

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
PIHII (Russia) = 0.126
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

International Scientific Journal Theoretical & Applied Science

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2020 Issue: 04 Volume: 84

Published: 24.04.2020 <http://T-Science.org>

QR – Issue



QR – Article



Raya Kanatovna Zhirenbaeva

Kazakh National Agrarian University
master's student,
Kazakhstan, Almaty

Maksat Rayev

Kazakh National Agrarian University
master's student,
Kazakhstan, Almaty

Bibinur Zhuzbaevna Kirgizbaeva

Kazakh National Agrarian University
professor, candidate of physical and mathematical sciences
Kazakhstan, Almaty

ANALYSIS OF THE PROCESS OF CREATING AND USING MOBILE APPLICATIONS

Abstract: At the present time, thanks to the fact that the world works in the information space with a large amount of information, it has become one of the most important measures for that, in a short time, for a short period of time. Therefore, for informing the world, only one computer connection became insufficient, it is very important to expand the connection through mobile devices, to use and develop mobile applications. Necessary information technologies for the integration of mobile communications are actively developing. This article discusses the widespread use of mobile applications for the development of mobile communications and their classification by structure and composition. Consider the features of the development of applications for mobile devices and their applications. There were analyzed tools and approaches for creating mobile applications and implemented methods of their use. Shows the process of creating mobile applications for a specific subject area.

Key words: mobile devices, applications, mobile applications, application types, Android Studio environment, development of mobile applications.

Language: Russian

Citation: Zhirenbaeva, R. K., Rayev, M., & Kirgizbaeva, B. Z. (2020). Analysis of the process of creating and using mobile applications. *ISJ Theoretical & Applied Science*, 04 (84), 508-513.

Soi: <http://s-o-i.org/1.1/TAS-04-84-85> **Doi:**  <https://dx.doi.org/10.15863/TAS.2020.04.84.85>

Scopus ASCC: 1700.

АНАЛИЗ ПРОЦЕССА СОЗДАНИЯ И ИСПОЛЬЗОВАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Аннотация: В настоящее время, благодаря тому, что мир работает в информационном пространстве с большим объемом информации, это стало одной из необходимых мер для того, чтобы человечество быстро, в короткие сроки, связалось, чтобы он своевременно мог получить необходимую информацию. Поэтому, для информирования мира одной только компьютерной связи стало недостаточно, очень важно расширять связь через мобильные устройства, использовать и разрабатывать мобильные приложения. Необходимые информационные технологии для внедрения мобильной связи активно развиваются. В данной статье обсуждается широкое использование мобильных приложений для развития мобильной связи и их классификация по структуре и составу. Рассмотрены особенности разработки приложения для мобильных устройств и их внедрение. Были проанализированы инструменты и подходы для создания мобильных

Impact Factor:

ISRA (India) = 4.971	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	РИИЦ (Russia) = 0.126	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

приложений и реализованы способы их применения. Показан процесс создания мобильных приложений для конкретной предметной области.

Ключевые слова: мобильные устройства, приложения, мобильные приложения, типы приложений, среда Android Studio, разработка мобильных приложений.

Введение

Информационные ресурсы, системы и современные технологии являются неотъемлемой, быстрорастущей областью человеческой деятельности. В 1997 году на рынке мобильной связи появилась технология WAP (Wireless Application Protocol), которая позволила устанавливать приложения непосредственно из Интернета на мобильные телефоны без использования кабеля для подключения к компьютеру. С тех пор начался процесс «мобилизации» общества. В начале 2000-х годов разработка мобильных приложений с телефонами с большими сенсорными экранами позволила создать новый уровень мобильных приложений, обеспечивающих качественную связь. С 2010 года мобильные устройства оснащаются высокопроизводительными процессорами, которые позволяют использовать современные информационные технологии для разработки приложений. Многие программисты начали осваивать новые сферы профессиональной деятельности и адаптировались к тенденциям рынка.

В настоящее время разработка мобильных приложений является одним из самых популярных направлений в сфере информационных технологий. Разработка мобильных приложений основана на алгоритмах и задачах, неизвестных пользователю, заранее осведомленных о потребностях потребителей.

