Труды МАИ. 2022. № 125
Trudy MAI, 2022, no. 125

Научная статья
УДК 004.4’2
DOI: 10.34759/trd-2022-125-21

ОСОБЕННОСТИ КРОСС-ПЛАТФОРМЕННОЙ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ XAMARIN

Светлана Валентиновна Кузнецова
Московский авиационный институт (национальный исследовательский университет), МАИ,
Москва, Россия
k_svetlana_valen@mail.ru

Аннотация. В условиях цифровизации всех отраслей экономики, включая авиационную, наблюдается возрастание спроса на мобильные приложения. Большинство разработчиков мобильных приложений ориентируется на нативную разработку, так как накоплен большой опыт создания приложений любой сложности. При таком подходе под каждую операционную систему (ОС) создается отдельное приложение, что достаточно дорого и трудоемко. Существование в настоящее время двух мобильных платформ Apple iOS (iPhone и iPad) и Google Android диктует необходимость применения кросс-платформенного подхода, который позволяет создавать единый программный код для нескольких операционных систем одновременно. Кросс-платформенная разработка накладывает некоторые ограничения на функциональные возможности приложения, но при этом

позволяет оптимизировать стоимость и скорость разработки, а также поддержки приложения, обеспечивая при этом результат на выходе не менее качественный, чем при нативной разработке.

Кросс-платформенный подход сформировался относительно недавно и реализуется в кросс-платформенных фреймворках. Поэтому актуальным становится освоение новых сложных кросс-платформенных технологий и инструментов разработки для эффективного их использования.

В статье исследуется технология разработки кросс-платформенных мобильных приложений Xamarin: изложены принципы разработки кросс-платформенных мобильных приложений, представлены основные возможности, предоставляемые разработчикам в рамках этого подхода, его преимущества и недостатки. Рассмотрен круг вопросов, связанных с проектированием, реализацией и тестированием мобильных приложений: использование архитектурного паттерна MVVM, языка разметки XAML для описания пользовательского интерфейса, возможная среда разработки - Visual Studio 2019 или Visual Studio для Mac и Xcode.

.Технология Xamarin.Forms, имеет ряд преимуществ, в том числе:

- в процессе разработки создается единый код для всех платформ;
- Xamarin предоставляет прямой доступ к нативным API каждой платформы;
- при создании приложений можно использовать мощные возможности платформы .NET Core и языка программирования C#.

Эта технология хорошо подходит для разработки корпоративных мобильных приложений в различных отраслях.

Представлена методика для создания кросс-платформенных мобильных приложений с использованием Xamarin, содержащая краткое последовательное и всестороннее описание всех этапов разработки.

Ключевые слова. Мобильное приложение, кросс-платформенная разработка, операционные системы, Apple iOS (iPhone и iPad), Google Android, технология Xamarin, Xamarin.Forms, язык C#, язык разметки XAML, паттерн проектирования MVVM (Model/View/ViewModel), среда разработки, Visual Studio 2019, Visual Studio для Mac и Xcode.

***Для цитирования*:** Кузнецова С.В. Особенности кросс-платформенной разработки мобильных приложений с использованием Xamarin // Труды МАИ. 2022. № 125. DOI: 10.34759/trd-2022-125-21

Original article

FEATURES OF CROSS-PLATFORM MOBILE APPLICATIONS DEVELOPMENT USING XAMARIN

Svetlana V. Kuznetsova
Moscow Aviation Institute (National Research University),
Moscow, Russia
k_svetlana_valen@mail.ru

Abstract. Today, the development of mobile applications is one of the most actively developing sectors of the IT industry. In this regard, there are many projects for the development of applications on mobile platforms.

Digital transformation in the aviation industry covers the production of aircraft, as well as passenger and cargo transportation. As for transportation, the work of airlines, airports, interaction with passengers, customers of cargo transportation, the functioning of a unified air traffic management system deserves special attention.

One of the technological trends actively developing today in terms of digitalization of the aviation industry is the development of mobile applications for aviation, including aircraft construction, air transportation and airport services.

Mobile applications provide significantly faster access to data and are an effective means of interaction regardless of the user's location. For example, they provide quick access to technical information and direct data entry, optimizing engineering and production processes.

The classic version of mobile application development is native development, in which a separate application is developed for each operating system (OS), which is quite expensive and time—consuming. An alternative is a cross-platform technology, the main purpose of which is to provide developers with a tool for parallel creation of applications for multiple operating systems at the same time. This will allow them to write a single source code for several mobile platforms, but the result of each individual build will be separate executable files. Obviously, cross-platform development is needed to optimize the cost and speed of development, as well as application support. At the same time, the output result is no less qualitative than with native development. Currently, there are the following most popular cross-platform frameworks: Xamarin, React Native, Flutter, and NativeScript. They are very different and not in all situations will be equally useful (or even necessary in principle).

The article discusses the technology of developing cross-platform mobile applications Xamarin: the principles of developing cross-platform mobile applications using it are outlined, the main features provided to developers within this approach, its advantages and disadvantages are presented. A range of issues related to the design, implementation and testing of mobile applications is considered: the use of the MVVM architectural pattern, the XAML markup language to describe the user interface, a possible development environment.

A methodology for creating cross-platform mobile applications on the Xamarin platform is presented, containing a brief, consistent and comprehensive description of all stages of development.

Keywords: mobile application, cross-platform development, operating systems (OS), Apple iOS (iPhone and iPad), Google Android, Xamarin technology, Xamarin.Forms, C# language, XAML markup language, MVVM design pattern (Model/View/ViewModel), Development environment, Visual Studio 2019, Visual Studio для Mac и Xcode.

