

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка мобильного приложения на платформе Android для
ООО «Первая типография»

Студент	С. А. Тырсенко	_____
Руководитель	О.В. Аникина	_____
Консультант по аннотации	Н.В. Ященко	_____

Допустить к защите
Заведующий кафедрой к.тех.н., доцент, А.В. Очеповский _____

« _____ » _____ 20 ____ г.

Тольятти 2017

Аннотация

Бакалаврскую работу выполнил студент: Тырсенко С.А.

Тема работы: «Разработка мобильного приложения на платформе Android для ООО «Первая типография».

Научный руководитель: к.т.н., доцент кафедры «Прикладная математика и информатика» О.В. Аникина.

Объект исследования – бизнес-процесс выполнения заказа.

Предмет исследования – частичная автоматизация бизнес-процесса выполнения заказа.

Цель исследования – разработка мобильного приложения на платформе Android для ООО «Первая типография».

Для того чтобы выполнить заданную цель в данной работе нужно справиться со следующими задачами:

- произвести идентификацию и анализ предметной области подвергающейся автоматизации;
- дать обоснование выбору методологии и технологии концептуального моделирования автоматизированной информационной системы;
- определить необходимые требования для автоматизированной информационной системы;
- спроектировать базу данных;
- разработать мобильное приложение.

В ходе выполнения данной работы будет спроектирована и реализована автоматизированная информационная система для осуществления online заказов продукции «Первая типография» посредством мобильного приложения на платформе Android.

Временным интервалом исследования является 2016 – 2017 гг.

Практическая значимость работы заключается в том, что разработанное приложение на платформе Android поможет клиентам оперативно реализовывать их запросы, и они могут быть обработаны специалистами ООО «Первая типография» сразу же после оформления заказа. Вследствие этого

тратится меньше времени на работу с клиентами, увеличивается качество и скорость информационных потоков, существенно экономятся средства и также, увеличивается число клиентов, а соответственно и прибыль.

Первая глава посвящена анализу предметной области, обоснованию выбора методологии и технологии концептуального и логического моделирования мобильного приложения для ООО «Первая типография», а также разработке и анализу модели бизнес-процесса «КАК ЕСТЬ», выявлению минусов имеющегося бизнес-процесса и рекомендациям по его улучшению благодаря современным мобильным технологиям.

Вторая глава посвящена непосредственно концептуальному и логическому моделированию мобильного приложения для ООО «Первая типография». А также разработке модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ», формулировке требований к внедряемой информационной системе, обоснованию и постановке задачи на разработку мобильного приложения.

Третья глава посвящена процессу реализации мобильного приложения с web-интерфейсом, необходимого для управления товарами и обработки заказов клиентов, осуществлённых через приложение.

Данная работа состоит из введения, 3-х разделов, заключения, списка литературы из 31 источников и 3 приложений. Общий объем работы – 55 страниц.

ABSTRACT

The title of the graduation work is "Development of a Mobile Application on the Android Platform for OOO "Pervaya Tipografiya"

The object of the research is the business process of fulfilling the order. The subject of the research is partial automation of the business process of order fulfillment.

The aim of the work is to develop a mobile application on the Android platform for OOO "Pervaya Tipografiya".

The practical importance of the work lies in the fact that the developed application on the Android platform will help customers to implement quickly their requests, and they can be processed by specialists of OOO "Pervaya Tipografiya" immediately after placing an order.

The first chapter is devoted to the analysis of the subject area, the substantiation of the choice of methodology and technology of conceptual and logical modeling of the mobile application for OOO "Pervaya Tipografiya", the development and analysis of the business process model "AS IS", as well as the identification of the shortcomings of the existing business process and recommendations for its improvement with the help of modern mobile technologies.

The second chapter is devoted directly to the conceptual and logical modeling of the mobile application for OOO "Pervaya Tipografiya".

The third chapter describes the process of implementing a mobile application with a web-interface, which is necessary for managing goods and processing orders of customers through the application.

The work consists of an introduction, 3 chapters, conclusion, references which contains 31 sources and 3 applications. Total amount of work - 55 pages.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
Глава 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗИРОВАННОЙ ИНФОРМИЦОННОЙ СИСТЕМЫ.....	6
1.1 Характеристика предприятия ООО «Первая типография» и ее деятельности.....	6
1.2 Выбор технологии анализа проектирования.....	7
1.3 Анализ бизнес-процессов организации ООО «Первая типография»	8
Глава 2 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА БАЗЕ ANDROID ДЛЯ ООО «Первая типография»	11
2.1 Формирование требований к будущей системе.....	11
2.2 Постановка задачи и ее обоснование на разработку автоматизированной информационной системы	11
2.3 Логическое моделирование мобильного приложения на платформе Android для ООО «Первая типография»	14
2.3.1 Разработка диаграммы вариантов использования мобильного приложения.....	14
2.3.2 Диаграмма классов мобильного приложения	20
2.3.3 Разработка логической модели данных мобильного приложения	22
2.4 Физическое моделирование автоматизированной системы для ООО «Первая типография»	23
2.4.1 Выбор архитектуры мобильного приложения.....	23
2.4.2 Выбор сервера базы данных для реализации автоматизированной информационной системы для мобильного приложения ООО «Первая типография».....	25
2.4.3 Выбор языка программирования для реализации мобильного приложения.....	26
2.4.4 Выбор web-сервера для мобильного приложения.....	28
2.4.5 Разработка физической модели данных для мобильного приложения	28
2.4.6 Диаграмма размещения компонентов мобильного приложения	29

3.1 Реализация функциональных требований	31
3.2 Расчет экономической эффективности внедрения мобильного приложения для online заказов продукции ООО «Первая типография».....	38
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	41
ПРИЛОЖЕНИЕ А Основная активность	44
ПРИЛОЖЕНИЕ Б Требования, предъявляемые к будущей системе.	49
ПРИЛОЖЕНИЕ В Главные экономические показатели деятельности ООО «Первая типография» за 2014 – 2016 гг.....	52

ВВЕДЕНИЕ

Коммерческие предприятия направляют все свои ресурсы на выполнение поставленных целей, достижение которых отображает эффективность работы компании и в большей части зависит от рационального применения организационных, трудовых и технологических возможностей.

Рейтингом каждой организации служит показатель дохода предприятия. Увеличению прибыли организации способствует внедрение новых технологий, соответствующих современному рынку.

На сегодняшний день все больше компаний вооружаются современными методами и программным обеспечением для того, чтобы увеличить прирост прибыли организации и, тем самым, твердо держаться на рынке товаров и услуг, несмотря на постоянно возрастающее число конкурентов. Данный факт стал основной причиной для научного поиска наиболее оптимальных решений рассматриваемой проблемы.

Исследование в данной работе преследует следующие цели: освоить теоретические основы, провести анализ и разработку программного обеспечения, удовлетворяющего потребностям выбранной организации в условиях современных тенденций развития бизнеса.

Выраженная выше цель порождает следующие задачи:

1. Освоить необходимые материалы для разработки приложений на платформе Android.
2. Провести исследование состояния и специфических особенностей представительской и сбытовой функций конкретного предприятия в интернете.
3. Выработать рекомендации для применения мобильного приложения в работе организации.

Объектом исследования данной работы является коммерческая компания ООО «Первая типография».

Предметом исследования служит демонстрация и продажа товаров компании, используя современные информационные технологии. Во время

исследования применялись методы: синтез, факторный анализ, статистическая обработка результатов, прогнозирование и программирование.

Реальная ценность данной работы состоит в том, что некоторые её положения в виде приложения на платформе Android могут быть применены специалистами «Первая типография» для улучшения информативности, увеличения заинтересованности, усовершенствования сервиса продуктов и услуг организации.

Структура данной работы включает в себя введение, три раздела, заключение, библиографический список и приложение.

Глава 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗИРОВАННОЙ ИНФОРМИЦОННОЙ СИСТЕМЫ

1.1 Характеристика предприятия ООО «Первая типография» и ее деятельности

ООО «Первая типография» - коммерческая организация - существует на рынке оперативной полиграфии уже более 6 лет, при этом перечень предоставляемых услуг содержит в себе, по сути, все различные виды работ, на которые есть спрос в нашем регионе. Основным достоинством данной компании служит её многофункциональность, благодаря этому предприятие хорошо обосновалось среди многих типографий города Тольятти, а также заработало отличную репутацию у клиентов. Однако опыт ведения бизнеса говорит о том, что в конкурентной борьбе всегда нужно идти в ногу со временем, поэтому необходимо улучшить деятельность ООО «Первая типография».

На рисунке 1.1 изображена организационная структура ООО «Первая типография».

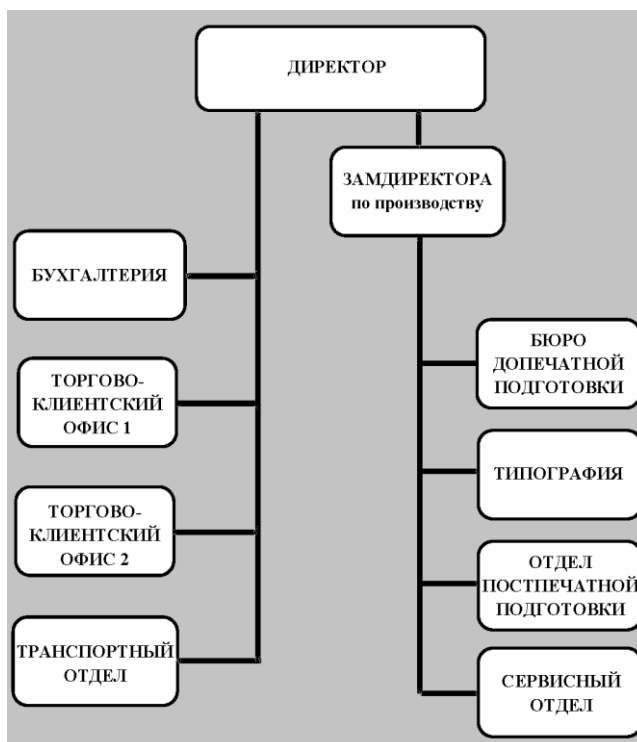


Рисунок 1.1 - Организационная структура ООО «Первая типография»

Основной вид деятельности ООО «Первая типография» является производство полиграфической продукции.

1.2 Выбор технологии анализа проектирования

Первым делом в проектировании информационной системы необходимо определиться с методологией проектирования.

Ознакомимся с описаниями самых популярных методологий:

- RUP;
- ICONIX;
- IDEF0.

Методология IDEF0 стала популярна по причине своей графически понятной и простой нотации.

Основная часть методологии представляет собой диаграммы. Диаграммы демонстрируют функции системы с помощью прямоугольников и существующих связей между внешней средой и функциями. Связи графически представлены в виде стрелок.

Методология IDEF0 базируется на графическом языке описания процессов. Модель в нотации в ней олицетворяет собой комплекс иерархически структурированных и взаимосвязанных диаграмм. Любая диаграмма обозначает единицу описания системы и находится на отдельном листе.

Rational Unified Process является методологией формирования процессов разработки программного продукта.

Процессы, формируемые на базе RUP, варьируются от самых легких - применяемых для малых проектов с коротким циклом разработки - до наиболее сложных процессов, обеспечивающих наиболее обширный перечень требований, может быть, даже разбившихся на группы разработчиков.

Методология ICONIX является чем-то средним между большим рациональным унифицированным процессом и небольшой методологией программирования extreme. Процесс ICONIX составлен на прецедентах. В

процессе также используется язык моделирования UML, тем не менее, основная часть отводится анализу требований.