Мобильное приложение предназначено для определенной группы потребителей и направлено на решение любых из проблем и трудностей. Мобильное приложение - это функциональное приложение, установленное на конкретной платформе, которое может выполнять определенные действия [1]. Это такая система, которая помогает вам взаимодействовать с различной информацией. В связи с этим мобильные приложения делятся на следующие типы:

- дополнительное событие: позволяет показать различные виды деятельности, такие как спортивные, культурные, образовательные и научные мероприятия.
- сервисное приложение: сервисное приложение, аналогичное сайтам, обслуживающим организации;

- приложения для различных игр для обучения и развития;
- интернет-магазины;
- рекламные приложения, используемые для рекламы различных брендов;
- бизнес-приложения: оптимизирует организационный процесс, обеспечивает доступ к бизнес-информации и интегрирует ее с базой данных;
- системное приложение: использование расширенных настроек и опций для телефона и его программного обеспечения;
- приложение с сервисами навигации и поиска с использованием модуля GPS, которое позволяет использовать телефон в качестве полноценного навигатора;
- приложения в мультимедийных программах, расширяющие возможности телефона при работе с видео и аудиоинформацией;
- социальные сети, онлайн-коммуникационные услуги, распространение информации и приложения для социальных сетей;
- контент приложений и т. д.

Следующим шагом является понимание того, почему возникла такая проблема, и поиск информации о том, как решить эту проблему. Необходимо провести как можно больше исследований в проблемной области. После того, как выявлена проблема, необходимо приступить к оценке влияния мобильного приложения на решение этой проблемы.

Моделирование.

На этом этапе информационная архитектура приложения должна быть разработана. Информационная архитектура - это процесс определения, какая информация и функции должны отображаться в приложении, и как эти данные и функции организованы. Как правило, этот процесс начинается со списка функций, связанных с запуском программы, и определяет, как отобразить его в приложении [3]. Таким образом, это действие является основным структурирующим блоком модели при моделировании приложения. Далее вам нужно начать создавать окно программы и назначить каждую функцию и местоположение данных. В этом случае вы можете быть уверены, что у каждого объекта есть свое место (Рис.1).

Impact Factor:

ISRA (India) = 4.971	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	РИИЦ (Russia) = 0.126	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

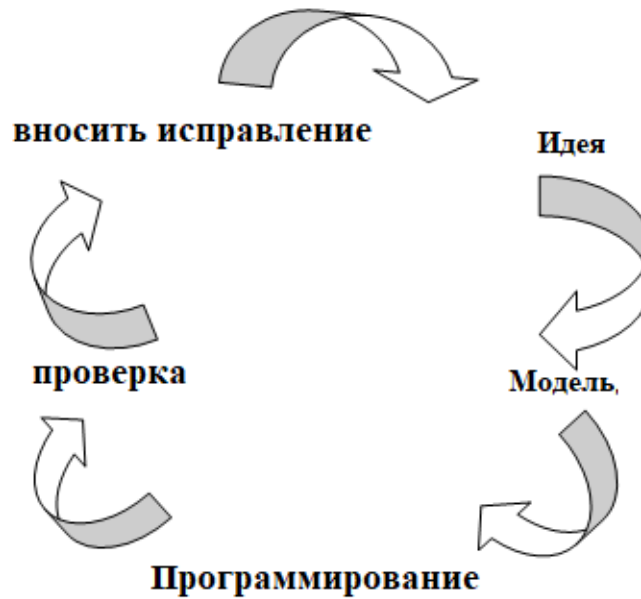


Рисунок 1 - Процесс разработки мобильного приложения

Пользователи должны заранее проанализировать неудобства, которые могут возникнуть при работе с приложением. Приложение определяет, какие объекты расположены на домашней странице, сколько щелчков нужно выполнить, чтобы выполнить действие в меню, и сколько раз вам нужно щелкнуть объект, чтобы выполнить действие на любой странице. Нажатие здесь означает нажатие на объект. Рекомендуется проверить, что каждый клик интуитивно понятен. Если для выполнения действия достаточно одного щелчка, это удобно для пользователя, но не рекомендуется повторно нажимать версию для других задач.

Следующий шаг - щелкнуть и заменить модель. Модель Click and Replace поможет вам протестировать приложение во время работы. Это в основном особый способ тестирования интерфейса приложения на мобильном телефоне [4]. Например, потребители получают ссылку по телефону, которая позволяет им выходить за пределы приложения, когда они его открывают. Хотя приложение в настоящее время не имеет никакой функциональности, они могут щелкнуть на каждой странице приложения и начать навигацию по приложению. Если возникнут проблемы, необходимо внести изменения в схему применения.

