

УДК 004

The model of accounting for the research activity of education system employees based on the IOS operating system

Olena Tolstoluzka, Krystyna Afanasieva

Tolstoluzka Olena

*professor of the Department of Theoretical and Applied Systems Engineering
Kharkiv National University named after V. N. Karazin, Kharkiv,
Svobody Square, 4, 61022*

e-mail: elena.tolstoluzka@karazin.ua;

<https://orcid.org/0000-0003-3085-2346>

Afanasieva Krystyna

*student of the Department of Theoretical and Applied Systems Engineering
Kharkiv National University named after V. N. Karazin, Kharkiv, Svobody
Square, 4, 61022*

e-mail: kristina.afanasieva.dev@gmail.com

<https://orcid.org/0000-0002-6262-4818>

The process of reviewing and analyzing of scientific information resources has been discussed in the paper. Analysis of software solutions, as well as, the conclusions as for the possibility of developing additional information arrays based on existing sources of scientific and scientometric information have been presented. The paper offers the approaches to storing scientometric information, the optimized methods of obtaining and processing data. That provides the ability to digitize publications to simplify searching and filtering scientific information and its further processing and replenishment of information arrays with a new data. The architectural templates have been analyzed and the Viper has been selected as the most suitable architectural template for the distribution of responsibilities and test coverage of the application. The choice of Swift programming language has been substantiated. Strengths and weaknesses of these technologies have been considered. The OS and Firebase data storage tool have been selected and discussed. The advantages of those technologies for the development of a mobile application have been listed. The purpose of the work is to create a mobile application for the operating system "iOS", version 11.0. The application must be a client that works with Firebase Cloud Firestore, which allows searching and filtering publications, as well as, adding and deleting records from Firebase Cloud Firestore for the administrators. Therefore it is necessary to implement the functionality that allows registering new and authorizing registered users. The object of the research is processing scientometric and scientific knowledge. The subject of research is the methods and tools of technology to automate the process of knowledge recording based on the scientific and scientometric information.

Keywords: filter, search, optimization, subject area, Scopus, Web of Science, Swift, iOS, application, Firebase, CocoaPods, operating system.

Модель урахування дослідницької діяльності працівників системи освіти на базі операційної системи IOS

О. Г. Толстолузка, Х. О. Афанасьєва

Толстолузка

Олена Геннадіївна

*д.т.н., с.н.с., професор кафедри теоретичної і прикладної системотехніки
факультету комп'ютерних наук;*

Харківський національний університет імені В. Н. Каразіна,

майдан Свободи, 4, Харків, Україна, 61022

e-mail: elena.tolstoluzka@karazin.ua;

<https://orcid.org/0000-0003-3085-2346>

Афанасьєва

Христина Олександрівна

студентка кафедри теоретичної та прикладної системотехніки;

*Харківський національний університет імені В. Н. Каразіна, м. Харків,
майдан Свободи, 4, 61022*

e-mail: kristina.afanasieva.dev@gmail.com

<https://orcid.org/0000-0002-6262-4818>

Робота присвячена огляду та аналізу ресурсів наукової інформації, аналізу засобів програмних рішень для досягнення поставленої мети, дослідження можливостей розробки додаткових інформаційних масивів на базі існуючих джерел наукової та наукометричної інформації. В рамках роботи запропоновані підходи зберігання наукометричної інформації, способи отримання та обробки даних, що дозволить оптимізувати час роботи з інформацією та надасть можливість оцифровки публікацій, що дозволяє залучати менше ресурсів для реалізації робіт з пошуку та фільтрації наукової інформації і подальшої її обробки та поповнення інформаційних масивів новими даними. В роботі були

проаналізовані архітектурні шаблони після чого зроблено висновок, що для найкращого розподілу обов'язків, і покриття додатку тестами, найбільше підходить архітектурний шаблон - VIPeR. Обґрунтований вибір мови програмування Swift. Було розглянуто сильні та слабкі сторони при використанні цих технологій. Розглянуто обрану ОС та засіб збереження даних Firebase. Перераховані переваги вищеперахованих технологій для розробки мобільного додатку. Метою роботи є створення мобільного додатку для операційної системи "iOS" компанії Apple, версії вище 11.0. Додаток має являти собою клієнт який працює з Firebase Cloud Firestore, який дозволяє здійснювати пошук та фільтрацію публікацій а також у випадку, якщо користувач є адміністратором додавати та видаляти записи з Firebase Cloud Firestore. В додатку необхідно реалізувати функціонал, що дозволяє виробляти реєстрацію і авторизацію зареєстрованим користувачам. Об'єктом дослідження є процес обробки наукометричних та наукових знань. Предметом дослідження є методи і засоби інформаційних технологій для автоматизації процесу обліку знань на базі ресурсів наукової та наукометричної інформації.

