Task:

Create a mobile Android application for storing and managing books. It should add new book by defining: the title, the first author and the year of publish. The application should modify and delete the selected book in the separate views.

1. Create an empty cross-platform application (Xamarin.Forms) and named it as Book. Specify the location of the project (Fig. 1). Choose empty template (Fig. 2).

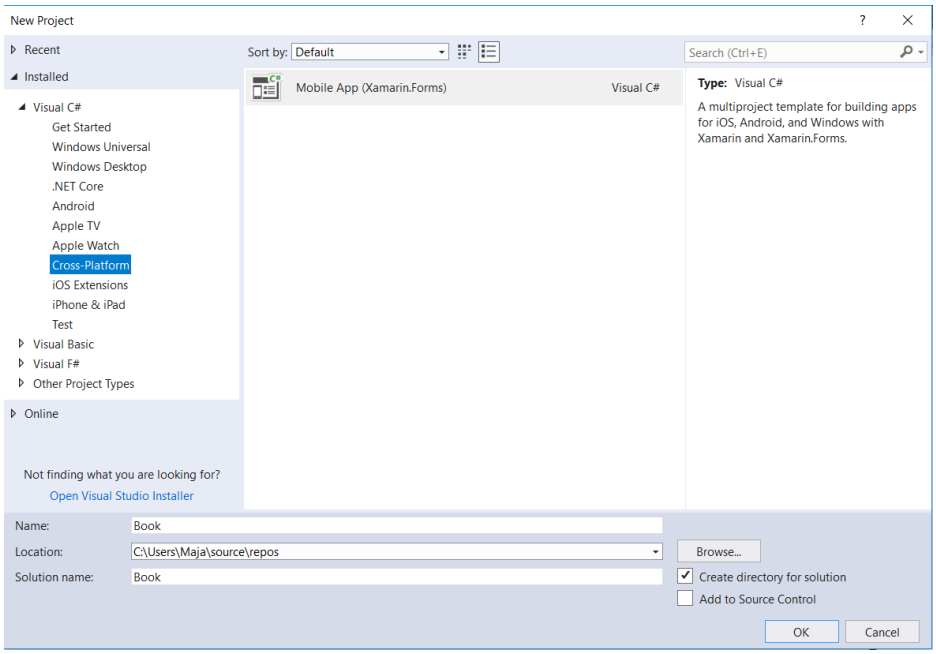


Fig. 1. Creating the new project.

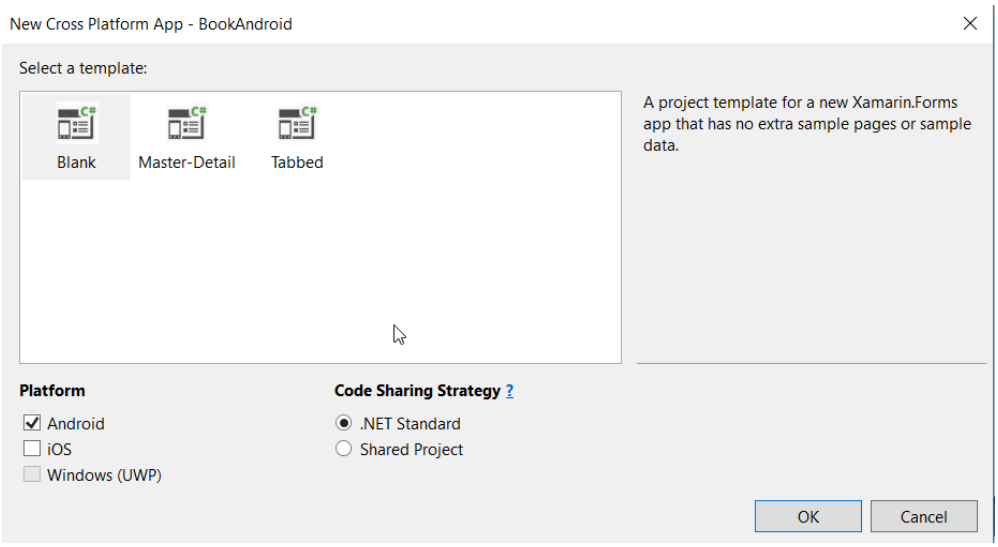


Fig. 2. Choosing the empty project for Android platform.

2. Go to Solution Explorer-> Project Name-> Manage NuGet packages for solution. Then, right-click to "Manage NuGet Packages" and search for SQLite.Net. Install the sqlite.net package (Fig. 3).

