

# CS MINI PROJECT

## FARMHOUSE

Prepared By Dhruv B. (191CH013) and Pranshu Shukla (191ME260)

### Program Synopsis

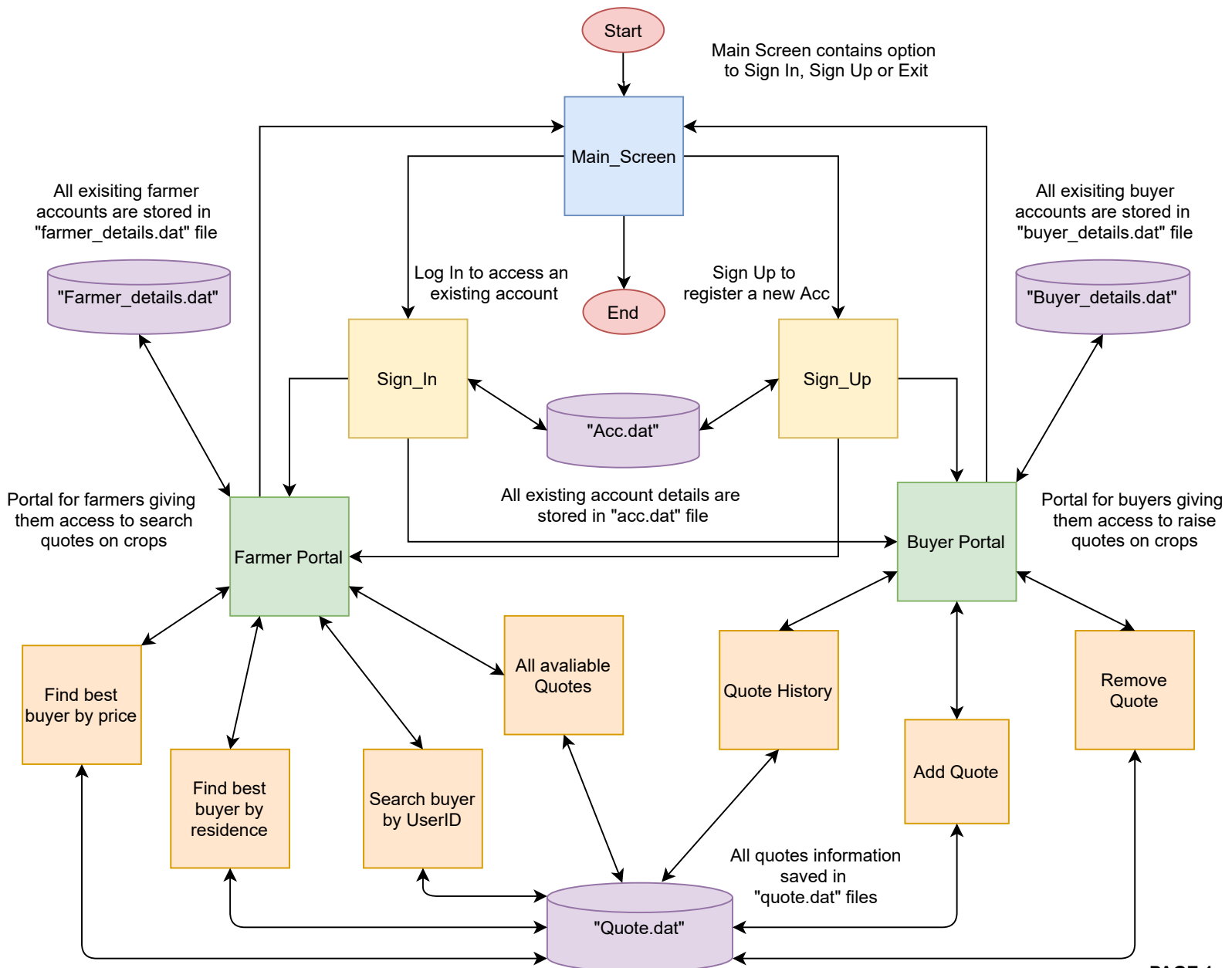
Farm-House is a user friendly program application that allows a quick and easy communication and transfer of information between a farmer and the buyer. Farm-House allows different users to choose which category they belong to (Farmer or Buyer) and provides them with appropriate options to make their interface as simple as possible.

### Program Structure

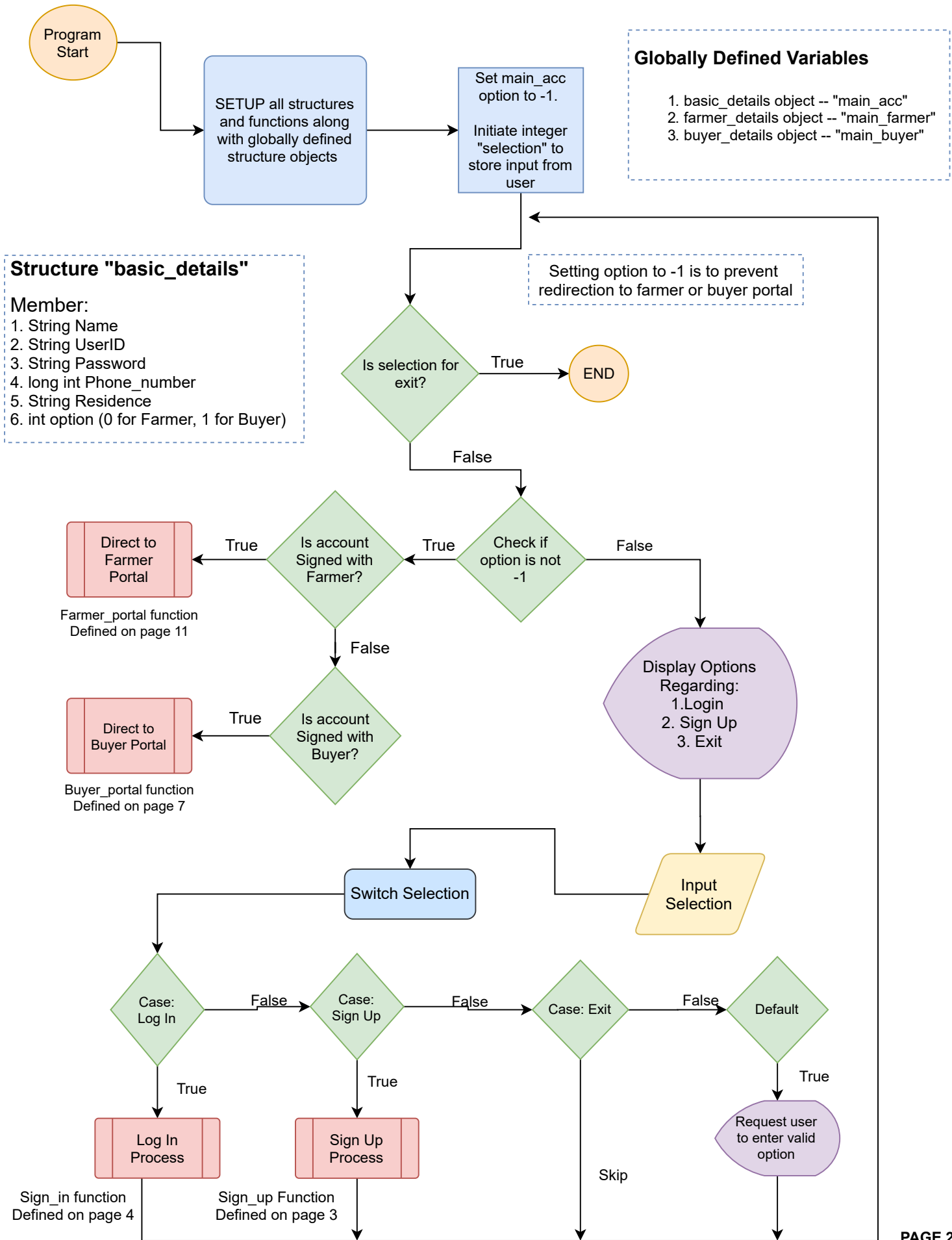
Farm-House uses 4 structures:

