Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет

имени В.Ф.Уткина»

Кафедра ЭВМ

Отчет по лабораторной работе №14

на тему

«Работа с SQLite»

по дисциплине  
**«Визуальное программирование»**

Выполнили:

Студенты группы №140

Бригада 5

Сафаров Д. А.

Тимохин Е. С.

Проверили:  
ст. преп. Хизриева Н.И.

ст. преп. Бастрычкин А.С.

**Цель работы**

Получение навыков работы с библиотекой Xamarin.Essentials.

**Задание**

Разработать приложение для работы с БД Supply, которая хранит в себе информацию о товарах, поставщиках и поставках.**Практическая часть**

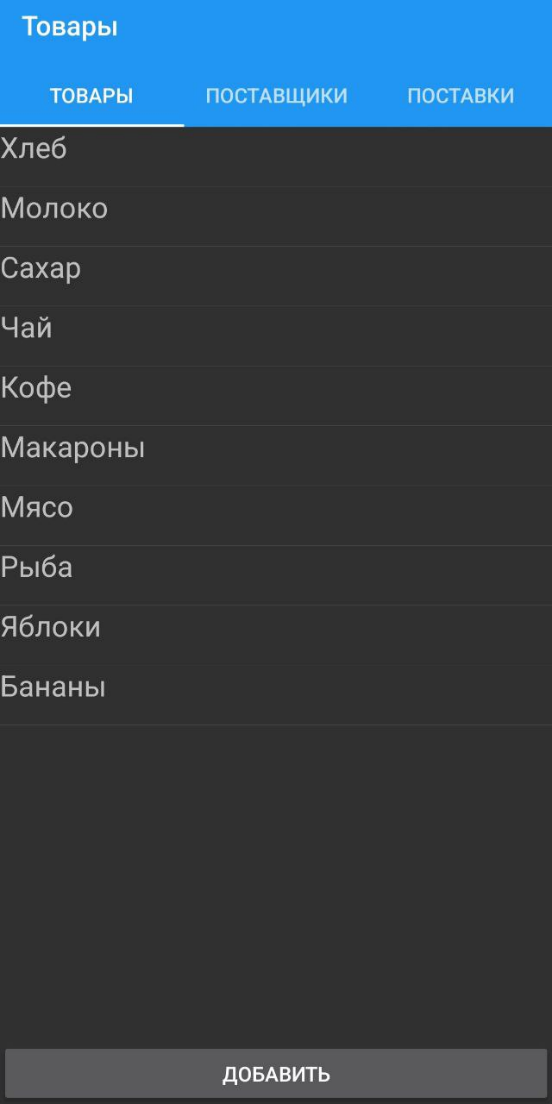
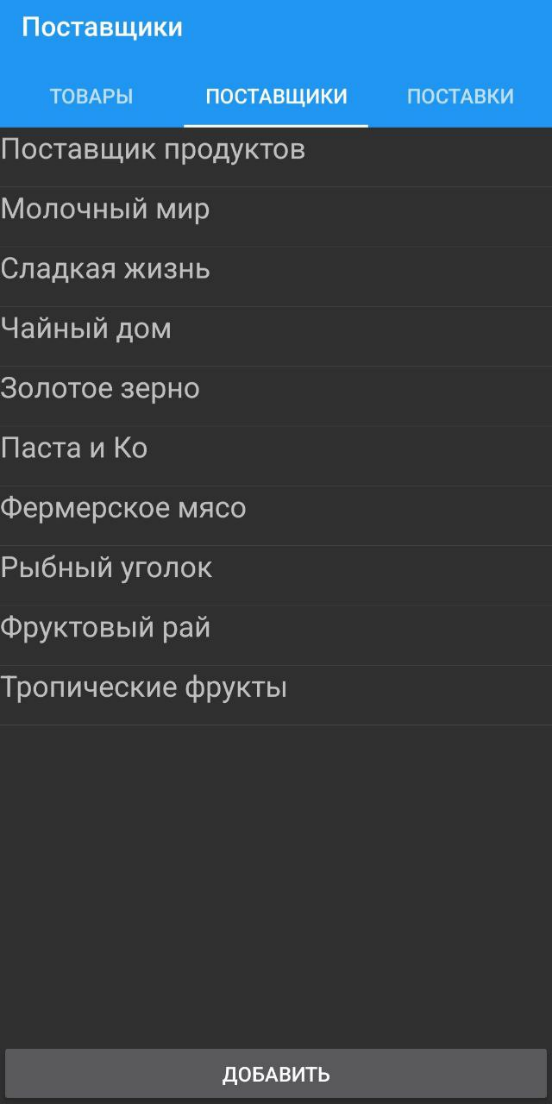
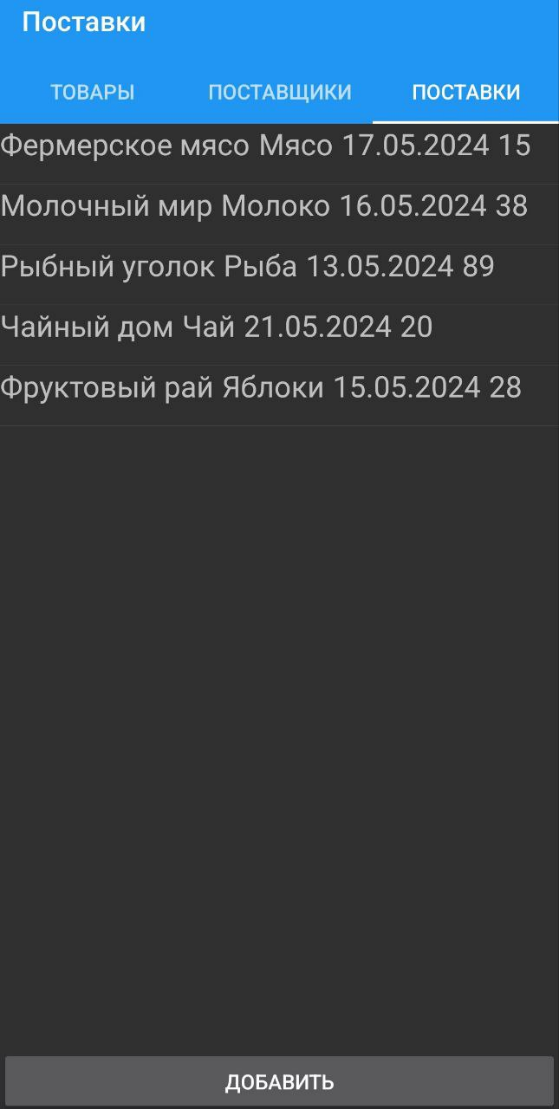
Код программы представлен в приложении А. Результат её работы представлен на рисунках 1 и 2.

