

## CONTACT's APP

ANDROID:::

### MainActivity.cs

```
using System;

using Android.App;
using Android.Content;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.OS;

using App4.mDatabase;
using Android.Database;
using Java.Util;
using Android.Database.Sqlite;

namespace App4
{
    [Activity(Label = "Contact-Save,Load and Delete", MainLauncher = true, Icon =
"@drawable/icon")]

    public class MainActivity : Activity
    {
        private ListView lv;

        private SearchView sv;
```

```

private EditText nameEditText,nameEditText1,getname;

private Button saveBtn, retrieveBtn;

JavaList<String> spaceCraft = new JavaList<String>();

int selectedItem = -1;

private ArrayAdapter adapter;

protected override void onCreate(Bundle bundle)
{
    base.onCreate(bundle);

    // Set our view from the "main" layout resource
    setContentView(Resource.Layout.Main);

    this.InitializeUI();

    // Get our button from the layout resource,
    // and attach an event to it

    adapter    =    new    ArrayAdapter(this,    Android.Resource.Layout.SimpleListItem1,
spaceCraft);

    saveBtn.Click += SaveBtn_Click;

    retrieveBtn.Click += RetrieveBtn_Click;

    sv.QueryTextChange += Sv_QueryTextChange;

    lv.ItemClick += Lv_ItemClick;

    lv.ItemLongClick += Lv_ItemLongClick;

}

private void Lv_ItemLongClick(object sender, AdapterView.ItemLongClickEventArgs e)
{

```

```

DBAdapter db = new DBAdapter(this);

String s1 = spaceCraft[e.Position];

String newString = s1.Substring(s1.IndexOf(' ') + 1);

String[] s2 = s1.Split(' ');

String s3 = s2[0];

this.selectedItem = e.Position;


var EditDialog = new AlertDialog.Builder(this);

EditDialog.SetMessage("Do you want to Edit or Delete");

EditDialog.SetNeutralButton("Edit", delegate {

    //For updating the database code is here

    nameEditText.Text = s3;

    nameEditText1.Text = newString;

});

EditDialog.SetPositiveButton("Cancel", delegate { });

EditDialog.SetNegativeButton("Delete", delegate { db.Delete(newString);

    Toast.MakeText(this,"Contact Deleted",ToastLength.Short).Show();

    adapter.NotifyDataSetChanged();

});

// Show the alert dialog to the user and wait for response.

EditDialog.Show();

}

private void Lv_ItemClick(object sender, AdapterView.ItemClickEventArgs e)

```

```

{

    double i = 0;

    String s1 = spaceCraft[e.Position];

    String newString = s1.Substring(s1.IndexOf(' ') + 1);

    bool result = double.TryParse(newString, out i);

    if (result == true)
    {

        var callDialog = new AlertDialog.Builder(this);

        callDialog.SetMessage("Call " + s1 + "?");

        callDialog.SetNeutralButton("Call", delegate {

            // Create intent to dial phone

            var callIntent = new Intent(Intent.ActionCall);

            callIntent.SetData(Android.Net.Uri.Parse("tel:" + i));

            StartActivity(callIntent);

        });

        callDialog.SetNegativeButton("Cancel", delegate { });

        // Show the alert dialog to the user and wait for response.

        callDialog.Show();

    }

    else

    {

        Toast.MakeText(this, "Not a valid Number", ToastLength.Short).Show();

    }
}

```

```
}
```

```
private void Sv_QueryTextChanged(object sender, SearchView.QueryTextChangedEventArgs e)
```

```
{
```

```
    String searchTerm = e.NewText;
```

```
    this.GetSpaceCrafts(searchTerm);
```

```
}
```

```
private void RetrieveBtn_Click(object sender, EventArgs e)
```

```
{
```

```
    GetSpaceCrafts(null);
```

```
    adapter.NotifyDataSetChanged();
```

```
}
```

```
private void SaveBtn_Click(object sender, EventArgs e)
```

```
{
```

```
    String name1 = nameEditText.Text;
```

```
    String name2 = nameEditText1.Text;
```

```
    if (!String.IsNullOrEmpty(name1) && !String.IsNullOrEmpty(name2))
```

```
    {
```

```
        Save(nameEditText.Text, nameEditText1.Text);
```

```
    }
```

```
else
```

```

        {
            Toast.makeText(this, "Name and Phone Number is not present",
ToastLength.Short).Show();
        }
    }

