

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА

Рецензент
Ведущий программист
ООО «ВОРТЕКСКОД»
_____ П.А. Михайлов
“ ___ ” _____ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,
д.ф.-м.н., профессор
_____ Л.Б. Соколинский
“ ___ ” _____ 2020 г.

**Разработка мобильного Android-приложения
«Конструктор тортов»**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2020. 308-031.ВКР

Научный руководитель,
к.т.н., доцент кафедры СП
_____ М.В. Сухов

Автор работы,
студент группы КЭ-401
_____ М.Т. Удесиани

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
“ ___ ” _____ 2020 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
03.02.2020

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра
студенту группы КЭ- 401
Удесиани Марии Тимуровне,
обучающемуся по направлению
02.03.02 «Фундаментальная информатика и информационные технологии»

1. **Тема работы** (утверждена приказом ректора от 24.04.2020 № 627)
Разработка мобильного Android-приложения «Конструктор тортов».
2. **Срок сдачи студентом законченной работы:** 08.06.2020.
3. **Исходные данные к работе**
 - 3.1. Общие сведения о платформе Android. [Электронный ресурс] URL:
<https://developer.android.com/guide/index.html>.
 - 3.2. Хорстманн К.С., Корнелл Г. Java. Библиотека профессионала. – М.:
Издательство «Вильямс», 2012. – 816 с.
4. **Перечень подлежащих разработке вопросов**
 - 4.1. Выполнить анализ предметной области.
 - 4.2. Спроектировать и реализовать мобильное приложение.
 - 4.3. Провести тестирование.
5. **Дата выдачи задания:** 03.02.2020.

Научный руководитель
к.т.н., доцент кафедры СП
Задание принял к исполнению

М.В. Сухов
М.Т. Удесиани

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1. Обзор аналогов.....	6
1.2. Обзор существующих средств разработки для ОС Android.....	8
1.3. Применение Google Firebase в Android-приложении.....	10
2. ПРОЕКТИРОВАНИЕ	12
2.1. Функциональные требования	12
2.2. Нефункциональные требования	12
2.3. Архитектура Android-приложения.....	12
2.4. Варианты использования мобильного приложения.....	13
2.4.1. Диаграмма деятельности.....	14
3. РЕАЛИЗАЦИЯ	16
3.1. Инструменты реализации.....	16
3.2. Реализация Android-приложения	16
3.2.1. Поддержка разных версий платформы Android	16
3.2.2. Реализация графического интерфейса.....	16
3.2.3. Реализация программного интерфейса.....	21
3.2.4. Диаграмма классов.....	22
3.3. Реализация серверной части приложения	23
4. ТЕСТИРОВАНИЕ.....	25
ЗАКЛЮЧЕНИЕ	28
ЛИТЕРАТУРА.....	29

ВВЕДЕНИЕ

АКТУАЛЬНОСТЬ ТЕМЫ

В настоящее время из-за увеличения доступности мобильного Интернета наблюдается склонность к использованию мобильных приложений для совершения таких действий, как заказ или покупка каких-либо товаров и услуг. Это дает возможность пользователю избежать необходимости самостоятельно совершать телефонный звонок, который может обернуться длительным ожиданием, недоступностью абонента и тратой денежных средств. Кроме того, у пользователя не всегда есть возможность говорить по телефону, что делает использование мобильного приложения удобнее, чем совершение звонка.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью данной работы является разработка Android-приложения «Конструктор тортов», которое предоставляет возможность заказа десерта.

«Конструктор тортов» – это приложение, которое позволяет пользователю собрать свой собственный торт, выбрав нужный вариант веса, количества ярусов, коржей, крема, начинки и декора. Так же пользователь может сразу оценить стоимость готового изделия.

Для достижения поставленной цели необходимо решить следующие задачи, перечисленные ниже.

1. Выполнить анализ предметной области.
2. Спроектировать и реализовать мобильное приложение.
3. Выполнить тестирование.

ОБЗОР ЛИТЕРАТУРЫ

В работах [1, 14, 20] представлены базовые понятия об ОС Android, основные принципы и решения, применяемые при разработке.

В источниках [15, 16, 17] описаны существующие платформы для создания приложений для ОС Android.

В работе [8] представлены основные принципы Material Design, которые стали основой при разработке графического интерфейса.

В источнике [21] описана работа платформы Firebase в Android-приложении.

СТРУКТУРА И ОБЪЕМ РАБОТЫ

Работа состоит из введения, четырех глав, заключения и библиографии. Объем работы составляет 30 страниц, объем библиографии – 21 источник.

СОДЕРЖАНИЕ РАБОТЫ

Первая глава «Анализ предметной области» содержит постановку задачи, обзор аналогичных проектов и существующих средств разработки для ОС Android.

Вторая глава «Проектирование мобильного приложения» содержит описание и анализ требований к приложению «Конструктор тортов», а также описание архитектур.

Третья глава «Реализация мобильного приложения» описывает подробности графической и программной реализации мобильного Android-приложения, а также его серверной части.

Четвертая глава «Тестирование мобильного приложения» посвящена результатам функционального тестирования мобильного Android-приложения.

В заключении описываются основные результаты, полученные при выполнении выпускной квалификационной работы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Целью данной работы является разработка мобильного приложения для сборки торта. В рамках текущей работы предлагается следующее решение поставленной задачи: разработка приложения, в функционал которого входит выбор веса готового изделия, количества ярусов, вида бисквита, крема, начинки и декора, а также дальнейший подсчет стоимости изделия. В качестве готового решения предполагается визуализация заказа в виде 2d-изображения готового изделия.

1.1. Обзор аналогов

Мобильных приложений, являющихся аналогами для «Конструктора тортов», не было найдено ни в App Store, ни в Google Play, но есть ряд веб-сайтов, которые предлагают такую функцию.

Favorite cake

На сайте кондитерской пользователю предложено собрать свой авторский торт [6]. Он должен выбрать количество и форму уровней, основу, крем для начинки и покрытия, топпинг, ягоды и декор. Также можно указать любую надпись, которую разместят на торте. Предусмотрен макет готового изделия.

