```
==============================
Filename : android_layout.txt




______________________________
Filename : activity_about.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="top"
    android:orientation="vertical" >

    <ScrollView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <HorizontalScrollView
            android:layout_width="match_parent"
            android:layout_height="wrap_content" >

            <LinearLayout
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:gravity="top" >

                <ImageView
                    android:id="@+id/imageview_About"
                    android:layout_width="400dp"
                    android:layout_height="1440dp"
                    android:src="@drawable/about" />
            </LinearLayout>
        </HorizontalScrollView>
    </ScrollView>

</LinearLayout>


______________________________
Filename : activity_factordetails.xml

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <ListView
        android:id="@+id/listview_Products"
        android:layout_width="match_parent"
        android:layout_height="376dp"
        android:layout_marginBottom="30dp" >
    </ListView>

    <Button
        android:id="@+id/button_Send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:enabled="false"
        android:text="@string/Send"
        android:visibility="visible" />

    <Button
        android:id="@+id/button_Delete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:enabled="false"
        android:text="@string/Delete"
        android:visibility="visible" >
    </Button>

</RelativeLayout>
```

Filename : activity_factors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ListView
        android:id="@+id/listview_factors"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
         >
    </ListView>

</LinearLayout>
```

Filename : activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ScrollView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >

        <RelativeLayout
            android:layout_width="fill_parent"
            android:layout_height="fill_parent" >

            <TextView
                android:id="@+id/textview_Username"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="30dp"
                android:text="@string/Username"
                android:textAppearance="?android:attr/textAppearanceLarge" />

            <EditText
                android:id="@+id/edittext_Username"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/textview_Username"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="20dp"
                android:ems="10"
                android:inputType="textPersonName" >

                <requestFocus />
            </EditText>

            <TextView
                android:id="@+id/textview_Password"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/edittext_Username"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="20dp"
                android:text="@string/Password"
                android:textAppearance="?android:attr/textAppearanceLarge" />

            <EditText
                android:id="@+id/edittext_Password"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/textview_Password"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="20dp"
                android:ems="10"
                android:inputType="textPassword" />

            <Button
                android:id="@+id/button_Login"
                style="?android:attr/buttonStyleSmall"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/edittext_Password"
                android:layout_centerHorizontal="true"
                android:layout_marginTop="20dp"
                android:text="@string/Login" />
        </RelativeLayout>
    </ScrollView>

</LinearLayout>
```

____

Filename : activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.rexaTB.chekehandroid.Activity_Main" >

    <ListView
        android:id="@+id/listview_Products"
        android:layout_width="match_parent"
        android:layout_height="376dp"
        android:layout_marginBottom="30dp" >
    </ListView>

    <Button
        android:id="@+id/button_Confirm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:enabled="false"
        android:text="@string/Confirm"
        android:visibility="visible" />
    <Button
        android:id="@+id/button_Refresh"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:enabled="true"
        android:text="@string/Refresh"
        android:visibility="visible" >
    </Button>
</RelativeLayout>



______________________________
Filename : activity_productdetails.xml

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textview_Title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="@string/NotLoaded"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <ImageView
        android:id="@+id/imageview_Picture"
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textview_Title"
        android:contentDescription="تصویر محصول در این قسمت قرار میگیرد"
        android:src="@android:drawable/ic_menu_gallery" />

    <TextView
        android:id="@+id/textview_Description"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_below="@+id/imageview_Picture"
        android:text="@string/NotLoaded"
        android:visibility="invisible" />

</RelativeLayout>



______________________________
Filename : activity_savelist.xml

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/textview_Title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/Title" />

    <EditText
        android:id="@+id/edittext_Title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview_Title"
        android:layout_centerHorizontal="true"
        android:ems="10" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button_Save"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/edittext_Title"
        android:layout_centerHorizontal="true"
        android:text="@string/Save" />

    <ListView
        android:id="@+id/listview_Products"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button_Save" >
    </ListView>

</RelativeLayout>
```

____
Filename : activity_userregister.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textview_Username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginRight="20dp"
        android:layout_marginTop="20dp"
        android:text="@string/Username" />

    <TextView
        android:id="@+id/textview_Firstname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textview_Username"
        android:layout_below="@+id/textview_Username"
        android:layout_marginTop="26dp"
        android:text="@string/Firstname" />

    <TextView
        android:id="@+id/textview_Lastname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textview_Firstname"
        android:layout_below="@+id/textview_Firstname"
        android:layout_marginTop="26dp"
        android:text="@string/Lastname" />

    <TextView
        android:id="@+id/textview_Password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textview_Lastname"
        android:layout_below="@+id/textview_Lastname"
        android:layout_marginTop="26dp"
        android:text="@string/Password" />

    <TextView
        android:id="@+id/textview_Confirm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textview_Password"
        android:layout_below="@+id/textview_Password"
        android:layout_marginTop="26dp"
        android:text="@string/Confirm" />

    <EditText
        android:id="@+id/edittext_Username"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textview_Username"
        android:layout_alignBottom="@+id/textview_Username"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="14dp"
        android:ems="10" />
    <EditText
        android:id="@+id/edittext_Firstname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textview_Firstname"
        android:layout_alignBottom="@+id/textview_Firstname"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="14dp"
        android:ems="10" />

    <EditText
        android:id="@+id/edittext_Lastname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textview_Lastname"
        android:layout_alignBottom="@+id/textview_Lastname"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="14dp"
        android:ems="10" />

    <EditText
        android:id="@+id/edittext_Password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textview_Password"
        android:layout_alignBottom="@+id/textview_Password"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="14dp"
        android:ems="10"
        android:inputType="textPassword" />

    <EditText
        android:id="@+id/edittext_Confirm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/textview_Confirm"
        android:layout_alignBottom="@+id/textview_Confirm"
        android:layout_alignParentLeft="true"
        android:layout_marginLeft="14dp"
        android:ems="10"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/button_Register"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="@string/Ok" />

</RelativeLayout>
```

____
Filename : menuitem_factor.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textview_Title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:text="@string/NotLoaded"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:textColor="@android:color/black" />

        <Button
            android:id="@+id/button_Details"
            style="?android:attr/buttonStyleSmall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/textview_Title"
            android:layout_centerHorizontal="true"
            android:text="@string/Details" />
    </RelativeLayout>

    <View
        android:layout_width="fill_parent"
        android:layout_height="1dp"
        android:background="#FF0000FF"
        android:rotation="180" />

</LinearLayout>
```

____
Filename : menuitem_product.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
```

```
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textview_Title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:text="@string/NotLoaded"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="@android:color/black" />

    <Button
        android:id="@+id/button_Plus"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/textview_Title"
        android:text="@string/Plus" />

    <Button
        android:id="@+id/button_Minus"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/textview_Title"
        android:text="@string/Minus" />

    <TextView
        android:id="@+id/textview_Count"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview_Title"
        android:layout_centerHorizontal="true"
        android:text="@string/Zero"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColor="@android:color/black" />

    <Button
        android:id="@+id/button_Details"
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview_Count"
        android:layout_centerHorizontal="true"
        android:text="@string/Details" />
</RelativeLayout>
```

```
    <View
        android:layout_width="fill_parent"
        android:layout_height="1dp"
        android:background="#FF0000FF"
        android:rotation="180" />

</LinearLayout>
```

```
==============================
Filename : android_manifest.txt
```

```
______________________________
Filename : AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.rexaTB.chekehandroid"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="9"
        android:targetSdkVersion="9" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".Activity_Main"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Activity_UserRegister"
            android:label="@string/Register" >
        </activity>
        <activity
            android:name=".Activity_Login"
            android:label="@string/Login" >
        </activity>
        <activity
            android:name=".Activity_SaveList"
            android:label="@string/Save" >
        </activity>
        <activity
            android:name=".Activity_About"
            android:label="@string/About" >
        </activity>
        <activity
            android:name=".Activity_Factors"
            android:label="@string/PrevLists" >
        </activity>
        <activity
            android:name=".Activity_ProductDetails"
            android:label="@string/Details" >
        </activity>
        <activity
            android:name=".Activity_FactorDetails"
            android:label="@string/Details" >
        </activity>
    </application>

</manifest>



==============================
Filename : android_menu.txt





______________________________
Filename : main_menu.xml

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.rexaTB.chekehandroid.Activity_Main" >

    <group android:id="@+id/group_UserDetails" >
        <item
            android:id="@+id/item_UserRegister"
            android:title="@string/Register">
        </item>
        <item
            android:id="@+id/item_UserLogin"
            android:title="@string/Login">
        </item>
    </group>
    <group android:id="@+id/product" >
        <item
            android:id="@+id/item_PrevLists"
            android:title="@string/PrevLists">
        </item>
    </group>
        <group android:id="@+id/App">
            <item android:id="@+id/item_About" android:title="@string/About"></item>
        </group>
</menu>


==============================
Filename : android_src.txt





______________________________
Filename : Activity_About.java

package com.rexaTB.chekehandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;

public class Activity_About extends Activity{

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_about);
                // ((ImageView)findViewById(R.id.imageview_About)).setBackgroundResource(R.drawable.about);
        }

}



______________________________
Filename : Activity_FactorDetails.java

package com.rexaTB.chekehandroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

public class Activity_FactorDetails extends Activity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_factordetails);

                final Button button_Send = (Button) findViewById(R.id.button_Send);
                final Button button_Delete = (Button) findViewById(R.id.button_Delete);

                Thread updateList = new Thread(new Runnable() {

                        @Override
                        public void run() {

                                try {
                                        final ListView listview_Products = (ListView) findViewById(R.id.listview_Products);

                                        MenuAdapter_Product menuadapter_main = new MenuAdapter_Product(getApplicationContext(),
                                                        Class_DownloadFactors.GetFactorItems(Class_About.LastFactorName));
                                        listview_Products.setAdapter(menuadapter_main);

                                        if (Class_About.isUserLogin)
                                                button_Send.setEnabled(true);
                                        button_Delete.setEnabled(true);

                                        button_Delete.setOnClickListener(new View.OnClickListener() {

                                                @Override
                                                public void onClick(View v) {

                                                        Class_FileManager CFM = new Class_FileManager();
                                                        CFM.Delete("List/" + Class_About.LastFactorName);

                                                        finish();
                                                }
                                        });

                                        button_Send.setOnClickListener(new View.OnClickListener() {
                                                @Override
                                                public void onClick(View v) {
                                                }
                                        });

                                } catch (Exception e) {
                                        Toast.makeText(new Activity().getApplicationContext(), "خطا در دریافت اطلاعات", Toast.LENGTH_LONG)
                                                        .show();
                                }
                        }
                });
                updateList.run();
        }
}
```

Filename : Activity_Factors.java

```
package com.rexaTB.chekehandroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.Toast;

public class Activity_Factors extends Activity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_factors);

                Thread updateList = new Thread(new Runnable() {

                        @Override
                        public void run() {

                                try {
                                        final ListView listview_ChooseProduct = (ListView) findViewById(R.id.listview_factors);

                                        MenuAdapter_Factor menuadapter_factor = new MenuAdapter_Factor(getApplicationContext(),
                                                        Class_DownloadFactors.GetFactors());
                                        listview_ChooseProduct.setAdapter(menuadapter_factor);

                                } catch (Exception e) {
                                        Toast.makeText(new Activity().getApplicationContext(), "خطا در دریافت اطلاعات از سرور",
                                                        Toast.LENGTH_LONG).show();
                                }
                        }
                });
                updateList.run();

        }

}
```

Filename : Activity_Login.java

```
package com.rexaTB.chekehandroid;

import android.app.Activity;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Activity_Login extends Activity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_login);

                ((Button) findViewById(R.id.button_Login)).setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                // Post login data to server

                                EditText et_Username = (EditText) findViewById(R.id.edittext_Username);
                                EditText et_Password = (EditText) findViewById(R.id.edittext_Password);

                                String Username = et_Username.getText().toString();
                                String Password = et_Password.getText().toString();

                                GET get = new GET();
                                String data = "";
                                try {
                                        data = get.execute(new StringBuilder("Username=" + Username + "&Password=" + Password).toString())
                                                        .get();
                                } catch (Exception e) {

                                }

                                if (data.equals("\"user\"")) {
                                        // Login done
                                        Class_About.isUserLogin = true;
                                        Class_About.Username = Username;
                                        Class_About.Password = Password;

                                        // Save login to text file
                                        try {
                                                Class_FileManager CFM = new Class_FileManager();
                                                CFM.Text_Write("Temp/Login", "Username=" + Username + "&Password=" + Password);
```

```
                    } catch (Exception e) {
                    }

                    // Return to main
                    finish();
                } else {
                    Toast.makeText(getApplicationContext(), "نام کاربری یا کلمه عبور معتبر نیست", Toast.LENGTH_LONG)
                            .show();
                }
            }
        });
    }

    class GET extends AsyncTask<String, String, String> {
        @Override
        protected String doInBackground(String... Params) {

            Class_WebClient client = new Class_WebClient();
            try {
                return client.Get("Login", "?" + Params[0]);
            } catch (Exception e) {
                return null;
            }
        }

    }

}
```

```
Filename : Activity_Main.java

package com.rexaTB.chekehandroid;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ListView;
import android.widget.Toast;

public class Activity_Main extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
Class_About.MyApplicationContext = getApplicationContext();

Class_FileManager CFM = new Class_FileManager();

CFM.CreateFolder("Temp");
CFM.CreateFolder("List");

final Button button_Confirm = (Button) findViewById(R.id.button_Confirm);
final Button button_Refresh = (Button) findViewById(R.id.button_Refresh);
final ListView listview_ChooseProduct = (ListView) findViewById(R.id.listview_Products);

try {int i = 0;
        for (String row : CFM.Text_Read("Temp/Login.txt").split("&")) {

                String[] fields = row.split("=");

                if (fields[0].equals("Username")) {
                        Class_About.Username = fields[1];
                        i++;
                } else if (fields[0].equals("Password")) {
                        Class_About.Password = fields[1];
                        i++;
                }
                if (i == 2)
                        Class_About.isUserLogin = true;
        }
} catch (Exception e) {

}

new Thread(new Runnable() {

        @Override
        public void run() {

                try {

                        MenuAdapter_Product menuadapter_main = new MenuAdapter_Product(getApplicationContext(),
                                        Class_DownloadProducts.GetOfflineList());
                        listview_ChooseProduct.setAdapter(menuadapter_main);

                        button_Confirm.setEnabled(true);

                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}).run();

button_Refresh.setOnClickListener(new View.OnClickListener() {
```

```
                @Override
                public void onClick(View v) {
                    Thread updateList = new Thread(new Runnable() {

                        @Override
                        public void run() {

                            try {

                                MenuAdapter_Product menuadapter_main = new MenuAdapter_Product(getApplicationContext(),
                                        Class_DownloadProducts.GetProducts());
                                listview_ChooseProduct.setAdapter(menuadapter_main);

                                button_Confirm.setEnabled(true);

                            } catch (Exception e) {
                                Toast.makeText(new Activity().getApplicationContext(), "خطا در دریافت اطلاعات از سرور",
                                        Toast.LENGTH_LONG).show();
                            }
                        }
                    });
                    updateList.run();
                }
            });
            button_Confirm.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View v) {
                    if (Class_About.ProductCount > 0) {
                        startActivity(new Intent(Activity_Main.this, Activity_SaveList.class));
                    } else {
                        Toast.makeText(Class_About.MyApplicationContext, "لطفا محصولی را به لیست خود اضافه کنید!",
                                Toast.LENGTH_LONG).show();
                    }
                }
            });
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items to the action bar if it is present.
            getMenuInflater().inflate(R.menu.main_menu, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks here. The action bar will
            // automatically handle clicks on the Home/Up button, so long
            // as you specify a parent activity in AndroidManifest.xml.
            int id = item.getItemId();
            if (id == R.id.item_UserRegister) {
```

```
                        startActivity(new Intent(Activity_Main.this, Activity_UserRegister.class));
                        return true;
                } else if (id == R.id.item_UserLogin) {
                        startActivity(new Intent(Activity_Main.this, Activity_Login.class));
                        return true;
                } else if (id == R.id.item_PrevLists) {
                        startActivity(new Intent(Activity_Main.this, Activity_Factors.class));
                        return true;
                } else if (id == R.id.item_About) {
                        startActivity(new Intent(Activity_Main.this, Activity_About.class));
                        return true;
                }
                return super.onOptionsItemSelected(item);
        }
}
```

____

Filename : Activity_ProductDetails.java

```
package com.rexaTB.chekehandroid;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLConnection;
import java.util.concurrent.ExecutionException;

import org.apache.http.client.ClientProtocolException;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.opengl.Visibility;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class Activity_ProductDetails extends Activity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_productdetails);
        final TextView textview_Title = (TextView) findViewById(R.id.textview_Title);
        final TextView textview_Description = (TextView) findViewById(R.id.textview_Description);

        Thread updateInfo = new Thread(new Runnable() {

                @Override
                public void run() {

                        textview_Description.setVisibility(1);
                        DownloadInfo di = new DownloadInfo();
                        try {
                                String data = di.execute("").get();
                                data = data.substring(1);
                                data = data.substring(0, data.length() - 1);

                                for (String field : data.split(",")) {
                                        String[] cell = field.split(":");

                                        cell[0] = cell[0].substring(1);
                                        cell[0] = cell[0].substring(0, cell[0].length() - 1);

                                        cell[1] = cell[1].substring(1);
                                        cell[1] = cell[1].substring(0, cell[1].length() - 1);

                                        if (cell[0].equals("ProductName")) {
                                                textview_Title.setText(cell[1]);
                                        } else if (cell[0].equals("ProductDescription")) {
                                                textview_Description.setText(cell[1]);
                                        }
                                }

                        } catch (Exception e) {
                                e.printStackTrace();
                        }

                        SetImage si = new SetImage();
                        try {
                                Toast.makeText(getApplicationContext(), si.execute("").get(), Toast.LENGTH_LONG).show();
                        } catch (Exception e) {
                        }
                }
        });
        updateInfo.run();
}

private class SetImage extends AsyncTask<String, String, String> {

        @Override
        protected String doInBackground(String... params) {
                try {
                        ImageView imageview_Picture = (ImageView) findViewById(R.id.imageview_Picture);
                        BitmapFactory.Options bmOptions;
                        bmOptions = new BitmapFactory.Options();
                        bmOptions.inSampleSize = 1;
                        Bitmap bm = LoadImage(Class_About.Serverlocation.substring(0, Class_About.Serverlocation.length() - 4)
                                        + "Download/ProductImage/" + Class_About.LastProductId, bmOptions);
                        imageview_Picture.setImageBitmap(bm);
                        return null;
                } catch (Exception e) {
                        return "تصویر بارگزاری نشد";
                }
        }

}

private Bitmap LoadImage(String URL, BitmapFactory.Options options) throws IOException {
        Bitmap bitmap = null;
        InputStream in = null;

        in = OpenHttpConnection(URL);
        bitmap = BitmapFactory.decodeStream(in, null, options);
        in.close();

        return bitmap;
}

private InputStream OpenHttpConnection(String strURL) throws IOException {
        InputStream inputStream = null;
        URL url = new URL(strURL);
        URLConnection conn = url.openConnection();

        HttpURLConnection httpConn = (HttpURLConnection) conn;
        httpConn.setRequestMethod("GET");
        httpConn.connect();

        if (httpConn.getResponseCode() == HttpURLConnection.HTTP_OK) {
                inputStream = httpConn.getInputStream();
        }

        return inputStream;
}

private class DownloadInfo extends AsyncTask<String, String, String> {

        @Override
        protected String doInBackground(String... params) {
                Class_WebClient webclient = new Class_WebClient();
                try {
                        return webclient.Get("Products", "/" + Class_About.LastProductId);
                } catch (ClientProtocolException e) {
                        Toast.makeText(Class_About.MyApplicationContext, "خطا در دریافت اطلاعات از سرور", Toast.LENGTH_LONG)
                                        .show();
                } catch (IOException e) {
```

```
                                Toast.makeText(Class_About.MyApplicationContext, "خطا در مدیریت سیستم فایل", Toast.LENGTH_LONG).show();
                        }
                        return null;
                }

        }
}
```

____

Filename : Activity_SaveList.java

```
package com.rexaTB.chekehandroid;

import java.io.IOException;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

public class Activity_SaveList extends Activity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_savelist);

                Thread updateList = new Thread(new Runnable() {

                        @Override
                        public void run() {

                                try {
                                        final ListView listview_ChooseProduct = (ListView) findViewById(R.id.listview_Products);
                                        MenuAdapter_Product menuadapter_main = new MenuAdapter_Product(getApplicationContext(), null);
                                        listview_ChooseProduct.setAdapter(menuadapter_main);
                                        ((Button) findViewById(R.id.button_Save)).setOnClickListener(new View.OnClickListener() {
                                                @Override
                                                public void onClick(View v) {
                                                        EditText edittext_Title = (EditText) findViewById(R.id.edittext_Title);
                                                        if (!"".equals(edittext_Title.getText().toString())) {

                                                                String data = "";
                                                                for (Class_Product product : MenuAdapter_Product.products) {
                                                                        if (product.Count > 0) {
                                                                                data += "[";
                                                                                data += "Id=" + product.Id;
                                        data += "&Name=" + product.Name;
                                        data += "&Count=" + product.Count;
                                        data += "]";
                                    }
                                }
                                Class_FileManager fileManager = new Class_FileManager();
                                try {
                                    fileManager.Text_Write("List/" + edittext_Title.getText().toString(), data);
                                    if (Class_About.isUserLogin) {
                                        startActivity(new Intent(Activity_SaveList.this, Activity_Factors.class));
                                    } else {
                                        finish();
                                    }
                                } catch (IOException e) {
                                    Toast.makeText(getApplicationContext(), "خطا در ذخیره فایل توسط سیستم عامل",
                                            Toast.LENGTH_LONG).show();
                                }
                            } else {
                                Toast.makeText(getApplicationContext(), "لطفا عنوانی را برای این فهرست انتخاب کنید",
                                        Toast.LENGTH_LONG).show();
                            }
                        }
                    });

                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(), "خطا", Toast.LENGTH_LONG).show();
                }
            }
        });
        updateList.run();

    }

}
```

