# Test Description

**Test Name or ID**: 01

**Test Type**: Black box

**Description**: Checks if the weight and volume of a package are valid.

**Setup:** Ensure the function is accessible and test inputs are ready.

**Test Function**:

isPackageValid(weight, volume)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Validate a valid package | Weight: 10.0 kg, Volume: 0.5 m³ | Return value: 1 (Valid package) | Return value: 1 (Valid package) | pass |
| Validate a package with negative weight | Weight: -5.0 kg, Volume: 1.0 m³ | Return value: 0 (Invalid package) | Return value: 0 (Invalid package) | pass |
| Validate a package with invalid volume | Weight: 50.0 kg, Volume: 3.0 m³ | Return value: 0 (Invalid package volume) | Return value: 0 (Invalid package volume) | pass |
| Validate an empty package | Weight: 0.0 kg, Volume: 0 m³ | Return value: 0 (Invalid) | Return value: 0 (Invalid) | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Test Name or ID**: 02

**Test Type**: Black box

**Description**: Verifies if a destination point on the map is valid (within bounds and not blocked).

**Setup:** Load the map and ensure the function is accessible.

**Test Function**:

isDestValid(map, destination)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Validate a valid destination point | Map: 25x25 grid, Destination Point: (5, 5) | Return value: 1 (Valid destination) | Return value: 1 (Valid destination) | pass |
| Validate a point out of map bounds | Map: 25x25 grid, Destination Point: (30, 30) | Return value: 0 (Invalid destination) | Return value: 0 (Invalid destination) | pass |
| Validate a point blocked by a building | Map: 25x25 grid, Destination Point: (10, 10) | |  | | --- | | Return value: 0 (Invalid destination) |  |  | | --- | |  | | Return value: 0 (Invalid destination) | pass |
| Validate a destination at the map boundary | Map: 25x25 grid, Destination Point: (24, 24) | Return value: 1 (Valid destination) | Return value: 1 (Valid destination) | Pass |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Test Name or ID**: 03

**Test Type**: Black box

**Description**: Ensures a package can be added to a truck without exceeding capacity.

**Setup:** Prepare an empty truck and valid/invalid package data.

**Test Function**:

addPackage(truck, package)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Add a valid package to an empty truck | Truck: Empty, Package: {Weight: 50 kg, Volume: 1 m³} | Truck contains package, updated weight and volume | Truck contains package, updated weight and volume | pass |
| Add a package exceeding weight capacity | Truck: {Weight: 1500 kg}, Package: {Weight: 10 kg} | Truck rejects package, weight exceeds capacity | Truck rejects package, weight exceeds capacity | pass |
| Add a package exceeding volume capacity | Truck: {Volume: 48 m³}, Package: {Volume: 2 m³} | |  |  |  | | --- | --- | --- | | |  | | --- | | Truck rejects package, volume exceeds capacity |  |  | | --- | |  | |  |  | | --- | |  | | Truck rejects package, volume exceeds capacity | pass |
| Add a package to a full truck | Truck: {Weight: 1500 kg, Volume: 48 m³}, Package: {Weight: 50 kg, Volume: 1 m³} | |  |  |  | | --- | --- | --- | | Truck rejects the package (no capacity left) |  |  | | |  |  |  | | --- | --- | --- | | Truck rejects the package (no capacity left) |  |  | | Pass |
|  |  |  |  |  |

**Test Name or ID**: 04

**Test Type**: Black box

**Description**: Checks if a valid path can be calculated from a route to a destination point.

**Setup:** Load the map, define a route, and set a destination point.

**Test Function**:

calcDeliveryPath(map, route, destination)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Calculate delivery path to valid destination | Map: 25x25, Route: Blue, Destination: (5, 5) | Path found, isPossible=1, valid route | Path found, isPossible=1, valid route | pass |
| Calculate delivery path to unreachable point | Map: 25x25, Route: Blue, Destination: (15, 15) blocked | Path not found, isPossible=0 | Path not found, isPossible=0 | pass |
| Calculate delivery path with empty route | Map: 25x25, Route: None, Destination: (5, 5) | |  |  |  | | --- | --- | --- | | |  | | --- | | Path not found, isPossible=0 |  |  | | --- | |  | |  |  | | --- | |  | | Path not found, isPossible=0 | pass |
| Calculate path where destination is the starting point | Map: 25x25, Route: Blue, Destination: (0, 0) | Path found, isPossible=1, zero-length path | Path found, isPossible=1, zero-length path | Pass |
|  |  |  |  |  |

**Test Name or ID**: 05

**Test Type**: Black box

**Description**: Checks if a package can be accepted by a truck based on its weight and volume capacity.

**Setup:** Prepare a truck with predefined current weight and volume. Provide test packages with varying weights and volumes.

**Test Function**:

isPackageAcceptable (truck, package)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Acceptable package within truck capacity | Truck: {Weight: 1000 kg, Volume: 20 m³}, Package: {Weight: 200 kg, Volume: 5 m³} | Return value: 1 (Acceptable) | Return value: 1 (Acceptable) | pass |
| Package exceeding weight capacity | Truck: {Weight: 1400 kg, Volume: 20 m³}, Package: {Weight: 200 kg, Volume: 5 m³} | Return value: 0 (Not Acceptable) | |  | | --- | | Return value: 0 (Not Acceptable) |  |  | | --- | |  | | pass |
| Package exceeding volume capacity | Truck: {Weight: 1000 kg, Volume: 45 m³}, Package: {Weight: 200 kg, Volume: 10 m³} | |  |  |  | | --- | --- | --- | | |  | | --- | | Return value: 0 (Not Acceptable) |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Return value: 0 (Not Acceptable) |  |  | | --- | |  | | pass |
| Package completely exceeds all capacities | Truck: {Weight: 1400 kg, Volume: 45 m³}, Package: {Weight: 200 kg, Volume: 10 m³} | |  | | --- | | Return value: 0 (Not Acceptable) |  |  | | --- | |  | | |  | | --- | | Return value: 0 (Not Acceptable) |  |  | | --- | |  | | Pass |
|  |  |  |  |  |

**Test Name or ID**: 06

**Test Type**: Black box

**Description**: Determines the closest point in a route to a given destination point.

**Setup:** Prepare a route with a set of points and provide a destination point to test.

**Test Function**:

getClosestPoint (route, destination)

**Test Scenarios:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| Closest point within the route | Route: {Points: [(0,0), (5,5), (10,10)]}, Point: (6,6) | Closest point index: 1 | Closest point index: 1 | pass |
| Destination matches a route point | Route: {Points: [(0,0), (5,5), (10,10)]}, Point: (5,5) | Closest point index: 1 | Closest point index: 1 | pass |
| No route points | Route: {Points: []}, Point: (5,5) | |  |  |  | | --- | --- | --- | | |  | | --- | | Closest point index: -1 |  |  | | --- | |  | |  |  | | --- | |  | | Closest point index: -1 | pass |
| Closest point with equidistant points | Route: {Points: [(0,0), (5,5), (10,10)]}, Point: (7,7) | Closest point index: 2 | Closest point index: 2 | Pass |
|  |  |  |  |  |

**Bugs Found**:

No bugs were found during testing. All tests were conducted carefully, and the system behaved as expected.