For citation*:* Kuznetsova S.V. Features of cross-platform mobile applications development using Xamarin. *Trudy MAI*, 2022, no. 125. DOI: 10.34759/trd-2022-125-21

Введение

На сегодняшний день разработка мобильных приложений – один из самых активно развивающихся секторов IT-индустрии. И в связи с этим появляется множество проектов по разработке приложений на мобильных платформах.

Цифровая трансформация в авиационной отрасли охватывает производство авиатехники, а также пассажирские и грузовые перевозки. Что касается перевозок,

то отдельного внимания заслуживает работа авиакомпаний, аэропортов, взаимодействие с пассажирами, заказчиками грузовых перевозок, функционирование единой системы организации воздушного движения. Многие тренды цифровой трансформации в остальных составляющих авиационной отрасли сфокусированы на взаимодействии с клиентом [2, 9]. Кроме пассажиров, цифровизация затрагивает перевозчиков. У пилотов — персональные устройства, на экранах которых они видят электронную документацию, имеют доступ к автоматизации выполнения инженерно-штурманских расчетов [10, 17], выводу изображения с камер наблюдения и др. У менеджеров — мониторы руководителей. У стюардесс — планшеты с информацией о каждом конкретном пассажире по питанию, пересадкам, стыковкам, сопровождающим и многое-многое другое.

Одним из активно развивающихся сегодня технологических трендов в части цифровизации отрасли является разработка мобильных приложений для авиации, включая авиастроение, авиаперевозки и обслуживание аэропортов [18].

Мобильные приложения предоставляют существенно более быстрый доступ к данным и являются эффективным средством взаимодействия независимо от местоположения пользователя. Например, обеспечивают быстрый доступ к технической информации и непосредственный ввод данных, оптимизируя инженерный и производственный процессы.

Классический вариант разработки мобильного приложения — это нативная разработка, при которой под каждую операционную систему (ОС) разрабатывается отдельное приложение, что достаточно дорого и трудоемко. Обзорный анализ технологий нативной разработки мобильного программного обеспечения (ПО)

можно встретить во многих статьях [5, 6, 7, 8, 11]. Альтернативой является кросс-платформенная технология [1, 4, 14, 16], основной целью которой является предоставление разработчикам ПО инструмента для параллельного создания приложений для нескольких операционных систем одновременно. Это позволяет написать единый исходный код для нескольких мобильных платформ, но результатом каждой отдельной сборки станут отдельные исполняемые файлы. Очевидно, что кросс-платформенная разработка нужна для оптимизации стоимости и скорости создания, а также поддержки приложений. При этом результат на выходе не менее качественный, чем при нативной разработке. В настоящее время наиболее популярными являются следующие кросс-платформенные фреймворки: Xamarin, React Native, Flutter, и NativeScript. Они сильно отличаются и не во всех ситуациях будут одинаково полезны (или даже нужны в принципе).

В статье исследуется технология разработки кросс-платформенных мобильных приложений Xamarin: изложены принципы разработки кросс-платформенных мобильных приложений, представлены основные возможности, предоставляемые разработчикам в рамках этого подхода, его преимущества и недостатки.

Разработана методика для создания мобильных приложений с использованием платформы Xamarin, содержащая краткое последовательное и всестороннее описание всех этапов разработки.

Платформа Xamarin

Xamarin – это платформа с открытым исходным кодом, предназначенная для построения современных приложений для iOS, Android и Windows с .NET. Основным новшеством, привнесенным технологией Xamarin в мир мобильной разработки, стало то, что эта технология обеспечивает кросс-платформенность кода.

Платформа Xamarin предлагает сложную кросс-платформенную поддержку для трех основных платформ iOS, Android и Windows. Приложения могут быть написаны для совместного использования до 80% их кода, а библиотека Xamarin..Essentials предлагает универсальный API-интерфейс для доступа к общим ресурсам на всех трех платформах. Это может значительно сократить как затраты на разработку, так и время выхода на рынок для мобильных разработчиков, ориентированных на эти самые популярные платформы.

Функционально платформа Xamarin представляет ряд субплатформ, в частности:

- Xamarin.Android - библиотеки для создания приложений для ОС Android;
- Xamarin.Mac - библиотеки для создания приложений для Mac OS X;
- Xamarin.iOS - библиотеки для создания приложений для iOS.

Эти субплатформы играют важную роль, так как через них приложения могут направлять запросы к прикладным интерфейсам на устройствах под управлением ОС Android или iOS. Благодаря этим субплатформам можно создавать отдельно приложения для Android, отдельно для iOS, но наиболее важной особенностью платформы является возможность создавать кросс-платформенные приложения, использующие одну логику для всех платформ. Можно определить визуальный

интерфейс, один раз к нему привязать какую-то логику на C#, и все это будет работать на Android и iOS.

Платформа Xamarin позволяет разработчикам создавать на языке C# мобильные приложения для операционных систем Apple iOS (iPhone и iPad) и Google Android. Она основана на сторонней открытой реализации платформы .NET под названием Mono. Уровень бизнес-логики может быть написан один раз и разделен между всеми мобильными платформами. Взаимодействие с интерфейсом пользователя (UI) и API различается на разных мобильных платформах, поэтому уровень пользовательского опыта обычно для каждой платформы свой.

Инструментарий Xamarin.Forms

Инструментарий Xamarin.Forms расширяет возможности Xamarin, упрощая кросс-платформенную мобильную разработку благодаря написанию общего кода для обеспечения требуемого пользовательского интерфейса, а также для реализации бизнес-логики. Для разработки бизнес-логики приложений используется язык C#. Бизнес-логика может, например, отвечать за обмен данных с сервером, работу с внутренней базой данных, конвертирование данных и т.д.

