



Univerzitet u Sarajevu
Fakultet za saobraćaj i komunikacije

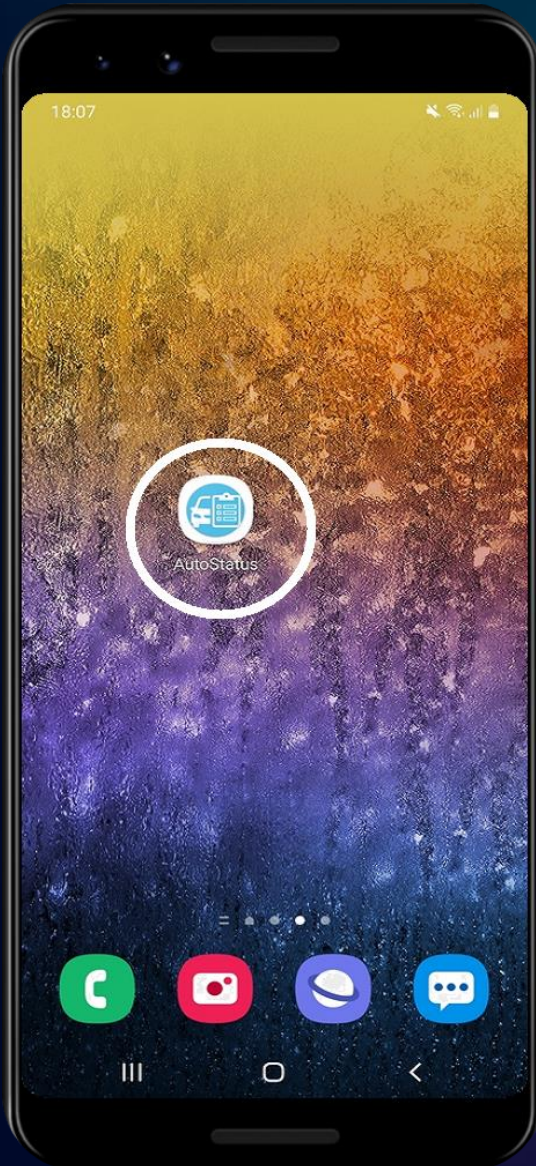


eAutomehaničar „AutoStatus“

Uvod

- Zbog konstantnog porasta prodaje i servisiranja automobila javlja se potreba za izradom mobilnih aplikacija koje će biti od koristi autoserviserima, sadašnjim vlasnicima vozila te budućim vlasnicima vozila u vidu tačne historije vozila. Kroz ovu aplikaciju korisnici će imati mogućnosti brzog pregleda svih servisnih podataka kroz historiju vozila, pregled dostupnih autoservisa i cjenovnik istih.
- Aplikacija se sastoji od: Početne stranice, Menija i Sekcija za: Pregled servisne historije automobila, Pregled dostupnih autoservisa, Kalkulatora servisnih intervala,
- NuGet paketa (pored defaultnih): Syncfusion.Xamarin.SfDataGrid, Xamarin.Forms.PinchZoomImage.

Launch icon

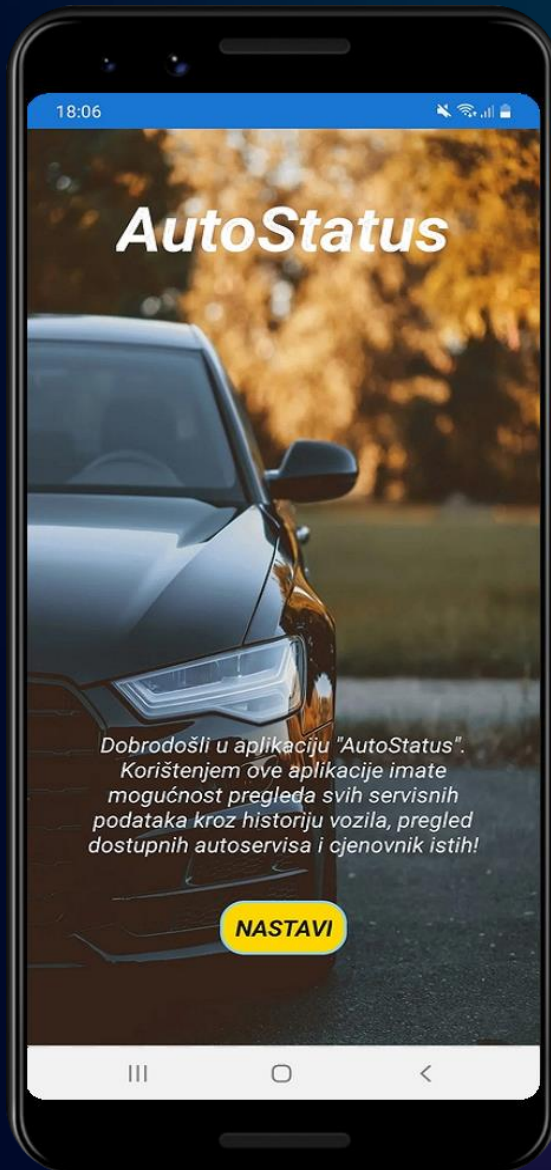


MainActiviti.cs

```
[Activity(Label = "AutoStatus", Icon = "@mipmap/launch", Theme = "@style/MainTheme",  
MainLauncher = true, ConfigurationChanges = ConfigChanges.ScreenSize |  
ConfigChanges.Orientation | ConfigChanges.UiMode | ConfigChanges.ScreenLayout |  
ConfigChanges.SmallestScreenSize )]
```

```
└─ mipmap-anydpi-v26  
   └─ icon.xml  
   └─ icon_round.xml  
└─ mipmap-hdpi  
   └─ icon.png  
   └─ launch.png  
   └─ launcher_foreground.png  
└─ mipmap-mdpi  
   └─ icon.png  
   └─ launch.png  
   └─ launcher_foreground.png  
└─ mipmap-xhdpi  
   └─ icon.png  
   └─ launch.png  
   └─ launcher_foreground.png  
└─ mipmap-xxhdpi  
   └─ icon.png  
   └─ launch.png  
   └─ launcher_foreground.png  
└─ mipmap-xxxhdpi  
   └─ icon.png  
   └─ launch.png  
   └─ launcher_foreground.png
```