____

Filename : Activity_UserRegister.java

```
package com.rexaTB.chekehandroid;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.app.Activity;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.EditTextPreference;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Activity_UserRegister extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_userregister);

        ((Button) findViewById(R.id.button_Register)).setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // Regsiter user
                try {
                    EditText et_FirstName = (EditText) findViewById(R.id.edittext_Firstname);
                    EditText et_LastName = (EditText) findViewById(R.id.edittext_Lastname);
                    EditText et_Password = (EditText) findViewById(R.id.edittext_Password);
                    EditText et_ConfirmPass = (EditText) findViewById(R.id.edittext_Confirm);
                    EditText et_Username = (EditText) findViewById(R.id.edittext_Username);

                    String PhoneNumber = "";

                    if (et_Password.getText().toString().equals(et_ConfirmPass.getText().toString())) {
                        List<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(5);
                        nameValuePairs.add(new BasicNameValuePair("FirstName", et_FirstName.getText().toString()));
                        nameValuePairs.add(new BasicNameValuePair("LastName", et_LastName.getText().toString()));
                        nameValuePairs.add(new BasicNameValuePair("Password", et_Password.getText().toString()));
                        nameValuePairs.add(new BasicNameValuePair("Username", et_Username.getText().toString()));
                        nameValuePairs.add(new BasicNameValuePair("PhoneNumber", PhoneNumber));

                        // Do post in background
                        POST post = new POST();
                        post.execute(nameValuePairs);

                        // redirect to login activity
                        startActivity(new Intent(Activity_UserRegister.this, Activity_Login.class));
                    } else {
                        Toast.makeText(getApplicationContext(), "کلمه عبور با تکرار آن همخوانی ندارد",
                                Toast.LENGTH_LONG).show();
                    }
                } catch (Exception e) {
                    Toast.makeText(getApplicationContext(), "مشکل هرچه باشد به خاطر سرور است", Toast.LENGTH_LONG)
                            .show();
                }

            }
        });
    }

    class MyParams {
        String Route;
        List<NameValuePair> NameValuePairs;

        public MyParams(String Route, List<NameValuePair> NameValuePairs) {
            this.Route = Route;
            this.NameValuePairs = NameValuePairs;
        }
    }

    class POST extends AsyncTask<List<NameValuePair>, String, String> {

        @Override
        protected String doInBackground(List<NameValuePair>... NameValuePairs) {
            Class_WebClient webClient = new Class_WebClient();
            try {
                return webClient.Post("Users", NameValuePairs[0]);
            } catch (Exception e) {
                return null;
            }
        }

    }
}
```

Filename : Class_About.java

```
package com.rexaTB.chekehandroid;

import java.io.File;

import android.content.Context;

public class Class_About {
    static public String Programmer = "MohammadReza Tayyebi - 3860995243";

    static public String Serverlocation = "http://192.168.29.1/api/";

    static public Context MyApplicationContext;

    static public boolean isUserLogin = false;
    static public String Username = "";
```

```
        static public String Password = "";

        static public int ProductCount;
        static public String LastProductId;
        static public String LastFactorName;
}
```

Filename : Class_DownloadFactors.java

```
package com.rexaTB.chekehandroid;

import java.io.File;
import java.io.FileNotFoundException;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.List;
import java.util.jar.Attributes.Name;
import java.util.regex.Pattern;

public class Class_DownloadFactors {
        static public Class_Factor[] GetFactors() {

                File file[] = new File(Class_About.MyApplicationContext.getFilesDir() + "/List").listFiles();
                List<Class_Factor> factor_List = new ArrayList<Class_Factor>();

                for (int i = 0; i < file.length; i++) {
                        Class_Factor factor = new Class_Factor();
                        factor.Name = file[i].getName();
                        factor_List.add(factor);
                }

                Class_Factor[] output = new Class_Factor[factor_List.size()];
                output = factor_List.toArray(output);
                return output;
        }

        static public Class_Product[] GetFactorItems(String FactorName) {

                Class_FileManager CFM = new Class_FileManager();

                List<Class_Product> products = new ArrayList<Class_Product>();

                try {
                        String data = CFM.Text_Read("List/" + FactorName);
                        data = data.substring(1);
                        data = data.substring(0, data.length() - 1);

                        String[] rows = data.split(Pattern.quote("]["));

                        data = null;
                        System.gc();

                        for (String row : rows) {
                                Class_Product product = new Class_Product();

                                for (String column : row.split("&")) {
                                        String[] values = column.split("=");

                                        if (values[0].equals("Id")) {
                                                product.Id = values[1];
                                        } else if (values[0].equals("Name")) {
                                                product.Name = values[1];
                                        } else if (values[0].equals("Count")) {
                                                product.Count = Integer.parseInt(values[1]);
                                        }
                                }
                                row = null;
                                System.gc();

                                products.add(product);
                        }

                } catch (FileNotFoundException e) {
                }
                Class_Product[] output = new Class_Product[products.size()];
                output = products.toArray(output);

                products = null;
                System.gc();

                return output;
        }
}
```

Filename : Class_DownloadProducts.java

```
package com.rexaTB.chekehandroid;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutionException;
import java.util.regex.Pattern;

import android.os.AsyncTask;

public class Class_DownloadProducts {
        static public Class_Product[] GetOfflineList() throws IOException {
```

```
        Class_FileManager CFM = new Class_FileManager();
        String data = CFM.Text_Read("Temp/_MainList.txt");

        List<Class_Product> product_List = new ArrayList<Class_Product>();

        String[] rows = data.split(Pattern.quote("},{"));


        for (String row : rows) {
                Class_Product product = new Class_Product();

                for (String column : row.split(",")) {
                        String[] values = column.split(":");

                        values[0] = values[0].substring(1);
                        values[0] = values[0].substring(0, values[0].length() - 1);

                        values[1] = values[1].substring(1);
                        values[1] = values[1].substring(0, values[1].length() - 1);

                        if (values[0].equals("ProductId")) {
                                product.Id = values[1];
                        } else if (values[0].equals("ProductName")) {
                                product.Name = values[1];
                        }
                }
                product_List.add(product);
        }

        Class_Product[] output = new Class_Product[product_List.size()];
        output = product_List.toArray(output);

        product_List = null;
        System.gc();

        CFM.Text_Write("_MainList", data);

        data = null;
        System.gc();

        return output;
}

static public Class_Product[] GetProducts() throws IOException, InterruptedException, ExecutionException {

        DownloadProducts dp = new DownloadProducts();

        String data = dp.execute("").get();

        data = data.substring(2);
        data = data.substring(0, data.length() - 2);
```

```
                List<Class_Product> product_List = new ArrayList<Class_Product>();

                String[] rows = data.split(Pattern.quote("},{"));

                dp = null;

                System.gc();

                for (String row : rows) {
                        Class_Product product = new Class_Product();

                        for (String column : row.split(",")) {
                                String[] values = column.split(":");

                                values[0] = values[0].substring(1);
                                values[0] = values[0].substring(0, values[0].length() - 1);

                                values[1] = values[1].substring(1);
                                values[1] = values[1].substring(0, values[1].length() - 1);

                                if (values[0].equals("ProductId")) {
                                        product.Id = values[1];
                                } else if (values[0].equals("ProductName")) {
                                        product.Name = values[1];
                                }
                        }
                        product_List.add(product);
                }

                Class_Product[] output = new Class_Product[product_List.size()];
                output = product_List.toArray(output);

                product_List = null;
                System.gc();

                Class_FileManager CFM = new Class_FileManager();
                CFM.Text_Write("Temp/_MainList", data);

                data = null;
                System.gc();

                return output;

        }
}

class DownloadProducts extends AsyncTask<String, String, String> {

        @Override
        protected String doInBackground(String... params) {
                try {
                        Class_WebClient webclient = new Class_WebClient();
                        return webclient.Get("Products", "");
                } catch (Exception e) {
                        return e.getMessage();
                }
        }

}
```

Filename : Class_Factor.java

```
package com.rexaTB.chekehandroid;

public class Class_Factor {
        public String Name;

        public String GetName() {
                return Name;
        }
}
```

Filename : Class_FileManager.java

```
package com.rexaTB.chekehandroid;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class Class_FileManager {
        public void Delete(String FileName) {
                new File(Class_About.MyApplicationContext.getFilesDir() + "/" + FileName).delete();
        }

        public String Text_Read(String FileName) throws FileNotFoundException {

                return new Scanner(new File(Class_About.MyApplicationContext.getFilesDir() + "/" + FileName), "UTF-8")
                                .useDelimiter("\\A").next();

        }

        public void Text_Write(String Filename, String Data) throws IOException {

                File outputFile = new File(Class_About.MyApplicationContext.getFilesDir(), Filename + ".txt");
                FileWriter writer = new FileWriter(outputFile);
                writer.append(Data);
                writer.flush();
                writer.close();
        }

        public void CreateFolder(String Name) {
                File dir = new File(Class_About.MyApplicationContext.getFilesDir() + "/" + Name);
                if (dir.exists() && dir.isDirectory()) {
                        // Do nothing!
                } else {
                        dir.mkdirs();
                }
        }
}



______________________________
Filename : Class_Product.java

package com.rexaTB.chekehandroid;

public class Class_Product {
        public String Id;
        public String Name;
        public int Count;

        // public Class_Product(String Id, String Name){
        // this.Id =Id;
        // this.Name = Name;
        // }

        public String getId(){
                return Id;
        }
        public String getName(){
                return Name;
        }
        public int getCount(){
                return Count;
        }
}



______________________________
Filename : Class_WebClient.java

package com.rexaTB.chekehandroid;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import android.os.AsyncTask;
import android.util.Log;

public class Class_WebClient {

        String Post(String Route, List<NameValuePair> nameValuePairs) throws ClientProtocolException, IOException {

                HttpClient httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(new String(Class_About.Serverlocation + Route));
                httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
                HttpResponse response = httpClient.execute(httpPost);
                return response.toString();

        }

        String Get(String Route, String Params) throws ClientProtocolException, IOException {
                HttpClient httpClient = new DefaultHttpClient();
                HttpGet httpGet = new HttpGet(new String(Class_About.Serverlocation + Route + Params));
                HttpResponse response = httpClient.execute(httpGet);
                BufferedReader rd = new BufferedReader(new InputStreamReader(response.getEntity().getContent()));

                StringBuffer result = new StringBuffer();
                String line = "";
                while ((line = rd.readLine()) != null) {
                        result.append(line);
                }
                return result.toString();
        }
}
```

Filename : MenuAdapter_Factor.java

```
package com.rexaTB.chekehandroid;
```

```
import java.util.Arrays;
import java.util.Comparator;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.TextView;

public class MenuAdapter_Factor extends BaseAdapter {

        static public Class_Factor[] factors;

        public MenuAdapter_Factor(Context context, Class_Factor[] rows) {

                MenuAdapter_Factor.factors = rows;

                inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        }

        @Override
        public int getCount() {
                return factors.length;
        }

        @Override
        public long getItemId(int position) {
                return position;
        }

        @Override
        public Object getItem(int position) {
                return factors[position];
        }

        static private LayoutInflater inflater;

        @Override
        public View getView(final int position, View convertView, final ViewGroup parent) {

                if (convertView == null)
                        convertView = inflater.inflate(R.layout.menuitem_factor, null);

                TextView textview_Title = (TextView) convertView.findViewById(R.id.textview_Title);
                textview_Title.setText(factors[position].GetName().substring(0, factors[position].GetName().length() - 4));

                Button button_Details = (Button) convertView.findViewById(R.id.button_Details);
                button_Details.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                                Class_About.LastFactorName = factors[position].GetName();
                                parent.getContext().startActivity(new Intent(parent.getContext(), Activity_FactorDetails.class));
                        }
                });

                return convertView;
        }

}
```

____
Filename : MenuAdapter_Product.java

```
package com.rexaTB.chekehandroid;

import java.util.Arrays;
import java.util.Comparator;
import java.util.zip.Inflater;

import android.app.Activity;
import android.app.Application;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MenuAdapter_Product extends BaseAdapter {

        static public Class_Product[] products;

        public MenuAdapter_Product(Context context, Class_Product[] rows) {
                if (rows != null) {
                        MenuAdapter_Product.products = rows;
                } else if (products != null) {
                        Arrays.sort(products, new Comparator<Class_Product>() {
                                @Override
                                public int compare(Class_Product o1, Class_Product o2) {
                                        return new StringBuilder().append(o2.getCount()).toString()
                                                        .compareTo(new StringBuilder().append(o1.getCount()).toString());
                                }
                        });
                }
                inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        }

        @Override
        public int getCount() {
                return products.length;
        }

        @Override
        public long getItemId(int position) {
                return position;
        }

        @Override
        public Object getItem(int position) {
                return products[position];
        }

        static private LayoutInflater inflater;

        @Override
        public View getView(final int position, View convertView, final ViewGroup parent) {

                if (convertView == null)
                        convertView = inflater.inflate(R.layout.menuitem_product, null);

                TextView textview_Title = (TextView) convertView.findViewById(R.id.textview_Title);
                textview_Title.setText(products[position].Name);
                final TextView textview_Count = (TextView) convertView.findViewById(R.id.textview_Count);
                textview_Count.setText(new StringBuilder().append(products[position].Count));

                ((Button) convertView.findViewById(R.id.button_Plus)).setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                products[position].Count += 1;
                                Class_About.ProductCount += 1;
                                textview_Count.setText(new StringBuilder().append(products[position].Count));
                        }
                });
                ((Button) convertView.findViewById(R.id.button_Minus)).setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                if (products[position].Count > 0) {
                                        products[position].Count -= 1;
                                        Class_About.ProductCount -= 1;
                                        textview_Count.setText(new StringBuilder().append(products[position].Count));
                                }
                        }
                });
```

```
                ((Button) convertView.findViewById(R.id.button_Details)).setOnClickListener(new View.OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                Class_About.LastProductId = products[position].Id;

                                Intent zoom = new Intent(parent.getContext(), Activity_ProductDetails.class);
                                parent.getContext().startActivity(zoom);

                        }
                });

                return convertView;
        }
}
```

```
==============================
Filename : android_values.txt
```

```
______________________________
Filename : dimens.xml

<resources>

    <!-- Default screen margins, per the Android Design guidelines. -->
    <dimen name="activity_horizontal_margin">16dp</dimen>
    <dimen name="activity_vertical_margin">16dp</dimen>

</resources>
```

```
______________________________
Filename : strings.xml

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

    <string name="app_name">چکه</string>
    <string name="About">درباره</string>
    <string name="Register">ثبت نام</string>
    <string name="Login">ورود به سیستم</string>
    <string name="Download">بارگذاری</string>
    <string name="Firstname">نام</string>
    <string name="Lastname">شهرت</string>
    <string name="Password">رمز عبور</string>
    <string name="Confirm">تائید</string>
    <string name="Username">نام کاربری</string>
    <string name="Ok">برو بریم</string>
    <string name="PrevLists">فهرست های پیشین</string>
    <string name="Save">ذخیره</string>
    <string name="Loaded">بارگزاری شد</string>
    <string name="NotLoaded">بارگزاری نشد</string>
    <string name="Plus">+</string>
    <string name="Minus">-</string>
    <string name="Zero">0</string>
    <string name="Details">جزئیات</string>
    <string name="Title">عنوان</string>
    <string name="Send">ارسال</string>
       <string name="Delete">حذف</string>
    <string name="Refresh">بارگزاری مجدد</string>

</resources>



______________________________
Filename : styles.xml

<resources>

    <!--
        Base application theme, dependent on API level. This theme is replaced
        by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
    -->
    <style name="AppBaseTheme" parent="android:Theme.Light">
        <!--
            Theme customizations available in newer API levels can go in
            res/values-vXX/styles.xml, while customizations related to
            backward-compatibility can go here.
        -->
    </style>

    <!-- Application theme. -->
    <style name="AppTheme" parent="AppBaseTheme">
        <!-- All customizations that are NOT specific to a particular API-level can go here. -->
    </style>

</resources>



==============================
Filename : api_controllers.txt




______________________________
Filename : AdminOwnerController.cs
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    public class AdminOwnerController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        [Authenticate]
        // GET: api/AdminOwner/5
        public string Get(string id)
        {
            try
            {
                // id is username
                return dc.vAdminOwner_GetStatus(dc.vHuman_GetIdFromUsername(id).FirstOrDefault().Id).FirstOrDefault().Status;
            }
            catch { }
            return null;
        }
    }
}
```

Filename : CitiesController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate(IsAdmin = true)]
    public class CitiesController : ApiController
    {
        DcDataContext dc = new DcDataContext();

        // GET: api/Cities
        public IEnumerable<vCity> Get()
        {
            return dc.vCities;
        }

        // GET: api/Cities/5
        public vCity Get(Guid id)
        {
            return dc.vCities.Where(x => x.Id == id).FirstOrDefault();
        }

        // POST: api/Cities
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                dc.City_Insert(values["name"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        // PUT: api/Cities/5
        public HttpResponseMessage Put(Guid id, FormDataCollection values)
        {
            try
            {
                dc.City_Update(id, values["name"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        // DELETE: api/Cities/5
        public HttpResponseMessage Delete(Guid id)
        {
            try
            {
                dc.City_Delete(id);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

```
______________________________
Filename : DownloadController.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace ChekehApi.Controllers
{
    public class DownloadController : Controller
    {
        DcDataContext dc = new DcDataContext();
        // GET: Download
        public ActionResult ProductImage(Guid id)
        {
            var file = dc.vProducts.Where(y => y.ProductId == id).FirstOrDefault();
            return File(file.Image.ToArray(), "image/png", file.ProductName);
        }
    }
}
```

```
______________________________
Filename : FeedBackController.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate]
    public class FeedBackController : ApiController
    {
        public DcDataContext dc = new DcDataContext();

        public IEnumerable<vFeedBack> GET(Guid Id) // Id is reciver
        {
            try {
                if (dc.vHuman_GetType(Id).FirstOrDefault().UserType.ToString() == "admin")
                    return (from a in dc.vFeedBacks select new vFeedBack { SenderId = a.SenderId, Id = a.Id, Value = a.Value }).OrderByDescending(m =>
m.Id).Take(10).OrderBy(n => n.Id);

                else return (from a in dc.vFeedBacks where (a.ReciverId == Id || a.SenderId == Id) select new vFeedBack { SenderId = a.SenderId, Id =
a.Id, Value = a.Value }).OrderByDescending(m => m.Id).Take(10).OrderBy(n => n.Id);
            }
            catch
            {
                return null;
            }
        }
        public HttpResponseMessage POST(Guid Id, FormDataCollection values) // Id is sender
        {
            try
            {
                if (values["ReciverId"] != "00000000-0000-0000-0000-000000000000")
                    dc.FeedBack_Insert(Id, new Guid(values["ReciverId"]), values["Value"].ToString());
                else if (values["ReciverId"] == "00000000-0000-0000-0000-000000000000")
                    dc.FeedBack_Insert(Id, null, values["Value"].ToString());
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

____
Filename : GroupsController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate(IsAdmin = true)]
    public class GroupsController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        public IEnumerable<vGroup> Get()
        {
            return dc.vGroups.ToList();
        }

        // GET: api/Groups/5
        public vGroup Get(Guid id)
        {
            return dc.vGroups.Where(x => x.Id == id).FirstOrDefault();
        }
        // POST: api/Groups
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                dc.Group_Insert(values["Name"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch {

                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }

        }
        // DELETE: api/Groups/5
        public HttpResponseMessage Delete(Guid id)
        {
            try
            {
                dc.Group_Delete(id);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch {

                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

____
Filename : HumanIdentifierController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    public class HumanIdentifierController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        [Authenticate]
        public Guid GET(string Id)
        {
            return dc.vHuman_GetIdFromUsername(Id).FirstOrDefault().Id;
        }
    }
}
```

____
Filename : Human_ImageController.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using System.Web.Mvc;

namespace ChekehApi.Controllers
{
    public class Human_ImageController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        [Authenticate]
        public HttpResponseMessage GET(string Id)
        {
            var image = dc.vHumans.Where(m => m.Id == new Guid(Id)).Select(z => z.Image).FirstOrDefault();

            HttpResponseMessage result = new HttpResponseMessage(HttpStatusCode.OK);
            try
            {
                result.Content = new StreamContent(new MemoryStream(image.ToArray()));
            }
            catch {
                return new HttpResponseMessage(HttpStatusCode.NoContent);
            }
            return result;
        }
    }
}
```

____
Filename : LoginController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
```

```
namespace ChekehApi.Controllers
{
    public class LoginController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        public string GET(string Username, string Password)
        {
            if (Convert.ToBoolean(dc.vCheckLogin(Username, Password)))
            {
                return dc.vHuman_GetType(dc.vHuman_GetIdFromUsername(Username).FirstOrDefault().Id).FirstOrDefault().UserType.ToString();
            }
            else
            {
                return null;
            }
        }
    }
}
```

