

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра информационных систем управления

Максимов Валерий Александрович

СОЗДАНИЕ СИСТЕМЫ СБОРА, ОБРАБОТКИ И АНАЛИЗА
БОЛЬШИХ ДАННЫХ РЫНКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по образовательной программе подготовки бакалавров по
направлению 09.03.03 – «Прикладная информатика»
профиль – «Прикладная информатика в экономике»

г. Владивосток
2019

Автор
работы

(подпись)

«__» _____ 2019г.

Руководите
ль, доцент кафедры
 ИСУ

Д.А.
Оськин

(подпись)

«__» _____ 2019г.

Защищена в ГАК с
оценкой

Секретарь ГАК

(подпись)

И.О. Фамилия

«__» _____ 2019г.

«Допустить к защите»
Заведующий кафедрой, к.т.н.,
доцент

А.И.
Сухомлинов

(подпись)

«__» _____ 2019г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение

высшего образования

«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра информационных систем управления

ЗАДАНИЕ

на выпускную квалификационную работу

студенту Максимову Валерию Александровичу группы Б8419
(фамилия, имя, отчество)

на тему «Создание системы сбора, обработки и анализа Больших
Данных рынка
мобильных приложений»

Вопросы, подлежащие разработке (исследованию):

1. Описание предметной области (описание деятельности компаний, занимающихся разработкой мобильных приложений; исследование процесса анализа рынка мобильных приложений; выявление проблем процесса анализа)
 2. Разработка проекта создания системы сбора, обработки и анализа данных
 3. Описание технологии решения задачи (описание постановки задачи; проектирование системы; формирование требований к системе; проектирование информационного обеспечения системы; поиск и обзор систем-аналогов; сравнительный анализ программных продуктов)
 4. Разработка системы сбора, обработки и анализа Больших Данных рынка мобильных приложений
 5. Расчет экономической эффективности
-

Основные источники информации и прочее, используемые для
разработки темы

1. Материалы по производственной практике по получению профессиональных умений и опыта профессиональной деятельности
 2. Материалы по производственной практике (научно-
-

исследовательской работе)

3. Материалы по преддипломной практике
4. Справочная, научная, учебная литература
5. Ресурсы интернет

Срок представления
работы «__» _____ 2019г.

Дата выдачи задания «__» _____ 2019г.

Руководитель ВКР доцент
кафедры информационных систем
управления

(подпись) Д.А. Оськин
(И.О. Фамилия)

Задание получил

(подпись) В.А.Максимов
(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение

высшего образования

«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

Кафедра информационных систем управления

Г Р А Ф И К

подготовки и оформления выпускной квалификационной работы

студента Максимова Валерия Александровича группы Б8419
(фамилия, имя, отчество)

на тему «Создание системы сбора, обработки и анализа Больших Данных
рынка

мобильных приложений»

№ п/ п	Выполняемые работы и мероприятия	Срок выполнения	Отметка о выполнен ии
1.	Выбор темы и согласование с руководителем	до 25 сентября	
2.	Подбор первичного материала, его изучение и обработка. Составление предварительной библиографии	до 20 ноября	

3.	Составление плана работы и согласования с руководителем	до 01 декабря	
4.	Разработка и представление руководителю первой части работы	до 01 февраля	
5.	Разработка и представление руководителю второй части работы	до 01 июня	
6.	Подготовка и согласование с руководителем выводов и предложений, введения и заключения. Подготовка презентации работы	до 22 июня	
7.	Доработка ВКР в соответствии с замечаниями руководителя	до 28 июня	
8.	Получение отзыва научного руководителя и защита ВКР на заседании выпускающей кафедры	до 01 июля	
9.	Доработка ВКР в соответствии с замечаниями, высказанными на защите, окончательное оформление	до 06 июля	
10.	Завершение подготовки к защите (доклад, раздаточный материал, презентация в Power Point)	до 06 июля	
11.	Защита ВКР в ГАК	июль	

Студент

«__» _____ 2019г.

(подпись)

В.А. Максимов

(И.О. Фамилия)

Руководитель ВКР

«__» _____ 2019г.

(подпись)

Д.А. Оськин

(И.О. Фамилия)

Аннотация

Тема выпускной квалификационной работы: «Создание системы обработки и анализа Больших Данных рынка мобильных приложений».

Объём работы 67 страниц, на которых размещены 39 рисунков и 10 таблиц. При написании диплома использовалось 25 литературных источников.

Ключевые слова: «Большие Данные», визуализация, мобильные приложения, анализ данных, обработка данных, Data Mining, Python.

Цель – разработка системы для проведения анализа рынка приложений для мобильных устройств.

Объектом исследования являются компании, занимающиеся разработкой мобильных приложений.

Предметом исследования является процесс анализа рынка мобильных приложений, выполняемый аналитиком.

Практическая значимость исследования заключается в разработке WEB-приложения автоматизирующего деятельность аналитика.

Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка используемых источников, приложений.

Введение раскрывает актуальность выбранной темы, цель исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе рассматривается существующее состояние предметной области («КАК ЕСТЬ»). Выявляются проблемы и недостатки в работе аналитика рынка мобильных

приложений и дается обоснование предложений по устранению выявленных недостатков.

Во второй главе описываются решения, принятые по всей вертикали проектирования информационной системы и дан контрольный пример.

В заключении подводятся итоги исследования, формируются окончательные выводы по рассматриваемой теме, оценка степени выполнения поставленных задач.

Оглавление

Введение.....	4
1 Анализ процесса анализа рынка мобильных приложений.....	7
1.1 Описание процесса анализа рынка мобильных приложений.....	7
1.2 Экономическая сущность комплекса экономических информационных задач аналитика.....	10
1.3 Обоснование необходимости и цели использования автоматизированной системы анализа.....	13
1.3.1 Обоснование автоматизации решения задачи.....	13
1.3.2 Цели и задачи автоматизации.....	17
1.3.3 Анализ существующих разработок и сравнительная характеристика программного обеспечения.....	17
1.4 Общая характеристика организации решения задачи автоматизации процессов сбора и обработки информации по мобильным приложениям.....	21
1.4.1 Постановка задачи.....	21
1.4.2 Входная информация.....	23
1.4.3 Выходная информация.....	24
2 Проектирование системы анализа и разработка программного обеспечения.....	30
2.1 Разработка проекта системы автоматизации работы аналитика..	30
2.1.1 Описание содержания проекта.....	30
2.1.2 Оценка длительности операций.....	33
2.2 Анализ требований к разрабатываемой системе.....	39
2.3 Проектирование системы анализа.....	44
2.4 Архитектура системы и ее программное обеспечение.....	46
2.5 Информационное обеспечение задачи анализа.....	49
2.6 Технологическое обеспечение задачи сбора и обработки информации о мобильных приложениях.....	53
2.7 Обоснование экономической эффективности проекта.....	55
2.8 Описание контрольного примера реализации проекта.....	61
Заключение.....	65
Список литературы.....	66
Приложение А.....	68
Приложение Б.....	71
Модуль считывания данных с Google Play Store.....	71
Фрагмент модуля обработки данных.....	76
Фрагмент модуля формирования графиков.....	78
Шаблон HTML-страницы коллекции.....	81
Приложение В.....	83

Введение

В настоящее время в рамках развития сферы информационных технологий активно развиваются мобильные технологии, что сопровождается развитием рынка мобильных приложений.

Если учитывать лишь приложения из самых популярных магазинов Android и iOS, рынок мобильных приложений уже насчитывает около 6 миллионов приложений, однако даже с такой статистикой появление как небольших групп разработчиков, так и целых компаний, занимающихся разработкой приложений, не прекращается.

Любая подобная новая компания сталкивается с вопросом о том, с чего какое приложение позволит компании удачно выйти на рынок и получить наибольшую прибыль. Для ответа на этот вопрос нужно определить целевую аудиторию популярных приложений, выявить наиболее актуальные категорий приложений, определить основную ценовую категорию наиболее популярных приложений и так далее.

Решением этих задач обычно занимается аналитик рынка мобильных приложений. Он занимается отслеживанием динамики появления новых пользователей у популярных приложений, изменение пользовательских оценок, собирает множество различных дополнительных данных о ценах, возрастных категориях и многом другом.

Объем всей этой информации огромен, а отслеживание динамики оценок, скачиваний и других показателей требует серьёзных трудозатрат, так как в свободном доступе подобных сведений нет, и получить их можно лишь путём ручного мониторинга каждого мобильного приложения

непосредственно в магазине. К тому же аналитику потребуется использовать дополнительные программы для агрегирования всех полученных данных и получения общей картины рынка. При ручном проведении анализа на такую трудоемкую работу будет затрачено множество времени, а, следовательно, и денежных средств.

В настоящий момент на рынке существует не так много готовых решений позволяющих проанализировать рынок мобильных приложений. При этом все доступные решения предоставляют лишь данные относительно одного конкретного приложения: количество пользователей этого приложения (число скачиваний), средняя пользовательская оценка приложения, рассчитанная на основе набора всех выставленных пользователями оценок и так далее.

То есть все существующие продукты не позволяют производить анализ рынка мобильных приложений, а позволяют отслеживать лишь определенные приложения, что ограничивает аналитика при принятии решений на начальных этапах разработки, когда принимается решение о назначении и функционале будущего приложения.

Поэтому становится актуальной задача создания системы, которая позволит автоматизировать сбор данных из магазина приложений, их обработку и представление в такой форме, которая будет удовлетворять всем потребностям аналитика при принятии решения о том, какое именно приложение требуется разработать.

Целью выпускной квалификационной работы является разработка системы для проведения анализа рынка приложений для мобильных устройств по выбранным

критериям, способной собирать, хранить и обрабатывать данные рынка мобильных приложений. Данная система избавит аналитика от выполнения трудоемкой, длительной и дорогой работы по сбору необходимых ему данных и будет предоставлять данные рынка мобильных приложений аналитику в форме, удобной для проведения анализа.

Для создания системы необходимо решить следующие задачи:

- проанализировать предметную область, охватываемую системой;
- определить требования к системе, путем анализа существующих решений в данной области, выделения минусов и плюсов аналогов, и выработка на их основе требований к системе;
- выполнить архитектурное проектирование будущей системы, и построить модели выполнения технологических процессов, которые она должна выполнять;
- реализовать систему, то есть выполнить работы по написанию кода, разработке базы данных системы, верстке веб-страниц, созданию серверной части системы, подключению базы данных к системе, тестированию и доработке системы.

Объектом исследования является отдел маркетинга в компании, занимающейся разработкой мобильных приложений.

Предметом исследования является процесс анализа рынка мобильных приложений, выполняемый аналитиком.

1 Анализ процесса анализа рынка мобильных приложений

1.1 Описание процесса анализа рынка мобильных приложений

Большие данные (англ. Big Data) – обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами. Однако в частности термин Big Data описывает не только сами данные, но и технологии их обработки и использования, а также методы поиска необходимой информации в больших массивах данных [3].

Google Play (прежнее название – Android Market) – магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения [10].

Знание статистики и тенденций присутствующих в Google Play Store данных является ценным инструментом как для разработчиков приложений, так и для разработчиков продуктов Android.

Каждое приложение включает в себя ряд мета-данных:

- количество скачиваний приложения;
- пользовательский рейтинг;
- стоимость приложения;
- наличие встроенных покупок;
- верхняя и нижняя граница стоимости встроенных покупок (при их наличии);
- возрастная категория приложения;

- жанр приложения.

Для компаний, занимающихся разработкой мобильных приложений агрегированная информация по приложениям, находящимся в топах, позволит решить множество вопросов, связанных с разработкой, которые возникают на её начальном этапе.

На рисунке 1 представлена предполагаемая организационная структура предприятия, занимающегося разработкой мобильных приложений под устройства с ОС Android. Данная организационная структура была составлена в результате сбора и агрегирования информации о реальных организационных структурах следующих компаний-разработчиков мобильных приложений: RedMadRobot, Agima, Omega-R и Touch Instinct.

Оргструктура предприятия позволяет выделить основные структуры компании и определить основные бизнес-процессы, выполняемые на предприятии.

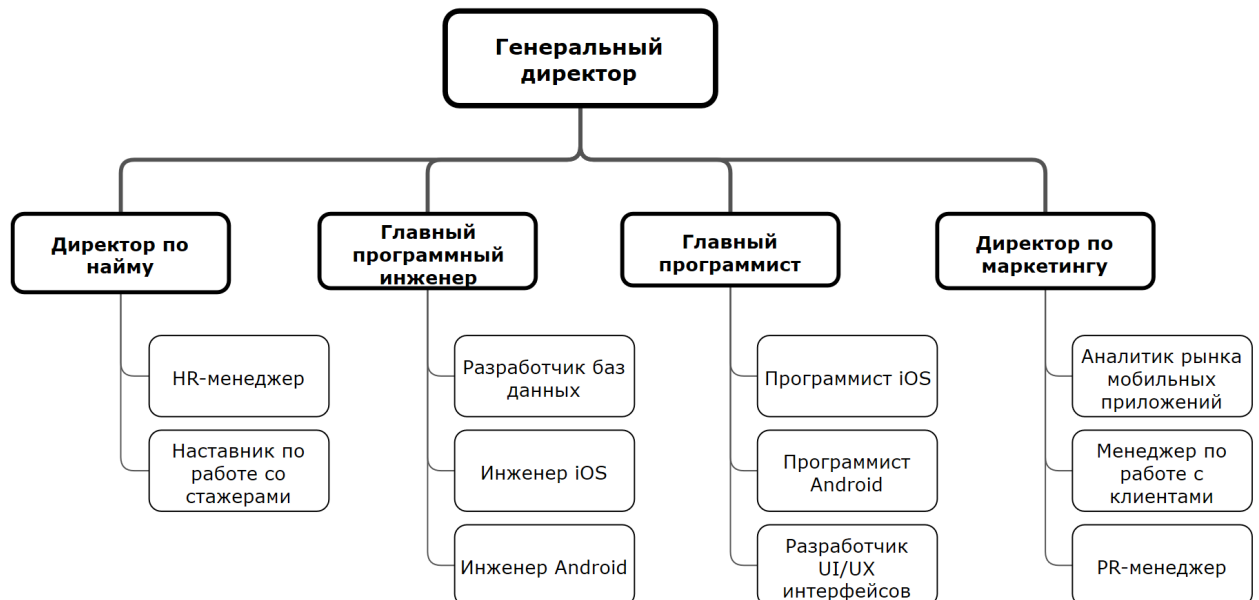


Рисунок 1 – Типовая организационная структура предприятия по разработке мобильных приложений

Основной деятельностью компании по разработке мобильных приложений является непосредственно сам процесс разработки мобильных приложений, однако помимо него компания осуществляет и другие процессы, поддерживающие её работу.

Директор по найму – руководитель отдела по работе с кадрами, в обязанности которого входит управление отделом найма сотрудников. В состав отдела, помимо директора, входят также HR менеджер – лицо, занимающееся непосредственно подбором персонала, требующегося компании, и наставник по работе со стажерами – лицо, занимающееся обучением стажеров и выполняющее наблюдение за выполнением их обязанностей.

Главный программный инженер – руководитель инженерного отдела, осуществляющий общий контроль над процессами проектирования iOS и Android приложений, а также баз данных к этим приложениям. В состав отдела, помимо главного инженера, также входят разработчик баз данных, занимающийся проектированием и разработкой баз данных к разрабатываемым приложениям, и инженеры iOS и Android, занимающиеся функциональным и структурным моделированием приложений под соответствующие платформы.

Главный программист – руководитель отдела программирования, выполняющий контроль над процессами разработки и тестирования разрабатываемых приложений. В состав отдела, кроме главного программиста, также входят программисты iOS и Android, занимающиеся непосредственно написанием приложений под

соответствующие платформы по составленным инженерным отделом моделям, и разработчик UI/UX интерфейсов, занимающийся разработкой дизайна пользовательских интерфейсов приложений и их программированием.

Директор по маркетингу – руководитель отдела маркетинга, принимающий ключевые стратегические решения по развитию предприятия и продвижению приложений. В состав отдела маркетинга также входят PR-менеджер, занимающийся вопросами рекламы приложений, а также разрабатывающий стратегии продвижения приложений на рынке, менеджер по работе с клиентами, выполняющий процессы сбора и анализа отзывов от клиентов по выпущенным приложениям для разработки стратегий продвижения этих приложений, и аналитик рынка мобильных приложений, занимающийся сбором и анализом отзывов, пользовательских оценок, количеств скачиваний и других метаданных приложений конкурентов, пользующихся популярностью у пользователей, с целью развития стратегий и формирования идей по созданию приложений, способных конкурировать с приложениями конкурентов.

1.2 Экономическая сущность комплекса экономических информационных задач аналитика

Одним из ключевых актеров в маркетинговой деятельности является аналитик. Он выполняет задачи по анализу рынка мобильных приложений, в которые входят: сбор необходимых для анализа данных, обработка собранных данных (проверка на ошибки и аномальные значения, а также приведение данных к виду, необходимому для

дальнейшей работы с ними), построение различных схем и диаграмм, необходимых для проведения анализа, анализ и принятие решений по результатам произведенного анализа.

Основанием для большинства решений, принятых в ходе маркетинговых исследований, служит информация, полученная именно аналитиком. Аналитик формирует отчеты, отображающие текущее состояние рынка (объем рынка, распределение приложений в топах по категориям), показатели, влияющие на спрос (категории, пользовательские оценки, наличие встроенных покупок, цена), и их динамика, особенности ценообразования (зависимость цен приложений от пользовательских оценок), по результатам проведенных исследований рынка и предоставляет их директору по маркетингу.

Для оценки временных затрат на сбор данных и обработку данных существует 3 основные методики:

- PERT (Program (Project) Evaluation and Review Technique, «оценка по трем точкам») – методика определения

ожидаемой продолжительности работы по формуле $\frac{t_n + 4t_{нв} + t_o}{6}$,

где t_n – пессимистичная оценка, $t_{нв}$ – наиболее вероятная оценка, t_o – оптимистичная оценка;

- экспертная оценка;

- опытная оценка (экспериментальная оценка).

Так как методика PERT подразумевает наличие представлений о времени, которое будет затрачено на выполнение задачи в лучшем случае (оптимистичная), в худшем случае (пессимистичная) и при обычных обстоятельствах (наиболее вероятная), то эта методика в

данном случае неприменима, так как никаких данных о длительности выполнения задач сбора и обработки данных нет.

Для применения экспертной оценки требуется наличие экспертов в изучаемом вопросе, что тоже в данном случае невозможно.

В связи с этим была выбрана методика экспериментальных исследований, позволившая изучить процессы сбора и обработки данных «изнутри», а также оценить реальные временные затраты на выполнение этих процессов. Для этого был создан файл Excel с созданными заранее таблицами, в которые, впоследствии, вносились данные из каждой коллекции (топа). Для изучения было решено собрать и обработать по 5 приложений из каждой коллекции, после чего полученные результаты было необходимо умножить на количество приложений в коллекциях.

Таким образом, на сбор данных по 5 приложениям в каждой коллекции было затрачено 20 минут, то есть по 5 минут на каждые 5 приложений. А значит в среднем, для сбора информации за одни сутки по всем приложениям, находящимся в топах (2000 приложений), аналитик затратит более 31-33 часов при наличии стабильного интернет-соединения, и при условии того, что аналитик является опытным ПК-пользователем.

На обработку собранной информации опытный аналитик затратит около одного часа, на визуализацию – также около часа.

Таким образом, на проведение анализа данных по коллекциям, сформированным на сутки, опытный аналитик затратит около 33-35 часов, что превышает длительность суток и тем более длительность рабочего дня. То есть для того, чтобы успеть выполнить сбор данных, аналитику придётся пожертвовать 1600 строками данных (придётся собирать данные лишь по 100 приложениям в каждой коллекции, вместо 500 возможных), что ухудшит общий показатель анализа. Также, если аналитик получает оплату за каждый час работы, то использование системы, ускоряющей процессы сбора и обработки информации, позволит компании сэкономить значительное количество средств.

Информация о необходимых аналитику данных была собрана из статей аналитиков изучаемой сферы деятельности. В частности, статья «Analysis of Apps in the Google Play Store» автора Yulia Norenko [9] позволила выделить основные показатели, необходимые аналитику при проведении анализа, а статья «A Statistical Analysis of the Apple App Store» автора Colin Eberhardt [8] разъяснила основной принцип мышления при проведении анализа агрегированных данных рынка мобильных приложений.

Так как Google не предусмотрели функционала в своих API, позволяющего программными средствами выполнять сбор необходимой информации, возникает ключевая проблема – невозможность быстрого получения мета-данных по приложениям.

Сбор этой информации в разрабатываемой системе было решено реализовать посредством WEB-Scraping'a. В

широком понимании веб-скрапинг – это сбор данных с различных интернет-ресурсов, общий принцип работы которого можно объяснить следующим образом: некий код, автоматизирующий выполнение запросов на целевой сайт, и получая ответ, исследует HTML-документ, ищет в нем необходимые данные и преобразует их в заданный формат [16][3][4].

1.3 Обоснование необходимости и цели использования автоматизированной системы анализа

1.3.1 Обоснование автоматизации решения задачи

Чтобы обосновать автоматизацию решения задачи необходимо дать характеристику, существующей технологии решения задачи. Для этого был проведен структурно-функциональный анализ решаемой задачи, в ходе которого были разработаны диаграммы модели IDEF0 «КАК-ЕСТЬ» (AS-IS), приведенные ниже на рисунках 2-5.

На рисунке 2 представлена контекстная AS-IS диаграмма процесса анализа рынка мобильных приложений.

На рисунке 3 приведена декомпозиция контекстной AS-IS диаграммы процесса анализа рынка мобильных приложений.

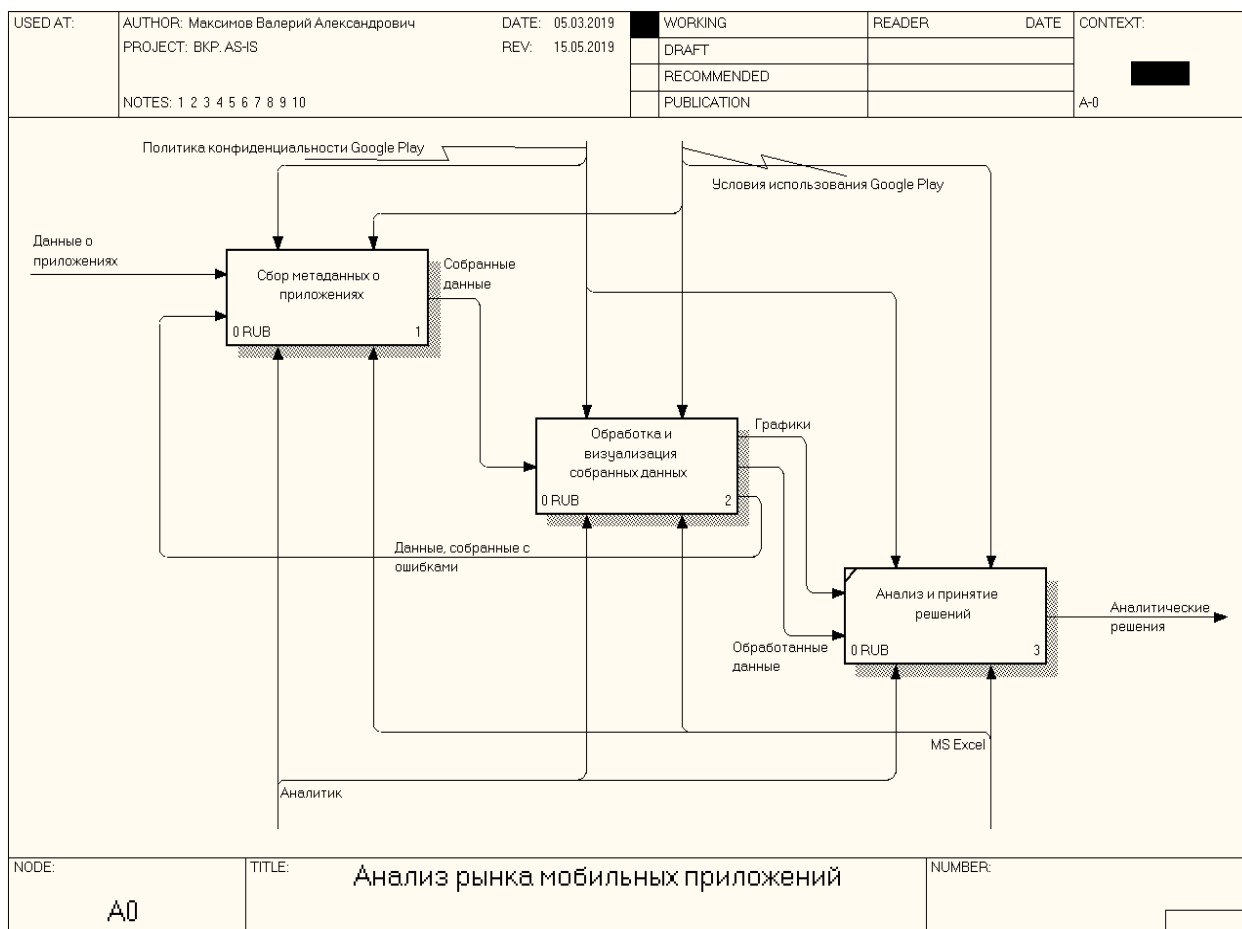


Рисунок 3 – Декомпозиция контекстной диаграммы

На рисунке 4 представлена декомпозиция процесса «Сбор мета-данных о приложениях», построенная по методологии IDEF3, позволяющая отобразить цикличность процесса сбора данных.

