WADE SMITH

PROG POE

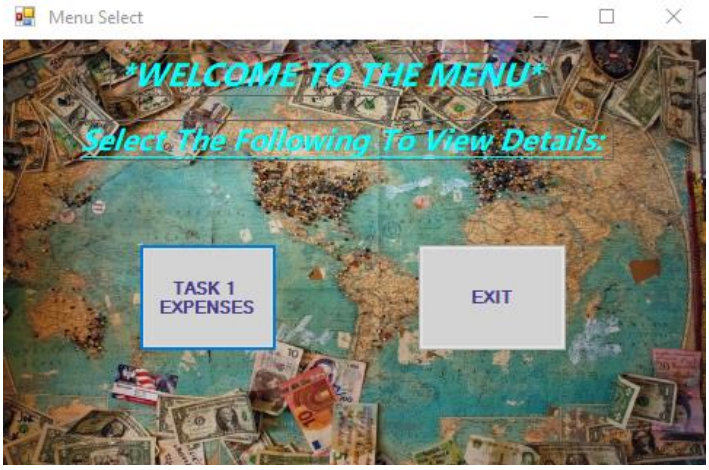
10117793

DISD

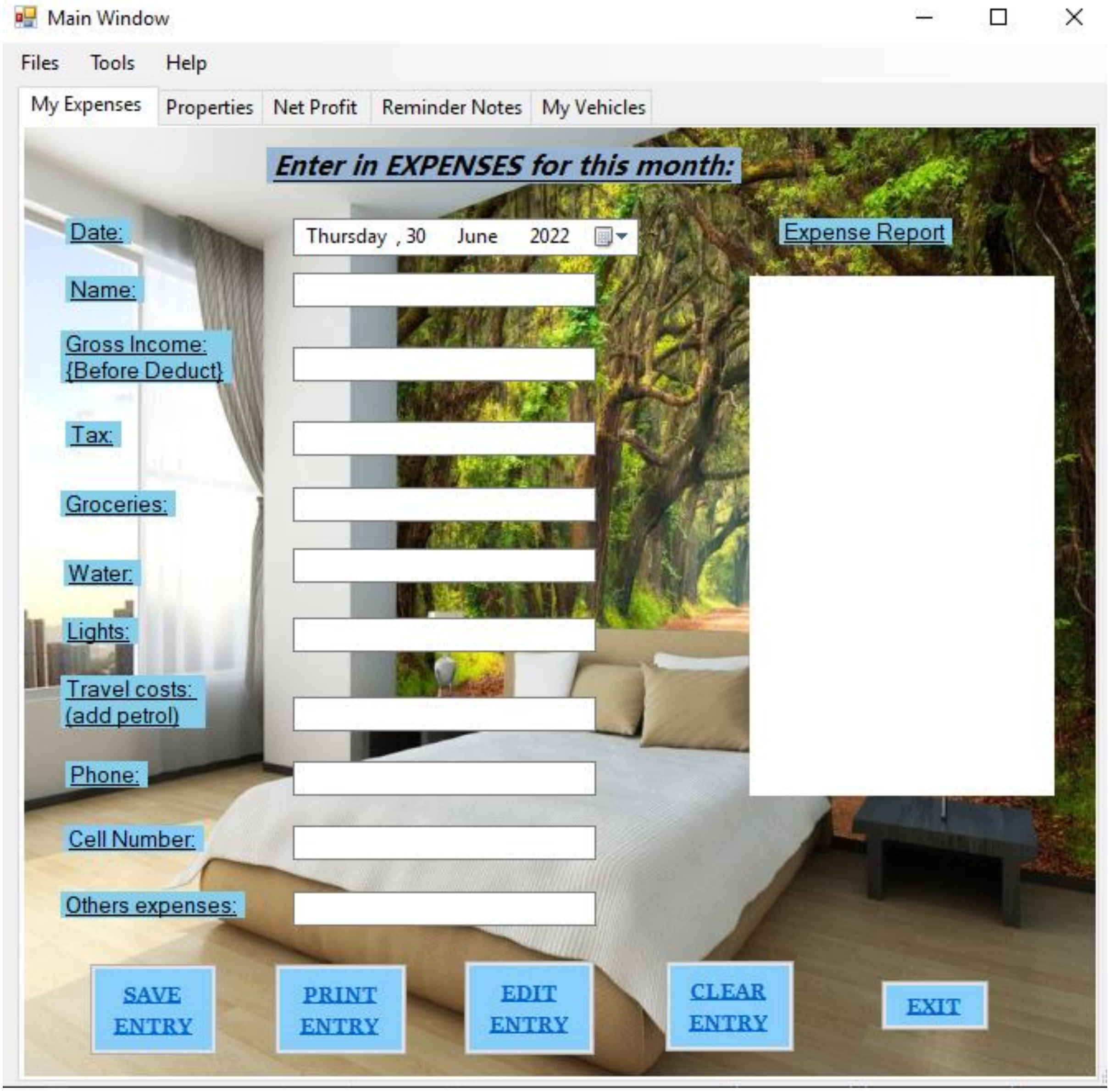
PROG POE

Reece Wanvig

I, WADE SMITH*hereby declare that I did not plagiarise any content of this assignment and that this is my own work.*

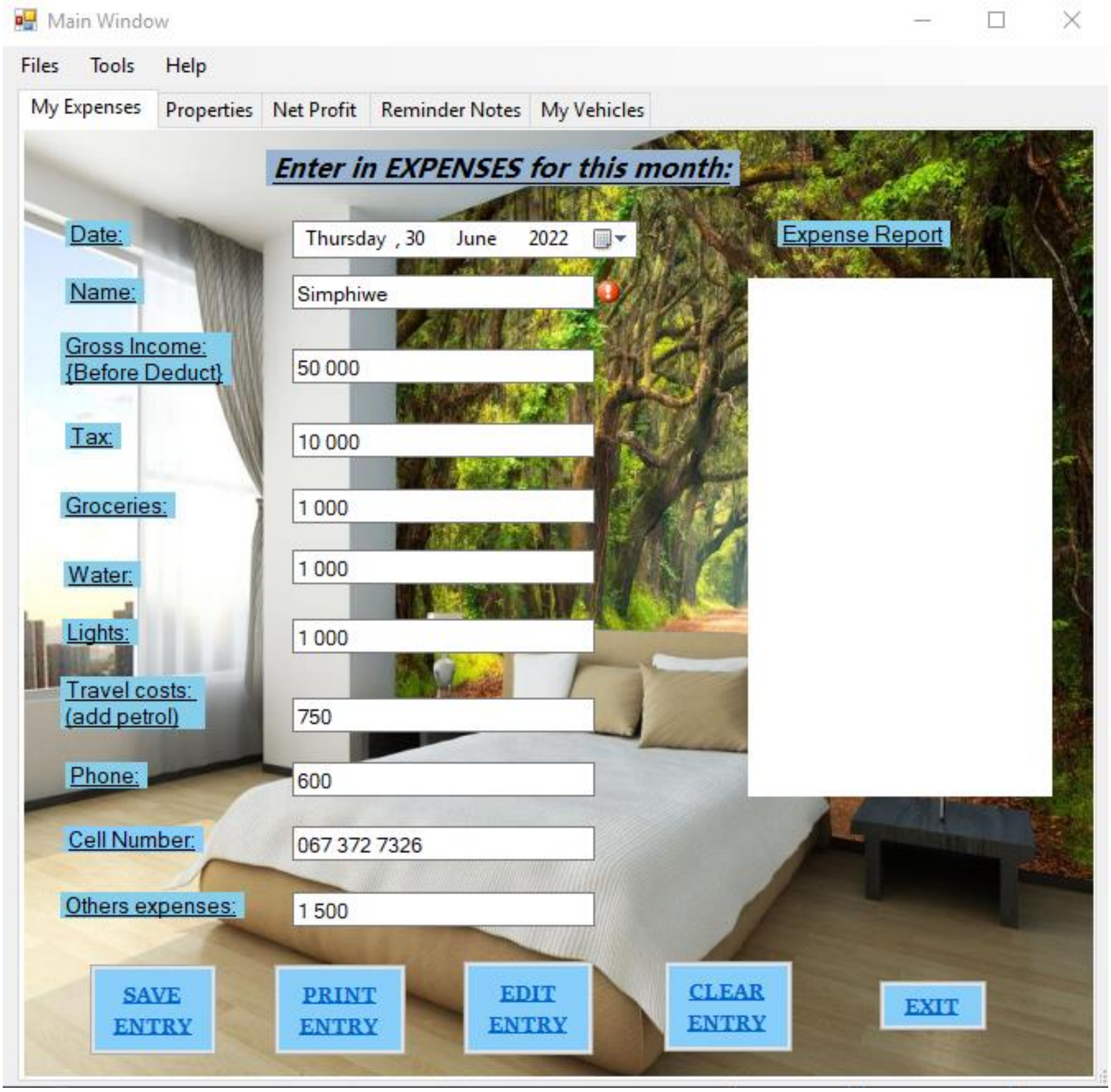


When the program is started up the welcome page introduces two options these two options being the expenses that were asked for in task 1 and a exit button to exit the program once task expenses are selected the following expenses tab appears



Thursday , 30 June 2022

This tab allows you to enter the expenses that Simphiwe has and all the deductions of his salary that are required to be entered in the fields. Once save entry button is clicked the stored data will then be saved.



