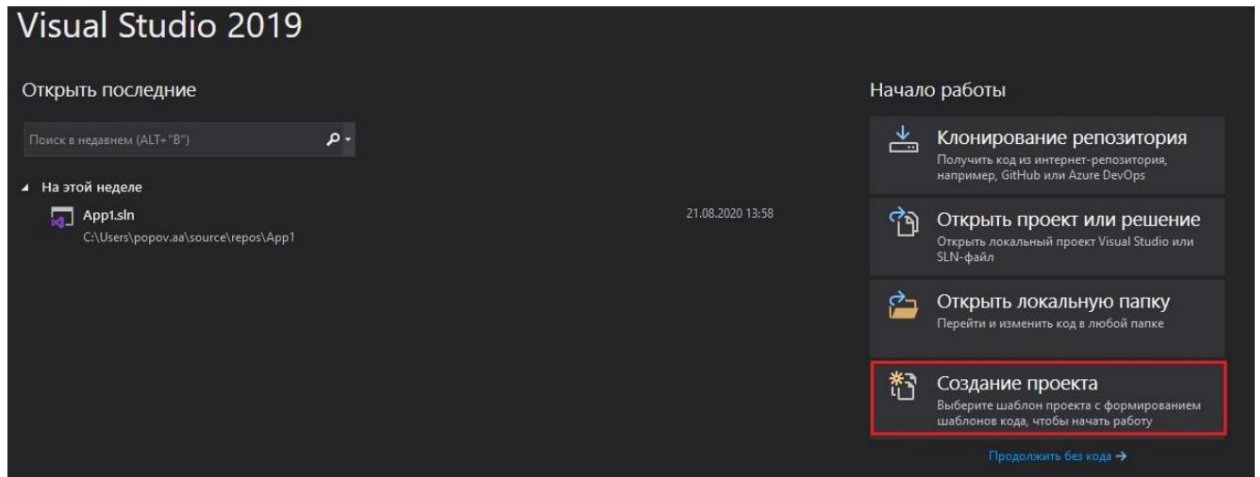
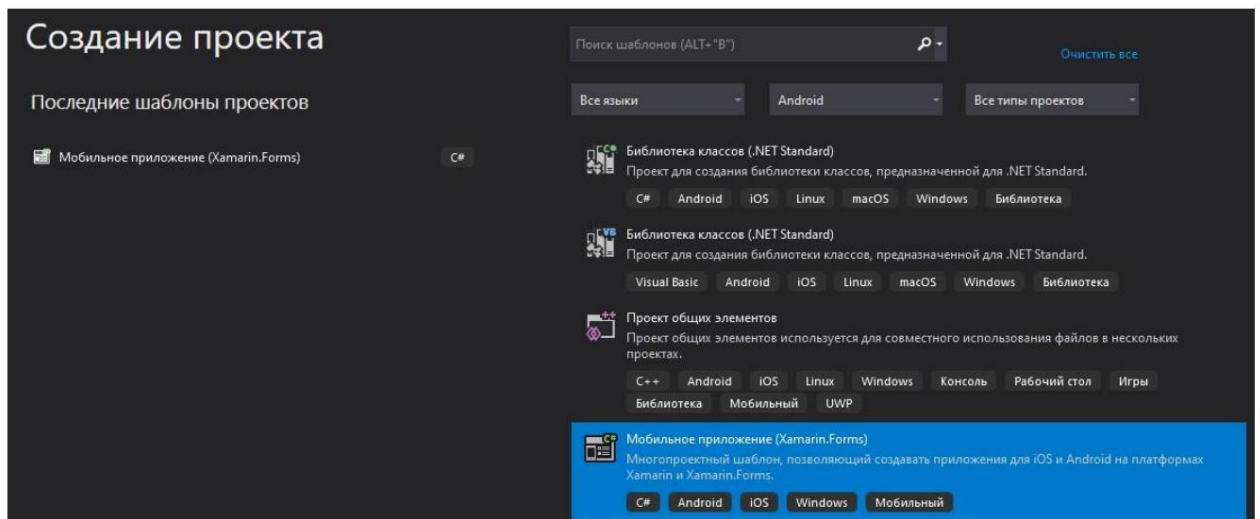


Tạo một dự án bằng Xamarin

Bài học này nói về việc tạo một dự án thí điểm với sử dụng nền tảng Xamarin. Để tạo một dự án, bạn phải chạy Visual Studio.Net. Tiếp theo, bạn cần nhấp vào Tạo dự án.



