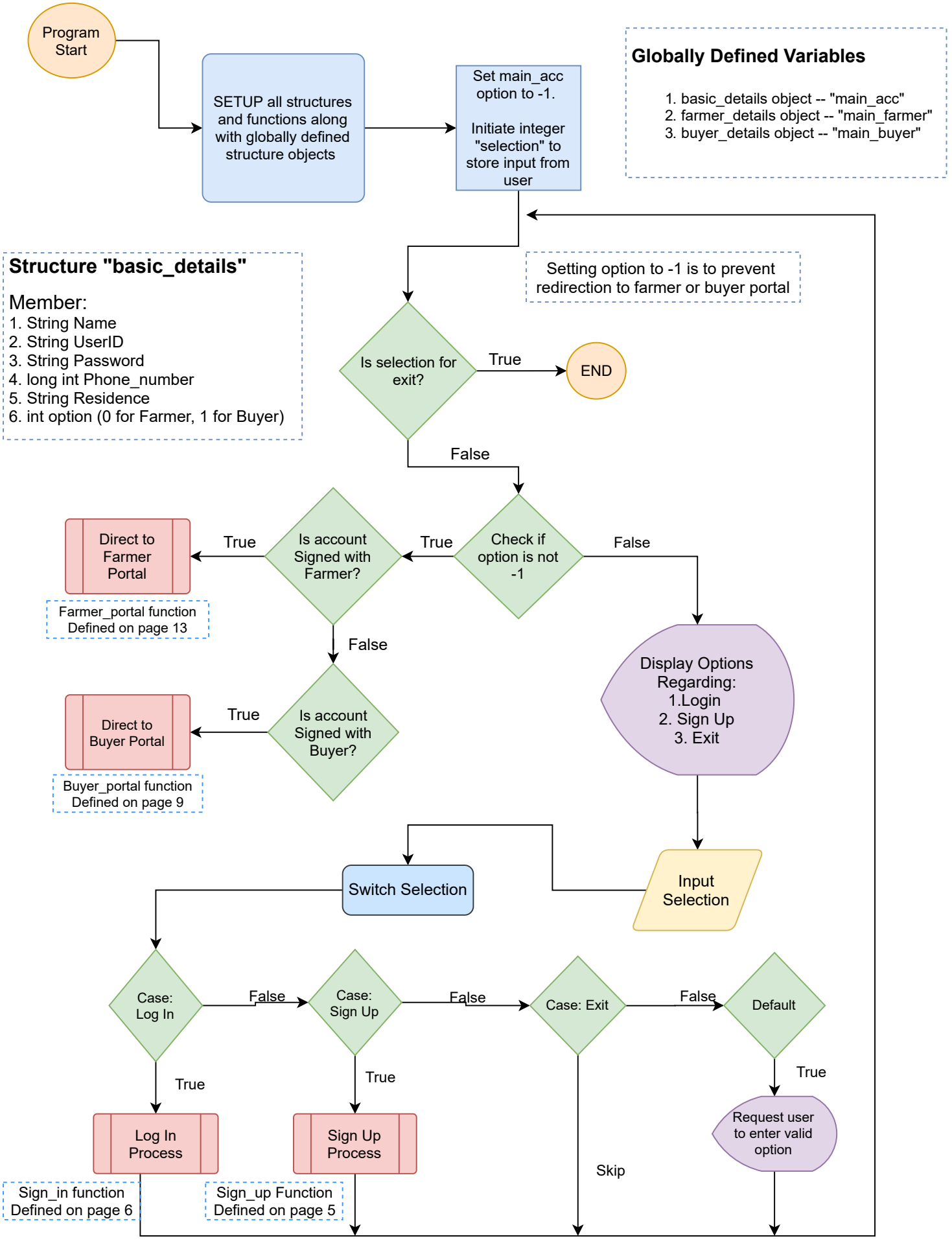
Function-Wise Flowchart



Sign_Up

SIGN UP FUNCTION

Used to Add a new account to the database

Variables used:

1. Structure basic_details object - acc
2. int selection (for farmer/buyer)
3. char input1 (for USERID check)

Sign Up Action

Initialize sample main_details structure object "acc"

Request main_details object variables from user:
1. Name
2. Phone number
3. Area of residence

False

USERID Exists

Request USERID for account

CHECK USERID EXISTS PROCESS

check_user_ID takes input userID as input and returns 1 and 0 depending of weather it can be used or not resp.

Check_user_function() Defined on page 8

True

Can be used

REQUEST PASSWORD

store UserID and password in "acc" object

All details are stored inside "acc" and are then put into "main_acc" and file "acc.dat"

Request Selection: Farmer or buyer
(0 for Farmer, 1 for Buyer)

Switch Selection

Case: Farmer

False

Case: Buyer

True

True

create_farmer_details function() Defined on page 7

Create - farmer - details action

Setting new Acc to active ID

Copy "acc" into "main_acc"

Create - buyer - details action

create_buyer_details function() Defined on page 8

acc.dat

Write into File

Write "acc" object into "acc.dat"

Open "acc.dat"

"acc.dat" stores basic_details structure objects

Close "acc.dat"

