**Федеральное государственное образовательное бюджетное учреждение высшего образования**

**«ФинансовЫЙ УНИВЕРСИТЕТ при Правительстве Российской Федерации»**

Факультет информационных технологий и анализа больших данных

Пояснительная записка к курсовой работе

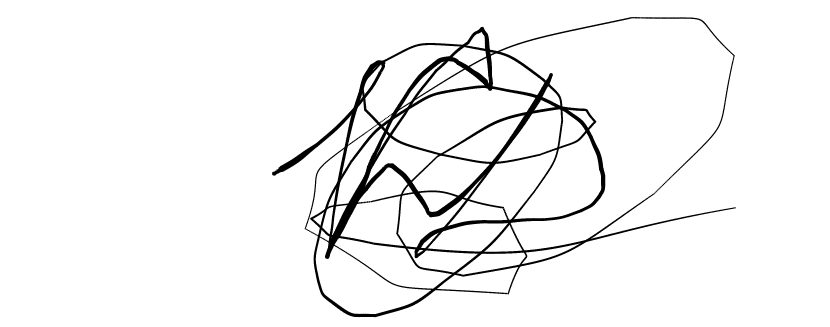
по дисциплине “Технологии разработки приложений для мобильных устройств”

на тему:

«Мобильное приложения для создания шаблонов электронных писем»

Студента группы ПИ20-5

Кравченко Д. Д.



Руководитель:

Доцент

Крахмалев О. Н.

Москва 2022

Оглавление

[**Введение** 2](#_Toc103868934)

[**Основная часть** 4](#_Toc103868935)

[**1.** **Разработка ТЗ** 4](#_Toc103868936)

[**1.1** **Критерии приложения** 4](#_Toc103868937)

[**1.2** **Требования приложения** 5](#_Toc103868938)

[**2.** **Разработка эскизного проекта** 6](#_Toc103868939)

[**2.1** **Структура эскиза** 6](#_Toc103868940)

[**2.2** **Эскиз** 7](#_Toc103868941)

[**3.** **Разработка ПО** 8](#_Toc103868942)

[3.1 **Выбор платформы для реализации** 8](#_Toc103868943)

[**3.2** **Блок-схема** 11](#_Toc103868944)

[3.3 **Алгоритмы** 13](#_Toc103868945)

[**4.** **Тестирование** 16](#_Toc103868946)

[**5.** **Внедрение** 17](#_Toc103868947)

[**5.1** **Инструкция по использованию** 17](#_Toc103868948)

[**Заключение** 26](#_Toc103868949)

[**Список используемых источников или интернет-ресурсов** 27](#_Toc103868950)

# **Введение**

Информационные технологии – это использование компьютеров, для создания, обработки, хранения, извлечения и обмена всеми видами информации.

На сегодняшний день информационные технологии заполнили практически каждый аспект жизни человека. Невозможно даже представить жизнь без ИТ, хотя еще половину века назад никто и понятия не имел о них. Информационные технологии очень успешно закрепились в каждой, не только специализированной деятельности, а во всей жизни человека. В какой-то мере, можно сказать, что ИТ стали ключевым достижением человечества. Ведь благодаря им произошел большой скачок в развитии всего мира.

Первые значимые в истории технологии представляли собой электронные вычислительные машины (ЭВМ). Электронные вычислительные машины впервые появились именно для произведения различных вычислений, что следует даже из названия. А одним из первых достижений раннего этапа развития ЭВМ стала, как раз, передача сообщений, обмен данными, между вычислительными машинами.

С ростом технологий росли и возможности ЭВМ, и вот уже поколение людей 21 века не представляет себе жизнь без различных социальных институтов, построенных в сфере информационных технологий. Огромные объемы информации пользователи получают из сети, а также совершают и огромное количество операций: запись на прием ко врачу, выпуск банковской карты, поиск рецептов блюд, методов лечения ОРЗ, организация досуга, просмотр обучающих видеороликов и, наконец, общений с коллегами по рабочим вопросам, с друзьями и семьей о домашних делах, с арендодателем по вопросам оплаты квартиры и еще бесконечное количество возможных вариантов.