```

```
private void InitializeUI()
```

```

{
    lv = findViewById<ListView>(Resource.Id.listView1);
    sv = findViewById<SearchView>(Resource.Id.searchView1);
    nameEditText = findViewById<EditText>(Resource.Id.editText1);
    nameEditText1 = findViewById<EditText>(Resource.Id.editText2);
    saveBtn = findViewById<Button>(Resource.Id.button1);
    retrieveBtn = findViewById<Button>(Resource.Id.button2);

    getname = findViewById<EditText>(Resource.Id.editText3);
}

```

```
private void Save(String name, String name1)//name1 added for phone
```

```

{
    DBAdapter db = new DBAdapter(this);
    db.openDB();

    bool saved = db.Add(name, name1);
    db.CloseDB();
    if (saved)//modified

```

```

{
    nameEditText.Text = "";
    nameEditText1.Text = "";
    Toast.MakeText(this, "Contact Saved", ToastLength.Short).Show();
}
else
{
    Toast.MakeText(this, "Unable to Save", ToastLength.Short).Show();
}
this.GetSpaceCrafts(null);

}

public void GetSpaceCrafts(string searchTerm)
{
    spaceCraft.Clear();
    DBAdapter db = new DBAdapter(this);
    db.openDB();
    ICursor c = db.Retrieve(searchTerm);
    //SpaceCraft s = null;

    if (c != null)
    {
        while (c.MoveNext())
        {
            string name = c.GetString(1);

```

```

        string phone = c.GetString(c.GetColumnIndex("phone"));

        string name1 = name + " " + phone;

        spaceCraft.Add(name1);

        Collections.Sort(spaceCraft);

    }

};

db.CloseDB();

if (spaceCraft.Size() > 0)

{

    lv.Adapter = adapter;

}

}

//Update Database for contact

public Boolean update(String newName, String newPhone, int id)

{

    try

    {

        spaceCraft.RemoveAt(id);

        return true;

    }

    catch (Exception)

    {

        return false;

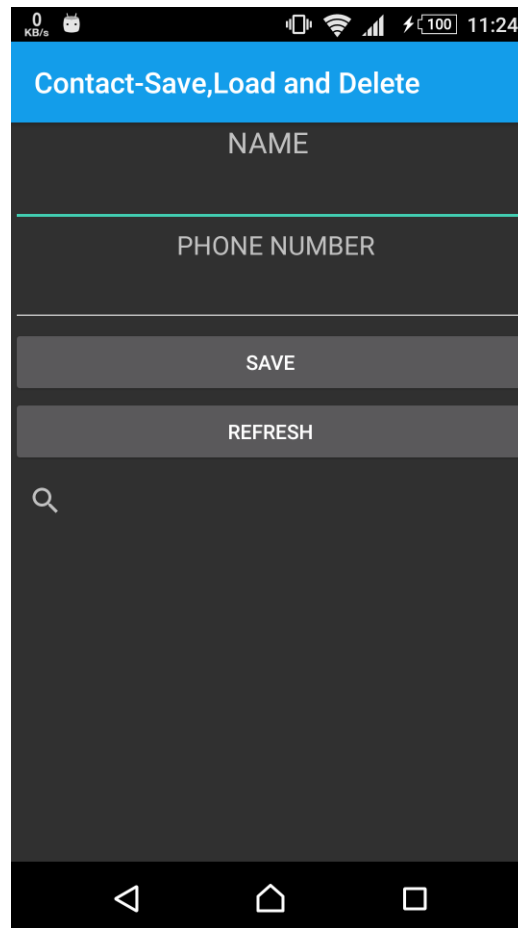
    }

}

```



```
}  
  
}
```



## Database Files:::

### Constants.cs

This file has necessary query and data field. Any changes to database can be done here.

