RETAABILE ZUZANI

IT PAT 2024

THE PROPERTY MARKET

Contents

TASK 0

TOPIC

A program related to the property market (A program for the purchasing and selling of property)

PURPOSE

It provides a platform where real estate agents interested in assisting with the sale of property may find what they're looking for (available property) with the aid of categorisation (price range that the owner is wanting to sell the property for, number of bedrooms, etc) and may be put in direct contact with the owners if interested. The platform also allows owners to upload their property and details needed for the real estate agent to help calculate the value of the home.

WHY IS THIS A SOLUTION

This program provides an easier, less stressful method of obtaining a real estate agent to help with the final sale of the property the owner is selling. It is helpful for both the owner and the real estate agent as the owner only needs to advertise the property by putting it up on the program and the real estate agent only needs to search through the filters to find owners to assist regarding the selling process.

SCOPE

This program can:

- Allow the search for property with specific features (number of bedrooms, etc)
- Advertise available property deliberately put up by owners
- Connect owner and future buyer by allowing owner contact detail access
- Allow owner to calculate average mortgage prices for their home
- Allow owner to put in contact details along with their property submission
- Allow property filtering according to price range

This program cannot:

- Allow the upload of more than one image for the propertu
- Work offline
- Work with speed all the time

TASK 1: RESEARCH

1A: TASK DEFINITION

The property market is a market that will always stay relevant in any number of years. As time goes on, society makes every market efficient with the development of technology designed specifically to make work easier. Every market has adapted the use of technologies to make products more accessible to their target audiences, and the property market is no different. Faster connection between real estate agents and owners are crucial to a more efficient property market.

Hence, in accordance to this year's topic, my project will be a platform where owners ready to sell their property can advertise it by adding images, details and the price the property is being sold by. Furthermore, real estate agents interested in helping home owners find the right buyer may make use of built in filters to find exactly what they're looking for, and have their contact details ready in case the owner does wish to work with them when notified of interest.

Following the brief of the topic, the program will allow for increased efficiency in the property market by being a platform where quick offers for a professional relationship between owner and real estate agent can be kickstarted. This allows for a speedier process of putting property on sale, and leads to faster offers on the property being made by the general public.

1B: USER REQUIREMENTS

The users of this program are the owners looking to sell their property through an independent real estate agent, as well as the real estate agents finding property that the owners would like to sell in exchange for paying a commission fee to the real estate agents.

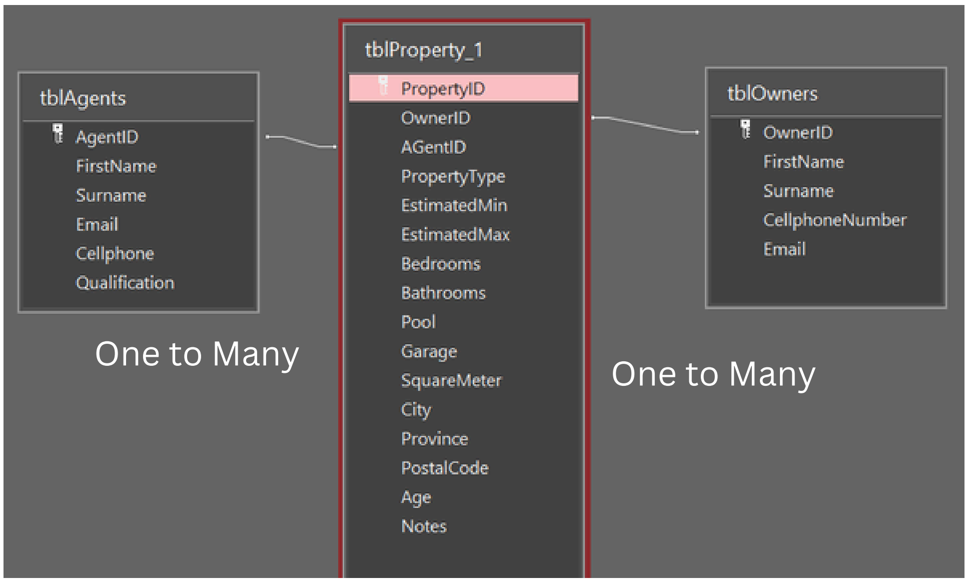
	Owners	**Real Estate Agents**	**Programmer**
Activity /User needs (what they need to be able to do)	Need to be able to upload images and details of the property (age, number of bedrooms, etc)	Need to be able to search through property based on specifics (number of bedrooms, area, etc)	• Need to ensure that image based file types and additional information can be uploaded • Need to allow for filtering
	Need to be able to edit already existing data on already uploaded property		Need to allow for the editing of already existing data in database tables
	Need to know the minimum and maximum commision the real estate agent is looking for	Need to set their rate to multiply the estimated minimum and maximum price of the home in order to have their commission worked out	Need to ensure correct calculations are being done
	Need to be able to edit/update their personal details/information	Need to be able to edit/update their personal details/information	Need to ensure these edits can be made and are saved

LIMITATIONS

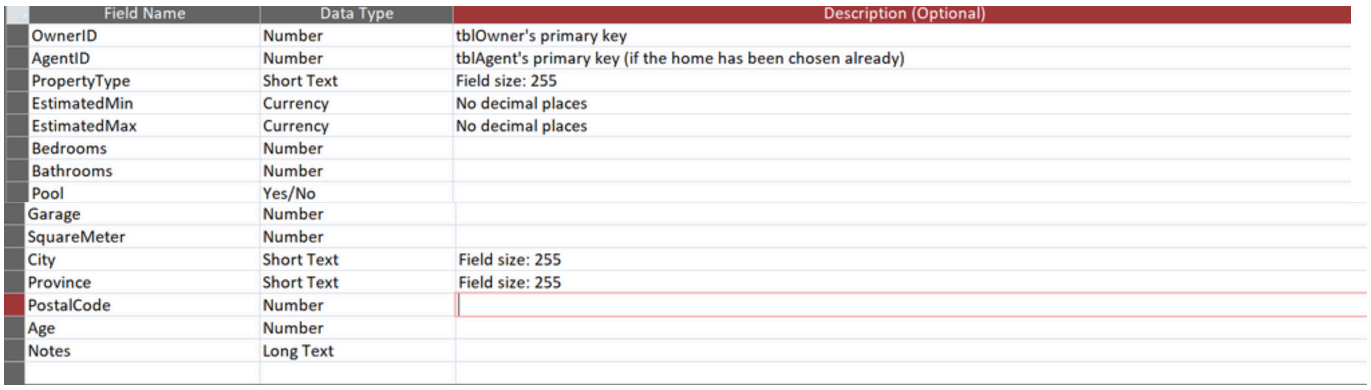
Owner	Real Estate Agent	Programmer
Needs to wait for real estate agent to reach out to them	Cannot see a lot of images of property at once	The programmer needs to constantly check the needed qualification to be a legally certified real estate agent, and make edits to the fixed variable as a result
Internet is needed to make use of the program	Internet is needed to make use of the program	

TASK 2: DATABASE

A database is defined as an organised collection of information in fields within records regarding a specific subject. My database, dbProperty, consists of three tables, namely tblOwners, tblAgents and tblProperty. tblOwners stores the contact information and basic details of the owner, tblAgent stores the contact information and basic details of the agents, and tblProperty stores all the relevant information regarding the property put up on the program. There is a One To Many relationship between the three.