Рисунок 1 – Страницы программы

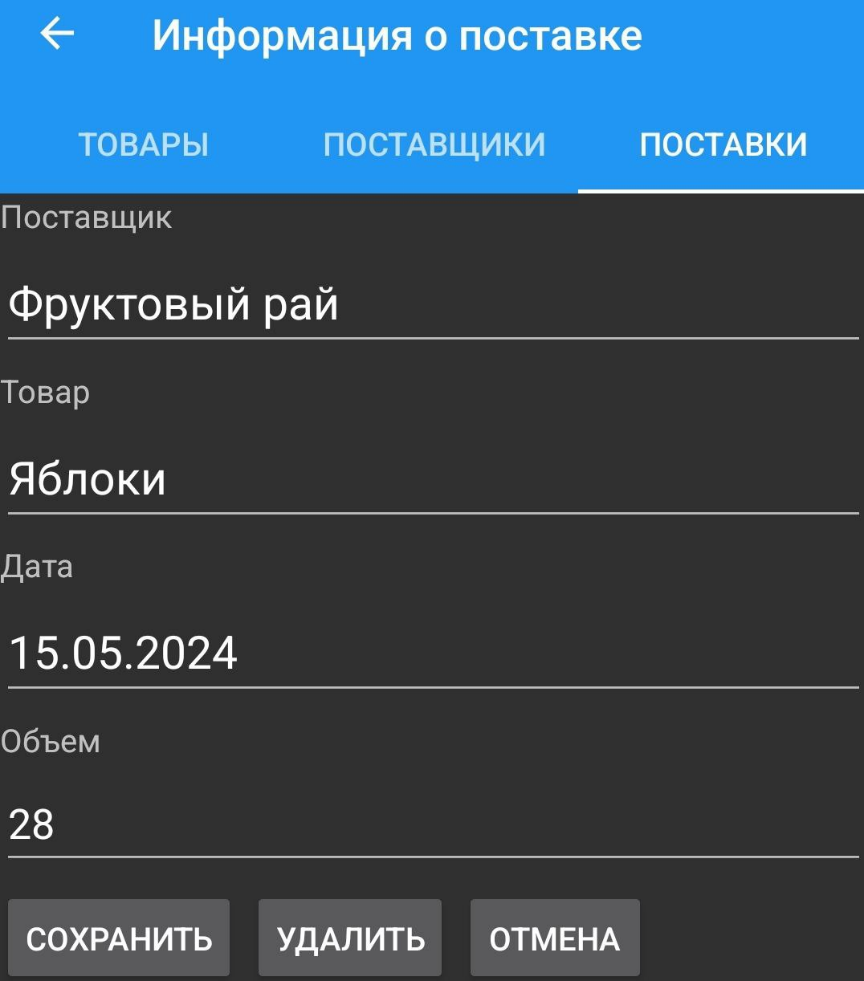
