

Министерство науки и высшего образования Российской Федерации  
Филиал Федерального государственного автономного образовательного учреждения  
высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
в г. Нижневартовске

Кафедра «Гуманитарные, естественно-научные и технические дисциплины»

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой «ГЕНТД»

к.филос.н., доцент

/И.Г. Рябова/

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

## **Разработка мобильного приложения для ресторана доставки «Суши АНТАНО»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ-09.03.04.2021.041.ПЗ ВКР

Консультанты  
Экономическая часть

к.э.н., доцент

/С.В.Данилова/

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Руководитель работы

Зам. Ген. Директора по производ-  
ству ООО «Северэнергосервис»

/ К.Г. Майфат /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Автор работы

Обучающийся группы НвФл-422

/ П.С. Земан /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Нормоконтролер

старший преподаватель

/ Л.Н. Буйлушкина /

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Нижневартовск, 2021

## АННОТАЦИЯ

Земан П.С. Разработка мобильного приложения для ресторана доставки «Суши АНТАНО» – Нижневартовск: филиал ЮУрГУ, НвФл-422: 2021, 62 стр., 27 ил., 4 табл., библиогр. список – 21 наим., 2 прил.

В рамках данной выпускной квалификационной работы рассматривается разработка мобильного приложения на платформе Android для ресторана доставки «Суши АНТАНО», практическая значимость которой заключается в том, что клиенты могут осуществлять заказы выбирая определенные позиции из перечня меню, что существенно упрощает процесс заказа блюд, также использование мобильного приложения способствует повышению конкурентоспособности за счет возможности добавления различных акций (Промо-кодов) для заказов, которые позволяют привлечь новых клиентов.

Проведен обзор литературных источников по теме разработки, обследована предметная область разработки, выявлены функциональные требования к предметной области, проведено проектирование мобильного приложения для хранения необходимой информации, разработано пользовательское приложение для доступа к данным и формирования отчетной документации.

Выполнен расчет технико-экономической эффективности после внедрения мобильного приложения.

					<b>ЮУрГУ-09.03.04.2021.041.ПЗ ВКР</b>							
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>		
<i>Разработал</i>		<i>Земан П.С.</i>			<b>Разработка мобильного приложения для ресторана доставки «Суши АНТАНО»</b>			<i>В</i>	<i>К</i>	<i>Р</i>		
<i>Проверил</i>		<i>Майфат К.Г.</i>									5	62
<i>Н.контр.</i>		<i>Буйлушкина Л.Н.</i>						<i>Филиал ФГАОУ ВО «ЮУрГУ (НИУ)» в г. Нижневартовске кафедры «ГЕНТД»</i>				
<i>Утвердил</i>		<i>Рябова И.Г.</i>										

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1 АНАЛИЗ ТРЕБОВАНИЙ И МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	7
1.1 Анализ предметной области .....	7
1.2 Обоснование выбора платформы .....	9
1.3 Назначение мобильного приложения .....	12
1.4 Обзор аналогов разработки .....	15
2 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ .....	22
2.1 Проектирование и разработка базы данных.....	22
2.2 Разработка серверного программного обеспечения.....	27
2.3 Разработка клиентского приложения.....	28
2.4 Проектирование интерфейса.....	39
2.5 Тестирование и отладка.....	43
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ .....	45
3.1 Организационно–экономическая характеристика деятельности предприятия .....	45
3.2 Анализ финансовых показателей деятельности предприятия .....	46
3.3 Расчет сметы затрат на реализацию проекта .....	47
3.4 Определение себестоимости приложения .....	48
3.5 Расчет доходов и финансовых результатов.....	50
ЗАКЛЮЧЕНИЕ .....	52
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	53

## ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ОБЕСПЕЧЕНИЯ СИСТЕМЫ УЧЕТА КОМПЬЮТЕРНОГО И ПРОГРАММНОГО .....	56
ПРИЛОЖЕНИЕ Б. КОМПАКТ-ДИСК .....	59

## ВВЕДЕНИЕ

Развитие мобильных сервисов является одним из основных составляющих для будущего любого предприятия. Создание своего собственного мобильного приложения для фирмы – это эффективный способ продвижения бизнеса.

Создание приложений для платформ Android или iOS — является признаком прогрессивной компании.

По данным аналитических агентств App Annie, Statista, Sensor Tower, Mary Meeker с каждым годом увеличивается разрыв между трафиком с десктопных и мобильных устройств.

Кроме того, люди все больше времени проводят в телефонах, растет разрыв между использованием мобильных браузеров и приложений. Судя по прогнозу eMarketer, в ближайшем будущем люди будут использовать мобильное устройство по 4 часа в день, а 88% этого времени будет занимать активность в приложениях.

Несмотря на то, что операционная система Android лидирует на мировом рынке, конкуренция здесь намного меньше, чем в Apple App Store. А так же, хоть и для Android представлено сравнительно меньше программ, новые позиции публикуются в три раза быстрее, чем для iOS. Каждый месяц в Google Play появляется более 100 000 новых приложений, когда в Apple Store около 30 000.

Новые приложения для iOS запускают реже, чем для Android. Это связано с тем, что работать с Google Play Store проще. Также, имеет вес тот факт, что пользователей Android больше, чем iOS.

Целью данной выпускной квалификационной работы является разработка мобильного приложения для ресторана доставки «Суши АНТАНО».

Задачи выпускной квалификационной работы:

- провести анализ предметной области разработки;
- выбрать программно-инструментальные средства разработки;
- спроектировать и реализовать мобильное приложение;

– рассчитать экономическую эффективность разработки.

Объект разработки – процесс заказ в ресторане доставки «Суши АНТАНО».

Предмет разработки – мобильное приложение для ресторана доставки «Суши АНТАНО».

Выпускная квалификационная работа состоит из введения, четырех разделов, заключения и приложений.

В первом разделе проведен анализ предметной области, обзор существующих систем и поставка задачи.

Во втором разделе представлено описание структуры мобильного приложения, а также подробное описание интерфейса приложения.

В третьем разделе рассчитана технико-экономическая эффективность предприятия.

Автоматизация процесса заказа за счет внедрение мобильного приложения на Android позволит сократить трудозатраты связанные с получением информации о заказе, увеличить конкурентоспособность за счет повышения скорости осуществления заказа.

# 1 АНАЛИЗ ТРЕБОВАНИЙ И МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Анализ предметной области

В рамках выпускной квалификационной работы рассматривается разработка мобильного приложения для ресторана доставки «Суши АНТАНО». Данная организация осуществляет свою деятельность в сфере общественного питания в Нижневартовске с октября 2019 года.

Предметом деятельности ресторана доставки «Суши АНТАНО» является:

– услуги по предоставлению питания потребителям, независимо от того, подаются ли они в специальных местах общепита или в ресторанах самообслуживания, едят их в помещении, забирают с собой или заказывают для доставки на дом;

– подготовку и подачу пищи для непосредственного потребления с транспортных средств или передвижных лавок;

– деятельность ресторанов, кафе, ресторанов быстрого обслуживания, мест с предоставлением еды на вынос, вагончиков для продажи мороженого, передвижных вагончиков для продажи пищи, деятельность по приготовлению пищи в торговых палатках;

– деятельность предприятий общественного питания с обслуживанием на вынос;

– деятельность ресторанов и баров, связанную с доставкой продуктов питания потребителям отдельными подразделениями предприятия.

Управляющий ресторана доставки «Суши АНТАНО» – лицо, осуществляющее повседневное управление делами фирмы, формирует заявки на поставку необходимых продуктов, изменяет ценны и меню, добавляет различные акции и промо-коды.

Клиент – лицо, осуществляющее заказ в ресторане доставки «Суши АНТАНО».

Для осуществления заказа в ресторане доставки «Суши АНТАНО» клиент заходит на сайт, проходит регистрацию или авторизацию (указывая свое имя, адрес и номер мобильного телефона), после чего добавляет в корзину нужные ему позиции или же он может позвонить по телефону и оставить свой заказ оператору.

Важным инструментом для моделирования требований с целью представления функциональных возможностей разрабатываемого программного обеспечения или системы в целом являются диаграммы вариантов использования. Данные диаграммы описывают взаимоотношения и зависимости между группами вариантов использования и действующими лицами (актерами), участвующими в процессе. Вариант использования обозначается на диаграмме эллипсом, внутри которого содержится его описание, обозначающее выполнение какой-либо операции или действия [2].

Исходя из вышеописанных функциональных составляющих предметной области, составим диаграмму вариантов использования

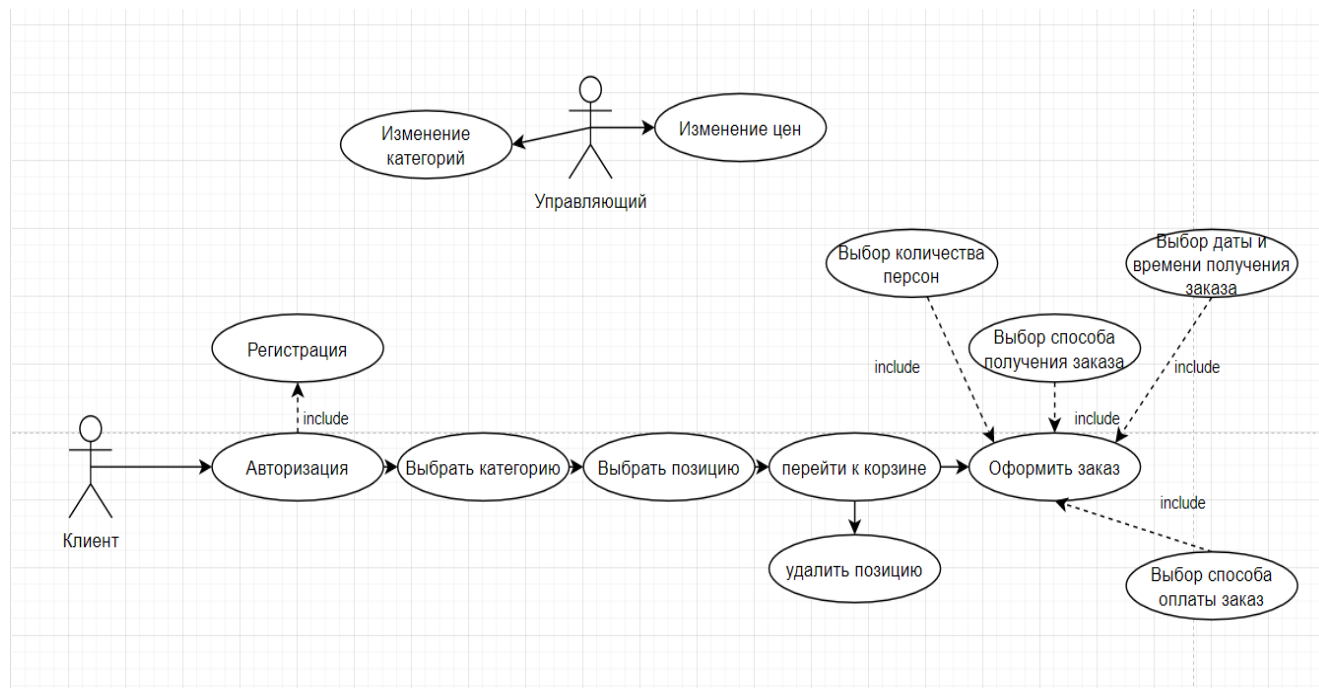


Рисунок 1.1 – Диаграмма вариантов использования для предметной области



## 1.2 Обоснование выбора платформы

Судя по прогнозированию аналитических агентств, в ближайшее время люди будут использовать мобильные устройства более 5 часов в день, и около 90 процентов этого времени будет занимать активность в различных приложениях. Из этого следует что использование мобильного приложения позволит не упустить 90 процентов потенциального трафика.



Рисунок 1.2 – Средний показатель ежедневного использования приложений и браузеров

На июнь 2020 года, на платформе Google Play было выгружено около 3 миллионов приложений, но не смотря на лидерование в мировом рынке, конкуренция здесь намного меньше, чем на других платформах.

