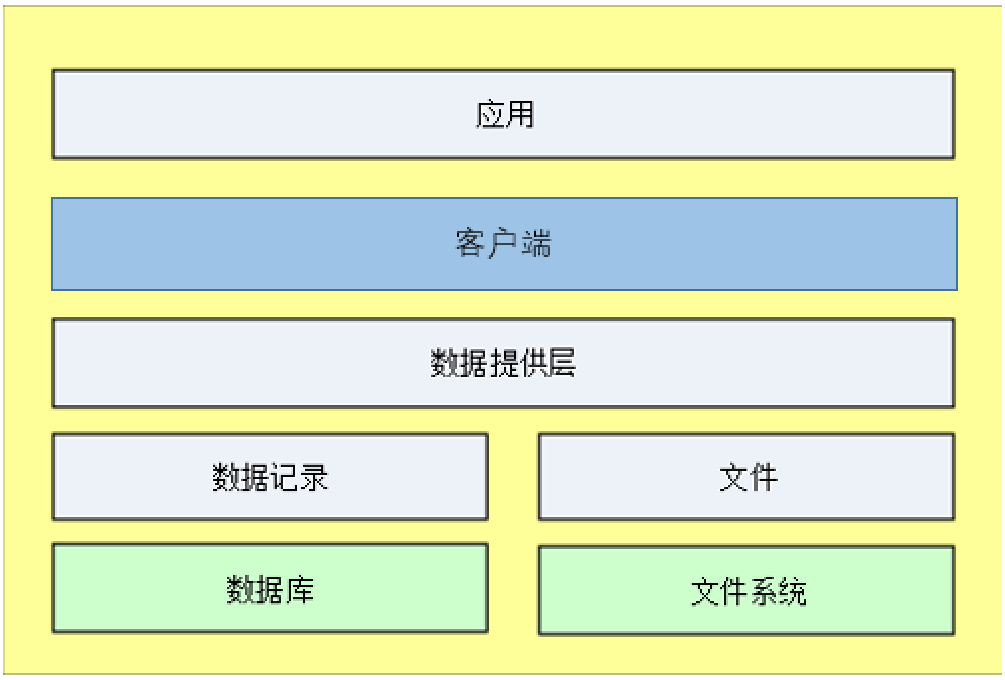
软件设计文档

一、 项目架构



二、模块划分

MainPage：主页模块

AddPage：信息添加模块

ContactsPage： 目录模块

Content：数据库结构模块

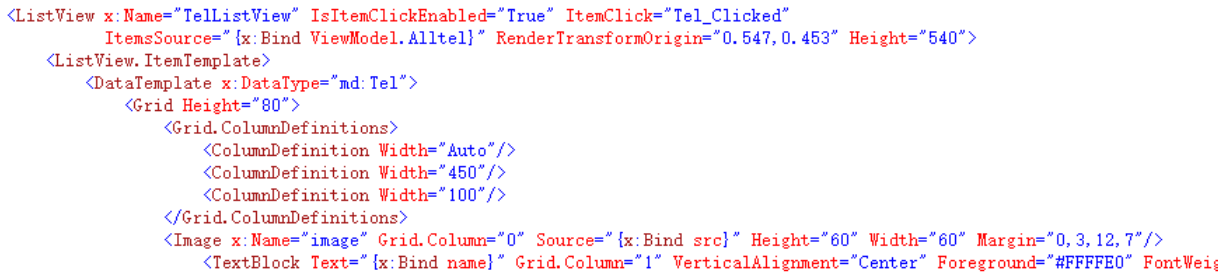
EditPage：信息编辑模块

Content1：归属地查询模块

三、 软件设计技术

1) DataBinding

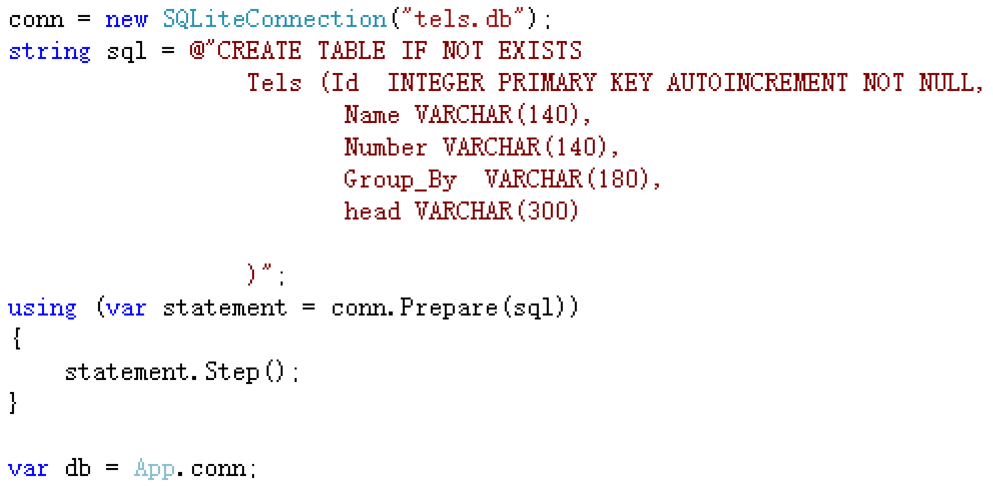
通过将 ListView 的 Item 设置为 DataBinding 来实现数据绑定，使得在联系人界面显示的信息与编辑界面显示的信息相同。



```
<ListView x:Name="TelListView" IsItemClickEnabled="True" ItemClick="Tel_Clicked"
          ItemsSource="{x:Bind ViewModel.Alltel}" RenderTransformOrigin="0.547,0.453" Height="540">
    <ListView.ItemTemplate>
        <DataTemplate x:DataType="md:Tel">
            <Grid Height="80">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto"/>
                    <ColumnDefinition Width="450"/>
                    <ColumnDefinition Width="100"/>
                </Grid.ColumnDefinitions>
                <Image x:Name="image" Grid.Column="0" Source="{x:Bind src}" Height="60" Width="60" Margin="0,3,12,7"/>
                    <TextBlock Text="{x:Bind name}" Grid.Column="1" VerticalAlignment="Center" Foreground="#FFFFE0" FontWeig
```