В итоге было решено применить методологию IDEF0 для проектирования информационной системы.

1.3 Анализ бизнес-процессов организации ООО «Первая типография»

Для обоснования необходимости использования мобильного приложения на платформе Android следует выделить и проанализировать процессы, нуждающиеся в облегчении работы с заявками клиентов. Наиболее понятное представление о данных процессах демонстрирует диаграмма IDEF0.

Концептуально-функциональная модель процесса деятельности «Выполнение заказа» продемонстрирована на рисунке 1.2.



Рисунок 1.2 – Концептуально-функциональная модель деятельности «Выполнение заказа» IDEF0

На предыдущей диаграмме можно увидеть бизнес-модель выполнения заказа в организации ООО «Первая типография» в общем виде. Данный процесс протекает следующим образом: клиент связывается по удобному для

него каналу связи (телефон, электронная почта, сайт и т.д.) или лично посещает и делает заказ, где менеджер данной организации должен понять и, если требуется, посоветовать клиенту, как лучше осуществить его пожелания. После того, как все согласовано в области заказа клиента и производства тиража, берется со склада нужное количество материала для его изготовления. Следующим этапом является процесс печати, а за ним – проверка качества данной продукции. В завершении готовый продукт упаковывается, после чего осуществляется оплата и доставка заказчику.

Следующая диаграмма олицетворяет декомпозицию главного процесса, она продемонстрирована на рисунке 1.3. Каждый блок взаимосвязан и, также все последующие блоки начинаются после завершения предыдущих блоков.

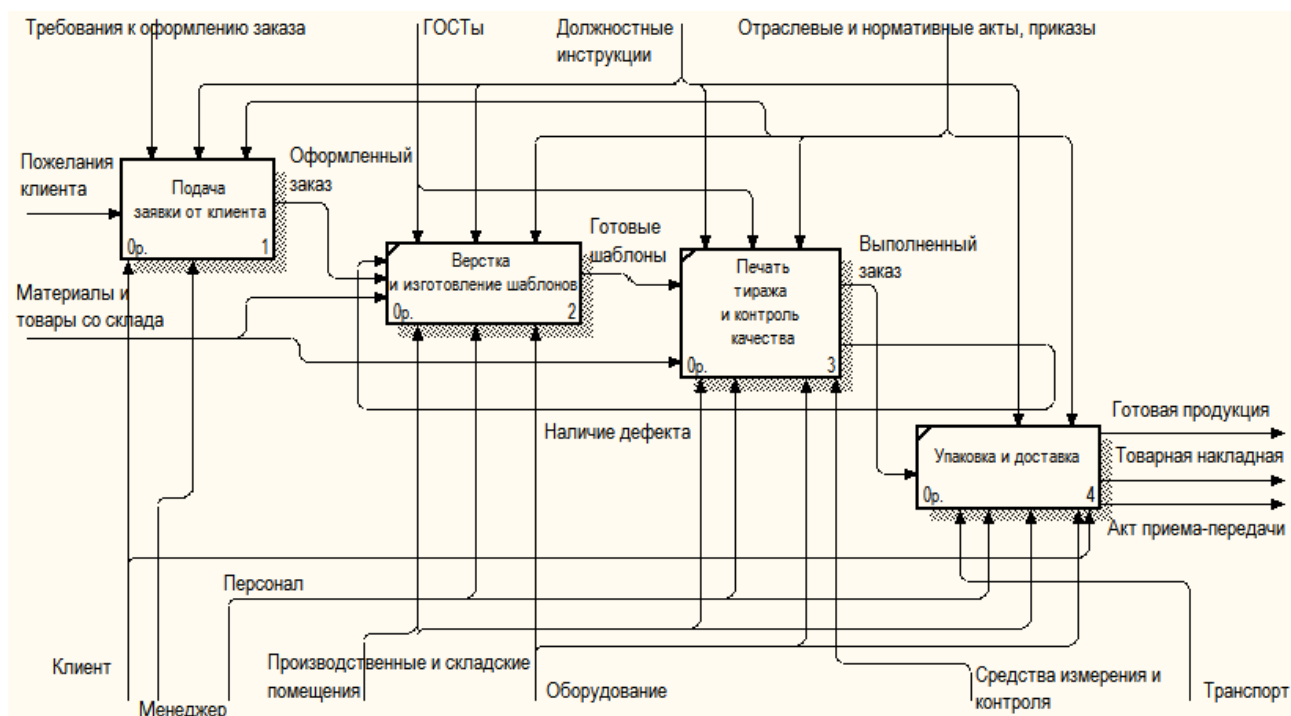


Рисунок 1.3 - Декомпозиция контекстной диаграммы
«Выполнение заказа»

На рисунке видно, что происходят четыре основных процесса:

- подача заявки от клиента;
- верстка и изготовление шаблонов;
- печать тиража и контроль качества;

- упаковка и доставка.

Процесс «Подача заявки от клиента», декомпозиция которого показана на рисунке 1.4, связан с подачей клиента заявки на нужный ему товар, с его необходимым оформлением и утверждением.

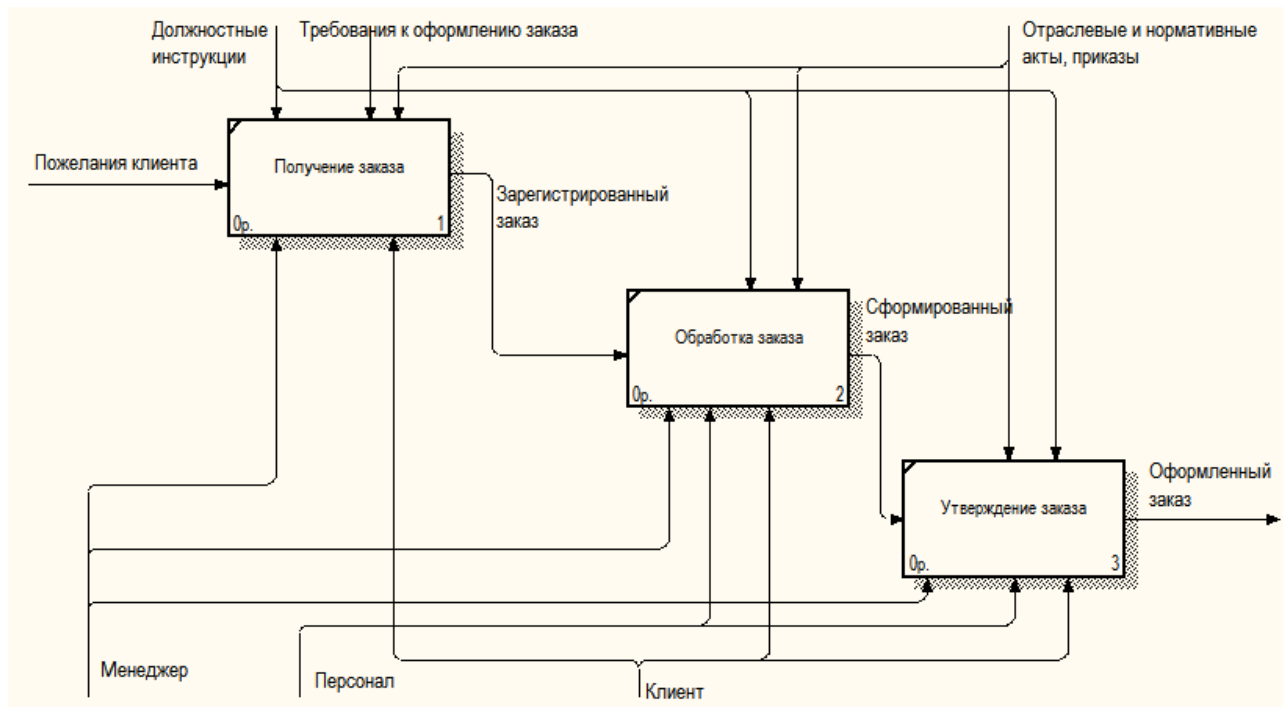


Рисунок 1.4 - Декомпозиция процесса «Подача заявки от клиента»

Исходя из описания бизнес-процессов организации ООО «Первая типография», можно увидеть, что персоналу приходится тратить достаточно много рабочего времени на обработку так называемых «мелких» заказов.

Вывод по первой главе

В итоге выполнения анализа предметной области было проведено сравнение методологий и технологий концептуального моделирования автоматизированной информационной системы.

В результате удалось определить процессы, требующие автоматизации, такие как отправка оформленного заказа от потребителя и его дальнейшая обработка. Частичная автоматизация данных процессов поможет упростить работу персоналу с заказами малого количества.

Глава 2 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА БАЗЕ ANDROID ДЛЯ ООО «Первая типография»

2.1 Формирование требований к будущей системе

Установим главные требования для будущей системы. Требования к ней расписаны по классификации FURPS+ (Таблица 2.1).

Аббревиатура FURPS обозначает следующее:

- Functionality, функционал.
- Usability, простота и удобство применения.
- Reliability, безотказность.
- Performance, продуктивность.
- Supportability, обслуживаемость.

В свою очередь нужно не забывать об ограничениях:

- design, проектирования;
- implementation, разработки;
- interface, интерфейса;
- physical, физические.

Требование определяются как «детальное изложение того, как предполагается реализовать». Есть два типа требований:

- Функциональные требования – это описание того, что система должна выполнять, то есть ее предназначение.
- Нефункциональные требования – это ограничение, представленное системе.

Вследствие анализа контекстов выявлены требования к будущей системе (см. Приложение Б).

2.2 Постановка задачи и ее обоснование на разработку автоматизированной информационной системы

Была сформирована контекстная диаграмма «КАК ДОЛЖНО БЫТЬ» (TO BE), она продемонстрирована на рисунке 2.1.

Модель «КАК ДОЛЖНО БЫТЬ» определяет будущий вид предметной области, в который она обратится вследствие оптимизации существующей системы и внедрения новых технологий.



Рисунок 2.1 - «Выполнение заказа IDEF0 (TO BE)»

Если сопоставить контекстную диаграмму «КАК ДОЛЖНО БЫТЬ», изображенную на рисунке 2.1, с диаграммой «КАК ЕСТЬ», продемонстрированной на рисунке 1.2, можно увидеть, что механизмов стало восемь потоков: персонал, производственные и складское помещение, оборудование, средства измерения и контроля, транспорт, клиент, менеджер, мобильное приложение.

Декомпозиция контекстной диаграммы представлена на рисунке 2.2. Теперь некоторая часть заказов, а именно «мелких» заказов осуществляется через мобильное приложение, что способствует значительному сокращению нагрузки на менеджера предприятия и тем самым у него появляется больше свободного времени, а значит и возможностей.