Как видно на диаграмме, основным исполнителем (объектом-инициатором) процесса сбора является аналитик, который вручную выполняет сбор данных.

При сборе данных аналитик использует электронные таблицы Microsoft Excel, позволяющие ему сохранить собранные данные, обработать их и выполнить необходимый ему анализ.

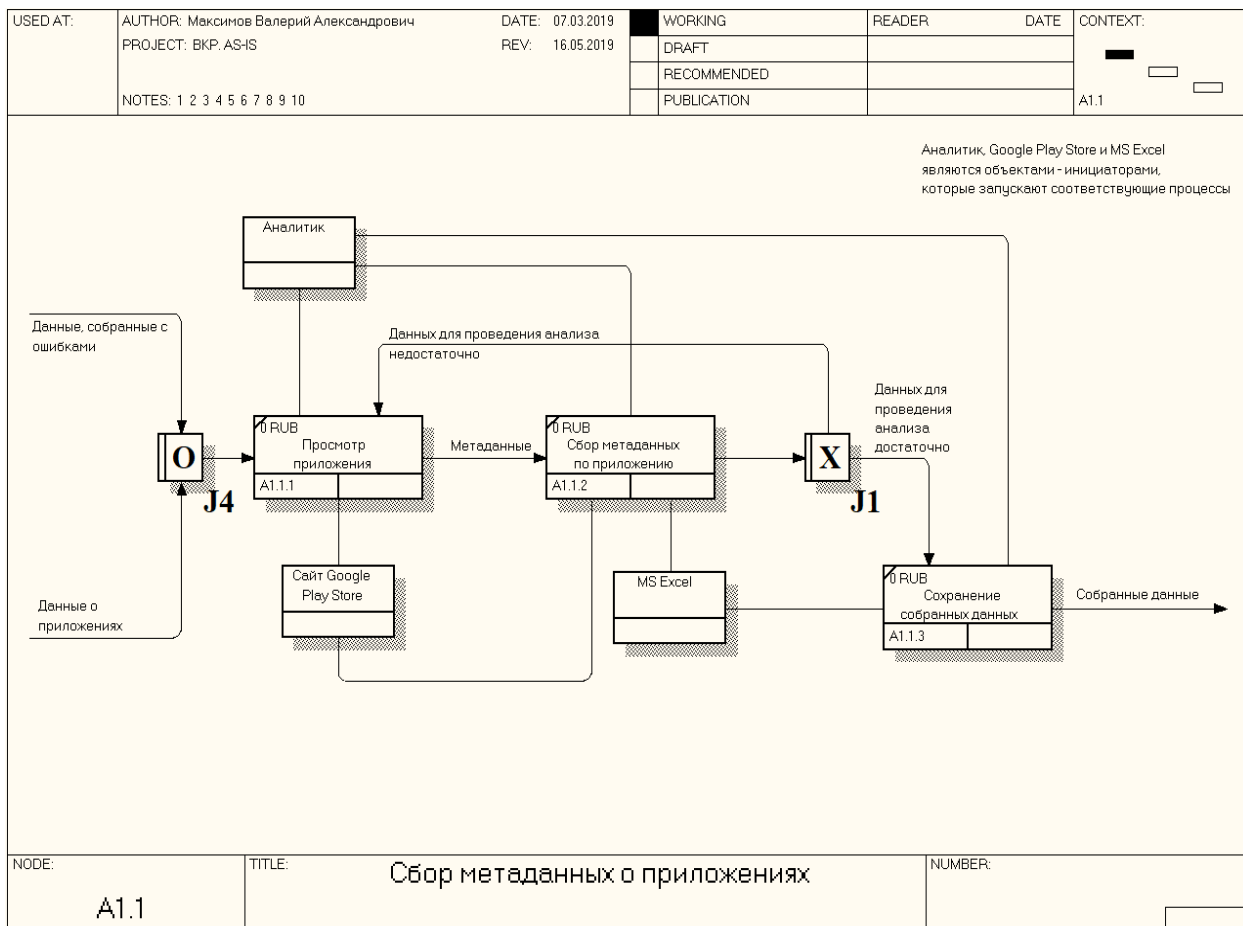


Рисунок 4 – Декомпозиция процесса «Сбор метаданных о приложениях»

На рисунке 5 представлена декомпозиция процесса «обработка и визуализация собранных данных», построенная по методологии IDEF3, позволяющая отобразить цикличность процесса обработки данных.

При этом есть вероятность обнаружить ошибку во время обработки данных, что вынудит аналитика повторить сбор данных по приложению с ошибкой.

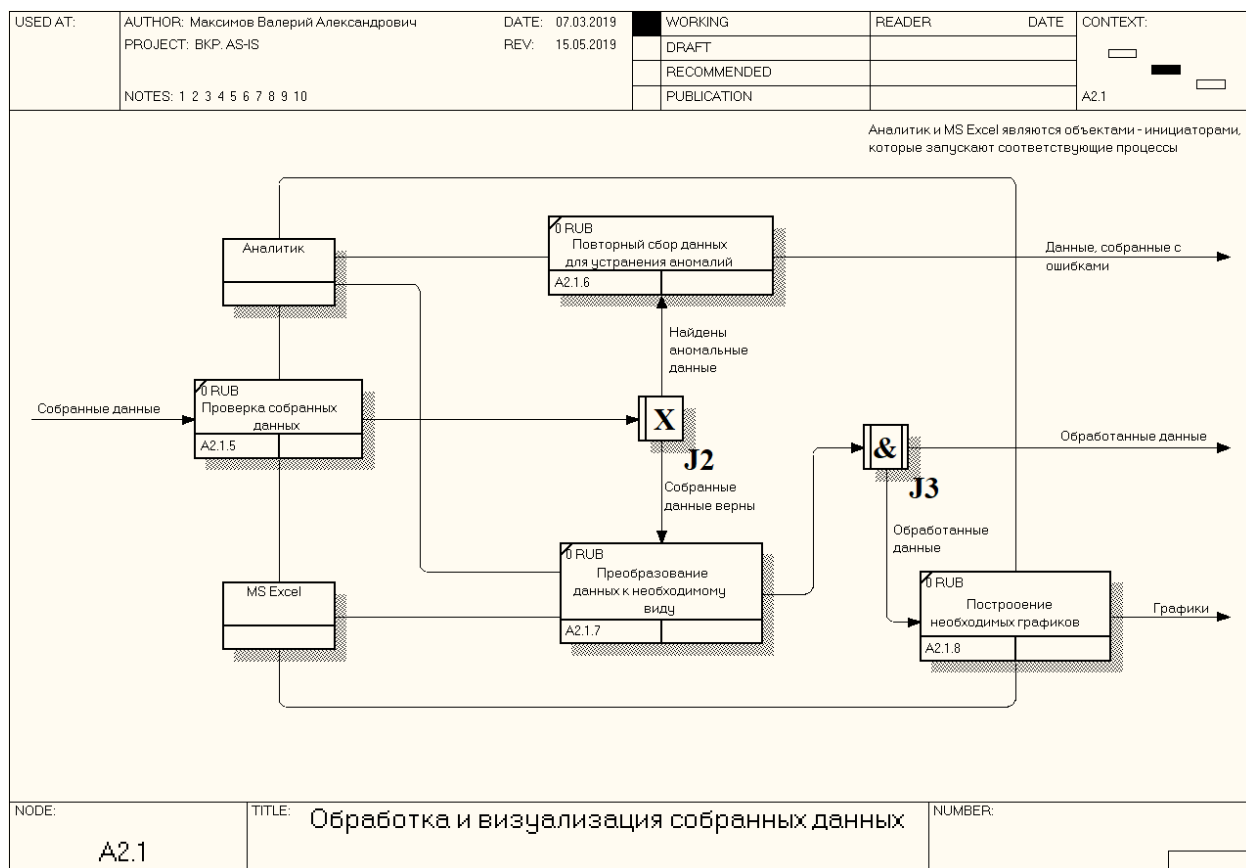


Рисунок 5 – Декомпозиция процесса «Обработка и визуализация собранных данных»

При анализе данных диаграмм были выделены следующие недостатки в работе аналитика:

1. Процесс сбора данных цикличен, при этом в зависимости от целей аналитика, могут потребоваться данные за разные даты, а значит этот цикл может продолжаться довольно длительное время.

2. Вызывает сложности процесс проверки собранных данных, так как аналитик, в силу человеческого фактора, может упустить аномальные (выбивающиеся из общего формата) данные из виду, следствием чего может стать

снижение качества будущего анализа и принимаемых решений.

3. Большое количество данных и построенных графиков может значительно замедлить скорость работы вычислительной машины, что также усложнит работу аналитика.

4. Высокая стоимость затрат (при условии, что аналитик получает почасовую оплату в размере 400 рублей в час) на проведение процессов сбора и обработки данных.

1.3.2 Цели и задачи автоматизации

Процесс сбора информации по мобильным приложениям является отправным для проведения анализа рынка мобильных приложений. Ручной сбор информации значительно увеличивает время проведения анализа рынка мобильных приложений. Автоматизация сбора данных позволит значительно ускорить процесс анализа и снизит нагрузку на аналитика.

Автоматизация процесса обработки данных также позволит снизить нагрузку на аналитика и ускорить процесс проведения анализа.

За счет использования разрабатываемой системы, аналитик сможет использовать собранные системой данные в построении собственных графиков, использовать графики, построенные системой. Также руководство предприятия сможет значительно снизить затраты на оплату работы аналитика.

1.3.3 Анализ существующих разработок и сравнительная характеристика программного обеспечения

Для оценки конкурентоспособности разрабатываемого продукта был произведен анализ и сравнение с выбранным аналогом по функциональному назначению, основным техническим и эксплуатационным параметрам, областям применения. Подобный анализ осуществляется при помощи оценки эксплуатационно-технического уровня разрабатываемого продукта с использованием метода бально-индексной оценки в соответствии с ГОСТом Р ИСО/МЭК 9126-93.

Эксплуатационно-технический уровень (ЭТУ) разрабатываемого продукта – это обобщенная характеристика его эксплуатационных свойств, возможностей, степени новизны, являющихся основой качества продукта.

При изучении предметной области решаемой задачи было выбрано два аналога разрабатываемой системы «MobAStA» – «AppAnnie» и «APPFOLLOW». Однако прямых аналогов разрабатываемой системе, которые выполняют те же функции, найдено не было. Поэтому к рассмотрению были приняты системы, которые имеют схожую направленность с разрабатываемой системой. Показатели качества были выбраны в соответствии с требованиями к разрабатываемой системе «MobAStA». Коэффициенты весомости для каждого показателя были выставлены в соответствии со значимостью этих показателей в разрабатываемой системе «MobAStA». По

системам-аналогам показатели качества были выставлены в соответствии с отзывами пользователей.

Расчет обобщенных показателей качества для каждой системы приведен в таблице 1.

Таблица 1 – Расчет обобщенных показателей качества

Показатели качества	Коэффициент весомости, B_j	MobAStA		App Annie		APPFOLLOW	
		X_j	$\frac{B_j^*}{X_j}$	X_j	$\frac{B_j^*}{X_j}$	X_j	$\frac{B_j^*}{X_j}$
1. Простота использования	0,15	4	0,6	3	0,45	5	0,75
2. Новизна (соответствие современным требованиям)	0,1	3	0,3	3	0,3	2	0,2
3. Агрегирование данных	0,25	5	1,25	3	0,75	2	0,5
4. Разнообразие представляемой информации	0,15	3	0,45	5	0,75	2	0,3
5. Стоимость	0,2	5	1	1	0,2	3	0,6
6. Визуализация предоставляемой информации	0,15	4	0,6	3	0,45	1	0,15
Обобщенный показатель качества $J_{ЭТУ}$		$J_1 = 4,20$		$J_2 = 2,90$		$J_3 = 2,50$	

Оправданность разработки собственной системы определяется коэффициентом технического уровня, который является отношением обобщенного показателя качества разрабатываемой системы к обобщенному показателю качества аналогов.

Коэффициент технического уровня для системы «App Annie»:

$$A_1 = \frac{J_1}{J_2} = \frac{4,20}{2,90} = 1,45$$

Коэффициент технического уровня для системы «APPFOLLOW»:

$$A_2 = \frac{J_1}{J_3} = \frac{4,20}{2,50} = 1,68$$

По полученным коэффициентам технического уровня видно, что разработка собственной системы «MobAStA» с технической точки зрения вполне оправдана, так как коэффициенты больше 1.

Низкие обобщенные показатели качества для систем-аналогов объясняются тем, что выбранные системы не имеют функций агрегирования данных и их визуализации, являющихся основными для разрабатываемой системы.

1.3.4 Обоснование выбора технологии проектирования

Методология проектирования представляет собой принципы проектирования, реализуемые набором методов проектирования, поддерживаемых средствами проектирования. Главный принцип построения различных систем – принцип иерархической декомпозиции. Данный принцип включает две группы методологий: объектно-ориентированная методология и структурно-функциональная методология [2][6].

Объектно-ориентированный подход строится на изначальном выделении объектов. Так как в рассматриваемой предметной области выделить полноценные объекты, а также определить связь между ними довольно сложно, для исследования предметной области выбран структурно-функциональный подход, в основе которого лежит принцип функциональный декомпозиции.

В структурном анализе используются в основном две группы средств, иллюстрирующих функции, выполняемые системой и отношения между данными. Каждой группе средств соответствуют определенные виды моделей (диаграмм), наиболее распространенными среди которых являются следующие:

- SADT (Structured Analysis and Design Technique) модели и соответствующие функциональные диаграммы;
- DFD (Data Flow Diagrams) диаграммы потоков данных;
- ERD (Entity-Relationship Diagrams) диаграммы «сущность-связь».

Модели SADT, а в частности IDEF0 и IDEF3, наиболее часто используемые при построении функциональных моделей, так как они наглядно отражают функциональную структуру объекта, а именно выполняемые процессы и связи между этими процессами. Благодаря чему можно определить логику процессов, выполняемых на предприятии, и взаимодействие их. Основным плюсом данной нотации является то, что можно получить достаточную информацию о каждом процессе. Благодаря ей можно определить все недостатки процесса и его реализации: дублирование функций, отсутствие механизмов, регламентирующих данный процесс, отсутствие контрольных переходов и т.д.

DFD предоставляет возможность рассмотреть информационное пространство системы. Она используется для описания потоков информации внутри ИС, что позволяет использовать DFD диаграммы как дополнение к модели бизнес-процессов, выполненной в IDEF0 (IDEF3) [6].

ERD модели позволяют отобразить набор объектов, их атрибутов и связей между ними в рассматриваемой предметной области. Эти модели значительно упрощают процесс проектирования базы данных системы. На основе данной модели строится процесс моделирования базы данных в методологии IDEF1X, позволяющей практически в полной мере описать конечный вид физической модели реляционной базы данных [7].

1.4 Общая характеристика организации решения задачи автоматизации процессов сбора и обработки информации по мобильным приложениям

1.4.1 Постановка задачи

Создание системы сбора и обработки информации о мобильных приложениях – одна из основных задач маркетингового отдела в компаниях, занимающихся разработкой мобильных приложений. Информация о приложениях, которую может предоставить разрабатываемая система, необходима при проведении анализа. Знать преобладающие категории в каждой коллекции, средние рейтинги приложений и подобную информацию необходимо для того, чтобы можно было наметить основополагающие характеристики разрабатываемого приложения. Важнейшим процессом в маркетинговом отделе является процесс принятия решений о дальнейшем пути разработки. Не имея требуемой информации (или при её ручном сборе), процесс принятия решений может значительно затянуться и снизить эффективность работы маркетингового отдела.

Использование ИС сбора и обработки данных о мобильных приложениях даст следующие эффекты:

- аналитик сможет сократить время на анализ информации;

- начальный набор графиков позволит аналитику сократить время на их построение;

- повысится качество исследования за счет компьютерной обработки собранных данных;

- эффективность деятельности маркетингового отдела значительно повысится за счет сокращения времени анализа.

В таблицах 2 и 3 отображено формализованное описание входных показателей и формализованное описание результатных показателей. Это необходимо для будущего проектирования программного продукта и базы данных.

Т а б л и ц а 2 – Входные данные

Наименование входного показателя	Идентификатор входного показателя
Дата считывания данных i-той коллекции	Date_point
Наименование приложения	Title
Цена приложения	Price
Платное / Бесплатное	Free
Встроенные покупки: есть / нет	IAP
Ранжирование встроенных покупок	IAP Range
Возрастной рейтинг	Content rating
Количество скачиваний	Installs
Категория	Category

Так как система помимо сбора и обработки данных выполняет и визуализацию основной статистики по

собранным данным, то в результативных данных будут именно графики, описывающие статистику по показателям, определенным на вход, а также описывающие отношение между некоторыми из этих показателей.

Т а б л и ц а 3 – Результатные данные

Наименование результатного показателя	Алгоритм предоставления результата
Среднее значение i-го показателя j-той коллекции по выборке из N приложений	$\frac{\sum_{k=1}^N i_k}{N}$
Отношение i-того и k-того параметров j-той коллекции по выборке из N приложений	$\frac{\sum_{n=1}^N i_n}{\sum_{m=1}^N k_m}$

1.4.2 Входная информация

На вход система получает код DOM-дерева страницы на языке разметки HTML, пример которого представлен на рисунке 6.

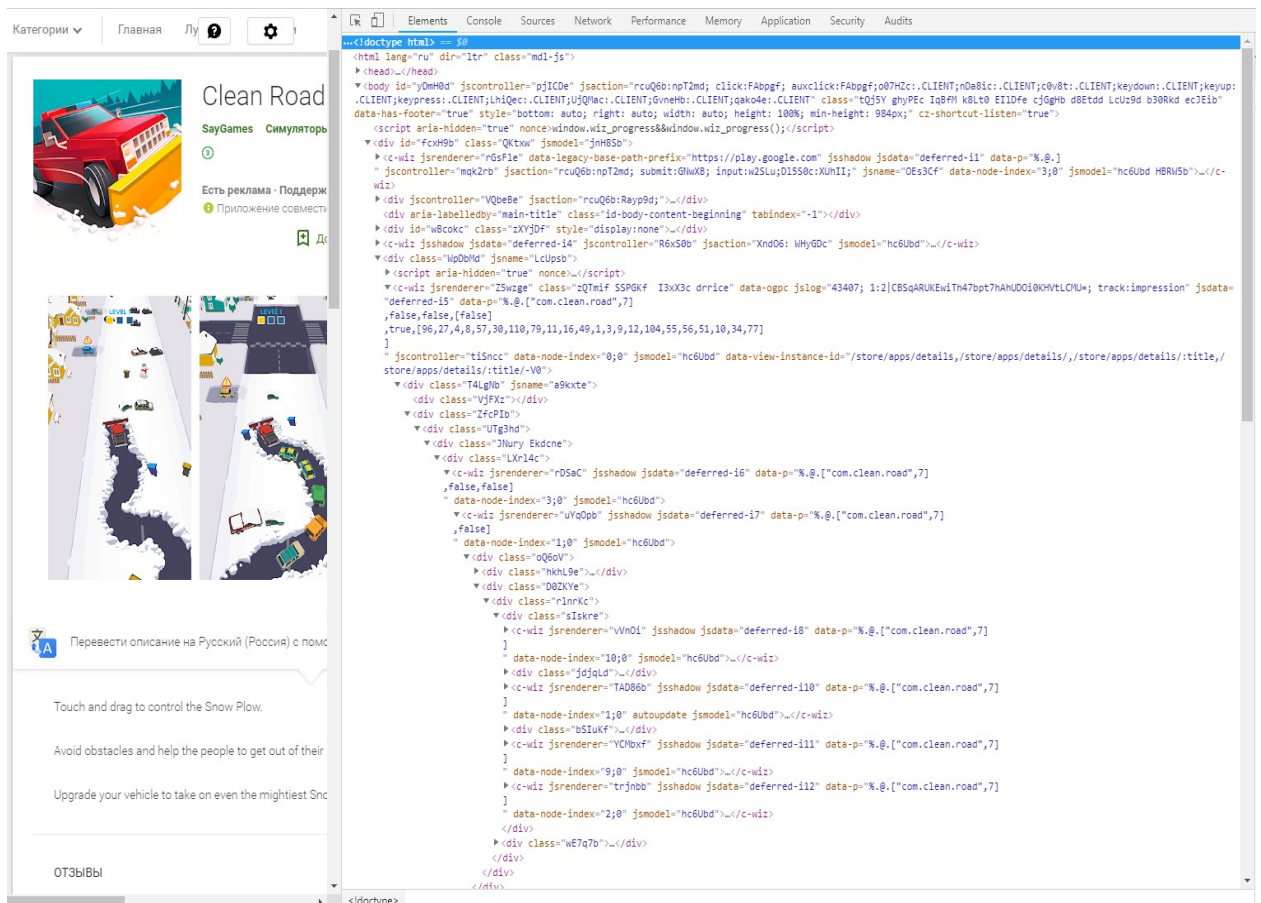


Рисунок 6 – Пример HTML-кода страницы, поступающего на вход

Данный код анализируется системой, после чего выделяются необходимые данные. Выделенные данные записываются системой в базу данных, откуда и используются системой в дальнейшем. Из базы в систему данные поступают в формате JSON-объекта, пример которого представлен на рисунке 7.


```

app = {
  'date_point': '2019-04-11',
  'app_id': 'a2z.Mobile.Event4960',
  'title': 'Craft Brewers Conference 2019',
  'score': 2.3,
  'price': '0',
  'free': 'Free',
  'iap': False,
  'iap_range': None,
  'size': '16M',
  'installs': '+100',
  'content_rating': 'Rated for 3+',
  'category': 'EVENTS',
}

```

Рисунок 7 – Пример JSON-объекта, поступающего на вход

1.4.3 Выходная информация

На выход система предоставляет набор графиков, которые могут помочь аналитику в принятии решений. Каждый график предоставляет определенный вид сводной информации. Так на рисунке 8 представлен график статистики по жанрам, отображающий количество приложений в каждом жанре.