1. "basic\_details" - for storing UserID and Password of an acc as well as other details
2. "farmer\_details" - for storing farmer related information of the account
3. "buyer\_details" - for storing buyer related information of the account
4. "buyer\_quotes" - for storing information on each quote issued by a buyer on the application

### Program Overview Design



# Function-Wise Flowchart

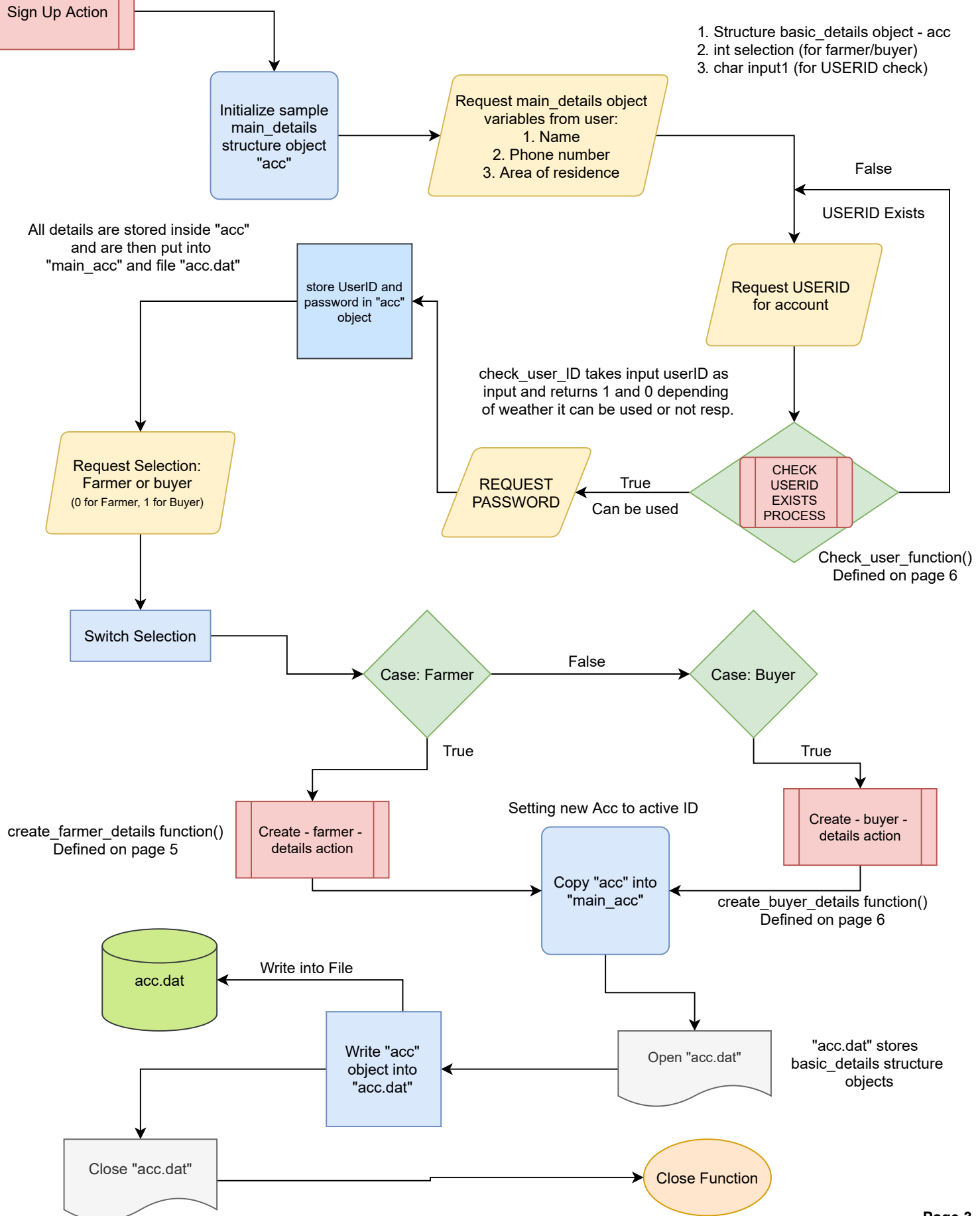


