| Модули | Описание | Количество строк кода | Размер (в Кбайтах) |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| .gitignore |  | 7 | 1 |
| App.xaml |  | 14 | 1 |
| App.xaml.cs |  | 27 | 1 |
| AppShell.xaml |  | 41 | 2 |
| AppShell.xaml.cs |  | 68 | 2 |
| Converters\BoolToIntConverter.cs |  | 17 | 1 |
| Converters\BooleanNegationConverter.cs |  | 19 | 1 |
| Converters\EnemyFilterConverter.cs |  | 22 | 1 |
| Converters\HitPointsConverter.cs |  | 22 | 1 |
| Converters\NpcFilterConverter.cs |  | 22 | 1 |
| Converters\ParticipantTypeToCollectionConverter.cs |  | 35 | 2 |
| DnDClient.csproj |  | 110 | 6 |
| DnDClient.csproj.user |  | 11 | 1 |
| DnDClient.sln |  | 16 | 1 |
| DnDClient.sln.DotSettings.user |  | 9 | 4 |
| DnDClient\_TemporaryKey.pfx |  | 11 | 3 |
| MauiProgram.cs |  | 46 | 2 |
| Models\Attack.cs |  | 34 | 2 |
| Models\Campaign.cs |  | 33 | 1 |
| Models\Combat.cs |  | 42 | 2 |
| Models\CombatLog.cs |  | 19 | 1 |
| Models\CombatParticipant.cs |  | 44 | 2 |
| Models\Enemy.cs |  | 83 | 3 |
| Models\Log.cs |  | 20 | 1 |
| Models\NPC.cs |  | 31 | 2 |
| Models\PlayerCharacter.cs |  | 342 | 11 |
| Models\SpecialAbility.cs |  | 26 | 1 |
| Models\StoryElement.cs |  | 26 | 1 |
| Models\User.cs |  | 28 | 1 |
| Platforms\Android\AndroidManifest.xml |  | 16 | 1 |
| Platforms\Android\MainActivity.cs |  | 11 | 1 |
| Platforms\Android\MainApplication.cs |  | 14 | 1 |
| Platforms\Android\Resources\values\colors.xml |  | 5 | 1 |
| Platforms\MacCatalyst\AppDelegate.cs |  | 8 | 1 |
| Platforms\MacCatalyst\Entitlements.plist |  | 14 | 1 |
| Platforms\MacCatalyst\Info.plist |  | 38 | 2 |
| Platforms\MacCatalyst\Program.cs |  | 14 | 1 |
| Platforms\Tizen\Main.cs |  | 15 | 1 |
| Platforms\Tizen\tizen-manifest.xml |  | 14 | 1 |
| Platforms\Windows\App.xaml |  | 8 | 1 |
| Platforms\Windows\App.xaml.cs |  | 22 | 1 |
| Platforms\Windows\Package.appxmanifest |  | 46 | 2 |
| Platforms\Windows\app.manifest |  | 15 | 1 |
| Platforms\iOS\AppDelegate.cs |  | 8 | 1 |
| Platforms\iOS\Info.plist |  | 32 | 2 |
| Platforms\iOS\Program.cs |  | 14 | 1 |
| Platforms\iOS\Resources\PrivacyInfo.xcprivacy |  | 51 | 2 |
| Properties\PublishProfiles\MSIX-win-x64.pubxml |  | 15 | 1 |
| Properties\launchSettings.json |  | 7 | 1 |
| Resources\Raw\AboutAssets.txt |  | 15 | 1 |
| Resources\Styles\Colors.xaml |  | 44 | 3 |
| Resources\Styles\Styles.xaml |  | 427 | 25 |
| Services\ApiHelper.cs |  | 170 | 6 |
| Services\Serdeser.cs |  | 15 | 1 |
| ViewModels\AuthViewModel.cs |  | 81 | 3 |
| ViewModels\CampaignListViewModel.cs |  | 77 | 3 |
| ViewModels\CampaignViewModel.cs |  | 160 | 6 |
| ViewModels\CharDetailsViewModel.cs |  | 50 | 2 |
| ViewModels\CharactersViewModel.cs |  | 98 | 4 |
| ViewModels\CombatParticipantsViewModel.cs |  | 390 | 13 |
| ViewModels\CombatViewModel.cs |  | 191 | 8 |
| ViewModels\CreateEnemyViewModel.cs |  | 144 | 5 |
| ViewModels\CreateNPCViewModel.cs |  | 137 | 5 |
| ViewModels\EntityManagementViewModel.cs |  | 280 | 9 |
| ViewModels\MainViewModel.cs |  | 16 | 1 |
| ViewModels\ProfileViewModel.cs |  | 79 | 3 |
| Views\AuthPage.xaml |  | 14 | 1 |
| Views\AuthPage.xaml.cs |  | 16 | 1 |
| Views\CampaignListPage.xaml |  | 19 | 2 |
| Views\CampaignListPage.xaml.cs |  | 16 | 1 |
| Views\CampaignPage.xaml |  | 142 | 9 |
| Views\CampaignPage.xaml.cs |  | 17 | 1 |
| Views\Cards\CampCard.xaml |  | 29 | 2 |
| Views\Cards\CampCard.xaml.cs |  | 14 | 1 |
| Views\Cards\CharCard.xaml |  | 172 | 9 |
| Views\Cards\CharCard.xaml.cs |  | 16 | 1 |
| Views\Cards\CombatCard.xaml |  | 24 | 2 |
| Views\Cards\CombatCard.xaml.cs |  | 21 | 1 |
| Views\Cards\NPCCard.xaml |  | 6 | 1 |
| Views\Cards\NPCCard.xaml.cs |  | 14 | 1 |
| Views\Cards\StoryCard.xaml |  | 13 | 1 |
| Views\Cards\StoryCard.xaml.cs |  | 29 | 1 |
| Views\CharacterDetailsPage.xaml |  | 521 | 49 |
| Views\CharacterDetailsPage.xaml.cs |  | 17 | 1 |
| Views\CharactersPage.xaml |  | 19 | 1 |
| Views\CharactersPage.xaml.cs |  | 16 | 1 |
| Views\CombatPage.xaml |  | 140 | 9 |
| Views\CombatPage.xaml.cs |  | 12 | 1 |
| Views\CombatParticipantsPage.xaml |  | 240 | 17 |
| Views\CombatParticipantsPage.xaml.cs |  | 17 | 1 |
| Views\CreateEnemyPage.xaml |  | 118 | 8 |
| Views\CreateEnemyPage.xaml.cs |  | 12 | 1 |
| Views\CreateNPCPage.xaml |  | 92 | 6 |
| Views\CreateNPCPage.xaml.cs |  | 12 | 1 |
| Views\EntityManagementPage.xaml |  | 157 | 10 |
| Views\EntityManagementPage.xaml.cs |  | 11 | 1 |
| Views\MainPage.xaml |  | 14 | 1 |
| Views\MainPage.xaml.cs |  | 13 | 1 |
| Views\ProfilePage.xaml |  | 20 | 2 |
| Views\ProfilePage.xaml.cs |  | 8 | 1 |

1. .gitignore

bin/

obj/

/packages/

riderModule.iml

/\_ReSharper.Caches/

.idea

.vs

publish/

2. App.xaml

﻿<?xml version = "1.0" encoding = "UTF-8" ?>

<Application xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:local="clr-namespace:DnDClient"

x:Class="DnDClient.App">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Resources/Styles/Colors.xaml" />

<ResourceDictionary Source="Resources/Styles/Styles.xaml" />

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

3. App.xaml.cs

﻿using DnDClient.Views;

namespace DnDClient;

public partial class App : Application

{

public App()

{

InitializeComponent();

Routing.RegisterRoute("CreateNPCPage", typeof(CreateNPCPage));

Routing.RegisterRoute("CreateEnemyPage", typeof(CreateEnemyPage));

MainPage = new AppShell();

}

protected override async void OnStart()

{

base.OnStart();

var token = await SecureStorage.GetAsync("auth\_token");

if (string.IsNullOrEmpty(token))

{

await Shell.Current.GoToAsync("//AuthPage");

}

else

{

await Shell.Current.GoToAsync("//MainPage");

}

}

}

4. AppShell.xaml

<?xml version="1.0" encoding="UTF-8"?>

<Shell

x:Class="DnDClient.AppShell"

xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:views="clr-namespace:DnDClient.Views"

Shell.FlyoutBehavior="Disabled"

Title="DnDClient">

<ShellContent

Title="Вход"

ContentTemplate="{DataTemplate views:AuthPage}"

Route="AuthPage"

FlyoutItemIsVisible="False" />

<ShellContent

Title="Главная"

ContentTemplate="{DataTemplate views:MainPage}"

Route="MainPage" />

<ShellContent

Title="Профиль"

ContentTemplate="{DataTemplate views:ProfilePage}"

Route="ProfilePage" />

<ShellContent

Title="Персонажи"

ContentTemplate="{DataTemplate views:CharactersPage}"

Route="CharactersPage" />

<ShellContent

Title="Кампании"

ContentTemplate="{DataTemplate views:CampaignListPage}"

Route="CampaignPage" />

<MenuItem

Text="Выход"

IconImageSource="logout.png"

Clicked="OnLogoutClicked"

IsDestructive="True" />

</Shell>

5. AppShell.xaml.cs

﻿namespace DnDClient;

public partial class AppShell : Shell

{

public AppShell()

{

InitializeComponent();

CheckAuthStateAsync();

}

private async void CheckAuthStateAsync()

{

var token = await SecureStorage.GetAsync("auth\_token");

if (!string.IsNullOrEmpty(token))

{

SetupAuthenticatedShell();

}

}

public void SetupAuthenticatedShell()

{

FlyoutBehavior = FlyoutBehavior.Flyout;

GoToAsync("//MainPage");

}

private async void OnLogoutClicked(object sender, EventArgs e)

{

try

{

SecureStorage.Remove("auth\_token");

ClearGlobalData();

ClearNavigationStack();

ForceGarbageCollection();

RestartApplication();

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при выходе: {ex.Message}");

}

}

private void ClearGlobalData()

{

Preferences.Clear();

SecureStorage.RemoveAll();

}

private void ClearNavigationStack()

{

foreach (var page in Navigation.NavigationStack.ToList())

{

Navigation.RemovePage(page);

}

}

private void ForceGarbageCollection()

{

GC.Collect();

GC.WaitForPendingFinalizers();

}

private void RestartApplication()

{

Application.Current.MainPage = new AppShell();

}

}

6. Converters\BoolToIntConverter.cs

﻿using System.Globalization;

namespace DnDClient.Converters;

public class BoolToIntConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

int intValue = (int)value;

int checkIndex = int.Parse(parameter.ToString());

return intValue >= checkIndex;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

7. Converters\BooleanNegationConverter.cs

﻿using System.Globalization;

namespace DnDClient.Converters;

public class BooleanNegationConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is bool b)

return !b;

return value;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is bool b)

return !b;

return value;

}

}

8. Converters\EnemyFilterConverter.cs

﻿using System.Globalization;

using DnDClient.Models;

namespace DnDClient.Converters

{

public class EnemyFilterConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is IEnumerable<CombatParticipant> participants)

{

return participants.Where(p => p.Type == ParticipantType.Enemy).ToList();

}

return null;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

}

9. Converters\HitPointsConverter.cs

﻿using System;

using System.Globalization;

using DnDClient.Models;

using Microsoft.Maui.Controls;

namespace DnDClient.Views;

public class HitPointsConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is CombatParticipant participant)

{

return $"{participant.CurrentHitPoints}/{participant.MaxHitPoints}";

}

return "0/0";

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

10. Converters\NpcFilterConverter.cs

﻿using System.Globalization;

using DnDClient.Models;

namespace DnDClient.Converters

{

public class NpcFilterConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is IEnumerable<CombatParticipant> participants)

{

return participants.Where(p => p.Type == ParticipantType.Npc).ToList();

}

return null;

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

}

11. Converters\ParticipantTypeToCollectionConverter.cs

﻿using DnDClient.ViewModels;

using System;

using System.Collections;

using System.Collections.ObjectModel;

using System.Globalization;

using DnDClient.Models;

using Microsoft.Maui.Controls;

namespace DnDClient.Views;

public class ParticipantTypeToCollectionConverter : IValueConverter

{

public object Convert(object value, Type targetType, object parameter, CultureInfo culture)

{

if (value is ParticipantType participantType && parameter is CombatParticipantsViewModel viewModel)

{

switch (participantType)

{

case ParticipantType.Player:

return viewModel.AvailableCharacters ?? new ObservableCollection<PlayerCharacter>();

case ParticipantType.Npc:

return viewModel.AvailableNPCs ?? new ObservableCollection<NPC>();

case ParticipantType.Enemy:

return viewModel.AvailableEnemies ?? new ObservableCollection<Enemy>();

default:

return new ObservableCollection<object>();

}

}

return new ObservableCollection<object>();

}

public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)

{

throw new NotImplementedException();

}

}

12. DnDClient.csproj

﻿<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<TargetFrameworks>net8.0-android;net8.0-windows10.0.19041.0</TargetFrameworks>

<!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->

<!-- <TargetFrameworks>$(TargetFrameworks);net8.0-tizen</TargetFrameworks> -->

<!-- Note for MacCatalyst:

The default runtime is maccatalyst-x64, except in Release config, in which case the default is maccatalyst-x64;maccatalyst-arm64.

When specifying both architectures, use the plural <RuntimeIdentifiers> instead of the singular <RuntimeIdentifier>.

The Mac App Store will NOT accept apps with ONLY maccatalyst-arm64 indicated;

either BOTH runtimes must be indicated or ONLY macatalyst-x64. -->

<!-- For example: <RuntimeIdentifiers>maccatalyst-x64;maccatalyst-arm64</RuntimeIdentifiers> -->

<OutputType>Exe</OutputType>

<RootNamespace>DnDClient</RootNamespace>

<UseMaui>true</UseMaui>

<SingleProject>true</SingleProject>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

<!-- Display name -->

<ApplicationTitle>DnDClient</ApplicationTitle>

<!-- App Identifier -->

<ApplicationId>com.strhzy.dndclient</ApplicationId>

<!-- Versions -->

<ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>

<ApplicationVersion>1</ApplicationVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'android'">30.0</SupportedOSPlatformVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'windows'">10.0.17763.0</SupportedOSPlatformVersion>

<TargetPlatformMinVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'windows'">10.0.17763.0</TargetPlatformMinVersion>

<SignAssembly>true</SignAssembly>

<GenerateAppInstallerFile>False</GenerateAppInstallerFile>

<AppxPackageSigningEnabled>True</AppxPackageSigningEnabled>

<PackageCertificateThumbprint>9CA2B8DF9813B45C4846CBEAFADFC3EDC4D940E3</PackageCertificateThumbprint>

<AppxPackageSigningTimestampDigestAlgorithm>SHA256</AppxPackageSigningTimestampDigestAlgorithm>

<AppxAutoIncrementPackageRevision>True</AppxAutoIncrementPackageRevision>

<AppxSymbolPackageEnabled>False</AppxSymbolPackageEnabled>

<GenerateTestArtifacts>True</GenerateTestArtifacts>

<HoursBetweenUpdateChecks>0</HoursBetweenUpdateChecks>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)'=='Release'">

<MtouchLink>None</MtouchLink> <!-- iOS/MacCatalyst -->

<AndroidLinkMode>None</AndroidLinkMode> <!-- Android -->

</PropertyGroup>

<ItemGroup>

<!-- App Icon -->

<MauiIcon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.svg" Color="#512BD4" />

<!-- Splash Screen -->

<MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

<!-- Images -->

<MauiImage Include="Resources\Images\\*" />

<!-- Custom Fonts -->

<MauiFont Include="Resources\Fonts\\*" />

<!-- Raw Assets (also remove the "Resources\Raw" prefix) -->

<MauiAsset Include="Resources\Raw\\*\*" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />

</ItemGroup>

<ItemGroup>

<PackageReference Include="BCrypt.Net-Next" Version="4.0.3" />

<PackageReference Include="CommunityToolkit.Maui" Version="9.1.1"/>

<PackageReference Include="CommunityToolkit.Mvvm" Version="8.4.0" />

<PackageReference Include="Microsoft.AspNetCore.SignalR.Client" Version="8.0.20" />

<PackageReference Include="Microsoft.AspNetCore.SignalR.Common" Version="8.0.20" />

<PackageReference Include="Microsoft.Maui.Controls" Version="$(MauiVersion)" />

<PackageReference Include="Microsoft.Maui.Controls.Compatibility" Version="$(MauiVersion)" />

<PackageReference Include="Microsoft.Extensions.Logging.Debug" Version="8.0.1" />

<PackageReference Include="Newtonsoft.Json" Version="13.0.3" />

<PackageReference Include="Microsoft.Extensions.Logging.Console" Version="8.0.1" />

</ItemGroup>

<ItemGroup>

<MauiXaml Update="Views\MainPage.xaml">

<SubType>Designer</SubType>

</MauiXaml>

</ItemGroup>

<ItemGroup>

<Compile Update="Views\MainPage.xaml.cs">

<DependentUpon>MainPage.xaml</DependentUpon>

<SubType>Code</SubType>

</Compile>

</ItemGroup>

<ItemGroup>

<TrimmerRootAssembly Include="BCrypt.Net-Next" Version="4.0.3" />

<TrimmerRootAssembly Include="CommunityToolkit.Mvvm" Version="8.4.0" />

<TrimmerRootAssembly Include="Microsoft.AspNetCore.SignalR.Client" Version="8.0.20" />

<TrimmerRootAssembly Include="Microsoft.AspNetCore.SignalR.Common" Version="8.0.20" />

<TrimmerRootAssembly Include="Microsoft.Maui.Controls" Version="$(MauiVersion)" />

<TrimmerRootAssembly Include="Microsoft.Maui.Controls.Compatibility" Version="$(MauiVersion)" />

<TrimmerRootAssembly Include="Microsoft.Extensions.Logging.Debug" Version="8.0.1" />

<TrimmerRootAssembly Include="Newtonsoft.Json" Version="13.0.3" />

<TrimmerRootAssembly Include="Microsoft.Extensions.Logging.Console" Version="8.0.1" />

</ItemGroup>

<ItemGroup>

<Folder Include="Resources\Images\" />

</ItemGroup>

</Project>

13. DnDClient.csproj.user

﻿<?xml version="1.0" encoding="utf-8"?>

<Project ToolsVersion="Current" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<PropertyGroup>

<IsFirstTimeProjectOpen>False</IsFirstTimeProjectOpen>

<ActiveDebugFramework>net8.0-windows10.0.19041.0</ActiveDebugFramework>

<ActiveDebugProfile>Windows Machine</ActiveDebugProfile>

<UapAppxPackageBuildMode>SideloadOnly</UapAppxPackageBuildMode>

<AppxShowAllApps>False</AppxShowAllApps>

<\_LastSelectedProfileId>MSIX-win-x64.pubxml</\_LastSelectedProfileId>

<PackageOptionalProjectsInIdeBuilds>True</PackageOptionalProjectsInIdeBuilds>

</PropertyGroup>

</Project>

14. DnDClient.sln

﻿

Microsoft Visual Studio Solution File, Format Version 12.00

Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "DnDClient", "DnDClient.csproj", "{24D8CE6A-4799-4336-B854-8AC2994840AD}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{24D8CE6A-4799-4336-B854-8AC2994840AD}.Debug|Any CPU.ActiveCfg = Debug|Any CPU

{24D8CE6A-4799-4336-B854-8AC2994840AD}.Debug|Any CPU.Build.0 = Debug|Any CPU

{24D8CE6A-4799-4336-B854-8AC2994840AD}.Release|Any CPU.ActiveCfg = Release|Any CPU

{24D8CE6A-4799-4336-B854-8AC2994840AD}.Release|Any CPU.Build.0 = Release|Any CPU

EndGlobalSection

EndGlobal

15. DnDClient.sln.DotSettings.user

﻿<wpf:ResourceDictionary xml:space="preserve" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:s="clr-namespace:System;assembly=mscorlib" xmlns:ss="urn:shemas-jetbrains-com:settings-storage-xaml" xmlns:wpf="http://schemas.microsoft.com/winfx/2006/xaml/presentation">

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003AMethodBaseInvoker\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FSourcesCache\_003Fd882146b4f265f10bcbec2663fce248db9ffec5fa1aeaf76e32a11ba5eafcd6\_003FMethodBaseInvoker\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003AMonitor\_002ECoreCLR\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E2\_003Fresharper\_002Dhost\_003FSourcesCache\_003F36f346b6c0454bc8a6afa7aed38119fe5bbffcc983298d9bfa4dbd5f49461f\_003FMonitor\_002ECoreCLR\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003AMonitor\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FDecompilerCache\_003Fdecompiler\_003F2832e8c2b81f4641b3863f406ce3a519c90938\_003F61\_003Fe27df3d4\_003FMonitor\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003ARelayCommand\_007BT\_007D\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FSourcesCache\_003Fd5f9ed2178f3d3dcf1fd45e7604d8217cd25a16edff8d7dc27769ff7c958e6\_003FRelayCommand\_007BT\_007D\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003ATask\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FSourcesCache\_003Fa77b231cea33ff338c41dc7869d6c493d4dae137ff51e342173efa61d933\_003FTask\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003ATask\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E2\_003Fresharper\_002Dhost\_003FSourcesCache\_003Fa77b231cea33ff338c41dc7869d6c493d4dae137ff51e342173efa61d933\_003FTask\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003AValidationUtils\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FSourcesCache\_003F982b7aa07be88fd96f55c1e9bf6ec5d9ac1fd3a01d9a3344dabc943a4c72\_003FValidationUtils\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

<s:String x:Key="/Default/CodeInspection/ExcludedFiles/FilesAndFoldersToSkip2/=7020124F\_002D9FFC\_002D4AC3\_002D8F3D\_002DAAB8E0240759\_002Ff\_003AViews\_005FCards\_005FCharCard\_002Examl\_002Esg\_002Ecs\_002Fl\_003AC\_0021\_003FUsers\_003Fstrhzy\_003FAppData\_003FRoaming\_003FJetBrains\_003FRider2025\_002E1\_003Fresharper\_002Dhost\_003FSourcesCache\_003F13b160926dc9b2f28b147f71416825355522c9\_003FViews\_005FCards\_005FCharCard\_002Examl\_002Esg\_002Ecs/@EntryIndexedValue">ForceIncluded</s:String>

</wpf:ResourceDictionary>

16. DnDClient\_TemporaryKey.pfx

0� ����0� �� \*�H�� ����� ��� }0� y0���� \*�H�� �����������0���0�����\*�H�� ��

������0���0��

\*�H�� ����0�����8E�cX���������-���.#3�W �)����`3I&~���ؽ�zw"�v���u��k�����|�������%�5ɭNQ�(L�'U{���EH:��s����D[����z���u[��#��wM1���\_��|SE����ϳ�3R�<�������-�"�Ggg2�FF�]r��O�C��w��0F����`d/��O�O�4�6)��l(����:���-�����O��[��𱉹�� �j�5�W桳���dא��6� �s�7