Структурные блоки дизайна программы основаны на стилистических инструкциях. Полезно обратить пристальное внимание на руководство по стилю звука при использовании программы. Благодаря индивидуальному языку дизайна, приложение будет удобным для пользователей. Таким образом, «срок службы»

мобильного приложения напрямую связан с его дизайном [5].

Программирование - это высокий уровень технического дизайна. Многие методы, технологии и языки программирования используются для создания мобильного приложения. У каждого из них есть свои плюсы и минусы. Некоторые из них дешевле в использовании, но менее производительны, в то время как другие требуют больше времени и более высокого уровня профессионализма. Самый простой способ, для создания приложения использовать распространенную надежную технологию, то есть без кодирования программного обеспечения можно создавать приложения на разные темы, используя готовые шаблоны [6].

Однако приложения, созданные с помощью таких готовых шаблонов, имеют много недостатков, таких как отсутствие полного контроля, зависимость от имеющегося дизайна, ограниченного интерфейса и т.д.

Системы проектирования Front-end и Back-end могут быть использованы при программировании мобильных приложений.

Предыдущий дизайн, Front-end дизайн - это проект приложения клиента. Другими словами, это дизайн интерфейса между пользователем и сервером приложений. Это позволяет пользователю вводить некоторые данные, а также первоначально обрабатывать их и отправлять на сервер через соответствующий API (интерфейс прикладного программирования).

Внутренний дизайн, Back-end design - это разработка серверной части приложения, которая отвечает за передачу данных между пользователями и ресурсами [7]. Есть несколько

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.126
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

вещей, которые нужно сделать перед написанием кода. Языки программирования. Наиболее часто используемыми языками для создания API являются Java, C #, Javascript, PHP и Python.

База данных основана на двух основных базах данных SQL и noSQL. Традиционными типами SQL являются MSSQL и MYSQL.

В ходе реализации инфраструктуры на этом этапе будут решаться вопросы развертывания API и базы данных. Принятые здесь решения помогут определить стоимость хостинга, масштабируемость, исполнение и надежность приложения. Обычные хостинг-провайдеры включают Amazon AWS и Rackspace. Помимо выбора провайдера, вам необходимо спланировать степень расширения системы. Облачные решения позволяют вам рассматривать ресурсы как полезные и использовать их по мере необходимости. Они также помогают выполнять резервное копирование базы данных, время работы сервера и обновления операционной системы [8].

Приложения - написанные на языках программирования для конкретной платформы и интегрированные в операционную систему, работают быстро и правильно, а также обладают преимуществами функциональности и скорости работы других мобильных приложений. Они позволяют этой платформе выполнять общие взаимодействия с интерфейсом программы самым простым способом [9]. Кроме того, доступны аппаратные приложения: видекамера, микрофон, акселерометр, телефонная книга и т. д. Конечно, для написания такой программы требуется много времени, поэтому вам необходимо иметь специальные знания для использования среды разработки программ, а также тот факт, что каждая платформа имеет свои собственные программы, что увеличивает спрос на программы.

Такие приложения имеют свои инструменты и языки программирования. Например, Java часто используется для Android Studio для написания программ для операционной системы Android, а также Xcode и Objective-C для iOS, а также недавно популярных языков программирования Swift, Visual Studio для Windows Phone и C #. эффективно работает [8].

Иногда вам нужны программы, которые могут работать на нескольких платформах, что позволяет создавать приложения за короткий промежуток времени в дополнение к собственным приложениям. В этом случае выбираются гибридные или веб-приложения, а для разработки

используются мобильные платформы с кроссплатформенностью [4].

Основы дизайна мобильных приложений.

Сегодня существует множество инструментов и структур, которые помогают вам создавать мобильные приложения. Примерами могут служить IONIC, Mobile Angular UI, Intel XDK, Titanium Devices, Sencha Touch, Kendo UI, PhoneGap.AppceleratorTitanium - платформы для написания приложений как на настольных компьютерах, так и на мобильных телефонах.