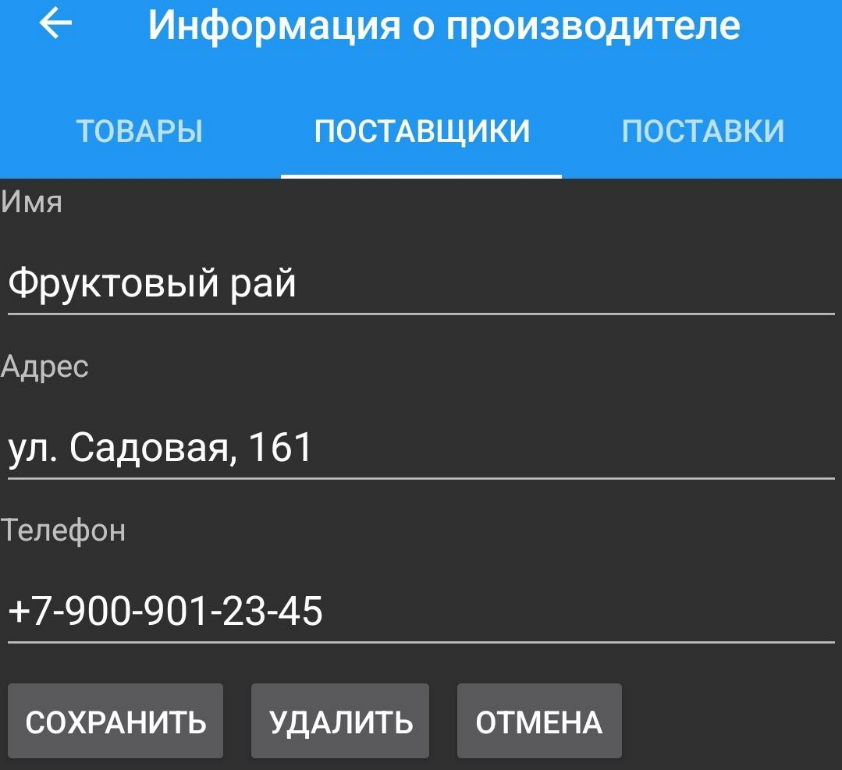
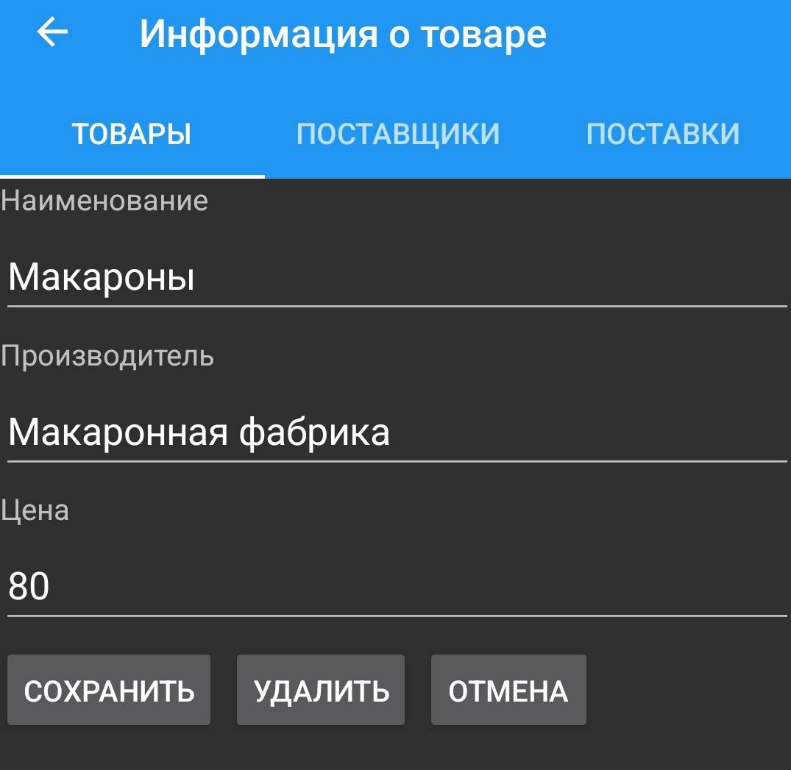