Date: Thursday , 30 June 2022

Name: Simphiwe

Gross Income: {Before Deduct} 50 000

Tax: 10 000

Groceries: 1 000

Water: 1 000

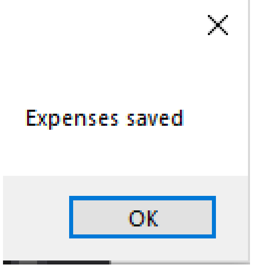
Lights: 1 000

Travel costs: (add petrol) 750

Phone: 600

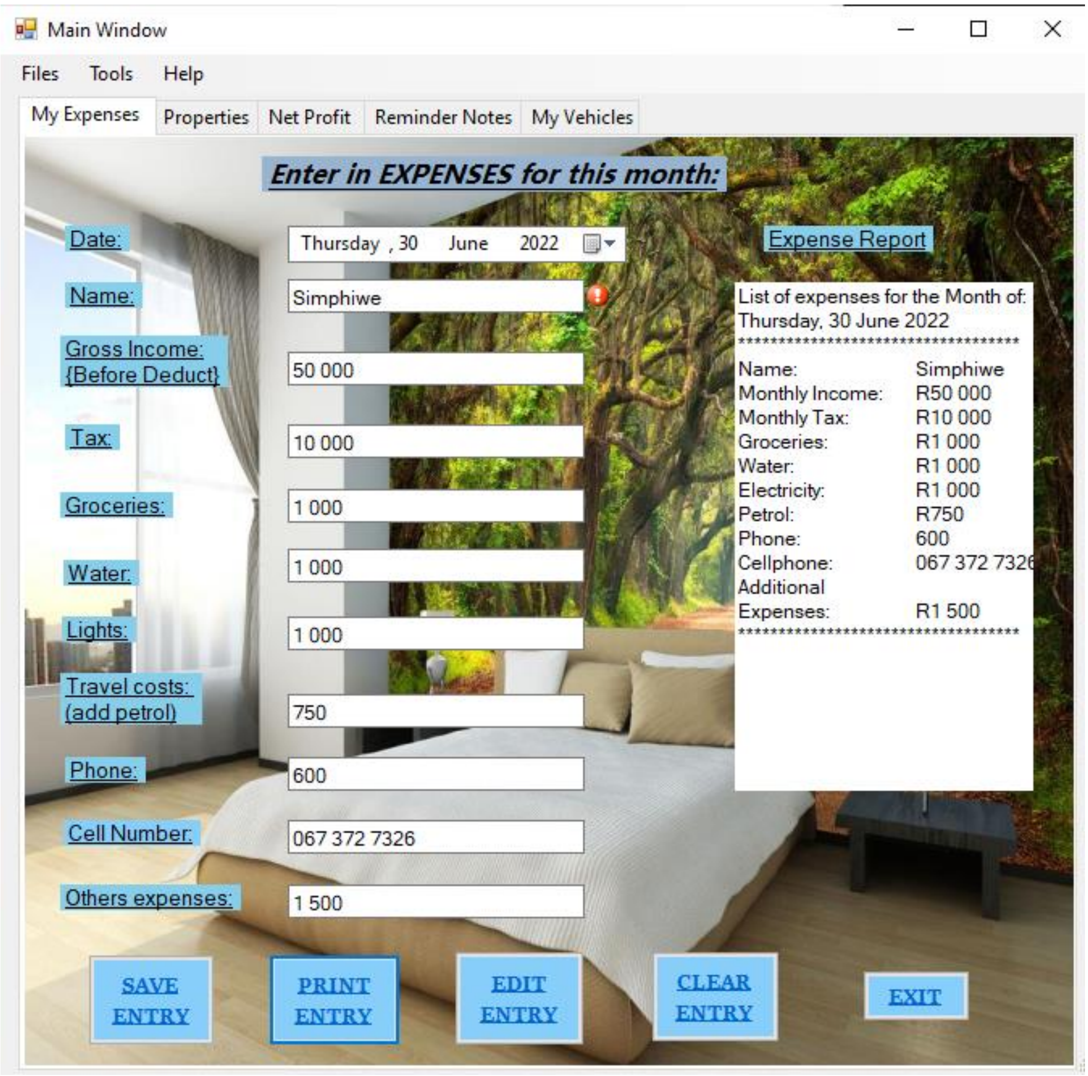
Cell Number: 067 372 7326

Others expenses: 1 500



Expenses saved

Once expenses are stored the user can then select to print the expenses to view the correct amounts and values have been assigned to the correct variables and a list is printed in the expense report showing all the values entered into expenses



Enter in EXPENSES for this month:

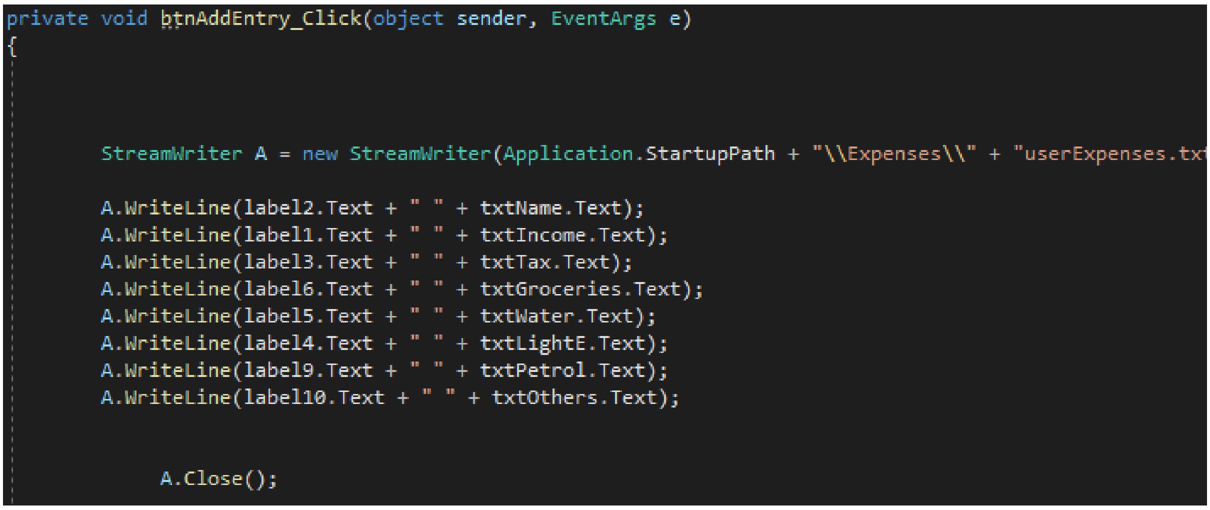
Date:	Thursday , 30 June 2022
Name:	Simphiwe
Gross Income: {Before Deduct}	50 000
Tax:	10 000
Groceries:	1 000
Water:	1 000
Lights:	1 000
Travel costs: (add petrol)	750
Phone:	600
Cell Number:	067 372 7326
Others expenses:	1 500

Expense Report

```
List of expenses for the Month of:
Thursday, 30 June 2022
***********************************
Name:              Simphiwe
Monthly Income:    R50 000
Monthly Tax:       R10 000
Groceries:         R1 000
Water:             R1 000
Electricity:       R1 000
Petrol:            R750
Phone:             600
Cellphone:         067 372 7326
Additional
Expenses:          R1 500
***********************************
```

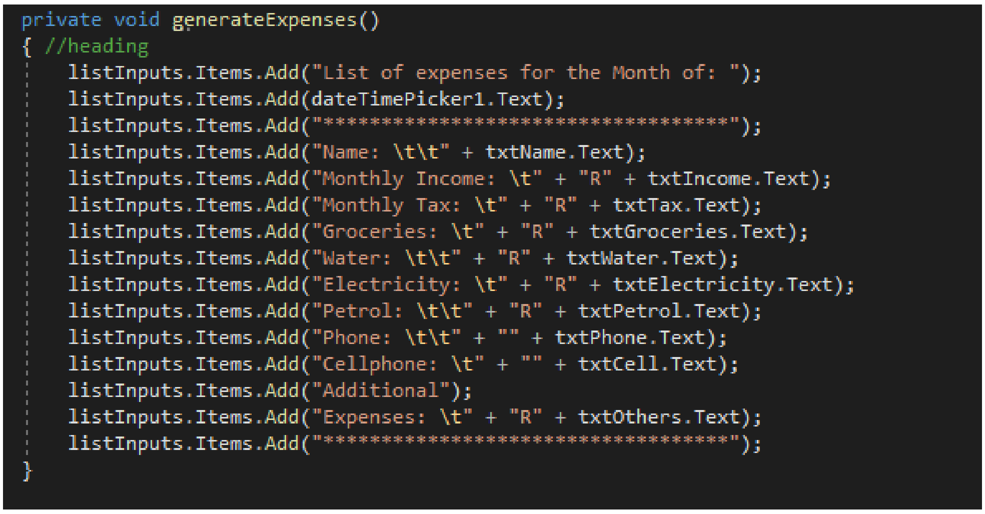
The user may then select whether to edit any incorrect entry or clear the entries to re-enter new values to the application.

