

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н.П. ОГАРЁВА»

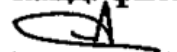
Факультет математики и информационных технологий

Кафедра фундаментальной информатики

УТВЕРЖДАЮ

Зав. кафедрой

канд. физ.-мат. наук, доцент



А.Г. Смольянов

(подпись)

«10» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

**ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «LET'S COOK» И
ЕГО РЕАЛИЗАЦИЯ СРЕДСТВАМИ ANDROID STUDIO**

Автор бакалаврской работы


(подпись)

10.06.19
(дата)

Т.С. Медведева

Обозначение бакалаврской работы БР-02069964-02.03.02-08-19

Направление 02.03.02 Фундаментальная информатика и информационные
технологии

Руководитель работы

канд. физ.-мат. наук, доцент


(подпись)

10.06.19
(дата)

Т.Е. Бадюкина

Нормоконтролер

канд. тех. наук, доцент


(подпись)

10.06.19
(дата)

С.В. Гарина

Рецензент

канд. физ.-мат. наук, доцент


(подпись)

10.06.19
(дата)

Р.В. Жалнин

Саранск

2019

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н.П. ОГАРЁВА»**

Факультет математики и информационных технологий

Кафедра фундаментальной информатики

УТВЕРЖДАЮ

Зав. кафедрой

канд. физ.-мат. наук, доцент

 **А.Г. Смольянов**
(подпись)

«12» 12 2019 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

**(в форме бакалаврской работы, дипломной работы, дипломного проекта,
магистерской диссертации)**

Студент Медведева Татьяна Сергеевна

**1 Тема «Проектирование мобильного приложения «Let's Cook» и его реализация
средствами Android Studio»**

Утверждена приказом № 10282-С от 12.12.2018

2 Срок предоставления работы к защите 24.06.2019

**3 Исходные данные для научного исследования (проектирования) научная и
учебно–методическая литература, периодические издания, электронные
ресурсы**

4 Содержание выпускной квалификационной работы

4.1 Теоретические аспекты разработки мобильного приложения

4.2 Проектирование мобильного приложения «Let's Cook»

4.3 Реализация приложения «Let's Cook»

5 Перечень графического материала: рисунки

Руководитель работы

Баг 12.12.2018
подпись, дата

Т.Е. Бадокина

Задание приняла к исполнению

Мед 12.12.2018
подпись, дата

Т.С. Медведева

РЕФЕРАТ

Выпускная квалификационная работа содержит 52 страницы, 31 рисунок, 1 таблицу, 15 использованных источников.

РЕЦЕПТЫ, ПРАВИЛЬНОЕ ПИТАНИЕ, ФОРМУЛА, БАЗЫ ДАННЫХ, SQL, ANDROID OS, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ANDROID STUDIO.

Объектом исследования является изучение возможностей среды разработки Android Studio на примере разработки собственного приложения.

Цель работы – разработать мобильное приложение для операционной системы Android для помощи в подборе рецептов и подсчете калорий.

В процессе работы была изучена научная и учебно–методическая литература, рассмотрены возможности Android Studio, проанализирован рынок приложений.

В результате исследования разработано мобильное приложение, которое помогает искать рецепты, создавать свои рецепты, рассчитывать суточную норму калорий и следить за её соблюдением.

Область применения – в жизни каждого человека.

Эффективность – данная работа направлена на улучшение качества жизни людей на примере питания.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1 Теоретические аспекты разработки мобильного приложения	
11	
1.1 Мобильные операционные системы	11
1.1.1 Android OS	11
1.1.2 iOS	15
1.1.3 Windows Phone	16
1.2 Основные конструкции мобильного приложения	17
1.3 Этапы разработки мобильного приложения	19
1.4 Языки программирования, используемые в мобильной разработке	21
1.4.1 Java	21
1.4.2 Kotlin	22
1.4.3 C#	23
1.5 Среды разработки мобильных приложений	24
1.5.1 Android Studio	24
1.5.2 Eclipse IDE	27
1.5.3 Visual Studio и Xamarin	28
1.6 Базы данных, используемые в мобильных приложениях	29
2 Проектирование мобильного приложения «Let's Cook»	31
2.1 Требования	31
2.2 Варианты использования	32
2.3 Структура базы данных	33
3 Реализация приложения «Let's Cook»	35
ЗАКЛЮЧЕНИЕ	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53

ВВЕДЕНИЕ

Необходимость создания мобильных приложений появилась после выхода на рынок мобильных телефонов с экраном. Первым программным обеспечением для телефонов являлись встроенные приложения, которые предназначались для выполнения определенных функций телефона и устанавливались в устройство самими производителями. Пожалуй, первым мобильным приложением, помимо ПО, отвечающего непосредственно за работу телефона, стала телефонная книга – приложение, которое содержало контакты пользователя. Изначально можно было записать только номер и имя абонента, но со временем функционал был расширен и появились такие дополнительные функции как адрес, день рождения, электронная почта, фотография. После появления технологии обмена сообщениями SMS, в телефоны было добавлено еще одно приложение, предоставляющее возможность отправки небольших текстов.

Первое приложение, которое появилось уже отдельно от исходного программного обеспечения, было реализовано в конце 90-х годов 20 века, когда сотовая связь получила широкое распространение во всем мире. К этому времени, производители телефонов уже были уверены, что софт для мобильного устройства – это перспективное направление, как с точки зрения продвижения технологий, так и с точки зрения их отдельного коммерческого использования. Поэтому, по умолчанию, в систему сотовых телефонов, помимо самых необходимых приложений, производители стали встраивать дополнительное ПО. Обычно это были различные мультимедийные приложения – редакторы рингтонов, небольшие аркадные игры, календари, калькуляторы и т. д.

Появились и сторонние разработчики, которые создавали приложения, аналогичные предустановленным, а также много другого полезного и иногда ненужного софта.

В момент появления в 1997 году на рынке сотовой связи технологии WAP (Wireless Application Protocol), которая позволяла выходить в интернет с

помощью мобильного устройства, количество программных приложений, как и их разработчиков, стало увеличиваться. Произошло это потому, что их установка значительно упростилась, и если раньше нужно было использовать DATA-кабель, то теперь пользователь не был привязан к персональному компьютеру и мог это делать в любом месте.

К 2000-ым годам на рынок мобильных устройств пришли смартфоны и коммуникаторы. Главными отличительными особенностями их от обычных телефонов были открытость системы и многозадачность. Открытость системы позволяла устанавливать любые приложения и изменять определенные настройки опытным пользователям, а многозадачность позволяла открывать несколько приложений и работать с ними одновременно. К примеру, можно слушать музыку и набирать сообщение, что в обычных телефонах невозможно.

Современный мир нельзя представить без приложений, они используются во всех окружающих нас сферах. С каждым днем их количество растет. В 2018 году пользователи загрузили более 400 миллионов бесплатных приложений. Это почти в 1,5 раза больше, чем в 2017-м. Из них можно получить информацию и воспользоваться любыми услугами такими как развлечения, путешествия, бизнес, общение, питание, спорт, образование, новости, медицина и др..

Последнее время в связи с развитием ресторанов быстрого питания, появлением новых вредных привычек, ухудшением экологии и быстрым ритмом жизни люди чаще стали задумываться о своем здоровье. Для того чтобы обезопасить и чувствовать себя комфортно в таких условиях нам помогают мобильные приложения. С помощью них мы можем контролировать все события, происходящие с нами в течении дня и стремиться к соблюдению здорового образа жизни.

Питание является важной составляющей здорового образа жизни человека. С питанием связаны все жизненно важные функции организма. Оно — источник развития клеток и тканей, их постоянного обновления, насыщения человека энергией. Неправильное питание — как недостаточное, так и избыточное — способно нанести огромный вред здоровью человека в любом возрасте. Это

можно заметить в снижении работоспособности, быстрой утомляемости, снижении уровня физического и умственного развития, неспособности оказывать сопротивление воздействию неблагоприятных факторов окружающей среды, преждевременном старении, и даже сокращении продолжительности жизни.

В нашем мире очень сложно уследить за потребляемой нами пищей, поэтому для облегчения процесса многие начали прибегать к помощи современных технологий, а именно использовать мобильные приложения.

Сейчас начинают становиться популярными приложения для обеспечения здорового образа жизни, которые позволяют контролировать вес, питание и физическую активность в течении дня. Зная это, актуально разработать свое мобильное приложение.

Цель выпускной квалификационной работы – изучить среду разработки Android Studio и разработать мобильное приложение.

Задачи выпускной квалификационной работы:

- на основе анализа литературы определить основные понятия и классификацию мобильных приложений;
- рассмотреть инструментальные среды разработки мобильных приложений;
- выявить достоинства и недостатки сред разработки;
- произвести проектирование архитектуры приложения;
- реализовать основные возможности Android Studio на конкретном примере.

1 Теоретические аспекты разработки мобильного приложения

1.1 Мобильные операционные системы

1.1.1 Android OS

Android - самая популярная мобильная платформа в мире. Согласно последним опросам, в мире свыше миллиарда активных Android-устройств и их количество продолжает стремительно расти. Android – полнофункциональная платформа с открытым кодом на базе Linux, разрабатываемая компанией Google. Это мощная платформа разработки, включающая всё необходимое для построения современных приложений из кода Java и XML. Более того, построенные приложения могут устанавливаться на множестве различных устройств – телефонах, планшетах и не только. [4]

История бренда

Компания Android была основана Ником Сирсом, Энди Рубином, Крисом Уайтом и Ричем Майнером в октябре 2003 года в городе Пало-Альто штата Калифорния. В начале своего пути они планировали разрабатывать операционную систему для цифровых камер, но быстро поняли, что данный рынок не имеет большого потенциала и сменили курс направления компании на создание мобильной операционной системы, которая смогла бы конкурировать с Symbian и Microsoft Windows Mobile популярные в те годы.

Финансирование компании происходило за деньги Энди Рубина, но вскоре его ресурсы закончились, и он не смог поддерживать фирму, тогда ему согласился помочь его друг, который передал 10 000 долларов, ничего не прося взамен, даже стать совладельцем компании.

Google, увидев потенциал маленькой компании, в 2005 году совершила сделку, сумма которой составила 50 миллионов долларов. Оставив основных сотрудников в рамках сделки, компания продолжила разработку, а Рубин был её руководителем. Google в свою очередь, предлагая взаимовыгодное

сотрудничество производителям смартфонов, начала продвигать компанию на массовом рынке.

Операционная система была изначально разработана для работы на смартфонах, имеющих физическую клавиатуру, но после выхода iPhone в 2007 году Google сосредоточилась на новом веянии рынка и изменила свои планы. В следствии этого Google объявили о поддержке сенсорной клавиатуры, хоть и сомневались в удобстве управления данным способом.

5 ноября 2007 года был представлен альянс Open Handset Alliance, эта дата стала одной из ключевых в процессе развития операционной системы. В обязанности этого объединения входит разработка приложений для платформы Android, внедрение инноваций в технологии, применяемые для разработки мобильных гаджетов, а также повышения удобства работы с устройствами Android при одновременном снижении затрат на производство и снижения цен для потребителей. В ходе своей работы альянс объединял производителей устройств, производителей чипсетов и значимых беспроводных операторов, а, помимо самой Google, в его состав вошли Motorola, HTC, Samsung, Qualcomm и T-Mobile.