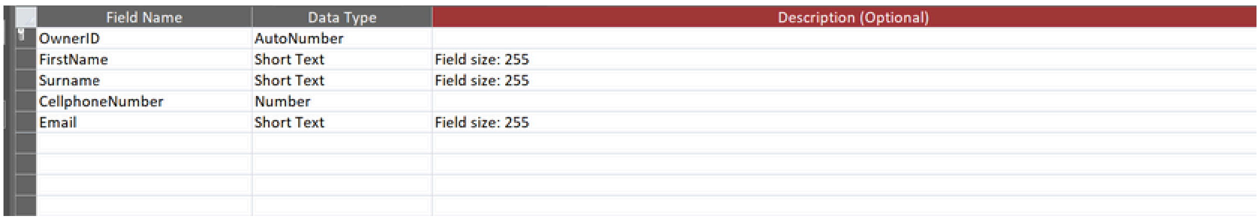
tblProperty



Field Name	Data Type	Description (Optional)
OwnerID	Number	tblOwner's primary key
AgentID	Number	tblAgent's primary key (if the home has been chosen already)
PropertyType	Short Text	Field size: 255
EstimatedMin	Currency	No decimal places
EstimatedMax	Currency	No decimal places
Bedrooms	Number	
Bathrooms	Number	
Pool	Yes/No	
Garage	Number	
SquareMeter	Number	
City	Short Text	Field size: 255
Province	Short Text	Field size: 255
PostalCode	Number	
Age	Number	
Notes	Long Text	

- PropertyID: uniquely identifies each property to be displayed on program
- OwnerID: foreign key to tblOwners, links the owner to the property
- AgentID: foreign key to tblAgents, links the owner to the property
- PropertyType: Holds the property type (apartment, townhouse, etc)
- EstimatedMin: The minimum amount the owner is looking to get for the property
- EstimatedMax: The maximum amount the owner is looking to get for the property
- Bedrooms: Holds the number of bedrooms the property has
- Bathrooms: Holds the number of bathrooms the property has
- Pool: Whether or not the property has a pool
- Garage: How many garages the property has (if any)
- SquareMeter: how many square meters the property is
- City: The city in which the property is
- Province: The province in which the property is
- PostalCode: The postal code of the property
- Age: How old the property is
- Notes: any additional information on the property

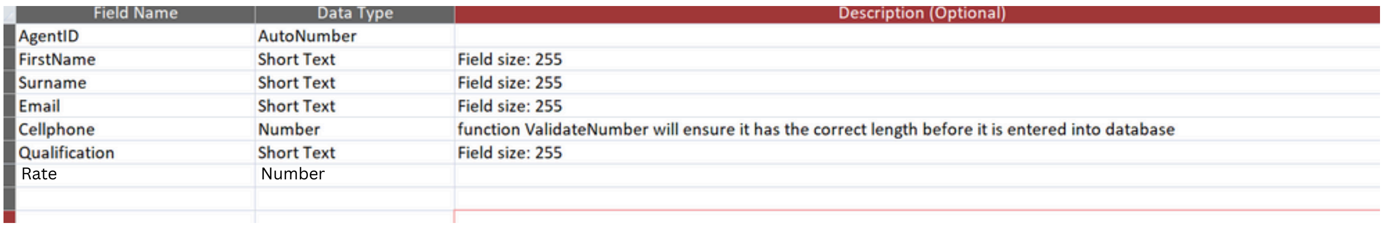
tblOwners



Field Name	Data Type	Description (Optional)
OwnerID	AutoNumber	
FirstName	Short Text	Field size: 255
Surname	Short Text	Field size: 255
CellphoneNumber	Number	
Email	Short Text	Field size: 255

- OwnerID: uniquely identifies every owner
- FirstName: holds the owner's first name
- Surname: holds the owner's last name
- CellPhoneNumber: holds the owner's phone number
- Email: holds the owner's email address

tblAgents



Field Name	Data Type	Description (Optional)
AgentID	AutoNumber	
FirstName	Short Text	Field size: 255
Surname	Short Text	Field size: 255
Email	Short Text	Field size: 255
Cellphone	Number	function ValidateNumber will ensure it has the correct length before it is entered into database
Qualification	Short Text	Field size: 255
Rate	Number	

- AgentID: uniquely identifies every agent
- FirstName: holds the agent's first name
- Surname: holds the agent's last name
- Email: holds the agent's email address
- CellPhoneNumber: holds the agent's phone number
- Qualification: holds the agent's qualification
- Rate: holds the rate (as a whole number, to be converted later) the Agent uses to calculate their commission

TASK 3: DATA DICTIONARY

3A: CLASSES AND OBJECTS

According to Embarcadero DocWiki, a class defines a structure made up of its componetnts: methods, fields and properties, with objects acting as its instances. The methods are the procedures (only make changes to the data) and functions (they return a single value). The class I will be making use of is (name): which will be used to verify the agents signing up for an account and their qualifications. It will also allow for the commission calculation (will be given as a range depending on the minimum price and maximum potential price of the household).

ATTRIBUTES

- fEmail (string)
- fPassword (string)
- fQualification (string)
- fFirstName (string)
- fSurname (string)
- fCellphone (integer)
- fRate (real)

METHODS

- Constructor create (sEmail, sPassword, sName, sSurname, sQualification : String; iCellphone : Integer; rRate: real);
  > this is to pass through all the variables needed for the following functions and procedures

- function CheckQualification : boolean;
  > this is passes the value of sQualification and checks if it matches with the needed qualification (a fixed variable). If it does, a boolean value of true is given as the result and leads to the agent's application to go through

- function CalculateMinCommission (rMinCommission : real) : real;
  > this takes the minimum price (put in by the owner) of the property and multiplies it by the rate the agent works by (fRate). It returns an integer value.

