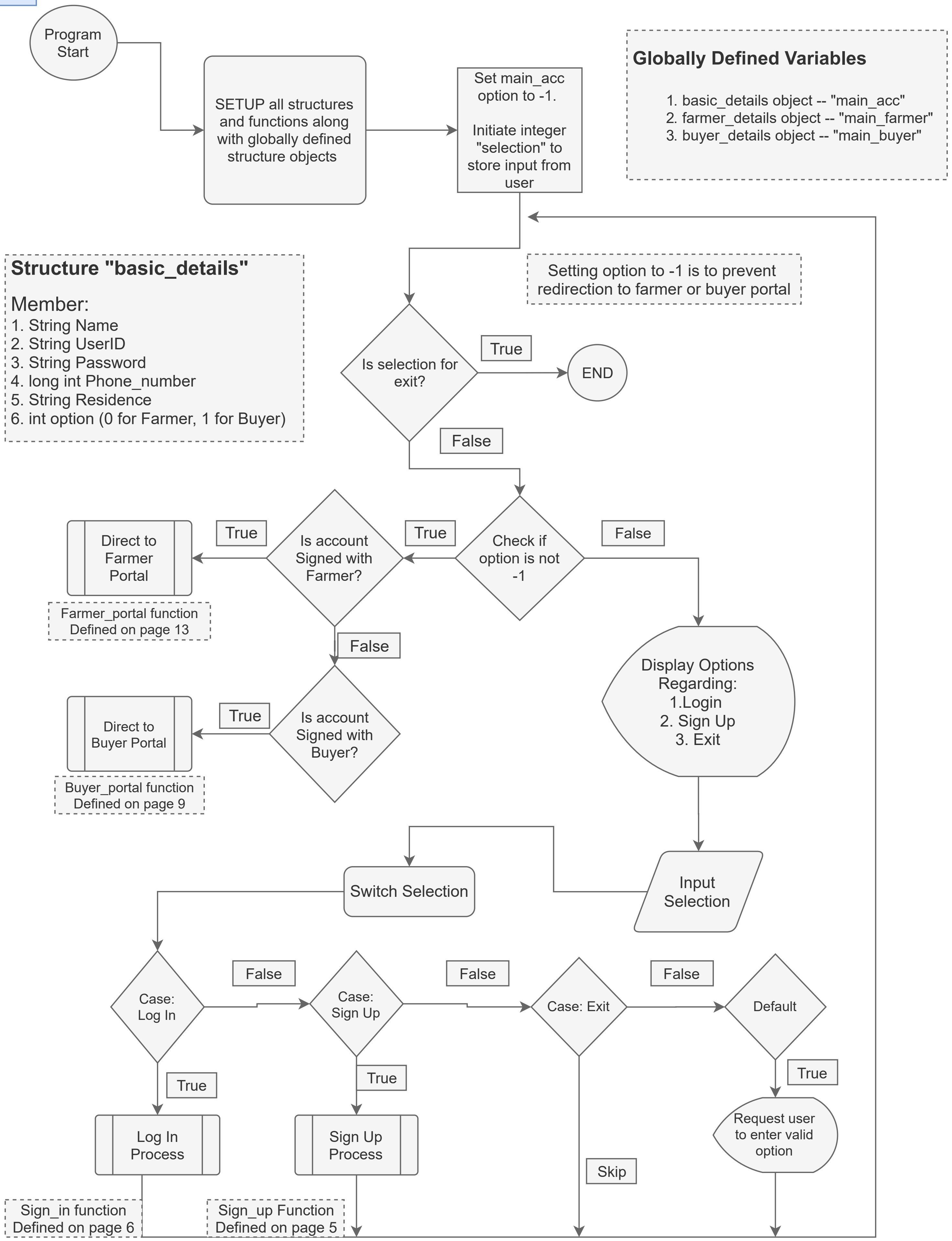
Main Screen

Function-Wise Flowchart


Program Start
SETUP all structures and functions along with globally defined structure objects
Set main_acc option to -1.
Initiate integer "selection" to store input from user
Setting option to -1 is to prevent redirection to farmer or buyer portal
Is selection for exit?
True
END
False
Check if option is not -1
True
False
Is account Signed with Farmer?
True
Direct to Farmer Portal
Farmer_portal function Defined on page 13
False
Is account Signed with Buyer?
True
Direct to Buyer Portal
Buyer_portal function Defined on page 9
Display Options Regarding:
1.Login
2. Sign Up
3. Exit
Input Selection
Switch Selection
Case: Log In
False
Case: Sign Up
False
Case: Exit
False
Default
True
Log In Process
Sign_in function Defined on page 6
True
Sign Up Process
Sign_up Function Defined on page 5
Skip
True
Request user to enter valid option

Globally Defined Variables

1. basic_details object -- "main_acc"
2. farmer_details object -- "main_farmer"
3. buyer_details object -- "main_buyer"

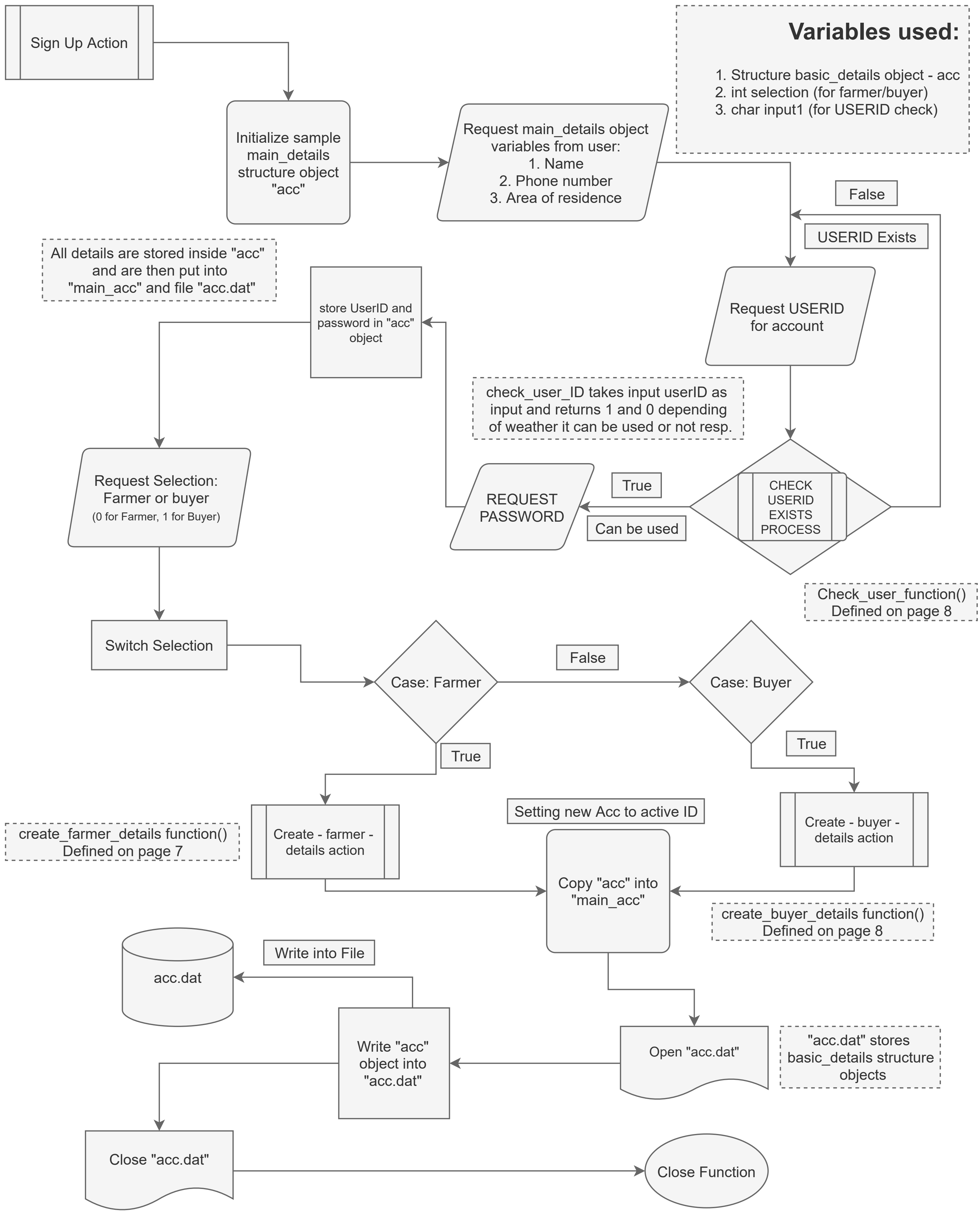
Structure "basic_details"

Member:
1. String Name
2. String UserID
3. String Password
4. long int Phone_number
5. String Residence
6. int option (0 for Farmer, 1 for Buyer)