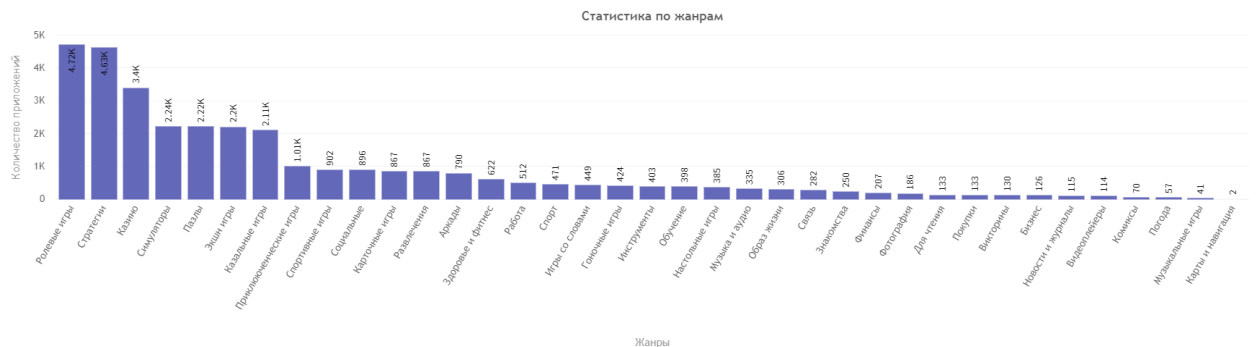


Рисунок 8 – Гистограмма статистики по количеству приложений в каждом жанре

На рисунке 9 изображена круговая диаграмма статистики по наличию встроенных покупок, отображающая процентное соотношение приложений, имеющих и не имеющих встроенные покупки к общему числу приложений,

а также средняя минимальная и средняя максимальная цены на единичную встроенную покупку.



Рисунок 9 – Круговая диаграмма по наличию встроенных покупок

На рисунке 10 приведена гистограмма, отображающая статистику по возрастным категориям, отображающая количество приложений в каждой возрастной категории.



Рисунок 10 – Гистограмма статистики по количеству приложений в каждой возрастной категории

Гистограмма, отображающая средние пользовательские рейтинги в каждой категории приведена на рисунке 11.

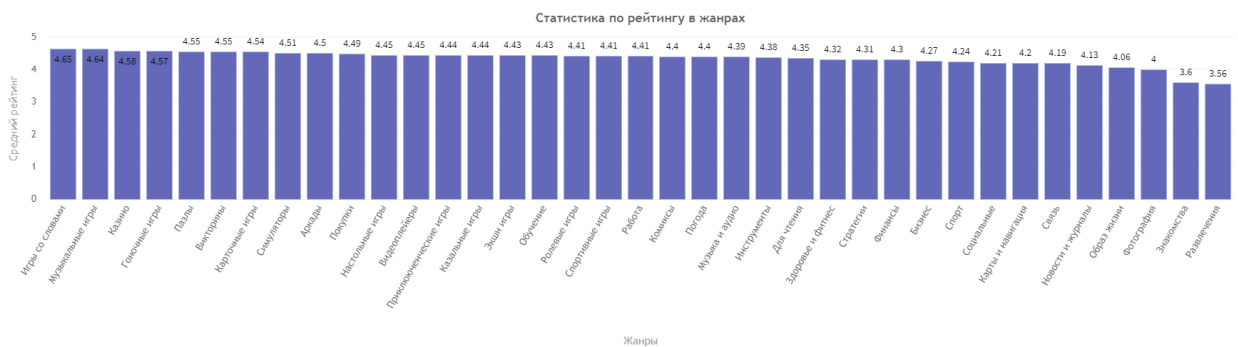


Рисунок 11 – Гистограмма средних рейтингов в каждой категории

Круговая диаграмма, отражающая процентное соотношение платных и бесплатных приложений к общему числу приложений представлена на рисунке 12.

Статистика по (бес)платным приложениям

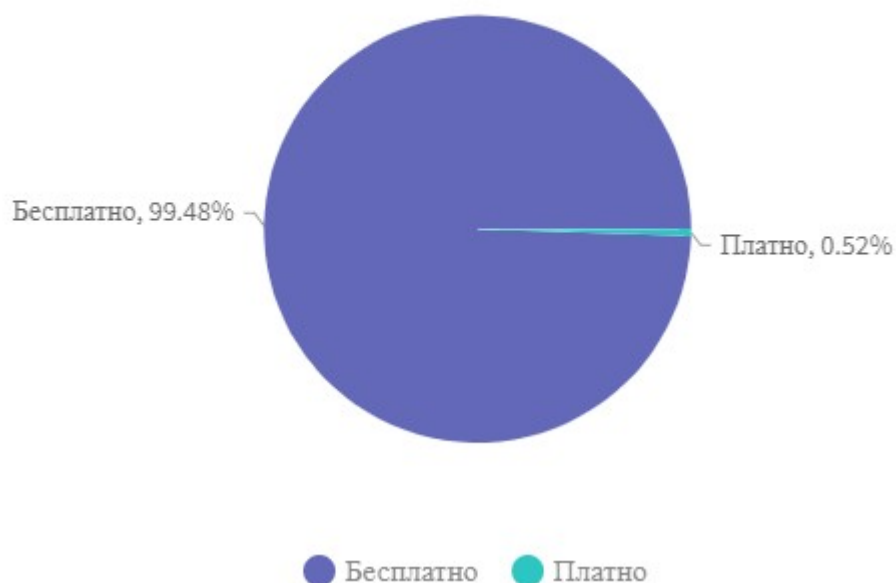


Рисунок 12 – Круговая диаграмма статистики по платным и бесплатным приложениям

Статистика по рейтингам при наличии или отсутствии встроенных покупок отображена на гистограмме, представленной на рисунке 13.

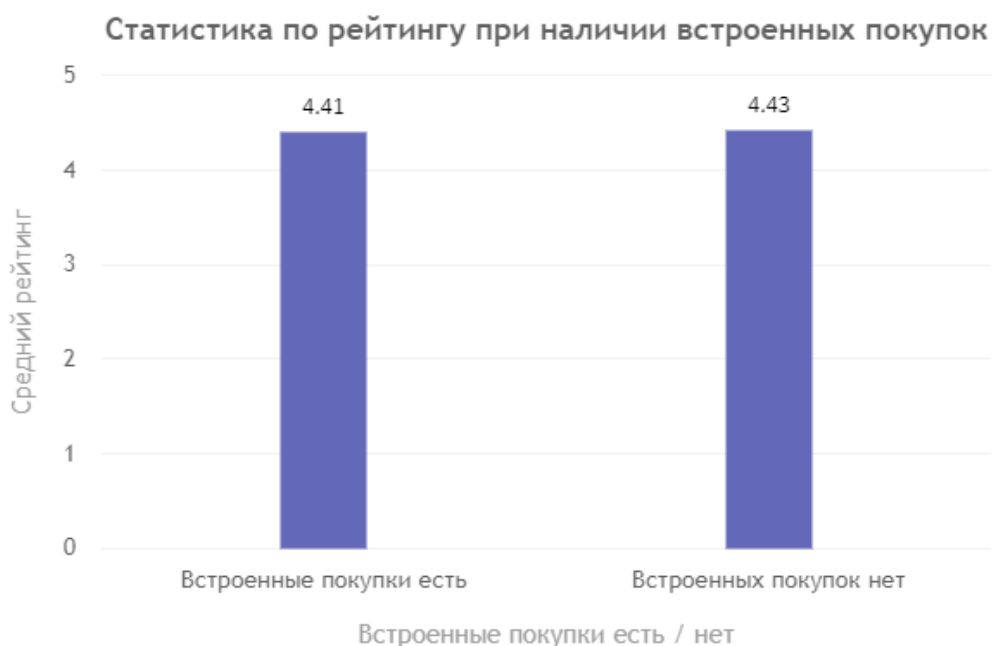


Рисунок 13 – Гистограмма статистики по рейтингам при наличии или отсутствии встроенных покупок

Статистика средних рейтингов в каждой возрастной категории приведена на гистограмме, представленной на рисунке 14.



Рисунок 14 – Гистограмма статистики по рейтингу в возрастных категориях

Гистограмма, отображающая статистику по средним количествам скачиваний в каждой возрастной категории, приведена на рисунке 15.



Рисунок 15 – Гистограмма статистики по скачиваниям в возрастных категориях

Статистика, отображающая среднее число скачиваний в каждой категории приложений отражена на гистограмме, приведенной на рисунке 16.

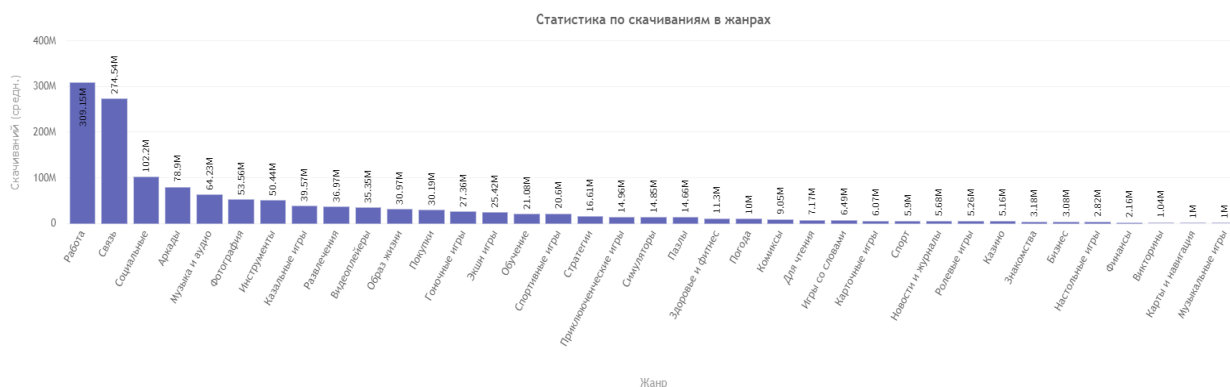


Рисунок 16 – Гистограмма статистики по скачиваниям в жанрах

Гистограмма, на которой отображена статистика по среднему количеству скачиваний в каждой группе пользовательских рейтингов.

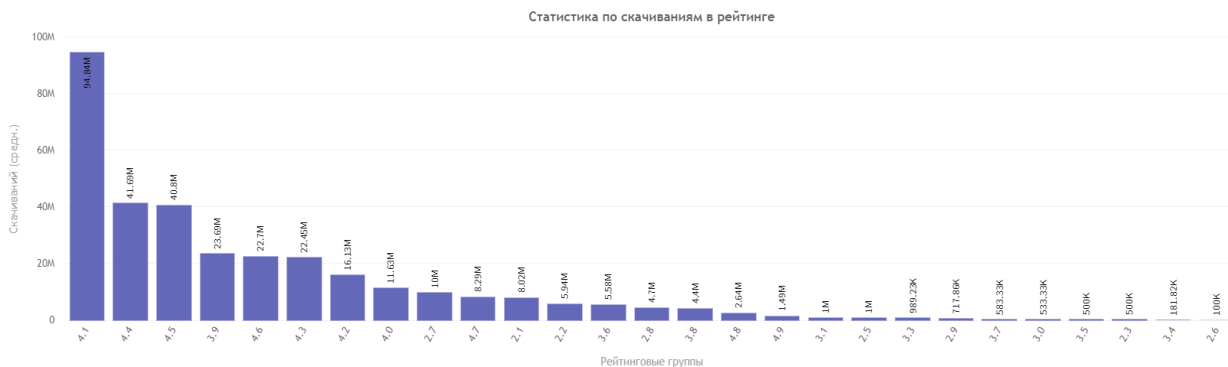


Рисунок 17 – Гистограмма статистики по скачиваниям в рейтинговых группах

Графики, формируемые системой, были выбраны на основании научных статей аналитиков в данной области, в результате изучения которых можно определить какая именно информация интересна аналитику при проведении анализа рынка мобильных приложений.

2 Проектирование системы анализа и разработка программного обеспечения

2.1 Разработка проекта системы автоматизации работы аналитика

2.1.1 Описание содержания проекта

Иерархическая структура работ – это метод разбиения всей работы, которую необходимо выполнить для достижения целей проекта, на более мелкие операции и действия до элементарных, то есть таких, которые могут быть оценены.

Первый этап при разработке проекта – составление иерархической структуры для разрабатываемого проекта. Целью данного проекта будет являться разработка системы для проведения анализа рынка приложений для мобильных устройств по выбранным критериям, способной собирать, хранить и обрабатывать данные рынка мобильных приложений. Данная система избавит аналитика от выполнения трудоемкой, длительной и дорогой работы по сбору необходимых ему данных и будет предоставлять данные рынка мобильных приложений аналитику в форме, удобной для проведения анализа. Плановый срок начала проекта – 01.12.2018, а крайний срок выполнения проекта – 23.06.19.

Итоговая структура работ проекта представлена на рисунке 18.

	i	Режим задачи ▼	Название ▼
1		→	▲ Создание системы обработки и анализа Больших Данных рынка мобильных приложений
2		→	▲ Формирование системных требований
3		→	Анализ предметной области
4		→	Анализ существующих решений
5		→	Определение и анализ требований к системе
6		→	Завершение разработки требований
7		→	▲ Проектирование системы
8		→	Структурное моделирование
9		→	Функциональное моделирование
10		→	Разработка макетов страниц
11		→	Проектирование базы данных
12		→	Завершение проектирования
13		→	▲ Реализация системы
14		→	Реализация механизма получения данных
15		→	Реализация алгоритмов обработки данных
16		→	Реализация базы данных
17		→	Реализация серверной части системы
18		→	Верстка пользовательской части системы
19		→	Подключение базы данных к системе
20		→	Тестирование системы
21		→	Доработка системы
22		→	Завершение проекта

Рисунок 18 – Структура проекта

Задачи, выполняемые при создании системы обработки и анализа Больших Данных рынка мобильных приложений, и представленные на рисунке 18, описывают последовательный подход к созданию системы.

Так первым при создании системы была выделена фаза «Формирование системных требований», включающая в себя следующие задачи:

1. Анализ предметной области – исследование предметной области с целью определения ее сущностей,

первоначальных требований к функциональности системы и определения границы проекта.

2. Анализ существующих решений – изучение существующих решений с целью выявления их плюсов и минусов.

3. Определение и анализ требований к разрабатываемой системе – разработка, на основе анализа существующих решений и анализа предметной области, требований к системе, их систематизация, выявление взаимосвязей, а также документирование.

Следующей была выделена фаза «Проектирование системы», которая определяет следующие задачи:

1. Структурное моделирование – построение диаграммы классов, диаграммы компонентов и диаграммы развертывания.

2. Функциональное моделирование – построение диаграммы системных прецедентов, диаграммы деятельности и диаграммы взаимодействия.

3. Разработка макетов страниц.

4. Проектирование базы данных.

Завершающей была определена фаза «Реализация системы», на которой выполняются задачи:

1. Реализация механизма получения данных – разработка модуля, занимающегося сбором требуемых данных.

2. Реализация алгоритмов обработки – реализация алгоритмов обработки собранных данных.

3. Реализация базы данных.

4. Реализация серверной части системы - реализация сервера, объединяющего в себе готовые части системы, работающие с данными.

5. Верстка пользовательской части системы - верстка веб-страниц, на которых данные будут представлены пользователю в форме, удобной для проведения анализа.

6. Тестирование системы.

7. Доработка системы.

2.1.2 Оценка длительности операций

Для определения ожидаемой продолжительности работы $T_{ож}$ применяется метод PERT (Program Evaluation and Review Technique). Оценка осуществляется по формуле 1:

$$T_{ож} = \frac{t_{мин} + 4t_{нв} + t_{макс}}{6} \quad (1)$$

где $t_{мин}$ - кратчайшая продолжительность данной работы (оптимистическая оценка);

$t_{макс}$ - самая большая продолжительность работы (пессимистическая оценка);

$t_{нв}$ - наиболее вероятная продолжительность работы (реалистическая оценка).

Для получения целочисленных значений, расчет был произведен в минутах.

Таким образом для каждой задачи был произведен расчет, результаты которого приведены в таблице 4.

Т а б л и ц а 4 - Оценка продолжительности работ

Виды работ	Оптимис т. оценка, $t_{мин}$	Реалист. оценка, $t_{нв}$	Пессимис т. оценка, $t_{макс}$	Ожидаем ая продолж. работы, $T_{ож}$
------------	---------------------------------------	---------------------------------	--------------------------------------	--------------------------------------------------

Анализ предметной области	2880	7200	11520	7200
Анализ существующих решений	1440	4320	7200	4320
Определение и анализ требований к системе	2016 0	2592 0	40320	27360
Структурное моделирование	5760	1296 0	20160	12960
Функциональное моделирование	8640	1440 0	37440	17280
Разработка макетов страниц	2880	7200	11520	7200
Проектирование базы данных	1440	4320	7200	4320

Окончание таблицы 4

Виды работ	Оптимист. т. оценка, $t_{мин}$	Реалист. оценка, $t_{нв}$	Пессимист. т. оценка, $t_{макс}$	Ожидаем ая продолж. работы, $T_{ож}$
Реализация механизма получения данных	4896 0	8064 0	86400	76320
Определение алгоритмов обработки данных	2016 0	2448 0	37440	25920
Реализация базы данных	8640	1008 0	20160	11520
Реализация серверной части системы	3600 0	4032 0	44640	40320
Верстка пользовательской части системы	2304 0	2880 0	43200	30240
Подключение базы данных к системе	1440	2880	4320	2880
Тестирование системы	2880	7200	11520	7200
Доработка системы	1440	2880	4320	2880
ИТОГО (в днях)	129	190	269	193

Как видно итоговые результаты таблицы, а в частности ожидаемая продолжительность работ близка к выбранной реалистичной оценке. Из этого следует, что план составлен с довольно высокой степенью оптимальности.

В процессе создания системы принимает участие не только ее непосредственный исполнитель – студент, но и руководитель дипломной работы. Именно он обеспечивает студента необходимыми материалами по дипломной работе, а также сопровождает студента при выполнении фазы формирования системных требований и отслеживает выполнение поставленных студентом задач, и корректирует их. Помимо научного руководителя, при написании дипломной работы, студент консультируется по вопросам моделирования с преподавателем, которого также требуется учесть.

Определенные в данном проекте трудовые и материальные ресурсы представлены на рисунке 19.


		Название ресурса	Тип	Единицы измерения материалов	Краткое название	Макс. единиц	Группа	Стандартная ставка
1		Научный руководитель (Оськин Д. А.)	Трудовой		НР	100%		110,00р./час
2		Преподаватель (Бедрина С. Л.)	Трудовой		п	100%		200,00р./час
3		Студент (Максимов В. А.)	Трудовой		С	100%		0,00р./час
4		Бумага	Материальный		Б			350,00р.
5		Компьютер	Материальный		К			0,00р.

Рисунок 19 – Ресурсы проекта

Как было описано ранее, роль научного руководителя (Оськина Д. А.) заключается в контроле исполнения студентом задач по созданию системы и консультировании по различным вопросам, а также в предоставлении материала студенту для его изучения, с целью использования при выполнении фазы «Формирование системных требований».

Студент (Максимов В. А.) – непосредственный исполнитель задач. Научный руководитель занимается поддержкой студента при выполнении научной работы, но выполнением работ занимается сам студент. Производство анализа предметной области, анализа аналогов, моделирования системы, её разработки и так далее – всё это входит в задачи студента.

Преподаватель (Бедрина С. Л.) выполняет роль консультанта, то есть помогает студенту при решении вопросов в области моделирования системы.

При распределении выделенных трудовых и материальных ресурсов на исполнение задач учитывался тот факт, что в большинстве задач научный руководитель, как и

преподаватель, либо не участвует вовсе, либо занимается контролем или поддержкой исполнения того или иного этапа выполнения работы студентом, что нельзя назвать незначительным, однако загрузка данных исполнителей при этом составит от 15% до 50% от их максимально возможной загруженности.

Результат распределения ресурсов по задачам можно увидеть на рисунке 20 в поле «Названия ресурсов».

	Режим задачи	Название	Названия ресурсов
1		▲ Создание системы обработки и анализа Больших Данных рынка мобильных приложений	
2		▲ Формирование системных требований	Бумага[1];Компьютер[1]
3		Анализ предметной области	Студент (Максимов В. А.)[80%]; Научный руководитель (Оськин Д. А.)[30%]
4		Анализ существующих решений	Студент (Максимов В. А.)[20%]; Научный руководитель (Оськин Д. А.)[15%]
5		Определение и анализ требований к системе	Студент (Максимов В. А.); Научный руководитель (Оськин Д. А.)[50%]
6		Завершение разработки требований	Студент (Максимов В. А.); Научный руководитель (Оськин Д. А.)
7		▲ Проектирование системы	Бумага[1];Компьютер[1]
8		Структурное моделирование	Студент (Максимов В. А.); Научный руководитель (Оськин Д. А.)[40%]; Преподаватель (Бедрина С. Л.)[15%]
9		Функциональное моделирование	Студент (Максимов В. А.)[80%]; Преподаватель (Бедрина С. Л.)[20%]
10		Разработка макетов страниц	Студент (Максимов В. А.)[20%]; Научный руководитель (Оськин Д. А.)[15%]
11		Проектирование базы данных	Студент (Максимов В. А.)
12		Завершение проектирования	Студент (Максимов В. А.); Научный руководитель (Оськин Д. А.)
13		▲ Реализация системы	Бумага[1];Компьютер[1]
14		Реализация механизма получения данных	Студент (Максимов В. А.)[70%]
15		Реализация алгоритмов обработки данных	Студент (Максимов В. А.)[15%]
16		Реализация базы данных	Студент (Максимов В. А.)[5%]
17		Реализация серверной части системы	Студент (Максимов В. А.)
18		Верстка пользовательской части системы	Студент (Максимов В. А.)[10%]
19		Подключение базы данных к системе	Студент (Максимов В. А.)
20		Тестирование системы	Студент (Максимов В. А.)
21		Доработка системы	Студент (Максимов В. А.)
22		Завершение проекта	Научный руководитель (Оськин Д. А.); Студент (Максимов В. А.)

Рисунок 20 – Распределение ресурсов по задачам

В результате было рассчитано, что сумма всех затрат на достижение цели проекта, учитывая стандартные ставки преподавательского состава ДВФУ и стандартную стоимость пачки бумаги, необходимой для набросков алгоритмов, чертежей блок-схем и других задач, составит 6 962 р. Из них 1 050 р. составляют затраты на материальные ресурсы, расходуемые в ходе выполнения задач.

2.1.4 Идентификации рисков и разработка стратегии их смягчения

Каждая задача, описанная в содержании проекта, тем или иным образом подвержена влиянию рисков.

Управление рисками проекта – это систематический процесс идентификации, анализа и реагирования на проектные риски.

Начать идентификацию рисков следует с определения задач, составляющих критический путь. Выполняется это с использованием встроенных средств программного продукта «Microsoft Project».

На рисунке 21 приведена диаграмма Ганта, где задачи, лежащие на критическом пути помечены красным.