- function CalculateMaxCommission (rMaxCommission: real) : real;
  > this takes the maximum price (put in by the owner) of the property and multiplies it by the rate the agent works by (fRate). It returns an integer value.

- function toString : String;
  > This returns the details of everything for the owner to see in string format

METHODS (CONTINUED)
• procedure SetName; *> this is to make fName = sName.*
• procedure SetSurname; *> this is to make fSurname = sSurname.*
• procedure SetEmail; *> this is to make fEmail = sEmail.*
• procedure SetPassword; *> this is to make fPassword = sPassword.*
• procedure SetQualification; *> this is to make fQualification = sQualification.*
• procedure SetRate; *> this is to make fRate = rRate.*

3B: TEXTFILES AND ARRAYS

Textfiles

A textfile is a form of data storage where information (either important for inputs or put together as an output) is stored onto one page in string format. There are two textfiles for this project.

tReviews: an output textfile designed to store the messages or reviews the users of the program leave for the developer. There is very little formatting for this textfile besides starting a new line whenever a review is left behind.

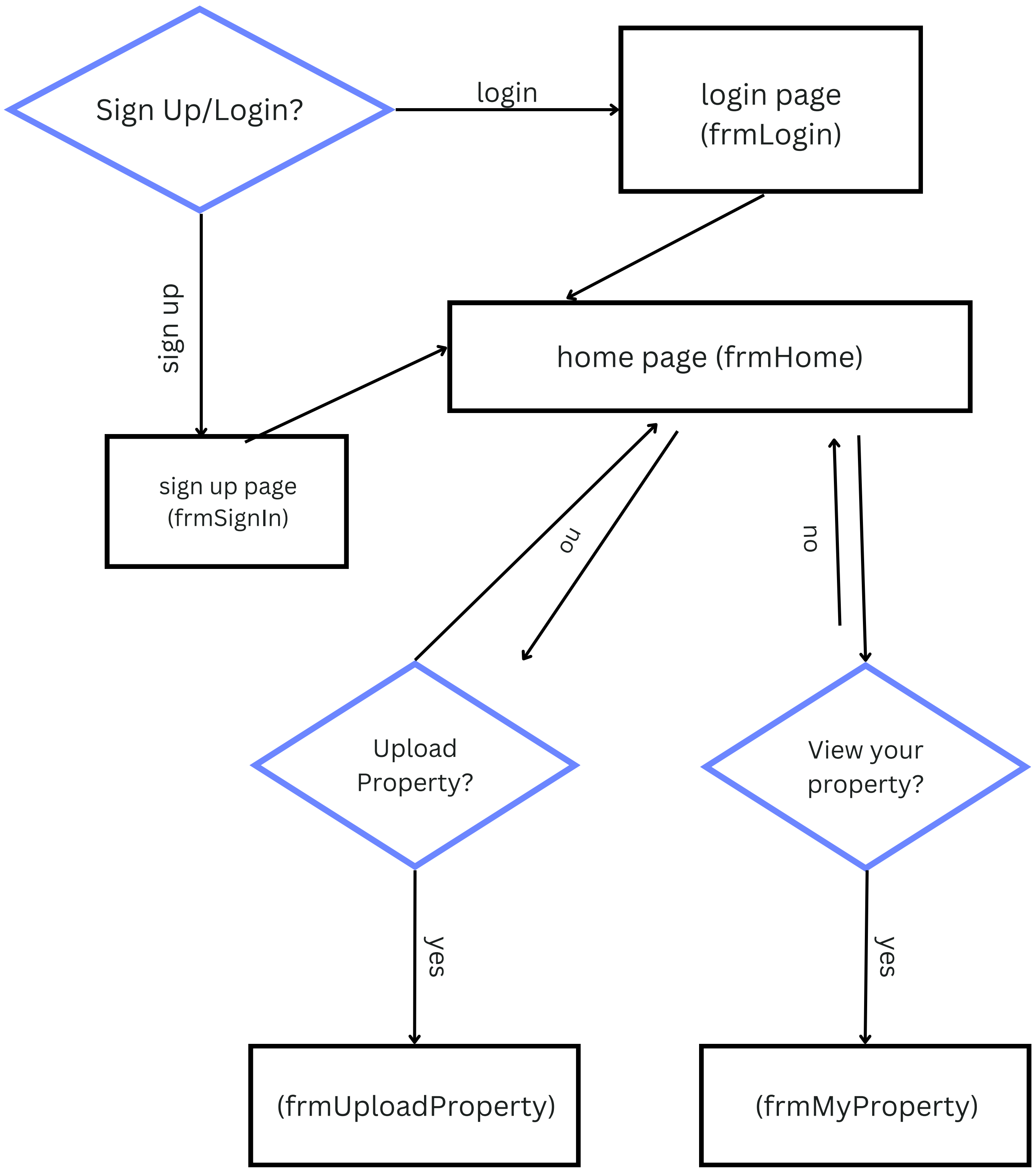
tOffers: an input textfile designed to store the details of the property that an agent is making an offer for, including the agent’s ID from dbProperty. Extensive string handling through Pos Copy Delete will be made use of to access this data.