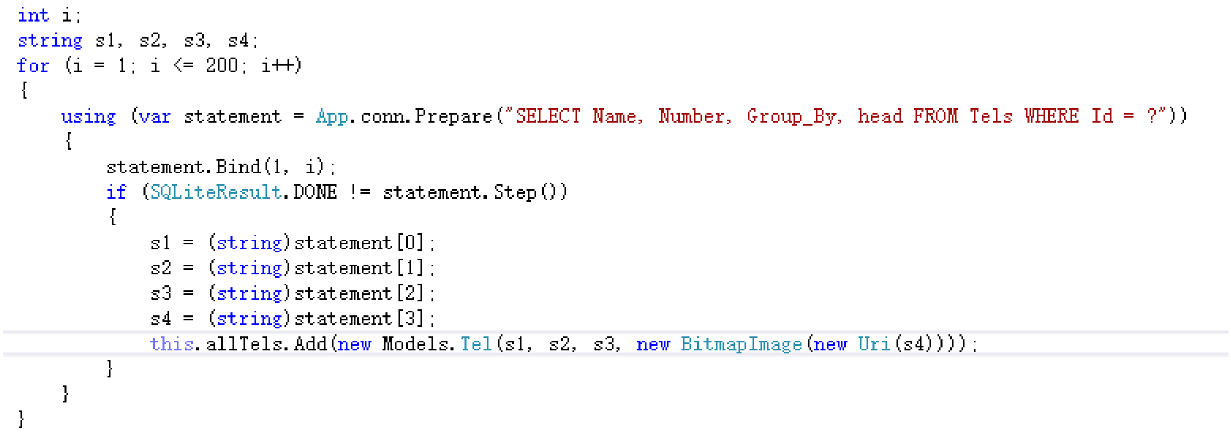
2) DataBase

在 App 文件中创建数据库文件并创建表格



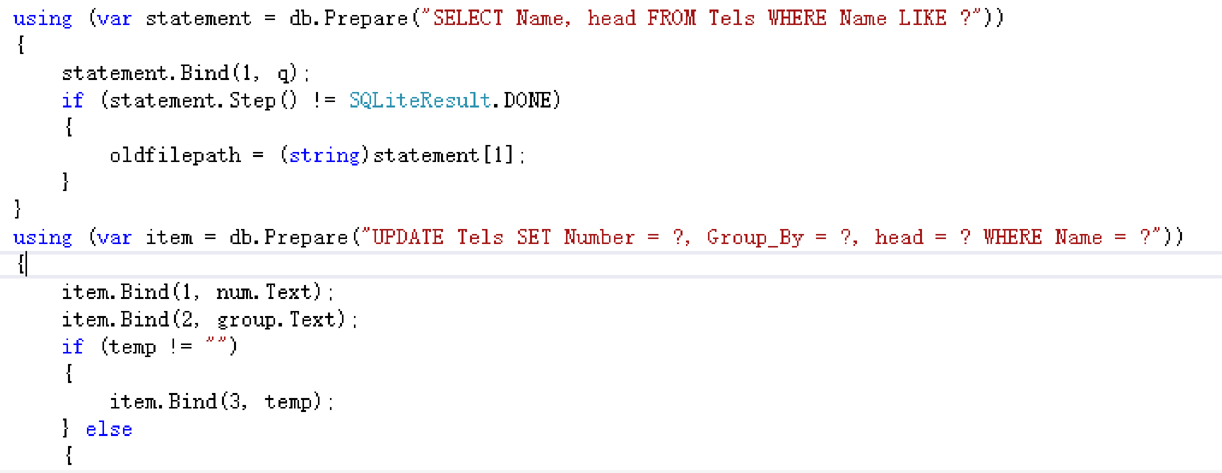
```
conn = new SQLiteConnection("tels.db");
string sql = @"CREATE TABLE IF NOT EXISTS
                 Tels (Id  INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
                        Name VARCHAR(140),
                        Number VARCHAR(140),
                        Group_By  VARCHAR(180),
                        head VARCHAR(300)

                 )";
using (var statement = conn.Prepare(sql))
{
    statement.Step();
}

var db = App.conn;
```

在 Telview 文件中根据数据库信息初始化界面

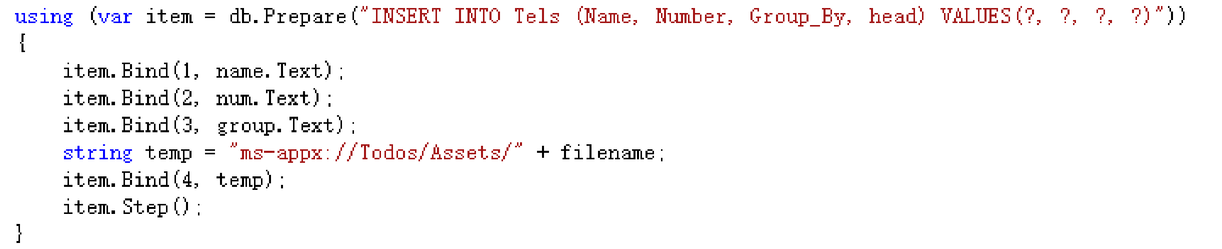


```
int i;
string s1, s2, s3, s4;
for (i = 1; i <= 200; i++)
{
    using (var statement = App.conn.Prepare("SELECT Name, Number, Group_By, head FROM Tels WHERE Id = ?"))
    {
        statement.Bind(1, i);
        if (SQLiteResult.DONE != statement.Step())
        {
            s1 = (string)statement[0];
            s2 = (string)statement[1];
            s3 = (string)statement[2];
            s4 = (string)statement[3];
            this.allTels.Add(new Models.Tel(s1, s2, s3, new BitmapImage(new Uri(s4))));
        }
    }
}
```