# SIGN UP FUNCTION

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 In function

Used to Log-In into a new acc

**struct farmer\_details**

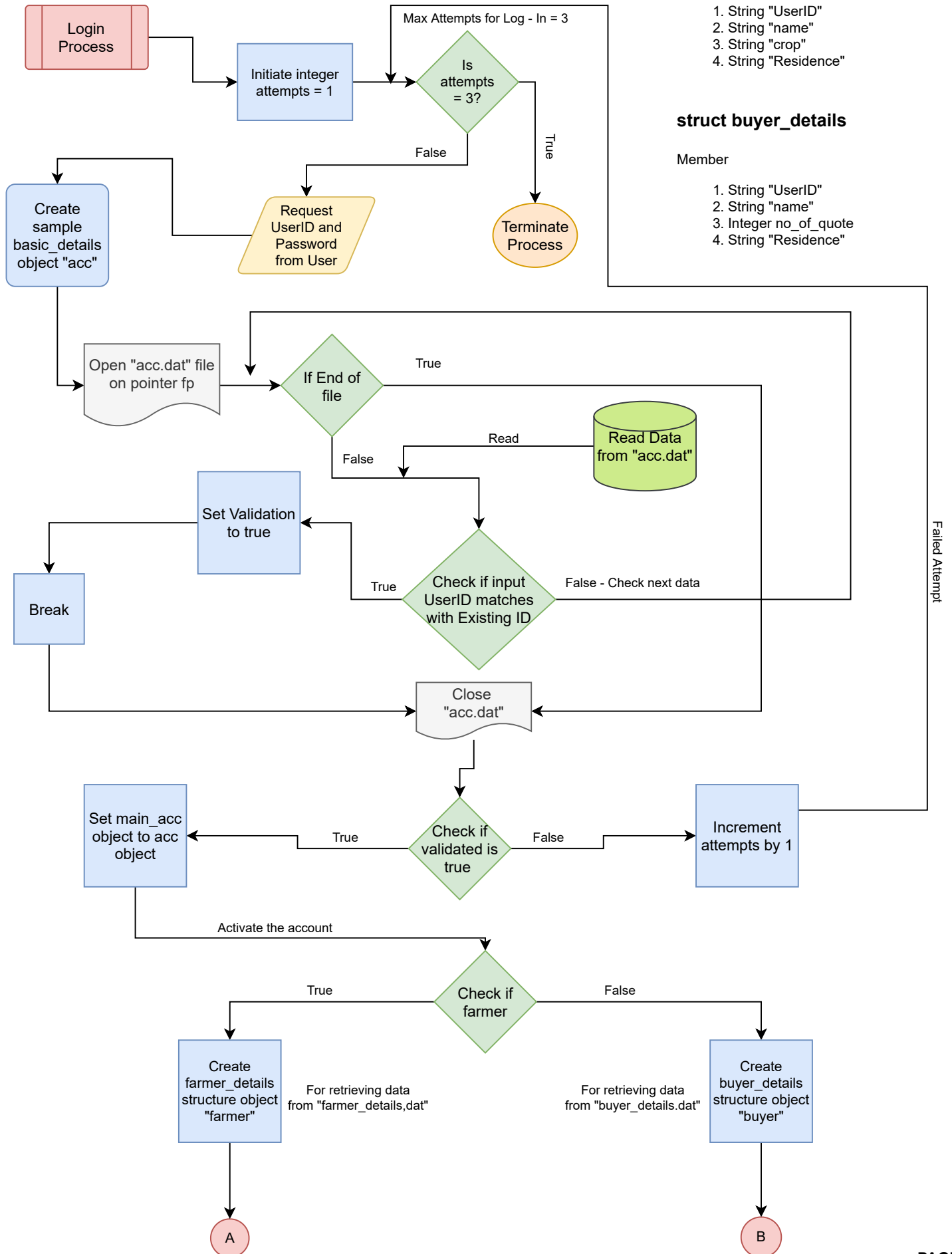
Member

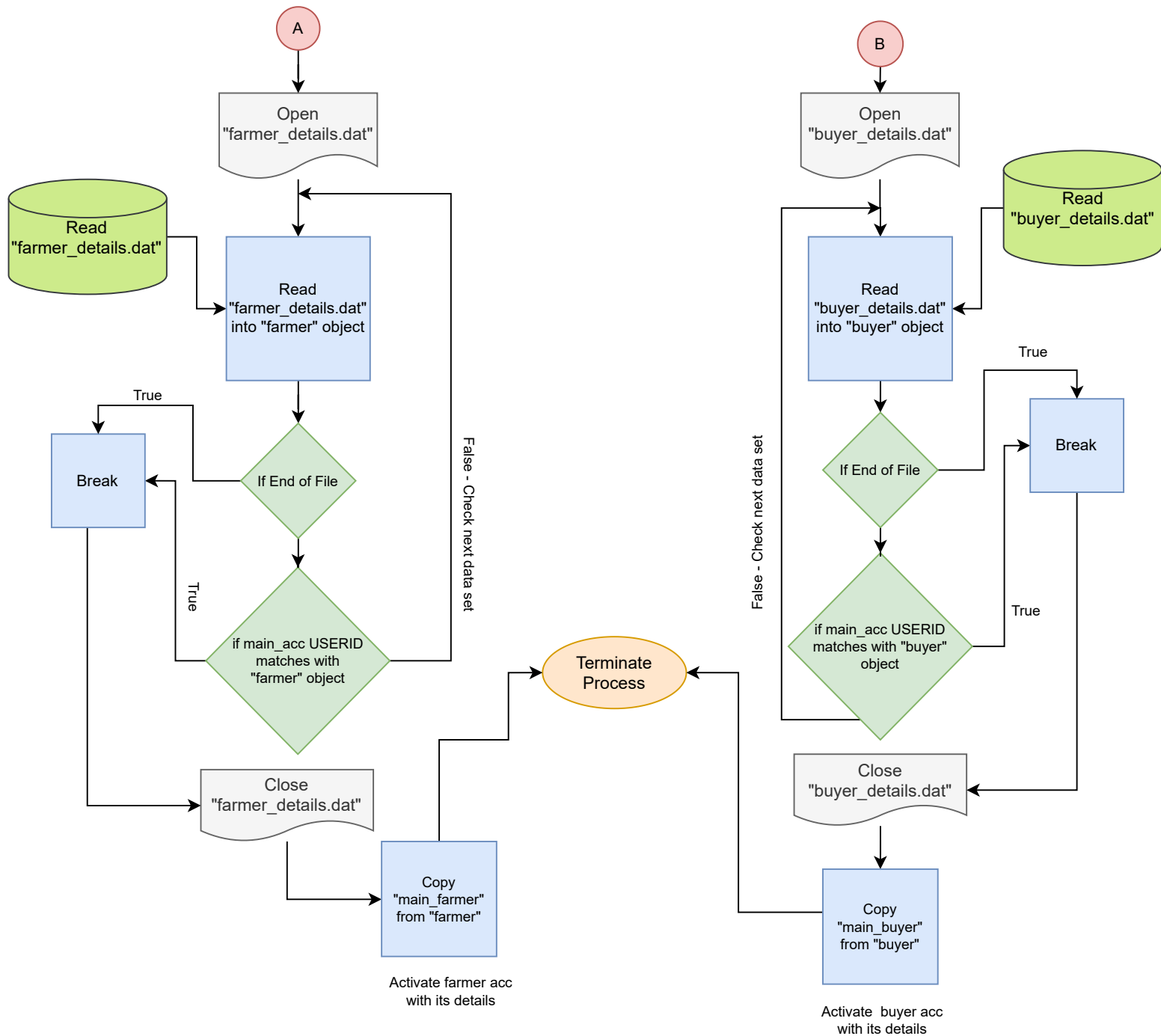
1. String "UserID"
2. String "name"
3. String "crop"
4. String "Residence"

**struct buyer\_details**

Member

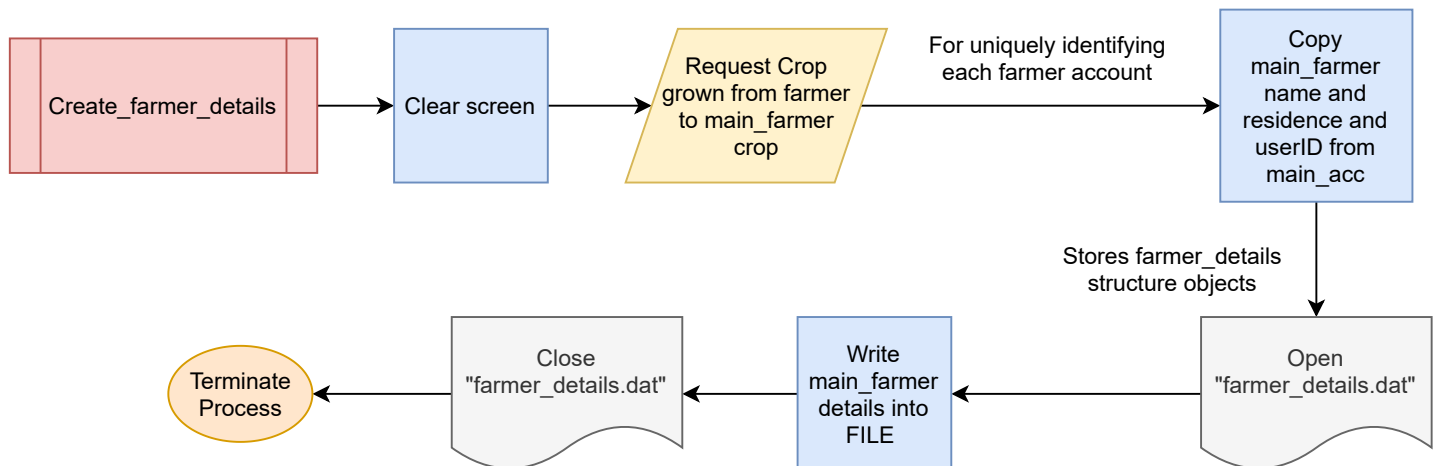
1. String "UserID"
2. String "name"
3. Integer no\_of\_quote
4. String "Residence"