Язык разметки XAML

Как и в приложениях для универсальной платформы Windows (UWP), в Xamarin.Forms применяется язык разметки XAML (рис. 1) для однократного определения пользовательского интерфейса для всех платформ сразу с помощью абстрагирования компонентов данного интерфейса, специфичных для той или иной

платформы. XAML включает основные четыре категории элементов: панели, элементы управления, элементы, связанные с документом и графические фигуры. Все созданное или реализованное с использованием XAML может быть выражено при помощи C#, однако, ключевым аспектом технологии является уменьшение сложности используемых для обработки XAML инструментов, так как XAML основан на XML. Приложения, созданные на основе Xamarin.Forms, формируют UI из нативных виджетов платформы, поэтому являются удобными и привлекательными, а также идеально подходят под целевую мобильную платформу.

Xaml-файлы можно также создавать и редактировать при помощи редактора и инструментов визуального конструирования, например, Microsoft Visual Studio. Процесс разработки приложений схож с разработкой на основе Windows.Forms и UWP. Чтобы переключаться между кодом и конструктором, нужно выбратьть на требуемую вкладку справа от рабочего окна (рис. 1).

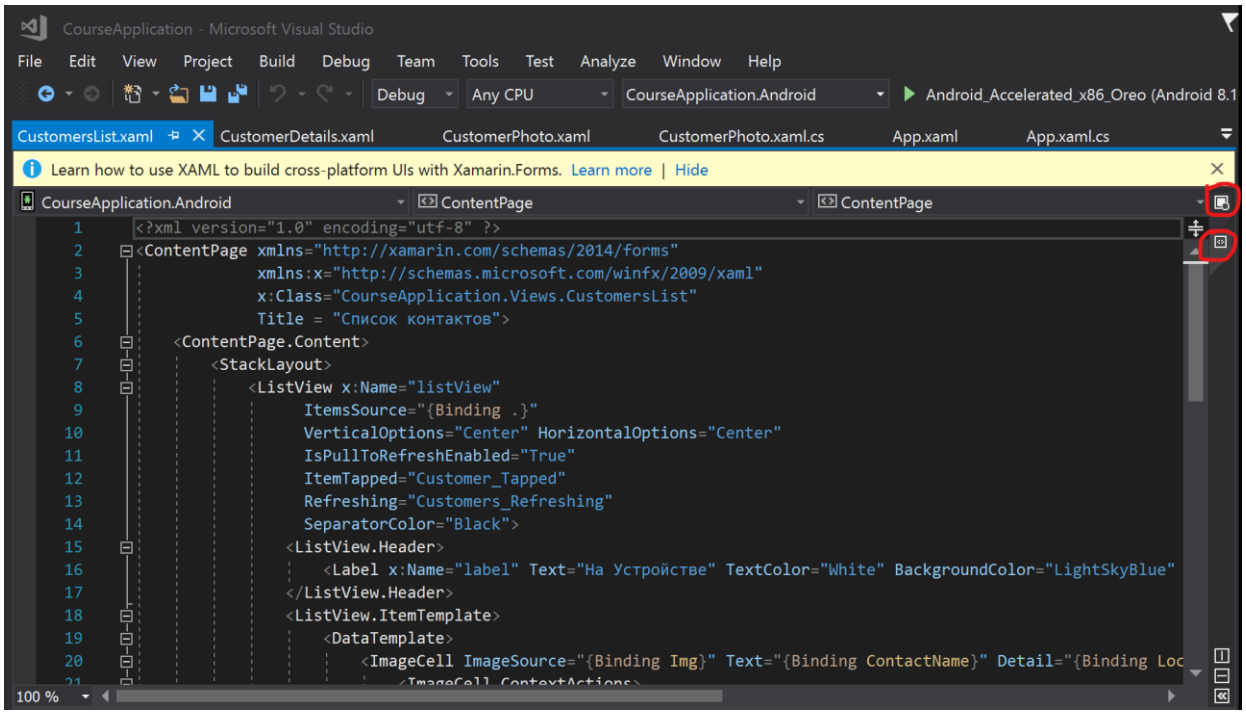

Рис.1. Код на XAML

Xamarin.Forms предоставляет готовые компоненты интерфейса приложения.

Страницы

При создании страницы в основном проекте приложения, можно выбрать ее тип (рис. 2) в зависимости от целей и рациональности использования.

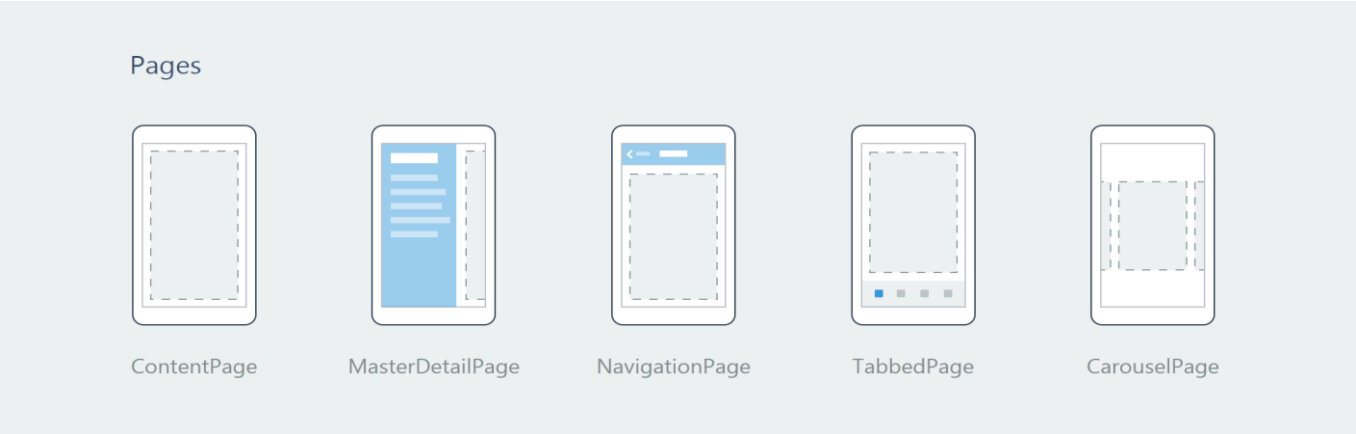

Рис. 2. Типы страниц

1) ContentPage – один элемент заполняет всю страницу.
2) MasterDetailPage – страница с выезжающим меню.
3) NavigationPage – страница с возможность вернуться назад.
4) TabbedPage – страница с вкладками.
5) CarouselPage – прокручиваемая страница.