HTC Dream, также известный как T-Mobile G1 был первым смартфоном, работающим на операционной системе Android. Смартфон имел необычный вид, хотя из-за большого многообразия различных форм в те годы, людей сложно было удивить. Он был представлен 23 сентября 2008 года.

Версия операционной системы для первого смартфона имела множество недостатков, но воодушевленная команда разработчиков не прекращала работу. Регулярно выпускали новые версии с исправление ошибок. Каждое крупное обновление получало название в честь какого-то десерта. Названия выбирались так, что первая буква в названии новой версии обязательно шла по алфавиту после первой буквы предыдущей версии. Во время анонса версии Android 4.4 (KitKat) Google объяснила выбор стратегии названия версий в честь десертов тем, что ее устройства делают жизнь пользователей очень сладкой. Так, первые версии Android получили название Cupcake, Donut и Eclair.

В таблице представлена статистика распространения версий на сентябрь 2018 года (таблица 1).

Таблица 1 – Статистика распространения версий:

Название	Версия	Год	Доля
Gingerbread	2.3.3 - 2.3.7	2010	0,2 %
Ice Cream Sandwich	4.0.3 - 4.0.4	2011	0,3 %
Jelly Bean	4.1.x	2012	1,1 %
	4.2.x	2012	1,5 %
	4.3	2013	0,4 %
KitKat	4.4	2013	7,6 %
Lollipop	5.0	2014	3,5 %
	5.1	2015	14,4 %
Marshmallow	6.0	2015	21,3 %
Nougat	7.0	2016	18,1 %
	7.1	2016	10,1 %
Oreo	8.0	2017	14,0 %
	8.1	2017	7,5 %
Pie	9.0	2018	< 0,1%
Q	10.0	2019	< 0,0%

В 2010 году Google представила серию устройств, которая получила название Nexus, по её словам, это сыграло важную роль в развитии Android. Задачей этой серии было внедрение аппаратных стандартов и новых версий программного обеспечения. В ходе работы над устройствами Nexus Google начала сотрудничество с крупными компаниями на рынке для разработки новых версий и устройств на Android. [11]

Начиная с 2009 года, Android начали устанавливать не только на смартфоны и планшеты, но и на другие устройства. Так, в массовое производство была запущена первая фоторамка на Android. Позже, в 2011 году итальянской

компанией Blue Sky были представлены умные часы G'm Watch которые так же работали под управлением Android ОС. Далее под управлением этой операционной системой была представлена первая в мире фотокамера, произведенная компанией Nexus. В линейке Nexus кроме смартфонов и планшетов представлен и медиаплеер Nexus Q, который также работает на ОС Android.

Главными особенностями операционной системы являются:

- кастомизация;
- огромный выбор устройств;
- поддержка двух сим – карт;
- простота обмена данных между устройствами.

Доля рынка

По данным исследовательской компании Canalys, доля Android на мировом рынке смартфонов составила 2,8% ко второму кварталу 2009 года. В четвертом квартале 2010 Android стала самой популярной операционной системой для смартфонов с показателем 33% всего рынка обогнав при этом Symbian. В третьем квартале 2011 года по данным Gartner этот показатель увеличился до 52,5%.

Дальше — больше! Популярность операционной системы росла с каждым днем. В 2011 году Google поделилась статистикой, что 400 000 устройств активируются каждый день в мае, а к июлю этот показатель составил 550 000 Android-устройств в день. К сентябрю 2012 количество активированных устройств составило 500 миллионов с показателем 1,3 миллиона активаций ежедневно, а доля платформы достигла 75% в мире. Запись о миллиарде активированных устройств была получена к сентябрю 2013 года. Далее, рост шел уже меньшими темпами, и в 2015 году доля составляла 76%. В целом же доля рынка Android зависит от местоположения. Например, в Китае широкое распространение платформа получила благодаря подъему собственных брендов.

Лидером по продажам устройств с этой операционной системой является Samsung, доля мирового рынка которого составляет 52,5% на конец 2016 года. В

след за ним идут такие компании как Huawei, LG, Lenovo, HTC, Sony, Asus, Xiaomi и Alcatel.

Говоря о магазине приложений Google Play, известно, что по состоянию на май 2013 года приложения были установлены 48 миллиардов раз, в мае 2016 уже 65 миллиардов раз. Всего в магазине содержится 3,2 миллиона приложений по состоянию на август 2018 года.

Анализируя данные StatCounter, который берет во внимание только устройства с использованием интернет-трафика, с августа 2013 года Android является самой распространенной мобильной операционной системой в мире. При этом в таких местах, как Саудовская Аравия, Южная Африка и Индия, объем трафика, получаемый устройствами на базе Android, выше, чем с использованием других платформ, включая их настольные версии.

1.1.2 iOS

iOS (до 24 июня 2010 года – iPhone OS) – операционная система от компании Apple, выпускаемая исключительно для устройств данной компании. Впервые представлена Стивом Джобсом 9 января 2007 года одновременно с первой моделью iPhone на выставке-конференции Macworld Conference & Expo и выпущена в июне того же года. Позже была разработана версия для iPad, Apple TV, Apple Watch.

Особенностями данной платформы являются:

- долгая поддержка. Apple регулярно обновляет свои устройства и это происходит на протяжении не менее 4 лет, сейчас этот период увеличивается. iPhone 5s, выпущенный в 2013 году до сих пор поддерживает последнюю версию операционной системы. Android – устройства не могут этим похвастаться так как производители могут не только не поддерживать их короткий период, но не поддерживать вообще;
- оптимизация системы под конкретное «железо» устройства. Даже самые первые версии после релиза с какими – либо ошибками работают плавно и без рывков. Из-за большого количество производителей с рядом собственных

настроек Android-устройствам это не доступно, так как они все оптимизированы по-разному;

- безопасность. Система от Apple не просто так является одной из самых безопасных. Это закрытая система. Нельзя просто так установить приложение из интернета, они доступны только в проверенном магазине приложений AppStore. Поэтому и отпадает необходимость в антивирусе. Так же в случае кражи злоумышленник не сможет украсть личные данные так как даже после сброса всех настроек устройство будет заблокировано;

- экосистема. Между устройствами компании Apple существует тесная связь. С любого устройства можно загрузить фото в общее облако, при написании текста на iPhone его редактирование можно продолжить на iPad и наоборот, так же существует одна связка паролей и так далее.

1.1.3 Windows Phone

Впервые система от Microsoft была представлена в Барселоне на Mobile World Congress 15 февраля 2010 года Стивом Балмером. В разработке мобильных устройств принимали участие Lg, Sumsung, HTC, Acer и многие другие. На конференции Nokia World были представлены первые смартфоны на базе этой платформы: Nokia Lumia 800 и Nokia Lumia 710.

Особенности операционной системы:

- плавность работы и необычный интерфейс. Рабочее пространство состоит из плиток, плавно перемещающихся по экрану и динамически отражающих информацию;

- безопасность. На устройство под управлением Windows Phone все приложения устанавливаются только из собственного магазина приложений, а установка стороннего ПО без перепрошивки невозможна;

- оптимизация. Приложения проходят строгую модерацию в следствии чего работают очень быстро и плавно;

– недостаточное количество приложений и ограниченность их функциональности. Существует большой процент приложений, чей функционал не дотягивает до аналогов на Android и iOS.

В октябре 2017 года, исполнительным директором Microsoft было заявлено о прекращении производства новых устройств и обновлений на Windows Phone, а 10 декабря 2019 года поддержка будет полностью приостановлена.

1.2 Основные конструкции мобильного приложения

Мобильное приложение – это специальный пакет, который пользователь может получить, установив со специального рынка приложений (Google Play, App Store). Мобильные приложения – это программы, поэтому они создаются под определенную операционную систему (iOS, Android, Windows Phone и т.д.).

Мобильные приложения – программное обеспечение для мобильных устройств. Их возможности частично зависят от возможностей мобильных устройств, частично от возможностей, заложенных разработчиком приложений. Используя мобильные приложения, пользователи получают услуги и товары, узнают новости, работают и отдыхают. Владельцы приложений продают информацию, услуги, товары, зарабатывают на рекламе, повышают мобильность рабочих мест, управляют бизнес-процессами. [3]

В ходе работы под мобильным приложением будем понимать, что это специальное программное обеспечение, разработанное под определенную мобильную платформу (iOS, Android, Windows Phone и др.) и выполняющее конкретные задачи. Оно предназначено для использования на смартфонах, планшетах, умных часах и других мобильных устройствах.

Мобильные приложения пишутся на языках программирования высокого уровня, а затем компилируются в машинный код операционной системы для получения максимальной производительности. Современные мобильные устройства оснащены дополнительными модулями, такими как фотокамеры, акселерометры, гироскопы, модули беспроводной связи дающими уникальные возможности для расширения функционала приложения.

Среди мобильных приложений, по процессу разработки, можно выделить три категории:

– веб-приложения – это мобильные версии привычных нам сайтов, адаптированные специально для смартфонов. Они являются кроссплатформенными и доступны каждому пользователю. Это самый часто встречающийся тип приложений;

– гибридные приложения – это приложения, включающие в себя функции нативных и веб-приложений, т.е. они имеют возможность использовать ПО смартфона и могут быть установлены на любую платформу, быть кроссплатформенным. Они могут быть установлены на устройство через магазин приложений и в то же время независимо обновлять информацию. Гибридные приложения требуют подключения к интернету, поскольку веб часть обновляется через интернет. Это, наверное, самый популярный способ построения мобильных приложений, так как у него органическая среда распространения, но разработка происходит быстрее и дешевле, чем в случае с нативными приложениями, так как, хотя оболочка и написана на «родном» языке программирования, «начинка» может быть написана в том или ином объеме на html5. Пользователь же скорее всего не заметит разницу между нативным приложением и гибридным;

– нативные приложения — это приложения, которые мы можем получить только из магазина приложений. Основным различием является то, что они разрабатываются под конкретную операционную систему (Windows Phone, Android, IOS и т.д.). Нативные приложения написаны на «родных» языках программирования, в следствии чего имеют высокую скорость работы и максимальный набор функций. В зависимости от назначения приложения, они могут как требовать, так и не требовать наличие интернет - соединения.

Классификация мобильных приложений по роду деятельности:

– бизнес - приложения. Данная категория приложений является наиболее интересной для инвесторов. Такие приложения помогают в расчетах, в офисной

работе, обмене служебными данными. Также обеспечивают доступ к банковским счетам, интернет – магазинам и платежным системам.;

– контентные приложения. Основные задачи данных приложений направлены на получение контента, что делает их очень популярными. С помощью таких приложений мы можем: просматривать фотографии и фильмы, слушать музыку, читать журналы и книги. В свою очередь к ним можно отнести и информационные приложения, которые предоставляют информацию о расписания городского транспорта, свежих новостях, о погоде, или специализированные приложения, разработанные для определенных событий, например к олимпиаде или чемпионату мира;

– мобильные социальные сети. С развитием мобильного интернета и изменения аппаратных особенностей мобильных устройств, данная категория является самой популярной, увеличивая свою аудиторию с каждым днем. С помощью данных приложений пользователи могут поддерживать связь друг с другом в любой точке мира;

– мобильные игры – это один из востребованных секторов мобильных приложений. [5]

Приложения также подразделяются по виду монетизации:

- бесплатное приложение с рекламой внутри приложения;
- бесплатное приложение со встроенными покупками;
- бесплатное приложение с платной подпиской;
- платное приложение, реализуемое посредством продажи в магазине.

