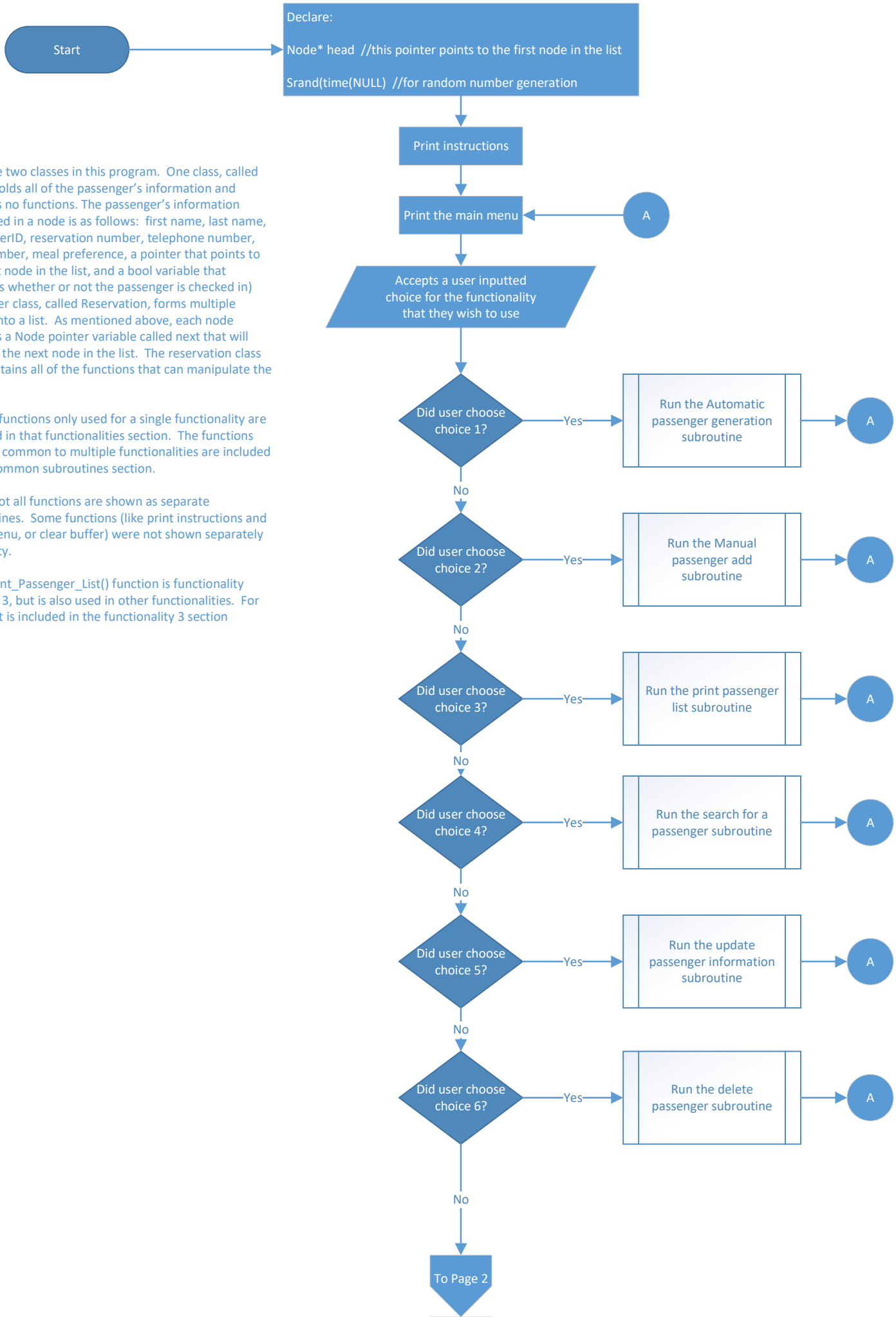
Notes:

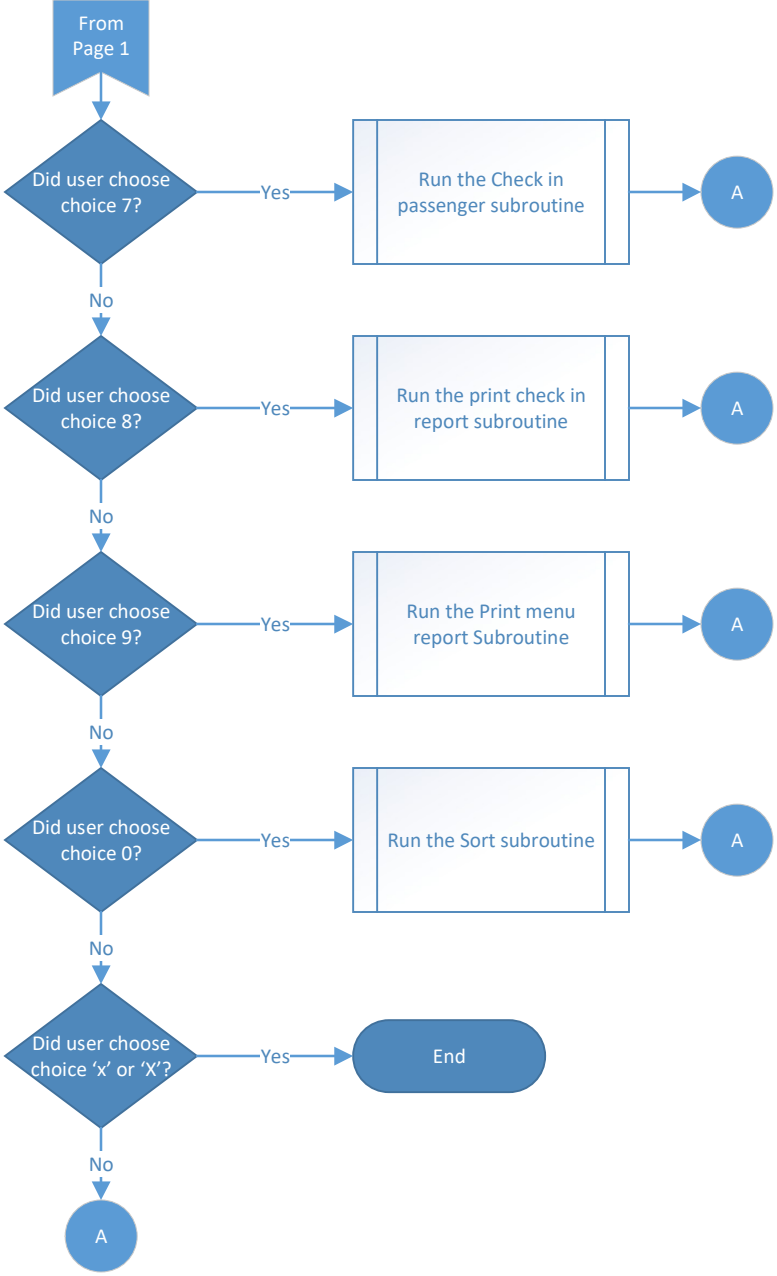
We use two classes in this program. One class, called Node, holds all of the passenger's information and contains no functions. The passenger's information contained in a node is as follows: first name, last name, passengerID, reservation number, telephone number, seat number, meal preference, a pointer that points to the next node in the list, and a bool variable that indicates whether or not the passenger is checked in) The other class, called Reservation, forms multiple Nodes into a list. As mentioned above, each node contains a Node pointer variable called next that will point to the next node in the list. The reservation class also contains all of the functions that can manipulate the list.

all the functions only used for a single functionality are included in that functionalities section. The functions that are common to multiple functionalities are included in the common subroutines section.

Also, not all functions are shown as separate subroutines. Some functions (like print instructions and main menu, or clear buffer) were not shown separately for clarity.

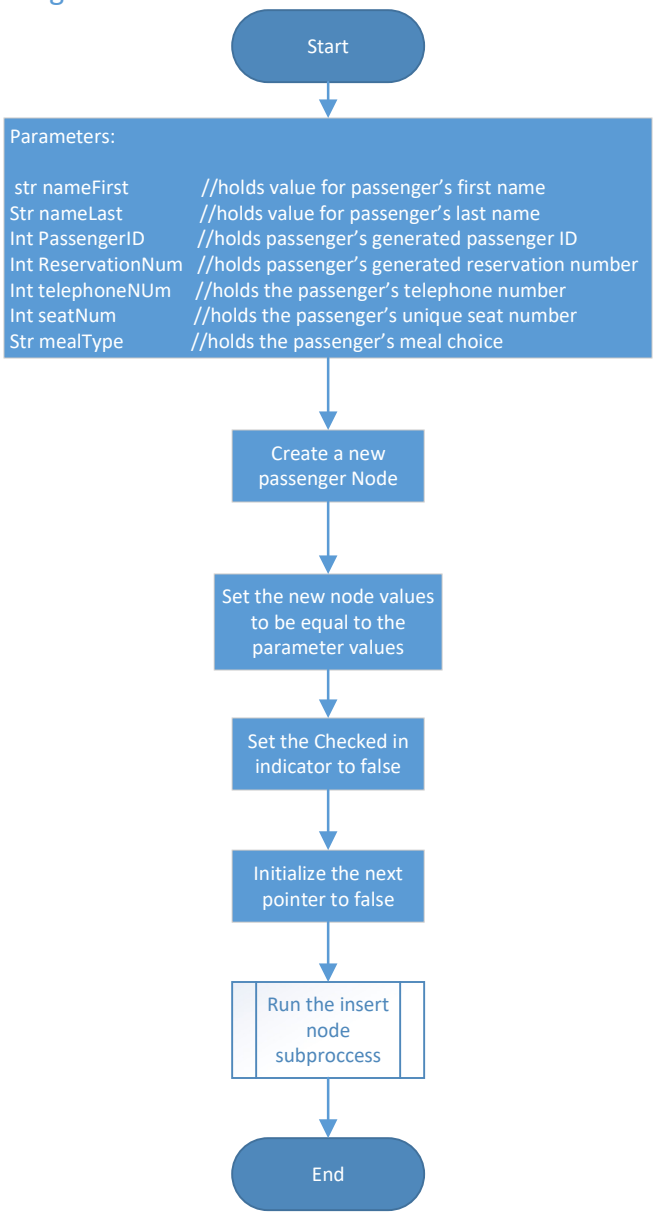
The Print_Passenger_List() function is functionality number 3, but is also used in other functionalities. For clarity, it is included in the functionality 3 section




From Page 1
Did user choose choice 7?
Yes
Run the Check in passenger subroutine
A
No
Did user choose choice 8?
Yes
Run the print check in report subroutine
A
No
Did user choose choice 9?
Yes
Run the Print menu report Subroutine
A
No
Did user choose choice 0?
Yes
Run the Sort subroutine
A
No
Did user choose choice 'x' or 'X'?
Yes
End
No
A

Create New Node

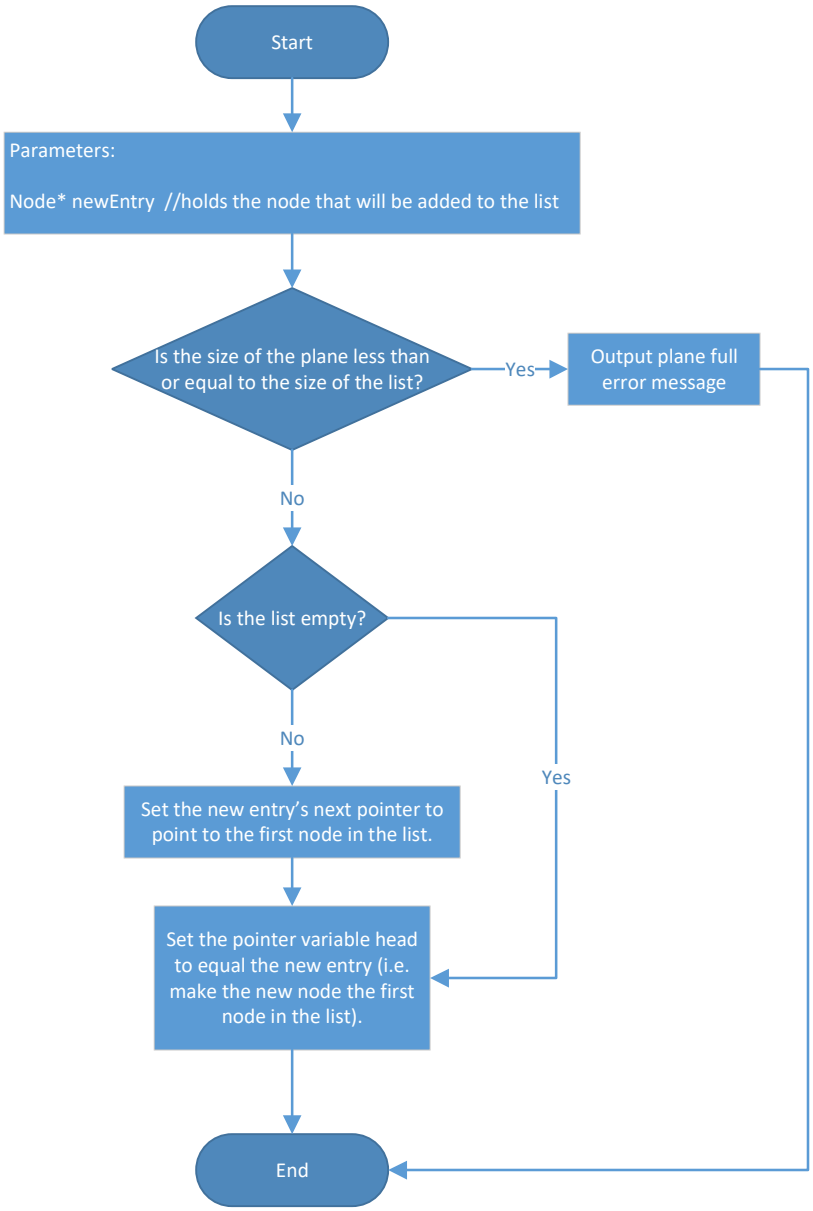
Refers to the function CreateNewNode()
Will add a node to the begginning of the list