Close Function

Sign In function

Used to Log-In into a new acc

Sign_In

struct farmer_details

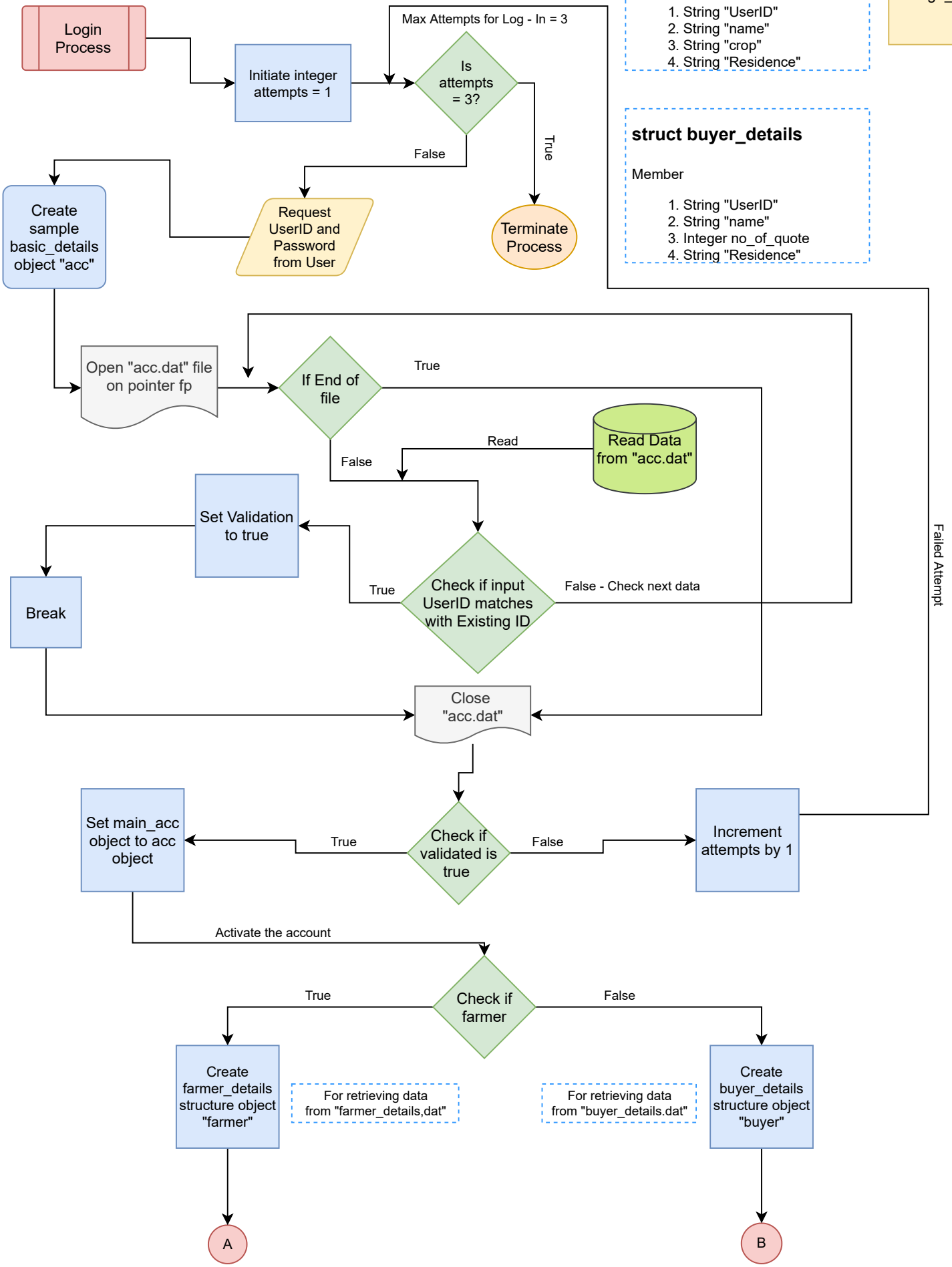
Member

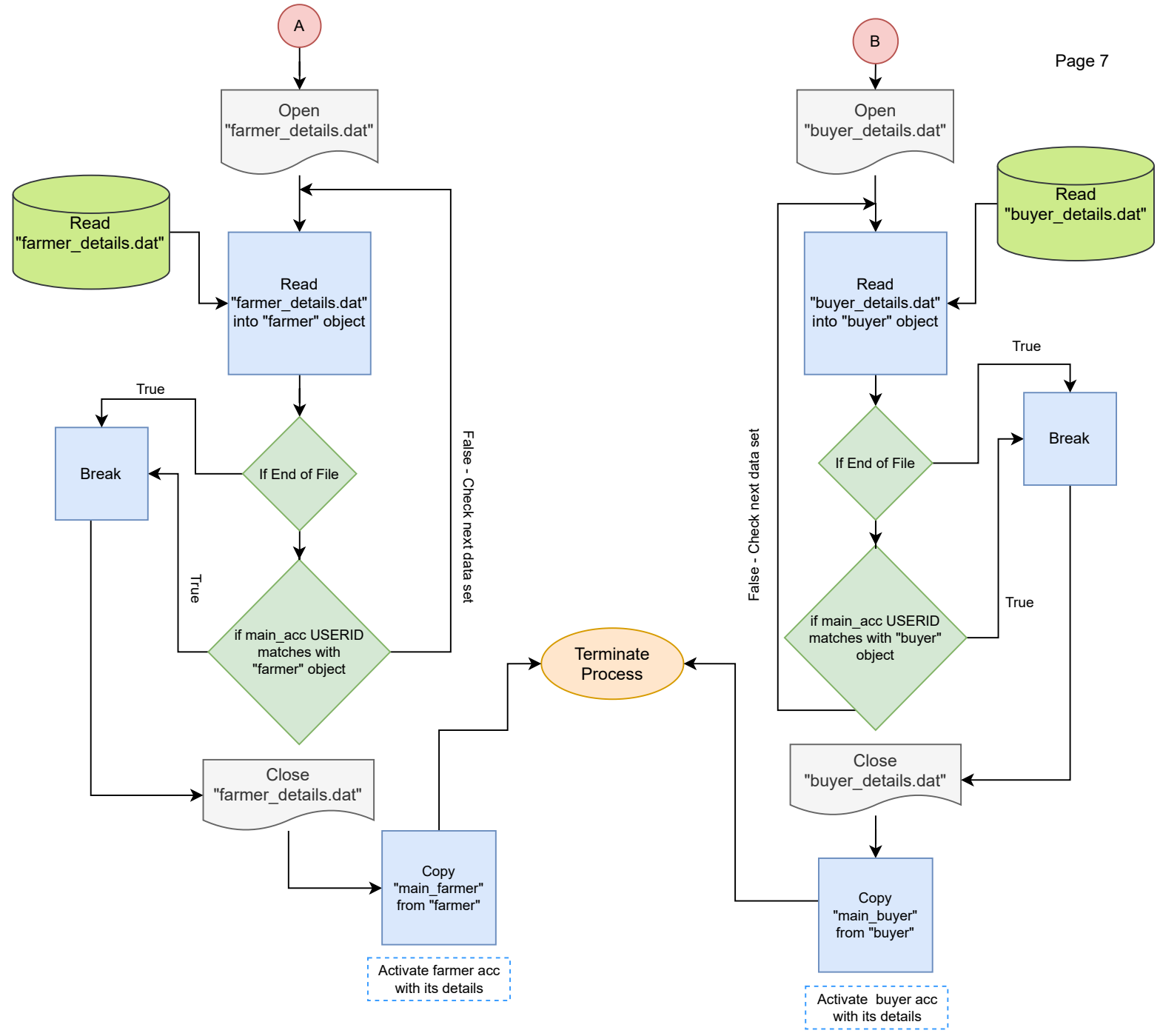
1. String "UserID"
2. String "name"
3. String "crop"
4. String "Residence"

struct buyer_details

Member

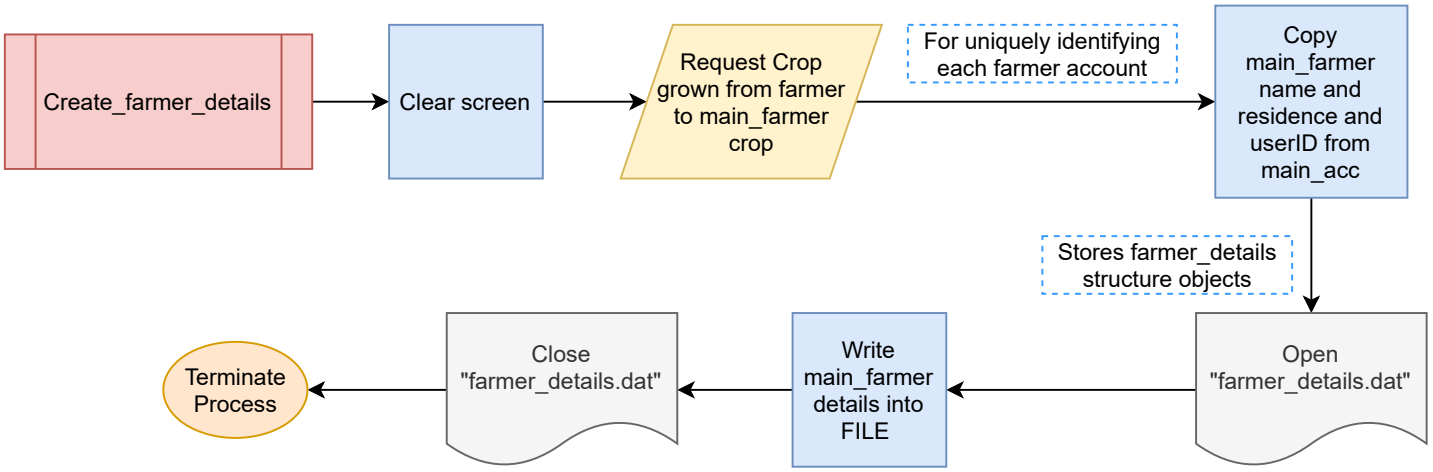
1. String "UserID"
2. String "name"
3. Integer no_of_quote
4. String "Residence"