Самыми распространенными магазинами приложений являются AppStore и Google play, именно они содержат основной объем приложений на рынке. [8]

1.3 Этапы разработки мобильного приложения

Разработка мобильного приложения состоит из следующих взаимосвязанных этапов:

1. Разработка технического задания. В первую очередь необходимо определить, какие потребности пользователей и клиента должно решать

приложение, а также сформулировать его основные задачи. Этому этапу уделяется особое внимание: от задания зависят технические особенности будущего продукта.

2. Прототипирование. Для того чтобы понять, как пользователь будет работать с мобильным приложением, создается графическая карта взаимодействия между различными экранами программы. На этом этапе осуществляется проработка практически всего функционала мобильного приложения. На стадии проектирования UI специалисты определяют принцип работы приложения, размещение функций и кнопок на каждом из экранов. На этом этапе:

- отрабатывается функционал приложения;
- разрабатываются схемы экранов приложения;
- продумывается связь экранов приложения и переходов по ним.

3. Дизайн мобильного приложения. Создается дизайн всех экранов будущего приложения и отрисовываются различные состояния для всех сценариев пользования. Создаются внутренние кнопки и иконки, а также все остальные графические элементы. Как правило отрисовка дизайна мобильного приложения предполагает юзабилити-исследования для того, чтобы убедиться, что разработанная дизайн-концепция максимально проста и удобна и поможет пользователям максимально быстро решать поставленные задачи.

4. Тестирование. Внутреннее тестирование как на эмуляторах, так и на реальных устройствах является обязательным на всех этапах разработки. Целью тестирования является проверка соответствия полученного приложения с предполагаемым на этапе прототипирования, проверка взаимодействия приложения с программной и аппаратной частью устройств.

5. Создание предрелизной версии. После успешного прохождения тестирования, данная версия становится предрелизной и её предстоит добавить в магазин приложений: Google Play, магазин приложений Windows Phone, App Store (в зависимости от того, для какой операционной системы создавалось приложение) или любой аналогичный сервис для дистрибуции приложений.

б. Добавление приложения в магазин. Последний этап работы – загрузка приложения на проверку в один из магазинов приложений. [10]

1.4 Языки программирования, используемые в мобильной разработке

1.4.1 Java

Одним из самых популярных языков для разработки приложений по под операционную систему Android является Java. Применение Java в создании мобильных приложений является неслучайным выбором, так как это открытый, свободный и мощный язык. Java применяется в разработке дополнительного функционала веб – серверов, полномасштабных корпоративных приложений, проектируют приложения для различных устройств помимо смартфонов и планшетов, и это только малая часть возможностей и области применения этого языка. Java – кроссплатформенный язык, с его помощью программисты могут создавать приложения, не обращая внимание на аппаратные особенности устройства, а с использованием Android API и иных вспомогательных средств разработчики способны быстро изучить проектирование приложений для операционной системы Android.

Главным преимуществом операционной системы Android является открытость. Данная платформа распространяется на свободной основе и основана на использовании открытого кода. Благодаря этому, разработчики могут получить доступ к исходному коду Android и на его основе реализовывать новые функции приложения.

За счет открытости кода более десяти OEM – производителей, имеющих лицензию на выпуск Android – устройств, производят гаджеты взяв за основу платформу Android, создавая тем самым конкуренцию и ускоряя внедрение инноваций.

С помощью языка Java, который относится к объектно-ориентированным языкам программирования, разработчики получают доступ к большому

количеству мощных библиотек, ускоряющих написание приложения. Создание пользовательского графического интерфейса является контролируемым событием. С его помощью можно собрать интерфейс из заготовленных элементов, в виде текстовых полей, кнопок и переключателей изменяя их размер, перетаскивая в любые части экрана и добавляя подписи. [9]

Другие преимущества Java:

Безопасность: для проверки достоверности данных используется методы, основанные на шифровании с открытым ключом.

Динамика: Java легко адаптируется под изменяющиеся условия и является более динамичным языком в отличии от C и C++. Программы могут выполнять обширное количество во время обработки информации, которая может быть использована для проверки и разрешения доступа к объектам на время выполнения.

Портативность: архитектурно-нейтральный и не имеющий зависимости от реализации аспектов спецификаций — все это делает Java портативным. Компилятор в Java написан на ANSI C с чистой переносимостью, который является подмножеством POSIX. [14]

Интерпретированность: Java байт-код переводится на лету в машинные инструкции и нигде не сохраняется, делая процесс более быстрым и аналитическим, поскольку связывание происходит как дополнительное с небольшим весом процесса.

Высокая производительность: введение Just-In-Time компилятора, позволило получить высокую производительность.

Многопоточность: можно писать программы, которые могут выполнять множество задач одновременно.

1.4.2 Kotlin

Язык был официально представлен в мае 2017 года на Google I/O и позиционируется Google как второй официальный язык программирования под Android после Java, только чуть более простой для понимания. Для его

использования необходимы знания Java, чтобы понимать его принципы работы, общую структуру языка и особенности. [15]

Kotlin совместим с Java и не вызывает снижения производительности и увеличения размера файлов. Отличие от Java в том, что он требует меньше служебного, так называемого boilerplate-кода, поэтому более обтекаемый и лёгкий для чтения. Его создателям удалось избежать nullpointexceptions, а компиляция больше не прерывается из-за мелочей вроде забытого знака «;».

Плюсы:

- лаконичность языка. Это язык с идеологией, что код читается как текст на английском языке;

- расширения, касты, именованные аргументы и множество других функций языка;

- язык поддерживается Google и уже существует несколько стабильных версий;

- полная совместимость с Java. Классы на котлине можно использовать в Java и наоборот и все это в одном проекте;

- Kotlin это не академический язык, который придумали в научном круге для специфичных задач. Это индустриальный язык, созданный разработчиками специально для разработчиков приложений Android;

- для того, чтобы подключить Kotlin не нужно переписывать код. Можно делать только новые части на котлине

Минусы:

- Маленькое сообщество и слабое продвижение.

1.4.3 C#

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. [13]

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), итераторы, атрибуты, события, обобщённые типы и методы, делегаты, свойства, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Плюсы:

- средний порог вхождения. Синтаксис похожий на C, C++ или Java облегчает переход для других программистов. Для новичков это также один из самых перспективных языков для изучения;

- добавлено функциональное программирование (F#);

- кроссплатформенность. Благодаря Xamarin на C# теперь можно писать под Android и iOS.

- множество конструкций для облегчения написания и понимания кода;

- большое сообщество программистов.

Минусы:

- ориентированность, в основном, только на .NET (на Windows платформу);

- бесплатность только для небольших компаний, учащихся и программистов-одиночек.

1.5 Среды разработки мобильных приложений

1.5.1 Android Studio

В 2013 году состоялась конференция разработчиков Google, на которой была представлена официальная среда разработки мобильных приложений для платформы Android - Android Studio. Она разработана на основе IntelliJ IDEA, имеет встроенный Android SDK и распространяется бесплатно.

Среда разработки Android Studio признана самой популярной и удобной платформой для разработки и тестирования приложений на Android. Google

разработала это программное обеспечение с огромным набором инструментов для упрощения процесса разработки мобильных приложений. По сравнению с предыдущим способом создания приложений в среде разработки Eclipse, процесс стал проще и динамичнее. Это произошло благодаря появлению возможности отображать главные рабочие элементы в самой структуре создаваемого приложения, что позволило эффективнее подойти к разработке. [2]

Разработчики выделяют возможность просмотра в реальном времени всех изменений проекта. Так же среда позволяет разрабатывать приложения для разных версий Android.

Благодаря новым встроенным инструментам и доработке интерфейса пользователя процесс взаимодействия со средой стал в разы удобнее Eclipse. В следствие чего написание кода стало более рационально, что позволило легко ориентироваться при разработке больших по объёму проектов. Есть функция перетаскивания функциональных элементов в самой программе, что упрощает редактирование информации.

Достоинства Android Studio:

- поддержка работы с несколькими языками;
- подсветка синтаксиса;
- возможность разработки приложений для планшетов, телевизоров, часов и других умных устройств;
- возможность быстро локализовать приложения;
- тестирование созданного приложения в эмуляторе;
- большая библиотека с готовыми компонентами и шаблонами;
- рефакторинг уже готового кода;
- предварительная проверка приложения на наличие ошибок;
- наличие литературы на русском языке;
- отправка push-сообщений для приложений через любые облачные сервисы сразу на устройства под Android.
- понятный интерфейс;

- удобная структура проекта;
- доступна разработка под разные разрешения экрана;
- поддержка отслеживания эффективной работы рекламных объявлений;
- наличие инструментов для повышения качества проектов и монетизации.

Инструменты Android Studio:

- AVD Manager – эмулятор для запуска приложений;
- Android Device Monitor – производит мониторинг виртуального или физического устройства во время работы;
- Android Debug Bridge - инструмент командной строки, с помощью которого можно копировать файлы с устройства и обратно, удалять и устанавливать приложения;
- FlowUp – помогает отслеживать общую производительность приложения;
- плагин Gradle для сборки приложений;
- JRebel – значительно сокращает время сборки приложения;
- Android Asset Studio – генерация различных типов иконок;
- Android Wi-Fi ADB – позволяет запускать приложение на устройстве используя Wi-Fi;
- FindBugs – поиск багов прямо в процессе разработки;
- Drawble Optimizer – оптимизация изображений.

Одним из основных преимуществ Android Studio является система сборки Gradle, которая интенсивно развивается компанией Google. Gradle обладает такими полезными функциями как:

- создание простых задач в виде скрипта;
- настройка хранилища ключей;
- создание различных вариантов сборки вашего приложения;
- возможность управления зависимостями и автоматически подгружать их. [12]

Недостатки Android Studio:

- достаточно медленная сборка проекта без дополнительных инструментов;

- высокие системные требования к ПК;

- медленная работа эмуляторов.

1.5.2 Eclipse IDE

Eclipse долгие годы уверенно держал пальму первенства по популярности среди Java IDE. Эта среда полностью бесплатная, с открытым исходным кодом, написанным преимущественно на Java. Тем не менее, её модульная архитектура позволяет использовать Eclipse и с другими языками. Проект Eclipse, инициированный IBM, появился в 2001 году. Портативность Java помогает Eclipse быть кроссплатформенной средой: эта IDE работает на Linux, Mac OS X, Solaris и Windows. Своей производительностью Eclipse обязана JVM. Eclipse работает довольно медленно. Даже сегодня она кажется медлительной, особенно если нацепить на неё много плагинов. Часть расходов ресурсов Eclipse можно отнести на счёт её встроенного инкрементного компилятора, который запускается всякий раз при загрузке файла или обновлении кода. Полезная штука, именно она ловит ошибки при вводе текста. Независимо от сборки, проект Eclipse поддерживает модель контента, которая содержит информацию об иерархии типов, ссылок и объявлениях Java-элементов. Последняя версия Eclipse носит имя 2019-03. Он содержит минимальную Eclipse SDK, а плагины добавляются по требованию. К слову, работа с плагинами в этой IDE — не для слаонервных. Сторонние плагины часто конфликтуют между собой, хотя в их официальной спецификации об этом ничего не сказано.