Početna stranica



- Naslov
- Kratki opis
- Dugme

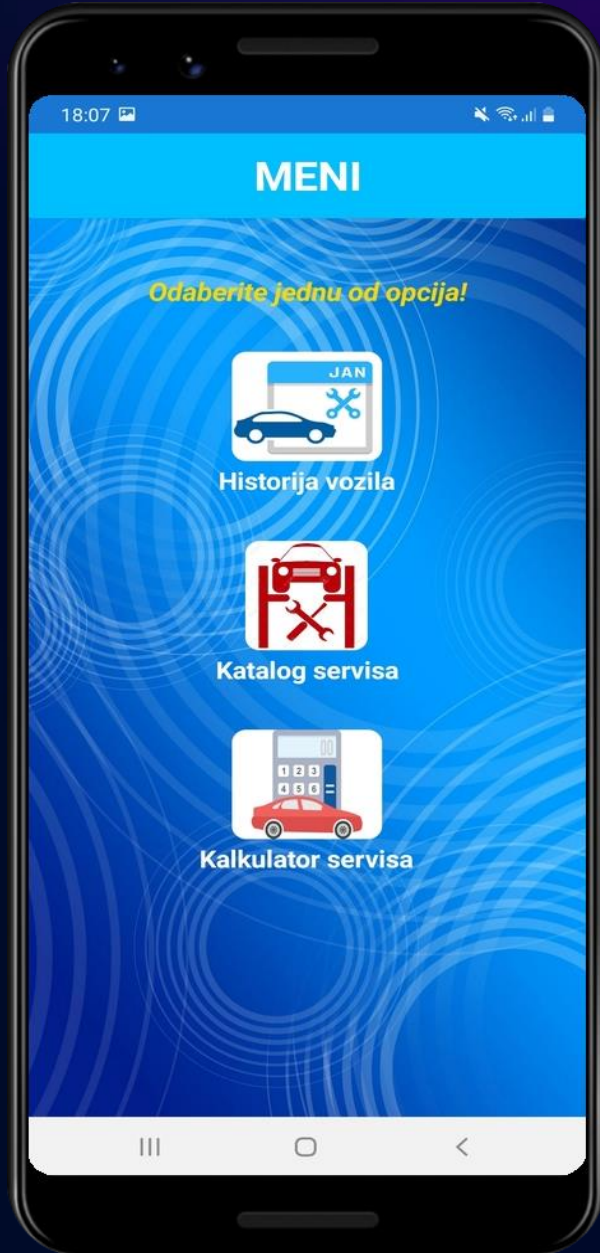
Pocetna.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="AutoStatus.Pocetna"
    BackgroundImage="pocetna.jpg"
    NavigationPage.HasNavigationBar="False">
    <Grid Margin="30,50,30,80">
        <Grid.RowDefinitions>
            <RowDefinition Height="370" />
            <RowDefinition />
        </Grid.RowDefinitions>
        <Label Grid.Row="0" Text="AutoStatus" HorizontalOptions="Center"
            FontAttributes="Italic,Bold"
            TextColor="White" FontSize="50"/>
        <Label Grid.Row="1" TranslationY="10" Text="Dobrodošli u aplikaciju "AutoStatus".
            Korištenjem ove aplikacije imate mogućnost pregleda svih servisnih podataka kroz
            historiju vozila, pregled dostupnih autoservisa i cjenovnik istih!"
            TextColor="AliceBlue" FontSize="17" FontAttributes="Italic"
            HorizontalTextAlignment="Center" VerticalOptions="Center" />
        <Button Grid.Row="1" BackgroundColor="Gold" Text="Nastavi" FontSize="18"
            FontAttributes="Italic,Bold" BorderColor="LightSkyBlue" BorderWidth="2"
            CornerRadius="20" VerticalOptions="End" HorizontalOptions="Center"
            Clicked="Button_Clicked" ></Button>
    </Grid>
</ContentPage>
```

Pocetna.xaml.cs

```
private async void Button_Clicked(object sender,
    EventArgs e)
    {
        await Navigation.PushAsync(new MainPage());
    }
```


MENI



- Isključen HasNavigationBar
- ImageButton-i
- Navigacija kroz stranice

MainPage.Xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage x:Class="AutoStatus.MainPage"
    NavigationPage.HasNavigationBar="False"
    BackgroundImage="meni">
    <StackLayout>
        <Frame Margin="0,0,0,40" Padding="10,10,10,10" BackgroundColor="DeepSkyBlue"
            HeightRequest="50">
            <Label Text="MENI" BackgroundColor="DeepSkyBlue"
                FontAttributes="Bold" VerticalOptions="Center" TextColor="White" FontSize="30"
                HorizontalTextAlignment="Center"/>
            </Frame>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition />
                    <RowDefinition />
                    <RowDefinition />
                </Grid.RowDefinitions>
                <ImageButton Margin="0,30,0,30" Grid.Row="0" Source="historija.jpg"
                    CornerRadius="10" HeightRequest="90" HorizontalOptions="Center" Clicked="ImageButton_Clicked" />
                <Label Grid.Row="0" Text="Historija vozila"
                    VerticalOptions="End" FontAttributes="Bold"
                    HorizontalOptions="Center" TextColor="White" FontSize="18" />
            </Grid>
        </StackLayout>
    </ContentPage>
```

MainPage.xaml.cs

```
private async void ImageButton_Clicked(object sender, EventArgs e)
{
    await Navigation.PushAsync(new Provjera());
}

private async void ImageButton_Clicked_1(object sender, EventArgs e)
{
    await Navigation.PushAsync(new Servis());
}

private async void ImageButton_Clicked_2(object sender, EventArgs e)
{
    await Navigation.PushAsync(new Kalkulator());
}
```