Array

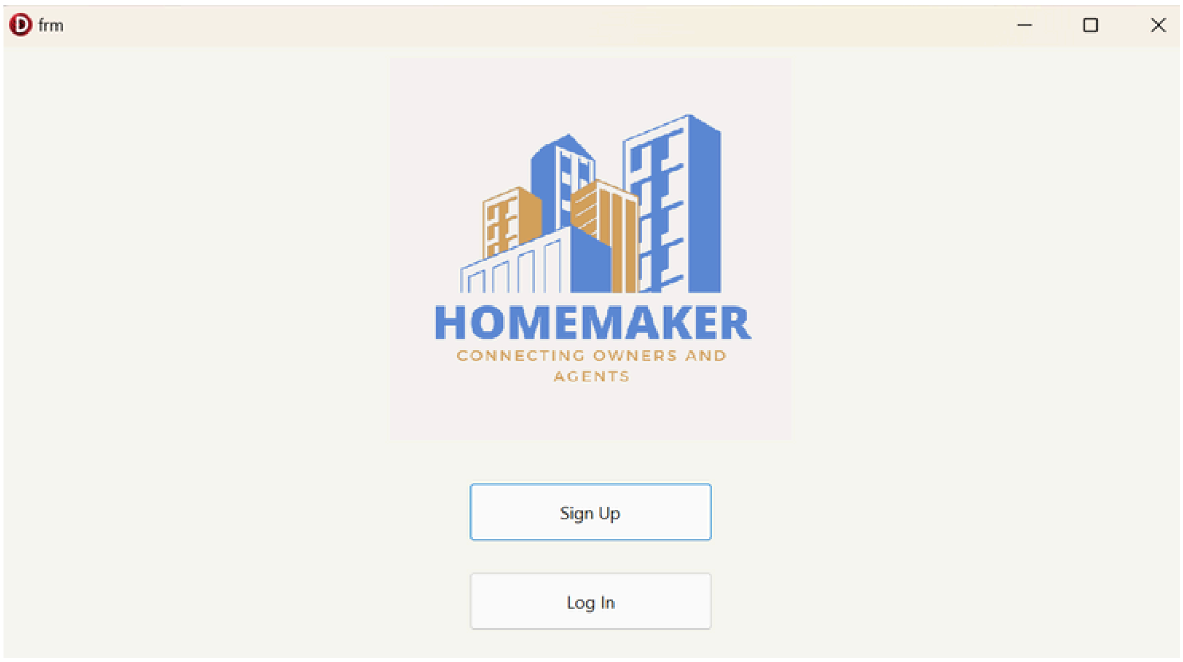
An array is defined as a variable storing more than one value of the same data type at a time. This program will be making use of **arrCity**: a **string array storing the names of the cities** to help simplify the classification of a property to be added. It will also make use of **arrProvince**: a **string array storing the names of the provinces** to help simplify the classification of a property to be added. The storing of city names ensures efficiency and limits the possible human error of spelling errors if the owner were to fill in the cities themselves.

TASK 4: NAVIGATION & GUI

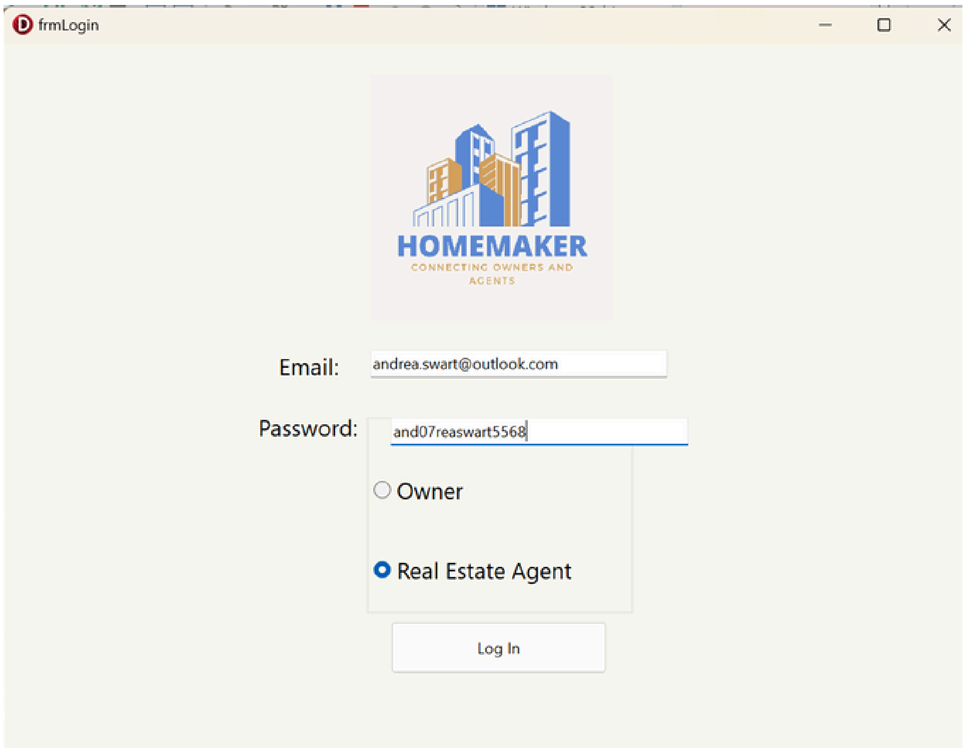
4A: NAVIGATION DIAGRAM



4B: GRAPHICAL USER INTERFACE



This is the first form the user will see when opening the program: frmOpening. From here the user (regardless of being an owner or an agent) can choose whether to go straight to logging in or signing up for a new account.

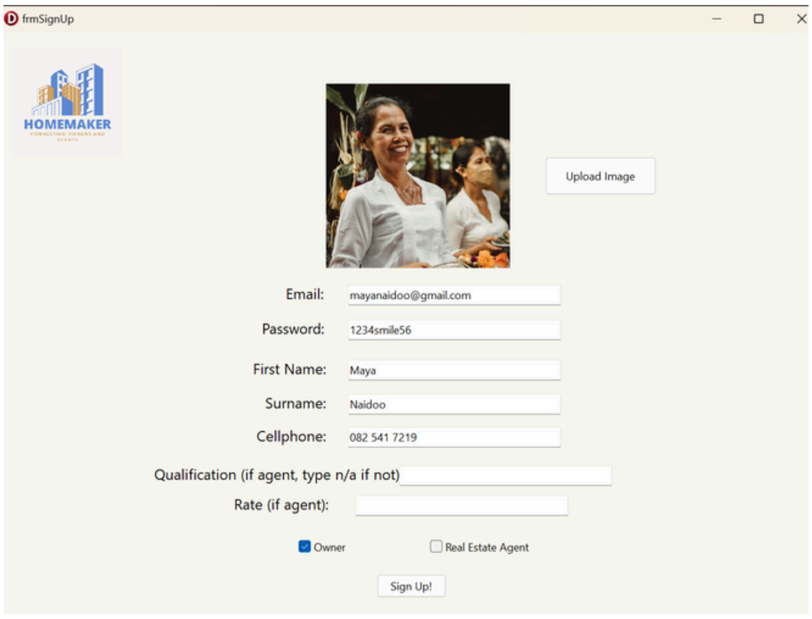


Email: andrea.swart@outlook.com

Password: and07reaswart5568

Owner [✓ Real Estate Agent]

If the user chooses to log in, they are taken to **frmLogin** where they can enter their email and password in the designated editboxes, as well as select what type of user they are on the radio group. This data then gets compared to the data found in either tblAgents or tblOwners (depending on the button of the radio group chosen) and if it doesnt match the Password and Email field of a record, an error message will pop up to notify the user that their log in was unsuccessful. If the login was successful, a message will pop up to notify the user before they are taken to the home page (frmHome).