Макеты

После создания страницы нужно определить, как все элементы будут расположены внутри страницы (рис.3).

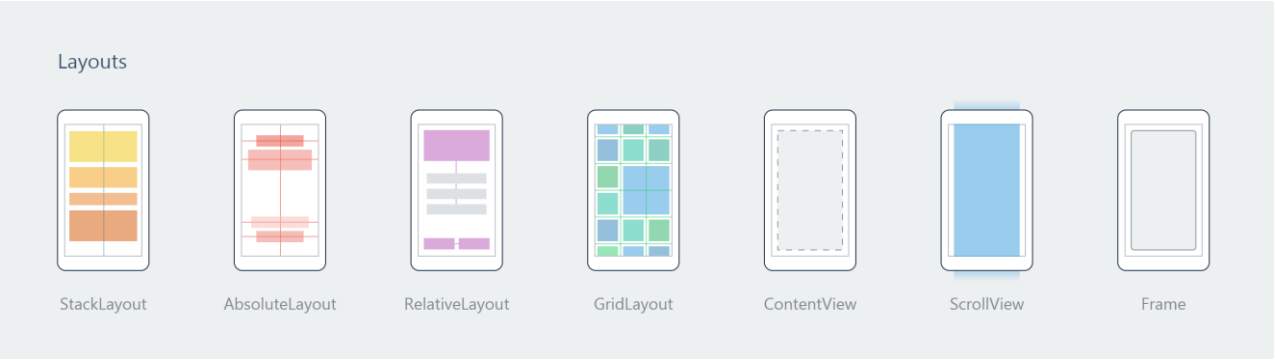

Рис.3. Макеты страниц

1) StackLayout – элементы выстраиваются один под другим сверху вниз.
2) AbsoluteLayout – размещение по абслолютным координатам.
3) RelativeLayout – все элементы связаны друг с другом.
4) GridLayout – сетка элементов.
5) ContentView – один элемент по всей странице.
6) ScrollView – прокручиваемый вид.
7) Frame – по сути аналог ContentView.

Макеты можно комбинировать, например, внутрь ScrollView можно поместить StackLayout для того чтобы последний имел функционал прокрутки.

Элементы управления

Когда страница создана, а макет задан, можно размещать на ней элементы управления (рис.4). Технология очень схожа с Windows.Forms.

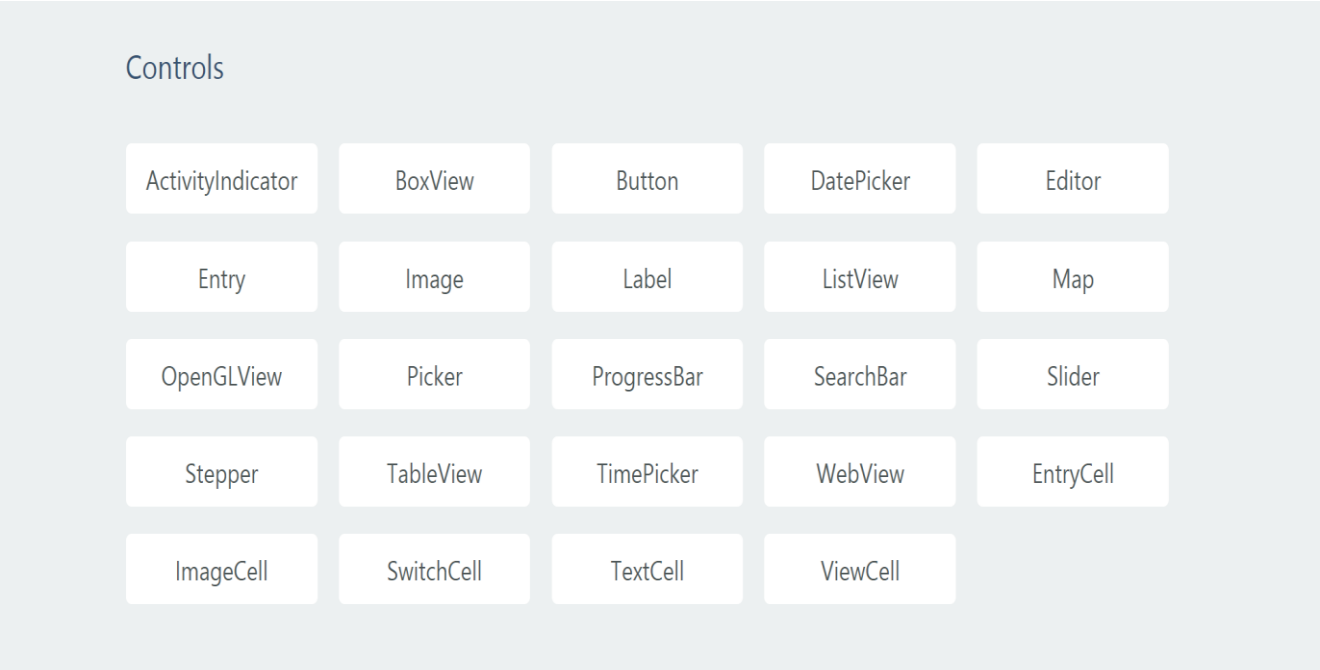

Рис.4. Элементы управления

Некоторые элементы вплоть до названия повторяют аналогичные компоненты Windows.Forms.

