Laborator 10 – Aplicatii mobile multi-platforma cu .NET MAUI

1. In laboratorul curent, vom dezvolta aplicatia dezvoltata in laboratorul 9. Deschidem Visual Studio si apoi alegem optiunea **Open a project or solution** si cautam proiectul **Nume_Pren_Lab7,** creat in laboratoarele anteriore. Laboratorul curent il vom dezvolta pe un branch nou. Utilizand pasii indicati in Lab 1, pct. 22 vom crea un branch nou (based on Laborator9) pe care il vom denumi Laborator.
2. Vom modifica aplicatia realizata in lab 9 astfel incat sa putem introduce adresa unui magazin de unde dorim sa facem cumparaturi. Pe baza adresei introduse aplicatia va putea deschide harta si vom obine indicatii de navigare. De asemenea daca distanta dintre locatia dispozitivului si locatia magazinului este mai mica de un 1 km, aplicatia ne va trimite o notificare.
3. Pentru a retine detalii despre magazine vom crea o noua clasa cu denumirea Shop astfel. In ferestra Solution Explorer, apasam click dreapta pe folderul Models -> Add->Class si vom denumi fisierul Shop.cs. Acesta va avea urmatorul continut:

```
public class Shop
    {

            [PrimaryKey, AutoIncrement]
            public int ID { get; set; }

            public string ShopName { get; set; }

            public string Adress { get; set; }

            public string ShopDetails { get { return ShopName+"
"+Adress;} }
            [OneToMany]
            public List<ShopList> ShopLists { get; set; }

    }
```

4. Intre ShopList si Shop va exista o relatie de One-To-Many deci modificam clasa ShopList adaugand cheia straina ShopID astfel:

```
    public class ShopList
        {
            [PrimaryKey, AutoIncrement]
            public int ID { get; set; }

            [MaxLength(250), Unique]
            public string Description { get; set; }

            public DateTime Date { get; set; }

            [ForeignKey(typeof(Shop))]
            public int ShopID { get; set; }


        }
```

5. In proiectul Nume_Pren_Lab7 adaugam o pagina de tip ContentPage. In fereastra Solution Explorer facem click dreapta pe numele proiectului Nume_Pren_Lab7, selectam Add->New Item si din categoria .NETMAUI alegem ContentPage pe care o denumim **ShopEntryPage.xaml.**

6. Vom utiliza un ListView care utilizeaza Databinding pentru a afisa toate magazinele la care dorim sa facem cumparaturi, selectarea unuia dintre aceste magazine va duce la ShopPage care permite modificarea detaliilor magazinului selectat. Vom folosi ToolbarItem pentru a crea un nou magazin la apasarea acestuia. Din fereastra **Solution Explorer** deschidem fisierul ShopEntryPage.xaml si modificam continutul acestuia astfel:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Nume_Pren_Lab7.ShopEntryPage"
             Title="ShopEntryPage">
    <ContentPage.ToolbarItems>
        <ToolbarItem Text="Add Shop"
                     Clicked="OnShopAddedClicked" />
    </ContentPage.ToolbarItems>
    <ListView x:Name="listView"
              Margin="20"
              ItemSelected="OnListViewItemSelected">
        <ListView.ItemTemplate>
            <DataTemplate>
                <TextCell Text="{Binding ShopName}"
                          Detail="{Binding Adress}" />
            </DataTemplate>
        </ListView.ItemTemplate>
    </ListView>
</ContentPage>
```

7. Din fereastra **Solution Explorer** deschidem fisierul ShopEntryPage.xaml.cs si adaugam la continutul acestuia codul de mai jos:

```
using Nume_Pren_Lab7.Models;

namespace Nume_Pren_Lab7;

public partial class ShopEntryPage : ContentPage
{
   public ShopEntryPage()
   {
         InitializeComponent();
   }

    protected override async void OnAppearing()
    {
        base.OnAppearing();

        listView.ItemsSource = await App.Database.GetShopsAsync();
    }

    async void OnShopAddedClicked(object sender, EventArgs e)
    {
        await Navigation.PushAsync(new ShopPage
        {
                BindingContext = new Shop()
            });
        }

        async void OnListViewItemSelected(object sender,
SelectedItemChangedEventArgs e)
        {
            if (e.SelectedItem != null)
            {
                await Navigation.PushAsync(new ShopPage
                {
                    BindingContext = e.SelectedItem as Shop
                });
            }