The user may also exit from this tab using the exit button.



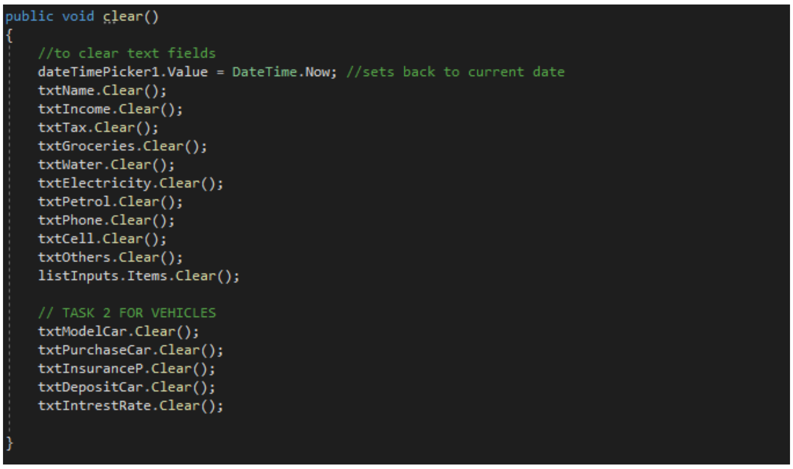
```
private void btnAddEntry_Click(object sender, EventArgs e)
{



        StreamWriter A = new StreamWriter(Application.StartupPath + "\\Expenses\\" + "userExpenses.tx

        A.WriteLine(label2.Text + " " + txtName.Text);
        A.WriteLine(label1.Text + " " + txtIncome.Text);
        A.WriteLine(label3.Text + " " + txtTax.Text);
        A.WriteLine(label6.Text + " " + txtGroceries.Text);
        A.WriteLine(label5.Text + " " + txtWater.Text);
        A.WriteLine(label4.Text + " " + txtLightE.Text);
        A.WriteLine(label9.Text + " " + txtPetrol.Text);
        A.WriteLine(label10.Text + " " + txtOthers.Text);


             A.Close();
```

All expenses are stored here for further use.



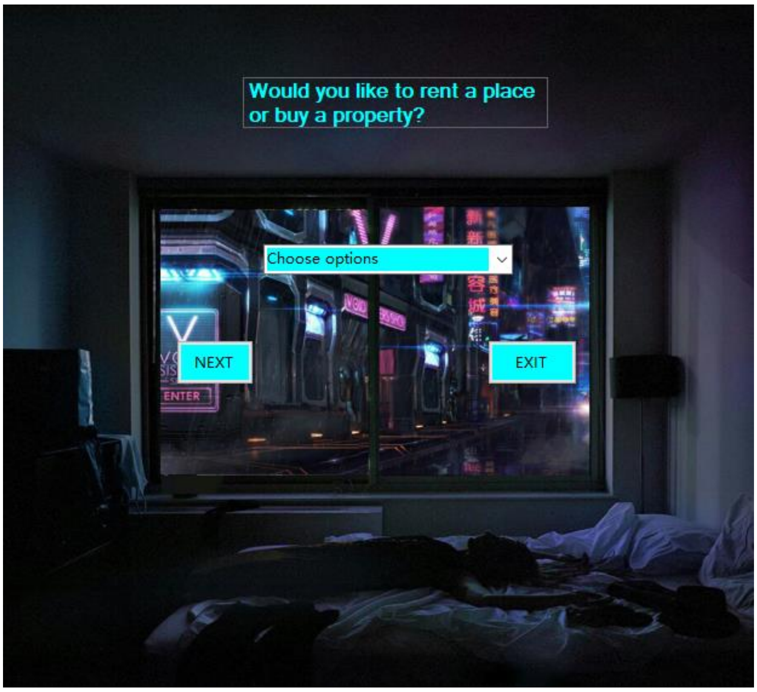
```
private void generateExpenses()
{ //heading
    listInputs.Items.Add("List of expenses for the Month of: ");
    listInputs.Items.Add(dateTimePicker1.Text);
    listInputs.Items.Add("***********************************");
    listInputs.Items.Add("Name: \t\t" + txtName.Text);
    listInputs.Items.Add("Monthly Income: \t" + "R" + txtIncome.Text);
    listInputs.Items.Add("Monthly Tax: \t" + "R" + txtTax.Text);
    listInputs.Items.Add("Groceries: \t" + "R" + txtGroceries.Text);
    listInputs.Items.Add("Water: \t\t" + "R" + txtWater.Text);
    listInputs.Items.Add("Electricity: \t" + "R" + txtElectricity.Text);
    listInputs.Items.Add("Petrol: \t\t" + "R" + txtPetrol.Text);
    listInputs.Items.Add("Phone: \t\t" + "" + txtPhone.Text);
    listInputs.Items.Add("Cellphone: \t" + "" + txtCell.Text);
    listInputs.Items.Add("Additional");
    listInputs.Items.Add("Expenses: \t" + "R" + txtOthers.Text);
    listInputs.Items.Add("***********************************");
}
```

The values that were entered are printed out to the text box.

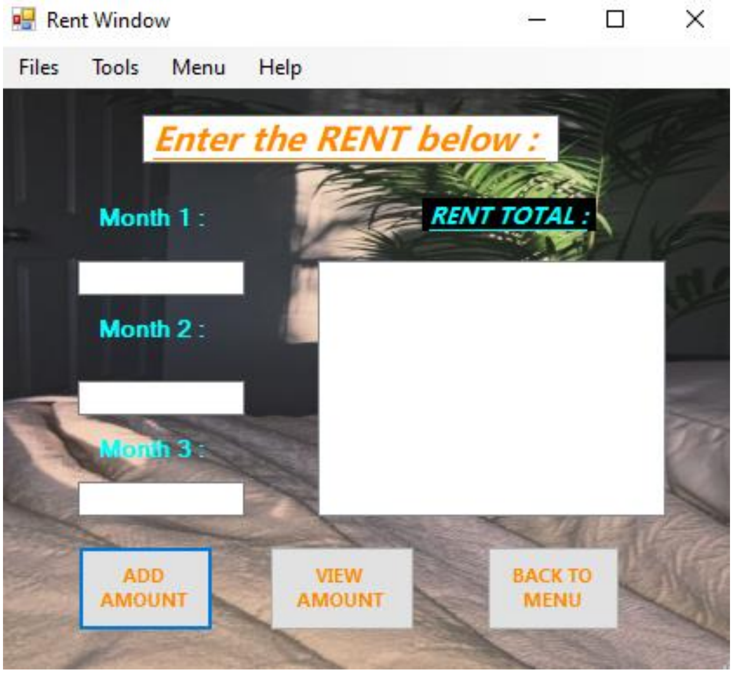


```
public void clear()
{
    //to clear text fields
    dateTimePicker1.Value = DateTime.Now; //sets back to current date
    txtName.Clear();
    txtIncome.Clear();
    txtTax.Clear();
    txtGroceries.Clear();
    txtWater.Clear();
    txtElectricity.Clear();
    txtPetrol.Clear();
    txtPhone.Clear();
    txtCell.Clear();
    txtOthers.Clear();
    listInputs.Items.Clear();

    // TASK 2 FOR VEHICLES
    txtModelCar.Clear();
    txtPurchaseCar.Clear();
    txtInsuranceP.Clear();
    txtDepositCar.Clear();
    txtIntrestRate.Clear();

}
```

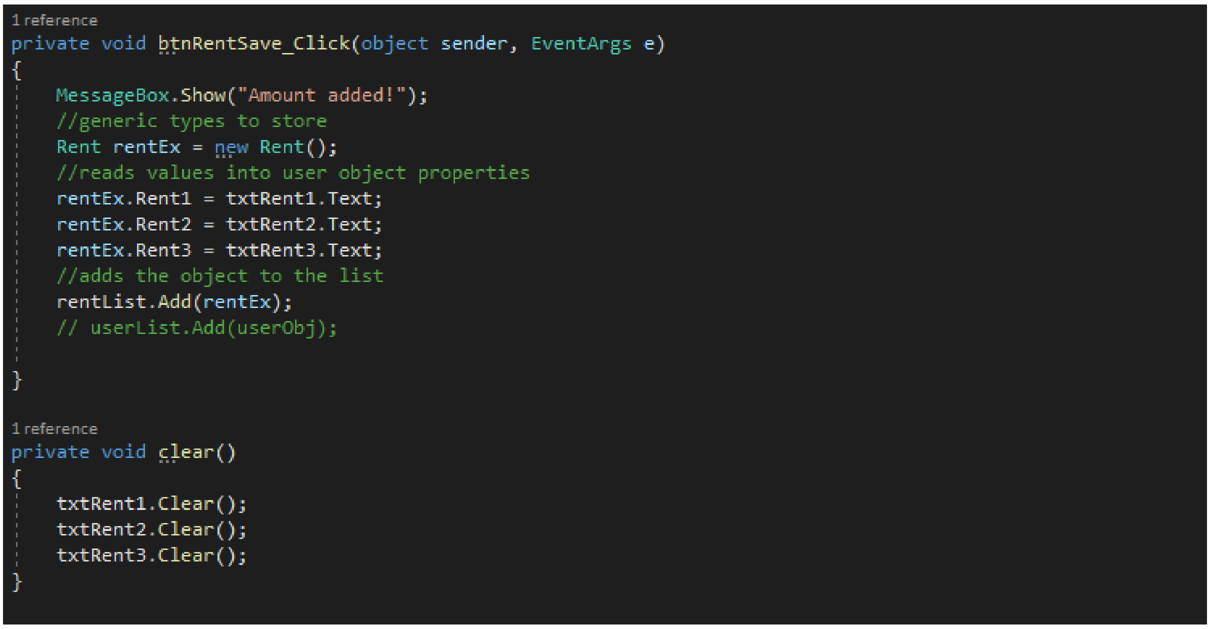
These methods clear all the text fields once the variables have been used and allows for the user to then enter new values if need be. Moving forward the entered values can be used to calculate all the expenses that the user has entered the next tab allows for the user to either select to rent or buy a property depending what they want to do.