На этой платформе вы можете создавать приложения для телефонов с такими операционными системами, как iOS, Android, BlackBerry. Вам нужно знать JavaScript, чтобы написать программу. Программа состоит из объектов со специальными свойствами и методами. Большой список объектов позволяет использовать все возможности операционной системы. Приложение JavaScript взаимодействует с Appcelerator Titanium API. Вы можете использовать различные компоненты пользовательского интерфейса для создания таких элементов, как текстовые поля, кнопки, списки. Точность этих объектов обеспечивается надлежащим управлением мобильной платформой. В большинстве случаев код, написанный на одной платформе, может работать на другой платформе без каких-либо изменений.

Однако не все различия в платформе отражаются правильно. В этом случае вам нужно написать отдельную таблицу кодов для каждой конкретной платформы. Чтобы создать небольшое приложение, необходимо знать только несколько объектов из Titanium.UI. Вот один из примеров: Titanium.UI.Window - один из наиболее важных объектов с различными свойствами и методами. Можно указать фоновое изображение, тему, расположение объектов и т.д. Для объекта можно указать:

```
var windows = Titanium.UI.createWindow
({backgroundColor: 'red', title: 'My window'}) можно
установить; windows.open (); itTitanium.UI. Здесь
кнопка используется для выполнения
определенных действий.
```

```
varbutton = Titanium.UI.createButton ({title:
'Hello'}); button.addEventListener ('click', function (e)
{Titanium.API.info ("button");});
```

```
itTitanium.UI.ImageView - это элемент для
отображения изображения. var image = Titanium.UI.
createImageView ({изображение: 'myimg.png',
ширина: 'авто', высота: 'авто'}); view.add (Рис. 2);
```

Impact Factor:

ISRA (India) = 4.971
ISI (Dubai, UAE) = 0.829
GIF (Australia) = 0.564
JIF = 1.500

SIS (USA) = 0.912
РИИЦ (Russia) = 0.126
ESJI (KZ) = 8.716
SJIF (Morocco) = 5.667

ICV (Poland) = 6.630
PIF (India) = 1.940
IBI (India) = 4.260
OAJI (USA) = 0.350

Mediacation-list листинг программы

```
<ion-title> Карта объектов </ion-title>
</ion-navbar>
</ion-header>
<ion-content>
<div>
<ion-segment [(ngModel)]="tab">
<ion-segment-button value="map">
Карта
</ion-segment-button>
<ion-segment-button value="list">
Тизим
</ion-segment-button>
</ion-segment>
</div>
```

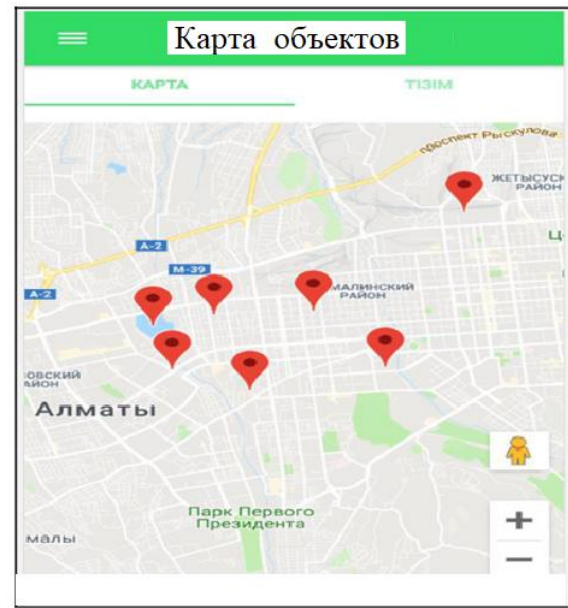


Рисунок 2 - Mediacation-list

Android основан на многочастном ядре Linux. Он обеспечивает системное разделение на функции и отвечает за безопасность, управление памятью, энергосистемами и процессами, а также использует модель драйверов с сетевым стеком.

На «верхнем» уровне ядра находится коллекция библиотек для основных базовых функций приложения как программного обеспечения, так и аппаратного. Таким образом, этот уровень представляет алгоритмы, подготовленные для более высоких уровней, поддерживает форматы файлов, кодирует информацию, выполняет декодирование, создает графику и так далее. Библиотеки создаются на C / C ++ на основе определенного оборудования [10].

