

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
профессионального образования  
Национальный исследовательский университет «МИЭТ»

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной работе по теме:

Разработка мобильного приложения сопровождения учебного процесса студентов в  
системе ОРИОКС  
(Шифр — МП СУПС)

Руководитель проекта

\_\_\_\_\_ Е.Л. Федотова

Исполнитель

\_\_\_\_\_ О.А. Фаткуллин

Москва 2018

## РЕФЕРАТ

Ключевые слова: разработка мобильного приложения, разработка для Android, язык программирования Kotlin, принципы проектирования SOLID, чистая архитектура, Material дизайн.

Отчет содержит 16 стр., 2 источника, 2 прил.

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| Введение .....                                      | 7  |
| 1 Исследовательский раздел.....                     | 9  |
| 1.1 Почему мобильное приложение? .....              | 9  |
| 1.2 Обзор аналогичных решений.....                  | 9  |
| 1.2.1 Сайт ОРИОКС .....                             | 10 |
| 1.2.2 Приложение “Расписание для МИЭТ” .....        | 10 |
| 1.2.3 Приложение “ОРИОКС Live” .....                | 10 |
| 1.2.4 Телеграм-бот “Open Orioks”.....               | 10 |
| 1.3 Обзор популярных мобильных платформ .....       | 10 |
| 1.4 Исследование структуры ОРИОКС .....             | 10 |
| 1.5 Входные и выходные данные .....                 | 10 |
| 1.6 Постановка задачи .....                         | 10 |
| 2 Конструкторский раздел.....                       | 11 |
| 2.1 Выбор языка программирования.....               | 11 |
| 2.2 Выбор среды разработки .....                    | 11 |
| 2.3 Выбор стека технологий .....                    | 11 |
| 2.4 Архитектура и алгоритм работы МП СУПС .....     | 11 |
| 2.5 Разработка пользовательского интерфейса .....   | 11 |
| 3 Технологический раздел.....                       | 12 |
| 3.1 Разработка под Android .....                    | 12 |
| 3.1.1 Структура проекта.....                        | 12 |
| 3.1.2 Разработка пользовательского интерфейса ..... | 12 |
| 3.1.3 Работа с данными .....                        | 12 |
| 3.1.4 Безопасность.....                             | 12 |
| 3.1.5 Многопоточность и работа с сетью .....        | 12 |

|  |    |
|--|----|
| 3.2 Сборка приложения .....                              | 12 |
| 3.2.1 Система автоматической сборки Gradle .....         | 12 |
| 3.2.2 Подключение зависимостей .....                     | 12 |
| 3.2.3 Подпись приложения .....                           | 12 |
| 3.2.4 Публикация приложения .....                        | 12 |
| 3.2.5 Непрерывная интеграция .....                       | 12 |
| 3.3 Тестирование и отладка в Android разработке .....    | 12 |
| 3.3.1 Методы тестирования программного обеспечения ..... | 12 |
| 3.3.2 Методология разработки TDD .....                   | 12 |
| 3.3.3 Тестирование пользовательского интерфейса .....    | 12 |
| 3.3.4 Отладка в IntelliJ IDEA .....                      | 12 |
| Заключение .....   | 13 |
| Список использованных источников .....                   | 14 |
| Приложение А Текст программы .....                       | 15 |
| Приложение Б Руководство программиста .....              | 16 |

## ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины с соответствующими определениями.

Push-уведомления — способ рассылки уведомлений. У конечного пользователя такие уведомления выглядят как небольшое всплывающее окошко на экране телефона или в браузере.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает — веб-серверю.

Кэширование — один из способов оптимизации приложений. Заключается в том, чтобы сохранять на некоторое время и переиспользовать результаты медленных операций..

Мобильное приложение — программное приложение, предназначенное для работы на смартфонах или планшетных компьютерах.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ОРИОКС — организация распределенного информационного обмена в корпоративных средах.

ПК — персональный компьютер.

API — application programming interface — внешний интерфейс взаимодействия с приложением.

БД — база данных.

HTML — hypertext markup language — язык гипертекстовой разметки.

## ВВЕДЕНИЕ

В современных учебных заведениях всё чаще практикуют использование систем сопровождения учебного процесса, которые позволяют учащимся видеть свои оценки, домашние задания, узнавать о событиях учебного процесса. В МИЭТе такой системой является ОРИОКС. Рассмотрим существующие недостатки ОРИОКСа.

Слабым местом ОРИОКСа является невозможность оперативного оповещения учащихся о событиях учебного процесса. Это обусловлено тем, что ОРИОКС представляет из себя веб-приложение и подразумевает регулярное посещение через браузер. Если же студент долгое время не заходит в систему, что бывает довольно часто, то он может пропустить информацию о важных событиях, например о пересдачах. Так же, в ОРИОКС невозможна установка напоминаний о предстоящих событиях.

Еще одна проблема веб-приложений – неудобство использования на мобильных устройствах. Большая часть студентов посещает ОРИОКС именно с мобильных устройств, и, хотя, дизайн сайта оптимизирован для работы на небольших экранах, всё же мобильные устройства накладывают свои ограничения. Скорость интернет соединения, чаще всего, ниже чем на стационарных ПК, кроме того, соединение может часто разрываться, что приводит к дискомфорту при использовании веб-приложения, т.к. для каждой страницы браузер должен загрузить, помимо данных, разметку и таблицу стилей.