Trong hộp thoại Tạo dự án , chọn mẫu Ứng dụng di động (Xamarin.Forms) và nhấp vào Tiếp theo



Đặt tên dự án thành Phoneword, xác định vị trí dự án và nhấp vào nút “Tạo”

Настроить новый проект

Мобильное приложение (Xamarin.Forms) C# Android iOS Windows Мобильный

Имя проекта

Phoneword

Расположение

C:\Users\popov.aa\Desktop\ ...

Имя решения ⓘ

Phoneword

☒ Поместить решение и проект в одном каталоге

Назад Создать

Trong hộp thoại Ứng dụng dành cho thiết bị di động mới , hãy nhấp vào mẫu "Trống" và nhấp vào nút "Tạo" để tạo dự án mới

Новое мобильное приложение

Выберите шаблон приложения.

Всплывающий элемент
Приложение с боковым меню, которое можно свернуть на небольших экранах.

С вкладками
Приложение, которое использует вкладки для перехода между разделами.

Пустой
Пустое приложение с одним начальным экраном.

Я хочу разработать приложение для следующих платформ:

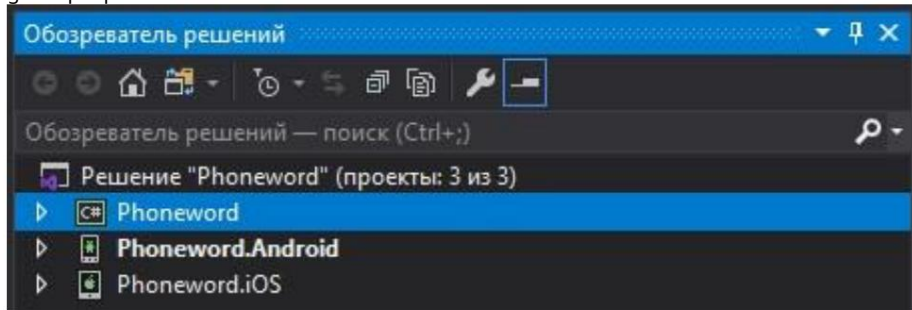
☒ Android

☒ iOS

☐ Windows (UWP)

Назад Создать

Sau khi tạo dự án, hãy vào Solution Explorer. Chúng ta có thể nhận thấy rằng có ba dự án được tạo trong giải pháp.



Tất cả chúng đều có cùng tên gốc, nhưng hai tên có hậu tố (".iOS" hoặc ".Android").

Chúng ta sẽ tạo một ứng dụng phần mềm một trang có chức năng chuyển đổi một chuỗi chữ và số do người dùng nhập thành một chuỗi số được hiển thị trong giao diện người dùng.