        }
    }
```

1. In proiectul Nume_Pren_Lab7 adaugam o pagina de tip ContentPage. In fereastra Solution Explorer facem click dreapta pe numele proiectului Nume_Pren_Lab7, selectam Add->New Item si din categoria .NET MAUI alegem ContentPage (XAML) pe care o denumim **ShopPage.xaml**
2. Vom utiliza doua elemente de tip Editor pentru a introduce Numele magazinului si Adresa precum si un buton Save.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Nume_Pren_Lab7.ShopPage"
             Title="ShopPage">
    <ContentPage.Content>
        <StackLayout Margin="20">
            <Editor Placeholder="Enter shop name" Margin="20"
                Text="{Binding ShopName}"
                HeightRequest="50" />
            <Editor Placeholder="Enter shop adress" Margin="20"
                Text="{Binding Adress}"
                HeightRequest="50" />
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="Auto" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="*" />
                </Grid.ColumnDefinitions>
                <Button Text="Save Shop" Grid.Column="0"
MaximumWidthRequest="200"
                    Clicked="OnSaveButtonClicked" />
            </Grid>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

3. Din fereastra **Solution Explorer** deschidem fisierul ShopPage.xaml.cs si adaugam la continutul acestuia codul de mai jos. Cand este apasat butonul Save, handler-ul de eveniment `OnSaveButtonClicked` este executat, lista de cumparaturi este salvata in baza de date si aplicatia navigheaza la pagina precedenta.

```
using Nume_Pren_Lab7.Models;

namespace Nume_Pren_Lab7;

public partial class ShopPage : ContentPage
{

    public ShopPage()
   {
          InitializeComponent();

   }

    async void OnSaveButtonClicked(object sender, EventArgs e)
    {
        var shop = (Shop)BindingContext;
        await App.Database.SaveShopAsync(shop);
        await Navigation.PopAsync();
    }
}
```

4. Deschidem fisierul AppShell.xaml si modificam codul astfel incat in TabBar sa apara noua pagina create ShopEntryPage

```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="Nume_Pren_Lab7.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:Nume_Pren_Lab7"
    Shell.FlyoutBehavior="Disabled">

    <TabBar>
     <ShellContent
         Title="Welcome"
         ContentTemplate="{DataTemplate local:MainPage}"
          />

     <ShellContent
         Title="My Shopping Lists"
         ContentTemplate="{DataTemplate local:ListEntryPage}"
          />

     <ShellContent
         Title="My Shops"
         ContentTemplate="{DataTemplate local:ShopEntryPage}"
          />

     <ShellContent
         Title="About"
         ContentTemplate="{DataTemplate local:AboutPage}"
             />
```

```
        </TabBar>

    </Shell>
```

8. In fisierul **Data/ShoppingListDatabase** adaugam cod pentru crearea si citirea datelor pentru clasa Shop. Adaugam in Clasa **ShoppingListDatabase** urmatorul continut:

```
using SQLite;
using System.Collections.Generic;
using System.Threading.Tasks;
using Nume_Pren_Lab7.Models;

namespace Nume_Pren_Lab7.Data
{
    public class ShopListDatabase
     {
        readonly SQLiteAsyncConnection _database;

        public ShopListDatabase(string dbPath)
        {
            _database = new SQLiteAsyncConnection(dbPath);
            _database.CreateTableAsync<ShopList>().Wait();
            _database.CreateTableAsync<Product>().Wait();
            _database.CreateTableAsync<ListProduct>().Wait();
              _database.CreateTableAsync<Shop>().Wait();
        }