## Create\_farmer\_details

Creates farmer details entry in file and sets to active account

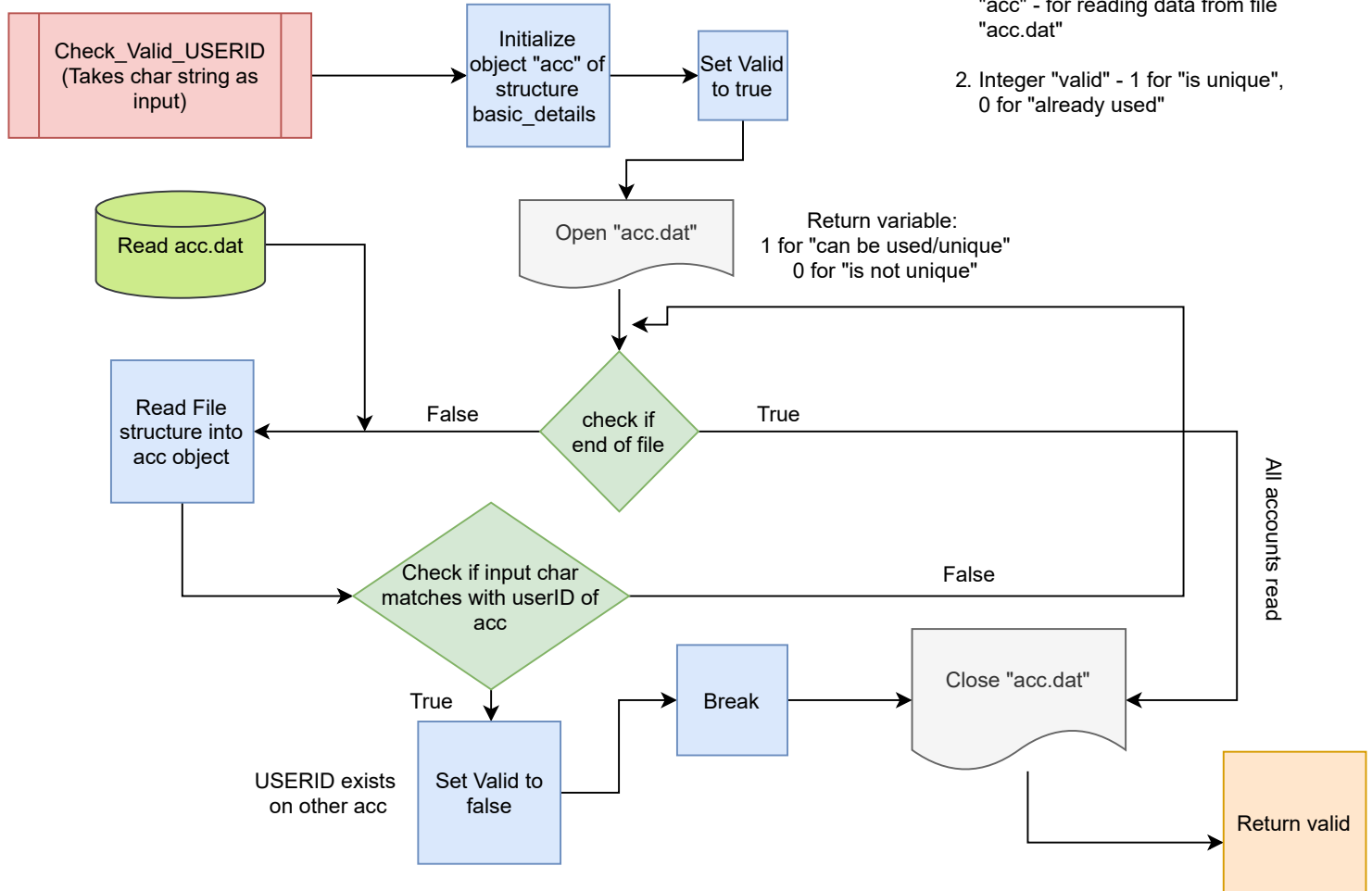


# check\_valid\_userID

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

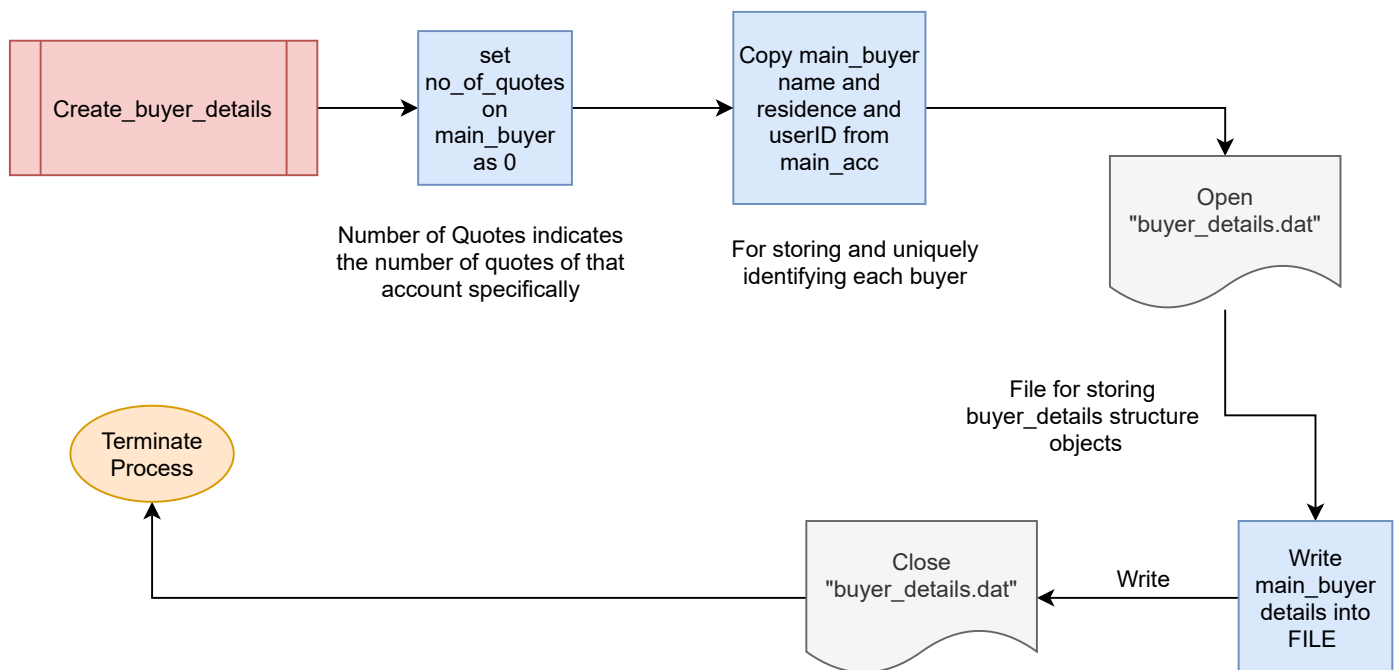
## Variables used

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.