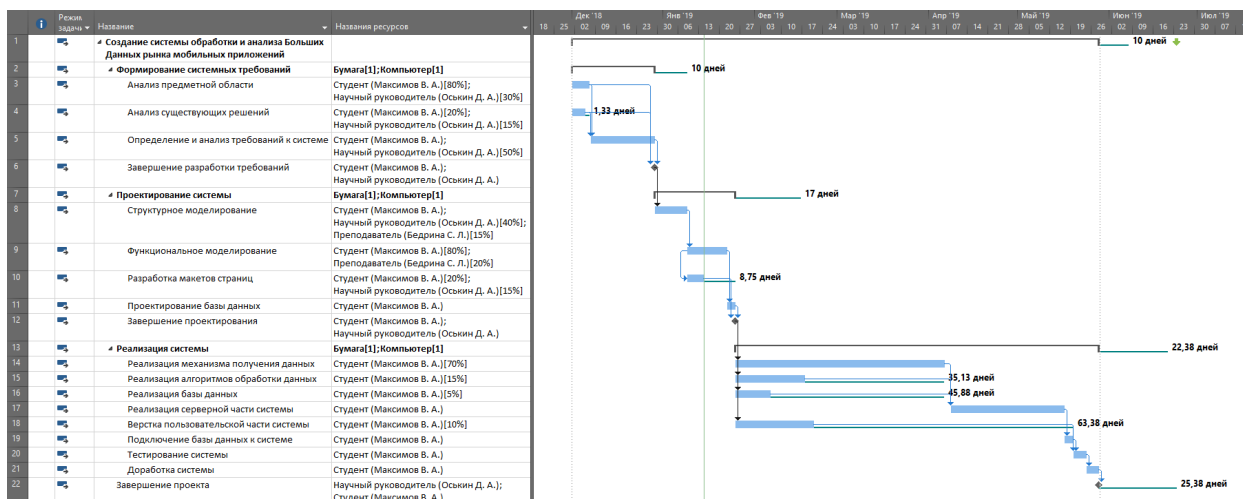


Рисунок 21 – Задачи критического пути проекта

Как видно на рисунке 21, при указании длительности работ равной определенной ранее ожидаемой длительности работ, все задачи проекта имеют довольно внушительный временной резерв, а окончание самого проекта происходит практически за месяц до крайнего срока.

Однако выделить риски, которые могут возникнуть при выполнении задач можно. Так, например, длительность наиболее трудоемкой задачи по реализации механизма получения данных может быть значительно увеличена по причине изменения формы хранения и представления данных на сайте Google Play. Так как механизм получения данных не является универсальным, то малейшие изменения в форме хранения этих данных может привести к полной неработоспособности этого механизма, а значит его разработку потребуется начинать с самого начала.

Планы смягчения данного риска приведены на рисунке 22.

13		Реализация системы	Бумага[1];Компьютер[1]
14		Реализация механизма получения данных	Студент (Максимов В. А.)[70%]
15		Обеспечение работоспособности системы	Студент (Максимов В. А.)[70%]

	Название задачи	Описание риска	Вероятность возникновения риска	План по смягчению риска	План реакции на риск
14	Реализация механизма получения данных	Риск изменения типа хранения данных на сайте Google Play, который приведёт к неработоспособности механизма получения данных	Низкая	Изучить дополнительную информацию по получению данных с веб-сайтов и исследовать возможность частичной универсализации текущего механизма	Изменить механизм получения данных, подстроившись под новую форму их представления

Рисунок 22 – Планы по смягчению риска задержки определения темы ВКР

Так как данный проект, в результате оптимизации, не имеет задач, лежащих на критическом пути, а временной резерв имеется практически у каждой задачи, то идентификация рисков, связанных с выполнением этих задач, не является сложной задачей.

Однако сложности может составить появление ресурсных рисков. Например, невозможность встречи с научным руководителем. Такие риски имеют невысокую вероятность, но требуют описания плана по их смягчению во избежание проблем, которые они могут составить.

Таким образом был составлен план по смягчению описанного ранее риска, приведенный на рисунке 23.

	Название ресурса	Описание риска	Вероятность возникновения риска	План по смягчению риска
1	Научный руководитель (Оськин Д. А.)	Риск невозможности встречи и консультации с научным руководителем, когда это необходимо; Недоступность научного руководителя по причине его занятости в МГУ	Средняя	Требуется обменяться контактами с научным руководителем, в том числе и электронной почтой, чтобы иметь возможность связи и демонстрации требуемой части выполнения работы
2	Преподаватель (Бедрина С. Л.)			
3	Студент (Максимов В. А.)			
4	Бумага			
5	Компьютер			

Рисунок 23 – План по смягчению ресурсного риска

Бюджетные риски – вид рисков, связанный с превышением ограничений по установленному бюджету.

Учитывая то, что при создании системы на студента ограничения по бюджету не накладываются, то и рассмотрение рисков данного вида не имеет смысла.

2.2 Анализ требований к разрабатываемой системе

После анализа модели AS-IS было выявлено, что процесс сбора данных цикличен, при этом в зависимости от целей аналитика, могут потребоваться данные за разные даты, а значит этот цикл может продолжаться довольно длительное время. А также опасным является процесс проверки собранных данных, так как аналитик, в силу человеческого фактора, может упустить аномальные (выбивающиеся из общего формата) данные из виду, следствием чего может стать снижение качества будущего анализа и принимаемых решений. При этом большое количество данных и построенных графиков может значительно замедлить скорость работы вычислительной машины, что также усложнит работу аналитика.

По результатам исследования недостатков деятельности аналитика была построена структурно-функциональная модель деятельности, в соответствии со стандартом IDEF0, описывающая организацию работы на предприятии «как должно быть» (ТО-ВЕ), приведенная ниже.

ТО-ВЕ модель автоматизируемого процесса анализа позволит определить основные требования к системе, направленной на автоматизацию деятельности производимой аналитиком в компаниях по разработке мобильных приложений.

На рисунке 24 представлена контекстная TO-BE диаграмма процесса анализа рынка мобильных приложений.

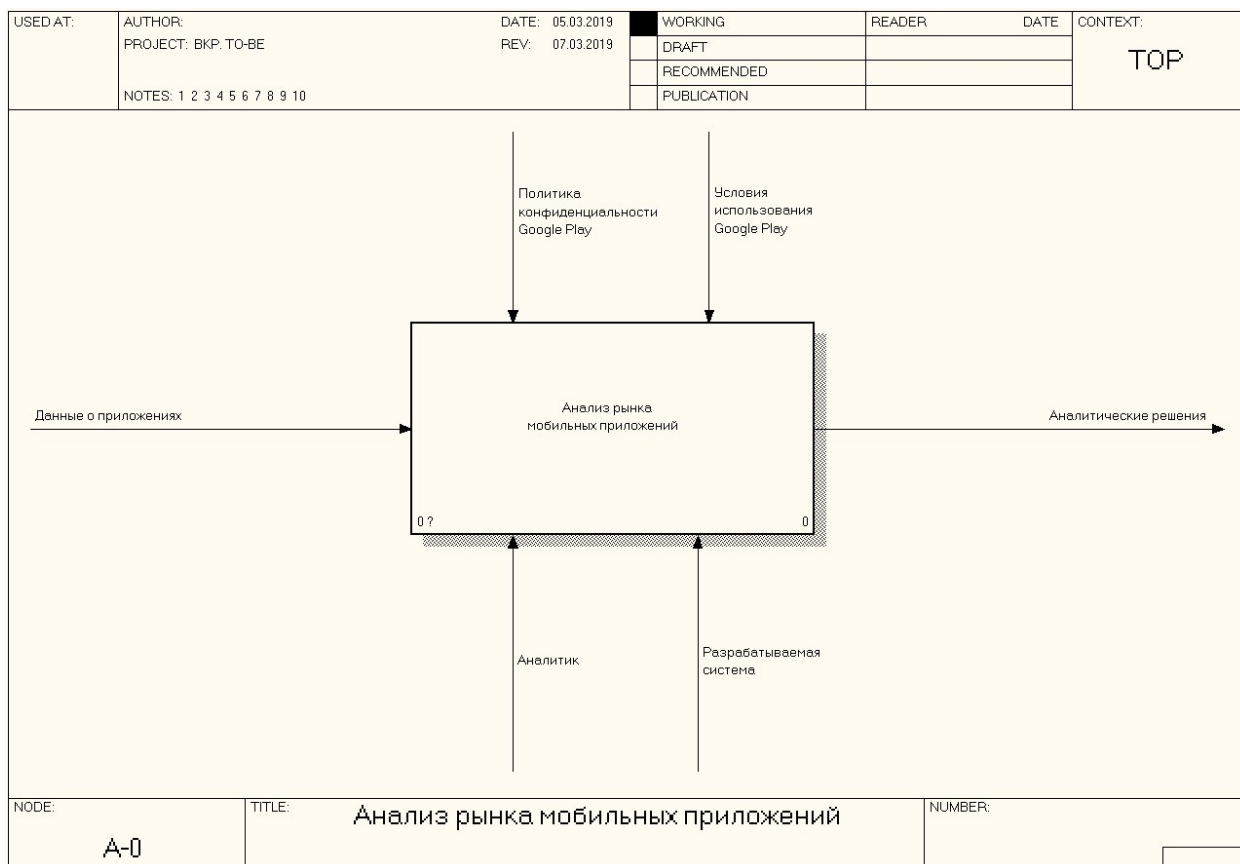


Рисунок 24 – Контекстная диаграмма модели TO-BE

Данная диаграмма описывает основные входные и выходные данные процесса анализа рынка мобильных приложений, но не отображает обмена данными между подпроцессами, происходящими внутри данного процесса.

На рисунке 25 приведена декомпозиция контекстной TO-BE диаграммы процесса анализа рынка мобильных приложений.

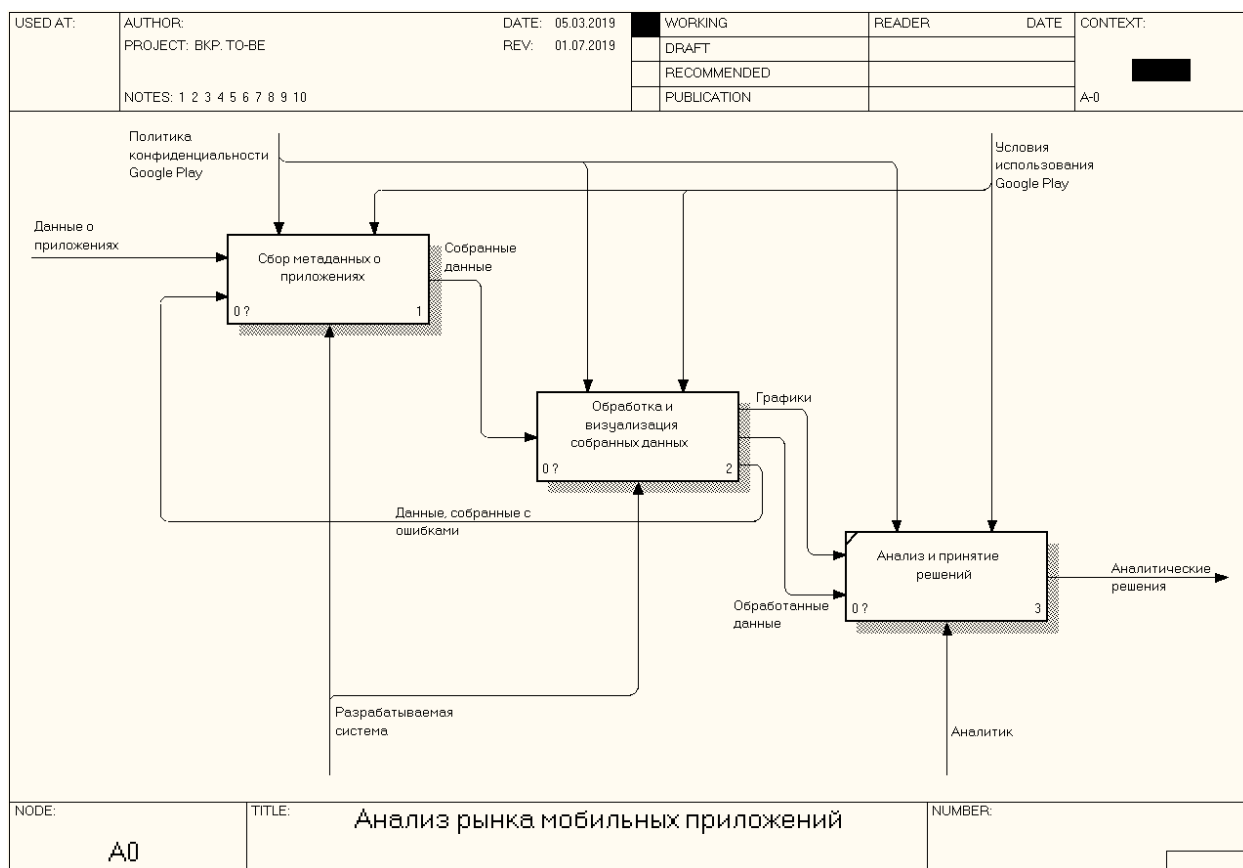


Рисунок 25 – Декомпозиция контекстной диаграммы

На рисунке 26 представлена декомпозиция процесса «Сбор мета-данных о приложениях», построенная по методологии IDEF3, позволяющая отобразить цикличность процесса сбора данных, а также изменение объектов инициаторов.

Как видно объект «Аналитик» был заменен объектом «Разрабатываемая система», а объект «MS Excel» был удалён, так как теперь хранением собранных данных также занимается «Разрабатываемая система».

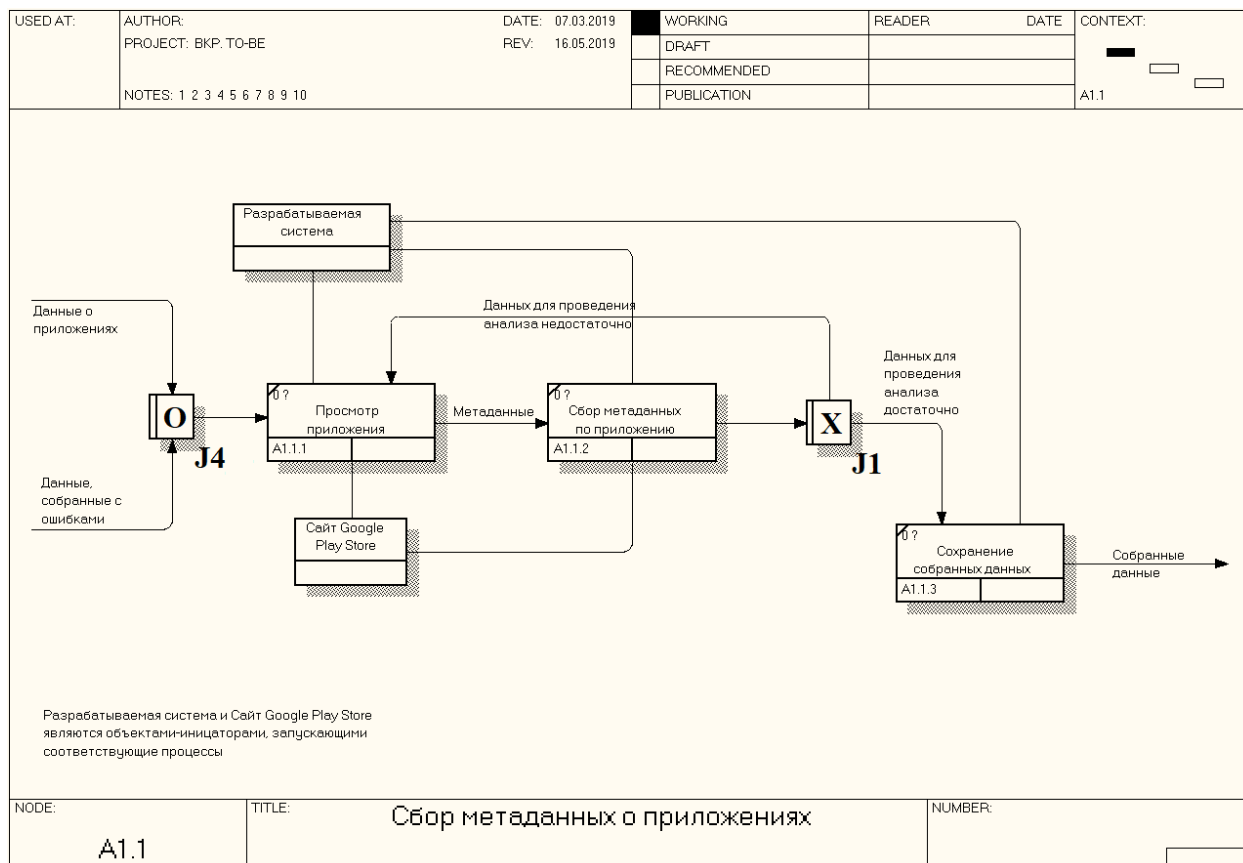


Рисунок 26 – Декомпозиция процесса «Сбор метаданных о приложениях»

На рисунке 27 представлена декомпозиция процесса «обработка и визуализация собранных данных», построенная по методологии IDEF3, позволяющая отобразить цикличность процесса обработки данных, а также изменение объектов-инициаторов.

Как видно объект «Аналитик», как и в предыдущем процессе, был заменен объектом «Разрабатываемая система», а объект «MS Excel» был удалён, так как все данные предоставляются единым хранилищем разрабатываемой системы.

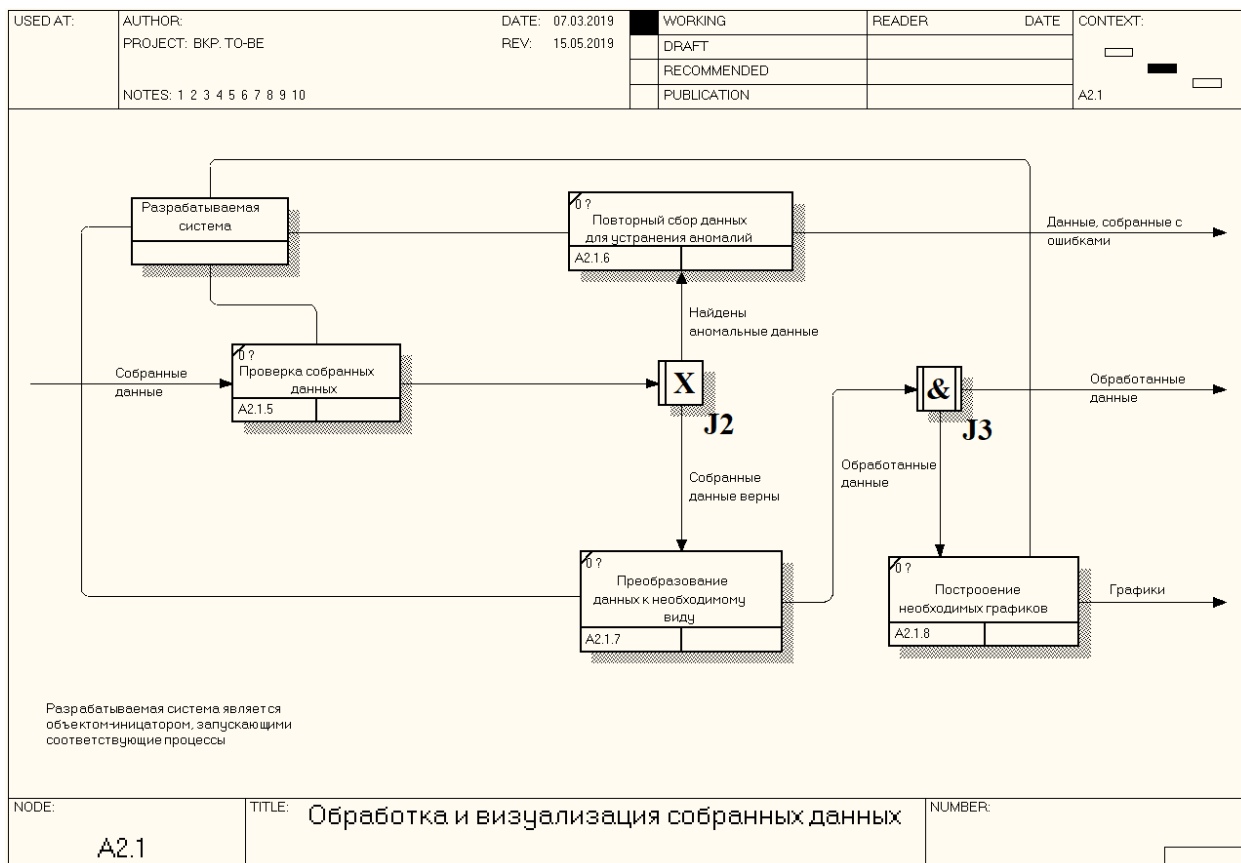


Рисунок 27 – Декомпозиция процесса «Обработка и визуализация собранных данных»

Как показано на рисунках, бизнес-процессы не понесли значительных технологических изменений, однако на рисунке 25 видно, что теперь аналитик участвует только в процессе анализа и принятия решений, а процессы сбора, обработки и визуализации данных теперь выполняются исключительно разрабатываемой системой. Таким образом наиболее трудоёмкие, а, следовательно, и затратные процессы, будут выполняться системой, что снизит нагрузку на аналитика.

По рисункам 26 и 27 можно определить основные требования к системе:

1. Разрабатываемая система должна выполнять сбор метаданных о приложениях из коллекций: «топ бесплатных», «топ платных», «бестселлеры» и «набирающие популярность»;

2. Разрабатываемая система должна выполнять обработку собранных данных (поиск и устранение аномалий, преобразование данных к форме, удобной для их дальнейшего использования);

3. Разрабатываемая система должна визуализировать обработанные данные и предоставить аналитику широкий набор графиков, которые помогут ему в проведении анализа и принятии решений;

4. Разрабатываемая система должна иметь простой и понятный пользователю интерфейс;

5. Разрабатываемая система должна быть легкодоступной для аналитика.

2.3 Проектирование системы анализа

Для определения требований к хранилищу данных была построена DFD-модель процесса сбора, обработки и визуализации данных магазина Google Play Store. Накопители данной диаграммы отображают отдельные объекты единой базы данных системы, с каждым из которых взаимодействует тот или иной процесс.

Эта модель позволит представить принцип взаимодействия процессов, выполняемых системой, с базой данных, а также изучить основные объекты, поставляющие

данные в систему, или получающие данные из неё. То есть DFD диаграмма позволит наглядно изучить с какими объектами базы данных будет взаимодействовать тот или иной процесс. Контекстная же диаграмма в частности позволит определить, какие лица, взаимодействующие с системой, какую информацию будут получать.

На рисунке 28 приведена контекстная диаграмма модели, которая описывает взаимодействие с системой внешних объектов: разработчика, аналитика и самого магазина Google Play Store. Основным пользователем системы будет являться аналитик, который и направляет запросы на получение данных (открытие WEB-страниц) к системе. Разработчик вносит начальные данные, которые система будет использовать в дальнейшем. Google Play Store же является основным источником данных для системы.

