Development of an Information Training System Based on Cross-Platform Technologies

Saule Akimova

Abstract**—**The mobile app market has shown unique growth over the past 3-5 years, creating a constantly growing demand among end users not only in the field of public administration, trade, financial structures, but also in the field of educational services. There is a demand for information training systems that run on various mobile platforms and operating systems. Cross-platform technologies allow you to create a single universal mobile application project for various mobile platforms and operating systems. In this aspect, the development of a cross-platform information training system is relevant.

Keywords**—**Cross-platform, Mobile Application, Xamarin.Forms, Frameworks.

I. Introduction

The aim of the study was to develop a cross-platform application in support of the educational process.

The scientific novelty of the research work is that it will analyze existing approaches to solving the problem of creating cross-platform applications, and offer an original application architecture that provides for the creation of mobile and desktop cross-platform applications.

The object of the research is a cross-platform development technology based on Xamarin, which allows developing cross-platform applications in the MS Visual Studio environment in C#. Xamarin is an open source platform designed to build modern, high-performance applications for iOS, Android, and Windows with .NET.

Expected result: will be developed a conceptual model of the information training system, based on which will be created a prototype of a cross-platform application, and the application will be tested on various mobile platforms.

In the modern era of the proliferation of computer networks and electronic devices (mobile phones, tablets, PCs), every person and organization is trying to access information and use these devices to promote themselves and increase productivity. Mobile phones have become so common that we have started using them in all areas of life, whether it's entertainment or education. The mobile device market is growing rapidly, which is why the demand for mobile applications is growing, but developing applications for mobile devices is not so easy and involves a number of problems. Mobile phones have a different operating system, and unlike a PC OS, the mobile OS decides the type of programming language needed for applications running on it. Therefore, we need to develop a mobile app in accordance with its platform. There are various categories of mobile phone app development, which are generally divided into four categories: native apps, mobile web apps, hybrid apps, and cross-platform apps.
Running an app on multiple platforms requires a cross-platform framework. In our research, we explore cross-platform strategies that would be useful for solving barrier problems in an application designed for mobile learning systems in a heterogeneous device environment.

S. Akimova, Higher School of Information Technologies of the Institute of Economics, Information Technologies and Professional Education of Zhangirkhan West Kazakhstan Agrarian Technical University, Uralsk, Republic of Kazakhstan, (e-mail: saule_akim@mail.ru).

II. Overview of Native and Cross-Platform Development Methods

Native development refers to the use of original languages and development tools for the mobile operating system. For iOS, apps are created in the XCode development environment in Objective-C, Swift, C, and C++. To create applications for Android, use the Android Studio environment and the Java language. Each development environment contains a whole set of utilities for writing code, designing the interface, debugging, profiling (monitoring), and building applications. Both the environment and the corresponding set of utilities are designed specifically for each mobile operating system and are the most convenient and powerful tools for developing mobile applications.

Cross-platform development involves the use of special utilities (frameworks) to create an application based on The JavaScript family of languages. The entire structure and logic of the application is created using such tools (PhoneGap, Titanium, Xamarin, Cordova, etc.) in JavaScript, and then wrapped in a native launching element, i.e. integrated into the base project for XCode or Android Studio, which allows you to create project builds with the same logic for several operating systems at once.

Advantages of cross-platform development

The cross-platform approach to development has the following positive aspects:

1. It requires less resources to implement an application for several platforms at once. This, in fact, is the essence of the cross – platform approach-the same code works on both iOS and Android.
2. Less development time. Due to the lack of unique interface elements and simpler technologies, the time to create simple products is usually less.
3. Simplified product update cycle. If you need to add something to the project or fix an error, this is done immediately for all platforms that the project applies to.
4. The ability to use the mobile version of the site. Most cross-platform solutions use the JavaScript family of languages.
5. Using a single application logic. The built-in logic of the application is guaranteed to work the same for all platforms.