٪E��r��J���"(�Q�G��OPbl���+L>�^��x�{�B^�"�T���@Y�r�L�t79�l��֏��c���'"]��S��%�)�33F��������h���&g�����M�}O^t�@��/�e���˺:�^�[ۢ4��Hd:\�A������I��j�����3������\_������� [�Xڗ�O���a��G��������8nXrۻ����L�Q+"��=E���P�J���ѓ�����r�ƆH���$������&��\_�����C�u��� �����R��fp�<p�3{��2���g�ڊÌ�LіWXﺖ�5��(�R�Y`���)Hx�b���C�,��v���d�&>i��M������E�� O�r�>��(f�������Pmx����=Y���yb.����@��g�-�h��}��S�f#�ئ+Z��~ Ur-�����݅Ʌzt��R�\_�r���2�`������Y�\_�܋�`,rU�����;�f�^]��&V2}�Eս����XwE44�

��T�������ԅ�`BM���ֻe� �YX'������0���F=��1��1�����a���x��l��w���`�B����L�I�fa&N�܈����d�5f����E-����++�r��w��6M�#N��Eu�[�w�����f'7��1Gj3VQa���� �r)�����I�Jj�����\* N%

j����������b��\�

h�b.)4Z�El��wAG#��y��˓���M�:ϋwI�v(�E���r73����z�~s�\_u�a��������\*}�����8�@��µk�C.��S��;�������(�����q���O����t6��-����:�

+��k��NPRSgLz�F�������:XAU�H��������x�2�'�iN���4����@��G��O�;�����u��ǝ��z(ّᝧ�˥a�0��V��}᜼���;5N��H�k¯g�/1߰���4I�G%z`.t��8�AI�2\*��H���C�׹^��~�>�`��Nz�$�#aP�EX=uy,Ï�0F��>�n�����B��g��-�SK2D��k��Z�%

����N�d��"y���(1��0�� \*�H�� � �1�������0W� \*�H�� � �1J�H�d�b�2�2�b�d�a�0�-�5�9�d�d�-�4�c�a�d�-�b�5�4�e�-�f�b�c�9�7�d�1�9�f�3�3�c0]� +�����7��1P�N�M�i�c�r�o�s�o�f�t� �S�t�r�o�n�g� �C�r�y�p�t�o�g�r�a�p�h�i�c� �P�r�o�v�i�d�e�r0��w� \*�H�� ������h0��d���0��]� \*�H�� ���0��

\*�H�� ����0����^�i㮭<���Ѐ��0H�n�gԪ!l���-"[~�`RT��(Ѧ��ӫsc�7��mhδĽ�i� �Tr���'�(��BJ��y���篠�9s���y��?�q���N���ATA1��IM���F���B����4B�!Ro�|Ee��h�7��@� �v��\*�=��߭��{���Z�):���j��x��&p;��.���a���!� ˄����brR׭��%�}�|JuԼ������9{��T��I���A%�3�A�����Ka�����k��}��� �g��\hw "�L��=� �K���|�T㙿f��O#�j͜+��̀�d:��V+~3�I�?�ze!�bp-��f�ɥ��8.5[#�A/�?��Ti`^1i�2��&� Q��-�#`����\_м����U�N!m��i7����T-�L�җ��i�LR�[Ђ?���'{�-F��Cs\*�f5­��I쬆�=���iMI���e�Q�g�DR�^���À�����l���X1�'����r��]���N���"�"��aJ]˝K�pZ��P��N����R|z���:\_y��A �)���p���h[��\���ۊ�/\_4lﺉ�&���.�����T4�ˉ)�\*��E���~ůRK�����h�K=��ͥ��OI���ʺ(T���DA�g����qlEj������;���~�8Ӓ<aM��'����������7

4W8!�)����@��G �\]���6��|)~SIx���T�$�Ts�b��,�1�gu�Xb��;��" ���b����ߒ<Q����\*\_i���YT��h�s�Bl�\*���A���1����\��Βn@������tF���ku�� A�'k�ڑ��\*��

A��W/����v��$8����\*�9�������l5;�ĕ.9�k{̑��6?Ww�0;0�0���+�������Ӥ��&��-yT����[��g�����,�/���%�OU0�ǽd˯!����

17. MauiProgram.cs

﻿using CommunityToolkit.Maui;

using DnDClient.ViewModels;

using DnDClient.Views;

using Microsoft.Extensions.Logging;

namespace DnDClient;

public static class MauiProgram

{

public static MauiApp CreateMauiApp()

{

var builder = MauiApp.CreateBuilder();

builder

.UseMauiApp<App>()

.UseMauiCommunityToolkit()

.ConfigureFonts(fonts =>

{

fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");

fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");

});

builder.Logging.AddConsole();

builder.Services.AddTransient<CharactersPage>();

builder.Services.AddTransient<CharacterDetailsPage>();

builder.Services.AddTransient<MainViewModel>();

builder.Services.AddTransient<AuthViewModel>();

builder.Services.AddTransient<CampaignListViewModel>();

builder.Services.AddTransient<CampaignViewModel>();

builder.Services.AddTransient<CharactersViewModel>();

builder.Services.AddTransient<CharDetailsViewModel>();

builder.Services.AddTransient<CombatViewModel>();

builder.Services.AddTransient<CombatPage>();

builder.Services.AddTransient<CombatParticipantsViewModel>();

builder.Services.AddTransient<EntityManagementViewModel>();

builder.Services.AddTransient<CreateEnemyViewModel>();

builder.Services.AddTransient<CreateNPCViewModel>();

builder.Services.AddTransient<CreateEnemyPage>();

builder.Services.AddTransient<CreateNPCPage>();

#if DEBUG

builder.Logging.AddDebug();

#endif

return builder.Build();

}

}

18. Models\Attack.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class Attack : ObservableObject

{

[ObservableProperty] private int? attackBonus;

[JsonPropertyName("damage\_dice")] [ObservableProperty]

private string damageDice = "default";

[ObservableProperty] private string description = "default";

[JsonIgnore] [ObservableProperty] private Enemy? enemy;

[ForeignKey("Enemy")] [ObservableProperty]

private Guid? enemyId;

[Key] [ObservableProperty] private Guid id = Guid.NewGuid();

[ObservableProperty] private string name = "default";

[JsonIgnore] [ObservableProperty] private NPC? npc;

[ForeignKey("NPC")] [ObservableProperty]

private Guid? npcId;

[JsonIgnore] [ObservableProperty] private PlayerCharacter? playerCharacter;

[ForeignKey("PlayerCharacter")] [ObservableProperty]

private Guid? playerCharacterId;

}

19. Models\Campaign.cs

﻿using System.Collections.ObjectModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class Campaign : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string name = string.Empty;

[ObservableProperty]

private ObservableCollection<StoryElement> plotItems = new();

[ObservableProperty]

private ObservableCollection<PlayerCharacter> playerCharacters = new();

[ObservableProperty]

private ObservableCollection<Combat> combats = new();

[ForeignKey("User")]

[ObservableProperty]

private Guid masterId;

[JsonIgnore]

[ObservableProperty]

private User? master;

}

20. Models\Combat.cs

﻿using System.Collections.ObjectModel;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class Combat : ObservableObject

{

[JsonIgnore] [ObservableProperty] private Campaign? campaign;

[ForeignKey("Campaign")] [ObservableProperty]

private Guid campaignId;

[JsonIgnore] [ObservableProperty] private ObservableCollection<CombatLog>? combatLogs;

[NotMapped] [ObservableProperty] private CombatParticipant? currentParticipant;

[ObservableProperty] private int currentRound = 1;

[ObservableProperty] private int currentTurnIndex = 0;

[Key] [ObservableProperty] private Guid id = Guid.NewGuid();

[ObservableProperty] private string name = string.Empty;

[ObservableProperty] private ObservableCollection<CombatParticipant> participants = new();

partial void OnCurrentTurnIndexChanged(int value)

{

CurrentParticipant = Participants != null && value >= 0 && value < Participants.Count

? Participants[value]

: null;

}

}

public enum ParticipantType

{

Player,

Npc,

Enemy

}

21. Models\CombatLog.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class CombatLog : ObservableObject

{

[Key] [ObservableProperty] private Guid id = Guid.NewGuid();

[ForeignKey("Combat")] [ObservableProperty] private Guid? combatId;

[ObservableProperty] private string type;

[ObservableProperty] private Guid sourceId;

[ObservableProperty] private Guid targetId;

[ObservableProperty] private int? damage;

[ObservableProperty] private string? message;

[JsonIgnore] [ObservableProperty] private Combat? combat;

[ObservableProperty] private CombatParticipant source;

[ObservableProperty] private CombatParticipant target;

}

22. Models\CombatParticipant.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class CombatParticipant : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string name = string.Empty;

[ObservableProperty]

private int initiative;

[ObservableProperty]

private int currentHitPoints;

[ObservableProperty]

private int maxHitPoints;

[ObservableProperty]

private int armorClass;

[ObservableProperty]

private bool isActive;

[ObservableProperty]

private ParticipantType type;

[ObservableProperty]

private Guid? sourceId;

[ForeignKey("Combat")]

[ObservableProperty]

private Guid combatId;

[JsonIgnore]

[ObservableProperty]

private Combat? combat;

}

23. Models\Enemy.cs

﻿using System.Collections.ObjectModel;

using System.ComponentModel.DataAnnotations;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class Enemy : ObservableObject

{

[ObservableProperty] private ObservableCollection<Attack> actions = new();

[ObservableProperty] private string? alignment;

[ObservableProperty] private int? armorClass;

[ObservableProperty] private string? armorDescription;

[ObservableProperty] private ObservableCollection<Attack> attacks;

[ObservableProperty] private ObservableCollection<Attack> bonusActions = new();

[ObservableProperty] private string? challengeRating;

[ObservableProperty] private double? challengeRatingValue;

[ObservableProperty] private int? charisma;

[ObservableProperty] private string? conditionImmunities;

[ObservableProperty] private int? constitution;

[ObservableProperty] private int? currentHitPoints;

[ObservableProperty] private string? damageImmunities;

[ObservableProperty] private string? damageResistances;

[ObservableProperty] private string? damageVulnerabilities;

[ObservableProperty] private string? description;

[ObservableProperty] private int? dexterity;

[ObservableProperty] private string? group;

[ObservableProperty] private string? hitDice;

[Key] [ObservableProperty] private Guid id = Guid.NewGuid();

[ObservableProperty] private int? intelligence;

[ObservableProperty] private string? languages;

[ObservableProperty] private ObservableCollection<Attack> legendaryActions = new();

[ObservableProperty] private string? legendaryDescription;

[ObservableProperty] private string name = string.Empty;

[ObservableProperty] private int? perception;

[ObservableProperty] private ObservableCollection<Attack> reactions = new();

[ObservableProperty] private string? senses;

[ObservableProperty] private string? size;

[ObservableProperty] private Dictionary<string, int>? skills;

[ObservableProperty] private string? slug;

[ObservableProperty] private ObservableCollection<SpecialAbility> specialAbilities = new();

[ObservableProperty] private Dictionary<string, string>? speed;

[ObservableProperty] private ObservableCollection<string> spellObservableCollection = new();

[ObservableProperty] private int? strength;

[ObservableProperty] private string? subtype;

[ObservableProperty] private string? type;

[ObservableProperty] private int? wisdom;

}

24. Models\Log.cs

﻿using System.ComponentModel.DataAnnotations;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class Log : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string tag = string.Empty;

[ObservableProperty]

private string message = string.Empty;

[ObservableProperty]

private DateTime date = DateTime.UtcNow;

}

25. Models\NPC.cs

﻿using System.Collections.ObjectModel;

using System.ComponentModel.DataAnnotations;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class NPC : ObservableObject

{

[ObservableProperty] private int armorClass = 10;

[ObservableProperty] private ObservableCollection<Attack> attacks;

[ObservableProperty] private string bonds = string.Empty;

[ObservableProperty] private string description = string.Empty;

[ObservableProperty] private string flaws = string.Empty;

[ObservableProperty] private int hitPoints = 10;

[Key] [ObservableProperty] private Guid id = Guid.NewGuid();

[ObservableProperty] private string ideals = string.Empty;

[ObservableProperty] private string name = string.Empty;

[ObservableProperty] private string occupation = string.Empty;

[ObservableProperty] private string personalityTraits = string.Empty;

[ObservableProperty] private string race = string.Empty;

}

26. Models\PlayerCharacter.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class PlayerCharacter : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string name = "default";

[ObservableProperty]

private string playerName = "default";

[ObservableProperty]

private string classType = "default";

[ObservableProperty]

private string background = "default";

[ObservableProperty]

private string race = "default";

[ObservableProperty]

private string alignment = "default";

[ObservableProperty]

private int experiencePoints = 0;

[ObservableProperty]

private int level = 1;

[ObservableProperty]

private bool inspiration;

[ForeignKey("User")]

[ObservableProperty]

private Guid userId;

[JsonIgnore]

[ObservableProperty]

private User? user;

// Валюта

[ObservableProperty]

private int copperPieces = 0;

[ObservableProperty]

private int silverPieces = 0;

[ObservableProperty]

private int electrumPieces = 0;

[ObservableProperty]

private int goldPieces = 0;

[ObservableProperty]

private int platinumPieces = 0;

// Черты характера

[ObservableProperty]

private string personalityTraits = "default";

[ObservableProperty]

private string ideals = "default";

[ObservableProperty]

private string bonds = "default";

[ObservableProperty]

private string flaws = "default";

// Характеристики

[ObservableProperty]

private int strength = 0;

[ObservableProperty]

private int dexterity = 0;

[ObservableProperty]

private int constitution = 0;

[ObservableProperty]

private int intelligence = 0;

[ObservableProperty]

private int wisdom = 0;

[ObservableProperty]

private int charisma = 0;

// Proficiency бонус

[ObservableProperty]

private int proficiencyBonus = 0;

// Saving throws proficiencies

[ObservableProperty]

private bool savingThrowStrengthProficiency;

[ObservableProperty]

private bool savingThrowDexterityProficiency;

[ObservableProperty]

private bool savingThrowConstitutionProficiency;

[ObservableProperty]

private bool savingThrowIntelligenceProficiency;

[ObservableProperty]

private bool savingThrowWisdomProficiency;

[ObservableProperty]

private bool savingThrowCharismaProficiency;

// Saving throws

[ObservableProperty]

private int savingThrowStrength = 0;

[ObservableProperty]

private int savingThrowDexterity = 0;

[ObservableProperty]

private int savingThrowConstitution = 0;

[ObservableProperty]

private int savingThrowIntelligence = 0;

[ObservableProperty]

private int savingThrowWisdom = 0;

[ObservableProperty]

private int savingThrowCharisma = 0;

// Навыки

[ObservableProperty]

private int acrobatics = 0;

[ObservableProperty]

private int animalHandling = 0;

[ObservableProperty]

private int arcana = 0;

[ObservableProperty]

private int athletics = 0;

[ObservableProperty]

private int deception = 0;

[ObservableProperty]

private int history = 0;

[ObservableProperty]

private int insight = 0;

[ObservableProperty]

private int intimidation = 0;

[ObservableProperty]

private int investigation = 0;

[ObservableProperty]

private int medicine = 0;

[ObservableProperty]

private int nature = 0;

[ObservableProperty]

private int perception = 0;

[ObservableProperty]

private int performance = 0;

[ObservableProperty]

private int persuasion = 0;

[ObservableProperty]

private int religion = 0;

[ObservableProperty]

private int sleightOfHand = 0;

[ObservableProperty]

private int stealth = 0;

[ObservableProperty]

private int survival = 0;

[ObservableProperty]

private int passiveWisdom = 0;

// Боевая статистика

[ObservableProperty]

private int armorClass = 0;

[ObservableProperty]

private int initiative = 0;

[ObservableProperty]

private int speed = 0;

[ObservableProperty]

private int maxHitPoints = 0;

[ObservableProperty]

private int currentHitPoints = 0;

[ObservableProperty]

private int temporaryHitPoints = 0;

[ObservableProperty]

private string hitDice = "default";

[ObservableProperty]

private int deathSaveSuccesses = 0;

[ObservableProperty]

private int deathSaveFailures = 0;

// Атаки и заклинания

[ObservableProperty]

private List<Attack> attacks = new();

[ObservableProperty]

private string featuresAndTraits = "default";

[ObservableProperty]

private string equipment = "default";

[ObservableProperty]

private string proficienciesAndLanguages = "default";

// Логика для saving throws

private readonly Dictionary<string, (int Score, bool IsProficient, string SavingThrowProperty)> \_abilityScores = new()

{

{ nameof(Strength), (0, false, nameof(SavingThrowStrength)) },

{ nameof(Dexterity), (0, false, nameof(SavingThrowDexterity)) },

{ nameof(Constitution), (0, false, nameof(SavingThrowConstitution)) },

{ nameof(Intelligence), (0, false, nameof(SavingThrowIntelligence)) },

{ nameof(Wisdom), (0, false, nameof(SavingThrowWisdom)) },

{ nameof(Charisma), (0, false, nameof(SavingThrowCharisma)) }

};

private int CalculateSavingThrow(int abilityScore, bool isProficient)

{

int modifier = (abilityScore - 10) / 2;

return isProficient ? modifier + ProficiencyBonus : modifier;

}

private void UpdateSavingThrow(string abilityName)

{

if (\_abilityScores.TryGetValue(abilityName, out var data))

{

int value = CalculateSavingThrow(data.Score, data.IsProficient);

typeof(PlayerCharacter).GetProperty(data.SavingThrowProperty)!.SetValue(this, value);

}

}

partial void OnStrengthChanged(int value)

{

\_abilityScores[nameof(Strength)] = (value, SavingThrowStrengthProficiency, nameof(SavingThrowStrength));

UpdateSavingThrow(nameof(Strength));

}

partial void OnDexterityChanged(int value)

{

\_abilityScores[nameof(Dexterity)] = (value, SavingThrowDexterityProficiency, nameof(SavingThrowDexterity));

UpdateSavingThrow(nameof(Dexterity));

}

partial void OnConstitutionChanged(int value)

{

\_abilityScores[nameof(Constitution)] = (value, SavingThrowConstitutionProficiency, nameof(SavingThrowConstitution));

UpdateSavingThrow(nameof(Constitution));

}

partial void OnIntelligenceChanged(int value)

{

\_abilityScores[nameof(Intelligence)] = (value, SavingThrowIntelligenceProficiency, nameof(SavingThrowIntelligence));

UpdateSavingThrow(nameof(Intelligence));

}

partial void OnWisdomChanged(int value)

{

\_abilityScores[nameof(Wisdom)] = (value, SavingThrowWisdomProficiency, nameof(SavingThrowWisdom));

UpdateSavingThrow(nameof(Wisdom));

}

partial void OnCharismaChanged(int value)

{

\_abilityScores[nameof(Charisma)] = (value, SavingThrowCharismaProficiency, nameof(SavingThrowCharisma));

UpdateSavingThrow(nameof(Charisma));

}

partial void OnSavingThrowStrengthProficiencyChanged(bool value)

{

\_abilityScores[nameof(Strength)] = (Strength, value, nameof(SavingThrowStrength));

UpdateSavingThrow(nameof(Strength));

}

partial void OnSavingThrowDexterityProficiencyChanged(bool value)

{

\_abilityScores[nameof(Dexterity)] = (Dexterity, value, nameof(SavingThrowDexterity));

UpdateSavingThrow(nameof(Dexterity));

}

partial void OnSavingThrowConstitutionProficiencyChanged(bool value)

{

\_abilityScores[nameof(Constitution)] = (Constitution, value, nameof(SavingThrowConstitution));

UpdateSavingThrow(nameof(Constitution));

}

partial void OnSavingThrowIntelligenceProficiencyChanged(bool value)

{

\_abilityScores[nameof(Intelligence)] = (Intelligence, value, nameof(SavingThrowIntelligence));

UpdateSavingThrow(nameof(Intelligence));

}

partial void OnSavingThrowWisdomProficiencyChanged(bool value)

{

\_abilityScores[nameof(Wisdom)] = (Wisdom, value, nameof(SavingThrowWisdom));

UpdateSavingThrow(nameof(Wisdom));

}

partial void OnSavingThrowCharismaProficiencyChanged(bool value)

{

\_abilityScores[nameof(Charisma)] = (Charisma, value, nameof(SavingThrowCharisma));

UpdateSavingThrow(nameof(Charisma));

}

partial void OnProficiencyBonusChanged(int value)

{

foreach (var key in \_abilityScores.Keys)

{

UpdateSavingThrow(key);

}

}

}

27. Models\SpecialAbility.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class SpecialAbility : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string? name;

[ObservableProperty]

private string? description;

[ForeignKey("Enemy")]

[ObservableProperty]

private Guid enemyId;

[JsonIgnore]

[ObservableProperty]

private Enemy? enemy;

}

28. Models\StoryElement.cs

﻿using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text.Json.Serialization;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class StoryElement : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string name = string.Empty;

[ObservableProperty]

private string description = string.Empty;

[ForeignKey("Campaign")]

[ObservableProperty]

private Guid campaignId;

[JsonIgnore]

[ObservableProperty]

private Campaign? campaign;

}

29. Models\User.cs

﻿using System.ComponentModel.DataAnnotations;

using CommunityToolkit.Mvvm.ComponentModel;

namespace DnDClient.Models;

public partial class User : ObservableObject

{

[Key]

[ObservableProperty]

private Guid id = Guid.NewGuid();

[ObservableProperty]

private string username = string.Empty;

[ObservableProperty]

private string passwordHash = string.Empty;

[ObservableProperty]

private UserRole role = UserRole.Player;

[ObservableProperty]

private List<PlayerCharacter> characters = new();

}

public enum UserRole

{

Master,

Player

}

30. Platforms\Android\AndroidManifest.xml

﻿<?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"

android:usesCleartextTraffic="true"

android:requestLegacyExternalStorage="true">

</application>

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.READ\_MEDIA\_IMAGES" />

<uses-permission android:name="android.permission.READ\_MEDIA\_VIDEO" />

<uses-permission android:name="android.permission.READ\_MEDIA\_AUDIO" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

</manifest>

31. Platforms\Android\MainActivity.cs

﻿using Android.App;

using Android.Content.PM;

using Android.OS;

namespace DnDClient;

[Activity(Theme = "@style/Maui.SplashTheme", MainLauncher = true, LaunchMode = LaunchMode.SingleTop,

ConfigurationChanges = ConfigChanges.ScreenSize | ConfigChanges.Orientation | ConfigChanges.UiMode |

ConfigChanges.ScreenLayout | ConfigChanges.SmallestScreenSize | ConfigChanges.Density)]

public class MainActivity : MauiAppCompatActivity

{

}

32. Platforms\Android\MainApplication.cs

﻿using Android.App;

using Android.Runtime;

namespace DnDClient;

[Application]

public class MainApplication : MauiApplication

{

public MainApplication(IntPtr handle, JniHandleOwnership ownership)

: base(handle, ownership)

{

}

protected override MauiApp CreateMauiApp() => MauiProgram.CreateMauiApp();

}

33. Platforms\Android\Resources\values\colors.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#512BD4</color>

<color name="colorPrimaryDark">#2B0B98</color>

<color name="colorAccent">#2B0B98</color>

</resources>

34. Platforms\MacCatalyst\AppDelegate.cs

﻿using Foundation;

namespace DnDClient;

[Register("AppDelegate")]

public class AppDelegate : MauiUIApplicationDelegate

{

protected override MauiApp CreateMauiApp() => MauiProgram.CreateMauiApp();

}

35. Platforms\MacCatalyst\Entitlements.plist

﻿<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<!-- See https://aka.ms/maui-publish-app-store#add-entitlements for more information about adding entitlements.-->

<dict>

<!-- App Sandbox must be enabled to distribute a MacCatalyst app through the Mac App Store. -->

<key>com.apple.security.app-sandbox</key>

<true/>

<!-- When App Sandbox is enabled, this value is required to open outgoing network connections. -->

<key>com.apple.security.network.client</key>

<true/>

</dict>

</plist>

36. Platforms\MacCatalyst\Info.plist

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<!-- The Mac App Store requires you specify if the app uses encryption. -->

<!-- Please consult https://developer.apple.com/documentation/bundleresources/information\_property\_list/itsappusesnonexemptencryption -->

<!-- <key>ITSAppUsesNonExemptEncryption</key> -->

<!-- Please indicate <true/> or <false/> here. -->

<!-- Specify the category for your app here. -->

<!-- Please consult https://developer.apple.com/documentation/bundleresources/information\_property\_list/lsapplicationcategorytype -->

<!-- <key>LSApplicationCategoryType</key> -->

<!-- <string>public.app-category.YOUR-CATEGORY-HERE</string> -->

<key>UIDeviceFamily</key>

<array>

<integer>2</integer>

</array>

<key>UIRequiredDeviceCapabilities</key>

<array>

<string>arm64</string>

</array>

<key>UISupportedInterfaceOrientations</key>

<array>

<string>UIInterfaceOrientationPortrait</string>

<string>UIInterfaceOrientationLandscapeLeft</string>

<string>UIInterfaceOrientationLandscapeRight</string>

</array>

<key>UISupportedInterfaceOrientations~ipad</key>

<array>

<string>UIInterfaceOrientationPortrait</string>

<string>UIInterfaceOrientationPortraitUpsideDown</string>

<string>UIInterfaceOrientationLandscapeLeft</string>

<string>UIInterfaceOrientationLandscapeRight</string>

</array>

<key>XSAppIconAssets</key>

<string>Assets.xcassets/appicon.appiconset</string>

</dict>

</plist>

37. Platforms\MacCatalyst\Program.cs

﻿using ObjCRuntime;

using UIKit;

namespace DnDClient;

public class Program

{

// This is the main entry point of the application.

static void Main(string[] args)

{

// if you want to use a different Application Delegate class from "AppDelegate"

// you can specify it here.