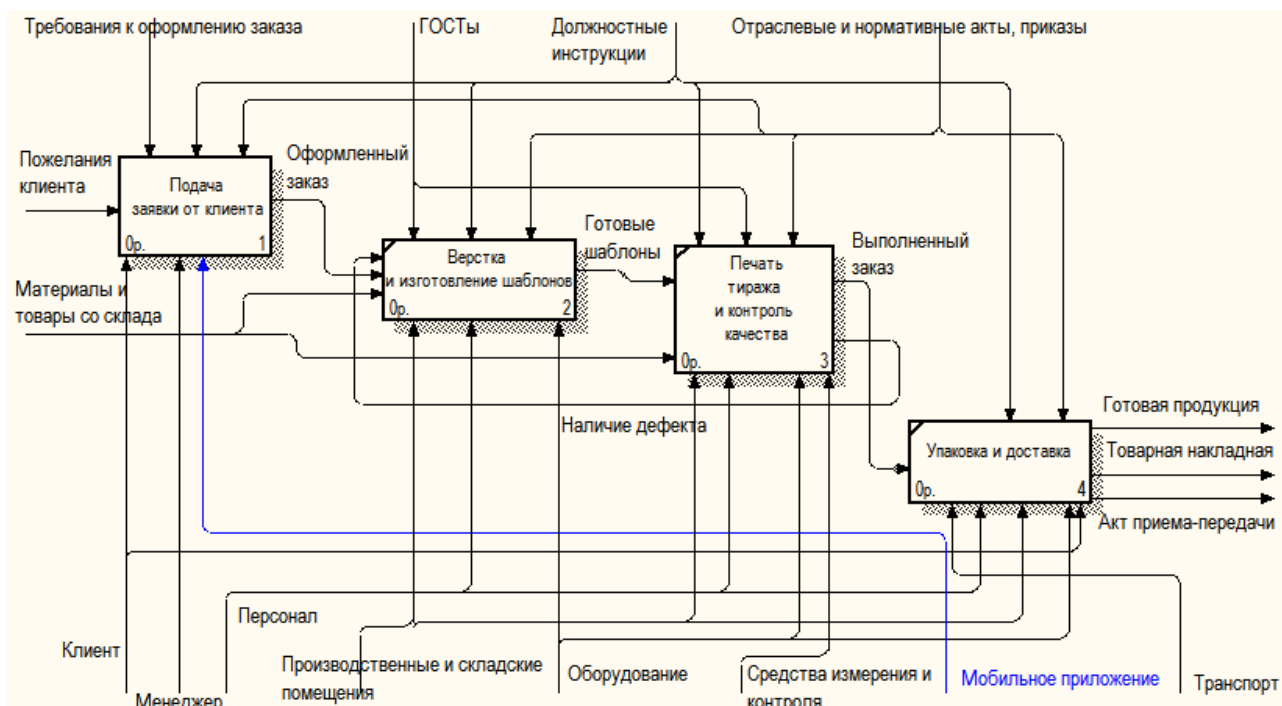


Рисунок 2.2 – Декомпозиция «Выполнение заказа»

На текущей фазе была проиллюстрирована декомпозиция главного процесса «Выполнение заказа».

Будущая автоматизированная информационная система обязана обеспечивать следующие функции:

- добавление новых товаров, а также их удаление и редактирование;
- создание заявки на продукцию;
- регистрация клиентов;
- обратная связь с клиентами.

Сначала пользователь выбирает нужную ему категорию товаров, после чего находит интересующий его продукт, так же он может воспользоваться поиском по наименованию. После этого он выбирает нужное ему количество товара и добавляет в его корзину. Далее пользователь может редактировать корзину, и после того, как он выбрал нужные ему товары, он может оформить заявку на приобретение выбранной им продукции, а также есть возможность регистрации, позволяющая значительно сократить время заполнения основных полей с данными о доставке.

2.3 Логическое моделирование мобильного приложения на платформе Android для ООО «Первая типография»

В итоге логического моделирования образуется логическая модель и объектная модель данных мобильного приложения.

Во время логического моделирования мобильного приложения применяется методология анализа и объектно-ориентированного проектирования, составленная на языке UML.

2.3.1 Разработка диаграммы вариантов использования мобильного приложения

Опишем будущую систему подачи заявки от потребителя с точки зрения диаграммы вариантов использования. Благодаря диаграмме вариантов использования появляется возможность обозначить внешние системы, взаимодействующие с системой, главные процессы и их взаимосвязанность. Диаграммы вариантов использования предоставляют возможность обозначить функциональную структуру системы, не вникая в детали ее реализации. А также происходит предварительное выявление и классификация системных объектов. В результате сформированной модели создается план разработки системы.

Для имеющейся предметной области обозначим актеров:

- пользователь;
- менеджер.

Определим, какие возможности обязана иметь разрабатываемая система:

- пользователь применяет систему для осуществления регистрации и подачи заявки на приобретение товаров;
- менеджер системы выполняет контроль поведения системы, наделяет описанием товары, редактирует данные.

Автоматизированная информационная система осуществляет хранение данных и предоставляет к ним доступ посредством сети Интернет с возможностью поиска необходимой информации. Названные выше пункты

позволяют обозначить следующие прецеденты (таблица 2.1), которые необходимо реализовать в будущей системе.

Таблица 2.1 - Краткое описание прецедентов

Прецеденты	Краткое описание
1	2
Регистрация	Регистрация пользователя в мобильном приложении
Добавление товаров	Добавление товаров с описанием и их количеством
Подача заявки на приобретение товаров	Формирование заявки и ее отправление исполнителю
Получение результата после отправки заявки	Предоставляет пользователю информацию о результате выполненного заказа
Предоставление справочной информации о приложении	Предоставление справочной информации о использовании мобильного приложения
Редактирование данных	Редактирование информации о продукции

Разработанная диаграмма вариантов использования для основных прецедентов проектируемой системы представлена на рисунке 2.3, где актеры, изображенные на ней – это группа лиц или другие системы, взаимодействующие с данной системой, а прецеденты (варианты использования) – это сервисы, обеспечиваемые системой. Сплошными линиями на диаграмме использования являются отношения ассоциаций, олицетворяющие способность применения актером прецедента. Также, на диаграмме использования можно заметить границы системы.

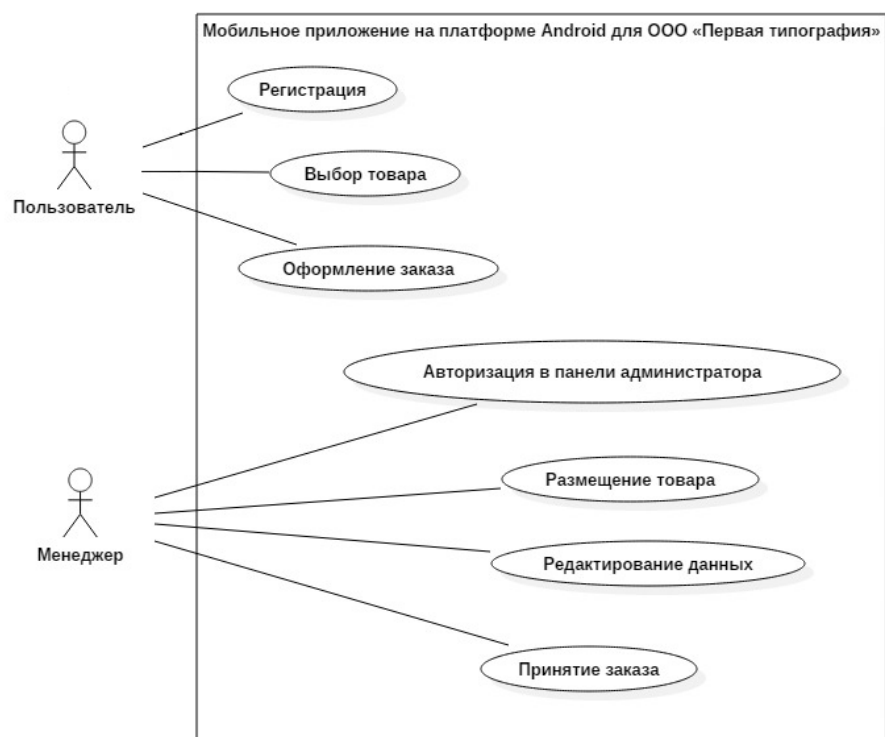


Рисунок 2.3 - Диаграмма вариантов использования

Опишем спецификации для главных прецедентов в табличном виде представления. А также в таблицах 2.2–2.8 продемонстрированы спецификации главных прецедентов.

Таблица 2.2 - Описание прецедента «Регистрация»

Прецедент: Регистрация
ID: 1
Краткое описание: Регистрация клиента
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловия: 1. Прецедент начинается по инициативе пользователя
Основной поток: 1. Пользователь запускает мобильное приложение. 2. Пользователь регистрируется в «Личном кабинете». 3. Вводит данные для регистрации.

Продолжение таблицы 2.2

<p>4. Если все поля для ввода заполнены корректно.</p> <p>4.1. Нажимает кнопку «Зарегистрироваться».</p> <p>4.2. Вводит необходимые данные для доставки товара.</p> <p>4.3. Нажимает кнопку «Сохранить».</p>
<p>Постусловия:</p> <p>1. Данные введены и актуальны</p>

Таблица 2.3 - Описание прецедента «Выбор товара»

Прецедент: Выбор товара
ID: 2
<p>Краткое описание:</p> <p>Пользователь просматривает перечень доступных товаров</p>
Главные актеры: Пользователь
<p>Второстепенные актеры:</p> <p>Нет</p>
<p>Предусловия:</p> <p>1. Прецедент начинается по инициативе пользователя</p>
<p>Основной поток:</p> <p>1. Пользователь запускает мобильное приложение.</p> <p>2. Входит в раздел «Товары».</p> <p>3. Выбирает нужный ему товар, а также его количество.</p> <p>4. Подтверждает перечень выбранных товаров в корзине.</p>
<p>Постусловия:</p> <p>1. Пользователь переходит к оформлению заказа.</p>

Таблица 2.4 - Описание прецедента «Оформление заказа»

<p>Прецедент:</p> <p>Оформление заказа</p>
ID: 3
<p>Краткое описание:</p> <p>Пользователь вводит необходимые данные для оформления заказа</p>
<p>Главные актеры:</p> <p>Пользователь</p>

Продолжение таблицы 2.4

Второстепенные актеры: Нет
Предусловия: 1. Прецедент начинается по инициативе пользователя
Основной поток: 1. Пользователь вводит нужные данные для оформления заказа. 2. Нажимает кнопку «Отправить».
Постусловия: 1. Пользователь получает сведения о дальнейших действиях.

Таблица 2.5 - Описание прецедента «Размещение товара»

Прецедент: Размещение товара
ID: 4
Краткое описание: Менеджер вводит необходимые данные для размещения товара
Главные актеры: Менеджер
Второстепенные актеры: Нет
Предусловия: 1. Прецедент начинается по инициативе пользователя
Основной поток: 1. Менеджер заполняет нужные поля для добавления нового товара. 2. Нажимает кнопку «Добавить».
Постусловия: Нет.

Таблица 2.6 - Описание прецедента «Редактирование данных»

Прецедент: Редактирование данных
ID: 5
Краткое описание:

Продолжение таблицы 2.6

Менеджер редактирует необходимые ему данные связанные с товаром
<p>Главные актеры:</p> <p>Менеджер</p>
<p>Второстепенные актеры:</p> <p>Нет</p>
<p>Предусловия:</p> <p>1. Прецедент начинается по инициативе пользователя</p>
<p>Основной поток:</p> <p>1. Менеджер заходит в «Панель администратора».</p> <p>2. Менеджер редактирует нужные ему поля.</p> <p>3. Нажимает кнопку «Обновить».</p>
<p>Постусловия:</p> <p>Нет.</p>

Таблица 2.7 - Описание прецедента «Принятие заказа»

<p>Прецедент:</p> <p>Принятие заказа</p>
ID: 6
<p>Краткое описание:</p> <p>Менеджер получает информацию о заказе клиента.</p>
<p>Главные актеры:</p> <p>Менеджер</p>
<p>Второстепенные актеры:</p> <p>Нет</p>
<p>Предусловия:</p> <p>1. Прецедент начинается по инициативе пользователя</p>
<p>Основной поток:</p> <p>1. Менеджер входит в «Панель администратора».</p> <p>2. Переходит в раздел «Заказы».</p> <p>3. Получает данные о заказе клиента.</p>
<p>Постусловия:</p> <p>1. Связывается с клиентом.</p>

Таблица 2.8 - Описание прецедента «Авторизация в панели администратора»

Прецедент: Авторизация в панели администратора
ID: 7
Краткое описание: Менеджер вводит логин и пароль для входа в панель администратора.
Главные актеры: Менеджер
Второстепенные актеры: Нет
Предусловия: 1 Прецедент начинается по инициативе пользователя
Основной поток: 4. Менеджер вводит логин и пароль в соответствующие поля. 5. Нажимает кнопку «Войти».
Постусловия: 2. Менеджер входит в «Панель администратора».