В настоящее время для Eclipse написано более 1700 плагинов, официальных и неофициальных. Плагины Eclipse, поддерживают более 100 языков программирования и почти 200 фреймворков для разработки приложений.

Eclipse практически всегда предлагает несколько способов решения той или иной задачи. Например, можно просматривать код с помощью ракурса

просмотра Java (Java browsing perspective). Специальный поиск Java позволяет найти объявления, ссылки и вхождения Java-пакетов, типов, методов, полей. Вы также можете использовать быстрый доступ к поиску и предпросмотр. Распространенные паттерны кода можно сгенерировать из шаблонов кода. Рефакторинг Java в Eclipse, поддерживает 23 операции, начиная от общепринятых операций по переименованию и заканчивая менее очевидными преобразованиями. Eclipse, поддерживает отладку как локально, так и удаленно, при условии, что используется JVM, которая поддерживает удаленную отладку. Отладка довольно стандартна: определяются контрольные точки, а затем просматриваются переменные в закладке отладки. Конечно, можно пошагово выполнять код и вычислять выражения.

У Eclipse — обширнейшая база документации самого разного возраста, ценности и полезности. Обнаружить несоответствующую текущей версии картинку в инструкции, например, с устаревшим интерфейсом и расположением кнопок — обычное дело для этой IDE. К сожалению, проблема запоздалого обновления документации очень характерна для любых проектов с исходным кодом.

1.5.3 Visual Studio и Xamarin

Xamarin — это платформа разработки мобильных приложений для создания нативных приложений iOS, Android и Windows из общего кода C# или .NET, которая позволяет многократно использовать между платформами от 75 % до почти 100 % кода. Приложения, написанные с помощью Xamarin и C#, имеют полный доступ к интерфейсам API базовой платформы и возможность создавать нативные пользовательские интерфейсы, а также компилировать код в машинный, поэтому влияние на производительность во время выполнения является незначительным.

Основной инструмент, с помощью которого компания Xamarin предлагает разрабатывать приложения для разных операционных систем, — это Xamarin Studio, которая работает на Windows или Mac OS X. Также можно вести разработку благодаря расширениям в среде Microsoft Visual Studio.

Для выполнения приложений в Android используется виртуальная Java-машина Dalvik. Нативные Java-приложения компилируются в промежуточный байт-код, который интерпретируется Dalvik'ом в команды процессора во время исполнения программы (Just-in-time (JIT) компиляция). Xamarin в свою очередь компилирует C#-код в промежуточный байт-код для виртуальной машины Mono, которая упаковывается в приложение. При запуске Xamarin-приложения две виртуальные машины Mono и Dalvik работают параллельно, обмениваясь данными через JNI.

Здесь нужно заметить, что приложение сможет работать не только под iOS и Android, но и под Windows Phone, т.к. на Windows также будет доступен Xamarin.Forms API ну, и конечно, будет доступ к оригинальной платформе.NET.

Результатом скомпилированного Xamarin-приложения являются .app-файл для iOS, .apk для Android и .appxbundle для UWP. Эти файлы неотличимы от пакетов приложений, созданных стандартными для платформ IDE, и развертываются точно таким же образом.

1.6 Базы данных, используемые в мобильных приложениях

Появление и относительно широкое распространение в начале 60-х годов XX века запоминающих устройств достаточно большой емкости с возможностью доступа к произвольным участкам памяти — магнитных дисков, — открыло широкие возможности для создания сложных структур долговременно хранимых данных. Высокая скорость обновления небольших объемов данных (доли секунды) создала условия для создания приложений, способных функционировать в режиме оперативной работы (on-line). В отличие от систем предшествующих поколений, время ответа стало измеряться не сутками, а секундами или долями секунды. Эти возможности, однако, привели к существенному усложнению кода приложений и, как следствие, к удорожанию их разработки и снижению надежности. В связи с этим появилась идея централизации функций управления данными, которая привела к появлению

систем, предоставляющих приложениям услуги по обработке данных. Такие системы получили название систем управления базами данных (СУБД).

Система управления базами данных (СУБД) — это программный комплекс, обеспечивающий централизованное хранение данных и предоставляющий приложениям услуги по обработке данных.

Совокупность данных, хранимых под управлением СУБД, называется базой данных. В оригинальном английском варианте словосочетание data base означает «основание, состоящее из данных». Этот смысл несколько искажается в русском словосочетании «база данных». На самом деле это — фундамент, на котором строятся приложения и который состоит из данных. Действительно, данные (а следовательно, база данных) являются очень существенной частью практически любой информационной системы. [7]

Одним из основных требований к системам управления базами данных является наличие высокоуровневых средств выполнения запросов. В системах, реализующих реляционную модель данных, в качестве такого средства используется язык SQL – язык структурированных запросов. Фактически этот язык содержит полный набор операций, необходимых для выполнения любых действий с базой данных. [1]

Для работы с мобильными приложениями требуется специальная библиотека – SQLite. Эта бесплатная библиотека, которая организует работу с автономными базами данных, не нуждающихся в серверах и не требовательных к ресурсам. SQLite читает и записывает данные в обычные файлы. Полная база данных SQL с несколькими таблицами, индексами, триггерами и представлениями, компактно хранится в одном файле. Все android-устройства поддерживают работу с SQLite.

С учетом вышесказанного, для решения поставленной в выпускной квалификационной работе задачи с учетом всех достоинств и недостатков было решено разработать приложение для операционной системы Android используя среду разработки Android Studio и язык программирования Java.

2 Проектирование мобильного приложения «Let's Cook»

В этой главе опишем проектирование создаваемого приложения и определим основные требования FURPS (Functionality, Usability, Reliability, Performance, Supportability).

2.1 Требования

Функциональные требования:

У пользователя есть возможность:

- искать рецепт по введенным продуктам.
- добавлять понравившиеся рецепты в избранное.
- создавать свой рецепт.
- редактировать рецепт.
- удалять рецепт.
- просматривать рецепты по категориям.
- просматривать все рецепты.
- считать количество полученных калорий от приготовленных блюд за день.
- посчитать индивидуальную суточную норму калорий.

Требования к удобству использования:

Для достижения функциональных требований должно быть создано меню для быстрого доступа к разделам.

Требования к надежности: программное средство (ПС) должно иметь свойство отказоустойчивости.

Требования к производительности: ПС должно иметь быструю скорость отклика и эффективное использование ресурсов.

Требования к поддержке: в приложении предусмотрена обратная связь с разработчиком. Разработчик должен выпускать частые обновления с исправлением ошибок и реализацией дополнительных функций.

2.2 Варианты использования

По требованиям составим use-case диаграмму для наглядного отображения возможностей.

Приложение «Let's cook» будет помогать следить за питанием человека в течении дня. Пользователь будет иметь доступ к 7 разделам:

- приготовить сейчас;
- избранное;
- мои рецепты;
- категории;
- все рецепты;
- приготовленные за день блюда;
- расчет суточной нормы калорий;

Составим use-case диаграмму вариантов использования (рисунок 1).

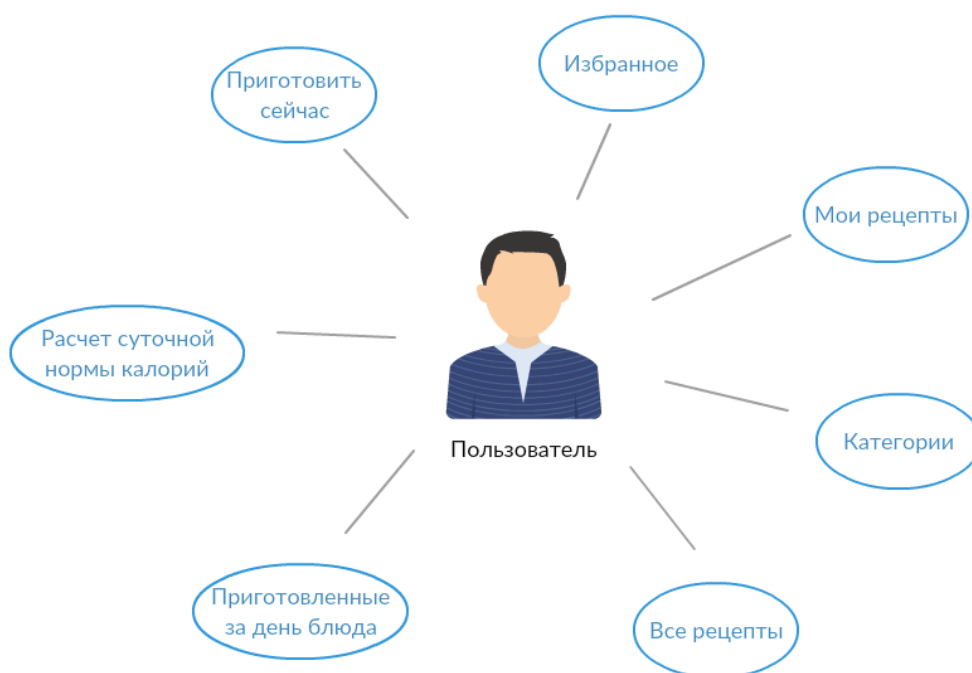


Рисунок 1 – Use-case диаграмма

Работая с базой данных в пункте «Мои рецепты», пользователь имеет возможности, отображенные на рисунке 2.