………

        public Task<List<Shop>> GetShopsAsync()
          {
              return _database.Table<Shop>().ToListAsync();
          }

          public Task<int> SaveShopAsync(Shop shop)
          {
              if (shop.ID != 0)
               {
                   return _database.UpdateAsync(shop);
               }
               else
               {
                   return _database.InsertAsync(shop);
              }
           }
     }
```

9. Dorim ca aplicatia sa deschisa harsta si sa afiseze modul in care putem ajunge la adresa indicate a magazinului. Astfel in fisierul ShopPage.xmls adaugam un buton care va deschide harta:

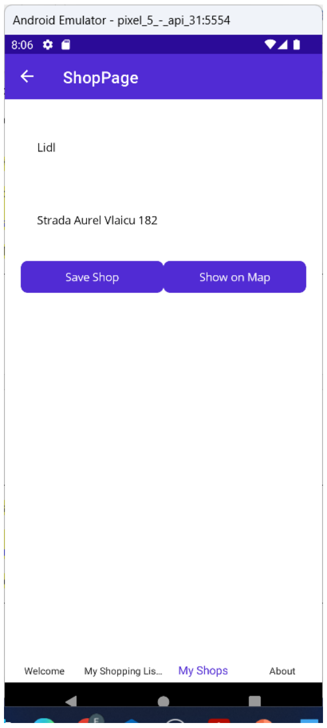
```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Nume_Pren_Lab7.ShopPage"
             Title="ShopPage">
    <ContentPage.Content>
        <StackLayout Margin="20">
            <Editor Placeholder="Enter shop name" Margin="20"
                Text="{Binding ShopName}"
                HeightRequest="50" />
            <Editor Placeholder="Enter shop adress" Margin="20"
                Text="{Binding Adress}"
                HeightRequest="50" />
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="Auto" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="*" />
                </Grid.ColumnDefinitions>
                <Button Text="Save Shop" Grid.Column="0"
MaximumWidthRequest="200"
                    Clicked="OnSaveButtonClicked" />
                <Button Text="Show on Map" Grid.Column="1"
MaximumWidthRequest="200"
                    Clicked="OnShowMapButtonClicked"></Button>

            </Grid>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

10. La apasarea butonului Show On Map se va declansa evenimentul Click care va fi gestionat de eventHandler-ul OnShowMapButtonClicked. Pentru a detecta locatia curenta a dispozitivului utilizam clasa Geolocation. Alternativ putem seta noi o locatie utilizand valori pentru latitudine si longitudine. In fisierul ShopPage.xaml.cs adaugam:

```
...

async void OnSaveButtonClicked(object sender, EventArgs e)
    {
        var shop = (Shop)BindingContext;
        await App.Database.SaveShopAsync(shop);
        await Navigation.PopAsync();
    }

    async void OnShowMapButtonClicked(object sender, EventArgs e)
    {
        var shop = (Shop)BindingContext;
        var address = shop.Adress;
        var locations = await Geocoding.GetLocationsAsync(address);

        var options = new MapLaunchOptions { Name = "Magazinul meu
preferat" };
```

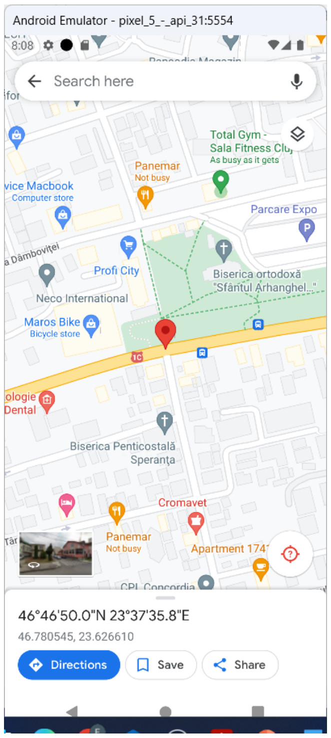
```
        var location = locations?.FirstOrDefault();

       // var myLocation = await Geolocation.GetLocationAsync();
         var myLocation = new Location(46.7731796289, 23.6213886738);

        await Map.OpenAsync(location, options);
    }
```