Once choosing whether to rent or buy a property the user will be prompted to fill the required text fields depending on which selection was made this then allows the user to enter in the expenses for the property that they will rent or buy.

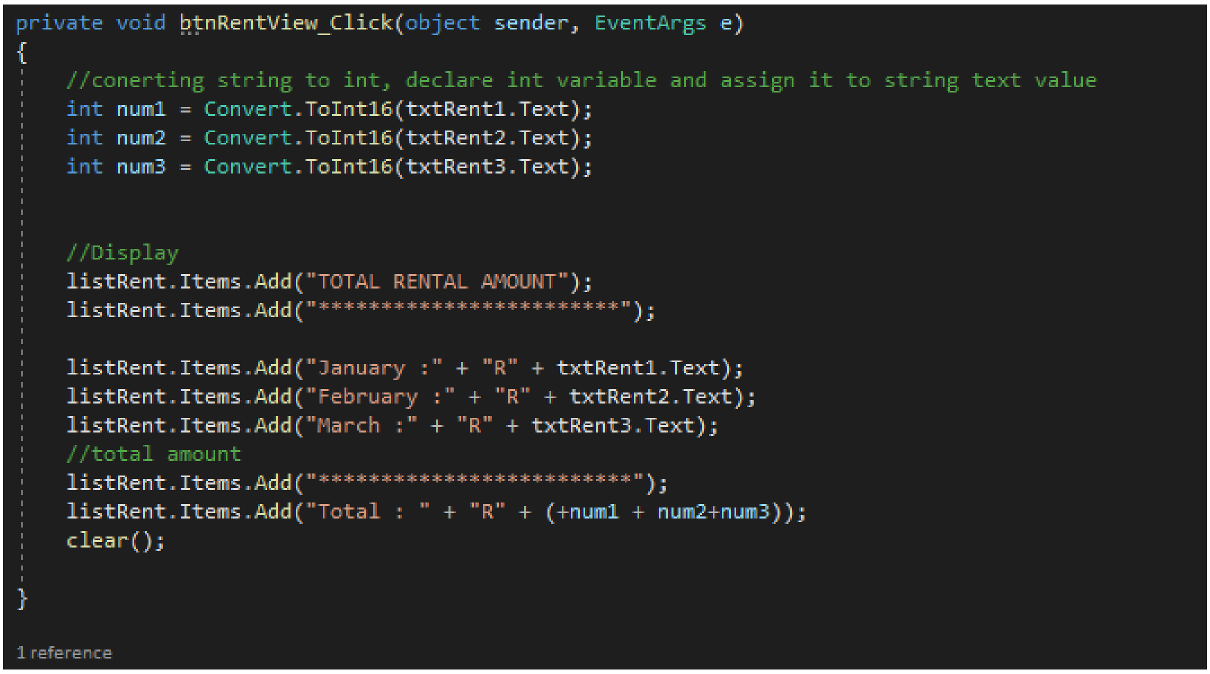


When choosing to rent a property the rent amount is required for each month and the amounts will be calculated to allow the user to see the amount that they will need to have for the property.



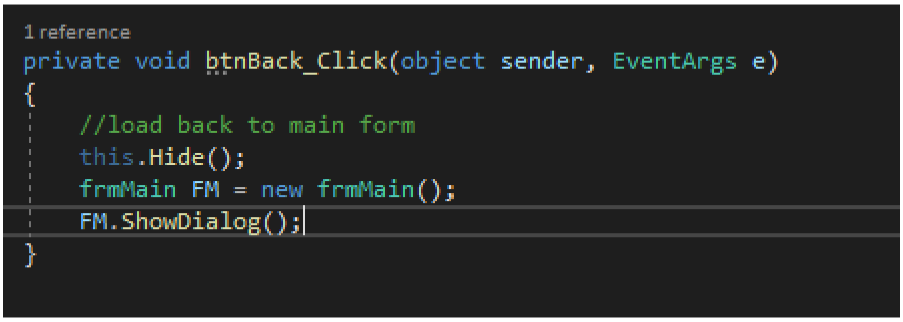
```
1 reference
private void btnRentSave_Click(object sender, EventArgs e)
{
    MessageBox.Show("Amount added!");
    //generic types to store
    Rent rentEx = new Rent();
    //reads values into user object properties
    rentEx.Rent1 = txtRent1.Text;
    rentEx.Rent2 = txtRent2.Text;
    rentEx.Rent3 = txtRent3.Text;
    //adds the object to the list
    rentList.Add(rentEx);
    // userList.Add(userObj);

}

1 reference
private void clear()
{
    txtRent1.Clear();
    txtRent2.Clear();
    txtRent3.Clear();
}
```

Storing all values added for the required rent amount are stored here for later use to use for the calculation of the rental property and whether the user can afford to rent the property or not, there is also a clear button if the user has to re-enter any information that was given.

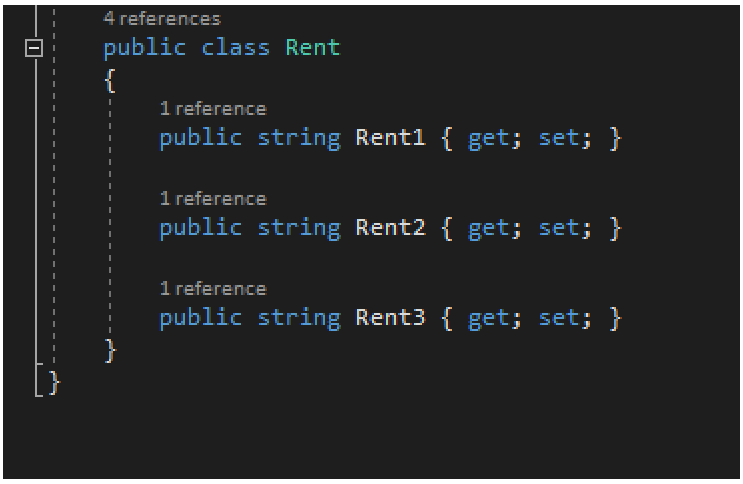


```
private void btnRentView_Click(object sender, EventArgs e)
{
    //conerting string to int, declare int variable and assign it to string text value
    int num1 = Convert.ToInt16(txtRent1.Text);
    int num2 = Convert.ToInt16(txtRent2.Text);
    int num3 = Convert.ToInt16(txtRent3.Text);


    //Display
    listRent.Items.Add("TOTAL RENTAL AMOUNT");
    listRent.Items.Add("************************");

    listRent.Items.Add("January :" + "R" + txtRent1.Text);
    listRent.Items.Add("February :" + "R" + txtRent2.Text);
    listRent.Items.Add("March :" + "R" + txtRent3.Text);
    //total amount
    listRent.Items.Add("*************************");
    listRent.Items.Add("Total : " + "R" + (+num1 + num2+num3));
    clear();

}

1 reference
```

The total rent amount will then be calculated and displayed for the user to see. There is also a back button which allows you to return back to the main menu to restart the process.



