

УДК 004.514+004.42

## **АВТОМАТИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ИНТЕРАКТИВНЫХ ПРОТОТИПОВ ANDROID-ПРИЛОЖЕНИЙ НА ОСНОВЕ НИЗКОДЕТАЛИЗИРОВАННЫХ МАКЕТОВ**

**А. С. Хлопунов<sup>1</sup>, И. С. Шахова<sup>2</sup>**

<sup>1,2</sup>*Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета*

<sup>1</sup>hlopunov598@gmail.com, <sup>2</sup>is@it.kfu.ru

### ***Аннотация***

Приведены механизмы автоматизации процесса разработки интерактивных прототипов мобильных приложений на основе рукописных макетов. Процесс автоматизации включает в себя использование методов машинного обучения для распознавания рукописных макетов. Для обеспечения взаимодействия пользователя с предложенными механизмами реализовано мобильное Android-приложение.

***Ключевые слова:*** прототипирование, UI, UX, мобильные приложения, пользовательский интерфейс

### **ВВЕДЕНИЕ**

Процесс разработки приложения состоит из нескольких этапов, и от продолжительности каждого из них зависят конечная стоимость продукта и время его разработки. Основные из этих этапов:

- 1) поиск идеи и анализ альтернативных решений;
- 2) сбор требований и составление технического задания;
- 3) разработка прототипа приложения и согласование его с заказчиком;
- 4) разработка дизайна приложения и реализация функционала;
- 5) тестирование и согласование с заказчиком [1].

По данным, опубликованным в статье [2], более 50% опрошенных разработчиков считают, что проектирование прототипов экранов является

первоначальной задачей. Это позволяет в кратчайшие сроки получить обратную связь от целевой аудитории и тем самым выявить проблемы в макете на начальном этапе и снизить время и стоимость разработки [3].

Прототипы могут быть как низкодетализированными (нарисованными на бумаге или с помощью специальных инструментов), так и полностью интерактивными, обладающими высокой детализацией и позволяющими взаимодействовать с ними. Соответственно, чем детальнее прототип, тем более точная обратная связь может быть по нему получена. Такие проработанные прототипы позволяют взаимодействовать с экранами макетов приложения: нажимать кнопки, вводить текст, тем самым давая больше отзывов о работе приложения [4]. Однако создание интерактивного прототипа представляет собой более трудозатратный процесс в сравнении с созданием низкодетализированных макетов, а также предполагает наличие навыков работы с инструментами прототипирования [5, 6].

Таким образом, целью данной работы является разработка инструмента автоматизации процесса создания интерактивного прототипа на основе рукописных макетов экранов приложения для операционной системы Android.

### **ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ**

При обзоре альтернативных решений для генерации элементов интерфейса из рукописных макетов были обнаружены следующие инструменты:

- Система проектирования Airbnb. Данная система позволяет распознавать рукописные макеты экранов и динамически генерировать из них код. На демонстрационных роликах ИИ данной системы показывает хорошие результаты распознавания различного типа элементов экрана. Однако данный инструмент предназначен для веб-приложений и не подходит для мобильных, а также на данный момент доступен только внутри корпоративной сети Airbnb.
- Приложение Prott. С помощью данного приложения можно сфотографировать нарисованные прототипы экранов, пометить переходы для кнопок, а затем загрузить все это в графический редактор Sketch, где будет сгенерирован интерактивный прототип. Однако данный программный инструмент не решает поставленную

проблему в полной мере, так как требует взаимодействия со сторонними инструментами.

### РАСПОЗНАВАЕМЫЕ ЭЛЕМЕНТЫ ИНТЕРФЕЙСА

Для прототипирования и обучения нейронной сети был определен следующий набор элементов, используемых в дизайне мобильных приложений. Данный выбор был обусловлен тем, что часть этих элементов является основным компонентом дизайна приложений на Android, а другие являются часто используемыми. Для каждого из них был определен свой стандартный внешний вид.

Были определены следующие элементы интерфейса:

- Button. Кнопка, которая состоит из текста или картинки и обрабатывает касания пользователей (рис. 1).

