|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования  **«Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (УрФУ)** |

Институт *Фундаментального образования (ИнФО)*

кафедра *интеллектуальных информационных технологий*

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель курсовой работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ**

**о курсовой работе**

по теме: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент:

Фамилия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Имя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Отчество \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Екатеринбург

2018

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc532918688)

[1 Цели 5](#_Toc532918689)

[2 Обзор внешнего рынка 6](#_Toc532918690)

[3 Внутренний обзор 7](#_Toc532918691)

[4 Описание реализации продукта 8](#_Toc532918692)

[ЗАКЛЮЧЕНИЕ 13](#_Toc532918693)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 15](#_Toc532918694)

# ВВЕДЕНИЕ

На данный момент у жилищно-коммунальных служб низкий уровень автоматизации системы. Заявки, полученные от жильцов, на устранение тех или иных неисправностей сканируются и хранятся в виде изображений. У жильцов есть несколько вариантов связаться с управляющей компанией: прийти самостоятельно, написать письмо на почту или позвонить в аварийно-диспетчерскую службу. Основной интерес для управляющих компаний составляет сокращение штата сотрудников и контролирование работников. До посещения управляющей компании мы предполагали сделать мобильное приложение для подачи заявок, однако после того, как выяснилось, что у управляющей компании нет ни базы данных заявок, никакой-либо другой автоматизированной системы, мы приняли решение двигаться в сторону интересов компании. Нами было выяснено, что необходимо для повышения производительности. Мы решили поменять концепцию нашего приложения и расширить функционал. Новой концепцией стало приложение не для жильцов, а для мастеров, электриков и других работников по устранению неполадок. Дополнением для приложения является сайт для диспетчеров по добавлению заявок, после добавления заявка отображается у мастера, которому она была отправлена, в приложении, мастер, находясь в любом месте может среагировать на заявку и отправиться на место выполнения. После устранения всех неисправностей мастер подтверждает выполнение работы, и заявка становится завершенной. Эту мысль мы развили дальше и, поскольку многие управляющие компании сейчас используют программное обеспечение 1С, то хорошим решением было бы использование нашего продукта как модуля для данной программы. Это бы помогло не перестраивать резко на наш продукт систему, а сделать это аккуратнее и безопаснее, поскольку тогда мы сможем сохранить действующую базу заявок. Помимо этого, также мы не полностью отказались от идеи приложения для жильцов, напротив, мы решили в будущем дополнить приложение до системы авторизации. Однако это оказалась слишком большая цель, поэтому на время осеннего семестра мы решили ограничиться только приложением для мастеров и сайтом для диспетчеров.