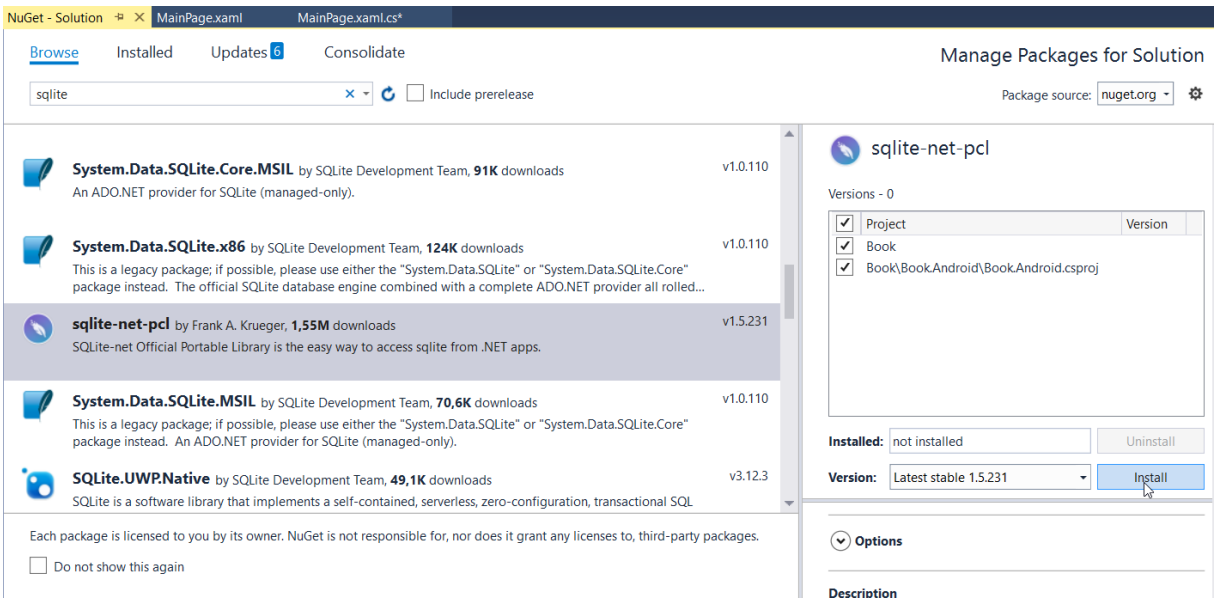


Fig. 3. Adding SQLite to the project.

3. Add a new class to create one data layer. Go to Solution Explorer-> Project Name and right click to Add item. Open a new Dialog box. Then, select **Class** and give it a name, such as, Book.cs. Click Add.

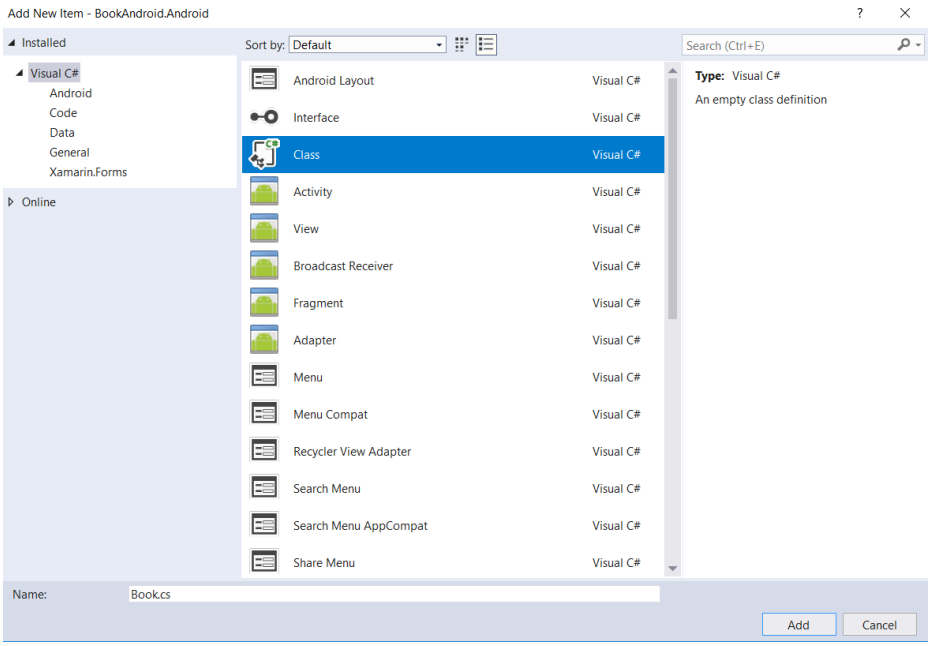


Fig. 4. Adding a class to the project.

4. In the Book.cs the table Book should be defined. Add id as primary key. It should be auto increment (Listing 1).

```
using System;
using System.Collections.Generic;
using System.Text;
using SQLite;

namespace Book
{
    public class Book
    {
        [PrimaryKey, AutoIncrement]
        public int id { get; set; }
        public string title { get; set; }
        public string firstAuthor { get; set; }
        public int publishYear { get; set; }
    }
}
```

Listing 1. Defining the Book class.