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
using Android.App;  
using Android.Content;  
using Android.OS;  
using Android.Runtime;
```

```
using Android.Views;
```

```
using Android.Widget;
```

```
namespace App4.mDatabase
```

```
{
```

```
    class Constants
```

```
    {                                     //query changed and String name is added
```

```
        public static String ROW_ID = "id";
```

```
        public static String NAME = "name";
```

```
        public static String PHONE = "phone";
```

```
        public static String DB_NAME = "b1_DB";
```

```
        public static String TB_NAME = "b1_TB";
```

```
        public static int DB_VERSION = 1;
```

```
        public static String CREATE_TB = "CREATE TABLE b1_TB(id INTEGER PRIMARY  
KEY AUTOINCREMENT,"
```

```
        + "name TEXT NOT NULL,phone TEXT NOT NULL);";//query change part 2
```

```
        public static String DROP_TB = "DROP TABLE IF EXISTS "+TB_NAME;
```

```
    }
```

```
}
```

## **DBAdapter.cs**

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using Android.App;
```

```
using Android.Content;
```

```
using Android.OS;
```

```
using Android.Runtime;
```

```
using Android.Views;
```

```
using Android.Widget;
using Android.Database.Sqlite;
using Android.Database;
```

```
namespace App4.mDatabase
{
    class DBAdapter
    {
        private Context c;
        private SQLiteDatabase db;
        private DBHelper helper;

        public DBAdapter(Context c)
        {
            this.c = c;
            helper = new DBHelper(c);
        }
        public DBAdapter openDB()
        {
            try
            {
                db = helper.WritableDatabase;
            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
            return this;
        }
        public void CloseDB()
        {
            try
            {
                helper.Close();
            }
        }
    }
}
```

```

    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
}

public bool Add(String name,String name1) //String name1 added for phone number
{
    try
    {
        ContentValues cv = new ContentValues();
        cv.Put(Constants.NAME, name);
        cv.Put(Constants.PHONE, name1);//name1 added for phone numbers
        db.Insert(Constants.TB_NAME, Constants.ROW_ID, cv);

        return true;
    }
    catch (Exception e)
    {

        Console.WriteLine(e.Message);
    }
    return false;
}

public ICursor Retrieve(String searchTerm)
{
    String[] columns =
{Constants.ROW_ID,Constants.NAME,Constants.PHONE};//columns added phone part 2

    ICursor c = null;

    if (!String.IsNullOrEmpty(searchTerm))
    {

```

```

        string sql = "SELECT * FROM " + Constants.TB_NAME + " WHERE " +
Constants.NAME + " LIKE '%" + searchTerm + "%'";
        c = db.RawQuery(sql, null);
    }
    else
    {
        c = db.Query(Constants.TB_NAME, columns, null, null, null, null, null);
    }
    return c;
}

public int Delete(String name) //String name1 added for phone number
{
    try
    {
        SQLiteDatabase db1 = helper.WritableDatabase;
        String[] whereArgs = { name };
        int count = db1.Delete(Constants.TB_NAME, Constants.PHONE+ "=", whereArgs);
        MainActivity m = new MainActivity();
        return count;

    }
    catch (Exception e)
    {

        Console.WriteLine(e.Message);
    }
    return 0;
}

public int updateName(String name,String newName)
{
    try
    {
        SQLiteDatabase db1 = helper.WritableDatabase;
        ContentValues cv = new ContentValues();
        cv.Put(Constants.NAME, newName);
        String[] whereArgs = { name };

```

```

        int count = db1.Update(Constants.TB_NAME, cv , Constants.NAME + "=?",
whereArgs);
        return count;

    }
    catch (Exception e)
    {

        Console.WriteLine(e.Message);
    }
    return 0;
}

public int updateName1(String name, String newName)
{
    try
    {
        SQLiteDatabase db1 = helper.WritableDatabase;
        ContentValues cv = new ContentValues();
        cv.Put(Constants.PHONE, newName);
        String[] whereArgs = { name };
        int count = db1.Update(Constants.TB_NAME, cv, Constants.PHONE + "=?",
whereArgs);
        return count;

    }
    catch (Exception e)
    {

        Console.WriteLine(e.Message);
    }
    return 0;
}
}
}

```

## DBHelper.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Android.App;
using Android.Content;
using Android.OS;
using Android.Runtime;
using Android.Views;
using Android.Widget;
using Android.Database.Sqlite;

namespace App4.mDatabase
{
    class DBHelper : SQLiteOpenHelper
    {
        public DBHelper(Context context) : base(context, Constants.DB_NAME, null,
Constants.DB_VERSION)
        {
        }
        public override void OnCreate(SQLiteDatabase db)
        {
            try //NO changes have to be made
            {
                db.ExecSQL(Constants.CREATE_TB);
            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

```

        public override void OnUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
        {
            db.ExecSQL(Constants.DROP_TB);
            OnCreate(db);
        }
    }
}