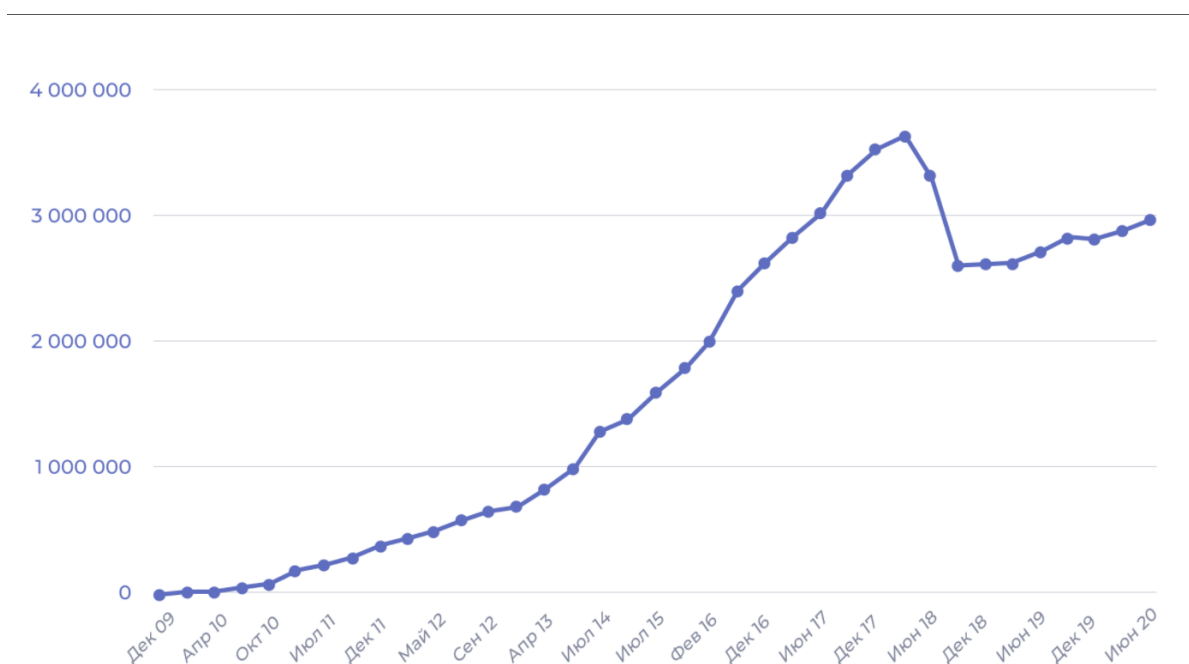


Рисунок 1.3 – Количество приложений на платформе Google Play

На июль 2020 года, в Apple App Store было опубликовано 4,37 млн. приложений. По данным Statista, на каждые четыре неигровых позиции для iOS, публикуется одна игра

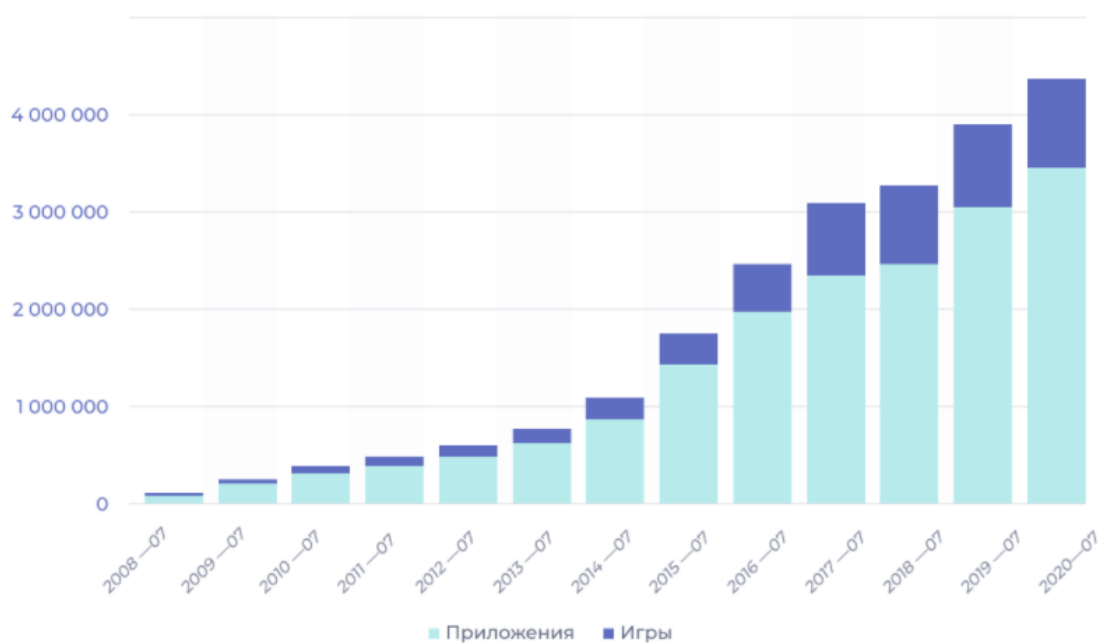


Рисунок 1.4 – Соотношение приложений на платформе App Store

Несмотря на то, что для Android представлено сравнительно меньше программ, новые позиции публикуются в три раза быстрее, в отличие от iOS. Каждый месяц в Google Play появляется более 100 000 новых приложений.

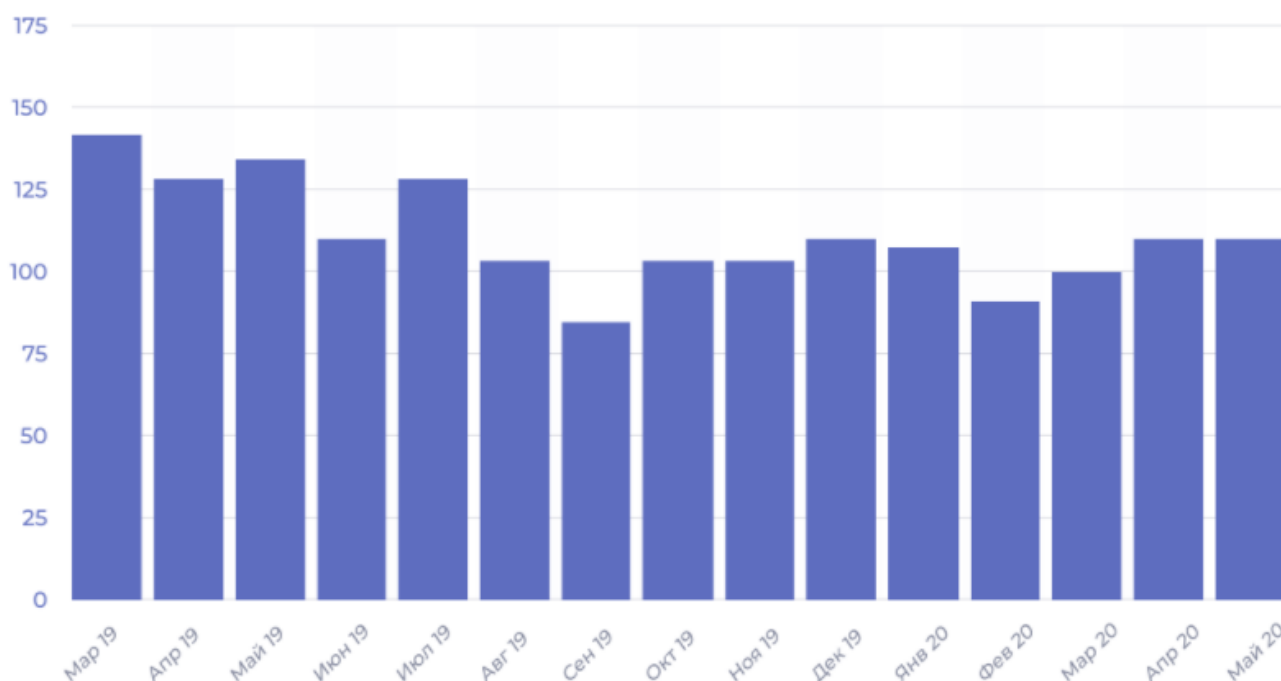


Рисунок 1.5 – Количество запусков приложений в месяц на платформе Google Play

Новые приложения для iOS запускают реже, чем для Android, связано с тем, что работа Google Play Store намного проще. Также, не стоит упускать тот факт, что пользователей Android намного больше. Google за счет своих, интегрированных в платформу Android, приложений смогла завоевать первое место по количеству приложений на рынке, так как их разработками пользуется почти каждый телефон на Android.

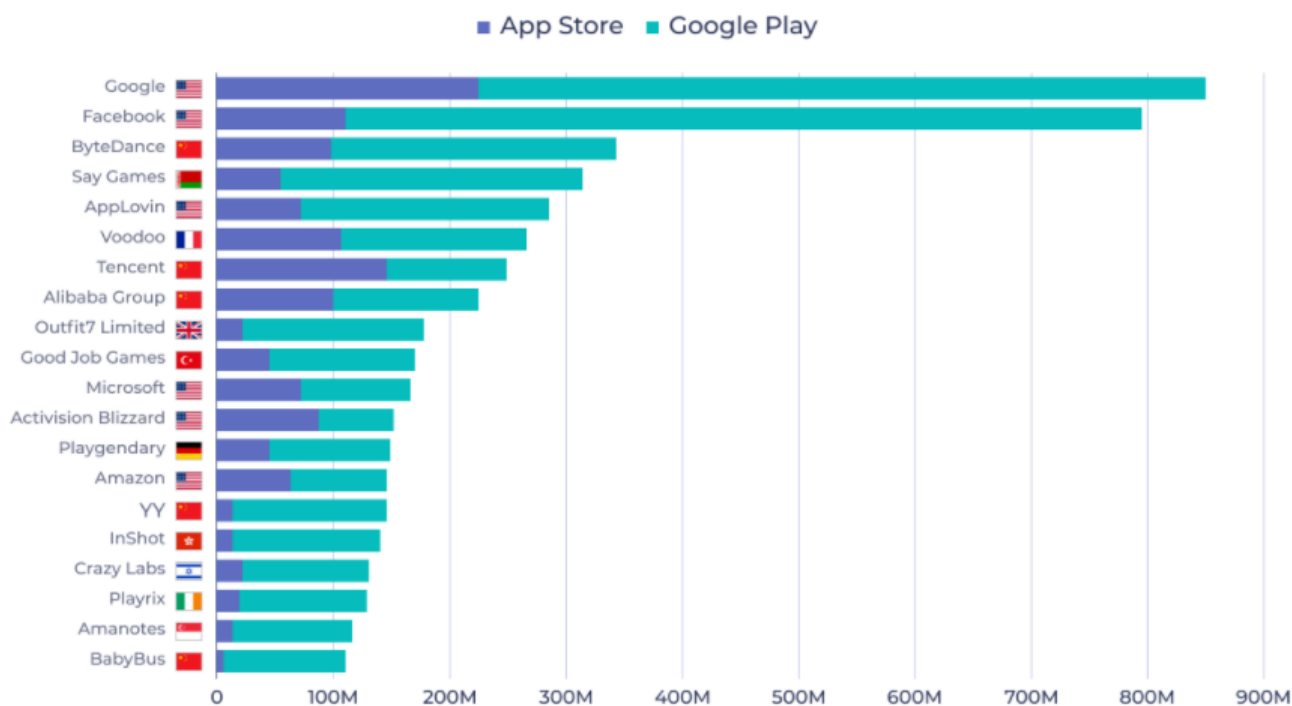


Рисунок 1.6 – Количество установленных приложений, по странам между платформами Google Play и AppStore

Исходя из проведенного обзора платформ и статистики скачиваний мобильных приложений, для разработки была выбрана мобильная платформа Android, а следовательно и платформа для размещения приложений – Google Play.

### 1.3 Назначение мобильного приложения

Мобильное приложения для ресторана доставки «Суши АНТАНО» должно решать следующие задачи:

1. Возможность осуществление клиентом заказа, путем добавления в корзину различных позиций и возможностью их удаления.
2. Возможность использования аукционных промо-кодов, для заказов.
3. Получение клиентом информации о ранее оформленных заказах.
4. Возможность регистрации пользователя в приложении.

Создание этого программного продукта преследует следующие цели:

1. Ускорение и упрощение осуществления заказа.

2. Увеличение потока новых клиентов.
3. Быстрый поиск желаемых позиций блюд.
4. Возможность ввода новых выгодных акций.

Целью данной работы является разработка мобильного приложения для ресторана доставки «Суши АНТАНО».

Формальной постановке задачи соответствует контекстная диаграмма методологии IDEF0, представленная на рисунке 1.7.

Входные данные:

- информация о меню;
- информация о клиенте (номер телефона, имя, адрес).

Выходные потоки:

- чек;
- заказ.

Входные управление:

- процедура выбора количества товара;
- процедура выбора блюда;
- процедура выбора типа оплаты;
- процедура выбора типа доставки.

