Каппушев М.Р., Кочкарова П.А.**Проектирование цифровой электронной библиотеки***Северо-Кавказская государственная академия
(Россия, Черкесск)**doi: 10.18411/lj-06-2021-60***Аннотация**

Тема статьи актуальна, т.к. создание электронной библиотеки будет способствовать сохранению языка, национальной самобытности, и культуры граждан, проживающих на территории Карачаево-Черкесской республики. Созданное мобильное приложение для платформы IOS упростит доступ пользователей к национальной художественной и учебной литературе на карачаево-балкарском языке.

Ключевые слова: цифровая электронная библиотека, мобильное приложение, облачная база данных, интерфейс, IOS проект.

Abstract

The topic of the article is relevant, since the creation of an electronic library will contribute to the preservation of the language, national identity, and culture of citizens living in the territory of the Karachay-Cherkess Republic. The created mobile application for the iOS platform will simplify users ' access to the national fiction and educational literature in the Karachay-Balkar language.

Keywords: digital electronic library, mobile application, cloud database, interface, iOS project.

На территории Карачаево-Черкесской Республики проживает пять коренных национальностей, из них четыре являются малочисленными. На сегодняшний день обозначилась тенденция утраты родного национального языка в семьях народов Карачаево-Черкесской республики, поскольку вся жизнь проходит в другой языковой среде. Воспитание в детском саду, обучение в школе, общение с одноклассниками происходит на русском языке. В семье родители разговаривают между собой на родном языке, но этого недостаточно.

В условиях снижения уровня владения родным языком среди представителей разных народов, очень важной становится работа по сохранению национальных языков как неотъемлемой части многонациональной культуры.

Целью работы является проектирование модели электронной национальной библиотеки, создание которой будет способствовать сохранению языка, национальной самобытности, и культуры граждан, проживающих на территории Карачаево-Черкесской республики. Благодаря этому, национальная литература будет легкодоступной для изучения.

Также Указ Президента РФ от 26 октября 2018 г. № 611 «О создании Фонда сохранения и изучения родных языков народов Российской Федерации» является еще одним доказательством того, что данная инициатива актуальна и имеет место быть. К тому же закон от 28 июля 2018 года подписанный президентом, Российской Федерации, «О родных языках» в соответствии с федеральным государственным образовательным стандартом дошкольного, начального и основного общего образования обеспечивают возможность получения образования на родных языках из числа языков народов РФ, изучения государственных языков республик РФ, в том числе русского языка как родного языка.

Закон к тому же определяет, что свободный выбор языка образования происходит путем заявления родителей обучающегося при поступлении (переводе) на

образование по обучающим планам дошкольного образования, обладающими госаккредитацию обучающим планам начального и основного общего образования.

Разработка приложения «Библиотека национальной художественной и учебной литературы Карачаево-Балкарского языка»

Сами, являясь представителями вышеупомянутой языковой группы, мы обеспокоены решением данной проблемы. Первым шагом на пути решения данной проблемы стало, разработка мобильного приложения (Библиотека национальной художественной и учебной литературы Карачаево-Балкарского языка) для операционной системы IOS, которая упростит доступ пользователям к литературе. Для юзеров, приложение должно быть интуитивно понятным, простым и позволяющим осуществлять легкий поиск всей необходимой информации.

Платформой для разработки и внедрения проекта была выбрана среда разработки Xcode, языком программирования – Swift, глобальной серверной частью – Firebase, локальной базой данных – Realm, системой резервного сохранения и контроля версий – GitHub. Для создания этого продукта, были определены нижеизложенные задачи:

- произвести анализ проблемной области;
- идентифицировать конструкции приложения;
- произвести анализ и определить инструментальные средства разработки;
- сформировать модель приложения, разработать макет интерфейса;
- разработать строение и связь баз данных (серверной и локальной);
- разработать главные функции и структуру распределения MVC (Model View Controller);
- наполнить контентом серверную часть приложения;
- произвести тестирование (Unit, пользовательское);
- отредактировать выявленные сбои и произвести финальные исправления;
- загрузить проект в AppStore.

Основной этап разработки мобильного приложения

Калибровка и внедрение Firebase | Realm

Firebase Realtime являют собой облачную базу данных. Информация хранятся как JSON, так и в виде файлов и обновляется в реальном времени с каждым подключенным пользователем. Для конфигурации Firebase, переходим на сайт firebase.google.com и создаем новый проект. Для настройки проекта необходимо получить идентификационный код IOS проекта.

Выполнив все необходимые пункты, приступаем к заполнению базы данных с помощью инструмента Data base

После окончания настройки Firebase, переходим к интеграции локальной базы данных. Realm — межплатформенная мобильная база данных для IOS (доступная в Swift).

Перед интегрированием Realm необходимо инициализировать CocoaPods, дающая возможность использовать внешние фреймворки. Внедрение Realm происходит с помощью команд в консоли проекта.

Чтобы инициализировать локальную базу данных, необходимо создать модели данных Realm. Данный проект будет содержать три модели название, обложка, синопсис.

Завершив наполнение базы данных, приступаем к созданию процессов связей проекта, Realm Base и Firebase Database.