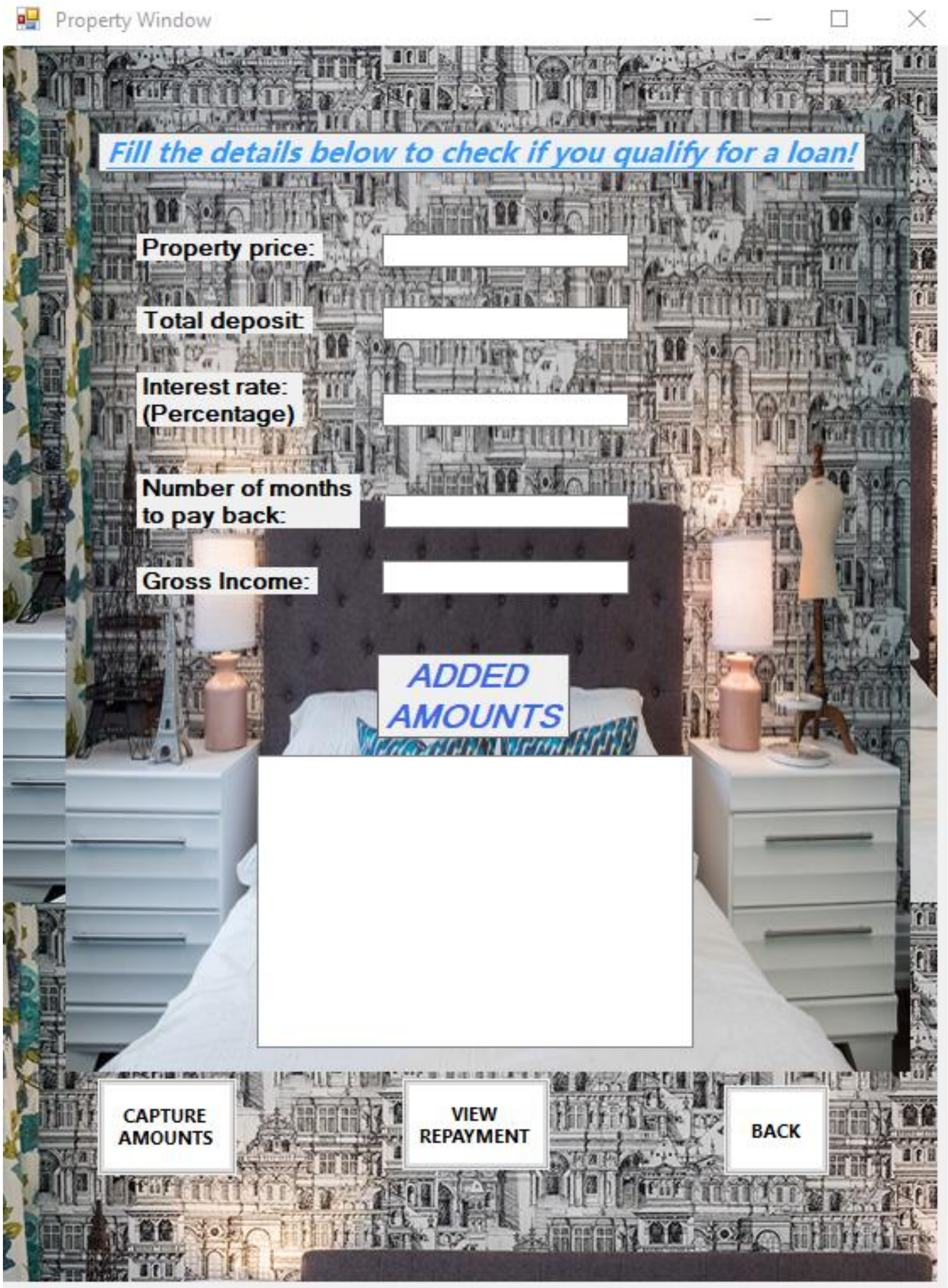
```
1 reference
private void btnBack_Click(object sender, EventArgs e)
{
    //load back to main form
    this.Hide();
    frmMain FM = new frmMain();
    FM.ShowDialog();
}
```



```
4 references
public class Rent
{
    1 reference
    public string Rent1 { get; set; }

    1 reference
    public string Rent2 { get; set; }

    1 reference
    public string Rent3 { get; set; }
}
}
```

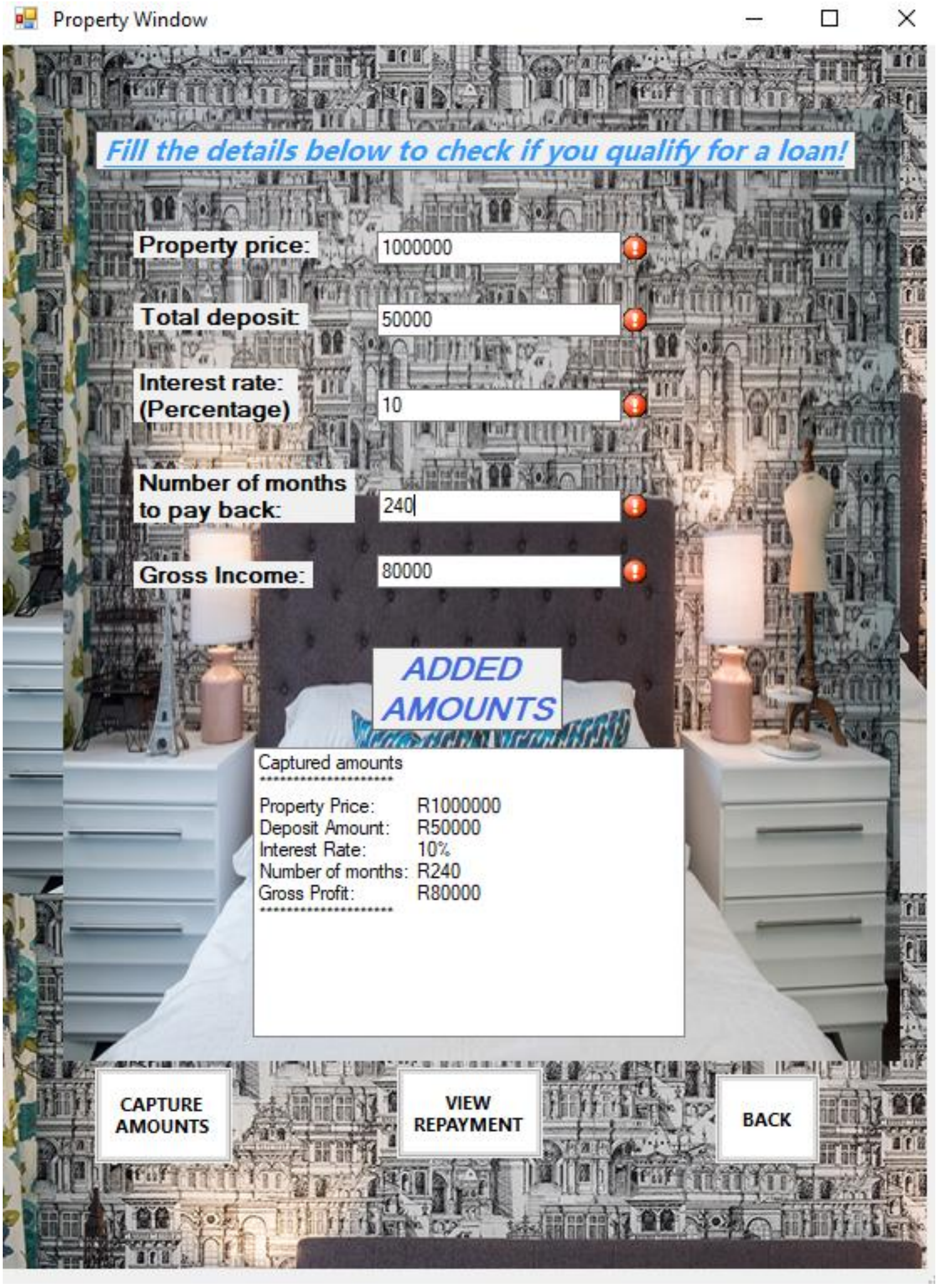
All the amounts that are entered for the rent are stored in the Rent gets and sets to be used for other expense calculations

If purchase property is selected a property tab will prompt you to enter in the values to purchase a property this allows the user to enter in values that will be calculated to see if the user is applicable to a home loan which will only be allowed if the users gross salary after deductions is more than 75% of the home loan.



Fill the details below to check if you qualify for a loan!

Once purchase property is selected the user has to enter in the values to calculate the amount that a home loan will be and the repayment that will be need to repay the loan.



Fill the details below to check if you qualify for a loan!