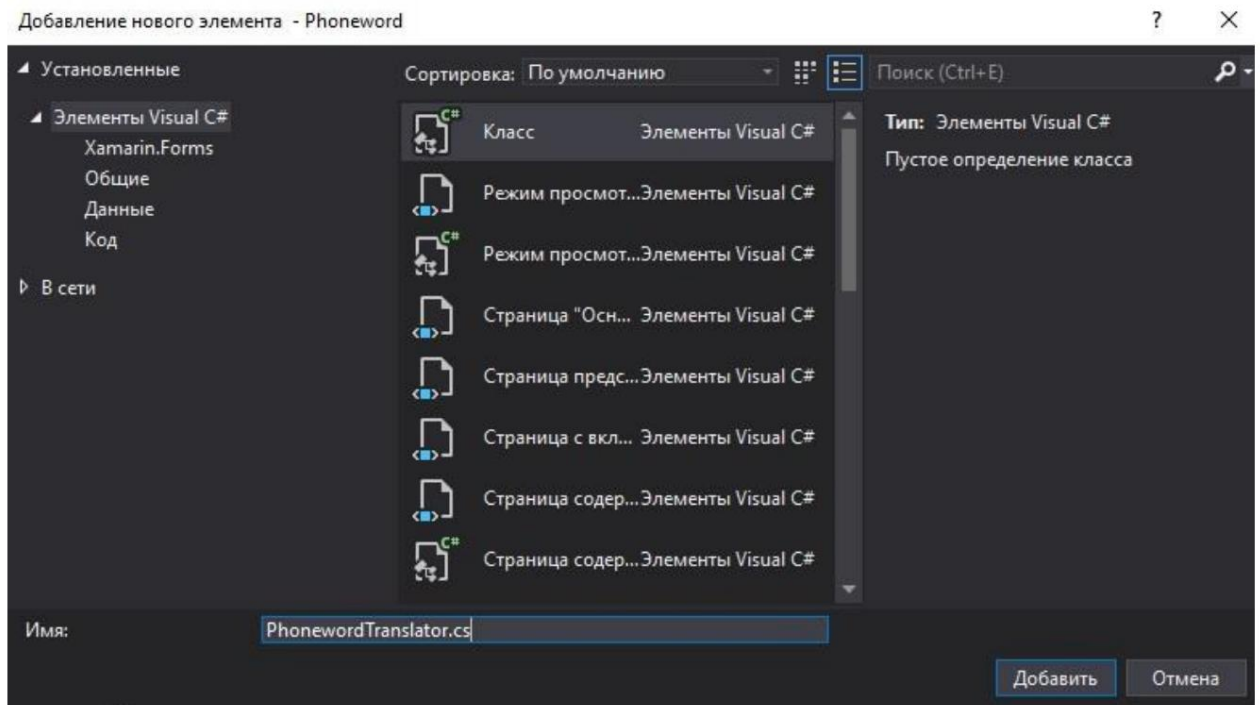
Để làm cơ sở cho việc dịch thuật, ứng dụng sẽ sử dụng các chữ cái tương ứng với các số được hiển thị trên bàn phím của điện thoại phổ thông.



Ví dụ: chuỗi "CAB" phải được chuyển đổi thành chuỗi "222", vì số "2" trên bàn phím tương ứng với cả ba chữ cái trong chuỗi "ABC". Hãy thêm một tệp mới có mã C# vào dự án. Để thực hiện việc này, bạn cần

nhấp chuột phải vào dự án có tên Phoneword và chọn Thêm>Tạo phần tử và trong

hộp thoại "Thêm phần tử mới" mở ra, chọn mẫu "Lớp", nhập tên lớp PhonewordTranslator.cs và nhấp vào nút "Thêm" .



Một lớp C Sharp PhonewordTranslator trống đư ợc tạo. Nhập
chư ơng trình sau vào lớp PhonewordTranslator

mã số:

lớp tĩnh công khai PhonewordTranslator

```
{ chuỗi tĩnh công khai ToNumber(chuỗi thô) {

    // Kiểm tra xem có thiếu chuỗi không
    if (string.IsNullOrEmpty(raw))
        return null;
    // Chuyển các ký tự chuỗi thành chữ hoa raw =
    raw.ToUpperInvariant();

    var newNumber = new StringBuilder();

    //Lặp qua các ký tự trong chuỗi raw
    foreach (var c in raw) {

        // Kiểm tra giá trị của thuộc tính // sự xuất
        hiện của một ký tự trong chuỗi "số" if (
        -0123456789".Contains(c)) // Nếu một ký tự
        đư ợc bao gồm trong chuỗi, thì // ký tự đó sẽ đư ợc thêm vào vào chuỗi
        newNumber newNumber.Append(c); else { // Nếu ký tự không có
        trong chuỗi thì // hàm dịch chữ
        cái

        thành số đư ợc gọi là var result = TranslateToNumber(c);
```

```
// kiểm tra kết quả thực thi // của hàm
TranslateToNumber if (result !=
null) // thêm một
    ký tự số // vào chuỗi newNumber thay
    vì một chữ cái newNumber.Append(result);

else // kết quả không tồn tại nếu hàm input // nhận đư ợc
    một ký tự không phải là chữ số // và không t ư ơng ứng với mảng chữ số return
    null;

}

} return newNumber.ToString();
}

// Hàm tìm kiếm một ký tự trong chuỗi và // trả về dấu
xuất hiện (không xuất hiện) // của ký tự trong chuỗi
static bool Chứa(chuỗi
này keyString, char c) {

    trả về keyString.IndexOf(c) >= 0;
}

// Mảng chuyển đổi chữ cái thành số chuỗi
chỉ đọc tĩnh[] chữ số = {
    "ABC", "DEF", "GHI", "JKL", "MNO", "PQRS", "TUV", "WXYZ" };

// Hàm chuyển chữ thành số static int?
TranslateToNumber(char c) {

    for (int i = 0; i < signature.Length; i+

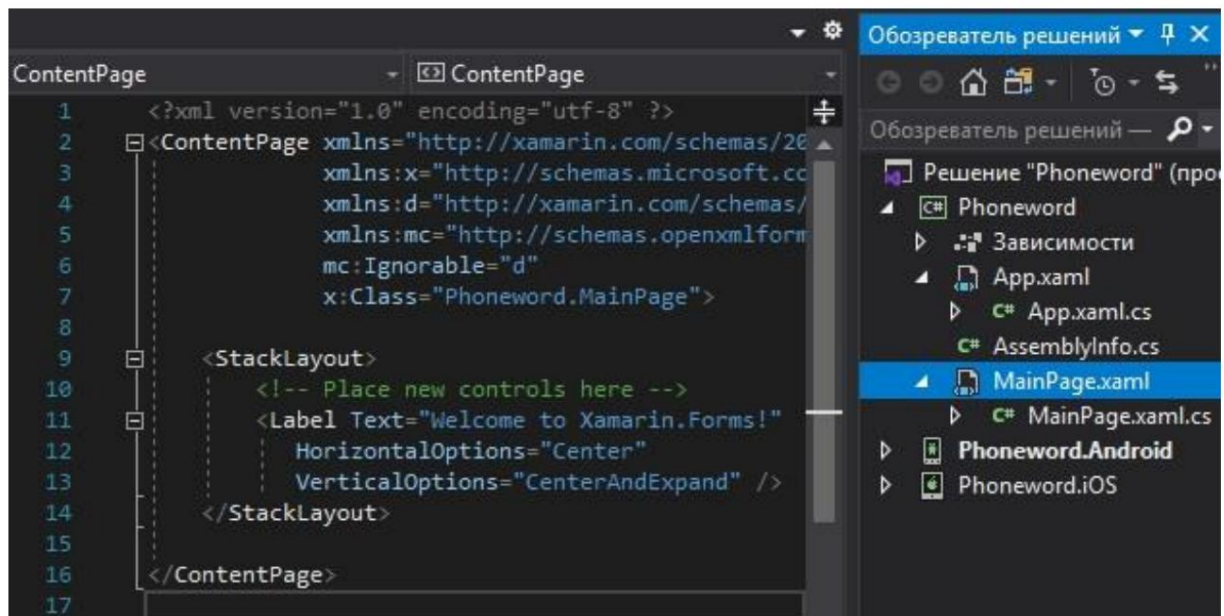
    +) { if (digits[i].Contains(c))
        return 2 + i;

    } trả về giá trị rỗng;
}
}
```