Provjera pristupa



- Polja za unos
- Dugme za pregled lozinke
- Validacija podataka

Provjera.xaml.cs

```
private void Nazad1(object sender, EventArgs e)
{
    Navigation.PopAsync();
}

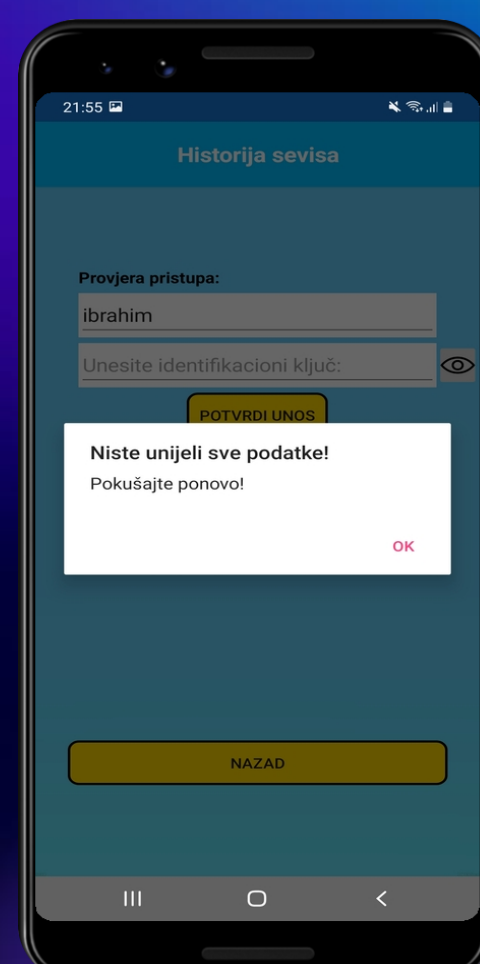
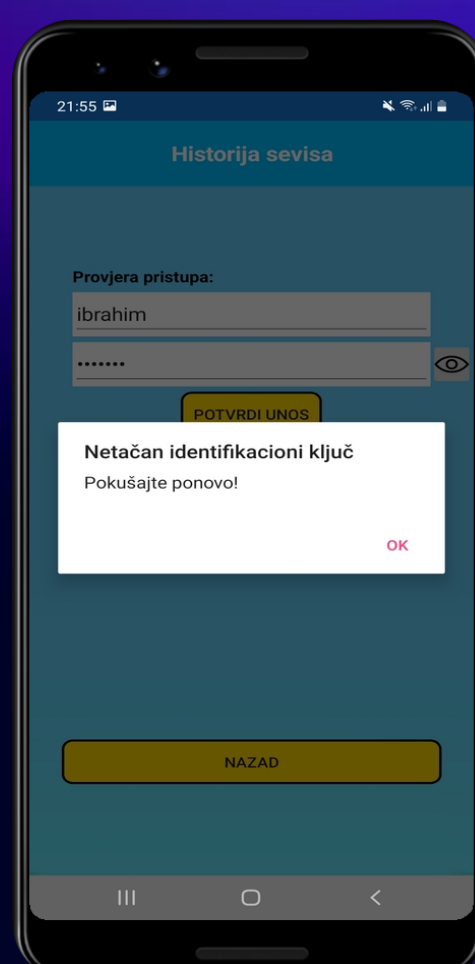
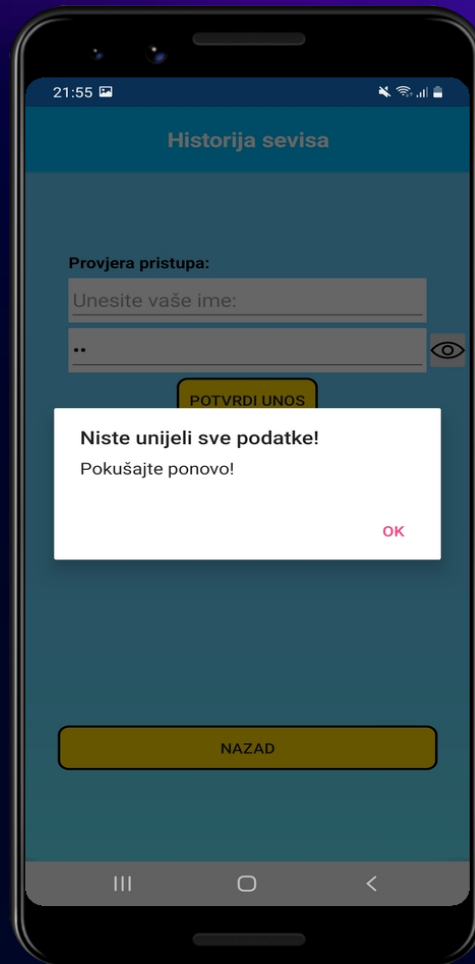
private async void Button_Clicked(object sender, EventArgs e)
{
    string x = Ime.Text;
    string y = Unos.Text;

    if (x == "emir" && y == "emir9999")
    {
        await Navigation.PushAsync(new Historija1());
    }
    else if (x == "ibrahim" && y == "ibrahim996")
    {
        await Navigation.PushAsync(new Historija2());
    }
    else if (x == "elvir" && y == "elvir9999")
    {
        await Navigation.PushAsync(new Historija3());
    }
    else if (x == string.Empty && y == string.Empty)
    {
        await DisplayAlert("Niste unijeli sve podatke!", "Pokušajte ponovo!", "OK");
    }
    else if (x.Any(char.IsDigit))
    {
        await DisplayAlert("Ime ne smije sadržavati brojeve", "Pokušajte ponovo!", "OK");
    }
    else if (x == "emir" && y != "emir9999")
    {
        await DisplayAlert("Netačan identifikacioni ključ", "Pokušajte ponovo!", "OK");
    }
}
```

```
        await DisplayAlert("Netačan identifikacioni ključ", "Pokušajte ponovo!", "OK");
    }
    else if (x == "ibrahim" && y != "ibrahim996")
    {
        await DisplayAlert("Netačan identifikacioni ključ", "Pokušajte ponovo!", "OK");
    }
    else if (x == "elvir" && y != "elvir9999")
    {
        await DisplayAlert("Netačan identifikacioni ključ", "Pokušajte ponovo!", "OK");
    }
    else
    {
        await DisplayAlert("Nemate pristup!", "Pokušajte ponovo!", "OK");
    }
}

private void ImageButton_Pressed(object sender, EventArgs e)
{
    Unos.IsPassword = false;
}

private void ImageButton_Released(object sender, EventArgs e)
{
    Unos.IsPassword = true;
}
```



Historija servisnih podataka (Informacije o vozilu)



- ScrollView
- ListView
- CarouselView
- Picker

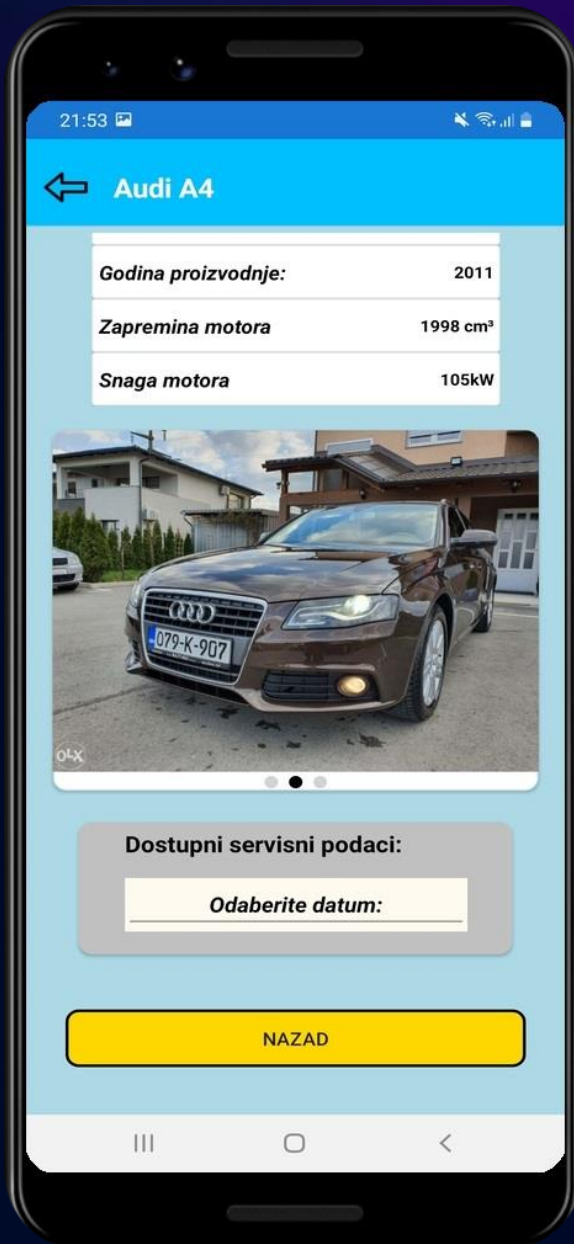
Historija1.Xaml

```
<ScrollView>
    <ListView ItemsSource="{Binding Historija}"
        Grid.Row="1" SelectionMode="None" IsEnabled="False" >
        <ListView.ItemTemplate>
            <DataTemplate>
                <ViewCell>
                    <Frame BackgroundColor="White" BorderColor="#F0F0F0"
                        Padding="5" >
                        <Grid HeightRequest="90" >
                            <Label Grid.Row="0" Text="{Binding Info}"
                                TextColor="Black" FontSize="15" FontAttributes="Bold,Italic"
                                VerticalTextAlignment="Center" />
                            <Label Text="{Binding Podatak}" Grid.Column="1"
                                TextColor="Black" FontSize="12" FontAttributes="Bold"
                                HorizontalTextAlignment="End" VerticalTextAlignment="Center"/>
                        </Grid>
                    </Frame>
                </ViewCell>
            </DataTemplate>
        </ListView.ItemTemplate>
    </ListView>

    <CarouselView x:Name="SlikeCarousel" IndicatorView="Indikator">
        <CarouselView.ItemTemplate>
            <DataTemplate>
                <Image Source="{Binding Slika}" Aspect="Fill" />
            </DataTemplate>
        </CarouselView.ItemTemplate>
    </CarouselView>

    <IndicatorView BackgroundColor="Transparent" x:Name="Indikator"
        TranslationY="-3" IndicatorColor="LightGray"
        SelectedIndicatorColor="Black" IndicatorSize="10"
        VerticalOptions="Center" />

    <Picker BackgroundColor="FloralWhite" Margin="30,0,30,0" FontSize="16"
        FontAttributes="Bold,Italic"
        HorizontalTextAlignment="Center" x:Name="Piker" Title="Odaberite datum:"
        TitleColor="Black" SelectedIndexChanged="Piker_SelectedIndexChanged">
    </Picker>
</ScrollView>
```