Property price: 1000000

Total deposit: 50000

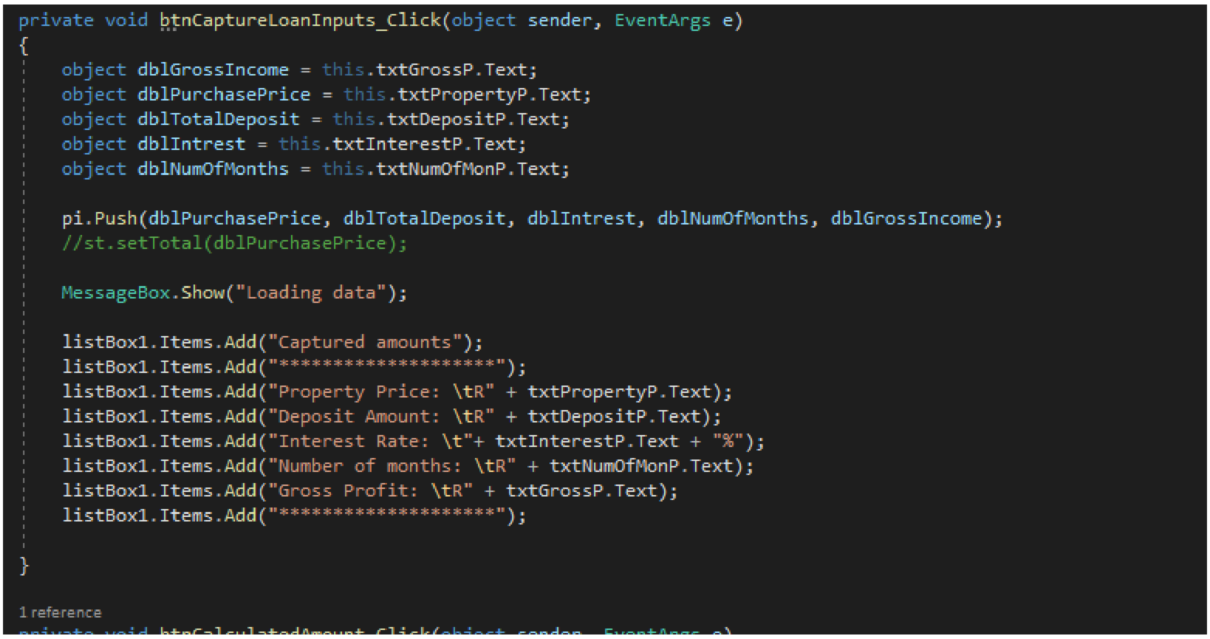
Interest rate: (Percentage) 10

Number of months to pay back: 240

Gross Income: 80000

ADDED AMOUNTS

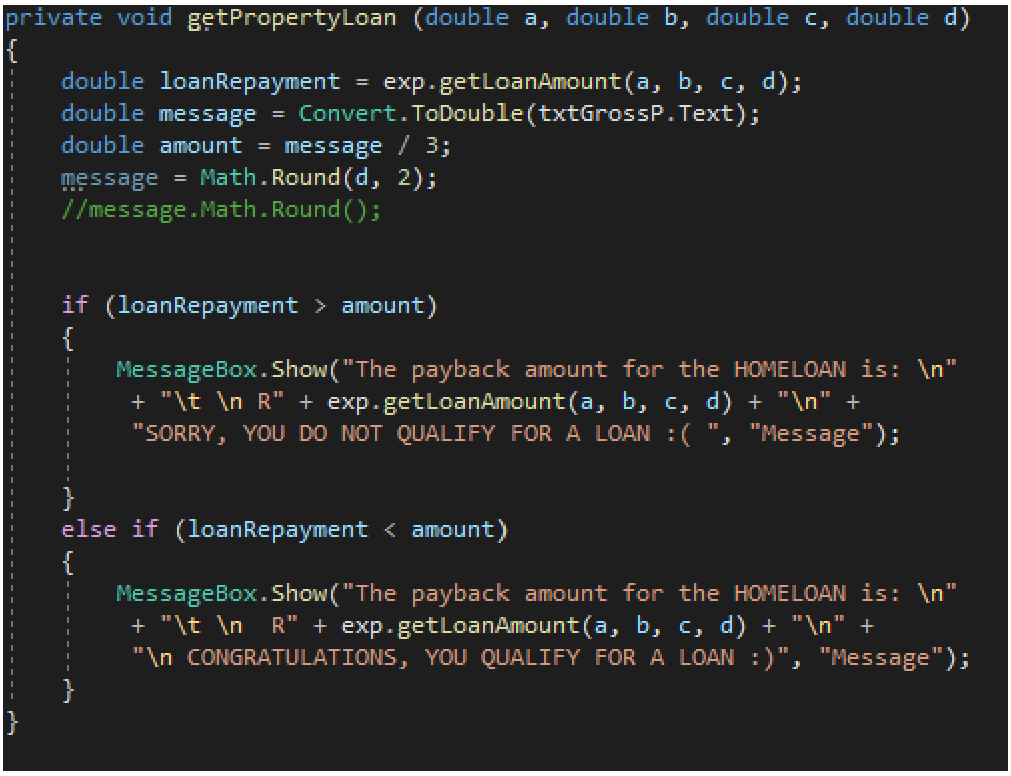
```
Captured amounts
********************
Property Price:     R1000000
Deposit Amount:     R50000
Interest Rate:      10%
Number of months:   R240
Gross Profit:       R80000
********************
```



```
private void btnCaptureLoanInputs_Click(object sender, EventArgs e)
{
    object dblGrossIncome = this.txtGrossP.Text;
    object dblPurchasePrice = this.txtPropertyP.Text;
    object dblTotalDeposit = this.txtDepositP.Text;
    object dblIntrest = this.txtInterestP.Text;
    object dblNumOfMonths = this.txtNumOfMonP.Text;

    pi.Push(dblPurchasePrice, dblTotalDeposit, dblIntrest, dblNumOfMonths, dblGrossIncome);
    //st.setTotal(dblPurchasePrice);

    MessageBox.Show("Loading data");

    listBox1.Items.Add("Captured amounts");
    listBox1.Items.Add("********************");
    listBox1.Items.Add("Property Price: \tR" + txtPropertyP.Text);
    listBox1.Items.Add("Deposit Amount: \tR" + txtDepositP.Text);
    listBox1.Items.Add("Interest Rate: \t"+ txtInterestP.Text + "%");
    listBox1.Items.Add("Number of months: \tR" + txtNumOfMonP.Text);
    listBox1.Items.Add("Gross Profit: \tR" + txtGrossP.Text);
    listBox1.Items.Add("********************");

}

1 reference
private void btnCalculatedAmount_Click(object sender, EventArgs e)
```

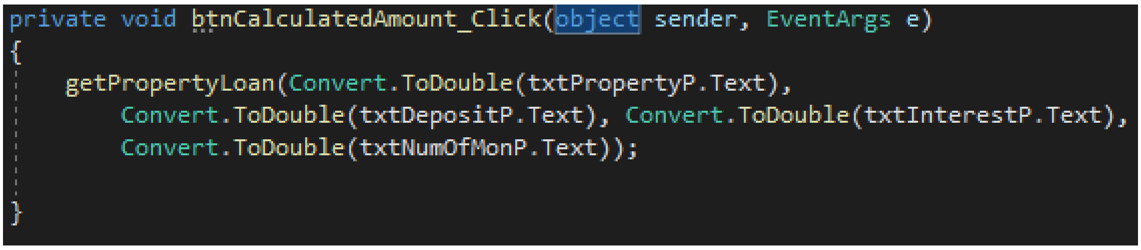
After the home loan repayments values are entered they are stored in the loan capture and to allow these values to be reprinted for the user to evaluate if it is the correct amounts entered.

When all the values have been added the user must the capture the amounts of the values and printed in the textbox below to display the details to the user once that is completed the user can then select the to view the monthly repayment cost of the home loan whether they qualify for the loan or not this is calculated a back button is also present in case any mistakes were made by the user.



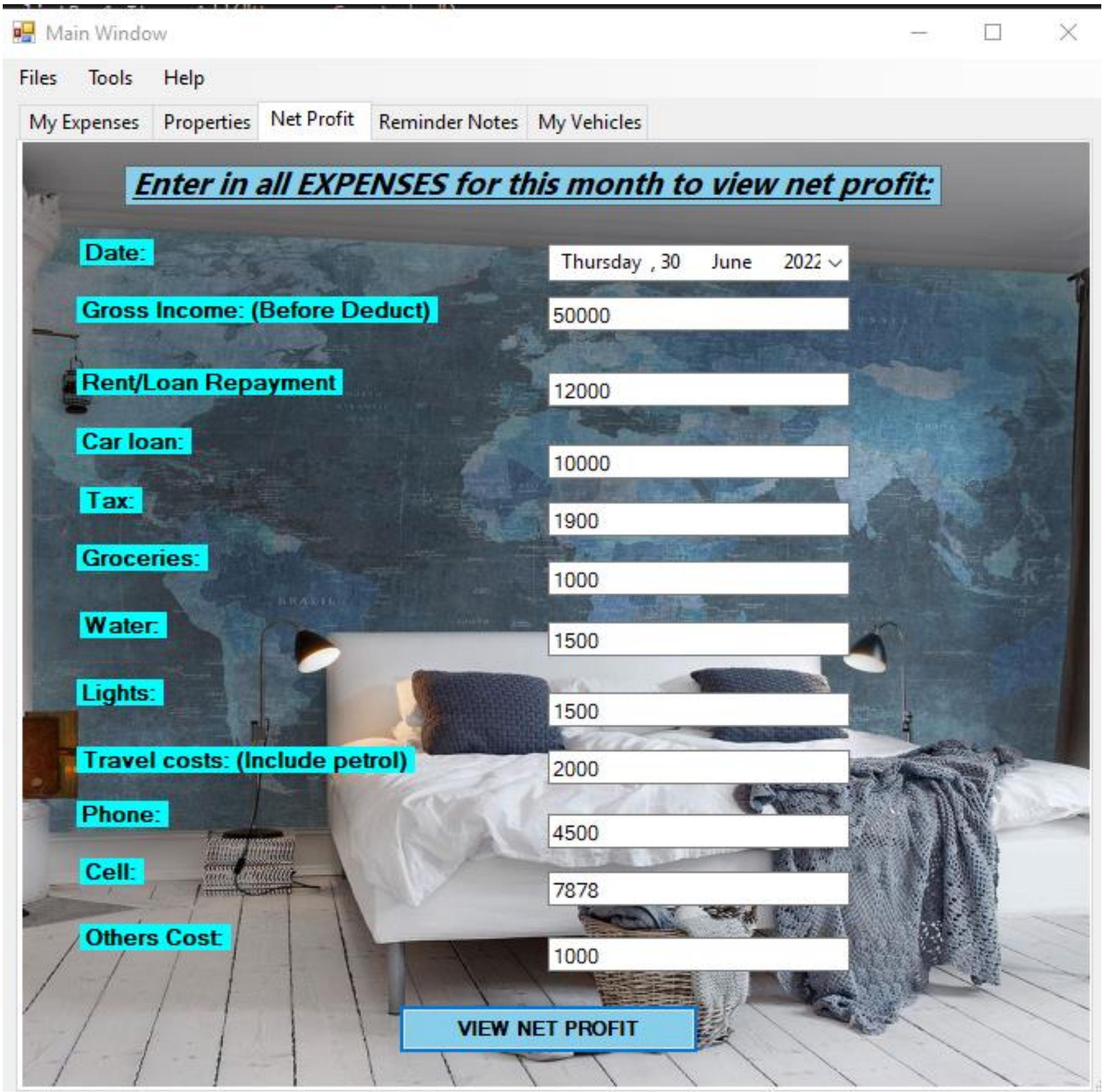
```
private void getPropertyLoan (double a, double b, double c, double d)
{
    double loanRepayment = exp.getLoanAmount(a, b, c, d);
    double message = Convert.ToDouble(txtGrossP.Text);
    double amount = message / 3;
    message = Math.Round(d, 2);
    //message.Math.Round();


    if (loanRepayment > amount)
    {
        MessageBox.Show("The payback amount for the HOMELOAN is: \n"
         + "\t \n R" + exp.getLoanAmount(a, b, c, d) + "\n" +
         "SORRY, YOU DO NOT QUALIFY FOR A LOAN :( ", "Message");

    }
    else if (loanRepayment < amount)
    {
        MessageBox.Show("The payback amount for the HOMELOAN is: \n"
         + "\t \n  R" + exp.getLoanAmount(a, b, c, d) + "\n" +
         "\n CONGRATULATIONS, YOU QUALIFY FOR A LOAN :)", "Message");
    }
}
```