Create_farmer_details

Creates farmer details entry in file and sets to active account

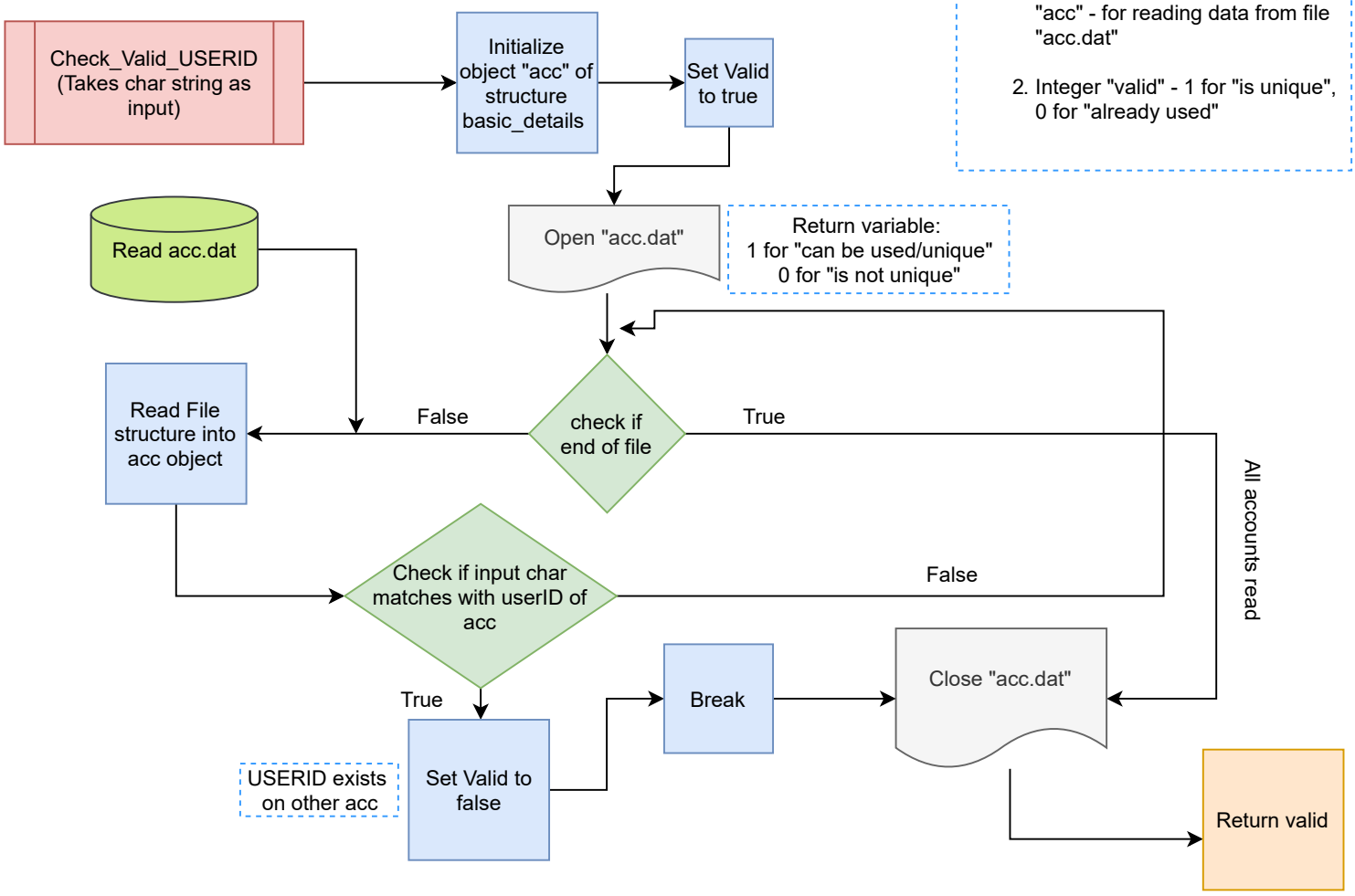


check_valid_userID

Used to check if the string the function is called is unique compared to other IDs

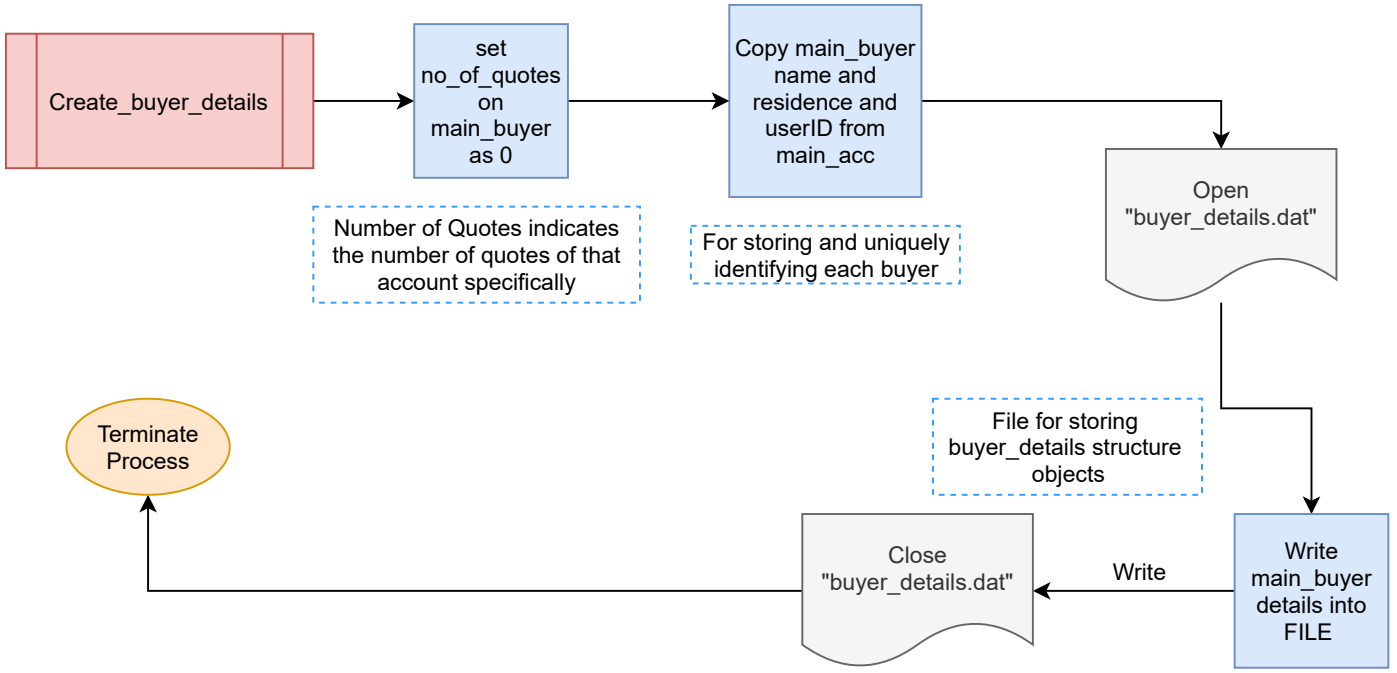
Variables used

1. Basic_details structure object "acc" - for reading data from file "acc.dat"
2. Integer "valid" - 1 for "is unique", 0 for "already used"



create_buyer_details

Creates new buyer account and makes it active.