- Klase (Slike, Vozilo)
- Lista slika za (CarouselView)
- Liste za prikaz informacija o vozilu
- Dodani podaci za piker i navigacija stranica

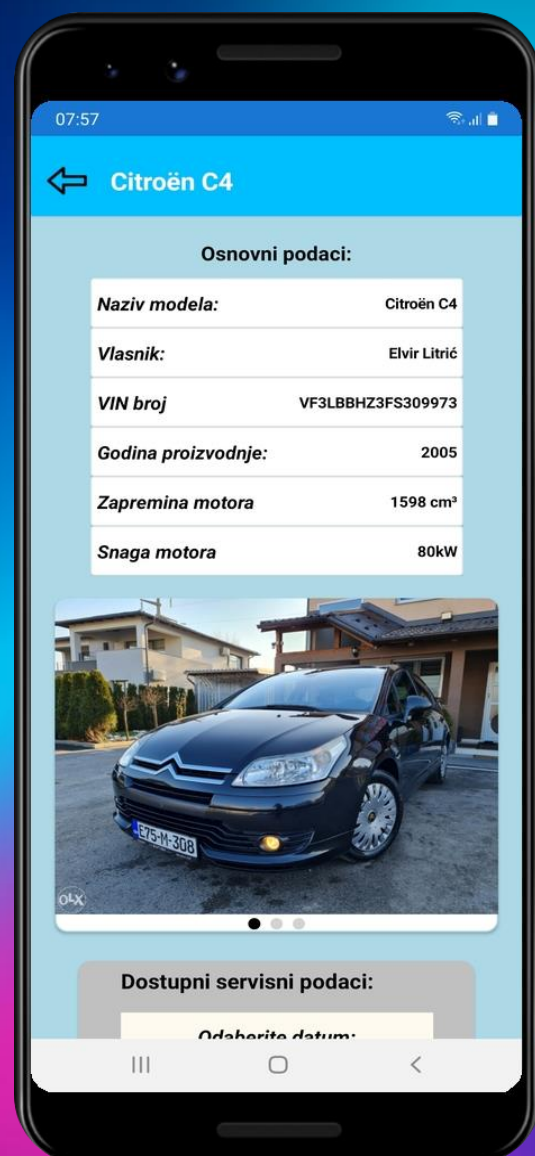
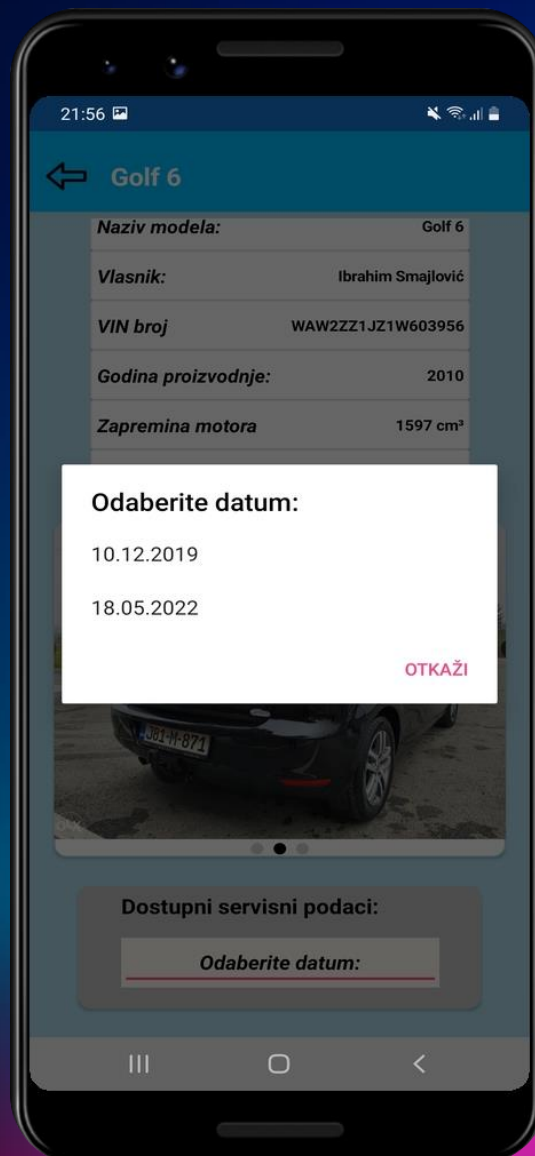
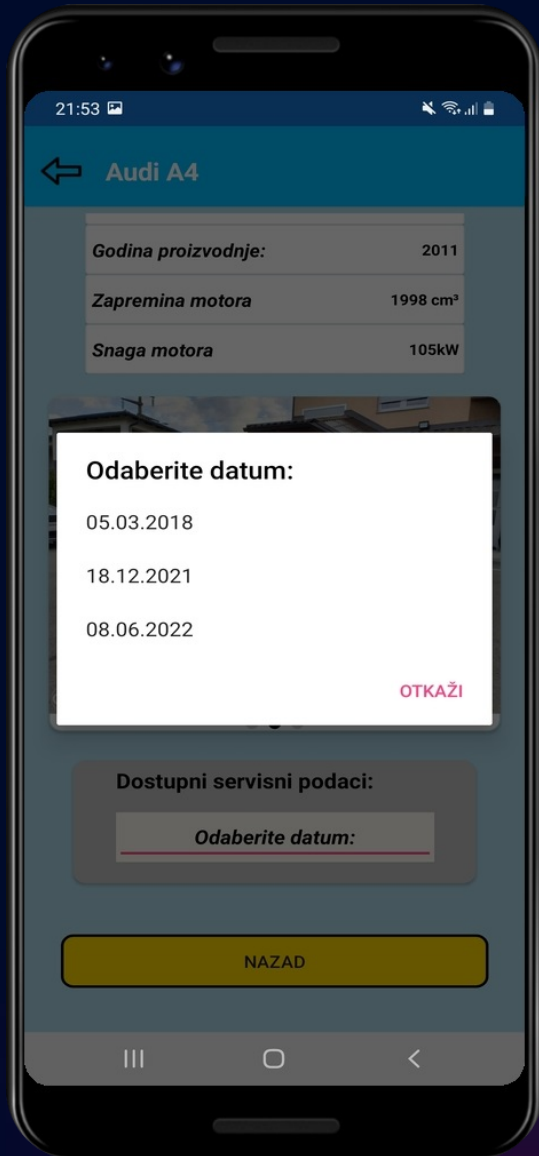
```
public Historija1()
{
    InitializeComponent();
    Picker.Items.Add("05.03.2018");
    Picker.Items.Add("18.12.2021");
    Picker.Items.Add("08.06.2022");

    List<Slike> slika = new List<Slike>()
    {
        new Slike(){ Naslov="Slika 1", Slika="audi1"},
    };
    SlikeCarousel.ItemsSource= slika;
}

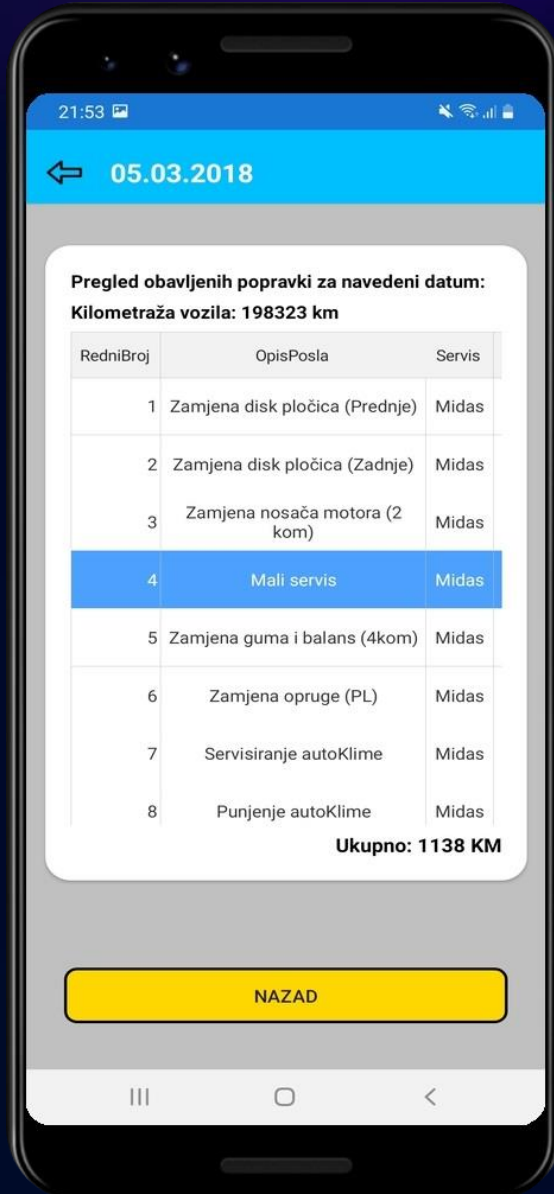
public List<Vozilo> Historija { get => Podaci(); }
private List<Vozilo> Podaci()
{
    var ServisiList = new List<Vozilo>();
    ServisiList.Add(new Vozilo { Info = "Naziv modela: ", Podatak =
"Audi A4" });
    ServisiList.Add(new Vozilo { Info = "Vlasnik: ", Podatak = "Emir
Ćato" });
    return ServisiList;
}
private async void Picker_SelectedIndexChanged(object sender, EventArgs
e)
{
    if (Picker.SelectedItem == "05.03.2018")
    {
        await Navigation.PushAsync(new Historija1a());
    }
    else if (Picker.SelectedItem == "18.12.2021")
    {
        await Navigation.PushAsync(new Historija1b());
    }
    else if (Picker.SelectedItem == "08.06.2022")
    {
        await Navigation.PushAsync(new Historija1c());
    }
}
}

public class Slike
{
    public string Naslov { get; set; }
    public string Slika { get; set; }
}

public class Vozilo
{
    public string Info { get; set; }
    public string Podatak { get; set; }
}
```