В результате данного этапа были выявлены главные роли, которые способствуют ограничению доступа к набору функций и прав потребителей.

2.3.2 Диаграмма классов мобильного приложения

Опираясь на диаграмму вариантов использования, была создана диаграмма классов (Рисунок 2.5), олицетворяющая классы, их операции и атрибуты, связи между ними.

На диаграмме представлены основные классы: *User*, *Category*, *Product*, *Order*, также интерфейс *Mobile App*.

Класс *User* имеет возможность просматривать, выбирать товары и производить заказы через *Mobile App*, данный класс хранит в себе данные такие как: имя, адрес, город, регион, номер телефона, эл. почта, пароль для авторизации и список выбранных им товаров.

Класс *Category* содержит виды продукции организации, а именно их наименование и изображение категории.

Класс *Product* содержит данные о товаре, такие как: наименование, категорию, цену товара, наличие товара на складе, его количество и его описание.

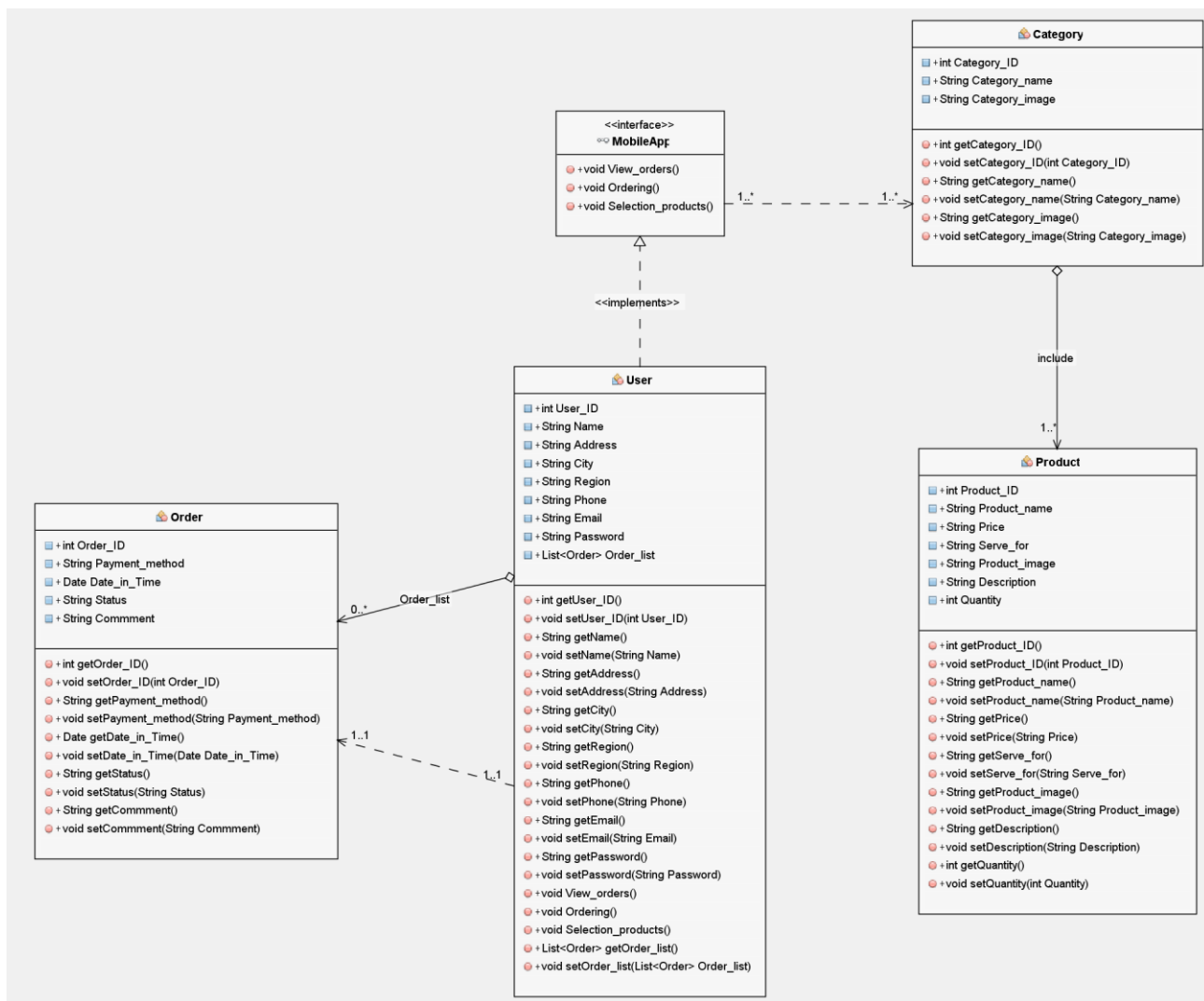


Рисунок 2.5 – Диаграмма классов

Mobile App предоставляет возможность клиенту просматривать товары, выбирать необходимую ему продукцию в нужном ему количестве, а также оформлять заказ.

Order хранит в себе данные о заказе сделанным пользователем например: способ оплаты, дату доставки, статус заказа и комментарий.

2.3.3 Разработка логической модели данных мобильного приложения

Процедура логического моделирования данных информационной системы представляет собой связи и интеграции двух видов технологий: объектной и реляционной.

На сегодняшний день системы управления базами данных считаются главным способом реализации баз данных и их хранилищ.

Для конструирования реляционных БД применяется такая методология, как IDEF1X.

В IDEF1X модель данных описывается на физическом и логическом уровнях.

Физическая модель данных включает в себя определения объектов, реализуемых в обозначенной СУБД и БД.

Логическая модель данных представляется свободным логическим образом концептуальной модели данных информационной системы.

На рисунке 2.6 изображена логическая модель данных, построенная на основе диаграммы классов.

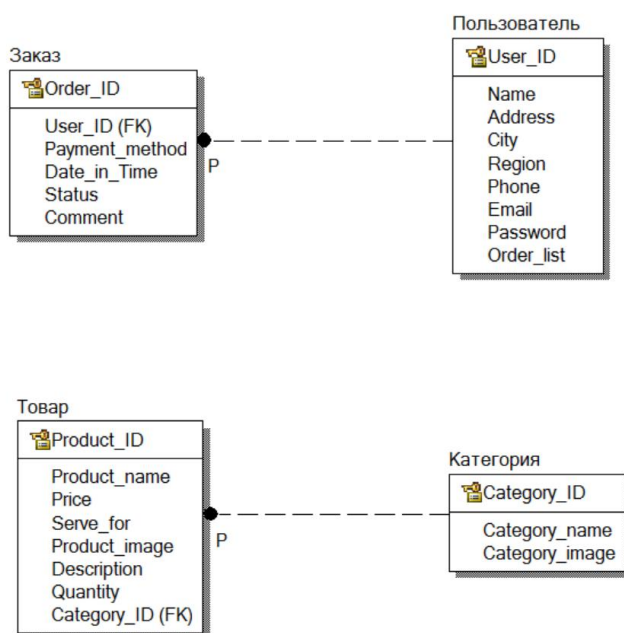


Рисунок 2.6 – Логическая модель данных

Используемый метод к моделированию данных основан на концепции «сущность - связь» и применяется для анализа информационных структур систем, имеющих разную направленность. Информационная модель отображает информацию о системных объектах в виде логической структуры.

На диаграмме проиллюстрированы следующие сущности: *Заказ*, *Пользователь*, *Категория* и *Товар* со связями различной кратности.

Один товар может принадлежать только к одной категории товаров, что показано связью типа «один-к-одному» между сущностями *Категория* и *Товар*, но одна категория может содержать несколько товаров, что отображено связью вида «один-ко-многим».

2.4 Физическое моделирование автоматизированной системы для ООО «Первая типография»

2.4.1 Выбор архитектуры мобильного приложения

Современные мобильные приложения реализуются в архитектуре «клиент-сервер».

Архитектура «клиент-сервер» исполняется в следующих вариантах:

- 1) двухзвенная архитектура «клиент - сервер».

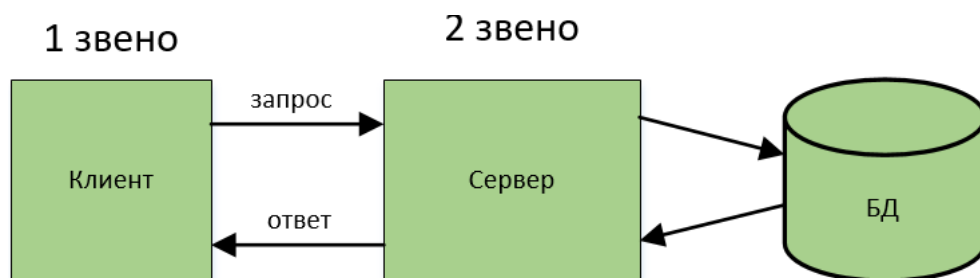


Рисунок 2.7 – Двухзвенная клиент-серверная архитектура

Первое звено – смартфоны пользователей с прикладными приложениями, благодаря которым потребители взаимодействуют с БД через сеть («толстый клиент»).

Второе звено - сервер баз данных - принимает участие в обработке информации.

Двухзвенная архитектура уменьшает нагрузку на информационную сеть по причине передачи запросов и ответов.

Плюсы:

- легкая реализация;
- дешевизна серверного оборудования.

Минусы:

- применение отсталых IT-решений;

2) трехзвенная архитектура «клиент - сервер».

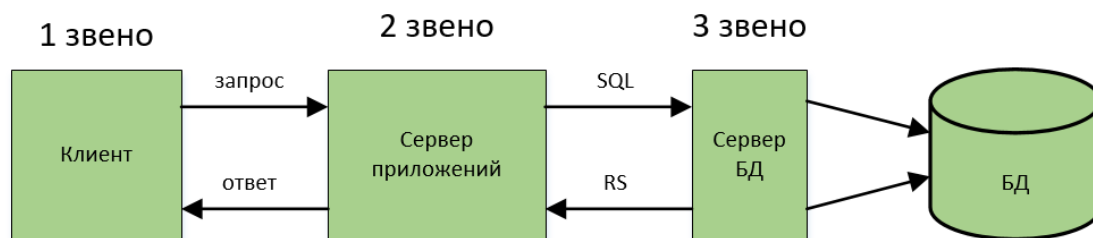


Рисунок 2.8 – Трехзвенная клиент-серверная архитектура

Первое звено – мобильное приложение на платформе Android – «тонкий клиент».

Второе звено – сервер приложений – программный продукт переходного уровня. Синхронизирует работу компонентов системы и создает их взаимосвязь. В основном, данную задачу выполняют программные обеспечения и сетевые службы.

Третье звено – сервер баз данных - не взаимодействует с клиентом напрямую, вследствие чего повышается безопасность системы.