11. Rulam aplicatia si introducem un magazin cu o adresa valida pentru acesta. Apasam apoi butonul Show on Map si observam ca se deschide Harta



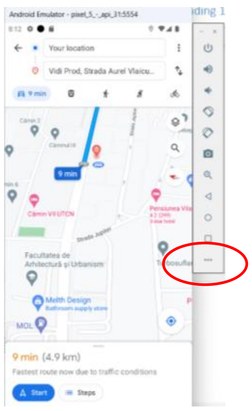
Lidl

Strada Aurel Vlaicu 182



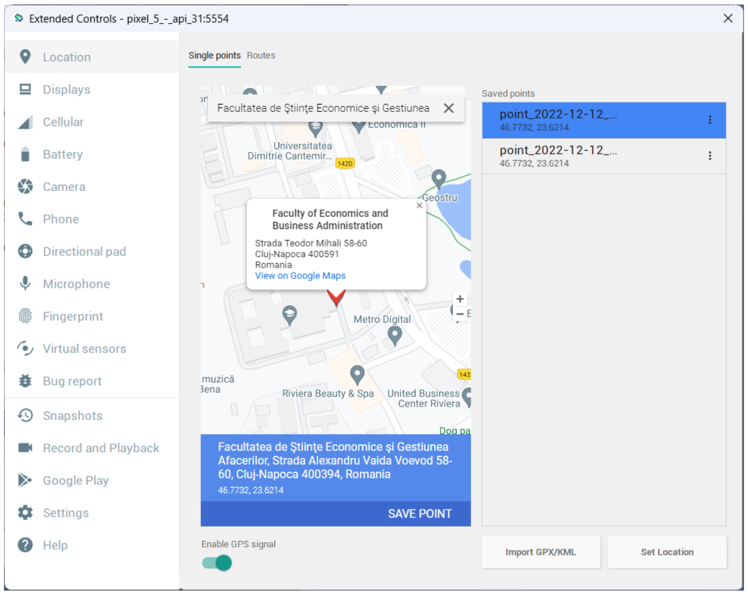
Android Emulator - pixel_5_-_api_31:5554
8:08
Search here
Total Gym - Sala Fitness Club
Panemar
Parcare Expo
Profi City
Neco International
Maros Bike
Biserica ortodoxă "Sfântul Arhanghel..."
Biserica Penticostală Speranța
Cromavet
Directions
Save
Share

46°46'50.0"N 23°37'35.8"E
46.780545, 23.626610



Android Emulator - pixel_5_-_api_31:5554
Your location
Vidi Prod, Strada Aurel Vlaicu...
9 min
9 min (4.9 km)
Fastest route now due to traffic conditions
Start
Steps

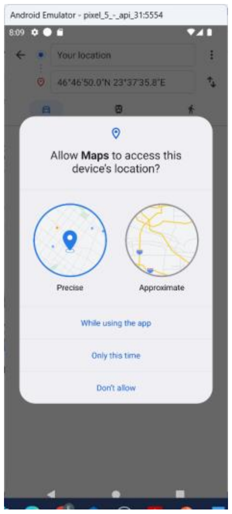
12. Pentru a seta locatia curenta in emulatorul Android apasam
Iar in fereastra care se deschide cautam o locatie apoi apasam butonul **Set Location**