Insert Node

Refers to the insertNode() function

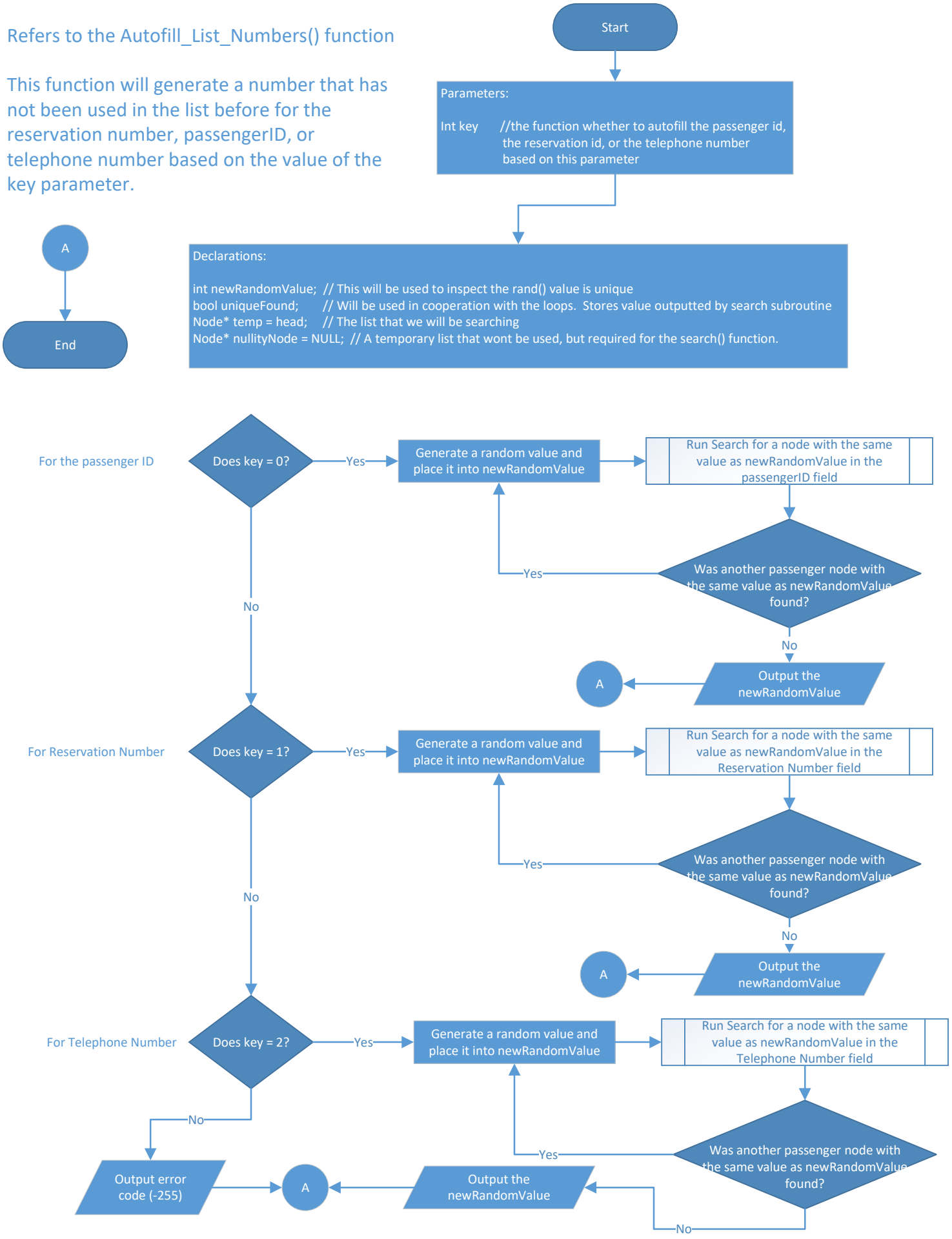
Only used when the create new node subroutine is called



Autofill List numbers

Refers to the Autofill_List_Numbers() function

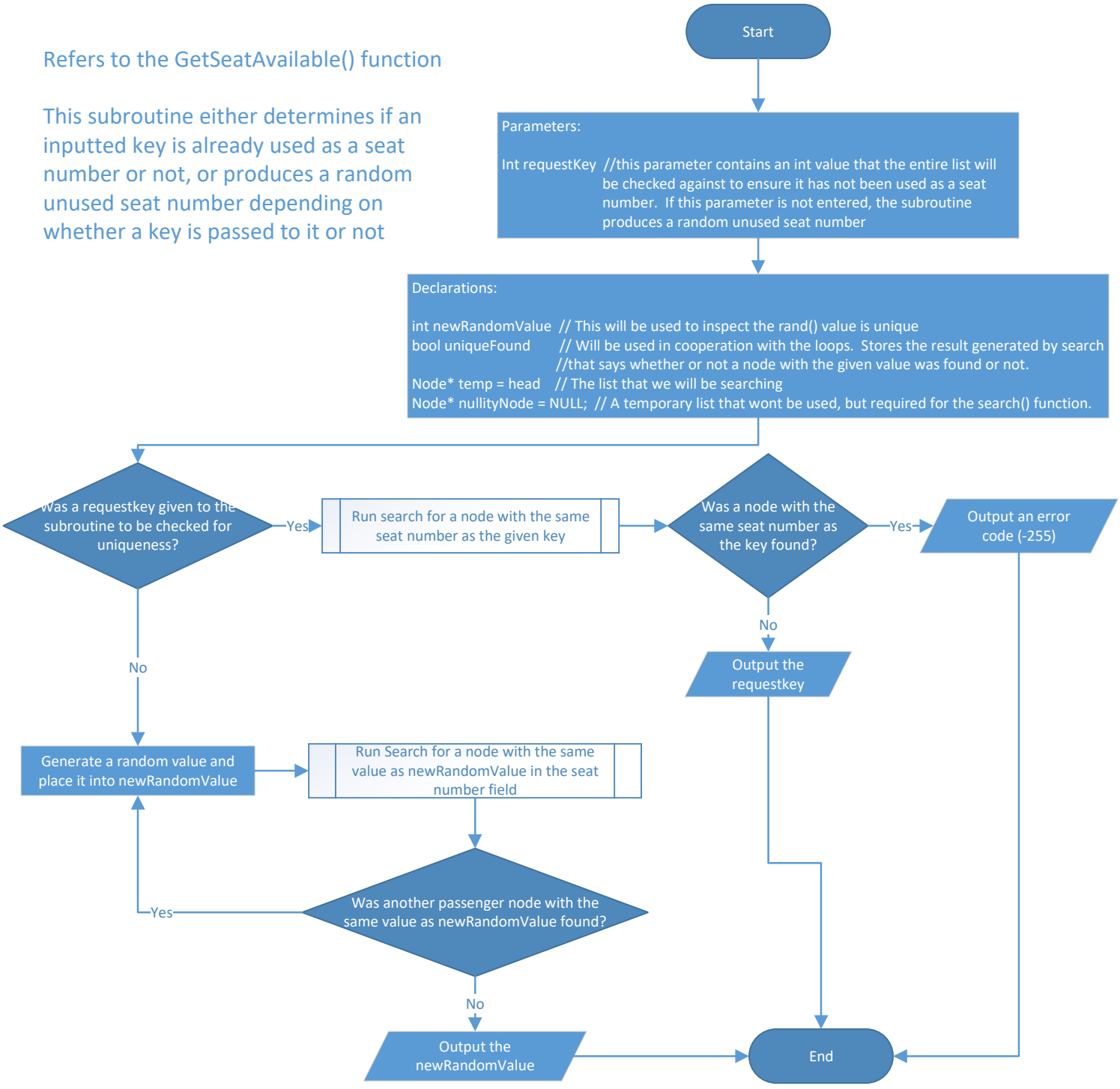
This function will generate a number that has not been used in the list before for the reservation number, passengerID, or telephone number based on the value of the key parameter.



Get Seat Available

Refers to the GetSeatAvailable() function

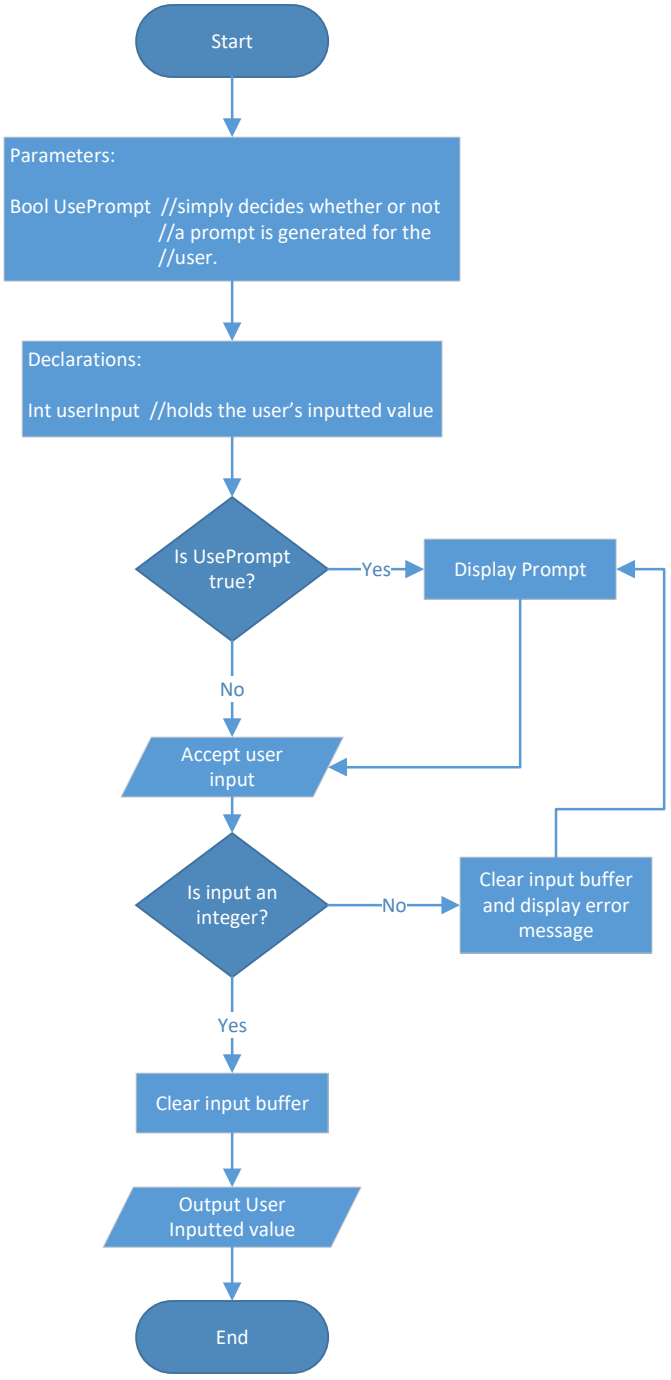
This subroutine either determines if an inputted key is already used as a seat number or not, or produces a random unused seat number depending on whether a key is passed to it or not



User Input Number

Refers to the UserInput_Number() function
This function was shown because it contains some
Error checking that ensures an integer value is entered.