UIApplication.Main(args, null, typeof(AppDelegate));

}

}

38. Platforms\Tizen\Main.cs

using System;

using Microsoft.Maui;

using Microsoft.Maui.Hosting;

namespace DnDClient;

class Program : MauiApplication

{

protected override MauiApp CreateMauiApp() => MauiProgram.CreateMauiApp();

static void Main(string[] args)

{

var app = new Program();

app.Run(args);

}

}

39. Platforms\Tizen\tizen-manifest.xml

﻿<?xml version="1.0" encoding="utf-8"?>

<manifest package="maui-application-id-placeholder" version="0.0.0" api-version="8" xmlns="http://tizen.org/ns/packages">

<profile name="common" />

<ui-application appid="maui-application-id-placeholder" exec="DnDClient.dll" multiple="false" nodisplay="false" taskmanage="true" type="dotnet" launch\_mode="single">

<label>maui-application-title-placeholder</label>

<icon>maui-appicon-placeholder</icon>

<metadata key="http://tizen.org/metadata/prefer\_dotnet\_aot" value="true" />

</ui-application>

<shortcut-list />

<privileges>

<privilege>http://tizen.org/privilege/internet</privilege>

</privileges>

<dependencies />

<provides-appdefined-privileges />

</manifest>

40. Platforms\Windows\App.xaml

﻿<maui:MauiWinUIApplication

x:Class="DnDClient.WinUI.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:maui="using:Microsoft.Maui"

xmlns:local="using:DnDClient.WinUI">

</maui:MauiWinUIApplication>

41. Platforms\Windows\App.xaml.cs

﻿using Microsoft.UI.Xaml;

// To learn more about WinUI, the WinUI project structure,

// and more about our project templates, see: http://aka.ms/winui-project-info.

namespace DnDClient.WinUI;

/// <summary>

/// Provides application-specific behavior to supplement the default Application class.

/// </summary>

public partial class App : MauiWinUIApplication

{

/// <summary>

/// Initializes the singleton application object. This is the first line of authored code

/// executed, and as such is the logical equivalent of main() or WinMain().

/// </summary>

public App()

{

this.InitializeComponent();

}

protected override MauiApp CreateMauiApp() => MauiProgram.CreateMauiApp();

}

42. Platforms\Windows\Package.appxmanifest

﻿<?xml version="1.0" encoding="utf-8"?>

<Package

xmlns="http://schemas.microsoft.com/appx/manifest/foundation/windows10"

xmlns:uap="http://schemas.microsoft.com/appx/manifest/uap/windows10"

xmlns:mp="http://schemas.microsoft.com/appx/2014/phone/manifest"

xmlns:rescap="http://schemas.microsoft.com/appx/manifest/foundation/windows10/restrictedcapabilities"

IgnorableNamespaces="uap rescap">

<Identity Name="maui-package-name-placeholder" Publisher="CN=strhzy" Version="0.0.1.0" />

<mp:PhoneIdentity PhoneProductId="12240ADE-CBA3-4A65-B30F-CA565F702E13" PhonePublisherId="00000000-0000-0000-0000-000000000000"/>

<Properties>

<DisplayName>$placeholder$</DisplayName>

<PublisherDisplayName>User Name</PublisherDisplayName>

<Logo>$placeholder$.png</Logo>

</Properties>

<Dependencies>

<TargetDeviceFamily Name="Windows.Universal" MinVersion="10.0.17763.0" MaxVersionTested="10.0.19041.0" />

<TargetDeviceFamily Name="Windows.Desktop" MinVersion="10.0.17763.0" MaxVersionTested="10.0.19041.0" />

</Dependencies>

<Resources>

<Resource Language="x-generate" />

</Resources>

<Applications>

<Application Id="App" Executable="$targetnametoken$.exe" EntryPoint="$targetentrypoint$">

<uap:VisualElements

DisplayName="$placeholder$"

Description="$placeholder$"

Square150x150Logo="$placeholder$.png"

Square44x44Logo="$placeholder$.png"

BackgroundColor="transparent">

<uap:DefaultTile Square71x71Logo="$placeholder$.png" Wide310x150Logo="$placeholder$.png" Square310x310Logo="$placeholder$.png" />

<uap:SplashScreen Image="$placeholder$.png" />

</uap:VisualElements>

</Application>

</Applications>

<Capabilities>

<rescap:Capability Name="runFullTrust" />

</Capabilities>

</Package>

43. Platforms\Windows\app.manifest

<?xml version="1.0" encoding="utf-8"?>

<assembly manifestVersion="1.0" xmlns="urn:schemas-microsoft-com:asm.v1">

<assemblyIdentity version="1.0.0.0" name="DnDClient.WinUI.app"/>

<application xmlns="urn:schemas-microsoft-com:asm.v3">

<windowsSettings>

<!-- The combination of below two tags have the following effect:

1) Per-Monitor for >= Windows 10 Anniversary Update

2) System < Windows 10 Anniversary Update

-->

<dpiAware xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">true/PM</dpiAware>

<dpiAwareness xmlns="http://schemas.microsoft.com/SMI/2016/WindowsSettings">PerMonitorV2, PerMonitor</dpiAwareness>

</windowsSettings>

</application>

</assembly>

44. Platforms\iOS\AppDelegate.cs

﻿using Foundation;

namespace DnDClient;

[Register("AppDelegate")]

public class AppDelegate : MauiUIApplicationDelegate

{

protected override MauiApp CreateMauiApp() => MauiProgram.CreateMauiApp();

}

45. Platforms\iOS\Info.plist

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>LSRequiresIPhoneOS</key>

<true/>

<key>UIDeviceFamily</key>

<array>

<integer>1</integer>

<integer>2</integer>

</array>

<key>UIRequiredDeviceCapabilities</key>

<array>

<string>arm64</string>

</array>

<key>UISupportedInterfaceOrientations</key>

<array>

<string>UIInterfaceOrientationPortrait</string>

<string>UIInterfaceOrientationLandscapeLeft</string>

<string>UIInterfaceOrientationLandscapeRight</string>

</array>

<key>UISupportedInterfaceOrientations~ipad</key>

<array>

<string>UIInterfaceOrientationPortrait</string>

<string>UIInterfaceOrientationPortraitUpsideDown</string>

<string>UIInterfaceOrientationLandscapeLeft</string>

<string>UIInterfaceOrientationLandscapeRight</string>

</array>

<key>XSAppIconAssets</key>

<string>Assets.xcassets/appicon.appiconset</string>

</dict>

</plist>

46. Platforms\iOS\Program.cs

﻿using ObjCRuntime;

using UIKit;

namespace DnDClient;

public class Program

{

// This is the main entry point of the application.

static void Main(string[] args)

{

// if you want to use a different Application Delegate class from "AppDelegate"

// you can specify it here.

UIApplication.Main(args, null, typeof(AppDelegate));

}

}

47. Platforms\iOS\Resources\PrivacyInfo.xcprivacy

<?xml version="1.0" encoding="UTF-8"?>

<!--

This is the minimum required version of the Apple Privacy Manifest for .NET MAUI apps.

The contents below are needed because of APIs that are used in the .NET framework and .NET MAUI SDK.

You are responsible for adding extra entries as needed for your application.

More information: https://aka.ms/maui-privacy-manifest

-->

<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">

<plist version="1.0">

<dict>

<key>NSPrivacyAccessedAPITypes</key>

<array>

<dict>

<key>NSPrivacyAccessedAPIType</key>

<string>NSPrivacyAccessedAPICategoryFileTimestamp</string>

<key>NSPrivacyAccessedAPITypeReasons</key>

<array>

<string>C617.1</string>

</array>

</dict>

<dict>

<key>NSPrivacyAccessedAPIType</key>

<string>NSPrivacyAccessedAPICategorySystemBootTime</string>

<key>NSPrivacyAccessedAPITypeReasons</key>

<array>

<string>35F9.1</string>

</array>

</dict>

<dict>

<key>NSPrivacyAccessedAPIType</key>

<string>NSPrivacyAccessedAPICategoryDiskSpace</string>

<key>NSPrivacyAccessedAPITypeReasons</key>

<array>

<string>E174.1</string>

</array>

</dict>

<!--

The entry below is only needed when you're using the Preferences API in your app.

<dict>

<key>NSPrivacyAccessedAPIType</key>

<string>NSPrivacyAccessedAPICategoryUserDefaults</string>

<key>NSPrivacyAccessedAPITypeReasons</key>

<array>

<string>CA92.1</string>

</array>

</dict> -->

</array>

</dict>

</plist>

48. Properties\PublishProfiles\MSIX-win-x64.pubxml

﻿<?xml version="1.0" encoding="utf-8"?>

<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<PropertyGroup>

<PublishDir>bin\Release\net8.0-android\publish\</PublishDir>

<PublishProtocol>FileSystem</PublishProtocol>

<RuntimeIdentifier>win-x64</RuntimeIdentifier>

<Platform>Any CPU</Platform>

<Configuration>Release</Configuration>

<TargetFramework>net8.0-windows10.0.19041.0</TargetFramework>

<PublishSingleFile>false</PublishSingleFile>

<PublishReadyToRun>false</PublishReadyToRun>

<SelfContained>True</SelfContained>

<PublishAppxPackage>true</PublishAppxPackage>

<AppxPackageDir>bin\Debug\net8.0-windows10.0.19041.0\win10-x64\AppPackages\</AppxPackageDir>

</PropertyGroup>

</Project>

49. Properties\launchSettings.json

{

"profiles": {

"Windows Machine": {

"commandName": "MsixPackage",

"nativeDebugging": false

}

}

}

50. Resources\Raw\AboutAssets.txt

﻿Any raw assets you want to be deployed with your application can be placed in

this directory (and child directories). Deployment of the asset to your application

is automatically handled by the following `MauiAsset` Build Action within your `.csproj`.

<MauiAsset Include="Resources\Raw\\*\*" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />

These files will be deployed with your package and will be accessible using Essentials:

async Task LoadMauiAsset()

{

using var stream = await FileSystem.OpenAppPackageFileAsync("AboutAssets.txt");

using var reader = new StreamReader(stream);

var contents = reader.ReadToEnd();

}

51. Resources\Styles\Colors.xaml

﻿<?xml version="1.0" encoding="UTF-8" ?>

<?xaml-comp compile="true" ?>

<ResourceDictionary

xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

<!-- Note: For Android please see also Platforms\Android\Resources\values\colors.xml -->

<Color x:Key="Primary">#512BD4</Color>

<Color x:Key="PrimaryDark">#ac99ea</Color>

<Color x:Key="PrimaryDarkText">#242424</Color>

<Color x:Key="Secondary">#DFD8F7</Color>

<Color x:Key="SecondaryDarkText">#9880e5</Color>

<Color x:Key="Tertiary">#2B0B98</Color>

<Color x:Key="White">White</Color>

<Color x:Key="Black">Black</Color>

<Color x:Key="Magenta">#D600AA</Color>

<Color x:Key="MidnightBlue">#190649</Color>

<Color x:Key="OffBlack">#1f1f1f</Color>

<Color x:Key="Gray100">#E1E1E1</Color>

<Color x:Key="Gray200">#C8C8C8</Color>

<Color x:Key="Gray300">#ACACAC</Color>

<Color x:Key="Gray400">#919191</Color>

<Color x:Key="Gray500">#6E6E6E</Color>

<Color x:Key="Gray600">#404040</Color>

<Color x:Key="Gray900">#212121</Color>

<Color x:Key="Gray950">#141414</Color>

<SolidColorBrush x:Key="PrimaryBrush" Color="{StaticResource Primary}"/>

<SolidColorBrush x:Key="SecondaryBrush" Color="{StaticResource Secondary}"/>

<SolidColorBrush x:Key="TertiaryBrush" Color="{StaticResource Tertiary}"/>

<SolidColorBrush x:Key="WhiteBrush" Color="{StaticResource White}"/>

<SolidColorBrush x:Key="BlackBrush" Color="{StaticResource Black}"/>

<SolidColorBrush x:Key="Gray100Brush" Color="{StaticResource Gray100}"/>

<SolidColorBrush x:Key="Gray200Brush" Color="{StaticResource Gray200}"/>

<SolidColorBrush x:Key="Gray300Brush" Color="{StaticResource Gray300}"/>

<SolidColorBrush x:Key="Gray400Brush" Color="{StaticResource Gray400}"/>

<SolidColorBrush x:Key="Gray500Brush" Color="{StaticResource Gray500}"/>

<SolidColorBrush x:Key="Gray600Brush" Color="{StaticResource Gray600}"/>

<SolidColorBrush x:Key="Gray900Brush" Color="{StaticResource Gray900}"/>

<SolidColorBrush x:Key="Gray950Brush" Color="{StaticResource Gray950}"/>

</ResourceDictionary>

52. Resources\Styles\Styles.xaml

﻿<?xml version="1.0" encoding="UTF-8" ?>

<?xaml-comp compile="true" ?>

<ResourceDictionary

xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml">

<Style TargetType="ActivityIndicator">

<Setter Property="Color" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

</Style>

<Style TargetType="IndicatorView">

<Setter Property="IndicatorColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}"/>

<Setter Property="SelectedIndicatorColor" Value="{AppThemeBinding Light={StaticResource Gray950}, Dark={StaticResource Gray100}}"/>

</Style>

<Style TargetType="Border">

<Setter Property="Stroke" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />

<Setter Property="StrokeShape" Value="Rectangle"/>

<Setter Property="StrokeThickness" Value="1"/>

</Style>

<Style TargetType="BoxView">

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource Gray950}, Dark={StaticResource Gray200}}" />

</Style>

<Style TargetType="Button">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource PrimaryDarkText}}" />

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource PrimaryDark}}" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="BorderWidth" Value="0"/>

<Setter Property="CornerRadius" Value="8"/>

<Setter Property="Padding" Value="14,10"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray950}, Dark={StaticResource Gray200}}" />

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

<VisualState x:Name="PointerOver" />

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="CheckBox">

<Setter Property="Color" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="Color" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="DatePicker">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource White}}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Editor">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource White}}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14" />

<Setter Property="PlaceholderColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Entry">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource White}}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14" />

<Setter Property="PlaceholderColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Frame">

<Setter Property="HasShadow" Value="False" />

<Setter Property="BorderColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray950}}" />

<Setter Property="CornerRadius" Value="8" />

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource Black}}" />

</Style>

<Style TargetType="ImageButton">

<Setter Property="Opacity" Value="1" />

<Setter Property="BorderColor" Value="Transparent"/>

<Setter Property="BorderWidth" Value="0"/>

<Setter Property="CornerRadius" Value="0"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="Opacity" Value="0.5" />

</VisualState.Setters>

</VisualState>

<VisualState x:Name="PointerOver" />

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Label">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource White}}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular" />

<Setter Property="FontSize" Value="14" />

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Span">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource White}}" />

</Style>

<Style TargetType="Label" x:Key="Headline">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource MidnightBlue}, Dark={StaticResource White}}" />

<Setter Property="FontSize" Value="32" />

<Setter Property="HorizontalOptions" Value="Center" />

<Setter Property="HorizontalTextAlignment" Value="Center" />

</Style>

<Style TargetType="Label" x:Key="SubHeadline">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource MidnightBlue}, Dark={StaticResource White}}" />

<Setter Property="FontSize" Value="24" />

<Setter Property="HorizontalOptions" Value="Center" />

<Setter Property="HorizontalTextAlignment" Value="Center" />

</Style>

<Style TargetType="ListView">

<Setter Property="SeparatorColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray500}}" />

<Setter Property="RefreshControlColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource Gray200}}" />

</Style>

<Style TargetType="Picker">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource White}}" />

<Setter Property="TitleColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource Gray200}}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

<Setter Property="TitleColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="ProgressBar">

<Setter Property="ProgressColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="ProgressColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="RadioButton">

<Setter Property="BackgroundColor" Value="Transparent"/>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource White}}" />

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="RefreshView">

<Setter Property="RefreshColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource Gray200}}" />

</Style>

<Style TargetType="SearchBar">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource White}}" />

<Setter Property="PlaceholderColor" Value="{StaticResource Gray500}" />

<Setter Property="CancelButtonColor" Value="{StaticResource Gray500}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular" />

<Setter Property="FontSize" Value="14" />

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

<Setter Property="PlaceholderColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="SearchHandler">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource White}}" />

<Setter Property="PlaceholderColor" Value="{StaticResource Gray500}" />

<Setter Property="BackgroundColor" Value="Transparent" />

<Setter Property="FontFamily" Value="OpenSansRegular" />

<Setter Property="FontSize" Value="14" />

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

<Setter Property="PlaceholderColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Shadow">

<Setter Property="Radius" Value="15" />

<Setter Property="Opacity" Value="0.5" />

<Setter Property="Brush" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource White}}" />

<Setter Property="Offset" Value="10,10" />

</Style>

<Style TargetType="Slider">

<Setter Property="MinimumTrackColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

<Setter Property="MaximumTrackColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray600}}" />

<Setter Property="ThumbColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="MinimumTrackColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}"/>

<Setter Property="MaximumTrackColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}"/>

<Setter Property="ThumbColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}"/>

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="SwipeItem">

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource Black}}" />

</Style>

<Style TargetType="Switch">

<Setter Property="OnColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

<Setter Property="ThumbColor" Value="{StaticResource White}" />

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="OnColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

<Setter Property="ThumbColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

<VisualState x:Name="On">

<VisualState.Setters>

<Setter Property="OnColor" Value="{AppThemeBinding Light={StaticResource Secondary}, Dark={StaticResource Gray200}}" />

<Setter Property="ThumbColor" Value="{AppThemeBinding Light={StaticResource Primary}, Dark={StaticResource White}}" />

</VisualState.Setters>

</VisualState>

<VisualState x:Name="Off">

<VisualState.Setters>

<Setter Property="ThumbColor" Value="{AppThemeBinding Light={StaticResource Gray400}, Dark={StaticResource Gray500}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="TimePicker">

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource White}}" />

<Setter Property="BackgroundColor" Value="Transparent"/>

<Setter Property="FontFamily" Value="OpenSansRegular"/>

<Setter Property="FontSize" Value="14"/>

<Setter Property="MinimumHeightRequest" Value="44"/>

<Setter Property="MinimumWidthRequest" Value="44"/>

<Setter Property="VisualStateManager.VisualStateGroups">

<VisualStateGroupList>

<VisualStateGroup x:Name="CommonStates">

<VisualState x:Name="Normal" />

<VisualState x:Name="Disabled">

<VisualState.Setters>

<Setter Property="TextColor" Value="{AppThemeBinding Light={StaticResource Gray300}, Dark={StaticResource Gray600}}" />

</VisualState.Setters>

</VisualState>

</VisualStateGroup>

</VisualStateGroupList>

</Setter>

</Style>

<Style TargetType="Page" ApplyToDerivedTypes="True">

<Setter Property="Padding" Value="0"/>

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource OffBlack}}" />

</Style>

<Style TargetType="Shell" ApplyToDerivedTypes="True">

<Setter Property="Shell.BackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource OffBlack}}" />

<Setter Property="Shell.ForegroundColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource SecondaryDarkText}}" />

<Setter Property="Shell.TitleColor" Value="{AppThemeBinding Light={StaticResource Black}, Dark={StaticResource SecondaryDarkText}}" />

<Setter Property="Shell.DisabledColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray950}}" />

<Setter Property="Shell.UnselectedColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray200}}" />

<Setter Property="Shell.NavBarHasShadow" Value="False" />

<Setter Property="Shell.TabBarBackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource Black}}" />

<Setter Property="Shell.TabBarForegroundColor" Value="{AppThemeBinding Light={StaticResource Magenta}, Dark={StaticResource White}}" />

<Setter Property="Shell.TabBarTitleColor" Value="{AppThemeBinding Light={StaticResource Magenta}, Dark={StaticResource White}}" />

<Setter Property="Shell.TabBarUnselectedColor" Value="{AppThemeBinding Light={StaticResource Gray900}, Dark={StaticResource Gray200}}" />

</Style>

<Style TargetType="NavigationPage">

<Setter Property="BarBackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource OffBlack}}" />

<Setter Property="BarTextColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource White}}" />

<Setter Property="IconColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource White}}" />

</Style>

<Style TargetType="TabbedPage">

<Setter Property="BarBackgroundColor" Value="{AppThemeBinding Light={StaticResource White}, Dark={StaticResource Gray950}}" />