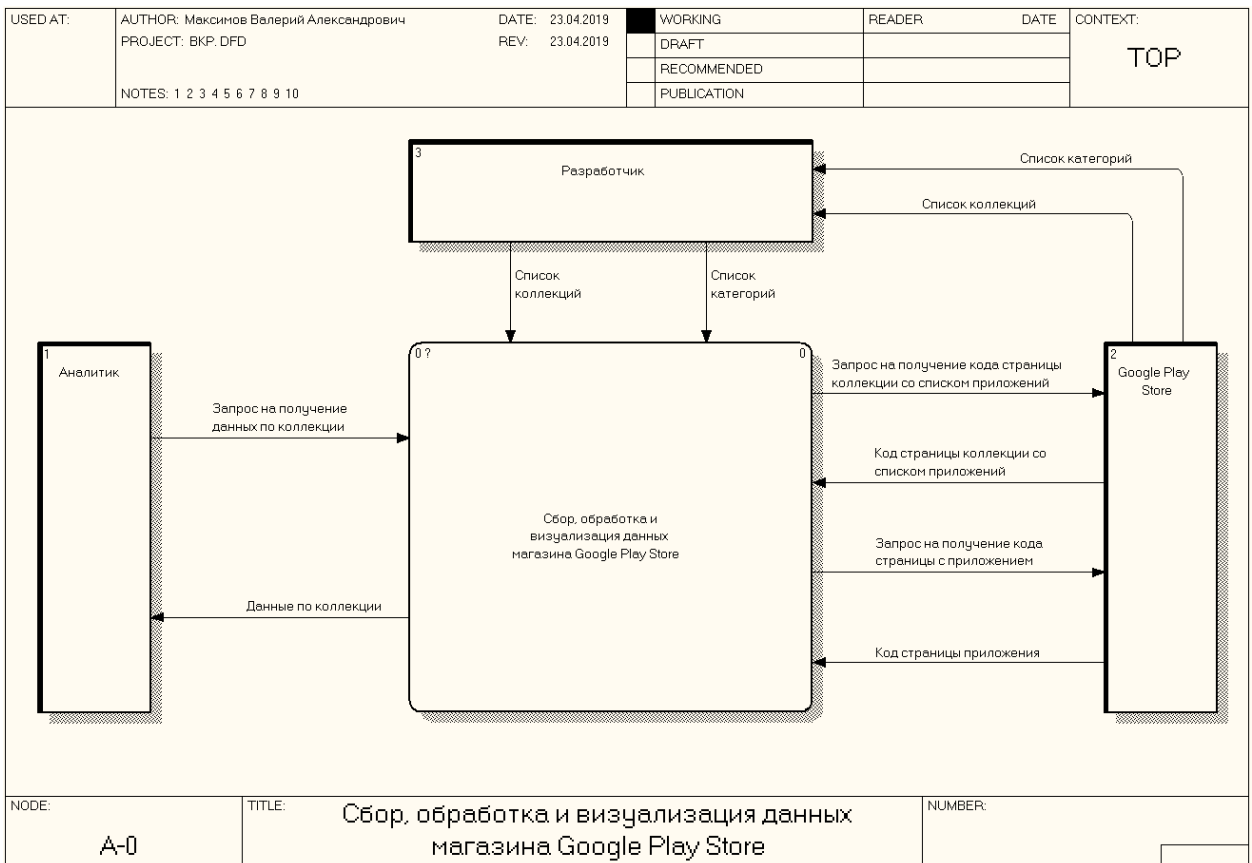


Рисунок 28 – Контекстная диаграмма DFD-модели процесса сбора обработки и визуализации данных магазина Google Play Store

На рисунке 29 приведена декомпозиция процесса, описывающая внутреннее перемещение данных между подпроцессами. Так запросы, поступающие от аналитика, обрабатываются рядом процессов, часть из которых получает данные непосредственно из самого магазина и записывает эти данные во внутреннее хранилище системы, представленное четырьмя основными накопителями, соответствующие таблицам единого хранилища, в которых будут храниться данные приложений каждой коллекции.

Другая часть процессов использует данные из внутреннего хранилища и использует их для формирования WEB-страниц, которые и предоставляются аналитику.

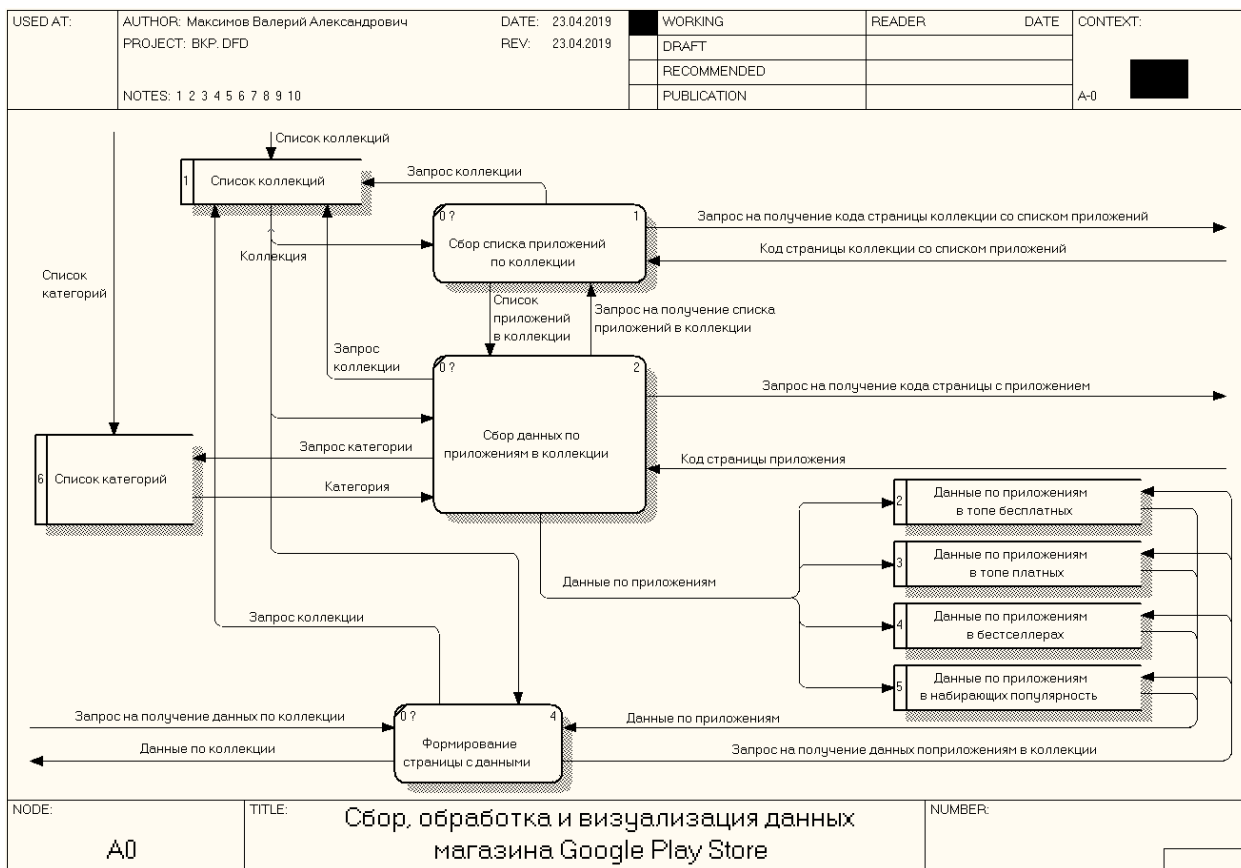


Рисунок 29 – Декомпозиция DFD-модели процесса сбора обработки и визуализации данных магазина Google Play Store

2.4 Архитектура системы и ее программное обеспечение

При разработке мобильного приложения использовались следующие программные средства, описанные ниже.

PyCharm - интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе

IntelliJ IDEA. Данная среда разработки доступна для Windows, OS X и Linux.

PhpStorm – коммерческая кросс-платформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

Anaconda – это платформа для научных исследований, основанная на языке программирования Python. Основная цель этого пакета – предоставить организациям возможность успешно защищать, интерпретировать, масштабировать и хранить данные, которые имеют решающее значение для повседневной работы.

phpMyAdmin – веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружелюбный интерфейс.

На рисунке 30 приведена диаграмма пакетов информационной системы, отображающая структуру проекта системы, а также принципы взаимодействия между модулями.

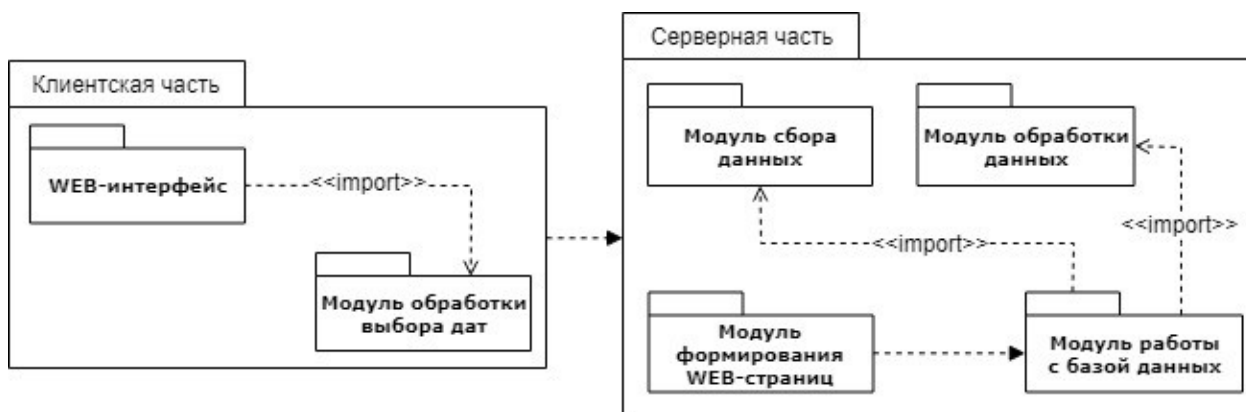


Рисунок 30 – Диаграмма пакетов

Как видно из рисунка 30, было выделено два основных пакета: клиентская и серверная части. Клиентская часть обрабатывает запросы пользователя, связанных с выбором коллекций, данные по которым необходимо выводить, а также с выбором периода и скачиванием данных за этот период в формате электронных таблиц MS Excel. Серверная часть выполняет внутренние обработки по сбору и обработке данных, а также по формированию страниц, представляемых пользователю.

В пакетах «Клиентская часть» и «Серверная часть» также были выделены пакеты, сформированные по основным функциональным направлениям системы.

Клиентская часть:

- WEB-интерфейс (соответствует фрагменту кода приложения А, пункта «Шаблон HTML-страницы коллекции»)
- пакет, отвечающий за обработку запросов пользователя со стороны клиента, в том числе и запросов, связанных со скачиванием данных за определенный период, импортируя функции модуля обработки выбора дат;

- модуль обработки выбора дат – пакет, занимающийся обработкой полей формы, предназначенной для выбора периода, за который клиенту необходимо получить данные.

Серверная часть:

- модуль сбора данных (соответствует фрагменту кода приложения А, пункта «Модуль считывания данных с Google Play Store») – пакет, функции которого выполняют подключение к Google Play Store и сбор данных по приложениям с него;

- модуль обработки данных (соответствует фрагменту кода приложения А, пункта «Фрагмент модуля обработки данных») – пакет, функции которого выполняют поиск аномальных значений в данных, а также форматируют данные, приводя их к виду, удобному для дальнейшего использования;

- модуль работы с базой данных – пакет, выполняющий подключение к базе данных и обрабатывающий процессы внесения данных в базу и получения данных из неё с использованием функций пакетов «Модуль сбора данных» и «Модуль обработки данных», а также выполняющий процесс формирования файла электронных таблиц MS Excel, который будет использован модулем формирования WEB-страниц для последующей передачи файла пользователю;

- модуль формирования WEB-страниц (соответствует фрагменту кода приложения А, пункта «Фрагмент модуля формирования графиков») – пакет, формирующий страницы, запрошенные пользователем, используя при этом данные, полученные из базы при помощи модуля работы с базой данных.

2.5 Информационное обеспечение задачи анализа

Даталогическая модель – модель данных, ориентированная на выбранный тип СУБД и представляет собой отображение логических связей между элементами данных, то есть модель хранилища системы, реализованного на основе реляционной базы данных и СУБД MySQL. Физическая модель системы представлена на рисунке 32 ниже.

Инфологическая модель базы данных – модель, представляющая собой описание объектов (сущностей), с набором атрибутов и связей между ними, выявленных в процессе исследования как входных, так и выходных данных [8, 9]. Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Инфологическая модель системы, описанная в нотации П. Чена, приведена на рисунке 31.

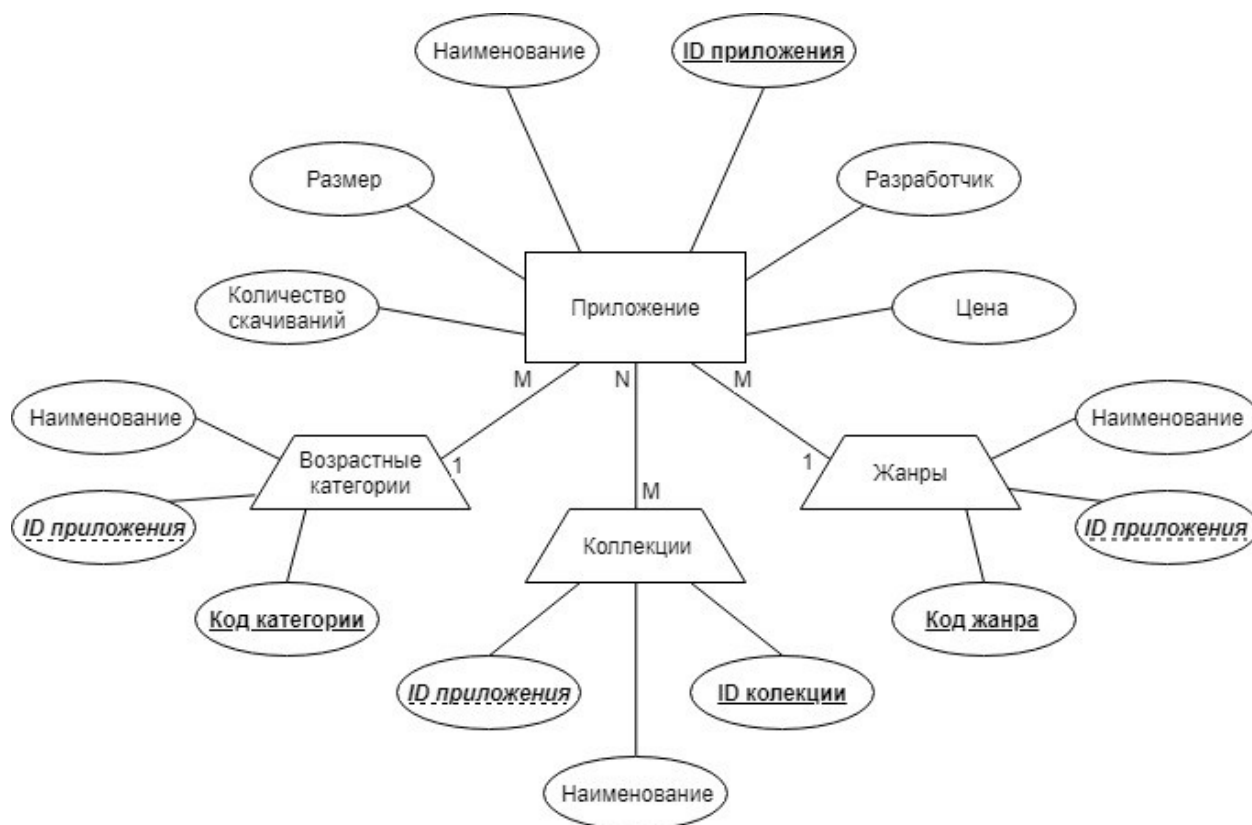


Рисунок 31 – Инфологическая модель разрабатываемой системы

Даталогическая модель – это модель данных, ориентированная на выбранный тип СУБД, которая представляет собой отражение логических связей между элементами данных [7, 9]. Она является представлением объектов (таблиц, атрибутов) в виде, который применим к выбранному типу СУБД.

Даталогическая модель, отражающая структуру базы данных, основанную на реляционной модели данных, разрабатываемой системы, приведена на рисунке 32.

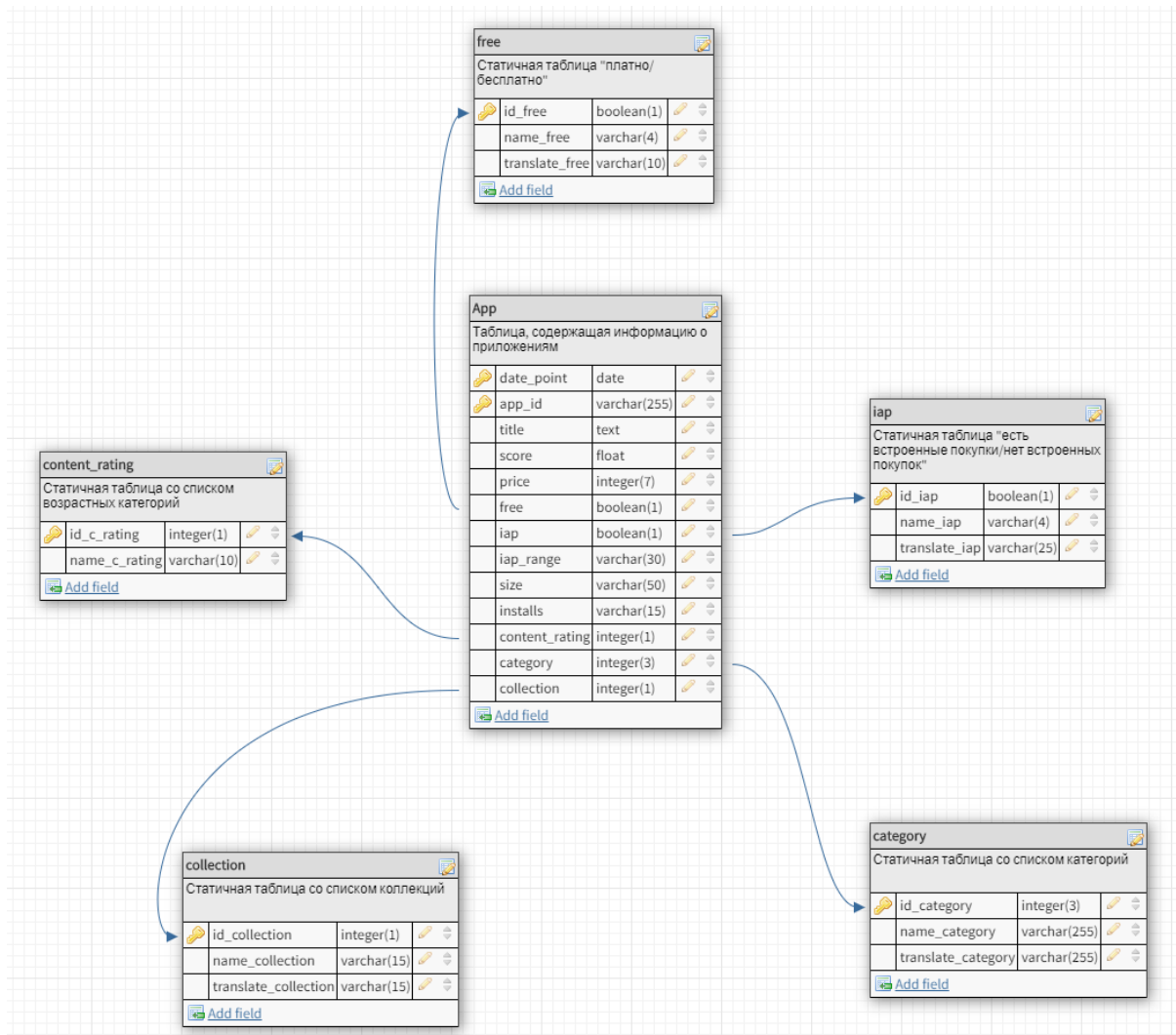


Рисунок 32 – Даталогическая модель разрабатываемой системы

Более подробное описание каждой сущности и атрибутов даталогической модели приведено в таблицах 1 – 6 приложения А.

На основе даталогической была разработана база данных, физическая модель которой представлена на рисунке 33.

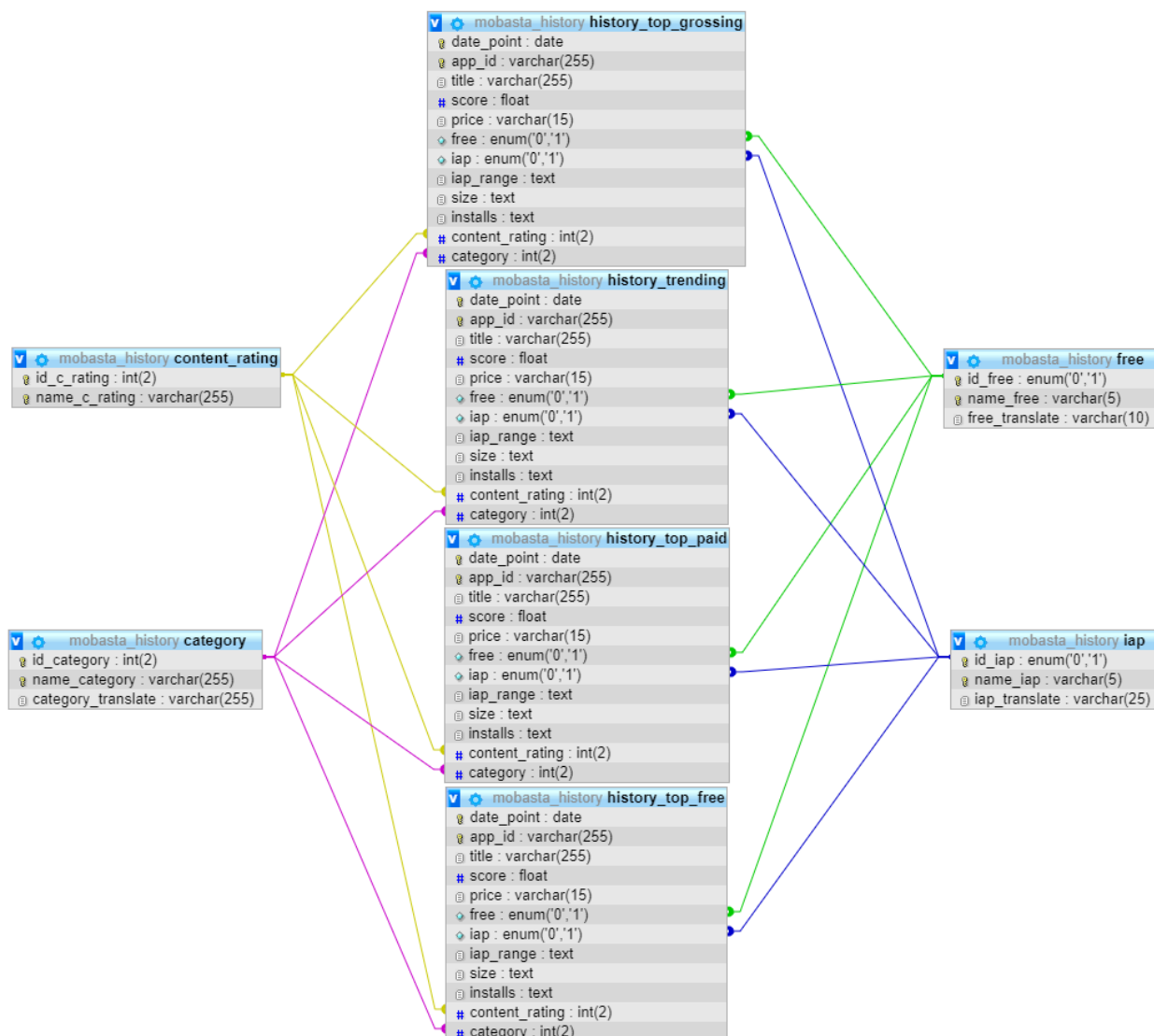


Рисунок 33 – Физическая модель хранилища данных

Как видно, в отличие от даталогической модели, при реализации базы данных было решено разделить сущность «App» на 4 таблицы в соответствии с коллекциями, в которых они собираются. Такой подход позволяет, жертвуя занимаемой памятью, значительно увеличить скорость отклика на запросы пользователя при загрузке WEB-страницы. Причиной этого является то, что при отправке пользователем запроса на агрегацию и визуализацию данных

по одной из коллекций вычислительная машина не будет затрачивать время на перебор и сравнение многочисленных записей, имеющих в базе, а выведет все записи из соответствующей запросу таблицы. Эмпирическим методом, с использованием монитора ресурсов, встроенного в Яндекс.Браузер, была выявлена разница между временем загрузки WEB-страницы «Бестселлеры» при использовании схемы базы данных с одной таблицей и с четырьмя таблицами приложений: при одной общей таблице загрузка страницы занимает приблизительно 26 секунд, а при четырёх таблицах – приблизительно 15 секунд.