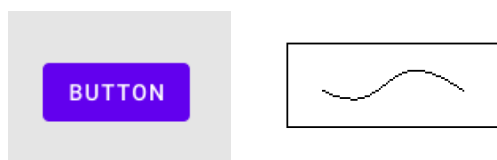


Рис. 1. Button в приложении и вид, заданный в рукописном макете

- ImageView. Данный компонент является контейнером для отображения изображений (рис. 2).



Рис. 2. ImageView в приложении и вид, заданный в рукописном макете

- Spinner. Выпадающий список, который предоставляет быстрый способ выбора одного значения из набора. В свернутом состоянии показывает текущее выбранное значение, а при касании отображается раскрывающееся меню со всеми остальными значениями (рис. 3).

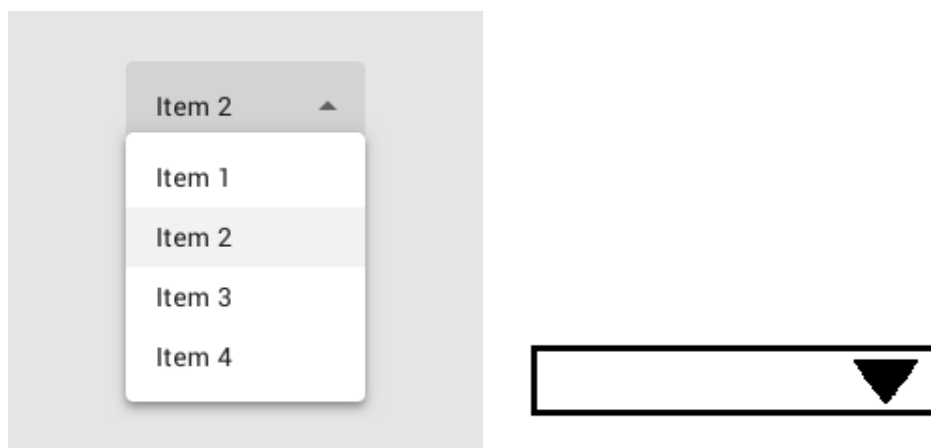


Рис. 3. Spinner в приложении и вид, заданный в рукописном макете

- EditText. Элемент пользовательского интерфейса для ввода и изменения текста (рис. 4).



Рис. 4. EditText в приложении и вид, заданный в рукописном макете

- TextView. Элемент для отображения текста (рис. 5).

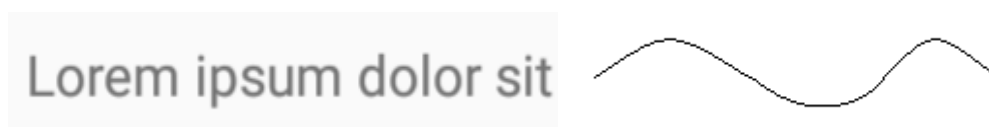


Рис. 5. TextView в приложении и вид, заданный в рукописном макете

- CheckBox. Данный компонент является флагом, с помощью которого можно выбрать нужные опции из определенного набора значений (рис. 6).

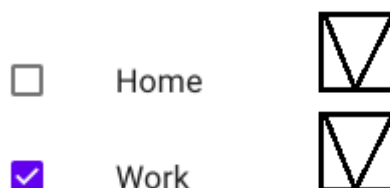


Рис. 6. CheckBox в приложении и вид, заданный в рукописном макете

---

- RadioButton. Элемент, который предоставляет выбрать только один пункт из predetermined набора значений. Никогда не используется в одиночестве, всегда есть несколько таких элементов, объединенных в одну группу (рис. 7).

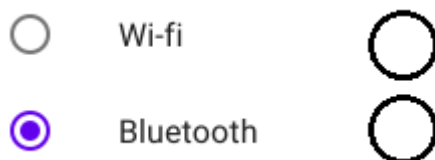


Рис. 7. RadioButton в приложении и вид, заданный в рукописном макете

- Switch. Компонент представляет собой полосу с двумя состояниями. Он используется, когда необходимо сделать выбор между двумя противоположными вариантами (рис. 8).

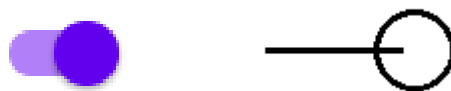


Рис. 8. Switch в приложении и вид, заданный в рукописном макете

- Floating Action Button. Плавающая кнопка, которая обычно выполняет основное или самое распространенное действие на экране. Отображается перед всем содержимым экрана (рис. 9).