____
Filename : OwnerController.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    public class Owner
    {
        public Guid Identifier { get; set; }

        public string Username { get; set; }
        public string Password { get; set; }
        public string NationalCode { get; set; }
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Address { get; set; }
        public string PostalCode { get; set; }
        public string PhoneNumber { get; set; }

        public HttpPostedFile Image { get; set; }
    }

    public class OwnerController : ApiController
    {
        [Authenticate]
        public Owner Get(Guid id)
        {
            var query = (from a in dc.vOwners where a.Id == id select a).FirstOrDefault();
            Owner owner = new Owner();
            owner.Identifier = query.Id;
            owner.Username = query.Username;
            owner.Password = null;
            owner.NationalCode = query.NationalCode;
            owner.FirstName = query.FirstName;
            owner.LastName = query.LastName;
            owner.Address = query.Address;
            owner.PostalCode = query.PostalCode;
            owner.PhoneNumber = query.PhoneNumber;
            return owner;
        }

        DcDataContext dc = new DcDataContext();
        public HttpResponseMessage Post([FromBody] Owner value)
        {
            try
            {
                byte[] data = null;
                if (value.Image != null)
                {
                    MemoryStream target = new MemoryStream();
                    value.Image.InputStream.CopyTo(target);
                    data = target.ToArray();
                }
                dc.Owner_Insert(value.Username, value.Password, value.FirstName, value.LastName, data, value.NationalCode, value.Address,
value.PostalCode, value.PhoneNumber);

                return Request.CreateResponse(HttpStatusCode.OK, value);
            }
            catch (Exception e)
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest, e.Message);
            }
        }

        //        [Authenticate]
        public HttpResponseMessage Put(Guid id, FormDataCollection values)
        {
            try
            {
                Owner value = new Owner { FirstName = values["FirstName"], Address = values["Address"], Identifier = id, LastName = values["LastName"],
NationalCode = values["NationalCode"], Password = values["Password"], PhoneNumber = values["PhoneNumber"], PostalCode = values["PostalCode"], Username
= values["Username"] };
                byte[] data = null;
                if (value.Image != null)
                {
                    MemoryStream target = new MemoryStream();
                    value.Image.InputStream.CopyTo(target);
                    data = target.ToArray();
                }
                dc.Owner_Update(id, value.Username, value.Password, value.FirstName, value.LastName, data, value.NationalCode, value.Address, 
value.PostalCode, value.PhoneNumber);
                return Request.CreateResponse(HttpStatusCode.OK, value);
            }
            catch (Exception e)
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest, e.Message);
            }
        }
    }
}
```

____

Filename : OwnerStatusController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate(IsAdmin = true)]
    public class OwnerStatusController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        public IEnumerable<vOwnerStatus> GET(string id = "NotConfirmed") // id is show all or not
        {
            var query = (from a in dc.vOwnerStatus select new vOwnerStatus { Id = a.Id, Username = a.Username, Status = a.Status });

            if (id == "Confirmed")
            {
                query = query.Where(a => a.Status == "Confirmed");
            }
            else if (id == "NotConfirmed")
            {
                query = query.Where(a => a.Status == "NotConfirmed");
            }
            return query.ToList();

        }
        public HttpResponseMessage POST(Guid id, FormDataCollection values) // id is ownerId
        {
            try
            {
                dc.vAdminOwner_SetStatus(id, null, values["Status"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

Filename : ProductsController.cs

```
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Threading.Tasks;
using System.Web;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate(IsAdmin = true)]
    public class ProductsController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        // GET: api/Products
        public IEnumerable<vProduct> Get()
        {
            return dc.vProducts;
        }

        // GET: api/Products/5
        public vProduct Get(Guid id)
        {
            return dc.vProducts.Where(x => x.ProductId == id).Select(x => new vProduct { ProductId = x.ProductId, ProductName = x.ProductName,
ProductDescription = x.ProductDescription }).FirstOrDefault();
        }

        // POST: api/Products
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                dc.Product_Update(new Guid(values["Id"]), values["Name"], null, values["Description"], new Guid(values["GroupId"]));
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        public async Task<HttpResponseMessage> Put(Guid id)
        {
            try
            {
                var values = HttpContext.Current.Request;

                byte[] bytes = null;
                if (values.Files.Count > 0)
                {
                   using (var binaryReader = new BinaryReader(values.Files[0].InputStream))
                    {
                        bytes = binaryReader.ReadBytes(values.Files[0].ContentLength);
                    }
                }
                dc.Product_Update (id, values["Name"], new System.Data.Linq.Binary(bytes), values["Description"], new Guid(values["GroupId"]));
                return new HttpResponseMessage { StatusCode = HttpStatusCode.OK };
            }
            catch
            {
                return new HttpResponseMessage { StatusCode = HttpStatusCode.BadRequest };
            }
        }

        // DELETE: api/Products/5
        public HttpResponseMessage Delete(Guid id)
        {
            try
            {
                dc.Product_Delete(id);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

Filename : StoreController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate]
    public class StoreController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        // GET: api/Store/5
        public vStore Get(Guid Id) // StoreId
        {
            return (from a in dc.vStores where a.StoreId == Id select a).FirstOrDefault();
        }
    }
}
```

Filename : StoreProductController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    public class StoreProductController : ApiController
    {

        DcDataContext dc = new DcDataContext();

        // GET: api/StoreProduct
        public IEnumerable<vProductAvailabiliityResult> Get(Guid Id) // Id is StoreId
        {
            return dc.vProductAvailabiliity(Id).ToList();
        }

        // POST: api/StoreProduct
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                bool wasAvailable = Convert.ToBoolean(values["Poseess"]);
                if (wasAvailable)
                {
                    dc.StoreProduct_Delete(new Guid(values["StoreId"]), new Guid(values["ProductId"]));
                }
                else
                {
                    dc.StoreProduct_Insert(new Guid(values["ProductId"]), new Guid(values["StoreId"]));
                }
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

    }
}
```

Filename : StoresController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
{
    [Authenticate]
    public class StoresController : ApiController
    {
        DcDataContext dc = new DcDataContext();
        // GET: api/Store
        public IEnumerable<vStore> Get(Guid Id) //OwnerId
        {
            var query = (from a in dc.vStores where a.OwnerId == Id select new vStore { StoreId = a.StoreId, StoreName = a.StoreName }).ToList();
            return query;
        }

        // POST: api/Store
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                dc.Store_Insert(values["Address"], values["postalCode"], new Guid(values["cityId"]), values["Name"],new Guid(values["OwnerId"]));
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        // PUT: api/Store
        public HttpResponseMessage Put(Guid id,FormDataCollection values)
        {
            try
            {
                dc.Store_Update(id, values["name"], values["Address"], values["postalCode"], new Guid(values["cityId"]));
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        // DELETE: api/Store/5
        public HttpResponseMessage Delete(Guid Id) //StoreId
        {
            try
            {
                dc.Store_Delete(Id);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);

            }
        }
    }
}
```

Filename : UsersController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Formatting;
using System.Web.Http;

namespace ChekehApi.Controllers
```

```
{
    public class UsersController : ApiController
    {
        DcDataContext dc = new DcDataContext();

        [Authenticate(IsAdmin = true)]
        public IEnumerable<vUser> Get()
        {
            return dc.vUsers.ToList();
        }

        // GET: api/Users/5

        [Authenticate]
        public vUser Get(Guid id)
        {
            return dc.vUsers.Where(x => x.Id == id).FirstOrDefault();
        }

        // POST: api/Users
        public HttpResponseMessage Post(FormDataCollection values)
        {
            try
            {
                dc.User_Insert(values["Username"], values["Password"], values["FirstName"], values["LastName"], null, values["PhoneNumber"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }

        // PUT: api/Users/5
        [Authenticate]
        public HttpResponseMessage Put(Guid id, FormDataCollection values)
        {
            try
            {
                dc.User_Update(id,values["Username"], values["Password"], values["FirstName"], values["LastName"], null, values["PhoneNumber"]);
                return Request.CreateResponse(HttpStatusCode.OK);
            }
            catch
            {
                return Request.CreateResponse(HttpStatusCode.BadRequest);
            }
        }
    }
}


==============================
Filename : api_main.txt




______________________________
Filename : AuthenticateAttribute.cs

using System;
using System.Net;
using System.Net.Http.Formatting;
using System.Web;
using System.Web.Http.Controllers;
using System.Web.Http.Filters;

namespace ChekehApi.Controllers
{
    [AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = true)]
    internal class AuthenticateAttribute : ActionFilterAttribute
    {
        public bool IsAdmin { get; set; }
        DcDataContext dc = new DcDataContext();
        public override void OnActionExecuting(HttpActionContext actionContext)
        {
            if (IsAdmin)
            {

            }
            /*
            try
            {
                var queryString = actionContext.Request.RequestUri.Query;
                if (!String.IsNullOrWhiteSpace(queryString))
                {
                    if (!Convert.ToBoolean(dc.CheckLogin(HttpUtility.ParseQueryString(queryString.Substring(1))["Username"],
HttpUtility.ParseQueryString(queryString.Substring(1))["Password"])))
                        actionContext.Response = new System.Net.Http.HttpResponseMessage(HttpStatusCode.Unauthorized);
                }
            }
            catch
            {
                actionContext.Response = new System.Net.Http.HttpResponseMessage(HttpStatusCode.Unauthorized);
            }
            base.OnActionExecuting(actionContext);
            */
        }
    }
}


______________________________
Filename : Web.config
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=301879
  -->
<configuration>
  <connectionStrings>
    <!--
    <add name="CheckehDBConnectionString" connectionString="Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=CheckehDB;User ID=IIS"
      providerName="System.Data.SqlClient" />
    -->
    <add name="CheckehDBConnectionString1" connectionString="Data Source=HP-PROBOOK;Initial Catalog=CheckehDB;User ID=iis;Password=iis"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" maxRequestLength="10000" />
  </system.web>
  <system.webServer>
    <handlers>
      <remove name="ExtensionlessUrlHandler-Integrated-4.0" />
      <remove name="OPTIONSVerbHandler" />
      <remove name="TRACEVerbHandler" />
      <add name="ExtensionlessUrlHandler-Integrated-4.0" path="*." verb="*" type="System.Web.Handlers.TransferRequestHandler"
preCondition="integratedMode,runtimeVersionv4.0" />
    </handlers>
  </system.webServer>
  <runtime>
    <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
      <dependentAssembly>
        <assemblyIdentity name="Newtonsoft.Json" culture="neutral" publicKeyToken="30ad4fe6b2a6aeed" />
        <bindingRedirect oldVersion="0.0.0.0-6.0.0.0" newVersion="6.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Optimization" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-1.1.0.0" newVersion="1.1.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="WebGrease" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-1.5.2.14234" newVersion="1.5.2.14234" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Helpers" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.WebPages" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
      </dependentAssembly>
      <dependentAssembly>
        <assemblyIdentity name="System.Web.Mvc" publicKeyToken="31bf3856ad364e35" />
        <bindingRedirect oldVersion="1.0.0.0-5.2.3.0" newVersion="5.2.3.0" />
      </dependentAssembly>
    </assemblyBinding>
  </runtime>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:6 /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb" type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:14 /nowarn:41008 /define:_MYTYPE=\&quot;Web\&quot; /optionInfer+" />
    </compilers>
  </system.codedom>
</configuration>
```

==============================

Filename : database.txt

```
USE [master]
GO
/****** Object:  Database [CheckehDB]    Script Date: 12/27/2015 12:15:28 AM ******/
CREATE DATABASE [CheckehDB]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'CheckehDB', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\CheckehDB.mdf' , SIZE = 10240KB , MAXSIZE =
UNLIMITED, FILEGROWTH = 1024KB ),
 FILEGROUP [FILES]
( NAME = N'ChekehDB_Files', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\ChekehDB_Files.ndf' , SIZE = 102400KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 50%)
 LOG ON
( NAME = N'CheckehDB_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\CheckehDB_log.ldf' , SIZE = 3456KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [CheckehDB] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [CheckehDB].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [CheckehDB] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [CheckehDB] SET ANSI_NULLS OFF
```

```
GO
ALTER DATABASE [CheckehDB] SET ANSI_PADDING OFF
GO
ALTER DATABASE [CheckehDB] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [CheckehDB] SET ARITHABORT OFF
GO
ALTER DATABASE [CheckehDB] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [CheckehDB] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [CheckehDB] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [CheckehDB] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [CheckehDB] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [CheckehDB] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [CheckehDB] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [CheckehDB] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [CheckehDB] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [CheckehDB] SET  DISABLE_BROKER
GO
ALTER DATABASE [CheckehDB] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [CheckehDB] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [CheckehDB] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [CheckehDB] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [CheckehDB] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [CheckehDB] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [CheckehDB] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [CheckehDB] SET RECOVERY SIMPLE
GO
ALTER DATABASE [CheckehDB] SET  MULTI_USER
GO
ALTER DATABASE [CheckehDB] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [CheckehDB] SET DB_CHAINING OFF
GO
ALTER DATABASE [CheckehDB] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [CheckehDB] SET TARGET_RECOVERY_TIME = 0 SECONDS
```

```
GO
ALTER DATABASE [CheckehDB] SET DELAYED_DURABILITY = DISABLED
GO
USE [CheckehDB]
GO
/****** Object:  UserDefinedFunction [dbo].[NationalCodeValidation]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author:            Mohammad Reza
-- Description: NationalCode validation
-- =============================================
CREATE FUNCTION [dbo].[NationalCodeValidation]
(
       @input numeric
)
RETURNS int
AS
BEGIN

declare @lenght int
set @lenght = DATALENGTH(CAST(@input as varchar))

IF (@lenght = 10)
return 1

return 0
END


GO
/****** Object:  Table [dbo].[AdminOwnerSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[AdminOwnerSet](
       [Id] [int] IDENTITY(1,1) NOT NULL,
       [AdminId] [uniqueidentifier] NULL,
       [OwnerId] [uniqueidentifier] NOT NULL,
       [Status] [varchar](50) NOT NULL,
 CONSTRAINT [PK_AdminOwnerSet] PRIMARY KEY CLUSTERED
(
       [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
```

```
GO
/****** Object:  Table [dbo].[AdminSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AdminSet](
	[Id] [uniqueidentifier] NOT NULL,
 CONSTRAINT [PK_AdminSet] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[CitySet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CitySet](
	[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF_City_Id]  DEFAULT (newid()),
	[Name] [nvarchar](50) NOT NULL,
 CONSTRAINT [PK_City] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[FactorProductSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FactorProductSet](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[FactorId] [int] NOT NULL,
	[ProductId] [uniqueidentifier] NOT NULL,
	[Count] [int] NOT NULL,
	[Price] [money] NULL,
 CONSTRAINT [PK_FactoryProduct] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[FactorSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

```
GO
CREATE TABLE [dbo].[FactorSet](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [CreateDate] [datetime] NOT NULL,
        [StoreId] [uniqueidentifier] NOT NULL,
        [UserId] [uniqueidentifier] NOT NULL,
 CONSTRAINT [PK_FactorSet] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[FeedbackSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FeedbackSet](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [SenderId] [uniqueidentifier] NOT NULL,
        [ReciverId] [uniqueidentifier] NULL,
        [Value] [nvarchar](max) NOT NULL,
 CONSTRAINT [PK_FeedbackSet] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
/****** Object:  Table [dbo].[GroupSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GroupSet](
        [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF_GroupSet_Id]  DEFAULT (newid()),
        [Name] [nvarchar](100) NOT NULL,
 CONSTRAINT [PK_GroupSet] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[HumanSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[HumanSet](
        [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF_HumanSet_Id]  DEFAULT (newid()),
        [Username] [nvarchar](50) NOT NULL,
        [Password] [nvarchar](150) NOT NULL,
        [FirstName] [nvarchar](50) NOT NULL,
        [LastName] [nvarchar](60) NOT NULL,
        [Image] [varbinary](max) NULL,
 CONSTRAINT [PK_HumanSet] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
UNIQUE NONCLUSTERED
(
        [Username] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [FILES]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[OwnerSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[OwnerSet](
        [Id] [uniqueidentifier] NOT NULL,
        [Address] [nvarchar](max) NOT NULL,
        [PostalCode] [varchar](50) NULL,
        [PhoneNumber] [varchar](50) NOT NULL,
        [NationalCode] [varchar](50) NOT NULL,
 CONSTRAINT [PK_OwnerSet] PRIMARY KEY CLUSTERED
(
        [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[OwnerStoreSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[OwnerStoreSet](
        [Id] [int] IDENTITY(1,1) NOT NULL,
        [StoreId] [uniqueidentifier] NOT NULL,
        [OwnerId] [uniqueidentifier] NOT NULL,
 CONSTRAINT [PK_OwnerStoreSet] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
 CONSTRAINT [IX_OwnerStoreUsUnique] UNIQUE NONCLUSTERED
(
	[OwnerId] ASC,
	[StoreId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  Table [dbo].[ProductSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ProductSet](
	[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF_ProductSet_Id]  DEFAULT (newid()),
	[Name] [nvarchar](150) NOT NULL,
	[Image] [varbinary](max) NULL,
	[Description] [nvarchar](max) NOT NULL,
	[GroupId] [uniqueidentifier] NULL,
 CONSTRAINT [PK_ProductSet] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[StoreProductSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[StoreProductSet](
	[Id] [int] IDENTITY(1,1) NOT NULL,
	[ProductId] [uniqueidentifier] NOT NULL,
	[StoreId] [uniqueidentifier] NOT NULL,
 CONSTRAINT [PK_StoreProductSet] PRIMARY KEY CLUSTERED
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
 CONSTRAINT [UniqueStoreProduct] UNIQUE NONCLUSTERED
(
	[ProductId] ASC,
	[StoreId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
/****** Object:  Table [dbo].[StoreSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[StoreSet](
	[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF_Store_Id]  DEFAULT (newid()),
	[Address] [nvarchar](max) NOT NULL,
	[PostalCode] [varchar](50) NOT NULL,
	[CityId] [uniqueidentifier] NOT NULL,
	[Name] [nvarchar](150) NOT NULL,
 CONSTRAINT [PK_Store] PRIMARY KEY CLUSTERED 
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[UserSet]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UserSet](
	[Id] [uniqueidentifier] NOT NULL,
	[RegisterDate] [datetime] NOT NULL CONSTRAINT [DF_UserSet_RegisterDate]  DEFAULT (getdate()),
	[PhoneNumber] [nchar](20) NOT NULL,
 CONSTRAINT [PK_UserSet] PRIMARY KEY CLUSTERED 
(
	[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/****** Object:  View [dbo].[vGroup]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create view [dbo].[vGroup]
as
select Id, Name from GroupSet

GO
/****** Object:  View [dbo].[vProduct]    Script Date: 12/27/2015 12:15:28 AM ******/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vProduct]
AS
SELECT        dbo.ProductSet.Id AS ProductId, dbo.ProductSet.Name AS ProductName, dbo.ProductSet.Description AS ProductDescription, 
dbo.ProductSet.GroupId, dbo.vGroup.Name AS GroupName, dbo.ProductSet.Image
FROM            dbo.ProductSet INNER JOIN
                         dbo.vGroup ON dbo.ProductSet.GroupId = dbo.vGroup.Id

GO
/****** Object:  View [dbo].[vOwner]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vOwner]
AS
SELECT        dbo.HumanSet.Id, dbo.HumanSet.Username, dbo.HumanSet.LastName, dbo.HumanSet.FirstName, dbo.HumanSet.[Image], dbo.OwnerSet.NationalCode, 
dbo.OwnerSet.[Address], dbo.OwnerSet.PostalCode, 
                         dbo.OwnerSet.PhoneNumber
FROM            dbo.OwnerSet INNER JOIN
                         dbo.HumanSet ON dbo.OwnerSet.Id = dbo.HumanSet.Id

GO
/****** Object:  View [dbo].[vOwnerStatus]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vOwnerStatus]
AS
SELECT        dbo.vOwner.Id, dbo.vOwner.Username, dbo.vOwner.LastName, dbo.vOwner.FirstName, dbo.AdminOwnerSet.Status, dbo.AdminOwnerSet.AdminId
FROM            dbo.AdminOwnerSet INNER JOIN
                         dbo.vOwner ON dbo.AdminOwnerSet.OwnerId = dbo.vOwner.Id


GO
/****** Object:  View [dbo].[vAdmin]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vAdmin]
AS
SELECT        dbo.HumanSet.Id, dbo.HumanSet.Username, dbo.HumanSet.FirstName, dbo.HumanSet.LastName, dbo.HumanSet.Image
FROM            dbo.AdminSet INNER JOIN
                         dbo.HumanSet ON dbo.AdminSet.Id = dbo.HumanSet.Id


GO
```

```
/****** Object:  View [dbo].[vCity]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vCity]
AS
SELECT        Id, Name
FROM            dbo.CitySet


GO
/****** Object:  View [dbo].[vFeedBack]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vFeedBack]
AS
SELECT        Id, SenderId, ReciverId, Value
FROM            dbo.FeedbackSet


GO
/****** Object:  View [dbo].[vHuman]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vHuman]
       AS SELECT HumanSet.Id, HumanSet.FirstName , HumanSet.LastName , HumanSet.Username , HumanSet.[Image] FROM HumanSet

GO
/****** Object:  View [dbo].[vStore]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create view [dbo].[vStore]
as

select

StoreId, StoreSet.Name as [StoreName], StoreSet.[Address] as [StoreAddress], StoreSet.PostalCode as [StorePostalCode],
CitySet.Name as [CityName],
OwnerSet.Id as [OwnerId]

from OwnerStoreSet

inner join StoreSet on StoreSet.Id = OwnerStoreSet.StoreId
inner join CitySet  on StoreSet.CityId = CitySet.Id
inner join OwnerSet on OwnerSet.Id = OwnerStoreSet.OwnerId
```