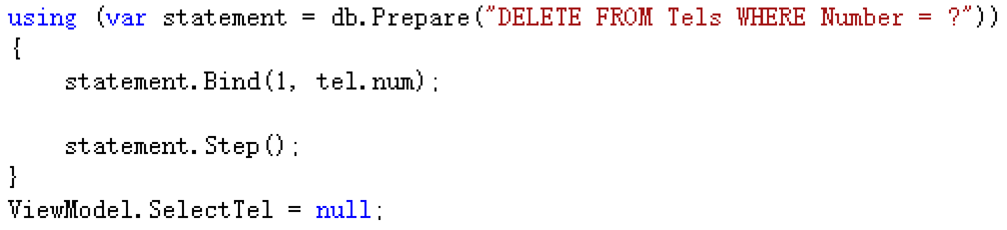
在 EditPage、AddPage、ContactsPage 等文件中通过 INSERT、UPDATE、DELETE、SELECT 等操作实现功能需要



```
using (var statement = db.Prepare("SELECT Name, head FROM Tels WHERE Name LIKE ?"))
{
    statement.Bind(1, q);
    if (statement.Step() != SQLiteResult.DONE)
    {
        oldfilepath = (string)statement[1];
    }
}
using (var item = db.Prepare("UPDATE Tels SET Number = ?, Group_By = ?, head = ? WHERE Name = ?"))
{
    item.Bind(1, num.Text);
    item.Bind(2, group.Text);
    if (temp != "")
    {
        item.Bind(3, temp);
    } else
    {
```



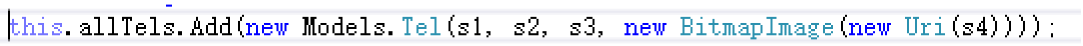
```
using (var item = db.Prepare("INSERT INTO Tels (Name, Number, Group_By, head) VALUES(?, ?, ?, ?)"))
{
    item.Bind(1, name.Text);
    item.Bind(2, num.Text);
    item.Bind(3, group.Text);
    string temp = "ms-appx://Todos/Assets/" + filename;
    item.Bind(4, temp);
    item.Step();
}
```



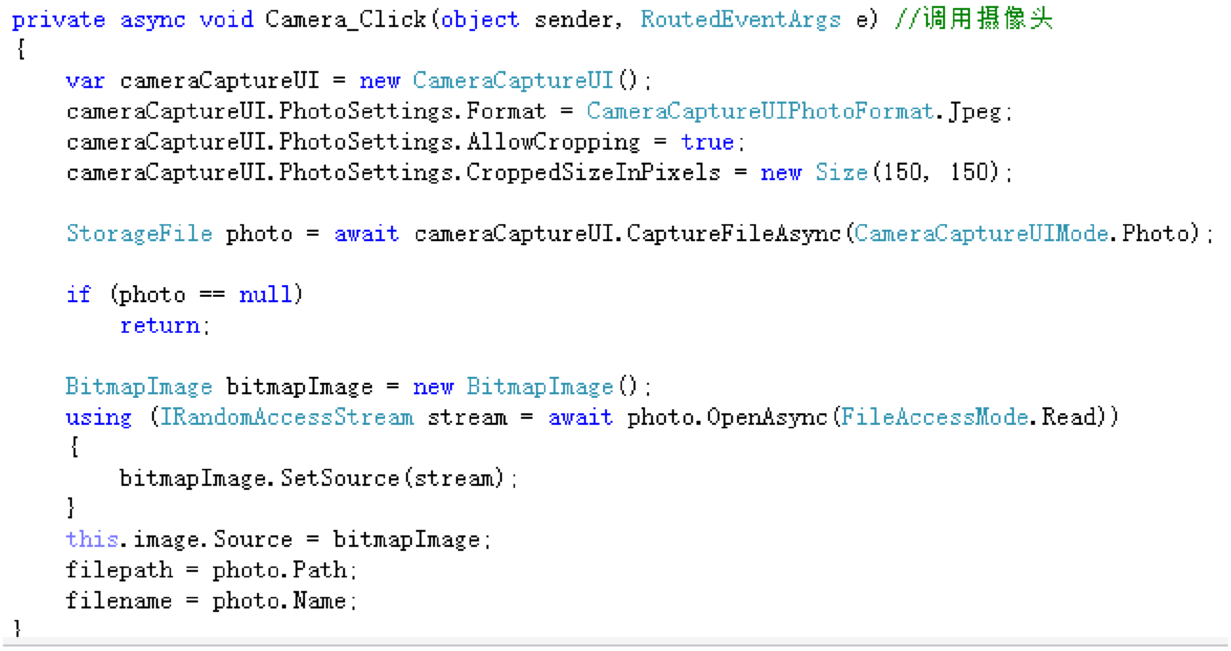
```
using (var statement = db.Prepare("DELETE FROM Tels WHERE Number = ?"))
{
    statement.Bind(1, tel.num);

    statement.Step();
}
ViewModel.SelectTel = null;
```