Входные ресурсы:

- платежная система;
- клиент.

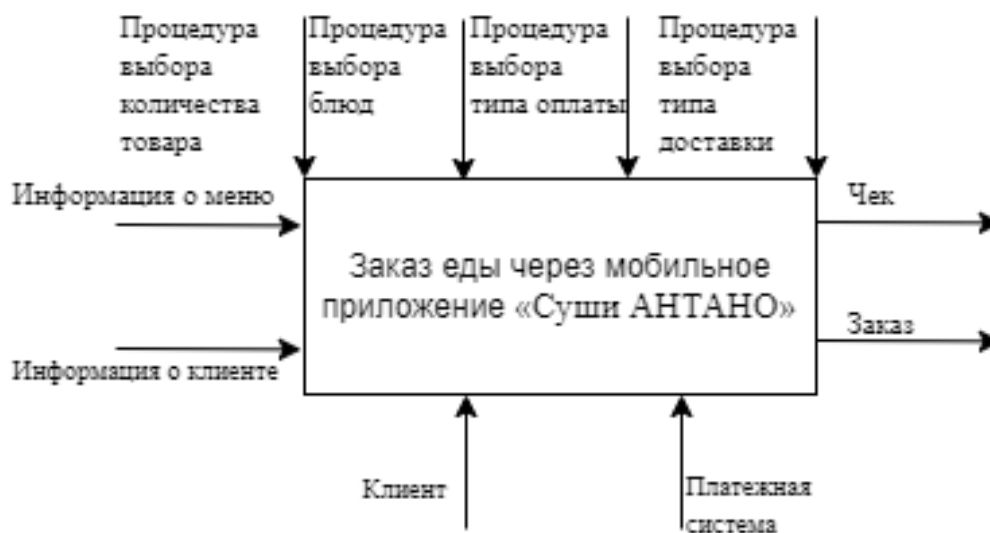


Рисунок 1.7 – Диаграмма процесса заказа еды через мобильное приложение в нотации IDEF0

Необходимо провести дальнейшую декомпозицию задачи на подзадачи, выделим следующие операции:

- авторизация клиента;
- выбор блюд (добавление в корзину);
- оформление заказа;

Декомпозиция процесса заказа еды через мобильное приложение ресторана доставки «Суши АНТАНО» представлена на рисунке 1.8.

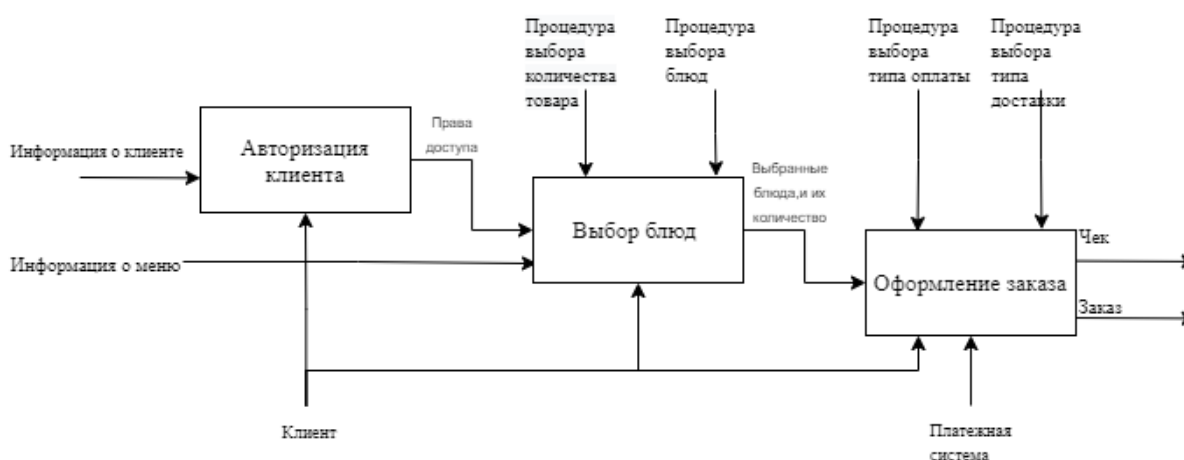


Рисунок 1.8 – Декомпозиция процесса заказа еды через мобильное приложение ресторана доставки «Суши АНТАНО» в нотации IDEF

Также, для более наглядной демонстрации процесса оформления заказа проведем дальнейшую декомпозицию, выделим следующие операции:

- подтверждение заказа;
- оплата;
- формирование заказа и чека.

На рисунке 1.9 представлена контекстная диаграмма данного процесса.



Рисунок 1.9 – Декомпозиция процесса оформления заказа в нотации IDEF

При выполнении процесса «оформления заказа» производится подтверждение заказа, клиентом, после производится оплата, только после этого совершается формирование заказа.

#### 1.4 Обзор аналогов разработки

На настоящий момент имеется ряд мобильных приложений, решающих поставленную задачу, но для определенных фирм. Каждая из них имеет свои особенности, преимущества и недостатки.

Рассмотрим требования для нашего мобильного приложения для осуществления еды в ресторане доставки «Суши АНТАНО». Мобильное приложение должно позволять клиенту осуществлять заказ в ресторане доставки «Суши АНТАНО». При разработке мобильного необходимо учесть следующие требования, позволяющие грамотно собрать модель предметной области:

- необходимо иметь возможность регистрации и авторизации пользователя.
- возможность добавление определенного количества блюд в корзину.
- необходимо иметь возможность осуществления заказа.
- необходимо иметь возможность оплаты заказа.

При анализе современных средств разработки были выбраны 3 основных кандидата:

- Додо пицца.
- **FOODBAND.**
- **DOSTAEВСКИЙ.**

Додо пицца — это мобильное приложение для Android, которое предназначено для осуществления заказа в сети пиццерий Додо пиццы. Уникальность мобильного приложения заключается в информативности. На рисунке 1.10, показана основное меню мобильного приложения.

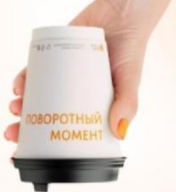


Нижевартовск ▾


На доставку В зале -10%


Выбрать адрес доставки >

1₽ с каждого  
напитка —  
на благотвори-  
тельность




Новое и популярное

 Пепперони  
от 435 ₽

 до 30%

Пицца Комбо Закуски Десерты

 2В1  
Пицца из половинок  
Соберите свою пиццу 35 см  
с двумя разными вкусами

Меню Профиль Контакты Корзина

Рисунок 1.10 – Мобильное приложение Додо пицца

Интерфейс интуитивно понятный. Программа многофункциональна. Хотелось бы отметить следующие преимущества:

— есть функционал для людей с ограниченными возможностями здоровья;

— информативность;

Недостатки мобильного приложения Додо пиццы:

— отсутствует функция привязки банковской карты;

— отсутствует раздел «Избранное».

Вторым кандидатом является мобильное приложение FOODBAND. Это мобильное приложение для Android, которое предназначено для осуществления заказа еды, основная форма мобильного приложения показана на рисунке 1.11.

Приложение FOODBAND имеет следующие плюсы:

— малый вес;

— информативность;

— функция повтора заказа;

— возможность привязки банковской карты;

Недостатки мобильного приложения FOODBAND:

— отсутствует возможность удалить аккаунт;

— отсутствует раздел «Избранное».

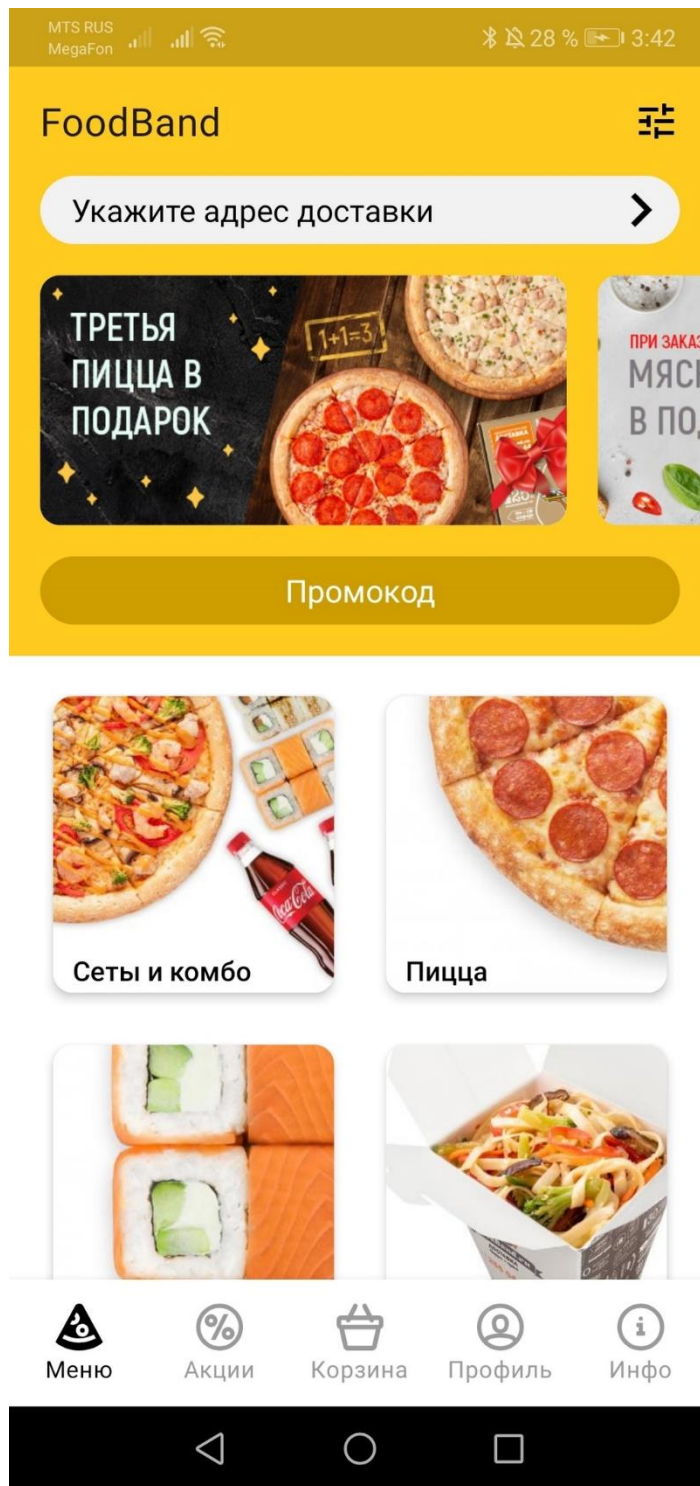


Рисунок 1.11 – Приложение FOODBAND

DOSTAЕВСКИЙ – позволяет осуществлять заказы в сети ресторанов DOSTAЕВСКИЙ. Основная форма мобильного приложения показана на рисунке 1.12.