<Setter Property="BarTextColor" Value="{AppThemeBinding Light={StaticResource Magenta}, Dark={StaticResource White}}" />

<Setter Property="UnselectedTabColor" Value="{AppThemeBinding Light={StaticResource Gray200}, Dark={StaticResource Gray950}}" />

<Setter Property="SelectedTabColor" Value="{AppThemeBinding Light={StaticResource Gray950}, Dark={StaticResource Gray200}}" />

</Style>

</ResourceDictionary>

53. Services\ApiHelper.cs

﻿#if ANDROID

using Environment = Android.OS.Environment;

#elif WINDOWS

using Windows.Storage;

#endif

using System.Net;

using System.Text;

using Newtonsoft.Json;

namespace DnDClient.Services;

public static class ApiHelper

{

//private static readonly string \_url = "http://192.168.1.167:5000/api";

private static readonly string \_url = "http://localhost:5000/api";

public static T? Get<T>(string model, Guid id = default)

{

try

{

using var client = new HttpClient();

var path = id == Guid.Empty ? $"{model}" : $"{model}/{id}";

var resp = client.GetAsync($"{\_url}/{path}").Result;

if (resp.StatusCode != HttpStatusCode.OK) return default;

var json = resp.Content.ReadAsStringAsync().Result;

return JsonConvert.DeserializeObject<T>(json);

}

catch (Exception ex)

{

return default;

}

}

public static bool Put<T>(string json, string model, Guid id)

{

try

{

using var client = new HttpClient();

HttpContent body = new StringContent(json, Encoding.UTF8, "application/json");

var resp = client.PutAsync($"{\_url}/{model}/{id}", body).Result;

return resp.StatusCode == HttpStatusCode.NoContent;

}

catch (Exception ex)

{

return false;

}

}

public static bool Post<T>(string json, string model)

{

try

{

using var client = new HttpClient();

HttpContent body = new StringContent(json, Encoding.UTF8, "application/json");

var resp = client.PostAsync($"{\_url}/{model}", body).Result;

return resp.StatusCode == HttpStatusCode.Created;

}

catch (Exception ex)

{

return false;

}

}

public static T? PostWithResponse<T>(string json, string model)

{

try

{

using var client = new HttpClient();

HttpContent body = new StringContent(json, Encoding.UTF8, "application/json");

var resp = client.PostAsync($"{\_url}/{model}", body).Result;

if (resp.StatusCode != HttpStatusCode.Created &&

resp.StatusCode != HttpStatusCode.OK)

return default;

var content = resp.Content.ReadAsStringAsync().Result;

return JsonConvert.DeserializeObject<T>(content);

}

catch (Exception ex)

{

return default;

}

}

public static bool Delete<T>(string model, Guid id)

{

try

{

using var client = new HttpClient();

var resp = client.DeleteAsync($"{\_url}/{model}/{id}").Result;

return resp.StatusCode == HttpStatusCode.NoContent;

}

catch (Exception ex)

{

return false;

}

}

public static async Task<string?> DownloadFileAsync(string model, string fileName, Guid id = default)

{

try

{

string? filePath = null;

await Task.Run(async () =>

{

using var client = new HttpClient();

var path = id == Guid.Empty ? $"{model}" : $"{model}/{id}/pdf";

var response = await client.GetAsync($"{\_url}/{path}", HttpCompletionOption.ResponseHeadersRead);

if (!response.IsSuccessStatusCode)

{

await ShowAlert("Ошибка", "Не удалось скачать файл!");

return;

}

await using var stream = await response.Content.ReadAsStreamAsync();

#if ANDROID

var status = await MainThread.InvokeOnMainThreadAsync(() =>

Permissions.RequestAsync<Permissions.StorageWrite>()

);

if (status != PermissionStatus.Granted)

{

await ShowAlert("Ошибка", "Нет доступа к хранилищу!");

return;

}

filePath = Path.Combine(Environment.ExternalStorageDirectory.AbsolutePath,

Environment.DirectoryDownloads, fileName);

using var fileStream = File.Create(filePath);

await stream.CopyToAsync(fileStream);

#elif WINDOWS

var folder = KnownFolders.DocumentsLibrary;

var file = await folder.CreateFileAsync(fileName, CreationCollisionOption.ReplaceExisting);

filePath = file.Path;

await using (var fileStream = await file.OpenStreamForWriteAsync())

{

await stream.CopyToAsync(fileStream);

}

#else

// iOS и другие платформы

filePath = Path.Combine(FileSystem.CacheDirectory, fileName);

await using var fileStream = File.Create(filePath);

await stream.CopyToAsync(fileStream);

#endif

});

if (filePath != null)

await ShowAlert("Успех", $"Файл сохранён: {filePath}");

return filePath;

}

catch (Exception ex)

{

await ShowAlert("Ошибка", $"Ошибка: {ex.Message}");

return null; // при ошибке возвращаем null

}

}

private static async Task ShowAlert(string title, string message)

{

await MainThread.InvokeOnMainThreadAsync(async () =>

{

if (Application.Current?.MainPage != null)

await Application.Current.MainPage.DisplayAlert(title, message, "OK");

});

}

}

54. Services\Serdeser.cs

﻿using Newtonsoft.Json;

namespace DnDClient.Services;

public class Serdeser

{

public static string Serialize(Object obj)

{

return JsonConvert.SerializeObject(obj);

}

public static T Deserialize<T>(string json)

{

return JsonConvert.DeserializeObject<T>(json);

}

}

55. ViewModels\AuthViewModel.cs

﻿using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using Newtonsoft.Json;

using DnDClient.Services;

namespace DnDClient.ViewModels;

public partial class AuthViewModel : ObservableObject

{

[ObservableProperty]

private string email = string.Empty;

[ObservableProperty]

private string password = string.Empty;

[ObservableProperty]

private string message = string.Empty;

[RelayCommand]

async Task Login()

{

try

{

var loginData = new { Email, PasswordHash = Password };

var json = JsonConvert.SerializeObject(loginData);

var response = ApiHelper.PostWithResponse<LoginResponse>(json, "Auth/login");

if (response != null && !string.IsNullOrEmpty(response.Token))

{

Preferences.Remove("current\_user");

Preferences.Remove("current\_user\_id");

await SecureStorage.SetAsync("auth\_token", response.Token);

Preferences.Set("current\_user", Email);

Preferences.Set("current\_user\_id", ApiHelper.Get<List<User>>("User?query=" + Email).FirstOrDefault().Id.ToString());

Email = string.Empty;

Password = string.Empty;

var shell = (AppShell)Shell.Current;

shell.FlyoutBehavior = FlyoutBehavior.Flyout;

await shell.GoToAsync("//MainPage");

}

else

{

Message = "Авторизация провалена! Проверьте данные или попробуйте позже";

}

}

catch (Exception ex)

{

Message = $"Ошибка: {ex.Message}";

}

}

[RelayCommand]

async Task Register()

{

try

{

var passwordHash = BCrypt.Net.BCrypt.HashPassword(Password);

var registerData = new { Email, PasswordHash = passwordHash };

var json = JsonConvert.SerializeObject(registerData);

var success = ApiHelper.Post<object>(json, "Auth/register");

if (success)

{

Message = "Регистрация успешна! Пожалуйста, войдите в свой аккаунт";

}

else

{

Message = "Регистрация провалена!";

}

}

catch (Exception ex)

{

Message = $"Ошибка: {ex.Message}";

}

}

}

public class LoginResponse

{

public string Token { get; set; } = string.Empty;

}

56. ViewModels\CampaignListViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using DnDClient.Views;

namespace DnDClient.ViewModels;

public partial class CampaignListViewModel : ObservableObject

{

private readonly INavigation \_navigation;

[ObservableProperty] private ObservableCollection<Campaign> campaigns;

public CampaignListViewModel(INavigation navigation)

{

\_navigation = navigation;

var userId = Preferences.Get("current\_user\_id", "");

var userCampaigns = ApiHelper.Get<List<Campaign>>("Campaign?userId=" + userId) ?? new List<Campaign>();

var masterCampaigns = ApiHelper.Get<List<Campaign>>("Campaign?masterId=" + userId) ?? new List<Campaign>();

var allCampaigns = userCampaigns.Union(masterCampaigns, new CampaignIdComparer()).ToList();

campaigns = new ObservableCollection<Campaign>(allCampaigns);

}

[RelayCommand]

private async Task TapCard(Campaign campaign)

{

if (campaign != null)

{

await \_navigation.PushAsync(new CampaignPage(campaign));

}

}

[RelayCommand]

private async Task AddCampaign()

{

var campaign = new Campaign();

var userId = Preferences.Get("current\_user\_id", "");

campaign.Name = "new campaign";

campaign.Master = ApiHelper.Get<User>("User", new Guid(userId));

campaign.MasterId = new Guid(userId);

string json = Serdeser.Serialize(campaign);

try

{

if (ApiHelper.Post<Campaign>(json, "Campaign"))

{

campaigns.Add(campaign);

}

}

catch (Exception e)

{

Console.WriteLine(e);

}

}

[RelayCommand]

private async Task DelCampaign(Campaign campaign)

{

try

{

if (ApiHelper.Delete<Campaign>("Campaign", campaign.Id))

{

campaigns.Remove(campaign);

}

}

catch (Exception e)

{

Console.WriteLine(e);

}

}

private class CampaignIdComparer : IEqualityComparer<Campaign>

{

public bool Equals(Campaign x, Campaign y) => x.Id == y.Id;

public int GetHashCode(Campaign obj) => obj.Id.GetHashCode();

}

}

57. ViewModels\CampaignViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using DnDClient.Views;

using Newtonsoft.Json;

namespace DnDClient.ViewModels;

public partial class CampaignViewModel : ObservableObject

{

private readonly INavigation \_navigation;

[ObservableProperty] private ObservableCollection<PlayerCharacter> availableCharacters = new();

[ObservableProperty] private Campaign campaign;

[ObservableProperty] private ObservableCollection<Combat> combats = new();

[ObservableProperty] private ObservableCollection<PlayerCharacter> filtredCharacters = new();

[ObservableProperty] private bool masterMode;

[ObservableProperty] private string newCombatName;

[ObservableProperty] private string newStoryDescription;

[ObservableProperty] private string newStoryName;

[ObservableProperty] private ObservableCollection<PlayerCharacter> players = new();

[ObservableProperty] private PlayerCharacter selectedCharacterToAdd;

[ObservableProperty] private ObservableCollection<StoryElement> stories = new();

public CampaignViewModel(Campaign? \_campaign)

{

if (\_campaign != null)

{

\_navigation = Application.Current.MainPage.Navigation;

campaign = \_campaign;

masterMode = Preferences.Get("current\_user\_id", "") == campaign.MasterId.ToString();

LoadData();

}

}

public void LoadData()

{

Players = campaign.PlayerCharacters;

var allChars = ApiHelper.Get<List<PlayerCharacter>>("PlayerCharacter") ?? new List<PlayerCharacter>();

AvailableCharacters =

new ObservableCollection<PlayerCharacter>(allChars.Except(Players, new PlayerCharacterIdComparer()));

Stories = ApiHelper.Get<ObservableCollection<StoryElement>>("StoryElement?campaignId=" +

Campaign.Id.ToString());

Combats = ApiHelper.Get<ObservableCollection<Combat>>("Combat?campaignId=" + Campaign.Id.ToString());

Campaign.Combats = Combats;

Campaign.PlotItems = Stories;

}

[RelayCommand]

public void AddPlayerToCampaign()

{

if (SelectedCharacterToAdd != null)

{

var json = JsonConvert.SerializeObject(SelectedCharacterToAdd.Id);

ApiHelper.Post<string>("", $"Campaign/{Campaign.Id}/add\_char/{SelectedCharacterToAdd.Id}");

campaign.PlayerCharacters.Add(SelectedCharacterToAdd);

LoadData();

}

}

[RelayCommand]

public void SaveCampaign()

{

ApiHelper.Put<Campaign>(Serdeser.Serialize(Campaign), "Campaign", Campaign.Id);

}

[RelayCommand]

public void AddStory()

{

if (!string.IsNullOrWhiteSpace(NewStoryName))

{

var story = new StoryElement

{

Name = NewStoryName,

Description = NewStoryDescription,

CampaignId = Campaign.Id

};

var json = JsonConvert.SerializeObject(story);

ApiHelper.Post<StoryElement>(json, "StoryElement");

NewStoryName = string.Empty;

NewStoryDescription = string.Empty;

LoadData();

}

}

[RelayCommand]

public void AddCombat()

{

var combat = new Combat

{

Name = NewCombatName,

CampaignId = campaign.Id,

Campaign = campaign,

Participants = new ObservableCollection<CombatParticipant>()

};

var json = JsonConvert.SerializeObject(combat);

bool success = ApiHelper.Post<Combat>(json, "Combat");

LoadData();

}

[RelayCommand]

public void DeleteStory(StoryElement story)

{

if (story != null)

{

ApiHelper.Delete<StoryElement>($"StoryElement", story.Id);

LoadData();

}

}

[RelayCommand]

public void DeleteCombat(Combat combat)

{

if (combat != null)

{

ApiHelper.Delete<Combat>("Combat", combat.Id);

LoadData();

}

}

[RelayCommand]

public async Task OpenCombatAsync(Combat combat)

{

if (combat == null) return;

var navParam = new Dictionary<string, object>

{

{ "Combat", combat },

{ "MasterMode", masterMode }

};

await \_navigation.PushAsync(new CombatPage(combat, masterMode));

}

[RelayCommand]

public async Task ManageCombatParticipants(Combat combat)

{

if (combat != null)

{

await \_navigation.PushAsync(new CombatParticipantsPage(combat));

}

}

[RelayCommand]

public async Task ManageEntities(Combat combat)

{

if (combat != null)

{

await \_navigation.PushAsync(new EntityManagementPage());

}

}

private class PlayerCharacterIdComparer : IEqualityComparer<PlayerCharacter>

{

public bool Equals(PlayerCharacter x, PlayerCharacter y) => x.Id == y.Id;

public int GetHashCode(PlayerCharacter obj) => obj.Id.GetHashCode();

}

}

58. ViewModels\CharDetailsViewModel.cs

﻿using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

namespace DnDClient.ViewModels;

public partial class CharDetailsViewModel : ObservableObject

{

private readonly INavigation \_navigation;

[ObservableProperty]

private PlayerCharacter \_char;

private PlayerCharacter oldChar;

[ObservableProperty]

private bool editMode = false;

partial void OnEditModeChanged(bool value)

{

if (value)

{

oldChar = new PlayerCharacter();

oldChar = \_char;

}

else if (!value)

{

bool success = ApiHelper.Put<PlayerCharacter>(Serdeser.Serialize(\_char), "PlayerCharacter", \_char.Id);

foreach (var attack in \_char.Attacks)

{

ApiHelper.Put<Attack>(Serdeser.Serialize(attack), "Attack", attack.Id);

}

}

}

[RelayCommand]

async Task AddAttack()

{

var attack = new Attack();

attack.PlayerCharacterId = \_char.Id;

\_char.Attacks.Add(attack);

ApiHelper.Post<Attack>(Serdeser.Serialize(attack), "Attack");

}

public CharDetailsViewModel(INavigation navigation, PlayerCharacter character)

{

\_navigation = navigation;

\_char = character;

}

}

59. ViewModels\CharactersViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using DnDClient.Views;

namespace DnDClient.ViewModels;

public partial class CharactersViewModel : ObservableObject

{

private readonly INavigation \_navigation;

[ObservableProperty] private ObservableCollection<PlayerCharacter> characters;

[RelayCommand]

private async Task TapCard(PlayerCharacter character)

{

if (character != null)

{

await \_navigation.PushAsync(new CharacterDetailsPage(character));

}

}

[RelayCommand]

private async Task Add()

{

var character = new PlayerCharacter();

var stringUserId = Preferences.Get("current\_user\_id", string.Empty);

if (Guid.TryParse(stringUserId, out var userId))

{

character.UserId = userId;

}

if (ApiHelper.Post<PlayerCharacter>(Serdeser.Serialize(character), "PlayerCharacter"))

{

characters.Add(character);

await \_navigation.PushAsync(new CharacterDetailsPage(character));

}

else

{

await Application.Current.MainPage.DisplayAlert("Ошибка", "Персонаж не добавлен", "Закрыть");

}

}

[RelayCommand]

private async Task Delete(PlayerCharacter character)

{

bool confirm = await Application.Current.MainPage.DisplayAlert(

"Подтверждение",

$"Вы уверены, что хотите удалить персонажа {character.Name}?",

"Да",

"Нет");

if (confirm)

{

if (ApiHelper.Delete<PlayerCharacter>("PlayerCharacter", character.Id))

{

characters.Remove(character);

await Application.Current.MainPage.DisplayAlert(

"Инфо",

"Персонаж успешно удален",

"Закрыть");

}

else

{

await Application.Current.MainPage.DisplayAlert(

"Ошибка",

"Не удалось удалить персонажа",

"Закрыть");

}

}

}

[RelayCommand]

public async Task Export(PlayerCharacter character)

{

try

{

var filePath = await ApiHelper.DownloadFileAsync("PlayerCharacter", $"{character.Name}.pdf", character.Id);

if (filePath != null)

{

// Файл успешно сохранён, можно что-то с ним сделать, например, открыть

await Launcher.OpenAsync(new OpenFileRequest { File = new ReadOnlyFile(filePath) });

}

}

catch (Exception ex)

{

await Application.Current.MainPage.DisplayAlert("Ошибка", ex.ToString(), "OK");

}

}

public CharactersViewModel(INavigation navigation)

{

\_navigation = navigation;

string userid = Preferences.Get("current\_user\_id", "default");

Characters = ApiHelper.Get<ObservableCollection<PlayerCharacter>>("PlayerCharacter?query=" + userid);

}

}

60. ViewModels\CombatParticipantsViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using DnDClient.Views;

using Newtonsoft.Json;

namespace DnDClient.ViewModels;

public partial class CombatParticipantsViewModel : ObservableObject

{

private INavigation \_navigation;

[ObservableProperty] private ObservableCollection<PlayerCharacter> availableCharacters = new();

[ObservableProperty] private ObservableCollection<Enemy> availableEnemies = new();

[ObservableProperty] private ObservableCollection<NPC> availableNPCs = new();

[ObservableProperty] private Combat combat;

[ObservableProperty] private bool isEditingParticipant = false;

[ObservableProperty] private ObservableCollection<CombatParticipant> participants = new();

[ObservableProperty] private object selectedEntityToAdd = new();

[ObservableProperty] private CombatParticipant selectedParticipant;

[ObservableProperty] private ParticipantType selectedParticipantType = ParticipantType.Player;

public CombatParticipantsViewModel(Combat \_combat)

{

combat = \_combat ?? throw new ArgumentNullException(nameof(\_combat));

Console.WriteLine($"CombatParticipantsViewModel initialized with Combat ID: {\_combat?.Id}");

combat.Participants ??= new ObservableCollection<CombatParticipant>();

LoadData();

}

public ObservableCollection<ParticipantType> ParticipantTypes { get; } = new ObservableCollection<ParticipantType>

{

ParticipantType.Player,

ParticipantType.Npc,

ParticipantType.Enemy

};

public void LoadData()

{

try

{

if (combat == null)

{

Console.WriteLine("Combat is null in LoadData");

Participants = new ObservableCollection<CombatParticipant>();

return;

}

Participants = combat.Participants != null

? new ObservableCollection<CombatParticipant>(combat.Participants)

: new ObservableCollection<CombatParticipant>();

Console.WriteLine($"Loaded {Participants.Count} participants");

LoadAvailableEntities();

}

catch (Exception ex)

{

Console.WriteLine($"Error loading data: {ex.Message}");

Participants = new ObservableCollection<CombatParticipant>();

}

}

private void LoadAvailableEntities()

{

try

{

var id = Combat.CampaignId.ToString();

availableCharacters =

ApiHelper.Get<ObservableCollection<PlayerCharacter>>(

$"PlayerCharacter?campaignId={combat.CampaignId.ToString()}");

Console.WriteLine($"Loaded {availableCharacters.Count} PlayerCharacters");

availableNPCs = ApiHelper.Get<ObservableCollection<NPC>>("NPC");

Console.WriteLine($"Loaded {availableNPCs.Count} NPCs");

availableEnemies = ApiHelper.Get<ObservableCollection<Enemy>>("Enemy");

Console.WriteLine($"Loaded {availableEnemies.Count} Enemies");

}

catch (Exception ex)

{

Console.WriteLine($"Error loading available entities: {ex.Message}");

availableCharacters = new ObservableCollection<PlayerCharacter>();

availableNPCs = new ObservableCollection<NPC>();

availableEnemies = new ObservableCollection<Enemy>();

}

}

[RelayCommand]

public async Task AddParticipant(object entity)

{

if (entity == null)

{

Console.WriteLine("AddParticipant: Entity is null");

return;

}

CombatParticipant participant = null;

string successMessage = "";

switch (entity)

{

case PlayerCharacter character:

participant = CreateParticipantFromCharacter(character);

successMessage = $"Персонаж {character.Name} добавлен в бой";

break;

case NPC npc:

participant = CreateParticipantFromNPC(npc);

successMessage = $"NPC {npc.Name} добавлен в бой";

break;

case Enemy enemy:

participant = CreateParticipantFromEnemy(enemy);

successMessage = $"Враг {enemy.Name} добавлен в бой";

break;

}

if (participant != null)

{

try

{

var json = JsonConvert.SerializeObject(participant);

var success = ApiHelper.Post<CombatParticipant>(json, "CombatParticipant");

if (success)

{

var result = ApiHelper.Get<CombatParticipant>("CombatParticipant", participant.Id);

if (result != null)

{

Participants.Add(result);

combat.Participants.Add(result);

await Shell.Current.DisplayAlert("Успех", successMessage, "OK");

}

else

{

Console.WriteLine($"Failed to retrieve CombatParticipant with ID {participant.Id}");

}

}

else

{

Console.WriteLine("API Post returned null");

}

}

catch (Exception ex)

{

Console.WriteLine($"Error adding participant: {ex.Message}");

await Shell.Current.DisplayAlert("Ошибка", $"Не удалось добавить участника: {ex.Message}", "OK");

}

}

LoadData();

}

private CombatParticipant CreateParticipantFromCharacter(PlayerCharacter character)

{

return new CombatParticipant

{

Name = character.Name,

CurrentHitPoints = character.CurrentHitPoints,

MaxHitPoints = character.MaxHitPoints,

ArmorClass = character.ArmorClass,

Initiative = 0,

IsActive = false,

Type = ParticipantType.Player,

SourceId = character.Id,

CombatId = combat.Id

};

}

private CombatParticipant CreateParticipantFromNPC(NPC npc)

{

return new CombatParticipant

{

Name = npc.Name,

CurrentHitPoints = npc.HitPoints,

MaxHitPoints = npc.HitPoints,

ArmorClass = npc.ArmorClass,

Initiative = 0,

IsActive = false,

Type = ParticipantType.Npc,

SourceId = npc.Id,

CombatId = combat.Id

};