This function is used to process a user inputted integer



Search

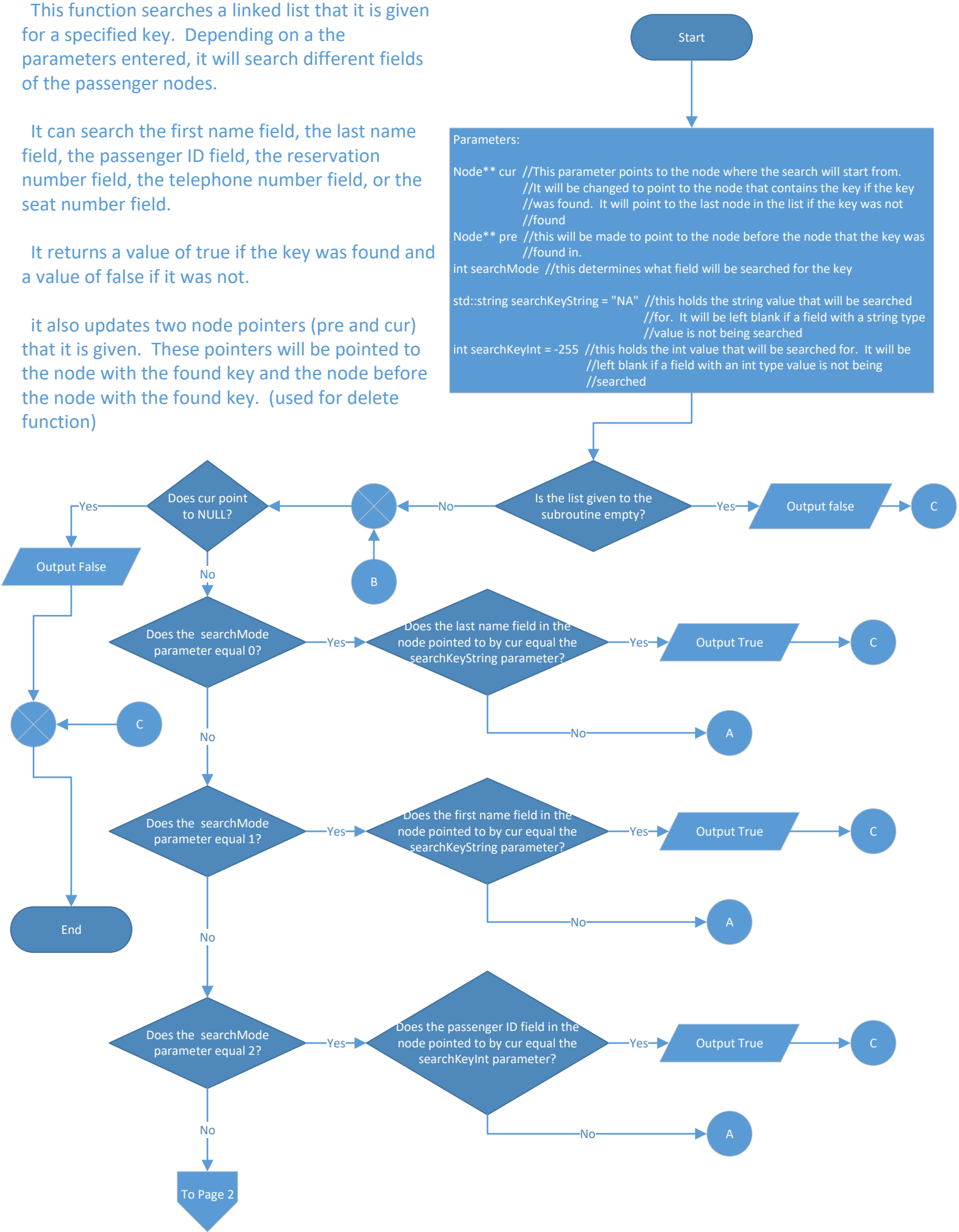
Refers to the Search() function

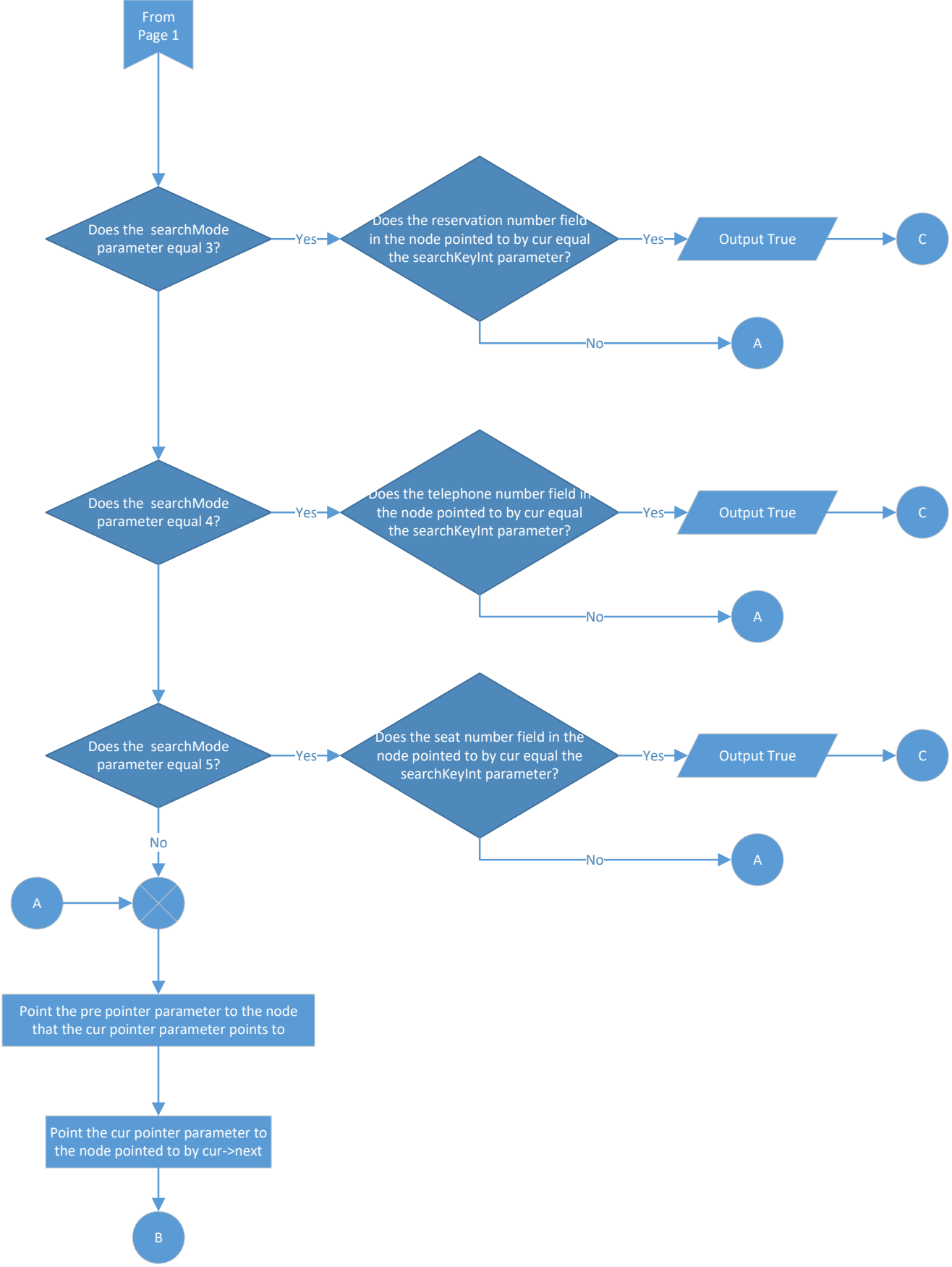
This function searches a linked list that it is given for a specified key. Depending on a the parameters entered, it will search different fields of the passenger nodes.

It can search the first name field, the last name field, the passenger ID field, the reservation number field, the telephone number field, or the seat number field.

It returns a value of true if the key was found and a value of false if it was not.

it also updates two node pointers (pre and cur) that it is given. These pointers will be pointed to the node with the found key and the node before the node with the found key. (used for delete function)




From Page 1
Does the searchMode parameter equal 3?
Yes
Does the reservation number field in the node pointed to by cur equal the searchKeyInt parameter?
Yes
Output True
C
No
A
Does the searchMode parameter equal 4?
Yes
Does the telephone number field in the node pointed to by cur equal the searchKeyInt parameter?
Yes
Output True
C
No
A
Does the searchMode parameter equal 5?
Yes
Does the seat number field in the node pointed to by cur equal the searchKeyInt parameter?
Yes
Output True
C
No
A
No
A
Point the pre pointer parameter to the node that the cur pointer parameter points to
Point the cur pointer parameter to the node pointed to by cur->next
B

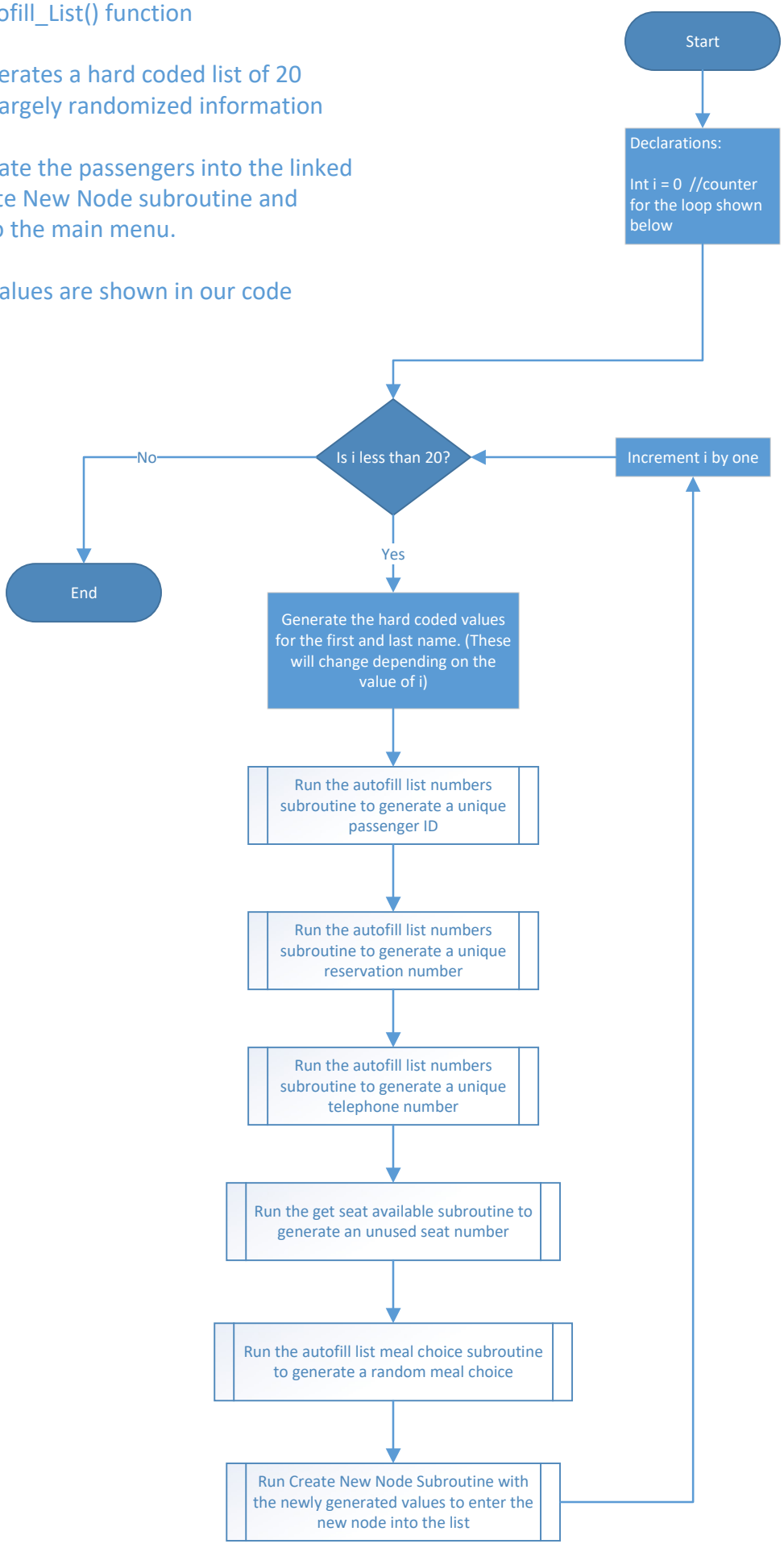
Automatic Passenger Generation

Refers to the Autofill_List() function

This function generates a hard coded list of 20 passengers with largely randomized information

Will simply generate the passengers into the linked list with the Create New Node subroutine and return the user to the main menu.