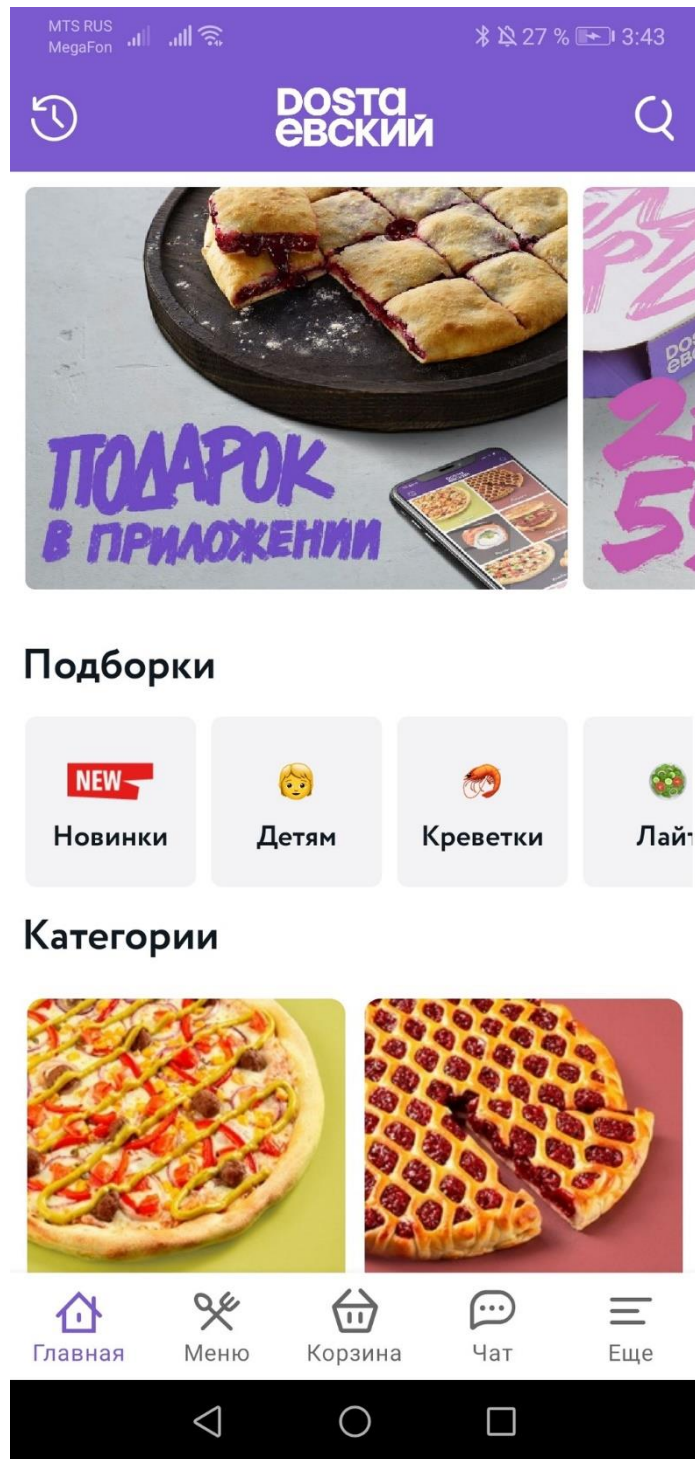


Рисунок 1.12 – Приложение DOSTAЕВСКИЙ

Интерфейс простой, интуитивно понятный. Основные достоинства заключаются в:

- малом весе мобильного приложения.
- хорошем функционале.

— наличии возможностей для людей с ограниченными возможностями здоровья.

Сравнение выбранных мобильных приложений представлено в таблице 1 по следующим критериям: информативность, интерфейс, дизайн, удобство для пользователя, достоинства.

Таблица 1.1 – Сравнение мобильных приложений

Критерий	Додо пицца	FOODBAND	DOSTAEBCKИЙ
Информативность	Информативно	Информативно	Мало информативно
Дизайн	Максимально удобный	Простой	Интуитивно понятный
Интерфейс	Хороший	Приемлемый	Хороший
Удобство для пользователя	Проста в использовании	Удобно	Удобно
Достоинства	Наличие возможностей для людей с ограниченными возможностями	Малый вес	Наличие возможностей для людей с ограниченными возможностями

Выводы по разделу один:

В данном разделе проанализирована предметная область, обоснована актуальность, выделены основные задачи и цели, которые должен решать программный комплекс. Определены требования к программному комплексу. Описана формальная постановка задачи. Также проведен обзор существующих программных продуктов.

## 2 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Разработка мобильного приложения включает в себя 5 этапов:

- 1) Проектирование и разработка базы данных.
- 2) Разработка серверного программного обеспечения.
- 3) Разработка клиентского приложения.
- 4) Проектирование интерфейса.
- 5) Тестирование продукта.

### 2.1 Проектирование и разработка базы данных

На основе анализа и назначения области была спроектирована база данных. База данных — это структура, используемая для хранения информации. В современных базах данных хранятся не только данные, но и информация.

В базе данных есть все необходимые сведения о клиентах, меню ресторана доставки «Суши АНТАНО» и тд. Кроме данных, база данных содержит методы и средства, позволяющие всем пользователям базы данных оперировать только с теми данными, которые входят в их компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным пользователям, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод, редактирование и получение данных.

Основная задача при проектировании базы данных - провести нормализацию до третьей стадии. После нормализации проектируются связи, далее дописываются необходимые хранимые функции.

Для хранения данных в разрабатываемой системе, предполагается хранить на сервере базы данных. Для организации хранения данных были рассмотрены бесплатные системы управления:

- MySQL – одна из самых популярных и распространенная СУБД;

- PostgreSQL - наиболее продвинутая СУБД.
- SQLite - мощная встраиваемая система управления.

MySQL - это самая распространенная полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. Обучиться использованию этой СУБД довольно просто, так как на просторах интернета вы легко найдете большее количество информации.

Несмотря на то, что в ней не реализован весь SQL функционал, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличии от SQLite работают со службами MySQL.

#### Преимущества MySQL:

- простота в работе - установить MySQL довольно просто. Дополнительные приложения, например, GUI, позволяет довольно легко работать с базами данных;
- богатый функционал - MySQL поддерживает большинство функционала SQL;
- безопасность - большое количество функций, обеспечивающих безопасность, которые поддерживается по умолчанию;
- масштабируемость - MySQL легко работает с большими объемами данных и легко масштабируется;
- скорость - упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

#### Недостатки MySQL

- известные ограничения - по задумке в MySQL заложены некоторые ограничения функционала, которые иногда необходимы в особо требовательных приложениях;
- проблемы с надежностью - из-за некоторых способов обработки данных MySQL (связи, транзакции, аудиты) иногда уступает другим СУБД по

надежности;

- медленная разработка - Хотя MySQL технически открытое ПО, существуют жалобы на процесс разработки.

PostgreSQL является самым профессиональным из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. PostgreSQL или Postgres стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий.

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID).) Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций.

Хотя PostgreSQL и не может похвастаться большой популярностью в отличии от MySQL, существует довольно большое число приложений облегчающих работу с PostgreSQL, несмотря на всю мощь функционала. Сейчас довольно легко установить эту СУБД используя стандартные менеджеры пакетов операционных систем.

#### Достоинства PostgreSQL

- открытое ПО соответствующее стандарту SQL - PostgreSQL - бесплатное ПО с открытым исходным кодом. Эта СУБД является очень мощной системой;

- большое сообщество - существует довольно большое сообщество, в котором вы запросто найдёте ответы на свои вопросы;

- большое количество дополнений - несмотря на огромное количество



встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;

- расширения - существует возможность расширения функционала за счет сохранения своих процедур;

- объектность - PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого.

#### Недостатки PostgreSQL

- производительность - при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MySQL;

- популярность - по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество;

- хостинг - в силу вышеперечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД;

Результаты анализа показывают, что по заявленным возможностям явным фаворитом является PostgreSQL.

SQLite – легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам, вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

#### Преимущества SQLite

- файловая структура - вся база данных состоит из одного файла, поэтому её очень легко переносить на разные машины;

- используемые стандарты - хотя может показаться, что эта СУБД примитивная, но она использует SQL. Некоторые особенности опущены, но основные все-таки поддерживаются;

- отличная при разработке и тестировании - в процессе разработки

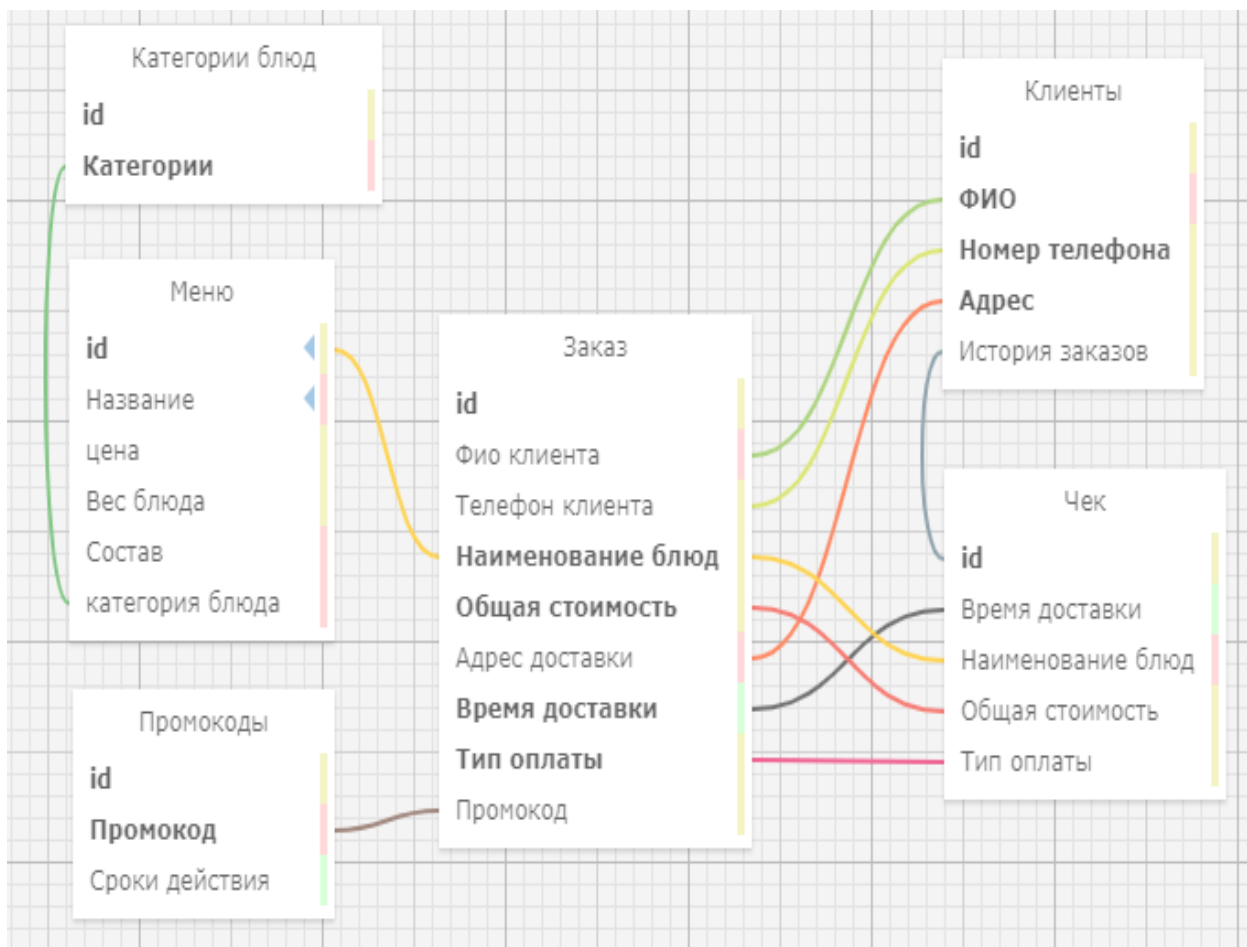
приложений часто появляется необходимость масштабирования. SQLite предлагает всё что необходимо для этих целей, так как состоит всего из одного файла и библиотеки, написанной на языке С.

### Недостатки SQLite

– отсутствие системы пользователей - более крупные СУБД включают в свой состав системы управления правами доступа пользователей. Обычно применения этой функции не так критично, так как эта СУБД используется в небольших приложениях;

– отсутствие возможности увеличения производительности - опять, исходя из проектирования, довольно сложно выжать что-то более производительное из этой СУБД.

В ходе анализа, для реализации базы данных была выбрана система MySQL из-за ее простоты, скорости работы, безопасности.



## Рисунок 2.1 – Схема базы данных

### 2.2 Разработка серверного программного обеспечения

Основной функционал серверного программного обеспечения опирается на необходимость клиентского приложения передавать или получать данные. В нашем случае все модели объектов. Основные задачи и команды, обрабатываемые сервером:

1. Авторизация.
2. Регистрация.
3. Выгрузка истории заказов клиента.
4. Получение информации о категориях блюд.
5. Получение информации о блюде.
6. Получение информации о цене блюда.
7. Получение информации о цене составе блюда.
8. Получение информации о весе составе блюда.
9. Подтверждение заказа.
10. Использование промокода.
11. Оплата.
12. Выбор времени доставки.
13. Выбор типа оплаты.
14. Изменение цен.
15. Выбор адреса доставки.
19. Добавление блюда.
20. Удаление блюда.
21. Добавление категории блюд.
22. Удаление категории блюд.
23. Добавление промокода.
24. Удаление промокода.

25. Получение информации о заказе.
26. Формирование чека.
27. Добавление изображения блюд и категорий.
28. Удаление изображения блюд и категорий.

### 2.3 Разработка клиентского приложения

Для разработки мобильного приложения был произведен анализ нескольких языков программирования, а именно:

- Java
- Kotlin
- Python
- C/C++
- JavaScript
- Dart
- C#

Java – это объектно-ориентированный язык программирования, который был разработан компанией Sun Microsystems в 1990-х годах.