```

## **Dialog\_Name.cs**

```

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Android.App;

using Android.Content;

using Android.OS;

using Android.Runtime;

using Android.Views;

using Android.Widget;

namespace App4

{

    class dialog_Name : DialogFragment

    {

        private Button updatebt;
    }
}

```



```

private EditText et;

public override View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
{

    base.onCreateView(inflater, container, savedInstanceState);

    var view = inflater.Inflate(Resource.Layout.layout1, container, false);

    return view;

}

}

}

```

### **Main.xml**

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:orientation="vertical"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:minWidth="25px"

    android:minHeight="25px">

    <TextView

        android:text="Enter the Name"

        android:textAppearance="?android:attr/textAppearanceMedium"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:textSize="20.0dp"

```

```
android:id="@+id/textView3"
```

```
android:layout_marginLeft="150.0dp" />
```

```
<EditText
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/editText3" />
```

```
<Button
```

```
android:text="Update"
```

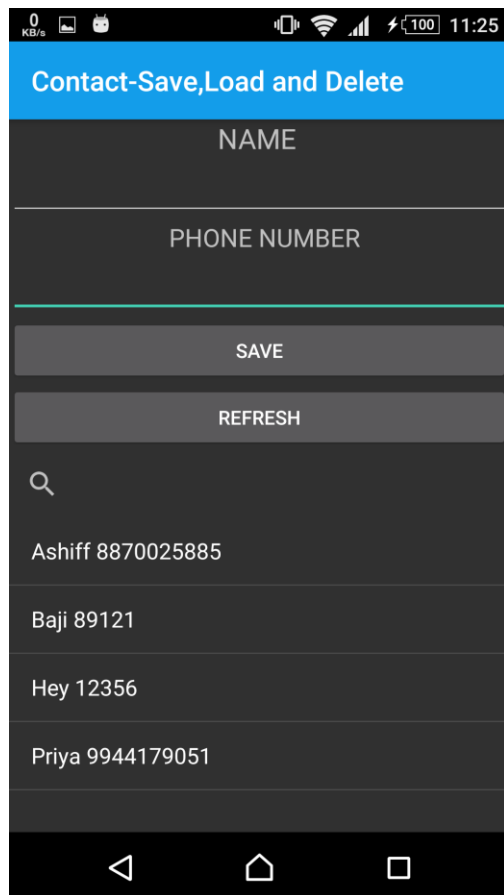
```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

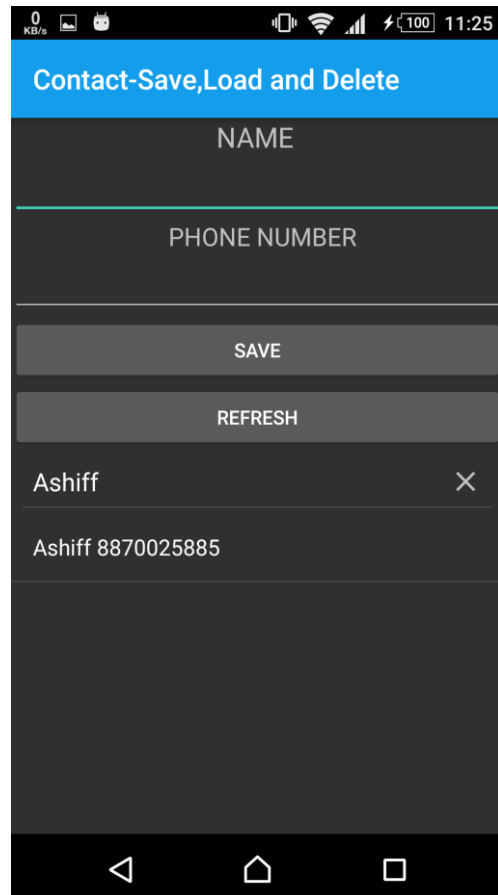
```
android:id="@+id/button3" />
```

```
</LinearLayout>
```