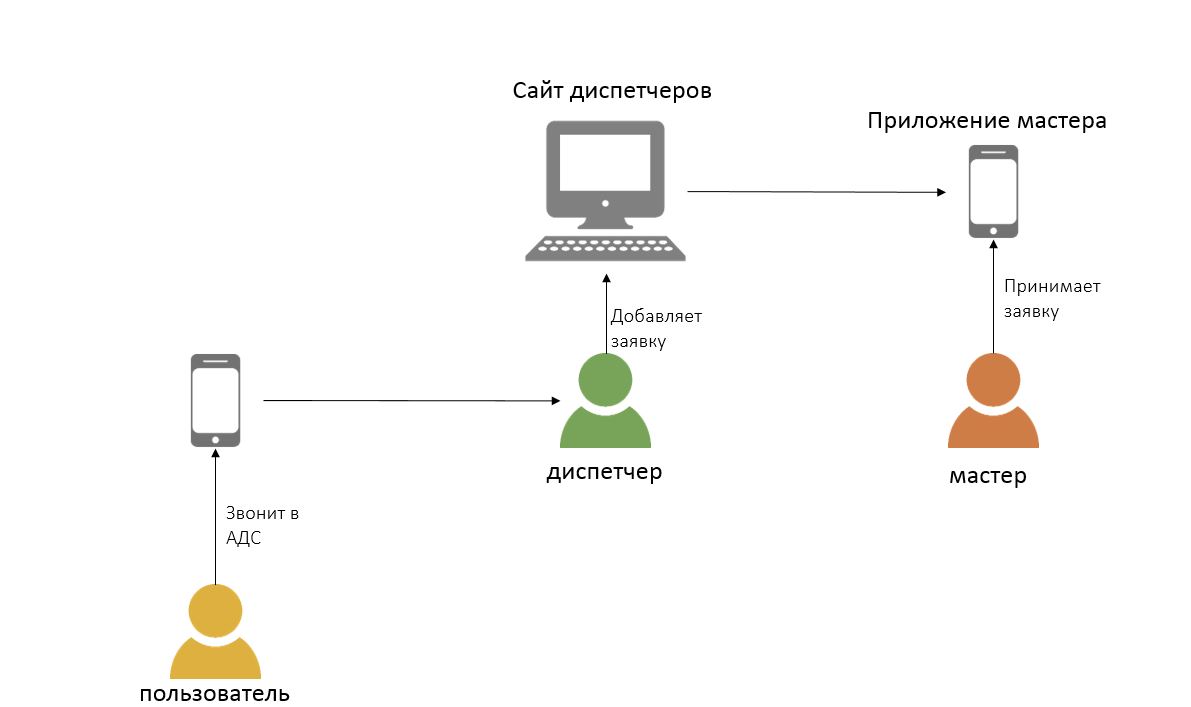


Рисунок 1 – Концепция приложения

# 1 Цели

Важной задачей любого проекта является выявление цели, к которой необходимо стремиться и задач, которые необходимо выполнять для достижения цели.

Цель: реализация мобильного приложения для мастеров и создание сайта для диспетчеров, а также обеспечение взаимодействия между ними.

Заданная цель оказалась достаточно велика и обширна, поскольку стояла задача не только реализация отдельного продукта, а сразу нескольких, отсюда использование нескольких языков, сред разработки и изучение с ранее незнакомыми технологиями. Мы уже имели опыт разработки мобильных приложений, поэтому представляли в каком направлении двигаться, также у некоторых членов команды были знания в языке PHP, который необходим для создания сайта и API для него. Однако никто из нас не сталкивался с реализацией API.

Задачи:

* 1. создание макетов приложения;
  2. создание макета сайта;
  3. проектирование БД;
  4. верстка экранов приложения;
  5. реализация логики приложения;
  6. верстка сайта;
  7. создание базы данных;
  8. реализация логики на сайте;
  9. написание API;
  10. подключение API к приложению;
  11. подключение API к сайту.

# 2 Обзор внешнего рынка

Основными нашими конкурентами на рынке являются «Квадо.ру», «Квартировод» и «1up». Все эти компании работают в других округах, однако это не единственное преимущество, которое мы имеем. Наше главное отличие от ни в том, что мы хотим сделать модуль для программного обеспечения 1С. Данным ПО сейчас пользуется большинство управляющих компаний. Мы позволим им не отказываться от того, что они используют, а доработать это. При посещении управляющей компании мы увидели, что там используется старое обеспечение, многим из них будет сложно перейти на что-то новое, однако они не против внедрить что-то новое постепенно. Именно это мы и хотим предложить. Также мы хотим сделать упор на контроль мастеров, что на данный момент мы не видели у других продуктов. Эта особенность позволит отслеживать нахождение работников и ход выполнения их работы. Диспетчера смогу из своего аккаунта на сайте видеть, где находится в данный момент мастер. Это упростит взаимодействие между работниками, а также позволит диспетчеру сообщить жильцу при запросе, когда прибудет мастер.

# 3 Внутренний обзор

Приложение, было принято решение, делать под платформу Android, поскольку на большинстве устройств именно она является операционной системой. Кроме того, аккаунт разработчика Google Play стоит 25$, а у AppStore стоимость составляет 99$ в месяц [1]. Для реализации мобильного приложения Android существует множество фреймворков: Xamarin, Corona SDK, Phonegap, Basic4Android, Mo Sync SDK и многие другие [4]. С Xamarin уже были знакомы некоторые члены команды, поэтому нашем выбором изначально был он, но так как Android Studio, одна из сред разработки, гораздо удобнее и быстрее для разработки на Android, мы решили остановиться на ней.