Sign_Up

SIGN UP FUNCTION

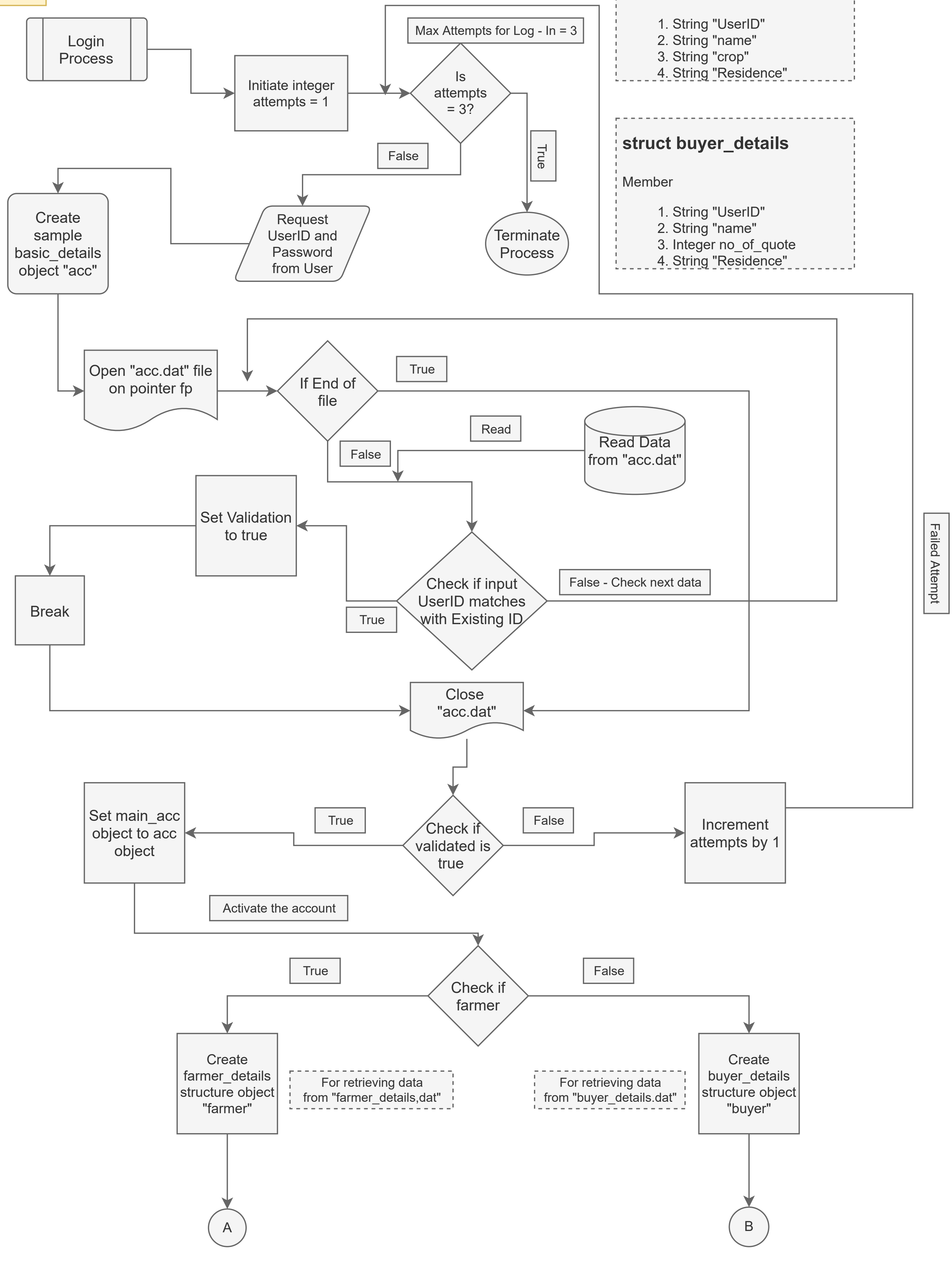
Used to Add a new account to the database


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
Variables used:
1. Structure basic_details object - acc
2. int selection (for farmer/buyer)
3. char input1 (for USERID check)
False
USERID Exists
Request USERID for account
All details are stored inside "acc" and are then put into "main_acc" and file "acc.dat"
store UserID and password in "acc" object
check_user_ID takes input userID as input and returns 1 and 0 depending of weather it can be used or not resp.
CHECK USERID EXISTS PROCESS
True
Can be used
REQUEST PASSWORD
Check_user_function()
Defined on page 8
Request Selection: Farmer or buyer
(0 for Farmer, 1 for Buyer)
Switch Selection
Case: Farmer
False
Case: Buyer
True
True
Setting new Acc to active ID
create_farmer_details function()
Defined on page 7
Create - farmer - details action
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

Sign In function

Used to Log-In into a new acc


Login Process
Initiate integer attempts = 1
Max Attempts for Log - In = 3
Is attempts = 3?
True
Terminate Process
False
Request UserID and Password from User
Create sample basic_details object "acc"
Open "acc.dat" file on pointer fp
If End of file
True
False
Read
Read Data from "acc.dat"
Check if input UserID matches with Existing ID
False - Check next data
True
Set Validation to true
Break
Close "acc.dat"
Check if validated is true
True
False
Set main_acc object to acc object
Increment attempts by 1
Failed Attempt
Activate the account
Check if farmer
True
False
Create farmer_details structure object "farmer"
For retrieving data from "farmer_details,dat"
For retrieving data from "buyer_details.dat"
Create buyer_details structure object "buyer"
A
B