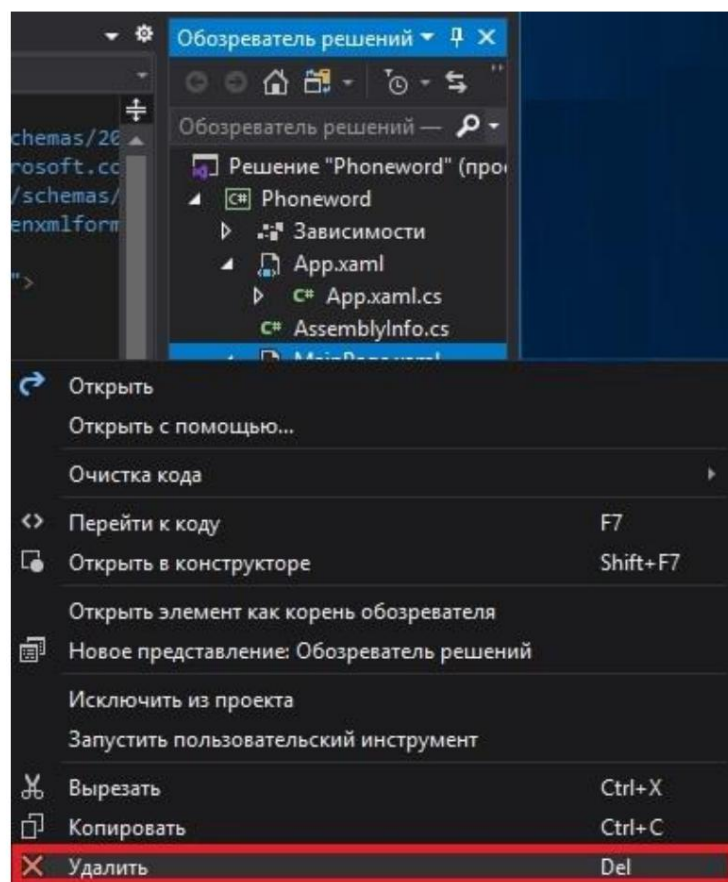
Lưu mã chương trình đã gõ.

Làm lại mã chương trình nằm trong file MainPage.xaml và tương ứng với giao diện người dùng.

Tạo giao diện người dùng của riêng bạn. Để thực hiện việc này, trước tiên bạn phải xóa mã giao diện người dùng được tạo tự động nằm trong tệp MainPage.xaml.



Để thực hiện việc này, trong Solution Explorer, bấm chuột phải vào tệp MainPage.xaml và chọn Xóa. Hành động này cũng xóa tệp mã MainPage.xaml.cs.



Тạo một lớp trong một tệp mới có tên MainPage.cs. Bạn có thể sử dụng quy trình tương tự được mô tả ở trên để tạo một tệp lớp mới.