Паттерн проектирования MVVM (Model/View/ViewModel)

Ключевым принципом построения кросс-платформенных приложений является создание архитектуры, которая поддается максимизации совместного использования кода на разных платформах. Существует несколько ключевых шаблонов, которые полезны для понимания при создании поддерживаемых/понятных мобильных приложений: Model/View/ViewModel (MVVM), Model/View/Controller (MVC), Business Façade, Singleton, Provider, Strategy Использование паттерна проектирования MVVM (Model/View/ViewModel) обеспечивает разделение функциональной части приложения на три ключевых компонента (рис.5):

- View - представление или пользовательский интерфейс;
- Model - модель или данные, которые используются в приложении;
- ViewModel - промежуточный слой между представлением и данными, который обеспечивает их взаимодействие, уменьшая связанность между компонентами и разделяя ответственности между ними.

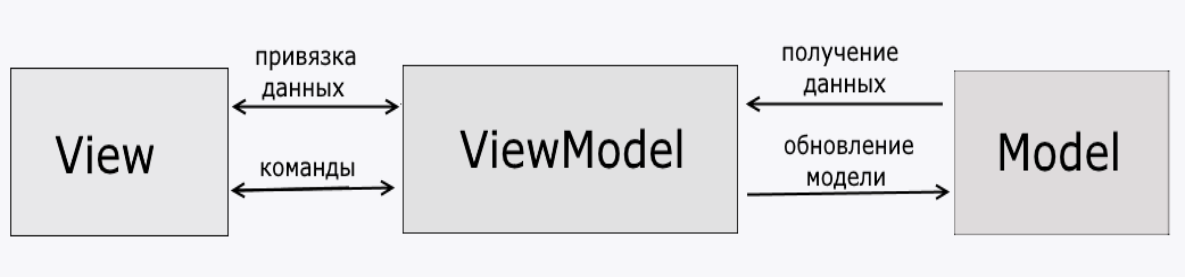

Рис. 5. Паттерн MVVM

Преимущества технологии Xamarin.Forms

Технология Xamarin.Forms, имеет ряд преимуществ, в том числе:

- в процессе разработки создается единый код для всех платформ;
- Xamarin предоставляет прямой доступ к нативным API каждой платформы;
- при создании приложений можно использовать мощные возможности платформы .NET Core и языка программирования C# (а также F#), который включает в себя значительные улучшения по сравнению с языками Objective-C/Swift и Java, такие как динамические языковые функции, функциональные конструкции, т Lambdas, LINQ, функции параллельного программирования, сложные Generics и многое другое.

Недостатки технологии Xamarin.Forms

Приложения, созданные на основе Xamarin.Forms могут работать медленнее [12, 13, 19, 20], чем нативные приложения.

Методика разработки мобильных приложений

Для разработки кросс-платформенных мобильных приложений необходимы понимание принципов объектно-ориентированного проектирования, знание шаблонов проектирования и алгоритмов, опыт проектирования программного обеспечения и программирования на C#, а также навыки работы с инструментальными средствами разработки Microsoft Visual Studio [14].

Для frontend-разработки (разработка пользовательского интерфейса) используется язык XAML. В xaml-файле описывается расположение элементов пользовательского интерфейса, а также зависимости проектов. В xaml-файле описано только расположение элементов управления и их «класс». В процессе компиляции кода на место элемента управления, описанного в xaml-файле, подставляется нативный элемент управления, в зависимости от используемой платформы. Кроме того, у разработчика есть возможность создавать на C# свои пользовательские элементы управления. Пользовательский элемент управления позволяет настроить внешний вид и свойства элемента для каждой платформы.

Для backend-разработки (бизнес-логики приложений) используется язык C#. В написании бизнес–логики и кроется главный плюс данной технологии, так как теперь ее нужно написать всего один раз, в отличии от нативной разработки. Код, написанный на C#, компилируется в нативный код для каждой платформы.

Проекты, использующие Xamarin, как правило, состоят из нескольких основных модулей:

1. Common code (общий код). В этом модуле находится логика, которую использует приложение, вне зависимости от того, на какой платформе оно запущено. Как правило, в этом модуле находятся xaml и xaml.cs - файлы разметки и, соответственно, обработчиков к ней; логика, отвечающая за обмен данных с сервером, работу с внутренней базой данных, конвертирование данных и тому подобное. Так же здесь находится общий файл program.cs, который отвечает за запуск приложения.
2. Custom controls (пользовательские элементы управления). В этом модуле находится объявления (общий набор свойств/методов) пользовательских элементов управления. Подробное и конкретное их описание находится в модулях конкретных платформ.
3. Платформенные модули. Их всего три – android, ios, windows. Каждый из этих модулей содержит подмодуль custom controlls, включающий в себя описание элементов управления, объявленных в модуле custom controls. С помощью этих классов описывается конкретный внешний вид, поведение, события и делегаты элементов управления для одной из трех платформ. Так же здесь находится подмодуль Resources, в котором содержатся ресурсы, необходимые приложению.

При компиляции приложения, Xamarin подставляет в общую xaml-разметку элементы управления, опираясь на зависимости, прописанные в файле разметки, прикрепляет к ним обработчики, компилирует код на C# в нативный и генерирует

выходные сборки для ios - формата .ipa, для android - .apk. Полученные файлы запускаются на целевых платформах так же, как и нативные.