Historija servisnih podataka (Pregled servisnih informacija)



- NuGet:
Syncfusion.Xamarin.Sf
DataGrid
- Tabelarni prikaz
- Moguće kretanje
kroz tabelu
- Selektovanje
redova posebno

Historija1a.Xaml

```
xmlns:Syncfusion="clr-
namespace:Syncfusion.SfDataGrid.XForms;assembly=Syncfusion.SfDataGrid.XForms"
xmlns:local="clr-namespace:AutoStatus"

<ContentPage.BindingContext >
  <local:Historija1A/>
</ContentPage.BindingContext>

<Frame x:Name="Frame1" Margin="15,30,15,40" CornerRadius="20">
  <StackLayout >
    <Label Text="Pregled obavljenih popravki za navedeni datum:"
      TextColor="Black" FontAttributes="Bold" FontSize="14"></Label>
    <Label Text="Kilometraža vozila: 198323 km"
      TextColor="Black" FontAttributes="Bold" FontSize="14"></Label>
    <Syncfusion:SfDataGrid x:Name="dataGrid"
      ItemsSource="{Binding ServisneInfo }" SelectionMode="Single"
      ColumnSizer="Auto">
      <Syncfusion:SfDataGrid.GridStyle>
        <Syncfusion:DataGridStyle >
        </Syncfusion:DataGridStyle>
      </Syncfusion:SfDataGrid.GridStyle>
    </Syncfusion:SfDataGrid>
    <Label FontSize="15" FontAttributes="Bold" TextColor="Black"
      HorizontalTextAlignment="End" Text="Ukupno: 1138 KM"></Label>
  </StackLayout>
</Frame>
```



- Biblioteka
- Konstruktor
- Kolekcija

Historija1a.xaml.cs

```
using System.Collections.ObjectModel;
```

```
public class ServisneInformacije
{
    private int redniBroj;
    private string opisPosla;
    private string servis;
    private string zamjenskidio;
    private int cijenaDijela;
    private int brRadnihSati;
    private int cijenaUsluge;
    private int ukupno;

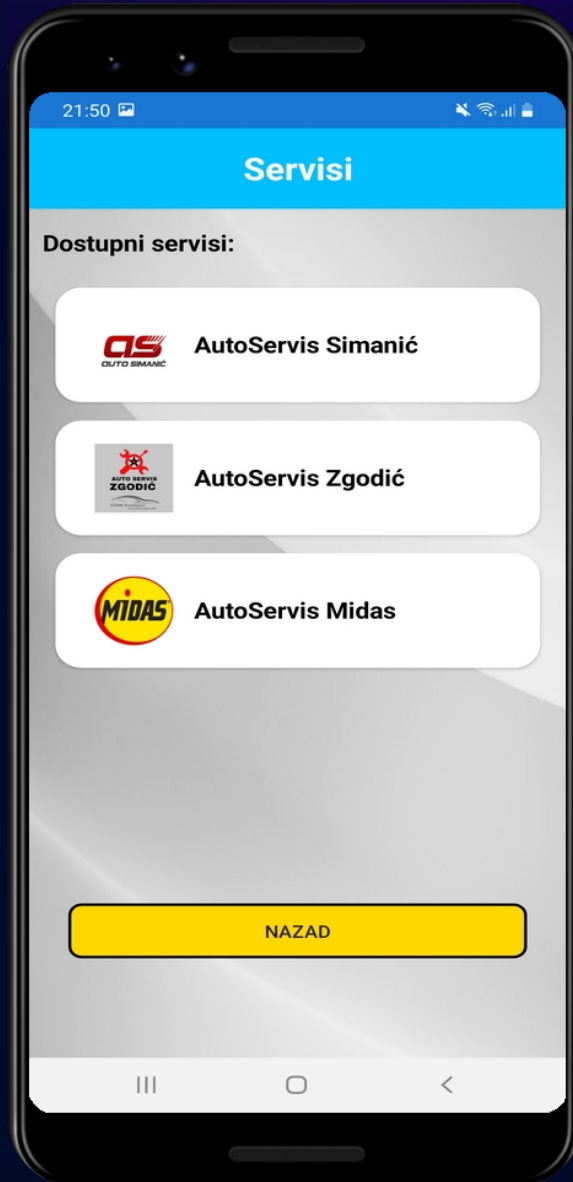
    public int RedniBroj
    {
        get { return redniBroj; }
        set
        {
            this.redniBroj = value;
        }
    }
}
```