Определение «объектно-ориентированный» относится к типу написания структуры кода Java, а именно: его разделение на «классы», которые запускаются вместе для того, чтобы обеспечить согласованное порождение объектов. Это позволяет организовать код, который можно легко редактировать. Java напрямую связан с такими языками программирования как C и C++, поэтому у них есть много общего. Большим преимуществом Java перед остальным множеством языков, является его платформенная независимость. Код, который вы пишете на одной платформе, можно легко запустить на другой. Java является официальным языком для разработки под Android, который поддерживает Android studio, является некоторой основой для языка kotlin.

Основные преимущества языка:

- большое количество материалов для нахождения нужной информации в процессе разработки;
- большое количество специалистов, работающих с данным языком программирования;
- среда Android Studio изначально заточена под работу с Java;

Kotlin – это язык программирования, разработанный российской компанией JetBrains. Появился он в 2011 году. В 2017 году Kotlin получил официальную поддержку для разработки Android-приложений. Как и Java, C и C++, Kotlin — является типизированным языком. Он поддерживает как объектно-ориентированное, так и процедурное программирование. Основной код Kotlin-программы пишется в функции main, которой передаётся массив аргументов командной строки:

```
1 // необязательный заголовочный файл пакета
2 package hello
3
4 fun main(args: Array < String > ) {
5     val scope = "world"
6     println("Hello, $scope!") //точки с запятыми необязат
7 }
```

Рисунок 2.2 – Функция main на языке программирования Kotlin

Синтаксис Java заочно предполагает более громоздкий код, чем в случае с Kotlin. Например, так будет выглядеть присваивание текст полю на этих двух языках программирования:

```
final TextView helloTextView = (TextView) findViewById(R.id.text_view_id);
helloTextView.setText("Hello world");
```

Рисунок 2.3 – Присваивание текст полю на языке программирования Java

```
helloTextView.text = "Hello world"
```

Рисунок 2.4 – Присваивание текст полю на языке программирования Kotlin

Основные преимущества языка:

- лаконичный код;
- поддерживается Android Studio;
- востребован из-за нехватки Kotlin-разработчиков;
- предоставляет корутины.

Python – это высокоуровневый язык программирования с динамической строгой типизацией и автоматическим управлением памятью, который при помощи Kivy и BeeWare может быть использован для разработки мобильного приложения.

Kivy — это open source библиотека для создания кроссплатформенных приложений, в том числе под Android и iOS, а BeeWare — это набор UI-инструментов для создания нативных Android-приложений.

C/C++ – это компилируемый, типизированный язык программирования который поддерживает обобщённое программирование, процедурное программирование, объектно-ориентированное программирование. У языка есть обширная стандартная библиотека, в которую включены распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. А вообще в целом, С-языки являются высокопроизводительными, и это важно, особенно если мы говорим о чём-то тяжеловесном вроде мобильной 3D-игры, но это безусловно не удовлетворяет нашим потребностям предметной области. Также в процессе разработки на C/C++, так или иначе, придется использовать Java, из-за того, что Android SDK позволит создать на C/C++ лишь часть приложения, которую через java можно подключить к программе.

JavaScript— мультипарадигменный язык программирования который поддерживает объектно-ориентированный, функциональный и императивный стили. JavaScript обычно используется как некоторый встраиваемый язык для

программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. В сочетании с фреймворком React Native возможно создание мобильного приложения с многофункциональными UI. React Native использует компоненты как и простые приложения андроид.

Основные преимущества языка:

- сборка быстрее, чем в Android Studio;
- для создания качественного UI есть flexbox;
- простая передача данных по сети с использованием API;

Dart – язык программирования который является альтернативой языку JavaScript. Один из разработчиков языка Марк Миллер утверждает, что JavaScript «имеет фундаментальные изъяны», которые невозможно исправить. Из-за которых и был создан Dart. Фреймворк Flutter — новая технология от Google, полноценный SDK, который был выпущен в 2018 году. По ряду параметров считается более удачным решением, чем React Native. Позволяет писать кроссплатформенные приложения, поэтому эффективен в разработке приложений для Android с общей базой кода на языке Dart. Dart компилируется в бинарный код, за счёт чего достигается высокая скорость выполнения операций. Вместо XML Dart использует так называемые деревья макетов.

Основные преимущества Dart:

- легко освоить, зная Java;
- высокая производительность программ;
- Hot Reload во Flutter — быстрая перезагрузка с сохранением состояния;

C# – это объектно-ориентированный язык программирования который был разработан в начале 1998 года группой инженеров компании Microsoft, как язык разработки приложений для платформы Microsoft .NET Framework. Программирование на Андроид подразумевает использование платформы Xamarin. С помощью, которой можно создавать единую логику для приложения на C# под платформы Android и iOS.

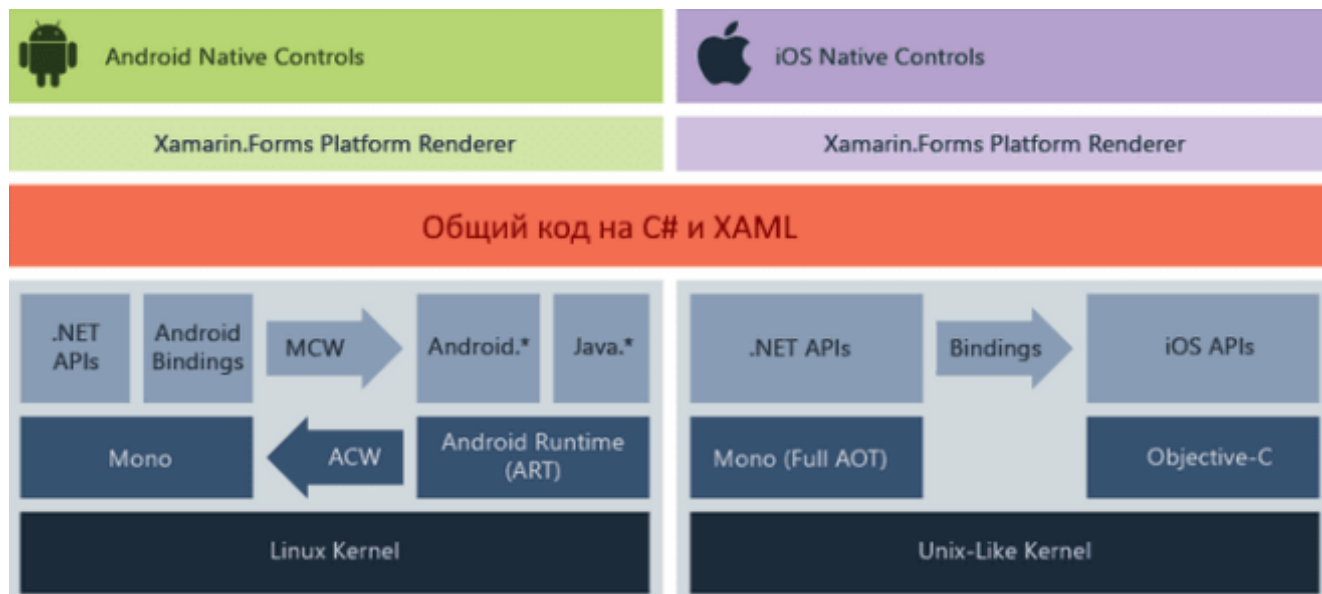


Рисунок 2.5 – Схема функционирования платформы Xamarin на языке программирования C#

Таблица 1.2 – Сравнение языков программирования

Наименование	Сложность языка	Количество информации о языке и его составляющих	Возможность кроссплатформенной разработки	Быстродействие
Kotlin	Легкий	Достаточно	нет	Достаточно быстрый
Python	Легкий	Достаточно мало	да	Быстрый
C/C++	Сложный	Достаточно	нет	Быстрый
JavaScript	Сложный	Много	нет	Достаточно быстрый
Dart	Легкий	Достаточно	да	Быстрый
C#	Сложный	Достаточно	да	медленный
Java	Легкий	Много	нет	медленный



Исходя из проведенного сравнения языков, основным языком разработки мобильного приложения был выбран язык Dart, благодаря поддержке кроссплатформенной разработки, и его быстродействию.

Для разработки мобильного приложения был выбран фреймворк Flutter, который был разработан компанией Google. Фреймворк Flutter позволяет создавать различные кроссплатформенные приложения, в основе которых может быть использован один и тот же программный код. На базе платформы может быть разработано как веб приложения и графические приложения под различные операционные системы (Windows, MacOS, Linux), так и мобильные приложения под Android и iOS.

Особенностью работы с Flutter является то, что приложения под разные платформы могут иметь один и тот же код. Поскольку используемые платформы не эквиваленты, то какие-то отдельные части кода необходимо настраивать под определенную ОС, например, под iOS, но тем не менее большая часть кода может совпадать. Это позволяет разработчикам существенно сэкономить время и ресурсы на создание приложений под обе платформы. Flutter относительно новый фреймворк. Хотя прототип фреймворка появился еще в 2015 году, а первая альфа-версия вышла в мае 2017 года, но первый стабильный релиз - Flutter 1.0 увидел свет только в декабре 2018 года. Тем не менее регулярно с небольшой периодичностью выходят промежуточные версии, добавляя новый функционал и исправляя имеющиеся баги.

Для выбора среды разработки мобильного приложения на Android были рассмотрены следующие варианты:

- Android Studio;
- Eclipse;
- IntelliJ Idea.

Android Studio - это официальная интегрированная среда разработки (IDE) для разработки приложений Android. Он основан на IntelliJ IDEA,

интегрированной среде разработки программного обеспечения на Java, и включает в себя инструменты редактирования кода и средства разработки.

Для поддержки разработки приложений в операционной системе Android Android Studio использует систему сборки на основе Gradle, эмулятор, шаблоны кода и интеграцию с Github.

Каждый проект в Android Studio имеет одну или несколько модальностей с исходным кодом и файлами ресурсов. К этим модальностям относятся модули приложений Android, модули библиотеки и модули Google AppEngine. Android Studio использует функцию Instant Push для отправки изменений кода и ресурсов в работающее приложение. Редактор кода помогает разработчику в написании кода и предлагает завершение, рефракцию и анализ кода. Приложения, созданные в Android Studio, затем компилируются в формат APK для отправки в Google Play Store.

Впервые программное обеспечение было анонсировано на Google в мае 2013 года, а первая стабильная сборка была выпущена в декабре 2014 года. Android Studio доступна для настольных платформ Mac, Windows и Linux. Он заменил Eclipse Android Development Tools (ADT) в качестве основной среды разработки приложений для Android.

Eclipse – это бесплатная IDE, разработанная некоммерческой компанией Eclipse Foundation. Эта программа является базой, которая регулирует процессы создания приложений.

Преимущества Эклипс:

- интерфейс переведен на грамотный русский язык (документация прилагается);
- отлично «гоняет» на компьютерах с низкой производительностью;
- имеет дополнительные функции (для серверной работы и анализа базы данных);
- может подключаться к модулям;

— может работать в групповом режиме (когда проект создают несколько человек одновременно).

IntelliJ Idea – это программа которая была разработана российской компанией JetBrains. Среда позволяет создавать приложения и программы на нескольких программных языках. Также IntelliJ Idea не перегружает ПК.

Преимущества IntelliJ Idea:

- более оперативная отладка значений;
- предусмотрен автозаполнитель методов;
- есть рефакторинг;
- интерфейс более понятный и лаконичный;
- подходит для тех, кто программирует на Java.

Недостатки IntelliJ Idea:

- IntelliJ Idea является платной средой разработки.

Исходя из сравнения средств разработки, для создание мобильного приложения была выбрана среда Android Studio, за его простоту и удобство в использовании.

Перейдем к рассмотрению разработки мобильного приложения. Разрабатываемое мобильное приложение предполагается для использование клиентом, поэтому в нем рассматривается только интерфейс клиента — который вмещает в себя все возможности пользователя и нового клиента организации

При разработке архитектуры был выбран шаблон проектирования MVP.

MVP (Model View Presenter) — шаблон, происходит от паттерна MVC, который набирает популярность при выборе методов разработки приложений для Android.

Так как система Android имеет связи между компонентами графического интерфейса и механизмами доступа к данным, то имеются некоторые проблемы, связанные с этим. Примером такой тесной связи является CursorAdapter, в котором адаптеры (часть графического представления с курсорами) имеют свою роль на уровне доступа к данным.

Для возможности разработки поддерживаемых приложений, следует разделять разные уровни компонентов системы. Архитектура MVP хорошо подходит для случаев, в которых возможно использования данных не только из базы данных, но и из web – сервиса, изменяя лишь определенные участки кода, в которых происходит обращение к данным.

