**AddViolationPage**

<?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="MauiApp3.AddViolationPage"

Title="AddViolationPage">

<VerticalStackLayout Padding="20" Spacing="10">

<!-- Поле ввода описания нарушения -->

<Editor x:Name="descriptionEditor"

Placeholder="Описание нарушения"

AutoSize="TextChanges"/>

<!-- Выбор пользователя -->

<Picker x:Name="userPicker" Title="Выберите пользователя" />

<!-- Поле ввода уровня серьезности -->

<Entry x:Name="severityEntry"

Placeholder="Серьезность (число)"

Keyboard="Numeric"/>

<!-- Выбор даты -->

<DatePicker x:Name="violationDatePicker" />

<!-- Кнопки управления -->

<HorizontalStackLayout Spacing="10">

<Button Text="Сохранить" Clicked="OnSaveClicked" />

<Button Text="Закрыть" Clicked="OnCloseClicked" />

</HorizontalStackLayout>

</VerticalStackLayout>

</ContentPage>

using Microsoft.Maui.Controls;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Threading.Tasks;

namespace MauiApp3;

public partial class AddViolationPage : ContentPage

{

private Dictionary<int, string> users;

public AddViolationPage()

{

InitializeComponent();

LoadUsers();

violationDatePicker.Date = DateTime.Today;

}

private async void LoadUsers()

{

users = new Dictionary<int, string>();

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

await connection.OpenAsync();

string query = "SELECT UserId, Username FROM Users";

using (SqlCommand command = new SqlCommand(query, connection))

using (SqlDataReader reader = await command.ExecuteReaderAsync())

{

while (await reader.ReadAsync())

{

int userId = reader.GetInt32(0);

string username = reader.GetString(1);

users.Add(userId, username);

}

}

foreach (var user in users)

{

userPicker.Items.Add(user.Value);

}

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", $"Не удалось загрузить пользователей: {ex.Message}", "OK");

}

}

}

private async void OnSaveClicked(object sender, EventArgs e)

{

if (userPicker.SelectedIndex == -1 || string.IsNullOrEmpty(descriptionEditor.Text) || string.IsNullOrEmpty(severityEntry.Text))

{

await DisplayAlert("Ошибка", "Пожалуйста, заполните все поля.", "OK");

return;

}

string description = descriptionEditor.Text;

int severity = int.Parse(severityEntry.Text);

DateTime date = violationDatePicker.Date;

int userId = users.FirstOrDefault(x => x.Value == userPicker.SelectedItem.ToString()).Key;

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

await connection.OpenAsync();

string query = "INSERT INTO Violations (UserId, Description, Date, SeverityLevel, ViolationCount) VALUES (@UserId, @Description, @Date, @SeverityLevel, 0)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserId", userId);

command.Parameters.AddWithValue("@Description", description);

command.Parameters.AddWithValue("@Date", date);

command.Parameters.AddWithValue("@SeverityLevel", severity);

await command.ExecuteNonQueryAsync();

}

await DisplayAlert("Успех", "Нарушение успешно добавлено.", "OK");

await Navigation.PopAsync();

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", $"Не удалось сохранить нарушение: {ex.Message}", "OK");

}

}

}

private async void OnCloseClicked(object sender, EventArgs e)

{

await Navigation.PopAsync();

}

}

**AdminPage**

<?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="MauiApp3.AdminPage"

Title="AdminPage">

<ScrollView>

<VerticalStackLayout Padding="20">

<!-- Заголовок -->

<Label Text="Список нарушений"

FontSize="24" HorizontalOptions="Center"/>

<!-- Список нарушений -->

<ListView x:Name="violationsListView"

SelectionMode="Single"

