

```
1 using Mraznicka.Helpers;
2 using Mraznicka.Models;
3 using Mraznicka.Services;
4 using System;
5 using System.Collections.Generic;
6 using System.Linq;
7 using System.Reflection;
8 using System.Text;
9 using Xamarin.Forms;
10 using System.Globalization;
11 using Plugin.Toast;
12 using Xamarin.Essentials;
13 using Xamarin.CommunityToolkit.Extensions;
14 using Plugin.NFC;
15
16 namespace Mraznicka.ViewModels
17 {
18     public class MainPageViewModel : BaseViewModel
19     {
20         ContentPage content_page;
21
22         public Command PushCommand { get; }
23         public Command PullCommand { get; }
24         public Command PreviewCommand { get; }
25         public Command SaveCommand { get; }
26
27         public string Version { get; set; }
28
29         private bool hasLicense;
30         public bool HasLicense {
31             get {
32                 return hasLicense && !LicenseValidate(null);
33             }
34             //set { SetAndNotify(ref hasLicense, value, () =>
35                 HasLicense); }
36             set { SetProperty(ref hasLicense, value); }
37         }
38
39         private string licKey;
40         public string LicKey
41         {
42             get { return licKey; }
43             set
44             {
45                 SetProperty(ref licKey, value);
46             }
47         }
48
49         public IDataStore<Models.Setting> DataStore =>
50             DependencyService.Get<IDataStore<Models.Setting>>();
51
52         public MainPageViewModel(ContentPage page)
53         {
54         }
```

```
52         content_page = page;
53         PushCommand = new Command(OnPushClicked, LicenseValidate);
54         SaveCommand = new Command(OnSaveClicked, SaveValidate);
55         PullCommand = new Command(OnPullClicked, LicenseValidate);
56         PreviewCommand = new Command(OnPreviewClicked,           ↗
            LicenseValidate);
57         Version = Assembly.GetExecutingAssembly().GetName       ↗
            ().Version.ToString();
58
59         HasLicense = !LicenseValidate(null);
60         ((App)Application.Current).Licencia = LicenseValidate  ↗
            (null);
61     }
62
63     private bool LicenseValidate(object arg)
64     {
65
66         var setting = DataStore.GetItems(false).FirstOrDefault(x => ↗
            x.Key == "LicenseKey");
67
68         var licences = new string[] {
69             "96587123",
70             "15678965",
71             "36541258",
72             "75698468",
73             "26798521",
74             "69489357",
75             "95487564",
76             "65423687",
77             "29768458",
78             "65041460"
79         };
80
81         return setting == null ? false : licences.Contains      ↗
            (setting.Val);
82
83     }
84
85     private bool SaveValidate(object arg)
86     {
87         return LicKey != "";
88     }
89
90     private void OnSaveClicked(object obj)
91     {
92         HasLicense = true;
93         DataStore.UpdateItem(new Setting() { Id = 1, Key =      ↗
            "LicenseKey", Val = LicKey });
94
95         PushCommand.ChangeCanExecute();
96         PullCommand.ChangeCanExecute();
97         PreviewCommand.ChangeCanExecute();
98         if (LicenseValidate(null))
```

```
99      {
100          Button button = content_page.FindByName<Button>
101              ("button_ulozit");
102          button.IsVisible = false;
103          Label label = content_page.FindByName<Label>
104              ("label_licencne_cislo");
105          label.IsVisible = false;
106          Entry entry = content_page.FindByName<Entry>
107              ("entry_licencne_cislo");
108          entry.IsVisible = false;
109          DMToast dt = new DMToast();
110          dt.ToastMessage
111              (Mraznicka.Resources.AppResources.registracia_prebehla_uspesne);
112      }
113      else
114      {
115          // CrossToastPopUp.Current.ShowToastError
116          (Mraznicka.Resources.AppResources.neplatny_licencny_k
117              luc, Plugin.Toast.Abstractions.ToastLength.Long);
118          DMToast dt = new DMToast();
119          dt.ToastError
120              (Mraznicka.Resources.AppResources.neplatny_licencny_k
121              luc);
122      }
123  }
124
125  private async void OnPushClicked(object obj)
126  {
127      Device.BeginInvokeOnMainThread(() =>
128      {
129          CrossNFC.Current.StopListening();
130      });
131      await Shell.Current.GoToAsync("VlozenieAllInOnePage");
132      //await Shell.Current.GoToAsync("PushPage");
133      //await Shell.Current.GoToAsync("ExpressPush");
134  }
135
136  private async void OnPullClicked(object obj)
137  {
138      Device.BeginInvokeOnMainThread(() =>
139      {
140          CrossNFC.Current.StopListening();
141      });
142      await Shell.Current.GoToAsync("PullPage");
143      //await Shell.Current.GoToAsync("ExpressPull");
144  }
145
146  private async void OnPreviewClicked(object obj)
147  {
148  }
```

```
143      // Prefixing with `//` switches to a different navigation  ↗
144      stack instead of pushing to the active one
145      //await Shell.Current.GoToAsync($"://{nameof(AboutPage)}");
146      Device.BeginInvokeOnMainThread(() =>
147      {
148          CrossNFC.Current.StopListening();
149      });
150      await Shell.Current.GoToAsync("PreviewPage");
151  }
152
153  }
154 }
155
```