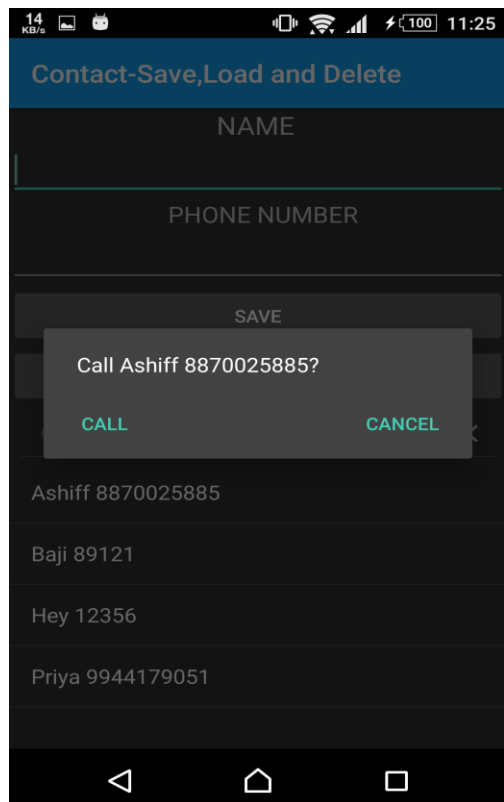
Adding Data to Database::



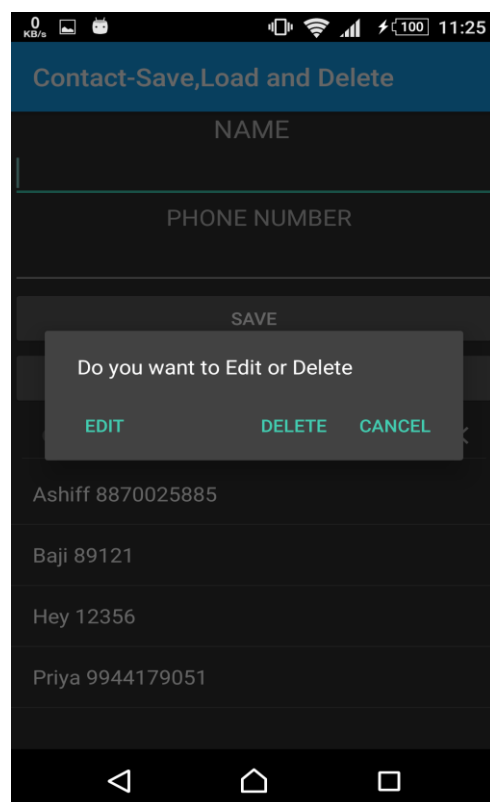
Searching a Data::



Calling by clicking a name::



By long tapping:: (update & delete)



End of Android Project::

## **IOS::**

### **Contact.cs**

```
using System;
using System.Collections.Generic;
using System.Text;

namespace App6
{
    class Contact
    {
        public string Name { get; set; }

        public string PhoneNumber { get; set; }

        public Contact(string name, string phone)
        {
            Name = name;
            PhoneNumber = phone;
        }

        public override string ToString()
        {
            return Name + " " + PhoneNumber;
        }
    }
}
```

### **ListController.cs**

```
using Foundation;
using System;
using UIKit;
using System.IO;

namespace App6
{
    public partial class ListController : UIViewController
    {
        public ListController(IntPtr handle) : base(handle)
        {
        }
    }
}
```

```

    {
    }

    public override void ViewDidLoad()
    {
        base.ViewDidLoad();

        var documents =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

        var filename = Path.Combine(documents, "MyContacts.txt");
        var text = File.ReadAllText(filename);
        string[] line = text.Split(',');
        Contact mycontact = new Contact(line[0], line[1]);
        Contact[] contactList = { mycontact };
        TableData.Source = new TableSource(contactList, this);
    }
}
}

```

### **TableController.cs**

```

using System;
using System.Collections.Generic;
using System.Text;
using Foundation;
using UIKit;
namespace App6
{
    class TableSource : UITableViewSource
    {
        protected Contact[] tableItems;
        protected string cellIdentifier = "TableCell";
        ListController owner;
        public TableSource(Contact[] items, ListController owner)
        {
            tableItems = items;