Advantages of native development

Development in native technologies and languages for iOS and Android has the following positive aspects:

1. The speed of the application. Since the application is created using the original development tools (XCode, Android Studio), the code obtained as a result of compiling the project is optimal for this platform. The application gets full hardware support for the device (processing the same images is carried out by a separate processor specially designed for this purpose-the GPU), multithreading is used to implement complex tasks and load content in the background. during development, programmers can measure the speed of all code sections and optimize them if necessary. they also have tools for monitoring RAM usage, searching for possible leaks, etc.
2. Flexibility in implementation. In contrast to the limitations in building the interface and the complexity of visual effects imposed by frameworks for cross-platform project Assembly, native development can implement everything that the technologies of a particular mobile operating system are capable of.
3. Use of the latest technologies and dependence on cross-platform frameworks. New software and hardware functionality provided by the device and operating system manufacturers becomes available for implementation immediately after the corresponding updates are released.

4. The ease and quality of testing. In addition to the tools mentioned in paragraph 1 for monitoring the application's use of device hardware resources, developers and testers have a whole range of technologies at their disposal. First, all system parameters are controlled automatically during the app's operation. If the application is using more memory than expected, or more CPU resources, this will not go unnoticed. Second, there are opportunities in the wide application of unit tests-automatic testing of almost every method in the application. If some part of the application stopped working correctly due to any code changes, the new version will simply not be built, and the programmer will immediately see the reason. Third, there are wide opportunities for integrating remote error monitoring systems: each native project has built-in functionality that allows you to see the error and its cause on any user's device.
5. Full support from the App Store and Google Play. Both companies, Apple and Google, are interested in ensuring that users get the most positive experience when using applications on their respective platforms, which is currently possible.

Summary.

From a technical point of view and from the point of view of the quality of the interface created, native development has much more advantages. However, there are areas where cross-platform technologies are justified: the gaming sector, test projects, and educational resources.

The advantages of cross-platform development, such as saving the project budget, development time, and reaching a larger audience, determined the choice in favor of this application development technology.

III. Xamarin for Cross-Platform Development

Xamarin is an open source platform for building modern, productive iOS, Android, and Windows applications with .NET. The Xamarin Platform is an abstraction layer that controls the interaction between shared code and base platform code.

Thanks to Xamarin, on average 90% of the application code can be used unchanged across platforms. With this pattern, a developer can write all business logic in one language (or use existing application code), while still gaining performance, styling, and behavior specific to each respective platform.

Xamarin is based on the open-source implementation of the .NET platform, Mono. This implementation includes the native C# compiler, runtime, and core .NET libraries.

Xamarin combines all the features of existing platforms and a number of its own, including the following:

1. Full compatibility with the basic SDK packages (software development kit). Xamarin contains compatibility features for almost all the basic SDK packages in iOS and Android. In addition, these bindings are strongly typed, which means that they are easy to navigate and use, and allow for high-quality type checking during compilation and development. Strongly typed bindings, which allow you to reduce the number of run-time errors and to improve the quality of applications.
2. Interaction of Objective-C, Java, C and C++ . Xamarin allows you to directly call the Objective-C, Java, C, and C++ libraries for more efficient use of a variety of third-party code. This feature allows you to use existing iOS and Android libraries written in Objective-C, Java, or C/C++. In addition, Xamarin offers binding projects for binding native Objective-C and Java libraries using declarative syntax.
3. Modern language constructions. Xamarin applications are written in modern C#, which is characterized by significant improvements over Objective-C and Java. This includes dynamic language functions, functional constructs such as lambda expressions, LINQ,

parallel programming functions, generic templates, and so on.

4. Reliable base class library (BCL). Xamarin applications use the .NET BCL library, a large collection of classes with comprehensive and simplified features, including support for XML, databases, serialization, I / o operations, strings, network functions, and so on.Existing C# code can be compiled for use in applications, providing access to thousands of libraries that contain additional features beyond the scope of BCL.
5. Modern integrated development environment (IDE). Xamarin uses the state-of-the-art Visual Studio environment, which includes features such as code completion, a more advanced project and solution management system, a comprehensive library of project templates, an integrated version control system, and much more.
6. Support for cross-platform mobile apps. Xamarin offers advanced cross-platform support for three major platforms-iOS, Android, and Windows. The total amount of code in the created applications can reach 90 %, and the Xamarin library. Essentials offers a universal API for accessing shared resources on all three platforms. This allows you to significantly reduce the development costs and time to release products to the market for developers who create mobile applications.
7. Xamarin.Forms