struct farmer_details

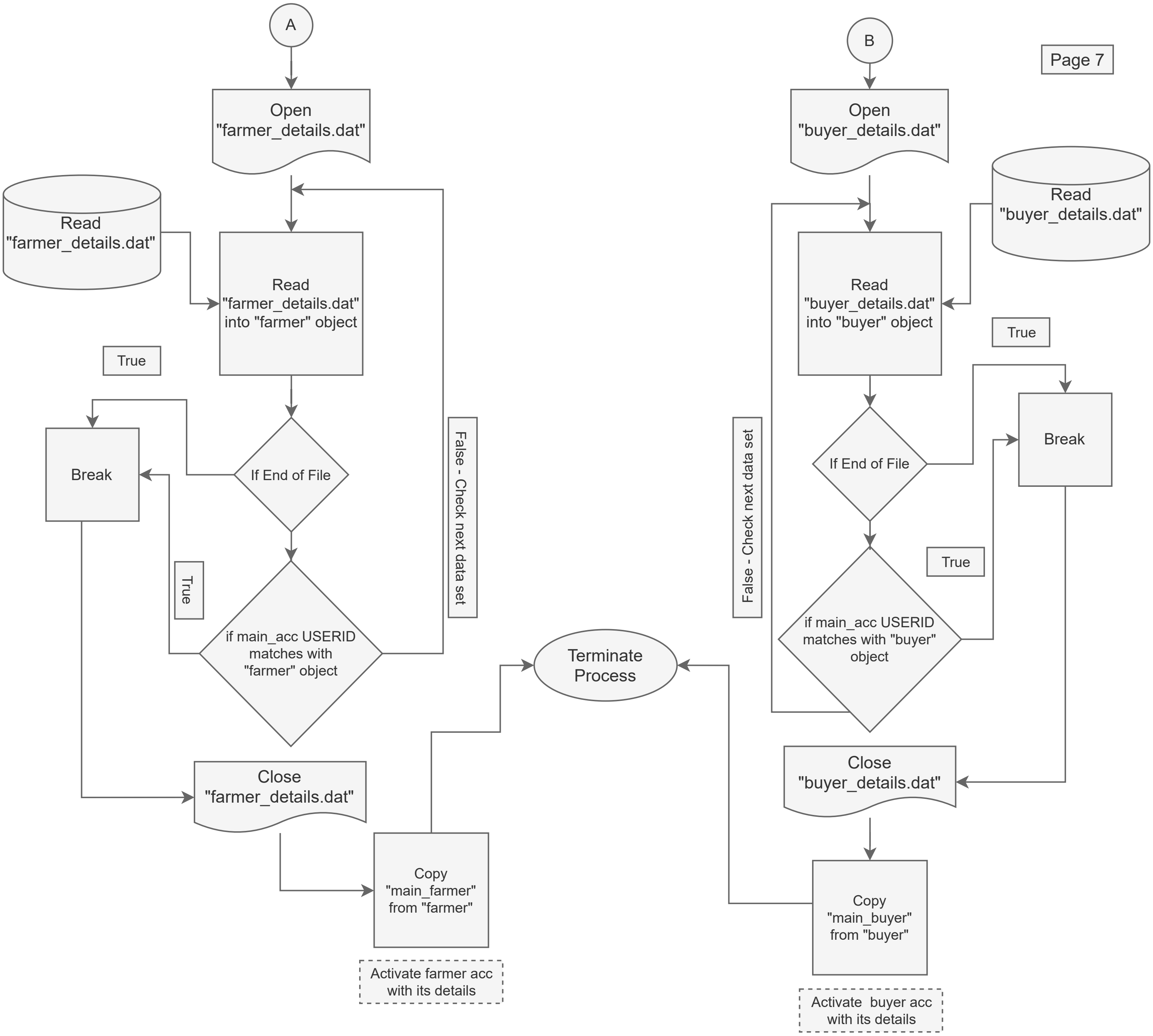
Member

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

struct buyer_details

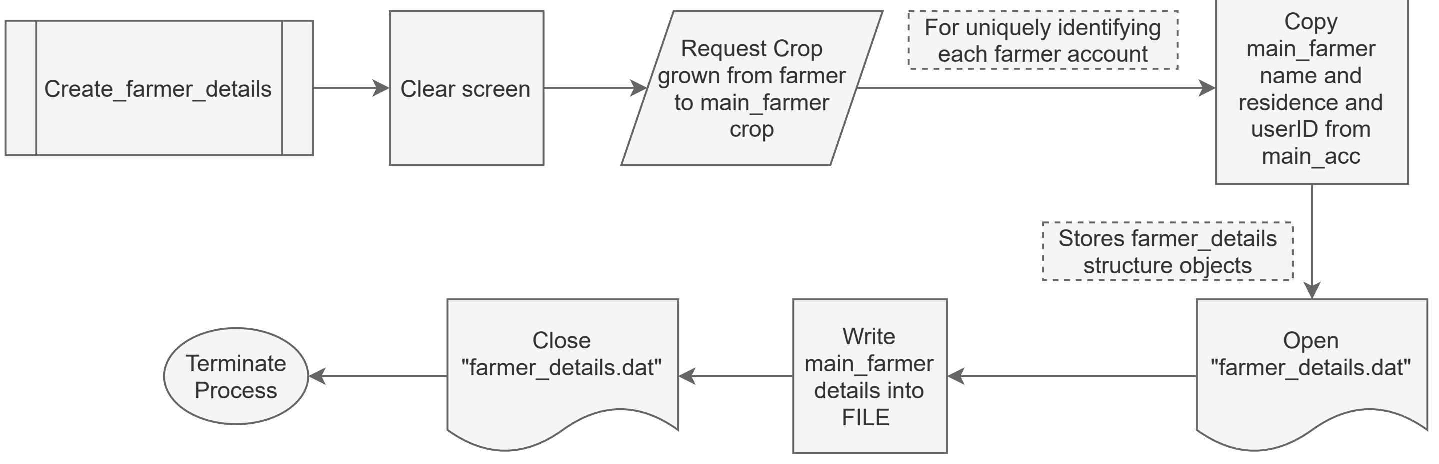
Member

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


A
Open "farmer_details.dat"
Read "farmer_details.dat"
Read "farmer_details.dat" into "farmer" object
True
Break
If End of File
True
if main_acc USERID matches with "farmer" object
False - Check next data set
Close "farmer_details.dat"
Copy "main_farmer" from "farmer"
Activate farmer acc with its details
Terminate Process
B
Open "buyer_details.dat"
Read "buyer_details.dat"
Read "buyer_details.dat" into "buyer" object
True
Break
If End of File
True
if main_acc USERID matches with "buyer" object
False - Check next data set
Close "buyer_details.dat"
Copy "main_buyer" from "buyer"
Activate buyer acc with its details

Create_farmer_details

Creates farmer details entry in file and sets to active account

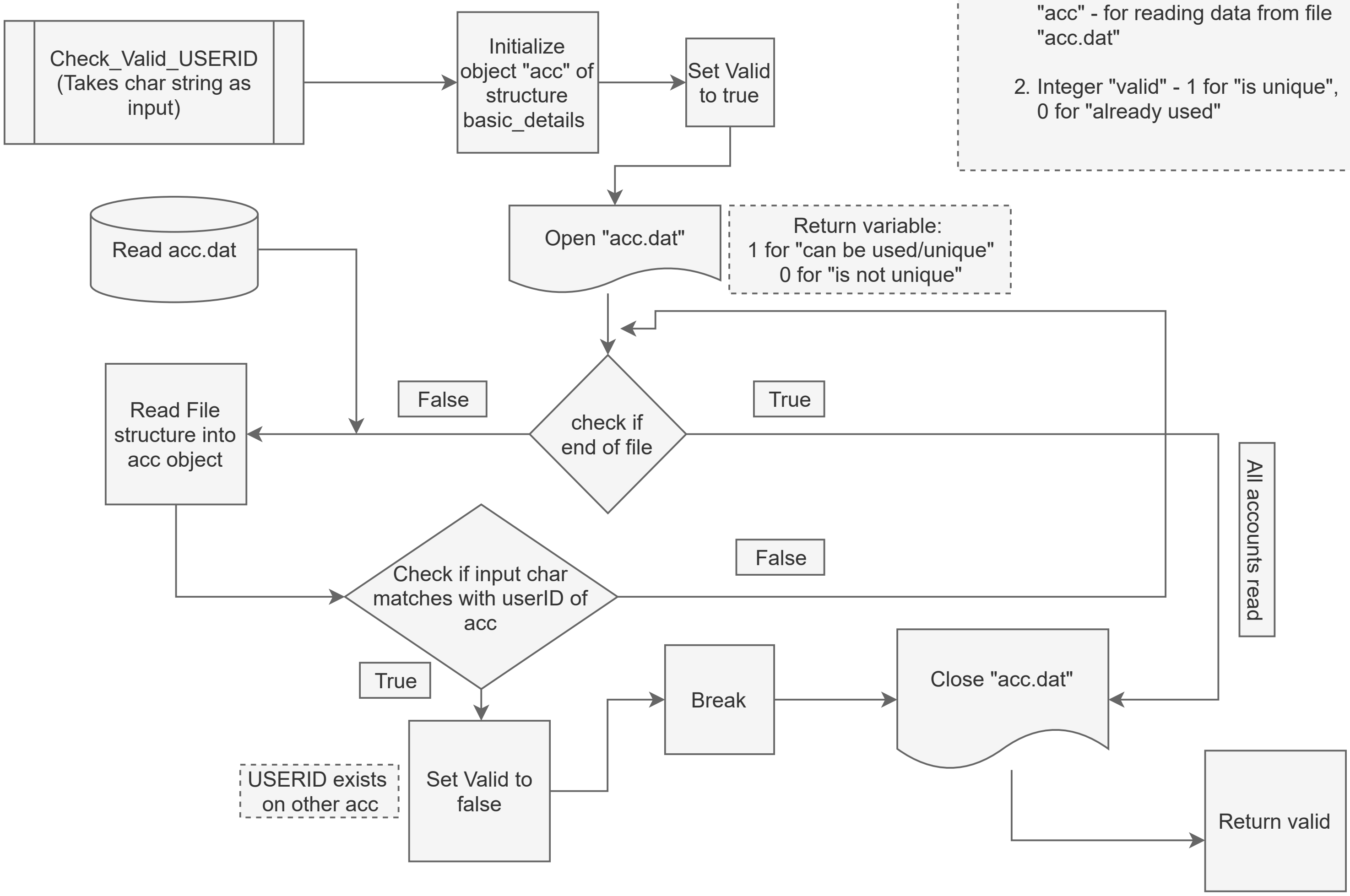

Create_farmer_details
Clear screen
Request Crop grown from farmer to main_farmer crop
For uniquely identifying each farmer account
Copy main_farmer name and residence and userID from main_acc
Stores farmer_details structure objects
Open "farmer_details.dat"
Write main_farmer details into FILE
Close "farmer_details.dat"
Terminate Process

check_valid_userID

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

Variables used

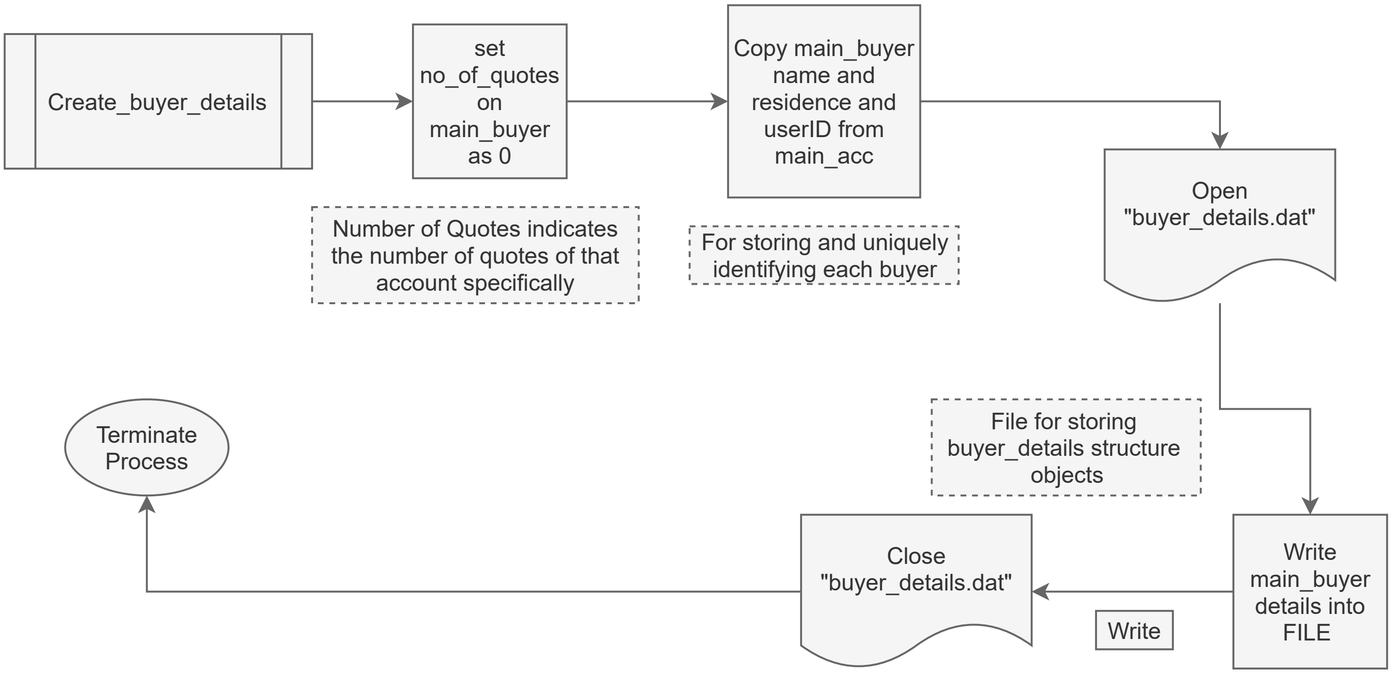
1. Basic_details structure object "acc" - for reading data from file "acc.dat"
2. Integer "valid" - 1 for "is unique", 0 for "already used"