Так как некоторые члены команды уже работали на языке программирования PHP, то сайт мы решила делать на нем. Кроме PHP для создания сайтов существует множество других языков программирования таких, как python, ruby, perl и другие [2]. PHP нам подходит, поскольку он прост в понимании и является объектно-ориентированным. Кроме того, он отлично взаимодействует с СУБД MySQL, а нам важно было использование данной СУБД, поскольку она достаточно быстрая, чтобы работать с большим объемом информации [7,15]. Выбор языка для написания апи не занял слишком много времени, поскольку лучшим вариантом, как мы решили, является использование того же языка, который мы используем для реализации сайта, а именно PHP. Помимо него можно было также использовать и другие языки, такие как python, java, C# и другие [14]. Но для нас важна работа с MySQL, поэтому лучшим выбором для нас стал php.

# 4 Описание реализации продукта

4.1 Средства

Написание приложения производилось на языке kotlin [12], он многофункциональный и объектно-ориентированный. Средой разработки является Android Studio. Реализация сайта была выполнена на языке php с использованием системы управления баз данных mysql. Для обеспечения взаимодействия базы данных, сайта и приложения было написано апи на языке программирования php [4].

4.2 Особенности реализации проекта

Перед началом работы над приложением необходимо продумать сценарий работы и создать макеты по этим сценариям. Необходимо было сделать такой дизайн, который был бы интуитивно понятен пользователю и удобен в использовании. Так как в начале работы мы хотели использовать фреймворк Xamarin, а также фреймворк EnCore для быстрой и кроссплатформенной разработки приложения, однако наша цель оказалась обширнее, чем предполагалось, поэтому мы решили ограничится одной платформой Android. В связи этим, мы решили сменить среду разработки с Visual Studio на Android Studio. Сменилась не только среда, сменился язык, а, следовательно, всю логику пришлось переписать. Логики было реализовано не много, в основном было реализовано много экранов. И в Xamarin.Android, и в Android Studio используется один формат для верстки экранов, это xml-файлы, поэтому перенести их не составило труда [16]. Основное время разработки приложения ушло на реализацию логики такой, как пролистывание заявок, боковое меню и добавление данных [13].

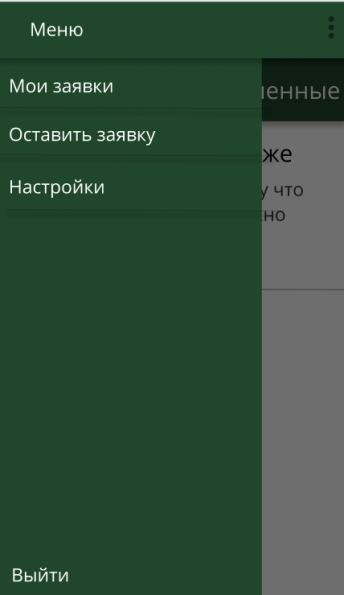
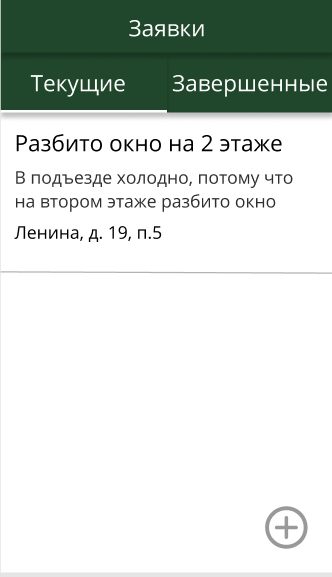
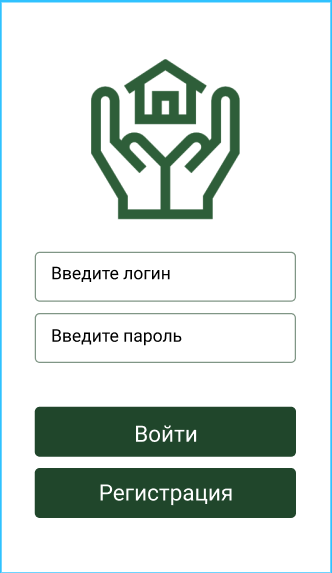