Плюсы:

- возможность масштабируемости;
- простота конфигурируемости.

Минусы:

- сложность производства программного продукта;
- относительно высокие требования к продуктивности сервера базы данных и серверов приложений.

Итоги сравнительного анализа архитектур продемонстрированы в таблице 2.9.

Таблица 2.9- Сопоставление архитектур систем

Характеристика	Клиент-сервер, двухзвенная модель	Клиент-сервер, трехзвенная модель
Масштабируемость	-	+
Конфигурируемость	-	+
Минимальные затраты на разработку и эксплуатацию	+	-
Минимальные требования к оборудованию	+	-
Поддержка современных технологий	-	+
Сумма баллов	2	3

По результатам анализа предлагается использовать при реализации мобильного приложения трехзвенную архитектуру «клиент-сервер».

2.4.2 Выбор сервера базы данных для реализации автоматизированной информационной системы для мобильного приложения ООО «Первая типография».

Проанализируем бесплатные системы управления базами данных FireBird, PostGreSQL и MySQL, а также сравним их.

PostgreSQL - поддерживает архитектуру «клиент-сервер» с разграничением процессов среди пользователей. Минусом данной системы управления базами данных есть низкая скорость выполнения команд манипулирования данными при больших информационных нагрузках.

MySQL – обладает высокой скоростью выполнения операций, надежностью и простотой использования, служит хорошим решением для небольших и средних программ.

Firebird – кросс-платформенная система управления базами данных. Главным минусом ее является отсутствие наличия кэша для результатов запросов, полнотекстовых индексов, ощутимое снижение продуктивности при увеличении внутренней фрагментации базы.

Итоги сравнительного анализа систем управления базами данных для мобильного приложения продемонстрированы в таблице 2.10.

Таблица 2.10- Сопоставление СУБД

Характеристики	PostgreSQL	MySQL	FireBird
Быстродействие	-	+	-
Поддержка транзакций	+	+	+
Поддержка реляционных баз данных	+	+	+
Сумма баллов	2	3	2

Из данных предыдущей таблицы можно сделать заключение, что СУБД MySQL предоставляет лучшую производительность, в связи с этим фактом выбираем ее.

2.4.3 Выбор языка программирования для реализации мобильного приложения

Для разработки программного продукта в пределах спиральной модели его жизненного цикла во время выбора средств разработки главное учесть критерии:

- поддержка технологии быстрой разработки программного продукта, базируемой на объектно-ориентированном программировании;
- поддержка трехзвенной архитектуры «клиент-сервер»;
- владение языка разработчиком программного обеспечения;
- поддержка системы управления базами данных MySQL;

На существующий момент для разработки мобильного программного обеспечения используются следующие популярные языки программирования: Java, Objective-C.

Java – объектно-ориентированный, мультиплатформенный язык программирования. Применяется для разработки приложений на таких платформах как Android. К недостаткам относится возможное снижение скорости выполнения приложений.

Objective-C – это строгое расширение языка Си. Применяется при разработке на базе Apple.

Результаты сравнительного анализа технологий программирования для разработки мобильного приложения приведены в таблице 2.11

Таблица 2.11 - Сравнительный анализ технологий программирования

Характеристики (баллы)	Java	Objective-C
Объектно-ориентированный подход к разработке ПО	+	+
Используется для разработки на платформе Android	+	-
Поддержка трехзвенной архитектуры «клиент-сервер»	+	+
Знание языка разработчиком	+	-
Поддержка СУБД MySQL	+	+
Сумма баллов	5	3

В итоге сравнительного анализа делаем выбор в пользу языка программирования Java. в качестве основного средства разработки мобильного приложения для ООО «Первая типография».

Для разработки ПО использована локальная среда разработки Java – Android Studio.

Android Studio предоставляет возможность запустить приложение, как на эмуляторе, так и на реальном устройстве. Установка данной среды разработки

очень проста. Она хорошо подходит для новичков в программирование на языке Java. Android Studio выделяется такими качествами: удобный и приятный интерфейс, автоматическая загрузка необходимых компонентов для разработки приложения и наличие большого количества справочной информации.

2.4.4 Выбор web-сервера для мобильного приложения

Осуществим сравнение бесплатных web-сервисов: Nginx, Apache, Tornado.

Apache – это независимое от платформы программное обеспечение. Главным плюсом ее является гибкость конфигураций и устойчивость к отказам.

Lighttpd – создавался соответственно сегодняшним требованиям с целью получения высокого быстродействия и безопасности. Минус Tornado состоит в том, что его сложно настроить и конфигурировать.

Итоги сравнения web-серверов продемонстрированы в таблице 2.13.

Таблица 2.12 - Сравнительный анализ web-серверов

Характеристики	Apache	Lighttpd	Tornado
Использование ресурсов	+	-	-
ПоддержкаPHP	+	+	+
Легкость настройки	+	-	-
Популярность	+	-	-
Сумма баллов	4	1	1

Как следует из таблицы 2.12, наилучшими характеристиками обладает web-сервер Apache, что и послужит основным поводом для его выбора.

2.4.5 Разработка физической модели данных для мобильного приложения

Физическая модель данных составляет отражение системного каталога и обуславливается конкретной системой управления базами данных.

В физической модели находится информация всех объектов баз данных.

На рисунке 2.9 продемонстрирована физическая модель базы данных мобильного приложения, сформированная для системы управления базами данных MySQL.

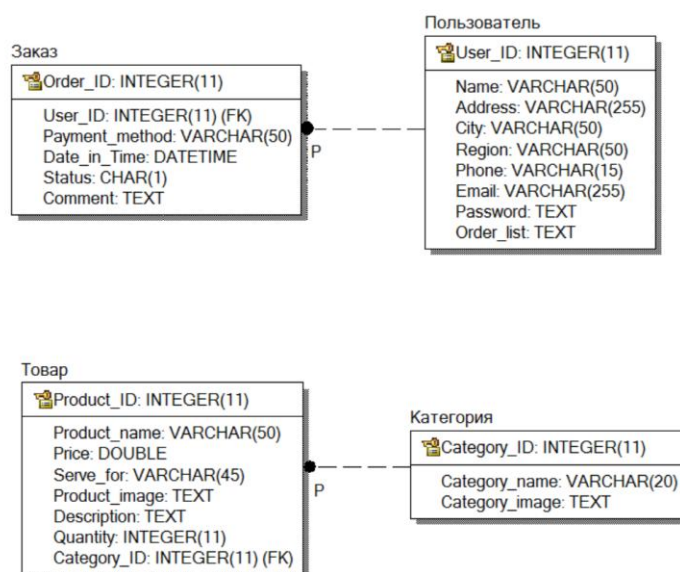


Рисунок 2.9 – Физическая модель данных

Физическая модель данных сформирована опираясь на логическую модель данных системы.

2.4.6 Диаграмма размещения компонентов мобильного приложения

Информационная система сформирована из компонентов:

- клиент – мобильное приложение на платформе Android;
- web-сервер – реализован на базе Apache;
- сервер баз данных –MySQL.

Доступ к серверам мобильного приложения осуществляется через Интернет.

Диаграмма компонентов дает возможность сформировать архитектуру будущей системы, обозначив соотношения программных компонентов. Главными графическими элементами диаграммы являются: интерфейсы и зависимости между ними, а также компоненты.

Сформированная диаграмма компонентов продемонстрирована на рисунке 2.10.

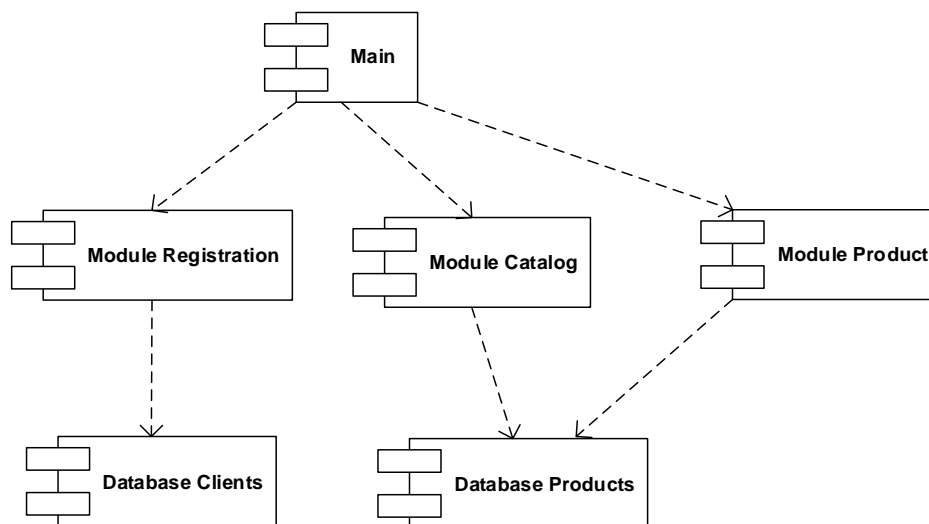


Рисунок 2.10 - Диаграмма компонентов

Вследствие чего, была сформирована диаграмма компонентов, иллюстрирующая модули аппаратного и программных уровней, и их взаимосвязь.

Вывод по второй главе

В данной главе было осуществлено проектирование АИС, где были проанализированы логические и физические моделирования БД. Рассмотренные выше пункты устанавливают структуру БД, нужные для её создания.

Создана концептуальная модель потоков данных для процесса отправки заказов от потребителя, опираясь на которую были определены функциональные требования, предъявляемые к автоматизированной информационной системе для подачи заявки клиентом, связанные с созданием мобильного приложения для отображения данных товара, и созданием (отправки) заявки на приобретение продукции ООО «Первая типография».

Глава 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ООО «Первая типография»

3.1 Реализация функциональных требований

Во время разработки мобильного приложения были достигнуты задачи:

Для начала необходимо зайти в Android Studio и создать проект с выдвинутой шторкой Navigation Drawer Activity (Рисунок 3.1), она необходима для хранения в себе навигационного меню мобильного приложения.