3) App to app communication

项目代码中通过添加图片、调用摄像头等功能实现 App to app communication。



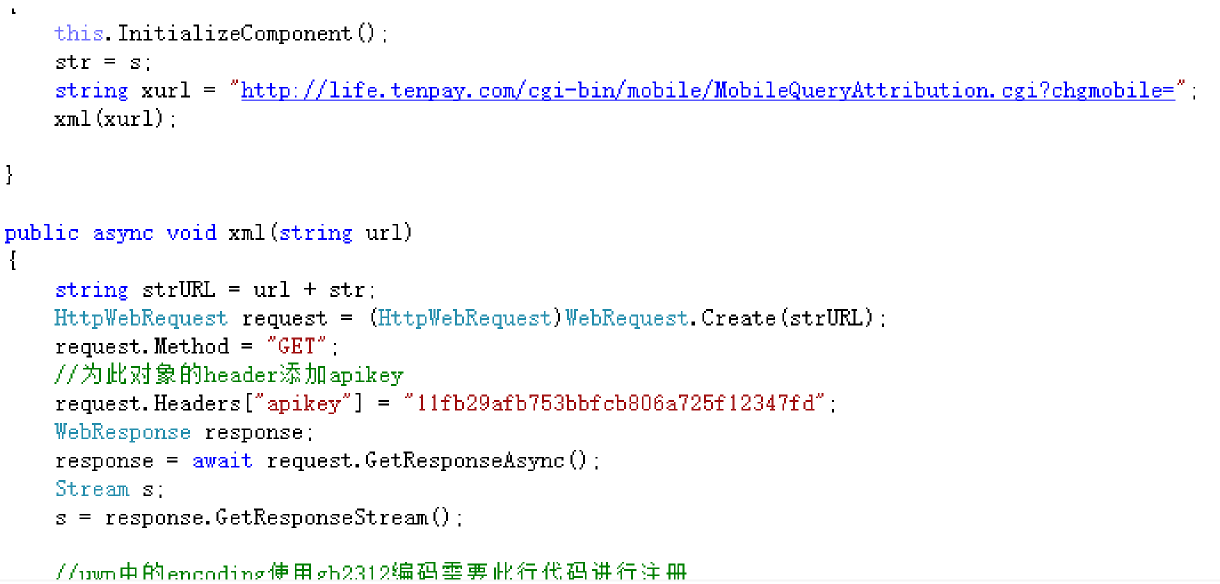
```
this.allTels.Add(new Models.Tel(s1, s2, s3, new BitmapImage(new Uri(s4))));
```



```
private async void Camera_Click(object sender, RoutedEventArgs e) //调用摄像头
{
    var cameraCaptureUI = new CameraCaptureUI();
    cameraCaptureUI.PhotoSettings.Format = CameraCaptureUIPhotoFormat.Jpeg;
    cameraCaptureUI.PhotoSettings.AllowCropping = true;
    cameraCaptureUI.PhotoSettings.CroppedSizeInPixels = new Size(150, 150);

    StorageFile photo = await cameraCaptureUI.CaptureFileAsync(CameraCaptureUIMode.Photo);

    if (photo == null)
        return;

    BitmapImage bitmapImage = new BitmapImage();
    using (IRandomAccessStream stream = await photo.OpenAsync(FileAccessMode.Read))
    {
        bitmapImage.SetSource(stream);
    }
    this.image.Source = bitmapImage;
    filepath = photo.Path;
    filename = photo.Name;
}
```