Check_Valid_USERID (Takes char string as input)
Initialize object "acc" of structure basic_details
Set Valid to true
Read acc.dat
Open "acc.dat"
Return variable: 1 for "can be used/unique" 0 for "is not unique"
False
check if end of file
True
Read File structure into acc object
All accounts read
Check if input char matches with userID of acc
False
True
Break
Close "acc.dat"
USERID exists on other acc
Set Valid to false
Return valid

---

create_buyer_details

Creates new buyer account and makes it active.


Create_buyer_details
set no_of_quotes on main_buyer as 0
Copy main_buyer name and residence and userID from main_acc
Open "buyer_details.dat"
Number of Quotes indicates the number of quotes of that account specifically
For storing and uniquely identifying each buyer
File for storing buyer_details structure objects
Terminate Process
Close "buyer_details.dat"
Write
Write main_buyer details into FILE

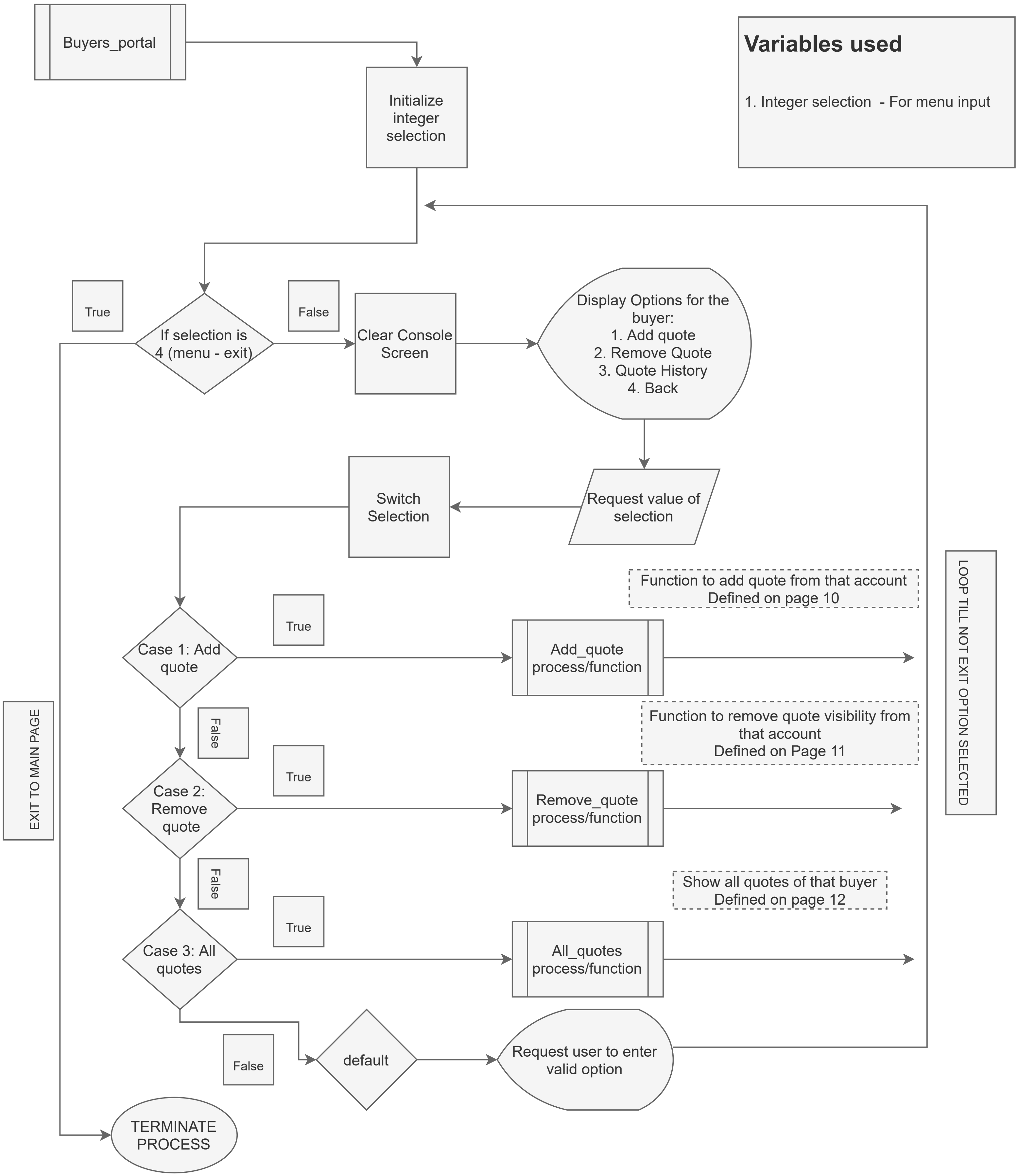
Buyer Portal

Buyer Portal

Displays Options for user if Buyer

Variables used

1. Integer selection - For menu input


Buyers_portal
Initialize integer selection
True
If selection is 4 (menu - exit)
False
Clear Console Screen
Display Options for the buyer:
1. Add quote
2. Remove Quote
3. Quote History
4. Back
Request value of selection
Switch Selection
Function to add quote from that account
Defined on page 10
True
Case 1: Add quote
Add_quote process/function
LOOP TILL NOT EXIT OPTION SELECTED
EXIT TO MAIN PAGE
False
Function to remove quote visibility from that account
Defined on Page 11
True
Case 2: Remove quote
Remove_quote process/function
False
Show all quotes of that buyer
Defined on page 12
True
Case 3: All quotes
All_quotes process/function
False
default
Request user to enter valid option
TERMINATE PROCESS

Add Quote

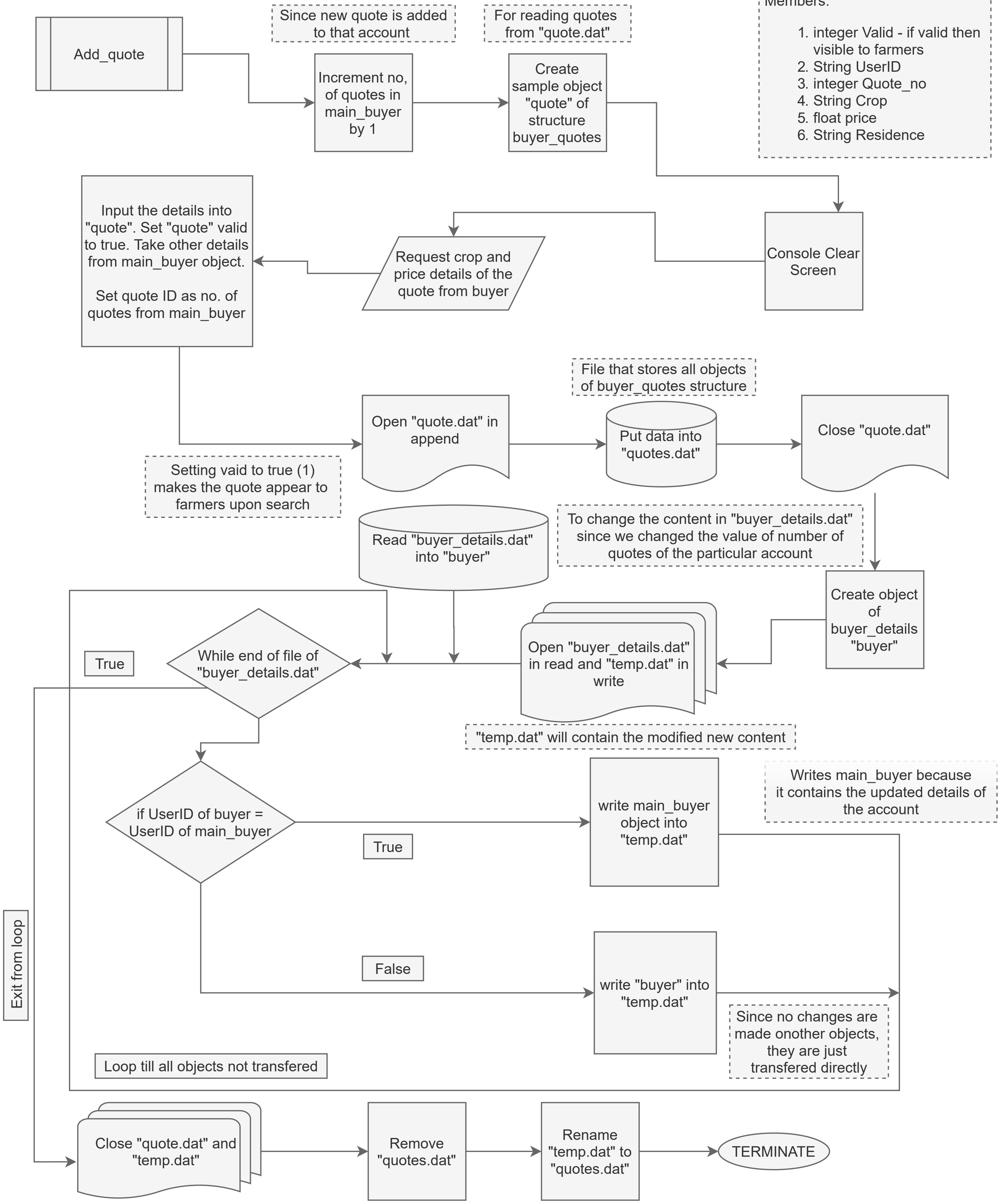
Add_quote()