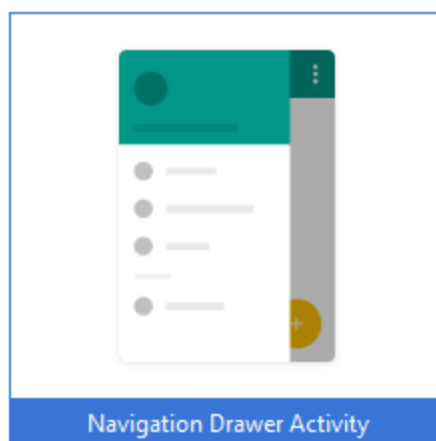


Рисунок 3.1 – Шаблон Navigation Drawer Activity

Далее приведен фрагмент программного кода (`MainActivity.java`), в котором происходит инициализация элементов в навигационном меню (Рисунок 3.2), а также на рисунке 3.3 продемонстрирован графический интерфейс приложения с «выдвинутой шторкой» на которой расположены все элементы навигационного меню, такие как: *Главная, Товары, Корзина, Личный кабинет, Информация, О программе, Поделиться, Контакты, Выход*. В разделе *Информация* хранится инструкция по применению программы. Раздел *О программе* предоставляет собой информацию о программе такую как: логотип программы, версию программного обеспечения, эл. почту и т.д. Для обратной связи создан раздел *Контакты*, где пользователь может написать письмо менеджеру в организацию ООО «Первая типография».

```
83     navDrawerItems = new ArrayList<NavDrawerItem>();
84
85     // Добавление навигационных элементов в массив
86     // Главная
87     navDrawerItems.add(new NavDrawerItem(navMenuTitles[0], navMenuIcons.getResourceId(0, -1)));
88     // Товары
89     navDrawerItems.add(new NavDrawerItem(navMenuTitles[1], navMenuIcons.getResourceId(1, -1)));
90     // Корзина
91     navDrawerItems.add(new NavDrawerItem(navMenuTitles[2], navMenuIcons.getResourceId(2, -1)));
92     // Личный кабинет
93     navDrawerItems.add(new NavDrawerItem(navMenuTitles[4], navMenuIcons.getResourceId(4, -1)));
94     // Информация
95     navDrawerItems.add(new NavDrawerItem(navMenuTitles[5], navMenuIcons.getResourceId(5, -1)));
96     // О программе
97     navDrawerItems.add(new NavDrawerItem(navMenuTitles[6], navMenuIcons.getResourceId(6, -1)));
98     // Поделиться
99     navDrawerItems.add(new NavDrawerItem(navMenuTitles[7], navMenuIcons.getResourceId(7, -1)));
100    // Контакты
101    navDrawerItems.add(new NavDrawerItem(navMenuTitles[8], navMenuIcons.getResourceId(8, -1)));
102    // Выход
103    navDrawerItems.add(new NavDrawerItem(navMenuTitles[9], navMenuIcons.getResourceId(9, -1)));
104
105
```

Рисунок 3.2 – Фрагмент программного кода инициализации элементов навигационного меню

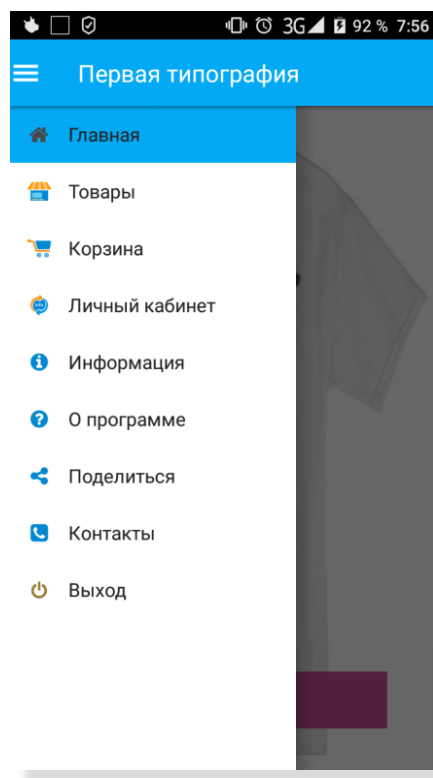


Рисунок 3.3 – Графический интерфейс навигационного меню приложения

После необходимо прописать код, отвечающий за реакцию на выбор того или иного элемента навигации (Рисунок 3.4)

```

276 private void displayView(int position) {
277
278     Fragment fragment = null;
279     switch (position) {
280     case 0:
281         fragment = new ActivityHome(); //активность отвечающая за отображение слайдера с изображениями предложений товаров
282         break;
283     case 1:
284         startActivity(new Intent(getApplicationContext(), ActivityCategoryList.class)); //запускает активность с категорией товаров
285         overridePendingTransition(R.anim.open_next, R.anim.close_next);
286         break;
287     case 2:
288         startActivity(new Intent(getApplicationContext(), ActivityCart.class)); //запускает активность Корзина
289         overridePendingTransition(R.anim.open_next, R.anim.close_next);
290         break;
291     case 3:
292         startActivity(new Intent(getApplicationContext(), ActivityProfile.class)); //запускает активность Личный кабинет
293         overridePendingTransition(R.anim.open_next, R.anim.close_next);
294         break;
295     case 4:
296         startActivity(new Intent(getApplicationContext(), ActivityInformation.class)); // запускает активность Информация
297         overridePendingTransition(R.anim.open_next, R.anim.close_next);
298         break;
299     case 5:
300         startActivity(new Intent(getApplicationContext(), ActivityAbout.class)); //запускает активность с информацией о Программе
301         overridePendingTransition(R.anim.open_next, R.anim.close_next);
302
303         break;

```

Рисунок 3.4 – Фрагмент программного кода отвечающего за обработку выбора элемента в навигационном меню

Далее рассмотрим программный код активности, отвечающий за оформление заказа пользователем. Который проверяет поля на наличие необходимых данных для оформления заказа с последующей отправкой его на сервер (Рисунок 3.5). После, продемонстрирован графический интерфейс данного фрагмента (Рисунок 3.6), на котором размещены поля для ввода: имени, адреса, города проживания, региона, номера телефона, электронной почты и комментария к заказу. Клиент может выбрать удобное для него время и дату доставки, нажав на кнопку «Дата доставки» и «Время доставки». После того как клиент заполнил все поля, он может пересмотреть список заказанных им продуктов с их стоимостью, включая НДС. После отправки оформленного заказа, клиента автоматически «перебрасывает» на экран «Заказ отправлен», где его благодарят за сделанный заказ и дают инструкции для дальнейших действий по получению выбранной им продукции ООО «Первая типография».

```

206 // слушает событие для обработки кнопки отправить заказ
207 btnSend.setOnClickListener((arg0) -> {
210 // TODO Auto-generated method stub
211
212 //получает все данные необходимые для отправки на сервер
213 Name = edtName.getText().toString();
214 Adress = edtAlamat.getText().toString();
215 City = edtKota.getText().toString();
216 Region = edtProvinsi.getText().toString();
217 Email = edtEmail.getText().toString();
218 Name2 = edtName2.getText().toString();
219 Date = btnDate.getText().toString();
220 Time = btnTime.getText().toString();
221 Phone = edtPhone.getText().toString();
222 Comment = edtComment.getText().toString();
223 Date_n_Time = Date+" "+Time;
224 if(Name.equalsIgnoreCase("") || Name2.equalsIgnoreCase("") || Email.equalsIgnoreCase("") || Alamat.equalsIgnoreCase("") ||
225 Date.equalsIgnoreCase("Дата доставки") ||
226 Time.equalsIgnoreCase("Время доставки") ||
227 Phone.equalsIgnoreCase("")){
228 Toast.makeText(ActivityCheckout.this, "Пожалуйста заполните поля.", Toast.LENGTH_SHORT).show();
229 }else if((data.size() == 0)){
230 Toast.makeText(ActivityCheckout.this, "У вас нет заказов", Toast.LENGTH_SHORT).show();
231 }else{
232 new sendData().execute();
233
234
235
236 });
237

```

Рисунок 3.5 – Фрагмент программного кода, отвечающего за обработку данных в полях необходимых для заказа

Рисунок 3.6 – Графический интерфейс оформления заказа

Для управления товарами и обработкой заказов клиентов разработана панель администратора, в которую может войти только менеджер предприятия,

так как он осуществляет взаимодействие с клиентами. Вход в панель администратора продемонстрирован на рисунке 3.7, где менеджеру необходимо ввести логин и пароль.

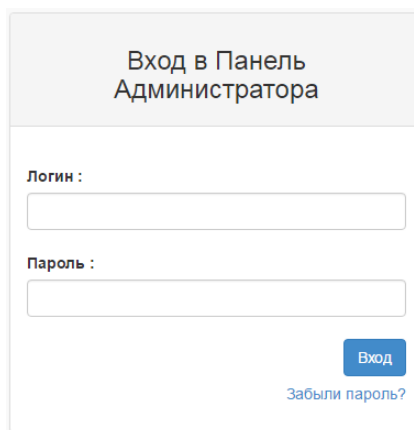


Рисунок 3.7 – Вход в панель администратора

На следующих изображениях предоставлен код с детальными пояснениями (Рисунок 3.8).

```
1 <?php
2 include_once('includes/connect_database.php');
3
4 // Начало сеанса
5 //session_start();
6
7 // Если пользователь нажал кнопку «Вход»
8 if(isset($_POST['btnLogin'])){
9
10 //Получаем имя пользователя и пароль
11 $username = $_POST['username'];
12 $password = $_POST['password'];
13
14 // Установить время ожидания сеанса
15 $currentTime = time() + 25200;
16 $expired = 3600;
17
18 // Создать переменную массива для обработки ошибки
19 $error = array();
20
21 // Проверяем, является ли $username пустым или нет
22 if(empty($username)){
23     $error['username'] = "*Username should be filled.";
24 }
25
26 // Проверяем, является ли $password пустым или нет
27 if(empty($password)){
28     $error['password'] = "*Password should be filled.";
29 }
30
31 // Если имя пользователя и пароль не пустые, проверяем в базе данных
32 if(!empty($username) && !empty($password)){
33
34     // Изменить имя пользователя на нижний регистр
```

Рисунок 3.8 – Первый фрагмент программного кода для входа в панель администратора

Продолжение кода авторизации в панели администратора продемонстрировано на рисунке 3.9.

```
35 $username = strtolower($username);
36
37 //Зашифровать пароль sha256
38 $password = hash('sha256', $username.$password);
39
40 // Получаем данные из таблицы пользователя
41 $sql_query = "SELECT *
42 FROM tbl_user
43 WHERE username = ? AND password = ?";
44
45 $stmt = $connect->stmt_init();
46 if($stmt->prepare($sql_query)) {
47     // ?s
48     $stmt->bind_param('ss', $username, $password);
49     // Выполняем запрос
50     $stmt->execute();
51
52     $stmt->store_result();
53     $num = $stmt->num_rows;
54
55     $stmt->close();
56     if($num == 1){
57         $_SESSION['user'] = $username;
58         $_SESSION['timeout'] = $currentTime + $expired;
59         header("location: dashboard.php");
60     }else{
61         $error['failed'] = "<span class='label label-danger'>Invalid Username or Password!</span>";
62     }
63 }
64
65 }
66 }
```

Рисунок 3.9 – Второй фрагмент программного кода для входа в панель администратора

В случае успешного прохождения авторизации менеджер попадает в панель инструментов (Рисунок 3.10), где перейдя в раздел «Заказы» он может увидеть таблицу со списком существующих на данный момент заказов клиентов (Рисунок 3.11).

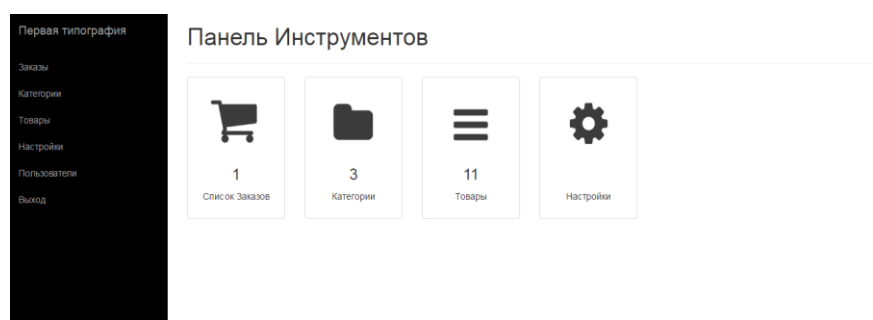


Рисунок 3.10 – Панель инструментов

Первая типография
 Заказы
 Категории
 Товары
 Настройки
 Пользователи
 Выход

Список Заказов

Искать по наименованию :

Имя	Адрес	Эл. почта	Способ оплаты	Дата и время	Телефон	Статус	Действие
Степан	московский 13	t.stepan774@yandex.ru	Оплата наличными	2017-04-22 02:56:00	89879682943	НЕ ОБРАБОТАННЫЙ	Подробнее Удалить

1

(total 1)

Рисунок 3.11 – Раздел заказов




Также в панели администратора предоставлена возможность редактирования, добавления и удаления категорий и товаров продукции ООО «Первая типография» (Рисунок 3.12 – 3.13).