The hard coded values are shown in our code



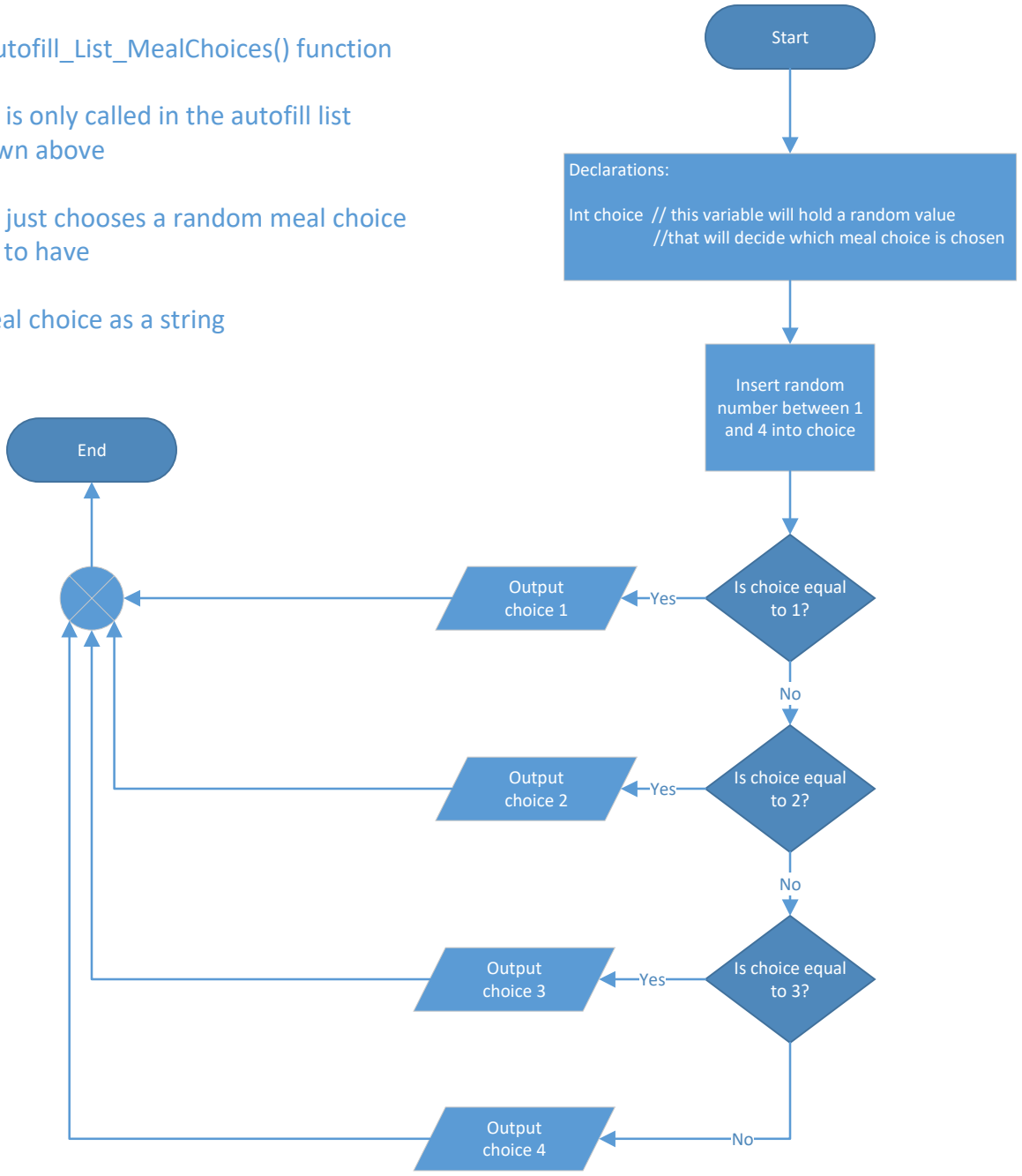
Autofill List Meal Choices

Refers to the Autofill_List_MealChoices() function

This subroutine is only called in the autofill list subroutine shown above

This subroutine just chooses a random meal choice for a passenger to have

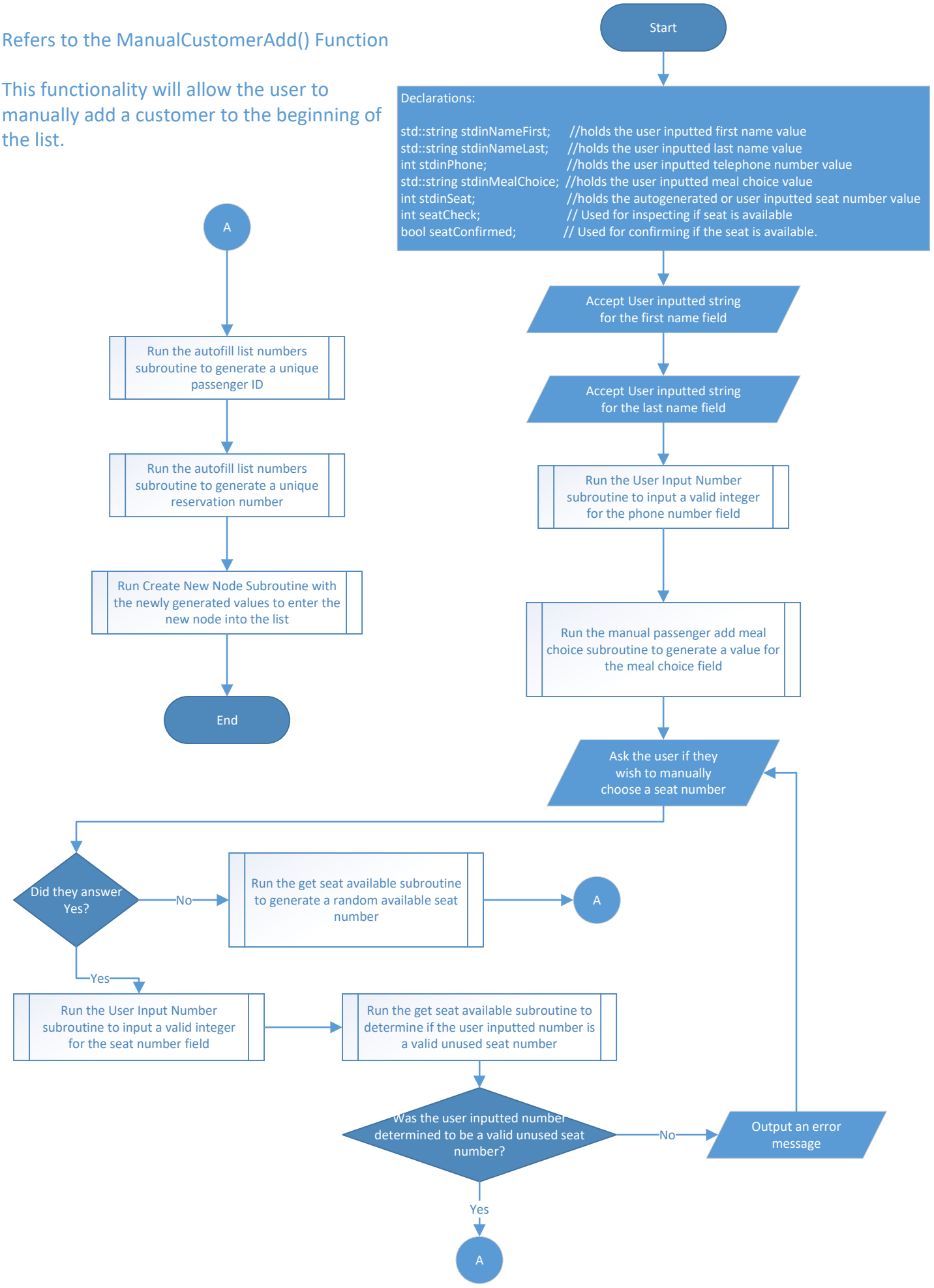
Outputs the meal choice as a string



Manual Passenger Add

Refers to the ManualCustomerAdd() Function

This functionality will allow the user to manually add a customer to the beginning of the list.



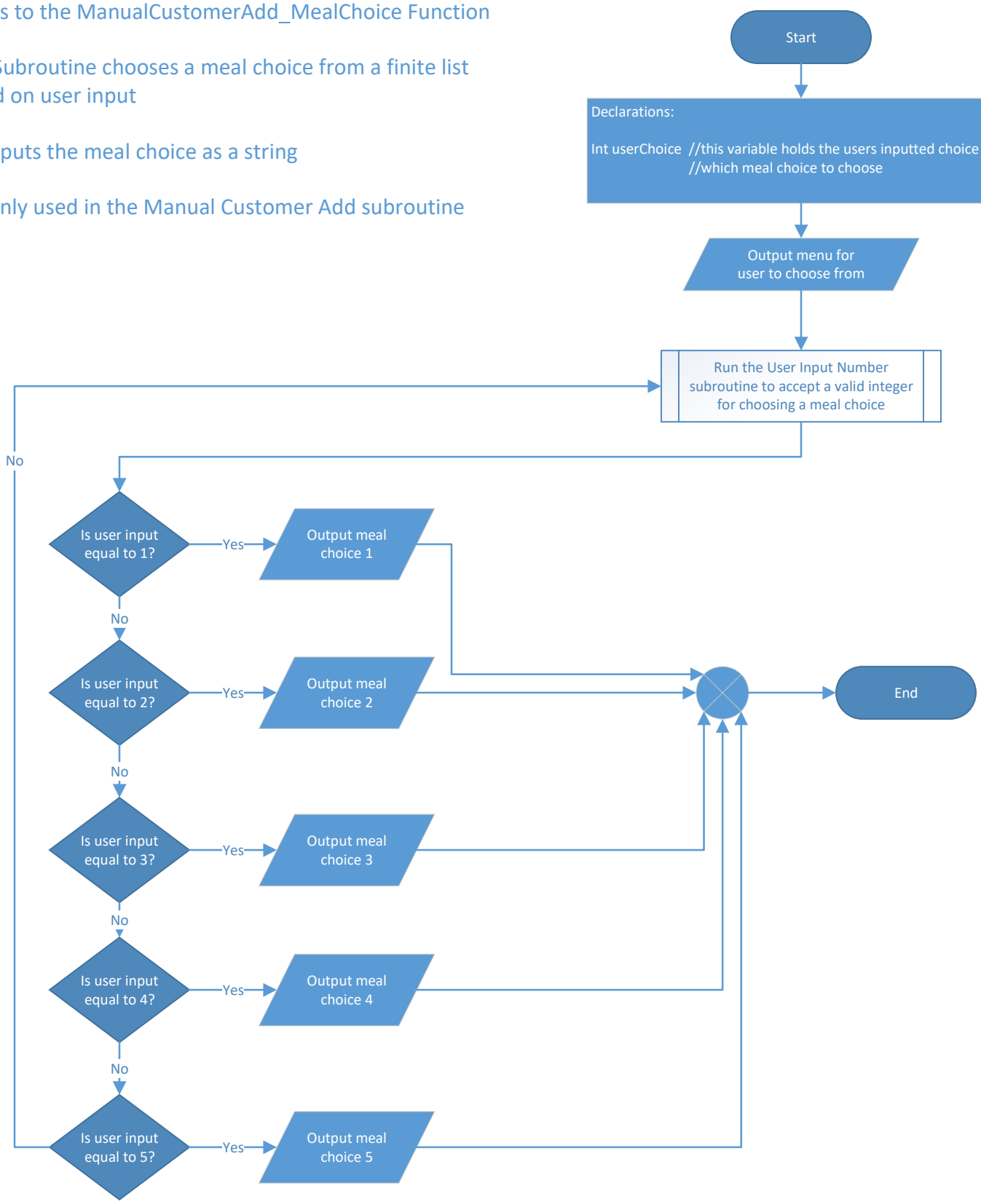
Manual Passenger Add Meal Choice

Refers to the ManualCustomerAdd_MealChoice Function

This Subroutine chooses a meal choice from a finite list based on user input

It outputs the meal choice as a string

It is only used in the Manual Customer Add subroutine



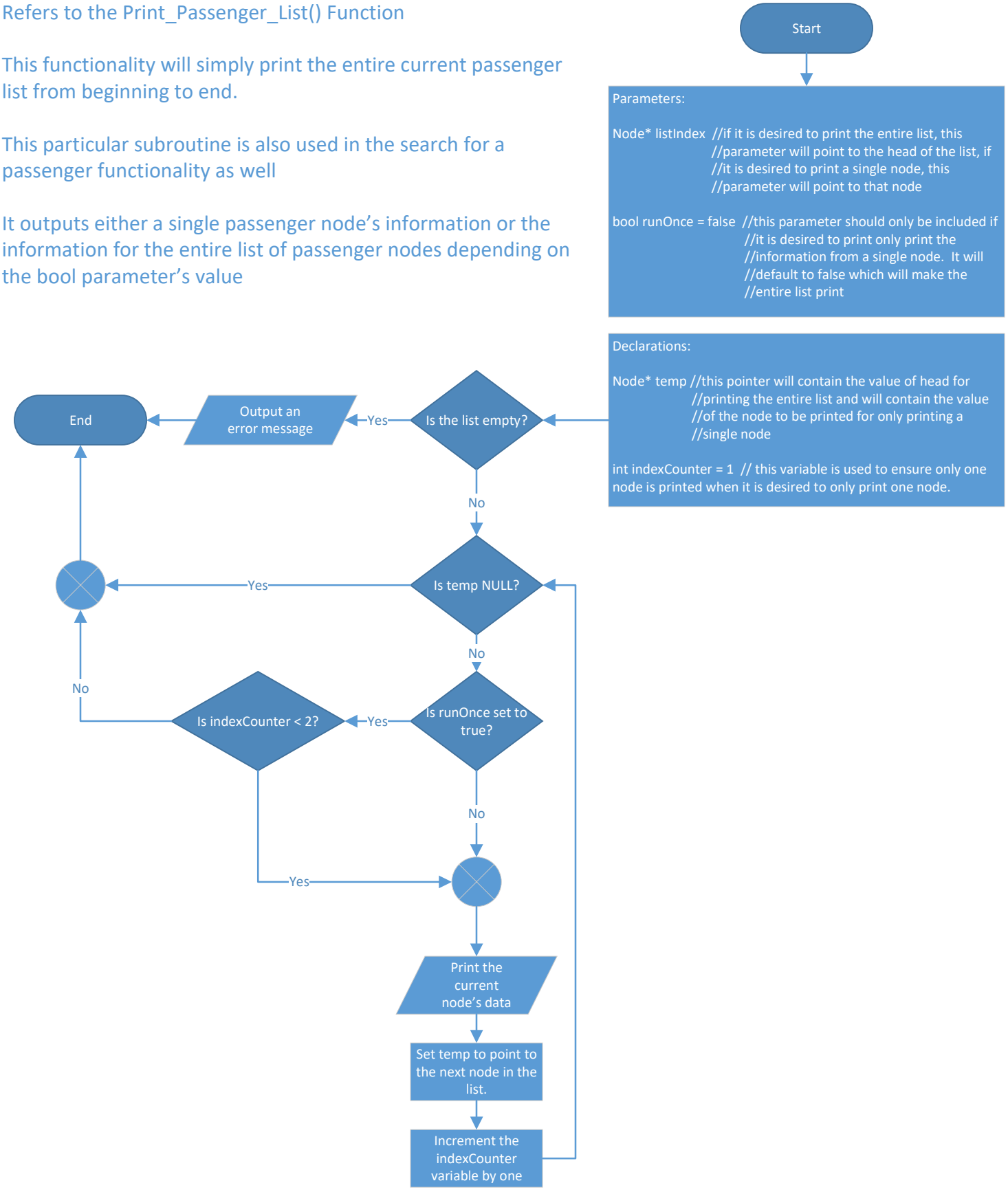
Print Passenger List

Refers to the Print_Passenger_List() Function

This functionality will simply print the entire current passenger list from beginning to end.