Historija1a.xaml.cs

```
public ServisneInformacije(int redniBroj,
string opisPosla, string servis, string zamjenskidio,
int cijenaDijela, int brRadnihSati,
int cijenaUsluge, int ukupno)
{
    this.redniBroj = redniBroj;
    this.opisPosla = opisPosla;
    this.servis = servis;
    this.zamjenskidio = zamjenskidio;
    this.cijenaDijela = cijenaDijela;
    this.brRadnihSati = brRadnihSati;
    this.cijenaUsluge = cijenaUsluge;
    this.ukupno = ukupno;
}

public class Historija1A
{
    public
    ObservableCollection<ServisneInformacije>
    ServisneInfo { get; set; }
    public Historija1A()
    {
        this.ServisneInfo = new
        ObservableCollection<ServisneInformacije>();
        Dodaj();
    }
    private void Dodaj()
    {
        ServisneInfo.Add(new
        ServisneInformacije(1, "Zamjena disk pločica
(Prednje)", "Midas", "Disk pločice", 80, 1, 30,
110));
        ServisneInfo.Add(new
        ServisneInformacije(2, "Zamjena disk pločica
(Zadnje)", "Midas", "Disk pločice", 40, 1, 20, 60));
    }
}
```


Katalog servisa



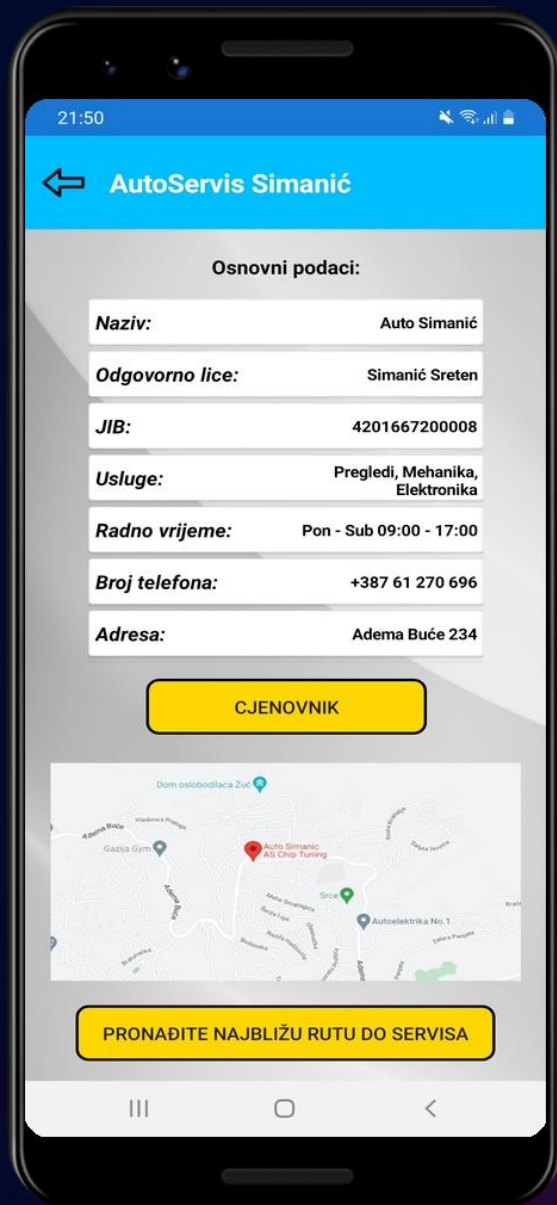
- Lista servisa
- Frame.GestureRecognizer
- Tapped

Servis.Xaml

```
<StackLayout >
    <Frame Padding="10,10,10,10" BackgroundColor="DeepSkyBlue"
    HeightRequest="50">
        <Label Text="Servisi"
        FontAttributes="Bold" VerticalOptions="Center"
        TextColor="White" FontSize="25" HorizontalTextAlignment="Center" />
    </Frame>
    <Label Padding="10,10,0,0" Text="Dostupni servisi:"
    TextColor="Black" FontSize="18" FontAttributes="Bold" />
    <Frame Margin="20,20,20,0" CornerRadius="20">
        <Grid >
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="80" />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>
            <Image Grid.Column="0" Source="simanic"
            HeightRequest="60" WidthRequest="60" />
            <Label Grid.Column="1" Text="AutoServis Simanić"
            VerticalOptions="Center" FontSize="18" TextColor="Black"
            FontAttributes="Bold" />
        </Grid>
    </Frame>
</StackLayout>
```

Servis1

(Osnovne informacije)



- Podaci o servisu
- ListView
- Cjenovnik
- Mogućnost zumiranja slike
- Lokacija

Servis1.Xaml

```
xmlns:pinch="clr-namespace:Xamarin.Forms.PinchZoomImage;assembly=Xamarin.Forms.PinchZoomImage"
<ScrollView>
    <Button x:Name="Cjenovnik" Margin="45,0,45,0"
        TranslationY="380" Text="Cjenovnik" CornerRadius="10" BackgroundColor="Gold"
        BorderColor="Black" BorderWidth="2" Clicked="CjenovnikKlik" />
    </Grid>
    <Grid x:Name="Grid2" Margin="0,-440,0,0" IsVisible="False" >
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <ListView ItemsSource="{Binding Cijene}"
            Grid.Row="0" SelectionMode="None" IsEnabled="False" Margin="10" >
            <ListView.ItemTemplate >
                <DataTemplate >
                    <ViewCell>
                        <Frame BackgroundColor="Transparent"
                            BorderColor="Transparent" Padding="0" IsEnabled="False" >
                            <Grid >
                                <Grid.ColumnDefinitions>
                                    <ColumnDefinition Width="40" />
                                    <ColumnDefinition Width="280" />
                                    <ColumnDefinition Width="130" />
                                </Grid.ColumnDefinitions>
                                <Grid Grid.Column="0" >
                                    <BoxView BackgroundColor="Black"/>
                                <Grid BackgroundColor="White"
                                    Margin="1" >
                                    <Label Grid.Column="0"
                                        Text="{Binding RBr}" TextColor="Black" FontSize="15" FontAttributes="Bold,Italic"
                                        HorizontalTextAlignment="Center" VerticalTextAlignment="Center" />
                                </Grid>
                            </Grid>
                        </ViewCell>
                    </DataTemplate>
                </ListView.ItemTemplate>
            </ListView>
        </Grid>
    </Grid>
    <StackLayout x:Name="Stack1" Margin="0,0,0,-450" TranslationY="-465">
        <pinch:PinchZoom x:Name="Zoom" TranslationY="25" >
            <pinch:PinchZoom.Content>
                <Image Margin="20" Source="simanicloc" ></Image>
            </pinch:PinchZoom.Content>
        </pinch:PinchZoom>
    </StackLayout>
</ScrollView>
```

