | FALSE | FALSE | PASS |
| 1. Tests to see if a shipment with a invalid weight is put on the truck | PARAMETERS:  Truck:  [5000, 250.00, "Blue"]  Shipment:  [-1, 5, "Red"] | FALSE | FALSE | PASS |
| 1. Tests to see if a shipment fit's with non matching id's, and a weight that does not fit the truck | PARAMETERS:  Truck:  [10, 250.00, "Blue"]  Shipment:  [3000,1, 2, "Green"] | FALSE | FALSE | PASS |

**Bugs Found**:

N/A (Not yet implemented/executed)

**Test Name or ID**: Test divert (by Kaitlyn)

**Test Type**: Black box

**Description**: Tests the divert function to ensure that trucks divert to the nearest/most efficient route to the destination, returns the index of the truck diverted

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_checkDivert

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests to see if a valid shipment fits on a truck with sufficient space | PARAMETERS:  Truck:  [100, 80.00, "Green"]  Shipment:  [500, 2, "Green"] | 0 (Ship on GREEN LINE, no diversion) | 0 | PASS |
| 1. Tests if a shipment is correctly assigned to a truck that will need to divert from the original route | PARAMETERS:  Truck:  [5000, 250.00, "Yellow"]  Shipment:  [1500, 2, "12E"] | 0 (Ship on YELLOW LINE, divert: D10, E10, E11, E12) | 0 | PASS |
| 1. Tests if a shipment/package exceeding truck capacity is properly handled | PARAMETERS:  Truck:  [5000, 200.00, "Blue"]  Shipment:  [6500, 5, "Blue"] | -1 (Invalid weight (must be 1-5000 Kg.)) | -1 | PASS |
| 1. Tests if a shipment/package with an invalid destination is properly handled | PARAMETERS:  Truck:  [5000, 250.00, "Blue"]  Shipment:  [1000, 2, "28X"] | -1 (Invalid destination) | -1 | PASS |

**Bugs Found**:

N/A (Not yet implemented/executed)

**Test Name or ID**: Test eqPt (by Kaitlyn)

**Test Type**: Black box

**Description**: Tests the eqPt function to ensure that the program prints an accurate map.

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_eqPt

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests if the function will return true if both points have the same values | PARAMETERS:  p1 = (9, 13)  p2 = (9, 13) | 1 | 1 | PASS |
| 1. Tests if the function will return false if both points have different row values | PARAMETERS:  p1 = (2, 10)  p2 = (5, 10) | 0 | 0 | PASS |
| 1. Tests if the function will return false if both points have different column values | PARAMETERS:  p1 = (3, 11)  p2 = (3, 20) | 0 | 0 | PASS |
| 1. Tests if the function will return false if both points have different row and column values | PARAMETERS:  p1 = (8, 16)  p2 = (7, 14) | 0 | 0 | PASS |
| 1. Tests if the function will return false if the points have opposite edge values of the map | PARAMETERS:  p1 = (0, 0)  p2 = (24, 24) | 0 | 0 | PASS |

**Bugs Found**:

N/A (Not yet implemented/executed)

**Test Name or ID**: Test distance (by Kaitlyn)

**Test Type**: Black box

**Description**: Tests the distance function to ensure that the program correctly computes the distance between two points.

**Setup:** C++ unit testing project file in Visual Studio 2022

**Test Function**: test\_distance

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| 1. Tests if the function handles when both points are the same | PARAMETERS:  p1: (3, 4)  p2: (3, 4) | 0.0 | 0.0 | PASS |
| 1. Tests if the function computes the horizontal distance between two points | PARAMETERS:  p1: (1, 2)  p2: (1, 6) | 4.0 | 4.0 | PASS |
| 1. Tests if the function handles diagonal distance between two points | PARAMETERS:  p1: (1, 1)  p2: (4, 5) | 5.0 | 5.0 | PASS |
| 1. Tests if the function computes the vertical distance between two points | PARAMETERS:  p1: (1, 4)  p2: (1, 10) | 6.0 | 6.0 | PASS |

**Bugs Found**:

N/A (Not yet implemented/executed)