

Практическая работа №14

Xamarin.Forms Shell

Цель работы: получить навыки написания мобильных приложений с использованием Xamarin.Forms Shell.

Теоретическая часть

Необходимо установить в Visual Studio пакеты для разработки приложений Xamarin.

Шаг 1

Создайте новый проект типа Mobile App (Xamarin.Forms), задайте проекту имя TextNotesApp. Тип проекта – Tabbed.

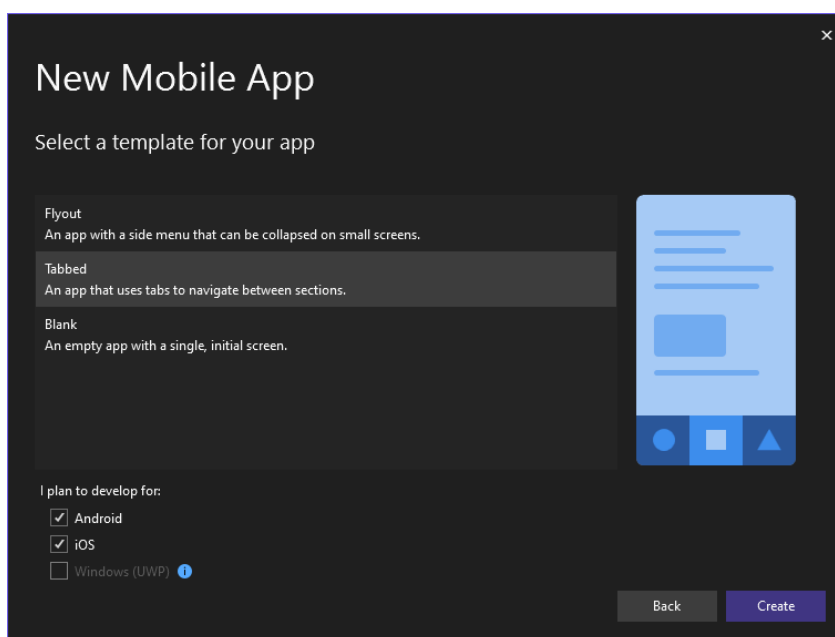


Рисунок 1 – Выбор типа проекта

Шаг 2

Нужно добавить android эмулятор. Можно выбрать настройки по умолчанию. Добавьте эмулятор или подключите телефон для отладки.

Шаг 3

Удалите лишние файлы и папки из проекта, а именно:

- Models
- Services
- ViewModels
- Views
- GettingStarted.txt

Шаг 4

Добавьте в проект новую папку с именем Views. Добавьте в созданную папку новый элемент типа Content Page (XAML) с именем TextNotesPage.

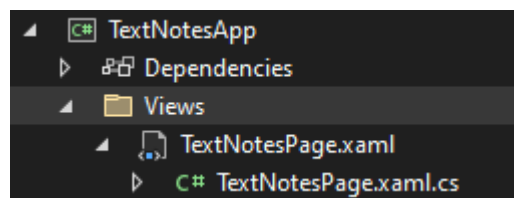


Рисунок 2 – Созданная страница

Шаг 5

Добавьте следующий код в файл TextNotesPage.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="TextNotesApp.Views.TextNotesPage"
              Title="Text Notes">
    <!-- Вертикальное расположение элементов -->
    <StackLayout Margin="20">
        <Editor x:Name="editor"
                Placeholder="Enter your note"
                HeightRequest="100" />
        <!-- Расположение элементов в две колонки -->
        <Grid ColumnDefinitions="*,*">
            <Button Text="Save"
                    Clicked="OnSaveButtonClicked" />
            <Button Grid.Column="1"
                    Text="Delete"
                    Clicked="OnDeleteButtonClicked"/>
        </Grid>
    </StackLayout>
</ContentPage>
```

Здесь мы добавляем на страницу редактор Editor и две кнопки для работы с файлом.

Шаг 6

Объявите в файле TextNotesPage.xaml.cs поле для хранения пути к файлу, в котором будет храниться содержимое заметки.

```
//Путь к файлу для хранения заметки
string _fileName =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "textnotes.txt");
```

При открытии страницы будем считывать текст из файла.

```
public TextNotesPage()
{
    InitializeComponent();

    //Считывание текста из существующего файла
    if (File.Exists(_fileName))
    {
        editor.Text = File.ReadAllText(_fileName);
    }
}
```

Добавим обработчики для кнопок работы с файлом.