Add Quote entry to database

Structure buyer_quotes

Members:

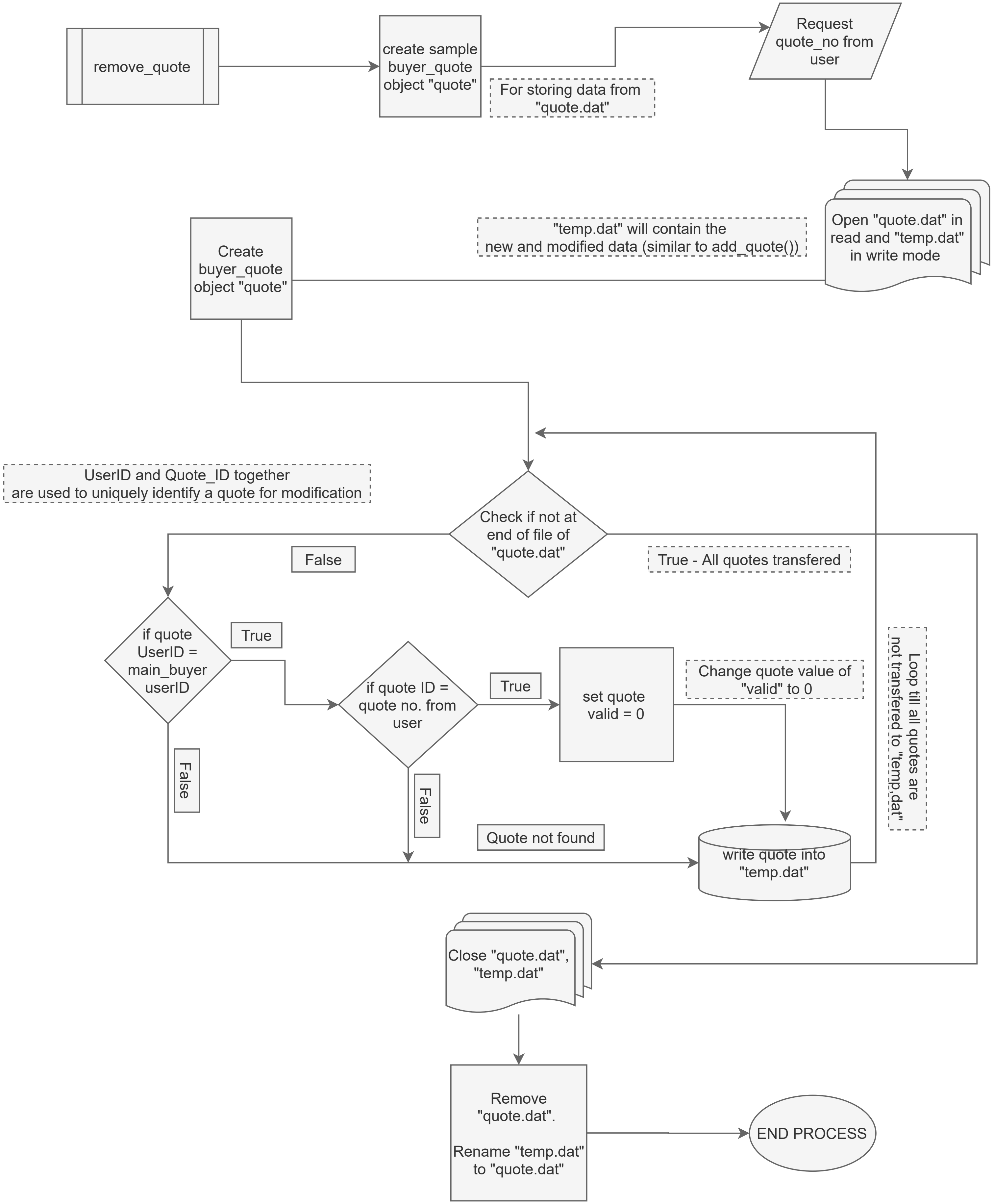
1. integer Valid - if valid then visible to farmers
2. String UserID
3. integer Quote_no
4. String Crop
5. float price
6. String Residence


Add_quote
Since new quote is added to that account
Increment no, of quotes in main_buyer by 1
For reading quotes from "quote.dat"
Create sample object "quote" of structure buyer_quotes
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
Setting vaid to true (1) makes the quote appear to farmers upon search
Open "quote.dat" in append
File that stores all objects of buyer_quotes structure
Put data into "quotes.dat"
Close "quote.dat"
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"
Read "buyer_details.dat" into "buyer"
Open "buyer_details.dat" in read and "temp.dat" in write
"temp.dat" will contain the modified new content
True
While end of file of "buyer_details.dat"
Exit from loop
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
Close "quote.dat" and "temp.dat"
Remove "quotes.dat"
Rename "temp.dat" to "quotes.dat"
TERMINATE

Remove Quote

Remove_quote()

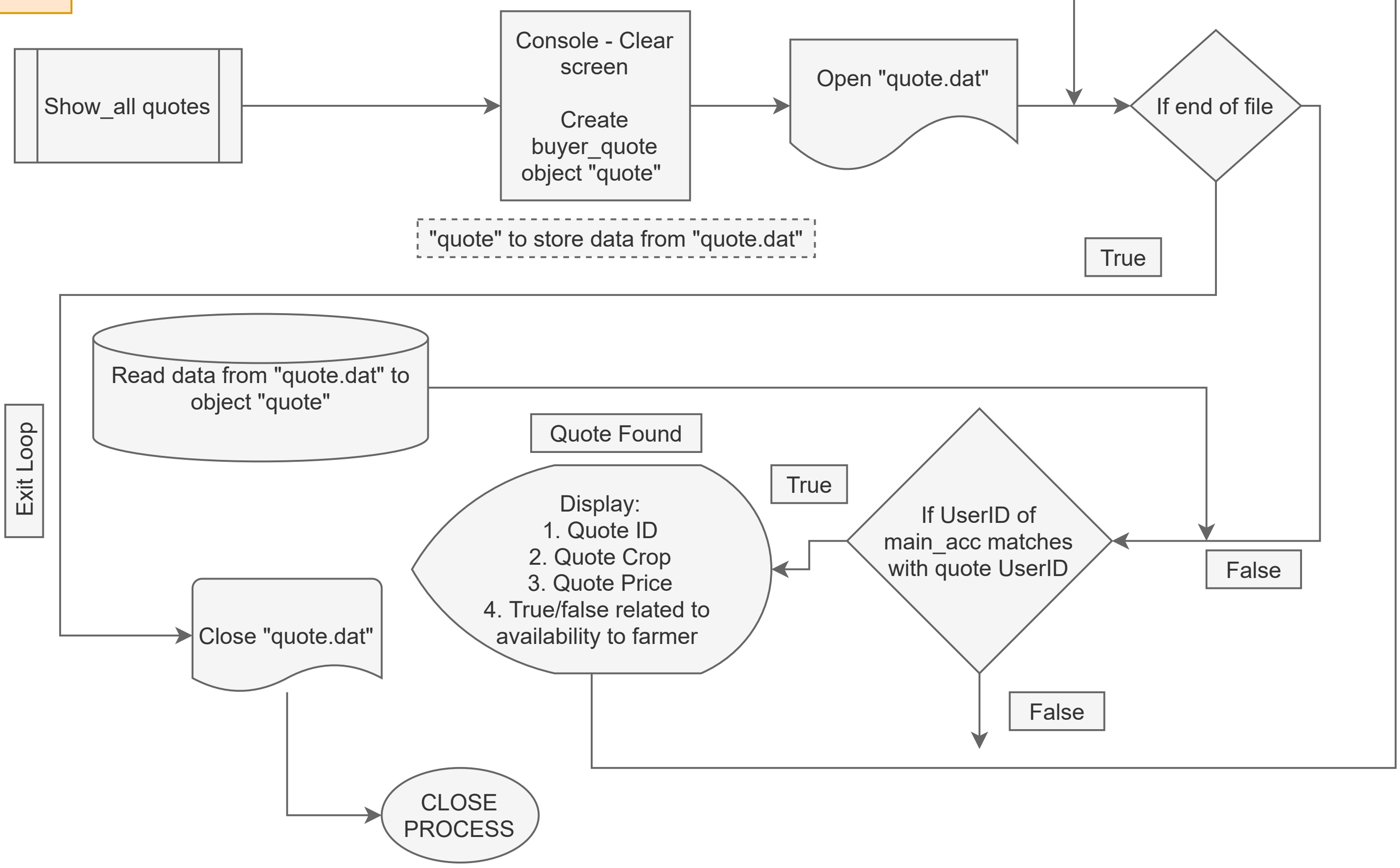
Removes quote availability for farmers (makes valid to false)