Генерируем базовый файл взаимодействия с базой данных `AddFirebaseToRealm.swift` (рисунок 1). Этот файл будет отправлять принятые данные в локальную базу данных Realm.

```

9  import UIKit
10 import RealmSwift
11 import FirebaseDatabase
12
13 struct DictionaryStruct {
14     let AudioURL : String!
15     let Details : String!
16     let KbrWord : String!
17     let RusWord : String!
18     let UID : String!
19 }
20
21 struct PhrasebookStruct {
22     let AudioURL : String!
23     let OrderID : String!
24     let KbrPhrase : String!
25     let RusPhrase : String!
26     let Sample : String!
27     let UID : String!
28 }
29
30 class AddFirebaseToRealm: NSObject {
31
32     var ref: DatabaseReference!
33
34     // MARK: - Dictionary stack
35
36     class func AddDictionaryToRealm() {
37         let UIRealm = try! Realm()
38
39         let ref = Database.database().reference()
40         ref.child("Dictionary").queryOrderedByKey().observe(.childAdded, with: { snapshot in
41             let value = snapshot.value as? NSDictionary
42             let AudioURL = value?["AudioURL"] as! String
43             let Details = value?["Details"] as! String
44             let KbrWord = value?["KbrWord"] as! String
45             let RusWord = value?["RusWord"] as! String
46             let UID = value?["UID"] as! String
47
48             let WordAdd = DictionaryModel()
49             WordAdd.UID = UID
50             WordAdd.RusWord = RusWord
51             WordAdd.KbrWord = KbrWord
52             WordAdd.Details = Details
53             WordAdd.AudioURL = AudioURL
54             try! UIRealm.write {
55                 UIRealm.add(WordAdd)
56                 print(" ----- Upload Dictionary ----- \ \(WordAdd.UID) to Realm")
57             }
58         })
59     }

```

Рисунок 1 –Фрагмент файла AddFirebaseToRealm.swift

Вышеуказанным способом, создаем файл GetFromRealm.swift, дающий возможность взаимодействовать с данными из локальной базы данных Realm. Этот протокол контактирует с API CocoaPods Realm, а после полученные данные передается в промежуточный массив (GetBookFromRealm, CoverFromRealm).

Применять этот протокол GetFromRealm можно путем вызова отдельные подфункции (GetDictionaryFromRealm, GetPhrasebookFromRealm).

Для наглядной части проекта требуется подготовить цифровые ресурсы (логотипы, иконка, изображения кнопок). В среде разработки Xcode ресурсы содержатся в каталоге Assets.xcassets.

При загрузке приложения стартовым экраном иницируется LaunchScreen.storyboard. Этот файл содержит визуальный облик дисплея загрузки.

В этом проекте будет встроен Tab Bar Controller, позволяющий осуществлять навигацию по разделам меню (словарь, разговорник, избранное). Также для передвижения по внедренным представлениям, есть необходимость внедрить Navigation Controller.

Окончив подготовку главных этапов, переходим к разработке ViewController. Сначала передвигаем новый ViewController на Main.storyboard из Object Library.

После этого, генерируем файл, ответственный за программную часть ViewController. В этом файле будут производиться все процессы и процедуры, которые необходимы для корректной работы данного View.

По предыдущему же паттерну, разрабатываются оставшиеся файлы программного кода и требующиеся представления, обеспечивающие функционал приложения.

1. Структура и интерпретация компьютерных программ / Харольд Абельсон, Джеральд Сассман - Добросвет, КДУ, 2018. - 608 с.
2. Cocoa Design Patterns / Eric Buck, Donald Yactman, 2009. - 456 с.
3. Core Data Programming Guide: Persistent Store Types and Behaviors [Электронный ресурс] / Apple Inc. – Режим доступа: <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/CoreData/>. – Загл. с экрана. – яз. англ.
4. Foundation Functions Reference – Apple Developer [Электронный ресурс] / Apple Inc. – Режим доступа: https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Miscellaneous/Foundation_Functions/. – Загл. с экрана. – яз. англ.
5. GitHub – GitHub [Электронный ресурс] / GitHub Inc. – Режим доступа: <https://github.com/>. – Загл. с экрана. – яз. англ.
6. iOS 11 App Development Essentials / Neil Smyth, 2018. - 812 с.
7. iOS Apprentice / Matthijs Hollemans, 2013. - 816 с.
8. iOS Programming: The Big Nerd Ranch Guide / Christian Keur, Aaron Hillegass, 2010. - 416 с.
9. iPhone App Development: The Missing Manual / Craig Hockenberry, 2010. - 352 с.
10. Model-View-Controller - Википедия [Электронный ресурс] / Wikimedia Foundation, Inc. – Режим доступа: <https://ru.wikipedia.org/wiki/Model-View-ControllerMVC>. – Загл. с экрана. – яз. рус., англ.
11. Pro Swift / Paul Hudson, 2016. - 147 с.
12. Push-уведомления - Википедия [Электронный ресурс] / Wikimedia Foundation, Inc. – Режим доступа: http://ru.wikipedia.org/wiki/Push_уведомления. – Загл. с экрана. – яз. рус., англ.
13. Госдума приняла закон об изучении родных языков <http://www.segodnia.ru/print/204564>

Каргин М.С., Сагаева И.Д.

Реализация программной множественной тахометрии на микроконтроллерах

*Саратовский государственный университет
(Россия, Саратов)*

doi: 10.18411/lj-06-2021-61

Аннотация

Статья посвящена практической реализации программной тахометрии для нескольких каналов. В статье сравнивается аппаратный и программный подход к измерению частоты сигнала, а также описан блок кода на языке C/C++, реализующий программный подход. В заключение приведен способ определения рациональности использования программного подхода перед аппаратным.

Ключевые слова: тахометрия, частотомер, микроконтроллеры, алгоритмы.

Abstract

The article is devoted to the practical implementation of software tachometry for several channels. The article compares the hardware and software approaches to measuring the signal frequency, and also describes a block of C / C ++ code that implements the software approach. In conclusion, a method for determining the rationality of using the software approach over the hardware one is presented.

Keywords: tachometry, frequency meter, microcontrollers, algorithms.

Для различных производственных и бытовых задач часто возникает потребность произвести измерение частоты некоторого физического процесса. Существует