Làm cho lớp bắt nguồn từ lớp `ContentPage`. Thêm câu lệnh sử dụng `Xamarin.Forms` vì `ContentPage` nằm trong không gian tên `Xamarin.Forms`.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using Xamarin.Forms;
5
6  namespace Phoneword
7  {
8      class MainPage: ContentPage
9      {
10     }
11 }
```

Bước tiếp theo là thêm mã để kết nối giao diện người dùng trong lớp `MainPage.cs` đã tạo. Ứng dụng phần mềm sử dụng bố cục `StackLayout`. Để làm việc với nó, bạn cần tạo một phiên bản của nó và đảm bảo rằng thuộc tính định hướng được đặt, thuộc tính này chúng ta cần đặt các điều khiển trong bố cục từ trên xuống dưới. Nếu sử dụng hướng bố cục dọc thì mỗi phần tử

các điều khiển bên trong bố cục sẽ lấp đầy toàn bộ chiều rộng của bố cục. Để

làm cho bố cục của ứng dụng phần mềm trông hấp dẫn, phải có không gian trống giữa các thành phần của nó. Có một số thuộc tính có thể được sử dụng cho việc này trong

tùy theo mục tiêu mà ứng dụng phần mềm giải quyết.

Mỗi điều khiển có một thuộc tính `Lề`, được sử dụng trong bố cục để đặt khoảng cách giữa các điều khiển được đặt trong bố cục. Tất cả các bố cục đều có thuộc tính `Đệm`, do đó tất cả các điều khiển bên trong bố cục đều ở một khoảng cách nhất định so với đường viền

cách trình bày.

Thuộc tính `Đệm` của trang được đặt thành 20 cho tất cả các trường. Ngoài ra, một tham số để đặt không gian trống trong giao diện người dùng là thuộc tính `Khoảng cách` của bố cục `StackLayout`. Thuộc tính này chỉ định khoảng cách giữa tất cả các phần tử con của `StackLayout`. Mã này đặt thuộc tính `Spacing` của `StackLayout` thành 15. Thuộc tính này bổ sung cho `Margin` của chính điều khiển, do đó khoảng trống thực tế

giữa các điều khiển sẽ là giá trị kết hợp của `lề` và `khoảng cách` (nếu cả hai thuộc tính đều được áp dụng).

Các điều khiển được thêm vào bố cục bằng thuộc tính `Trẻ em`. Các phần tử sau đã được tạo trong mã chương

trình và được thêm vào bố cục:

điều khiển cho giao diện người dùng của ứng dụng phần mềm:

1. Điều khiển `Nhãn` có thuộc tính `Văn bản`, trong đó văn bản phụ trợ “Nhập từ điện thoại” được đặt (Nhập số từ - chuỗi chữ và số). 2. Kiểm soát mục nhập cho phép bạn nhập hoặc chỉnh sửa chuỗi chữ và số và được sử dụng để nhận dữ liệu đầu vào

từ người dùng. Bên trong điều khiển `Mục nhập`, thuộc tính `Văn bản` được khởi tạo thành “1-855-XAMARIN”. Người dùng có thể thay thế văn bản nguồn bằng văn bản của riêng họ, nhưng việc điền trước điều khiển là cần thiết để kiểm tra ứng dụng phần mềm. 3. Phần tử `Nút` chạy logic chương trình để chuyển đổi một chuỗi chữ và số thành chuỗi số. Thuộc tính `Text` của nó phải được đặt thành “Dịch”. 4. Phần tử `Nút` thứ hai hiển thị chuỗi số do chuỗi chữ và số được nhập trước đó. Thuộc tính `Text` của nó phải được đặt thành “Gọi”. Theo mặc định, khả năng nhấp của `Nút` này phải bị tắt bằng cách đặt thuộc tính `IsEnabled` thành `false`. Sau khi xây dựng

`StackLayout` với các điều khiển bên trong, bạn cần thêm bố cục vào thuộc tính `Nội dung` của `ContentPage`. Trong trường hợp này, `MainPage` sẽ hiển thị nội dung của `StackLayout` bằng cách sử dụng các quy tắc bố cục điều khiển được đặt trong bố cục của lớp `StackLayout`, cũng như các giá trị thuộc tính và giá trị của điều khiển.

cách trình bày.

Tiếp theo, bạn cần kết nối các sự kiện `Clicked` của nút `Translate` để chuyển chuỗi chữ cái thành chuỗi số. Để thực hiện việc này, hãy sử dụng trình xử lý sự kiện

`OnTranslate`.

Lấy đối tượng và `EventArgs` làm tham số.

Trình xử lý sự kiện phải chuyển đổi chuỗi chữ và số được nhập trong điều khiển `Entry` thành chuỗi số bằng cách sử dụng phương thức `PhonewordTranslator.ToNumber` của lớp `PhonewordTranslator` đã phát triển trước đó.

Chuỗi số được lưu trữ trong một chuỗi có tên là `số dịch`. Chuỗi chữ và số để chuyển đổi thêm sang

Chuỗi số được lấy từ thuộc tính `Text` của tùy chọn kiểm soát `Mục nhập`.

Phương thức `PhonewordTranslator.ToNumber` trả về `null` nếu chuỗi chữ và số không thể chuyển đổi thành chuỗi số. Tiếp theo, chúng ta sẽ thêm mã để thay đổi thuộc tính `Văn bản` của nút `Gọi` để nó hiển thị chuỗi số nếu được chuyển đổi thành công từ chuỗi chữ và số.

Nút Gọi có sẵn hoặc không có sẵn để nhấn tùy thuộc vào sự thành công của việc chuyển đổi cấu trúc chữ và số thành cấu trúc kỹ thuật số.

Nếu hàm TranslateNumber trả về giá trị rỗng thì nút Gọi sẽ không khả dụng và nếu chuyển đổi thành công thì nút này sẽ khả dụng. Để làm cho một nút có sẵn hoặc không thể bấm được

Thuộc tính IsEnabled được sử dụng như đã đề cập trước đó.

Nếu không thể chuyển đổi một chuỗi chữ và số (chuỗi chứa các ký tự không phải số và chữ cái) thì quá trình chuyển đổi sẽ không thành công và thuộc tính Văn bản của nút (điều khiển Button2) được đặt lại thành "Gọi" và trường Mục nhập sẽ xuất hiện trong trường Mục nhập của điều khiển. thông báo "Chuỗi lỗi".

sử dụng Hệ

thống; sử dụng System.Collections.Generic;

sử dụng System.Text;

sử dụng Xamarin.Forms;

không gian tên Từ ngữ {

lớp công khai MainPage: ContentPage {

Nhập số điện thoạiVăn bản;

Nút dịch Nút;

Nút gọiNút; chuỗi

được dịchSố;

MainPage công khai()

{

this.Padding = Độ dày mới (20, 20, 20, 20);

// Đặt thuộc tính bố cục

Bảng điều khiển StackLayout = StackLayout mới

{

Khoảng cách =

15, Định hướng = StackOrientation.Vertical };

// Thêm các điều khiển vào bố cục StackLayout panel.Children.Add(new Label {

Text = "Nhập từ điện thoại:",

FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)) }));

panel.Children.Add(phoneNumberText = Mục nhập mới {

Text="1-855-XAMARIN", }));

panel.Children.Add(translateButton = Nút mới {