Для нас, в данной работе, интересны технологии, направленные на коммуникацию людей. Инструментом коммуникации послужит электронная почта. Каждый современный человек знает о таком инструменте и имеет свой почтовый ящик. Причиной этому служит высокая востребованность электронной почты. В 1997 году появился первый общедоступный сервис в интернете – “Yahoo! Mail”. [1] Вскоре появились и остальные крупные мировые сервисы электронных почт, которыми мы пользуемся и по сей день. Тогда электронная почта использовалась для личного общения людей друг с другом. Со временем появлялись все более удобные для общения электронные сервисы почты оказалась на втором плане. Но она не исчезла из жизни людей, сегодня, в 77% [7] случаев, человек создает почтовый ящик в рабочих целях. В каждой организации используется электронная почта для связи внутри коллектива. Или же почтовый ящик может понадобиться для регистрации на том или ином сайте в интернете.

Завершив вступительную часть по ознакомлению с электронной почты, вернемся к теме работы. Приложения, или же, веб-сервисы для создания шаблонов электронных писем упрощают людям пользование стандартных сервисов для отправки писем. С помощью таких сервисов можно создавать письма для масштабной маркетинговой рассылки. Речь идет не просто о письмах с темой и телом письма, в виде текста, а о множестве компонентов внутри тела письма, такие как, картинки, видео, кнопки, стилизованные текста и различные ссылки.

Целью моего проекта является создать мобильное приложение для создания шаблонов и отправки писем. Приложение должно создавать письмо, в котором пользователь сам пишет отправителя или отправителя, текст письма и прикрепляет нужный документ, в виде картинки, видео или другого файла, в письмо.

# **Основная часть**

1. **Разработка ТЗ**
   1. **Критерии приложения**

Приложение должно соответствовать следующим критериям:

1. Функциональность
2. Совокупность функций всех задач и целей пользования
3. Производительность
4. Эффективность – связь между достигнутого результата и использованными ресурсами
5. Время загрузки или отклика
6. Совместимость
7. Аппаратная совместимость мобильного приложения
8. Программная совместимость мобильного приложения
9. Удобство пользования
10. Понятность мобильного приложения
11. Изучаемость мобильного приложения – количество усилий пользователя по обучению полноценным пользованием приложением
12. Простота пользования
13. Эстетика пользовательского интерфейса
14. Доступность мобильного приложения
    1. **Требования приложения**

Требования:

1. Пользователь должен заполнять все поля вручную
2. Программа должна осуществлять обработку ошибок, неправильно введенных данных

Нефункциональные требования:

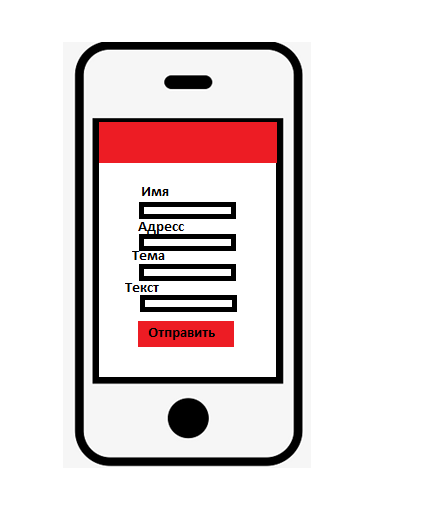
1. Приложение должны быть разработано для 2 мобильных платформ (IOS, Android)
2. Приложение должно работать в онлайн режиме (подключенное к интернету)
3. **Разработка эскизного проекта**
   1. **Структура эскиза**

Приложение состоит только из одного экрана так как, для полного требуемого функционала этого достаточно.

Экран состоит из следующих компонентов:

1. Форма письма
2. Поле ввода своего имени
3. Поле ввода электронного адреса почты получателя
4. Поле ввода темы письма
5. Поле ввода текста в теле письма
6. Кнопка проверяющая правильность введенного Email адреса
7. Кнопка, открывающая сервис (“Gmail”, “Яндекс.Почта”, “Mail.ru”) для отправки письма
   1. **Эскиз**

Набросок того, как должно выглядеть мое приложение:

****

1. **Разработка ПО**
   1. **Выбор платформы для реализации**

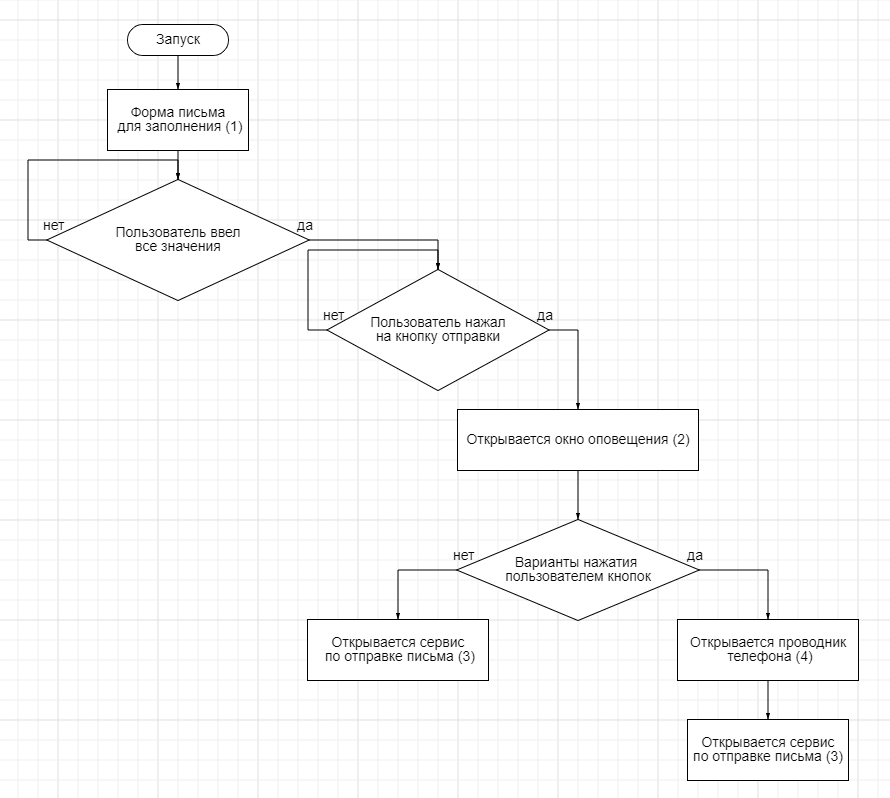
Создание приложения начинается с выбором платформы и языка программирования. Я, в свою очередь, выбрал платформу – React Native [4].

React Native [4] – это популярная платформа мобильных приложений основанная на JavaScript, которая позволяет создавать мобильное приложение с собственным интерфейсом для IOS и Android.

Инструментом разработчика послужит оболочка React Native [4] – Expo [11]. Это набор инструментов, построенных вокруг React Native. Самая распространенная проблема, которую решает Expo [11], заключается в том, что она сама связывает библиотеки в проекте и устраняет возможные проблемы с этим, вместо разработчика.

Выбор этой платформы основывается на большом количестве преимуществ перед другими языками, такими как, Java, C# и т. д. Мобильное приложение на основе React Native [4] не является мобильным, гибридным веб-приложение или HTML5. Это больше похоже на настоящее мобильное приложение. Для более полной картины я приведу значительные преимущества React-native [4]:

1. React Native позволяет разработчикам создавать приложения с использованием web-технологий.
2. React Native позволяет создавать кросс-платформенные приложения, то есть написанный код мобильного приложения поддерживается сразу на двух платформах Android и IOS. Это благодаря тому, что React-native использует JavaScript компоненты, которые построенные сразу на двух мобильных платформ.
3. React Native использует фундаментальные строительные блоки Android и IOS для компиляции собственных приложений для обеих платформ в JavaScript. Это облегчает обработку кодовой базы, а также упрощает добавление новых функций даже во время работы приложения.
4. React Native позволяет создавать приложения быстрее. Вместо перекомпиляции вы можете мгновенно перезагрузить приложение. Используя функцию горячей перезагрузки, вы даже можете запускать новые функции, сохраняя при этом состояние приложения.
5. В React Native мы можем легко разрабатывать и тестировать функции, используя различные библиотеки и инструменты, такие как Expo [11], ESLint, Jest и Redux.
6. React Native обеспечивает хорошее сообщество разработчиков.
   1. **Блок-схема**



Комментарии к блок-схеме:

1. Форма, состоящая из полей ввода (имя, адрес почты получателя, тема письма, тело письма)
2. Окно оповещения, состоит из двух кнопок “Да” и “Нет”, в котором пользователя спрашивают, хочет ли он прикрепить вложение к письму, в виде файла, картинки или видео
   1. Кнопка “Да” – после нажатия открывается проводник телефона, в котором пользователь выбирает вложение и после чего открывается сервис по отправке письма
   2. Кнопка “Нет” – после нажатия сразу открывается сервис по отправке письма
3. Сервис по отправке письма это другое приложение на устройстве пользователя, например Gmail
4. Проводник телефона, в котором пользователь выбирает вложение к письму, которое он хочет прикрепить
   1. **Алгоритмы**

В моем приложении используется алгоритм по созданию письма. В вышеописанной блок-схеме я показал последовательность действий. Сейчас я опишу как именно создается письмо в моем приложении.