ItemSelected="OnViolationSelected">

<ListView.ItemTemplate>

<DataTemplate>

<TextCell Text="{Binding Description}" Detail="{Binding Date}" />

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<!-- Кнопки управления -->

<HorizontalStackLayout Spacing="10" VerticalOptions="EndAndExpand">

<Button Text="Добавить нарушение" Clicked="OnAddViolationClicked" />

<Button Text="Все апелляции" Clicked="OnViewAppealsClicked" />

</HorizontalStackLayout>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

using Microsoft.Maui.Controls;

using System;

using System.Collections.ObjectModel;

using System.Data.SqlClient;

namespace MauiApp3;

public partial class AdminPage : ContentPage

{

private ObservableCollection<Violation> violations;

public AdminPage()

{

InitializeComponent();

violations = new ObservableCollection<Violation>();

violationsListView.ItemsSource = violations;

LoadViolations();

}

private async void LoadViolations()

{

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

string query = "SELECT ViolationId, Description, Date FROM Violations";

using (SqlCommand command = new SqlCommand(query, connection))

using (SqlDataReader reader = await command.ExecuteReaderAsync())

{

violations.Clear();

while (await reader.ReadAsync())

{

violations.Add(new Violation

{

ViolationId = reader.GetInt32(0),

Description = reader.GetString(1),

Date = reader.GetDateTime(2) // Теперь типы совпадают

});

}

}

}

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", $"Не удалось загрузить нарушения: {ex.Message}", "OK");

}

}

private async void OnAddViolationClicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new AddViolationPage());

}

private async void OnViewAppealsClicked(object sender, EventArgs e)

{

await Navigation.PushAsync(new ProcessAppealPage());

}

private void OnViolationSelected(object sender, SelectedItemChangedEventArgs e)

{

if (e.SelectedItem != null)

{

// Действия при выборе нарушения, если нужно

var selectedViolation = (Violation)e.SelectedItem;

Console.WriteLine($"Выбрано нарушение: {selectedViolation.Description}");

}

}

public class Violation

{

public int ViolationId { get; set; }

public int UserId { get; set; } // Добавляем это свойство

public string Description { get; set; }

public DateTime Date { get; set; }

public int SeverityLevel { get; set; }

public int ViolationCount { get; set; }

}

//private void OnViolationSelected(object sender, SelectedItemChangedEventArgs e)

//{

// if (e.SelectedItem is Violation selectedViolation)

// {

// // Действия с выбранным нарушением

// DisplayAlert("Выбрано нарушение", $"Описание: {selectedViolation.Description}", "OK");

// }

//// Снять выделение

//((ListView)sender).SelectedItem = null;

//}

}

**App**

<?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:MauiApp3"

x:Class="MauiApp3.App">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

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

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

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

namespace MauiApp3

{

public partial class App : Application

{

public App()

{

InitializeComponent();

}

protected override Window CreateWindow(IActivationState? activationState)

{

return new Window(new AppShell());

}

}

}

**AppealPage**

<?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="MauiApp3.AppealPage"

Title="AppealPage">

<ScrollView>

<VerticalStackLayout Spacing="20" Padding="30">

<!-- Отображение данных о нарушении -->

<Label Text="Нарушение:"

FontSize="24" HorizontalOptions="Center"/>

<Label x:Name="violationDescriptionLabel"

FontSize="18"

TextColor="Gray"

HorizontalOptions="Center"/>

<!-- TextBox для ввода апелляции -->

<Editor x:Name="appealEditor"

Placeholder="Опишите ваше обжалование..."

AutoSize="TextChanges"

HeightRequest="150"/>

<!-- Кнопка для отправки апелляции -->

<Button Text="Отправить"

Clicked="OnSubmitAppealClicked"/>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

using Microsoft.Maui.Controls;

using System;

using System.Data.SqlClient;

using System.Threading.Tasks;

namespace MauiApp3;

public partial class AppealPage : ContentPage

{

private int violationId;

private string violationDescription;

public AppealPage(int violationId, string violationDescription)

{

InitializeComponent();

this.violationId = violationId;

this.violationDescription = violationDescription;

violationDescriptionLabel.Text = violationDescription;

}

private async void OnSubmitAppealClicked(object sender, EventArgs e)