Значительным недостатком является отсутствие примерной стоимости готового торта и отсутствие выбора веса десерта.

Забыли сахар

Кондитерская предлагает выбрать основу, начинку, количество ярусов, а также покрытие и дополнения для торта. Еще пользователь должен указать количество персон, на которых рассчитан десерт, чтобы определить вес изделия. В конце пользователь может увидеть примерный расчет стоимости [11].

Кондитерская Олега Ильина

Автор заверяет, что пользователь может выбрать коржи, начинку и топпинг, создав уникальный торт, но конструктор заключается в том, что

на уже готовый торт из списка Вы можете добавить надпись, фигурку или фотопечать [12]. То есть категории выбора по составляющим торта отсутствуют.

Kuzina

Кондитерская предлагает на выбор уже готовые сочетания бисквита, крема и начинки. Также пользователь может выбрать только из двух вариантов веса торта [7]. Есть итоговая стоимость.

Британские пекарни

Как основу пользователь может выбрать уже готовый торт, его форму и вес [9]. Также можно добавить шоколадную табличку с текстом или фото. Рассчитана цена изделия.

Cake Art 44

Потенциальному клиенту предложено собрать торт, который подходит именно ему [5]. Основой являются уже готовые сочетания классических тортов, пользователь выбирает дату, на которую хочет заказать изделие, вес и тематику будущего десерта.

Недостатком является обязательная регистрация на сайте, а также отсутствие стоимости.

Мастер шоколад

Пользователям предложено самим выбрать состав торта из уже готовых сочетаний, указать количество ярусов и вес продукта [13]. Можно добавить фотопечать, загрузив фотографию, или надпись. В виде декора предложены только фигурки из мастики. Показана примерная стоимость изделия.

Славишна

Кондитерская предлагает выбрать форму заготовки, подставки для торта, покрытие, начинку, коржи и украшение. Пользователь самостоятельно указывает вес изделия, количество ярусов и желаемую надпись [18]. Есть возможность прикрепить эскиз торта. Отсутствует стоимость изделия.

Студия Людмилы Мостаковой

Пользователь выбирает количество уровней, начинку, ингредиенты, форму уровней торта, цвет глазури, декор, цветы и фигурки. В конце можно посчитать примерную стоимость [19].

Джем

На сайте кондитерской пользователю предложено собрать свой торт. Он должен указать вариант бисквита, крема, начинки и декора, а также желаемый вес. Присутствует стоимость изделия [10].

1.2. Обзор существующих средств разработки для ОС Android

На данный момент существует несколько платформ для создания приложений для Android. Некоторые из них представлены ниже.

1. Android Studio [15], основанная на языке Java и Android SDK [14].
2. Eclipse [16], поддерживающий язык Java и Android Development Tools [4].
3. Xamarin [17], основанная на языке программирования C# и платформе .NET.

Android SDK

Android SDK включает в себя разнообразные библиотеки, документацию и инструменты, которые помогают разрабатывать мобильные приложения для ОС Android.

1. API Android SDK – API библиотеки Android, предоставляемые для разработки приложений.
2. Документация SDK – включает обширную справочную информацию, детализирующую, что включено в каждый пакет и класс и как это использовать при разработке приложений.
3. AVD (Android Virtual Device) – интерактивный эмулятор мобильного устройства Android. Используя эмулятор, можно запускать и тестировать приложения без использования реального Android устройства.

4. Development Tools – SDK включает несколько инструментальных средств для разработки, которые позволяют компилировать и отлаживать создаваемые приложения.

5. Sample Code – Android SDK предоставляет типовые приложения, которые демонстрируют некоторые из возможностей Android, и простые программы, которые показывают, как использовать индивидуальные особенности API в вашем коде.

Android Studio

Android Studio – это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google. Является официальной средой разработки приложений для платформы Android. Данная среда разработки имеет следующие особенности, представленные ниже.

1. Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана.

2. Различные виды сборок и генерация нескольких .apk файлов.

3. Рефакторинг кода.

4. Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.

5. Шаблоны основных макетов и компонентов Android.

6. Поддержка разработки приложений для Android Wear и Android TV.

7. Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine.

Eclipse

Eclipse – свободная интегрированная среда разработки кроссплатформенных приложений. Android Development Tools (ADT) – это плагин для Eclipse IDE, позволяющий разрабатывать приложения для платформы Android. Данная среда разработки позволяет создавать

настолько же эффективные приложения для Android, но т.к. Android Studio является инструментом, специально созданным для мобильной разработки, то она позволяет проще и быстрее создавать приложения по сравнению с Eclipse.

Xamarin

Xamarin – это фреймворк для кроссплатформенной разработки мобильных приложений с использованием языка C#. Xamarin обладает следующими преимуществами и недостатками, перечисленными ниже.

1. Позволяет создавать кроссплатформенные мобильные приложения (iOS, Android, Windows Phone).
2. Не требует знаний стандартных средств реализации Android приложений (основан на C# и платформе .NET).
3. Платная лицензия.
4. Неполная поддержка стандартных API платформы.
5. Относительно низкая производительность.

1.3. Применение Google Firebase в Android-приложении

Большинство мобильных приложений используют сервер для хранения и передачи данных. Альтернативой классической разработки являются облачные сервисы, предоставляющие возможность хранить гибкую систему данных у себя на сервере.

Бэкенд как сервис (**BaaS**) – модель, которая предоставляет разработчикам мобильных и веб-приложений возможность хранить и обрабатывать данные в облаке, не поднимая собственный сервер, а также предоставляет такие функции, как управление пользователями, push-уведомления и интеграцию с сервисами социальных сетей. Эти услуги предоставляются посредством использования комплектов для разработки специального программного обеспечения (SDK) и интерфейсов прикладного программирования (API) [21].

Одним из самых популярных BaaS является **Firestore** от компании Google. Основным направлением Firestore является облачная NoSQL база данных для real-time приложений, которая предоставляет API, позволяющее разработчикам хранить и синхронизировать данные между несколькими клиентами. База данных позволяет работать с данными, которые синхронизируются в реальном времени и доступны при отсутствии интернета. Важно отметить, что для работы с Firestore в Android Studio уже встроен плагин (начиная с версии 2.2), с помощью которого можно легко настраивать сервисы, а также автоматически подключаться к приложению и добавлять зависимости.