This particular subroutine is also used in the search for a passenger functionality as well

It outputs either a single passenger node's information or the information for the entire list of passenger nodes depending on the bool parameter's value



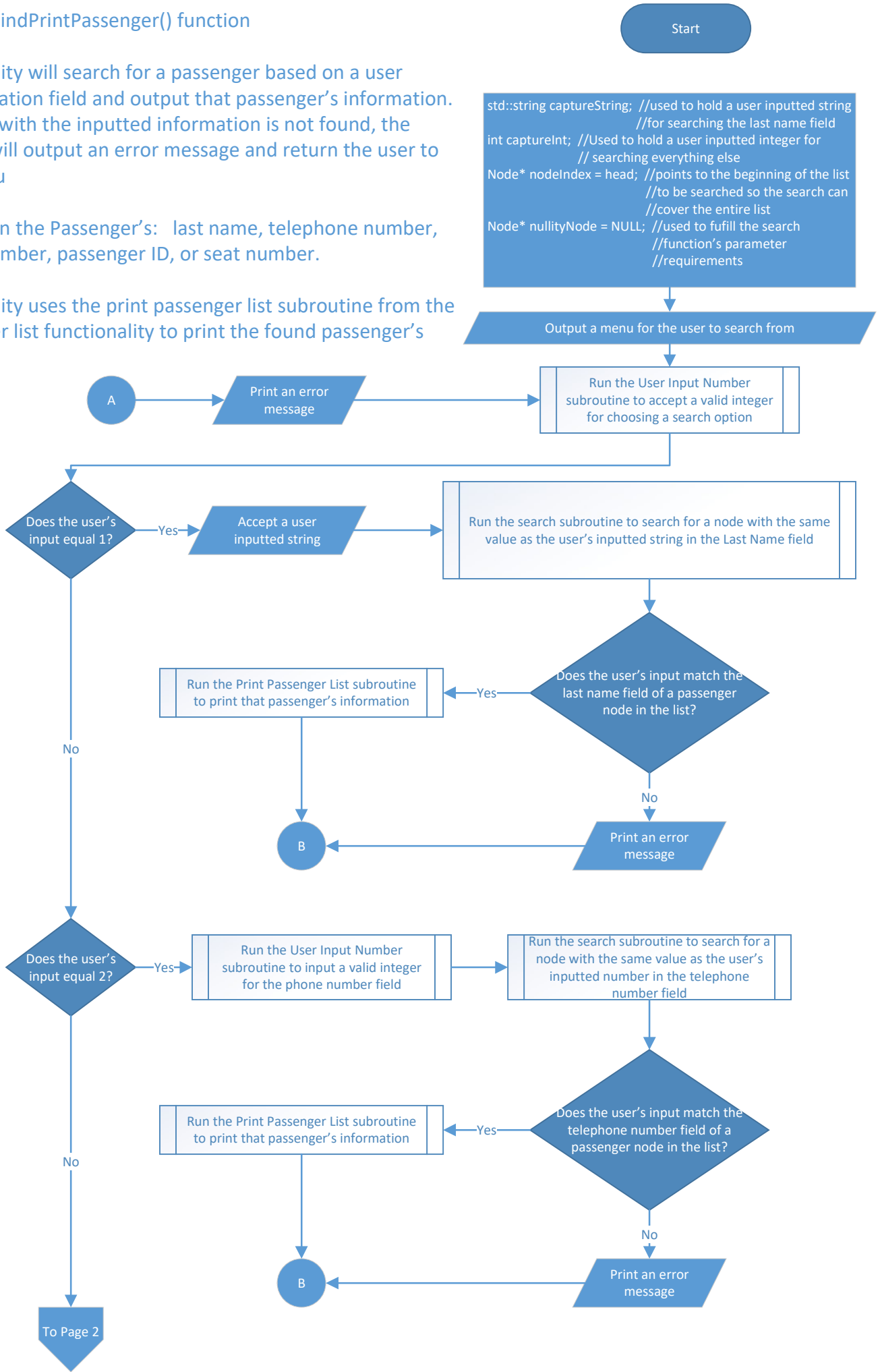
Search for a Passenger

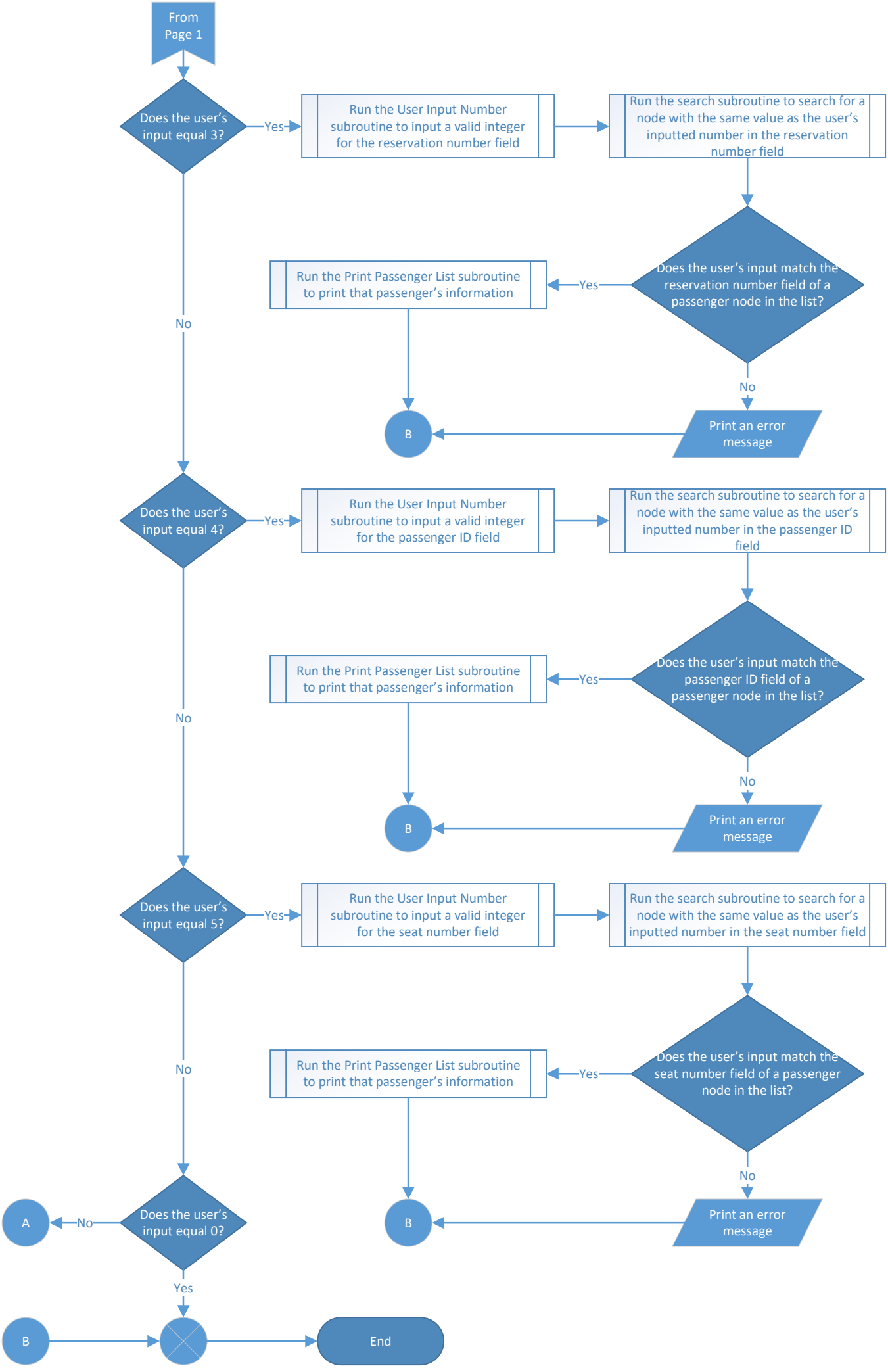
Refers to the FindPrintPassenger() function

This functionality will search for a passenger based on a user chosen information field and output that passenger's information. If a passenger with the inputted information is not found, the functionality will output an error message and return the user to the main menu

It can search on the Passenger's: last name, telephone number, reservation number, passenger ID, or seat number.

This functionality uses the print passenger list subroutine from the print passenger list functionality to print the found passenger's information




From Page 1
Does the user's input equal 3?
Yes
Run the User Input Number subroutine to input a valid integer for the reservation number field
Run the search subroutine to search for a node with the same value as the user's inputted number in the reservation number field
Does the user's input match the reservation number field of a passenger node in the list?
Yes
Run the Print Passenger List subroutine to print that passenger's information
No
Print an error message
B
No
Does the user's input equal 4?
Yes
Run the User Input Number subroutine to input a valid integer for the passenger ID field
Run the search subroutine to search for a node with the same value as the user's inputted number in the passenger ID field
Does the user's input match the passenger ID field of a passenger node in the list?
Yes
Run the Print Passenger List subroutine to print that passenger's information
No
Print an error message
B
No
Does the user's input equal 5?
Yes
Run the User Input Number subroutine to input a valid integer for the seat number field
Run the search subroutine to search for a node with the same value as the user's inputted number in the seat number field
Does the user's input match the seat number field of a passenger node in the list?
Yes
Run the Print Passenger List subroutine to print that passenger's information
No
Print an error message
B
No
Does the user's input equal 0?
No
A
Yes
B
End

Update Passenger Information

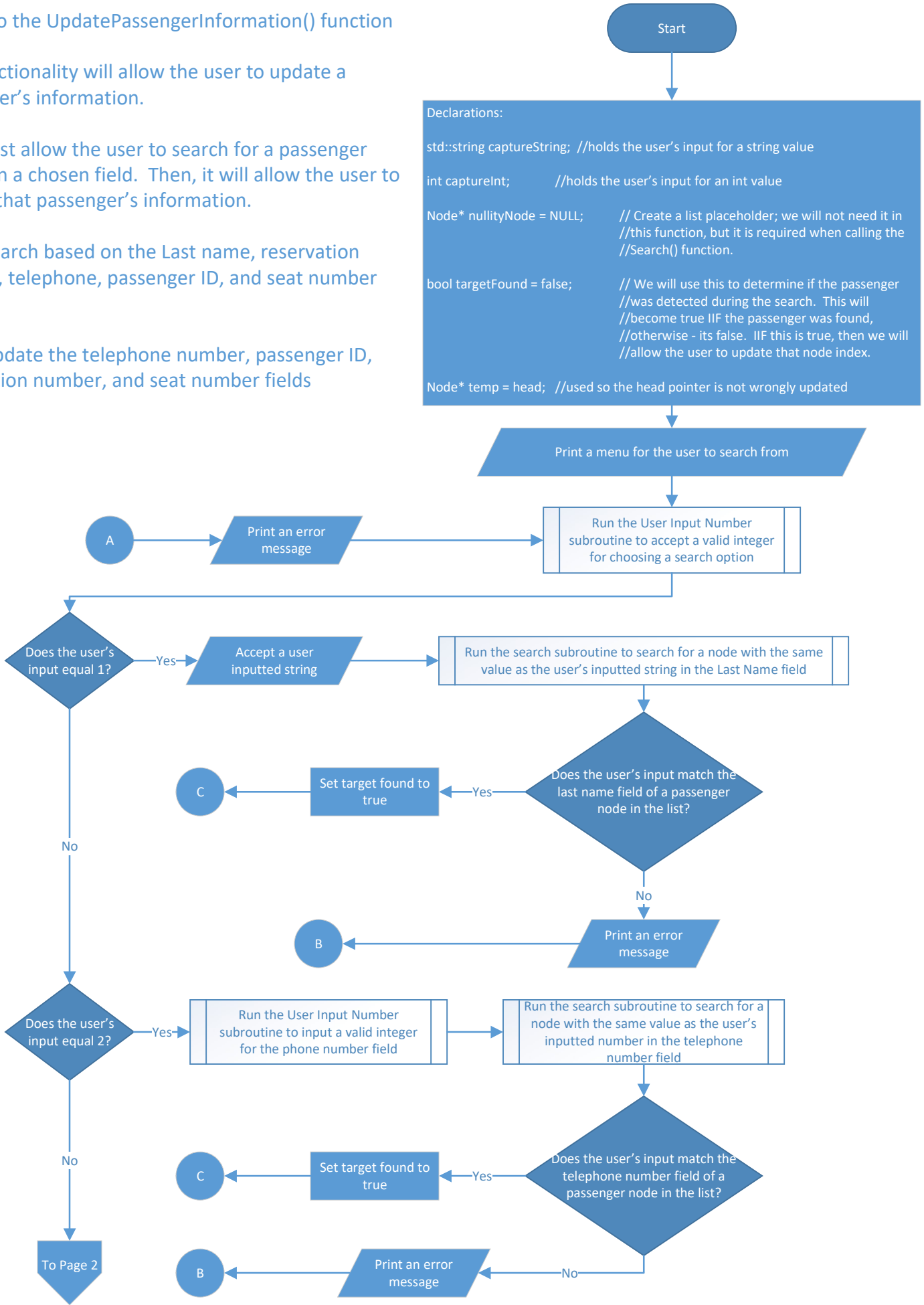
Refers to the UpdatePassengerInformation() function