MVP осуществляет графическое отображение независимым от источника данных. Разрабатываемое нами приложение состоит из трех уровней, что позволяет нам тестировать каждый из них отдельно.

#### Presenter

Presenter посредник графического компонента и модели. Он извлекает данные из модели и возвращает их в удобной форме. В отличии от типичного MVC он так же решает, что произойдет, при взаимодействии с представлением.

#### View

Зачастую это Activity, Fragment или View. В идеале, добавление presenter осуществляется с помощью механизма внедрения зависимостей, таких как Dagger, в противном случаи в view должен быть создан объект presenter. Единственное, что view будет делать это вызывать методы presenter, каждый раз когда происходит взаимодействие с графическим интерфейсом (нажатие кнопок, свайпы и т.д).

#### Model

В приложении с хорошей многоуровневой архитектурой, model является точкой доступа к данным или бизнес логики. В "стерильной" архитектуре, model вероятно будет Interactor, который реализует какое-то действие.

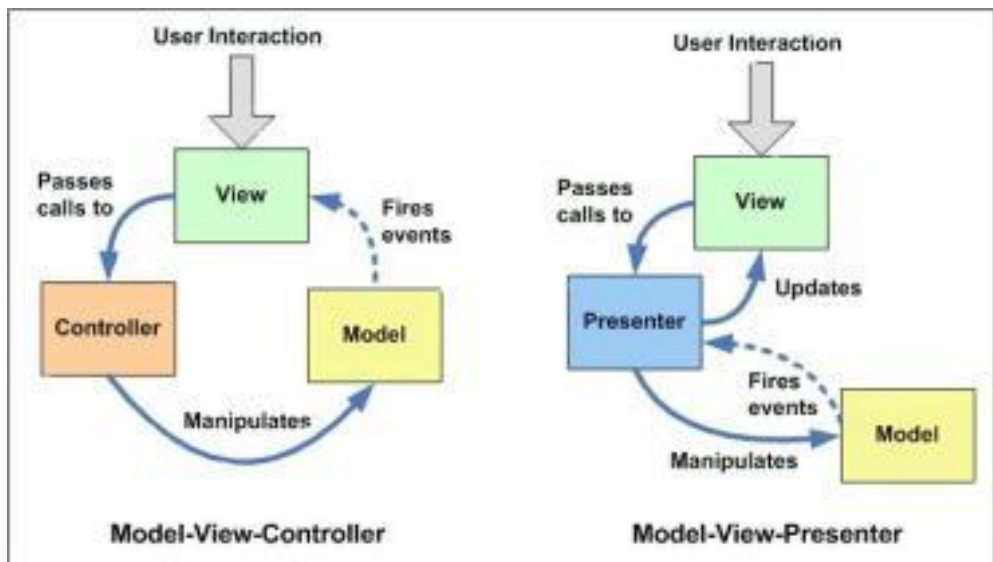


Рисунок 2.6 – Различия между паттернами MVC и MVP

Классы компонента паттерна MVP – View

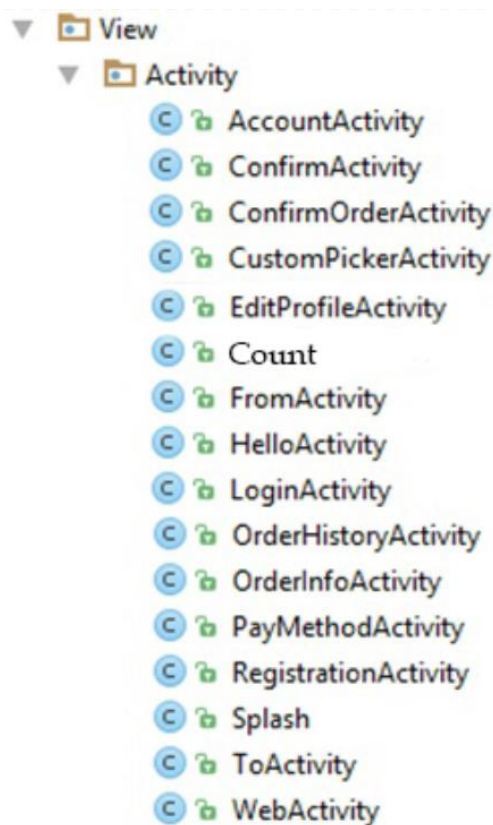


Рисунок 2.7 – Классы компонента паттерна MVP – View

## Классы компонента паттерна MVP – Model



Рисунок 2.8 – Классы компонента паттерна MVP – Model

Классы компонента Model, в двух приложениях, представляют из себя «сущности», содержащие только их свойства (члены классов) и set/get методы.

## Классы компонента паттерна MVP – Presenter

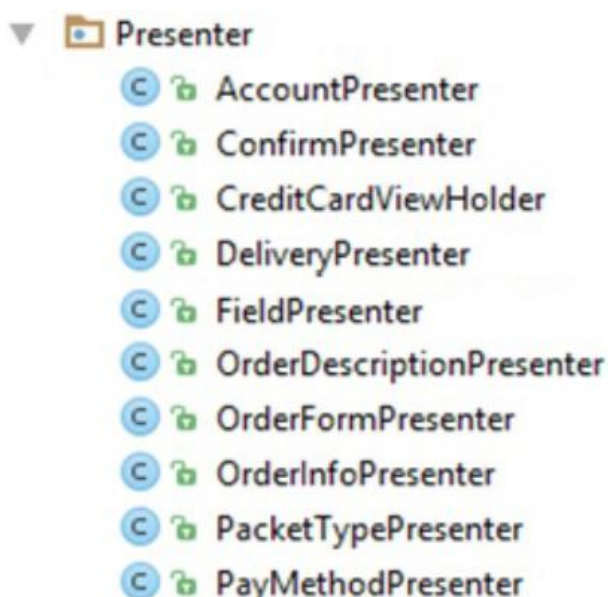


Рисунок 2.9 – Классы компонента паттерна MVP – Presenter

## 2.4 Проектирование интерфейса

Представление функционала в виде графического интерфейса с последующим введением основных моделей объектов, переходов, цветовой гаммы дизайна, и основных функциональных элементов.

Прототипирование интерфейса.

При прототипировании интерфейса будет произведено создание наброска продукта, в котором будет отображен его внешний вид, логика работы, а также будет представлена основная функциональность.

Работа над ним начинается с создания макета. Внешне он выглядит как структурное представление продукта и порядок взаимодействия пользователя с ним.



Рисунок 2.10 – Прототип мобильного интерфейса

Детализированный прототип — следующий шаг по созданию пользовательского интерфейса. Этот макет является более конкретным.



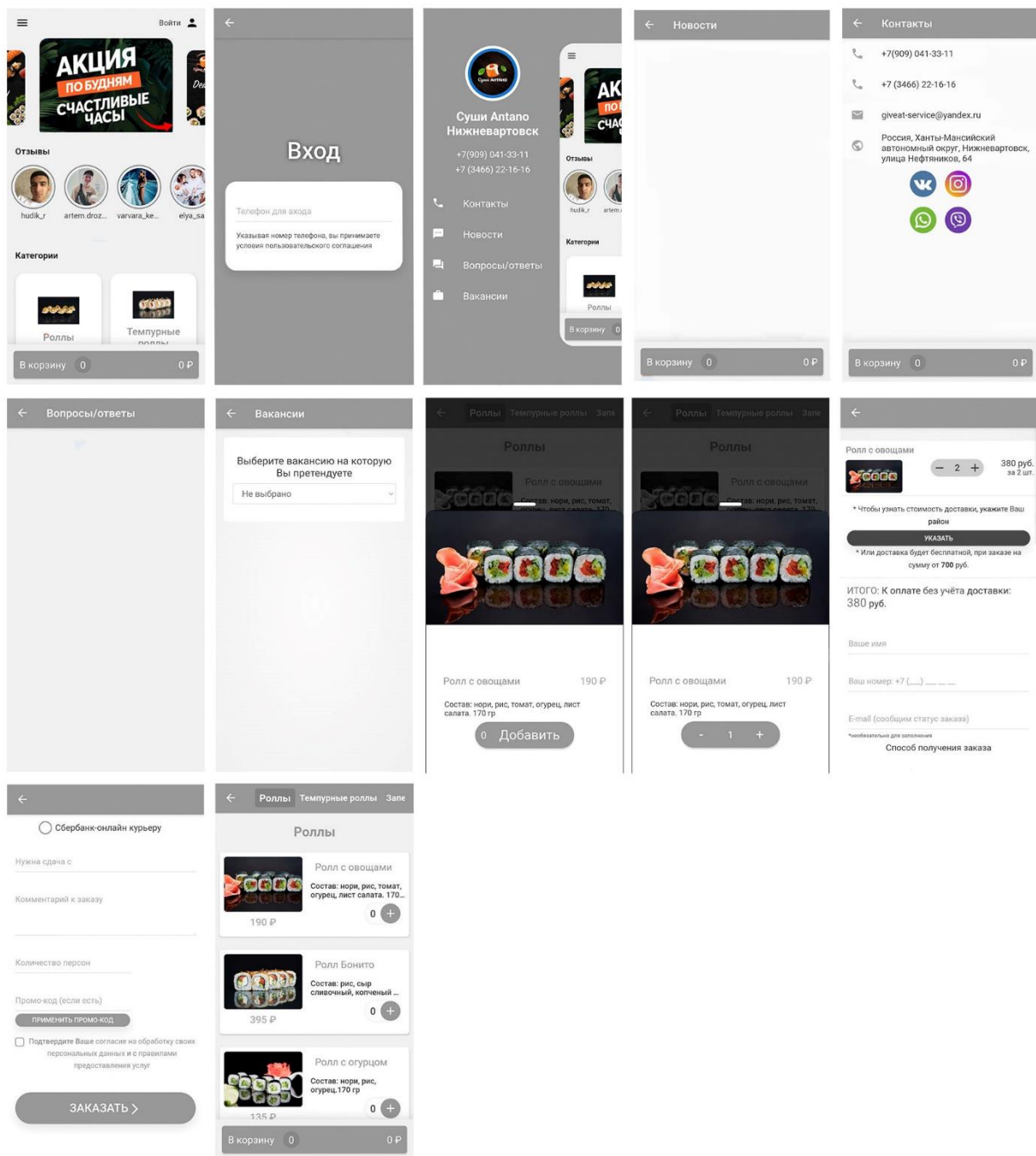


Рисунок 2.11 – Детализированный прототип мобильного интерфейса

Следующим шагом нужно показать, к чему и куда приводит взаимодействие будущего пользователя с элементами интерфейса. Связав элементы линиями с другими экранами, на которые попадёт пользователь, мы получаем пользовательские сценарии использования приложения.

Сценарии использования приложения — карта навигации, по которой видно поведение пользователя мобильного приложения, как он достигает цели и как легко ему это удаётся. Внешне он выглядит как логически связанные друг с другом прямоугольники, акцент в которых сделан на действиях пользователя.