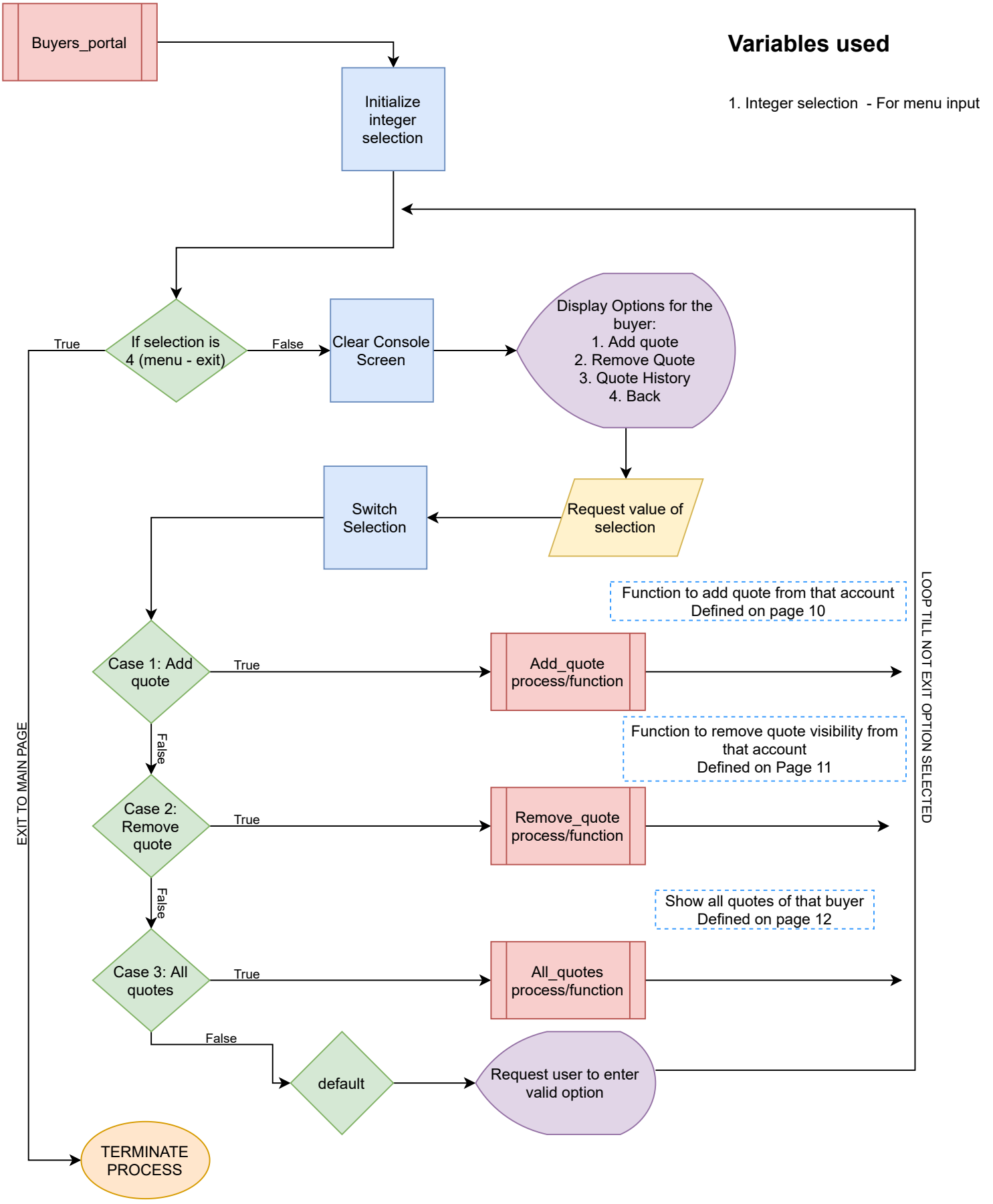
Buyer Portal

Displays Options for user if Buyer

Buyer Portal

Variables used

1. Integer selection - For menu input



Add_quote()

Add Quote entry to database

Add Quote

Add_quote

Since new quote is added to that account

For reading quotes from "quote.dat"

Increment no, of quotes in main_buyer by 1

Create sample object "quote" of structure buyer_quotes

Structure buyer_quotes

Members:

1. integer Valid - if valid then visible to farmers
2. String UserID
3. integer Quote_no
4. String Crop
5. float price
6. String Residence

Console Clear Screen

Request crop and price details of the quote from buyer

Input the details into "quote". Set "quote" valid to true. Take other details from main_buyer object.

Set quote ID as no. of quotes from main_buyer

File that stores all objects of buyer_quotes structure

Open "quote.dat" in append

Put data into "quotes.dat"

Close "quote.dat"

Setting vaid to true (1) makes the quote appear to farmers upon search

Read "buyer_details.dat" into "buyer"

To change the content in "buyer_details.dat" since we changed the value of number of quotes of the particular account

Create object of buyer_details "buyer"

Open "buyer_details.dat" in read and "temp.dat" in write

While end of file of "buyer_details.dat"

True

"temp.dat" will contain the modified new content

if UserID of buyer = UserID of main_buyer

True

write main_buyer object into "temp.dat"

Writes main_buyer because it contains the updated details of the account

False

write "buyer" into "temp.dat"

Since no changes are made onother objects, they are just transfered directly

Loop till all objects not transfered

Exit from loop

Close "quote.dat" and "temp.dat"

Remove "quotes.dat"