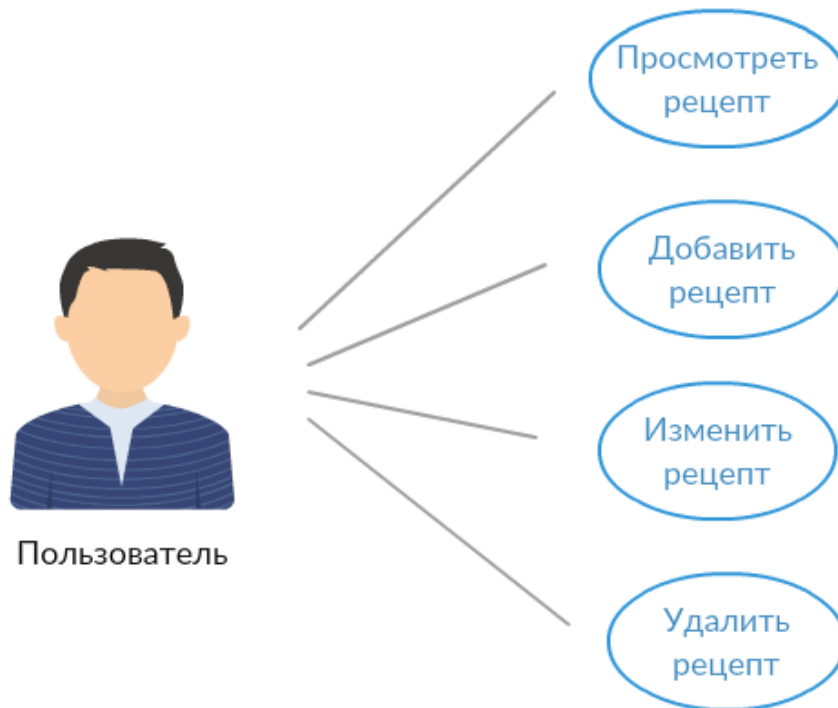


Рисунок 2 - Работа с базой данных

2.3 Структура базы данных

С помощью DB Browser for SQLite создадим базу данных recipes.db, которая будет состоять из трех таблиц:

- categoryTab;
- recipesTab;
- myrecipesTab;

Таблица categoryTab будет содержать название категорий и иметь следующие поля (рисунок 3):

- `_id` – номер категории;
- `nameCategory` – название категории;



 <code>_id</code>	INTEGER
 <code>nameCategory</code>	TEXT

Рисунок 3 - Название и типы полей в таблице categoryTab

В таблице recipesTab будут храниться рецепты. Добавим следующие поля (рисунок 4):

- _id - уникальный номер;
- title – название блюда;
- products – список необходимых продуктов;
- description – описание приготовления блюда;
- calories – количество калорий;
- image – название картинки для блюда;
- category – номер категории, к которой относится блюдо;
- favorite – является избранным или нет;






 _id	INTEGER
 title	TEXT
 products	TEXT
 description	TEXT
 calories	INTEGER
 image	TEXT
 category	TEXT
 favorite	INTEGER

Рисунок 4 - Название и типы полей в таблице recipesTab

Таблица myrecipeTab будет аналогична предыдущей, но в ней будут содержаться рецепты, которые добавил сам пользователь.

3 Реализация приложения «Let's Cook»

Изучив все достоинства и недостатки различных сред разработки, выберем Android studio, так как она является официальной средой разработки мобильных приложений для Android и имеет множество встроенных функций и библиотек. В качестве языка программирования будем использовать Java.

Для начала создадим меню, в котором отобразим все доступные пользователю функции. При создании проекта, в качестве шаблона выберем Navigation Drawer Activity и затем добавим туда свои разделы. Меню будет состоять из фрагментов, входящих в MainActivity.

Activity — это элемент приложения, который выводит полученный результат на экран путем взаимодействия пользователя с устройством и осуществления тех или иных действий, к примеру пользователь может написать сообщение, посмотреть новости, сделать фотографию или набрать номер.

Фрагмент – объект контроллера, которому активность передает выполнение операций. Фрагменты используются для обеспечения гибкости и большей динамичности пользовательских интерфейсов. Для реализации фрагмента необходимо создать подкласс класса Fragment (или его существующего подкласса). Класс Fragment имеет код, во многом схожий с кодом Activity. Он включает методы обратного вызова, аналогичные методам операции, такие как onCreate(), onStart(), onPause() и onStop().

Добавим первый пункт в меню. Для этого создадим новый .java файл, который будет содержать публичный подкласс FragmentCookNow (рисунок 5).

```
public class FragmentCookNow extends Fragment {  
    public FragmentCookNow() {}  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_cook_now, container, attachToRoot: false);  
    }  
}
```

Рисунок 5 - Реализация фрагмента FragmentCookNow

Далее создадим файл `fragment_cook_now.xml`. С помощью него будет определен дизайн приложения на языке расширяемой разметки XML (eXtensible Markup Language).

В файле `MainActivity.java` свяжем эти два файла используя транзакции. Для транзакций (добавление, удаление, замена) будем использовать класс-помощник `android.app.FragmentTransaction`. При вызове из меню получим следующий код, отображенный на рисунке 6.

```
FragmentCookNow fragment = new FragmentCookNow();
FragmentTransaction fragmentTransaction = getSupportFragmentManager().beginTransaction();
fragmentTransaction.replace(R.id.frame, fragment, "FragmentCookNow");
fragmentTransaction.commit();
```

Рисунок 6 - Реализация фрагмента `FragmentCookNow` в `MainActivity`

Аналогично, для остальных пунктов. В итоге получим меню, состоящее из 7 фрагментов (рисунок 7).

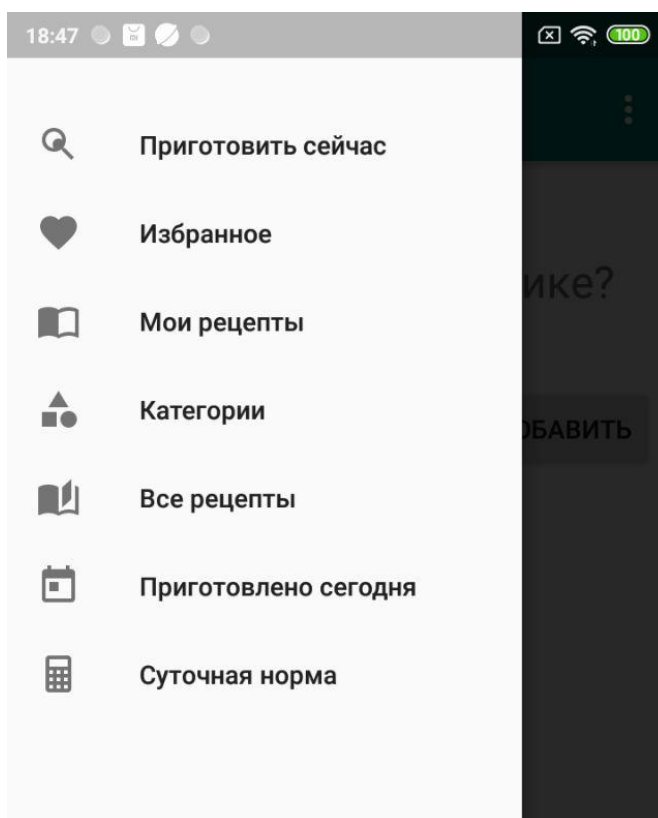


Рисунок 7 - Меню приложения

2. Приготовить сейчас

Приложение «Let's Cook» будет помогать экономить время при выборе блюда для приготовления. Основной его функцией будет подбор рецепта по имеющимся продуктам.

Принцип работы:

- вводим продукты, имеющиеся в холодильнике, нажимаем «Начать!»;
- приложение сверяет их с продуктами, содержащимися в рецептах;
- выводит на экран блюда с наибольшим совпадением продуктов;
- выбираем подходящее блюдо;
- выводится его рецепт.

Для работы с базой данных добавим её в проект и создадим класс DataBaseHelper в котором определим путь к базе, название таблиц и столбцов (рисунок 8).

```
private static String DB_NAME = "recipes.db";
private static final int SCHEMA = 1; // версия базы данных
static final String TABLE = "recipesTab"; // название таблицы в бд
// названия столбцов
static final String COLUMN_ID = "_id";
static final String COLUMN_TITLE = "title";
static final String COLUMN_PRODUCTS = "products";
static final String COLUMN_DESCRIPTION = "description";
static final String COLUMN_IMAGE = "image";
static final String COLUMN_CATEGORY = "category";
static final String COLUMN_FAVORITE = "favorite";
static final String COLUMN_CALORIES = "calories";
```

Рисунок 8 - Реализация подключения базы данных

Создадим XML – файл разметки. Добавим поле ввода продуктов и кнопку, по нажатии которой начинается поиск (рисунок 9). Будем вводить по одному продукту, добавляя его в список.

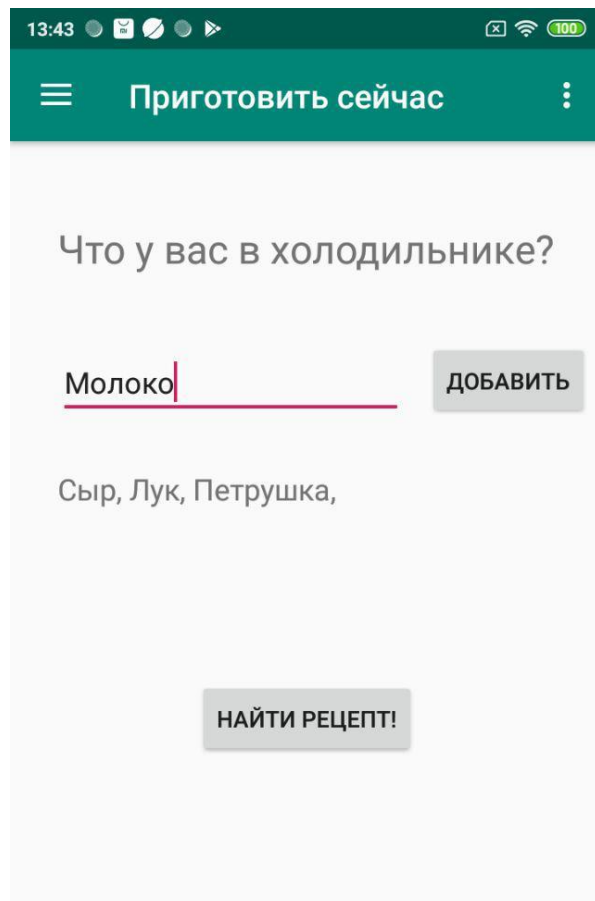


Рисунок 9 - Ввод продуктов для поиска

Поиск будет происходить следующим образом: берем первый продукт из списка введенных и сравниваем его с каждым продуктом в первом рецепте, если находим совпадение, то добавляем к счетчику совпадений +1, иначе переходим к следующему рецепту, аналогично с каждым введенным продуктом (рисунок 10).

```
for (int p = 0; p < kol_vo_prod; p++) {  
    for (int i = 0; i < row; i++)  
        for (int j = 0; j < rec_mas_prod[i].length; j++) {  
            if (ish_mas_prod[p] == rec_mas_prod[i][j]) {  
                k[i] = k[i] + 1;  
            }  
        }  
}
```

Рисунок 10 - Реализация поиска

Далее сортируем счетчик совпадений по возрастанию и выводим несколько первых, наиболее подходящих рецептов (рисунок 11).

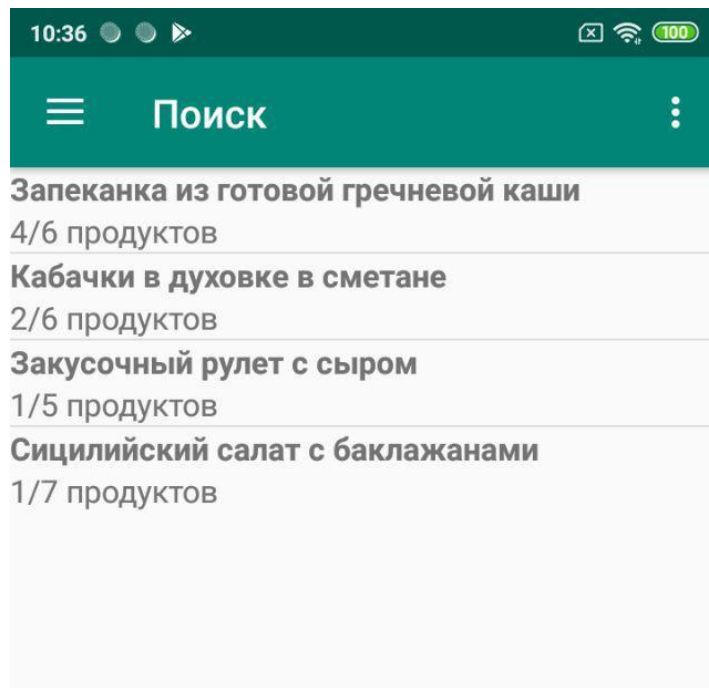


Рисунок 11 - Результат поиска

После выбора нужного нам рецепта, мы переходим на новый activity `RecipeDetail.java`, где уже подробно можем изучить рецепт, добавить в избранное, или приготовленное сегодня. Более подробно рассмотрим этот файл в пункте меню «Все рецепты».