remove_quote
create sample buyer_quote object "quote"
For storing data from "quote.dat"
Request quote_no from user
Open "quote.dat" in read and "temp.dat" in write mode
"temp.dat" will contain the new and modified data (similar to add_quote())
Create buyer_quote object "quote"
UserID and Quote_ID together are used to uniquely identify a quote for modification
Check if not at end of file of "quote.dat"
False
True - All quotes transfered
if quote UserID = main_buyer userID
True
if quote ID = quote no. from user
True
set quote valid = 0
Change quote value of "valid" to 0
Loop till all quotes are not transfered to "temp.dat"
False
False
Quote not found
write quote into "temp.dat"
Close "quote.dat", "temp.dat"
Remove "quote.dat".
Rename "temp.dat" to "quote.dat"
END PROCESS

Quote History

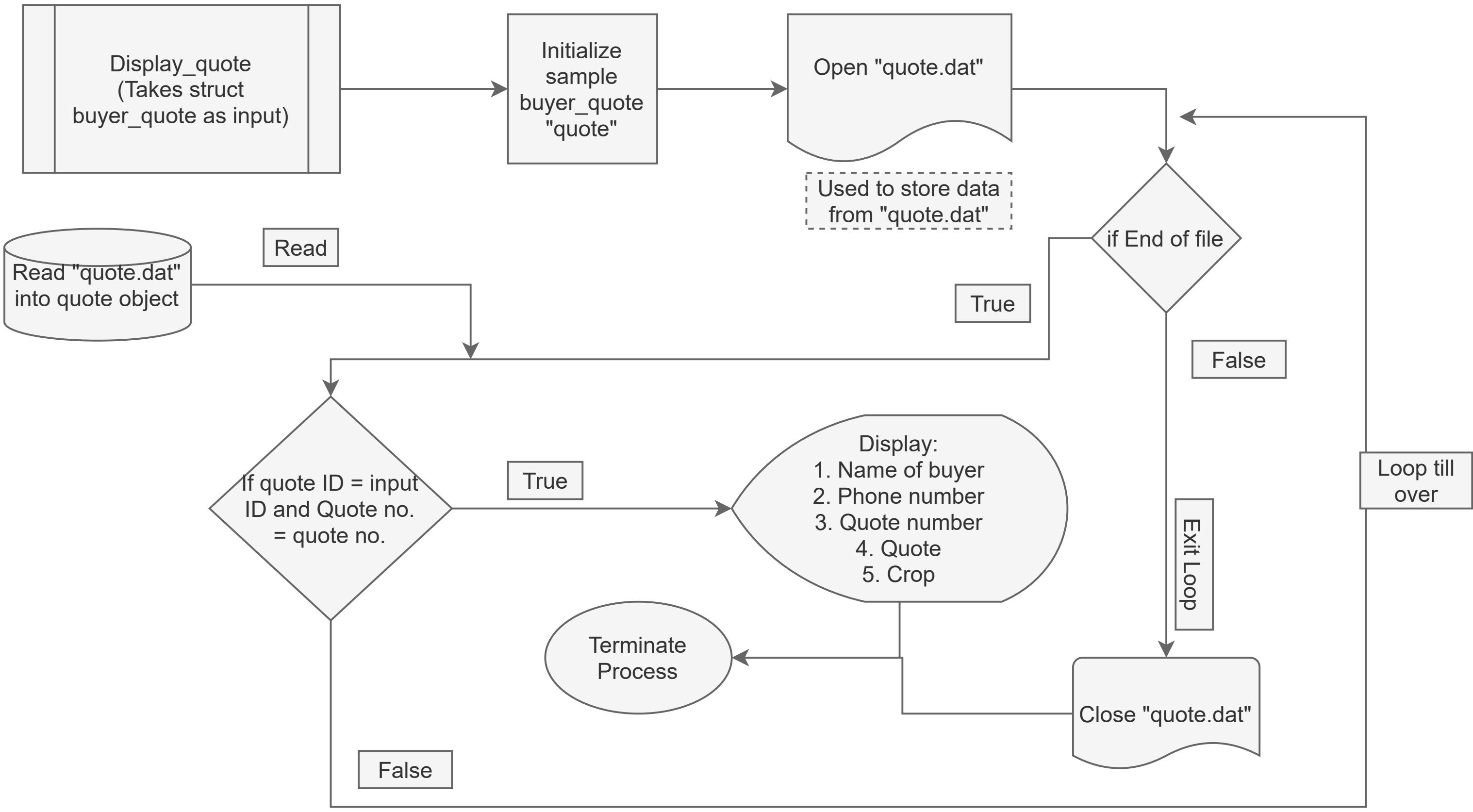
Quote History for Buyer

To show all quotes of that particular account


Show_all quotes
Console - Clear screen
Create buyer_quote object "quote"
"quote" to store data from "quote.dat"
Open "quote.dat"
If end of file
True
False
Exit Loop
Read data from "quote.dat" to object "quote"
Quote Found
Display:
1. Quote ID
2. Quote Crop
3. Quote Price
4. True/false related to availability to farmer
True
If UserID of main_acc matches with quote UserID
False
Close "quote.dat"
CLOSE PROCESS

Display_quote()

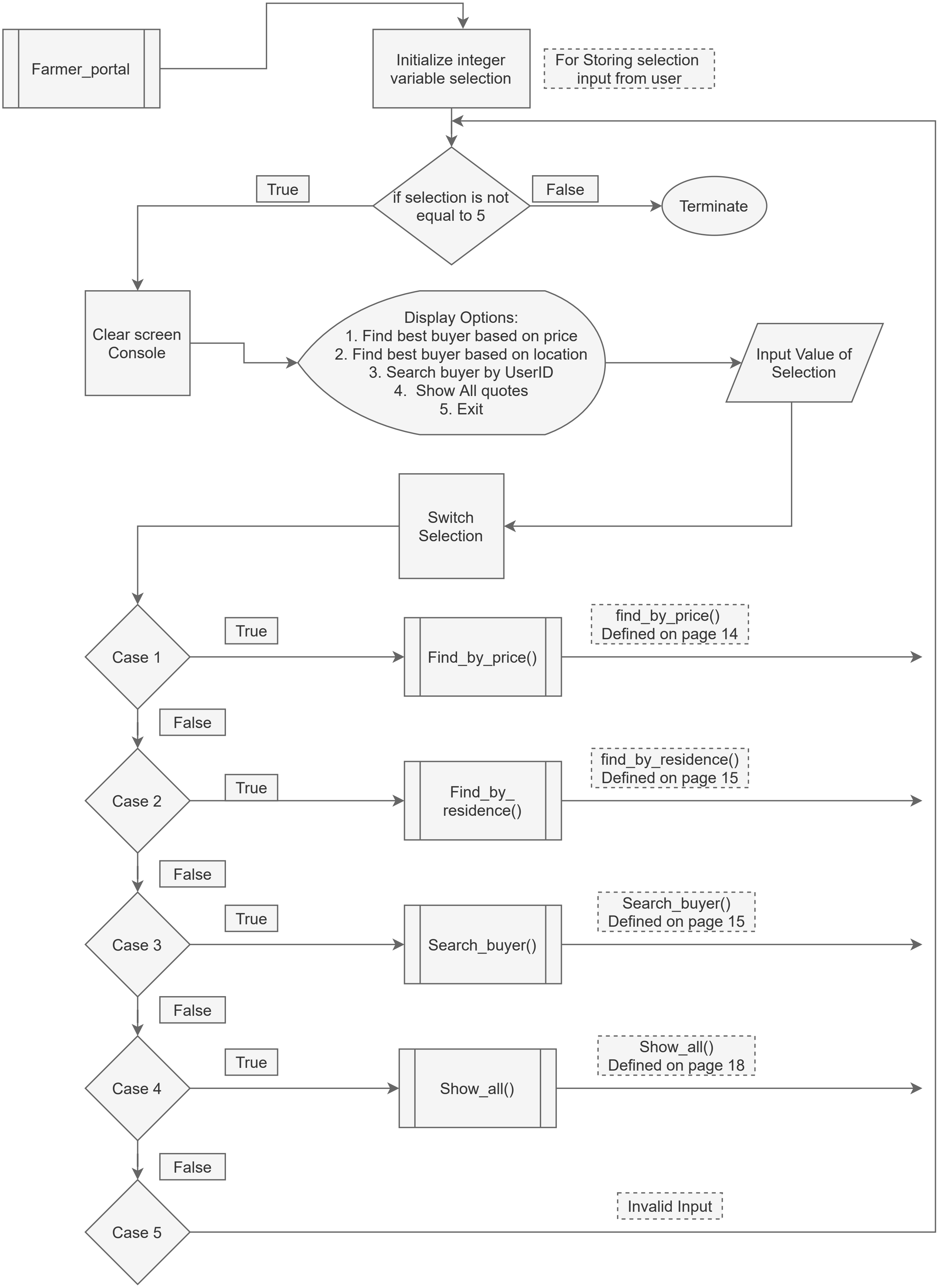
Takes in structure buyer_quotes object as input and displays its corresponding details