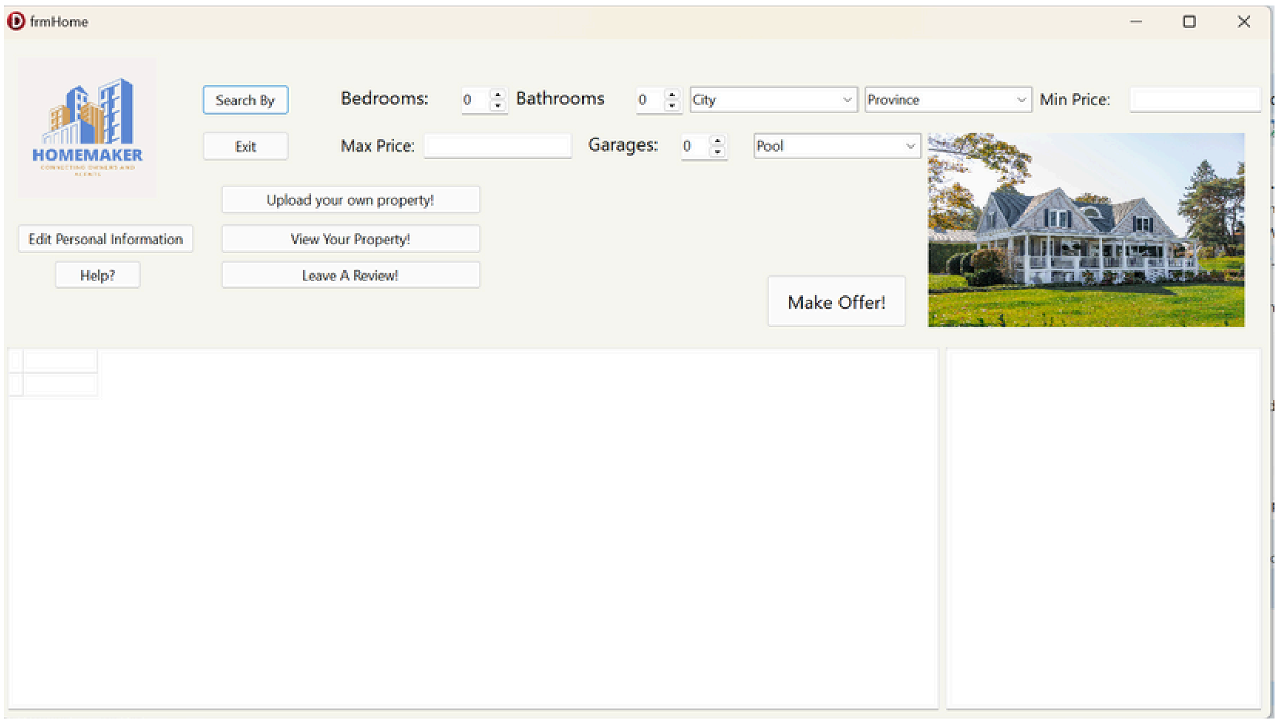


Email: mayanaidoo@gmail.com
Password: 1234smile56
First Name: Maya
Surname: Naidoo
Cellphone: 082 541 7219
Qualification (if agent, type n/a if not)
Rate (if agent):
[✓ Owner] Real Estate Agent

This is the **sign up form (frmSIgnUp)** where users can create an account for this program. Depending on whether the user signs up as an owner or an agent, a new record with the detailed information will be added either to tblOwners or tblAgents if there are no errors to be found with the data. An error message pops up if an editbox for the email, password, first name, surname or cellphone is left empty to notify the user to fill them in. An error message also pops up if neither owner or real estate agent are checked to notify the user to check one. If 'Real Estate Agent' is checked but the edit box for rate and/or qualification are left blank, an error message will pop up to notify the user of this error so they may fill in what's needed to successfully sign up.

By pressing 'Upload Image', the user is allowed to upload a picture of themselves to link with their personal information.

Further validation for cellphone will be done (for agent: function Validate Cellphone and for owner: in btnSignUp). If the signup is successful, a message will pop up to let the user know before the form closes and frmLogin opens, where they can log in with their new accounts.



Bedrooms: 0 Bathrooms 0 City Province Min Price:
Max Price: Garages: 0 Pool

After successfully logging in, the user is taken to frmHome (Home Page) where property available to real estate agents can be found. imgProperty loads the image of any property selected on the dbGrid, and details of the owner (Name, Surname, Email) will be displayed on the rich edit (redOutput). If the user logged in is an owner, they can successfully make an offer by pressing on the button "Make Offer". This will add a new line of data to tOffers, a textfile for offers made, holding the AgentID, PropertyID, OwnerID and date of the offer made. If the user making an offer is an owner, an error message will pop up, reminding them that owners cannot make offers to other owners.

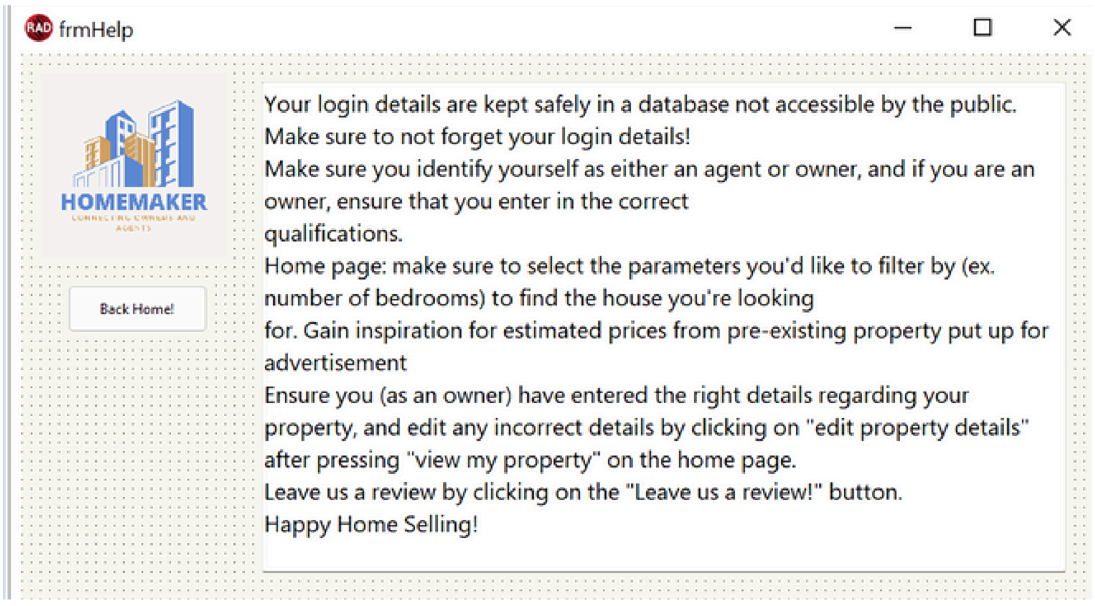
Conversely, if an agent tries to press on the "View your property" or "Upload your own property" button (btnUpload and btnViewProperty), an error message will pop up, reminding them that agents cannot upload any property or view property they do not have uploaded.