{

string appealText = appealEditor.Text;

if (string.IsNullOrEmpty(appealText))

{

await DisplayAlert("Ошибка", "Пожалуйста, напишите текст апелляции.", "OK");

return;

}

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

await connection.OpenAsync();

// Вставляем апелляцию в таблицу Appeals

string query = "INSERT INTO Appeals (ViolationId, AppealText, Status, DateSubmitted) VALUES (@ViolationId, @AppealText, @Status, @DateSubmitted)";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@ViolationId", violationId);

command.Parameters.AddWithValue("@AppealText", appealText);

command.Parameters.AddWithValue("@Status", "В ожидании"); // Статус по умолчанию

command.Parameters.AddWithValue("@DateSubmitted", DateTime.Now);

int rowsAffected = await command.ExecuteNonQueryAsync();

if (rowsAffected > 0)

{

await DisplayAlert("Успех", "Апелляция успешно отправлена.", "OK");

await Navigation.PopAsync(); // Возвращаемся на предыдущую страницу

}

else

{

await DisplayAlert("Ошибка", "Ошибка при отправке апелляции.", "OK");

}

}

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", "Произошла ошибка при подключении к базе данных: " + ex.Message, "OK");

}

}

}

}

**AppShell**

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

<Shell

x:Class="MauiApp3.AppShell"

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

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

xmlns:local="clr-namespace:MauiApp3"

Shell.FlyoutBehavior="Flyout"

Title="MauiApp3">

<ShellContent

Title="Home"

ContentTemplate="{DataTemplate local:MainPage}"

Route="MainPage" />

</Shell>

namespace MauiApp3

{

public partial class AppShell : Shell

{

public AppShell()

{

InitializeComponent();

}

}

}

**CourierPage**

<?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="MauiApp3.CourierPage"

Title="CourierPage">

<ScrollView>

<VerticalStackLayout Spacing="20" Padding="30">

<Label Text="Список нарушений"

FontSize="24"

HorizontalOptions="Center" />

<ListView x:Name="violationsListView"

ItemSelected="OnViolationSelected">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<StackLayout Padding="10">

<Label Text="{Binding Description}" FontSize="18"/>

<Label Text="{Binding Date}" FontSize="14" TextColor="Gray"/>

<Label Text="{Binding SeverityLevel}" FontSize="16"/>

</StackLayout>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button Text="Выговор" Clicked="OnPunishClicked"/>

<Button Text="Обжаловать" Clicked="OnAppealClicked"/>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

using Microsoft.Maui.Controls;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Threading.Tasks;

namespace MauiApp3;

public partial class CourierPage : ContentPage

{

public CourierPage()

{

InitializeComponent();

LoadViolations();

}

private List<Violation> violationsList = new List<Violation>();

// Загрузка нарушений из базы данных

private async void LoadViolations()

{

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

string query = "SELECT ViolationId, Description, Date, SeverityLevel FROM Violations WHERE UserID = @UserID";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@UserID", 1); // Тут можно заменить на динамическое значение UserID

SqlDataReader reader = await command.ExecuteReaderAsync();

while (await reader.ReadAsync())

{

violationsList.Add(new Violation

{

ViolationId = reader.GetInt32(0),

Description = reader.GetString(1),

Date = reader.GetDateTime(2).ToString("dd-MM-yyyy"),

SeverityLevel = reader.GetInt32(3)

});

}

violationsListView.ItemsSource = violationsList;

}

}

}

// Обработчик нажатия на кнопку "Выговор"

private async void OnPunishClicked(object sender, EventArgs e)

{

if (violationsListView.SelectedItem is Violation selectedViolation)

{

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

string query = "SELECT PunishmentDescription FROM Punishments WHERE SeverityLevel = @SeverityLevel";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@SeverityLevel", selectedViolation.SeverityLevel);

var punishmentDescription = await command.ExecuteScalarAsync();

if (punishmentDescription != null)