Слой взаимодействия с пользователем, созданный с помощью Xamarin.Forms, никогда не будет идеально сочетаться с каждой конкретной платформой, будучи клиентской специализированной сборкой с применением Xamarin.Forms. Поэтому для каждой операционной системы должен быть написан отдельный код для работы с системными функциями, например, такими как звонки, открытие приложений и т.д.

Таким образом, Xamarin обеспечивает создание 4 проектов: три для выполнения под управлением определенных ОС (Android, iOS, UWP) и один проект содержащий логику работы приложения и пользовательского интерфейса.

Среда разработки мобильных приложений с использованием Xamarin.Forms

Xamarin имеет собственную интегрированную среду разработки Xamarin Studio. Также можно создавать приложения для Android, iOS и Windows - от бизнес-логики до пользовательского интерфейса - с почти 80% -ным общим кодом с использованием Xamarin.Forms в Visual Studio Community, Professional и Enterprise.
Для создания приложений Xamarin.Forms разработчики могут пользоваться Visual Studio 2019 или Visual Studio для Mac. Для компиляции приложений для Android потребуется операционная система Windows 10 и среда разработки Visual Studio 2019. Для компиляции приложений для iOS понадобится операционная система macOS, Visual Studio для Mac и Xcode.

Для разработки на самой последней версии Xamarin.Forms нужен компьютер с Windows 10 или Mac OS Mojave (или более поздняя).

Ниже приведена последовательность действий при разработки мобильного приложения.

1. Установить Visual Studio 2019 с официального сайта Microsoft visualstudio.microsoft.com. Далее приводится описание для версии “Community”
2. Когда будет предложено установить компоненты, обязательно выбрать “Mobile Development with .NET” во вкладке “Mobile & Gaming”. Но для полного функционала рекомендуется установить соответствующие пакеты для разработки под Windows.
3. При первом запуске среды разработки следует установить Android SDK (Software Development Kit) соответствующего выпуска. Через меню приложения перейти в Android SDK Manager и установить нужный пакет(ы), в зависимости от версии Android, на которую нацелена разработка.

   Если эмулятор Android не был установлен автоматически, его можно установить вручную во вкладке Tools.
4. В меню перейти по ветви File -> New -> Project

   В левом дереве выбрать CrossPlatform и создать, задав имя и расположение, проект Mobile App (Xamarin.Forms) (рис.6).

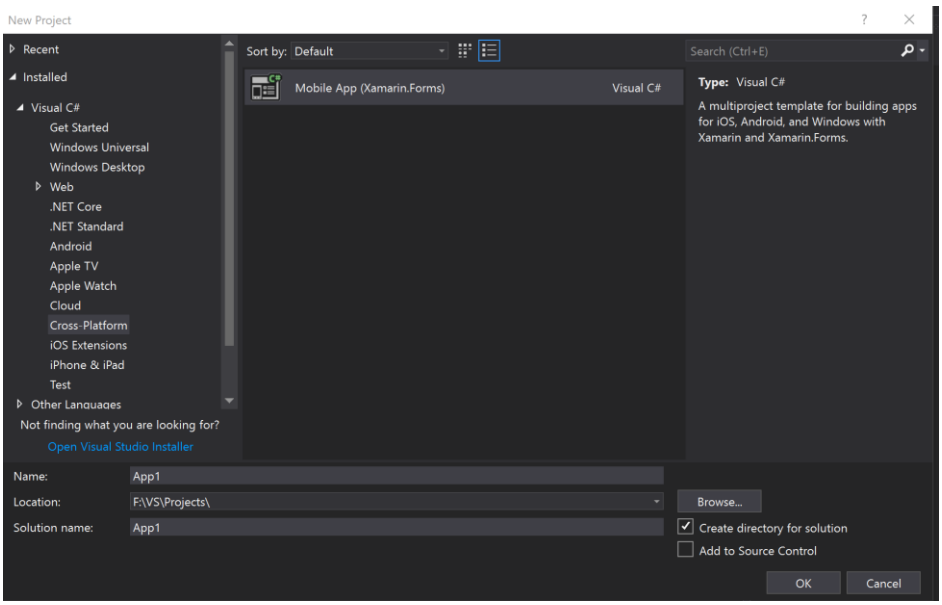

Рис. 6, Создание проекта Mobile App

В следующем окне выбрать шаблон приложения и платформы, под которые будет вестись разработка и стратегию обмена кодом. Последний подготовительный шаг - обновление Nu-Get пакетов. Щелкнуть правой кнопкой мыши на решении App1 и выберать пункт Update NuGet Packages.

5. На этом шаге создаются папки Models и View и пишется соответствующий код приложения. В автоматически сгенерированном файле App1.xaml.cs обрабатываются события, возникающие при старте, сне, выходе из сна и т.д.. В этом же файле указывается страница, которая будет отображаться как основная. По умолчанию – MainPage.xaml.

   Разработка на Xamarin.Forms ведется как раз созданием таких страниц, связей между ними и написанием бизнес-логики.

6. Отладку и тестирование можно производить как с помощью эмулятора, так и с помощью подключенного устройства с включенной USB отладкой.

7. Сборка и архивация приложения для Android

Для того, чтобы приложение можно было установить на устройство или опубликовать в магазине приложений, его нужно заархивировать и подписать. Когда приложение уже полностью готово к выпуску, нужно выбрать в верхней строке конфигурацию текущего решения “Release”, вместо “Debug”.

Использование облачных сервисов

Мобильные приложения часто взаимодействуют с облачными сервисами. Создать сервис веб-API ASP.NET Core для поддержки мобильного приложения можно благодаря среде разработки Visual Studio.

Платформа ASP.NET Core представляет технологию от компании Microsoft, предназначенную для создания различного рода веб-приложений: от небольших веб-сайтов до крупных веб-порталов и веб-сервисов [14].