```
        Text = "Dịch" });

panel.Children.Add(callButton = Nút mới {

    Text = "Gọi",
    IsEnabled = sai, });

// Xử lý việc nhấp vào nút Dịch
TranslateButton.Clicked += OnTranslate;

// Thêm bố cục vào trang ContentPage this.Content
= panel;
}

khoảng trống riêng tư OnTranslate(ngư ời gửi đối tư ợng, EventArgs e)

{ chuỗi đã nhậpNumber = phoneNumberText.Text; đã dịchSố =

    Phoneword.PhonewordTranslator.ToNumber(enteredNumber);

    if (!string.IsNullOrEmpty(translationNumber)) {

        callButton.IsEnabled = true;
        callButton.Text = "Gọi " + TranslateNumber;

    }

    khác {
        callButton.IsEnabled = false;
        callButton.Text = "Gọi";
        phoneNumberText.Text = "Chuỗi lỗi";
    }
}

}

}
```

Tiếp theo bạn cần phải lắp ráp một giải pháp. Để thực hiện việc này, hãy nhấp vào menu Xây dựng và chọn Giải pháp xây dựng từ menu thả xuống. Nếu giải pháp đã đư ợc tập hợp và các thay đổi đã đư ợc thực hiện đối với mã chư ơng trình thì bạn phải chọn Xây dựng lại giải pháp.

Đảm bảo không có lỗi trong hộp thoại Đầu ra