{

await DisplayAlert("Выговор", punishmentDescription.ToString(), "OK");

}

else

{

await DisplayAlert("Ошибка", "Штраф для этого уровня не найден.", "OK");

}

}

}

}

else

{

await DisplayAlert("Ошибка", "Пожалуйста, выберите нарушение.", "OK");

}

}

// Обработчик нажатия на кнопку "Обжаловать"

private async void OnAppealClicked(object sender, EventArgs e)

{

if (violationsListView.SelectedItem is Violation selectedViolation)

{

// Переход на форму обжалования с передачей ViolationId и Description

await Navigation.PushAsync(new AppealPage(selectedViolation.ViolationId, selectedViolation.Description));

}

else

{

await DisplayAlert("Ошибка", "Пожалуйста, выберите нарушение для обжалования.", "OK");

}

}

// Обработчик для выбора нарушения

private void OnViolationSelected(object sender, SelectedItemChangedEventArgs e)

{

// Можно добавить логику для подсветки или получения данных о выбранном нарушении

}

}

// Класс для хранения данных нарушения

public class Violation

{

public int ViolationId { get; set; }

public string Description { get; set; }

public string Date { get; set; }

public int SeverityLevel { get; set; }

}

**MainPage**

<?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="MauiApp3.MainPage">

<VerticalStackLayout Spacing="25" Padding="30">

<Label Text="Авторизация"

FontSize="32"

HorizontalOptions="Center" />

<Entry x:Name="usernameEntry" Placeholder="Введите имя пользователя" />

<Entry x:Name="passwordEntry" Placeholder="Введите пароль" IsPassword="True" />

<Button Text="Войти" Clicked="OnLoginClicked" />

</VerticalStackLayout>

</ContentPage>

using System;

using System.Data.SqlClient;

namespace MauiApp3

{

public partial class MainPage : ContentPage

{

int count = 0;

public MainPage()

{

InitializeComponent();

}

private async void OnLoginClicked(object sender, EventArgs e)

{

string username = usernameEntry.Text;

string password = passwordEntry.Text;

if (string.IsNullOrEmpty(username) || string.IsNullOrEmpty(password))

{

await DisplayAlert("Ошибка", "Пожалуйста, введите имя пользователя и пароль.", "OK");

return;

}

string role = await AuthenticateUser(username, password);

if (role != null)

{

// Переход на соответствующую форму в зависимости от роли

if (role == "Courier")

{

// Переход на форму для курьера

await Navigation.PushAsync(new CourierPage()); // Страница для курьера

}

else if (role == "Admin")

{

// Переход на форму для администратора

await Navigation.PushAsync(new AdminPage()); // Страница для администратора

}

}

else

{

await DisplayAlert("Ошибка", "Неверное имя пользователя или пароль.", "OK");

}

}

// Метод для аутентификации пользователя с учетом роли

private async Task<string> AuthenticateUser(string username, string password)

{

try

{

string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

// Запрос для получения роли пользователя

string query = "SELECT role FROM Users WHERE username = @username AND password = @password";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@username", username);

command.Parameters.AddWithValue("@password", password);

var result = await command.ExecuteScalarAsync();

// Если роль найдена, возвращаем ее

if (result != null)

{

return result.ToString(); // Возвращаем роль (Courier или Admin)

}

else

{

return null; // Если пользователя с такими данными нет

}

}

}

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", "Ошибка подключения к базе данных: " + ex.Message, "OK");

return null;

}

}

}

}

**MauiProgram**

using Microsoft.Extensions.Logging;

namespace MauiApp3

{

public static class MauiProgram

{

public static MauiApp CreateMauiApp()

{

var builder = MauiApp.CreateBuilder();

builder

.UseMauiApp<App>()

.ConfigureFonts(fonts =>

{

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

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

});

#if DEBUG

builder.Logging.AddDebug();

#endif

return builder.Build();

}

}

}

**ProcessAppealPage**

<?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="MauiApp3.ProcessAppealPage"