Servis1

(Cjenovnik usluga)



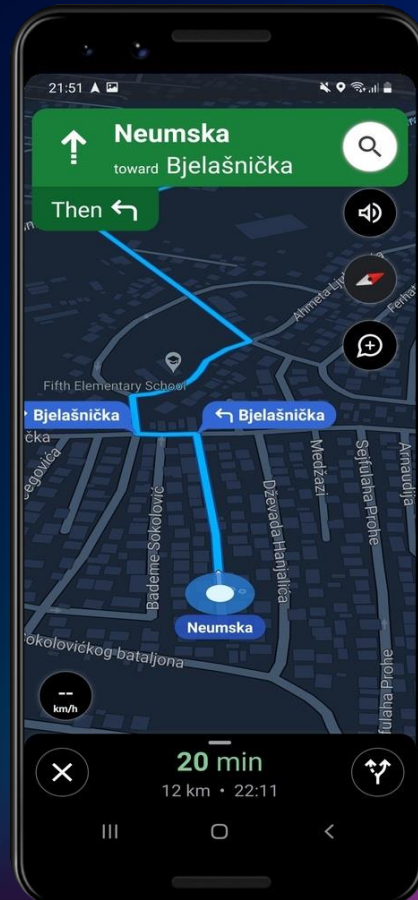
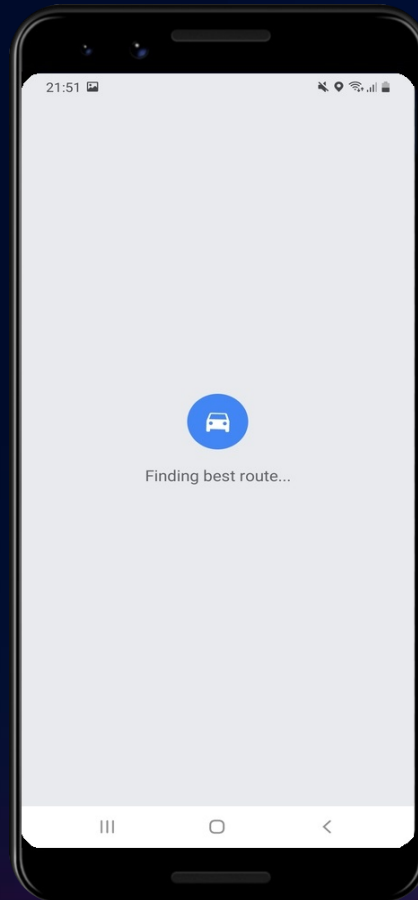
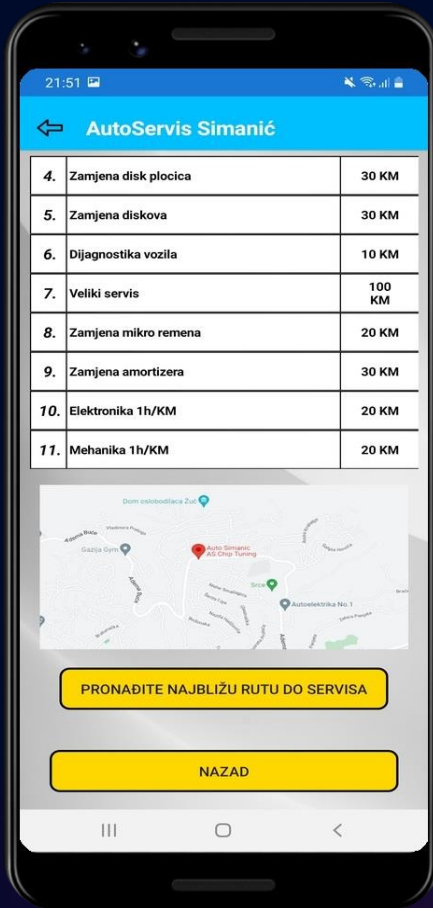
- ScrollView
- ListView
- Promjena property-a prilikom klika

Servis1.xaml.cs

```
private void CjenovnikKlik(object sender, EventArgs e)
{
    Dugme.IsVisible = true;
    Grid2.IsVisible = true;
    Stack1.Margin = new Thickness(0,0,0,-300);
    Stack1.TranslationY = -320;
}
```

Prilikom klika dolazi do promjene vrijednosti prikazanih property-a, kako bi se prikazao cjenovnik, a ispod njega slika lokacije.

Servis1 (Lokacija)



Servis1.xaml.cs

```
using Xamarin.Essentials;

private void Lokacija(object sender,
EventArgs e)
{
    Map.OpenAsync(43.86273015775319,
18.358952542164147, new MapLaunchOptions
    {
        Name = "simanic",
        NavigationMode =
NavigationMode.Driving
    });
}
```

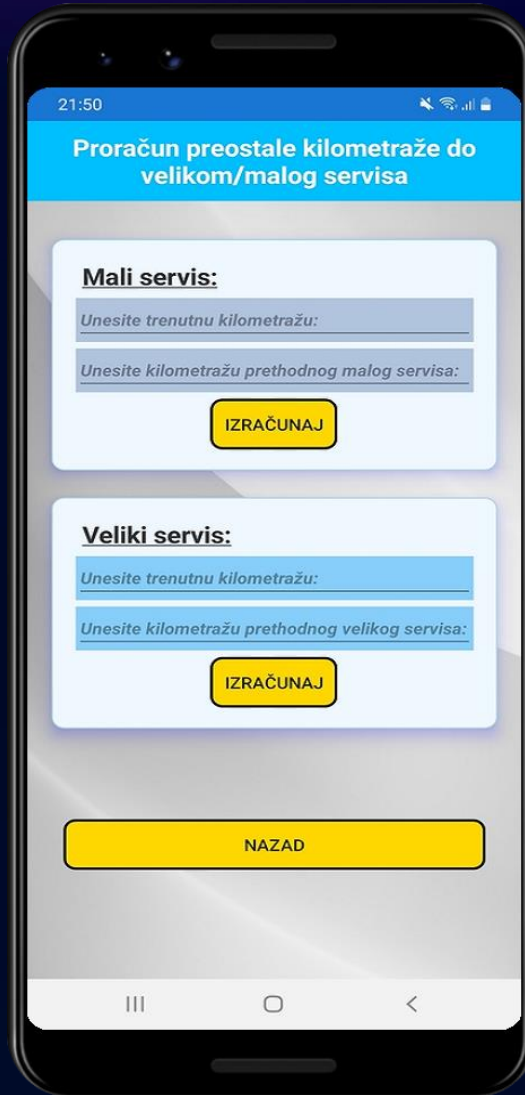
MAP

Uključena biblioteka Xamarin.Essentials

Korištenje latitude i longitude za navigaciju.