5. Create a class that represents the database and keep all the logic to access the database and its table within this class. For this, you should create a class named **BookDatabase.cs**.
6. In the class you should add methods for: creating the database, inserting a row to the table, and selecting all rows (Listing 2).

```
class BookDatabase
    {
        string path =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);

        public bool createDatabase()
        {
            try {
                using (var connection = new
                SQLiteConnection(System.IO.Path.Combine(path, "Book.db")))
                {
                    connection.CreateTable<Book>();
                    return true;
                }
            }
            catch (SQLiteException ex)
            {
                Debug.WriteLine("SQLiteEx" + ex.Message);
                return false;
            }

        }

        //Insert new book
        public bool insertNewBook(Book newBook)
        {
            try
            {
                using (var connection = new
                 SQLiteConnection(System.IO.Path.Combine(path, "Book.db")))
                {
                    connection.Insert(newBook);
                    return true;
                }
            }
            catch (SQLiteException ex)
            {
                Debug.WriteLine("SQLiteEx" + ex.Message);
                return false;
            }
```

```
        }

        select all rows
        public List<Book> selectAllBooks()
        {
            try
            {
                using (var connection = new
              SQLiteConnection(System.IO.Path.Combine(path, "Book.db")))
                {
                    return connection.Table<Book>().ToList();
                }
            }
            catch (SQLiteException ex)
            {
                Debug.WriteLine("SQLiteEx" + ex.Message);
                return null;
            }
        }

}
```

Listing 2. Creating the database, inserting a row to the table, and selecting all rows to a list.

7. In MainPage.xaml define a button “Add book” and a list view that consists of three labels (for title, first author and year of publish). The code is presented in Listing 3.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:Book"
             x:Class="Book.MainPage">

    <StackLayout>
        <!-- Place new controls here -->
        <Button Text="Add book" Clicked="Button_Clicked"/>
        <ListView x:Name="bookList" ItemSelected="BookList_ItemSelected">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout Orientation="Horizontal">

                            <Label Text="{Binding firstAuthor}"/>
                            <Label Text="."/>
                            <Label Text="{Binding title}"/>
                            <Label Text=":"/>
                            <Label Text="{Binding publishYear}"/>

                        </StackLayout>
                    </ViewCell>
                </DataTemplate>

            </ListView.ItemTemplate>
        </ListView>
    </StackLayout>

</ContentPage>
```

Listing 3. The code for controls in MainPage.

8. Add creating the database and method for button clicked (Listing 4).

```
public MainPage()
        {
            InitializeComponent();
            db = new BookDatabase();
            db.createDatabase();
            bookList.ItemsSource = db.selectAllBooks();

        }
private void Button_Clicked(object sender, EventArgs e)
        {
            Navigation.PushModalAsync(new AddingBook());
        }
```

Listing 4. The codein MainPage.cs.

9. Add a new content page “AddingBook” (Fig. 5).

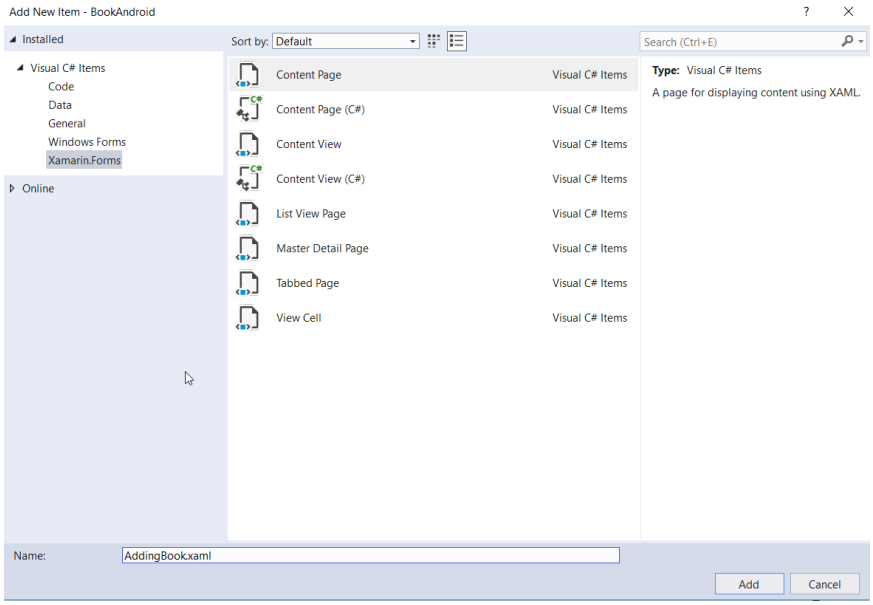


Fig. 5. Adding AddingBook content page.

10. In the AddingBook.xaml define three Entry controls and two buttons: “Add book” and “Cancel”. After tapping “Add book” the program should check if all entries are filled in, then add a new record and go back to the main page. “Cancel” button should move the users to the main page. The controls are presented in Listing 5, while button clicked method in Listing 6.