Рис. 9. Floating Action Button приложение и вид, заданный в рукописном макете

### **АЛГОРИТМ РАСПОЗНАВАНИЯ И ГЕНЕРАЦИИ ЭЛЕМЕНТОВ**

Алгоритм распознавания элементов с фотографий нарисованных макетов и последующей генерации экранов на их основе использует предварительно подготовленную и обученную нейронную сеть. Для ее обучения использовалось 150 рукописных шаблонов и 2000 сгенерированных с помощью скрипта. Данный алгоритм состоит из следующих основных шагов (рис. 10):

- 1) добавление фотографии в приложение и сохранение пути до нее;
- 2) создание классификатора для распознавания изображений. На вход подаются модель нейронной сети, файл с метками элементов, которые могут быть распознаны, и ожидаемые размеры изображений после сжатия;
- 3) первый запуск метода `decodeResource()` для `Bitmap` с флагом `inJustDecodeBounds = true`, что позволяет узнать размеры изображения без его загрузки в оперативную память;
- 4) вычисление коэффициента сжатия;
- 5) второй запуск метода `decodeResource()` со сжатым изображением;
- 6) запуск распознавания на `Bitmap`, полученном из изображения;
- 7) заполнение листа элементами, генерируемыми из параметров, полученных посредством распознавания: идентификатора, метки (названия элемента), точности распознавания, координат прямоугольной области, в которой находится элемент;
- 8) вычисление метрик экрана и коэффициентов отношений высоты и ширины между устройством и сжатым изображением;
- 9) парсинг пришедших результатов и генерация элементов интерфейса в зависимости от метки;
- 10) вычисление расположения и размеров элементов на основе координат прямоугольной области и коэффициентов отношений ширины и высоты;
- 11) генерация из полученных элементов экрана.

Запуск нейронной сети и процесс распознавания элементов происходит только один раз для каждого изображения. После процесса распознавания полученные из него элементы, удовлетворяющие по точности распознавания алгоритму генерации (в данном случае 65%), записываются в таблицу в базе данных. При последующих запусках элементы извлекаются из базы данных, конвертируются в соответствующий формат и подаются на вход алгоритму генерации экрана.