Prilikom klika na dugme „Pronađite najbližu rutu do servisa“ dolazi do otvaranja google maps i bira najkraću rutu do određenog servisa, uz način navigacije za automobil.

Kalkulator servisnih intervala

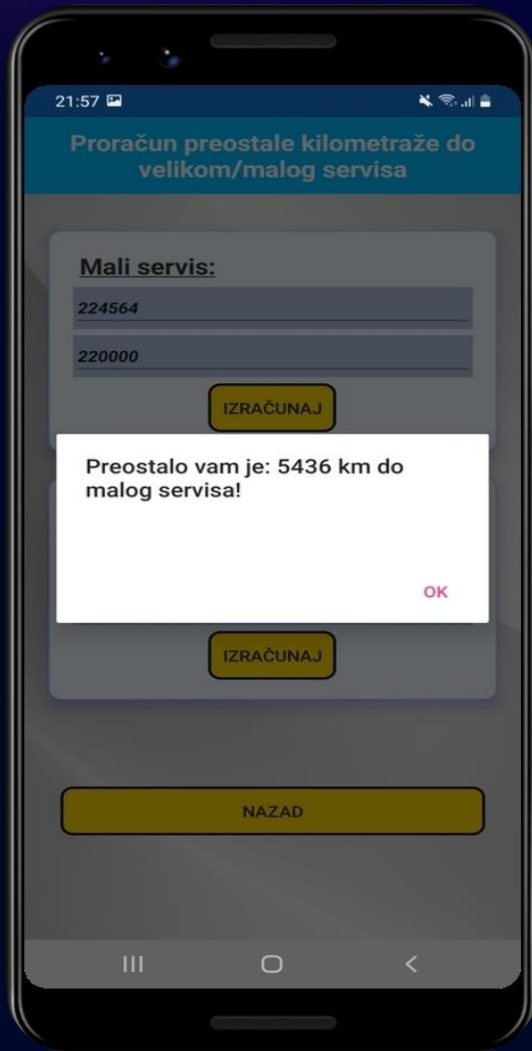


- Proračun preostale kilometraže do sljedećeg velikog/malog servisa vozila
- Polja za unos

Kalkulator.Xaml

```
<Frame HasShadow="True" BackgroundColor="AliceBlue"
CornerRadius="10" Margin="20,20,20,0"
xct:ShadowEffect.Color="Blue" BorderColor="LightBlue" >
  <StackLayout >
    <Label TranslationX="5"
BackgroundColor="#46878" TextColor="Black" Text="Veliki
servis:" TextDecorations="Underline" Opacity="0.8"
FontAttributes="Bold" FontSize="20"
HorizontalTextAlignment="Start" HorizontalOptions="Start" />
    <Entry x:Name="Velikiservis"
BackgroundColor="LightSkyBlue" Placeholder="Unesite trenutnu
kilometražu:" FontSize="14" FontAttributes="Bold,Italic" />
    <Entry x:Name="Velikiservis2"
BackgroundColor="LightSkyBlue" Placeholder="Unesite
kilometražu prethodnog velikog servisa: " FontSize="14"
FontAttributes="Bold,Italic" />
    <Label x:Name="rez2" FontSize="20"
FontAttributes="Bold" HorizontalTextAlignment="Center"
TextColor="Black" IsVisible="Hidden" />
    <Button HorizontalOptions="Center"
Text="Izračunaj" CornerRadius="10" BackgroundColor="Gold"
BorderColor="Black" BorderWidth="2" Clicked="Drugi"
BindingContext="{x:Reference Name=rez2 }" />
  </StackLayout>
</Frame>
```

Mali servis

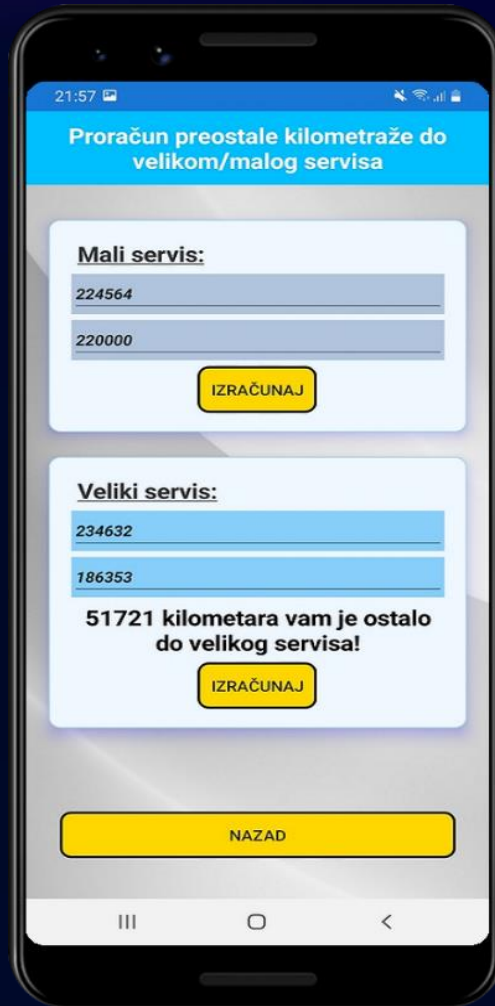


- Ispis rezultata u DisplayAlert

Kalkulator.xaml.cs

```
private async void Prvi(object sender, EventArgs e)
{
    int prva = int.Parse(Maliservis.Text);
    int druga = int.Parse(Maliservis2.Text);
    int x = 10000 - (prva - druga);
    int vrijednost = x;
    if (prva < druga)
    {
        await DisplayAlert("Neispravan unos 'Trenutna kilometraža manja od prijašnje!", "", "Ok");
    }
    else if ((prva - druga) > 10000)
    {
        int c = prva - druga - 10000;
        await DisplayAlert("Trebali ste uraditi mali servis prije " + c + " kilometara!", "", "Ok");
    }
    else
    {
        await DisplayAlert("Preostalo vam je: " + vrijednost + " km do malog servisa!", "", "Ok");
    }
}
```

Veliki servis



- Ispis rezultata u Labelu
- Promjena propertya labele prilikom pritiska na dugme

Kalkulator.xaml.cs

```
private void Drugi(object sender, EventArgs e)
{
    rez2.IsVisible = true;
    var prva = int.Parse(Velikiservis.Text);
    var druga = int.Parse(Velikiservis2.Text);
    var x = 100000 - (prva - druga);
    var vrijednost = Convert.ToString(x);

    if (prva < druga)
    {
        rez2.Text = "Neispravan unos 'Trenutna kilometraža manja od prijašnje!";
    }
    else if ((prva - druga) > 100000)
    {
        var c = prva - druga - 100000;
        rez2.Text = "Trebali ste uraditi veliki servis prije " + c + " kilometara!";
    }
    else
    {
        rez2.Text = vrijednost + " kilometara vam je ostalo do velikog servisa!";
    }
}
```

Video prezentacija