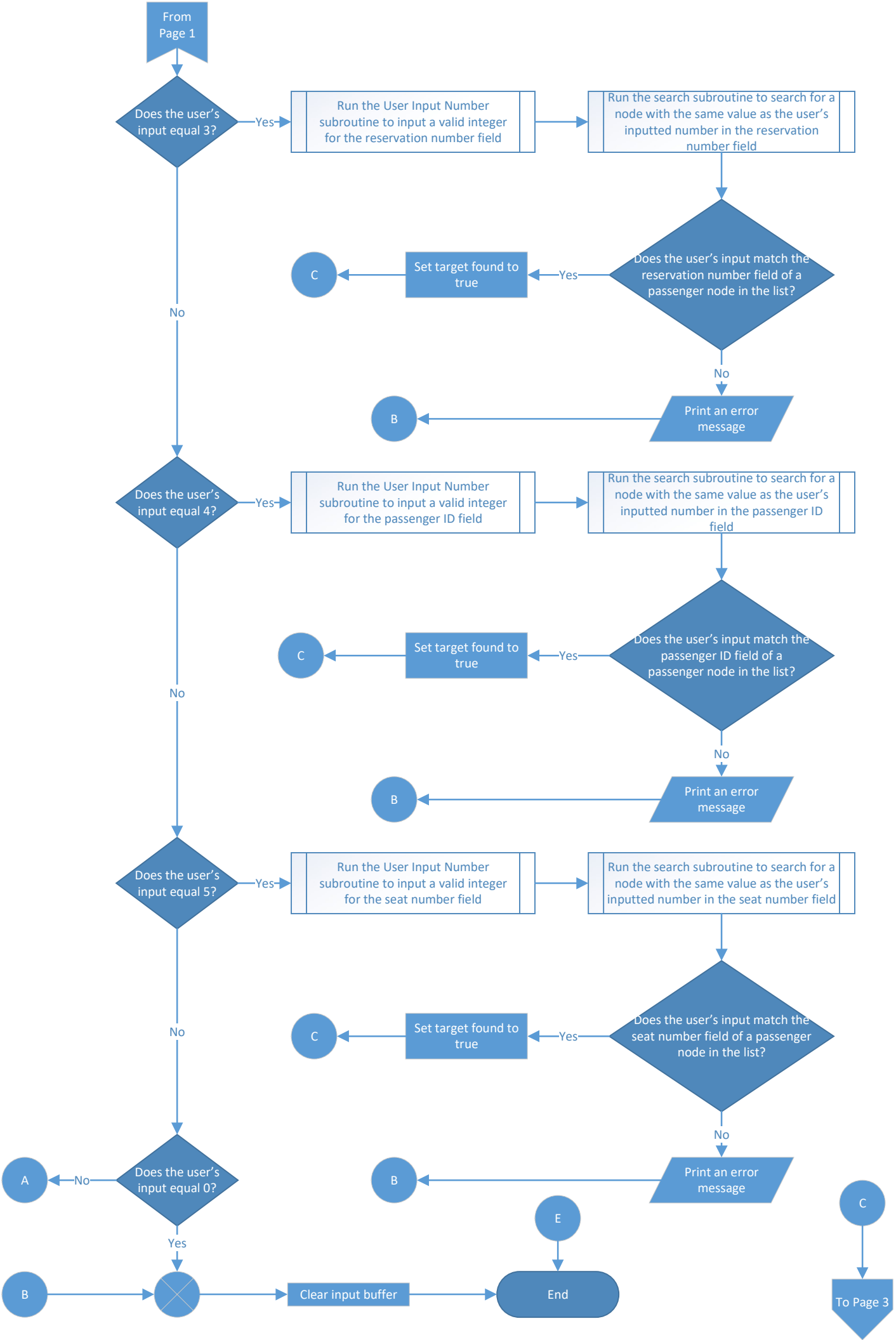
This functionality will allow the user to update a passenger's information.

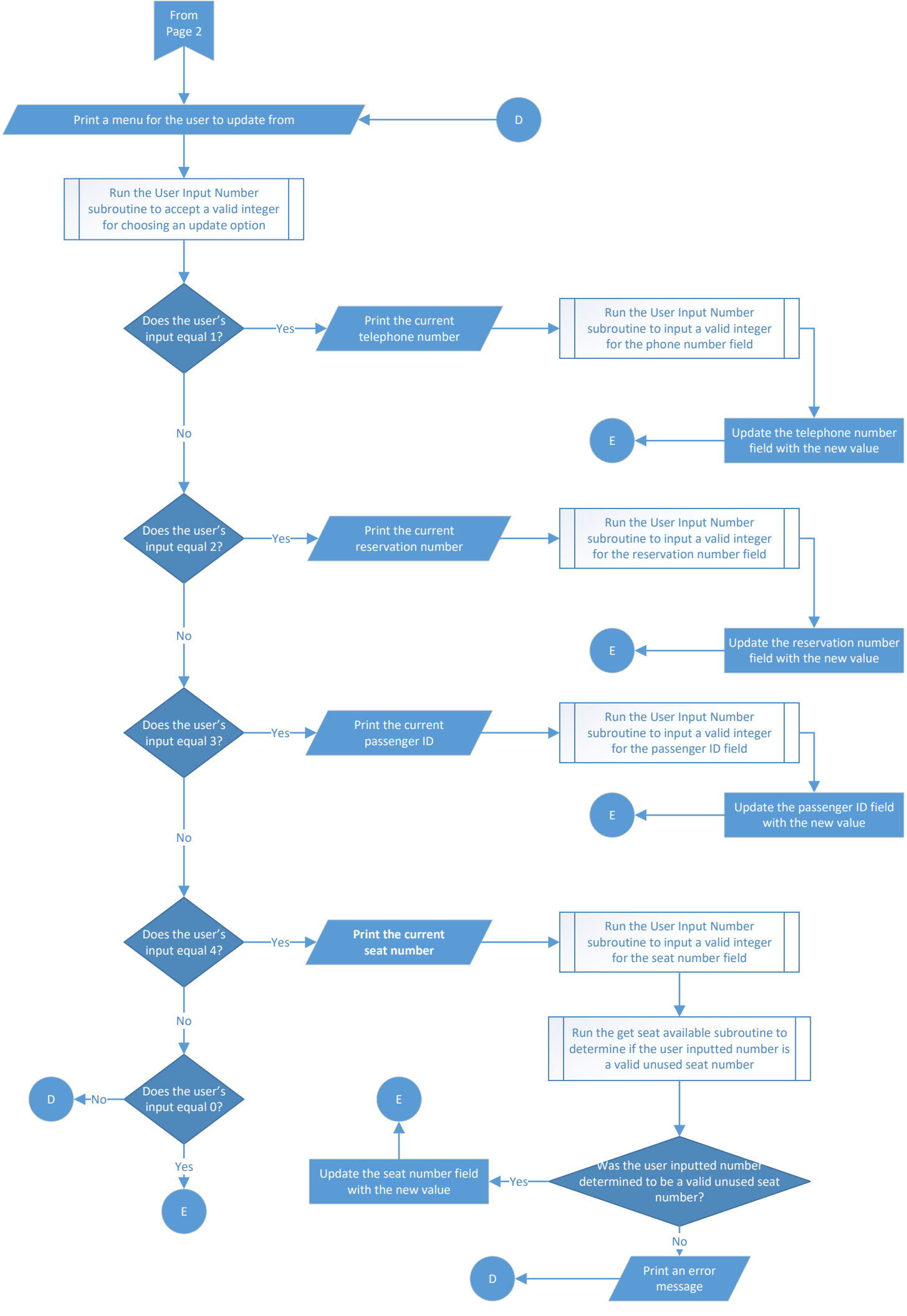
It will first allow the user to search for a passenger based on a chosen field. Then, it will allow the user to update that passenger's information.

It can search based on the Last name, reservation number, telephone, passenger ID, and seat number fields.

It can update the telephone number, passenger ID, reservation number, and seat number fields




From Page 1
Does the user's input equal 3?
Yes
Run the User Input Number subroutine to input a valid integer for the reservation number field
Run the search subroutine to search for a node with the same value as the user's inputted number in the reservation number field
Does the user's input match the reservation number field of a passenger node in the list?
Yes
Set target found to true
C
No
Print an error message
B
No
Does the user's input equal 4?
Yes
Run the User Input Number subroutine to input a valid integer for the passenger ID field
Run the search subroutine to search for a node with the same value as the user's inputted number in the passenger ID field
Does the user's input match the passenger ID field of a passenger node in the list?
Yes
Set target found to true
C
No
Print an error message
B
No
Does the user's input equal 5?
Yes
Run the User Input Number subroutine to input a valid integer for the seat number field
Run the search subroutine to search for a node with the same value as the user's inputted number in the seat number field
Does the user's input match the seat number field of a passenger node in the list?
Yes
Set target found to true
C
No
Print an error message
B
No
Does the user's input equal 0?
No
A
Yes
B
Clear input buffer
End
E
C
To Page 3

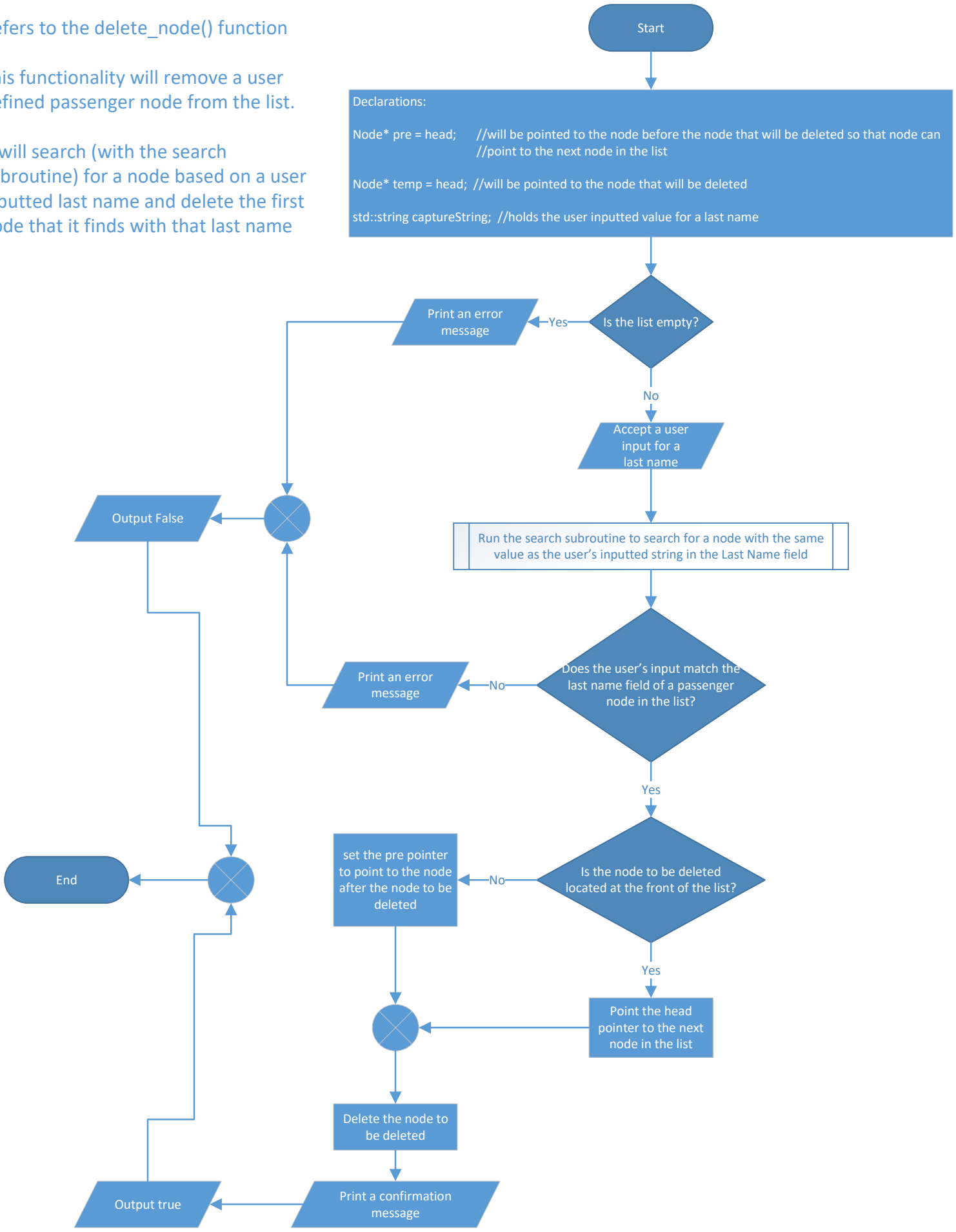

From Page 2
Print a menu for the user to update from
D
Run the User Input Number subroutine to accept a valid integer for choosing an update option
Does the user's input equal 1?
Yes
Print the current telephone number
Run the User Input Number subroutine to input a valid integer for the phone number field
Update the telephone number field with the new value
E
No
Does the user's input equal 2?
Yes
Print the current reservation number
Run the User Input Number subroutine to input a valid integer for the reservation number field
Update the reservation number field with the new value
E
No
Does the user's input equal 3?
Yes
Print the current passenger ID
Run the User Input Number subroutine to input a valid integer for the passenger ID field
Update the passenger ID field with the new value
E
No
Does the user's input equal 4?
Yes
Print the current seat number
Run the User Input Number subroutine to input a valid integer for the seat number field
Run the get seat available subroutine to determine if the user inputted number is a valid unused seat number
Was the user inputted number determined to be a valid unused seat number?
Yes
Update the seat number field with the new value
E
No
Print an error message
D
No
Does the user's input equal 0?
D
No
Yes
E

Delete Passenger

Refers to the delete_node() function

This functionality will remove a user defined passenger node from the list.