3. Избранное

В этом разделе будут находиться понравившиеся пользователю рецепты. Для добавления будем использовать специальную кнопку `Floating Action Button`, которая отображается поверх всего контента (рисунок 12).

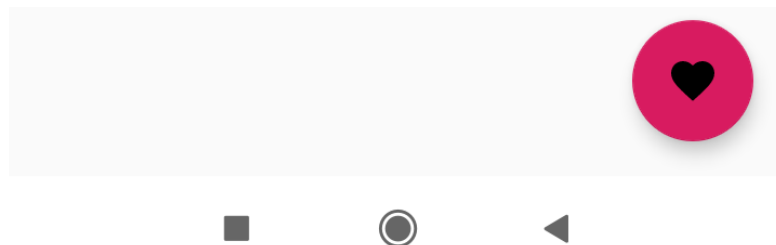


Рисунок 12 - Оформление кнопки `Floating Action Button`

При нажатии на кнопку, в поле столбца favorite в базе данных будет ставиться “1”. Для получения полного списка блюд воспользуемся следующим SQL – запросом:

```
SELECT * FROM myrecipeTab WHERE favorite = “1”
```

В результате получим список, отображенный на рисунке 13.

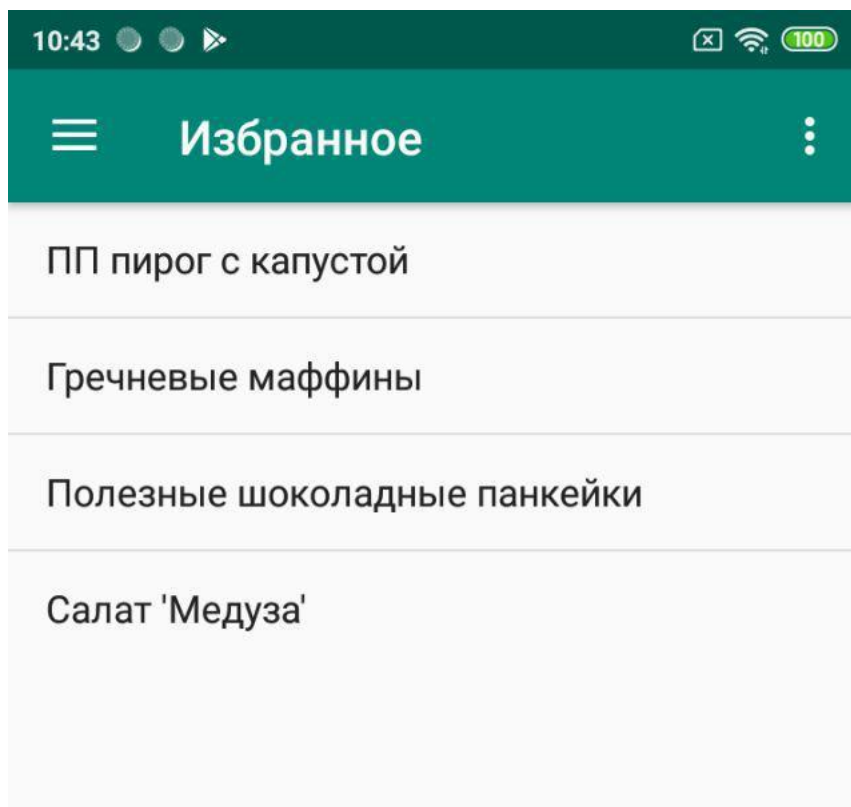


Рисунок 13 - Результат вывода избранных рецептов

4. Мои рецепты

В этом разделе будут храниться рецепты, добавленные самостоятельно. Для быстроты доступа вынесем их в отдельный пункт меню. Добавим в файл XML прокручиваемый список ListView для удобного отображения нескольких рецептов. Наш список сейчас пуст, чтобы его заполнить, сделаем возможность добавления новых блюд. Чтобы не загромождать интерфейс новыми кнопками будем использовать toolbar. Добавим первый пункт меню согласно рисунку 14:

```

<item
    android:id="@+id/action_settings"
    android:orderInCategory="100"
    android:title="Добавить новый рецепт"
    app:showAsAction="never" />

```

Рисунок 14 - XML реализация пункта меню

Также, нам понадобится функция редактирования и удаления рецепта, поэтому аналогично добавим ещё два пункта меню. В итоге получим меню с тремя функциями (рисунок 15).

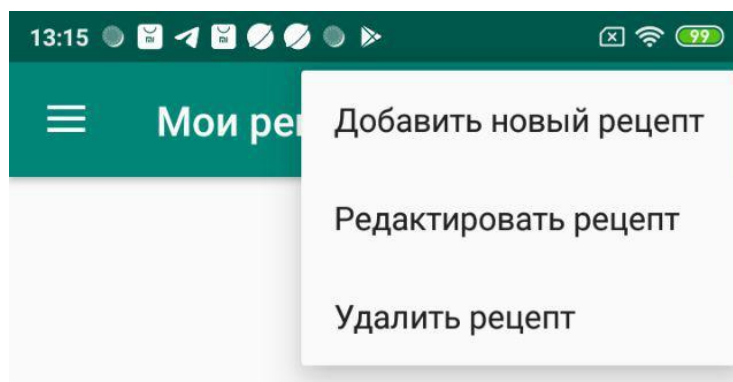


Рисунок 15 - Результат создания меню в toolbar

При нажатии на первый пункт меню будет выполняться код (рисунок 16), который перенесет нас на новую activity AddRecipe.

```

Intent intent = new Intent( packageContext: this, AddRecipe.class);
startActivity(intent);

```

Рисунок 16 - Реализация перехода на новый Activity

Далее определим интерфейс этой activity используя разметку XML (рисунок 18). После создания всех полей ввода приступим к редактированию java файла. Для добавления новых рецептов будем использовать таблицу `myrecipeTab`. Также создадим класс `RecipeClass` со свойствами эквивалентными полям таблицы, а именно `id`, `title`, `description`, `image`, `calories`, `category`, `favorite`. Создадим класс `DataBaseHelper` для определения полей новой таблицы (рисунок 17).

```
private static final String TABLE_RECIPE = "myrecipeTab";
private static final String RECIPE_ID = "ID";
private static final String RECIPE_TITLE = "title";
private static final String RECIPE_PRODUCTS = "products";
private static final String RECIPE_DESCRIPTION = "description";
private static final String RECIPE_IMAGE = "image";
private static final String RECIPE_CALORIES = "calories";
private static final String RECIPE_FAVORITE = "favorite";
```

Рисунок 17 - Реализация подключения базы данных

Создадим объект myrecipe класса RecipeClass и заполним его свойства, введенными пользователем данными.

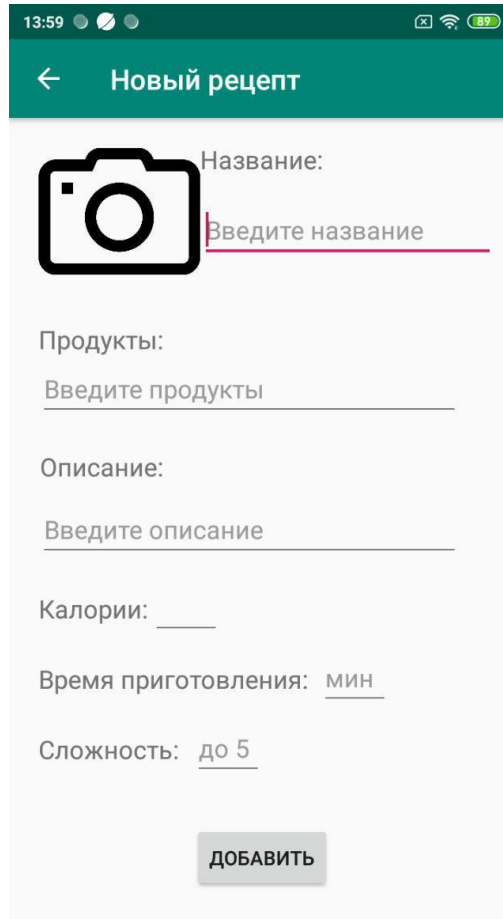


Рисунок 18 - Результат создания интерфейса для добавления новых рецептов

Далее добавим эти данные в базу данных (рисунок 19).


```

public void addNewRecipe(RecipeClass myRecipe){
    ContentValues values = new ContentValues();
    values.put(RECIPE_TITLE, myRecipe.getTitle());
    values.put(RECIPE_PRODUCTS, myRecipe.getProducts());
    values.put(RECIPE_DESCRIPTION, myRecipe.getDescription());
    values.put(RECIPE_IMAGE, myRecipe.getImage());
    values.put(RECIPE_CALORIES, myRecipe.getCalories());
    values.put(RECIPE_FAVORITE, myRecipe.getFavorite());
    dbDatabase.insert(TABLE_RECIPE, nullColumnHack: null, values);
}

```

Рисунок 19 - Реализация добавления введенных значений в базу данных

Результат отображен на рисунке 20.



Рисунок 20 - Результат добавления нового рецепта

При нажатии пункта меню «Редактировать рецепт» будет открываться это же окно с возможностью внести изменения и пересохранить рецепт.

Также имеется возможность удаления. Для этого будет использоваться код, отображенный на рисунке 21.

```
int result = dbDatabase.delete(TABLE_RECIPE,
    whereClause: RECIPE_ID + " = ?", new String[] {String.valueOf(id)});
dbDatabase.close();
```

Рисунок 21 - Реализация удаления рецепта

После этого рецепт пропадет из нашей базы данных.

5. Категории

В этом разделе пользователь сможет сгруппировать все свои рецепты. В данной версии приложения будет 5 категорий (рисунок 22). Они будут храниться в таблице categoryTab. Добавим их в таблицу.

<u>id</u>	nameCategory
Фил...	Фильтр
1	Супы
2	Основные блюда
3	Закуски
4	Салаты
5	Десерты

Рисунок 22 - Структура таблицы categoryTab

Для отображения так же будем использовать ListView. Добавим код, отображенный на рисунке 23 в файл FragmentCategory.java

```

ArrayAdapter<String> listViewAdapter = new ArrayAdapter<>(
    getActivity(),
    android.R.layout.simple_list_item_1,
    menuItems
);

```

Рисунок 23 - Реализация listview

В итоге получим список, состоящий из добавленных нами категорий (рисунок 24).