}

private CombatParticipant CreateParticipantFromEnemy(Enemy enemy)

{

return new CombatParticipant

{

Name = enemy.Name,

CurrentHitPoints = enemy.CurrentHitPoints ?? 0,

MaxHitPoints = enemy.CurrentHitPoints ?? 0,

ArmorClass = enemy.ArmorClass ?? 10,

Initiative = 0,

IsActive = false,

Type = ParticipantType.Enemy,

SourceId = enemy.Id,

CombatId = combat.Id

};

}

[RelayCommand]

public void RemoveParticipant(CombatParticipant participant)

{

if (participant != null)

{

try

{

ApiHelper.Delete<CombatParticipant>("CombatParticipant", participant.Id);

Participants.Remove(participant);

combat.Participants.Remove(participant);

Console.WriteLine($"Removed participant: {participant.Name}");

}

catch (Exception ex)

{

Console.WriteLine($"Error removing participant: {ex.Message}");

}

}

else

{

Console.WriteLine("RemoveParticipant: Participant is null");

}

}

[RelayCommand]

public void UpdateParticipant(CombatParticipant participant)

{

if (participant != null)

{

try

{

var json = JsonConvert.SerializeObject(participant);

ApiHelper.Put<CombatParticipant>(json, "CombatParticipant", participant.Id);

Console.WriteLine($"Updated participant: {participant.Name}");

}

catch (Exception ex)

{

Console.WriteLine($"Error updating participant: {ex.Message}");

}

}

else

{

Console.WriteLine("UpdateParticipant: Participant is null");

}

}

[RelayCommand]

public async Task OpenNPCCreation()

{

await Shell.Current.Navigation.PushAsync(new CreateNPCPage());

}

[RelayCommand]

public async Task OpenEnemyCreation()

{

await Shell.Current.Navigation.PushAsync(new CreateEnemyPage());

}

[RelayCommand]

public async Task EditNPC(NPC npc)

{

if (npc == null) return;

var navigationParameter = new ShellNavigationQueryParameters

{

{ "NPC", npc }

};

await Shell.Current.GoToAsync(nameof(CreateNPCPage), navigationParameter);

}

[RelayCommand]

public async Task EditEnemy(Enemy enemy)

{

if (enemy == null) return;

var navigationParameter = new Dictionary<string, object>

{

{ "Enemy", enemy }

};

await Shell.Current.GoToAsync(nameof(CreateEnemyPage), navigationParameter);

}

[RelayCommand]

public void EditParticipant(CombatParticipant participant)

{

if (participant == null) return;

SelectedParticipant = new CombatParticipant

{

Id = participant.Id,

Name = participant.Name,

Initiative = participant.Initiative,

CurrentHitPoints = participant.CurrentHitPoints,

MaxHitPoints = participant.MaxHitPoints,

ArmorClass = participant.ArmorClass,

Type = participant.Type,

SourceId = participant.SourceId,

CombatId = participant.CombatId

};

IsEditingParticipant = true;

}

[RelayCommand]

public void SaveParticipantChanges()

{

if (SelectedParticipant == null) return;

var json = JsonConvert.SerializeObject(SelectedParticipant);

var result = ApiHelper.Put<CombatParticipant>(json, "CombatParticipant", SelectedParticipant.Id);

if (result)

{

var index = Participants.FirstOrDefault(p => p.Id == SelectedParticipant.Id);

if (index != null)

{

var participantIndex = Participants.IndexOf(index);

if (participantIndex != -1)

{

Participants[participantIndex] = SelectedParticipant;

}

}

IsEditingParticipant = false;

SelectedParticipant = null;

LoadData();

}

}

[RelayCommand]

public void DeleteParticipant(CombatParticipant participant)

{

if (participant == null) return;

var result = ApiHelper.Delete<CombatParticipant>("CombatParticipant", participant.Id);

if (result)

{

Participants.Remove(participant);

combat.Participants.Remove(participant);

}

}

[RelayCommand]

private void CancelEdit()

{

IsEditingParticipant = false;

SelectedParticipant = null;

}

[RelayCommand]

public void UpdateParticipantHealth(CombatParticipant participant)

{

if (participant == null) return;

var json = JsonConvert.SerializeObject(participant);

var result = ApiHelper.Put<CombatParticipant>(json, "CombatParticipant/health", participant.Id);

if (result)

{

LoadData();

}

}

[RelayCommand]

public void DeleteNPC(NPC npc)

{

if (npc == null) return;

var result = ApiHelper.Delete<NPC>("NPC", npc.Id);

if (result)

{

AvailableNPCs.Remove(npc);

}

}

[RelayCommand]

public void DeleteEnemy(Enemy enemy)

{

if (enemy == null) return;

var result = ApiHelper.Delete<Enemy>("Enemy", enemy.Id);

if (result)

{

AvailableEnemies.Remove(enemy);

}

}

}

61. ViewModels\CombatViewModel.cs

﻿using System.Collections.ObjectModel;

using System.Text.Json;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

namespace DnDClient.ViewModels;

public partial class CombatViewModel : ObservableObject, IAsyncDisposable

{

private CancellationTokenSource? \_cts;

private Task? \_listenTask;

[ObservableProperty] private int attackDamage;

[ObservableProperty] private Combat combat;

[ObservableProperty] private ObservableCollection<CombatLog> combatLogs = new();

[ObservableProperty] private bool isPlayerTurn;

[ObservableProperty] private bool masterMode;

[ObservableProperty] private ObservableCollection<CombatLog> pendingLogs = new();

[ObservableProperty] private string selectedActionType;

[ObservableProperty] private Attack selectedAttack;

[ObservableProperty] private CombatParticipant selectedEnemy;

[ObservableProperty] private CombatParticipant selectedNpc;

[ObservableProperty] private CombatParticipant selectedTarget;

public CombatViewModel(Combat \_combat, bool \_masterMode = false)

{

if (\_combat != null)

{

Combat = \_combat;

MasterMode = \_masterMode;

IsPlayerTurn = !MasterMode && Combat.CurrentParticipant.Type == ParticipantType.Player;

CombatLogs = ApiHelper.Get<ObservableCollection<CombatLog>>("CombatLog");

combat.CombatLogs = CombatLogs;

}

}

public async ValueTask DisposeAsync()

{

await DisconnectAsync();

}

public async Task ConnectAsync()

{

if (Combat == null) return;

\_cts = new CancellationTokenSource();

var client = new HttpClient();

var stream = await client.GetStreamAsync($"http://localhost:5000/api/combat/stream/{Combat.Id}", \_cts.Token);

var reader = new StreamReader(stream);

\_listenTask = Task.Run(async () =>

{

while (!\_cts.Token.IsCancellationRequested)

{

var line = await reader.ReadLineAsync();

if (string.IsNullOrWhiteSpace(line)) continue;

if (!line.StartsWith("data:")) continue;

var json = line.Substring(5).Trim();

try

{

var evt = JsonSerializer.Deserialize<CombatEvent>(json);

if (evt == null) continue;

switch (evt.EventType)

{

case "PendingMove":

if (MasterMode && evt.Log != null) PendingLogs.Add(evt.Log);

break;

case "MasterConfirm":

if (evt.Combat != null) Combat = evt.Combat;

if (evt.Log != null)

{

if (!CombatLogs.Contains(evt.Log)) CombatLogs.Add(evt.Log);

PendingLogs.Remove(evt.Log);

}

break;

case "NpcMove":

case "EnemyMove":

if (evt.Log != null && !CombatLogs.Contains(evt.Log)) CombatLogs.Add(evt.Log);

if (evt.Combat != null) Combat = evt.Combat;

break;

}

}

catch (Exception ex)

{

Console.WriteLine($"Parse error: {ex.Message}");

}

}

}, \_cts.Token);

}

[RelayCommand]

public async Task SendPlayerActionAsync()

{

if (!IsPlayerTurn || Combat == null || SelectedTarget == null ||

string.IsNullOrEmpty(SelectedActionType))

return;

var log = new CombatLog

{

CombatId = Combat.Id,

Type = SelectedActionType,

SourceId = Combat.CurrentParticipant.Id,

TargetId = SelectedTarget.Id,

Damage = AttackDamage,

Message = $"Игрок {Combat.CurrentParticipant.Name} использует " +

$"{SelectedAttack?.Name ?? SelectedActionType} на {SelectedTarget.Name}" +

$" ({(SelectedActionType == "attack" ? "урон" : "лечение")}: {AttackDamage})"

};

ApiHelper.Post<CombatLog>(Serdeser.Serialize(log), "combat/player-move");

}

[RelayCommand]

public async Task ConfirmActionAsync(CombatLog log)

{

if (!MasterMode || Combat == null) return;

var request = new

{

CombatId = Combat.Id,

Combat,

Log = log

};

ApiHelper.Post<object>(Serdeser.Serialize(request), "combat/master-confirm");

}

[RelayCommand]

public async Task SendNpcActionAsync()

{

if (!MasterMode || Combat == null || SelectedNpc == null ||

string.IsNullOrEmpty(SelectedActionType))

return;

var log = new CombatLog

{

CombatId = Combat.Id,

Type = SelectedActionType,

SourceId = SelectedNpc.Id,

TargetId = SelectedTarget.Id,

Damage = AttackDamage,

Message = $"НПС {SelectedNpc.Name} использует " +

$"{SelectedAttack?.Name ?? SelectedActionType} на {SelectedTarget?.Name ?? "цель"}" +

$" ({(SelectedActionType == "attack" ? "урон" : "лечение")}: {AttackDamage})"

};

ApiHelper.Post<CombatLog>(Serdeser.Serialize(log), "combat/npc-move");

}

[RelayCommand]

public async Task SendEnemyActionAsync()

{

if (!MasterMode || Combat == null || SelectedEnemy == null ||

string.IsNullOrEmpty(SelectedActionType))

return;

var log = new CombatLog

{

CombatId = Combat.Id,

Type = SelectedActionType,

SourceId = SelectedEnemy.Id,

TargetId = SelectedTarget.Id,

Damage = AttackDamage,

Message = $"Враг {SelectedEnemy.Name} использует " +

$"{SelectedAttack?.Name ?? SelectedActionType} на {SelectedTarget?.Name ?? "цель"}" +

$" ({(SelectedActionType == "attack" ? "урон" : "лечение")}: {AttackDamage})"

};

ApiHelper.Post<CombatLog>(Serdeser.Serialize(log), "combat/enemy-move");

}

public async Task DisconnectAsync()

{

\_cts?.Cancel();

if (\_listenTask != null) await \_listenTask;

\_cts?.Dispose();

}

}

public class CombatEvent

{

public string EventType { get; set; }

public Combat Combat { get; set; }

public CombatLog Log { get; set; }

}

62. ViewModels\CreateEnemyViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using Newtonsoft.Json;

namespace DnDClient.ViewModels;

public partial class CreateEnemyViewModel : ObservableObject, IQueryAttributable

{

[ObservableProperty] private string actionButtonText = "Создать";

[ObservableProperty] private int? armorClass;

[ObservableProperty] private ObservableCollection<Attack> attacks = new();

[ObservableProperty] private string challengeRating = string.Empty;

[ObservableProperty] private string description = string.Empty;

[ObservableProperty] private int? hitPoints;

[ObservableProperty] private Guid id;

[ObservableProperty] private bool isEditMode;

[ObservableProperty] private string name = string.Empty;

[ObservableProperty] private ObservableCollection<SpecialAbility> specialAbilities = new();

[ObservableProperty] private string title = "Создать врага";

[ObservableProperty] private string type = string.Empty;

public void ApplyQueryAttributes(IDictionary<string, object> query)

{

if (query.TryGetValue("Enemy", out var enemy))

{

if (enemy is Enemy existingEnemy)

{

IsEditMode = true;

Title = "Редактировать врага";

ActionButtonText = "Сохранить";

Id = existingEnemy.Id;

Name = existingEnemy.Name;

Type = existingEnemy.Type ?? string.Empty;

Description = existingEnemy.Description ?? string.Empty;

HitPoints = existingEnemy.CurrentHitPoints;

ArmorClass = existingEnemy.ArmorClass;

ChallengeRating = existingEnemy.ChallengeRating ?? string.Empty;

Attacks = new ObservableCollection<Attack>(existingEnemy.Attacks ?? new ObservableCollection<Attack>());

SpecialAbilities = new ObservableCollection<SpecialAbility>(existingEnemy.SpecialAbilities ??

new ObservableCollection<SpecialAbility>());

}

}

}

[RelayCommand]

private async Task SaveAsync()

{

try

{

var enemy = new Enemy

{

Id = IsEditMode ? Id : Guid.NewGuid(),

Name = Name,

Type = Type,

Description = Description,

CurrentHitPoints = HitPoints,

ArmorClass = ArmorClass,

ChallengeRating = ChallengeRating,

Attacks = Attacks,

SpecialAbilities = SpecialAbilities

};

var json = JsonConvert.SerializeObject(enemy);

bool success;

if (IsEditMode)

{

success = ApiHelper.Put<Enemy>(json, "Enemy", enemy.Id);

}

else

{

success = ApiHelper.Post<Enemy>(json, "Enemy");

}

if (success)

{

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

}

else

{

var action = IsEditMode ? "обновить" : "создать";

await Shell.Current.DisplayAlert("Ошибка", $"Не удалось {action} врага", "OK");

}

}

catch (Exception ex)

{

await Shell.Current.DisplayAlert("Ошибка", ex.Message, "OK");

}

}

[RelayCommand]

private void AddAttack()

{

Attacks.Add(new Attack());

}

[RelayCommand]

private void DelAttack(Attack attack)

{

if (attack != null)

{

if (attack.Id != Guid.Empty)

{

ApiHelper.Delete<Attack>("Attack", attack.Id);

}

Attacks.Remove(attack);

}

}

[RelayCommand]

private void AddAbility()

{

SpecialAbilities.Add(new SpecialAbility());

}

[RelayCommand]

private void DelAbility(SpecialAbility ability)

{

if (ability != null)

{

if (ability.Id != Guid.Empty)

{

ApiHelper.Delete<SpecialAbility>("SpecialAbility", ability.Id);

}

SpecialAbilities.Remove(ability);

}

}

}

63. ViewModels\CreateNPCViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using Newtonsoft.Json;

namespace DnDClient.ViewModels;

public partial class CreateNPCViewModel : ObservableObject

{

[ObservableProperty] private string actionButtonText = "Создать";

[ObservableProperty] private int armorClass;

[ObservableProperty] private ObservableCollection<Attack> attacks = new();

[ObservableProperty] private string bonds = string.Empty;

[ObservableProperty] private string description = string.Empty;

[ObservableProperty] private string flaws = string.Empty;

[ObservableProperty] private int hitPoints;

[ObservableProperty] private Guid id;

[ObservableProperty] private string ideals = string.Empty;

[ObservableProperty] private bool isEditMode;

[ObservableProperty] private string name = string.Empty;

[ObservableProperty] private string occupation = string.Empty;

[ObservableProperty] private string personalityTraits = string.Empty;

[ObservableProperty] private string race = string.Empty;

[ObservableProperty] private string role = string.Empty;

[ObservableProperty] private string title = "Создать NPC";

public void ApplyQueryAttributes(IDictionary<string, object> query)

{

if (query.TryGetValue("NPC", out var npc))

{

if (npc is NPC existingNpc)

{

IsEditMode = true;

Title = "Редактировать NPC";

ActionButtonText = "Сохранить";

Id = existingNpc.Id;

Name = existingNpc.Name;

Description = existingNpc.Description;

Race = existingNpc.Race;

Occupation = existingNpc.Occupation;

HitPoints = existingNpc.HitPoints;

ArmorClass = existingNpc.ArmorClass;

PersonalityTraits = existingNpc.PersonalityTraits;

Ideals = existingNpc.Ideals;

Bonds = existingNpc.Bonds;

Flaws = existingNpc.Flaws;

Attacks = new ObservableCollection<Attack>(existingNpc.Attacks ?? new ObservableCollection<Attack>());

}

}

}

[RelayCommand]

private async Task SaveAsync()

{

try

{

var npc = new NPC

{

Id = IsEditMode ? Id : Guid.NewGuid(),

Name = Name,

Description = Description,

Race = Race,

Occupation = Occupation,

HitPoints = HitPoints,

ArmorClass = ArmorClass,

PersonalityTraits = PersonalityTraits,

Ideals = Ideals,

Bonds = Bonds,

Flaws = Flaws,

Attacks = Attacks

};

var json = JsonConvert.SerializeObject(npc);

bool success;

if (IsEditMode)

{

success = ApiHelper.Put<NPC>(json, "NPC", npc.Id);

}

else

{

success = ApiHelper.Post<NPC>(json, "NPC");

}

if (success)

{

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

}

else

{

var action = IsEditMode ? "обновить" : "создать";

await Shell.Current.DisplayAlert("Ошибка", $"Не удалось {action} NPC", "OK");

}

}

catch (Exception ex)

{

await Shell.Current.DisplayAlert("Ошибка", ex.Message, "OK");

}

}

[RelayCommand]

private void AddAttack()

{

Attacks.Add(new Attack());

}

[RelayCommand]

private void DelAttack(Attack attack)

{

if (attack != null)

{

if (attack.Id != Guid.Empty)

{

ApiHelper.Delete<Attack>("Attack", attack.Id);

}

Attacks.Remove(attack);

}

}

}

64. ViewModels\EntityManagementViewModel.cs

﻿using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

using Newtonsoft.Json;

namespace DnDClient.ViewModels;

public partial class EntityManagementViewModel : ObservableObject

{

[ObservableProperty] private ObservableCollection<Enemy> enemies = new();

[ObservableProperty] private bool isEditingEnemy = false;

[ObservableProperty] private bool isEditingNPC = false;

[ObservableProperty] private Enemy newEnemy = new Enemy();

[ObservableProperty] private NPC newNPC = new NPC();

// [ObservableProperty]

// private Guid campaignId;

[ObservableProperty] private ObservableCollection<NPC> npcs = new();

[ObservableProperty] private Enemy selectedEnemy;

[ObservableProperty] private NPC selectedNPC;

public EntityManagementViewModel()

{

//CampaignId = campaignId;

LoadData();

}

public void LoadData()

{

Npcs = new ObservableCollection<NPC>(

ApiHelper.Get<List<NPC>>("NPC") ?? new List<NPC>());

Enemies = new ObservableCollection<Enemy>(

ApiHelper.Get<List<Enemy>>("Enemy") ?? new List<Enemy>());

}

// [RelayCommand]

// public async Task AddNpcToCampaignAsync(NPC npc)

// {

// if (npc == null) return;

//

// try

// {

// var result = ApiHelper.Post<string>($"", $"Campaign/{CampaignId}/add\_npc/{npc.Id}");

//

// if (result != null)

// {

// await Shell.Current.DisplayAlert("Успех", $"NPC {npc.Name} добавлен в кампанию", "OK");

// }

// else

// {

// await Shell.Current.DisplayAlert("Ошибка", "Не удалось добавить NPC в кампанию", "OK");

// }

// }

// catch (Exception ex)

// {

// await Shell.Current.DisplayAlert("Ошибка", $"Ошибка при добавлении NPC: {ex.Message}", "OK");

// }

// }

//

// [RelayCommand]

// public async Task AddEnemyToCampaignAsync(Enemy enemy)

// {

// if (enemy == null) return;

//

// try

// {

// var result = ApiHelper.Post<string>($"", $"Campaign/{CampaignId}/add\_enemy/{enemy.Id}");

//

// if (result != null)

// {

// await Shell.Current.DisplayAlert("Успех", $"Враг {enemy.Name} добавлен в кампанию", "OK");

// }

// else

// {

// await Shell.Current.DisplayAlert("Ошибка", "Не удалось добавить врага в кампанию", "OK");

// }

// }

// catch (Exception ex)

// {

// await Shell.Current.DisplayAlert("Ошибка", $"Ошибка при добавлении врага: {ex.Message}", "OK");

// }

// }

// NPC CRUD operations

[RelayCommand]

public void CreateNPC()

{

var json = JsonConvert.SerializeObject(NewNPC);

var success = ApiHelper.Post<NPC>(json, "NPC");

if (success)

{

var result = ApiHelper.Get<NPC>("NPC", NewNPC.Id);

if (result != null)

{

Npcs.Add(result);

NewNPC = new NPC();

}

}

}

[RelayCommand]

public void UpdateNPC()

{

if (SelectedNPC != null)

{

var json = JsonConvert.SerializeObject(SelectedNPC);

ApiHelper.Put<NPC>(json, "NPC", SelectedNPC.Id);

IsEditingNPC = false;

}

}

[RelayCommand]

public void EditNPC(NPC npc)

{

if (npc == null) return;

SelectedNPC = npc;

NewNPC = new NPC

{

Id = npc.Id,

Name = npc.Name,

Description = npc.Description,

HitPoints = npc.HitPoints,

ArmorClass = npc.ArmorClass,

};

IsEditingNPC = true;

}

[RelayCommand]

public void SaveNPCChanges()

{

if (NewNPC == null) return;

var json = JsonConvert.SerializeObject(NewNPC);

var result = ApiHelper.Put<NPC>(json, "NPC", NewNPC.Id);

if (result)

{

var index = Npcs.IndexOf(SelectedNPC);

if (index != -1)

{

Npcs[index] = NewNPC;

}

IsEditingNPC = false;

NewNPC = new NPC();

LoadData();

}

}

[RelayCommand]

public void DeleteNPC(NPC npc)

{

if (npc == null) return;

var result = ApiHelper.Delete<NPC>("NPC", npc.Id);

if (result)

{

Npcs.Remove(npc);

}

}

// Enemy CRUD operations

[RelayCommand]

public void CreateEnemy()

{

var json = JsonConvert.SerializeObject(NewEnemy);

var success = ApiHelper.Post<Enemy>(json, "Enemy");

if (success)

{

var result = ApiHelper.Get<Enemy>("Enemy", NewEnemy.Id);

if (result != null)

{

Enemies.Add(result);

NewEnemy = new Enemy();

}

}

}

[RelayCommand]

public void UpdateEnemy()

{

if (SelectedEnemy != null)

{

var json = JsonConvert.SerializeObject(SelectedEnemy);

ApiHelper.Put<Enemy>(json, "Enemy", SelectedEnemy.Id);

IsEditingEnemy = false;

}

}

[RelayCommand]

public void EditEnemy(Enemy enemy)

{

if (enemy == null) return;

SelectedEnemy = enemy;

NewEnemy = new Enemy

{

Id = enemy.Id,

Name = enemy.Name,

Description = enemy.Description,

CurrentHitPoints = enemy.CurrentHitPoints,

ArmorClass = enemy.ArmorClass,

SpecialAbilities =

new ObservableCollection<SpecialAbility>(enemy.SpecialAbilities ??

new ObservableCollection<SpecialAbility>())

};

IsEditingEnemy = true;

}

[RelayCommand]

public void SaveEnemyChanges()

{

if (NewEnemy == null) return;

var json = JsonConvert.SerializeObject(NewEnemy);

var result = ApiHelper.Put<Enemy>(json, "Enemy", NewEnemy.Id);

if (result)

{

var index = Enemies.IndexOf(SelectedEnemy);

if (index != -1)

