INTEGRATION TEST CASE DOCUMENT

# Function: getUserInput

## Test Case 1

**Test Description**: Test the function with valid input for weight, box size and destination. This serves to ensure that all input is handled and parsed correctly. All the required validation functions must be called as needed to check if input entered by the user is valid and fits the requirements.

**Precondition**: The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the input is provided, ensure the validation functions validWeight, validVolume, validDestination are called to check the input.

3.When input has been validated and respects the requirements of the program, the function chooseTruckLine is called for follow up details one on the appropriate line or route the trucks are to take to make the delivery.

**Test Data:**

Weight: 10

Box Size: 3

Destination: 5B

**Expected Results:**

validWeight returns 1.  
validVolume returns 1.  
validDestination returns 1.  
chooseTruckLine is called and an accurate truck line to take for delivery should be displayed.

## Test Case 2

**Test Description:** Test the function handling user input when one value given is not meeting the requirements. Checking the weight validation function called for its error handling and accurate display of error message for invalid weight given by the user.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the input is provided, ensure the validation functions validWeight, validVolume, validDestination are called to check the input.

3. When input has been validated and respects the requirements of the program, the function chooseTruckLine is called for follow up details one on the appropriate line or route the trucks are to take to make the delivery.

**Test Data:**

Weight: 2560

Box Size: 3

Destination: 5B

**Expected Results:**

The program should not crash due to the invalid input given by the user.

validWeight returns 0.  
The message for invalid weight input should be displayed.

The user should be prompted to re-enter valid inputs.

## Test Case 3

**Test Description:** Test the function handling user input when one value given is not meeting the requirements. Checking the box size validation function called for its error handling and accurate display of error message for invalid size given by the user.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the input is provided, ensure the validation functions validWeight, validVolume, validDestination are called to check the input.

3. When input has been validated and respects the requirements of the program, the function chooseTruckLine is called for follow up details one on the appropriate line or route the trucks are to take to make the delivery.

**Test Data:**

Weight: 12

Box Size: 4

Destination: 5B

**Expected Results:**

The program should not crash due to invalid input for box size.

validWeight returns 1.  
validVolume returns 0.  
The message for invalid size input should be displayed.

The user should be prompted to re-enter valid inputs.

## Test Case 4

**Test Description:** Test the function handling user input when one value given is not meeting the requirements. Checking the destination validation function called for its error handling and accurate display of error message for invalid destination given by the user.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the input is provided, ensure the validation functions validWeight, validVolume, validDestination are called to check the input.

3. When input has been validated and respects the requirements of the program, the function chooseTruckLine is called for follow up details one on the appropriate line or route the trucks are to take to make the delivery.

**Test Data:**

Weight: 10

Box Size: 4

Destination: 34Z

**Expected Results:**

The program should not crash due to the invalid destination entered by the user.

validWeight returns 1.  
validVolume returns 1.  
validDestination returns 0.  
The message for invalid destination input should be displayed.

The program should prompt the user to re-enter valid inputs.

# Function: chooseTruckLine

## Test Case 1

**Test Description:** Test function for selecting the appropriate truck line needed to take for delivery when all trucks available have sufficient capacity. This serves to ensure that the function call of canCarry, compareTrucks, printDiversion happen smoothly and the returned results are used to display the required and accurate information about the delivery.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the validation of the input has been executed, the function canCarry is called to verify if there is a truck available to carry out the delivery.

3. After trucks have been confirmed to have the capacity to carry the package, the function compareTrucks function is called to check in the available trucks for delivery, which would be the most appropriate for delivery.

4. After the appropriate truck has been chose for the delivery, the function printDiversion is called to display the accurate route the truck is to take to ensure that the package reaches the entered destination by the user.

**Test Data:**

Weight: 500

Box Size: 5

Destination: 10J

**Expected Results:**

**canCarry** returns 1

**compareTrucks** returns 0 (indicating both trucks have enough space)  
The appropriate truck line to take for delivery should be displayed.

## Test Case 2

**Test Description:** Test function for handling the case when all trucks do not have sufficient capacity to make a delivery. This serves to ensure that the function call of canCarry, compareTrucks, printDiversion happen smoothly and the returned results are used to display the required and accurate information about the delivery.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the validation of the input has been executed, the function canCarry is called to verify if there is a truck available to carry out the delivery.

3. After trucks have been confirmed to have the capacity to carry the package, the function compareTrucks function is called to check in the available trucks for delivery, which would be the most appropriate for delivery.

4. After the appropriate truck has been chose for the delivery, the function printDiversion is called to display the accurate route the truck is to take to ensure that the package reaches the entered destination by the user.

**Test Data:**

Weight: 2500

Box Size: 5

Destination: 10J

**Expected Results:**

canCarry returns 0.