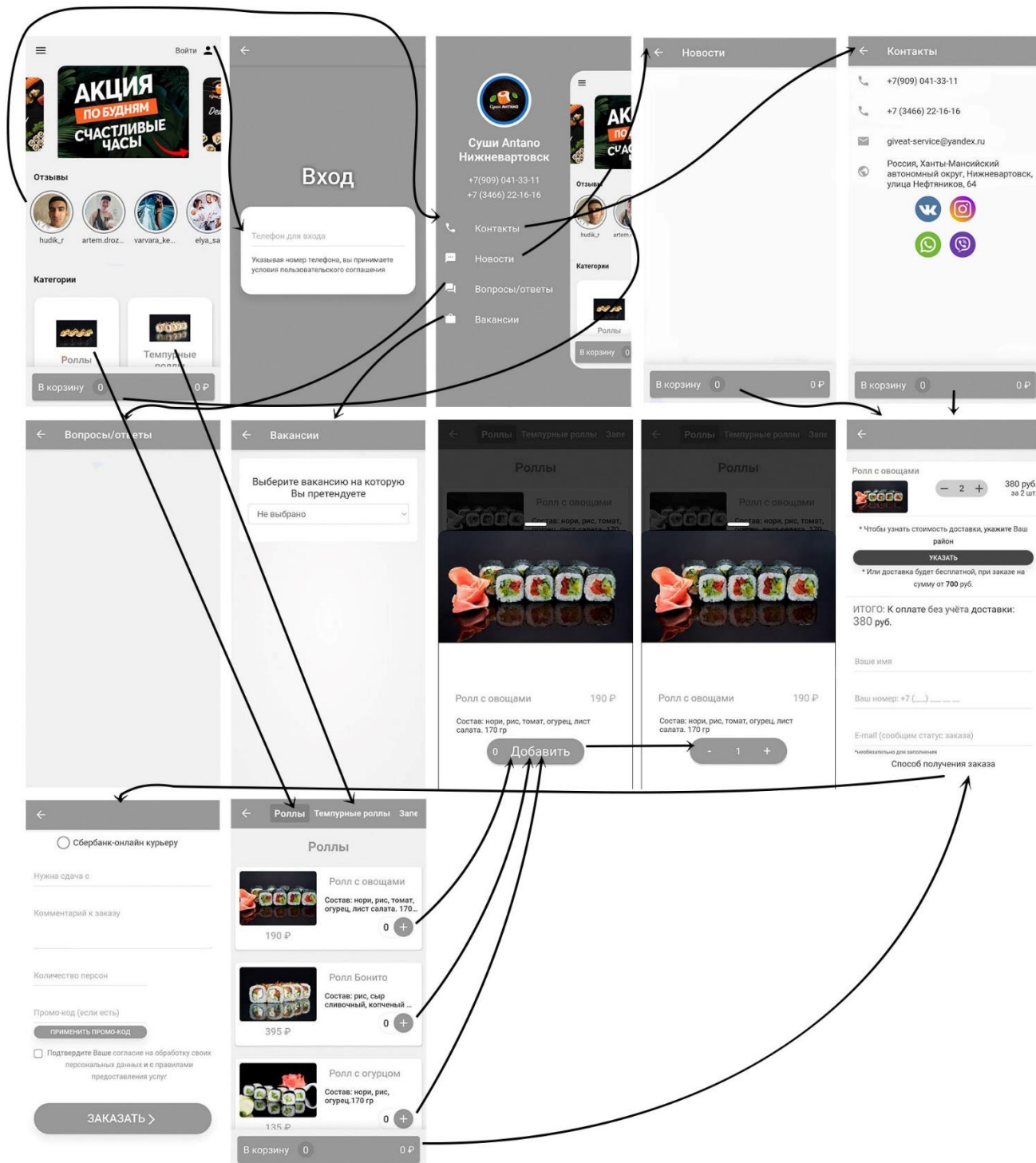


Рисунок 2.12 – Сценарий использования мобильного интерфейса

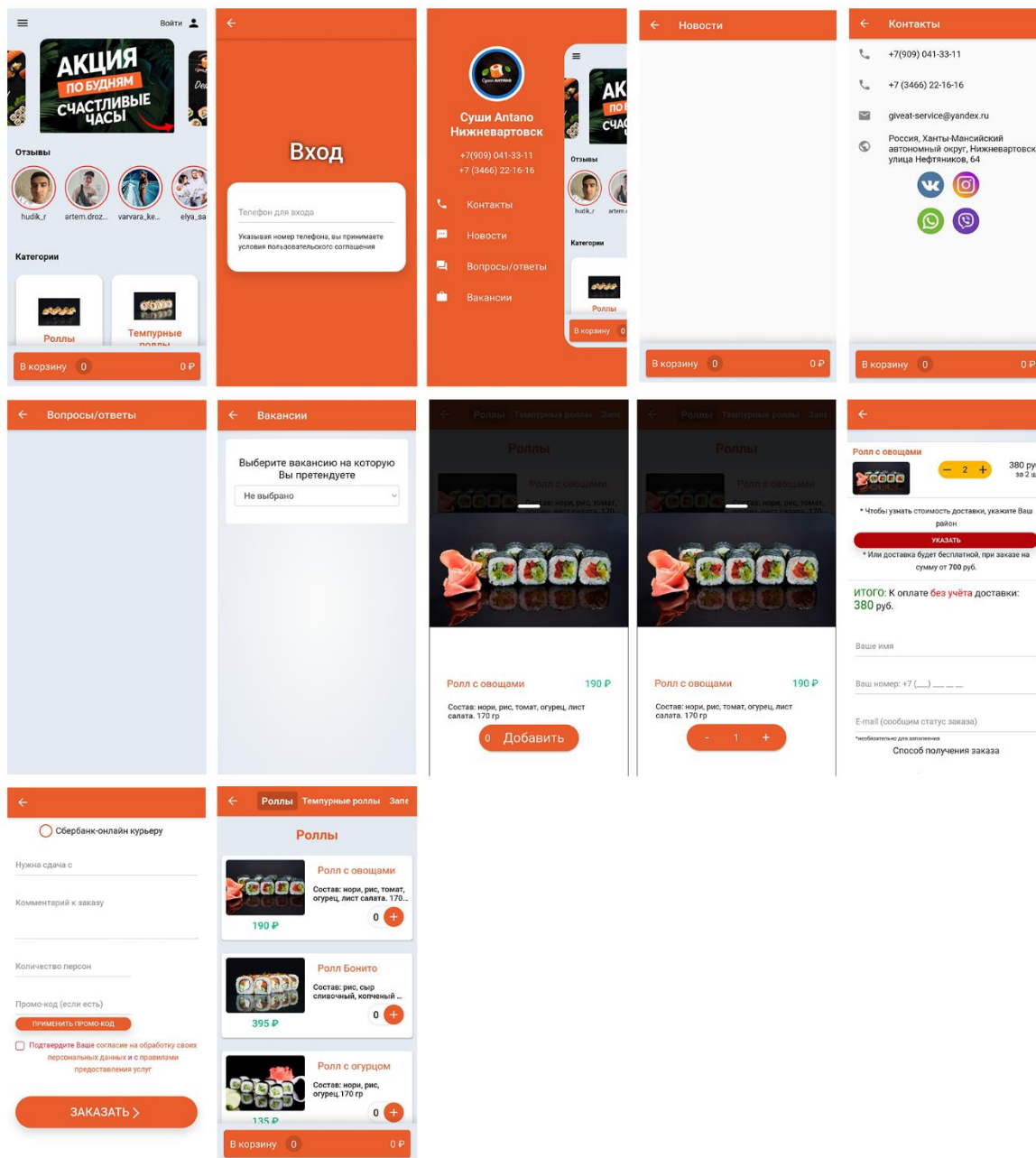


Рисунок 2.13 – Стилизация мобильного интерфейса

## 2.5 Тестирование и отладка

Тестирование программы – процесс выполнения программы с целью обнаружения ошибок в программе на некотором наборе данных, для которых заранее известен результат применения или известны правила поведения этих программ. Тестирование программ является одной из составных частей более общего понятия - «отладка программ». Под отладкой понимается процесс,

позволяющий получить программу, функционирующую с требуемыми характеристиками в заданной области изменения входных данных [6].

Тестирование мобильного приложения и программного обеспечения прошло успешно, программа функционирует без ошибок.

Данная программа не нуждается в отладке, так как внесенная информация заполнялась тестовыми данными, внесенные данные работают корректно.

Выводы по разделу два:

В процессе работы были рассмотрены этапы разработки программного продукта, такие как: проектирование и разработка базы данных, разработка сервиса, разработка мобильного приложения, проектирование интерфейса, тестирование продукта. Также были представлены фрагменты спроектированного интерфейса мобильного приложения, схема базы данных мобильного приложения.

### 3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

#### 3.1 Организационно–экономическая характеристика деятельности предприятия

Ресторан доставки «Суши АНТАНО» осуществляет свою деятельность в сфере общественного питания в Нижневартовске с октября 2019 года.

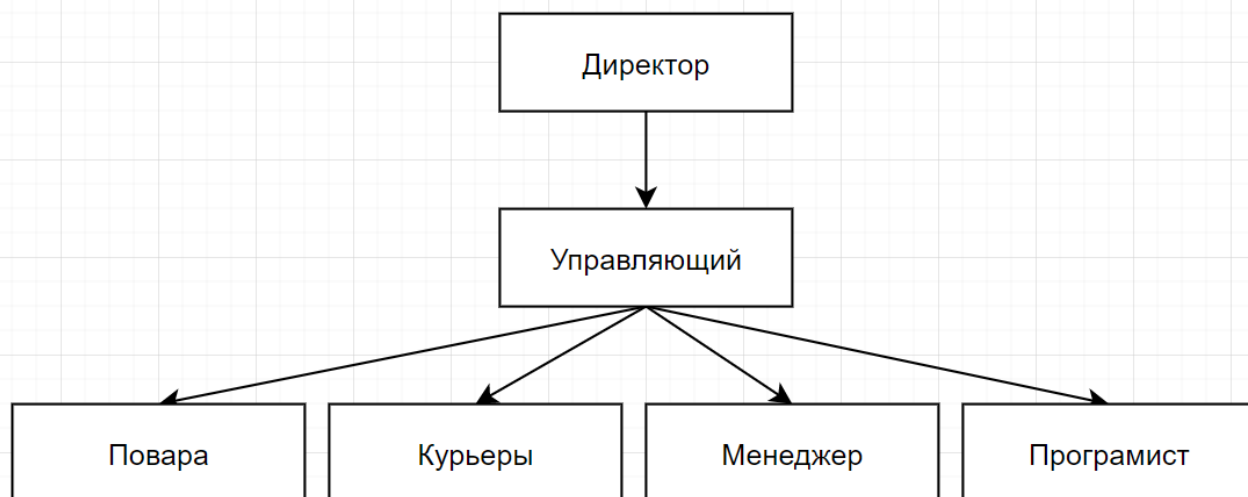


Рисунок 3.1 – Структура управления ресторана доставки «Суши АНТАНО»

Программист выполняет следующие должностные обязанности:

- устанавливает на серверы и рабочие станции сетевое программное обеспечение;
- конфигурирует систему на сервере;
- обеспечивает интегрирование программного обеспечения на файл-серверах, серверах систем управления базами данных и на рабочих станциях;
- поддерживает рабочее состояние программного обеспечения сервера и сайта;

— участвует в восстановлении работоспособности системы при сбоях и выходе из строя сетевого оборудования;

— готовит предложения по модернизации и приобретению сетевого оборудования;

— выполняет отдельные служебные поручения управляющего.

### 3.2 Анализ финансовых показателей деятельности предприятия

В данном пункте рассматриваются финансовые показатели ресторана доставки «Суши АНТАНО» за период с марта 2020 года по февраль 2021 года.

Месяц	Выручка общая	Расходы	Доход
Март	552 722	400 722	152 000
Апрель	747 355	653 646	93 709
Май	753 515	575 537	177 978
Июнь	1 025 422	713 281	312 141
Июль	819 414	531 971	287 443
Август	824 305	573 027	251 278
Сентябрь	678 296	642 438	35 858
Октябрь	817 117	545 368	271 749
Ноябрь	710 216	609 662	100 554
Декабрь	784 856	605 478	179 378
Январь	776 896	588 176	188 720
Февраль	894 763	711237	183526

Рисунок 3.2 – Финансовые показатели ресторана доставки «Суши АНТАНО»

Исходя из данной таблицы финансовых показателей ресторана доставки «Суши АНТАНО», данное заведение можно считать рентабельным, заведение за данный период работало без минусовых показателей.

Доход за данный период с марта 2020 года по февраль 2021 года составил 2234334 (два миллиона двести тридцать четыре тысячи триста тридцать четыре) рубля.

В среднем в день приходит от 30 до 65 заказов, средний чек заказов 565 рублей.

### 3.3 Расчет сметы затрат на реализацию проекта

Создание мобильного приложения производит один программист с зарплатой 30000 руб./мес.

Часовая ставка заработной платы вычисляется по формуле (1):

$$Z_{\text{час}} = \frac{\text{Оклад в Месяц}}{\text{Фонд рабочего времени}} = \frac{30000}{160} = 187,5 \text{ руб.} \quad (1)$$

Размер основной заработной платы за  $i$ -ый этап определяется как часовая ставка, умноженная на время в часах, необходимое для выполнения  $i$ -того этапа рассчитывается по формуле (2):

$$Z_{\text{этап}} = Z_{\text{час}} \times t_{\text{вып}} \quad (2)$$

Размер основной заработной платы при восьмичасовом рабочем дне, требуемых на разработку, вычисляется по формуле (3):

$$Z_{\text{осн}} = \sum Z_{i\text{-тый этап}} = \sum (Z_{\text{час}} \times t_{\text{вып}}) \quad (3)$$

Таблица 3.3 – Расчет длительности разработки графического приложения и размера заработной платы

Этапы	Виды работ	Исполнители		Часовая ставка, руб.	Длительность выполнения, часы	Размер Зарплаты, Руб.
		Количество	Должность			
Начальный	Формирование требований к системе учета	1	Программист	187,5 руб.	12	2 250 руб.
Подготовительный	Подбор информации, используемой для системы учета	1	Программист	187,5 руб.	24	4 500 руб.
Внешнее проектирование	Разработка архитектуры прогр. комплекса и его интерфейса	1	Программист	187,5 руб.	24	4 500 руб.
Основной	Написание кода	1	Программист	187,5 руб.	56	10 500 руб.
Заключительный	Отладка и исправление	1	Программист	218,75 руб.	24	4 500 руб.
	Итоговое тестирование	1	Программист	218,75 руб.	8	1 500 руб.
Всего					148	27750 руб.
Дополнительная зарплата (10%)						2775 руб.
Итого						30 525 руб.