Title="ProcessAppealPage">

<ScrollView>

<VerticalStackLayout Padding="20">

<!-- Заголовок -->

<Label Text="Список апелляций"

FontSize="24" HorizontalOptions="Center"/>

<!-- Список апелляций -->

<ListView x:Name="appealsListView"

SelectionMode="Single">

<ListView.ItemTemplate>

<DataTemplate>

<TextCell Text="{Binding AppealText}" Detail="{Binding Status}" />

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<!-- Кнопки управления -->

<HorizontalStackLayout Spacing="10" VerticalOptions="EndAndExpand">

<Button Text="Одобрить" Clicked="OnApproveClicked" />

<Button Text="Отклонить" Clicked="OnRejectClicked" />

<Button Text="Закрыть" Clicked="OnCloseClicked" />

</HorizontalStackLayout>

</VerticalStackLayout>

</ScrollView>

</ContentPage>

using System.Collections.ObjectModel;

using System.Data.SqlClient;

using Microsoft.Maui.Controls;

namespace MauiApp3;

public partial class ProcessAppealPage : ContentPage

{

private ObservableCollection<Appeal> appeals = new ObservableCollection<Appeal>();

private const string connectionString = "Server=sql6033.site4now.net;Database=db\_aafddd\_kyrs;User Id=db\_aafddd\_kyrs\_admin;Password=kyrsovik23";

public ProcessAppealPage()

{

InitializeComponent();

LoadAppeals();

appealsListView.ItemsSource = appeals;

}

private async void LoadAppeals()

{

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

string query = "SELECT AppealId, ViolationId, AppealText, Status, DateSubmitted FROM Appeals";

using (SqlCommand command = new SqlCommand(query, connection))

using (SqlDataReader reader = await command.ExecuteReaderAsync())

{

appeals.Clear();

while (await reader.ReadAsync())

{

appeals.Add(new Appeal

{

AppealId = reader.GetInt32(0),

ViolationId = reader.GetInt32(1),

AppealText = reader.GetString(2),

Status = reader.GetString(3),

DateSubmitted = reader.GetDateTime(4)

});

}

}

}

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", $"Не удалось загрузить апелляции: {ex.Message}", "OK");

}

}

private async void OnApproveClicked(object sender, EventArgs e)

{

if (appealsListView.SelectedItem is Appeal selectedAppeal)

{

await UpdateAppealStatus(selectedAppeal.AppealId, "Approved");

}

else

{

await DisplayAlert("Ошибка", "Выберите апелляцию для одобрения.", "OK");

}

}

private async void OnRejectClicked(object sender, EventArgs e)

{

if (appealsListView.SelectedItem is Appeal selectedAppeal)

{

await UpdateAppealStatus(selectedAppeal.AppealId, "Rejected");

}

else

{

await DisplayAlert("Ошибка", "Выберите апелляцию для отклонения.", "OK");

}

}

private async Task UpdateAppealStatus(int appealId, string status)

{

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

await connection.OpenAsync();

string query = "UPDATE Appeals SET Status = @Status WHERE AppealId = @AppealId";

using (SqlCommand command = new SqlCommand(query, connection))

{

command.Parameters.AddWithValue("@Status", status);

command.Parameters.AddWithValue("@AppealId", appealId);

await command.ExecuteNonQueryAsync();

}

}

await DisplayAlert("Успех", $"Статус апелляции обновлен на \"{status}\".", "OK");

LoadAppeals(); // Обновляем список после изменения

}

catch (Exception ex)

{

await DisplayAlert("Ошибка", $"Не удалось обновить статус: {ex.Message}", "OK");

}

}

private async void OnCloseClicked(object sender, EventArgs e)

{

await Navigation.PopAsync();

}

}

public class Appeal

{

public int AppealId { get; set; }

public int ViolationId { get; set; }

public string AppealText { get; set; }

public string Status { get; set; }

public DateTime DateSubmitted { get; set; }

}