```
void OnSaveButtonClicked(object sender, EventArgs e)
{
    File.WriteAllText(_fileName, editor.Text);
}

void OnDeleteButtonClicked(object sender, EventArgs e)
{
    if (File.Exists(_fileName))
    {
        File.Delete(_fileName);
    }
    editor.Text = string.Empty;
}
```

Шаг 7

Добавьте ещё одну страницу с именем AboutPage. Здесь будет размещаться информация о вас.

```
<?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="TextNotesApp.Views.AboutPage"
    Title="About">
    <!-- Расположим элементы в два столбца -->
    <Grid RowDefinitions="Auto,*">
        <Image Source="xamarin_logo.png"
            BackgroundColor="{OnPlatform iOS=LightSlateGray, Android=#2196F3}"
            VerticalOptions="Center"
            HeightRequest="64" />
        <!-- Расположим элементы вертикально -->
        <StackLayout Grid.Row="1"
            Margin="20"
```

```

        Spacing="20">
    <Label FontSize="22">
        <Label.FormattedText>
            <FormattedString>
                <FormattedString.Spans>
                    <Span Text="TextNotes"
                        FontAttributes="Bold"
                        FontSize="22" />
                    <Span Text=" v1.0" />
                </FormattedString.Spans>
            </FormattedString>
        </Label.FormattedText>
    </Label>
    <Label Text="I am the best Xamarin developer." />
    <Button Text="Learn more"
        Clicked="OnButtonClicked" />
</StackLayout>
</Grid>
</ContentPage>

```

Шаг 8

Добавим обработчик для кнопки. При нажатии на неё будет открываться ссылка на ваш профиль на GitHub.

```

async void OnButtonClicked(object sender, EventArgs e)
{
    await Launcher.OpenAsync("https://github.com/");
}

```

Шаг 9

Откройте файл AppShell.xaml. Этот код отвечает за определение визуальной иерархии приложения. На его основе Shell создает UI для содержимого приложения.

Измените строку внутри тега Shell:

```
xmlns:views="clr-namespace:TextNotesApp.Views"
```

Уберите тег `<Shell.Resources>` вместе с содержимым и измените содержимое тега `<TabBar>`

```

<TabBar>
    <ShellContent Title="Text Notes"
        Icon="icon_feed.png"
        ContentTemplate="{DataTemplate views:TextNotesPage}" />
    <ShellContent Title="About"
        Icon="icon_about.png"
        ContentTemplate="{DataTemplate views:AboutPage}" />
</TabBar>

```

Далее откройте файл AppShell.xaml.cs и уберите из конструктора весь код, кроме метода `InitializeComponent()`

Шаг 10

Откройте файл App.xaml. Здесь определяется класс App для создания экземпляра приложения.

Удалите тег `<Application.Resources>` вместе с содержимым.

Откройте файл App.xaml.cs и удалите из конструктора код:

```
DependencyService.Register<MockDataStore>();
```

Шаг 11

Предыдущий код позволяет работать с одной заметкой. Для удобства добавим возможность работать с несколькими заметками.

Добавьте в проект папку Models. Создайте в этой папке класс TextNote. Опишите в классе свойства для хранения информации в заметке, а именно путь к файлу, текст заметки и дату.

```
public string Filename { get; set; }  
public string Text { get; set; }  
public DateTime Date { get; set; }
```

Шаг 12

Добавьте новую страницу с именем TextNoteEntryPage. На этой странице будем работать с заметкой. Вместо тега `<ContentPage.Content>` добавьте код:

```
<StackLayout Margin="20">  
  <Editor Placeholder="Enter your text note"  
    Text="{Binding Text}"  
    HeightRequest="100" />  
  <Grid ColumnDefinitions="*,*">  
    <Button Text="Save"  
      Clicked="OnSaveButtonClicked" />  
    <Button Grid.Column="1"  
      Text="Delete"  
      Clicked="OnDeleteButtonClicked"/>  
  </Grid>  
</StackLayout>
```

Здесь мы привязали данные из Editor к свойству Text класса TextNote.

Шаг 13

Добавим в класс App свойство FolderPath и код для его инициализации в конструкторе:

```
public static string FolderPath { get; private set; }

FolderPath = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.
LocalApplicationData));
```

Шаг 14

Откройте файл TextNoteEntryPage.xaml.cs и добавьте код, который позволит передавать данные на страницу с помощью параметров запроса. Здесь указывается имя свойства и идентификатор параметра запроса.