Рисунок 2 – Страницы добавления записей

Приложение А. Текст программы

**DBService.cs**

using SQLite;

using System;

using System.Collections.Generic;

namespace SupplyMobileApp

{

public class DBService

{

SQLiteConnection \_database;

public DBService(string databasePath)

{

\_database = new SQLiteConnection(databasePath);

\_database.CreateTable<Item>();

\_database.CreateTable<Supplier>();

\_database.CreateTable<Supply>();

}

/// Item

public IEnumerable<Item> GetItems()

{

return \_database.Table<Item>().ToList();

}

public Item GetItem(int id)

{

return \_database.Get<Item>(id);

}

public int DeleteItem(int id)

{

return \_database.Delete<Item>(id);

}

public int SaveItem(Item item)

{

if (item.Id != 0)

{

\_database.Update(item);

return item.Id;

}

else

{

return \_database.Insert(item);

}

}

/// Supplier

public IEnumerable<Supplier> GetSuppliers()

{

return \_database.Table<Supplier>().ToList();

}

public Supplier GetSupplier(int id)

{

return \_database.Get<Supplier>(id);

}

public int DeleteSupplier(int id)

{

return \_database.Delete<Supplier>(id);

}

public int SaveSupplier(Supplier supplier)

{

if (supplier.Id != 0)

{

\_database.Update(supplier);

return supplier.Id;

}

else

{

return \_database.Insert(supplier);

}

}

/// Supply

public IEnumerable<Supply> GetSupplies()

{

//\_database.DeleteAll<Supply>();

return \_database.Table<Supply>().ToList();

}

public Supply GetSupply(int supplier, int item, DateTime date)

{

return \_database.Find<Supply>(supply =>

supply.Supplier == supplier &&

supply.Item == item &&

supply.Date == date);

}

public void DeleteSupply(int supplier, int item, DateTime date)

{

\_database.Execute(

"DELETE FROM Supply WHERE Supplier = ? AND Item = ? AND Date = ?",

supplier, item, date);

}

public void SaveSupply(Supply supply)

{

var existingSupply = GetSupply(supply.Supplier, supply.Item, supply.Date);

if (existingSupply != null)

{

\_database.Execute(

"UPDATE Supply " +

"SET Volume = ? " +

"WHERE Supplier = ? AND Item = ? AND Date = ?",

supply.Volume, supply.Supplier, supply.Item, supply.Date);

}

else

{

\_database.Execute(

"INSERT INTO Supply (Supplier, Item, Date, Volume) " +

"VALUES (?, ?, ?, ?)",

supply.Supplier, supply.Item, supply.Date, supply.Volume);

}

}

}

}

**Item.cs**

using SQLite;

namespace SupplyMobileApp

{

[Table("Item")]

public class Item

{

[PrimaryKey, AutoIncrement]

public int Id { get; set; }

[MaxLength(50)]

public string Name { get; set; }

[MaxLength(50)]

public string Manufacturer { get; set; }

public decimal Price { get; set; }

}

}

**ItemPage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.ItemPage" Title="Информация о товаре">

<StackLayout>

<Label Text="Наименование" />

<Entry Text="{Binding Name}" />

<Label Text="Производитель" />

<Entry Text="{Binding Manufacturer}" />

<Label Text="Цена" />

<Entry Text="{Binding Price}" />

<StackLayout Orientation="Horizontal">

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

<Button Text="Удалить" Clicked="DeleteItem" />

<Button Text="Отмена" Clicked="Cancel" />

</StackLayout>

</StackLayout>

</ContentPage>

**ItemPage.xaml.cs**

using System;

using Xamarin.Forms;

namespace SupplyMobileApp

{

public partial class ItemPage : ContentPage

{

public ItemPage()