It will search (with the search subroutine) for a node based on a user inputted last name and delete the first node that it finds with that last name

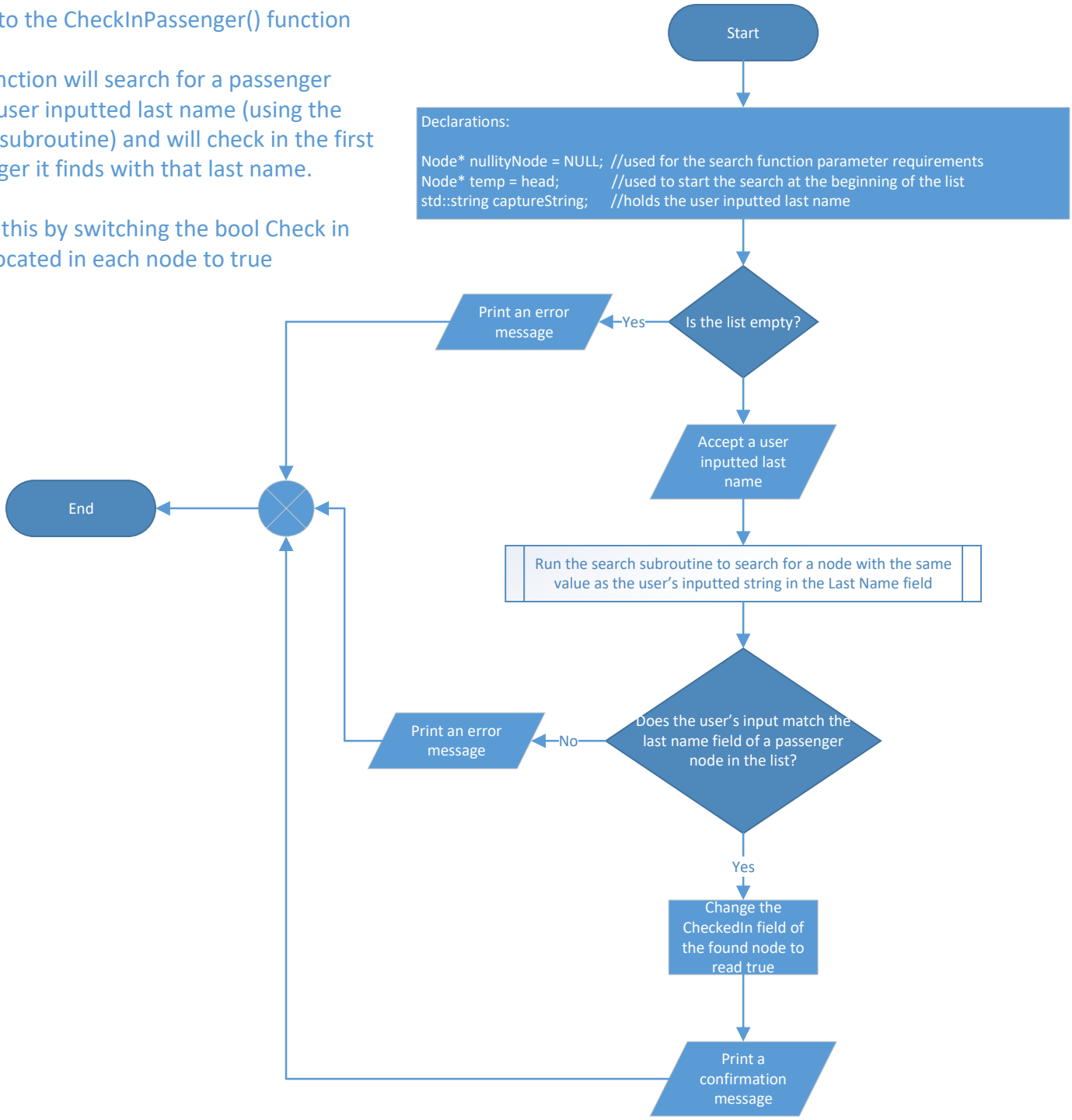


Check In Passenger

Refers to the CheckInPassenger() function

This function will search for a passenger with a user inputted last name (using the search subroutine) and will check in the first passenger it finds with that last name.

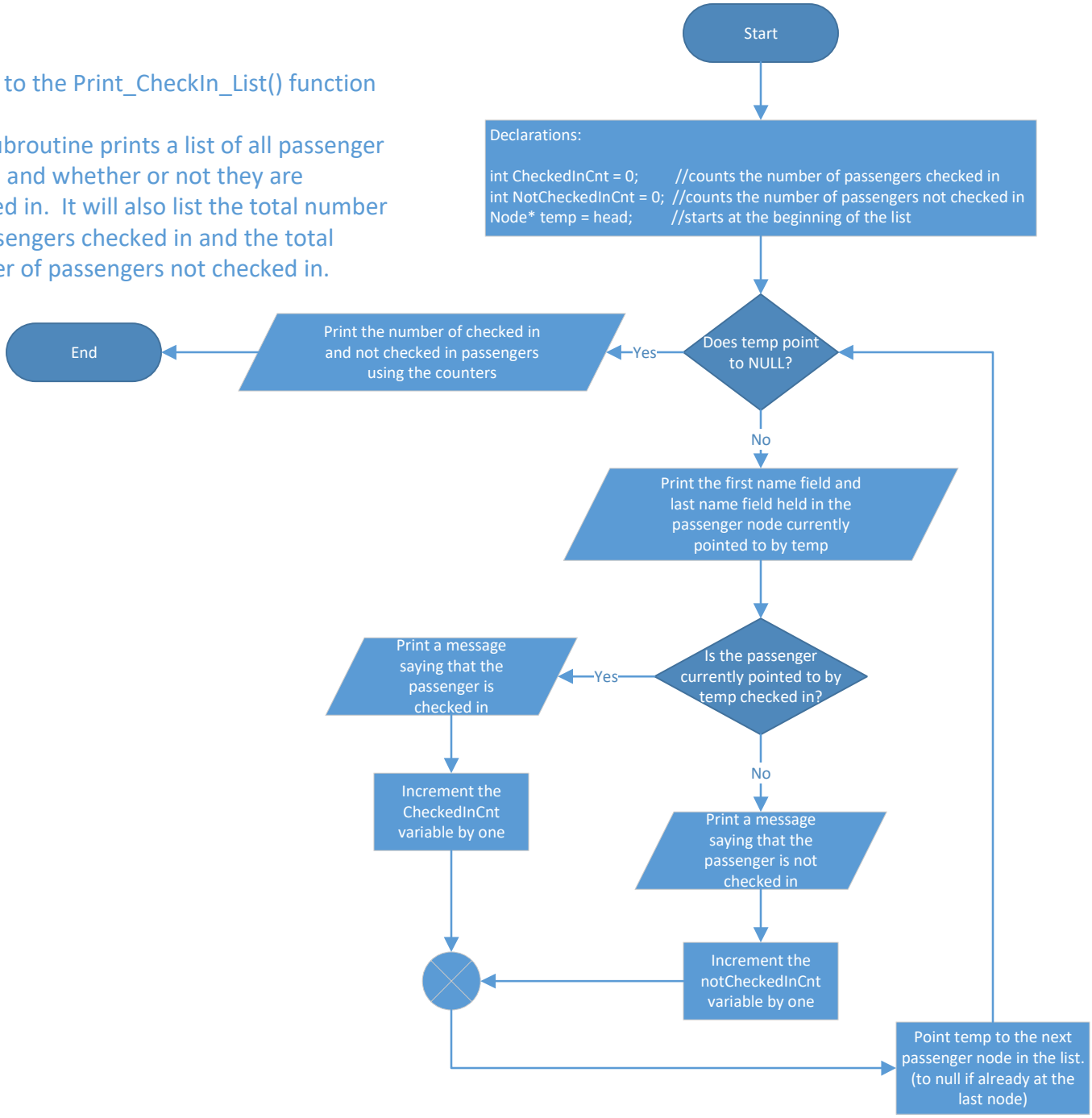
It does this by switching the bool Check in value located in each node to true



Print Check In Report

Refers to the Print_CheckIn_List() function

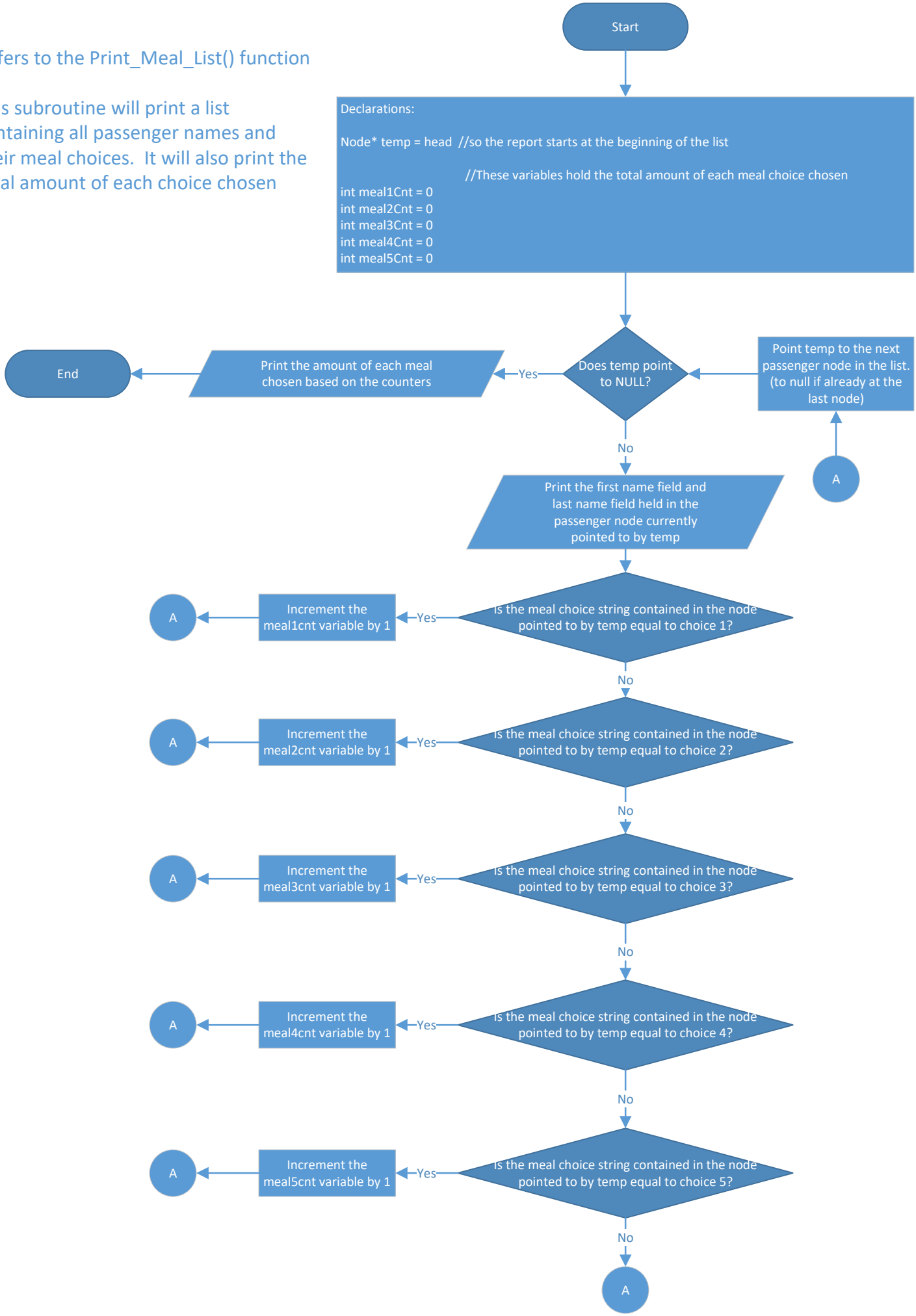
This subroutine prints a list of all passenger names and whether or not they are checked in. It will also list the total number of passengers checked in and the total number of passengers not checked in.



Print Menu Report

Refers to the Print_Meal_List() function

This subroutine will print a list containing all passenger names and their meal choices. It will also print the total amount of each choice chosen