The method below is calculation of the monthly repayment required to repay the home loan



```
private void btnCalculatedAmount_Click(object sender, EventArgs e)
{
    getPropertyLoan(Convert.ToDouble(txtPropertyP.Text),
        Convert.ToDouble(txtDepositP.Text), Convert.ToDouble(txtInterestP.Text),
        Convert.ToDouble(txtNumOfMonP.Text));

}
```

The next tab is the net profit this allows for the user to view the net profit after all expenses and deductions have been made to view the net profit after home loan and car repayments the loan and the car repayments can be after the calculations are done in the other tabs for the home loan and car purchase



Enter in all EXPENSES for this month to view net profit:

Date: Thursday , 30 June 2022

Gross Income: (Before Deduct) 50000

Rent/Loan Repayment 12000

Car loan: 10000

Tax: 1900

Groceries: 1000

Water: 1500

Lights: 1500

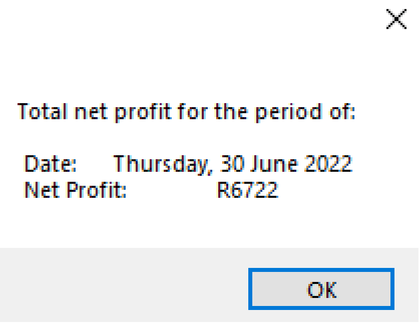
Travel costs: (Include petrol) 2000

Phone: 4500

Cell: 7878

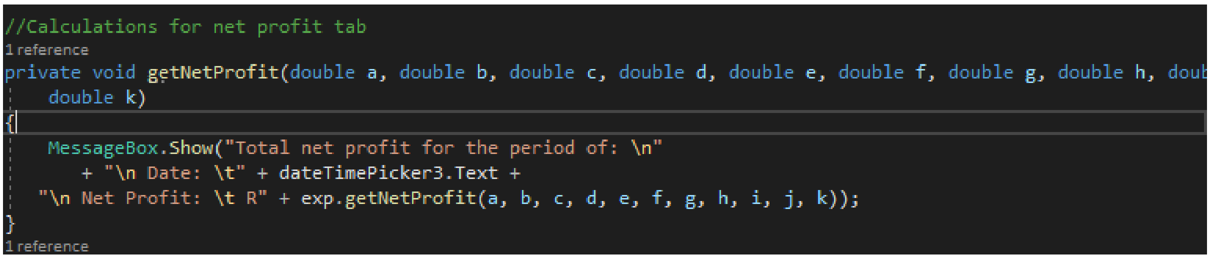
Others Cost 1000

After all the text fields are entered you can view the net profit of the user entered details and message box will display the results of the net profit calculations.



```
Total net profit for the period of:

 Date:     Thursday, 30 June 2022
 Net Profit:           R6722
```

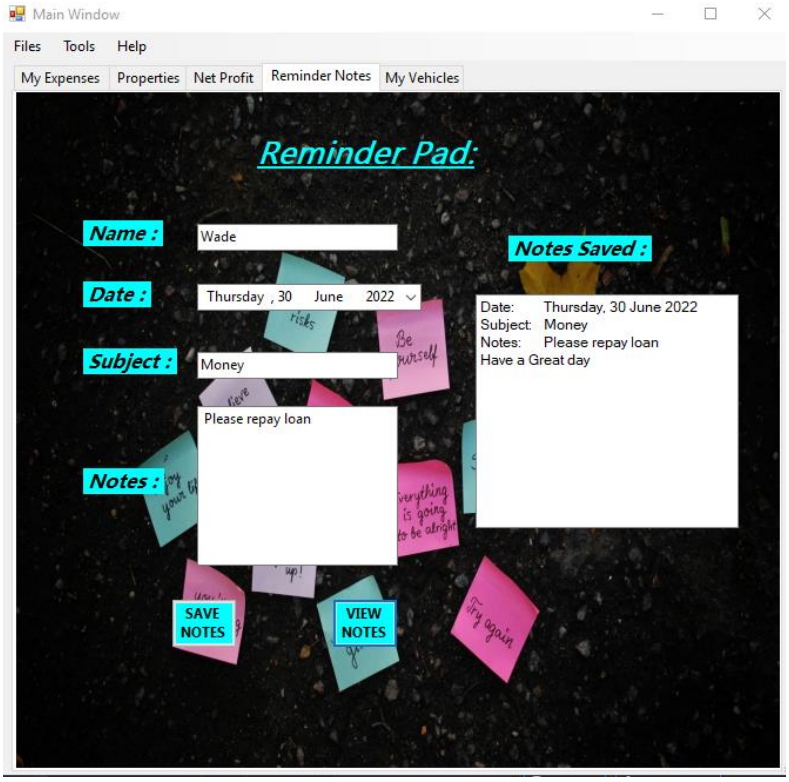
This is the money left over after all deductions have been made including the home loan repayment and the car purchase.



```
//Calculations for net profit tab
1 reference
private void getNetProfit(double a, double b, double c, double d, double e, double f, double g, double h, doub
    double k)
{
    MessageBox.Show("Total net profit for the period of: \n"
       + "\n Date: \t" + dateTimePicker3.Text +
   "\n Net Profit: \t R" + exp.getNetProfit(a, b, c, d, e, f, g, h, i, j, k));
}
1 reference
```

The calculation to the net profit of the user according to the user entered values.

The next tab is to allow the user to add notes that they may need to remind them oof certain functions they still need to do or whether they need notes to remind them of repayments that still need to be done or purchases that still need to be added