После того, как пользователь заполнил все поля и нажал на кнопку, вызывается функция, которая, в свою очередь, вызывает окно оповещения.

**const** showAlert = () =>

Alert.**alert**(

"Прикрепить вложение",

"Хотите ли вы прикрепить вложение?",

[

{

text: "Нет",

onPress: () => {sendEmail([])},

style: "cancel"

},

{ text: "Да", onPress: sendEmailWithAttachment }

]

);

Каждая кнопка вызывает другую функцию, в зависимости от значения.

Кнопка “Нет” вызывает функция отправки без вложения:

**const** sendEmail = async(file) => {

var options = {}

**if**(file.length < 1){

options = {

        subject: subject,

        recipients: [email],

        body: message,

}

}**else**{

options = {

        subject: subject,

        recipients: [email],

        body: message,

        attachments: file

}

}

let promise = new Promise((resolve, reject) => ***{***

MailComposer.composeAsync(options)

.then((result) => ***{***

resolve(result)

***}***)

.catch((error) => ***{***

reject(error)

***}***)

***}***)

Здесь создается переменная “options”, которая включает в себя все значения из полей ввода. И далее вызывается функция MailComposer.composeAsync [8] библиотеки expo-mail-composer. Эта библиотека позволяет быстро и легко создавать, и отправлять электронные письма с помощью пользовательского интерфейса операционной системы.

Кнопка “Да” в окне оповещения вызывает следующую функцию:

**const** sendEmailWithAttachment = async() => {

let result = await ImagePicker.launchImageLibraryAsync(***{***

mediaTypes: ImagePicker.MediaTypeOptions.All,

allowsEditing: false,

aspect: [4, 3],

quality: 1,

***}***);

**if** (!result.cancelled) {

console.log(result.uri)

sendEmail([result.uri]);

}**else**{

sendEmail([])

}

}

Здесь задаются параметры вставки вложения и вызывается та же функция sendEmail с заданным параметром. Переменная result представляет собой функцию библиотеки expo-image-picker [9]. Данная библиотека предоставляет доступ к пользовательскому интерфейсу системы для выбора изображений, видео из библиотеки телефона или фотосъемки с помощью камеры.

Так же существует функция для проверки правильности написания электронной почты. Она проверяет значение, которое ввел пользователь на символы, которые должны присутствовать в адресе почты.

**const** validateEmail = (e) => {

**if** (validator.isEmail(email)) {

Alert.**alert**('Email is valid')

console.**log**("Email is valid")

} **else** {

Alert.**alert**('Email is not valid')

console.**log**("Email is not valid")

}

}

Для этого используется специальная библиотека validator [10]. Она предназначена на то, чтобы проверять текс на существующий адрес электронной почты. Для пользователя я создал кнопку, нажав на которую, пользователю сообщится существует ли введенная им почта или нет.

1. **Тестирование**

Я протестировал свое приложения по следующим пунктам:

1. Функциональное тестирование

Данное тестирование проверяет, что приложение соответствует нужной функциональной спецификации.

1. Запуск приложения
2. Работоспособность основного функционала приложения
3. Корректное отображение ошибок
4. Работа с файлами
5. Тестирование pop-up, алертов
6. Тестирование удобства пользования

Данное тестирование помогает удостоверится насколько легко и эффективно пользователь может использовать приложение.

1. Корректное отображение элементов на устройствах с различными разрешениями экранов
2. Правильно выравнивание текстов
3. Сообщения об ошибках без грамматических и орфографических ошибок
4. Пиксель-перфект

1. Тестирование производительности

Тестирование производительности помогает понять, насколько приложение потребляет время и ресурсы для функционирования.