Заключение

В статье предложена методика для создания мобильных приложений с применением технологии Xamarin, содержащая краткое последовательное и всестороннее описание этапов проектирования, реализации, отладки и тестирования мобильного приложения.

Изложены принципы разработки кросс-платформенных мобильных приложений с использованием Xamarin.Forms.

Рассмотрены архитектурный паттерн MVVM, язык разметки XAML для описания пользовательского интерфейса, и возможная среда разработки.

Представлены основные возможности, предоставляемые разработчикам в рамках этого подхода, выявлены его преимущества и недостатки.

Демонстрируется подробный процесс разработки мобильного приложения.

Сделан вывод, что инструментарий Xamarin.Forms целесообразно использовать, когда разработка ведется на языке C# (или существующий код C# портируется на разные платформы), ограничен бюджет, важна скорость разработки и/или интерфейс приложения не очень сложен. Эта технология хорошо подходит для разработки корпоративных мобильных приложений в различных отраслях.

Список источников

1. Валиуллина Д.И., Зиганшин И.А. Применение фреймворка IONIC для разработки кросс-платформенных мобильных приложений // Международный научно-исследовательский конкурс «Ученый XXI века» (Пенза, 05 января 2022): сборник статей. – Пенза: Наука и Просвещение, 2022. С. 32-34

2. Долгова М.И., Сливинский Д.В. Продвижение инновационных информационных технологий на воздушном транспорте // Экономика и бизнес: теория и практика. 2021. № 3-1. С. 172-177.

3. Золотой С.А., Страшко И.Б., Котов Д.С., Нестерович И.М., Роубо В.В., Костюк К.И. Совершенствование инструментальных средств обработки и анализа космической информации // Информатика. 2021. Т. 18. № 3. С. 106-114.

4. Орлова М.С., Слива М.В. Анализ кроссплатформенных технологий для разработки мобильных приложений // II Международная научно-практическая конференция «Современное программирование» (Нижневартовск,14–16 ноября

2019): сборник трудов. – Нижневартовск: Нижневартовский государственный университет, 2019. С. 93-96.

5. Седлецкий К.В. Анализ технологий разработки мобильных приложений // Молодежный Вестник УГАТУ. 2020. № 1 (22). С. 136-138.

6. Сирбаев И.Ш., Хисаметдинов Ф.З. Технология разработки мобильных приложений // Всероссийская научно-практической конференции с международным участием «Неделя науки и технологий» (Сибай, 12-16 апреля 2021): сборник трудов. – Сибай: Издательский дом «Республика Башкортостан», 2021. С. 274-276.

7. Тлембаев А.А., Даненова Г.Т., Коккоз М.М. Исследование программных средств разработчиков мобильных приложений // Международный журнал прикладных и фундаментальных исследований. 2018. № 2. С. 53-58.

8. Трапезникова П.В., Беззатеева В.С. Этапы разработки мобильного приложения // Международная научно-практическая конференция «Информационные технологии как основа эффективного инновационного развития» (Самара, 10 января 2022): сборник статей. – Уфа: Аэтерна, 2022. Т. 1. С. 66-70.

9. Федосеева М.С., Тюменев А.В. Информационные технологии в гражданской авиации // Теория и практика проектного образования. 2020. № 2 (14). С. 43-45.

10. Шевелев А.А. Внедрение применения электронного полетного планшета в боевой авиации на этапах подготовки и выполнения полетного задания // Universum: технические науки. 2021. № 11-1 (92). С. 20-24. DOI: 10.32743/UniTech.2021.92.11.12536

11. Ямских М.Е. Сравнительный анализ программ для создания мобильных приложений // Актуальные проблемы авиации и космонавтики. 2019. Т. 3. С. 685-687.

12. Biorn-Hansen A. An empirical investigation of performance overhead in cross-platform mobile development frameworks // Empirical Software Engineering, 2020, vol. 25, no. 4, pp. 2997-3040. DOI:10.1007/s10664-020-09827-6

13. Biorn-Hansen A., Gronli T. M., Ghinea G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development // ACM Computing Surveys (CSUR), 2018, vol. 51, no. 5, pp. 1-34. DOI:10.1145/3241739

14. Прайс Марк Дж. «C# 7 и .NET Core. Кросс-платформенная разработка для профессионалов». - СПб.: Питер, 2018. - 640 с.

15. Jia Y. Multi-dimensional teaching mode for introduction to aerospace // Journal of Physics: Conference Series. IOP Publishing, 2020, vol. 1509, no. 1, pp. 1-6. DOI: 10.1088/1742-6596/1509/1/012011

16. Christoph J., Rösch D., Schuster T., Waidelich L. Current Progress in Cross-Platform Application Development Evaluation of Frameworks for Mobile Application Development // International Journal on Advances in Software, 2019, vol. 12, no.1 -2, pp. 30-45.

17. Rieger C., Kuchen H. A model-driven cross-platform app development process for heterogeneous device classes // Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019, pp. 7431-7440.

18. Smy P. Training air traffic controllers through digital mobile applications versus traditional methods // 14th European Conference on Games Based Learning, 2020, pp. 547-557.

19. Syeed A.B., Bhat S.H., Kaur D. Study of mobile app development industry // International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2021, vol. 7, no. 6, pp. 154-170. DOI: 10.32628/cseit217638

20. Yi J., He J., Yang L. Platform heterogeneity, platform governance and complementors' product performance: an empirical study of the mobile application industry // Frontiers of Business Research in China, 29, vol. 13, no. 1, pp. 1-20.