```sql
GO
/****** Object:  View [dbo].[vUser]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[vUser]
AS
SELECT        dbo.HumanSet.Id, dbo.HumanSet.Username, dbo.HumanSet.FirstName, dbo.HumanSet.LastName, dbo.UserSet.RegisterDate, dbo.UserSet.PhoneNumber, 
dbo.HumanSet.Image
FROM            dbo.UserSet INNER JOIN
                         dbo.HumanSet ON dbo.UserSet.Id = dbo.HumanSet.Id

GO
ALTER TABLE [dbo].[FactorSet] ADD  CONSTRAINT [DF_FactorSet_CreateDate]  DEFAULT (getdate()) FOR [CreateDate]
GO
ALTER TABLE [dbo].[AdminOwnerSet]  WITH CHECK ADD  CONSTRAINT [FK_AdminOwnerSet_AdminSet] FOREIGN KEY([AdminId])
REFERENCES [dbo].[AdminSet] ([Id])
GO
ALTER TABLE [dbo].[AdminOwnerSet] CHECK CONSTRAINT [FK_AdminOwnerSet_AdminSet]
GO
ALTER TABLE [dbo].[AdminOwnerSet]  WITH CHECK ADD  CONSTRAINT [FK_AdminOwnerSet_OwnerSet] FOREIGN KEY([OwnerId])
REFERENCES [dbo].[OwnerSet] ([Id])
GO
ALTER TABLE [dbo].[AdminOwnerSet] CHECK CONSTRAINT [FK_AdminOwnerSet_OwnerSet]
GO
ALTER TABLE [dbo].[AdminSet]  WITH CHECK ADD  CONSTRAINT [FK_AdminSet_HumanSet] FOREIGN KEY([Id])
REFERENCES [dbo].[HumanSet] ([Id])
GO
ALTER TABLE [dbo].[AdminSet] CHECK CONSTRAINT [FK_AdminSet_HumanSet]
GO
ALTER TABLE [dbo].[FactorProductSet]  WITH CHECK ADD  CONSTRAINT [FK_FactoryProduct_FactorSet] FOREIGN KEY([FactorId])
REFERENCES [dbo].[FactorSet] ([Id])
GO
ALTER TABLE [dbo].[FactorProductSet] CHECK CONSTRAINT [FK_FactoryProduct_FactorSet]
GO
ALTER TABLE [dbo].[FactorProductSet]  WITH CHECK ADD  CONSTRAINT [FK_FactoryProduct_ProductSet] FOREIGN KEY([ProductId])
REFERENCES [dbo].[ProductSet] ([Id])
GO
ALTER TABLE [dbo].[FactorProductSet] CHECK CONSTRAINT [FK_FactoryProduct_ProductSet]
GO
ALTER TABLE [dbo].[FactorSet]  WITH CHECK ADD  CONSTRAINT [FK_FactorSet_StoreSet] FOREIGN KEY([StoreId])
REFERENCES [dbo].[StoreSet] ([Id])
GO
ALTER TABLE [dbo].[FactorSet] CHECK CONSTRAINT [FK_FactorSet_StoreSet]
GO
ALTER TABLE [dbo].[FactorSet]  WITH CHECK ADD  CONSTRAINT [FK_FactorSet_UserSet] FOREIGN KEY([UserId])
REFERENCES [dbo].[UserSet] ([Id])
GO
ALTER TABLE [dbo].[FactorSet] CHECK CONSTRAINT [FK_FactorSet_UserSet]
GO
```

```
ALTER TABLE [dbo].[FeedbackSet]  WITH CHECK ADD  CONSTRAINT [FK_FeedbackSet_HumanSet] FOREIGN KEY([SenderId])
REFERENCES [dbo].[HumanSet] ([Id])
GO
ALTER TABLE [dbo].[FeedbackSet] CHECK CONSTRAINT [FK_FeedbackSet_HumanSet]
GO
ALTER TABLE [dbo].[FeedbackSet]  WITH CHECK ADD  CONSTRAINT [FK_FeedbackSet_HumanSet1] FOREIGN KEY([ReciverId])
REFERENCES [dbo].[HumanSet] ([Id])
GO
ALTER TABLE [dbo].[FeedbackSet] CHECK CONSTRAINT [FK_FeedbackSet_HumanSet1]
GO
ALTER TABLE [dbo].[OwnerSet]  WITH CHECK ADD  CONSTRAINT [FK_OwnerSet_HumanSet] FOREIGN KEY([Id])
REFERENCES [dbo].[HumanSet] ([Id])
GO
ALTER TABLE [dbo].[OwnerSet] CHECK CONSTRAINT [FK_OwnerSet_HumanSet]
GO
ALTER TABLE [dbo].[OwnerStoreSet]  WITH CHECK ADD  CONSTRAINT [FK_OwnerStoreSet_OwnerSet1] FOREIGN KEY([OwnerId])
REFERENCES [dbo].[OwnerSet] ([Id])
GO
ALTER TABLE [dbo].[OwnerStoreSet] CHECK CONSTRAINT [FK_OwnerStoreSet_OwnerSet1]
GO
ALTER TABLE [dbo].[OwnerStoreSet]  WITH CHECK ADD  CONSTRAINT [FK_OwnerStoreSet_StoreSet] FOREIGN KEY([StoreId])
REFERENCES [dbo].[StoreSet] ([Id])
GO
ALTER TABLE [dbo].[OwnerStoreSet] CHECK CONSTRAINT [FK_OwnerStoreSet_StoreSet]
GO
ALTER TABLE [dbo].[ProductSet]  WITH CHECK ADD  CONSTRAINT [FK_ProductSet_GroupSet] FOREIGN KEY([GroupId])
REFERENCES [dbo].[GroupSet] ([Id])
GO
ALTER TABLE [dbo].[ProductSet] CHECK CONSTRAINT [FK_ProductSet_GroupSet]
GO
ALTER TABLE [dbo].[StoreProductSet]  WITH CHECK ADD  CONSTRAINT [FK_StoreProductSet_ProductSet1] FOREIGN KEY([ProductId])
REFERENCES [dbo].[ProductSet] ([Id])
GO
ALTER TABLE [dbo].[StoreProductSet] CHECK CONSTRAINT [FK_StoreProductSet_ProductSet1]
GO
ALTER TABLE [dbo].[StoreProductSet]  WITH CHECK ADD  CONSTRAINT [FK_StoreProductSet_StoreSet1] FOREIGN KEY([StoreId])
REFERENCES [dbo].[StoreSet] ([Id])
GO
ALTER TABLE [dbo].[StoreProductSet] CHECK CONSTRAINT [FK_StoreProductSet_StoreSet1]
GO
ALTER TABLE [dbo].[StoreSet]  WITH CHECK ADD  CONSTRAINT [FK_Store_City] FOREIGN KEY([CityId])
REFERENCES [dbo].[CitySet] ([Id])
GO
ALTER TABLE [dbo].[StoreSet] CHECK CONSTRAINT [FK_Store_City]
GO
ALTER TABLE [dbo].[UserSet]  WITH CHECK ADD  CONSTRAINT [FK_UserSet_UserSet] FOREIGN KEY([Id])
REFERENCES [dbo].[HumanSet] ([Id])
GO
ALTER TABLE [dbo].[UserSet] CHECK CONSTRAINT [FK_UserSet_UserSet]
GO
ALTER TABLE [dbo].[AdminOwnerSet]  WITH CHECK ADD  CONSTRAINT [CK_Status] CHECK  (([Status]='Confirmed' OR [Status]='NotConfirmed'))
GO
```

```
ALTER TABLE [dbo].[AdminOwnerSet] CHECK CONSTRAINT [CK_Status]
GO
/****** Object:  StoredProcedure [dbo].[Admin_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Admin_Delete]
	@Id uniqueidentifier
AS
BEGIN
	begin tran
	Delete HumanSet where Id=@Id
	delete AdminSet where Id=@Id
	commit
END


GO
/****** Object:  StoredProcedure [dbo].[Admin_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Admin_Insert]
@Username nvarchar(50),
@Password nvarchar(150),
@FirstName nvarchar(50),
@LastName nvarchar(60),
@Image varbinary(max)
AS
BEGIN
	declare @Id uniqueidentifier
	set @Id = NEWID()

	begin tran
	insert into HumanSet(Id, Username, [Password], FirstName, LastName, [Image]) values (@Id, @Username, @Password, @FirstName, @LastName, @Image)
	insert into AdminSet(Id) values (@Id)
	commit
END


GO
/****** Object:  StoredProcedure [dbo].[Admin_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Admin_Update]
@Id uniqueidentifier,
@Username nvarchar(50),
@Password nvarchar(150),
@FirstName nvarchar(50),
@LastName nvarchar(60),
@Image varbinary(max)
AS
BEGIN

       UPDATE HumanSet Set Username=@Username, [Password]=@Password, FirstName=@FirstName,@LastName=@LastName, [Image]=@Image WHERE Id=@Id
       -- UPDATE OwnerSet where Id=@Id

END


GO
/****** Object:  StoredProcedure [dbo].[City_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[City_Delete]
       @Id uniqueidentifier
AS
BEGIN
delete CitySet where Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[City_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[City_Insert]
@Name nvarchar(50)
AS
BEGIN
       insert into CitySet(Id, Name) VALUES(NEWID(), @Name)
END


GO
/****** Object:  StoredProcedure [dbo].[City_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[City_Update]
       @id uniqueidentifier,
       @Name nvarchar(50)
AS
BEGIN
       UPDATE CitySet Set Name=@Name WHERE Id=@Id
```

```
END


GO
/****** Object:  StoredProcedure [dbo].[Factor_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Factor_Insert]
       @StoreId uniqueidentifier,
       @UserId uniqueidentifier
AS
BEGIN
       Insert into FactorSet(CreateDate, StoreId, UserId) values (GETDATE(), @StoreId, @UserId)
END


GO
/****** Object:  StoredProcedure [dbo].[FactorProduct_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FactorProduct_Delete]
       @Id int
AS
BEGIN
       Delete FactorProductSet where Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[FactorProduct_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FactorProduct_Insert]
       @FactorId int,
       @productId     uniqueidentifier,
       @Count int,
       @Price money
AS
BEGIN
       Insert into FactorProductSet(FactorId, ProductId, [Count] , Price) values(@FactorId, @ProductId, @Count, @Price)
END


GO
/****** Object:  StoredProcedure [dbo].[FeedBack_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
```

```
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FeedBack_Delete]
@Id int
as
BEGIN
Delete from FeedBackSet where Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[FeedBack_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[FeedBack_Insert]
@SenderId uniqueidentifier,
@ReciverId uniqueidentifier null,
@Value nvarchar(MAX)
as
BEGIN
insert into FeedbackSet(SenderId, ReciverId, Value) VALUES(@SenderId, @ReciverId, @Value)
END


GO
/****** Object:  StoredProcedure [dbo].[Group_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Group_Delete]
	@id uniqueidentifier
AS
BEGIN
	DELETE GroupSet WHERE Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[Group_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Group_Insert]
@Name nvarchar(50)
AS
BEGIN
	insert into GroupSet(Id, Name) VALUES(NEWID(), @Name)
```

```
END


GO
/****** Object:  StoredProcedure [dbo].[Group_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Group_Update]
	@id uniqueidentifier,
	@Name nvarchar(50)
AS
BEGIN
	UPDATE GroupSet Set Name=@Name WHERE Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[Owner_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Owner_Delete]
	@Id uniqueidentifier
AS
BEGIN
	begin tran
	Delete HumanSet where Id=@Id
	delete OwnerSet where Id=@Id
	commit
END


GO
/****** Object:  StoredProcedure [dbo].[Owner_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Owner_Insert]
@Username nvarchar(50),
@Password nvarchar(150),
@FirstName nvarchar(50),
@LastName nvarchar(60),
@Image varbinary(max),
@NationalCode varchar(50),
@Address nvarchar(max),
@PostalCode varchar(50),
@PhoneNumber varchar(50)
AS
```

```
BEGIN
        declare @Id uniqueidentifier
        set @Id = NEWID()

        IF (dbo.NationalCodeValidation(@NationalCode) = 1)
        begin
        begin tran
                insert into HumanSet(Id, Username, [Password], FirstName, LastName, [Image]) values (@Id, @Username, @Password, @FirstName, @LastName, 
@Image)
                insert into OwnerSet(Id, NationalCode, [Address], PostalCode, PhoneNumber) values (@Id, @NationalCode, @Address, @PostalCode, 
@PhoneNumber)

                -- SET Status
                Declare @RC int
                EXECUTE @RC = [dbo].[AdminOwner_SetStatus]
                   @Id
                  ,null
                  ,'NotConfirmed'
                End
        commit
END


GO
/****** Object:  StoredProcedure [dbo].[Owner_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Owner_Update]
        @Id uniqueidentifier,
        @Username nvarchar(50),
        @Password nvarchar(150),
        @FirstName nvarchar(50),
        @LastName nvarchar(60),
        @Image varbinary(max),
        @NationalCode varchar(50),
        @Address nvarchar(max),
        @PostalCode varchar(50),
        @PhoneNumber varchar(50)
AS
BEGIN
        IF (dbo.NationalCodeValidation(@NationalCode) = 1)
        begin
        begin tran
                UPDATE HumanSet Set Username=@Username, [Password]=@Password, FirstName=@FirstName,@LastName=@LastName, [Image]=@Image WHERE Id=@Id
                update OwnerSet set NationalCode=@NationalCode, [Address]=@Address, PostalCode=@PostalCode, PhoneNumber=@PhoneNumber where Id=@Id

                -- set status
                Declare @RC int
                EXECUTE @RC = [dbo].[AdminOwner_SetStatus]
                        @Id
```

```
                      ,null
                      ,'NotConfirmed'
               End
        commit
        end


GO
/****** Object:  StoredProcedure [dbo].[Product_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Product_Delete]
@Id uniqueidentifier
AS
BEGIN
DELETE ProductSet WHERE Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[Product_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Product_Insert]
@Name nvarchar(150),
@Image varbinary(max),
@Description nvarchar(max),
@GroupId uniqueidentifier
AS
BEGIN
INSERT INTO ProductSet(Id, Name,[Image], [Description],[GroupId]) VALUES(NEWID(), @Name, @Image, @Description, @GroupId)
END


GO
/****** Object:  StoredProcedure [dbo].[Product_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Product_Update]
@Id uniqueidentifier,
@Name nvarchar(150),
@Image varbinary(max),
@Description nvarchar(max),
@GroupId uniqueidentifier
AS
BEGIN
```

```sql
UPDATE ProductSet SET Name=@Name, [Image]=@Image, [Description]=@Description, GroupId=@GroupId WHERE Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[Store_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Store_Delete]
@Id uniqueidentifier
AS
BEGIN
begin tran
begin try
delete OwnerStoreSet where StoreId=@Id
delete StoreSet where Id=@Id
commit
end try
begin catch
rollback
end catch
END


GO
/****** Object:  StoredProcedure [dbo].[Store_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Store_Insert]
@Address nvarchar(max),
@PostalCode varchar(50),
@CityId uniqueidentifier,
@Name nvarchar(150),
@OwnerId uniqueidentifier
AS
BEGIN
Begin Tran
begin try
	declare @StoreId uniqueidentifier
	set @StoreId = newid()
	insert into StoreSet (Id, [Address], PostalCode, CityId, Name) Values (@StoreId, @Address, @PostalCode,@CityId, @Name)
	insert into OwnerStoreSet( OwnerId, StoreId) values ( @OwnerId, @StoreId)
commit
end try
begin catch
rollback
end catch
```

```
END


GO
/****** Object:  StoredProcedure [dbo].[Store_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Store_Update]
@Id uniqueidentifier,
@Name nvarchar(150),
@Address nvarchar(max),
@PostalCode varchar(50),
@CityId uniqueidentifier
AS
BEGIN
update StoreSet set [Address]=@Address, PostalCode=@PostalCode, CityId=@CityId where Id=@Id
END


GO
/****** Object:  StoredProcedure [dbo].[StoreProduct_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[StoreProduct_Delete]
	@StoreId uniqueidentifier,
	@ProductId uniqueidentifier
AS
BEGIN
	Delete StoreProductSet where StoreId=@StoreId and ProductId=@ProductId
END


GO
/****** Object:  StoredProcedure [dbo].[StoreProduct_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[StoreProduct_Insert]
@ProductId uniqueidentifier,
@StoreId uniqueidentifier
AS
BEGIN
Insert into StoreProductSet(ProductId, StoreId) values (@ProductId, @StoreId)
END


GO
```

```
/****** Object:  StoredProcedure [dbo].[User_Delete]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create PROCEDURE [dbo].[User_Delete]
@Id uniqueidentifier
AS
BEGIN
        begin tran

        begin try
                delete UserSet where Id=@Id
                delete HumanSet where Id=@Id
                commit
        end try
        begin catch
                rollback
        end catch
END


GO
/****** Object:  StoredProcedure [dbo].[User_Insert]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[User_Insert]
        @Username nvarchar(50),
        @Password nvarchar(150),
        @FirstName nvarchar(50),
        @LastName nvarchar(60),
        @Image varbinary(max),
        @PhoneNumber nchar(10)
AS
BEGIN
        declare @Id uniqueidentifier
        set @Id = NEWID()

        begin tran

        begin try
                insert into HumanSet(Id, Username, [Password], FirstName, LastName, [Image]) values (@Id, @Username, @Password, @FirstName, @LastName, 
@Image)
                insert into UserSet(Id, RegisterDate, PhoneNumber) values (@Id, GETDATE(), @PhoneNumber)
                commit
        end try
        begin catch
                rollback
        end catch
END
```

```
GO
/****** Object:  StoredProcedure [dbo].[User_Update]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[User_Update]
	@Id uniqueidentifier,
	@Username nvarchar(50),
	@Password nvarchar(150),
	@FirstName nvarchar(50),
	@LastName nvarchar(60),
	@Image varbinary(max),
	@PhoneNumber nchar(10)
AS
BEGIN
	begin tran

	begin try
		update HumanSet set Username=@Username, [Password]=@Password, FirstName=@FirstName,@LastName=@LastName, [Image]=@Image where Id=@Id
		update UserSet set PhoneNumber=@PhoneNumber where Id=@Id
		commit
	end try
	begin catch
		rollback
	end catch
END


GO
/****** Object:  StoredProcedure [dbo].[vAdminOwner_GetStatus]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vAdminOwner_GetStatus]
	@OwnerId uniqueidentifier
AS
BEGIN
	Declare @Status varchar(50)
	select top 1 [Status] from AdminOwnerSet where OwnerId=@OwnerId order by Id desc
END

GO
/****** Object:  StoredProcedure [dbo].[vAdminOwner_SetStatus]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vAdminOwner_SetStatus]
```

```
@OwnerId uniqueidentifier,
@AdminId uniqueidentifier null,
@Status varchar(50)
AS
BEGIN
begin tran
delete from AdminOwnerSet where OwnerId = @OwnerId
insert into AdminOwnerSet(OwnerId,AdminId,[Status]) values(@OwnerId,@AdminId,@Status)
commit
END


GO
/****** Object:  StoredProcedure [dbo].[vCheckLogin]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vCheckLogin]
@Username nvarchar(50),
@Password nvarchar(150)
AS
        Declare @StoredPassword nvarchar(150)
        set @StoredPassword = (select [Password] from HumanSet where Username=@Username)
        IF @StoredPassword = @Password
                Return 1
        Else
                RETURN 0

GO
/****** Object:  StoredProcedure [dbo].[vHuman_GetIdFromUsername]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vHuman_GetIdFromUsername]
        @Username nvarchar(50)
AS
        select Id from HumanSet where Username=@Username

GO
/****** Object:  StoredProcedure [dbo].[vHuman_GetType]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vHuman_GetType]
        @Id     uniqueidentifier
AS
        Declare @adminCount int
        Declare @ownerCount int
        Declare @userCount int
        set @adminCount = (select count(*) as 'Admin' from AdminSet where Id=@Id)
        set @ownerCount = (select count(*) as 'Owner' from OwnerSet where Id=@Id)
        set @userCount = (select count(*) as 'User' from UserSet where Id=@Id)

        if @userCount > 0
                select 'user' as 'UserType'

        else if @ownerCount > 0
                select 'owner' as 'UserType'

        else if @adminCount > 0
                select 'admin' as 'UserType'



GO
/****** Object:  StoredProcedure [dbo].[vProductAvailabiliity]    Script Date: 12/27/2015 12:15:28 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[vProductAvailabiliity]
        @StoreId uniqueidentifier
AS
BEGIN
        select B.ProductId, ProductName, Available = case [Count]

        when '0' then 'False'
        when '1' then 'True'
        else 'error'

        end

        from
        (
                select A.ProductId, Count(*)-1 as [Count] from
                (
                        select ProductId, NULL as StoreId from vProduct
                        union
                        select ProductId, StoreId from StoreProductSet
                        where StoreId=@StoreId
                ) as A
                group by
                A.ProductId
        ) as B inner join vProduct on vProduct.ProductId = B.ProductId
END


GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
Begin PaneConfigurations =
   Begin PaneConfiguration = 0
      NumPanes = 4
      Configuration = "(H (1[40] 4[20] 2[20] 3) )"
   End
   Begin PaneConfiguration = 1
      NumPanes = 3
      Configuration = "(H (1 [50] 4 [25] 3))"
   End
   Begin PaneConfiguration = 2
      NumPanes = 3
      Configuration = "(H (1 [50] 2 [25] 3))"
   End
   Begin PaneConfiguration = 3
      NumPanes = 3
      Configuration = "(H (4 [30] 2 [40] 3))"
   End
   Begin PaneConfiguration = 4
      NumPanes = 2
      Configuration = "(H (1 [56] 3))"
   End
   Begin PaneConfiguration = 5
      NumPanes = 2
      Configuration = "(H (2 [66] 3))"
   End
   Begin PaneConfiguration = 6
      NumPanes = 2
      Configuration = "(H (4 [50] 3))"
   End
   Begin PaneConfiguration = 7
      NumPanes = 1
      Configuration = "(V (3))"
   End
   Begin PaneConfiguration = 8
      NumPanes = 3
      Configuration = "(H (1[56] 4[18] 2) )"
   End
   Begin PaneConfiguration = 9
      NumPanes = 2
      Configuration = "(H (1 [75] 4))"
   End
   Begin PaneConfiguration = 10
      NumPanes = 2
      Configuration = "(H (1[66] 2) )"
   End
   Begin PaneConfiguration = 11
      NumPanes = 2
      Configuration = "(H (4 [60] 2))"
   End
   Begin PaneConfiguration = 12
      NumPanes = 1
      Configuration = "(H (1) )"
   End
   Begin PaneConfiguration = 13
      NumPanes = 1
      Configuration = "(V (4))"
   End
   Begin PaneConfiguration = 14
      NumPanes = 1
      Configuration = "(V (2))"
   End
   ActivePaneConfig = 0
End
Begin DiagramPane =
   Begin Origin =
      Top = 0
      Left = 0
   End
   Begin Tables =
      Begin Table = "AdminSet"
         Begin Extent =
            Top = 6
            Left = 38
            Bottom = 85
            Right = 208
         End
         DisplayFlags = 280
         TopColumn = 0
      End
      Begin Table = "HumanSet"
         Begin Extent =
            Top = 6
            Left = 246
            Bottom = 136
            Right = 416
         End
         DisplayFlags = 280
         TopColumn = 2
      End
   End
End
Begin SQLPane =
End
Begin DataPane =
   Begin ParameterDefaults = ""
   End
End
Begin CriteriaPane =
   Begin ColumnWidths = 11
      Column = 1440
      Alias = 900
      Table = 1170
      Output = 720
      Append = 1400
```

```
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vAdmin'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vAdmin'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "CitySet"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 215
               Right = 208
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
   End
   Begin CriteriaPane = 
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vCity'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vCity'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties = 
   Begin PaneConfigurations = 
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "FeedbackSet"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 136
               Right = 208
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane = 
   End
   Begin DataPane = 
      Begin ParameterDefaults = ""
      End
   End
   Begin CriteriaPane = 
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vFeedBack'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vFeedBack'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties = 
   Begin PaneConfigurations = 
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[42] 4[20] 2[18] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "OwnerSet"
```

```
         Begin Extent = 
            Top = 6
            Left = 38
            Bottom = 136
            Right = 208
         End
         DisplayFlags = 280
         TopColumn = 0
      End
      Begin Table = "HumanSet"
         Begin Extent = 
            Top = 6
            Left = 246
            Bottom = 136
            Right = 416
         End
         DisplayFlags = 280
         TopColumn = 0
      End
   End
End
Begin SQLPane = 
End
Begin DataPane = 
   Begin ParameterDefaults = ""
   End
   Begin ColumnWidths = 9
      Width = 284
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
      Width = 1500
   End
End
Begin CriteriaPane = 
   Begin ColumnWidths = 11
      Column = 1440
      Alias = 900
      Table = 1170
      Output = 720
      Append = 1400
      NewValue = 1170
      SortType = 1350
      SortOrder = 1410
      GroupBy = 1350
      Filter = 1350
      Or = 1350
      Or = 1350
```

```
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vOwner'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vOwner'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "AdminOwnerSet"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 136
               Right = 208
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "vOwner"
            Begin Extent =
               Top = 6
               Left = 246
               Bottom = 202
               Right = 416
            End
            DisplayFlags = 280
            TopColumn = 0
         End
      End
   End
   Begin SQLPane =
   End
```

```
   Begin DataPane = 
      Begin ParameterDefaults = ""
      End
      Begin ColumnWidths = 9
         Width = 284
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
         Width = 1500
      End
   End
   Begin CriteriaPane = 
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vOwnerStatus'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo', 
@level1type=N'VIEW',@level1name=N'vOwnerStatus'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties = 
   Begin PaneConfigurations = 
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
         Begin Origin =
            Top = 0
            Left = 0
         End
         Begin Tables =
            Begin Table = "ProductSet"
               Begin Extent =
                  Top = 6
                  Left = 38
                  Bottom = 136
                  Right = 208
               End
               DisplayFlags = 280
               TopColumn = 0
            End
            Begin Table = "vGroup"
               Begin Extent =
                  Top = 6
                  Left = 246
                  Bottom = 102
                  Right = 416
               End
               DisplayFlags = 280
               TopColumn = 0
            End
         End
      End
      Begin SQLPane =
      End
      Begin DataPane =
         Begin ParameterDefaults = ""
         End
      End
      Begin CriteriaPane =
         Begin ColumnWidths = 11
            Column = 1440
            Alias = 900
            Table = 1170
            Output = 720
            Append = 1400
            NewValue = 1170
            SortType = 1350
            SortOrder = 1410
            GroupBy = 1350
            Filter = 1350
            Or = 1350
            Or = 1350
            Or = 1350
         End
      End
   End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vProduct'
```

```
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vProduct'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPane1', @value=N'[0E232FF0-B466-11cf-A24F-00AA00A3EFFF, 1.00]
Begin DesignProperties =
   Begin PaneConfigurations =
      Begin PaneConfiguration = 0
         NumPanes = 4
         Configuration = "(H (1[40] 4[20] 2[20] 3) )"
      End
      Begin PaneConfiguration = 1
         NumPanes = 3
         Configuration = "(H (1 [50] 4 [25] 3))"
      End
      Begin PaneConfiguration = 2
         NumPanes = 3
         Configuration = "(H (1 [50] 2 [25] 3))"
      End
      Begin PaneConfiguration = 3
         NumPanes = 3
         Configuration = "(H (4 [30] 2 [40] 3))"
      End
      Begin PaneConfiguration = 4
         NumPanes = 2
         Configuration = "(H (1 [56] 3))"
      End
      Begin PaneConfiguration = 5
         NumPanes = 2
         Configuration = "(H (2 [66] 3))"
      End
      Begin PaneConfiguration = 6
         NumPanes = 2
         Configuration = "(H (4 [50] 3))"
      End
      Begin PaneConfiguration = 7
         NumPanes = 1
         Configuration = "(V (3))"
      End
      Begin PaneConfiguration = 8
         NumPanes = 3
         Configuration = "(H (1[56] 4[18] 2) )"
      End
      Begin PaneConfiguration = 9
         NumPanes = 2
         Configuration = "(H (1 [75] 4))"
      End
      Begin PaneConfiguration = 10
         NumPanes = 2
         Configuration = "(H (1[66] 2) )"
      End
      Begin PaneConfiguration = 11
         NumPanes = 2
         Configuration = "(H (4 [60] 2))"
      End
      Begin PaneConfiguration = 12
         NumPanes = 1
         Configuration = "(H (1) )"
      End
      Begin PaneConfiguration = 13
         NumPanes = 1
         Configuration = "(V (4))"
      End
      Begin PaneConfiguration = 14
         NumPanes = 1
         Configuration = "(V (2))"
      End
      ActivePaneConfig = 0
   End
   Begin DiagramPane =
      Begin Origin =
         Top = 0
         Left = 0
      End
      Begin Tables =
         Begin Table = "UserSet"
            Begin Extent =
               Top = 6
               Left = 38
               Bottom = 119
               Right = 208
            End
            DisplayFlags = 280
            TopColumn = 0
         End
         Begin Table = "HumanSet"
            Begin Extent =
               Top = 6
               Left = 246
               Bottom = 136
               Right = 416
            End
            DisplayFlags = 280
            TopColumn = 2
         End
      End
   End
   Begin SQLPane =
   End
   Begin DataPane =
      Begin ParameterDefaults = ""
      End
   End
   Begin CriteriaPane =
      Begin ColumnWidths = 11
         Column = 1440
         Alias = 900
         Table = 1170
         Output = 720
         Append = 1400
         NewValue = 1170
         SortType = 1350
         SortOrder = 1410
         GroupBy = 1350
         Filter = 1350
         Or = 1350
         Or = 1350
         Or = 1350
      End
   End
End
' , @level0type=N'SCHEMA',@level0name=N'dbo', @level1type=N'VIEW',@level1name=N'vUser'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_DiagramPaneCount', @value=1 , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'VIEW',@level1name=N'vUser'
GO
USE [master]
GO
ALTER DATABASE [CheckehDB] SET  READ_WRITE
GO
```

==============================
Filename : windows.txt

____
Filename : About.cs

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

namespace Chekeh
{
    class About
    {
        static public string Server = "http://192.168.29.1/api/";

        static public string Username;
        static public string Status;
        static public string Identifier;
        static public string Password;
    }
}
```

```
______________________________
Filename : AdminChat.xaml

<Window x:Class="Chekeh.AdminChat"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        Title="گفت و گو" Height="300" Width="300" FlowDirection="RightToLeft" Loaded="Window_Loaded" ShowInTaskbar="False" Icon="favicon.ico"
WindowStartupLocation="CenterScreen">
    <Grid>
        <local:ChatUserControl x:Name="thisChat" HorizontalAlignment="Stretch" Margin="10,10,10,10" VerticalAlignment="Stretch"/>

    </Grid>
</Window>
```

```
______________________________
Filename : AdminChat.xaml.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for AdminChat.xaml
    /// </summary>
    public partial class AdminChat : Window
    {
        public AdminChat()
        {
            InitializeComponent();
        }
        public Guid ReciverId { get; set; }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            thisChat.ReciverId = ReciverId;
            thisChat.SenderId = new Guid(About.Identifier);
        }
    }
}
```

____
Filename : AdminMain.xaml

```
<Window x:Name="AdminMainWindow" x:Class="Chekeh.AdminMain"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        Title="اپراتور های چکه!" Height="251.292" Width="640.959" FlowDirection="RightToLeft" WindowStartupLocation="CenterScreen"
Loaded="AdminMainWindow_Loaded" Icon="favicon.ico">
    <Grid VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
        <TabControl VerticalAlignment="Stretch" HorizontalAlignment="Stretch" x:Name="tabControl">
            <TabItem Header="تائید نشده ها" Loaded="TabItem_Loaded" >
                <ScrollViewer>
                    <Grid>
                        <StackPanel x:Name="status_notconfirmed_list">

                        </StackPanel>
                    </Grid>
                </ScrollViewer>
            </TabItem>
            <TabItem Header="تائید شده ها" x:Name="Status_Confirmed" Loaded="Status_Confirmed_Loaded" >
                <ScrollViewer>
                    <Grid>
                        <StackPanel x:Name="status_confirmed_list">

                        </StackPanel>
                    </Grid>
                </ScrollViewer>
            </TabItem>
            <TabItem Header="گروه ها" x:Name="Groups" GotFocus="Groups_GotFocus">
                <Grid>
                    <Label x:Name="label_group" Content="عنوان گروه:" HorizontalAlignment="Left" Margin="10,9,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textbox_Group" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Margin="81,10,0,0" Width="369"/>
                    <Button x:Name="button_GroupCreate" Content="ایجاد" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23"
```

```
Click="button_GroupCreate_Click" Margin="455,12,0,0" />
                    <ScrollViewer HorizontalAlignment="Stretch" Margin="10,38,10,10" VerticalAlignment="Stretch">
                        <Grid>
                            <StackPanel x:Name="GroupList">

                            </StackPanel>
                        </Grid>
                    </ScrollViewer>
                </Grid>
            </TabItem>
            <TabItem Header="محصولات" x:Name="Products" GotFocus="Products_GotFocus">
                <Grid>
                    <Label x:Name="label2" Content="توضیحات" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label1" Content="عنوان" HorizontalAlignment="Left" Margin="10,41,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textbox_Product" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Margin="70,10,0,0" Width="342"/>
                    <ComboBox x:Name="comboBox_ProductCombo" HorizontalAlignment="Left" Margin="417,10,0,0" VerticalAlignment="Top" Width="120" />
                    <TextBox x:Name="textbox_Description" Height="46" Margin="70,41,0,0" TextWrapping="Wrap" HorizontalAlignment="Left"
VerticalAlignment="Top" AcceptsReturn="True" Text="" Width="547"/>
                    <Button x:Name="button_CreateProduct" Content="ایجاد" HorizontalAlignment="Left" Margin="542,10,0,0" VerticalAlignment="Top"
Width="75" Height="23" Click="button_CreateProduct_Click"/>
                    <ScrollViewer HorizontalAlignment="Stretch" Margin="10,92,10,10" VerticalAlignment="Stretch">
                        <Grid>
                            <StackPanel x:Name="ProductList">

                            </StackPanel>
                        </Grid>
                    </ScrollViewer>
                </Grid>
            </TabItem>
            <TabItem Header="شهر ها" x:Name="Cities" GotFocus="Cities_GotFocus">
                <Grid>
                    <Label x:Name="label_City" Content="نام شهر:" HorizontalAlignment="Left" Margin="10,9,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textbox_City" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Margin="81,10,0,0" Width="369"/>
                    <Button x:Name="button_CityCreate" Content="ایجاد" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23"
Click="button_CityCreate_Click" Margin="455,12,0,0" />
                    <ScrollViewer HorizontalAlignment="Stretch" Margin="10,38,10,10" VerticalAlignment="Stretch">
                        <Grid>
                            <StackPanel x:Name="CityList">

                            </StackPanel>
                        </Grid>
                    </ScrollViewer>
                </Grid>
            </TabItem>
        </TabControl>
    </Grid>
</Window>
```

Filename : AdminMain.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for AdminMain.xaml
    /// </summary>
    public partial class AdminMain : Window
    {
        public AdminMain()
        {
            InitializeComponent();
        }

        static int GroupGotFocus = 0;
        static public int CityGotFocus = 0;
        static int ProductGotFocus = 0;

        private void AdminMainWindow_Loaded(object sender, RoutedEventArgs e)
        {

        }

        private void TabItem_Loaded(object sender, RoutedEventArgs e)
        {
            status_notconfirmed_list.Children.Clear();

            try
            {
                using (WebClient client_notconfirmed = new WebClient())
                {
                    client_notconfirmed.Encoding = Encoding.UTF8;

                    string Data_notconfirmed = client_notconfirmed.DownloadString(About.Server + "OwnerStatus");
                    Data_notconfirmed = Data_notconfirmed.Remove(Data_notconfirmed.Length - 2);
                    Data_notconfirmed = Data_notconfirmed.Substring(2, Data_notconfirmed.Length - 2);

                    foreach (var row in Data_notconfirmed.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {

                        string Username = String.Empty;
                        Guid Identifier = new Guid();

                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "Username")
                                Username = values[1];
                            else if (values[0] == "Id")
                                Identifier = new Guid(values[1]);
                        }
                        status_notconfirmed_list.Children.Add(new Status { Username = Username, OwnerId = Identifier });
                    }

                }
            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات از سرور");
            }
        }

        private void Status_Confirmed_Loaded(object sender, RoutedEventArgs e)
        {
            status_confirmed_list.Children.Clear();
            try
            {
                using (WebClient client_confirmed = new WebClient())
                {
                    client_confirmed.Encoding = Encoding.UTF8;

                    string Data_confirmed = client_confirmed.DownloadString(About.Server + "OwnerStatus/Confirmed");
                    Data_confirmed = Data_confirmed.Remove(Data_confirmed.Length - 2);
                    Data_confirmed = Data_confirmed.Substring(2, Data_confirmed.Length - 2);

                    foreach (var row in Data_confirmed.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {

                        string Username = String.Empty;
                        Guid Identifier = new Guid();
                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "Username")
                                Username = values[1];
                            else if (values[0] == "Id")
                                Identifier = new Guid(values[1]);
                        }
                        status_confirmed_list.Children.Add(new Status { Username = Username, OwnerId = Identifier });
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات از سرور");
            }
        }

        private void button_GroupCreate_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                using (WebClient client = new WebClient())
                {
                    client.Encoding = Encoding.UTF8;
                    NameValueCollection nvc = new NameValueCollection();
                    nvc["Name"] = textbox_Group.Text;
                    string response = Encoding.UTF8.GetString(client.UploadValues(About.Server + "Groups", nvc));
                    MessageBox.Show("عملیات با موفقیت انجام شد");
                   textbox_Group.Text = String.Empty;
                    GetGroups();
                }
            }
            catch { }
        }
        public void GetProducts()
        {
            try
            {
                using (WebClient client = new WebClient())
                {

                    client.Encoding = Encoding.UTF8;
                    string data = client.DownloadString(About.Server + "Products");
                    data = data.Remove(data.Length - 2);
                    data = data.Substring(2, data.Length - 2);

                    ProductList.Children.Clear();

                    foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {
                        ProductUserControl productUserControl = new ProductUserControl();

                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "ProductName")
                            {
                                productUserControl.ProductName = values[1];
                            }
                            else if (values[0] == "ProductId")
                            {
                                productUserControl.Id = new Guid(values[1]);
                            }

                        }
                        ProductList.Children.Add(productUserControl);
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات از سرور");

            }
        }
        public void GetGroups()
        {
            try
            {
                using (WebClient client = new WebClient())
                {

                    client.Encoding = Encoding.UTF8;

                    string data = client.DownloadString(About.Server + "Groups");
                    data = data.Remove(data.Length - 2);
                    data = data.Substring(2, data.Length - 2);
                    GroupList.Children.Clear();
                    comboBox_ProductCombo.Items.Clear();

                    foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {
                        GroupUserControl groupUserControl = new GroupUserControl();
                        ComboBoxItem comboItem = new ComboBoxItem();

                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "Name")
                            {
                                groupUserControl.GroupName = values[1];
                                comboItem.Content = values[1];
                            }
                            else if (values[0] == "Id")
                            {
                                groupUserControl.Id = new Guid(values[1]);
                                comboItem.Tag = values[1];
                            }

                        }
                        comboBox_ProductCombo.Items.Add(comboItem);
                        GroupList.Children.Add(groupUserControl);
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات از سرور");

            }
        }

        private void Groups_GotFocus(object sender, RoutedEventArgs e)
        {
            if (GroupGotFocus < 1)
            {
                GetGroups();
                GroupGotFocus++;
            }
        }
```

```
private void button_CreateProduct_Click(object sender, RoutedEventArgs e)
{
    try
    {
        using (WebClient client = new WebClient())
        {
            client.Encoding = Encoding.UTF8;

            NameValueCollection nvc = new NameValueCollection();
            nvc["Name"] = textbox_Product.Text;
            nvc["Description"] = textbox_Description.Text;
            nvc["GroupId"] = ((ComboBoxItem)comboBox_ProductCombo.SelectedItem).Tag.ToString();
            client.UploadValues(About.Server + "Products", nvc);
            MessageBox.Show("عملیات با موفقیت انجام شد");
            textbox_Product.Text = textbox_Description.Text = String.Empty;
            GetGroups();
            GetProducts();
        }
    }
    catch
    {
        MessageBox.Show("ارتباط با سرور برقرار نیست");
    }
}

private void Products_GotFocus(object sender, RoutedEventArgs e)
{
    if (GroupGotFocus < 1)
    {
        GetGroups();
        GroupGotFocus++;
    }
    if (ProductGotFocus < 1)
    {
        GetProducts();
        ProductGotFocus++;
    }
}

public void GetCities()
{
    try
    {
        using (WebClient client = new WebClient())
        {
            client.Encoding = Encoding.UTF8;
            string data = client.DownloadString(About.Server + "Cities");
            data = data.Remove(data.Length - 2);
            data = data.Substring(2, data.Length - 2);

            CityList.Children.Clear();
            foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
            {
                CityUserControl cityUserControl = new CityUserControl();

                foreach (var column in row.Split(','))
                {
                    string[] values = column.Split(':');

                    values[0] = values[0].Remove(values[0].Length - 1);
                    values[0] = values[0].Substring(1, values[0].Length - 1);

                    values[1] = values[1].Remove(values[1].Length - 1);
                    values[1] = values[1].Substring(1, values[1].Length - 1);

                    if (values[0] == "Name")
                    {
                        cityUserControl.CityName = values[1];
                    }
                    else if (values[0] == "Id")
                    {
                        cityUserControl.Id = new Guid(values[1]);
                    }

                }
                CityList.Children.Add(cityUserControl);
            }
        }
    }
    catch
    {
        MessageBox.Show("ارتباط با سرور برقرار نیست");
    }
}

private void button_CityCreate_Click(object sender, RoutedEventArgs e)
{
    try
    {
        WebRequest request = WebRequest.Create(About.Server + "Cities");
        request.Method = "POST";
        string postData = "Name=" + textbox_City.Text;
        byte[] byteArray = Encoding.UTF8.GetBytes(postData);
        request.ContentType = "application/x-www-form-urlencoded";
        request.ContentLength = byteArray.Length;
        Stream dataStream = request.GetRequestStream();
        dataStream.Write(byteArray, 0, byteArray.Length);
        dataStream.Close();
        WebResponse response = request.GetResponse();
        // Console.WriteLine(((HttpWebResponse)response).StatusDescription);
        dataStream = response.GetResponseStream();
        StreamReader reader = new StreamReader(dataStream);
        string responseFromServer = reader.ReadToEnd();
```

```
                // Console.WriteLine(responseFromServer);
                reader.Close();
                dataStream.Close();
                response.Close();
                MessageBox.Show("عملیات با موفقیت انجام شد");
                textbox_City.Text= String.Empty;
                GetCities();
            }
            catch
            {
                MessageBox.Show("پاسخ مناسب از سرور دریافت نشد");
            }
        }

        private void Cities_GotFocus(object sender, RoutedEventArgs e)
        {
            if (CityGotFocus < 1)
            {
                GetCities();
                CityGotFocus++;
            }
        }
    }
}
```

____

Filename : App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6" />
    </startup>
</configuration>
```