{

Enemies[index] = NewEnemy;

}

IsEditingEnemy = false;

NewEnemy = new Enemy();

LoadData();

}

}

[RelayCommand]

public void DeleteEnemy(Enemy enemy)

{

if (enemy == null) return;

var result = ApiHelper.Delete<Enemy>("Enemy", enemy.Id);

if (result)

{

Enemies.Remove(enemy);

}

}

[RelayCommand]

private void CancelEdit()

{

if (IsEditingNPC)

{

IsEditingNPC = false;

NewNPC = new NPC();

}

if (IsEditingEnemy)

{

IsEditingEnemy = false;

NewEnemy = new Enemy();

}

}

public void AddSpecialAbilityToEnemy(SpecialAbility ability)

{

if (IsEditingEnemy && NewEnemy != null)

{

if (NewEnemy.SpecialAbilities == null)

NewEnemy.SpecialAbilities = new ObservableCollection<SpecialAbility>();

NewEnemy.SpecialAbilities.Add(ability);

}

}

public void RemoveSpecialAbilityFromEnemy(SpecialAbility ability)

{

if (IsEditingEnemy && NewEnemy?.SpecialAbilities != null)

{

NewEnemy.SpecialAbilities.Remove(ability);

}

}

}

65. ViewModels\MainViewModel.cs

﻿using CommunityToolkit.Mvvm.ComponentModel;

using DnDClient.Models;

using DnDClient.Services;

namespace DnDClient.ViewModels;

public partial class MainViewModel : ObservableObject

{

[ObservableProperty]

string welcome;

public MainViewModel()

{

string user = Preferences.Get("current\_user", "default");

Welcome = "Добро пожаловать, " + user;

}

}

66. ViewModels\ProfileViewModel.cs

using CommunityToolkit.Mvvm.ComponentModel;

using CommunityToolkit.Mvvm.Input;

using DnDClient.Models;

using DnDClient.Services;

namespace DnDClient.ViewModels;

public partial class ProfileViewModel : ObservableObject

{

[ObservableProperty] private string checkNewPassword;

[ObservableProperty] private User currentUser;

[ObservableProperty] private string newPassword;

[ObservableProperty] private string newUsername;

[ObservableProperty] private string userId;

public ProfileViewModel()

{

userId = Preferences.Get("current\_user\_id", string.Empty);

if (userId != string.Empty)

{

currentUser = ApiHelper.Get<User>("User", new Guid(userId));

}

}

[RelayCommand]

public void ChangeUsername()

{

if (newUsername != String.Empty)

{

try

{

currentUser.Username = newUsername;

if (ApiHelper.Put<User>(Serdeser.Serialize(currentUser), "User", new Guid(userId)))

{

Shell.Current.DisplayAlert("Успех", "Вы успешно сменили имя пользователя", "OK");

}

else

{

Shell.Current.DisplayAlert("Ошибка", "Не удалось поменять имя пользователя", "OK");

}

}

catch (Exception e)

{

Shell.Current.DisplayAlert("Ошибка", $"{e}", "OK");

}

}

else

{

Shell.Current.DisplayAlert("Ошибка", "Вы не ввели новое имя пользователя", "OK");

}

}

[RelayCommand]

public void ChangePassword()

{

if (newPassword == checkNewPassword && newPassword != String.Empty && checkNewPassword != String.Empty)

{

try

{

currentUser.PasswordHash = BCrypt.Net.BCrypt.HashPassword(newPassword);

if (ApiHelper.Put<User>(Serdeser.Serialize(currentUser), "User", new Guid(userId)))

{

Shell.Current.DisplayAlert("Успех", "Вы успешно сменили пароль", "OK");

}

else

{

Shell.Current.DisplayAlert("Ошибка", "Не удалось поменять пароль", "OK");

}

}

catch (Exception e)

{

Shell.Current.DisplayAlert("Ошибка", $"{e}", "OK");

}

}

else

{

Shell.Current.DisplayAlert("Ошибка", "Пароли не совпадают", "OK");

}

}

}

67. Views\AuthPage.xaml

﻿<?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="DnDClient.Views.AuthPage">

<ContentPage.Content>

<VerticalStackLayout Padding="25" Spacing="10">

<Entry Placeholder="Имя пользователя" Text="{Binding Email}" />

<Entry Placeholder="Пароль" IsPassword="True" Text="{Binding Password}" />

<Button Text="Вход" Command="{Binding LoginCommand}" />

<Button Text="Регистрация" Command="{Binding RegisterCommand}" />

<Label Text="{Binding Message}" />

</VerticalStackLayout>

</ContentPage.Content>

</ContentPage>

68. Views\AuthPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class AuthPage : ContentPage

{

public AuthPage()

{

InitializeComponent();

BindingContext = new AuthViewModel();

}

}

69. Views\CampaignListPage.xaml

﻿<?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"

xmlns:cards="clr-namespace:DnDClient.Views.Cards"

xmlns:vm="using:DnDClient.ViewModels"

x:Class="DnDClient.Views.CampaignListPage">

<ContentPage.Content>

<VerticalStackLayout>

<Button Text="Добавить кампанию" Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignListViewModel}}, Path=AddCampaignCommand}"/>

<CollectionView ItemsSource="{Binding Campaigns}">

<CollectionView.ItemTemplate>

<DataTemplate>

<cards:CampCard BindingContext="{Binding .}" />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</VerticalStackLayout>

</ContentPage.Content>

</ContentPage>

70. Views\CampaignListPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CampaignListPage : ContentPage

{

public CampaignListPage()

{

InitializeComponent();

BindingContext = new CampaignListViewModel(Navigation);

}

}

71. Views\CampaignPage.xaml

﻿<?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"

xmlns:views="clr-namespace:DnDClient.Views.Cards"

xmlns:conv="clr-namespace:DnDClient.Converters"

xmlns:local="clr-namespace:DnDClient.Views"

x:Class="DnDClient.Views.CampaignPage">

<ContentPage.Resources>

<conv:BooleanNegationConverter x:Key="BooleanNegationConverter" />

</ContentPage.Resources>

<ContentPage.Content>

<ScrollView>

<Grid Margin="10" Padding="10">

<!-- Режим мастера -->

<StackLayout IsVisible="{Binding MasterMode}">

<Label Text="Мастер: управление кампанией" FontAttributes="Bold" FontSize="20" />

<Entry Text="{Binding Campaign.Name}" Placeholder="Название кампании" />

<Editor Text="{Binding Campaign.Description}" Placeholder="Описание кампании" />

<Button Text="Сохранить изменения" Command="{Binding SaveCampaignCommand}" />

<Button Text="Управление сущностями" Command="{Binding ManageEntitiesCommand}" />

<Label Text="Игроки кампании:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Players}" Margin="0,0,0,10">

<CollectionView.ItemTemplate>

<DataTemplate>

<views:CharCard />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

<Label Text="Добавить персонажа:" FontAttributes="Bold" />

<Picker ItemsSource="{Binding AvailableCharacters}"

ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedCharacterToAdd, Mode=TwoWay}" />

<Button Text="Добавить в кампанию" Command="{Binding AddPlayerToCampaignCommand}" />

<Grid RowDefinitions="{OnPlatform Default='\*,\*,\*,\*,\*,\*', WinUI='\*,\*,\*', MacCatalyst='\*,\*,\*'}"

ColumnDefinitions="{OnPlatform Default='\*', WinUI='\*,\*', MacCatalyst='\*,\*'}">

<Frame Margin="5" Padding="10" BorderColor="DarkGreen" Grid.Row="0" Grid.Column="0">

<StackLayout>

<Label Text="Добавить сюжет" FontAttributes="Bold" />

<Entry Placeholder="Название" Text="{Binding NewStoryName, Mode=TwoWay}" />

<Editor Placeholder="Описание" Text="{Binding NewStoryDescription, Mode=TwoWay}" />

<Button Text="Добавить" Command="{Binding AddStoryCommand}" />

</StackLayout>

</Frame>

<StackLayout Grid.Row="1" Grid.Column="0">

<Label Text="Сюжеты кампании:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Stories}" Grid.Row="2" Grid.Column="0">

<CollectionView.ItemTemplate>

<DataTemplate>

<views:StoryCard />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

<Frame Margin="5" Padding="10" BorderColor="DarkBlue"

Grid.Row="{OnPlatform Default='3', WinUI='0', MacCatalyst='0'}"

Grid.Column="1">

<StackLayout>

<Label Text="Добавить бой" FontAttributes="Bold" />

<Entry Placeholder="Название боя" Text="{Binding NewCombatName, Mode=TwoWay}" />

<Button Text="Добавить" Command="{Binding AddCombatCommand}" />

</StackLayout>

</Frame>

<StackLayout Grid.Row="{OnPlatform Default='4', WinUI='1', MacCatalyst='1'}"

Grid.Column="1">

<Label Text="Бои кампании:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Combats}"

Grid.Row="{OnPlatform Default='5', WinUI='2', MacCatalyst='2'}"

Grid.Column="1">

<CollectionView.ItemTemplate>

<DataTemplate>

<StackLayout>

<views:CombatCard />

<Button Text="👥 Управление участниками"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CampaignPage}},

Path=BindingContext.ManageCombatParticipantsCommand}"

CommandParameter="{Binding}"

Margin="5"

BackgroundColor="#FF9800"

TextColor="White" />

</StackLayout>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Grid>

</StackLayout>

<!-- Обычный режим -->

<StackLayout IsVisible="{Binding MasterMode, Converter={StaticResource BooleanNegationConverter}}">

<Label Text="Кампания" FontAttributes="Bold" FontSize="20" />

<Label Text="{Binding Campaign.Name}" FontSize="18" />

<Label Text="{Binding Campaign.Description}" FontSize="14" />

<Label Text="Игроки:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Players}" Margin="0,0,0,10">

<CollectionView.ItemTemplate>

<DataTemplate>

<views:CharCard />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

<Label Text="Сюжеты:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Stories}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="5" Padding="10" BorderColor="Gray">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<Label Text="{Binding Description}" FontSize="12" />

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

<Label Text="Бои:" FontAttributes="Bold" Margin="0,10,0,0" />

<CollectionView ItemsSource="{Binding Combats}">

<CollectionView.ItemTemplate>

<DataTemplate>

<views:CombatCard />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Grid>

</ScrollView>

</ContentPage.Content>

</ContentPage>

72. Views\CampaignPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.Models;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CampaignPage : ContentPage

{

public CampaignPage(Campaign campaign)

{

InitializeComponent();

BindingContext = new CampaignViewModel(campaign);

}

}

73. Views\Cards\CampCard.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<ContentView xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:vm="clr-namespace:DnDClient.ViewModels"

x:Class="DnDClient.Views.Cards.CampCard">

<Frame

CornerRadius="10"

Margin="10"

Padding="15"

BorderColor="Purple"

Background="{AppThemeBinding Light=LightGray, Dark=#454545}"

HasShadow="True">

<Frame.GestureRecognizers>

<TapGestureRecognizer

Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignListViewModel}}, Path=TapCardCommand}"

CommandParameter="{Binding .}" />

</Frame.GestureRecognizers>

<Grid RowDefinitions="\*,\*" ColumnDefinitions="5\*,5\*,2\*">

<Label Grid.Row="0" Grid.ColumnSpan="2" Text="{Binding Name, StringFormat='Название: {0}'}" FontSize="25"

HorizontalTextAlignment="Center">

</Label>

<Label Grid.Row="1" Grid.Column="0" Text="{Binding Master.Username, StringFormat='Мастер: {0}'}"></Label>

<Label Grid.Row="1" Grid.Column="1" Text="{Binding PlayerCharacters.Count, StringFormat='Игроков: {0}'}"></Label>

<Button Text="Удалить" BackgroundColor="Red" Grid.RowSpan="2" Grid.Column="2"

Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignListViewModel}}, Path=DelCampaignCommand}"

CommandParameter="{Binding .}" />

</Grid>

</Frame>

</ContentView>

74. Views\Cards\CampCard.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DnDClient.Views.Cards;

public partial class CampCard : ContentView

{

public CampCard()

{

InitializeComponent();

}

}

75. Views\Cards\CharCard.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<ContentView xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:vm="clr-namespace:DnDClient.ViewModels"

x:Class="DnDClient.Views.Cards.CharCard">

<Frame

CornerRadius="10"

Margin="10"

Padding="15"

BorderColor="Purple"

Background="{AppThemeBinding Light=LightGray, Dark=#454545}"

HasShadow="True">

<Frame.GestureRecognizers>

<TapGestureRecognizer

Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CharactersViewModel}}, Path=TapCardCommand}"

CommandParameter="{Binding .}" />

</Frame.GestureRecognizers>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!-- Имя персонажа -->

<Label

Grid.Row="0"

Grid.Column="0"

Grid.ColumnSpan="2"

Text="{Binding Name}"

FontSize="24"

FontAttributes="Bold"

HorizontalOptions="Center"

Margin="0,0,0,10" />

<!-- Класс и уровень -->

<Label

Grid.Row="1"

Grid.Column="0"

Text="{Binding ClassType, StringFormat='Class: {0}'}"

FontSize="16" />

<Label

Grid.Row="1"

Grid.Column="1"

Text="{Binding Level, StringFormat='Level: {0}'}"

FontSize="16"

HorizontalOptions="End" />

<!-- Раса и выравнивание -->

<Label

Grid.Row="2"

Grid.Column="0"

Text="{Binding Race, StringFormat='Race: {0}'}"

FontSize="16" />

<Label

Grid.Row="2"

Grid.Column="1"

Text="{Binding Alignment, StringFormat='Alignment: {0}'}"

FontSize="16"

HorizontalOptions="End" />

<!-- Характеристики -->

<Label

Grid.Row="3"

Grid.Column="0"

Grid.ColumnSpan="2"

Text="Attributes"

FontSize="18"

FontAttributes="Bold"

Margin="0,10,0,5" />

<Grid Grid.Row="4"

Grid.Column="0"

Grid.ColumnSpan="2"

HorizontalOptions="CenterAndExpand" ColumnSpacing="20" RowSpacing="20">

<!-- Правильное определение RowDefinitions с OnPlatform -->

<Grid.RowDefinitions>

<OnPlatform x:TypeArguments="RowDefinitionCollection">

<On Platform="iOS, Android">

<RowDefinitionCollection>

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

</RowDefinitionCollection>

</On>

<On Platform="WinUI, MacCatalyst">

<RowDefinitionCollection>

<RowDefinition Height="\*" />

</RowDefinitionCollection>

</On>

</OnPlatform>

</Grid.RowDefinitions>

<!-- Правильное определение ColumnDefinitions с OnPlatform -->

<Grid.ColumnDefinitions>

<OnPlatform x:TypeArguments="ColumnDefinitionCollection">

<On Platform="iOS, Android">

<ColumnDefinitionCollection>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</ColumnDefinitionCollection>

</On>

<On Platform="WinUI, MacCatalyst">

<ColumnDefinitionCollection>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</ColumnDefinitionCollection>

</On>

</OnPlatform>

</Grid.ColumnDefinitions>

<!-- Элементы с упрощенным назначением строк/колонок -->

<Label Text="{Binding Strength, StringFormat='Сила: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=0, Android=0, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=0, Android=0, WinUI=0, MacCatalyst=0}" />

<Label Text="{Binding Dexterity, StringFormat='Ловкость: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=0, Android=0, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=1, Android=1, WinUI=1, MacCatalyst=1}" />

<Label Text="{Binding Constitution, StringFormat='Телосложение: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=1, Android=1, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=0, Android=0, WinUI=2, MacCatalyst=2}" />

<Label Text="{Binding Intelligence, StringFormat='Интеллект: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=1, Android=1, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=1, Android=1, WinUI=3, MacCatalyst=3}" />

<Label Text="{Binding Wisdom, StringFormat='Мудрость: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=2, Android=2, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=0, Android=0, WinUI=4, MacCatalyst=4}" />

<Label Text="{Binding Charisma, StringFormat='Харизма: {0}'}"

FontSize="14"

HorizontalOptions="Center"

Grid.Row="{OnPlatform iOS=2, Android=2, WinUI=0, MacCatalyst=0}"

Grid.Column="{OnPlatform iOS=1, Android=1, WinUI=5, MacCatalyst=5}" />

</Grid>

<StackLayout Orientation="Horizontal" Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="2" Padding="0,0" Margin="0,25" HorizontalOptions="Center" Spacing="20">

<Button Text="Удалить"

BackgroundColor="Red"

Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CharactersViewModel}}, Path=DeleteCommand}"

CommandParameter="{Binding .}" />

<Button Text="Скачать pdf"

BackgroundColor="Green"

Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CharactersViewModel}}, Path=ExportCommand}"

CommandParameter="{Binding .}" />

</StackLayout>

</Grid>

</Frame>

</ContentView>

76. Views\Cards\CharCard.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Input;

using DnDClient.ViewModels;

namespace DnDClient.Views.Cards;

public partial class CharCard : ContentView

{

public CharCard()

{

InitializeComponent();

}

}

77. Views\Cards\CombatCard.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<ContentView xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:vm="using:DnDClient.ViewModels"

x:Class="DnDClient.Views.Cards.CombatCard">

<Frame Padding="10" Margin="5" BorderColor="Gray" CornerRadius="10">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Label Text="{Binding Name}" FontAttributes="Bold" FontSize="Medium"/>

<Label Text="{Binding CombatCurrentRound}" FontSize="Small" Grid.Row="1"/>

<Button Text="Открыть бой" Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignViewModel}}, Path=OpenCombatCommand}" CommandParameter="{Binding .}" BackgroundColor="DarkSlateGray" TextColor="White" Grid.Row="2"></Button>

<Button Text="Меню участников боя" Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignViewModel}}, Path=ManageCombatParticipantsCommand}" CommandParameter="{Binding .}" BackgroundColor="DarkRed" TextColor="White" Grid.Row="3" Margin="0,5,0,0"/>

<Button Text="Удалить" Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignViewModel}}, Path=DeleteCombatCommand}" CommandParameter="{Binding .}" BackgroundColor="DarkRed" TextColor="White" Grid.Row="3" Margin="0,5,0,0"/>

<Grid.GestureRecognizers>

<TapGestureRecognizer Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignViewModel}}, Path=OpenCombatCommand}" CommandParameter="{Binding .}"/>

</Grid.GestureRecognizers>

</Grid>

</Frame>

</ContentView>

78. Views\Cards\CombatCard.xaml.cs

﻿namespace DnDClient.Views.Cards;

public partial class CombatCard : ContentView

{

public CombatCard()

{

InitializeComponent();

}

// public static readonly BindableProperty OpenCombatCommandProperty =

// BindableProperty.Create(

// nameof(OpenCombatCommand),

// typeof(ICommand),

// typeof(CombatCard),

// null);

//

// public ICommand OpenCombatCommand

// {

// get => (ICommand)GetValue(OpenCombatCommandProperty);

// set => SetValue(OpenCombatCommandProperty, value);

// }

}

79. Views\Cards\NPCCard.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<ContentView xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="DnDClient.Views.Cards.NPCCard">

</ContentView>

80. Views\Cards\NPCCard.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DnDClient.Views.Cards;

public partial class NPCCard : ContentView

{

public NPCCard()

{

InitializeComponent();

}

}

81. Views\Cards\StoryCard.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<ContentView xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:vm="using:DnDClient.ViewModels"

x:Class="DnDClient.Views.Cards.StoryCard">

<Frame Margin="5" Padding="10" BorderColor="Gray" CornerRadius="10">

<StackLayout>

<Entry Text="{Binding Name, Mode=TwoWay}" Placeholder="Название сюжета" Completed="OnNameCompleted"/>

<Editor Text="{Binding Description, Mode=TwoWay}" Placeholder="Описание" Completed="OnDescriptionCompleted"/>

<Button Text="Удалить" Command="{Binding Source={RelativeSource AncestorType={x:Type vm:CampaignViewModel}}, Path=DeleteStoryCommand}" CommandParameter="{Binding .}" BackgroundColor="DarkRed" TextColor="White" Margin="0,5,0,0"/>

</StackLayout>

</Frame>

</ContentView>

82. Views\Cards\StoryCard.xaml.cs

﻿using DnDClient.Models;

using DnDClient.Services;

namespace DnDClient.Views.Cards;

public partial class StoryCard : ContentView

{

public StoryCard()

{

InitializeComponent();

}

private void OnNameCompleted(object sender, EventArgs e)

{

if (BindingContext is StoryElement story)

{

var json = Serdeser.Serialize(story);

ApiHelper.Put<StoryElement>(json, "StoryElement", story.Id);

}

}

private void OnDescriptionCompleted(object sender, EventArgs e)

{

if (BindingContext is StoryElement story)

{

var json = Serdeser.Serialize(story);

ApiHelper.Put<StoryElement>(json, "StoryElement", story.Id);

}

}

}

83. Views\CharacterDetailsPage.xaml

﻿<?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"

xmlns:converters="clr-namespace:DnDClient.Converters"

xmlns:vm="clr-namespace:DnDClient.ViewModels"

x:Class="DnDClient.Views.CharacterDetailsPage">

<ContentPage.Resources>

<converters:BoolToIntConverter x:Key="BoolToIntConverter" />

<converters:BooleanNegationConverter x:Key="BooleanNegationConverter"/>

<Style x:Key="LabelStyle" TargetType="Label">

<Setter Property="FontSize" Value="{OnPlatform Default=Micro, WinUI=Small, iOS=Small, Android=Small}" />

<Setter Property="HorizontalOptions" Value="Center" />

<Setter Property="VerticalOptions" Value="Center" />

<Setter Property="TextColor" Value="{AppThemeBinding Light=Black, Dark=White}" />

</Style>

<Style x:Key="ShowLabelStyle" TargetType="Label">

<Setter Property="IsVisible" Value="{Binding EditMode, Converter={StaticResource BooleanNegationConverter}}" />

<Setter Property="FontSize" Value="{OnPlatform Default=Micro, WinUI=Small, iOS=Small, Android=Small}" />

<Setter Property="Background" Value="DimGray"></Setter>

<Setter Property="HorizontalOptions" Value="Center" />

<Setter Property="VerticalOptions" Value="Center" />

<Setter Property="TextColor" Value="{AppThemeBinding Light=Black, Dark=White}" />

</Style>

<Style x:Key="EntryStyle" TargetType="Entry">

<Setter Property="IsVisible" Value="{Binding EditMode}"></Setter>

<Setter Property="FontSize" Value="{OnPlatform Default=Small, WinUI=Medium}" />

<Setter Property="HorizontalOptions" Value="Center" />

<Setter Property="VerticalOptions" Value="Center" />

<Setter Property="Margin" Value="2" />

<Setter Property="TextColor" Value="{AppThemeBinding Light=Black, Dark=White}" />

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light=White, Dark=#2D2D2D}" />

<Setter Property="PlaceholderColor" Value="{AppThemeBinding Light=Gray, Dark=LightGray}" />

</Style>

<Style x:Key="FrameStyle" TargetType="Frame">

<Setter Property="BorderColor" Value="{AppThemeBinding Light=Black, Dark=Silver}" />