21. Литвиненко А.О. Программный комплекс автоматизированного планирования задействования средств наземного автоматизированного комплекса управления // Труды МАИ. 2016. № 86. URL: https://trudymai.ru/published.php?ID=67829

22. Качалин А.М., Задорожная О.Н. Система спутникового контроля авиационных систем (Единая Информационная Система Взаимодействия (SWIM.ru)) // Труды МАИ. 2016. № 85. URL: https://trudymai.ru/published.php?ID=66220

23. Ронжин А.Л., Нгуен В.В., Соленая О.Я. Анализ проблем разработки беспилотных летательных манипуляторов и физического взаимодействия БЛА с наземными объектами // Труды МАИ. 2018. № 98. URL: https://trudymai.ru/published.php?ID=90439

References

1. Valiullina D.I., Ziganshin I.A. *Mezhdunarodnyi nauchno-issledovatel'skii konkurs «Uchenyi XXI veka»*: sbornik statei, Penza, Nauka i Prosveshchenie, 2022, pp. 32-34.

2. Dolgova M.I., Slivinskii D.V. *Ekonomika i biznes: teoriya i praktika*, 2021, no. 3-1, pp. 172-177.

3. Zolotoi S.A., Strashko I.B., Kotov D.S., Nesterovich I.M., Roubo V.V., Kostyuk K.I. *Informatika*, 2021, vol. 18, no. 3, pp. 106-114.

4. Orlova M.S., Sliva M.V. *II Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Sovremennoe programmirovanie»* sbornik trudov, Nizhnevartovsk, Nizhnevartovskii gosudarstvennyi universitet, 2019, pp. 93-96.

5. Sedletskii K.V. *Molodezhnyi Vestnik UGATU,* 2020, no. 1 (22), pp. 136-138.

6. Sirbaev I.Sh., Khisametdinov F.Z. *Vserossiiskaya nauchno-prakticheskoi konferentsii s mezhdunarodnym uchastiem «Nedelya nauki i tekhnologii»*: sbornik trudov, Sibai, Izdatel'skii dom «Respublika Bashkortostan», 2021, pp. 274-276.

7. Tlembaev A.A., Danenova G.T., Kokkoz M.M. *Mezhdunarodnyi zhurnal prikladnykh i fundamental'nykh issledovanii,* 2018, no. 2, pp. 53-58.

8. Trapeznikova P.V., Bezzateeva V.S. *Mezhdunarodnaya nauchno-prakticheskaya konferentsiya «Informatsionnye tekhnologii kak osnova effektivnogo innovatsionnogo razvitiya»*: sbornik statei. Ufa, Aeterna, 2022, vol. 1, pp. 66-70.

9. Fedoseeva M.S., Tyumenev A.V. *Teoriya i praktika proektnogo obrazovaniya,* 2020, no. 2 (14), pp. 43-45.

10. Shevelev A.A. *Universum: tekhnicheskie nauki*, 2021, no. 11-1 (92), pp. 20-24. DOI: 10.32743/UniTech.2021.92.11.12536

11. Yamskikh M.E. *Aktual'nye problemy aviatsii i kosmonavtiki*, 2019, vol. 3, pp. 685-687.

12. Biorn-Hansen A. An empirical investigation of performance overhead in cross-platform mobile development frameworks, *Empirical Software Engineering*, 2020, vol. 25, no. 4, pp. 2997-3040. DOI:10.1007/s10664-020-09827-6

13. Biorn-Hansen A., Gronli T. M., Ghinea G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development, *ACM Computing Surveys (CSUR)*, 2018, vol. 51, no. 5, pp. 1-34. DOI:10.1145/3241739

14. Prais Mark Dzh. *C# 7 i .NET Core. Kross-platformennaya razrabotka dlya professionalov* (C# 7 and .NET Core. Cross-platform development for professionals), Saint Petersburg, Piter, 2018, 640 p.

15. Jia Y. Multi-dimensional teaching mode for introduction to aerospace, *Journal of Physics: Conference Series*. IOP Publishing, 2020, vol. 1509, no. 1, pp. 1-6. DOI: 10.1088/1742-6596/1509/1/012011

16. Christoph J., Rösch D., Schuster T., Waidelich L. Current Progress in Cross-Platform Application Development Evaluation of Frameworks for Mobile Application Development, *International Journal on Advances in Software*, 2019, vol. 12, no.1 -2, pp. 30-45.

17. Rieger C., Kuchen H. A model-driven cross-platform app development process for heterogeneous device classes, *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019, pp. 7431-7440.

18. Smy P. Training air traffic controllers through digital mobile applications versus traditional methods, *14th European Conference on Games Based Learning*, 2020, pp. 547-557.

19. Syeed A.B., Bhat S.H., Kaur D. Study of mobile app development industry, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2021, vol. 7, no. 6, pp. 154-170. DOI: 10.32628/cseit217638

20. Yi J., He J., Yang L. Platform heterogeneity, platform governance and complementors' product performance: an empirical study of the mobile application industry, *Frontiers of Business Research in China*, 29, vol. 13, no. 1, pp. 1-20.

21. Litvinenko A.O. *Trudy MAI,* 2016, no. 86. URL: https://trudymai.ru/eng/published.php?ID=67829

22. Kachalin A.M., Zadorozhnaya O.N. *Trudy MAI*, 2016, no. 85. URL: https://trudymai.ru/eng/published.php?ID=66220

23. Ronzhin A.L., Nguen V.V., Solenaya O.Ya. *Trudy MAI*, 2018, no. 98. URL: https://trudymai.ru/eng/published.php?ID=90439

Статья поступила в редакцию 02.06.2022
Статья после доработки 04.06.2022
Одобрена после рецензирования 10.07.2022
Принята к публикации 25.08.2022
The article was submitted on 02.06.2022; approved after reviewing on 10.07.2022; accepted for publication on 25.08.2022