{

InitializeComponent();

}

private void SaveItem(object sender, EventArgs e)

{

var item = (Item)BindingContext;

if (!String.IsNullOrEmpty(item.Name))

{

App.Database.SaveItem(item);

}

Navigation.PopAsync();

}

private void DeleteItem(object sender, EventArgs e)

{

var item = (Item)BindingContext;

App.Database.DeleteItem(item.Id);

Navigation.PopAsync();

}

private void Cancel(object sender, EventArgs e)

{

Navigation.PopAsync();

}

}

}

**ItemTablePage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.ItemTablePage" Title="Товары">

<StackLayout>

<ListView x:Name="itemList" ItemsSource="{Binding}"

ItemSelected="OnItemSelected">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<ViewCell.View>

<StackLayout Orientation="Horizontal">

<Label Text="{Binding Name}" FontSize="Medium" />

</StackLayout>

</ViewCell.View>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button Text="Добавить" Clicked="CreateItem" />

</StackLayout>

</ContentPage>

**ItemTablePage.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace SupplyMobileApp

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class ItemTablePage : ContentPage

{

public ItemTablePage()

{

InitializeComponent();

}

protected override void OnAppearing()

{

itemList.ItemsSource = App.Database.GetItems();

base.OnAppearing();

}

private async void OnItemSelected(object sender, SelectedItemChangedEventArgs e)

{

Item selectedItem = (Item)e.SelectedItem;

ItemPage itemPage = new ItemPage();

itemPage.BindingContext = selectedItem;

await Navigation.PushAsync(itemPage);

}

private async void CreateItem(object sender, EventArgs e)

{

Item item = new Item();

ItemPage itemPage = new ItemPage();

itemPage.BindingContext = item;

await Navigation.PushAsync(itemPage);

}

}

}

**MainPage.xaml**

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

<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"

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

xmlns:local="clr-namespace:SupplyMobileApp"

x:Class="SupplyMobileApp.MainPage">

<NavigationPage Title="Товары">

<x:Arguments>

<local:ItemTablePage />

</x:Arguments>

</NavigationPage>

<NavigationPage Title="Поставщики">

<x:Arguments>

<local:SupplierTablePage />

</x:Arguments>

</NavigationPage>

<NavigationPage Title="Поставки">

<x:Arguments>

<local:SupplyTablePage />

</x:Arguments>

</NavigationPage>

</TabbedPage>

**MainPage.xaml.cs**

using System;

using Xamarin.Forms;

namespace SupplyMobileApp

{

public partial class MainPage : TabbedPage

{

public MainPage()

{

InitializeComponent();

}

}

}

**Supplier.cs**

using SQLite;

namespace SupplyMobileApp

{

[Table("Supplier")]

public class Supplier

{

[PrimaryKey, AutoIncrement]

public int Id { get; set; }

[MaxLength(50)]

public string Name { get; set; }

[MaxLength(100)]

public string Address { get; set; }

[MaxLength(17)]

public string Phone { get; set; }

}

}

**SupplierPage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.SupplierPage" Title="Информация о производителе">

<StackLayout>

<Label Text="Имя" />

<Entry Text="{Binding Name}" />

<Label Text="Адрес" />

<Entry Text="{Binding Address}" />

<Label Text="Телефон" />

<Entry Text="{Binding Phone}" />

<StackLayout Orientation="Horizontal">

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

<Button Text="Удалить" Clicked="DeleteSupplier" />

<Button Text="Отмена" Clicked="Cancel" />

</StackLayout>

</StackLayout>

</ContentPage>

**SupplierPage.xaml.cs**

using System;

using Xamarin.Forms;

namespace SupplyMobileApp

{

public partial class SupplierPage : ContentPage

{

public SupplierPage()

{

InitializeComponent();

}

private void SaveSupplier(object sender, EventArgs e)

{

var supplier = (Supplier)BindingContext;

if (!String.IsNullOrEmpty(supplier.Name))