<Setter Property="CornerRadius" Value="8" />

<Setter Property="Margin" Value="{OnPlatform Default=3, WinUI=5}" />

<Setter Property="Padding" Value="5" />

<Setter Property="BackgroundColor" Value="{AppThemeBinding Light=#F5F5F5, Dark=#1E1E1E}" />

</Style>

</ContentPage.Resources>

<ContentPage.Content>

<ScrollView>

<Grid RowDefinitions="Auto,Auto,Auto,Auto,Auto" ColumnDefinitions="{OnPlatform Default='\*', WinUI='\*,\*,\*'}" Padding="{OnPlatform Default=5, WinUI=10}">

<StackLayout Orientation="Horizontal" Grid.Column="{OnPlatform Default=0, WinUI=3}" HorizontalOptions="Center">

<Label Text="Редактирование" Style="{StaticResource LabelStyle}"></Label>

<CheckBox IsChecked="{Binding EditMode}"></CheckBox>

</StackLayout>

<!-- Character Info -->

<Frame Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="{OnPlatform Default=1, WinUI=3}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto,Auto" ColumnDefinitions="{OnPlatform Default='\*,\*', WinUI='\*,\*,\*,\*'}">

<!-- Character Name -->

<StackLayout Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="{OnPlatform Default=1, WinUI=2}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.Name, Mode=TwoWay}" Placeholder="Имя персонажа" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Name}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Имя персонажа" Style="{StaticResource LabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="0" Grid.Column="{OnPlatform Default=1, WinUI=2}" Grid.ColumnSpan="{OnPlatform Default=1, WinUI=2}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.PlayerName}" Placeholder="Имя игрока" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.PlayerName}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Имя игрока" Style="{StaticResource LabelStyle}" />

</StackLayout>

<!-- Class and Race -->

<StackLayout Grid.Row="1" Grid.Column="0" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.ClassType}" Placeholder="Класс" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.ClassType}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Класс" Style="{StaticResource LabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="1" Grid.Column="{OnPlatform Default=1, WinUI=1}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.Race}" Placeholder="Раса" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Race}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Раса" Style="{StaticResource LabelStyle}" />

</StackLayout>

<!-- Background and Alignment -->

<StackLayout Grid.Row="2" Grid.Column="0" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.Background}" Placeholder="Предыстория" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Background}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Предыстория" Style="{StaticResource LabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="2" Grid.Column="{OnPlatform Default=1, WinUI=1}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.Alignment}" Placeholder="Мировоззрение" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Alignment}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Мировоззрение" Style="{StaticResource LabelStyle}" />

</StackLayout>

<!-- Experience and Level -->

<StackLayout Grid.Row="3" Grid.Column="{OnPlatform Default=1, WinUI=2}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.ExperiencePoints}" Placeholder="Опыт" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.ExperiencePoints}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Опыт" Style="{StaticResource LabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="3" Grid.Column="{OnPlatform Default=0, WinUI=3}" Orientation="Vertical" Padding="5">

<Entry Text="{Binding Char.Level}" Placeholder="Уровень" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Level}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Уровень" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Grid>

</Frame>

<!-- Abilities and Saves -->

<Grid Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="{OnPlatform Default=1, WinUI=2}" Margin="5">

<Grid RowDefinitions="Auto,Auto,Auto,Auto" ColumnDefinitions="{OnPlatform Default='\*', WinUI='\*,\*'}">

<Frame Grid.Row="0" Grid.RowSpan="4" Grid.Column="0" Style="{StaticResource FrameStyle}">

<StackLayout Spacing="10">

<!-- Strength -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Сила" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Strength, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Strength}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

<!-- Dexterity -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Ловкость" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Dexterity, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Dexterity}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

<!-- Constitution -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Телосложение" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Constitution, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Constitution}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

<!-- Intelligence -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Интеллект" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Intelligence, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Intelligence}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

<!-- Wisdom -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Мудрость" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Wisdom, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Wisdom}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

<!-- Charisma -->

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<Label Text="Харизма" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

<Label Text="{Binding Char.Charisma, StringFormat='{0}'}" FontSize="{OnPlatform Default=Medium, WinUI=Large}" HorizontalOptions="Center" Style="{StaticResource ShowLabelStyle}" />

<Entry Text="{Binding Char.Charisma}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

</StackLayout>

</Frame>

</StackLayout>

</Frame>

<Frame Grid.Row="0" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<StackLayout Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.Inspiration}" />

<Label Text="Вдохновение" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="1" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<StackLayout Orientation="Horizontal">

<Label Text="+" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.ProficiencyBonus}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.ProficiencyBonus, StringFormat='+{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Бонус владения" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="2" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto,Auto,Auto,Auto,Auto">

<StackLayout Grid.Row="0" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowStrengthProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowStrength}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowStrength, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Сила" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="1" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowDexterityProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowDexterity}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowDexterity, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Ловкость" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="2" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowConstitutionProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowConstitution}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowConstitution, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Телосложение" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="3" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowIntelligenceProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowIntelligence}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowIntelligence, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Интеллект" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="4" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowWisdomProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowWisdom}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowWisdom, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Мудрость" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="5" Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.SavingThrowCharismaProficiency}" IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SavingThrowCharisma}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SavingThrowCharisma, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Харизма" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<Label Grid.Row="6" Text="СПАСБРОСКИ" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

<Frame Grid.Row="3" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto,Auto">

<StackLayout Grid.Row="0" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Acrobatics}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Acrobatics, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Акробатика (Лов)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="1" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.AnimalHandling}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.AnimalHandling, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Уход за животными (Муд)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="2" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Arcana}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Arcana, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Магия (Инт)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="3" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Athletics}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Athletics, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Атлетика (Сил)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="4" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Deception}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Deception, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Обман (Хар)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="5" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.History}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.History, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="История (Инт)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="6" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Insight}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Insight, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Проницательность (Муд)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="7" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Intimidation}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Intimidation, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Запугивание (Хар)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="8" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Investigation}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Investigation, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Анализ (Инт)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="9" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Medicine}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Medicine, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Медицина (Муд)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="10" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Nature}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Nature, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Природа (Инт)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="11" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Perception}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Perception, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Восприятие (Муд)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="12" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Performance}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Performance, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Выступление (Хар)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="13" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Persuasion}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Persuasion, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Убеждение (Хар)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="14" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Religion}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Religion, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Религия (Инт)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="15" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.SleightOfHand}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SleightOfHand, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Ловкость рук (Лов)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="16" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Stealth}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Stealth, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Скрытность (Лов)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<StackLayout Grid.Row="17" Orientation="Horizontal">

<CheckBox IsVisible="{Binding EditMode}" />

<Entry Text="{Binding Char.Survival}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Survival, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Выживание (Муд)" Style="{StaticResource LabelStyle}" Margin="5,0" />

</StackLayout>

<Label Grid.Row="18" Text="НАВЫКИ" FontAttributes="Bold" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

</Grid>

</Grid>

<!-- Combat Stats -->

<Frame Grid.Row="2" Grid.Column="{OnPlatform Default=0, WinUI=2}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto,Auto" ColumnDefinitions="{OnPlatform Default='\*,\*,\*', WinUI='\*,\*,\*'}">

<Frame Grid.Row="0" Grid.Column="0" Style="{StaticResource FrameStyle}">

<StackLayout>

<Entry Text="{Binding Char.ArmorClass}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.ArmorClass, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="КЗ" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="0" Grid.Column="1" Style="{StaticResource FrameStyle}">

<StackLayout>

<StackLayout Orientation="Horizontal" HorizontalOptions="Center">

<Label Text="+" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.Initiative}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Initiative, StringFormat='+{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<Label Text="Инициатива" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="0" Grid.Column="2" Style="{StaticResource FrameStyle}">

<StackLayout>

<Entry Text="{Binding Char.Speed}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.Speed, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Скорость" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="1" Grid.ColumnSpan="{OnPlatform Default=3, WinUI=3}" Style="{StaticResource FrameStyle}">

<StackLayout>

<StackLayout Orientation="Horizontal">

<Label Text="Максимум хитов" Style="{StaticResource LabelStyle}" Margin="0,0,5,0" />

<Entry Text="{Binding Char.MaxHitPoints}" WidthRequest="100" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.MaxHitPoints, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<Entry Text="{Binding Char.CurrentHitPoints}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.CurrentHitPoints, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Текущие хиты" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="2" Grid.ColumnSpan="{OnPlatform Default=3, WinUI=3}" Style="{StaticResource FrameStyle}">

<StackLayout>

<Entry Text="{Binding Char.TemporaryHitPoints}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.TemporaryHitPoints, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Временные хиты" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Grid.Row="3" Grid.ColumnSpan="3" Style="{StaticResource FrameStyle}">

<StackLayout Orientation="Horizontal" HorizontalOptions="Center">

<Frame Style="{StaticResource FrameStyle}">

<StackLayout HorizontalOptions="Fill">

<StackLayout Orientation="Horizontal" HorizontalOptions="Center">

<Label Text="Всего" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.HitDice}" WidthRequest="40" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.HitDice}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<Label Text="Кость хитов" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

<Frame Style="{StaticResource FrameStyle}">

<StackLayout>

<StackLayout Orientation="Horizontal" HorizontalOptions="Center">

<Label Text="Успехи" Style="{StaticResource LabelStyle}" />

<StackLayout Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.DeathSaveSuccesses, Converter={StaticResource BoolToIntConverter}, ConverterParameter=1}"/>

<CheckBox IsChecked="{Binding Char.DeathSaveSuccesses, Converter={StaticResource BoolToIntConverter}, ConverterParameter=2}"/>

<CheckBox IsChecked="{Binding Char.DeathSaveSuccesses, Converter={StaticResource BoolToIntConverter}, ConverterParameter=3}"/>

</StackLayout>

</StackLayout>

<StackLayout Orientation="Horizontal" HorizontalOptions="Center">

<Label Text="Провалы" Style="{StaticResource LabelStyle}" />

<StackLayout Orientation="Horizontal">

<CheckBox IsChecked="{Binding Char.DeathSaveFailures, Converter={StaticResource BoolToIntConverter}, ConverterParameter=1}" />

<CheckBox IsChecked="{Binding Char.DeathSaveFailures, Converter={StaticResource BoolToIntConverter}, ConverterParameter=2}"/>

<CheckBox IsChecked="{Binding Char.DeathSaveFailures, Converter={StaticResource BoolToIntConverter}, ConverterParameter=3}"/>

</StackLayout>

</StackLayout>

<Label Text="Спасброски от смерти" Style="{StaticResource LabelStyle}" />

</StackLayout>

</Frame>

</StackLayout>

</Frame>

</Grid>

</Frame>

<!-- Personality -->

<Frame Grid.Row="3" Grid.Column="{OnPlatform Default=0, WinUI=2}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto,Auto">

<Grid Grid.Row="0" RowDefinitions="Auto,Auto">

<Editor Grid.Row="0" Text="{Binding Char.PersonalityTraits}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.PersonalityTraits}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Черты характера" Style="{StaticResource LabelStyle}" />

</Grid>

<Grid Grid.Row="1" RowDefinitions="Auto,Auto">

<Editor Grid.Row="0" Text="{Binding Char.Ideals}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.Ideals}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Идеалы" Style="{StaticResource LabelStyle}" />

</Grid>

<Grid Grid.Row="2" RowDefinitions="Auto,Auto">

<Editor Grid.Row="0" Text="{Binding Char.Bonds}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.Bonds}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Привязанности" Style="{StaticResource LabelStyle}" />

</Grid>

<Grid Grid.Row="3" RowDefinitions="Auto,Auto">

<Editor Grid.Row="0" Text="{Binding Char.Flaws}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.Flaws}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Слабости" Style="{StaticResource LabelStyle}" />

</Grid>

</Grid>

</Frame>

<!-- Attacks and Spells -->

<Frame Grid.Row="3" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="\*,\*,Auto" ColumnDefinitions="9\*,\*">

<Button Grid.Row="0" Grid.Column="1" Command="{Binding AddAttackCommand}" Text="+"></Button>

<CollectionView Grid.Row="1" Grid.ColumnSpan="2" ItemsSource="{Binding Char.Attacks}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Grid ColumnDefinitions="{OnPlatform Default='\*,\*,\*', WinUI='\*,\*,\*'}">

<Entry Grid.Column="0" Text="{Binding Name}" Placeholder="Название" Style="{StaticResource EntryStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode}"/>

<Entry Grid.Column="1" Text="{Binding AttackBonus}" Placeholder="Бонус" Style="{StaticResource EntryStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode}"/>

<Entry Grid.Column="2" Text="{Binding DamageDice}" Placeholder="Урон" Style="{StaticResource EntryStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode}"/>

<Label Grid.Column="0" Text="{Binding Name}" Style="{StaticResource ShowLabelStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode, Converter={StaticResource BooleanNegationConverter}}"/>

<Label Grid.Column="1" Text="{Binding AttackBonus}" Style="{StaticResource ShowLabelStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode, Converter={StaticResource BooleanNegationConverter}}"/>

<Label Grid.Column="2" Text="{Binding DamageDice}" Style="{StaticResource ShowLabelStyle}" IsVisible="{Binding Source={RelativeSource AncestorType={x:Type vm:CharDetailsViewModel}}, Path=EditMode, Converter={StaticResource BooleanNegationConverter}}"/>

</Grid>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

<Label Grid.Row="2" Grid.ColumnSpan="2" Text="Атаки и заклинания" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

<!-- Passive Wisdom and Proficiencies -->

<Grid Grid.Row="4" Grid.Column="0" Margin="5">

<Grid RowDefinitions="Auto,Auto">

<StackLayout Orientation="Horizontal">

<Entry Text="{Binding Char.PassiveWisdom}" WidthRequest="50" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.PassiveWisdom, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

<Label Text="Пассивная мудрость (Восприятие)" Style="{StaticResource LabelStyle}" />

</StackLayout>

<Frame Grid.Row="1" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="\*,Auto">

<Editor Grid.Row="0" Text="{Binding Char.ProficienciesAndLanguages}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.ProficienciesAndLanguages}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Прочие владения и языки" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

</Grid>

</Grid>

<!-- Equipment and Currency -->

<Frame Grid.Row="4" Grid.Column="{OnPlatform Default=0, WinUI=1}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="Auto,Auto,Auto" ColumnDefinitions="{OnPlatform Default='\*,\*,\*,\*,\*', WinUI='\*,\*,\*,\*'}">

<StackLayout Grid.Row="0" Grid.Column="0" Margin="2">

<Label Text="ММ" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.CopperPieces}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.CopperPieces, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="0" Grid.Column="1" Margin="2">

<Label Text="СМ" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.SilverPieces}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.SilverPieces, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="0" Grid.Column="2" Margin="2">

<Label Text="ЗМ" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.GoldPieces}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.GoldPieces, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="0" Grid.Column="3" Margin="2">

<Label Text="ЭМ" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.ElectrumPieces}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.ElectrumPieces, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<StackLayout Grid.Row="0" Grid.Column="4" Margin="2" IsVisible="{OnPlatform Default=False, WinUI=True}">

<Label Text="ПМ" Style="{StaticResource LabelStyle}" />

<Entry Text="{Binding Char.PlatinumPieces}" Style="{StaticResource EntryStyle}" />

<Label Text="{Binding Char.PlatinumPieces, StringFormat='{0}'}" Style="{StaticResource ShowLabelStyle}" />

</StackLayout>

<Editor Grid.Row="1" Grid.ColumnSpan="{OnPlatform Default=5, WinUI=4}" Text="{Binding Char.Equipment}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="1" Grid.ColumnSpan="{OnPlatform Default=5, WinUI=4}" Text="{Binding Char.Equipment}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="2" Grid.ColumnSpan="{OnPlatform Default=5, WinUI=4}" Text="Снаряжение" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

<!-- Features and Traits -->

<Frame Grid.Row="4" Grid.Column="{OnPlatform Default=0, WinUI=2}" Style="{StaticResource FrameStyle}">

<Grid RowDefinitions="\*,Auto">

<Editor Grid.Row="0" Text="{Binding Char.FeaturesAndTraits}" Style="{StaticResource EntryStyle}" />

<Label Grid.Row="0" Text="{Binding Char.FeaturesAndTraits}" Style="{StaticResource ShowLabelStyle}" />

<Label Grid.Row="1" Text="Умения и способности" Style="{StaticResource LabelStyle}" />

</Grid>

</Frame>

</Grid>

</ScrollView>

</ContentPage.Content>

</ContentPage>

84. Views\CharacterDetailsPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.Models;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CharacterDetailsPage : ContentPage

{

public CharacterDetailsPage(PlayerCharacter character)

{

InitializeComponent();

BindingContext = new CharDetailsViewModel(Navigation, character);

}

}

85. Views\CharactersPage.xaml

﻿<?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"

xmlns:cards="clr-namespace:DnDClient.Views.Cards"

xmlns:vm="clr-namespace:DnDClient.ViewModels"

x:Class="DnDClient.Views.CharactersPage">

<ContentPage.Content>

<StackLayout Orientation="Vertical">

<Button Text="Добавить персонажа" Command="{Binding AddCommand}"></Button>

<CollectionView ItemsSource="{Binding Characters}">

<CollectionView.ItemTemplate>

<DataTemplate>

<cards:CharCard BindingContext="{Binding .}" />

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</ContentPage.Content>

</ContentPage>

86. Views\CharactersPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CharactersPage : ContentPage

{

public CharactersPage()

{

InitializeComponent();

BindingContext = new CharactersViewModel(Navigation);

}

}

87. Views\CombatPage.xaml

﻿<?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"

xmlns:conv="clr-namespace:DnDClient.Converters"

x:Class="DnDClient.Views.CombatPage">

<ContentPage.Resources>

<conv:NpcFilterConverter x:Key="NpcFilterConverter" />

<conv:EnemyFilterConverter x:Key="EnemyFilterConverter" />

</ContentPage.Resources>

<ContentPage.Content>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!-- Информация о текущем ходе -->

<StackLayout Grid.Row="0" Grid.ColumnSpan="2" Padding="10">

<Button Text="Управлять участниками" Command="{Binding ManageParticipantsCommand}"></Button>

<Label Text="Раунд:" FontAttributes="Bold" />

<Label Text="{Binding Combat.CurrentRound}" />

<Label Text="Текущий ходящий:" FontAttributes="Bold" />

<Label Text="{Binding Combat.CurrentParticipant.Name}" />

</StackLayout>

<!-- Основная область: логи и участники -->

<Grid Grid.Row="1" Grid.ColumnSpan="2">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!-- Логи боя -->

<StackLayout Grid.Column="0" Grid.ColumnSpan="2" Padding="10">

<Label Text="Логи боя" FontAttributes="Bold" />

<CollectionView ItemsSource="{Binding CombatLogs}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="2" Padding="5" BorderColor="Gray">

<Label Text="{Binding Message}" />

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

<StackLayout Grid.Row="1" Grid.Column="0" Padding="10" IsVisible="{Binding MasterMode}">

<Label Text="Ожидающие ходы" FontAttributes="Bold" />

<CollectionView ItemsSource="{Binding PendingLogs}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="2" Padding="5" BorderColor="Gray">

<StackLayout>

<Label Text="{Binding Message}" />

<Button Text="Подтвердить"

Command="{Binding BindingContext.ConfirmActionCommand, Source={x:Reference Name=CombatPage}}"

CommandParameter="{Binding .}" />

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

<!-- Участники боя -->

<StackLayout Grid.Column="1" Grid.ColumnSpan="2" Padding="10">

<Label Text="Участники" FontAttributes="Bold" />

<CollectionView ItemsSource="{Binding Combat.Participants}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="2" Padding="5" BorderColor="Gray">

<StackLayout>

<Label Text="{Binding Name}" />

<Label Text="{Binding CurrentHitPoints}" FontSize="10" />

<Label Text="{Binding Type}" FontSize="10" />

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Grid>

<!-- UI для игрока -->

<StackLayout Grid.Row="2" Grid.ColumnSpan="2" Padding="10" IsVisible="{Binding IsPlayerTurn}">

<Label Text="Ваш ход" FontAttributes="Bold" />

<Picker Title="Тип действия" ItemsSource="{Binding ActionTypes}"

SelectedItem="{Binding SelectedActionType}" />

<Picker Title="Цель" ItemsSource="{Binding Combat.Participants}" ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedTarget}" />

<Picker Title="Атака" ItemsSource="{Binding Combat.CurrentParticipant.Attacks}"

ItemDisplayBinding="{Binding Name}" SelectedItem="{Binding SelectedAttack}"

IsVisible="{Binding SelectedActionType}" />

<Entry Placeholder="Урон/лечение" Keyboard="Numeric" Text="{Binding AttackDamage}" />

<Button Text="Отправить ход" Command="{Binding SendPlayerActionCommand}" />

</StackLayout>

<!-- UI для мастера -->

<StackLayout Grid.Row="2" Grid.Column="0" Padding="10" IsVisible="{Binding MasterMode}">

<Label Text="Управление НПС" FontAttributes="Bold" />

<Picker Title="Выберите НПС"

ItemsSource="{Binding Combat.Participants, Converter={StaticResource NpcFilterConverter}}"

ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedNpc}" />

<Picker Title="Тип действия" ItemsSource="{Binding ActionTypes}"

SelectedItem="{Binding SelectedActionType}" />

<Picker Title="Цель" ItemsSource="{Binding Combat.Participants}" ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedTarget}" />

<Picker Title="Атака" ItemsSource="{Binding SelectedNpc.Attacks}" ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedAttack}" IsVisible="{Binding SelectedActionType}" />

<Entry Placeholder="Урон/лечение" Keyboard="Numeric" Text="{Binding AttackDamage}" />

<Button Text="Отправить ход за НПС" Command="{Binding SendNpcActionCommand}" BackgroundColor="#4CAF50"

TextColor="White" />

</StackLayout>

<StackLayout Grid.Row="2" Grid.Column="1" Padding="10" IsVisible="{Binding MasterMode}">

<Label Text="Управление врагами" FontAttributes="Bold" />

<Picker Title="Выберите врага"

ItemsSource="{Binding Combat.Participants, Converter={StaticResource EnemyFilterConverter}}"

ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedEnemy}" />

<Picker Title="Тип действия" ItemsSource="{Binding ActionTypes}"

SelectedItem="{Binding SelectedActionType}" />

<Picker Title="Цель" ItemsSource="{Binding Combat.Participants}" ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedTarget}" />

<Picker Title="Атака" ItemsSource="{Binding SelectedEnemy.Attacks}" ItemDisplayBinding="{Binding Name}"

SelectedItem="{Binding SelectedAttack}" IsVisible="{Binding SelectedActionType}" />

<Entry Placeholder="Урон/лечение" Keyboard="Numeric" Text="{Binding AttackDamage}" />

<Button Text="Отправить ход за врага" Command="{Binding SendEnemyActionCommand}"

BackgroundColor="#FF5722" TextColor="White" />

</StackLayout>

</Grid>

</ContentPage.Content>

</ContentPage>

88. Views\CombatPage.xaml.cs

﻿using DnDClient.Models;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CombatPage : ContentPage

{

public CombatPage(Combat combat, bool masterMode)

{

InitializeComponent();