Display_quote (Takes struct buyer_quote as input)
Initialize sample buyer_quote "quote"
Open "quote.dat"
Used to store data from "quote.dat"
if End of file
True
False
Read
Read "quote.dat" into quote object
If quote ID = input ID and Quote no. = quote no.
True
Display:
1. Name of buyer
2. Phone number
3. Quote number
4. Quote
5. Crop
Exit Loop
Loop till over
Terminate Process
Close "quote.dat"
False

Farmer Portal

Farmer_portal()

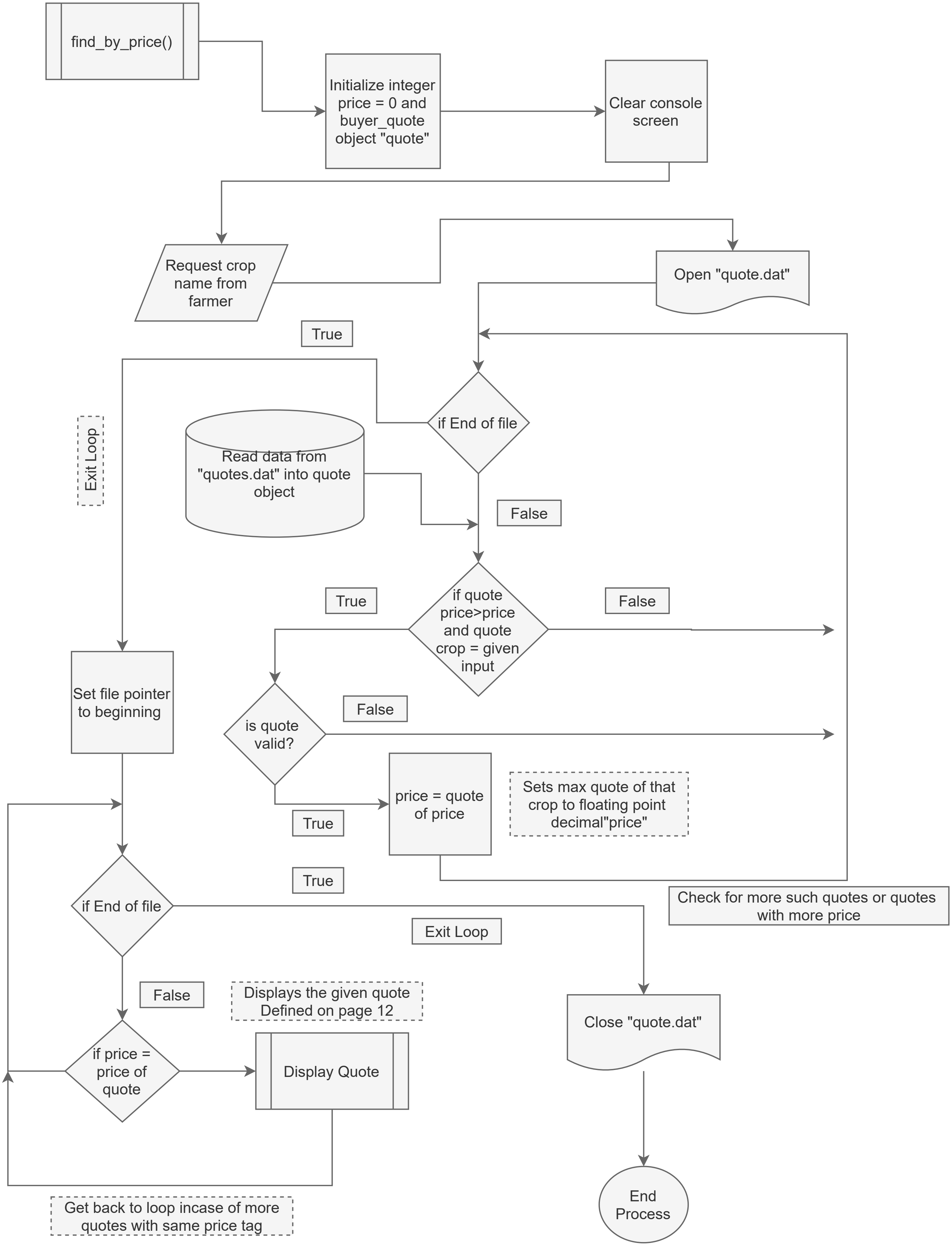
Shows Options for Farmer to access and search for quotes


Farmer_portal
Initialize integer variable selection
For Storing selection input from user
True
if selection is not equal to 5
False
Terminate
Clear screen Console
Display Options:
1. Find best buyer based on price
2. Find best buyer based on location
3. Search buyer by UserID
4. Show All quotes
5. Exit
Input Value of Selection
Switch Selection
Case 1
True
Find_by_price()
find_by_price()
Defined on page 14
False
Case 2
True
Find_by_
residence()
find_by_residence()
Defined on page 15
False
Case 3
True
Search_buyer()
Search_buyer()
Defined on page 15
False
Case 4
True
Show_all()
Show_all()
Defined on page 18
False
Case 5
Invalid Input

Find by Price

Find_by_price

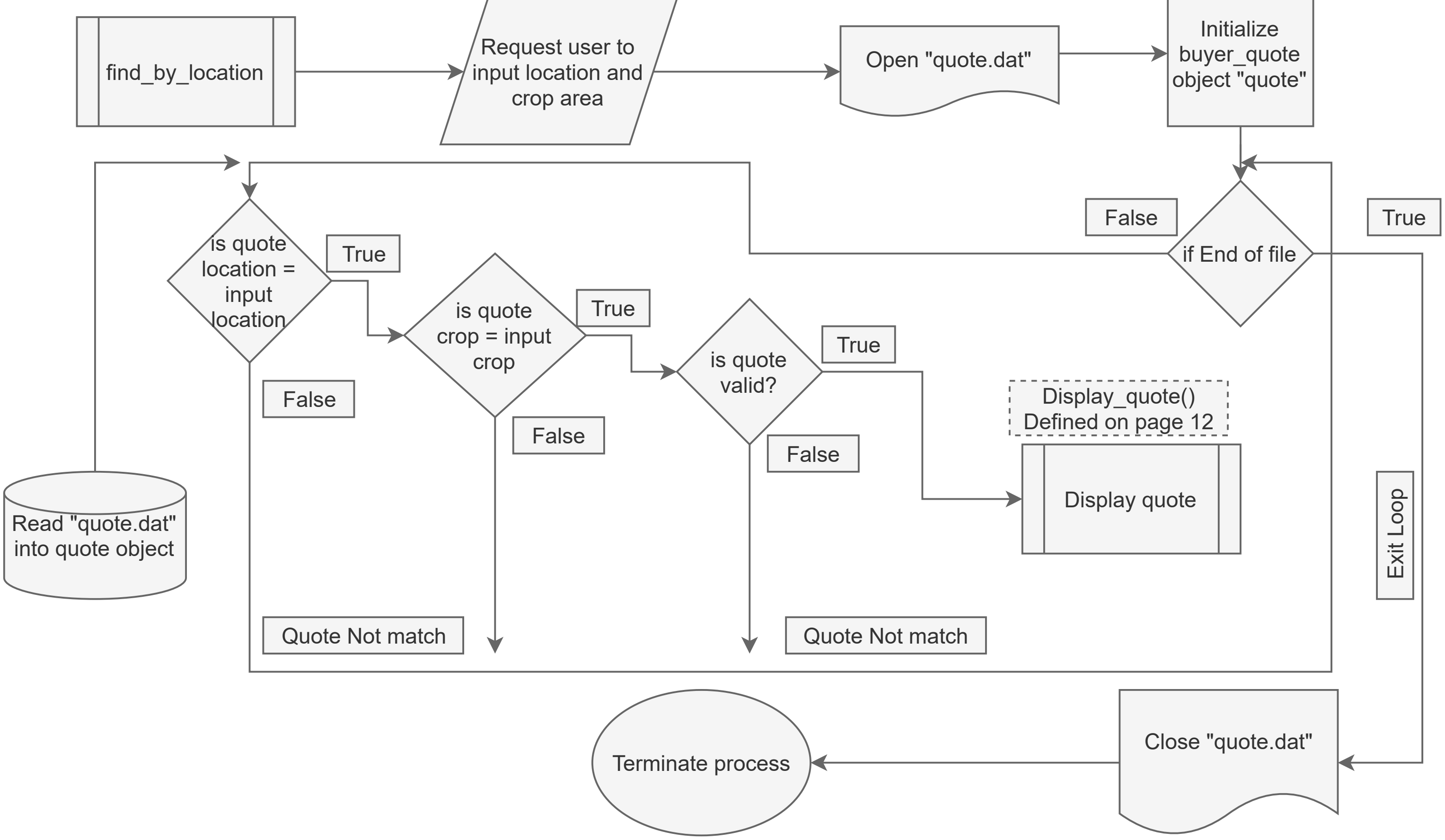
Finds best quote by price for a buyer depending on a given crop


find_by_price()
Initialize integer price = 0 and buyer_quote object "quote"
Clear console screen
Request crop name from farmer
Open "quote.dat"
True
Exit Loop
if End of file
Read data from "quotes.dat" into quote object
False
True
if quote price>price and quote crop = given input
False
Set file pointer to beginning
is quote valid?
False
price = quote of price
Sets max quote of that crop to floating point decimal"price"
True
True
if End of file
Check for more such quotes or quotes with more price
Exit Loop
False
Displays the given quote Defined on page 12
if price = price of quote
Display Quote
Close "quote.dat"
End Process
Get back to loop incase of more quotes with same price tag