```

```

        this.owner = owner;
    }

    public override nint RowsInSection(UITableView tableview, nint section)
    {
        return tableItems.Length;
    }

    public override void RowSelected(UITableView tableView, NSIndexPath indexPath)
    {
        //String s1 = tableItems[indexPath.Row];
        //UIApplication.SharedApplication.OpenUrl(new NSURL("tel:" + ph));
        tableView.DeselectRow(indexPath, true);
    }

    public override UITableViewCell GetCell(UITableView tableView, NSIndexPath
indexPath)
    {
        UITableViewCell cell = tableView.DequeueReusableCell(cellIdentifier);
        if (cell == null)
        {
            cell = new UITableViewCell(UITableViewCellStyle.Default, cellIdentifier);
            cell.TextLabel.Text = tableItems[indexPath.Row].ToString();
            String s1 = cell.TextLabel.Text;
            var alert = UIAlertController.Create("Confirm",s1 , UIAlertControllerStyle.Alert);
            alert.AddAction(UIAlertAction.Create("OK", UIAlertActionStyle.Default, null));
            // PresentViewController(alert, true, null);
            return cell;
        }
    }
}

```

### **ViewController.cs**

```

using System;
using System.IO;
using UIKit;
namespace App6

```

```

{
    public partial class ViewController : UIViewController
    {

        public ViewController(IntPtr handle) : base(handle)
        {
        }

        public override void ViewDidLoad()
        {
            base.ViewDidLoad();

            // Perform any additional setup after loading the view, typically from a nib.
        }

        public override void DidReceiveMemoryWarning()
        {
            base.DidReceiveMemoryWarning();

            // Release any cached data, images, etc that aren't in use.
        }

        partial void Add_TouchUpInside(UIButton sender)
        {
            String name = entername.Text;
            String ph = phone.Text;
            if (!String.IsNullOrEmpty(name) && !String.IsNullOrEmpty(ph))
            {
                String line = String.Format("{0},{1}", name, ph);

                var documents = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

                var filename = Path.Combine(documents, "MyContacts.txt");
                File.WriteAllText(filename, line);

                var alert = UIAlertController.Create("Confirm", "Contact Added",
                UIAlertControllerStyle.Alert);
            }
        }
    }
}

```

```

        alert.AddAction(UAAlertAction.Create("OK", UAAlertActionStyle.Default, null));
        PresentViewController(alert, true, null);

        entername.Text = "";
        phone.Text = "";
    }

    else
    {
        var alert = UIAlertController.Create("Warning!", "Name and Number is not present",
        UIAlertControllerStyle.Alert);

        alert.AddAction(UAAlertAction.Create("ok", UAAlertActionStyle.Default, null));

        PresentViewController(alert, true, null);
    }
}
}
}
}

```

### **Main.cs**

```

using UIKit;

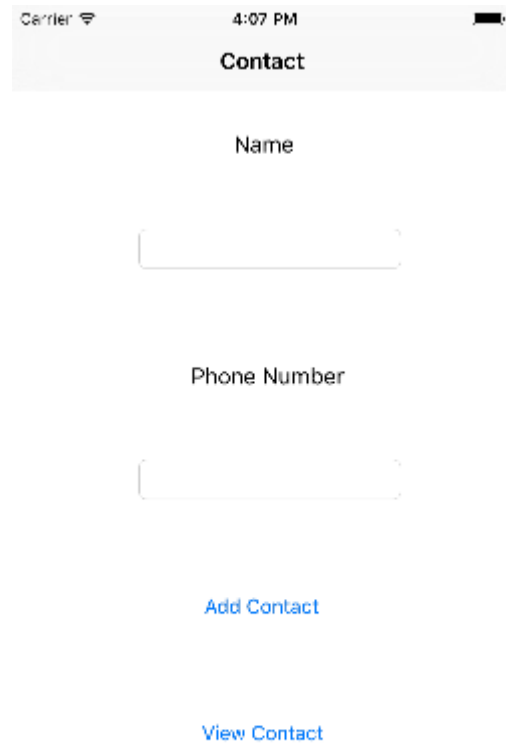
namespace App6
{
    public class Application
    {
        // This is the main entry point of the application.
        static void Main(string[] args)
        {
            // if you want to use a different Application Delegate class from "AppDelegate"
            // you can specify it here.

            UIApplication.Main(args, null, "AppDelegate");
        }
    }
}

```



Front page of the App::