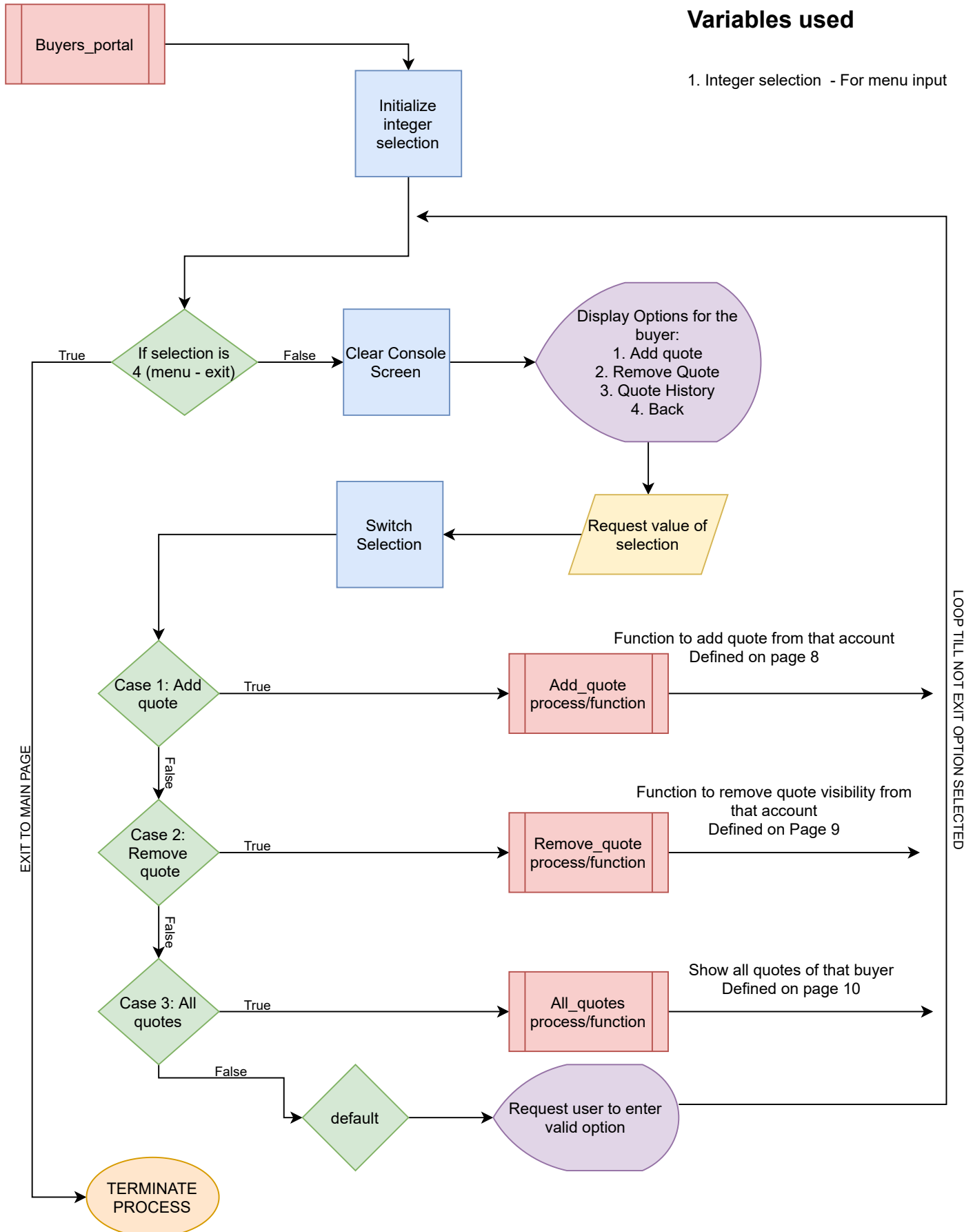


# Buyer Portal

Displays Options for user if Buyer

## Variables used

1. Integer selection - For menu input



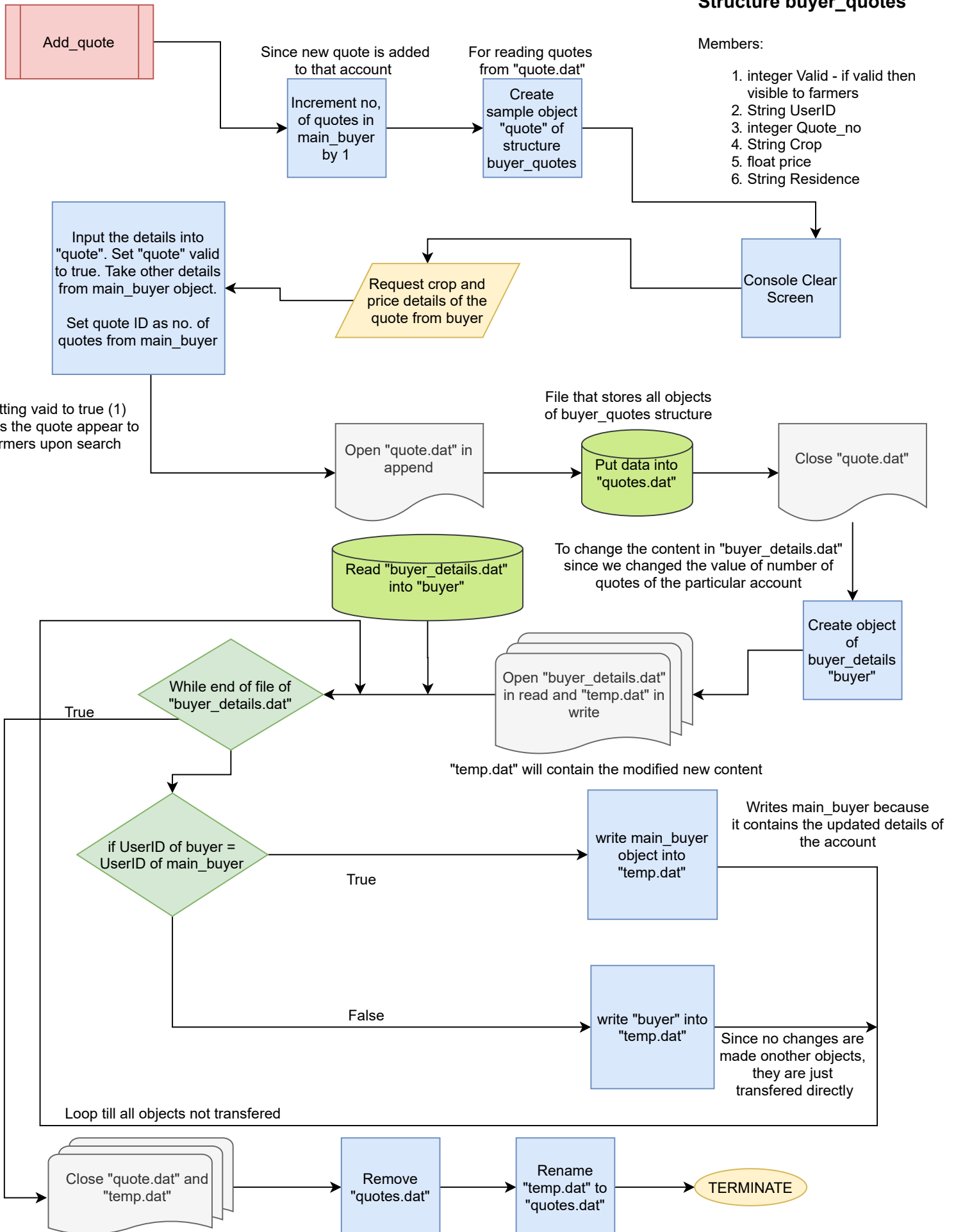
# Add\_quote()

Add Quote entry to database

## Structure buyer\_quotes

Members:

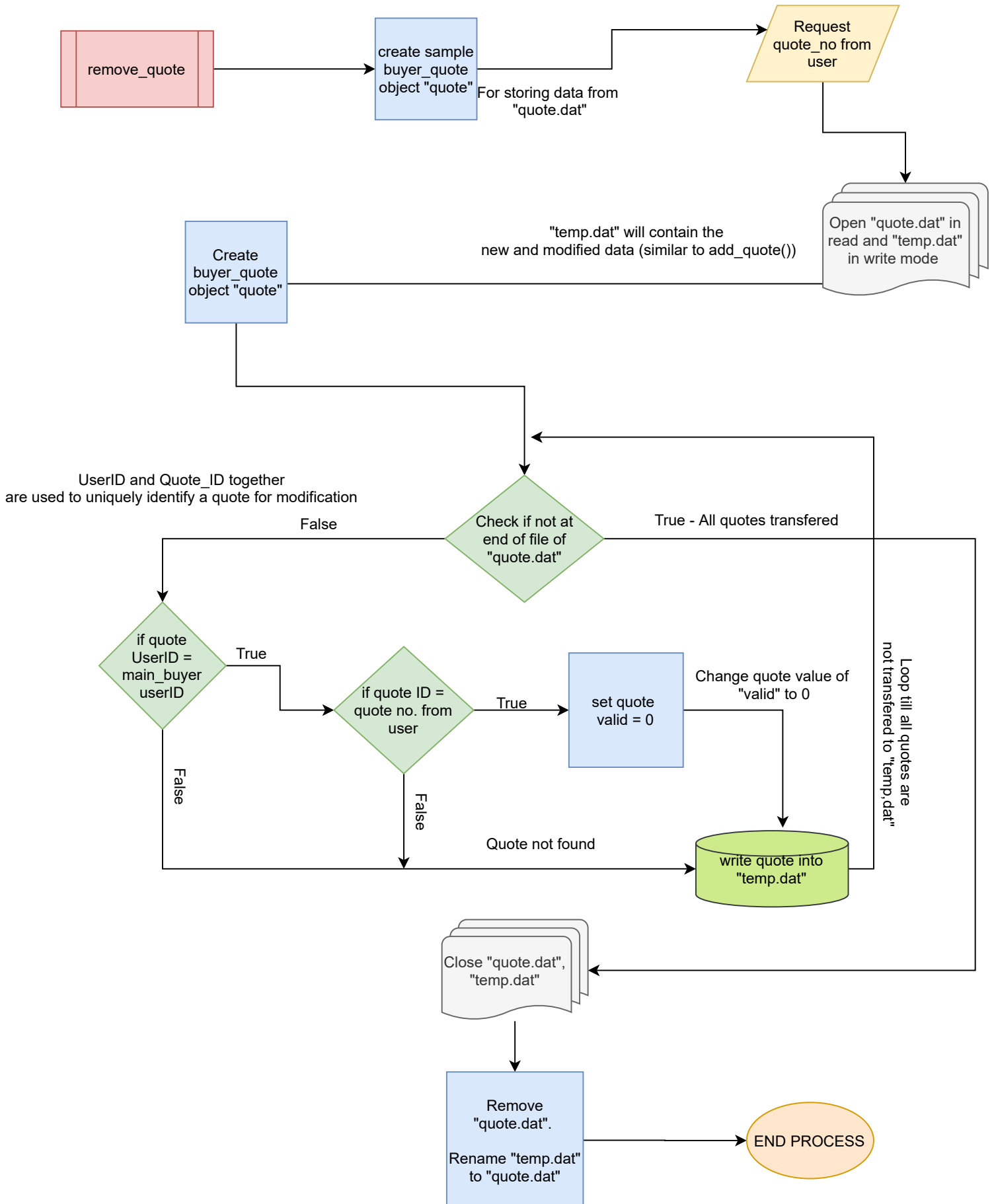
1. integer Valid - if valid then visible to farmers
2. String UserID
3. integer Quote\_no
4. String Crop
5. float price
6. String Residence