Поля даталогической модели имеют полностью соответствуют полям физической модели, включая наименование и типы данных.

На рисунке 34 приведён пример хранимой информации из таблицы «history_top_grossing».

2019-03-14	com.klockigame.klocki	klocki	4.6	RUB 15.00	0	0	NULL	37M	100.000+	1	32
2019-03-24	com.zigzagame.evertale	Evertale	4.3	RUB 15.00	0	1	RUB 65.00 - RUB 1,190.00	74M	1.000+	2	34
2019-03-14	com.zitga.ninja.stickman.legends.shadow.wars	Stickman Legends: Shadow War Offline Fighting Game	4.6	RUB 15.00	0	1	RUB 15.00 - RUB 1499.00	98M	5.000.000+	3	23
2019-03-02	com.codebrewgames.pocketcitygame	Pocket City	4.6	RUB 149.00	0	0	NULL	23M	100.000+	1	35
2019-03-03	com.ndiviapps.qrcode.pro	QR Code Reader (Pro)	4.7	RUB 139.00	0	0	NULL	4.4M	10.000+	1	55
2019-03-04	com.ndiviapps.qrcode.pro	QR Code Reader (Pro)	4.7	RUB 139.00	0	0	NULL	4.4M	10.000+	1	55
2019-03-20	com.lgdstudio.ilivegoldedition	i Live - Gold Edition	4.2	RUB 139.00	0	1	RUB 69.00 - RUB 1490.00	25M	100.000+	1	35
2019-03-22	com.farm.hell.clickers.pro	Hell Clicker PRO	4.4	RUB 129.00	0	1	RUB 15.00 - RUB 3790.00	80M	50.000+	3	35
2019-03-22	tv.rd6	TV RD - Dominican Television	4.5	0	1	0	NULL	5.8M	100.000+	1	12
2019-03-16	air.EAh	Beauty Games 📱📱	4.7	0	1	0	NULL	67M	100.000+	1	35
2019-03-17	air.EAh	Beauty Games 📱📱	4.7	0	1	0	NULL	67M	100.000+	1	35
2019-03-18	air.EAh	Beauty Games 📱📱	4.7	0	1	0	NULL	67M	100.000+	1	35
2019-03-19	air.EAh	Beauty Games 📱📱	4.7	0	1	0	NULL	67M	100.000+	1	35
2019-03-22	air.EAh	Beauty Games 📱📱	4.6	0	1	0	NULL	67M	100.000+	1	35
2019-03-23	air.EAh	Beauty Games 📱📱	4.6	0	1	0	NULL	67M	100.000+	1	35
2019-03-06	com.HBO	HBO GO: Stream with TV Package	3.7	0	1	0	NULL	47M	10.000.000+	3	12
2019-03-07	com.HBO	HBO GO: Stream with TV Package	3.7	0	1	0	NULL	47M	10.000.000+	3	12
2019-03-08	com.HBO	HBO GO: Stream with TV Package	3.7	0	1	0	NULL	47M	10.000.000+	3	12
2019-03-23	com.aaf	Alliance of American Football	3.8	0	1	0	NULL	16M	100.000+	1	54
2019-03-24	com.aaf	Alliance of American Football	3.8	0	1	0	NULL	16M	100.000+	1	54
2019-03-03	com.amc	AMC Theatres	4.5	0	1	0	NULL	78M	1.000.000+	1	12
2019-03-04	com.amc	AMC Theatres	4.5	0	1	0	NULL	78M	1.000.000+	1	12
2019-03-08	com.amc	AMC Theatres	4.5	0	1	0	NULL	78M	1.000.000+	1	12
2019-03-10	com.amc	AMC Theatres	4.5	0	1	0	NULL	78M	1.000.000+	1	12

Рисунок 34 – Пример хранения информации в базе

2.6 Технологическое обеспечение задачи сбора и обработки информации о мобильных приложениях

Так как данные по Android-приложениям возможно получить лишь непосредственно с сайта Google Play Store, было решено реализовать сбор этих данных методом WEB-scraping'a, который подразумевает считывание исходного кода страницы и вычленение из него необходимой информации [4, 5].

Общий принцип, соответствующий фрагменту кода, приведенному в приложении А пункте «Модуль считывания данных с Google Play Store» работы системы приведен на диаграмме деятельности, представленной на рисунке 35.

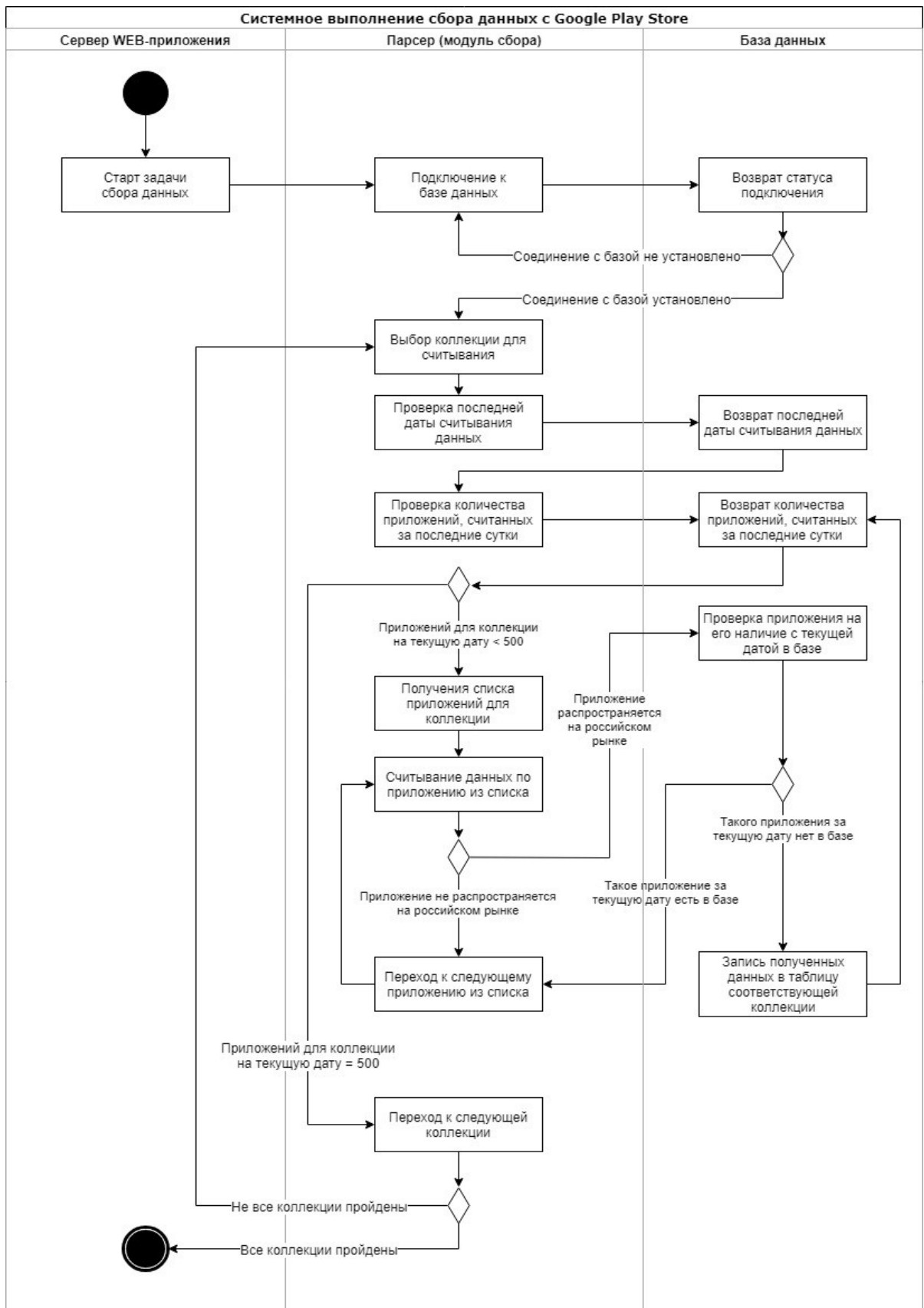


Рисунок 35 – Диаграмма деятельности, описывающая процесс сбора данных

Основной для любого WEB-сайта является модель его структуры, а именно «Карта сайта». В данной работе был построен вариант модели «Карта сайта» в форме графа, который показывает не только структуру сайта, но и порядок перехода между страницами сайта (рисунок 36).

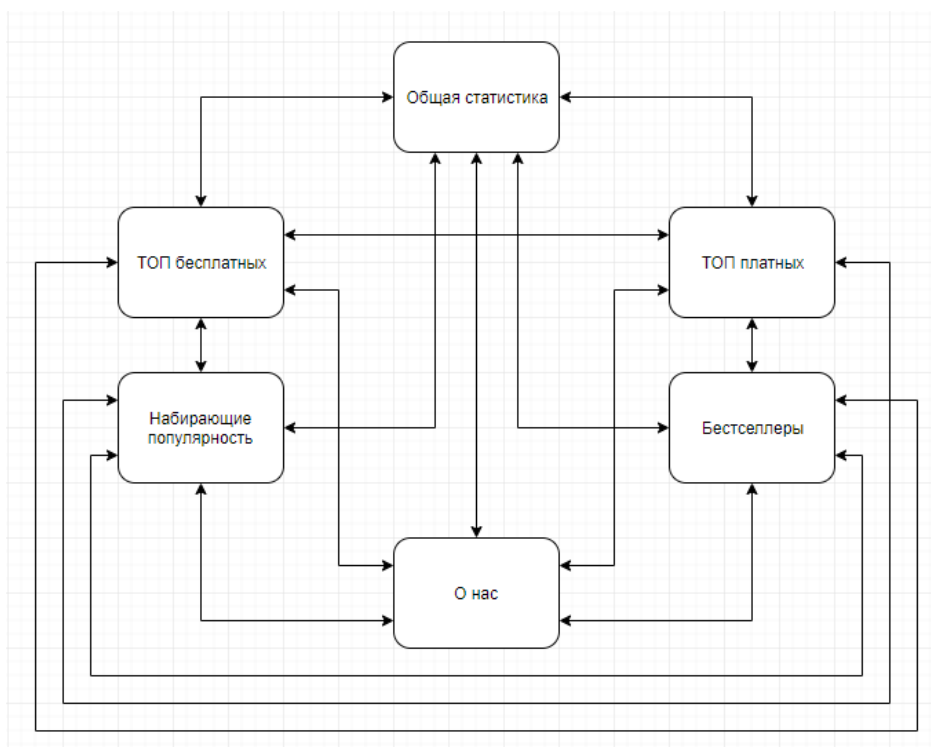


Рисунок 36 – Модель «Граф страниц»

Как видно из модели, приведенной на рисунке 35, пользователь должен иметь возможность перейти по гиперссылкам из любой страницы в любую страницу.

2.7 Обоснование экономической эффективности проекта

Расчет был произведен по отраслевому стандарту ОСТ 4.071.030.

Для исследования экономической эффективности разрабатываемой системы был выбран метод

инвестиционного анализа, позволяющий оценить эффективность разрабатываемой системы на основе затрат.

Чтобы рассчитать показатели экономической эффективности, необходимо сравнить данные по затратам до внедрения информационной системы анализа деятельности предприятия и после внедрения.

Трудоемкость работ по внедрению ИС определяется с учетом срока окончания работ, объема выполняемых функций, выбранной среды программирования. Рассчитанная трудоемкость работ, согласно графика работ, представлена в таблице 5.

Т а б л и ц а 5 – Комплекс работ по разработке проекта

Содержание работ	Исполнители	Длительность	Загрузка	
			Дни	Проценты
Формирование системных требований				
Анализ предметной области	<i>Научный руководитель</i>	5	1,5	30%
	<i>Студент</i>		4	80%
Анализ существующих решений	<i>Научный руководитель</i>	3	0,45	15%
	<i>Студент</i>		0,6	20%
Определение и анализ требований к системе	<i>Научный руководитель</i>	19	9,5	50%
	<i>Студент</i>		19	100%
Итого по этапу «Формирование системных требований»	<i>Научный руководитель</i>	27	11,45	
	<i>Студент</i>		23,6	
Проектирование системы				
Структурное моделирование	<i>Научный руководитель</i>	9	3,6	40%
	<i>Преподаватель</i>		1,35	15%
	<i>Студент</i>		9	100%
Функциональн	<i>Преподаватель</i>	12	2,4	20%

ое моделирование	<i>Студент</i>		9,6	80%
Разработка макетов страниц	<i>Научный руководитель</i>	5	0,75	15%
	<i>Студент</i>		1	20%
Проектирование базы данных	<i>Студент</i>	3	3	100%
Итого по этапу «Проектирование системы»	<i>Научный руководитель</i>	29	4,35	
	<i>Преподаватель</i>		3,75	
	<i>Студент</i>		22,6	
Реализация системы				
Реализация механизма получения данных	<i>Студент</i>	53	37,1	70%
Реализация алгоритмов обработки данных	<i>Студент</i>	18	2,7	15%

Окончание таблицы 5

Содержание работ	Исполнители	Длительность	Загрузка	
			Дни	Проценты
Реализация базы данных	<i>Студент</i>	8	0,4	5%
Реализация серверной части системы	<i>Студент</i>	28	28	100%
Верстка пользовательской части системы	<i>Студент</i>	21	2,1	10%
Подключение базы данных к системе	<i>Студент</i>	2	2	100%
Тестирование системы	<i>Студент</i>	5	5	100%
Доработка системы	<i>Студент</i>	2	2	100%
Итого по этапу «Реализация системы»	<i>Студент</i>	137	79,3	
Итого по	<i>Студент</i>	193	125,5	

проекту	Научный руководитель		15,8
	Преподаватель		3,75

Капитальные вложения, связанные с автоматизацией обработки информации рассчитываются по формуле $K=K_n+K_p$, где K_n – капитальные вложения на проектирование (руб.), а K_p – капитальные вложения на реализацию проекта (руб).

Суммарные затраты на проектирование системы и ее разработку, и отладку на компьютере определяются по формуле 2:

$$K_n = \sum_{i=1}^m Z_{oi} \cdot t_i \quad (2)$$

где m – количество работников, участвующих в разработке проекта;

Z_{oi} – затраты на основную заработную плату работника i -й категории, руб. (таблица 6);

Т а б л и ц а 6 – Расчет основной зарплаты работников

Должность	Должностной оклад, руб.	Средняя дневная	Затраты времени на разработку t_i , человеко-	ОЗП Z_{oi} , руб.
Научный руководитель	17950	854,76	15,8	13505,24
Преподаватель	35900	1709,52	3,75	6410,71
Итого Z_o				19915,9

W_d – коэффициент, учитывающий дополнительную заработную плату в долях к основной заработной плате ($W_d = 0,4$);

W_c – коэффициент, учитывающий отчисления на социальные нужды, в долях к сумме основной и

дополнительной заработной платы разработчиков ($W_c = 0,262$);

W_n - коэффициент, учитывающий накладные расходы организации, в долях к основной заработной плате разработчиков ($W_n = 0,6$);

S_m - затраты на материалы (таблица 7);

Таблица 7 - Расчет затрат на материалы

Материалы	Единица измерения	Требуемое количество	Цена за единицу, руб.	Сумма, руб.
Тетрадь общая	шт.	2	50	100
Ручка гелевая	шт.	3	30	90
Бумага офисная	пачка	1	350	350
Итого				540

M_b - затраты на использование машинного времени (таблица 8) рассчитываются по формуле 3.

Таблица 8 - Расчет затрат на использование машинного времени

Затраты на использование машинного времени M_b	Машинное время на проект t_{mb}	Стоимость одного часа $S_{мч}$	Коэф. мультипрограммности K_m
506,50	953,87	0,531	1

$$M_b = t_{mb} S_{мч} K_m, \quad (3)$$

где t_{mb} - машинное время компьютера, необходимое для разработки системы;

$S_{мч}$ - стоимость 1 часа машинного времени (взята средняя стоимость в бюджетных учреждениях в г. Владивосток);

K_m – коэффициент мультипрограммности (из расчета, что машина будет использоваться исключительно в целях работы над проектом, коэффициент будет равен 1).

Список затрат на разработку приведен в таблице 9.

Таблица 9 – Общие затраты на разработку

Статьи затрат	Сумма, руб.
Основная заработная плата Z_o	19915,95
Дополнительная зарплата W_d	7966,38
Отчисления на социальные нужды	7305,17
Затраты на материалы	540,00
Затраты на машинное время	506,50
Накладные расходы организации	11949,57
ИТОГО	48183,58

Капитальные вложения на реализацию проекта рассчитываются по формуле 4:

$$K_p = K_o + K_{дд} + K_{пп} + K_{св} + K_{иб} + K_{пк}, \quad (4)$$

где K_o – затраты на основное и вспомогательное оборудование, руб.;

$K_{зд}$ – затраты на строительство, реконструкцию здания и помещений, руб.;

$K_{пп}$ – затраты на приобретение типовых разработок, пакетов, руб.;

$K_{св}$ – затраты на прокладку линий связи, руб.;

$K_{иб}$ – затраты на создание информационной базы, руб.;

$K_{пк}$ – затраты на подготовку и переподготовку кадров, руб.

В связи с тем, что для внедрения системы, рассматриваемой в данном проекте, не было затрат, связанных с прокладкой линии связи, затрат на основное и вспомогательное оборудование, затрат на реконструкцию и строительство зданий, то данные затраты для внедрения

системы не учитывают. Также не принимаются в расчет затраты по подготовке и переподготовке кадров, затраты на создание информационной базы и затраты на приобретение типовых разработок. Однако учитывая то, что для функционирования системы, разрабатываемой в форме сайта, требуется оплатить хостинг.

Затраты на хостинг будут отнесены к капитальным вложениям на реализацию проекта и будут браться как средняя стоимость хостинга за месяц.

В итоге суммарные затраты на разработку проекта составят: $K = K_n + K_p = 48183,58 + 150 = 48333,58$ руб.

Так как разрабатываемая система и системы-аналоги позиционируются как сторонние WEB-сайты, то следует учесть затраты на оплату хостинга. Так как все хостинги, как и аналоги разрабатываемой системы, имеют ежемесячную оплату, то расчет их стоимости будет произведен за год (т.е. 12 месяцев).

Таким образом эксплуатационные затраты будут равны $Z_3 = Z_m = 150 * 12 = 1800$ руб. из расчета, что средняя сумма ежемесячного платежа за услуги хостинга равна 150 руб.

Затраты же, связанные с использованием аналогов, получаются из расчета, что каждый аналог требует ежемесячной оплаты, и исчисляются за год. То есть затраты в год на использование аналогов, определенных в первой лабораторной работе, будут составлять $100\$ * 12 = 1200\$ \approx 79000$ рублей для сайта-аналога «AppAnnie» и $5500 * 12 = 66000$ рублей для сайта-аналога «APPFOLLOW».

Для расчета показателя экономического эффекта требуется рассчитать приведенные затраты на единицу

работ, выполняемых с помощью разрабатываемой системы и системы-аналога.

Учитывая, что системы являются сторонними WEB-сервисами, то затрат на их внедрение не предполагается, однако каждая система имеет себестоимость, в данном случае, равную стоимости хостинга в разрабатываемой системе и ежемесячный платёж для системы-аналога. Расчет будет произведен за год (т.е. 12 месяцев).

Таким образом показатель экономического эффекта будет равен:

$$\mathcal{E} = (Z_1 * A_k - Z_2) * N = (79000 * 1,45 - 1800) * 1 = 112750$$

После определения годового экономического эффекта необходимо рассчитать срок окупаемости затрат на разработку продукта:

$$T_{ок} = \frac{K}{\mathcal{E}} = \frac{48333,58}{112750} \approx 0,4$$

Затем был рассчитан фактический коэффициент экономической эффективности разработки (E_{ϕ}) и сопоставлен с нормативным значением коэффициента эффективности капитальных вложений $E_n = 0,33$:

$$E_{\phi} = \frac{1}{T_{ок}} = \frac{1}{0,4} = 2,5$$

Фактический коэффициент экономической эффективности разработки получился больше, чем нормативный, поэтому разработка и внедрение разрабатываемого продукта является эффективной.

Таким образом, в ходе проделанной работы найдены все необходимые данные, доказывающие целесообразность и эффективность данной разработки, которые приведены в таблице 10.

Таблица 10 - Результаты экономического обоснования проекта

Характеристика проекта	Значение
Затраты на разработку и внедрение проекта, руб.	48333,58
Общие эксплуатационные затраты, руб.	1800
Экономический эффект, руб.	112750
Коэффициент экономической эффективности	2,5
Срок окупаемости, лет	0,4

2.8 Описание контрольного примера реализации проекта

Так как разрабатываемая система представляется в виде сайта, в качестве контрольного примера будут приведены страницы разработанного сайта. На рисунках 37-38 приведена страница, отображающая набор графиков, построенных на основе данных приложений, попавших в коллекцию «Бестселлеры».

Статистика по бестселлерам

01.06.2019 01.07.2019 [Скачать данные](#)

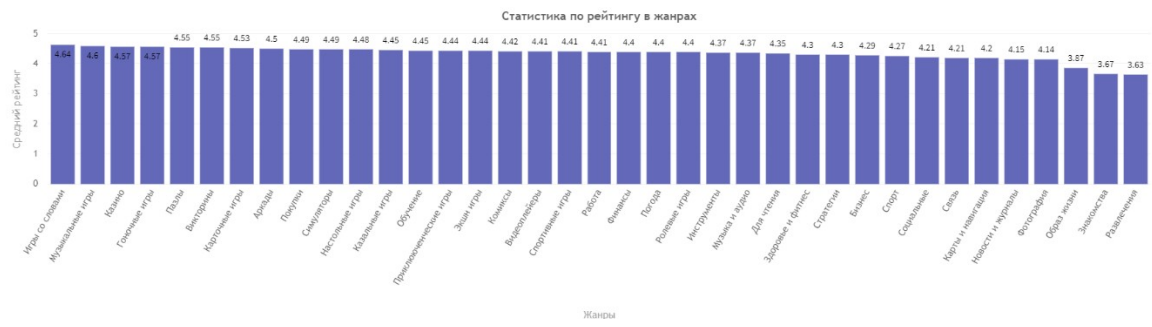
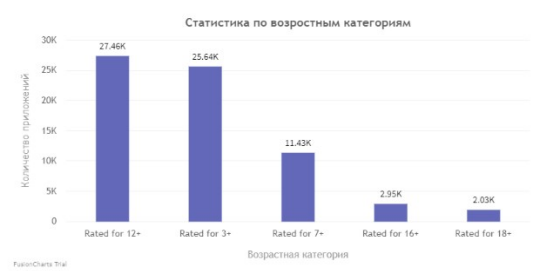
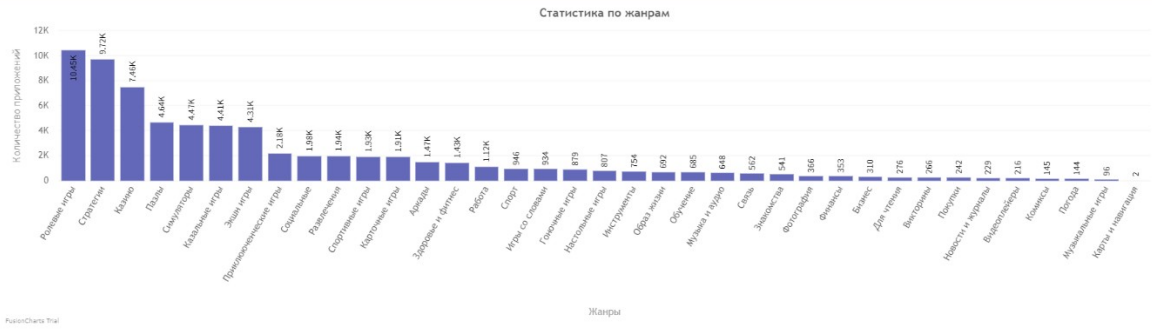


Рисунок 37 – Первая часть страницы системы, отображающая статистику по коллекции «Бестселлеры»

На рисунке 37 приведены графики, отражающие общую информацию о количестве приложений каждого жанра, о количестве приложений, имеющих и не имеющих встроенные покупки, средние минимальную и максимальную цены встроенных покупок, о количестве приложений в каждой возрастной категории и о средних пользовательских оценках для каждого жанра.