BindingContext = new CombatViewModel(combat, masterMode);

}

}

89. Views\CombatParticipantsPage.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:local="using:DnDClient.Views"

x:Class="DnDClient.Views.CombatParticipantsPage"

Title="Управление участниками боя"

xmlns:android="clr-namespace:Microsoft.Maui.Controls.PlatformConfiguration.AndroidSpecific;assembly=Microsoft.Maui.Controls"

android:TabbedPage.ToolbarPlacement="Bottom">

<TabbedPage.Resources>

<local:HitPointsConverter x:Key="HitPointsConverter" />

</TabbedPage.Resources>

<!-- Вкладки с участниками -->

<TabbedPage.Children>

<!-- Вкладка для игроков -->

<ContentPage Title="Игроки" IconImageSource="user\_icon.png">

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<!-- Добавление игроков -->

<Frame>

<StackLayout Spacing="10">

<Label Text="Добавить игрока" FontAttributes="Bold" FontSize="16" />

<CollectionView ItemsSource="{Binding AvailableCharacters}"

HeightRequest="200"

EmptyView="Нет доступных персонажей">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="10" Margin="2" BackgroundColor="#2F4F4F">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Label Text="{Binding Type, StringFormat='Тип: Игрок'}"

FontSize="12" />

<Label

Text="{Binding CurrentHitPoints, StringFormat='HP: {0}/{1}'}"

FontSize="12" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}"

FontSize="12" />

</StackLayout>

<Button Text="➕ Добавить в бой"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.AddParticipantCommand}"

CommandParameter="{Binding}"

BackgroundColor="#4CAF50"

TextColor="White"

Margin="0,5,0,0" />

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Frame>

<!-- Список игроков в бою -->

<Label Text="Игроки в бою:" FontAttributes="Bold" FontSize="16" />

<CollectionView ItemsSource="{Binding Participants}"

EmptyView="Нет игроков в бою">

<CollectionView.ItemsLayout>

<LinearItemsLayout Orientation="Vertical" ItemSpacing="10" />

</CollectionView.ItemsLayout>

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="15" Margin="2" BackgroundColor="#2F4F4F">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<Label Text="{Binding Type, StringFormat='Тип: {0}'}" FontSize="12" />

<Label

Text="{Binding CurrentHitPoints, StringFormat='HP: {0}/{1}', Converter={StaticResource HitPointsConverter}}"

FontSize="12" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}"

FontSize="12" />

<Label Text="{Binding Initiative, StringFormat='Инициатива: {0}'}"

FontSize="12" />

<StackLayout Orientation="Horizontal" Spacing="5"

Margin="0,10,0,0">

<Button Text="❤️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.UpdateParticipantHealthCommand}"

CommandParameter="{Binding}"

BackgroundColor="#2196F3"

TextColor="White"

WidthRequest="40" />

<Button Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.DeleteParticipantCommand}"

CommandParameter="{Binding}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</StackLayout>

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</ScrollView>

</ContentPage>

<!-- Вкладка NPC -->

<ContentPage Title="NPC" IconImageSource="npc\_icon.png">

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<!-- Добавление NPC -->

<Frame>

<StackLayout Spacing="10">

<Label Text="Добавить NPC" FontAttributes="Bold" FontSize="16" />

<Button Text="Создать нового NPC"

Command="{Binding OpenNPCCreationCommand}"

BackgroundColor="#2196F3"

TextColor="White"

Margin="0,0,0,10" />

<CollectionView ItemsSource="{Binding AvailableNPCs}"

HeightRequest="200"

EmptyView="Нет доступных NPC">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="10" Margin="2" BackgroundColor="#2F4F4F">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Label Text="{Binding Race}" FontSize="12" />

<Label Text="{Binding HitPoints, StringFormat='HP: {0}'}"

FontSize="12" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}"

FontSize="12" />

</StackLayout>

<StackLayout Orientation="Horizontal" Spacing="5">

<Button Text="✏️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.EditNPCCommand}"

CommandParameter="{Binding}"

BackgroundColor="#FFC107"

WidthRequest="40" />

<Button Text="➕"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.AddParticipantCommand}"

CommandParameter="{Binding}"

BackgroundColor="#4CAF50"

TextColor="White"

WidthRequest="40" />

<Button Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.DeleteNPCCommand}"

CommandParameter="{Binding}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</StackLayout>

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Frame>

</StackLayout>

</ScrollView>

</ContentPage>

<!-- Вкладка врагов -->

<ContentPage Title="Враги" IconImageSource="enemy\_icon.png">

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<!-- Добавление врагов -->

<Frame>

<StackLayout Spacing="10">

<Label Text="Добавить врага" FontAttributes="Bold" FontSize="16" />

<Button Text="Создать нового врага"

Command="{Binding OpenEnemyCreationCommand}"

BackgroundColor="#2196F3"

TextColor="White"

Margin="0,0,0,10" />

<CollectionView ItemsSource="{Binding AvailableEnemies}"

HeightRequest="200"

EmptyView="Нет доступных врагов">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="10" Margin="2" BackgroundColor="#2F4F4F">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Label Text="{Binding Type}" FontSize="12" />

<Label Text="{Binding HitPoints, StringFormat='HP: {0}'}"

FontSize="12" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}"

FontSize="12" />

<Label Text="{Binding ChallengeRating, StringFormat='CR: {0}'}"

FontSize="12" />

</StackLayout>

<StackLayout Orientation="Horizontal" Spacing="5">

<Button Text="✏️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.EditEnemyCommand}"

CommandParameter="{Binding}"

BackgroundColor="#FFC107"

WidthRequest="40" />

<Button Text="➕"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.AddParticipantCommand}"

CommandParameter="{Binding}"

BackgroundColor="#4CAF50"

TextColor="White"

WidthRequest="40" />

<Button Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CombatParticipantsPage}},

Path=BindingContext.DeleteEnemyCommand}"

CommandParameter="{Binding}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</StackLayout>

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</Frame>

</StackLayout>

</ScrollView>

</ContentPage>

</TabbedPage.Children>

</TabbedPage>

90. Views\CombatParticipantsPage.xaml.cs

﻿using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using DnDClient.Models;

using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class CombatParticipantsPage : TabbedPage

{

public CombatParticipantsPage(Combat combat)

{

InitializeComponent();

BindingContext = new CombatParticipantsViewModel(combat);

}

}

91. Views\CreateEnemyPage.xaml

﻿<?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"

xmlns:local="using:DnDClient.Views"

x:Class="DnDClient.Views.CreateEnemyPage"

xmlns:toolkit="http://schemas.microsoft.com/dotnet/2022/maui/toolkit"

Title="{Binding Title}">

<ContentPage.Resources>

<Style x:Key="InvalidEntryStyle" TargetType="Entry">

<Setter Property="TextColor" Value="Red" />

<!-- Можно добавить другие свойства, например, BackgroundColor -->

</Style>

<Style x:Key="ValidEntryStyle" TargetType="Entry">

<Setter Property="TextColor" Value="Green" />

</Style>

</ContentPage.Resources>

<ContentPage.Content>

<ScrollView>

<VerticalStackLayout Padding="20" Spacing="16">

<Entry Placeholder="Имя" Text="{Binding Name}" />

<Entry Placeholder="Тип" Text="{Binding Type}" />

<Entry Placeholder="Здоровье" Keyboard="Numeric" Text="{Binding HitPoints}">

<Entry.Behaviors>

<toolkit:NumericValidationBehavior

InvalidStyle="{StaticResource InvalidEntryStyle}"

ValidStyle="{StaticResource ValidEntryStyle}"

Flags="ValidateOnValueChanged"

MinimumValue="1.0"

MaximumValue="100.0"

MaximumDecimalPlaces="2" />

</Entry.Behaviors>

</Entry>

<Entry Placeholder="Класс брони" Keyboard="Numeric" Text="{Binding ArmorClass}">

<Entry.Behaviors>

<toolkit:NumericValidationBehavior

InvalidStyle="{StaticResource InvalidEntryStyle}"

ValidStyle="{StaticResource ValidEntryStyle}"

Flags="ValidateOnValueChanged"

MinimumValue="1.0"

MaximumValue="100.0"

MaximumDecimalPlaces="2" />

</Entry.Behaviors>

</Entry>

<Entry Placeholder="CR" Text="{Binding ChallengeRating}" />

<Editor Placeholder="Описание" HeightRequest="100" Text="{Binding Description}" />

<Frame Padding="10" Margin="0,10">

<VerticalStackLayout Spacing="10">

<Label Text="Специальные способности" FontAttributes="Bold" />

<Button Text="Добавить способность" Command="{Binding AddAbilityCommand}" />

<CollectionView ItemsSource="{Binding SpecialAbilities}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="0,5" Padding="10">

<Grid ColumnDefinitions="\*, Auto">

<VerticalStackLayout Grid.Column="0" Spacing="5">

<Entry Text="{Binding Name}" Placeholder="Название" />

<Editor Text="{Binding Description}"

Placeholder="Описание"

HeightRequest="60"

AutoSize="TextChanges" />

</VerticalStackLayout>

<Button Grid.Column="1"

Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CreateEnemyPage}}, Path=BindingContext.DelAbilityCommand}"

CommandParameter="{Binding .}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</Grid>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</VerticalStackLayout>

</Frame>

<Frame Padding="10" Margin="0,10">

<VerticalStackLayout Spacing="10">

<Label Text="Атаки" FontAttributes="Bold" />

<Button Text="Добавить атаку" Command="{Binding AddAttackCommand}" />

<CollectionView ItemsSource="{Binding Attacks}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="0,5" Padding="10">

<Grid ColumnDefinitions="\*, \*, Auto">

<VerticalStackLayout Grid.Column="0" Spacing="5">

<Entry Text="{Binding Name}" Placeholder="Название" />

<Entry Text="{Binding AttackBonus}" Placeholder="Бонус атаки" />

</VerticalStackLayout>

<VerticalStackLayout Grid.Column="1" Spacing="5">

<Entry Text="{Binding DamageDice}" Placeholder="Кость урона" />

<Entry Text="{Binding DamageBonus}" Placeholder="Бонус урона" />

</VerticalStackLayout>

<Button Grid.Column="2"

Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CreateEnemyPage}}, Path=BindingContext.DelAttackCommand}"

CommandParameter="{Binding .}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</Grid>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</VerticalStackLayout>

</Frame>

<Button Text="{Binding ActionButtonText}"

Command="{Binding SaveCommand}"

BackgroundColor="#4CAF50"

TextColor="White"

Margin="0,20,0,0" />

</VerticalStackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

92. Views\CreateEnemyPage.xaml.cs

﻿using DnDClient.ViewModels;

namespace DnDClient.Views

{

public partial class CreateEnemyPage : ContentPage

{

public CreateEnemyPage()

{

InitializeComponent();

BindingContext = new CreateEnemyViewModel();

}

}

}

93. Views\CreateNPCPage.xaml

﻿<?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"

xmlns:local="using:DnDClient.Views"

x:Class="DnDClient.Views.CreateNPCPage"

xmlns:toolkit="http://schemas.microsoft.com/dotnet/2022/maui/toolkit"

Title="{Binding Title}">

<ContentPage.Resources>

<Style x:Key="InvalidEntryStyle" TargetType="Entry">

<Setter Property="TextColor" Value="Red" />

<!-- Можно добавить другие свойства, например, BackgroundColor -->

</Style>

<Style x:Key="ValidEntryStyle" TargetType="Entry">

<Setter Property="TextColor" Value="Green" />

</Style>

</ContentPage.Resources>

<ContentPage.Content>

<ScrollView>

<VerticalStackLayout Padding="20" Spacing="16">

<Entry Placeholder="Имя" Text="{Binding Name}" />

<Entry Placeholder="Роль" Text="{Binding Role}" />

<Entry Placeholder="Раса" Text="{Binding Race}" />

<Entry Placeholder="Профессия" Text="{Binding Occupation}" />

<Entry Placeholder="Здоровье" Keyboard="Numeric" Text="{Binding HitPoints}">

<Entry.Behaviors>

<toolkit:NumericValidationBehavior

InvalidStyle="{StaticResource InvalidEntryStyle}"

ValidStyle="{StaticResource ValidEntryStyle}"

Flags="ValidateOnValueChanged"

MinimumValue="1.0"

MaximumValue="100.0"

MaximumDecimalPlaces="2" />

</Entry.Behaviors>

</Entry>

<Entry Placeholder="Класс брони" Keyboard="Numeric" Text="{Binding ArmorClass}">

<Entry.Behaviors>

<toolkit:NumericValidationBehavior

InvalidStyle="{StaticResource InvalidEntryStyle}"

ValidStyle="{StaticResource ValidEntryStyle}"

Flags="ValidateOnValueChanged"

MinimumValue="1.0"

MaximumValue="100.0"

MaximumDecimalPlaces="2" />

</Entry.Behaviors>

</Entry>

<Editor Placeholder="Описание" HeightRequest="100" Text="{Binding Description}" />

<Editor Placeholder="Черты характера" HeightRequest="60" Text="{Binding PersonalityTraits}" />

<Editor Placeholder="Идеалы" HeightRequest="60" Text="{Binding Ideals}" />

<Editor Placeholder="Узы" HeightRequest="60" Text="{Binding Bonds}" />

<Editor Placeholder="Изъяны" HeightRequest="60" Text="{Binding Flaws}" />

<Frame Padding="10" Margin="0,10">

<VerticalStackLayout Spacing="10">

<Label Text="Атаки" FontAttributes="Bold" />

<Button Text="Добавить атаку" Command="{Binding AddAttackCommand}" />

<CollectionView ItemsSource="{Binding Attacks}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Margin="0,5" Padding="10">

<Grid ColumnDefinitions="\*, \*, Auto">

<VerticalStackLayout Grid.Column="0" Spacing="5">

<Entry Text="{Binding Name}" Placeholder="Название" />

<Entry Text="{Binding AttackBonus}" Placeholder="Бонус атаки" />

</VerticalStackLayout>

<VerticalStackLayout Grid.Column="1" Spacing="5">

<Entry Text="{Binding DamageDice}" Placeholder="Кость урона" />

<Entry Text="{Binding DamageBonus}" Placeholder="Бонус урона" />

</VerticalStackLayout>

<Button Grid.Column="2"

Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:CreateNPCPage}}, Path=BindingContext.DelAttackCommand}"

CommandParameter="{Binding .}"

BackgroundColor="#F44336"

TextColor="White"

WidthRequest="40" />

</Grid>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</VerticalStackLayout>

</Frame>

<Button Text="{Binding ActionButtonText}"

Command="{Binding SaveCommand}"

BackgroundColor="#4CAF50"

TextColor="White"

Margin="0,20,0,0" />

</VerticalStackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

94. Views\CreateNPCPage.xaml.cs

﻿using DnDClient.ViewModels;

namespace DnDClient.Views

{

public partial class CreateNPCPage

{

public CreateNPCPage()

{

InitializeComponent();

BindingContext = new CreateNPCViewModel();

}

}

}

95. Views\EntityManagementPage.xaml

﻿<?xml version="1.0" encoding="utf-8"?>

<TabbedPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

xmlns:local="using:DnDClient.Views"

x:Class="DnDClient.Views.EntityManagementPage"

Title="Управление сущностями">

<!-- Вкладка NPC -->

<ContentPage Title="NPC">

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<!-- Форма редактирования NPC -->

<Frame IsVisible="{Binding IsEditingNPC}">

<StackLayout Spacing="10">

<Label Text="Редактировать NPC" FontAttributes="Bold" />

<Entry Placeholder="Имя" Text="{Binding NewNPC.Name}" />

<Entry Placeholder="Раса" Text="{Binding NewNPC.Race}" />

<Entry Placeholder="Профессия" Text="{Binding NewNPC.Occupation}" />

<Entry Placeholder="HP" Text="{Binding NewNPC.HitPoints}" Keyboard="Numeric" />

<Entry Placeholder="AC" Text="{Binding NewNPC.ArmorClass}" Keyboard="Numeric" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Button Text="Сохранить" Command="{Binding SaveNPCChangesCommand}"

BackgroundColor="#4CAF50" TextColor="White" />

<Button Text="Отмена" Command="{Binding CancelEditCommand}"

BackgroundColor="#9E9E9E" TextColor="White" />

</StackLayout>

</StackLayout>

</Frame>

<!-- Создание нового NPC -->

<Frame IsVisible="{Binding IsEditingNPC, Converter={StaticResource BooleanNegationConverter}}">

<StackLayout Spacing="10">

<Label Text="Создать нового NPC" FontAttributes="Bold" />

<Entry Placeholder="Имя" Text="{Binding NewNPC.Name}" />

<Entry Placeholder="Раса" Text="{Binding NewNPC.Race}" />

<Entry Placeholder="Профессия" Text="{Binding NewNPC.Occupation}" />

<Entry Placeholder="HP" Text="{Binding NewNPC.HitPoints}" Keyboard="Numeric" />

<Entry Placeholder="AC" Text="{Binding NewNPC.ArmorClass}" Keyboard="Numeric" />

<Button Text="Создать" Command="{Binding CreateNPCCommand}" />

</StackLayout>

</Frame>

<!-- Список NPC -->

<Label Text="Список NPC:" FontAttributes="Bold" />

<CollectionView ItemsSource="{Binding Npcs}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="15" Margin="5" BackgroundColor="LightGray">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<Label Text="{Binding Race, StringFormat='Раса: {0}'}" />

<Label Text="{Binding Occupation, StringFormat='Профессия: {0}'}" />

<Label Text="{Binding HitPoints, StringFormat='HP: {0}'}" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Button Text="✏️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:EntityManagementPage}},

Path=BindingContext.EditNPCCommand}"

CommandParameter="{Binding}"

BackgroundColor="#FFC107" />

<Button Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:EntityManagementPage}},

Path=BindingContext.DeleteNPCCommand}"

CommandParameter="{Binding}"

BackgroundColor="#F44336"

TextColor="White" />

</StackLayout>

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</ScrollView>

</ContentPage>

<!-- Вкладка Enemy -->

<ContentPage Title="Враги">

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<!-- Форма редактирования Enemy -->

<Frame IsVisible="{Binding IsEditingEnemy}">

<StackLayout Spacing="10">

<Label Text="Редактировать врага" FontAttributes="Bold" />

<Entry Placeholder="Имя" Text="{Binding NewEnemy.Name}" />

<Entry Placeholder="Тип" Text="{Binding NewEnemy.Type}" />

<Entry Placeholder="HP" Text="{Binding NewEnemy.HitPoints}" Keyboard="Numeric" />

<Entry Placeholder="AC" Text="{Binding NewEnemy.ArmorClass}" Keyboard="Numeric" />

<Entry Placeholder="CR" Text="{Binding NewEnemy.ChallengeRating}" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Button Text="Сохранить" Command="{Binding SaveEnemyChangesCommand}"

BackgroundColor="#4CAF50" TextColor="White" />

<Button Text="Отмена" Command="{Binding CancelEditCommand}"

BackgroundColor="#9E9E9E" TextColor="White" />

</StackLayout>

</StackLayout>

</Frame>

<!-- Создание нового Enemy -->

<Frame IsVisible="{Binding IsEditingEnemy, Converter={StaticResource BooleanNegationConverter}}">

<StackLayout Spacing="10">

<Label Text="Создать нового врага" FontAttributes="Bold" />

<Entry Placeholder="Имя" Text="{Binding NewEnemy.Name}" />

<Entry Placeholder="Тип" Text="{Binding NewEnemy.Type}" />

<Entry Placeholder="HP" Text="{Binding NewEnemy.HitPoints}" Keyboard="Numeric" />

<Entry Placeholder="AC" Text="{Binding NewEnemy.ArmorClass}" Keyboard="Numeric" />

<Entry Placeholder="CR" Text="{Binding NewEnemy.ChallengeRating}" />

<Button Text="Создать" Command="{Binding CreateEnemyCommand}" />

</StackLayout>

</Frame>

<!-- Список Enemy -->

<Label Text="Список врагов:" FontAttributes="Bold" />

<CollectionView ItemsSource="{Binding Enemies}">

<CollectionView.ItemTemplate>

<DataTemplate>

<Frame Padding="15" Margin="5" BackgroundColor="LightGray">

<StackLayout>

<Label Text="{Binding Name}" FontAttributes="Bold" />

<Label Text="{Binding Type, StringFormat='Тип: {0}'}" />

<Label Text="{Binding HitPoints, StringFormat='HP: {0}'}" />

<Label Text="{Binding ArmorClass, StringFormat='AC: {0}'}" />

<Label Text="{Binding ChallengeRating, StringFormat='CR: {0}'}" />

<StackLayout Orientation="Horizontal" Spacing="10">

<Button Text="✏️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:EntityManagementPage}},

Path=BindingContext.EditEnemyCommand}"

CommandParameter="{Binding}"

BackgroundColor="#FFC107" />

<Button Text="🗑️"

Command="{Binding Source={RelativeSource AncestorType={x:Type local:EntityManagementPage}},

Path=BindingContext.DeleteEnemyCommand}"

CommandParameter="{Binding}"

BackgroundColor="#F44336"

TextColor="White" />

</StackLayout>

</StackLayout>

</Frame>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

</StackLayout>

</ScrollView>

</ContentPage>

</TabbedPage>

96. Views\EntityManagementPage.xaml.cs

﻿using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class EntityManagementPage : TabbedPage

{

public EntityManagementPage()

{

InitializeComponent();

BindingContext = new EntityManagementViewModel();

}

}

97. Views\MainPage.xaml

﻿<?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="DnDClient.Views.MainPage">

<ScrollView>

<VerticalStackLayout

Padding="30,0"

Spacing="25">

<Label Text="{Binding Welcome}"/>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

98. Views\MainPage.xaml.cs

﻿using DnDClient.ViewModels;

namespace DnDClient.Views;

public partial class MainPage : ContentPage

{

int count = 0;

public MainPage()

{

InitializeComponent();

BindingContext = new MainViewModel();

}

}

99. Views\ProfilePage.xaml

<?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"

xmlns:viewModels="using:DnDClient.ViewModels"

x:Class="DnDClient.Views.ProfilePage">

<ContentPage.BindingContext>

<viewModels:ProfileViewModel />

</ContentPage.BindingContext>

<ContentPage.Content>

<Grid RowDefinitions="\*,\*,\*,\*,\*">

<Entry Placeholder="Имя пользователя" Text="{Binding NewUsername}" Grid.Row="0"></Entry>

<Button Text="Сменить имя пользователя" Grid.Row="1" Command="{Binding ChangeUsernameCommand}" />

<Entry Placeholder="Новый пароль" Grid.Row="2" IsPassword="True" Text="{Binding NewPassword}"></Entry>

<Entry Placeholder="Подтверрждение нового пароля" Grid.Row="3" IsPassword="True"

Text="{Binding CheckNewPassword}">

</Entry>

<Button Text="Сменить пароль" Grid.Row="4" Command="{Binding ChangePasswordCommand}"></Button>

</Grid>

</ContentPage.Content>

</ContentPage>

100. Views\ProfilePage.xaml.cs

namespace DnDClient.Views;

public partial class ProfilePage : ContentPage

{

public ProfilePage()

{

InitializeComponent();

}

}