No truck line is selected for this delivery.  
'Ships tomorrow' message should be displayed.

## Test Case 3

**Test Description:** Test function for handling the case where in the trucks available, the left has more capacity than the right truck. This serves to ensure that the function call of canCarry, compareTrucks, printDiversion happen smoothly and the returned results are used to display the required and accurate information about the delivery.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the validation of the input has been executed, the function canCarry is called to verify if there is a truck available to carry out the delivery.

3. After trucks have been confirmed to have the capacity to carry the package, the function compareTrucks function is called to check in the available trucks for delivery, which would be the most appropriate for delivery.

4. After the appropriate truck has been chose for the delivery, the function printDiversion is called to display the accurate route the truck is to take to ensure that the package reaches the entered destination by the user.

**Test Data:**

Weight: 560

Box Size: 3

Destination: 15D

**Expected Results:**

**canCarry** returns 1

**compareTrucks** returns 1 (indicating left truck has more space)

A suitable truck is chosen to make the delivery.  
The appropriate truck line to take for delivery should be displayed.

## Test Case 4

**Test Description:** Test function for handling the case where in the trucks available, the right has more capacity compared to the left truck. This serves to ensure that the function call of canCarry, compareTrucks, printDiversion happen smoothly and the returned results are used to display the required and accurate information about the delivery.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. After the validation of the input has been executed, the function canCarry is called to verify if there is a truck available to carry out the delivery.

3. After trucks have been confirmed to have the capacity to carry the package, the function compareTrucks function is called to check in the available trucks for delivery, which would be the most appropriate for delivery.

4. After the appropriate truck has been chose for the delivery, the function printDiversion is called to display the accurate route the truck is to take to ensure that the package reaches the entered destination by the user.

**Test Data:**

Weight: 560

Box Size: 3

Destination: 15D

**Expected Results:**

**canCarry** returns 1

**compareTrucks** returns -1 (indicating right truck has more space)

A suitable truck is chosen to make the delivery.  
The appropriate truck line to take for delivery should be displayed.

# Function: printDiversion

## Test Case 1

**Test Description:** Test the function handling cases where no diversion is needed by the truck to make a delivery. Ensure the correct message is displayed when no diversions are necessary. The function call for convertPoint should be able to accurately convert a point into printable values.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. Function convertPoint is called to convert a point into printable values to help display the points in a easier more readable format.

**Test Data:**

Weight: 1500 kg

Box size: 3 cubic meters

Destination: 7E

**Expected Results:**

convertPoint should return an integer containing the row.

“No diversion” should be displayed.

YELLOW LINE should be the route to displayed to take.

## Test Case 2

**Test Description:** Test the function handling cases where a simple diversion is needed by the truck. Ensure the correct message is displayed for the diversions necessary to make the delivery. The function call for convertPoint should be able to accurately convert a point into printable values.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. Function convertPoint is called to convert a point into printable values to help display the points in a easier more readable format.

**Test Data:**

Weight: 150

Box size: 1

Destination: 8B

Expected Results:

convertPoint should return an integer containing the row.

YELLOW LINE should be the route to displayed to take.

The statement: “divert: 8D, 8C, 8B" should be displayed as well.

## Test Case 3

**Test Description:** Test the function handling cases where a diversion of less than 3 diversion points is needed by the truck. Ensure the correct message is displayed for the diversions necessary to make the delivery. The function call for convertPoint should be able to accurately convert a point into printable values.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. Function convertPoint is called to convert a point into printable values to help display the points in an easier more readable format.

**Test Data:**

Weight: 15

Box size: 1

Destination: 3C

Expected Results:

convertPoint should return an integer containing the row.

BLUE LINE should be the route to displayed to take.

The statement: 'divert: 1A, 2B, 3C' message should be displayed.

## Test Case 4

**Test Description:** Test the function handling cases where a complex diversion is needed by the truck. Ensure the correct message is displayed for the diversions necessary to make the delivery. The function call for convertPoint should be able to accurately convert a point into printable values.

**Precondition:** The program must be able to start and run smoothly. The program should also be able to request and accept input from the user.

**Steps:**

1. Provide the following input values:

(Add space after every input)

a. Enter weight (kilograms)

b. Enter box size (cubic meters)

c. Enter destination

2. Function convertPoint is called to convert a point into printable values to help display the points in an easier more readable format.

**Test Data:**

Weight: 560 kg

Box size: 3 cubic meters

Destination: 15D

**Expected Results:**

convertPoint should return an integer containing the row.

YELLOW LINE should be the route to displayed to take.

The statement: “divert: 20D, 19D, 18D, 17D, 16D, 15D" should be displayed.