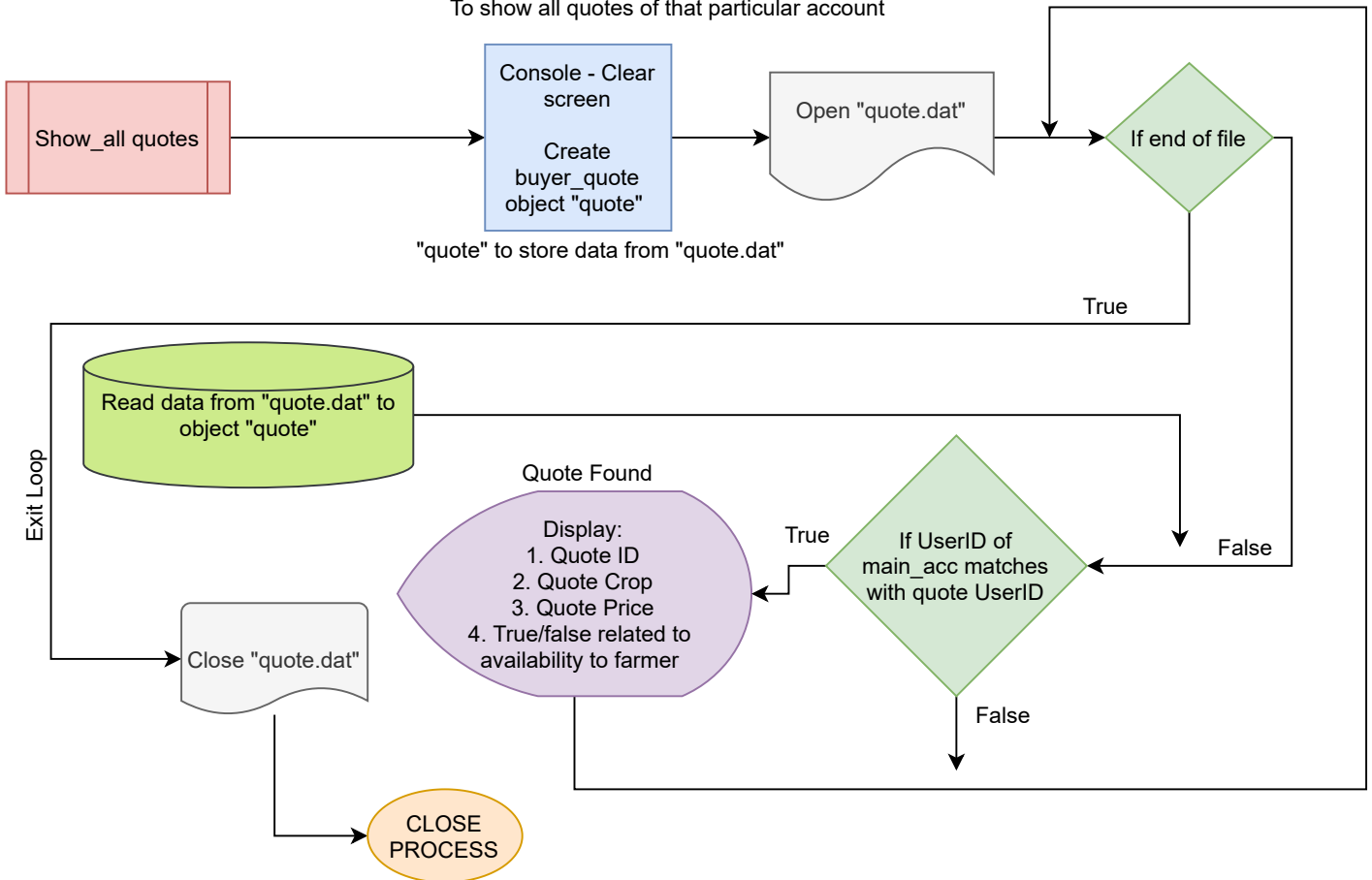
# Remove\_quote()

Removes quote availability for farmers (makes valid to false)