Учитывая преимущества и недостатки рассмотренных инструментов, для реализации приложения была выбрана среда разработки Android Studio и платформа Firestore.

2. ПРОЕКТИРОВАНИЕ

2.1. Функциональные требования

Разрабатываемое мобильное приложение для сборки тортов должно удовлетворять следующим функциональным требованиям:

- приложение должно позволять пользователю просматривать без подключения к Интернету информацию по сборке торта;
- приложение должно предоставлять пользователю возможность выбирать позиции во всех категориях (вес, количество ярусов, бисквит, крем, начинка и декор);
- приложение должно предоставлять пользователю возможность просматривать выбранные позиции перед осуществлением заказа;
- приложение должно предоставлять пользователю информацию о стоимости готового изделия;
- приложение должно предоставлять возможность пользователю визуальную модель будущего изделия в формате картинки;
- приложение должно позволять пользователю осуществлять заказ собранного торта;

2.2. Нефункциональные требования

В результате анализа функциональных требований и обзора аналогичных проектов, были сформулированы следующие нефункциональные требования:

- приложение должно быть написано на языке Java под ОС Android 5.0 и выше;
- серверная часть приложения должна быть реализована с помощью Firebase.

2.3. Архитектура Android-приложения

Архитектура приложений, как правило, проектируется в терминах архитектуры «клиент-сервер», при которой один или несколько

клиентских устройств запрашивают информацию с сервера. Сервер обычно отвечает необходимой информацией (рисунок 1).

Для описания архитектуры принято использовать UML – унифицированный язык моделирования. Этот открытый стандарт использует графические обозначения для моделирования определенных аспектов системы, таких как отношения классов или жизненный цикл какого-либо события.

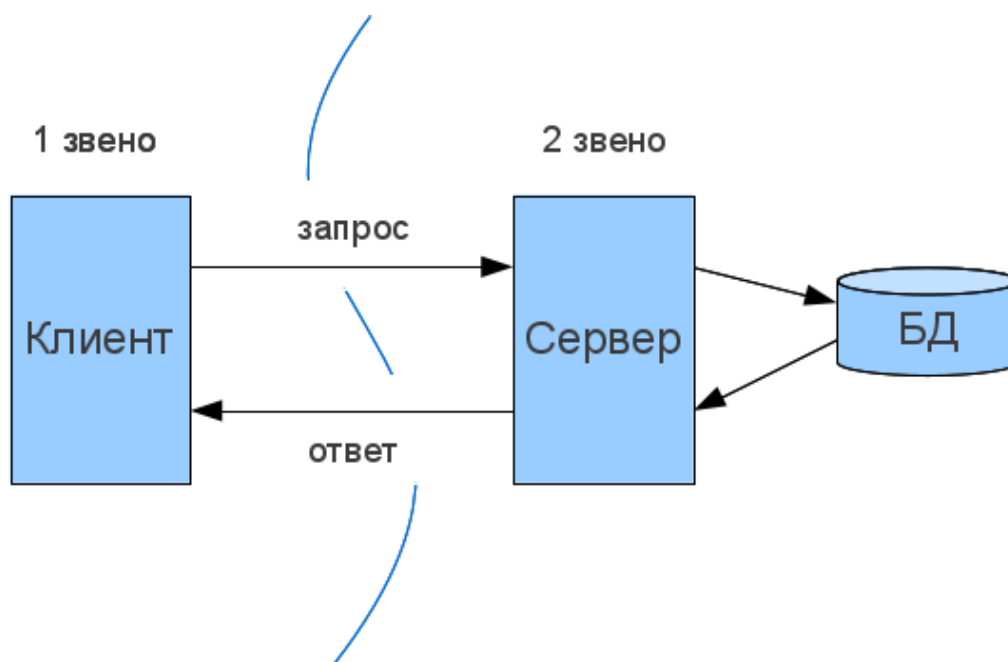


Рис. 1. Клиент-серверная архитектура

2.4. Варианты использования мобильного приложения

Диаграмма вариантов использования мобильного приложения представлена на рисунке 2. С приложением взаимодействует один актер – пользователь.

Пользователь может *собрать торт*, указав вес торта, количество ярусов, бисквит, крем, начинку и декор.



Рис. 2. Варианты использования мобильного приложения «Конструктор тортов»

2.4.1. Диаграмма деятельности

На рисунке 3 изображена диаграмма деятельности. Она предназначена для моделирования процесса как деятельности, состоящей из коллекции соединенных ребрами узлов. Диаграмма деятельности применяется для моделирования деталей алгоритма сборки торта.