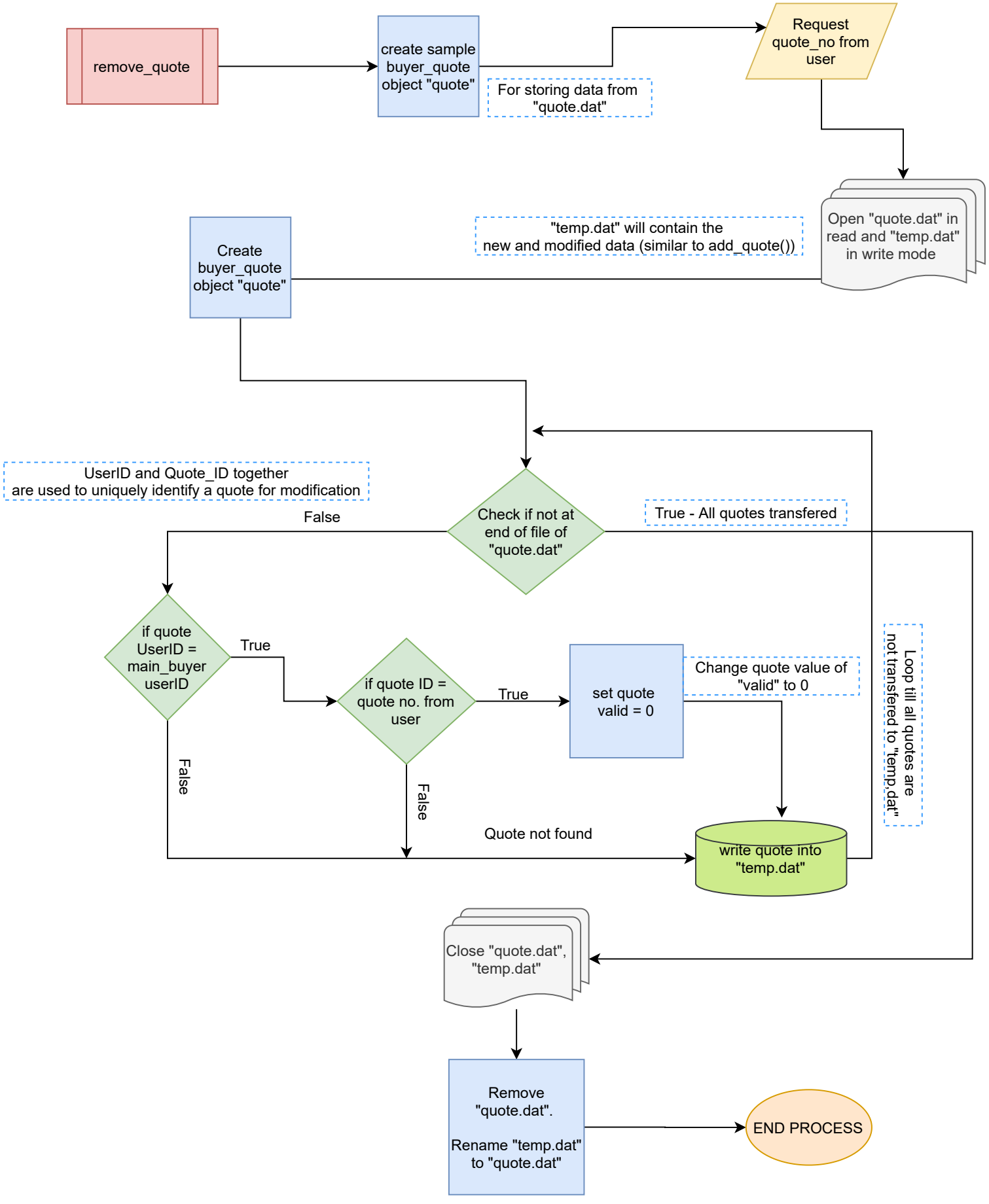
Rename "temp.dat" to "quotes.dat"

TERMINATE

Remove Quote

Remove_quote()

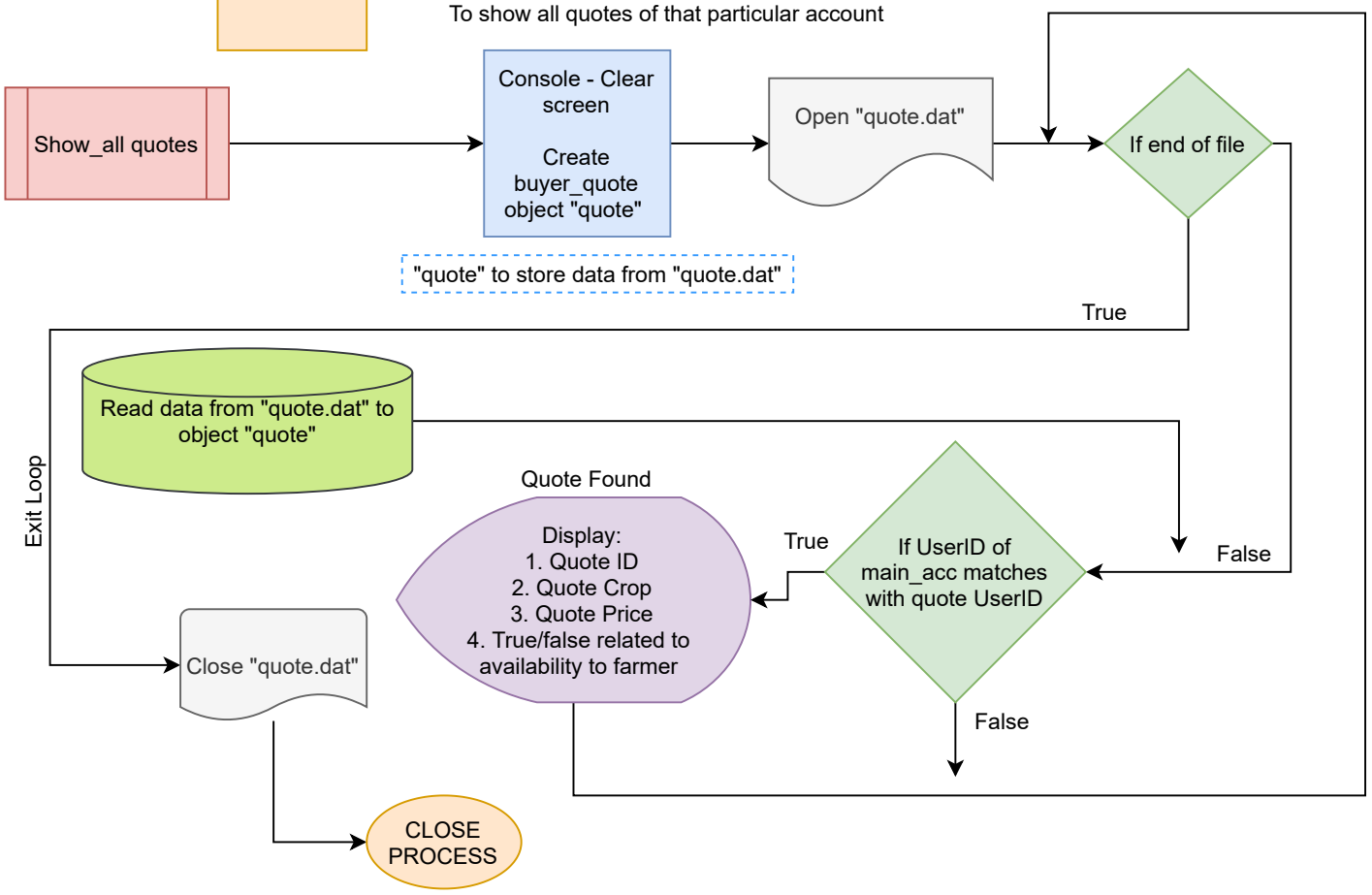
Removes quote availability for farmers (makes valid to false)