____

Filename : App.xaml

```
<Application x:Class="Chekeh.App"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:local="clr-namespace:Chekeh"
             StartupUri="Login.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="Resources\MyStyle\Theme.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
```

```
</Application>
```

Filename : App.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

Filename : ChatUserControl.xaml

```
<UserControl x:Class="Chekeh.ChatUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d"
             d:DesignHeight="300" d:DesignWidth="300" Loaded="UserControl_Loaded">
    <Grid VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
        <ScrollViewer Margin="0,0,0,23" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
            <StackPanel x:Name="stackpanel_Main" HorizontalAlignment="Stretch" VerticalAlignment="Stretch">


            </StackPanel>
        </ScrollViewer>
        <TextBox x:Name="textbox_Value" Margin="39,0,69,0" Height="23" HorizontalAlignment="Stretch" VerticalAlignment="Bottom" TextWrapping="Wrap"
Text="" Background="{x:Null}" />
        <Button x:Name="button_Send" Content="ارسال" HorizontalAlignment="Right"
                VerticalAlignment="Bottom" Width="60" Click="button_Send_Click">
            <Button.Background>
                <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                    <GradientStop Color="#00000000" Offset="0"/>
                    <GradientStop Color="White" Offset="1"/>
                </LinearGradientBrush>
            </Button.Background>
        </Button>
        <Label x:Name="label_Comment" Content="پیام: " HorizontalAlignment="Left" VerticalAlignment="Bottom"/>

    </Grid>
</UserControl>
```

```
______________________________
Filename : ChatUserControl.xaml.cs

using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Chat.xaml
    /// </summary>
    public partial class ChatUserControl : UserControl
    {
        public ChatUserControl()
        {
            InitializeComponent();
        }

        public Guid ReciverId { get; set; }
        public Guid SenderId { get; set; }

        public void refresh()
        {
            stackpanel_Main.Children.Clear();
            using (WebClient client = new WebClient())
            {
                client.Encoding = Encoding.UTF8;
                try
                {
                    string Content = String.Empty;

                        Content = client.DownloadString(About.Server + "FeedBack/" + SenderId);

                    Content = Content.Remove(Content.Length - 2);
                    Content = Content.Substring(2, Content.Length - 2);

                    foreach (var row in Content.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {
                        string text = String.Empty;
                        bool IsSender = false;
                        foreach (var column in row.Split(','))
                        {

                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            try
                            {
                                Convert.ToInt32(values[1]);
                            }
                            catch
                            {
                                values[1] = values[1].Remove(values[1].Length - 1);
                                values[1] = values[1].Substring(1, values[1].Length - 1);
                            }
                            if (values[0] == "SenderId")
                            {
                                if (values[1] == SenderId.ToString())
                                {
                                    IsSender = true;
                                }
                            }
                            else if (values[0] == "Value")
                            {
                                text += values[1];
                            }
                        }
                        stackpanel_Main.Children.Add(new Dialogue { Content = text, IsSender = IsSender });
                    }
                }
                catch
                {
                    stackpanel_Main.Children.Add(new Label { Content = "هیچ پیامی دریافت نشد" });
                }
            }
        }
```

```
        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            refresh();
        }

        private void button_Send_Click(object sender, RoutedEventArgs e)
        {
            using (WebClient client = new WebClient())
            {
                client.Encoding = Encoding.UTF8;

                try
                {
                    NameValueCollection nvc = new NameValueCollection();
                    nvc["Value"] = textbox_Value.Text;

                        nvc["ReciverId"] = ReciverId.ToString();

                    string response = Encoding.UTF8.GetString(client.UploadValues(About.Server + "FeedBack/" + SenderId.ToString(), nvc));
                    refresh();
                    textbox_Value.Text = String.Empty;
                }
                catch {
                    MessageBox.Show("ارتباط با سرور برقرار نیست.");
                }
            }
        }
    }
}
```

Filename : Chekeh.csproj

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.props" Condition="Exists('$(MSBuildExtensionsPath)\$
(MSBuildToolsVersion)\Microsoft.Common.props')" />
  <PropertyGroup>
    <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
    <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
    <ProjectGuid>{DE4003B0-87B2-43F6-B659-493C9FE0B0AC}</ProjectGuid>
    <OutputType>WinExe</OutputType>
    <AppDesignerFolder>Properties</AppDesignerFolder>
    <RootNamespace>Chekeh</RootNamespace>
    <AssemblyName>Chekeh</AssemblyName>
    <TargetFrameworkVersion>v4.6</TargetFrameworkVersion>
    <FileAlignment>512</FileAlignment>
    <ProjectTypeGuids>{60dc8134-eba5-43b8-bcc9-bb4bc16c2548};{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}</ProjectTypeGuids>
    <WarningLevel>4</WarningLevel>
    <AutoGenerateBindingRedirects>true</AutoGenerateBindingRedirects>
```

```
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugSymbols>true</DebugSymbols>
  <DebugType>full</DebugType>
  <Optimize>false</Optimize>
  <OutputPath>bin\Debug\</OutputPath>
  <DefineConstants>DEBUG;TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
  <PlatformTarget>AnyCPU</PlatformTarget>
  <DebugType>pdbonly</DebugType>
  <Optimize>true</Optimize>
  <OutputPath>bin\Release\</OutputPath>
  <DefineConstants>TRACE</DefineConstants>
  <ErrorReport>prompt</ErrorReport>
  <WarningLevel>4</WarningLevel>
</PropertyGroup>
<PropertyGroup>
  <ApplicationIcon>favicon.ico</ApplicationIcon>
</PropertyGroup>
<ItemGroup>
  <Reference Include="System" />
  <Reference Include="System.Data" />
  <Reference Include="System.Drawing" />
  <Reference Include="System.Windows.Forms" />
  <Reference Include="System.Xml" />
  <Reference Include="Microsoft.CSharp" />
  <Reference Include="System.Core" />
  <Reference Include="System.Xml.Linq" />
  <Reference Include="System.Data.DataSetExtensions" />
  <Reference Include="System.Net.Http" />
  <Reference Include="System.Xaml">
    <RequiredTargetFramework>4.0</RequiredTargetFramework>
  </Reference>
  <Reference Include="WindowsBase" />
  <Reference Include="PresentationCore" />
  <Reference Include="PresentationFramework" />
</ItemGroup>
<ItemGroup>
  <ApplicationDefinition Include="App.xaml">
    <Generator>MSBuild:Compile</Generator>
    <SubType>Designer</SubType>
  </ApplicationDefinition>
  <Compile Include="AdminChat.xaml.cs">
    <DependentUpon>AdminChat.xaml</DependentUpon>
  </Compile>
  <Compile Include="AdminMain.xaml.cs">
    <DependentUpon>AdminMain.xaml</DependentUpon>
  </Compile>
```

```
<Compile Include="ChatUserControl.xaml.cs">
  <DependentUpon>ChatUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="CityDetails.xaml.cs">
  <DependentUpon>CityDetails.xaml</DependentUpon>
</Compile>
<Compile Include="CityUserControl.xaml.cs">
  <DependentUpon>CityUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="Dialogue.xaml.cs">
  <DependentUpon>Dialogue.xaml</DependentUpon>
</Compile>
<Compile Include="GroupUserControl.xaml.cs">
  <DependentUpon>GroupUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="PossessingUserControl.xaml.cs">
  <DependentUpon>PossessingUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="ProductDetails.xaml.cs">
  <DependentUpon>ProductDetails.xaml</DependentUpon>
</Compile>
<Compile Include="ProductUserControl.xaml.cs">
  <DependentUpon>ProductUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="Register.xaml.cs">
  <DependentUpon>Register.xaml</DependentUpon>
</Compile>
<Compile Include="Status.xaml.cs">
  <DependentUpon>Status.xaml</DependentUpon>
</Compile>
<Compile Include="StoreDetails.xaml.cs">
  <DependentUpon>StoreDetails.xaml</DependentUpon>
</Compile>
<Compile Include="StoreUserControl.xaml.cs">
  <DependentUpon>StoreUserControl.xaml</DependentUpon>
</Compile>
<Compile Include="UserDetails.xaml.cs">
  <DependentUpon>UserDetails.xaml</DependentUpon>
</Compile>
<Page Include="AdminChat.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="AdminMain.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="ChatUserControl.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="CityDetails.xaml">
```

```
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="CityUserControl.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Dialogue.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="GroupUserControl.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Login.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Main.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Compile Include="About.cs" />
<Compile Include="App.xaml.cs">
  <DependentUpon>App.xaml</DependentUpon>
  <SubType>Code</SubType>
</Compile>
<Compile Include="Login.xaml.cs">
  <DependentUpon>Login.xaml</DependentUpon>
</Compile>
<Compile Include="Main.xaml.cs">
  <DependentUpon>Main.xaml</DependentUpon>
  <SubType>Code</SubType>
</Compile>
<Page Include="PossessingUserControl.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="ProductDetails.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="ProductUserControl.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Register.xaml">
  <SubType>Designer</SubType>
  <Generator>MSBuild:Compile</Generator>
</Page>
<Page Include="Resources\BubbleCreme\Theme.xaml">
```

```
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\BureauBlack\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\BureauBlue\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\DavesGlossyControls\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\MyStyle\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ExpressionDark\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ExpressionLight\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\RainierOrange\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\RainierPurple\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\RainierRadialBlue\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ShinyBlue\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ShinyDarkGreen\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ShinyDarkPurple\Theme.xaml">
  <Generator>MSBuild:Compile</Generator>
  <SubType>Designer</SubType>
</Page>
<Page Include="Resources\ShinyDarkTeal\Theme.xaml">
```

```
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\ShinyRed\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\TwilightBlue\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\UXMusingsBubblyBlue\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\UXMusingsGreen\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\UXMusingsRed\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\UXMusingsRoughGreen\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\UXMusingsRoughRed\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Resources\WhistlerBlue\Theme.xaml">
      <Generator>MSBuild:Compile</Generator>
      <SubType>Designer</SubType>
    </Page>
    <Page Include="Status.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
    <Page Include="StoreDetails.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
    <Page Include="StoreUserControl.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
    <Page Include="UserDetails.xaml">
      <SubType>Designer</SubType>
      <Generator>MSBuild:Compile</Generator>
    </Page>
  </ItemGroup>
  <ItemGroup>
    <Compile Include="Properties\AssemblyInfo.cs">
      <SubType>Code</SubType>
    </Compile>
    <Compile Include="Properties\Resources.Designer.cs">
      <AutoGen>True</AutoGen>
      <DesignTime>True</DesignTime>
      <DependentUpon>Resources.resx</DependentUpon>
    </Compile>
    <Compile Include="Properties\Settings.Designer.cs">
      <AutoGen>True</AutoGen>
      <DependentUpon>Settings.settings</DependentUpon>
      <DesignTimeSharedInput>True</DesignTimeSharedInput>
    </Compile>
    <EmbeddedResource Include="Properties\Resources.resx">
      <Generator>ResXFileCodeGenerator</Generator>
      <LastGenOutput>Resources.Designer.cs</LastGenOutput>
    </EmbeddedResource>
    <None Include="Properties\Settings.settings">
      <Generator>SettingsSingleFileGenerator</Generator>
      <LastGenOutput>Settings.Designer.cs</LastGenOutput>
    </None>
    <AppDesigner Include="Properties\" />
  </ItemGroup>
  <ItemGroup>
    <None Include="App.config">
      <SubType>Designer</SubType>
    </None>
  </ItemGroup>
  <ItemGroup>
    <Resource Include="favicon.ico" />
  </ItemGroup>
  <Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```

Filename : Chekeh.csproj.user

```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="14.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    <ProjectView>ShowAllFiles</ProjectView>
  </PropertyGroup>
</Project>
```

```
______________________________
Filename : CityDetails.xaml

<Window x:Class="Chekeh.CityDetails"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
      WindowStartupLocation="CenterScreen"  Title="جزئیات شهر" Height="76.384" Width="400" FlowDirection="RightToLeft" Loaded="Window_Loaded"
Icon="favicon.ico">
    <Grid>
        <Button x:Name="button_Update" Content="به روز رسانی" HorizontalAlignment="Left" Margin="307,10,0,0" VerticalAlignment="Top" Width="75"
Click="button_Update_Click"/>
        <TextBox x:Name="textbox_Name" HorizontalAlignment="Stretch" Height="23" Margin="10,10,90,0"  TextWrapping="Wrap" VerticalAlignment="Top"/>
    </Grid>
</Window>
```

```
______________________________
Filename : CityDetails.xaml.cs

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for CityDetails.xaml
    /// </summary>
    public partial class CityDetails : Window
    {
        public CityDetails()
        {
            InitializeComponent();
        }
        public Guid Id { get; set; }
        public string CityName { get; set; }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            textbox_Name.Text = CityName;
        }

        private void button_Update_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                WebRequest request = WebRequest.Create(About.Server + "Cities/" + Id.ToString());
                request.Method = "PUT";
                string postData = "Name=" + textbox_Name.Text;
                byte[] byteArray = Encoding.UTF8.GetBytes(postData);
                request.ContentType = "application/x-www-form-urlencoded";
                request.ContentLength = byteArray.Length;
                Stream dataStream = request.GetRequestStream();
                dataStream.Write(byteArray, 0, byteArray.Length);
                dataStream.Close();
                WebResponse response = request.GetResponse();
                // Console.WriteLine(((HttpWebResponse)response).StatusDescription);
                dataStream = response.GetResponseStream();
                StreamReader reader = new StreamReader(dataStream);
                string responseFromServer = reader.ReadToEnd();
                // Console.WriteLine(responseFromServer);
                reader.Close();
                dataStream.Close();
                response.Close();
                AdminMain.CityGotFocus = 0;
                MessageBox.Show("به روز رسانی انجام شد");
            }
            catch
            {
                MessageBox.Show("پاسخ مناسب از سرور دریافت نشد");
            }
        }
    }
}
```

____

```
Filename : CityUserControl.xaml

<UserControl x:Class="Chekeh.CityUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d"
              Height="33.321" Width="338.705"  Loaded="UserControl_Loaded">
    <Grid>
        <Label x:Name="label_Title" Content="نام شهر بارگزاری نشده است" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="button_Details" Content="جزئیات" HorizontalAlignment="Left" Margin="264,3,0,0" VerticalAlignment="Top" Width="75"
Click="button_Details_Click" Height="20" />
        <Button x:Name="button_Delete" Content="حذف" HorizontalAlignment="Left" Margin="184,3,0,0" VerticalAlignment="Top" Width="75"
Click="button_Delete_Click"/>
    </Grid>
</UserControl>
```

____

Filename : CityUserControl.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for CityUserControl.xaml
    /// </summary>
    public partial class CityUserControl : UserControl
    {
        public CityUserControl()
        {
            InitializeComponent();
        }

        public string CityName { get; set; }
        public Guid Id { get; set; }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            label_Title.Content = CityName;
        }
        private void button_Details_Click(object sender, RoutedEventArgs e)
        {
            new CityDetails { CityName = CityName, Id = Id }.ShowDialog();
        }

        private void button_Delete_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                var request = WebRequest.Create(About.Server + "Cities/" + Id.ToString());
                request.Method = "DELETE";
                var response = (HttpWebResponse)request.GetResponse();
                Visibility = Visibility.Hidden;
            }
            catch {
                MessageBox.Show("خطا در اعلام درخواست به سرور");
            }
        }
    }
}
```

____

Filename : Dialogue.xaml

```
<UserControl x:Class="Chekeh.Dialogue"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d" Height="auto" Width="auto" Loaded="UserControl_Loaded">

    <Label x:Name="label_Text" Content="متن بارگذاری نشده است" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="5.94,3.168,0,0"/>

</UserControl>
```

____

Filename : Dialogue.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Dialogue.xaml
    /// </summary>
    public partial class Dialogue : UserControl
    {
        public Dialogue()
        {
            InitializeComponent();
        }
        public bool IsSender { get; set; }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            if (!IsSender)
                this.Content = "پاسخ: " + this.Content;
            else
                this.Content = "شما: " + this.Content;
            label_Text.Content = this.Content;
        }
    }
}
```

____
Filename : GroupUserControl.xaml

```
<UserControl x:Class="Chekeh.GroupUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d"
             Height="30" Width="auto" Loaded="UserControl_Loaded">
    <Grid>
        <Label x:Name="label_Title" Content="عنوان گروه بارگزاری نشده است" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="button_Delete" Content="حذف" HorizontalAlignment="Left" Margin="184,0,0,6" VerticalAlignment="Bottom" Width="75"
Click="button_Delete_Click"/>
    </Grid>
</UserControl>
```

____

Filename : GroupUserControl.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for GroupUserControl.xaml
    /// </summary>
    public partial class GroupUserControl : UserControl
    {
        public string GroupName { get; set; }
        public Guid Id { get; set; }

        public GroupUserControl()
        {
            InitializeComponent();
        }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            label_Title.Content = GroupName;
        }

        private void button_Delete_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                var request = WebRequest.Create(About.Server + "Groups/" + Id.ToString());
                request.Method = "DELETE";
                var response = (HttpWebResponse)request.GetResponse();
                Visibility = Visibility.Hidden;
            }
            catch {
                MessageBox.Show("خطا در اعلام درخواست به سرور");
            }
        }
    }
}



______________________________
Filename : Login.xaml

<Window x:Class="Chekeh.Login"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        Title="ورود به سیستم" Height="187.085" Width="300" WindowStartupLocation="CenterScreen" ResizeMode="NoResize" ShowInTaskbar="False"
Icon="favicon.ico">
    <Grid>
        <Label x:Name="label_Username" Content="نام کاربری" HorizontalAlignment="Right" VerticalAlignment="Top" Margin="0,20,46,0"/>
        <Label x:Name="label_Password" Content="کلمه عبور" HorizontalAlignment="Right" VerticalAlignment="Top" Margin="0,48,46,0"/>
        <TextBox x:Name="textbox_Username" HorizontalAlignment="Left" Height="23" TextAlignment="Center" TextWrapping="Wrap" Text=""
VerticalAlignment="Top" Width="120" Margin="54,24,0,0"/>
        <PasswordBox x:Name="textbox_Password" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" Margin="54,52,0,0"/>
        <Button x:Name="button_Login" Content="ورود" HorizontalAlignment="Left" VerticalAlignment="Top" Width="75" Margin="99,80,0,0"
Click="button_Login_Click"/>
        <Button x:Name="button_Register" Content="ثبت نام" HorizontalAlignment="Left" VerticalAlignment="Top" Width="75" Margin="99,105,0,0"
Click="button_Register_Click"/>
    </Grid>
</Window>



______________________________
Filename : Login.xaml.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
```

```
namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Login.xaml
    /// </summary>
    public partial class Login : Window
    {
        public Login()
        {
            InitializeComponent();
        }

        private void button_Login_Click(object sender, RoutedEventArgs e)
        {
            textbox_Username.IsEnabled = textbox_Password.IsEnabled = false;
            try
            {
                using (WebClient client = new WebClient())
                {
                    client.Encoding = Encoding.UTF8;

                    string userType = client.DownloadString(About.Server + "Login?" + "Username=" + textbox_Username.Text + "&Password=" +
textbox_Password.Password).Replace("\"", null);
                    if (userType == "owner")
                    {
                        About.Username = textbox_Username.Text;
                        About.Status = client.DownloadString(About.Server + "AdminOwner/" + textbox_Username.Text).Replace("\"", null);
                            About.Identifier = client.DownloadString(About.Server + "HumanIdentifier/" + About.Username).Replace("\"", null);

                        Main mw = new Main();
                        mw.Show();
                        this.Close();

                        using (WebClient client2 = new WebClient())
                        {
                            client.Encoding = Encoding.UTF8;
                            About.Identifier = client.DownloadString(About.Server + "HumanIdentifier/" + About.Username).Replace("\"", null);
                        }
                    }
                    else if (userType == "admin")
                    {
                        About.Username = textbox_Username.Text;
                        About.Status = null;
                        About.Password = textbox_Password.Password;
                            About.Identifier = client.DownloadString(About.Server + "HumanIdentifier/" + About.Username).Replace("\"", null);


                        new AdminMain { }.Show();
                        this.Close();
                    }
                    else
                    {
                        MessageBox.Show("نام کاربری یا کلمه عبور معتبر نیست. آیا پیشتر ثبت نام کرده اید؟");
                    }
                }
            }
            catch {
                MessageBox.Show("ارتباط با سرور برقرار نیست");
            }
            textbox_Username.IsEnabled = textbox_Password.IsEnabled = true;
        }

        private void button_Register_Click(object sender, RoutedEventArgs e)
        {
            ((Window)(new Register { })).Show();
            this.Close();
        }
    }
}
```

____

Filename : Main.xaml

```
<Window x:Class="Chekeh.Main"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        Title="چکه!" Height="500" Width="700" WindowStartupLocation="CenterScreen" Closed="Window_Closed" Loaded="Window_Loaded"
FlowDirection="RightToLeft" Icon="favicon.ico">
    <Grid>
        <TabControl x:Name="tabControl" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Margin="10">
            <TabItem x:Name="Feedback" Header="بازخورد">
                <Grid VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
                    <Label x:Name="label_Status" Content="وضعیت" HorizontalAlignment="Stretch" Margin="10,10,10,0" VerticalAlignment="Top"/>
                    <local:ChatUserControl x:Name="thisChat" HorizontalAlignment="Stretch" Margin="10,41,0,0" VerticalAlignment="Stretch"/>
                </Grid>
            </TabItem>
            <TabItem x:Name="Profile" Header="مشخصات" GotFocus="Profile_GotFocus">
                <Grid>
                    <Label x:Name="label_Username" Content="نام کاربری" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,8,0,0"/>
                    <Label x:Name="label_Password" Content="کلمه عبور" HorizontalAlignment="Left" Margin="10,39,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_ConfirmPass" Content="تکرار کلمه عبور" HorizontalAlignment="Left" Margin="10,70,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_Name" Content="نام" HorizontalAlignment="Left" Margin="10,100,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_LastName" Content="نام خانوادگی" HorizontalAlignment="Left" Margin="10,131,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_NationalCode" Content="کد ملی" HorizontalAlignment="Left" Margin="10,162,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_PostalCode1" Content="کد پستی" HorizontalAlignment="Left" Margin="10,193,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_PhoneNumber" Content="تلفن" HorizontalAlignment="Left" Margin="10,224,0,0" VerticalAlignment="Top"/>
                    <Label x:Name="label_Address1" Content="آدرس" HorizontalAlignment="Left" Margin="10,255,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textbox_Username" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="104,13,0,0"/>
```