Всего на разработку приложения при учете восьмичасового рабочего дня, было затрачено 30 525 руб.

### 3.4 Определение себестоимости приложения

В себестоимость включаются следующие статьи затрат:

- основная зарплата;
- дополнительная зарплата;



- отчисления на социальные взносы (30%);
- накладные расходы;
- прочие прямые расходы.

Основная и дополнительная зарплата были определены выше. К дополнительной зарплате относятся выплаты, предусмотренные законодательством (оплата отпусков, доплаты за выслугу лет и т.п.). В научных организациях дополнительная зарплата составляет 10-12% от основной (в коммерческих 10 – 30%). Таким образом, сумма отчислений во внебюджетный фонд:  $O_{\text{Страх}} = 30\,000 \times 0.3 = 9000$  руб.

Затраты на амортизацию оборудования проводятся за период их использования, т.е. за период внедрения и создания дополнений к программному обеспечению.

Денежное выражение амортизации является амортизационным отчислением, которое входит в текущие затраты.

Величина амортизационных отчислений определяется на основе норм амортизации.

Норма амортизации – это установленный размер амортизационных отчислений на полное восстановление, выраженное в %. Норма амортизации устанавливается на основе экономически целесообразного срока службы и должна обеспечить возмещение износа основных средств к моменту возможного их морального и физического износа и создать экономическую основу для замены.

$$A = \frac{30000}{5} \times \frac{148}{2000} = 444$$

К прочим прямым расходам относятся расходы на обслуживание компьютера и плата за потребляемую электроэнергию, а также расходы на прочие материалы. По данным документов энергосбытовой компании – акционерное общество «Нижевартовская энергосбытовая компания», стоимость 1 кВт-час

электроэнергии – 6 руб., потребляемая ЭВМ мощность 0,4 кВт, получим затраты на электроэнергию в размере:

$$P_{\text{проч}} = 148 \times 6 \times 0,4 = 355,2 \text{ руб.}$$

Также в процессе разработке были использованы канцелярские товары, такие как ручка (цена которой составляет 25 рублей за штуку) и пачка бумаги для принтера, А4 стоимостью в 350 рублей.

Рассчитанные величины сведем в таблицу 3.4:

Таблица 3.4 – Себестоимость разработки программного комплекса

Статья расходов	Сумма, руб.	Удельный вес, %
Основная зарплата	27750	68,18
Дополнительная зарплата	2775	6,81
Отчисления во внебюджетные фонды (30%)	9000	22,11
Амортизация	444	1,10
Прочие прямые расходы	730,2	1,79
Итого	40 699,2	100

Себестоимость разработки приложения при учете основной и дополнительной заработной платы, амортизации, отчислениях во внебюджетные фонды и прочих расходов было затрачено 40 699,2 руб.

### 3.5 Расчет доходов и финансовых результатов

Без использования мобильного приложения, разработанного в рамках выпускной квалификационной работы, происходит уменьшение конкурентоспособности среди ресторанов доставки.

Разрабатываемый программный комплекс предназначен для облегчения осуществления заказа. В среднем в день приходит от 30 до 65 заказов. По

статистике мобильное приложение способствует повышению продаж на 2 процента и больше.  $\overline{СрКз} = \frac{30+65}{2} = 47.5$ ,  $\overline{КнЗк} = 47.5 \times 0,02 = 0,95$

При внедрении мобильного приложения ресторана доставки «Суши АНТАНО» количество заказов будет увеличено на  $\overline{0,95}$  заказов за день, возьмем средний чек заказа за день, который равняется 565 рублей.

За месяц, прибыль от внедрения мобильного приложения составит:  
 $\Pi = 30 \times 0,95 \times 565 = 16\ 102$  руб. С учетом, что на разработку было затрачено 40 699,2 руб., период окупаемости составит:  $40\ 699,2 / 0,95 \times 565 = 76,42$ .

В результате мобильное приложение начнет окупать себя уже на 77 день после запуска в эксплуатацию.

Выводы по разделу три:

В случае внедрения разработанного мобильного приложения рестораном доставки «Суши АНТАНО» срок окупаемости наступит на 77 день после внедрения, и будет приносить прибыль, за месяц равную 16 102 рублей. Мобильное приложение поможет повысить конкурентноспособность ресторана и увеличить количество потенциальных клиентов.

## ЗАКЛЮЧЕНИЕ

Цель данной выпускной квалификационной работы – разработка мобильного приложения для ресторана доставки «Суши АНТАНО».

Исходя из цели и поставленных задач, была определена структура данной выпускной квалификационной работы, в ходе которой был произведён анализ деятельности фирмы, анализ средств для создания мобильного приложения, представлена программная реализация для написания и описан результат работы мобильного приложения.

В выпускной квалификационной работе изложены теоретические основы построения мобильных приложений, представлены официальные данные аналитических агентств App Annie, Statista, Sensor Tower, Mary Meeker.

Рассмотрены известные аналогичные разработки.

Результатом решения поставленных задач является создание программного мобильного приложения, которое позволяет клиентам самостоятельно осуществлять заказ.

Таким образом, поставленные задачи решены в полном объеме. Цель выпускной квалификационной работы достигнута.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2012. – 400 с.
- 2 Семенчук, В. Мобильное приложение как инструмент бизнеса / В. Семенчук. — Москва : Альпина Паблишер, 2017. — 240 с.
- 3 Баккет, К. Dart в действии : учебное пособие / К. Баккет. — Москва : ДМК Пресс, 2013. — 528 с.
- 4 Роберт, Дж. Мюллер. Проектирование баз данных и UML / Роберт Дж. Мюллер – М.: Лори, 2013. – 432 с.
- 5 EPC – диаграммы [Электронный ресурс].- режим доступа URL: [https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema8/tema8\\_3](https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema8/tema8_3), свободный. [дата обращения – 25.05.2020]
- 6 Абрамов, И. В. Интерфейс инструментария business studio : учебное пособие / И. В. Абрамов, А. В. Копытин, М. Г. Матвеев. — Воронеж : ВГУ, 2016. — 32 с.
- 7 Черников, В. Разработка мобильных приложений на C# для iOS и Android учебное пособие / В. Черников. — Москва : ДМК Пресс, 2020. — 188 с.
- 8 Бойко, В.В. Проектирование баз данных информационных систем / В.В. Бойко, В.М. Савинков – М.: Финансы и статистика, 1989. - 351 с.
- 9 Диго, С.М. Проектирование и использование баз данных / С.М. Диго – М.: Финансы и статистика, 1995. – 208 с.
- 10 Трипкош, В. А. Информатизация институтов управления и финансовых структур : учебное пособие / В. А. Трипкош. — Оренбург : ОГУ, 2019. — 111 с.
- 11 Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2012. - 400 с.

12 Коликова, Т.В. Основы тестирования программного обеспечения / Т.В. Коликова, В.П. Котляров. – М., Бином, 2010, 285 стр.

13 Заяц, А. М. Введение в гибридные технологии разработки мобильных приложений : учебное пособие для спо / А. М. Заяц, Н. П. Васильев. Санкт-Петербург : Лань, 2021. — 160 с.

14 Ретабоуил, С. Android NDK: руководство для начинающих : руководство  
С. Ретабоуил ; перевод с английского А. Н. Киселев. — 2-е изд. — Москва : ДМК Пресс, 2016. — 518 с.

15 Беспалов, Д. А. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения : учебное пособие : в 2 частях / Д. А. Беспалов. — Ростов-на-Дону : ЮФУ, 2019 — Часть 2 : Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения — 2019. — 168 с.

16 Соколова, В. В. Разработка мобильных приложений : учебное пособие / В. В. Соколова. — Томск : ТПУ, 2014. — 176 с.

17 ГОСТ 22.771-77 Требования к информационному обеспечению – Москва: Издательство стандартов, 2011. – 56 с.

18 Информационно-аналитическая система Универис [Электронный ресурс] – Режим доступа: <http://www.univeris.susu.ru/SitePages/Home.aspx> – Загл. с экрана. [дата обращения - 25.05.2021]

19 Методические рекомендации по подготовке и оформлению выпускной квалификационной работы (проекта) для технических направлений подготовки 09.03.01 Информатика и вычислительная техника, 09.03.04 Программная инженерия, 12.03.01 Приборостроение, 23.03.01 Технология транспортных процессов / сост. Л.Н.Буйлушкина. - Нижневартовск, 2017. – 35 с.

20 Маркин, Е.И. Использование реактивного программирования при

разработке мобильных приложений / Е.И. Маркин, К.М. Рябова // Computational nanotechnology. — 2016. — № 2. — С. 170-173. — ISSN 2313-223X. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/journal/issue/298484> (дата обращения - 25.05.2021).

21 Стартапы, бизнес, технологии [Электронный ресурс] – Режим доступа: <https://vc.ru/marketing/245003-statistika-mobilnyh-prilozheniy-2021-zagruzki-trendy-i-dohodnost-industrii> – Загл. с экрана. [дата обращения - 25.05.2021]

## ПРИЛОЖЕНИЯ

### ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ СИСТЕМЫ УЧЕТА КОМПЬЮТЕРНОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

#### 1 Наименование разработки

Разработка мобильного приложения для ресторана доставки «Суши АНТАНО»

#### 2 Основание для разработки

Разработка ведется на основании задания выпускной квалификационной работы по направлению 09.03.04 Программная инженерия.

#### 3 Исполнитель

Исполнитель – Земан Павел Сергеевич

#### 4 Назначение и цель разработки

Разрабатываемое мобильно приложение создаться для осуществления заказов клиентами. Система учета должна позволять быстро и легко осуществлять заказ, и оплачивать его.

#### 5 Содержание работы



5.1.1 Задачи, подлежащие решению

5.1.2 Анализ предметной области.

5.1.3 Определение функций, выполняемых системой.

Продолжение приложения А

5.1.4 Выбор средств разработки.

5.1.5 Проектирование базы данных.

5.1.6 Разработка интерфейса программы.

5.1.7 Оформление документации.

5.2 Требования к программному продукту

5.2.1 Требования к функциональным характеристикам:

Мобильное приложение должно выполнять следующие функции:

1. Возможность осуществления заказа.
2. Возможность оплаты заказа.
3. Получение информации о меню.

5.2.2 Требование к составу программных компонентов

5.4 Требование к архитектуре системы

Система должна включать в себя:

– приложение, производящее работу с базой данных и реализующую интерфейс пользователя;

– базу данных для хранения информации о заказах, клиентах, меню.

5.5 Требование к базе данных

5.6 Требования к входным и выходным данным

### 5.6.1 Входные данные

Входными данными является информация о меню, клиенте, заказах.

Окончание приложения А

### 5.6.2 Выходные данные

Выходными данным является информация о заказе, чек.

### 5.7 Требования к составу и характеристикам технических средств

Приложение должно функционировать на мобильном устройстве следующей конфигурации:

- версия Android от 5.0;
- свободное место на устройстве должно быть больше 100 мегабайт.

#### 5.7.1 Требования к лингвистическому обеспечению

Интерфейс пользователя должен быть реализован на русском языке.

## 6 Требования к документации

Для приема программного продукта должны быть предоставлены следующие документы:

7.1. Техническое задание.

7.2. Пояснительная записка.

## 7 Порядок приема

Прием и контроль программного продукта проводится приемочной комиссией, в состав которой входят представители заказчика и исполнитель, в течение трех рабочих дней после завершения работ.

Дополнительные условия

Данное техническое задание может уточняться и изменяться в установленном порядке.

## ПРИЛОЖЕНИЕ Б. КОМПАКТ-ДИСК

Содержание:

1. Пояснительная записка к выпускной квалификационной работе.
2. Разработанный программный продукт.
3. Презентация.