Первая типография
 Заказы
 Категории
 Товары
 Настройки
 Пользователи
 Выход

Список Категорий

Добавить новую Категорию

Искать по наименованию :

Наименование	Изображение	Действие
Мужские футболки		Изменить Удалить
Женские футболки		Изменить Удалить
Подарки		Изменить Удалить

1

(total 3)

Рисунок 3.12 – Раздел управления категориями продукции

Первая типография

Заказы

Категории

Товары

Настройки

Пользователи

Выход

Список Товаров

Добавить новый товар

Искать по наименованию :

Search









Наименование	Изображение	Статус	Количество	Цена	Категория	Действие
Кружка «8 марта»		Available	17	5 RUB	Подарки	Подробнее Изменить Удалить
Чашка с надписью 'Hug panda'		Available	30	5 RUB	Подарки	Подробнее Изменить Удалить
Чашка с надписью 'Drink tea'		Available	9	5 RUB	Подарки	Подробнее Изменить Удалить
Кот шредингера		Available	75	15 RUB	Мужские футболки	Подробнее Изменить Удалить
Siberian Fast Food		Available	45	28 RUB	Мужские футболки	Подробнее Изменить Удалить
Assassins Creed		Available	40	20 RUB	Мужские футболки	Подробнее Изменить Удалить
Подружка невесты, сердечки		Available	15	10 RUB	Женские футболки	Подробнее Изменить Удалить
Oxxxymiron		Available	10	15 RUB	Женские футболки	Подробнее Изменить Удалить

Рисунок 3.13 – Раздел управления товарами

Тем самым было разработано мобильное приложение на платформе Android для ООО «Первая типография», под управлением web-интерфейсом.

3.2 Расчет экономической эффективности внедрения мобильного приложения для online заказов продукции ООО «Первая типография»

Внедрение мобильного приложения в ООО «Первая типография» позволит уменьшить нагрузку на персонал и увеличит поток клиентов за счёт удобства и простоты использования, а также популярности приложений Android.

Ожидаемый экономический эффект от новых клиентов — рост прибыли организации на 5 %. Осуществим оценку предложенных усовершенствований и вычислим предположительный экономический эффект.

Ожидаемый результат от внедрения:

$$P = B \times 0,05 = 12156000 \times 0,05 = 607800 \text{ руб.},$$

где B – выручка за год (см. Приложение В).

Вследствие того, что организация заказывала разработку мобильного приложения в контексте производственной практики студента ТГУ, то по обоюдному согласию затраты на внедрение программного обеспечения для ООО «Первая типография» составили 0 руб.:

$Z = 0$ руб.

Вследствие чего, доход от внедрения:

$\Pi = P - Z = 607800 - 0 = 607800$ руб.

Высчитаем экономический эффект:

$\Xi = P / Z = 607800 / 1 = 607800$ руб. / руб. затрат

Ожидаемое увеличение роста прибыли от внедрения мобильного приложения предоставлен в виде диаграммы на рисунке 3.14.

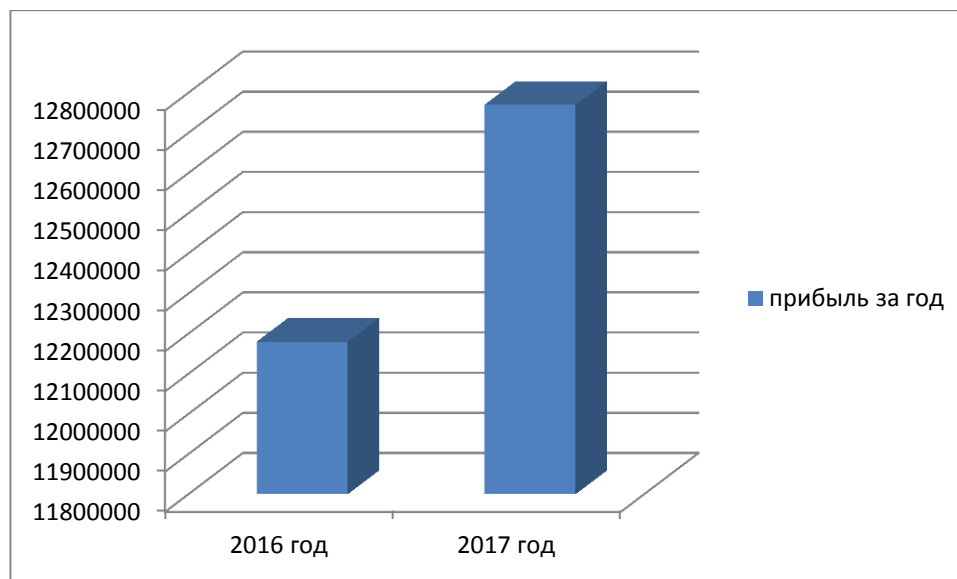


Рисунок 3.14 – Ожидаемое увеличение роста прибыли за 2017 год от внедрения сервиса

Экономическая эффективность от внедрения мобильного приложения в дальнейшем будет зависеть от ряда факторов конъюнктуры рынка.

Вывод по третьей главе

Была разработано мобильное приложение с управлением через web-интерфейс, в которой были реализованы механизмы предоставления информации товаров, а также возможность добавлять, удалять и редактировать продукцию и ее категории менеджером организации.

ЗАКЛЮЧЕНИЕ

Во время выполнения данной работы было создано мобильное приложение для осуществления online заказов продукции ООО «Первая типография» с управлением через web-интерфейс.

Исследование предметной области осуществлялось, опираясь на концептуальную модель по методологиям UML, IDEF0 и DFD с применением CASE-технологий, благодаря которым удалось сильно облегчить проектирование концептуальной модели системы.

Концептуальная модель предметной области способствовала выявлению основных элементов созданной системы. Были сформированы требования к проектируемой системе благодаря диаграмме вариантов использования, а также сделана начальная концептуальная модель системы для ее дальнейшей детализации в виде логической модели.

Основываясь на модели “ТО ВЕ”, были сформированы задачи на проектирование автоматизированной информационной системы оформления заказов от потребителей.

Моделирование системы способствовало полному удовлетворению всех требований, предъявленных системе в задании.

Для системы была создана информационная модель, а также сформирована логическая и физическая схема БД.

Благодаря технологиям: Java, HTML, PHP, CSS, MySQL реализована АИС для осуществления online заказов посредством мобильного приложения ООО «Первая типография» под управлением web-интерфейса.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ-34 РД 50-680-88. Руководящий документ по стандартизации. Методические указания. Автоматизированные системы. Основные положения.
2. ГОСТ-34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.
3. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.

Научная и методическая литература

4. Сеттер, Р.В. Изучаем Java на примерах и задачах – Наука и техника, 2016. – 240 с.
5. Дейтел П., Дейтел Х., Дейтел Э., Моргано М. Android для разработчиков, Питер, 2015. – 384 с.
6. Вязовик, Н. А. Программирование на Java, Интуит 2016. – 600 с.
7. Роберт Лафоре. Структура данных и алгоритмы Java. - Питер 2013. – 704с.
8. Фрэд Лонг. Руководство для программиста на Java: 75 рекомендаций по написанию надёжных и защищённых программ. – Вильямс , 2014. – 256 с.
9. Дэнни Гудман, Майкл Моррисон. JavaScript. Библия пользователя. - Вильямс, 2012. – 1184 с.
10. Джошуа Блох. Java. Эффективное программирование: Лори, 2014. – 440с.
11. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5: Питер, 2016. – 768 с.
12. Пьюривал Сэмми. Основы разработки веб-приложений: Питер, 2015. – 272 с.
13. Денис Колисниченко. PHP и MySQL. Разработка Web-приложений: БВХ-Петербург, 2015. – 593 с.

14. Балдин, К.В Информационные системы в экономике: Учебник / К.В Балдин, В.Б. Уткин. - М.: Дашков и К, 2015. – 395 с.
15. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы / Д.В. Александров. - М.: Финансы и статистика, 2011. – 224 с.
16. Васильков, А.В. Информационные системы и их безопасность: Учебное пособие / А.В. Васильков, А.А. Васильков, И.А. Васильков. - М.: Форум, 2013. – 528 с.
17. Мезенцев, К.Н. Автоматизированные информационные системы: Учебник для студентов учреждений среднего профессионального образования / К.Н. Мезенцев. - М.: ИЦ Академия, 2013. – 176 с.
18. Филиппов, В. А. Многомерные СУБД при создании корпоративных информационных систем / В.А. Филиппов. - М.: Едиториал УРСС, 2014. – 465 с.
19. Шелухин, О. И. Моделирование информационных систем / О.И. Шелухин, А.М. Тенякшев, А.В. Осин. - М.: Радиотехника, 2011. – 368 с.
20. Ярочкин, В. Безопасность информационных систем: моногр. / В. Ярочкин. - М.: Ось-89, 2015. – 320 с.
21. Гвоздева, В. А. Основы построения автоматизированных информационных систем / В.А. Гвоздева, И.Ю. Лаврентьева. - М.: Форум, Инфра-М, 2016. – 320 с.
22. Хоган Б., Уоррен К., Уэбер М., Джонсон К., Годин А. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов; Питер - Москва, 2013. – 288 с.
23. Тарасов, С.В. СУБД для программиста. Базы данных изнутри: Солон-Пресс , 2015. – 320 с.
24. Фуфаев Э.В., Фуфаев Д.Э. Базы данных: Академия , 2015. – 320 с.
25. Бураков М.В., Латыпова Р.Р. Базы данных и язык SQL: СПб.: ГУАП, 2014. – 120 стр.

26. Кузнецов С.Д. Введение в модель данных SQL (2-е изд.): М.: НОУ "Интуит" , 2015. – 350 с.