Quote History

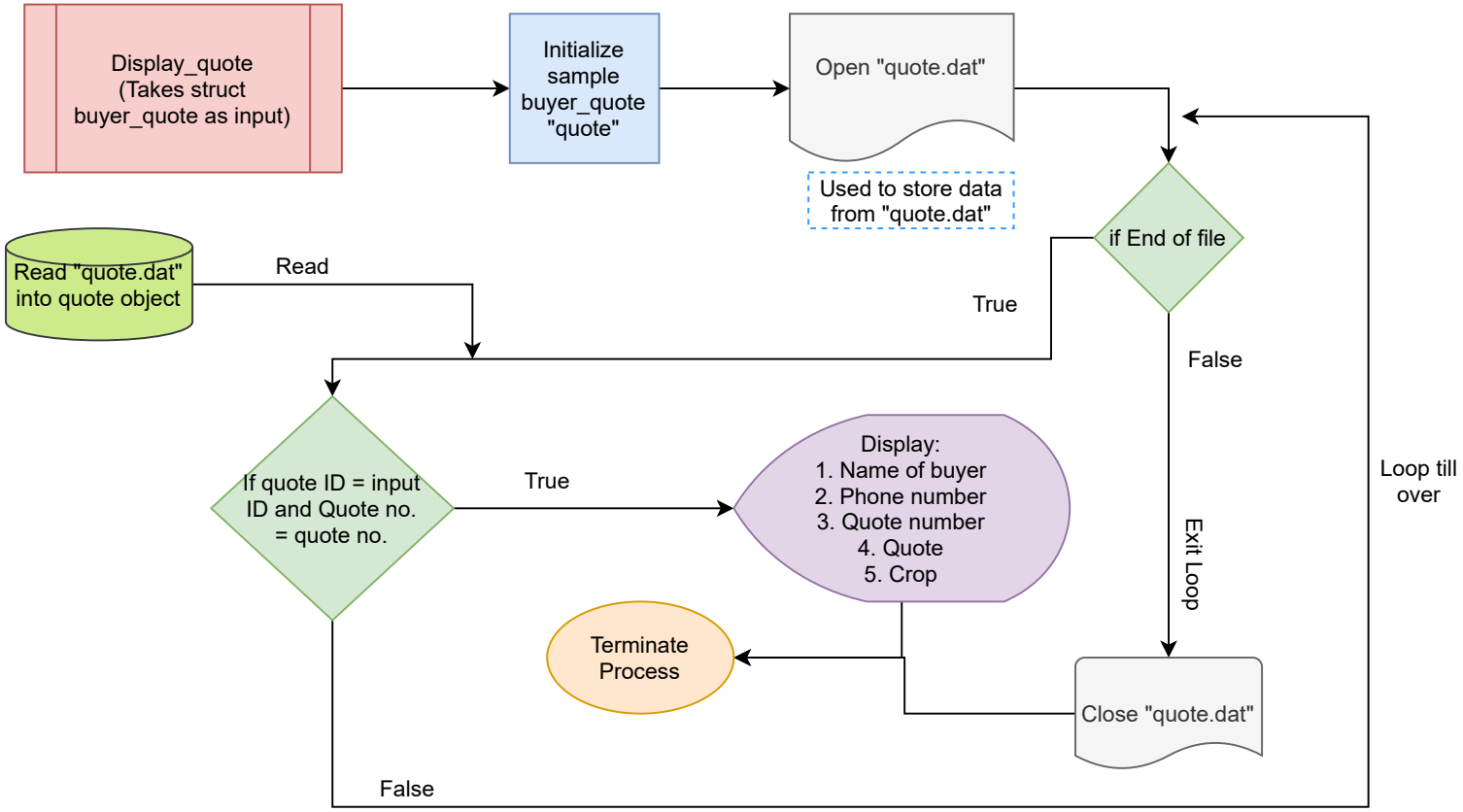
Quote History for Buyer

To show all quotes of that particular account



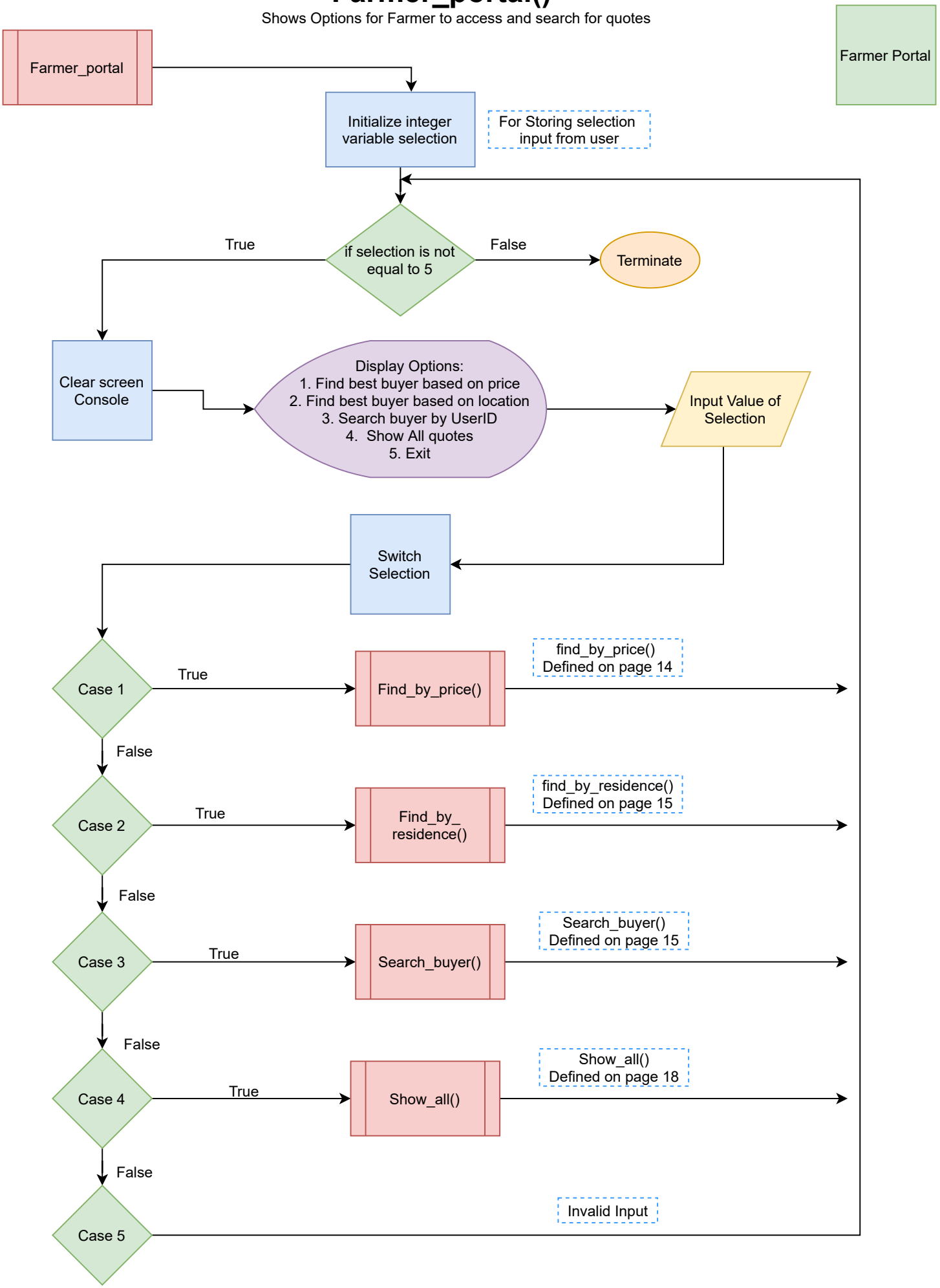
Display_quote()

Takes in structure buyer_quotes object as input and displays its corresponding details



Farmer_portal()