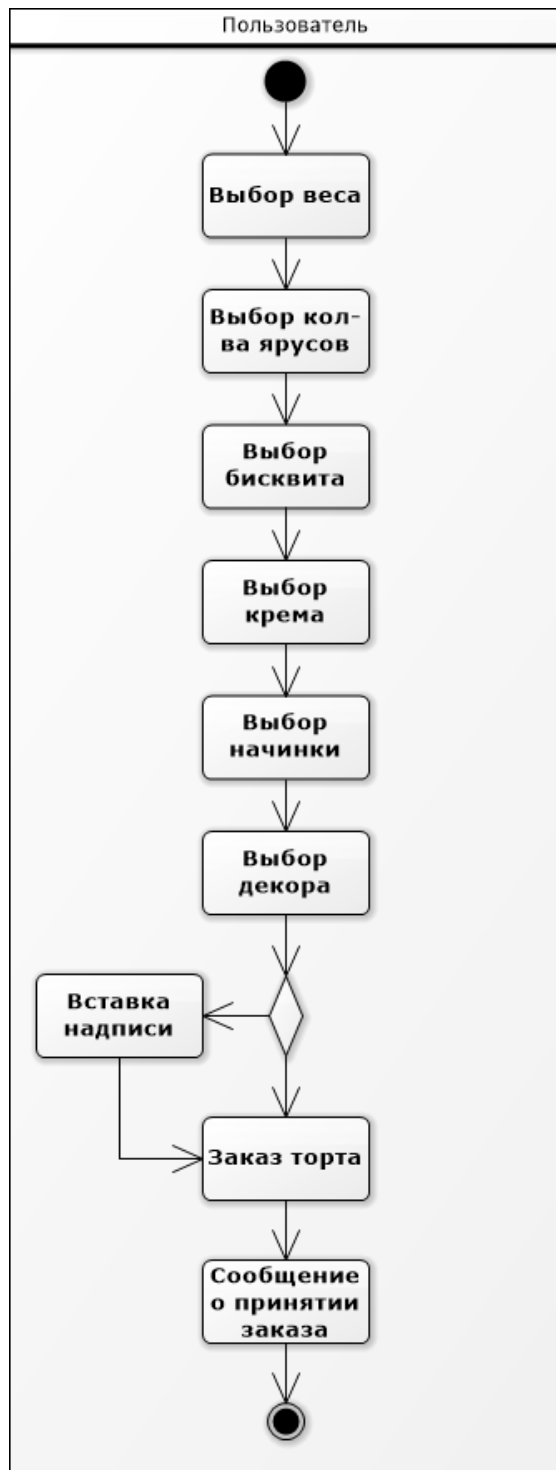


Рис. 3. Диаграмма деятельности

3. РЕАЛИЗАЦИЯ

3.1. Инструменты реализации

Клиентская часть системы реализована для ОС Android, являющейся наиболее популярной мобильной операционной системой в мире [3] на высокоуровневом объектно-ориентированном языке Java. Серверная часть разработана с использованием Firebase.

3.2. Реализация Android-приложения

3.2.1. Поддержка разных версий платформы Android

В каждом Android-проекте необходимо обозначить минимальную поддерживаемую приложением версию платформы Android. Чем меньше версия, тем на большее число устройств будет возможна установка приложения, но будет невозможна или ограничена работа с некоторыми возможностями API более поздних версий платформы [20].

Согласно официальному сайту платформы Android [2], доля Android-устройств версии 5.0 и выше составляет 94,1 % на момент начала апреля 2020 года. Поэтому в качестве минимальной поддерживаемой версии было решено выбрать версию 5.0 (Lollipop, API 21).

3.2.2. Реализация графического интерфейса

Дизайн приложения разрабатывался с опорой на принципы Material Design. *Material Design* представляет собой комплексную концепцию создания визуальных, движущихся и интерактивных элементов для различных платформ и устройств от компании Google [8].

Для навигации по основным экранам приложения используются кнопки. Рассмотрим вышеописанный подход на примере определения конкретного стиля кнопок, используемых в приложении для стилизации элементов управления. На рисунках 4 и 5 приведено определение стиля кнопки на главном экране для платформы Android и ресурсы,

используемые для определения типа изменения внешнего вида элемента при взаимодействии пользователя с ним.

```
<Button
    android:id="@+id/button"
    android:layout_width="185dp"
    android:layout_height="69dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:background="@drawable/buttonshape"
    android:shadowColor="#8B48A8"
    android:shadowDx="0"
    android:shadowDy="0"
    android:shadowRadius="5"
    android:text="@string/act2"
    android:textColor="#333333"
    android:textSize="18sp"
    app:layout_constraintVertical_bias="0.876" />
```