Литература на иностранном языке

27. Farrell J. Java Programming. — Course Technology, 2015.

28. Cay S. Horstmann Big Java Early Objects – Wiley, 2015.

29. Harvey M. Deitel, Paul J. Deitel Java How to Program, Prentice Hall. – 9th Edition, 2015.

30. Riaz Ahmed. SQL – The Shortest Route For Beginners. — CreateSpace Independent Publishing Platform, 2015.

31. Donald Bales. Beginning Oracle PL/SQL. – Apress, 2015

ПРИЛОЖЕНИЕ А

Основная активность

```
public class MainActivity extends FragmentActivity {
    private DrawerLayout mDrawerLayout;
    private ListView mDrawerList;
    private ActionBarDrawerToggle mDrawerToggle;

    private CharSequence mDrawerTitle;
    private CharSequence mTitle;

    private String[] navMenuTitles;
    private TypedArray navMenuIcons;

    private ArrayList<NavDrawerItem> navDrawerItems;
    private AdapterNavDrawerList adapter;
    static DBHelper dbHelper;
    AdapterMainMenu mma;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nav_drawer_main);
        Parse.initialize(this, getString(R.string.parse_application_id),
        getString(R.string.parse_client_key));
        ParseAnalytics.trackAppOpened(getIntent());
        PushService.setDefaultPushCallback(this, MainActivity.class);
        ParseInstallation.getCurrentInstallation().saveInBackground();
        mTitle = mDrawerTitle = getTitle();
        navMenuTitles = getResources().getStringArray(R.array.nav_drawer_items);
        navMenuIcons = getResources().obtainTypedArray(R.array.nav_drawer_icons);
        mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
        mDrawerList = (ListView) findViewById(R.id.list_slidermenu);
        mDrawerLayout.setDrawerShadow(R.drawable.navigation_drawer_shadow,
        GravityCompat.START);
        navDrawerItems = new ArrayList<NavDrawerItem>();
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[0],
        navMenuIcons.getResourceId(0, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[1],
        navMenuIcons.getResourceId(1, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[2],
        navMenuIcons.getResourceId(2, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[4],
        navMenuIcons.getResourceId(4, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[5],
        navMenuIcons.getResourceId(5, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[6],
        navMenuIcons.getResourceId(6, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[7],
        navMenuIcons.getResourceId(7, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[8],
        navMenuIcons.getResourceId(8, -1)));
        navDrawerItems.add(new NavDrawerItem(navMenuTitles[9],
        navMenuIcons.getResourceId(9, -1)));
        navMenuIcons.recycle();
        mDrawerList.setOnItemClickListener(new SlideMenuClickListener());
        adapter = new AdapterNavDrawerList(getApplicationContext(),
        navDrawerItems);
        mDrawerList.setAdapter(adapter);
        getActionBar().setDisplayHomeAsUpEnabled(true);
        getActionBar().setHomeButtonEnabled(true);
        ActionBar bar = getActionBar();
        bar.setBackgroundDrawable(new
```

```

ColorDrawable(getResources().getColor(R.color.header));
DisplayMetrics dm = new DisplayMetrics();
getWindowManager().getDefaultDisplay().getMetrics(dm);
if (!Constant.isNetworkAvailable(MainActivity.this)) {
    Toast.makeText(MainActivity.this, getString(R.string.no_internet),
Toast.LENGTH_SHORT).show();
}
mma = new AdapterMainMenu(this);
dbhelper = new DBHelper(this);
try {
    dbhelper.createDataBase();
} catch (IOException ioe) {
    throw new Error("Unable to create database");
}
try {
    dbhelper.openDataBase();
} catch (SQLException sqle) {
    throw sqle;
}
if (dbhelper.isPreviousDataExist()) {
    showAlertDialog();
}
mDrawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout,
R.drawable.ic_drawer,
    R.string.app_name,
    R.string.app_name
) {
    public void onDrawerClosed(View view) {
        getActionBar().setTitle(mTitle);
        invalidateOptionsMenu();
    }

    public void onDrawerOpened(View drawerView) {
        getActionBar().setTitle(mDrawerTitle);
        invalidateOptionsMenu();
    }
};
mDrawerLayout.setDrawerListener(mDrawerToggle);

if (savedInstanceState == null) {
    displayView(0);
}
}
void showAlertDialog() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle(R.string.confirm);
    builder.setMessage(getString(R.string.db_exist_alert));
    builder.setCancelable(false);
    builder.setPositiveButton(getString(R.string.yes), new
DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {
            dbhelper.deleteAllData();
            dbhelper.close();
        }
    });
    builder.setNegativeButton(getString(R.string.no), new
DialogInterface.OnClickListener() {

        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            dbhelper.close();
            dialog.cancel();
        }
    });
}

```



```

    }
    });
    AlertDialog alert = builder.create();
    alert.show();
}
@Override
public void onBackPressed() {
    // TODO Auto-generated method stub
    dbHelper.deleteAllData();
    dbHelper.close();
    finish();
    overridePendingTransition(R.anim.open_main, R.anim.close_next);
}
private class SlideMenuClickListener implements ListView.OnItemClickListener
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        displayView(position);
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }
    switch (item.getItemId()) {
        case R.id.rate_app:
            try {
                startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("market://details?id=" + getPackageName())));
            } catch (android.content.ActivityNotFoundException anfe) {
                startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse("http://play.google.com/store/apps/details?id=" +
getPackageName())));
            }
            return true;
        case R.id.more_app:
            startActivity(new Intent(Intent.ACTION_VIEW,
Uri.parse(getString(R.string.more_apps))));
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    // if nav drawer is opened, hide the action items
    boolean drawerOpen = mDrawerLayout.isDrawerOpen(mDrawerList);
    menu.findItem(R.id.ic_menu).setVisible(!drawerOpen);
    return super.onPrepareOptionsMenu(menu);
}
private void displayView(int position) {
    Fragment fragment = null;
    switch (position) {
        case 0:
            fragment = new ActivityHome();
            break;
        case 1:

```

```

        startActivity(new Intent(getApplicationContext(),
ActivityCategoryList.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    case 2:
        startActivity(new Intent(getApplicationContext(), ActivityCart.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    case 3:
        startActivity(new Intent(getApplicationContext(),
ActivityProfile.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    case 4:
        startActivity(new Intent(getApplicationContext(),
ActivityInformation.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    case 5:
        startActivity(new Intent(getApplicationContext(),
ActivityAbout.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    case 6:
        Intent sendInt = new Intent(Intent.ACTION_SEND);
        sendInt.putExtra(Intent.EXTRA_SUBJECT, getString(R.string.app_name));
        sendInt.putExtra(Intent.EXTRA_TEXT, "First Tipograf Android App\n\n" +
getString(R.string.app_name)
        + "\n\nhttps://play.google.com/store/apps/details?id=" +
getPackageName());
        sendInt.setType("text/plain");
        startActivity(Intent.createChooser(sendInt, "Share"));
        break;
    case 7:
        startActivity(new Intent(getApplicationContext(),
ActivityContactUs.class));
        overridePendingTransition(R.anim.open_next, R.anim.close_next);

        break;
    case 8:
        dbHelper.deleteAllData();
        dbHelper.close();
        MainActivity.this.finish();
        overridePendingTransition(R.anim.open_next, R.anim.close_next);
        break;
    default:
        break;
}
if (fragment != null) {
    android.app.FragmentManager fragmentManager = getFragmentManager();
    fragmentManager.beginTransaction().replace(R.id.frame_container,
fragment).commit();
    mDrawerList.setItemChecked(position, true);
    mDrawerList.setSelection(position);
    setTitle(navMenuTitles[position]);
    mDrawerLayout.closeDrawer(mDrawerList);
} else {
    Log.e("MainActivity", "Error in creating fragment");
}
}

@Override
public void setTitle(CharSequence title) {
    mTitle = title;
}

```

```
        getActionBar().setTitle(mTitle);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mDrawerToggle.syncState();
    }
    @Override
    public void onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
        mDrawerToggle.onConfigurationChanged(newConfig);
    }
}
```

ПРИЛОЖЕНИЕ Б

Требования, предъявляемые к будущей системе.

ID	Требование	Статус	Полезность	Риск	Стабильность
Функциональные требования					
1	Предоставлять информацию о товаре в мобильном приложении	Одобренные	Критичное	Средний	Низкая
2	Регистрировать клиентов	Одобренные	Критичное	Средний	Низкая
3	Принимать заявки на приобретение товаров	Одобренные	Критичное	Средний	Низкая
4	Отправлять результаты о сделанном заказе	Предложенные	Критичное	Средний	Средняя
5	Предоставлять возможность ознакомиться с правилами пользования мобильным приложением	Предложенные	Критичное	Средний	Средняя
Требования к удобству использования					
8	Названия товаров и их описание понятны и просты для прочтения	Одобренные	Критичное	Низкий	Низкая
9	Удобный поиск товара	Одобренные	Критичное	Низкий	Низкая

ID	Требование	Статус	Полезность	Риск	Стабильность
10	Товары должны быть разбиты на категории	Одобренные	Критичное	Низкий	Низкая
11	Поддержка версии Android 6.0	Одобренные	Критичное	Низкий	Низкая
12	Размер основного шрифта должен быть не меньше 12 px	Одобренные	Критичное	Низкий	Низкая
13	Дизайн мобильного приложения должен быть выполнен в единой цветовой гамме и едином стиле.	Одобренные	Критичное	Низкий	Низкая
Требования к надежности					
14	Среднее время восстановления работоспособного состояния не более 1-2 суток	Предложенные	Критичное	Низкий	Низкая
15	Организация бесперебойного питания технических средств	Предложенные	Критичное	Средний	Средняя

ID	Требование	Статус	Полезность	Риск	Стабильность
Требования к производительности					
16	Загрузка списка товаров не должна быть более 60 секунд	Предложенные	Критичное	Средний	Средняя
Требования к поддержке					
16	Время устранения возникших проблем с серверной частью 30-60 мин	Предложенные	Критичное	Средний	Средняя
Ограничения реализации					
17	Реализация на языках java, php, javascript, html и css	Одобренные	Критичное	Средний	Низкая

ПРИЛОЖЕНИЕ В

Главные экономические показатели деятельности

ООО «Первая типография» за 2014 – 2016 гг.

Показатели	2014 г.	2015 г.	2016 г.	Изменение					
				2015 -2014 гг.		2016 -2015 гг.		2016 -2014 гг.	
				Абс. изм (+/-)	Темп прироста, %	Абс. изм (+/-)	Темп прироста, %	Абс. изм (+/-)	Темп прироста, %
1	2	3	4	5	6	7	8	9	10
1. Выручка ¹ , тыс. руб.	13004	13842	12156	838	6	-1686	-12	-848	-6,5
2. Себестоимость продаж ¹ , тыс. руб.	10643	10992	10493	349	3	-499	-4,5	-150	-1,4
3. Валовая прибыль ¹ (убыток), тыс. руб.	2361	2850	1662	489	21	-1188	-41,7	-699	-29,6
4. Управленческие расходы ¹ , тыс. руб.	1365	1367	1363	2	0,2	-4	-0,3	-2	-0,2
5. Коммерческие расходы ¹ , тыс. руб.	51	56	48	5	10	-8	-14	-3	-6
6. Прибыль (убыток) от продаж, тыс. руб.	996	1483	299	487	49	-1184	-80	-697	-70
7. Чистая прибыль ¹ , тыс. руб.	634	1050	207	416	66	-843	-80	-427	-67
8. Основные средства, тыс. руб.	11167	10592	9723	-575	-0,9	-869	-8,2	-1444	-12,9
9. Оборотные активы ² , тыс. руб.	3818	4704	4065	886	23	-639	-13,6	247	6,5
10. Численность ППП, чел.	25	25	25	0	0	0	0	0	0
11. Фонд оплаты труда ППП ³ , тыс. руб.	4112	4357	4572	245	6	215	5	460	11
12. Производительность труда работающего, тыс. руб. (стр1/стр10)	520,2	553,7	486,2	33,5	6,4	-67,5	-12	-34	-6,5
13. Среднегодовая заработная плата работающего, тыс. руб. (стр11/стр10)	164,5	174,3	182,9	9,8	6	8,6	5	18,4	11
14. Фондоотдача (стр1/стр8)	1,2	1,3	1,3	0,1	8	0	0	0,1	8
15. Оборачиваемость активов, раз (стр1/стр9)	3,4	2,9	3	-0,5	-15	0,1	3	-0,4	-12
16. Рентабельность продаж, % (стр6/стр1) ×100%	7,7	10,7	2,5	3	39	-8,2	-77	-5,2	-68
17. Рентабельность производства, % (стр7/стр1) ×100%	4,9	7,6	1,7	2,7	55	-5,9	-78	-3,2	-65
18. Затраты на рубль выручки, (стр2+стр4+стр5)/стр1 *100 коп.)	93	90	98	-3	-3	8	9	5	5

1 – форма 2. Отчет о финансовых результатах (Отчет о прибылях и убытках)

2 –Бухгалтерский баланс. Итоговая сумма всех оборотных активов, указанных в бланке бухгалтерского баланса

3 – Отчет о движении денежных средств.