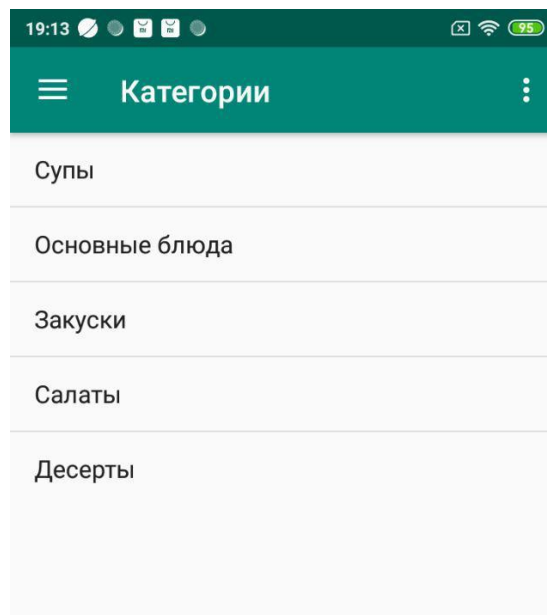


Рисунок 24 - Результат вывода категорий

6. Все рецепты

В данном разделе пользователь сможет просмотреть все доступные рецепты (рисунок 25). Они будут складываться из имеющихся в базе и добавленных самостоятельно. Вывод также будет с использованием ListView для удобства отображения большого количества рецептов.

```

//открываем подключение
db = databaseHelper.open();
//в виде курсора получим данные
userCursor = db.rawQuery("select * from "+ DatabaseHelper.TABLE,
null);
//выберем необходимые столбцы для вывода

```

```
String[] headers = new String[] {DatabaseHelper.COLUMN_TITLE,
DatabaseHelper.COLUMN_PRODUCTS};

//создаем адаптер и передаем курсор, аналогично категориям в пункте 4
userAdapter = new SimpleCursorAdapter(getActivity(),
android.R.layout.two_line_list_item,
userCursor, headers, new int[]{android.R.id.text1, android.R.id.text2}, 0);
userList.setAdapter(userAdapter);
```



Рисунок 25 - Результат отображения всех рецептов

Далее создадим новый акивити в котором уже можно будет просмотреть детали рецепта. Для перехода в него используем код отображенный на рисунке 26:

```
Intent intent = new Intent(getActivity(), RecipeDetails.class);
startActivity(intent);
```

Рисунок 26 - Реализация перехода к новому Activity

Создадим XML - файл, в который добавим TextView и ImageView элементы для отображения всех деталей рецепта. В файле RecipeDetails.java снова подключимся к базе данных и присвоим полям данные из БД (рисунок 27).

```
userCursor = db.rawQuery( sql: "select * from " + DatabaseHelper.TABLE + " where " +
| DatabaseHelper.COLUMN_ID + "=?", new String[]{String.valueOf(userId)});
userCursor.moveToFirst();
setTitle(userCursor.getString( columnIndex: 1));
productsBox.setText(userCursor.getString( columnIndex: 2));
descriptionBox.setText(userCursor.getString( columnIndex: 3));
pic.setImageResource(R.drawable.pic1);
calories.setText(userCursor.getString( columnIndex: 4));
userCursor.close();
```

Рисунок 27 - Реализация вывода деталей рецепта

Запустим приложение и проверим как отображается наш рецепт (рисунок 28).

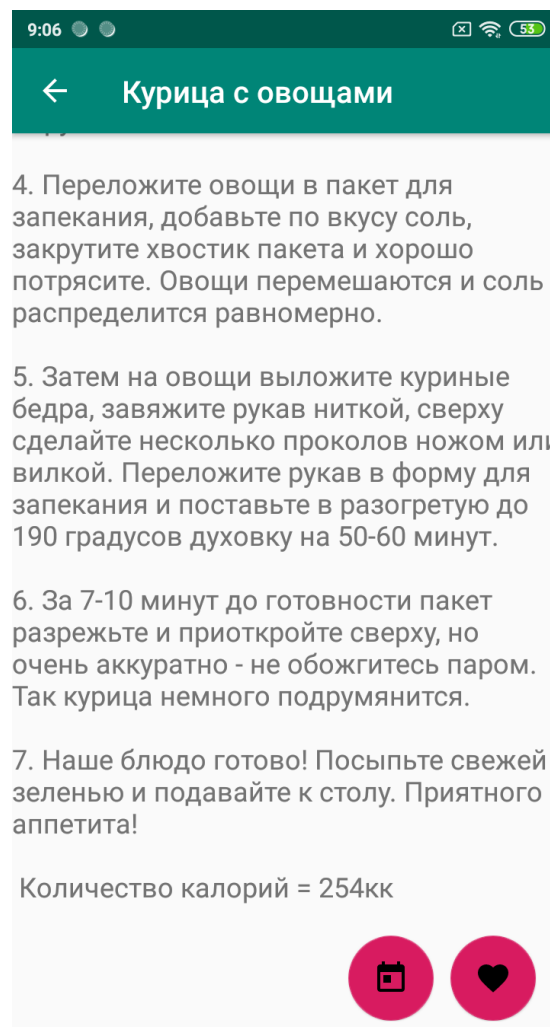


Рисунок 28 - Результат вывода деталей рецепта

7. Приготовлено сегодня

Для поддержания здорового образа жизни будем следить за количеством потребляемых калорий. Калории — это количество энергии, которую мы получаем из принимаемой пищи и теряем при физических нагрузках, если баланс между расходом и поступлением не будет соблюден, получим неправильное функционирование организма, что приведет к проблемам со здоровьем. Для этого после приготовления блюда добавляем его в раздел «Приготовлено сегодня». В этом разделе отображаются все блюда, приготовленные за день. Калорийность всех блюд складывается, и мы получаем число энергии, приобретенное организмом в текущий день.

Для добавления будем использовать специальную кнопку Floating Action Button, которая отображается поверх всего контента, аналогично кнопке «Избранное» (рисунок 29).

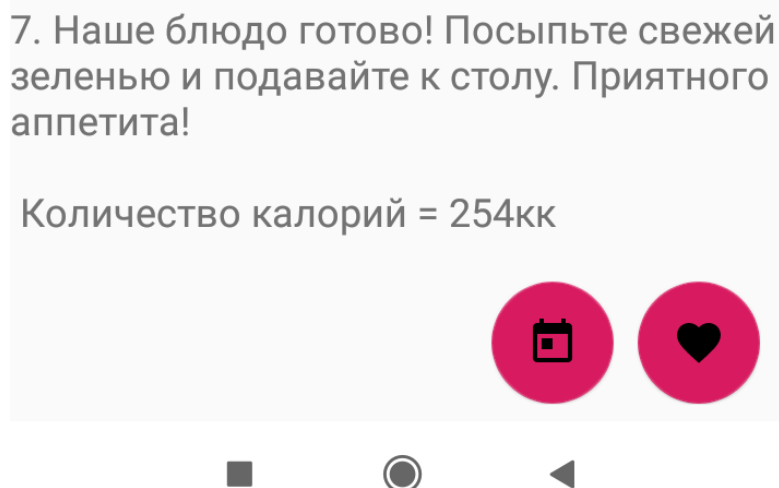


Рисунок 29 - Оформление кнопки Floating Action Button

К концу дня получим список всех приготовленных нами блюд (рисунок 30).

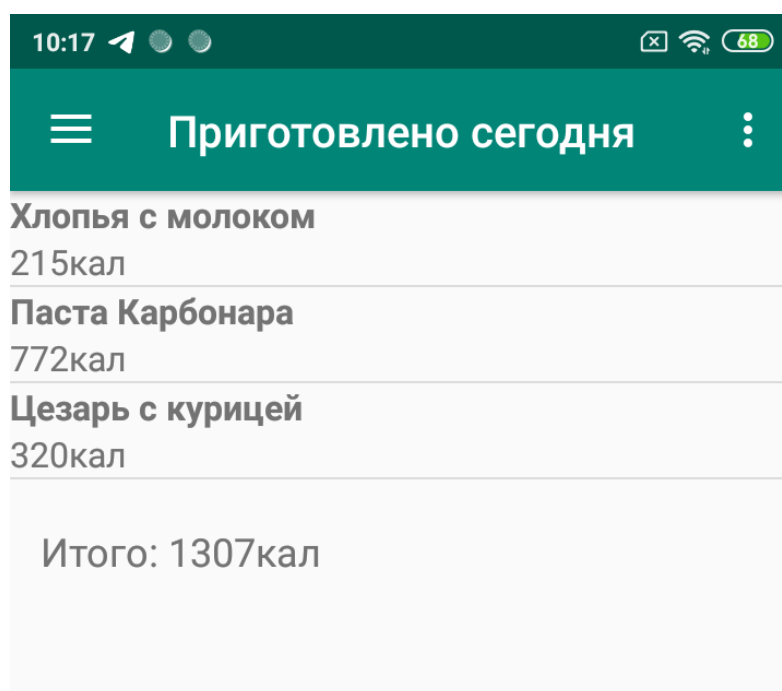


Рисунок 30 - Результат вывода всех приготовленных блюд за день

7. Суточная норма

Чтобы избежать проблем со здоровьем, рассчитаем суточную норму калорий для человека, которой будем придерживаться. Существует три формулы для получения этого числа: Формула Харриса-Бенедикта, Формула Маффины-Джеора, Формула Кэтча – МакАрдла. Формула Харриса-Бенедикта разработана в 1919 году что достаточно давно и основывается на исследовании физически крепких и молодых людей, ведущих активный образ жизни. В современном мире она завышает суточную норму потребления калорий. Данная формула не подходит для нашего времени в связи с изменением питания и уровня активности. Самой точной из формул является Формула Кэтча – МакАрдла. При расчетах по этой формуле используется процент жира в организме. Так как не все знают данную характеристику своего организма от этой формулы тоже придется отказаться. Остается формула Маффины-Джеора, которая была разработана в 90-х годах 20 века. Она является самой популярной и более реалистичной для сегодняшнего образа жизни и питания. Так же, как и Формула Харриса-Бенедикта она не берет в расчет процент жира в организме. Считается, что эта формула тоже несколько завышает потребности в калориях.

В формуле используются такие особенности организма как возраст, вес (кг), рост (см). Сначала посчитаем количество энергии, расходуемое на физиологические процессы в состоянии покоя (на обновление клеток, работу легких, сердца, поддержание температуры тела, циркуляцию крови и т. д.). Для мужчин (1) и женщин (2) применяются разные формулы:

$$\text{BMR} = 9.99 * \text{вес} + 6.25 * \text{рост} - 4.92 * \text{возраст} + 5 \quad (1)$$

$$\text{BMR} = 9.99 * \text{вес} + 6.25 * \text{рост} - 4.92 * \text{возраст} - 161 \quad (2)$$

Полученной энергии недостаточно для поддержания нормальной жизнедеятельности человека, чтобы скорректировать результаты, необходимо учесть энергию, затраченную на физические нагрузки. Для каждого организма количество этой энергии различно, поэтому существует несколько коэффициентов в зависимости от образа жизни:

- минимальные затраты (коэффициент = 1.2);

- низкий уровень активности (коэффициент = 1.375);
- средний уровень активности (коэффициент = 1.55);
- высокий уровень активности (коэффициент = 1.725);
- очень высокий уровень активности (коэффициент = 1.9);

В итоге получим число равное $BMR * \text{Коэффициент активности}$ – это и будет суточной нормой потребления калорий. Запустим приложение и посмотрим на результат (рисунок 31).