```
[QueryProperty(nameof(ItemId), nameof(ItemId))]
```

Свойство ItemId будет получать данные, переданные в параметр запроса ItemId из URI, указанного в вызове метода GoToAsync.

Добавьте свойство:

```
public string ItemId
{
    set { LoadNote(value); }
}
```

Оно вызывает метод LoadNote для создания объекта TextNote из файла на устройстве и задает для BindingContext страницы объект TextNote.

В конструктор добавьте код:

```
BindingContext = new TextNote();
```

Здесь мы сохраняем заметку в BindingContext страницы.

Добавим метод для загрузки заметки:

```

void LoadNote(string filename)
{
    try
    {
        TextNote note = new TextNote
        {
            Filename = filename,
            Text = File.ReadAllText(filename),
            Date = File.GetCreationTime(filename)
        };
        BindingContext = note;
    }
    catch (Exception)
    {
        Console.WriteLine("Can't load note");
    }
}

```

Добавим обработчик для кнопки «Сохранить».

```

async void OnSaveButtonClicked(object sender, EventArgs e)
{
    var note = (TextNote)BindingContext;

    if (string.IsNullOrEmpty(note.Filename))
    {
        var filename = Path.Combine(App.FolderPath,
            $"{Path.GetRandomFileName()}.notes.txt");
        File.WriteAllText(filename, note.Text);
    }
    else
    {
        File.WriteAllText(note.Filename, note.Text);
    }

    await Shell.Current.GoToAsync("..");
}

```

И обработчик для кнопки «Удалить».

```

async void OnDeleteButtonClicked(object sender, EventArgs e)
{
    var note = (TextNote)BindingContext;

    if (File.Exists(note.Filename))
    {
        File.Delete(note.Filename);
    }

    await Shell.Current.GoToAsync("..");
}

```

В обоих обработчиках после выполнения действия приложение возвращается на предыдущую страницу.

Шаг 15

Теперь изменим код в файле TextNotesPage.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="TextNotesApp.Views.TextNotesPage"
              Title="Text Notes">
    <ContentPage.ToolbarItems>
        <ToolBarItem Text="Add"
                      Clicked="OnAddClicked" />
    </ContentPage.ToolbarItems>

    <CollectionView x:Name="collectionView"
                   Margin="20"
                   SelectionMode="Single"
                   SelectionChanged="OnSelectionChanged">
        <CollectionView.ItemsLayout>
            <LinearItemsLayout Orientation="Vertical"
                               ItemSpacing="10" />
        </CollectionView.ItemsLayout>

        <CollectionView.ItemTemplate>
            <DataTemplate>
                <StackLayout>
                    <Label Text="{Binding Text}"
                           FontSize="Medium" />
                    <Label Text="{Binding Date}"
                           TextColor="Silver"
                           FontSize="Small" />
                </StackLayout>
            </DataTemplate>
        </CollectionView.ItemTemplate>
    </CollectionView>
</ContentPage>

```

CollectionView используется для отображения всех заметок приложения. При выборе пользователем заметки открывается страница с содержимым заметки. Для создания заметки используем кнопку ToolbarItem.

Шаг 16

Далее изменим код файла TextNotesPage.xaml.cs:

```

public TextNotesPage()
{
    InitializeComponent();
}

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

    var notes = new List<TextNote>();

    var files = Directory.EnumerateFiles(App.FolderPath, "*.notes.txt");
    foreach (var filename in files)
    {
        notes.Add(new TextNote
        {
            Filename = filename,
            Text = File.ReadAllText(filename),
            Date = File.GetCreationTime(filename)
        });
    }
}

```



```

    }

    collectionView.ItemsSource = notes
        .OrderBy(d => d.Date)
        .ToList();
}

```

Этот метод при открытии страницы получает все заметки из локальной папки приложения.

Добавим обработчик для добавления новой заметки.

```

async void OnAddClicked(object sender, EventArgs e)
{
    await Shell.Current.GoToAsync(nameof(TextNoteEntryPage));
}

```

Добавим обработчик для перехода к заметке.

```

async void OnSelectionChanged(object sender, SelectionChangedEventArgs e)
{
    if (e.CurrentSelection != null)
    {
        TextNote note = (TextNote)e.CurrentSelection.FirstOrDefault();
        await
Shell.Current.GoToAsync($"{nameof(TextNoteEntryPage)}?{nameof(TextNoteEntryPage.Item
Id)}={note.Filename}");
    }
}

```

Шаг 17

Теперь нужно поменять код в файле AppShell.xaml.cs. Добавьте в конструктор метод для добавления маршрута новой страницы, которую не представили в файле AppShell.xaml:

```

Routing.RegisterRoute(nameof(TextNoteEntryPage), typeof(TextNoteEntryPage));

```

Шаг 18

Запустите приложение.

Практическая часть

Добавьте в разработанное приложение возможность привязывать к заметке изображение.