Рис. 4. Определение стиля кнопки в файле activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector
xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true" >
        <shape android:shape="rectangle" >
            <corners android:radius="50dip" />
            <stroke android:width="1dip" android:color="#462d4f" />
            <solid android:color="#8b819b"/>
        </shape>
    </item>
    <item android:state_focused="true">
        <shape android:shape="rectangle" >
            <corners android:radius="50dip" />
            <stroke android:width="1dip" android:color="#a773d1" />
            <solid android:color="#8b819b"/>
        </shape>
    </item>
</selector>
```

Рис. 5. Определение стиля кнопки в файле drawable/buttonshape.xml

Результаты отображения интерфейса на устройстве Pixel 2, 5.0", Android 9.0

На рисунке 6 представлен главный экран приложения. Нажав на кнопку «Собрать торт», пользователь переходит на следующий экран.

На рисунке 7 показан экран с выбором веса. Пользователь выбирает желаемый вес от 2 до 5 кг. с шагом в 500 гр.

На рисунке 8 изображен экран с выбором количества ярусов (от 1 до 3).

На рисунке 9 представлен экран с категорией «бисквит». Пользователь выбирает один из предложенных вариантов (шоколадный, ванильный, красный бархат)

На рисунке 10 показан экран выбора крема, на котором пользователь указывает нужную позицию (шоколадный, карамельный, кремчиз, ореховый, сливочный с маскарпоне).

На рисунке 11 изображен экран с вариантами начинки, где пользователь может выбрать несколько позиций сразу (клубника, малина, вишня, соленая карамель).

На рисунке 12 представлен экран с выбором декора. Пользователь также может указать несколько понравившихся позиций (ягоды, карамель, ганаш, орехи).

На рисунке 13 показан экран с информацией о заказе. Пользователь видит выбранные им позиции по каждой из категорий, стоимость готового торта, а также 2d-модель будущего изделия.

На рисунке 14 изображен экран подачи заявки для заказа собранного торта. Пользователь должен ввести свои данные для осуществления заказа, а также согласиться на обработку персональных данных.

На рисунке 15 изображен финальный экран. Он становится доступным после того, как пользователь ввел все данные.

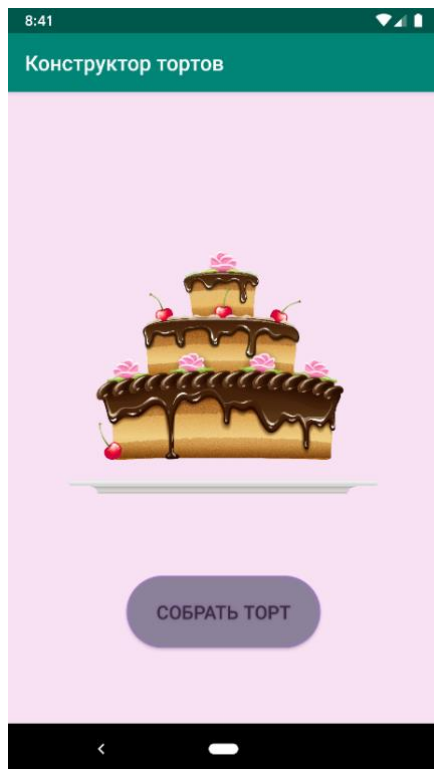


Рис. 6. Главный экран



Рис. 7. Экран выбора веса

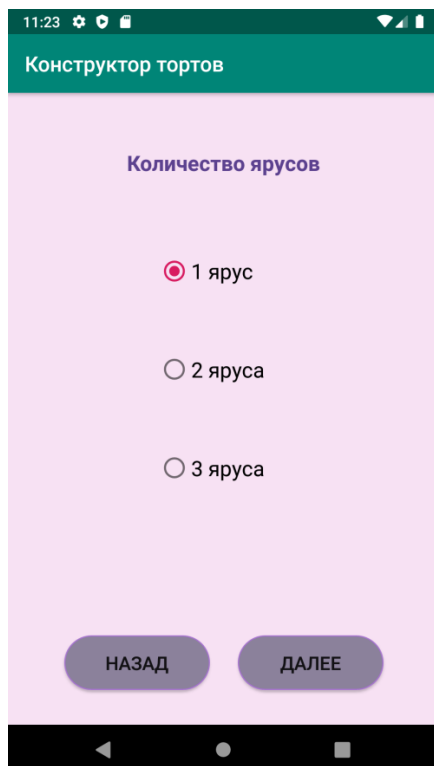


Рис. 8. Экран выбора количества ярусов

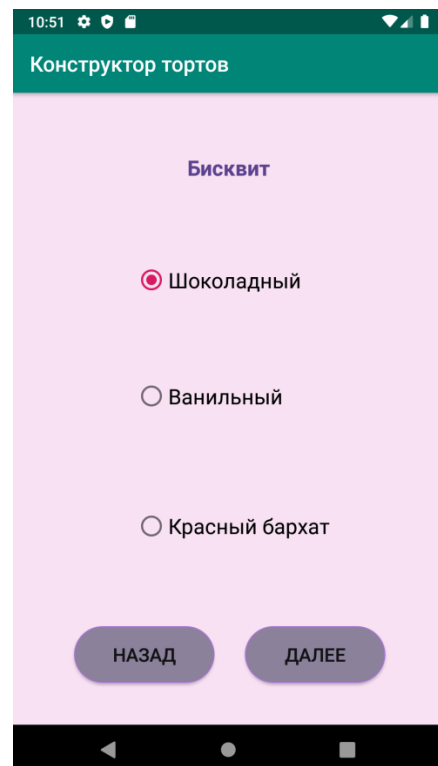


Рис. 9. Экран выбора бисквита



Рис. 10. Экран выбора крема

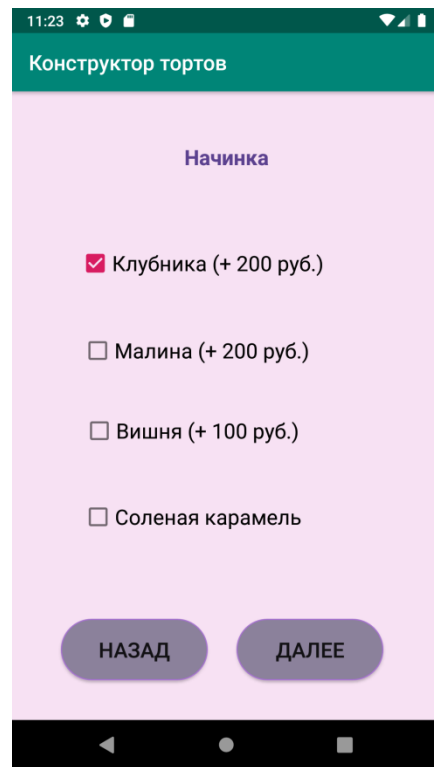


Рис. 11. Экран выбора начинки

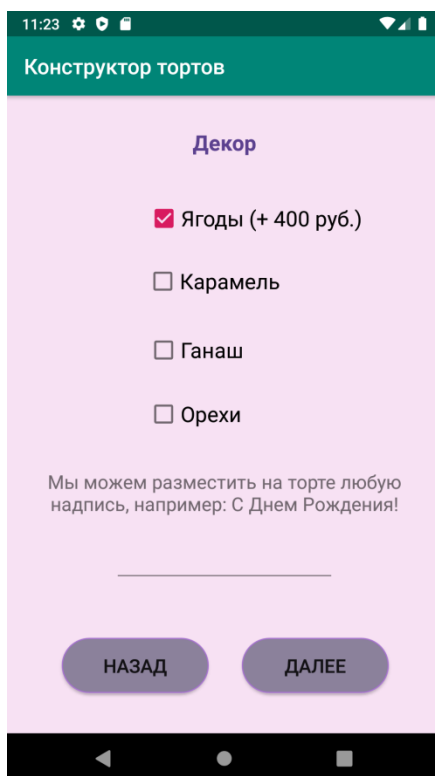


Рис. 12. Экран выбора декора



Рис. 13. Экран с информацией о заказе

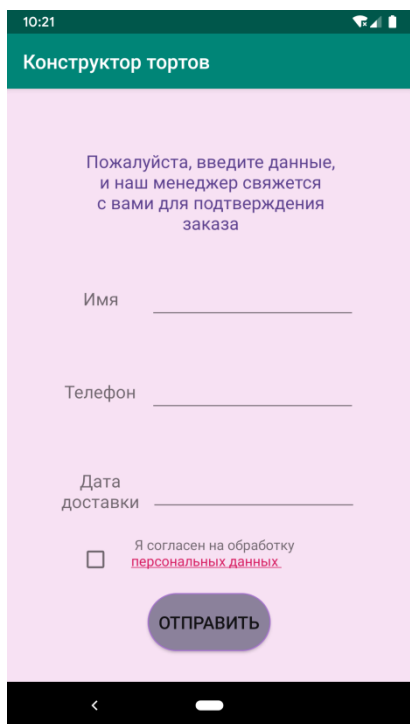


Рис. 14. Экран заявки

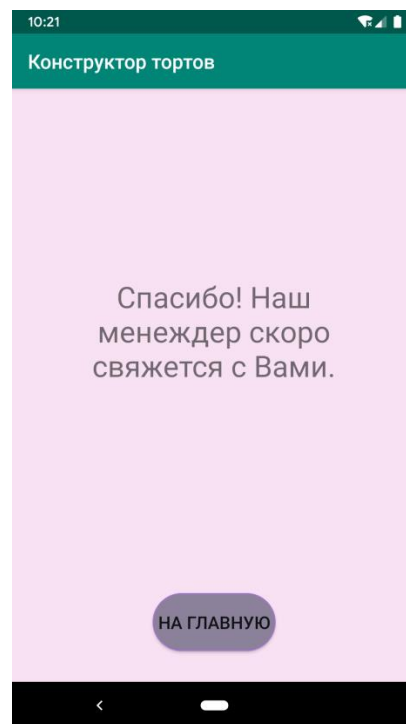


Рис. 15. Финальный экран

3.2.3. Реализация программного интерфейса

Для навигации по экранам приложения используются кнопки. На рисунке 16 изображен исходный код функции обработки нажатия пользователем на кнопку «Далее» на экране с выбором веса. Для обработки нажатия на кнопку в Android предусмотрен специальный объект `OnClickListener` [1], в котором можно реализовать метод, выполняющийся при нажатии кнопки.

При нажатии на кнопку «Далее» происходит переход на следующий экран, но только при условии, что одна из позиций выбрана. Реализовано всплывающее уведомление (`Toast Notification`), как обработчик ошибки, которое появляется на поверхности окна приложения. При этом текущая деятельность приложения остаётся работоспособной для пользователя. В течение нескольких секунд сообщение плавно закрывается. Аналогичным образом осуществляется обработчик событий и на других экранах.