Name : Wade

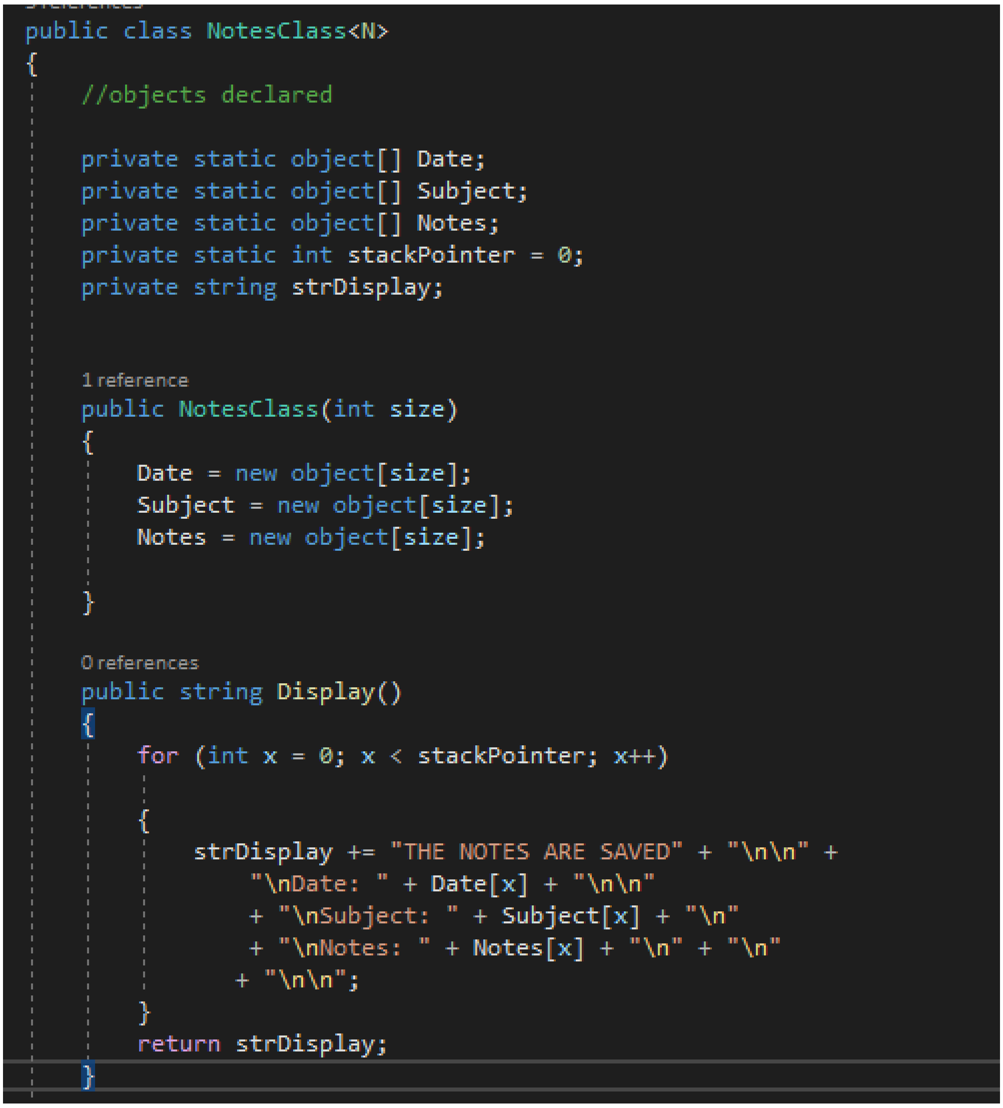
Date : Thursday , 30 June 2022

Subject : Money

Notes : Please repay loan

Notes Saved :

```
Date:     Thursday, 30 June 2022
Subject:  Money
Notes:    Please repay loan
Have a Great day
```

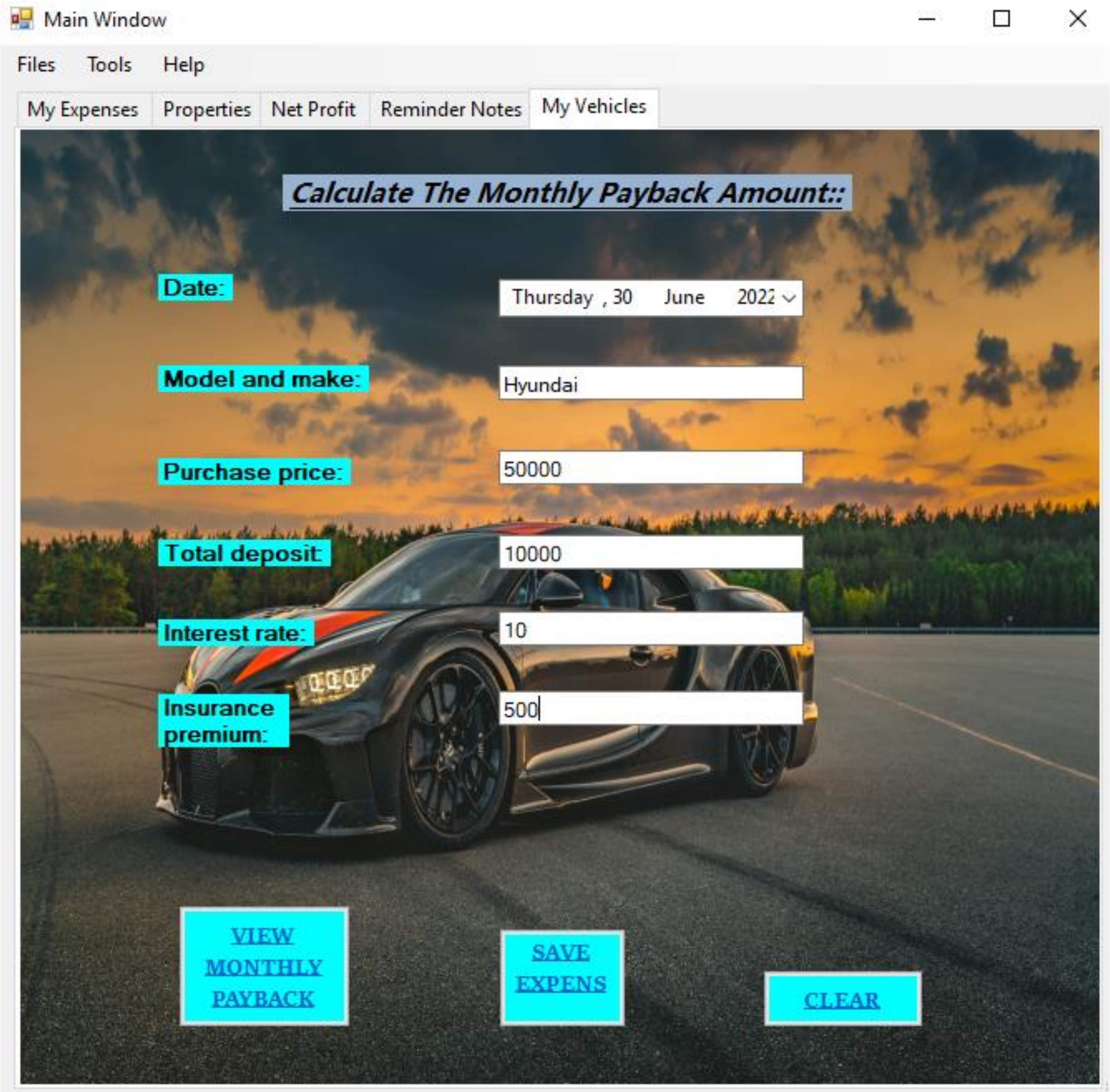


```
public class NotesClass<N>
{
    //objects declared

    private static object[] Date;
    private static object[] Subject;
    private static object[] Notes;
    private static int stackPointer = 0;
    private string strDisplay;


    1 reference
    public NotesClass(int size)
    {
        Date = new object[size];
        Subject = new object[size];
        Notes = new object[size];

    }

    0 references
    public string Display()
    {
        for (int x = 0; x < stackPointer; x++)

        {
            strDisplay += "THE NOTES ARE SAVED" + "\n\n" +
                "\nDate: " + Date[x] + "\n\n"
                + "\nSubject: " + Subject[x] + "\n"
                + "\nNotes: " + Notes[x] + "\n" + "\n"
               + "\n\n";
        }
        return strDisplay;
    }
```

The notes are stored and saved for later use in any of the calculations or expenses.

This are here as a reminder to the use in an effort to help them remember all the deductions and notes they may need to help with the calculations of the home loan or the car purchase.

The car purchase is the next tab this tab allows the user to enter the values needed to purchase a car the purchase price and interest rate is included and then calculated to then allow the user to calculate the amount needed to repay for the car



Date: Thursday , 30 June 2022

Model and make: Hyundai

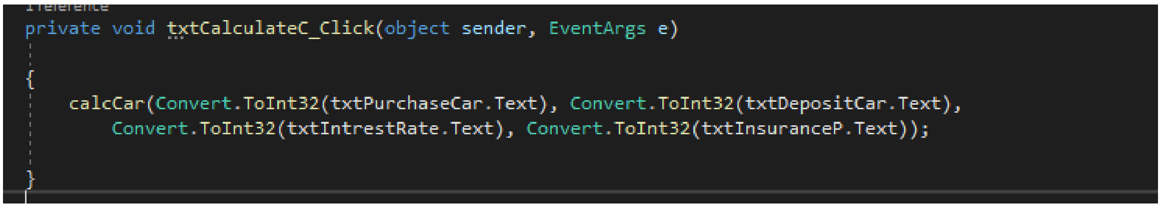
Purchase price: 50000

Total deposit: 10000

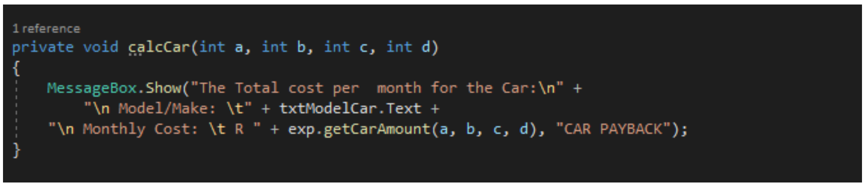
Interest rate: 10

Insurance premium: 500

The expenses of the car then saved and will then be able to view from the monthly payback after the calculations are made for the repayment



```
1 reference
private void txtCalculateC_Click(object sender, EventArgs e)

{
    calcCar(Convert.ToInt32(txtPurchaseCar.Text), Convert.ToInt32(txtDepositCar.Text),
        Convert.ToInt32(txtIntrestRate.Text), Convert.ToInt32(txtInsuranceP.Text));

}
```



```
1 reference
private void calcCar(int a, int b, int c, int d)
{
    MessageBox.Show("The Total cost per  month for the Car:\n" +
        "\n Model/Make: \t" + txtModelCar.Text +
    "\n Monthly Cost: \t R " + exp.getCarAmount(a, b, c, d), "CAR PAYBACK");
}
```

This is for all the car calculations and the display of the expenses of the car these expenses can be added to the notes to then be re-used in the monthly expenses or the net profit calculations all the interest and loan repayments a=can also be added to the notes to be re-used later in monthly calculations or the expense calculations