13:33

Посчитать калории

Дневная норма калорий

Возраст: 21

Вес: 57

Рост: 168

Активность: Низкая

Пол: м ж

Суточная норма: 1928 калорий

РАСЧИТАТЬ

Рисунок 31 - Результат работы операции "Посчитать калории"

В итоге было создано мобильное приложение способное помочь каждому человеку контролировать приемы пищи в течении дня, следить за нормой калорий и выбирать те блюда, которые способствуют нормальному пищеварению, сохранению баланса полезных веществ в организме и положительно влияющих на здоровье человека.

ЗАКЛЮЧЕНИЕ

В современном мире считается актуальным разработка мобильных приложений. Их использование позволяет сократить время на получение и передачу информации. Благодаря приложениям мы получаем информацию в компактной и точной форме.

В ходе выпускной квалификационной работы был проведен обзор платформ для разработки мобильных приложения для операционной системы Android, решены следующие задачи:

- на основе анализа литературы определили основные понятия и классификацию мобильных приложений;
- рассмотрены инструментальные среды разработки мобильных приложений;
- выявлены достоинства и недостатки сред разработки;
- реализованы основные возможности Android Studio на примере создания мобильного приложения «LetsCook», которое определяет список блюд, которые мы можем приготовить из введенных продуктов. В дальнейшем созданное приложение требует дизайнерского оформления и заполнения базы данных большим количеством рецептов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Бейли Л. Изучаем SQL / Л. Бейли – СПб. : Питер, 2012. – 592 с.
- 2 Дайте П., Дайте Х., Уолдо А. Android для разработчиков / П. Дайте., Х. Дайте, А. Уолдо – Изд. 3-е, – СПб. : Питер, 2016. – 512 с.
- 3 Дейтел Х., Дейтел П., Дейтел Э., Моргано М. Android для программистов. Создаем приложения / Х. Дейтел, П. Дейтел, Э. Дейтел, М. Моргано, – СПб. : Питер, 2013. – 560 с.
- 4 Гриффитс Дон, Гриффитс Дэвид Head First. Программирование для Android / Дон Гриффитс, Дэвид Гриффитс – СПб. : Питер, 2016. – 912 с.
- 5 Карпюк И. А. Сравнительный анализ мобильных приложений и инструментальных средств их разработки [Электронный ресурс] / И. А. Карпюк, Н. М. Куляшова // Научно-методический электронный журнал «Концепт». – 2017. – Т. 31. – С. 826–830. – Режим доступа: <https://e-koncept.ru/2017/970180.htm>.
- 6 Кэмпбелл К., Кэмпбелл Т. Китайское исследование. Результаты самого масштабного исследования связи питания и здоровья / К. Кэмпбелл, Т. Кэмпбелл – Изд. 7-е, – М. : Манн, Иванов и Фербер, 2017. – 400 с.
- 7 Новиков Б. А., Горшкова Е. А. Основы технологий баз данных / Б. А. Новиков, Е. А. Горшкова – М. : ДМК Пресс, 2019. – 240 с.
- 8 Семенчук В. В. Мобильное приложение как инструмент бизнеса / В. В. Семенчук. – М. : Альпина Паблишер, 2017. – 240 с.
- 9 Сиерра К., Бейтс Б. Изучаем Java / К. Сиерра, Б. Бейтс – Изд. 2-е, – М. : Эксмо, 2018. – 720 с.
- 10 Фадеев А. Ю. Сравнительный анализ мобильных приложений и инструментальных средств их разработки / А. Ю. Фадеев, Е. А. Волкова // Наука и перспективы. – 2016. – №3. – С. 8 - 13.
- 11 Android [Электронный ресурс] // Режим доступа: <https://ru.wikipedia.org/wiki/Android>.
- 12 Android Studio [Электронный ресурс] // Режим доступа: <http://developer.android.com/>.

13 C# [Электронный ресурс] // Режим доступа:
https://ru.wikipedia.org/wiki/C_Sharp.

14 Java [Электронный ресурс] // Режим доступа:
<https://ru.wikipedia.org/wiki/Java>.

15 Kotlin [Электронный ресурс] // Режим доступа:
<https://ru.wikipedia.org/wiki/Kotlin>.

ОТЗЫВ

о бакалаврской работе

студента Медведевой Татьяны Сергеевны,

обучающейся по направлению подготовки 02.03.02 - Фундаментальная

информатика и информационные технологии

на тему «Проектирование мобильного приложения «Let's Cook» и его

реализация средствами Android Studio»

Выпускная квалификационная работа Медведевой Т.С. посвящена разработке мобильного Android приложения для помощи в подборе рецептов и подсчете калорий.

Актуальность выбранной для квалификационной работы темы predeterminedена все большим проникновением в различные сферы нашей жизни мобильных устройств, к тому же вопросы правильного питания с каждым годом становятся все актуальнее.

Медведевой Т.С. рассмотрены теоретические аспекты разработки веб-приложений и проанализированы возможности различных языков программирования и сред разработки для создания мобильных приложений, спроектировано и разработано приложение «Let's Cook». Разработанное приложение с определенными доработками может быть размещено в Google Play и им можно полноценно пользоваться.

Структура работы логична, материал излагается последовательно. Тема работы, заявленная к рассмотрению на страницах выпускной работы, раскрыта надлежащим образом.

При выполнении Медведева Т.С. проявила самостоятельность и глубокое понимание рассматриваемого материала. Работа написана грамотным профессиональным языком, содержит описание полученных результатов.

Работа Медведевой Т.С. удовлетворяет всем требованиям, предъявляемым к выпускным квалификационным работам, и заслуживает оценки «отлично», а ее автор присуждения квалификации «Бакалавр» по специальности 02.03.02 «Фундаментальная информатика и информационные технологии».

Научный руководитель



Бадюкина Т.Е., к.ф.-м.н,
доцент

«10» _____ 06 _____ 2019 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА”

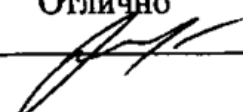
ОТЗЫВ РЕЦЕНЗЕНТА
О ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Студента Медведевой Татьяны Сергеевны
Факультет математики и информационных технологий
Кафедра фундаментальной информатики и информационных технологий
Группа 402
Направление подготовки Фундаментальная информатика и информационные технологии
Квалификация (степень) бакалавр
Наименование темы: Проектирование мобильного приложения «Let's Cook» и его реализация средствами Android Studio
Рецензент Жалнин Р.В., МГУ им. Н.П. Огарева, кандидат физико-математических наук, доцент

ОЦЕНКА ВЫПУСКНОЙ РАБОТЫ

№ п/п	Показатели	Оценка				
		5	4	3	2	0*
1.	Актуальность тематики работы	+				
2.	Степень полноты обзора состояния вопроса и корректность постановки задачи	+				
3.	Уровень и корректность использования в работе методов исследования, математического моделирования		+			
4.	Степень комплексности работы, применение в ней знаний естественнонаучных, социально-экономических, общепрофессиональных и специальных дисциплин		+			
5.	Ясность, четкость, последовательность и обоснованность изложения	+				
6.	Применение современного математического и программного обеспечения, компьютерных технологий в работе	+				
7.	Качество оформления (общий уровень грамотности, стиль изложения, качество иллюстраций, соответствие требованиям стандарта)	+				
8.	Оригинальность и новизна полученных результатов (научных, конструкторских и технологических решений)	+				
9.	Тип работы					
	фундаментальная с оригинальными результатами					
	реферативная					
	прикладная	+				
10.	Рекомендации					
	к опубликованию					
	к внедрению	+				
ИТОГОВАЯ ОЦЕНКА		Отлично				

* – не оценивается (трудно оценить)



Отмеченные достоинства: 1) Приложение имеет интуитивно - понятный интерфейс. 2) Разработано под одну из самых популярных мобильных операционных систем. 3) В работе описываются все этапы разработки мобильного приложения. 4) Может использоваться в повседневной жизни каждого человека.

Отмеченные недостатки: небольшой объем базы данных, что затрудняет оценку эффективности используемых алгоритмов.

Заключение: работа Медведевой Татьяны Сергеевны удовлетворяет всем требованиям, предъявляемым к выпускным квалификационным работам, и заслуживает оценки отлично.

10.06

2019 г.

Рецензент


(подпись)

Заявление о самостоятельном характере выполнения работы

Я, Медведева Татьяна Сергеевна, обучающаяся 4 курса направления подготовки 02.03.02 – Фундаментальная информатика и информационные технологии, заявляю, что в моей работе на тему: «Проектирование мобильного приложения «Let's Cook» и его реализация средствами Android Studio», представленной в Государственную экзаменационную комиссию для публичной защиты, не содержится элементов неправомерных заимствований.

Все прямые заимствования из печатных и электронных источников, а также ранее защищённых письменных работ, кандидатских и докторских диссертаций имеют соответствующие ссылки.

Я ознакомлена с действующим в Университете Положением о проверке работ обучающихся ФГБОУ ВО «МГУ им. Н.П. Огарёва» на наличие заимствований, в соответствии с которым обнаружение неправомерных заимствований является основанием для отрицательного отзыва руководителя работы.


Подпись обучающегося

03.06.2019
Дата

Работа представлена для проверки в Системе «Антиплагиат.ВУЗ»

03.06.2019 г.
Дата представления работы


подпись руководителя

ОТЧЕТ
о результатах проверки работы обучающегося
на наличие заимствований

Ф.И.О. автора работы Медведева Татьяна Сергеевна

Тема работы Проектирование мобильного приложения «Let's Cook» и его реализация средствами Android Studio

Руководитель канд. физ.-мат. наук Бадюкина Т.Е.

Представленная работа прошла проверку на наличие заимствований в системе «Антиплагиат.ВУЗ».

Результаты автоматической проверки: оригинальность 79,14 %
цитирования 0,65 %
заимствования 20,21 %

Результаты анализа полного отчета на наличие заимствований:

правомерные заимствования: да, 20,21 %
да/нет, количество (%), обоснованность

корректные цитирования: да, 0,65
да/нет, количество (%), обоснованность


неправомерные заимствования: нет
да/нет, количество (%), обоснованность

признаки обхода системы: нет
(да/нет, описание)

Общее заключение об итоговой оригинальности работы и возможности ее допуска к защите:

Итоговая оригинальность работы составляет 79,14 %, Медведева Т.С. допускается к защите выпускной квалификационной работы (в форме бакалаврской работы).

Руководитель
Доцент кафедры фундаментальной информатика

 Т. Е. Бадюкина
подпись
04.06.19
дата

Заведующему кафедрой
фундаментальной информатики
А. Г. Смольянову
студента 4 курса
очной формы обучения
на бесплатной основе
направления подготовки
02.03.02 – Фундаментальная
информатика и информационные
технологии факультета математики
и информационных технологий
ФГБОУ ВО «МГУ им. Н.П.Огарёва»
Медведевой Татьяны Сергеевны

заявление.

Прошу разместить мою выпускную квалификационную работу на тему «Проектирование мобильного приложения «Let's Cook» и его реализация средствами Android Studio» в электронной библиотечной системе университета в полном объеме.

20.06.2019
Дата

Медведева
Подпись