```
                    <TextBox x:Name="textbox_FirstName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="104,105,0,0"/>
                    <TextBox x:Name="textbox_LastName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="104,136,0,0"/>
                    <TextBox x:Name="textbox_NationalCode" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top"
Width="120" Margin="104,167,0,0"/>
                    <TextBox x:Name="textbox_PostalCode1" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top"
Width="120" Margin="104,198,0,0"/>
                    <TextBox x:Name="textbox_PhoneNumber" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top"
Width="120" Margin="104,229,0,0"/>
                    <TextBox x:Name="textbox_Address1" AcceptsReturn="True" HorizontalAlignment="Left" Height="153" TextWrapping="Wrap"
VerticalAlignment="Top" Width="120" Margin="104,258,0,0"/>
                    <PasswordBox x:Name="textbox_Password" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120"
Margin="104,43,0,0"/>
                    <PasswordBox x:Name="textbox_ConfirmPass" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120"
Margin="104,74,0,0"/>
                    <Image x:Name="Image_Profile" HorizontalAlignment="Left" Height="100" VerticalAlignment="Top" Width="100" Margin="272,88,0,0"
Stretch="UniformToFill"/>
                    <Button x:Name="button_Image" Content="تصویر" HorizontalAlignment="Left" Margin="272,12,0,0" VerticalAlignment="Top" Width="100"
Click="button_Image_Click"/>
                    <Button x:Name="button_Delete" Content="حذف تصویر" HorizontalAlignment="Left" Margin="272,37,0,0" VerticalAlignment="Top"
Width="100" Click="button_Delete_Click"/>
                    <Button x:Name="button_Update" Content="ثبت اطلاعات" HorizontalAlignment="Left" VerticalAlignment="Top" Width="178"
Margin="229,198,0,0" Click="button_Update_Click" />
                </Grid>
            </TabItem>
            <TabItem x:Name="Store" Header="فروشگاه" IsEnabled="False" GotFocus="Store_GotFocus">
                <Grid>
                    <Label x:Name="label_Store" Content="نام:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,10,0,0"/>
                    <TextBox x:Name="textbox_Store" HorizontalAlignment="Left" VerticalAlignment="Top" Width="306" Height="23" Margin="69,13,0,0"
TextWrapping="Wrap"/>
                    <Label x:Name="label_City" Content="شهر:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,41,0,0"/>
                    <ComboBox x:Name="comboBox_City" Height="23" VerticalAlignment="Top" HorizontalAlignment="Left" Width="120" Margin="69,44,0,0"  />
                    <Label x:Name="label_PostalCode2" Content="کد پستی:" HorizontalAlignment="Left" Margin="194,41,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textBox_PostalCode2" HorizontalAlignment="Left" Height="23" Margin="255,44,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="120"/>
                    <Label x:Name="label_Address2" Content="آدرس:" HorizontalAlignment="Left" Margin="396,41,0,0" VerticalAlignment="Top"/>
                    <TextBox x:Name="textbox_Address2" HorizontalAlignment="Left" Height="23" Margin="441,44,0,0" TextWrapping="Wrap"
VerticalAlignment="Top" Width="120"/>
                    <Button x:Name="button_Store" Content="ثبت" HorizontalAlignment="Left" Margin="581,44,0,0" VerticalAlignment="Top" Width="75"
Click="button_Store_Click"/>

                    <ScrollViewer Margin="0,72,0,0">
                        <Grid>
                            <StackPanel  x:Name="storesList"></StackPanel>
                        </Grid>
                    </ScrollViewer>
                </Grid>
            </TabItem>
            <TabItem x:Name="Product" Header="محصول" IsEnabled="False" GotFocus="Product_GotFocus">
                <Grid>
                    <Label Content="انتخاب فروشگاه" Height="auto" HorizontalAlignment="Left" Width="100" Margin="0,10,0,0" />
                    <ComboBox x:Name="comboBox_Store1" HorizontalAlignment="Left" VerticalAlignment="Top" Width="300" Margin="115,10,0,0"
SelectionChanged="comboBox_Store1_SelectionChanged">
                    </ComboBox>
                    <ScrollViewer HorizontalAlignment="Stretch" Height="374" Margin="0,42,0,0" VerticalAlignment="Stretch">
                        <Grid  VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
                            <StackPanel x:Name="ProductsList">

                            </StackPanel>
                        </Grid>
                    </ScrollViewer>
                </Grid>
            </TabItem>
            <TabItem x:Name="Partner" Header="مشتری" IsEnabled="False" GotFocus="Partner_GotFocus">
                <Grid>
                    <Label Content="انتخاب فروشگاه" Height="auto" HorizontalAlignment="Left" Width="100" Margin="0,10,0,0" />
                    <ComboBox x:Name="comboBox_Store2" HorizontalAlignment="Left" VerticalAlignment="Top" Width="300" Margin="115,10,0,0">
                    </ComboBox>
                    <ScrollViewer HorizontalAlignment="Stretch" Height="374" Margin="0,42,0,0" VerticalAlignment="Stretch">
                        <Grid x:Name="CustomersList" VerticalAlignment="Stretch" HorizontalAlignment="Stretch"/>
                    </ScrollViewer>
                </Grid>
            </TabItem>
        </TabControl>
    </Grid>
</Window>
```

Filename : Main.xaml.cs

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
```

```
public partial class Main : Window
{
    public Main()
    {
        InitializeComponent();
    }

    private void Window_Closed(object sender, EventArgs e)
    {
        Application.Current.Shutdown();
    }

    private void Window_Loaded(object sender, RoutedEventArgs e)
    {
        try
        {
            thisChat.SenderId = new Guid(About.Identifier);


            if (About.Status == "NotConfirmed")
            {
                label_Status.Content = "در انتظار تائید مدیر سیستم";
            }
            else if (About.Status == "Confirmed")
            {
                Partner.IsEnabled = false;
                Store.IsEnabled = Product.IsEnabled = true;
                label_Status.Content = "کاربر توسط هوش انسانی تائید شده است؛ از افتخار همکاری با شما خرسندیم";
            }
        }
        catch
        {
            MessageBox.Show("ارتباط با سرور برقرار نیست");
        }
    }
    private void button_Update_Click(object sender, RoutedEventArgs e)
    {
        try
        {
            if (textbox_Password.Password == textbox_ConfirmPass.Password)
            {
                WebRequest request = WebRequest.Create(About.Server + "Owner/" + About.Identifier);
                request.Method = "PUT";
                string postData =
                      "Username=" + textbox_Username.Text +
                      "&FirstName=" + textbox_FirstName.Text +
                      "&LastName=" + textbox_LastName.Text +
                      "&NationalCode=" + textbox_NationalCode.Text +
                      "&Address=" + textbox_Address1.Text +
                      "&PostalCode=" + textbox_PostalCode1.Text +
                      "&PhoneNumber=" + textbox_PhoneNumber.Text +
                      "&Password=" + textbox_Password.Password;
                byte[] byteArray = Encoding.UTF8.GetBytes(postData);
                request.ContentType = "application/x-www-form-urlencoded";
                request.ContentLength = byteArray.Length;
                Stream dataStream = request.GetRequestStream();
                dataStream.Write(byteArray, 0, byteArray.Length);
                dataStream.Close();
                WebResponse response = request.GetResponse();
                // Console.WriteLine(((HttpWebResponse)response).StatusDescription);
                dataStream = response.GetResponseStream();
                StreamReader reader = new StreamReader(dataStream);
                string responseFromServer = reader.ReadToEnd();
                // Console.WriteLine(responseFromServer);
                reader.Close();
                dataStream.Close();
                response.Close();
                MessageBox.Show("به روز رسانی انجام شد");
            }

            else
            {
                MessageBox.Show("کلمه عبور با تکرار آن همخوانی ندارد");
            }
        }
        catch (Exception g)
        {
            MessageBox.Show("پاسخ مناسب از سرور دریافت نشد");
        }
    }

    int CountsGotFocused = 0;
    static public string ImagePath = null;

    private void Profile_GotFocus(object sender, RoutedEventArgs e)
    {
        if (CountsGotFocused < 1) // Lazy load on first focus
        {
            try
            {
                using (WebClient client = new WebClient())
                {
                    client.Encoding = Encoding.UTF8;

                    string data = client.DownloadString(About.Server + "Owner/" + About.Identifier);
                    data = data.Remove(data.Length - 1);
                    data = data.Substring(1, data.Length - 1);

                    foreach (string column in data.Split(new string[] { "," }, StringSplitOptions.None))
                    {
                        string[] item = column.Split(':');

                        item[0] = item[0].Remove(item[0].Length - 1);
                        item[0] = item[0].Substring(1, item[0].Length - 1);
                        item[1] = item[1].Remove(item[1].Length - 1);
                        item[1] = item[1].Substring(1, item[1].Length - 1);

                        if (item[0] == "Username")
                            textbox_Username.Text = item[1];
                        else if (item[0] == "FirstName")
                            textbox_FirstName.Text = item[1];
                        else if (item[0] == "LastName")
                            textbox_LastName.Text = item[1];
                        else if (item[0] == "NationalCode")
                            textbox_NationalCode.Text = item[1];
                        else if (item[0] == "Address")
                            textbox_Address1.Text = item[1];
                        else if (item[0] == "PostalCode")
                            textbox_PostalCode1.Text = item[1];
                        else if (item[0] == "PhoneNumber")
                            textbox_PhoneNumber.Text = item[1];

                        Image_Profile.Source = new BitmapImage(new Uri(About.Server + "Human_Image/" + About.Identifier));
                    }
                }
                CountsGotFocused++;
            }
            catch
            {
                MessageBox.Show("اطلاعات دریافت نشده اند و یا نا معتبر تشخیص داده شده اند");
            }
        }
    }

    private void button_Image_Click(object sender, RoutedEventArgs e)
    {

        Stream checkStream = null;
        Microsoft.Win32.OpenFileDialog openFileDialog = new Microsoft.Win32.OpenFileDialog();
        openFileDialog.Multiselect = false;
        openFileDialog.Filter = "تصاویر|*.bmp;*.jpg;*.png";
        if ((bool)openFileDialog.ShowDialog())
        {
            try
            {
                if ((checkStream = openFileDialog.OpenFile()) != null)
                {
                    ImagePath = openFileDialog.FileName;
                    Image_Profile.Source = new BitmapImage(new Uri(openFileDialog.FileName, UriKind.RelativeOrAbsolute));
                }
            }
            catch
            {
                MessageBox.Show("خطا: فایل ارسال شده قابل خواندن نیست");
            }
        }
    }

    private void button_Delete_Click(object sender, RoutedEventArgs e)
    {
        ImagePath = null;
        Image_Profile.Source = null;
    }
    public void GetStores()
    {
        try
        {
            using (WebClient client_City = new WebClient())
            {
                client_City.Encoding = Encoding.UTF8;
                string data = client_City.DownloadString(About.Server + "Cities");
                data = data.Remove(data.Length - 2);
                data = data.Substring(2, data.Length - 2);

                comboBox_City.Items.Clear();

                foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
                {
                    ComboBoxItem cityComboItem = new ComboBoxItem();

                    foreach (var column in row.Split(','))
                    {
                        string[] values = column.Split(':');

                        values[0] = values[0].Remove(values[0].Length - 1);
                        values[0] = values[0].Substring(1, values[0].Length - 1);

                        values[1] = values[1].Remove(values[1].Length - 1);
                        values[1] = values[1].Substring(1, values[1].Length - 1);

                        if (values[0] == "Name")
                        {
                            cityComboItem.Content = values[1];
                        }
                        else if (values[0] == "Id")
                        {
                            cityComboItem.Tag = new Guid(values[1]);
                        }

                    }
                    comboBox_City.Items.Add(cityComboItem);

                }
            }
```

```
using (WebClient client_stores = new WebClient())
{
    client_stores.Encoding = Encoding.UTF8;
    string data = client_stores.DownloadString(About.Server + "Stores/" + About.Identifier);

    data = data.Remove(data.Length - 2);
    data = data.Substring(2, data.Length - 2);

    storesList.Children.Clear();
    comboBox_Store1.Items.Clear();
    comboBox_Store2.Items.Clear();




    foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
    {

        StoreUserControl storeUserControl = new StoreUserControl();
        ComboBoxItem storeComboItem1 = new ComboBoxItem();


        foreach (var column in row.Split(','))
        {
            string[] values = column.Split(':');

            values[0] = values[0].Remove(values[0].Length - 1);
            values[0] = values[0].Substring(1, values[0].Length - 1);

            values[1] = values[1].Remove(values[1].Length - 1);
            values[1] = values[1].Substring(1, values[1].Length - 1);

            if (values[0] == "StoreName")
            {
                storeComboItem1.Content = values[1];
                storeUserControl.StoreName = values[1];
            }
            else if (values[0] == "StoreId")
            {
                storeUserControl.StoreId = new Guid(values[1]);
                storeComboItem1.Tag = values[1];
            }

        }
        ComboBoxItem storeComboItem2 = new ComboBoxItem { Tag = storeComboItem1.Tag, Content = storeComboItem1.Content };

        comboBox_Store1.Items.Add(storeComboItem1);
        comboBox_Store2.Items.Add(storeComboItem2);
        storesList.Children.Add(storeUserControl);

    }
```

```
                }

            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات");
            }
        }
        public static int StoreGotFocus = 0;
        private void Store_GotFocus(object sender, RoutedEventArgs e)
        {
            if (StoreGotFocus < 1)
            {
                GetStores();
                StoreGotFocus++;
            }
        }

        int ProductsGotFocus = 0;
        private void Product_GotFocus(object sender, RoutedEventArgs e)
        {
            if (StoreGotFocus < 1)
            {
                GetStores();
                StoreGotFocus++;
            }
            if (ProductsGotFocus < 1)
            {

                ProductsGotFocus++;
            }
        }

        private void button_Store_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                WebRequest request = WebRequest.Create(About.Server + "Stores");
                request.Method = "POST";
                string postData = "Name=" + textbox_Store.Text + "&Address=" + textbox_Address2.Text + "&postalCode=" + textBox_PostalCode2.Text +
"&CityId=" + ((ComboBoxItem)comboBox_City.SelectedItem).Tag.ToString() + "&OwnerId=" + About.Identifier;
                byte[] byteArray = Encoding.UTF8.GetBytes(postData);
                request.ContentType = "application/x-www-form-urlencoded";
                request.ContentLength = byteArray.Length;
                Stream dataStream = request.GetRequestStream();
                dataStream.Write(byteArray, 0, byteArray.Length);
                dataStream.Close();
                WebResponse response = request.GetResponse();
                // Console.WriteLine(((HttpWebResponse)response).StatusDescription);
                dataStream = response.GetResponseStream();
                StreamReader reader = new StreamReader(dataStream);
                string responseFromServer = reader.ReadToEnd();
                // Console.WriteLine(responseFromServer);
                reader.Close();
                dataStream.Close();
                response.Close();
                MessageBox.Show("عملیات با موفقیت انجام شد");
                textbox_Address2.Text = textbox_Store.Text = String.Empty;
                GetStores();
            }
            catch
            {
                MessageBox.Show("پاسخ مناسب از سرور دریافت نشد");
            }

        }

        private void Partner_GotFocus(object sender, RoutedEventArgs e)
        {
            if (StoreGotFocus < 1)
            {
                GetStores();
                StoreGotFocus++;
            }
        }

        private void comboBox_Store1_SelectionChanged(object sender, SelectionChangedEventArgs e)
        {
            try
            {
                using (WebClient client = new WebClient())
                {
                    client.Encoding = Encoding.UTF8;

                    string data = client.DownloadString(About.Server + "StoreProduct/" + ((ComboBoxItem)comboBox_Store1.SelectedItem).Tag.ToString());
                    data = data.Remove(data.Length - 2);
                    data = data.Substring(2, data.Length - 2);

                    ProductsList.Children.Clear();



                    foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {

                        PossessingUserControl possessingUserControl = new PossessingUserControl();
                        possessingUserControl.StoreId = new Guid(((ComboBoxItem)comboBox_Store1.SelectedItem).Tag.ToString());


                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
```

```
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "ProductId")
                            {
                                possessingUserControl.ProductId = new Guid(values[1]);
                            }
                            else if (values[0] == "ProductName")
                            {
                                possessingUserControl.ProductName = values[1];
                            }
                            else if (values[0] == "Available")
                            {
                                possessingUserControl.Possess = Convert.ToBoolean(values[1]);
                            }

                        }
                        ProductsList.Children.Add(possessingUserControl);
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در برقراری ارتباط با سرور");
            }
        }
    }
}
```

____
```
Filename : PossessingUserControl.xaml

<UserControl x:Class="Chekeh.PossessingUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d" FlowDirection="RightToLeft" d:DesignWidth="300" Height="auto" Loaded="UserControl_Loaded">
    <Grid Margin="0,0,0,5">
        <Label x:Name="label_Title" Content="عنوان " HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,7,0,0" Width="200"/>
        <Button x:Name="button_Possess" Content="موجود است | موجود نیست" HorizontalAlignment="Left" Margin="215,10,0,0" VerticalAlignment="Top"
Width="75" Click="button_Possess_Click"/>

    </Grid>
</UserControl>
```

Filename : PossessingUserControl.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for PossessingUserControl.xaml
    /// </summary>
    public partial class PossessingUserControl : UserControl
    {
        public Guid ProductId { get; set; }
        public Guid StoreId { get; set; }
        public string ProductName { get; set; }
        public bool Possess { get; set; }


        public PossessingUserControl()
        {
            InitializeComponent();
        }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            if (Possess)
            {
                button_Possess.Content = "موجود است";
                button_Possess.ToolTip = "برای اعلام موجود نبود کالا کلیک کنید";
            }
            else if (!Possess)
            {
                button_Possess.Content = "موجود نیست";
                button_Possess.ToolTip = "برای اعلام موجود بودن کالا کلیک کنید";
            }
            label_Title.Content = ProductName;
        }
        private void button_Possess_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                using (WebClient client = new WebClient())
                {
                    NameValueCollection nvc = new NameValueCollection();
                    nvc["ProductId"] = ProductId.ToString();
                    nvc["StoreId"] = StoreId.ToString();

                    if (Possess)
                    {
                        nvc["Possess"] = "False";
                        client.UploadValues(About.Server + "StoreProduct", nvc);

                        button_Possess.Content = "موجود نیست";
                        button_Possess.ToolTip = "برای اعلام موجود نبود کالا کلیک کنید";
                        Possess = false;
                    }
                    else if (!Possess)
                    {
                        nvc["Possess"] = "True";
                        client.UploadValues(About.Server + "StoreProduct", nvc);

                        button_Possess.Content = "موجود است";
                        button_Possess.ToolTip = "برای اعلام موجود بودن کالا کلیک کنید";
                        Possess = true;
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در برقراری ارتباط با سرور");
            }
        }
    }
}
```

Filename : ProductDetails.xaml

```
<Window x:Class="Chekeh.ProductDetails"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        FlowDirection="RightToLeft"
        WindowStartupLocation="CenterScreen" Title="جزئیات محصول" Height="300" Width="300" Icon="favicon.ico" ShowInTaskbar="False"
```

```
Loaded="Window_Loaded">
    <Grid>
        <Button x:Name="button_Choose" Content="انتخاب تصویر" HorizontalAlignment="Center" Margin="0,10,0,0" VerticalAlignment="Top" Width="138"
Click="button_Choose_Click"/>
        <Button x:Name="button_Submit" Content="ارسال" HorizontalAlignment="Right" Margin="0,239,10,0" VerticalAlignment="Top" Width="75"
Click="button_Submit_Click"/>

        <Label x:Name="label_Name" Content="عنوان:" HorizontalAlignment="Right" Margin="0,44,241,0" VerticalAlignment="Top"/>
        <Label x:Name="label_Description" Content="جزئیات:" HorizontalAlignment="Right" Margin="0,75,236,0" VerticalAlignment="Top"/>
        <Label x:Name="label_Group" Content="گروه:" HorizontalAlignment="Right" Margin="0,202,247,0" VerticalAlignment="Top"/>

        <TextBox x:Name="textbox_Name" HorizontalAlignment="Left" Height="23" Margin="77,47,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"
Width="205"/>
        <TextBox x:Name="textbox_Description" HorizontalAlignment="Left" AcceptsReturn="True" Height="114" Margin="77,79,0,0" TextWrapping="Wrap"
Text="" VerticalAlignment="Top" Width="205"/>

        <ComboBox x:Name="comboBox_Group" HorizontalAlignment="Left" Margin="77,202,0,0" VerticalAlignment="Top" Width="138"/>