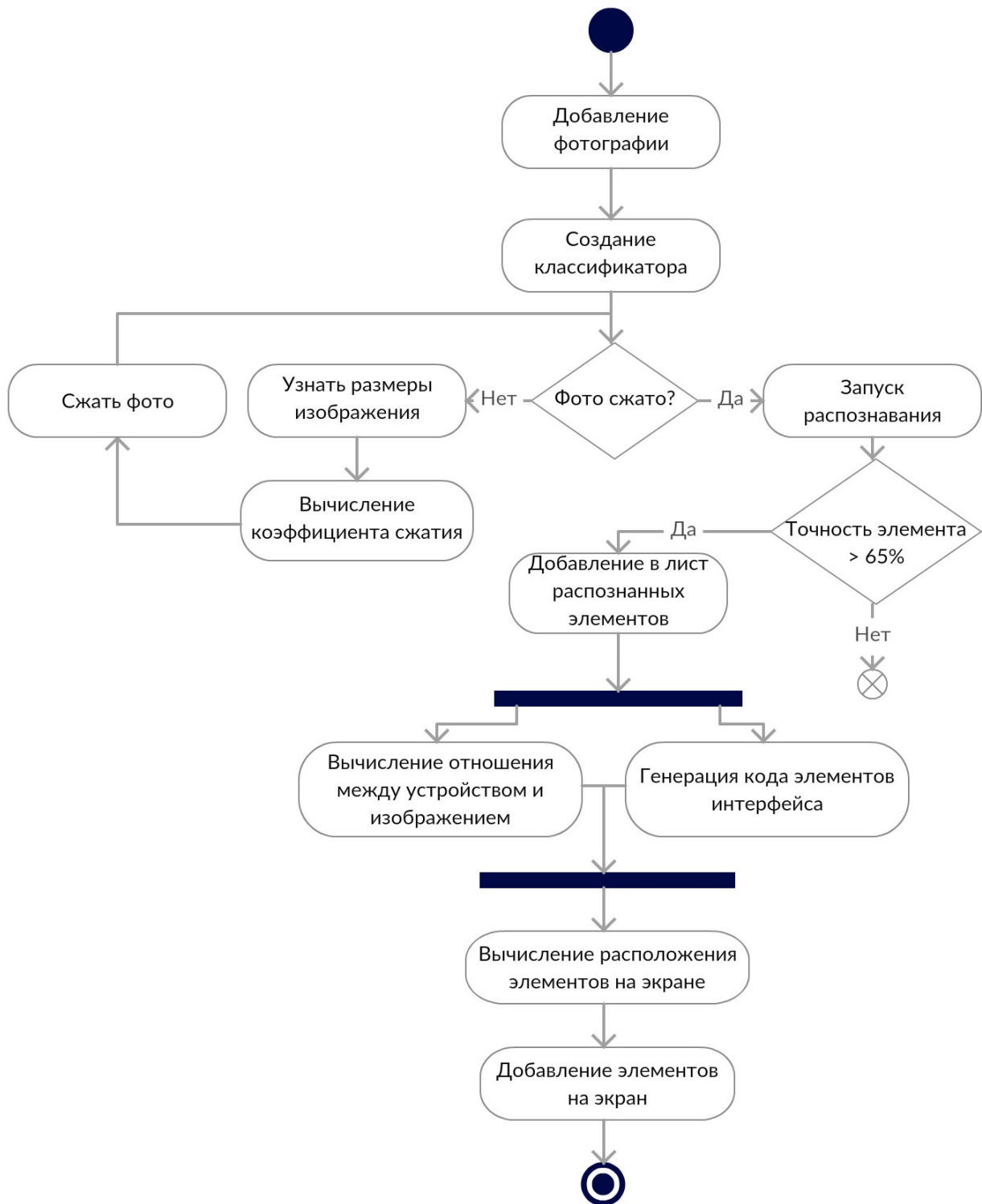


Рис. 10. Алгоритм распознавания и генерации элементов

## ОПИСАНИЕ РАБОТЫ ПРИЛОЖЕНИЯ

Работа пользователя с разработанным инструментом осуществляется через графический интерфейс мобильного приложения. На рисунке 11 изображена диаграмма сценариев использования мобильного приложения.



Рис. 11. Диаграмма сценариев использования приложения

После запуска приложения и выдачи ему разрешений на доступ к файлам на устройстве пользователю предлагается создать новый проект – это область, где хранятся фотографии макетов экранов для каждого отдельного заказчика или проекта. После выбора проекта открывается галерея приложения, в которой хранятся загруженные шаблоны. Здесь же можно загрузить и новые фотографии с устройства (рис. 12). При клике по шаблону его изображение передается в нейронную сеть и открывается сгенерированное на его основе окно.