Рисунок 2 –Макет приложения

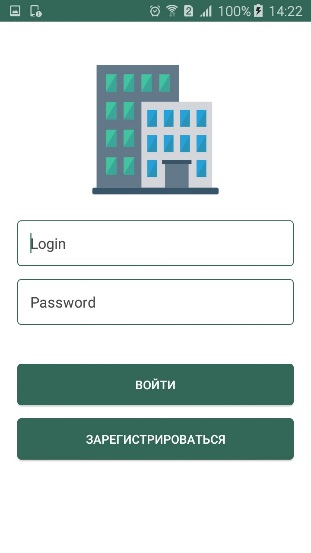
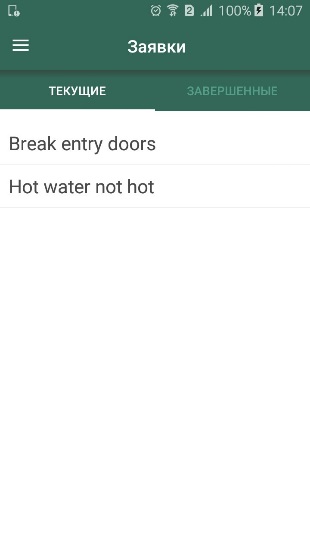
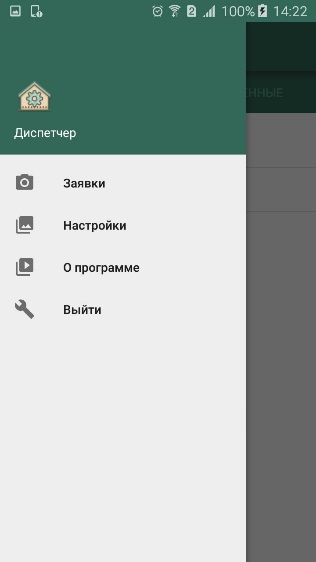
  

Рисунок 3 – Приложение

Как можно заметить по рисунку 2 и рисунку 3, изменился дизайн, цветовая гамма. Это связано с тем, что в команде мы не могли прийти к консенсусу и определиться с тем, как должно выглядеть приложение. Однако финальная версия всем понравилась.

Параллельно с созданием приложения, осуществлялось написание апи. Апи необходимо для взаимодействия базы данных и приложения/сайта. С базой данных MySQL мы связываемся с помощью расширения PDO, поскольку он имеет ряд преимуществ по сравнению с MySQLi. PDO безопаснее, проще в использовании, занимает меньше места в коде, а главное – позволяет выгружать данные из базы в объекты [6].

Для проверки http запросов, нам необходимо было создать базу данных, но перед созданием необходимо ее спроектировать. При ее проектировании мы учли не только мастеров, но и диспетчеров. В нашей бд есть таблица users, содержащая информацию о пользователях такую, как имя, логин, хешированный пароль и должность. Есть таблица post, хранящая список должностей таких, как диспетчер, мастер, жилец и администратор. Таблица requests содержит данные о заявках: название, описание, идентификатор пользователя, отправившего заявку, идентификатор статуса заявки. Для статусов есть отдельная таблица под названием status, она хранит названия статусов: одобрена, принята, завершена. Кроме того, есть также таблица, хранящая фотографии заявки по ее идентификатору. База спроектирована с заделом на будущее, чтобы ее можно было дополнять и совершенствовать, поскольку мы планируем добавить возможность жильцам отправлять заявки через приложение [9].