```
?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="Book.AddingBook">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Please insert book title" />
            <Entry x:Name="title" Placeholder="Title"/>
            <Label Text="Please insert book first author" />
            <Entry x:Name="author" Placeholder="First author"/>
            <Label Text="Please insert year of publishment" />
            <Entry x:Name="year" Placeholder="Year of publishment"
                                          Keyboard="Numeric" />
```

```
            <Button Text="Add new book" Clicked="Button_Clicked"/>
            <Button Text=" Cancel" Clicked="Button_Clicked_1"/>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Listing 5. The code for controls in AddingBook.

```
namespace Book
{
       [XamlCompilation(XamlCompilationOptions.Compile)]
       public partial class AddingBook : ContentPage
       {
        BookDatabase bookdb;

        public AddingBook ()
             {
                    InitializeComponent ();
                    bookdb = new BookDatabase();
             }

        private void Button_Clicked(object sender, EventArgs e)
        {
            //create a new Book object
            var newBook = new Book();
            newBook.title = title.Text;
            newBook.firstAuthor = author.Text;
            newBook.publishYear = Convert.ToInt32(year.Text);

            if ((newBook.title != String.Empty) && (newBook.firstAuthor !=
                String.Empty) && (newBook.publishYear != 0))
            {
                //insert new row
                bookdb.insertNewBook(newBook);
                //go back to main page
                Navigation.PopModalAsync();
            }
        }
        //Cancel
        private void Button_Clicked_1(object sender, EventArgs e)
        {
            //go back to main page
            Navigation.PopModalAsync();
        }
    }
}
```

Listing 6. The code in AddingBook.

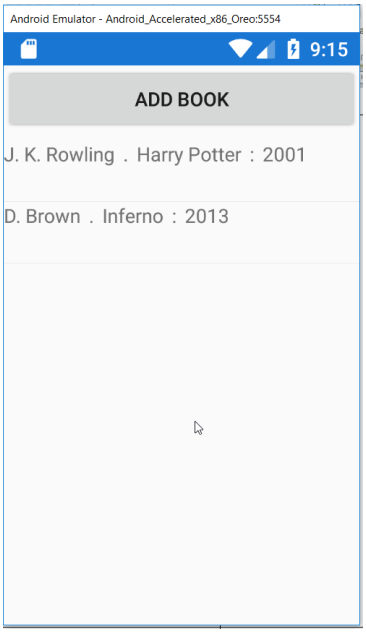
11. In the main page add

```
protected override void OnAppearing()
        {
            //base.OnAppearing();
            bookList.ItemsSource = db.selectAllBooks();
        }
```

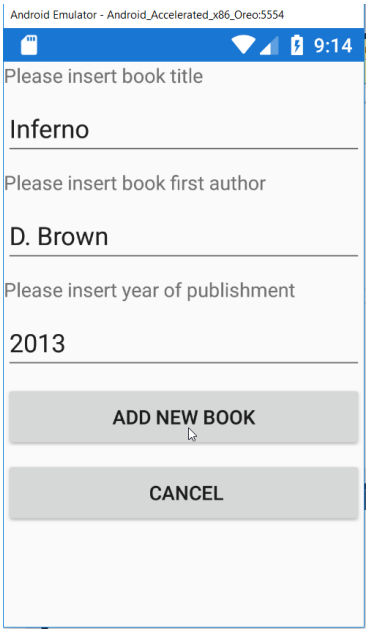
Listing 7. OnAppearing method.

12. Create a new content page for editing and deleting rows. The user should be moved to this content after tapping a list item on the main page.

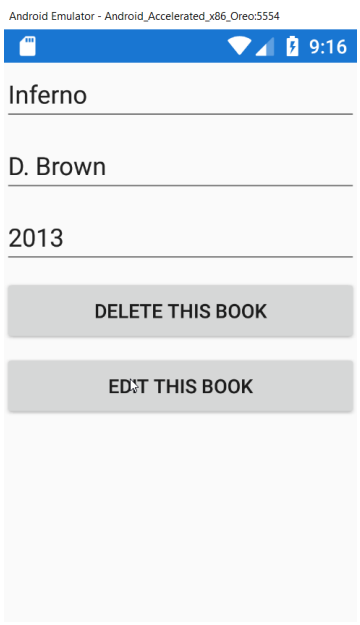
The application should look like this:



MainPage



AddingBook



EditingBook