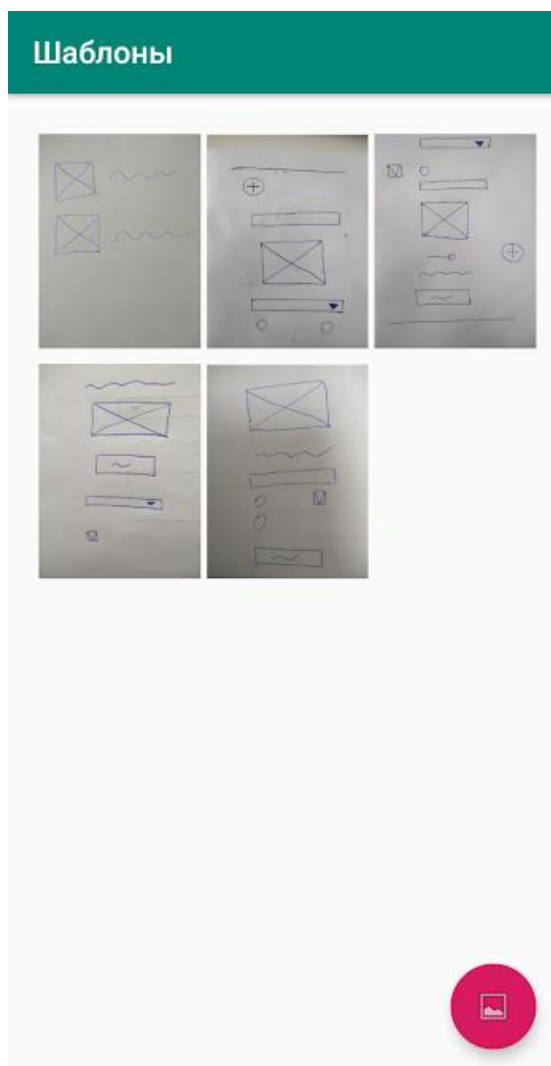


Рис. 12. Экран галереи приложения

Все элементы являются интерактивными, то есть в поля для ввода текста можно вводить текст, в выпадающих списках можно выбрать одно из значений и др. Также есть возможность добавления перехода с одного экрана на другой через нажатие обычной кнопки или плавающей кнопки. Для этого необходимо удерживать кнопку в течение 1–2 секунд, и после этого будет предложено выбрать экран из этого же проекта, на который необходимо производить переход. В следующий раз при клике на эту кнопку произойдет переход на выбранный экран. Пример генерации интерактивного экрана по низкодетализированному шаблону изображен на рисунке 13.

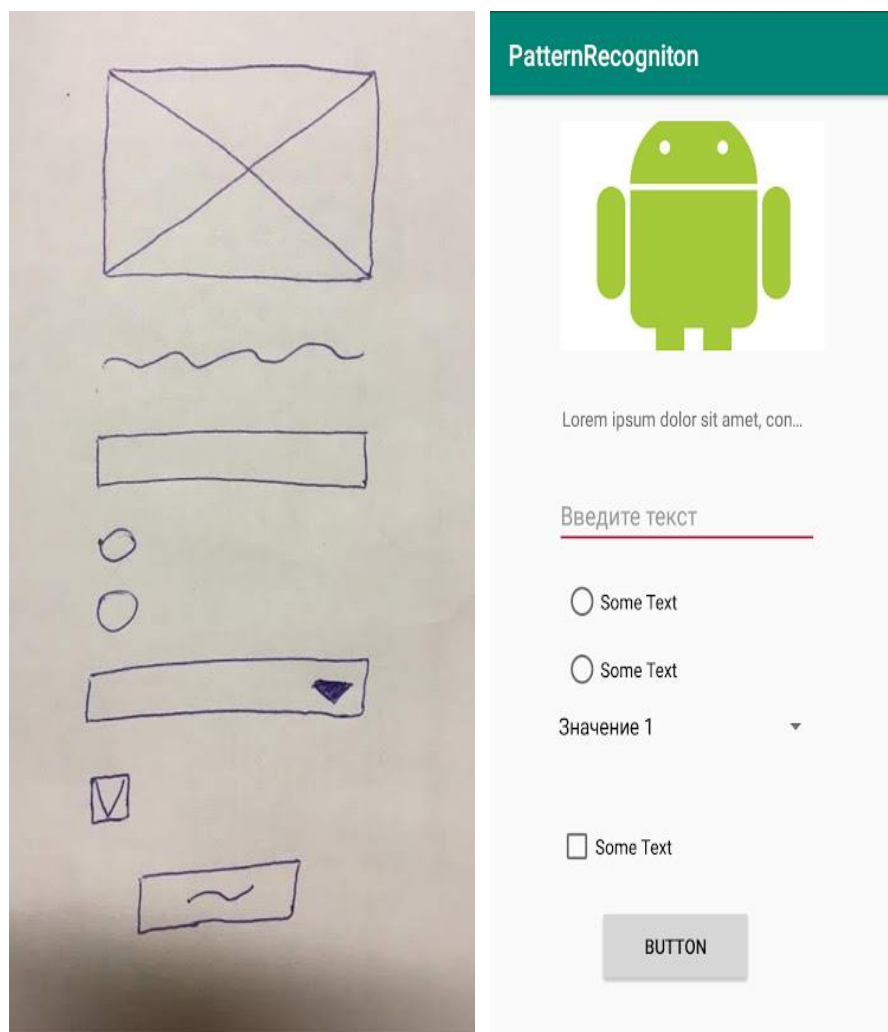


Рис. 13. Низкодетализированный прототип и сгенерированный на его основе экран

## **ЗАКЛЮЧЕНИЕ**

Описан разработанный инструмент для автоматической генерации интерактивных прототипов на основе низкодетализированных макетов для устройств под управлением операционной системы Android. Разработанное мобильное приложение дает пользователю возможность генерации интерактивного интерфейса приложения по нарисованному шаблону.