Xamarin.Forms is an open source user interface platform. Using Xamarin.Forms developers can create applications for Xamarin. iOS, Xamarin.Android and Windows based on a common code base. Xamarin.Forms allows developers to create user interfaces in XAML using C#code. These user interfaces on each platform are prepared for viewing as their own controls. The following are some examples of features provided by Xamarin.Forms:

- XAML user interface language
- Data binding
- Gestures
- Effect produced
- Setting the style

IV. Developed a Technical Specification for Creating an Information System

1. General information

1.1 Product Name (design)

Information training system based on cross-platform technologies

1.2 Purpose and scope

The information system is designed to work on various hardware platforms in support of the educational process. It can be used in the classroom as a training application, as well as when organizing and conducting distance learning.

2. Requirements for the program

2.1. General requirements

2.1.1 Platform for client parts of the application:

- iOS version 9 and higher
- Android version 8.0 and higher
- Windows 10

2.1.2 The Mobile device must support portrait and landscape orientation.

2.1.3 All data is stored in the database.

2.2. Requirements to the functional characteristics

The program should provide the ability to perform the following functions:

2.2.1. User authorization

Two-factor authorization must be implemented in the system.

2.2.2. Providing content for students

2.2.2.1. The training section of the application user

1) The software must contain an introductory section that starts when you first install the app. You must also ensure that you can call from the app.

The section should contain the following information:

- Introduction. The introductory part must contain content in the format of text or video information.
- Instructions for using the app. The instructions section must contain content in text or video format.

2) Requirements for the educational material

1. must be strictly structured

2. it should contain the following types of material: theoretical, practical tasks, control block, assessment, self-study material, links to sources.

3. the material may contain video material, links to video sources.

2.2.2.3. The organization of training

1. The choice of discipline of study;

2. Course;

3. Evaluation of the student;

4. The control of education;

5. Completion of training.

The content of the content should be moderated depending on the course being studied.

3. User group

The project provides the following user groups:

Administrator-the administrator can add/remove / change existing users. Assign or change user roles. The administrator must be able to create new user roles. Performs user authorization and a number of other information system administration functions.

Teacher-creates a course, fills in content, evaluates classes, and leaves comments,

A student can choose a course, go back to the completed material, answer a test, or take an exam.

V. Conclusion

Cross-platform solutions are effective; however, they did not gain enough mileage since their birth due to fragmentation in mobility platforms. Now that the mobile app development world is largely divided into two large platforms – Android & iOS – cross-platform mobile app development is expected to experience much more advancement very soon.

The purpose of using a Cross-Platform approach is to target as many students as possible by providing the application for different platforms and reaching the broadest possible audience. One advantage of mobile learning is that many learners own a deviceand carry it wherever they go. However, designing learning activities for mobile devices requires reliance on instructional design that fits this kind of technology.

Terms of reference for the development of an information system and a conceptual model of the information system has been developed. The prototype of the application is at the stage of creation.

ACKNOWLEDGMENT

This publication is the result of the project implementation ERASMUS+ ACeSYRI:
Advanced Centre for PhD Students and Young Researchers in Informatics
reg.no. 610166-EPP-1-2019-1-SK-EPPKA2-CBHE-JP

Co-funded by the
Erasmus+ Programme
of the European Union

ACeSYRI

REFERENCES

[1] ST RK 2193-2012 " Information technologies. Recommended practice for developing mobile web applications". [Electronic resource] URL: https://online.zakon.kz

[2] Inyushkina O. G. Design of information systems: textbook / O. G. Inyushkina-Yekaterinburg-Fort-Dialog Iset, 2014-240 p.

[3] Isaev, G. N. Design of information systems: textbook / G. N. Isaev - Omega-L, 2015-424 p.

[4] Programming Guide for Xamarin Forms // METANIT.COM -Website about programming URL: https://metanit.com/sharp/xamarin/

[5] Andrew Troelsen, Philip Jetix C# 6.0 programming Language and platform .NET 4.6. - Moscow: Williams, 2016.

[6] 'Xamarin Documentation - Xamarin | Microsoft Docs'. [Online]. Available: https://docs.microsoft.com/en-us/xamarin/. [Accessed: 20-Oct-2020].

[7] D. Hermes, Xamarin Mobile Application: Cross-Platform C# and Xamarin. Forms Fundamentals, 2015.