Все эти недостатки можно устранить, создав мобильное приложение, которое будет получать данные от сервера ОРИОКС через API). Это позволит запрашивать и получать только ту информацию, которая нужна для работы приложения в конкретный момент времени, так как приложению не нужно скачивать таблицу стилей и разметку. В виду малого потребления трафика приложение сможет работать на смартфоне в фоновом режиме и отображать уведомления и напоминания в тот момент когда они только пришли.

Удобство работы с системой ОРИОКС при нестабильном интернет соединении тоже повысится за счёт меньшего объёма пересылаемых данных. Кроме того, приложение может кэшировать данные и использовать их даже при отсутствии интернет соединения.

На данный момент нет ни одного приложения работающего с ОРИОКС через API. Более того, нет ни одного приложения, позволяющего студентам использовать все возможности ОРИОКС. Целью данной работы является создание такого приложения, чтобы повысить оперативность оповещения студентов о событиях учебного процесса.



## 1 Исследовательский раздел

### 1.1 Почему мобильное приложение?

Мобильные приложения могут быть инструментом для быстрой доставки информации, чего нельзя добиться при помощи обычного веб-приложения.

Индустрия мобильных устройств очень быстро развивается. На смену старым устройствам приходят более новые, современные и обладающие большим спектром возможностей. Количество пользователей с каждым годом растёт. Сейчас смартфоны есть почти у всех студентов и они редко с ними расстаются надолго. Конечно, важную роль играет программная составляющая — мобильные приложения. Они существуют совершенно разной направленности:

- развлекательные (игры, музыкальные и видео проигрыватели и т.д.);
- коммуникационные (мессенджеры, навигаторы и т.д.);
- справочные (словари, базы знаний);
- прикладные (все остальные от графического редактора до калькулятора).

Кроме того, популярность мобильных приложений повлекла за собой появление мощных инструментов разработки, большого количества библиотек и фреймворков, что, в свою очередь, сделало разработку приложений быстрой, лёгкой и продуктивной.

### 1.2 Обзор аналогичных решений

Был произведен поиск существующих приложений в Windows Phone Store, Google Play, Apple App Store, на GitHub, и было найдено только 3 аналога. К ним можно, так же, добавить сам сайт ОРИОКС.

Рассмотрим преимущества и недостатки каждого решения по отдельности. В качестве критериев будем брать:

- способ получения данных;
- возможность push-уведомлений;

- возможность просмотра расписания;
- возможность просмотра текущих предметов;
- возможность просмотра успеваемости;
- возможность просмотра списка долгов;
- возможность просмотра списка пересдач;
- наличие графического интерфейса;
- оффлайн доступ.

#### 1.2.1 Сайт ОРИОКС

Сайт ОРИОКС получает информацию напрямую из БД, но конечному пользователю для работы сайта нужно подгрузить таблицы стилей и HTML разметку.

#### 1.2.2 Приложение “Расписание для МИЭТ”

#### 1.2.3 Приложение “ОРИОКС Live”

#### 1.2.4 Телеграм-бот “Open Orioks”

### 1.3 Обзор популярных мобильных платформ

### 1.4 Исследование структуры ОРИОКС

### 1.5 Входные и выходные данные

### 1.6 Постановка задачи

## 2 Конструкторский раздел

### 2.1 Выбор языка программирования

### 2.2 Выбор среды разработки

### 2.3 Выбор стека технологий

### 2.4 Архитектура и алгоритм работы МП СУПС

### 2.5 Разработка пользовательского интерфейса

### 3 Технологический раздел

#### 3.1 Разработка под Android

##### 3.1.1 Структура проекта

##### 3.1.2 Разработка пользовательского интерфейса

##### 3.1.3 Работа с данными

##### 3.1.4 Безопасность

##### 3.1.5 Многопоточность и работа с сетью

#### 3.2 Сборка приложения

##### 3.2.1 Система автоматической сборки Gradle

##### 3.2.2 Подключение зависимостей

##### 3.2.3 Подпись приложения

##### 3.2.4 Публикация приложения

##### 3.2.5 Непрерывная интеграция

#### 3.3 Тестирование и отладка в Android разработке

##### 3.3.1 Методы тестирования программного обеспечения

##### 3.3.2 Методология разработки TDD

##### 3.3.3 Тестирование пользовательского интерфейса

##### 3.3.4 Отладка в IntelliJ IDEA

## ЗАКЛЮЧЕНИЕ

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по подготовке выпускной квалификационной работы по направлению подготовки бакалавров 09.03.04 «Программная инженерия» / Гагарина Л.Г., Касимов Р.А., Коваленко Д.Г. и др. ; Под ред. Б.В. Черникова. — МИЭТ, 2016. — 20 с.

2. Wikipedia. Типографика — Википедия. — 2012. — Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0> (дата обращения: 25.01.2012).

ПРИЛОЖЕНИЕ А

ТЕКСТ ПРОГРАММЫ

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

ПРИЛОЖЕНИЕ Б

РУКОВОДСТВО ПРОГРАММИСТА

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.