Разработанный инструмент позволяет сократить временные затраты на создание интерактивного прототипа, тем самым обеспечивая получение детальной обратной связи по прототипу на начальном этапе разработки.

## **СПИСОК ЛИТЕРАТУРЫ**

---

1. *Vithanti T., Kumar A.* Modeling the Mobile Application Development Lifecycle // International MultiConference of Engineers and Computer Scientists. 2014. V. 1. P. 596-600.
2. *Harleen K.F., Xiaofeng W., Swati V.C.* An Investigation into Mobile Application Development Processes: Challenges and Best Practices // I.J. Modern Education and Computer Science. 2014. V. 6. P. 1–9.
3. *Why Prototyping is Essential to Your Design Process.* URL: <https://www.webfx.com/blog/web-design/prototyping-is-essential/>.
4. *Kamushken R.* The advantages of interactive prototyping // Medium. 2017. URL: <https://medium.com/@kamushken/the-advantages-of-interactive-prototyping-855203728b83>.
5. *Babich N.* The Magic of Paper Prototyping // UX Planet. 2018. URL: <https://uxplanet.org/the-magic-of-paper-prototyping-51693eac6bc3>
6. *Morson S.* Using Wireframes to Design Your App // Learn Design for iOS Development. 2014. P. 69–84.

## AUTOMATION OF ANDROID APPLICATIONS INTERACTIVE PROTOTYPES DEVELOPMENT BASED ON LOW-FIDELITY WIREFRAMES

A. S. Khlopunov<sup>1</sup>, I. S. Shakhova<sup>2</sup>

*Higher School of Information Technologies and Intelligent Systems at Kazan Federal University*

<sup>1</sup>hlopunov598@gmail.com, <sup>2</sup>is@it.kfu.ru

### **Abstract**

Some mechanisms for automation of Android applications interactive prototypes development based on handwritten wireframes are described in the paper. The process of automation includes machine learning methods used for the handwritten wireframes recognition. The mobile Android application is developed to ensure user interaction with these mechanisms.

**Keywords:** *prototyping, UI, UX, mobile applications, user interface*

### **REFERENCES**

1. Vithanti T., Kumar A. Modeling the Mobile Application Development Lifecycle // International MultiConference of Engineers and Computer Scientists. 2014. V. 1. P. 596–600.
2. Harleen K.F., Xiaofeng W., Swati V.C. An Investigation into Mobile Application Development Processes: Challenges and Best Practices // I.J. Modern Education and Computer Science. 2014. V. 6. P. 1–9.
3. *Why Prototyping is Essential to Your Design Process.* URL: <https://www.webfx.com/blog/web-design/prototyping-is-essential/>.
4. Kamushken R. The advantages of interactive prototyping // Medium. 2017. URL: <https://medium.com/@kamushken/the-advantages-of-interactive-prototyping-855203728b83>.
5. Babich N. The Magic of Paper Prototyping // UX Planet. 2018. URL: <https://uxplanet.org/the-magic-of-paper-prototyping-51693eac6bc3>
6. Morson S. Using Wireframes to Design Your App // Learn Design for iOS Development. 2014. P. 69–84

## СВЕДЕНИЯ ОБ АВТОРАХ



**ХЛОПУНОВ Анатолий Сергеевич** – студент Высшей школы информационных технологий и интеллектуальных систем Казанского федерального университета, Android-разработчик.

**Anatoly Sergeevich KHLOPUNOV** – student of the Higher School of Information Technologies and Intelligent Systems of Kazan Federal University, developer of software for Android.

email: hlopunov598@gmail.com



**ШАХОВА Ирина Сергеевна** – ассистент кафедры программной инженерии Высшей школы информационных технологий и интеллектуальных систем Казанского федерального университета. Сфера научных интересов – мобильные приложения, цифровые образовательные системы, индивидуализация образования, мобильное обучение.

**Irina Sergeevna SHAKHOVA** – teacher of the Higher School of Information Technologies and Intelligent Systems, Kazan Federal University. Research interests include mobile applications, digital educational systems, individualization in education, mobile learning.

email: is@it.kfu.ru

*Материал поступил в редакцию 23 июня 2019 года*