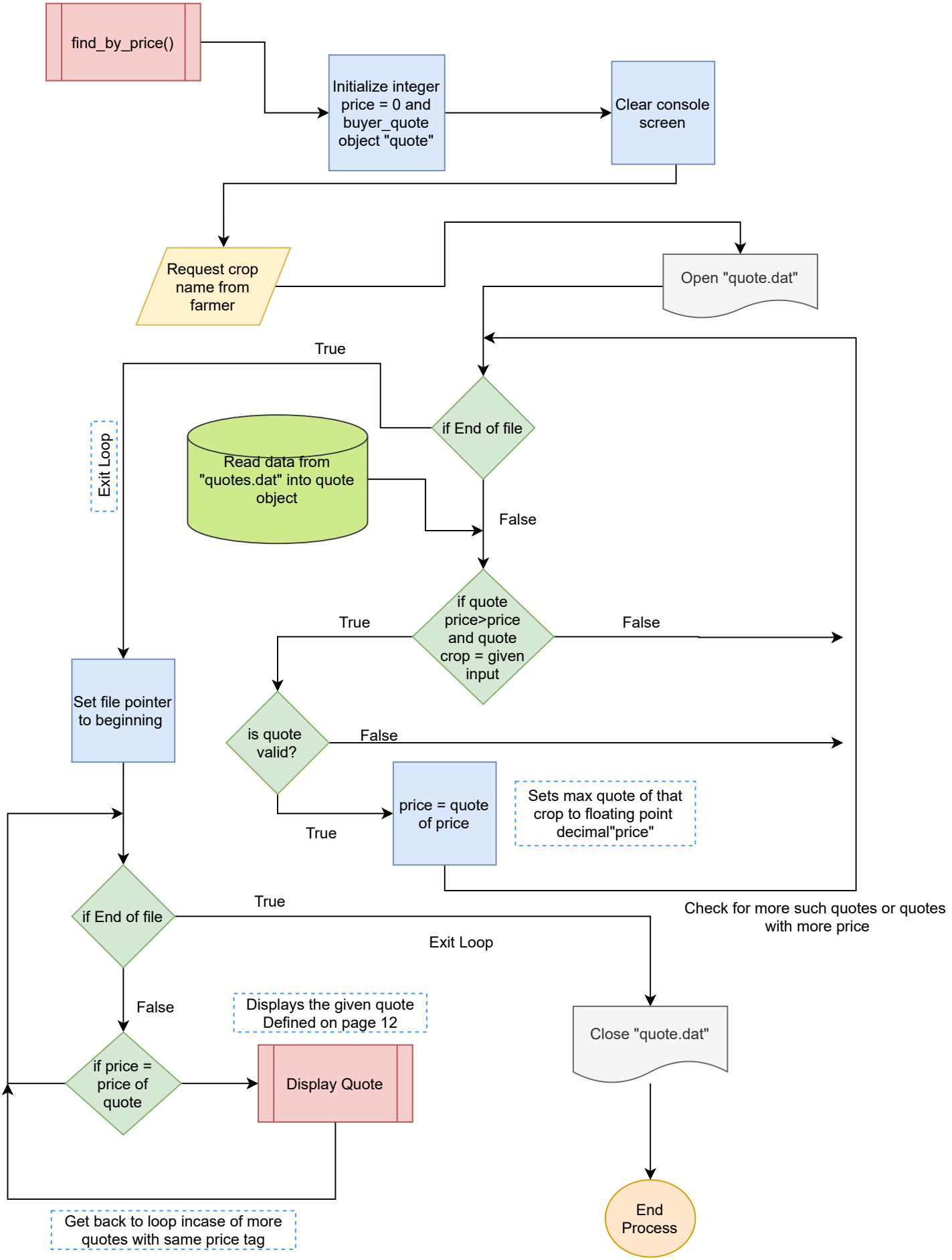
Shows Options for Farmer to access and search for quotes



Find_by_price

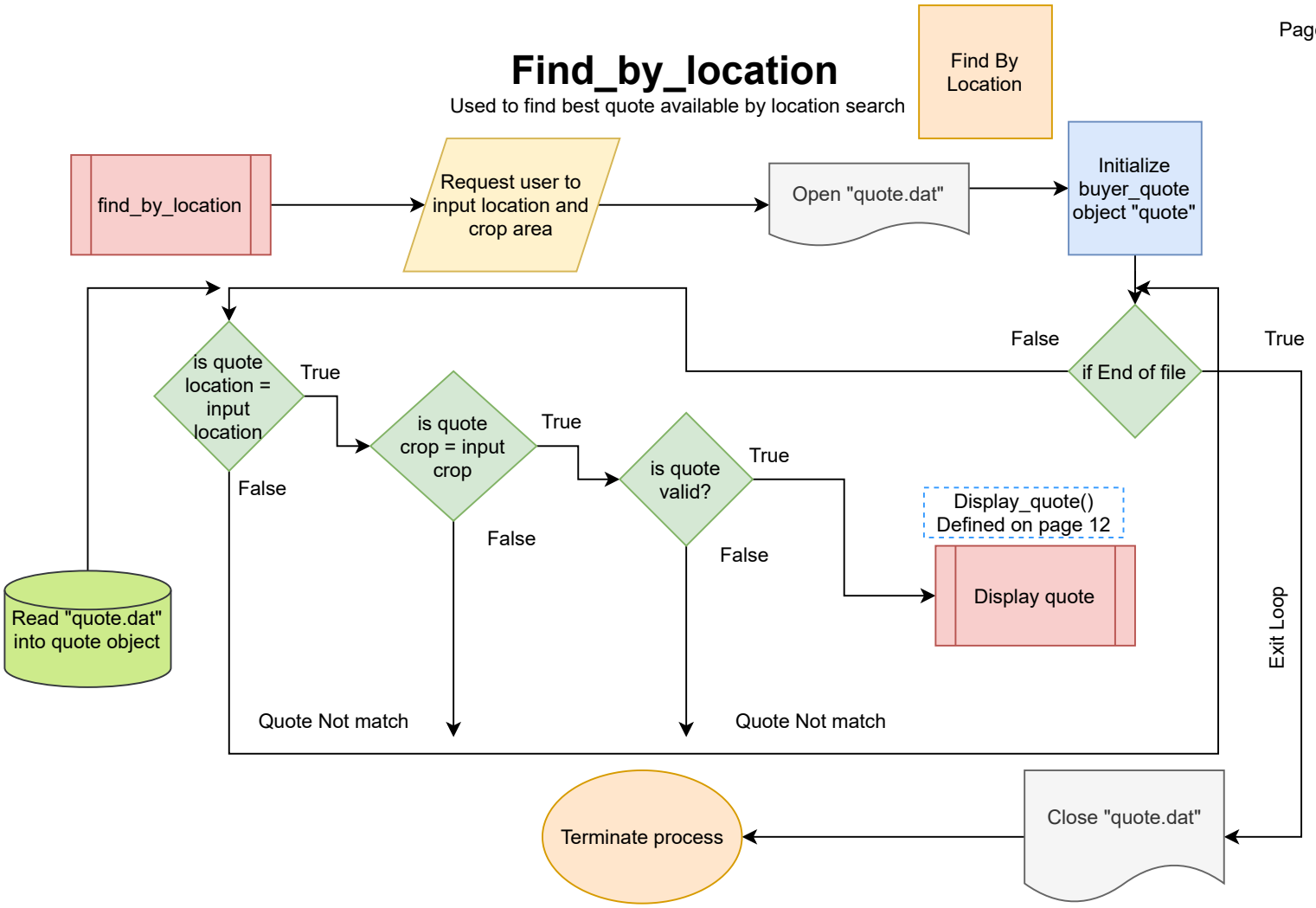
Finds best quote by price for a buyer depending on a given crop