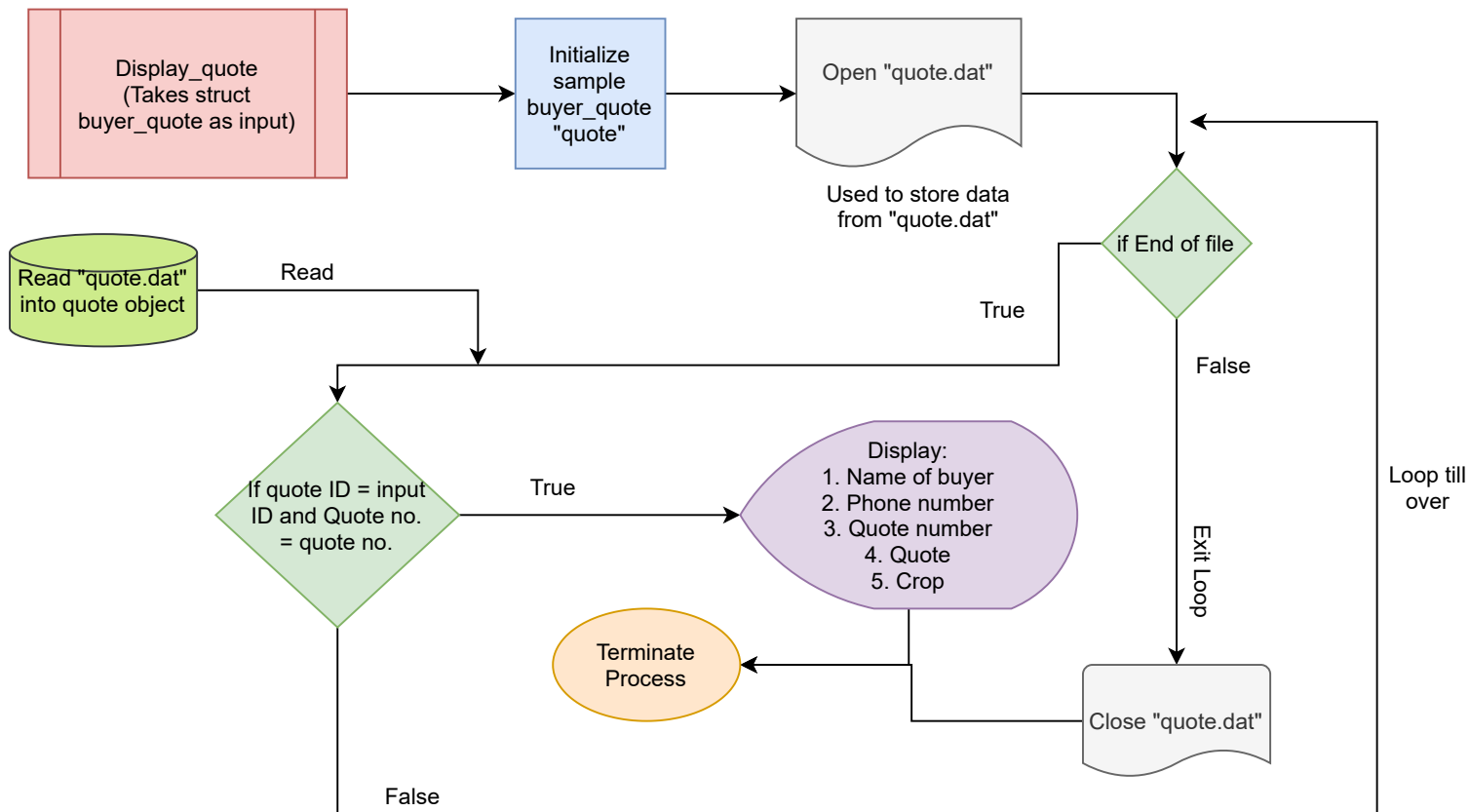
# Quote History for Buyer

To show all quotes of that particular account



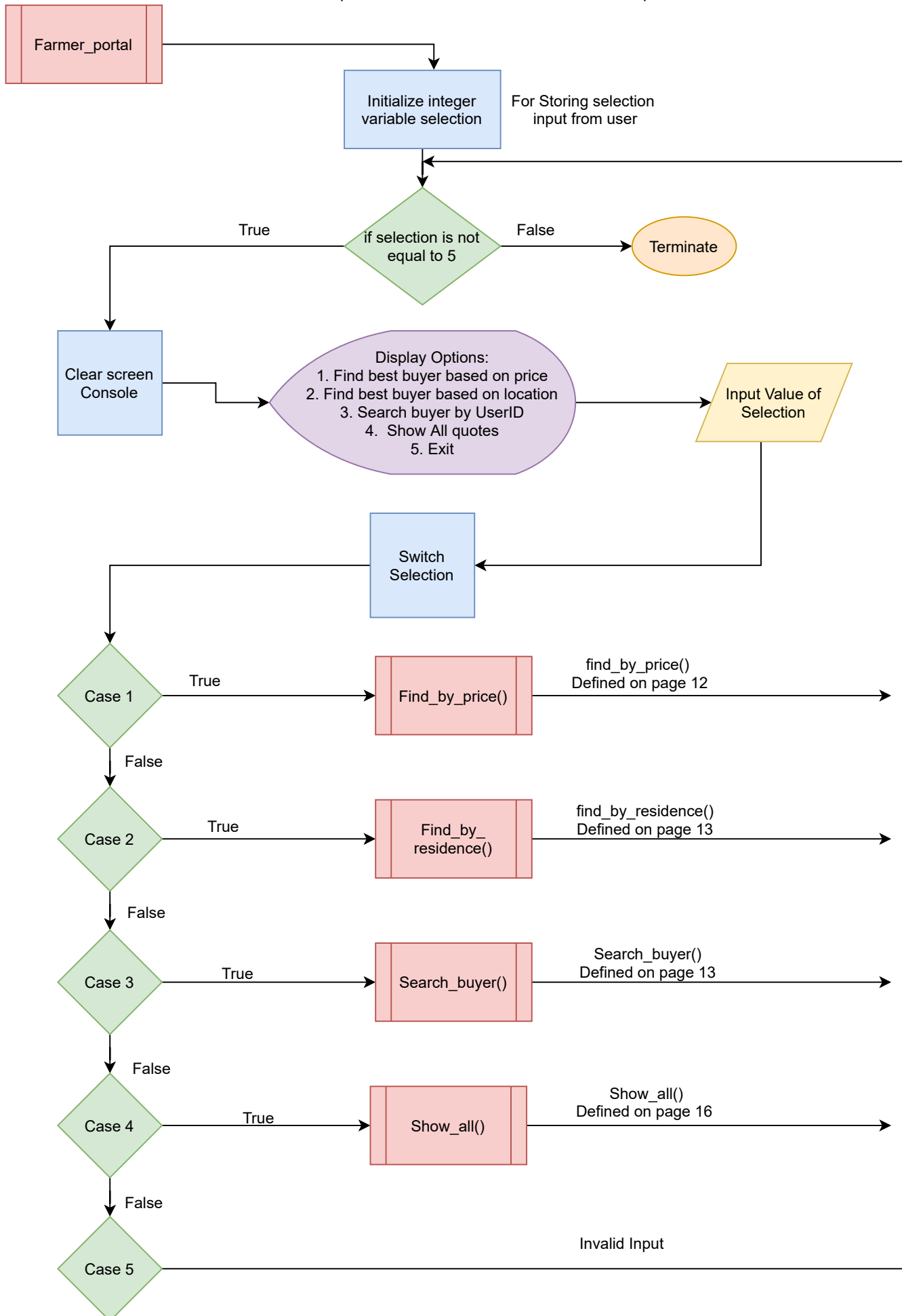
## Display\_quote()

Takes in structure buyer\_quotes object as input and displays its corresponding details



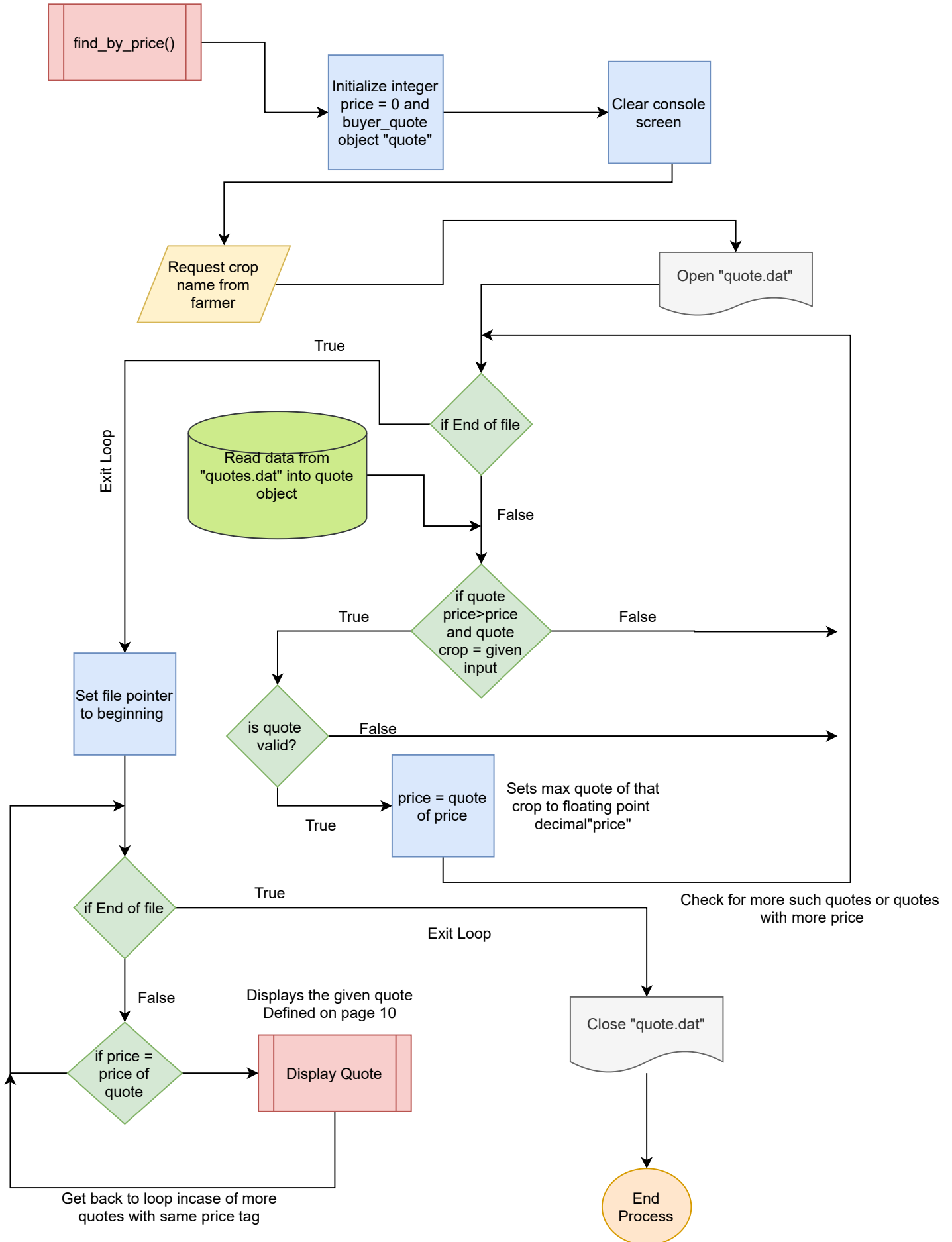
# Farmer\_portal()