"Search By" button (btnSearch) allows the user to filter the database table tblProperty displayed on the dbGrid by using the editboxes, spinedits and/or the comboboxes of the different aspects of property. This allows agents and owners alike to browse through different properties based on the restrictions they set.

"Help" button (btnHelp) takes the user to frmHelp.

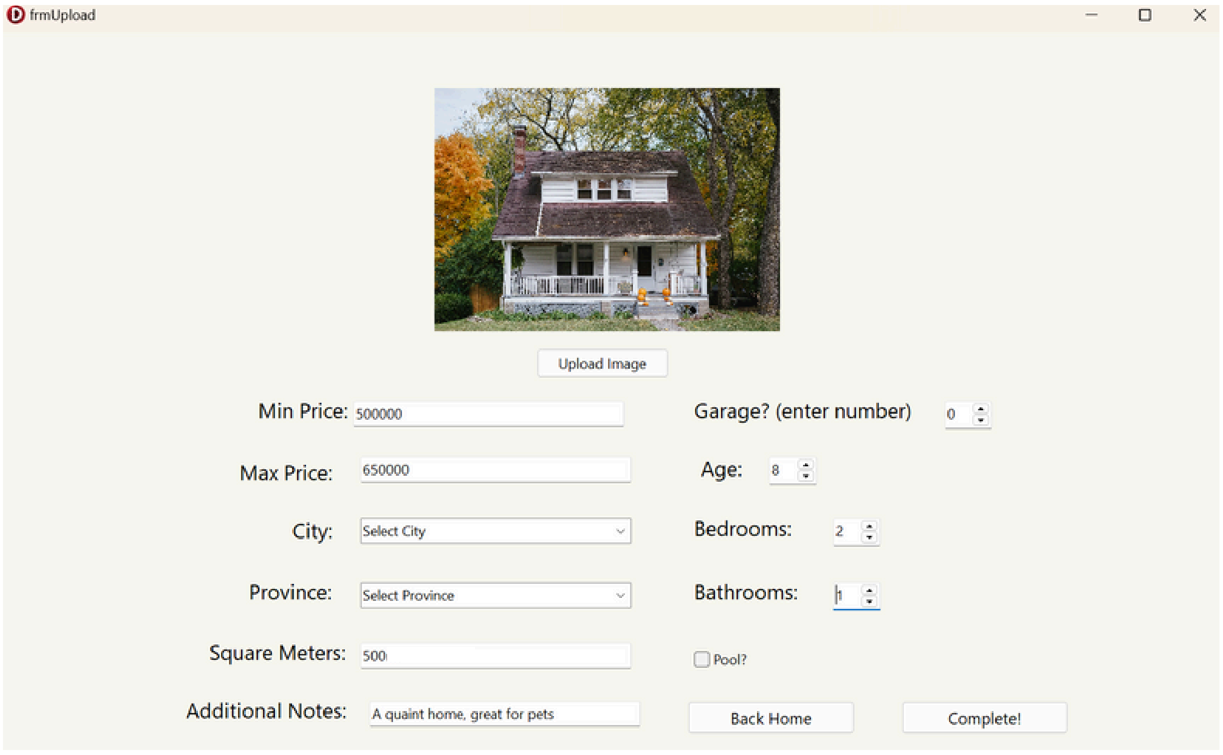
"Leave a review" button (btnReview) opens up an inputbox where the user can type out what they think about the program. This will be added to tReviews to be viewed by the developer.

"exit" button takes the user back to frmOpening.



Your login details are kept safely in a database not accessible by the public.
Make sure to not forget your login details!
Make sure you identify yourself as either an agent or owner, and if you are an owner, ensure that you enter in the correct
qualifications.
Home page: make sure to select the parameters you'd like to filter by (ex. number of bedrooms) to find the house you're looking
for. Gain inspiration for estimated prices from pre-existing property put up for advertisement
Ensure you (as an owner) have entered the right details regarding your property, and edit any incorrect details by clicking on "edit property details" after pressing "view my property" on the home page.
Leave us a review by clicking on the "Leave us a review!" button.
Happy Home Selling!

This if the Help form (frmHelp) where there are detailed steps for making use of the program in a rich edit. "Back Home" takes the user back to frmHome.