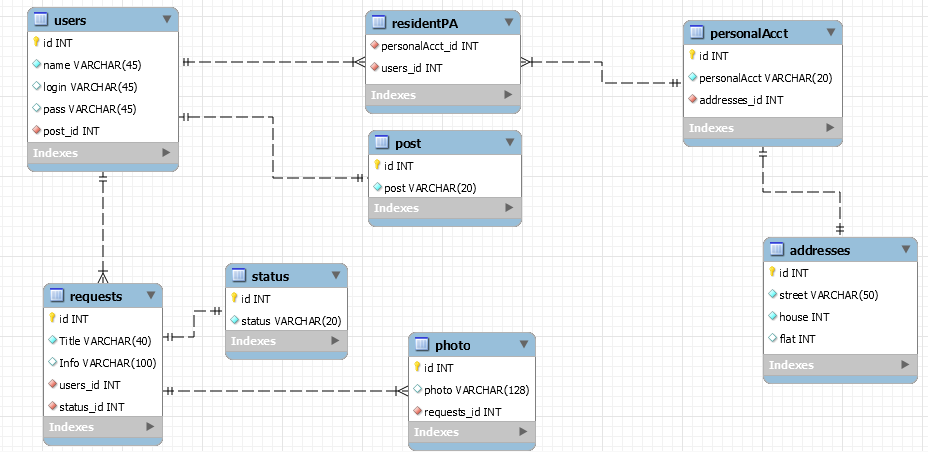


Рисунок 4 – Схема реляционной базы данных

После проверки апи на работоспособность нам предстояло обеспечить отправку запроса через приложение. Проблемы возникали в выборе инструмента для осуществления запросов. В сети много вариантов на решение этой задачи. Одним из них была библиотека apache, однако с последней версией Android Studio пропала возможность ее использования. Другим была библиотека OkHTTP, с помощью которой получилось выполнить простейшие запросы, однако при попытке выгрузки заявок возникла проблема с работой потоков, хотя данная библиотека предназначена для снятия груза работы над потоками с разработчика. Снова углубившись в тему поиска надежного инструмента для осуществления запросов, совершенно случайно был найден встроенный в kotlin класс HttpURLConnection [6]. Мы создали отдельный асинхронный класс для get и post запросов с принятием url и других данных для post запроса.

Сайт был сверстан с помощью фреймворка bootstrap, он предоставляет выбор из современных и лаконичных элементов [7, 15]. Сайт также работает с апи, получая через него информацию о пользователях и заявках.

4.3 График работ

В таблице 1 и таблице 2 представлен ход выполнения работ и распределение задач между участниками проекта.

Таблица 1 – График выполнения работ

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование этапа | Даты | Результат |
| 1 | Продумывание концепции | 12.10.2018-14.10.2018 | Сформирована идея, продумано поведение приложения |
| 2 | Создание макетов | 15.10.2018-22.10.2018 | Созданы макеты приложения |
| 3 | Встреча с представителями управляющей компании | 23.10.2018 | Обсуждена концепция приложения, выявлены узкие места управляющей компании |
| 4 | Переосмысление идеи и переделывание макетов | 24.10.2018-29.10.2018 | Идея приложения поменялась. Макеты переделаны исходя из новых потребностей |
| 5 | Проектирование бд и ее реализация | 30.10.2018-04.11.2018 | Спроектирована база данных и реализована |
| 6 | Разработка мобильного приложения | 05.11.2018-29.11.2018 | Разработан основной функционал приложения, созданы экраны |
| 7 | Написание API | 30.11.2018-12.11.2018 | Написано апи для приложения и сайта |
| 8 | Подключения апи к мобильному приложению | 13.11.2018-16.11.2018 | Теперь в приложении появляются данные из базы данных |
| 9 | Верстка сайта и добавление функционала | 17.11.2018-13.12.2018 | Появился сайт с функционалом |
| 10 | Подключение апи к сайту | 14.12.2018-19.12.2018 | На сайте теперь появляется информация из базы данных |
| 11 | Тестирование и исправление | 20.12.2018-24.12.2018 | Исправлены баги, код отрефакторен |

Таблица 2 – Распределение работ.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Разработка БД | Создание макетов | Разработка приложения |
| Веселов | да | да | да |
| Глухова | нет | да | да |
| Габайдуллин | да | да | нет |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Разработка сайта | Разработка API | Тестирование | Исправление |
| Веселов | нет | нет | да | да |
| Глухова | нет | да | да | да |
| Габайдуллин | да | нет | да | да |