1. Время загрузки приложения
2. Обработка запросов
3. Потребление ресурсов приложением
4. **Внедрение**
   1. **Инструкция по использованию**

Для пользования приложением необходимо мобильное приложение Expo [11], скачать его можно в App Store или Play Market. Чтобы открыть приложение на телефоне, следует открыть QR-код в проекте и навести на него камеру телефона (IOS) или открыть на телефоне приложение Expo [11] и отсканировать QR-код через него (Android). При запуске приложения будет отображаться следующие:

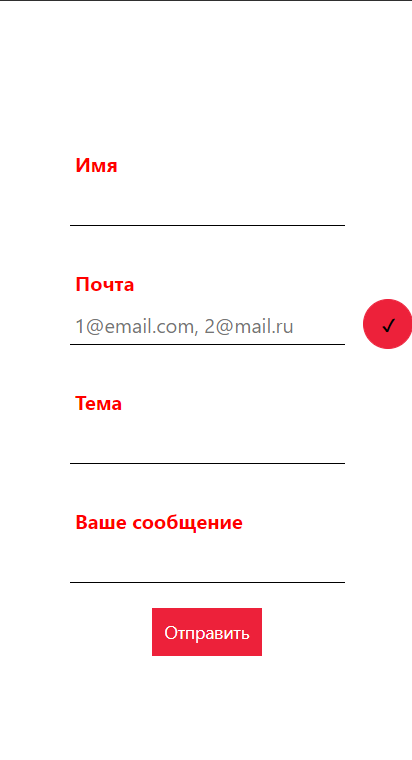


Рис 1. Экран приложения

Пример работы кнопки проверки адреса почты:



Рис 2. В поле ввода введены значения для примера

Изображение выглядит как текст

Автоматически созданное описание

Рис 3. Нажата кнопка проверки адреса почты

Изображение выглядит как текст

Автоматически созданное описание

Рис 4. Введено правильно значение в поле ввода

Изображение выглядит как текст

Автоматически созданное описание

Рис 5. Нажата кнопка проверки адреса почты

Далее я покажу отправку письма:

Изображение выглядит как текст

Автоматически созданное описание

Рис 6. Заполненная форма письма

Изображение выглядит как текст

Автоматически созданное описание

Рис. 7 Всплывающее окно

На рис. 7 показано всплывающее окно после нажатия кнопки “Отправить”.

Если нажать “Да”, то появится проводник телефона с выбором вложения, после выбора откроется сервис электронной почты (рис. 9):