Find by Price



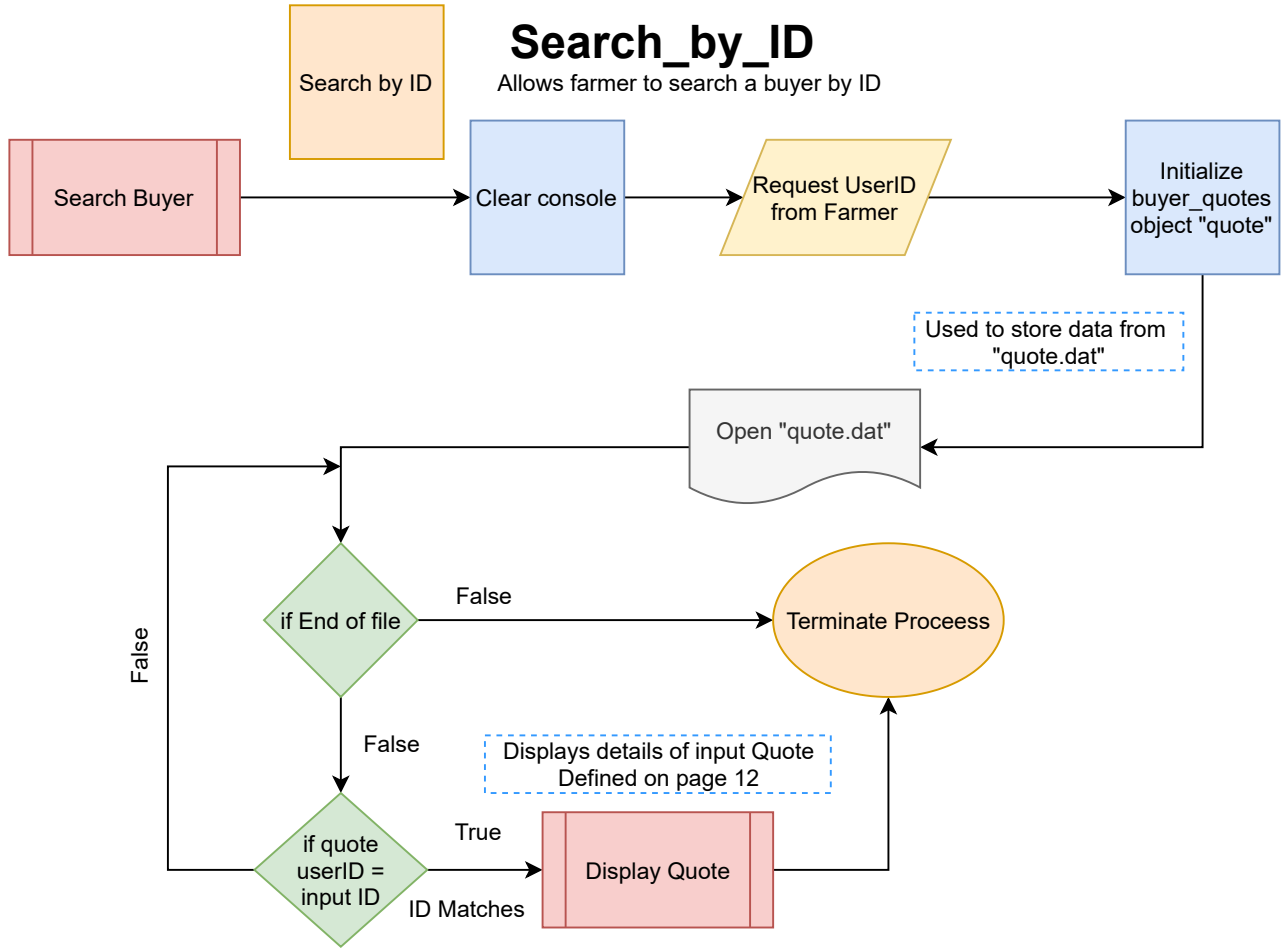
Find_by_location

Used to find best quote available by location search

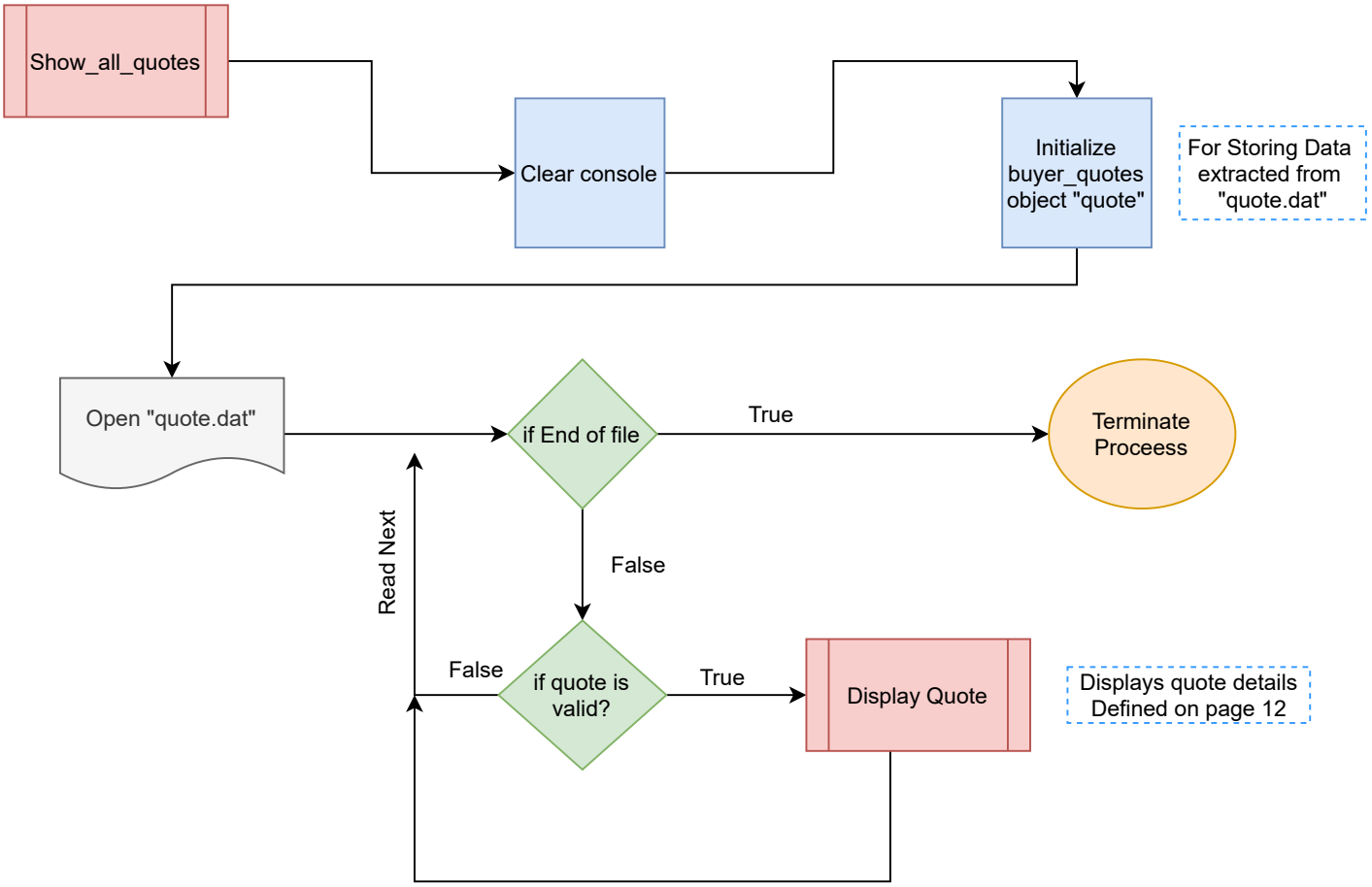


Search_by_ID

Allows farmer to search a buyer by ID



Show_all_quote

Shows all quotes that is accessible to farmers (Quote.valid is 1)

Show All Quotes