```

act6.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        int id = getIntent().getIntExtra("id", 1);
        if (rb1.isChecked()) {
            Intent intent = new Intent();
            intent.setClass(Activity3.this, Activity10.class);
            weight = "2 кг.";
            sum = 2500;
            intent.putExtra("id", id);
            intent.putExtra("res", sum );
            intent.putExtra("res2", weight );
            startActivity(intent);
        } else if (rb2.isChecked()) {
            Intent intent = new Intent();
            intent.setClass(Activity3.this, Activity10.class);
            weight = "2,5 кг.";
            sum = 3000;
            intent.putExtra("id", id);
            intent.putExtra("res", sum );
            intent.putExtra("res2", weight );
            startActivity(intent);
        }

        /*здесь представлен программный код, который
        .по такому же алгоритму производит обработку
        .нажатия кнопки*/

        else if (!rb1.isChecked() && !rb2.isChecked() &&
        !rb3.isChecked() && !rb4.isChecked() &&
        !rb5.isChecked() && !rb6.isChecked() &&
        !rb7.isChecked()) {
            Toast toast =
            Toast.makeText(getApplicationContext(), "Выберите
            позицию", Toast.LENGTH_SHORT);
            toast.setGravity(Gravity.BOTTOM, 0, 230);
            toast.show();
        }
    }
});

```

Рис. 16. Фрагмент листинга обработки нажатия кнопки «Далее»

3.2.4. Диаграмма классов

В результате реализации приложения была разработана диаграмма классов, которая изображена на рисунке 17.

На диаграмме представлены основные классы мобильного приложения. Классы не наследуются. В каждом классе реализован обязательный метод `onCreate` – метод, который задаёт начальную установку параметров при инициализации активити.

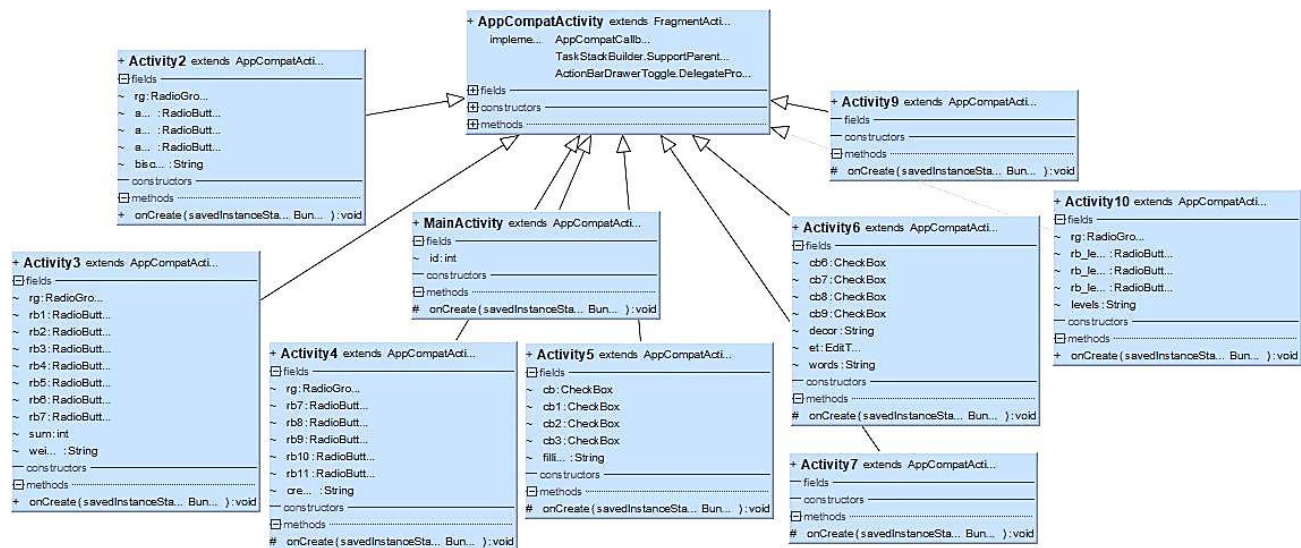


Рис. 17. Диаграмма классов

3.3. Реализация серверной части приложения

Серверная часть мобильного приложения была разработана с использованием платформы Firebase. После подключения плагина Firebase к проекту в Android Studio можно представить базу данных в реальном времени как объект JSON.

С помощью метода `getInstance()` класса `FirebaseDatabase` мы получаем ссылку на `Realtime Database`. Затем с помощью метода `getReference()` мы можем получить ссылку на любой путь внутри базы данных. Путь может содержать либо один ключ, либо несколько ключей, расположенных иерархически. Когда есть `DatabaseReference` к пути, можно установить его значение с помощью метода `setValue()`. На рисунке 18 показано, как добавить в базу данных простую пару «ключ-значение».