Ключові слова: *фільтр, пошук, оптимізація, предметна область, Scopus, Web of Science, Swift, iOS, application, Firebase, CocoaPods, операційна система.*

Модель учета исследовательской деятельности работников системы образования на базе операционной системы IOS

Е. Г. Толстолужская, К. А. Афанасьева

**Толстолужская
Елена Геннадьевна**

*д. т. н., с. н. с. профессор кафедры теоретической и прикладной системотехники факультета компьютерных наук
Харьковский национальный университет имени В. Н. Каразина
г. Харьков, площадь Свободы, 4, 61022
e-mail: elena.tolstoluzka@karazin.ua;*

<https://orcid.org/0000-0003-3085-2346>

**Афанасьева
Кристина
Александровна**

*студентка кафедры теоретической и прикладной системотехники
Харьковский национальный университет имени В. Н. Каразина
г. Харьков, площадь Свободы, 4, 61022
e-mail: kristina.afanasieva.dev@gmail.com*

<https://orcid.org/0000-0002-6262-4818>

Работа посвящена обзору и анализу ресурсов научной информации, анализа средств программных решений для достижения поставленной цели, исследования возможностей разработки дополнительных информационных массивов на базе существующих источников научной и наукометрической информации. В рамках работы предложены подходы хранения наукометрической информации, способы получения и обработки данных, что позволит оптимизировать время работы с информацией и позволит оцифровки публикаций, позволяет привлекать меньше ресурсов для реализации работ по поиску и фильтрации научной информации и последующей ее обработки и пополнение информационных массивов новыми данными. В работе были проанализированы архитектурные шаблоны после чего сделан вывод, что для лучшего распределения обязанностей, и покрытие приложении тестами, больше подходит архитектурный шаблон - VIPeR. Обоснован выбор языка программирования Swift. Были рассмотрены сильные и слабые стороны при использовании этих технологий. Рассмотрены выбранную ОС и средство хранения данных Firebase. Перечисленные преимущества вышперечисленных технологий для разработки мобильного приложения. Целью работы является создание мобильного приложения для операционной системы "iOS" компании Apple, версии выше 11.0. Приложение должно представлять собой клиент работающий с Firebase Cloud Firestore, который позволяет осуществлять поиск и фильтрацию публикаций а также в случае, если пользователь является администратором добавлять и удалять записи из Firebase Cloud Firestore. В приложении необходимо реализовать функционал, позволяющий производить регистрацию и авторизацию зарегистрированным пользователям. Объектом исследования является процесс обработки наукометрических и научных знаний. Предметом исследования являются методы и средства информационных технологий для автоматизации процесса учета знаний на базе ресурсов научной и наукометрической информации.

Ключевые слова: *фильтр, поиск, оптимізація, предметная область, Scopus, Web of Science, Swift, iOS, application, Firebase, CocoaPods, операционная система.*

1. Introduction

The rapid development of information technology, its introduction into all spheres of modern life has led to a sharp expansion of the concept of knowledge processing in computer systems. At the beginning of the last century the total amount of human knowledge doubled about every fifty years, nowadays it doubles every five years. That is why the problem of creating methods that will help rationally organize the process of storing and operating information is of great importance.

Every year, the volume of scientific information grows. The number of experiments and publications in various subject areas has been increasing constantly, resulting in an ever-growing need for recording of scientometric information.

The development and use of scientometric knowledge by modern digital technologies will form a comprehensive view of the analysis of publishing activity and the possibility of using obtained data to make important management decisions.

It is difficult to imagine a modern man without a mobile device in his hands. Whether it's a phone, smartphone or other communicators - these devices are firmly and, apparently, for a long time, have become established in society as indispensable everyday means of communication. With the advent of new mobile devices and the variety of platforms on which they are developed, the business of creating mobile applications for various purposes is expanding actively.

Due to the spread of COVID-19 we are facing with the issue of distancing all educational and scientific processes, and recording the scientific information is not an exception.

2. Formulation of the problem

The purpose of the work is to create a mobile application for optimizing information about the scientific activities of the education system employees based on the IOS operating system.

To achieve this goal, the following tasks have been formulated:

- to analyze the literature on existing software solutions and choose technologies for creating a mobile application;
- to create a data storage model using Firebase;
- to develop a mobile application for optimizing the research activities of the education system employees by using the Swift programming language;
- to test the developed software;
- to create the documentation for the developed application;
- to analyze the quality and speed of test results of the developed application;
- to form an explanatory note.