frmUpload

Upload Image

Min Price: 500000

Max Price: 650000

City: Select City

Province: Select Province

Square Meters: 500

Additional Notes: A quaint home, great for pets

Garage? (enter number) 0

Age: 8

Bedrooms: 2

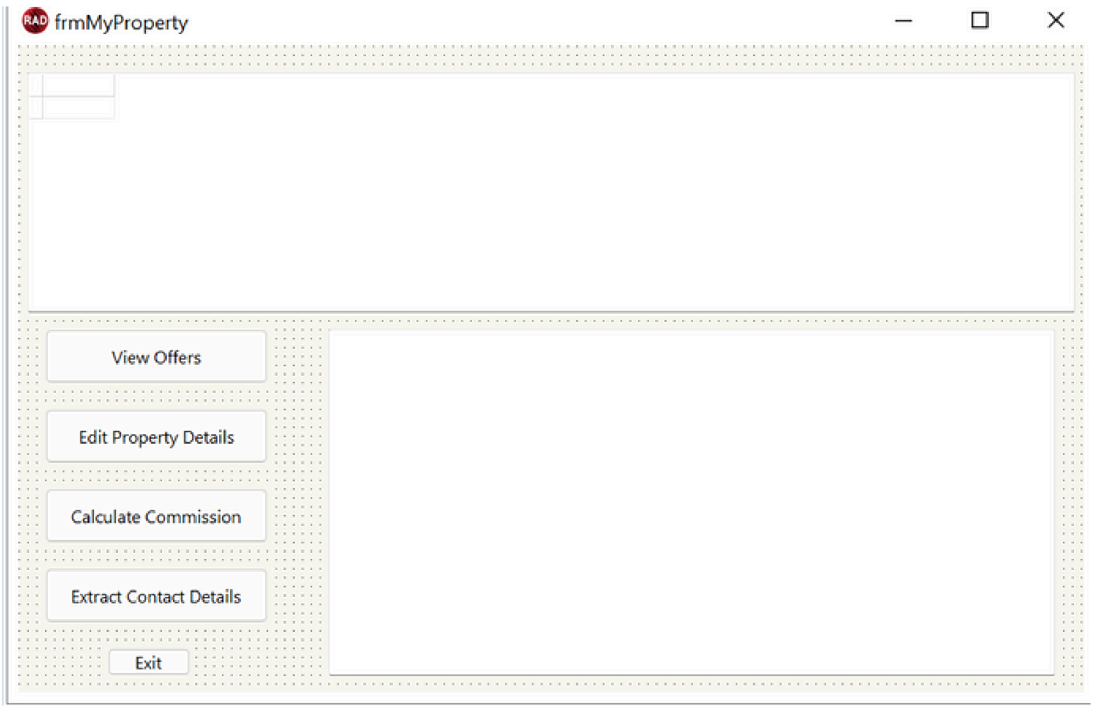
Bathrooms: 1

Pool?

Back Home

Complete!

This is **frmUpload**, where owners can upload the details of their home, along with an image of it to be added by the user via the "Upload Image" button. If successfully completed, tblProperty has a new record with this information added to it and the user will receive a popup message of confirmation. Error messages warning of empty textboxes/spnBedrooms/spnBathrooms/spnAge/empty comboboxes will be displayed appropriately.


frmMyProperty

View Offers

Edit Property Details

Calculate Commission

Extract Contact Details

Exit

This is **frmMyProperty**, where owners can view details of their previously uploaded property. View Offers allows for the extraction of data from tOffers to redOutput. Edit Property Details allows the owner to re-enter data on their property to correct any mistakes. Calculate Commission calls for the usage of the class file, specifically the functions CalculateMinCommission and CalculateMaxCommission. the dbGrid displays property with an OwnerID equal to the owner logged in. Extract contact details extracts the email and cellphone number of the agent who's made an offer.

TASK 5: IPO

OPENING PAGE

components

- btnLogin: if pressed, it closes frmOpening and opens frmLogin
- btnSIgnup: if pressed, it closes frmOpening and opens frmSignup

LOGIN PAGE

data validation

- if edtEmail.text = ‘’, message ‘Please enter your email’ will show up.
- if edtPassword.text = ‘’, message ‘Please enter your password’ will show up.

input

- sEmail (string): from editbox (edtEmail). Information from keyboard.
- sPassword (string): from editbox (edtPassword). Information from keyboard.
- bOwner (boolean): value from radio group (rgpType). Information from keyboard.
- bAgent (boolean): value from radio group (rgpType). Information from keyboard.

processing

- Validating sEmail: **sEmail needs to match with Email record** in dbProperty in order to allow login (message: ‘incorrect email’ if sEmail doesn’t match Email record)
- Validating sPassword: **sPassword needs to match with Password record** in dbProperty in order to allow login (message: ‘incorrect password’ if sEmail doesn’t match Password record)
- bOwner and bAgent **cannot BOTH** be true or false: message stating ‘Please select Owner or Agent’ will appear and user will have to check one box to login
- if bAgent is true, the password and email will be searched for through every line (pos

output

- if sEmail and sPassword match Email and Password field of either tblOwners or tblAgents, the user will be logged in. frmLogin will close and frmHome will open.

SIGNUP PAGE

data validation

- if edtEmail.text = ‘’, message ‘Please enter your email’ will show up.
- if edtPassword.text = ‘’, message ‘Please enter your password’ will show up.
- if edtName.text = ‘’, message ‘Please enter your name’ will show up.
- if edtSurname.text = ‘’, message ‘Please enter your surname’ will show up.
- if edtCellphone.text = ‘’, message ‘Please enter your cellphone number’ will show up.
- function ValidateNumber from class file
- if cbxOwner.checked = True AND edtQualification.text = ‘’, message ‘Please enter your qualification’ will show up.
- if imgProfile is empty, message ‘Please enter a picture of yourself’ will show up

input

- sEmail (string): from edit box (edtEmail). Information from keyboard.
- sPassword (string): from edit box (edtPassword). Information from keyboard.
- sName (string): from edit box (edtEmail). Information from keyboard.
- sSurname (string): from edit box (edtEmail). Information from keyboard.
- iCellphone (string): from edit box (edtEmail). Information from keyboard.
- sQualif (string): from edit box (edtEmail). Information from keyboard.
- cbxOwner (string): from edit box (edtEmail). Information from keyboard.
- cbxAgent (boolean):

processing

- if **cbxAgent** is checked, then a new record in tblAgents will be added.
  > sEmail will be added to the Email field
  > sPassword will be added to the Password field
  > sName will be added to the FirstName field
  > sSurname will be added to the Surname field
  > iCellphone will be added to the Cellphone field
  > sQualif will be added to the Qualification field
  > AgentID will be added automatically (Autonumber primary key)

- if **cbxOwner** is checked, then a new record in tblOwner will be added.
  > sEmail will be added to the Email field
  > sPassword will be added to the Password field
  > sName will be added to the FirstName field
  > sSurname will be added to the Surname field
  > iCellphone will be added to the Cellphone field
  > OwnerID will automatically be added (autonumber primary key)

output

- MessageDlg (“You have successfully signed up! Please log in!”, mbOk) will show up. Once mbOk is clicked, frmSignup will close and frmLogin will open