    </Grid>
</Window>
```

Filename : ProductDetails.xaml.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Forms;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for ProductDetails.xaml
    /// </summary>
    public partial class ProductDetails : Window
    {
        public Guid ProductId { get; set; }
        public ProductDetails()
        {
            InitializeComponent();
        }


        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            try
            {
                using (WebClient client = new WebClient())
                {

                    client.Encoding = Encoding.UTF8;

                    string data = client.DownloadString(About.Server + "Groups");
                    data = data.Remove(data.Length - 2);
                    data = data.Substring(2, data.Length - 2);


                    comboBox_Group.Items.Clear();

                    foreach (var row in data.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {
                        GroupUserControl groupUserControl = new GroupUserControl();
                        ComboBoxItem comboItem = new ComboBoxItem();

                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "Name")
                            {
                                groupUserControl.GroupName = values[1];
                                comboItem.Content = values[1];
                            }
                            else if (values[0] == "Id")
                            {
                                groupUserControl.Id = new Guid(values[1]);
                                comboItem.Tag = values[1];
                            }

                        }
                        comboBox_Group.Items.Add(comboItem);
                        data = client.DownloadString(About.Server + "Products/" + ProductId);
                        data = data.Remove(data.Length - 1);
                        data = data.Substring(1, data.Length - 1);

                        foreach (string column in data.Split(new string[] { "," }, StringSplitOptions.None))
                        {
                            string[] item = column.Split(':');

                            item[0] = item[0].Remove(item[0].Length - 1);
                            item[0] = item[0].Substring(1, item[0].Length - 1);

                            item[1] = item[1].Remove(item[1].Length - 1);
                            item[1] = item[1].Substring(1, item[1].Length - 1);

                            if (item[0] == "ProductName")
                                textbox_Name.Text = item[1];
                            else if (item[0] == "ProductDescription")
                                textbox_Description.Text = item[1];
                        }
                    }
                }
            }
            catch
            {
                System.Windows.MessageBox.Show("خطا در دریافت اطلاعات از سرور");
                Close();
            }
        }


        private string path;

        private void button_Choose_Click(object sender, RoutedEventArgs e)
        {
            Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();
            dlg.DefaultExt = ".png";
            dlg.Filter = "JPEG Files (*.jpeg)|*.jpeg|PNG Files (*.png)|*.png|JPG Files (*.jpg)|*.jpg|GIF Files (*.gif)|*.gif";
            Nullable<bool> result = dlg.ShowDialog();
            if (result == true)
            {
                path = dlg.FileName;
                button_Choose.Content = "تصویر انتخاب شد";
            }
        }

        private async void button_Submit_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                byte[] data = File.ReadAllBytes(path);

                HttpClient httpClient = new HttpClient();
```

```
                MultipartFormDataContent form = new MultipartFormDataContent();

                form.Add(new StringContent(textbox_Name.Text), "Name");
                form.Add(new StringContent(((ComboBoxItem)comboBox_Group.SelectedItem).Tag.ToString()), "GroupId");
                form.Add(new StringContent(textbox_Description.Text), "Description");
                form.Add(new ByteArrayContent(data, 0, data.Count()), "Image", "PostedImageFile");
                HttpResponseMessage response = await httpClient.PutAsync(About.Server + "Products/" + ProductId, form);

                System.Windows.MessageBox.Show("عملیات با موفقیت انجام شد");
                Close();
            }
            catch {
                System.Windows.MessageBox.Show("خطا در برقراری ارتباط با سرور");
            }
        }

    }
}
```

____

Filename : ProductUserControl.xaml

```
<UserControl x:Class="Chekeh.ProductUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d"
           Height="33.321" Width="338.705"  Loaded="UserControl_Loaded">
    <Grid>
        <Label x:Name="label_Title" Content="عنوان محصول بارگزاری نشده است" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="button_Details" Content="جزئیات" HorizontalAlignment="Left" Margin="264,3,0,0" VerticalAlignment="Top" Width="75"
Click="button_Details_Click" Height="20" />
        <Button x:Name="button_Delete" Content="حذف" HorizontalAlignment="Left" Margin="184,3,0,0" VerticalAlignment="Top" Width="75"
Click="button_Delete_Click"/>
    </Grid>
</UserControl>
```

____

Filename : ProductUserControl.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for ProductUserControl.xaml
    /// </summary>
    public partial class ProductUserControl : UserControl
    {
        public ProductUserControl()
        {
            InitializeComponent();
        }

        public string ProductName { get; set; }
        public Guid Id { get; set; }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            label_Title.Content = ProductName;
        }
        private void button_Details_Click(object sender, RoutedEventArgs e)
        {
            new ProductDetails { ProductId = Id }.ShowDialog();
        }
        private void button_Delete_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                var request = WebRequest.Create(About.Server + "Products/" + Id.ToString());
                request.Method = "DELETE";
                var response = (HttpWebResponse)request.GetResponse();
                Visibility = Visibility.Hidden;
            }
            catch
            {
                MessageBox.Show("خطا در اعلام درخواست به سرور");
            }
        }
    }
}
```

Filename : Register.xaml

```
<Window x:Class="Chekeh.Register"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
        Title="Reigster" Height="438.376" Width="300" WindowStartupLocation="CenterScreen" Icon="favicon.ico">
    <Grid>
        <Label x:Name="label_Username" Content="نام کاربری" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="170,5,0,0"/>
        <Label x:Name="label_Password" Content="کلمه عبور" HorizontalAlignment="Left" Margin="170,36,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_ConfirmPass" Content="تکرار کلمه عبور" HorizontalAlignment="Left" Margin="170,67,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_Name" Content="نام" HorizontalAlignment="Left" Margin="170,97,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_LastName" Content="نام خانوادگی" HorizontalAlignment="Left" Margin="170,128,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_NationalCode" Content="کد ملی" HorizontalAlignment="Left" Margin="170,159,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_PostalCode" Content="کد پستی" HorizontalAlignment="Left" Margin="170,190,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_PhoneNumber" Content="تلفن" HorizontalAlignment="Left" Margin="170,221,0,0" VerticalAlignment="Top"/>
        <Label x:Name="label_Address" Content="آدرس" HorizontalAlignment="Left" Margin="170,252,0,0" VerticalAlignment="Top"/>
        <Button x:Name="button_Regsiter" Content="ثبت نام" HorizontalAlignment="Left" VerticalAlignment="Top" Width="75" Margin="45,366,0,0"
Click="button_Regsiter_Click"/>
        <TextBox x:Name="textbox_Username" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,9,0,0"/>
        <TextBox x:Name="textbox_FirstName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,101,0,0"/>
        <TextBox x:Name="textbox_LastName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,132,0,0"/>
        <TextBox x:Name="textbox_NationalCode" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,163,0,0"/>
        <TextBox x:Name="textbox_PostalCode" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,194,0,0"/>
        <TextBox x:Name="textbox_PhoneNumber" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"
Margin="45,225,0,0"/>
        <RichTextBox x:Name="textbox_Address" HorizontalAlignment="Left" Height="100" Margin="45,257,0,0" VerticalAlignment="Top"
Width="120"></RichTextBox>
        <PasswordBox x:Name="textbox_Password" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" Margin="45,39,0,0"/>
        <PasswordBox x:Name="textbox_ConfirmPass" HorizontalAlignment="Left" VerticalAlignment="Top" Height="23" Width="120" Margin="45,70,0,0"/>
    </Grid>
</Window>
```

Filename : Register.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Reigster.xaml
    /// </summary>
    public partial class Register : Window
    {
        public Register()
        {
            InitializeComponent();
        }

        private void button_Regsiter_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                if (textbox_Password.Password == textbox_ConfirmPass.Password)
                {
                    using (WebClient client = new WebClient())
                    {
                        client.Encoding = Encoding.UTF8;

                        NameValueCollection data = new NameValueCollection();
                        data["Username"] = textbox_Username.Text;
                        data["Password"] = textbox_Password.Password;
                        data["FirstName"] = textbox_FirstName.Text;
                        data["LastName"] = textbox_LastName.Text;
                        data["NationalCode"] = textbox_NationalCode.Text;
                        data["Address"] = new TextRange(textbox_Address.Document.ContentStart, textbox_Address.Document.ContentEnd).Text;
                        data["PostalCode"] = textbox_PostalCode.Text;
                        data["PhoneNumber"] = textbox_PhoneNumber.Text;
                        string response = Encoding.UTF8.GetString(client.UploadValues(About.Server + "Owner", data));

                        About.Username = textbox_Username.Text;
                        About.Status = "NotConfirmed";

                        new AdminMain { }.Show();
                        this.Close();
                    }

                }
                else
                {
                    MessageBox.Show("کلمه عبور با تکرار آن همخوانی ندارد");
                }
            }
            catch
            {
                MessageBox.Show("ارتباط با سرور برقرار نیست");
            }
        }
    }
}
```

```
______________________________
Filename : Status.xaml

<UserControl x:Class="Chekeh.Status"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d"
             Width="auto"
             Height="auto" Loaded="UserControl_Loaded"
>
    <Grid VerticalAlignment="Stretch" HorizontalAlignment="Stretch">
        <Label x:Name="label_username" Content="متن بارگذاری نشده است" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="button_accept" Content="تائید کردن" HorizontalAlignment="Left" Margin="128,0,0,0" VerticalAlignment="Top" Width="80"
Click="button_accept_Click"/>
        <Button x:Name="button_drop" Content="تائید نکردن" HorizontalAlignment="Left" Margin="208,0,0,0" VerticalAlignment="Top" Width="80"
Click="button_drop_Click"/>
        <Button x:Name="button_details" Content="جزئیات" HorizontalAlignment="Left" Margin="288,0,-75,0" VerticalAlignment="Top" Width="75"
Click="button_details_Click"/>

    </Grid>
</UserControl>
```

```
______________________________
Filename : Status.xaml.cs

using System;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Status.xaml
    /// </summary>
    public partial class Status : UserControl
    {
        public Guid OwnerId { get; set; }
        public string Username { get; set; }

        public Status()
        {
            InitializeComponent();
        }
        public void done()
        {
            button_accept.IsEnabled = button_details.IsEnabled = button_drop.IsEnabled = false;
            label_username.Content = "انجام شد";
        }

        public void SendStatus(string Status)
        {
            using (WebClient client = new WebClient())
            {
                client.Encoding = Encoding.UTF8;

                NameValueCollection nvc = new NameValueCollection();
                nvc["Status"] = Status;
                client.UploadValues(About.Server + "OwnerStatus/" + OwnerId, nvc);
            }
        }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            label_username.Content = Username;
        }

        private void button_details_Click(object sender, RoutedEventArgs e)
        {
            new UserDetails { UserId = OwnerId }.Show();

        }

        private void button_drop_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                SendStatus("NotConfirmed");
                done();
            }
            catch { MessageBox.Show("اعلام خطا از طرف سرور"); }

        }

        private void button_accept_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                SendStatus("Confirmed");
                done();
            }
            catch { MessageBox.Show("اعلام خطا از طرف سرور"); }
        }
    }
}
```

Filename : StoreDetails.xaml

```
<Window x:Class="Chekeh.StoreDetails"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
     WindowStartupLocation="CenterScreen"   Title="جزئیات مغازه" FlowDirection="RightToLeft" Height="300" Width="300" Loaded="Window_Loaded"
Icon="favicon.ico">
    <Grid>
        <Label x:Name="label_Name" Content="نام فروشگاه:" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top"/>
        <TextBox x:Name="textBox_Name" HorizontalAlignment="Left" Height="23" Margin="86,14,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="196"/>
        <Label x:Name="label_PostalCode" Content="کد پستی:" HorizontalAlignment="Left" Margin="10,41,0,0" VerticalAlignment="Top"/>
        <TextBox x:Name="textBox_PostalCode" HorizontalAlignment="Left" Height="23" Margin="86,44,0,0" TextWrapping="Wrap" VerticalAlignment="Top"
Width="196"/>
        <Label x:Name="label_Address" Content="آدرس:" HorizontalAlignment="Left" Margin="10,110,0,0" VerticalAlignment="Top"/>
        <TextBox x:Name="textBox_Address" HorizontalAlignment="Left" Height="112" Margin="86,110,0,0" TextWrapping="Wrap" AcceptsReturn="True"
VerticalAlignment="Top" Width="196"/>
        <Button x:Name="button_Updte" Content="به روز رسانی" HorizontalAlignment="Left" Margin="207,233,0,0" VerticalAlignment="Top" Width="75"
Click="button_Updte_Click"/>
        <Label x:Name="label_City" Content="شهر:" HorizontalAlignment="Left" Margin="12,72,0,0" VerticalAlignment="Top"/>
        <ComboBox x:Name="comboBox" HorizontalAlignment="Left" Margin="86,76,0,0" VerticalAlignment="Top" Width="196"/>
    </Grid>
</Window>
```

Filename : StoreDetails.xaml.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for StoreDetails.xaml
    /// </summary>
    public partial class StoreDetails : Window
    {
        public StoreDetails()
        {
            InitializeComponent();
        }

        public Guid Id { get; set; }

        private void button_Updte_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                WebRequest request = WebRequest.Create(About.Server + "Stores/" + Id.ToString());
                request.Method = "PUT";
                string postData = "Name=" + textBox_Name.Text + "&Address=" + textBox_Address.Text + "&postalCode=" + textBox_PostalCode.Text + "&CityId=" + ((ComboBoxItem)comboBox.SelectedItem).Tag.ToString() + "&OwnerId=" + About.Identifier;
                byte[] byteArray = Encoding.UTF8.GetBytes(postData);
                request.ContentType = "application/x-www-form-urlencoded";
                request.ContentLength = byteArray.Length;
                Stream dataStream = request.GetRequestStream();
                dataStream.Write(byteArray, 0, byteArray.Length);
                dataStream.Close();
                WebResponse response = request.GetResponse();
                // Console.WriteLine(((HttpWebResponse)response).StatusDescription);
                dataStream = response.GetResponseStream();
                StreamReader reader = new StreamReader(dataStream);
                string responseFromServer = reader.ReadToEnd();
                // Console.WriteLine(responseFromServer);
                reader.Close();
                dataStream.Close();
                response.Close();
                MessageBox.Show("عملیات با موفقیت انجام شد");

            }
            catch
            {
                MessageBox.Show("پاسخ مناسب از سرور دریافت نشد");
            }
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            try
            {
                using (WebClient client_City = new WebClient())
                {
                    client_City.Encoding = Encoding.UTF8;
                    string data1 = client_City.DownloadString(About.Server + "Cities");
                    data1 = data1.Remove(data1.Length - 2);
                    data1 = data1.Substring(2, data1.Length - 2);

                    comboBox.Items.Clear();

                    foreach (var row in data1.Split(new string[] { "},{" }, StringSplitOptions.None))
                    {
                        ComboBoxItem cityComboItem = new ComboBoxItem();

                        foreach (var column in row.Split(','))
                        {
                            string[] values = column.Split(':');

                            values[0] = values[0].Remove(values[0].Length - 1);
                            values[0] = values[0].Substring(1, values[0].Length - 1);

                            values[1] = values[1].Remove(values[1].Length - 1);
                            values[1] = values[1].Substring(1, values[1].Length - 1);

                            if (values[0] == "Name")
                            {
                                cityComboItem.Content = values[1];
                            }
                            else if (values[0] == "Id")
                            {
                                cityComboItem.Tag = new Guid(values[1]);
                            }

                        }
                        comboBox.Items.Add(cityComboItem);
```

```
                    }

                    using (WebClient client = new WebClient())
                    {
                        client.Encoding = Encoding.UTF8;

                        string data2 = client.DownloadString(About.Server + "Store/" + Id.ToString());
                        data2 = data2.Remove(data2.Length - 1);
                        data2 = data2.Substring(1, data2.Length - 1);

                        foreach (string item in data2.Split(new string[] { "," }, StringSplitOptions.None))
                        {
                            string column = item;
                            column = column.Remove(column.Length - 1);
                            column = column.Substring(1, column.Length - 1);

                            string[] field = column.Split(new string[] { "\":\"" }, StringSplitOptions.None);

                            if (field[0] == "StoreName")
                                textBox_Name.Text = field[1];
                            else if (field[0] == "StorePostalCode")
                                textBox_PostalCode.Text = field[1];
                            else if (field[0] == "StoreAddress")
                                textBox_Address.Text = field[1];
                        }

                    }
                }
                Main.StoreGotFocus = 0;
            }
            catch {
                MessageBox.Show("خطا در دریافت اطلاعات از سرور");
                this.Close();
            }

        }
    }
}


______________________________
Filename : StoreUserControl.xaml

<UserControl x:Class="Chekeh.StoreUserControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             xmlns:local="clr-namespace:Chekeh"
             mc:Ignorable="d" Height="26.614" Width="auto" FlowDirection="RightToLeft" Loaded="UserControl_Loaded">
    <Grid>
```

```
        <Label x:Name="label_Title" Content="عنوان فروشگاه بارگزاری نشده است" HorizontalAlignment="Left" VerticalAlignment="Top"/>
        <Button x:Name="button_Details" Content="جزئیات" HorizontalAlignment="Left" Margin="184,0,0,0" VerticalAlignment="Top" Width="75"
Click="button_Details_Click"/>
        <Button x:Name="button_Delete" Content="حذف" HorizontalAlignment="Left" Margin="260,0,0,0" VerticalAlignment="Top" Width="75"
Click="button_Delete_Click"/>
    </Grid>
</UserControl>
```

____
Filename : StoreUserControl.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for Store.xaml
    /// </summary>
    public partial class StoreUserControl : UserControl
    {
        public StoreUserControl()
        {
            InitializeComponent();
        }
        public string StoreName { get; set; }
        public Guid StoreId { get; set; }

        private void button_Details_Click(object sender, RoutedEventArgs e)
        {
            new StoreDetails { Id = StoreId }.ShowDialog();
        }

        private void UserControl_Loaded(object sender, RoutedEventArgs e)
        {
            label_Title.Content = StoreName;
        }
        private void button_Delete_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                var request = WebRequest.Create(About.Server + "Stores/" + StoreId.ToString());
                request.Method = "DELETE";
                var response = (HttpWebResponse)request.GetResponse();
                Visibility = Visibility.Hidden;
            }
            catch
            {
                MessageBox.Show("خطا در اعلام درخواست به سرور");
            }
        }
    }
}
```

Filename : UserDetails.xaml

```
<Window x:Class="Chekeh.UserDetails"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:Chekeh"
        mc:Ignorable="d"
      WindowStartupLocation="CenterScreen"  Title="جزئیات کاربر" Height="400" Width="300" FlowDirection="RightToLeft" Loaded="Window_Loaded"
ResizeMode="NoResize" ShowInTaskbar="False" Icon="favicon.ico">
    <Grid>
        <Button x:Name="button_Chat" Content="گفت‌وگو" HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Width="75"
Click="button_Chat_Click"/>
        <Label x:Name="label_Username" Content="نام کاربری: " HorizontalAlignment="Stretch" VerticalAlignment="Top" Margin="10,35,0,0"
FontWeight="Bold" FontSize="16"/>
        <Label x:Name="label_FirstName" Content="نام: " HorizontalAlignment="Stretch" Margin="10,71,10,0" VerticalAlignment="Top"/>
        <Label x:Name="label_LastName" Content="نام خانوادگی: " HorizontalAlignment="Stretch" Margin="10,102,10,0" VerticalAlignment="Top"/>
        <Label x:Name="label_NationalCode" Content="کد ملی: " HorizontalAlignment="Stretch" Margin="10,133,10,0" VerticalAlignment="Top"/>
        <Label x:Name="label_PostalCode" Content="کد پستی: " HorizontalAlignment="Stretch" Margin="10,164,10,0" VerticalAlignment="Top"/>
        <Label x:Name="label_PhoneNumber" Content="تلفن: " HorizontalAlignment="Stretch" Margin="10,195,10,0" VerticalAlignment="Top"/>
        <Label x:Name="label_Address" Content="آدرس: " HorizontalAlignment="Stretch" Margin="10,226,10,0" VerticalAlignment="Top"/>
    </Grid>
</Window>
```

Filename : UserDetails.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace Chekeh
{
    /// <summary>
    /// Interaction logic for UserDetails.xaml
    /// </summary>
    public partial class UserDetails : Window
    {
        public UserDetails()
        {
            InitializeComponent();
        }
        public Guid UserId { get; set; }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            try
            {
                using (WebClient client = new WebClient())
                {
                    client.Encoding = Encoding.UTF8;

                    string data = client.DownloadString(About.Server + "Owner/" + UserId);
                    data = data.Remove(data.Length - 1);
                    data = data.Substring(1, data.Length - 1);

                    foreach (string column in data.Split(new string[] { "," }, StringSplitOptions.None))
                    {
                        string[] item = column.Split(':');

                        item[0] = item[0].Remove(item[0].Length - 1);
                        item[0] = item[0].Substring(1, item[0].Length - 1);

                        item[1] = item[1].Remove(item[1].Length - 1);
                        item[1] = item[1].Substring(1, item[1].Length - 1);

                        if (item[0] == "Username")
                            label_Username.Content = item[1];
                        else if (item[0] == "FirstName")
                            label_FirstName.Content += item[1];
                        else if (item[0] == "LastName")
                            label_LastName.Content += item[1];
                        else if (item[0] == "NationalCode")
                            label_NationalCode.Content += item[1];
                        else if (item[0] == "Address")
                            label_Address.Content += item[1];
                        else if (item[0] == "PostalCode")
                            label_PostalCode.Content += item[1];
                        else if (item[0] == "PhoneNumber")
                            label_PhoneNumber.Content += item[1];
                    }
                }
            }
            catch
            {
                MessageBox.Show("خطا در دریافت اطلاعات");
                this.Close();
            }
        }

        private void button_Chat_Click(object sender, RoutedEventArgs e)
        {
            new AdminChat { ReciverId = UserId }.ShowDialog();
        }
    }
}
```