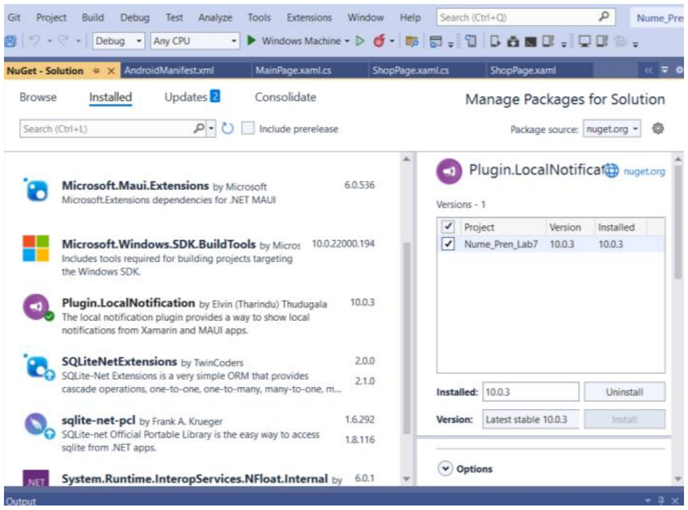
13. Pentru a permite aplicatiei sa acceseze locatia curenta pe dispozitive Android este necesar sa adaugam in fisierul Platforms->Android-> AndroidManifest.xml urmatoarele instructiuni:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
      <application android:allowBackup="true" android:icon="@mipmap/appicon"
android:roundIcon="@mipmap/appicon_round"
android:supportsRtl="true"></application>
      <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
      <uses-permission android:name="android.permission.INTERNET" />

  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
/>
</manifest>
```

14. Rulam aplicatia din nou si observam ca daca dorim sa utilizam locatia curenta se cere permisiunea din partea utilizatorului:


Android Emulator - pixel_5_-_api_31:5554
Your location
46°46'50.0"N 23°37'35.8"E
Allow Maps to access this device's location?
Precise
Approximate
While using the app
Only this time
Don't allow

15. Dorim ca aplicatia sa ne trimita o notificare daca ne aflam la o distanta relativ mica fata de magazin. Vom adauga pachetul **Plugin.LocalNotification** de la meniul Tools->NuGet Package Manager->Manage Nuget Packages for Solution



Manage Packages for Solution

Package source: nuget.org

Microsoft.Maui.Extensions by Microsoft 6.0.536
Microsoft.Extensions dependencies for .NET MAUI

Microsoft.Windows.SDK.BuildTools by Micros 10.0.22000.194
Includes tools required for building projects targeting the Windows SDK.

Plugin.LocalNotification by Elvin (Tharindu) Thudugala 10.0.3
The local notification plugin provides a way to show local notifications from Xamarin and MAUI apps.

SQLiteNetExtensions by TwinCoders 2.0.0 2.1.0
SQLite-Net Extensions is a very simple ORM that provides cascade operations, one-to-one, one-to-many, many-to-one, m...

sqlite-net-pcl by Frank A. Krueger 1.6.292 1.8.116
SQLite-net Official Portable Library is the easy way to access sqlite from .NET apps.

System.Runtime.InteropServices.NFloat.Internal by 6.0.1

Plugin.LocalNotificat nuget.org

Versions - 1

✓ Project	Version	Installed
✓ Nume_Pren_Lab7	10.0.3	10.0.3

Installed: 10.0.3

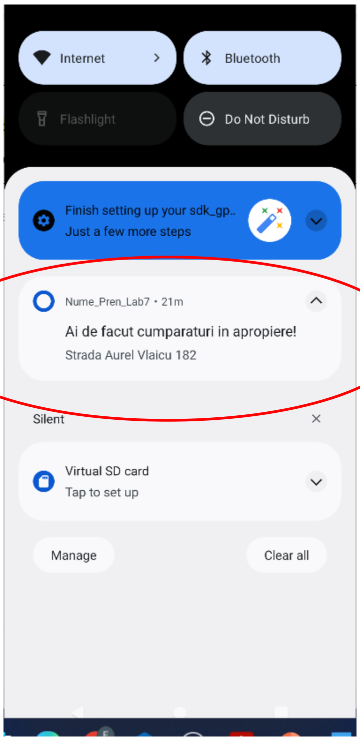
Version: Latest stable 10.0.3

16. In fisierul ShopPage.xaml.cs modificam hander-ul de eveniment. Calculam distanta locatiei curente fata de adresa magazinukui. Daca distanta este mai mica de 4 km vom trimite o notiifcare astfel:

```
async void OnShowMapButtonClicked(object sender, EventArgs e)
    {
        var shop = (Shop)BindingContext;
        var address = shop.Adress;
        var locations = await Geocoding.GetLocationsAsync(address);

        var options = new MapLaunchOptions { Name = "Magazinul meu 
preferat" };

        var location = locations?.FirstOrDefault();

       // var myLocation = await Geolocation.GetLocationAsync();
         var myLocation = new Location(46.7731796289, 23.6213886738);

        var distance = myLocation.CalculateDistance(location, 
DistanceUnits.Kilometers);

        if (distance < 4)
        {

            var request = new NotificationRequest
            {

                Title = "Ai de facut cumparaturi in apropiere!",
                Description = address,
                Schedule = new NotificationRequestSchedule
                {
                    NotifyTime = DateTime.Now.AddSeconds(1)
                }

            };
```

```
                LocalNotificationCenter.Current.Show(request);
            }
            await Map.OpenAsync(location, options);
        }
```

17. Rulam aplicatia si observam ca daca se indeplineste conditia se trimite notificarea:



Nume_Pren_Lab7 • 21m
Ai de facut cumparaturi in apropiere!
Strada Aurel Vlaicu 182

18. Adaugam in fisierul ListPage.xaml un Picker in care vom incarca toate magazinele introduse in aplicatie. Utilizatorul va alege din lista un magazin atasat listei de cumparaturi curente:

```
...
                <Editor.Behaviors>
                    <local:ValidationBehaviour />
                </Editor.Behaviors>
            </Editor>
            <Picker x:Name="ShopPicker" Title="Enter Shop Name"/>
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="*" />
                </Grid.ColumnDefinitions>
                <Button Text="Save"
                    Clicked="OnSaveButtonClicked" />
                <Button Grid.Column="1"
                    Text="Delete"
                    Clicked="OnDeleteButtonClicked"/>
            </Grid>
...
```

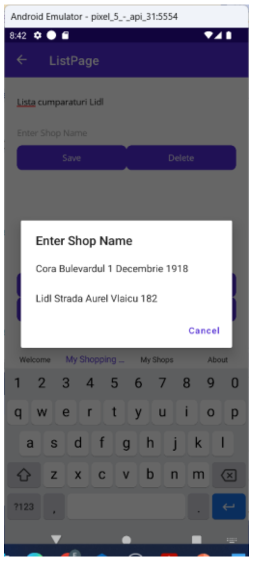
19. In fisierul ListPage.xaml.cs adaugam in metoda OnAppearing instructiunile care vor incarca magazinele disponibile cu adresele acestora:

```
      protected override async void OnAppearing()
        {
            base.OnAppearing();

            var items =  await App.Database.GetShopsAsync();
            ShopPicker.ItemsSource = (System.Collections.IList)items;

            ShopPicker.ItemDisplayBinding = new Binding("ShopDetails");

            var shopl = (ShopList)BindingContext;

            listView.ItemsSource = await
    App.Database.GetListProductsAsync(shopl.ID);
```

20. In fisierul ListPage.xaml.cs adaugam in hander-ul de eveniment `OnSaveButtonClicked` instructiunile care vor salva id-ul magazinului la lista de cumparaturi atasata.

```
async void OnSaveButtonClicked(object sender, EventArgs e)
    {
        var slist = (ShopList)BindingContext;
        slist.Date = DateTime.UtcNow;
        Shop selectedShop = (ShopPicker.SelectedItem as Shop);
        slist.ShopID = selectedShop.ID;
        await App.Database.SaveShopListAsync(slist);
        await Navigation.PopAsync();
    }
```

21. Rulam aplicatia si observam ca la adaugarea unei liste de cumparaturi putem alege acum din lista de magazine predefinite:



Lista cumparaturi Lidl

Enter Shop Name

Enter Shop Name

Cora Bulevardul 1 Decembrie 1918

Lidl Strada Aurel Vlaicu 182

Cancel

Sarcina laborator: Realizati implementarea operatiunii de stergere a unui magazin (in pagina ShopPage adaugati un buton Delete si implementarea functionalitatilor aferente)