HOME PAGE

components

- btnSearch (button): if pressed, it applies the filters selected by the user to show property that fit under the restrictions
- btnHelp (button): if pressed, it closes frmHome and opens frmHelp
- btnExit (button): if pressed, it shows a message dialog asking if the user is sure. If ‘ok’ is pressed, the user is taken back to frmOpening
- btnViewProperty (button): if pressed, it filters to show property that matches the ID of the owner logged in
- btnEditPersonalInfo (button): if pressed, it queues a series of input boxes where the user can re-enter their personal information, changing all they wish to change
- btnUploadProperty (button): if pressed, it closes frmHome and opens frmUpload
- btnReview (button): if pressed, it opens an inputbox where the user can leave comments for the developer
- btnOffer (button): allows agents to show interest in selling the property (added to tOffers)
- dbGrid: displays tblProperty
- redOutput (rich edit): Displays information on owner property

input

- iBedroomNo(integer): from spnBedroom. Information from keyboard.
- iBathroomNo(integer): from spnBathroom. Information from keyboard.
- sCity(string): from cbxCity. Information from arrCity.
- sProvince(string): from cbxProvince. Information from arrProvince.
- iMin(integer): from edtMin (StrToInt conversion). Information from keyboard.
- iMax(integer): from edtMax (StrToInt conversion). Information from keyboard.
- iGarage(integer): from spnGarage, information from keyboard.
- bPool(boolean): from cbxPool. Yes/No selected from keyboard.
- tOffers(textfile)
- tReviews(textfile)

processing

- btnSearch
  > The database will be filtered through after the needed information is added.
- btnViewProperty
  > The form will close and frmMyProperty will open if the user is classified as an owner.
- btnMakeOffer
  > This will add a new line to tOffers (textfile) connected to tOffers, capturing the AgentID and the PropertyID of the selected property (AgentID#OwnerID#PropertyID#DateOfOffer)
- btnEditInfo
  > This will create a series of InputBoxes for the user to re-enter their information, replacing the information already stored in tblOwner or tblAgent (depending on if the user has logged in as an agent/owner)
- btnReview: this will append tReviews and save the review typed out in InputBox to tReviews

output

- After btnSearch is clicked:
  > fields in tblProperty matching the parameters selected by the user and implemmented via the filters will display in the database table. If a field is selected in particular, an image of the property linked with that field will display through the imgProperty component. More information on the owner will be listed in the redDetails (Name, Surname, Cellphone number), along with the information on the property.

- After btnHelp is clicked:
  > frmHome is closed and frmHelp is opened
- After btnExit is clicked:
  > frmHome is closed and frmLogin is opened

UPLOAD PROPERTY PAGE

components

- edtMinPrice (edit box):
- edtMaxPrice (edit box)
- edtCity (combobox). information from arrCity
- edtProvince (combobox). information from arrProvince
- edtSquareMeters (edit box)
- edtNotes (edit box)
- spnAge (spin edit)
- spnGarage (spin edit)
- spnBedrooms (spin edit)
- spnBathrooms (spin edit)
- cbxPool (check box)
- btnUploadImage (button)
- btnBackHome (button)
- btnComplete (button)

input

- iMinPrice (integer): taken from edtMinPrice, information from keyboard.
- iMaxPrice (integer): taken from edtMaxPrice, information from keyboard.
- sCity (String): taken from code, information from arrCity
- sProvince(String): taken from code, information from arrProvince
- iSquareMeters (string): taken from edtSquareMeters, information from keyboard.
- sNotes (String): taken from edtNotes, information from keyboard.
- iAge (Integer): taken from spnAge, information from keyboard.
- iGarage (Integer): taken from spnGarage, information from keyboard.
- iBathrooms (Integer): taken from spnBathrooms, information from keyboard.
- iBedrooms (Integer): taken from spnBedrooms, information from keyboard.
- bPool (boolean): taken from cbxPool, information from keyboard.

data validation

- if iMinPrice, iMaxPrice, iSquareMeters, sNotes = ‘’ then Showmessage(‘Please enter information in text boxes’)
- if iAge, iBathrooms or iBedrooms = ‘’ then Showmessage(“Please enter numbers into spin edits”)

processing

- iMinPrice is stored into MinPrice field in dbProperty
- iMaxPrice is stored into MaxPrice field in dbProperty
- sCity is stored into City field in dbProperty
- sProvince is stored into Province field in dbProperty
- iSquareMeters is stored into SquareMeters field in dbProperty
- sNotes is stored into Notes field in dbProperty
- iAge is stored into Age field in dbProperty
- iBathrooms is stored into Bathrooms field in dbProperty
- iBedrooms is stored into Bedrooms field in dbProperty
- if bPool = True, Pool field is made true/’yes’ in dbProperty

output

- a new record of the newly listed property is to be found in tblProperty once btnComplete is pressed. After pressing it, a Showmessage(‘Your property has been listed! Good luck!’) will display.
- if btnHome is pressed, frmUploadProperty will close and frmHome will open

MY PROPERTY FORM (FOR OWNERS)

components

- btnViewOffers (button): when pressed, it allows for the string handling of tOffers to display the offers linked with the OwnerID
- btnEditProperty (button): allows for the editing of details of the selected property
- dbgMyProperty (database grid): shows all the property in tblProperty entered under the same OwnerID
- btnCalculateCommission (button): when clicked, it calls for functionCalculateCommision from clsAgents
- btnExtractDetails (button): when pressed, it takes out the details of the agent making an offer and formats them into the redOutput

input

- tOffers (textfile): has information regarding the property offers have been made on as well as the real estate agent making them
- iMinPrice(integer): this is taken from MinPrice of tblProperty (displayed on dbgMyProperty)
- iMaxPrice(integer): this is taken from MaxPrice of tblProperty (displayed on dbgMyProperty)
- rRate(real): taken from the Rate field of tblAgent divided by 100.
- sName, sSurname, sEmail, sQualification(string): these are taken from the FirstName, Surname, Email and Qualification fields of tblAgents
- iCellphone(integer): taken from the Cellphone field of tblAgents

processing

- btnViewOffers:
  > The database will be filtered according to the information stored in the textfile tOffers. Extensive Pos Copy Delete handling will be used to extract this data needed for the filtering.
- btnEditPropertyDetails:
  > data given to several edit boxes will be taken and replace the information of the selected property with it.
- btnCalculateCommission:
  > this will call on functions CalculateMinCommission and CalculateMaxCommission of clsAgents and display the results in redOutput (after formatting (FloatToStr) is done)
- btnExtractContactDetails:
  > this will call on function toString from clsAgents to be displayed into redOutput

output

- closes frmMyProperty and opens frmHome if btnClose is pressed
- Details of Agent's name, surname, email and cellphone number will be displayed in redOutput if btnExtractContactDetails is clicked
- The minimum commission price and maximum commission price the agent expects for selling the property will be displayed in redOutput if btnCalculateCommission is pressed
- The database will be filtered if btnViewOffers is pressed
- The record of the edited property will be displayed if btnEditProperty is clicked

BIBLIOGRAPHY

- https://www.delphibasics.co.uk/
- ChatGPT (for Array usage)
- Xhanti Luke’s Grade 12 PAT
- https://learndelphi.org/