```

FirebaseDatabase db = FirebaseDatabase.getInstance();
DatabaseReference ref = db.getReference("name"); // Key
ref.setValue("maria"); // Value

```

Рис.18. Добавление данных в Realtime Database

Перейдя в раздел «Database» консоли Firebase, можно просмотреть все данные, находящиеся внутри базы данных в режиме реального времени (рисунок19).

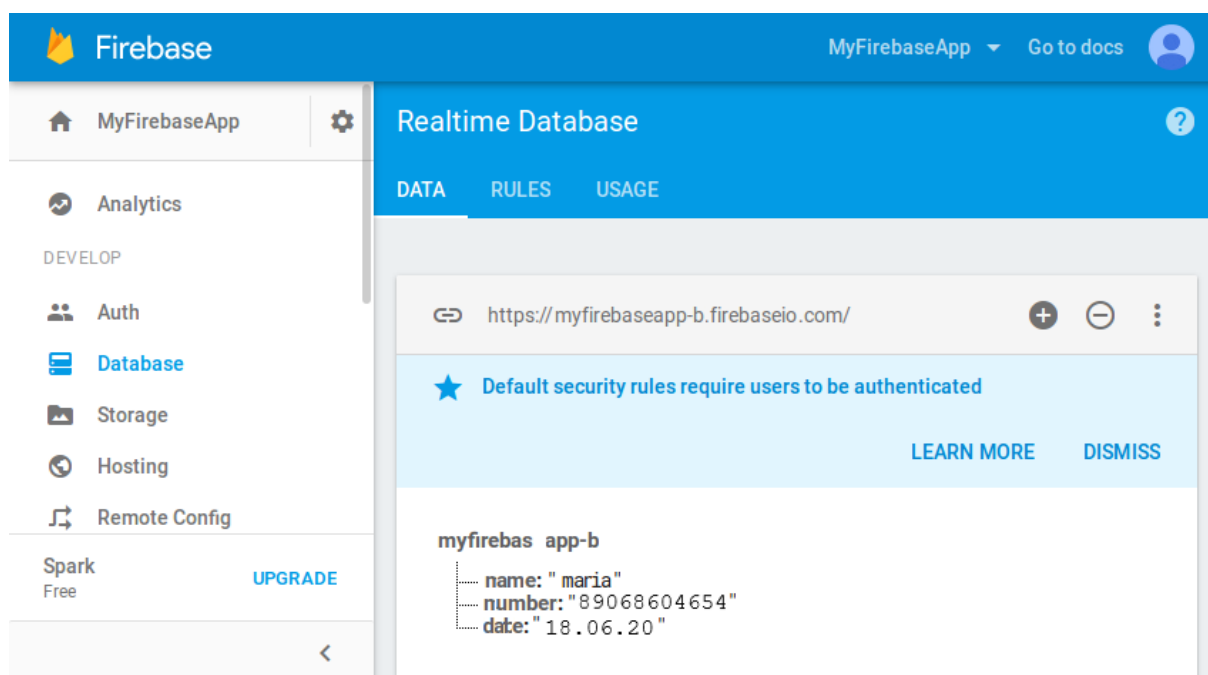


Рис.19. Отображение данных в Firebase

4. ТЕСТИРОВАНИЕ

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способность ПО решать задачи, необходимые пользователям. Было проведено функциональное тестирование разработанного приложения, результаты которого приведены в таблице 1.

Табл. 1. Тестирование приложения

№	Аспект работы	Действия	Результат	Тест пройден ?
1	Выбор веса	1. Указать желаемый вес торта, выбрав один из вариантов. 2. Нажать на кнопку «Далее».	Происходит перенаправление на экран с выбором количества ярусов.	Да
		1. Оставить позицию невыбранной 2. Нажать на кнопку «Далее».	Переход на следующий экран не осуществляется, выводится сообщение об ошибке.	Да
2	Выбор количества ярусов	1. На экране выбрать количество уровней. 2. Нажать на кнопку «Далее».	Происходит перенаправление на экран с выбором бисквита.	Да
		1. Оставить позицию не выбранной 2. Нажать на кнопку «Далее».	Перенаправление на следующий экран не осуществляется, выводится сообщение об ошибке	Да
3	Выбор бисквита	1. На экране выбрать один из вариантов бисквита. 2. Нажать на кнопку «Далее».	Происходит переход на экран с выбором крема.	Да
		1. Оставить позицию не выбранной 2. Нажать на кнопку «Далее».	Перенаправление на следующий экран не осуществляется, выводится сообщение об ошибке.	Да

№	Аспект работы	Действия	Результат	Тест пройден?
4	Выбор крема	1. На экране выбрать один из вариантов бисквита. 2. Нажать на кнопку «Далее».	Происходит перенаправление на экран с выбором начинки.	Да
		1. Оставить позицию не выбранной 2. Нажать на кнопку «Далее».	Переход на следующий экран не осуществляется, выводится сообщение об ошибке.	Да
5	Выбор начинки	1. Выбрать один или несколько вариантов начинки. 2. Нажать на кнопку «Далее».	Происходит перенаправление на экран с выбором декора.	Да
		1. Оставить позицию не выбранной 2. Нажать на кнопку «Далее».	Происходит перенаправление на экран с выбором декора.	Да
6	Выбор декора	1. Выбрать один или несколько вариантов декора. 2. Нажать на кнопку «Далее».	Происходит переход на экран с информацией о заказе.	Да
		1. Оставить позицию не выбранной 2. Нажать на кнопку «Далее».	Перенаправление на экран с информацией о заказе.	Да
		1. В поле с надписью попытаться ввести строку длиной более чем 40 символов.	Более чем 40 символов ввести невозможно.	Да
7	Отправка заявки	1. В поле имя, телефон и дата доставки ввести данные, согласиться на обработку персональных данных. 2. Нажать на кнопку «Отправить».	Происходит переход на финальный экран.	Да

№	Аспект работы	Действия	Результат	Тест пройден?
		<ol style="list-style-type: none"> 1. Одно или несколько из полей оставить незаполненным. 2. Нажать на кнопку «Отправить». 	Переход на следующий экран не осуществляется, выводится сообщение об ошибке.	Да
		<ol style="list-style-type: none"> 1. В поле «телефон» указать номер меньше 11 символов и/или начинающийся не с 8 2. Нажать на кнопку «Отправить» 	Переход на следующий экран не осуществляется, выводится сообщение об ошибке.	Да
		<ol style="list-style-type: none"> 1. В поле «Имя» попытаться ввести строку длиной более чем 40 символов. 	Более чем 40 символов ввести невозможно.	Да
8	Передача данных в Firebase	<ol style="list-style-type: none"> 1. Собрать торт и ввести свои данные. 2. Нажать на кнопку «Отправить». 	Отображение данных в консоли Realtime Database в Firebase, переход на финальный экран.	Да
		<ol style="list-style-type: none"> 1. Собрать торт и ввести свои данные. 2. Не нажать на кнопку «Отправить». 	Введенные данные не отображаются в Firebase	Да