Find By Location

Find_by_location

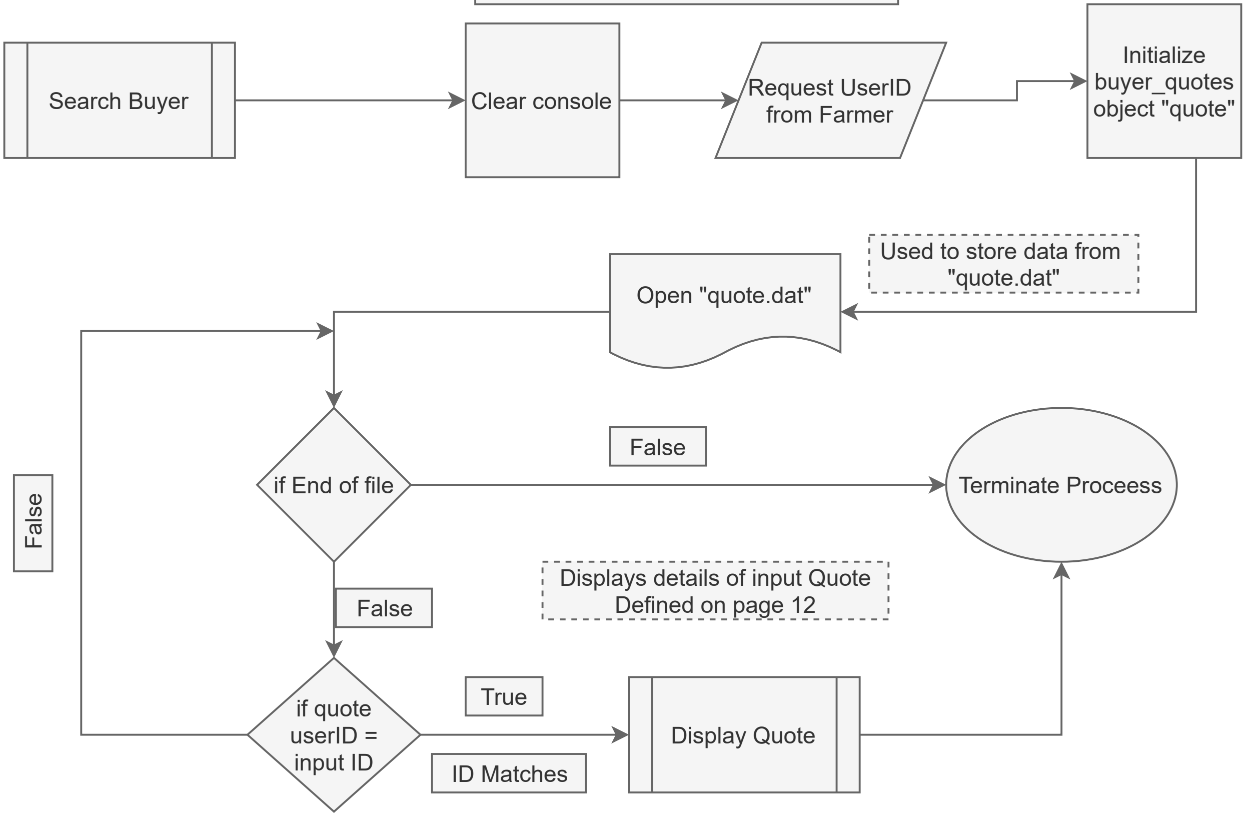
Used to find best quote available by location search


find_by_location
Request user to input location and crop area
Open "quote.dat"
Initialize buyer_quote object "quote"
if End of file
False
True
is quote location = input location
True
False
is quote crop = input crop
True
False
is quote valid?
True
False
Display_quote() Defined on page 12
Display quote
Read "quote.dat" into quote object
Exit Loop
Quote Not match
Quote Not match
Terminate process
Close "quote.dat"

Search by ID

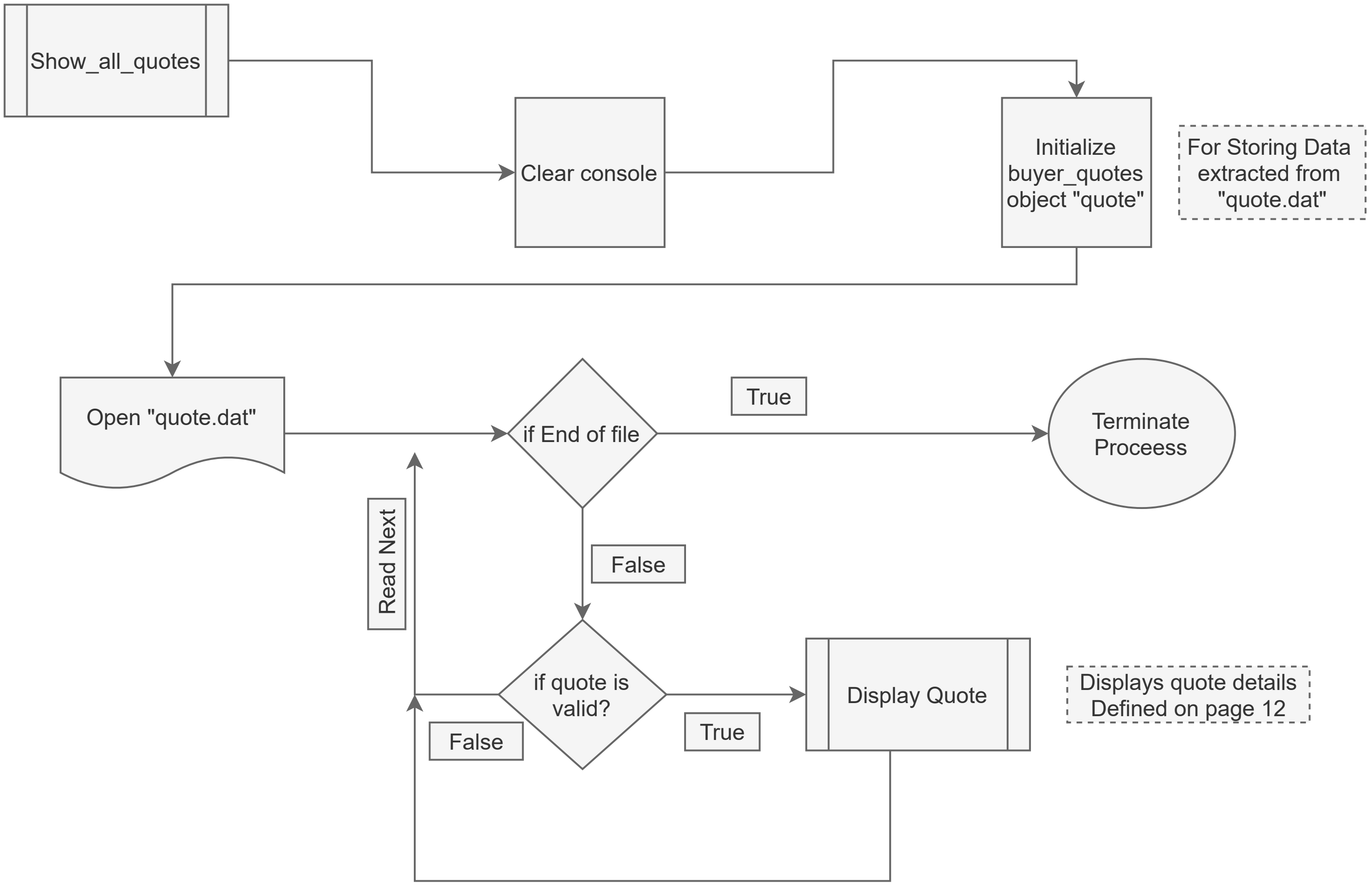
Search_by_ID

Allows farmer to search a buyer by ID


Search Buyer
Clear console
Request UserID from Farmer
Initialize buyer_quotes object "quote"
Used to store data from "quote.dat"
Open "quote.dat"
if End of file
False
Terminate Proceess
False
False
Displays details of input Quote Defined on page 12
if quote userID = input ID
True
ID Matches
Display Quote

Show_all_quote

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


Show_all_quotes
Clear console
Initialize buyer_quotes object "quote"
Open "quote.dat"
if End of file
True
Terminate Proceess
False
Read Next
if quote is valid?
False
True
Display Quote

For Storing Data extracted from "quote.dat"

Displays quote details Defined on page 12