Carrier 4:07 PM

**Contact**

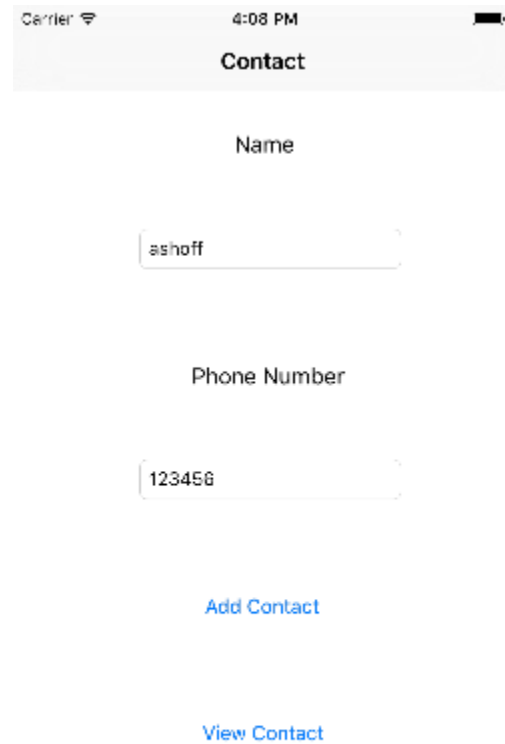
Name

Phone Number

[Add Contact](#)

[View Contact](#)

Adding the Data::



Carrier 4:08 PM

**Contact**

Name

ashoff

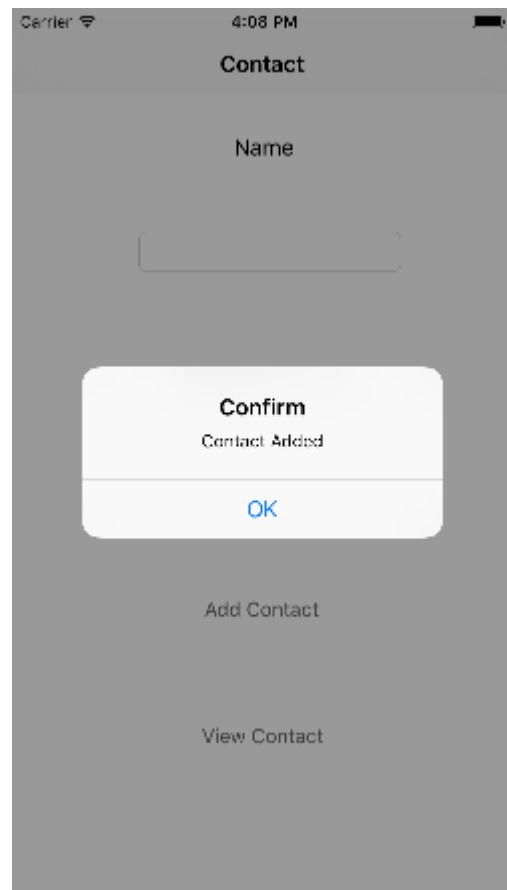
Phone Number

123456

[Add Contact](#)

[View Contact](#)

Click Add Contact Saves the data



Carrier 4:08 PM

**Contact**

Name

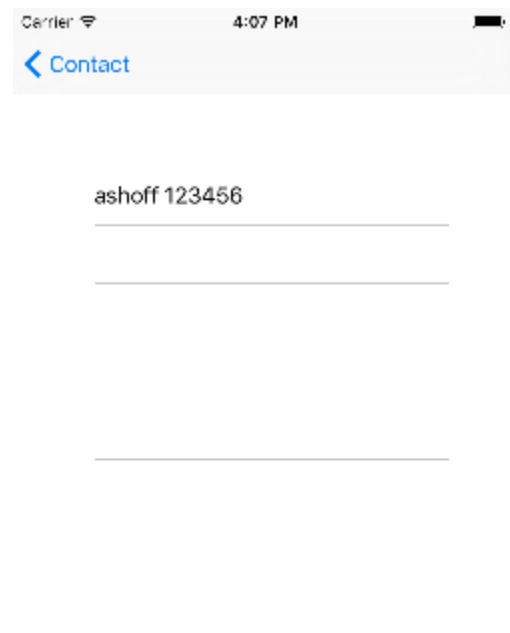
**Confirm**  
Contact Added

[OK](#)

[Add Contact](#)

[View Contact](#)

TableView to list the data: (Call Option)



Carrier 4:07 PM

[Contact](#)

ashoff 123456