Анализируя информацию, приведенную на странице, можно заметить, что приложения в категориях «Знакомства» и «Развлечения» имеют самые низкие средние показатели рейтингов ~3,6 баллов.

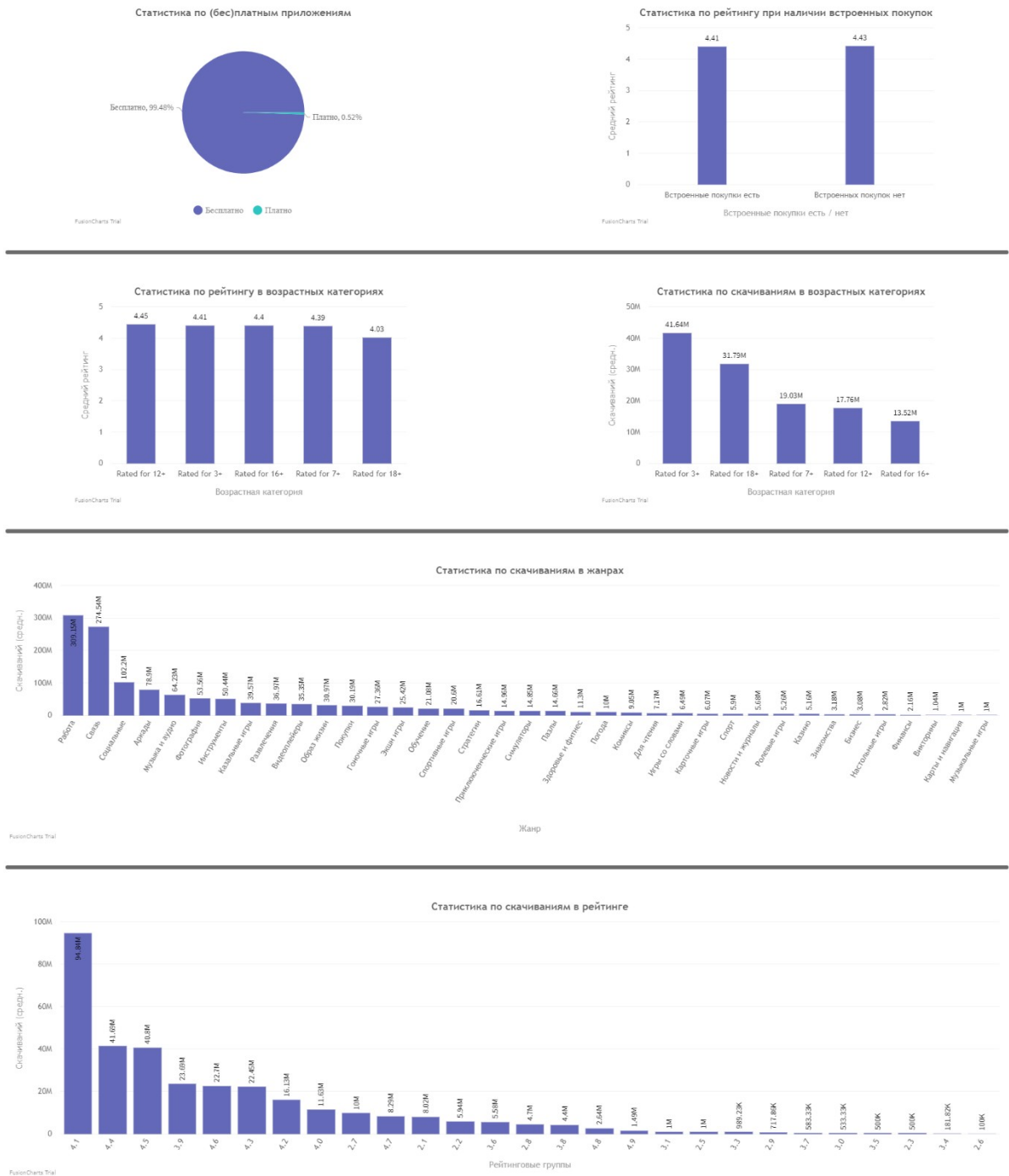


Рисунок 38 – Первая часть страницы системы, отображающая статистику по коллекции «Бестселлеры»

На рисунке 38 приведены графики, отражающие информацию о количестве платных и бесплатных

приложений, о средних пользовательских оценках приложений, имеющих и не имеющих встроенные покупки, о средних пользовательских оценках в каждой возрастной категории, о среднем количестве скачиваний в каждой возрастной категории, о количестве скачиваний в каждой рейтинговой группе.

По этим графикам можно заметить, что группа развлечений находится не на последнем месте по числу скачиваний. По этой информации разработчик мобильных приложений может сделать вывод о том, что пользователи, в силу широкой популярности этой категории, возможно, будут искать альтернативы с более высоким рейтингом, а значит, разработка такого приложения имеет шанс попасть в поле зрения пользователя [8, 9].

Аналогичными методами можно произвести широкий спектр аналитических расчётов и построения гипотез по различным показателям: категории пользователей (возрастной рейтинг), влияние наличия встроенных покупок на количество скачиваний и рейтинги пользователей, влияние цен приложений на скачивания и рейтинги и так далее.

Подобный подход к анализу данных позволит разработчику широко оценить текущее состояние рынка «лучших» Android-приложений и принять различные решения, которые повлияют на качество, направленность и формат разрабатываемого приложения.

Помимо формирования стандартных графиков система позволяет пользователю получить выгрузку данных за определенный период. Для этого пользователю требуется

выбрать необходимый период, находясь на странице выгружаемой коллекции и нажать на кнопку «Скачать данные» в верхней части страницы, приведенной на рисунке 39.

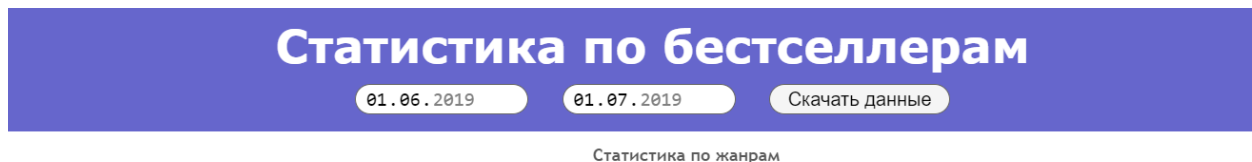


Рисунок 39 – Поле выбора дат и загрузки необходимых данных за выбранный период

Результат сформированной выборки приведен в приложении В на рисунке В.1.

Заключение

В ходе выполнения выпускной квалификационной работы были закреплены и углублены знания в области информационных систем и экономики путем самостоятельного решения реальных производственно-хозяйственных, экономических и управленческих проблем, полученные в ходе обучения.

Рассмотрено состояния существующей предметной области («КАК ЕСТЬ») процесса проведения анализа, характеристики объекта и аппарата управления.

Выявлены проблемы и недостатки в процессе проведения анализа рынка мобильных приложений. Также был представлен и обоснован путь устранения выявленных недостатков. Произведено описание входной и выходной информации. Определены цели автоматизации и произведено описание решений, принятых по всей вертикали проектирования и продемонстрирован контрольный пример.

При выполнении данной выпускной квалификационной работы были решены следующие задачи:

- произведен анализ предметной области отдела маркетинга компаний, занимающихся разработкой мобильных приложений;

- выявлены недостатки в процессе анализа и предложены решения по автоматизации процессов;

- была спроектирована информационная система сбора, обработки и визуализации информации по мобильным приложениям;

- была реализована спроектированная информационная система.

В дальнейшем планируется усовершенствование интерфейса приложения, а также добавление функций по выделению негативных отзывов пользователей, по выбору временного интервала, для которого была выведена статистика и т.д.

Список литературы

Нормативно-правовые акты

1. Отраслевой стандарт «Автоматизированная система управления предприятием. Создание системы» // ОСТ 4.071.030

Учебники и учебные пособия

1. Бабич А.В. Введение в UML. Учебное пособие/ А.В. Бабич — М.: НОУ ИНТУИТ, 2016. - 209 с.

2. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining - СПб.: БХВ-П., 2007 - 331 с.

3. Барсегян А.А., Куприянов М.С. Технологии анализа данных. Data Mining, Visual Mining, Text Mining, OLAP. - СПб.: БХВ-П., 2007 - 384 с.

4. Бриггс, Джейсон Python для детей. Самоучитель по программированию / Джейсон Бриггс. - Москва: Огни, 2013. - 177 с.

5. Бэрри, Пол Изучаем программирование на Python / Пол Бэрри. - М.: Эксмо, 2016. - 332 с.

6. Васильев, А. Н. Python на примерах. Практический курс по программированию / А.Н. Васильев. - М.: Наука и техника, 2016. - 432 с.

7. Васильев, Александр Николаевич Python на примерах. Практический курс по программированию. Руководство. - М.: Наука и техника, 2017. - 752 с.

8. Гуриков, С.Р. Основы алгоритмизации и программирования на Python / С.Р. Гуриков. - М.: Форум, 2018. - 991 с.

9. Гуриков, С.Р. Основы алгоритмизации и

программирования на Python. Учебное пособие. Гриф МО РФ.
- М.: Инфра-М, Форум, 2018. - 707 с.

10. Златопольский, Д. М. Основы программирования на языке Python / Д.М. Златопольский. - М.: ДМК Пресс, 2017. - 277 с.

11. Каюмова А.В. Визуальное моделирование систем в StarUML. Учебное пособие. Казань. – Казанский федеральный университет, 2013. –104с.

12. МакГрат, Майк Python. Программирование для начинающих / Майк МакГрат. - М.: Эксмо, 2013. - 727 с.

13. Маклаков, С.В. BPWIN и ERWIN. Case - средства разработки информационных систем. - М.: ДИАЛОГ-МИФИ, 2009. - с. 163.

14. Туманов В.Е. Проектирование реляционных хранилищ данных. Учебное пособие - М.: Диалог-МИФИ, 2007. - 333с.

Электронные ресурсы

1. A Statistical Analysis of the Apple App Store [Электронный источник] - URL: <https://blog.scottlogic.com/2014/03/20/app-store-analysis.html>

2. Analysis of Apps in the Google Play Store [Электронный источник] - URL: <https://nycdatascience.com/blog/student-works/web-scraping/analysis-of-apps-in-the-google-play-store/>

3. Google Play Store Apps [Электронный источник] - URL: <https://play.google.com/store/apps>

4. Number of Android apps on Google Play [Электронный источник] - URL: <https://www.appbrain.com/stats/number-of-android-apps>

5. SQLite Studio [Электронный ресурс] // «SQLiteStudio»

[сайт]. Режим доступа: <https://sqlitestudio.pl/index.rvt>

6. Малышева, Е.Н. Проектирование информационных систем. Объектно-ориентированная case-технология проектирования информационных систем [Электронный ресурс]: учебное пособие. — Электрон. дан. — Кемерово: КемГУКИ (Кемеровский государственный университет культуры и искусств), 2009. — 70 с.

7. Обзор [Электронный ресурс] // “FileZilla” [сайт]. Режим доступа: <https://filezilla-project.org/>

8. Руководство к своду знаний по управлению проектами, 5 издание [Электронный ресурс]. Режим доступа: http://pm-files.com/sites/default/files/file/C/C-1/C-1-1/pmbok_5th_2013_rus.pdf

9. Современные методы анализа данных [Электронный источник] — URL: <http://riep.ru/upload/iblock/031/031173bb40e099800b248497db44cb88.pdf>

10. Юридический отдел: обязанности и функции [Электронный ресурс] // “Генеральный директор” [сайт]. Режим доступа: <https://www.gd.ru/articles/8920-yuridicheskiy-otdel>

Приложение А

Описание сущностей и атрибутов даталогической модели

Сущность «App» – основная сущность, которая отражает представление самого приложения в базе данных.

Таблица 1 – Описание сущности «App»

Наименование атрибута	Тип атрибута	Описание атрибута
-----------------------	--------------	-------------------

date_point	date	Первичный ключ. Хранит дату записи приложения в базу данных
app_id	varchar (255)	Первичный ключ. Хранит текстовый ID приложения, присвоенный ему разработчиком при публикации приложения в магазине
title	text	Наименование приложения
score	float	Средняя пользовательская оценка
price	integer (7)	Цена приложения
free	boolean	Внешний ключ. Хранит одно из двух значений: «платное приложение» или «бесплатное приложение»
iap	boolean	Внешний ключ. Хранит одно из двух значений: «приложение имеет встроенные покупки» или «приложение не имеет встроенных покупок»
iap_range	varchar (30)	Диапазон стоимостей встроенных покупок
size	varchar (50)	Размер приложения
installs	varchar (15)	Число скачиваний приложения
content_rating	integer (1)	Внешний ключ. Хранит код возрастной категории приложения
category	integer (3)	Внешний ключ. Хранит код жанра приложения
collection	integer (1)	Внешний ключ. Хранит код коллекции приложения

Сущность «free» - вспомогательная сущность, которая хранит в себе варианты распространения приложения (платно или бесплатно).

Таблица 2 - Описание сущности «free»

Наименование атрибута	Тип атрибута	Описание атрибута
id_free	boolean	Первичный ключ. Хранит код значения

name_free	varchar (4)	Текстовое представление значения («True» / «False»)
translate_free	varchar (10)	Текстовое представление значения на русском языке

Сущность «iap» – вспомогательная сущность, которая хранит в себе варианты наличия встроенных покупок в приложении (есть или нет).

Т а б л и ц а 3 – Описание сущности «iap»

Наименование атрибута	Тип атрибута	Описание атрибута
id_iap	boolean	Первичный ключ. Хранит код значения
name_iap	varchar (4)	Текстовое представление значения («True» / «False»)
translate_iap	varchar (10)	Текстовое представление значения на русском языке

Сущность «category» – вспомогательная сущность, которая хранит в себе список всех жанров, которые могут иметь приложения.

Т а б л и ц а 4 – Описание сущности «category»

Наименование атрибута	Тип атрибута	Описание атрибута
id_category	integer (3)	Первичный ключ. Хранит код жанра
name_category	varchar (255)	Текстовое представление жанра на английском языке
translate_category	varchar (255)	Текстовое представление жанра на русском языке

Сущность «collection» – вспомогательная сущность, которая хранит в себе список коллекций, к которым могут относиться приложения.

Таблица 5 - Описание сущности «collection»

Наименование атрибута	Тип атрибута	Описание атрибута
id_collection	integer (1)	Первичный ключ. Хранит код коллекции
name_collection	varchar (15)	Текстовое представление коллекции на английском языке
translate_collection	varchar (15)	Текстовое представление коллекции на русском языке

Сущность «content_rating» - вспомогательная сущность, которая хранит в себе список возрастных категорий, которые могут иметь приложения.

Таблица 6 - Описание сущности «content_rating»

Наименование атрибута	Тип атрибута	Описание атрибута
id_c_rating	integer (1)	Первичный ключ. Хранит код возрастной категории
name_c_rating	varchar (10)	Текстовое представление возрастной категории

Приложение Б

Фрагменты кода разрабатываемой системы

В данном приложении приведены только основные фрагменты кода, так как полный листинг системы слишком большой.

Модуль считывания данных с Google Play Store

```
import play_scraper
import MySQLdb
import datetime
import time
from colorama import Fore
from threading import Thread, BoundedSemaphore as bs

start_time = time.time()

sem = bs() # Семафор, способный держать в себе 1 поток (default).
Аргумент 'value' хранит количество потоков в очереди.

def scrapy(table, collection, nums, page):
    while True:
        print(Fore.LIGHTBLUE_EX, page, "Repeat!")
        global apps_collection, row

        # Флаг на продолжение цикла for, если одно из приложений
        окажется битым и вылетит exception
        continue_flag = False

        # Флаг на прерывание цикла for, если стек приложений коллекции
        заполнен (500 из 500)
        full_stack_flag = False #
        try:
            apps_collection = play_scraper.collection(collection=collection,
            results=nums, page=page)
            print(page, "Получили коллекцию приложений!")

            conn = MySQLdb.connect('localhost', 'VProgramMist',
            '1998Vm0000', 'mobasta_history', charset='utf8mb4',
            use_unicode=False)
            cursor = conn.cursor()
            cursor.execute("SET CHARACTER SET utf8mb4;")
            cursor.execute("SET character_set_connection=utf8mb4;")
```

```

# Курсоры для получения внешних ключей
cursor_cat = conn.cursor()
cursor_rait = conn.cursor()
cursor_free = conn.cursor()
cursor_iap = conn.cursor()
cursor_key_checker = conn.cursor()
print(Fore.WHITE, page, "Связались с базой и получили
курсоры!")

for app in apps_collection:
    while True:
        try:
            sql_str = "SELECT * FROM " + table + " WHERE app_id =
%s AND date_point = %s;"
            cursor_key_checker.execute(sql_str,
                                      (app['app_id'], datetime.date.today()))

            if not cursor_key_checker.fetchone() is None:
                print(page, Fore.YELLOW + "Приложение ",
app['app_id'], " уже в базе!")
                break
            print(Fore.GREEN, page, "Совпадений нет!")
            try:
                row = play_scraper.details(
                    app['app_id']) # Ищем каждое приложение, для
получения детальных записей
            except:
                print(Fore.RED, page, "Приложение болезненное!
Переходим к другому...")
                continue_flag = True
                break

            print(Fore.WHITE, page, "Нашли приложение и собрали по
нему данные!", row['title'])

            if not row['iap_range'] is None:
                if not row['iap_range'][0].strip().startswith('RUB'):
                    row['iap_range'] = play_scraper.details(row['app_id'])
['iap_range'][0]
                    row['iap'] = 'True'
                else:
                    row['iap'] = 'False'

            sql_str = "SELECT id_category FROM category WHERE
name_category = %s;"
            cursor_cat.execute(sql_str,
                              [row['category'][0]])

            sql_str = "SELECT id_c_rating FROM content_rating WHERE
name_c_rating = %s;"
            cursor_rait.execute(sql_str,

```

```

        [row['content_rating'][0]])

    sql_str = "SELECT id_free FROM free WHERE name_free =
%s;"
    cursor_free.execute(sql_str,
                        [str(row['free'])])

    sql_str = "SELECT id_iap FROM iap WHERE name_iap =
%s;"
    cursor_iap.execute(sql_str,
                       [str(row['iap'])])

    print(Fore.CYAN, page, "Получили внешние ключи!")

    sem.acquire()
    # Через семафор позволяем одному потоку выполнить
запись строки в базу
    # и блокируем доступ к записи в базу строк другими
потоками

    sql_str = "SELECT * FROM " + table + " WHERE app_id =
%s AND date_point = %s;"
    cursor_key_checker.execute(sql_str,
                              (app['app_id'], datetime.date.today()))

    if not cursor_key_checker.fetchone() is None:
        print(page, Fore.YELLOW + "Приложение ",
app['app_id'], " уже в базе!")
        sem.release()
        break

    sql_str = """INSERT
    INTO """ + table + """(app_id, title, score, price, free, iap,
iap_range, size, installs, content_rating,
date_point, category)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
%s);"""

    # Блок проверки числа записей. Если число записей за
текущий день = 500, то прерываем
    # обновление. Требуется для получения более
достоверных данных при анализе данных.
    cursor.execute(
        "SELECT COUNT(*), DATEDIFF(MAX(date_point),
MIN(date_point)), MAX(date_point) FROM " + table)
    row_count = cursor.fetchone()
    if (row_count[0] - 500 * row_count[1]) >= 500 and
row_count[2] == datetime.date.today():
        continue_flag = False
        full_stack_flag = True
        sem.release()

```

```

        break

    cursor.execute(sql_str, (row['app_id'], row['title'],
row['score'],
                                row['price'], str(int(cursor_free.fetchone()
[0])),
                                str(int(cursor_iap.fetchone()[0])),
                                row['iap_range'], row["size"], row['installs'],
                                cursor_rait.fetchone()[0],
datetime.date.today(),
                                cursor_cat.fetchone()[0]))

    print(Fore.GREEN, page, "Записали строку в базу!")

    conn.commit() # Фиксируем все изменения в базе

    sem.release() # Через семафор освобождаем очередь
ПОТОКОВ ДЛЯ ЗАПИСИ

    print(Fore.GREEN, page, row['app_id'])

except:
    print(Fore.RED, page, "Проблемы с базой!")

    try:
        sem.release()
    except:
        pass

    conn.commit() # Фиксируем все изменения в базе

    if row['price'] is None:
        break

    cursor_cat.close()
    cursor_rait.close()
    cursor_free.close()
    cursor_iap.close()
    cursor.close()
    conn.close()

    print(Fore.YELLOW, page, "Закрыли проблемное
соединение!")

    conn = MySQLdb.connect('localhost', 'VProgramMist',
'1998Vm0000', 'mobasta_history',
                                charset='utf8mb4',
                                use_unicode=False)
    cursor = conn.cursor()
    cursor.execute("SET CHARACTER SET utf8mb4;")
    cursor.execute("SET character_set_connection=utf8mb4;")

```

```

        # Курсоры для получения внешних ключей
        cursor_cat = conn.cursor()
        cursor_rait = conn.cursor()
        cursor_free = conn.cursor()
        cursor_iap = conn.cursor()
        cursor_key_checker = conn.cursor()

        print(Fore.YELLOW, page, "Открыли соединение
повторно!")

        continue
        break
    if continue_flag:
        continue
    if full_stack_flag:
        break

    cursor_cat.close()
    cursor_rait.close()
    cursor_free.close()
    cursor_iap.close()
    cursor.close()
    conn.close()
    print(Fore.LIGHTRED_EX, page, "Закрыли соединение!")
except:
    print(Fore.RED, page, "Проблемы с сетью!")
    continue
break

dic_coll = {
    'history_top_free': 'TOP_FREE',
    'history_top_paid': 'TOP_PAID',
    'history_trending': 'TRENDING',
    'history_top_grossing': 'TOP_GROSSING',
}

for i in dic_coll:

    # Блок проверки числа записей. Если число записей за текущий день
    = 500, то прерываем обновление.
    # Требуется для получения более достоверных данных при анализе
    данных.

    conn = MySQLdb.connect('localhost', 'VProgramMist', '1998Vm0000',
'mobasta_history', charset='utf8mb4',
        use_unicode=False)
    cursor = conn.cursor()
    cursor.execute("SELECT COUNT(*), DATEDIFF(MAX(date_point),
MIN(date_point)), MAX(date_point) FROM " + i)