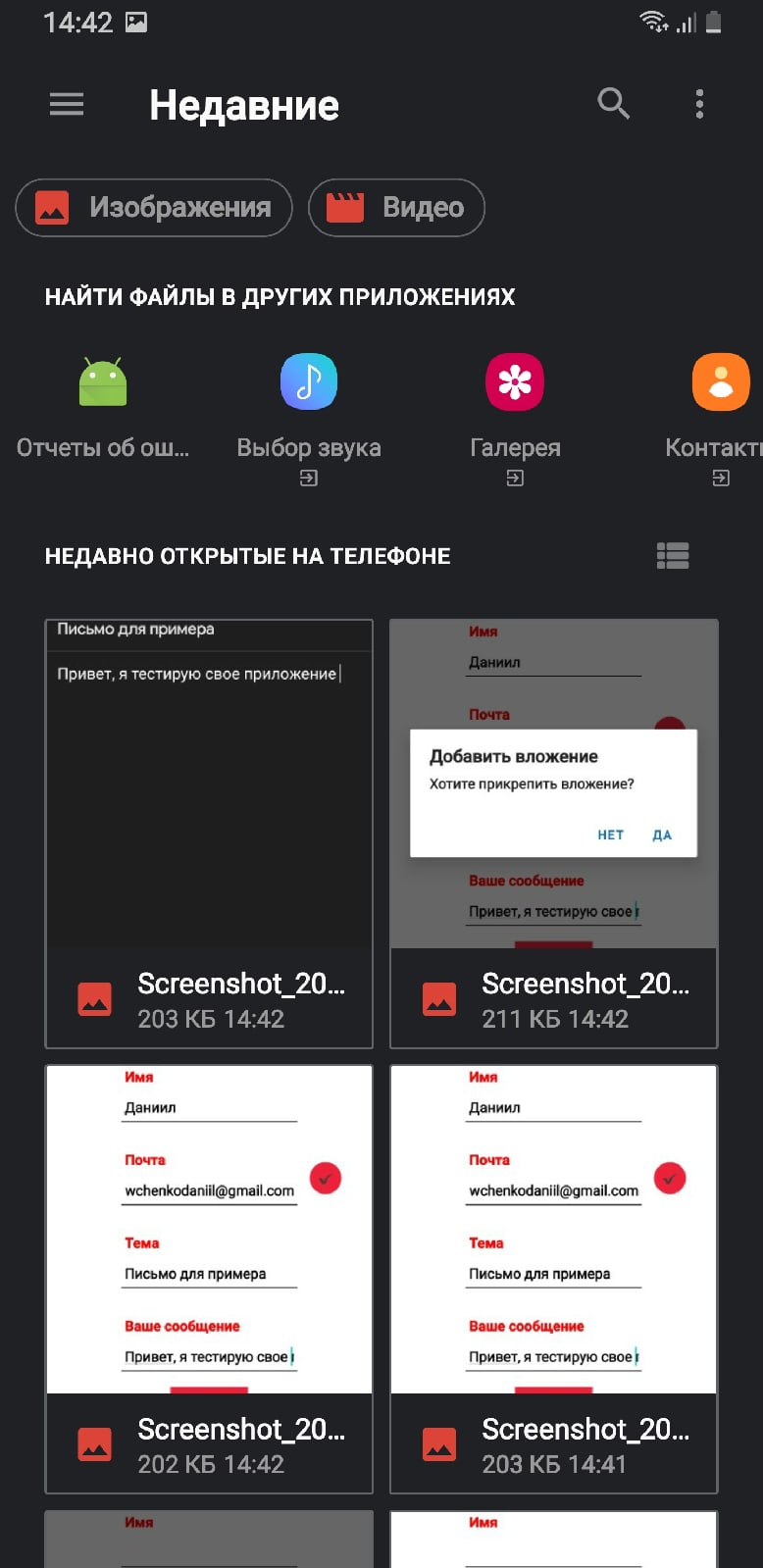


Рис. 8 Выбор вложения

Если же нажать на кнопку “Нет”, то откроется сервис электронной почты:

Изображение выглядит как текст

Автоматически созданное описание

Рис. 9 Отправка письма

# **Заключение**

В наше время сервис по созданию шаблонов письма может быть очень полезен для упрощения работы с рассылками. В рамках моей работы было разработано приложение для IOS и Android. Были достигнуты следующие задачи:

1. Разработано ТЗ, поставлены требования и критерии
2. Разработан эскиз приложения
3. Обоснован выбор платформы, описаны алгоритмы работы приложения
4. Проведено тестирование приложения
5. Описана инструкция по пользованию приложения

Задачи, которые были поставлены, достигнуты и реализованы. В моем видении приложение не на конечном варианте реализации, для полной полноты картины могут быть следующие добавления:

1. Сохранение каждого шаблона на отдельном экране
2. Авторизация и регистрация пользователей

# **Список используемых источников или интернет-ресурсов**

1. Информационные технологии. [Электронный ресурс] URL: <https://translated.turbopages.org/proxy_u/en-ru.ru.8c81ef62-628635ac-51de3cd3-74722d776562/https/en.wikipedia.org/wiki/Information_technology>
2. Что такое React Native? Комплексное руководство 2021. [Электронный ресурс] URL: <https://habr.com/ru/post/596183/>
3. Российская система качества мобильные приложения для смартфонов. [Электронный ресурс] URL: <https://d-russia.ru/wp-content/uploads/2017/04/Roskachestvo_mob_app_3-04-2017.pdf>
4. React Native: плюсы и минусы фреймворка в 2022 году. UPD. [Электронный ресурс] URL: <https://www.purrweb.com/ru/blog/react-native-plyusy-i-minusy/>
5. Why You Should Choose React Native? [Электронный ресурс] URL: <https://www.geeksforgeeks.org/why-you-should-choose-react-native/>
6. Чек-лист тестирования мобильных приложений. [Электронный ресурс] URL: <https://habr.com/ru/post/534190/>
7. Как люди используют электронную почту в 2018 году — статистика. [Электронный ресурс] URL: <https://emailsoldiers.ru/blog/email-in-2018-research>
8. MailComposer – Expo documentation. [Электронный ресурс] URL: <https://docs.expo.dev/versions/latest/sdk/mail-composer/>
9. ImagePicker - Expo documentation. [Электронный ресурс] URL: <https://docs.expo.dev/versions/latest/sdk/imagepicker/>
10. React native form validator. [Электронный ресурс] URL: <https://www.npmjs.com/package/react-native-form-validator>
11. Платформа для разработки мобильных приложений – Expo. [Электронный ресурс] URL: <https://expo.dev/>