4) Network Accessing

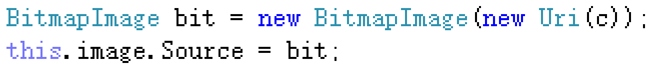
号码归属地查询的功能使用到了网络访问，在 content1 文件中。



```
        this.InitializeComponent();
        str = s;
        string xurl = "http://life.tenpay.com/cgi-bin/mobile/MobileQueryAttribution.cgi?chgmobile=";
        xml(xurl);

    }

    public async void xml(string url)
    {
        string strURL = url + str;
        HttpWebRequest request = (HttpWebRequest)WebRequest.Create(strURL);
        request.Method = "GET";
        //为此对象的header添加apikey
        request.Headers["apikey"] = "11fb29afb753bbfcb806a725f12347fd";
        WebResponse response;
        response = await request.GetResponseAsync();
        Stream s;
        s = response.GetResponseStream();

        //uwp中的encoding使用gb2312编码需要此行代码进行注册
```

5) File Management

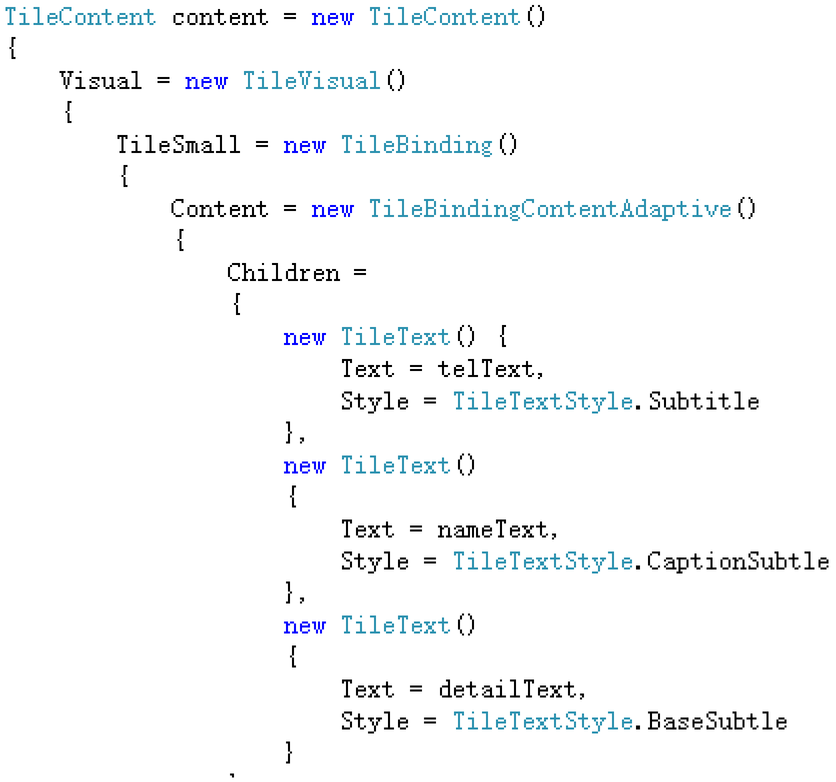
项目中在多处添加图片时使用 uri 访问了图片文件。



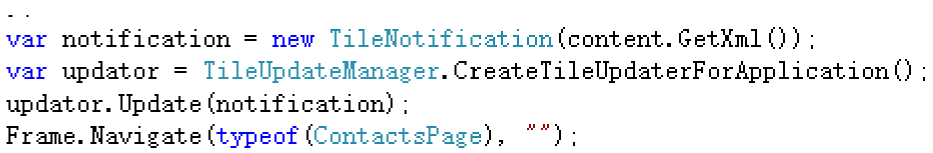
```
BitmapImage bit = new BitmapImage(new Uri(c));
this.image.Source = bit;
```

6) Live Tiles

项目中实现了 win10 的 3D 动态磁贴，在 MainPage 的文件中。



```
TileContent content = new TileContent()
{
    Visual = new TileVisual()
    {
        TileSmall = new TileBinding()
        {
            Content = new TileBindingContentAdaptive()
            {
                Children =
                {
                    new TileText() {
                        Text = telText,
                        Style = TileTextStyle.Subtitle
                    },
                    new TileText()
                    {
                        Text = nameText,
                        Style = TileTextStyle.CaptionSubtle
                    },
                    new TileText()
                    {
                        Text = detailText,
                        Style = TileTextStyle.BaseSubtle
                    }
```



```
..
var notification = new TileNotification(content.GetXml());
var updator = TileUpdateManager.CreateTileUpdaterForApplication();
updator.Update(notification);
Frame.Navigate(typeof(ContactsPage), "");
```