```

Вывод
Показать выходные данные из: Сборка
1>----- Перестроение всех файлов начато: проект: Phoneword, Конфигурация: Release Any CPU -----
1>Phoneword -> C:\Users\Popov.aa\Desktop\Phoneword\Phoneword\bin\Release\netstandard2.0\Phoneword.dll
2>----- Перестроение всех файлов начато: проект: Phoneword.Android, Конфигурация: Release Any CPU -----
Соединение для связывания с Mac не установлено, поэтому сборка будет проведена автономно. Установите соединение и повт
3>----- Перестроение всех файлов начато: проект: Phoneword.iOS, Конфигурация: Release iPhone -----
3> Executing SayHello Task to establish a connection to a Mac Server.
3> Properties: SessionId=c9862f5e9a7ecd3710da6cc023d2a18f,
3> ServerPort=,
3> ServerAddress=,
3> ServerUser=,
3> ServerPassword=,
3> SSHKey=,
3> SSHPassPhrase=,
3> AppName=Phoneword.iOS,
3> ContinueOnDisconnected=True
3>C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\MSBuild\Xamarin\iOS\Xamarin.Messaging.targets(56,3)
3> Phoneword.iOS -> C:\Users\Popov.aa\Desktop\Phoneword\Phoneword.iOS\bin\iPhone\Release\Phoneword.iOS.exe
2> Phoneword.Android -> C:\Users\Popov.aa\Desktop\Phoneword\Phoneword.Android\bin\Release\Phoneword.Android.dll
===== Перестроение всех проектов: успешно: 3, с ошибками: 0, пропущено: 0 =====

```

Xem video hướng dẫn cách hoạt động của phần mềm

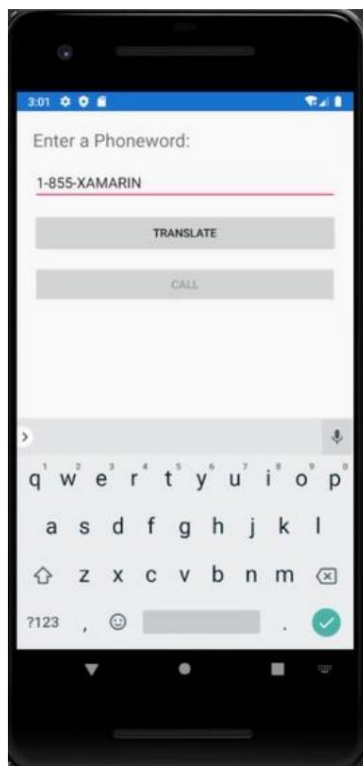
ứng dụng có thể được khởi chạy bằng cách khởi chạy một tệp video hiện có.

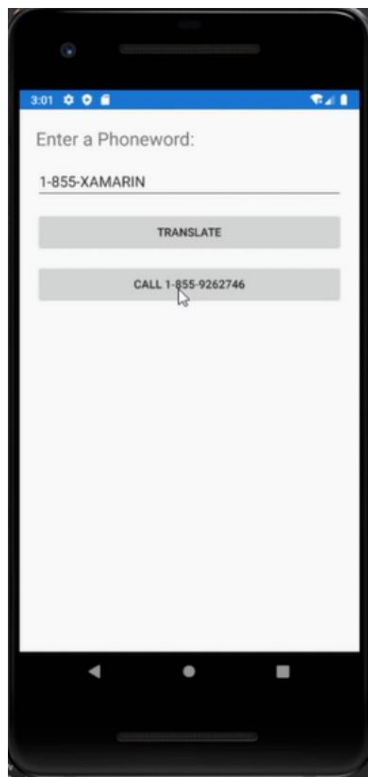
Bạn cần kiểm tra ứng dụng phần mềm bằng cách chạy nó trên thiết bị hoặc trình mô phỏng Android đã được định cấu hình trước đó.

Tiếp theo, bạn cần nhấp vào nút "TRANSLATE" để chuyển đổi chuỗi chữ và số "1-855-XAMARIN" thành chuỗi số.

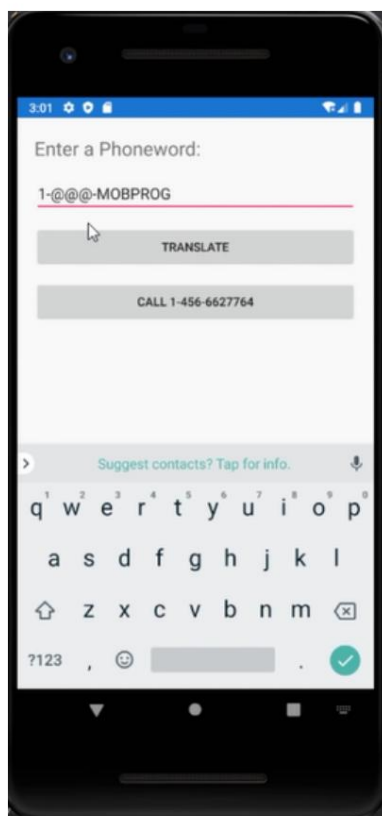
Nút có dòng chữ "Gọi" hiển thị một chuỗi văn bản bao gồm "Gọi" và chuỗi kỹ thuật số kết quả.

Trong trường hợp này, nút dưới cùng sẽ có sẵn để nhấn





Bây giờ, trong Entry control, bạn phải nhập chuỗi chữ và số "không chính xác" "1-@@@-MOBPROG", chứa ba ký tự không phải chữ cái cũng không phải số.



Tiếp theo, bạn cần nhấn vào nút có dòng chữ "TRANSLATE" để chuyển chuỗi chữ và số "1-@@@-MOBPROG" thành

dòng số. Trong trường hợp này, việc chuyển đổi chuỗi chữ và số "không chính xác" sẽ không thành công và giá trị của thuộc tính Văn bản phần dư ới cùng đư ợc gán giá trị "Gọi", nó sẽ không khả dụng đối với nhấn và chuỗi "Chuỗi lỗi" xuất hiện trong trường nhập Mục nhập (Hình.).