# ЗАКЛЮЧЕНИЕ

Итогом проделанной нами работы стал продукт, состоящий из мобильного приложения для мастеров и сайта для диспетчеров. Было обеспечено взаимодействие между сайтом, базой данных и приложением. Диспетчер может отправлять заявки, оформляя их в личном кабинете на сайте, получив статус диспетчера после регистрации, который может выдать администратор. Диспетчер также может отслеживать ход выполнения заявки, принял ли мастер, выполнил ли. Оформленная диспетчером заявка имеет статус одобренной и отправляется мастеру, выбранному диспетчером. После получения заявки, мастер должен приять ее и отправиться на выполнение, принять он ее может в любом месте, имея при себе лишь телефон на платформе андроид с доступом в интернет. При выполнении принятой мастером заявки он должен нажать одну кнопку, тем самым подтвердив выполнение заявки. Тогда диспетчер увидит, что заявка завершена. Однако мы не планируем останавливаться на достигнутом, мы хотим развивать наше приложение дальше и у нас имеется план работ на будущее. Главной и первоочередной задачей для нас теперь является интеграция с программным обеспечением 1С, которое заинтересует большее количество управляющих компаний, поскольку многие из них используют данное ПО. Кроме того, мы хотим добавить возможность отслеживания месторасположения мастеров для диспетчеров, это позволит диспетчерам держать в курсе жильцов о ходе выполняемой работы, а также выявлять недобросовестных работников. Если данное приложение приживется в управляющих компаниях, мы планируем добавить возможность авторизации жильцов в приложении для оформления заявок, это снимет нагрузку с диспетчеров и позволит жильцам сообщать о неполадках дистанционно.

Данный опыт помог нам во многом: работе в команде, распределению обязанностей, планированию времени, коммуникации с потенциальными заказчиками, а также поиску клиентов. Мы все узнали что-то новое, освоили новые технологии, познакомились с тем, что раньше казалось сложным и непонятным. У нас были разногласия и споры по поводу оформления, организации проектов, технологий и языков программирования. Однако важно приходить к компромиссу для достижения поставленной цели. Мы не исключаем той возможности, что приложение может не заинтересовать управляющие компании, но это не станет концом работы над проектом, напротив, это повод совершенствовать и развивать приложение и дальше, искать новых клиентов и воплощать намеченные задумки.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мнения экспертов о платформах для мобильной разработки [Электронный ресурс] URL: https://tproger.ru/experts/18/ (дата обращения – 20.11.2018).
2. Курс PHP [Электронный ресурс] URL: https://webshake.ru/ (дата обращения – 26.10.2018).
3. Steve Suehring PHP, MySQL, JavaScript & HTML5 All-in-One For Dummies, 2013. – 720 с
4. Хабр, Пишем свой API для сайта [Электронный ресурс] URL: https://habr.com/post/143317/ (дата обращения – 11.12.2018).
5. Stackoverflow [Электронный ресурс] URL: https://stackoverflow.com (дата обращения – 15.12.2018).
6. Bootstrap, сайт фреймворка [Электронный ресурс] URL: bootstrap-3.ru (дата обращения – 07.12.2018).
7. GeekBrains, Выбираем язык для веб-разработки [Электронный ресурс] URL: https://geekbrains.ru/posts/road\_to\_web\_development (дата обращения – 17.10.2018).
8. Грофф, Вайнберг, Оппель SQL. Полное руководство, 2014. – 960 с.
9. Джон Дакетт  HTML и CSS. Разработка и дизайн  веб-сайтов, 2015. – 480 с.
10. Линн Бейли, Майкл Моррисон Изучаем PHP и MySQL, 2010. – 800с.
11. Сайт Александра Климова [Электронный ресурс] URL: http://developer.alexanderklimov.ru/android/kotlin/ (дата обращения – 18.11.2018).
12. Сильвио Морето Bootstrap в примерах, 2015. – 314 с.
13. Openedu [Электронный ресурс] URL: https://openedu.ru/ (дата обращения – 15.11.2018).