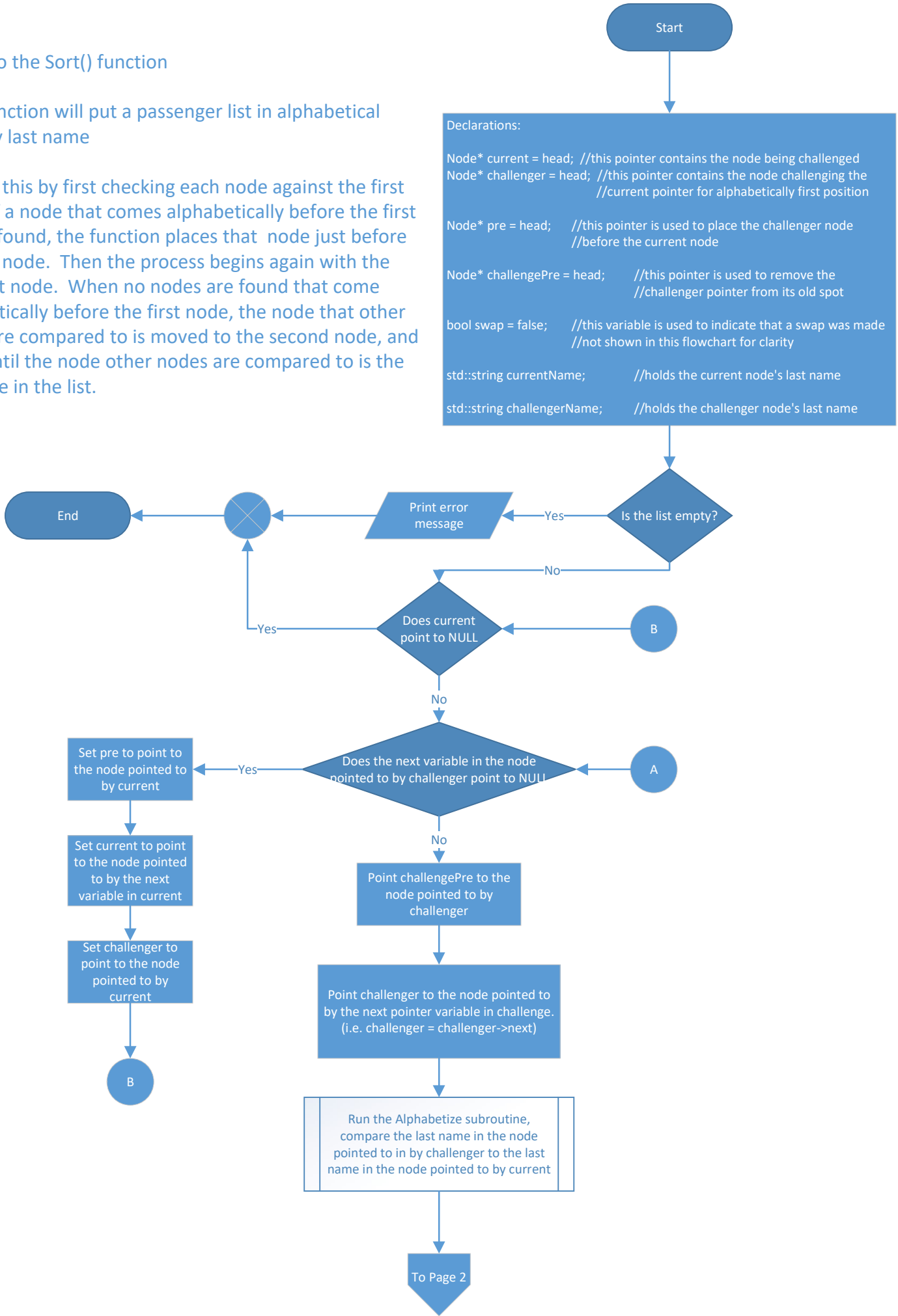


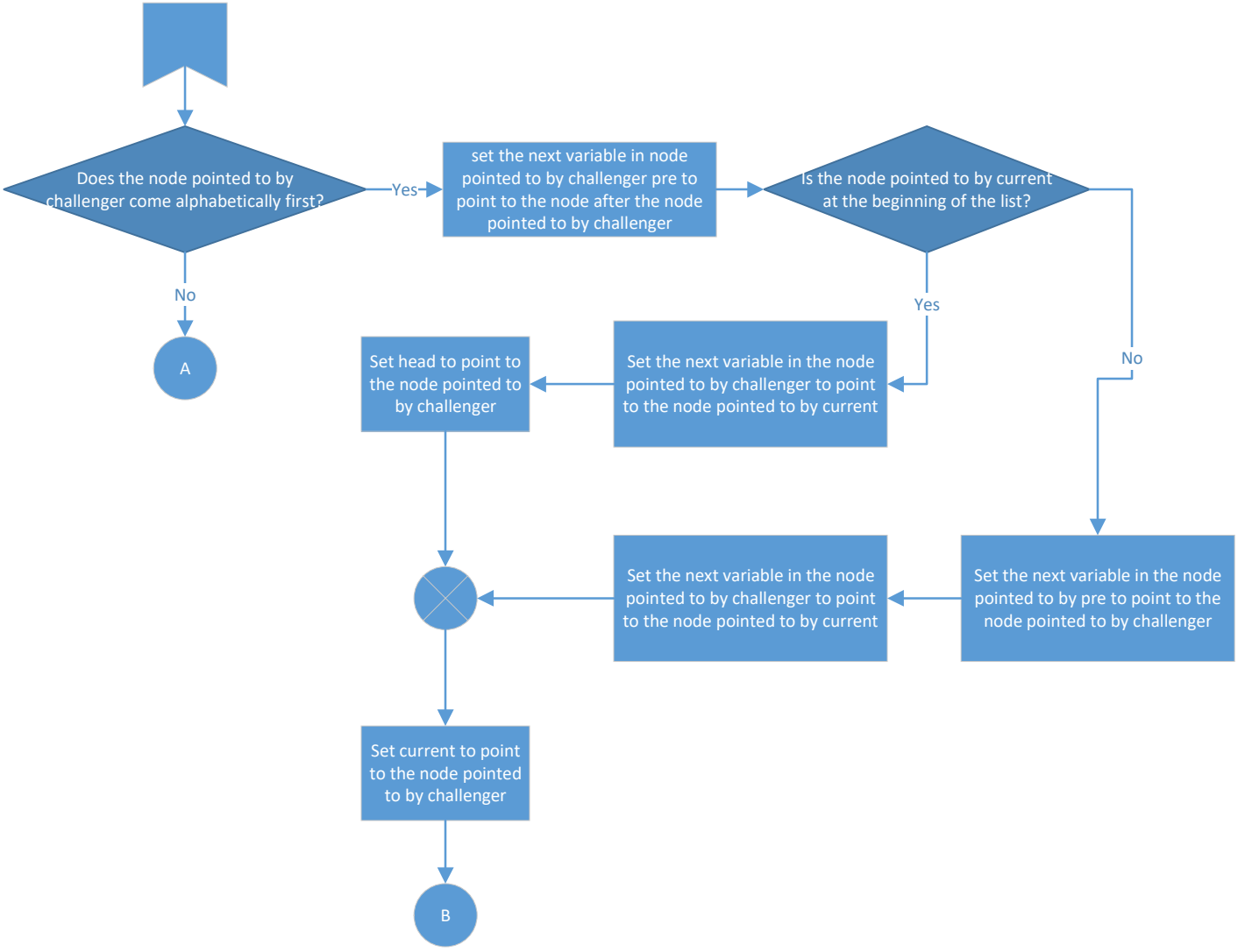
Sort

Refers to the Sort() function

This function will put a passenger list in alphabetical order by last name

It does this by first checking each node against the first node. if a node that comes alphabetically before the first node is found, the function places that node just before the first node. Then the process begins again with the new first node. When no nodes are found that come alphabetically before the first node, the node that other nodes are compared to is moved to the second node, and so on until the node other nodes are compared to is the last node in the list.




Does the node pointed to by challenger come alphabetically first?
Yes
No
A
set the next variable in node pointed to by challenger pre to point to the node after the node pointed to by challenger
Is the node pointed to by current at the beginning of the list?
Yes
No
Set the next variable in the node pointed to by challenger to point to the node pointed to by current
Set head to point to the node pointed to by challenger
Set the next variable in the node pointed to by pre to point to the node pointed to by challenger
Set the next variable in the node pointed to by challenger to point to the node pointed to by current
Set current to point to the node pointed to by challenger
B

Alphabetize

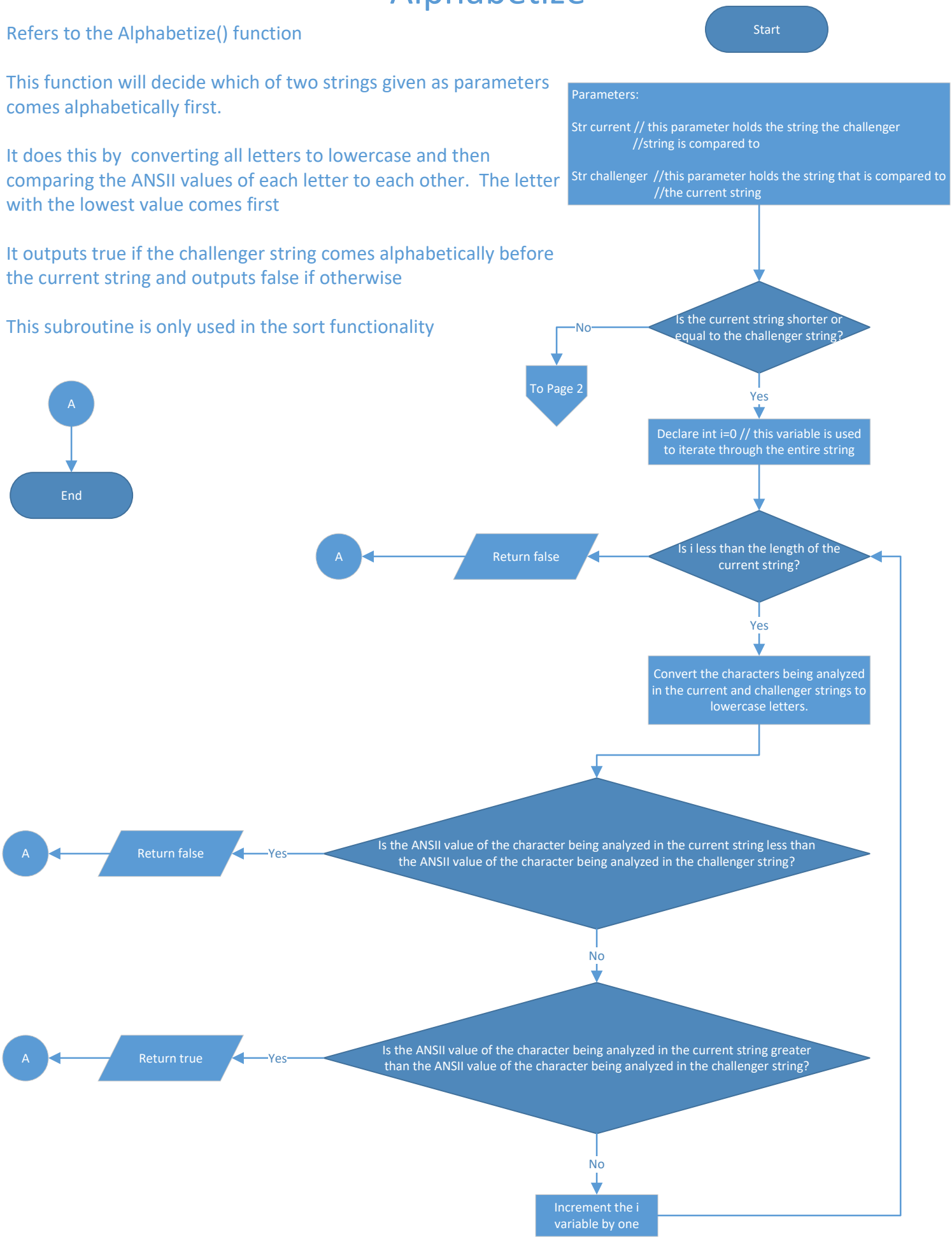
Refers to the Alphabetize() function

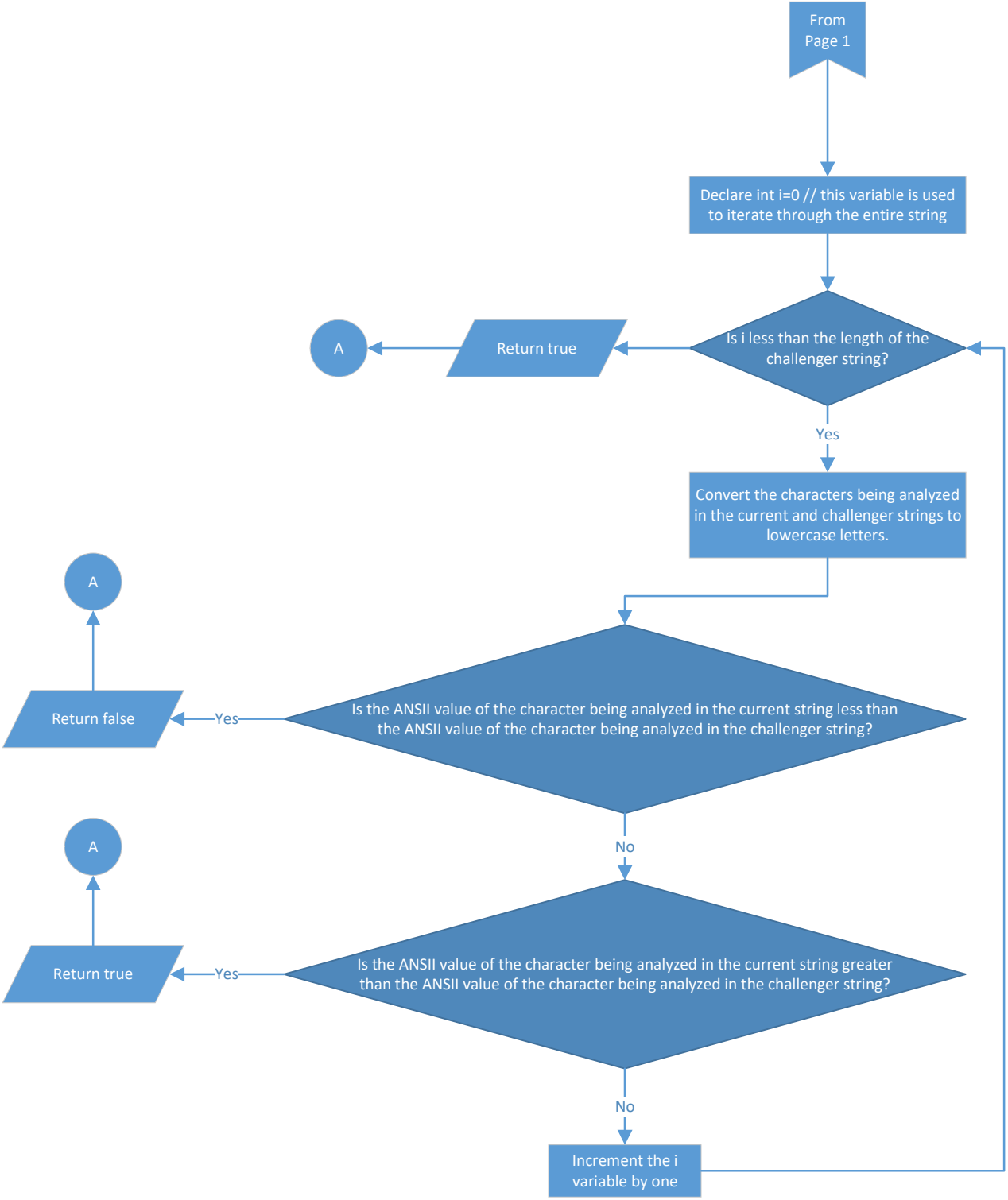
This function will decide which of two strings given as parameters comes alphabetically first.

It does this by converting all letters to lowercase and then comparing the ANSII values of each letter to each other. The letter with the lowest value comes first

It outputs true if the challenger string comes alphabetically before the current string and outputs false if otherwise

This subroutine is only used in the sort functionality




From Page 1
Declare int i=0 // this variable is used to iterate through the entire string
Is i less than the length of the challenger string?
Return true
A
Yes
Convert the characters being analyzed in the current and challenger strings to lowercase letters.
Is the ANSII value of the character being analyzed in the current string less than the ANSII value of the character being analyzed in the challenger string?
Yes
Return false
A
No
Is the ANSII value of the character being analyzed in the current string greater than the ANSII value of the character being analyzed in the challenger string?
Yes
Return true
A
No
Increment the i variable by one