3. Main part

The application prototype is the computer model of accounting for the scientific activities of the education system employees created by using the C++ language, which has been developed as a bachelor's thesis.

The main advantages of mobile applications are the simplicity of the communication between the brand and the user, economic benefits and ease of use.

Interactive versions of printed publications are becoming more and more popular, which is why we would like to overcome the limitations of working only with publications that are part of Scopus or Web of Science and cover most of this subject area.

A mobile application can be an effective tool for attracting new students to participation in scientific conferences or a convenient for working with existing publications.

The developed product has the following functionality:

- Integration with Firebase, namely:
 - Google Authorization.
 - Authorization via Facebook.
 - Authorization via mail.
 - Data storage.
- The authorization not only for administrators but also for regular users, which will allow user-friendly functionality, has been added.
- The automatic check for belonging to the administrators' list has been implemented for authorization via mail. Successful check provides access to the functionality intended for the administrators only, namely:
 - Adding new records with the possibility to attach a document.
 - Deleting old records.
- Adding publications to the favorites list.

- Viewing all posts.
- Filtering the mail.
- Searching for the publications by specified parameters.
- Downloading the document.

4. Choice of programming language

After analyzing the advantages and disadvantages of Swift and Objective-C, Swift has been selected as more convenient. Swift is a more productive language that allows faster creating of new applications. The main differences between these languages are presented in Table 1.

Table 1 - Comparison of programming languages Swift and Objective-C

	Swift	Objective-C
Accessibility to users of other programming languages	The syntax is similar to other well-known programming languages such as: Python, Go, JavaScript.	The syntax is much different from other programming languages, so it is quite difficult to learn.
Code length	The code is more streamlined, which reduces its length and development time.	The implementation of the same functions requires 2 times more lines of code than in Swift.
Interactivity	You can create applications interactively.	You cannot create applications interactively.
Community using the language	Swift is much easier to learn, but there are far fewer professionals now than in Objective-C.	Most of the iOS applications in the industry are written in Objective-C.
Safety	Security is one of the core concepts of Swift. Its design eliminates several types of potential errors in Objective-C.	Not as secure as Swift.
Open source	Swift is an open source language. It is also used in Linux, which creates the possibility of its application in the development of programs for Android OS.	Has no source code, and is exclusively focused on Apple devices.
Completeness and stability of the language	Swift is still under development and Apple is constantly releasing and adding new features.	Objective-C has already been created and used consistently.

When using Swift, the development process is simplified, easily accessible for novice programmers, thereby reducing the cost of highly qualified specialists. The Apple support for language development plays an important role as well. Rapid growth of the Swift community and its open-source character

should be mentioned. Swift has been chosen because it is free and therefore available to a large number of developers. Fig. 1 shows a graph of the growth of the Swift developer community since its launch.