Используя этот код, разработчик увеличивает скорость по сравнению с Java или Swift, но скорость работы программы может уменьшиться, потому что в программах увеличивается объем выполняемых операций. Таким образом, программист должен предварительно выбрать конкретную задачу или скорость выполнения.

Платформа Xamarin может использоваться для создания мобильных приложений. Xamarin использует язык C #. С помощью этой системы можно создать логику одного приложения на нескольких платформах одновременно - Android, iOS, Windows Mobile.

Это приложение работает на Android, iOS и Windows Phone. Технология Xamarin.Forms это позволяет. Использование платформы Xamarin основано на том факте, что приложение должно содержать большую часть многоплатформенного кода.

Инструменты Android - Включает эмулятор виртуальных устройств Android (AVD). Эмулятор AVD позволяет тестировать приложение на виртуальном мобильном устройстве с ОС Android. Эмулятор имеет возможность создавать несколько виртуальных устройств с различными конфигурациями. Чтобы создать виртуальное устройство Android, нажмите «Создать» в меню «Окно → Android SDK и AVD Manager», введите имя виртуального устройства и укажите его характеристики [8].

Заключение.

В настоящее время широко используются современные технологии проектирования мобильных приложений, широко используется мобильное программирование. Достижения в области технологий мобильного программирования позволяет разрабатывать, тестировать и вводить в эксплуатацию мобильные приложения.

Мобильные приложения публикуются в Google Play и Apple App Store и используются широко. Это наиболее распространенные типы мобильных приложений, предназначенные для общения в различных играх и социальных сетях, а также доступно множество приложений для электронной коммерции. Разработанные приложения, если они профессионально разработаны, то мобильные приложения занимают особое место в жизни человечества, облегчая их повседневную деятельность.

Impact Factor:	ISRA (India) = 4.971	SIS (USA) = 0.912	ICV (Poland) = 6.630
	ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.126	PIF (India) = 1.940
	GIF (Australia) = 0.564	ESJI (KZ) = 8.716	IBI (India) = 4.260
	JIF = 1.500	SJIF (Morocco) = 5.667	OAJI (USA) = 0.350

References:

1. Kulyashova, N.M., Karpyuk, I.A., & Drozdov, S.A. (2013). *Osnovnyye printsipy postroyeniya i klassifikatsii informatsionno-poiskovykh sistem*. Materialy XVI nauch.konf. molodykh uchenykh, aspirantov i studentov Mordov. gos. un-ta. im. N. P. Ogarova: v 6 ch, CH.3. Yestestvennyye nauki. (pp.66-68). Saransk: Izd-vo Mordov. un-ta.
2. Pantileykin, N.V. (2016). Mobil'nyye prilozheniya i ikh vidy. *Nauchno-metodicheskiy elektronnyy zhurnal Kontsept*, T. 26, pp. 776-780.
3. Khardi, B., & Fillips, B. (2014). *Programmirovaniye pod Android. Dlya professionalov*. (Seriya «Dlya professionalov»). (p.592). SPb.: Piter.
4. Alipbekov, A. S., Aldashova, Z. T., Makovetskaya, A. A., Dulatbaeva, S. S., & Tulenbayev, M. S. (2019). Features of the use of mobile android applications in the educational environment. *ISJ Theoretical & Applied Science*, 05 (73), 479-483. SoI: <http://s-o-i.org/1.1/TAS-05-73-73> Doi: <https://dx.doi.org/10.15863/TAS.05.73.73>.
5. Varakin, M.V. (2012). *Razrabotka mobil'nykh prilozheniy pod Android - UTS «Spetsialist»*, p.592.
6. (2016). Head First. Programmirovaniye dlya Android. (Seriya«Head First O'Reilly»). (p.704). SPb.: Piter.
7. Vogel, L. (2011). *Android Development Tutorial*. (data obrashcheniya: 20.04.). Retrieved from <http://www.vogella.de/android.html>
8. Buch, G., Rambo, D., & Yakobson, I. (2007). *Yazyk UML. Rukovodstvo pol'zovatelya*. 2-ye izd. Moscow: DMK Press.
9. Goloshchapov, A.L. (2011). *GoogleAndroid: Programmirovaniye dlya mobil'nykh ustroystv*. SPb.: BKHV-Peterburg.
10. Khashimi, S., Komatineni, S., & Maklin, D. (2011). *Razrabotka prilozheniy dlya Android*, Piter.