{

App.Database.SaveSupplier(supplier);

}

Navigation.PopAsync();

}

private void DeleteSupplier(object sender, EventArgs e)

{

var supplier = (Supplier)BindingContext;

App.Database.DeleteSupplier(supplier.Id);

Navigation.PopAsync();

}

private void Cancel(object sender, EventArgs e)

{

Navigation.PopAsync();

}

}

}

**SupplierTablePage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.SupplierTablePage"

Title="Поставщики">

<StackLayout>

<ListView x:Name="supplierList" ItemsSource="{Binding}" ItemSelected="OnSupplierSelected">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<ViewCell.View>

<StackLayout Orientation="Horizontal">

<Label Text="{Binding Name}" FontSize="Medium" />

</StackLayout>

</ViewCell.View>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button Text="Добавить" Clicked="CreateSupplier" />

</StackLayout>

</ContentPage>

**SupplierTablePage.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace SupplyMobileApp

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class SupplierTablePage : ContentPage

{

public SupplierTablePage()

{

InitializeComponent();

}

protected override void OnAppearing()

{

supplierList.ItemsSource = App.Database.GetSuppliers();

base.OnAppearing();

}

private async void OnSupplierSelected(object sender, SelectedItemChangedEventArgs e)

{

Supplier selectedSupplier = (Supplier)e.SelectedItem;

SupplierPage itemPage = new SupplierPage();

itemPage.BindingContext = selectedSupplier;

await Navigation.PushAsync(itemPage);

}

private async void CreateSupplier(object sender, EventArgs e)

{

Supplier item = new Supplier();

SupplierPage itemPage = new SupplierPage();

itemPage.BindingContext = item;

await Navigation.PushAsync(itemPage);

}

}

}

**Sypply.cs**

using SQLite;

using System;

namespace SupplyMobileApp

{

[Table("Supply")]

public class Supply

{

[Indexed(Name = "PK\_Supply", Order = 1, Unique = true)]

public int Supplier { get; set; }

[Indexed(Name = "PK\_Supply", Order = 2, Unique = true)]

public int Item { get; set; }

[Indexed(Name = "PK\_Supply", Order = 3, Unique = true)]

public DateTime Date { get; set; }

public int Volume { get; set; }

}

}

**SupplyPage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.SupplyPage"

Title="Информация о поставке">

<StackLayout>

<Label Text="Поставщик" />

<Picker x:Name="supplierPicker" />

<Label Text="Товар" />

<Picker x:Name="itemPicker" />

<Label Text="Дата" />

<DatePicker x:Name="datePicker" />

<Label Text="Объем" />

<Entry Text="{Binding Volume}" Keyboard="Numeric" />

<StackLayout Orientation="Horizontal">

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

<Button Text="Удалить" Clicked="DeleteSupply" />

<Button Text="Отмена" Clicked="Cancel" />

</StackLayout>

</StackLayout>

</ContentPage>

**SupplyPage.xaml.cs**

using System;

using System.Collections.Generic;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace SupplyMobileApp

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class SupplyPage : ContentPage

{

public Supply supply { get; set; }

public List<Supplier> Suppliers { get; set; }

public List<Item> Items { get; set; }

public SupplyPage(Supply supply = null)

{

InitializeComponent();

if (supply == null)

{

supply = new Supply

{

Date = DateTime.Now // Устанавливаем текущую дату по умолчанию

};

}

this.supply = supply;

LoadData();

// Устанавливаем текущую дату в DatePicker

if (supply.Date == default(DateTime))

{

datePicker.Date = DateTime.Now;

}

else

{

datePicker.Date = supply.Date;

}

BindingContext = this.supply;

}

private void LoadData()

{

Suppliers = App.Database.GetSuppliers() as List<Supplier>;

Items = App.Database.GetItems() as List<Item>;

supplierPicker.ItemsSource = Suppliers;

itemPicker.ItemsSource = Items;

supplierPicker.ItemDisplayBinding = new Binding("Name");

itemPicker.ItemDisplayBinding = new Binding("Name");

if (supply.Supplier != 0)

{

var selectedSupplier = Suppliers.Find(s => s.Id == supply.Supplier);

if (selectedSupplier != null)

{

supplierPicker.SelectedItem = selectedSupplier;

}

}

if (supply.Item != 0)