ЗАКЛЮЧЕНИЕ

В соответствии с целями данной работы был проведен обзор научной литературы и существующих аналогов, который подтвердил актуальность поставленной задачи.

Были определены функциональные и нефункциональные требования и представлены диаграммы деятельности и вариантов использования, а также спроектирована система, состоящая из клиентского приложения и серверной части.

Представлена графическая и программная реализация мобильного Android-приложения, а также его серверной части. Разработанная система соответствует всем требованиям.

Было выполнено функциональное тестирование системы. Все тесты успешно пройдены.

В рамках дипломного проекта было разработано и реализовано мобильное Android-приложение «Конструктор тортов» состоящее из клиентского приложения и серверной части. Код приложения составил свыше 1500 строк кода на языке Java и около 1600 строк на языке разметки XML.

Были решены следующие задачи.

1. Выполнен анализ предметной области.
2. Спроектировано и реализовано мобильное приложение.
3. Выполнено тестирование.

ЛИТЕРАТУРА

1. Android Developers. [Электронный ресурс] URL: <https://developer.android.com/> (дата обращения: 15.02.2020).
2. Android Developers. [Электронный ресурс] URL: <https://developer.android.com/about/dashboards/index.html> (дата обращения: 20.03.2020).
3. Android Development Tool. [Электронный ресурс] URL: <https://developer.android.com> (дата обращения: 17.02.2020).
4. Android Development Tools for Eclipse. [Электронный ресурс] URL: <https://marketplace.eclipse.org/content/android-development-toolseclipse> (дата обращения: 06.04.2020).
5. Cake Art 44. [Электронный ресурс] URL: <https://cakeart44.ru/konstruktor-tortov/> (дата обращения: 15.03.2020).
6. Favorite cake. [Электронный ресурс] URL: <http://www.ilovecake.ru/designer/> (дата обращения: 19.03.2020).
7. Kuzina. [Электронный ресурс] URL: <https://kuzina.ru/tort> (дата обращения: 14.03.2020).
8. Material design guidelines SendGrid. [Электронный ресурс] URL: <https://material.io/guidelines/material-design/introduction.html> (дата обращения: 20.04.2020).
9. Британские пекарни. [Электронный ресурс] URL: <https://british-bakery.ru/calс/> (дата обращения: 17.03.2020)
10. Джем. [Электронный ресурс] URL: <http://jamjars.ru/> (дата обращения: 14.04.2020).
11. Забыли сахар. [Электронный ресурс] URL: <https://xn--80aaac2aiu7bg5b9c.xn--p1ai/constructor> (дата обращения: 15.03.2020).
12. Кондитерская Олега Ильина. [Электронный ресурс] URL: <http://sweetstore.ru/konstruktor/> (дата обращения: 15.03.2020).

13. Мастер шоколад. [Электронный ресурс] URL: <https://master-chocolate.ru/constructor/> (дата обращения: 15.03.2020).
14. Общие сведения о платформе Android. [Электронный ресурс] URL: <https://developer.android.com/guide/index.html> (дата обращения: 29.03.2020).
15. Сайт Android Studio. [Электронный ресурс] URL: <https://developer.android.com/studio/features.html> (дата обращения: 06.04.2020).
16. Сайт Eclipse. [Электронный ресурс] URL: <https://eclipse.org/> (дата обращения: 06.04.2020).
17. Сайт Xamarin. [Электронный ресурс] URL: <https://www.xamarin.com/> (дата обращения: 19.03.2020).
18. Славишна. [Электронный ресурс] URL: <http://slavishna.ru/zakaz/> (дата обращения: 15.03.2020)
19. Студия Людмилы Мостаковой. [Электронный ресурс] URL: <https://vipcake.by/calculate> (дата обращения: 17.03.2020)
20. Харди Б., Филлипс Б. Программирование под Android. – СПб.: Питер, 2014. – 592 с.
21. Чевжик, Е. А. Использование Google Firebase в Android-приложении / Е. А. Чевжик // 68-я научно-техническая конференция учащихся, студентов и магистрантов, 17-22 апреля, Минск : сборник научных работ : в 4 ч. Ч. 4 / Белорусский государственный технологический университет. – Минск : БГТУ, 2017. – С. 8-11. [Электронный ресурс] URL: <https://elib.belstu.by/handle/123456789/23623> (дата обращения: 20.04.2020).