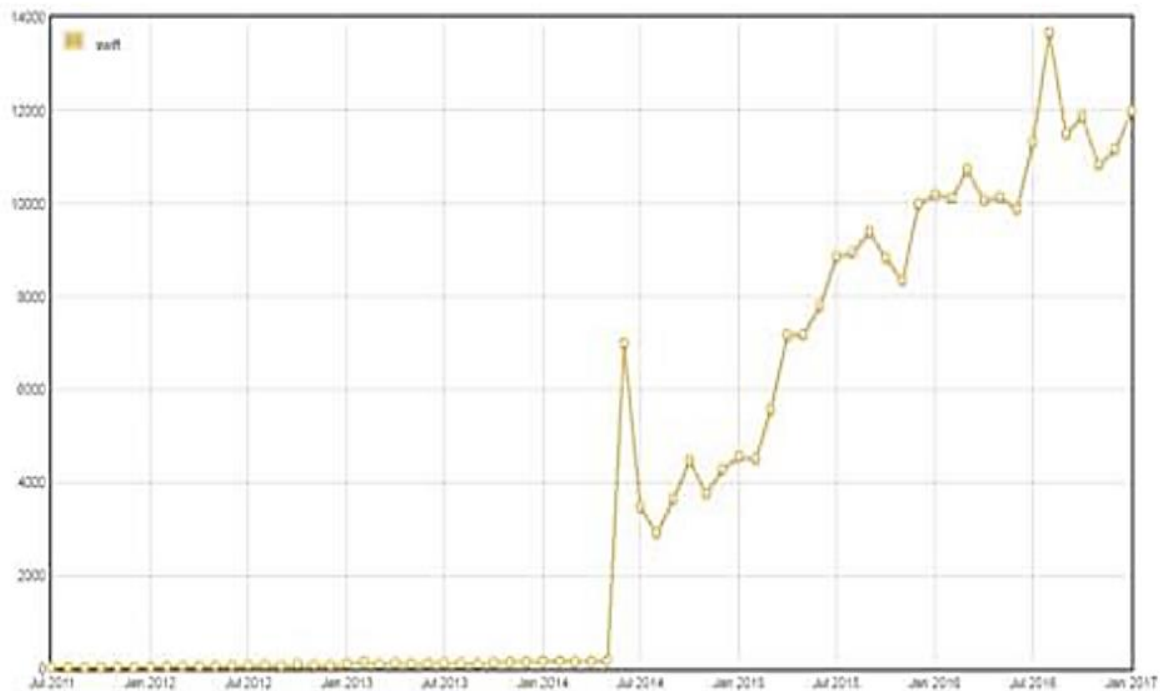


Fig. 1 - Swift community growth graph

Reducing the time to write an application allows you to reduce costs and increase the company's profit. Swift is suitable for writing new applications in short time.

5. Choosing an Application Architecture Pattern

As a result of the analysis of architectural patterns, the following conclusions can be made:

- 1) Traditional MVC is not applicable to modern IOS development.
- 2) Apple MVC is the best architectural pattern, in terms of development speed, but noticeably complicates the application support.
- 3) MVP offers good testability, but also means a significant increase in code size. Its features lie in the distribution because most of the responsibilities are divided between the Presenter and the Model, while the View has a minimum of tasks. Testability is also an advantage that is you can test most of the business logic because of the practical empty View. MVP, as well as traditional MVC, is not suitable for the IOS development.
- 4) The peculiarities of MVVM are in propagation. The View in MVVM has more responsibilities than the View in MVP, since MVVM updates the state from the ViewModel, and the MVP only passes all events to the Presenter and does not update itself. In testability the ViewModel knows nothing about the View, which allows you to test it without problems. The View can also be tested, but since it depends on UIKit it is not advisable to test it. Thus, MVVM combines the advantages of all the previously considered approaches to the architecture of IOS applications and it does not require additional code to update the View. Testability is also advantageous.

5) VIPER surpasses all presented architectural patterns in distribution, testability, and ease of learning new modules. The main features of VIPER are distribution, namely an excellent ability to assign responsibilities, and testability. It is ease of use as well. A small amount of code in VIPER blocks allows easily switching from one task to another, thereby increasing the usefulness of the programmers' work.

We can conclude that for the best distribution of responsibilities, and tests coverage of the application, the VIPER architectural pattern is the most suitable.

6. Conclusions

The main functions and capabilities of the Firebase database have been analyzed, and the Swift programming language has been chosen for developing a mobile application by using the VIPER architectural pattern.

REFERENCES/ЛІТЕРАТУРА

1. Swift. Open Source Programming Language [Electronic resource]. - access mode: URL: <https://www.apple.com/ru/swift/> (date of access - 04.10.2020)
2. Introduction to Obj-C [Electronic resource]. - access mode: URL: <http://macbug.ru/cocoa/objc> (date of access - 17.10.2020)
3. React Native Tutorial: Build app with JavaScript [Electronic resource]. - access mode: URL: <https://www.raywenderlich.com/126063/reactnative-tutorial> (date of access - 15.10.2020)
4. The MVC design pattern [Electronic resource]. - mode Access: URL: <https://webformyself.com/shablon-proektirovaniya-mvc/> (date requests - 17.10.2020)
5. Model-View-Controller (MVC) in IOS: A Modern Approach [Electronic resource]. - access mode: URL: <https://www.raywenderlich.com/132662/mvc-in-ios-a-modern-approach> (date requests - 17.10.2020)
6. IOS Architecture Patterns [Electronic resource]. - access mode: URL: <https://techblog.badoo.com/blog/2016/03/21/ios-architecture-patterns/> (date requests - 17.10.2020)
7. Implementation of MVVM in IOS using RxSwift [Electronic resource]. - access mode: URL: <https://habrahabr.ru/post/273455/> (date requests - 17.10.2020)
8. The Book of VIPER [Electronic resource]. - access mode: URL: <https://github.com/strongself/The-Book-of-VIPER> (date accessed - 17.10.2020)
9. ANON The Swift Programming Language (Swift 2.1) / ANON – Cupertino: Apple Inc., 2014. 528 p.
10. Vandad Nahavandipoor iOS 8 Swift Programming Cookbook / Vandad Nahavandipoor – Boston: O'Reilly Media., 2014. – 902 p.