```

```
t = cursor.fetchone()
if (t[0] - 500 * t[1]) >= 500 and t[2] == datetime.date.today():
    continue
```

Если число записей за текущие сутки < 500 то приступаем к выполнению скрипта

```
thread1 = Thread(target=scrapy, args=(i, dic_coll.get(i), 108, 1))
thread2 = Thread(target=scrapy, args=(i, dic_coll.get(i), 108, 2))
thread3 = Thread(target=scrapy, args=(i, dic_coll.get(i), 108, 3))
thread4 = Thread(target=scrapy, args=(i, dic_coll.get(i), 108, 4))
thread5 = Thread(target=scrapy, args=(i, dic_coll.get(i), 108, 5))
```

```
thread1.start()
thread2.start()
thread3.start()
thread4.start()
thread5.start()
```

```
thread1.join()
thread2.join()
thread3.join()
thread4.join()
thread5.join()
```

Фрагмент модуля обработки данных

```
def get_data_from_table(table, key):
    warnings.filterwarnings("ignore")

    conn = MySQLdb.connect('localhost', 'VProgramMist', '1998Vm0000',
        'mobasta_history', charset='utf8mb4',
        use_unicode=False)
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM {table} LIMIT 1".format(table=table))

    cursor.execute("""SELECT date_point, app_id, title, score, price,
free_translate, iap_translate, iap_range,
size, installs, name_c_rating, category_translate
FROM ((( {table} LEFT JOIN category
ON {table}.category = category.id_category) LEFT JOIN
content_rating
ON {table}.content_rating = content_rating.id_c_rating) LEFT JOIN
free
ON {table}.free = free.id_free) LEFT JOIN iap
ON {table}.iap = iap.id_iap""").format(table=table))

    cf = cursor.fetchall()

    cursor.close()
    conn.close()
```

```

# Коллекция для данных, полученных из базы

dt = {"Date point": [],
      "ID": [],
      "Title": [],
      "Rating score": [],
      "Price": [],
      "Free": [],
      "IAP": [],
      "IAP Range": [],
      "Size": [],
      "Installs": [],
      "Content rating": [],
      "Category": []}

for i in cf:
    for j, c in zip(i, dt.keys()):
        if c == "Category":
            dt[c].append(j.decode(encoding="utf-8", errors="strict"))
        else:
            dt[c].append(j)

data = pd.DataFrame(dt, columns=dt.keys())

# Реформат полей из байт-строк в строки
data['Installs'] = data['Installs'].apply(
    lambda x: float(x.decode(encoding="utf-8", errors="strict").replace('+',
'').replace(',', '')))

for i in dt.keys():
    bstr_to_str(data, i)

# Группировка приложений по жанрам

if key == 'Category count':
    return (data.groupby('Category')
            ['Installs'].agg(np.size)).sort_values(ascending=False).to_dict()

# Группировка приложений по наличию встроенных покупок

elif key == 'IAP count':
    return (data.groupby('IAP')['Installs'].agg(np.size)).to_dict()

# Группировка приложений по возрастным категориям

elif key == 'Content rating count':
    return (data.groupby('Content rating')
            ['Installs'].agg(np.size)).sort_values(ascending=False).to_dict()

# Средняя цена встроенных покупок (ранжирование)

```



```

elif key == 'IAP Range':
    dic = {0: [], 1: []}
    for i in data['IAP Range']:
        if not i is None:
            a = i.replace('RUB ', '').replace('RUB\xa0', '').replace(',', '')
            if '-' in a:
                dic[0].append(float(a.split(' - ')[0]))
                dic[1].append(float(a.split(' - ')[1]))
            else:
                dic[0].append(float(a))
                dic[1].append(float(a))

    return {0: np.average(dic[0]), 1: np.average(dic[1])}

# Средние оценки по категориям

elif key == 'Category and Rating':
    return (data.dropna().groupby('Category')['Rating
score'].agg(np.average)).sort_values(ascending=False). \
        to_dict()

# Количество платных / бесплатных приложений

elif key == 'Free count':
    return (data.groupby('Free')['Installs'].agg(np.size)).to_dict()

# Средняя оценка по приложениям с наличием встроенных покупок
и без...

elif key == 'IAP and Rating':
    return (data.dropna().groupby('IAP')['Rating
score'].agg(np.average)).to_dict()

# Средняя оценка по возрастным категориям

elif key == 'Content and Rating':
    return (data.dropna().groupby('Content rating')['Rating
score'].agg(np.average)).sort_values(ascending=False). \
        to_dict()

# Среднее количество скачиваний по возрастным категориям
elif key == 'Content rating and Installs':
    return (data.groupby('Content rating')
['Installs'].agg(np.average)).sort_values(ascending=False).to_dict()

# Среднее количество скачиваний по категориям

elif key == 'Category and Installs':
    return (data.groupby('Category')
['Installs'].agg(np.average)).sort_values(ascending=False).to_dict()

```

```

# Среднее количество скачиваний по оценкам

elif key == 'Installs and Rating':
    return (data.dropna().groupby('Rating score')
['Installs'].agg(np.average)).sort_values(ascending=False). \
        to_dict()

```

Фрагмент модуля формирования графиков

```

def top_grossing(request):
    category_count_char = FusionCharts("column2d", "category-count",
"95%", "500", "category-count-char", "json",
        get_chart('history_top_grossing', 'category_count'))

    iap_count_char = FusionCharts("pie2d", "iap-count", "600", "400", "iap-
count-char", "json",
        get_chart('history_top_grossing', 'iap_count'))

    cont_rating_count_char = FusionCharts("column2d", "cont-rait-count",
"800", "400", "cont-rating-count-char", "json",
        get_chart('history_top_grossing',
'content_rating_count'))

    category_and_rating_char = FusionCharts("column2d", "cat-and-rait",
"95%", "500", "category-and-rating-char",
        "json", get_chart('history_top_grossing',
'category_and_rating'))

    free_count_char = FusionCharts("pie2d", "free-count", "600", "400",
"free-count-char", "json",
        get_chart('history_top_grossing', 'free_count'))

    iap_and_rating_char = FusionCharts("column2d", "iap-and-rating", "600",
"400", "iap-and-rating-char",
        "json", get_chart('history_top_grossing',
'iap_and_rating'))

    content_and_rating_char = FusionCharts("column2d", "content-and-
rating", "600", "400", "content-and-rating-char",
        "json", get_chart('history_top_grossing',
'content_and_rating'))

    content_rating_and_installs_char = FusionCharts("column2d", "content-
rating-and-installs", "600", "400",
        "content-rating-and-installs-char", "json",
        get_chart('history_top_grossing',
'content_rating_and_installs'))

    category_and_installs_char = FusionCharts("column2d", "category-and-
installs", "95%", "500",
        "category-and-installs-char", "json",

```

```

        get_chart('history_top_grossing',
'category_and_installs'))

    installs_and_rating_char = FusionCharts("column2d", "installs-and-
rating", "95%", "500", "installs-and-rating-char",
        "json", get_chart('history_top_grossing',
'installs_and_rating'))

```

```

return render(request, 'index.html', {
    'output': [
        iap_count_char.render(),
        category_count_char.render(),
        cont_rating_count_char.render(),
        category_and_rating_char.render(),
        free_count_char.render(),
        iap_and_rating_char.render(),
        content_and_rating_char.render(),
        content_rating_and_installs_char.render(),
        category_and_installs_char.render(),
        installs_and_rating_char.render()
    ],
    'title': 'бестселлерам',
    'iap_range': get_chart('history_top_grossing', 'iap_range'),
    'today': datetime.date.today().strftime("%Y-%m-%d"),
    'table': "top_grossing",
})

```

```

import json
from collections import OrderedDict

```

```

# Common base class for FC
class FusionCharts:

```

```

    baseTemplate = """
        <script type="text/javascript">
            FusionCharts.ready(function () {
                __FC__
            });
        </script>"""

```

```

    constructorTemplate = """new FusionCharts(__constructorOptions__);"""
    renderTemplate = """FusionCharts("__chartId__").render();"""
    eventTemplate =
    """FusionCharts("__chartId__").addEventListener("__fceventname__", __fceventb
ody__);"""

```

```

    # constructor
    def __init__(self, type, id, width, height, renderAt, dataFormat,
dataSource):

```

```

self.eventOptions = OrderedDict()
self.constructorOptions = {}
self.constructorOptions['type'] = type
self.constructorOptions['id'] = id
self.constructorOptions['width'] = width
self.constructorOptions['height'] = height
self.constructorOptions['renderAt'] = renderAt
self.constructorOptions['dataFormat'] = dataFormat
#dataSource = unicode(dataSource, errors='replace')
self.constructorOptions['dataSource'] = dataSource

def addEvent(self, eventName, funcName):
    self.eventOptions[eventName] = funcName

def addMessage(self, messageName, messageValue):
    self.constructorOptions[messageName] = messageValue

# render the chart created
# It prints a script and calls the FusionCharts javascript render method
of created chart
def render(self):

    # Serialize constructorOptions to a JSON formatted
    self.readyJson = json.dumps(self.constructorOptions,
ensure_ascii=False)

    # Create Fusioncharts constructor from template and insert JSON data
in it
    self.readyJson =
FusionCharts.constructorTemplate.replace('__constructorOptions__',
self.readyJson)

    # Iterate and attach EventHandler from template
for key, value in self.eventOptions.items():
        self.readyJson = self.readyJson +
FusionCharts.eventTemplate.replace('__chartId__',
self.constructorOptions['id'])
        self.readyJson = self.readyJson.replace("_fceventname_",
key).replace("_fceventbody_", value)

    # FusionCharts Render method will create chart
    self.readyJson = self.readyJson +
FusionCharts.renderTemplate.replace('__chartId__',
self.constructorOptions['id'])
    self.readyJson = FusionCharts.baseTemplate.replace("__FC__",
self.readyJson)

    self.readyJson = self.readyJson.replace("\n", "")
    self.readyJson = self.readyJson.replace("\t", "")

    if(self.constructorOptions['dataFormat'] == 'json'):

```

```

self.readyJson = self.readyJson.replace('\', '')
self.readyJson = self.readyJson.replace('{', '{')
self.readyJson = self.readyJson.replace('}', '}')

return self.readyJson

```

Шаблон HTML-страницы коллекции

```

<!DOCTYPE html>
{% load staticfiles %}
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="{% static 'css/styles.css' %}">
  {% load static %}
  <script type="text/javascript" src="{% static 'js/fusioncharts.js'
%}"></script>
  <script type="text/javascript" src="{% static
'js/themes/fusioncharts.theme.fusion.js' %}"></script>
  <script src="https://d3js.org/d3.v5.min.js"></script>
  <title>Apps Analytics</title>
</head>
<body>

<header>
  <!--  -->
  <h1 id="logo-title">Статистика по {{ title }}</h1>
</header>

<div class="overlay">
  <nav class="overlayMenu">
    <ul role="menu">
      <li><a role="menuitem" href="{% url 'all' %}">Все показатели!
</a></li>
      <li><a role="menuitem" href="{% url 'top_grossing'
%}">Бестселлеры!</a></li>
      <li><a role="menuitem" href="{% url 'top_free' %}">Топ
бесплатных!</a></li>
      <li><a role="menuitem" href="{% url 'top_paid' %}">Топ
платных!</a></li>
      <li><a role="menuitem" href="{% url 'trending'
%}">Набирающие популярность!</a></li>
    </ul>
  </nav>
</div>

<div class="navBurger" role="navigation" id="navToggle"></div>

<div class="date-picker">

```

```

    <form action="{% url 'download' %}" class="date-picker"
method="post" target="_blank">
    {% csrf_token %}
    <a href="#" class="show-data">Отобразить</a>
    <input hidden type="text" name="table" value="{{ table }}">
    <input min="2019-03-01" max="{{ today }}" type="date"
name="date_start" id="date_start" value="{{ today }}">
    <input min="2019-03-01" max="{{ today }}" type="date"
name="date_end" id="date_end" value="{{ today }}">
    <button type="submit" class="download">Скачать данные</button>
    </form>
</div>

<main>
    <div id="category-count-char"></div>
    {{ output.0|safe }}
    <div class="line"></div>
    <div class="iap">
        <div id="iap-count-char"></div>
        {{ output.1|safe }}
        <h2 id="iap-min">Средняя минимальная цена: {{iap_range.0|
floatformat:2}}</h2>
        <h2 id="iap-max">Средняя максимальная цена: {{iap_range.1|
floatformat:2}}</h2>
    </div>
    <div id="cont-rating-count-char"></div>
    {{ output.2|safe }}
    <div class="line"></div>
    <div id="category-and-rating-char"></div>
    {{ output.3|safe }}
    <div class="line"></div>
    <div id="free-count-char"></div>
    {{ output.4|safe }}
    <div id="iap-and-rating-char"></div>
    {{ output.5|safe }}
    <div class="line"></div>
    <div id="content-and-rating-char"></div>
    {{ output.6|safe }}
    <div id="content-rating-and-installs-char"></div>
    {{ output.7|safe }}
    <div class="line"></div>
    <div id="category-and-installs-char"></div>
    {{ output.8|safe }}
    <div class="line"></div>
    <div id="installs-and-rating-char"></div>
    {{ output.9|safe }}
</main>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js'></
script>
</script>

```

```
$("#navToggle").click(function() {
    $(this).toggleClass("active");
    $(".overlay").toggleClass("open");
    $("body").toggleClass("locked");
});
$('.overlay').click(function() {
    $(this).removeClass('open');
    $('.navBurger').removeClass('active');
});
</script>
</body>
</html>
```

Приложение В

Пример выгрузки данных

	B	C	D	E	F	G	H	I	J	K	L	M
1	Date point	ID	Title	Rating score	Price	Free	IAP	IAP Range	Size	Installs	Content rating	Category
2	2019-06-01	com.basevelocity.radarscope	RadarScope	4,6	650.00746	Платно	Встроенных покупок нет		Varies with device	100,000+	Rated for 3+	Погода
3	2019-06-01	com.aniplex.fategrandorder.en	Fate/Grand Order (English)	4,4	0	Бесплатно	Встроенных покупок нет		63M	1,000,000+	Rated for 12+	Ролевые игры
4	2019-06-01	com.bandainamcoent.dblegends_ww	DRAGON BALL LEGENDS	4,6	0	Бесплатно	Встроенных покупок нет		61M	10,000,000+	Rated for 12+	Экшн игры
5	2019-06-01	com.bandainamcoent.saomdna	SWORD ART ONLINE:Memory Defrag	4,8	0	Бесплатно	Встроенных покупок нет		23M	1,000,000+	Rated for 7+	Ролевые игры
6	2019-06-01	com.bandainamcogames.dbzdokkanww	DRAGON BALL Z DOKKAN BATTLE	4,2	0	Бесплатно	Встроенных покупок нет		85M	10,000,000+	Rated for 12+	Экшн игры
7	2019-06-01	com.bydeluxe.d3.android.program.starz	STARZ	4,3	0	Бесплатно	Встроенных покупок нет		Varies with device	10,000,000+	Rated for 18+	Развлечения
8	2019-06-01	com.cbs.app	CBS - Full Episodes & Live TV	3,8	0	Бесплатно	Встроенных покупок нет		22M	10,000,000+	Rated for 12+	Развлечения
9	2019-06-01	com.clearchannel.iheartradio.controller	iHeartRadio - Free Music, Radio & Podcasts	4,7	0	Бесплатно	Встроенных покупок нет		Varies with device	50,000,000+	Rated for 12+	Музыка и аудио
10	2019-06-01	com.cyberstep.toreba.android	Crane Game Toreba	2,9	0	Бесплатно	Встроенных покупок нет		16M	1,000,000+	Rated for 12+	Развлечения
11	2019-06-01	com.dazn	DAZN Live Fight Sports: Boxing, MMA & More	3,8	0	Бесплатно	Встроенных покупок нет		Varies with device	5,000,000+	Rated for 3+	Спорт
12	2019-06-01	com.ember.meowmatch	Meow Match: Cats Matching 3 Puzzle & Ball Blast	4,7	0	Бесплатно	Встроенных покупок нет		105M	1,000,000+	Rated for 3+	Пазлы
13	2019-06-01	com.everydollar.android	EveryDollar: Budget Tool and Expense Tracker	4	0	Бесплатно	Встроенных покупок нет		13M	1,000,000+	Rated for 3+	Финансы
14	2019-06-01	com.GF.apcguscn.hwyad.google	Royal Chaos-The Greatest Royal Romance	4,2	0	Бесплатно	Встроенных покупок нет		30M	1,000,000+	Rated for 3+	Ролевые игры
15	2019-06-01	com.hbo.hbonow	HBO NOW: Stream TV & Movies	3,8	0	Бесплатно	Встроенных покупок нет		46M	10,000,000+	Rated for 12+	Развлечения
16	2019-06-01	com.heroesofchaos.ggplay.koramgame.na	Goddess: Primal Chaos - Free 3D Action MMORPG Game	4,6	0	Бесплатно	Встроенных покупок нет		26M	1,000,000+	Rated for 12+	Ролевые игры
17	2019-06-01	com.hily.app	Hily Dating: Chat, Match & Meet Singles	4,2	0	Бесплатно	Встроенных покупок нет		65M	1,000,000+	Rated for 18+	Знакомства
18	2019-06-01	air.com.and.games505.gemsofwar	Gems of War - Match 3 RPG	4,6	0	Бесплатно	Встроенные покупки есть	RUB 149.00 - RUB 7,290.00	45M	1,000,000+	Rated for 7+	Пазлы
19	2019-06-01	air.com.buffalo_studios.newflashbingo	Bingo Blitz™ - Bingo Games	4,4	0	Бесплатно	Встроенные покупки есть	RUB 32.02 - RUB 32,790.00	Varies with device	10,000,000+	Rated for 12+	Казино
20	2019-06-01	air.com.goodgamestudios.empirefourkingdoms	Empire: Four Kingdoms	4,2	0	Бесплатно	Встроенные покупки есть	RUB 75.00 - RUB 24,990.00	84M	10,000,000+	Rated for 3+	Стратегии
21	2019-06-01	air.com.pixelfederation.diggy	Diggy's Adventure: Escape this 2D Mine Maze Puzzle	4,6	0	Бесплатно	Встроенные покупки есть	RUB 85.00 - RUB 17,499.00	78M	10,000,000+	Rated for 3+	Приключенчески
22	2019-06-01	air.com.playtika.cvs	Vegas Downtown Slots™ - Slot Machines & Word Games	4,7	0	Бесплатно	Встроенные покупки есть	RUB 59.00 - RUB 28,090.00	Varies with device	1,000,000+	Rated for 12+	Казино
23	2019-06-01	air.com.playtika.slotomania	Slotomania™ Slots Casino: Vegas Slot Machine Games	4,4	0	Бесплатно	Встроенные покупки есть	RUB 40.69 - RUB 26,090.00	Varies with device	10,000,000+	Rated for 12+	Казино
24	2019-06-01	air.com.rosetastone.mobile.CoursePlayer	Rosetta Stone: Learn Languages	4,5	0	Бесплатно	Встроенные покупки есть	RUB 75.00 - RUB 13,690.00	32M	5,000,000+	Rated for 3+	Обучение
25	2019-06-01	air.com.sgn.cookiejam.gp	Cookie Jam™ Match 3 Games & Free Puzzle Game	4,4	0	Бесплатно	Встроенные покупки есть	RUB 69.00 - RUB 18,690.00	Varies with device	50,000,000+	Rated for 3+	Пазлы
26	2019-06-01	air.com.sgn.cookiejamblast.gp	Cookie Jam Blast™ New Match 3 Puzzle Saga Game	4,6	0	Бесплатно	Встроенные покупки есть	RUB 69.00 - RUB 18,690.00	74M	5,000,000+	Rated for 3+	Пазлы
27	2019-06-01	air.com.spicrackmedia.bingoshowdown	Bingo Showdown: Free Bingo Game – Live Bingo	4,7	0	Бесплатно	Встроенные покупки есть	RUB 59.00 - RUB 6,690.00	73M	1,000,000+	Rated for 12+	Казино
28	2019-06-01	and.onemt.war.ar	انتقام السلاطين	4,4	0	Бесплатно	Встроенные покупки есть	RUB 55.00 - RUB 7,899.00	90M	5,000,000+	Rated for 3+	Стратегии
29	2019-06-01	and.onemt.war.en	Revenge of Sultans	4,1	0	Бесплатно	Встроенные покупки есть	RUB 55.00 - RUB 25,590.00	90M	1,000,000+	Rated for 7+	Стратегии
30	2019-06-01	ata.kraken.heckfire	Kingdoms of Heckfire	4,4	0	Бесплатно	Встроенные покупки есть	RUB 55.00 - RUB 14,990.00	Varies with device	1,000,000+	Rated for 12+	Стратегии
31	2019-06-01	ata.squid.pimd	Party in my Dorm	3,8	0	Бесплатно	Встроенные покупки есть	RUB 75.00 - RUB 7,490.00	106M	5,000,000+	Rated for 16+	Казальные игры
32	2019-06-01	co.thefabulous.app	Fabulous: Self Care	4,6	0	Бесплатно	Встроенные покупки есть	RUB 279.00 - RUB 25,890.00	36M	5,000,000+	Rated for 3+	Здоровье и фитнес
33	2019-06-01	com.abradoodle.bingo	Bingo Abradoodle : Best Free Bingo Games	4,8	0	Бесплатно	Встроенные покупки есть	RUB 68.53 - RUB 4,190.00	61M	1,000,000+	Rated for 12+	Настольные игры
34	2019-06-01	com.abtnprojects.ambatana	letgo: Buy & Sell Used Stuff, Cars, Furniture	4,5	0	Бесплатно	Встроенные покупки есть	RUB 45.00 - RUB 3,399.00	26M	50,000,000+	Rated for 12+	Покупки
35	2019-06-01	com.adobe.reader	Adobe Acrobat Reader: PDF Viewer, Editor & Creator	4,3	0	Бесплатно	Встроенные покупки есть	RUB 67.77 - RUB 3,701.63	Varies with device	100,000,000+	Rated for 3+	Работа
36	2019-06-01	com.alien.shooter.galaxy.attack	Galaxy Attack: Alien Shooter	4,6	0	Бесплатно	Встроенные покупки есть	RUB 15.00 - RUB 6,190.00	59M	50,000,000+	Rated for 3+	Аркады
37	2019-06-01	com.alltrails.alltrails	AllTrails: Hiking, Running & Mountain Bike Trails	4,5	0	Бесплатно	Встроенные покупки есть	RUB 35.60 - RUB 10,290.00	29M	1,000,000+	Rated for 3+	Здоровье и фитнес
38	2019-06-01	com.ancestry.android.apps.ancestry	Ancestry	4,4	0	Бесплатно	Встроенные покупки есть	RUB 31.01 - RUB 3,190.00	Varies with device	5,000,000+	Rated for 3+	Для чтения
39	2019-06-01	com.and.riseofthekings	Rise of the Kings	4,3	0	Бесплатно	Встроенные покупки есть	RUB 59.00 - RUB 7,199.00	97M	5,000,000+	Rated for 12+	Стратегии
40	2019-06-01	com.antivirus	AVG AntiVirus 2019 for Android Security FREE	4,5	0	Бесплатно	Встроенные покупки есть	RUB 25.00 - RUB 3,169.00	29M	100,000,000+	Rated for 3+	Инструменты
41	2019-06-01	com.analon.weatherradar.free	NOAA Weather Radar Live & Alerts	4,4	0	Бесплатно	Встроенные покупки есть	RUB 45.00 - RUB 1,890.00	8.5M	10,000,000+	Rated for 3+	Погода

Рисунок В.1 – Пример выгрузки данных, сформированной системой, за 06.2019г. по коллекции «Бестселлеры»