Shows Options for Farmer to access and search for quotes



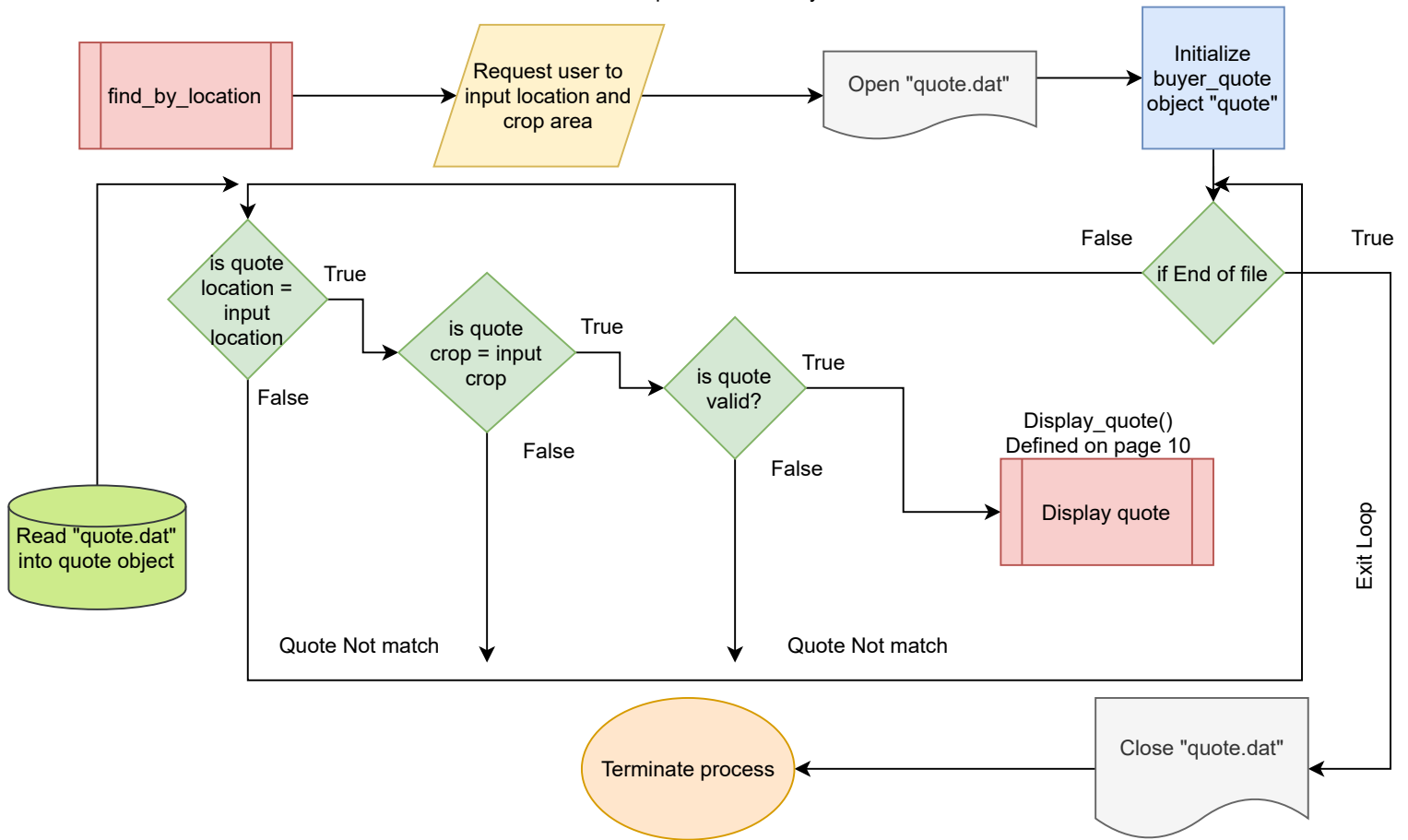
# Find\_by\_price

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



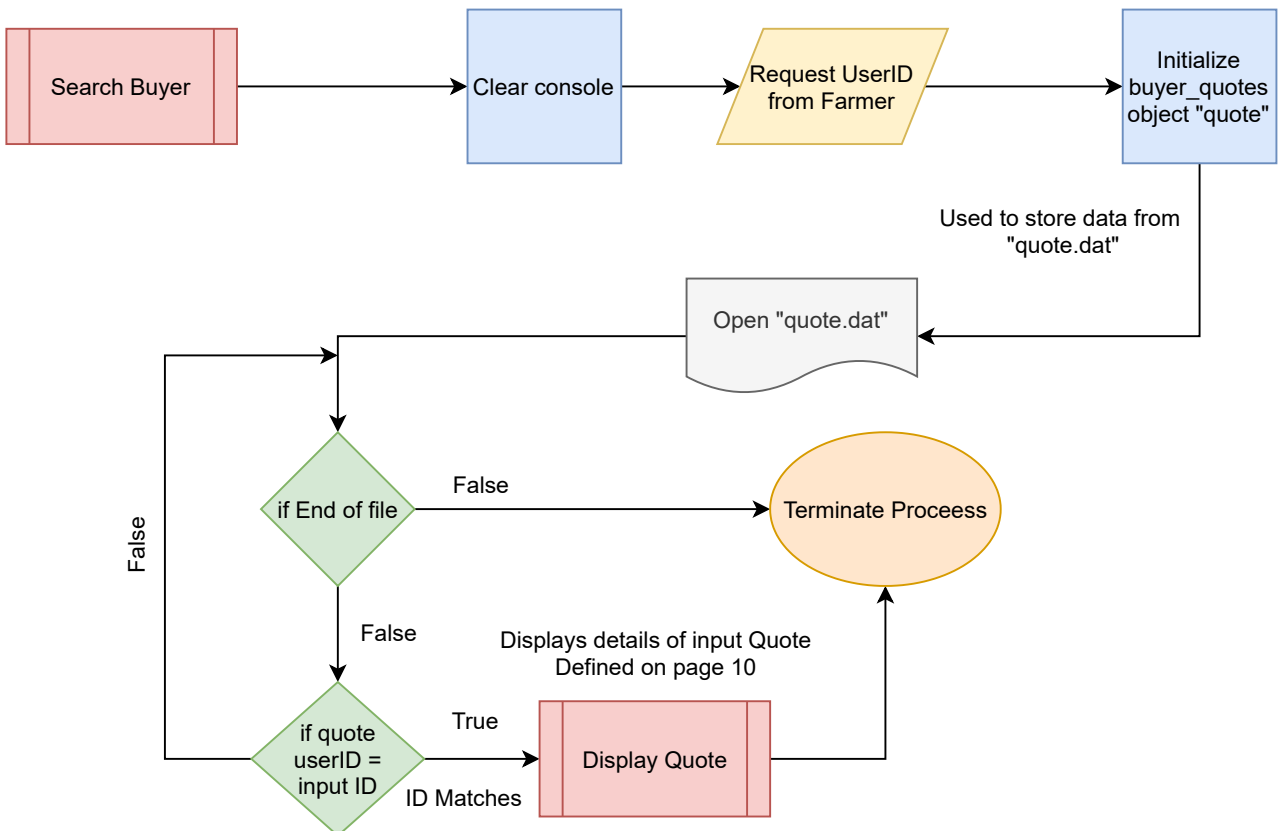
## Find\_by\_location

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)