{

var selectedItem = Items.Find(i => i.Id == supply.Item);

if (selectedItem != null)

{

itemPicker.SelectedItem = selectedItem;

}

}

}

private async void SaveSupply(object sender, EventArgs e)

{

if (supplierPicker.SelectedItem == null || itemPicker.SelectedItem == null)

{

await DisplayAlert("Ошибка", "Пожалуйста, выберите поставщика и товар", "OK");

return;

}

var bindingInfo = BindingContext as Supply;

supply.Supplier = ((Supplier)supplierPicker.SelectedItem).Id;

supply.Item = ((Item)itemPicker.SelectedItem).Id;

supply.Date = datePicker.Date; // Используем выбранную дату

supply.Volume = bindingInfo.Volume;

App.Database.SaveSupply(supply);

await Navigation.PopAsync();

}

private async void DeleteSupply(object sender, EventArgs e)

{

App.Database.DeleteSupply(supply.Supplier, supply.Item, supply.Date);

await Navigation.PopAsync();

}

private async void Cancel(object sender, EventArgs e)

{

await Navigation.PopAsync();

}

}

}

**SupplyTablePage.xaml**

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

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

x:Class="SupplyMobileApp.SupplyTablePage"

Title="Поставки">

<StackLayout>

<ListView x:Name="supplyList"

ItemsSource="{Binding}"

ItemSelected="OnSupplySelected">

<ListView.ItemTemplate>

<DataTemplate>

<ViewCell>

<ViewCell.View>

<StackLayout Orientation="Horizontal">

<Label Text="{Binding SupplierName}" FontSize="Medium" />

<Label Text="{Binding ItemName}" FontSize="Medium" />

<Label Text="{Binding Date, StringFormat='{0:dd.MM.yyyy}'}" FontSize="Medium" />

<Label Text="{Binding Volume}" FontSize="Medium" />

</StackLayout>

</ViewCell.View>

</ViewCell>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<Button Text="Добавить" Clicked="CreateSupply" />

</StackLayout>

</ContentPage>

**SupplyTablePage.xaml.cs**

using System;

using System.Linq;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

using System.Collections.Generic;

namespace SupplyMobileApp

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class SupplyTablePage : ContentPage

{

public SupplyTablePage()

{

InitializeComponent();

}

protected override void OnAppearing()

{

base.OnAppearing();

LoadSupplies();

}

private void LoadSupplies()

{

var supplies = App.Database.GetSupplies();

var suppliers = App.Database.GetSuppliers().ToDictionary(s => s.Id, s => s.Name);

var items = App.Database.GetItems().ToDictionary(i => i.Id, i => i.Name);

var supplyDisplayModels = supplies.Select(supply => new SupplyDisplayModel

{

SupplierName = suppliers.ContainsKey(supply.Supplier) ? suppliers[supply.Supplier] : "Unknown",

ItemName = items.ContainsKey(supply.Item) ? items[supply.Item] : "Unknown",

Date = supply.Date,

Volume = supply.Volume,

SupplierId = supply.Supplier,

ItemId = supply.Item

}).ToList();

supplyList.ItemsSource = supplyDisplayModels;

}

private async void OnSupplySelected(object sender, SelectedItemChangedEventArgs e)

{

if (e.SelectedItem == null)

return;

var selectedSupplyDisplayModel = e.SelectedItem as SupplyDisplayModel;

var selectedSupply = App.Database.GetSupplies().FirstOrDefault(s =>

s.Supplier == selectedSupplyDisplayModel.SupplierId &&

s.Item == selectedSupplyDisplayModel.ItemId &&

s.Date == selectedSupplyDisplayModel.Date);

if (selectedSupply != null)

{

SupplyPage supplyPage = new SupplyPage(selectedSupply);

await Navigation.PushAsync(supplyPage);

}

((ListView)sender).SelectedItem = null;

}

private async void CreateSupply(object sender, EventArgs e)

{

Supply supply = new Supply();

SupplyPage supplyPage = new SupplyPage(supply);

await Navigation.PushAsync(supplyPage);

}

}

public class SupplyDisplayModel

{

public string SupplierName { get; set; }

public string ItemName { get; set; }

public DateTime Date { get; set; }

public int Volume { get; set; }

public int SupplierId { get; set; }

public int ItemId { get; set; }

}

}