

УДК 004.4

Руководитель курсовой работы: ст. преп. Рачков Андрей Владимирович

Мухаметшин А. Р., Курсовая работа направления подготовки «Программная инженерия» на тему «Проектирование ИС „Электронные визитки“»: Москва, 2024 г., МИРЭА – Российский технологический университет (РТУ МИРЭА), Институт информационных технологий (ИТ), кафедра инструментального и прикладного программного обеспечения (ИиППО) – 33 стр., 12 рис., 16 источн.  
 Ключевые слова: проектирование, электронные визитки, клиент-серверные системы, UML, IDEF0, Entity Relationship, База данных, Модель, Бизнес процесс, Диаграмма.  
 Целью работы является проектирование комплексной информационной системы для автоматизации процессов создания, управления и обмена электронными визитками. Разработаны модели процессов и архитектура клиент-серверной системы, логическая модель базы данных, а также техническая документация.

Muhametshin A. R., Course Work for the program "Software Engineering" on the topic "Design of the information system 'Electronic business cards": Moscow, 2024, MIREA – Russian Technological University (RTU MIREA), Institute of Information Technologies (IT), Department of Instrumental and Applied Software (IiPPO) – 33 pages, 12 figures, 16 sources.  
 Keywords: design, electronic business cards, client-server systems, UML, IDEF0, Entity Relationship, Database, Model, Business Process, Diagram.   
 The purpose of the work is to design an integrated information system for automating the processes of creating, managing and exchanging electronic business cards. Process models and architecture of the client-server system, a logical database model, as well as technical documentation have been developed.

РТУ МИРЭА: 119454, Москва, пр-т Вернадского, д. 78  
 Кафедра инструментального и прикладного программного обеспечения (ИиППО)  
 Тираж: 1 экз. (на правах рукописи)  
 Файл: «ПЗ\_ПКСС\_ИКБО-20-21\_МухаметшинАР.pdf», исполнитель Мухаметшин Александр Ринатович  
 © Мухаметшин А. Р.

СОДЕРЖАНИЕ

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 7](#_Toc24)

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 8](#_Toc25)

[ВВЕДЕНИЕ 9](#_Toc26)

[1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ 11](#_Toc27)

[1.1 Анализ существующих решений 11](#_Toc28)

[1.2 Определение требований к решению 14](#_Toc29)

[1.3 Выбор инструментов и методов создания информационной системы 17](#_Toc30)

[1.4 Постановка задачи к проектированию и разработке информационной системы 20](#_Toc31)

[Вывод по разделу 1 20](#_Toc32)

[2 ПРОЕКТНЫЙ РАЗДЕЛ 21](#_Toc33)

[2.1 Проектирование функциональной схемы 21](#_Toc34)

[2.2 Проектирование архитектуры информационной системы 21](#_Toc35)

[2.3 Проектирование клиентской части информационной системы 21](#_Toc36)

[2.4 Проектирование серверной части информационной системы 22](#_Toc37)

[2.5 Проектирование жизненного цикла информационной системы 23](#_Toc38)

[2.6 Проектирование схемы базы данных 23](#_Toc39)

[Вывод по разделу 2 23](#_Toc40)

[ЗАКЛЮЧЕНИЕ 24](#_Toc41)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчёте применяются следующие термины и определения.

|  |  |  |
| --- | --- | --- |
| **Масштабируемость** | – | **возможность увеличивать ресурсы или добавлять дополнительные экземпляры сервисов, чтобы поддерживать высокий уровень производительности системы при росте нагрузки** |
| **Микросервисная архитектура** | – | подход к разработке, при котором приложение разбивается на независимые, автономные сервисы, каждый из которых отвечает за конкретную функцию системы |

# **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

В настоящем отчёте применяются следующие сокращения и обозначения.

|  |  |  |
| --- | --- | --- |
| ИС | – | Информационная система |
| ПО | – | Программное Обеспечение |

**ВВЕДЕНИЕ**

Целью выпускной квалификационной работы является разработка интеллектуальной системы (ИС) для автоматизированной оценки спроса на продукт, которая позволит бизнесу прогнозировать рыночные потребности на основе анализа данных из разнородных источников с использованием технологий машинного обучения (ML). Система направлена на обеспечение компаний инструментом для повышения конкурентоспособности за счет точного предсказания спроса и оптимизации бизнес-процессов.

Актуальность разработки обусловлена стремительными изменениями рыночных условий, в которых традиционные методы оценки спроса, основанные на экспертных суждениях или ручном анализе данных, становятся недостаточно эффективными. Современные компании сталкиваются с необходимостью оперативной адаптации к динамике рынка, что требует автоматизации процессов сбора, обработки и анализа данных. Исследования показывают, что внедрение интеллектуальных систем анализа данных позволяет сократить издержки на 15–20% за счет минимизации избыточных запасов и упущенных возможностей [1]. Кроме того, использование технологий машинного обучения для прогнозирования спроса демонстрирует рост точности предсказаний до 85% в сравнении с традиционными подходами [2]. В условиях цифровизации бизнеса разработка таких систем приобретает стратегическое значение.

Новизна работы заключается в создании универсальной информационной системы, которая интегрирует сбор данных из внутренних источников (CRM, ERP), внешних API (например, Google Trends) и веб-скрапинга (данные маркетплейсов), их обработку с применением ML-моделей и предоставление результатов через удобный веб-интерфейс и API. В отличие от существующих решений, предлагаемая система обладает гибкостью настройки источников данных и прогнозных моделей. Это обеспечивает высокую адаптивность системы к различным бизнес-сценариям и требованиям пользователей.

В ходе работы должны быть выполнены следующие задачи:

* провести анализ предметной области и существующих решений;
* спроектировать архитектуру системы;
* выбрать средства и методы для разработки;
* разработать информационную систему и провести тестирование;
* рассчитать стоимость проведения работ.

Объектом исследования является интеллектуальная система оценки спроса на продукт, включающая модули сбора, предобработки и анализа данных с использованием технологий машинного обучения для прогнозирования рыночных потребностей.

На защиту выносится информационная система, обеспечивающая автоматизированный сбор данных из различных источников, их анализ с помощью ML-моделей и предоставление прогнозов спроса через API и веб-интерфейс.

Стандарты, используемые в работе: СМКО МИРЭА 7.5.1/03.П.30-19 [3], ГОСТ 7.32-2017 [4], ГОСТ Р 7.0.100-2018 [5], СМКО МИРЭА 7.5.1/03.П.67-19 [6], ФГОС ВО 3++ по направлению подготовки 09.03.04 Программная инженерия [7].

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ
   1. Анализ существующих решений

В условиях цифровизации бизнеса и роста объемов данных системы автоматической оценки спроса становятся важным инструментом для компаний, стремящихся оптимизировать управление запасами и повысить эффективность маркетинговых стратегий. Для определения требований к разрабатываемой информационной системе проведен сравнительный анализ существующих решений.

* + 1. Ozon Seller

Ozon Seller [8] – это инструмент аналитики, разработанный для продавцов на платформе Ozon, одной из крупнейших российских торговых онлайн-площадок. Платформа предоставляет пользователям доступ к статистике продаж, прогнозам спроса и аналитике конкурентной среды. Веб-интерфейс приложения позволяет визуализировать ключевые показатели, такие как объемы продаж по категориям товаров и динамика спроса в определенные периоды. Пример веб-интерфейса представлен на рисунке 1.1.

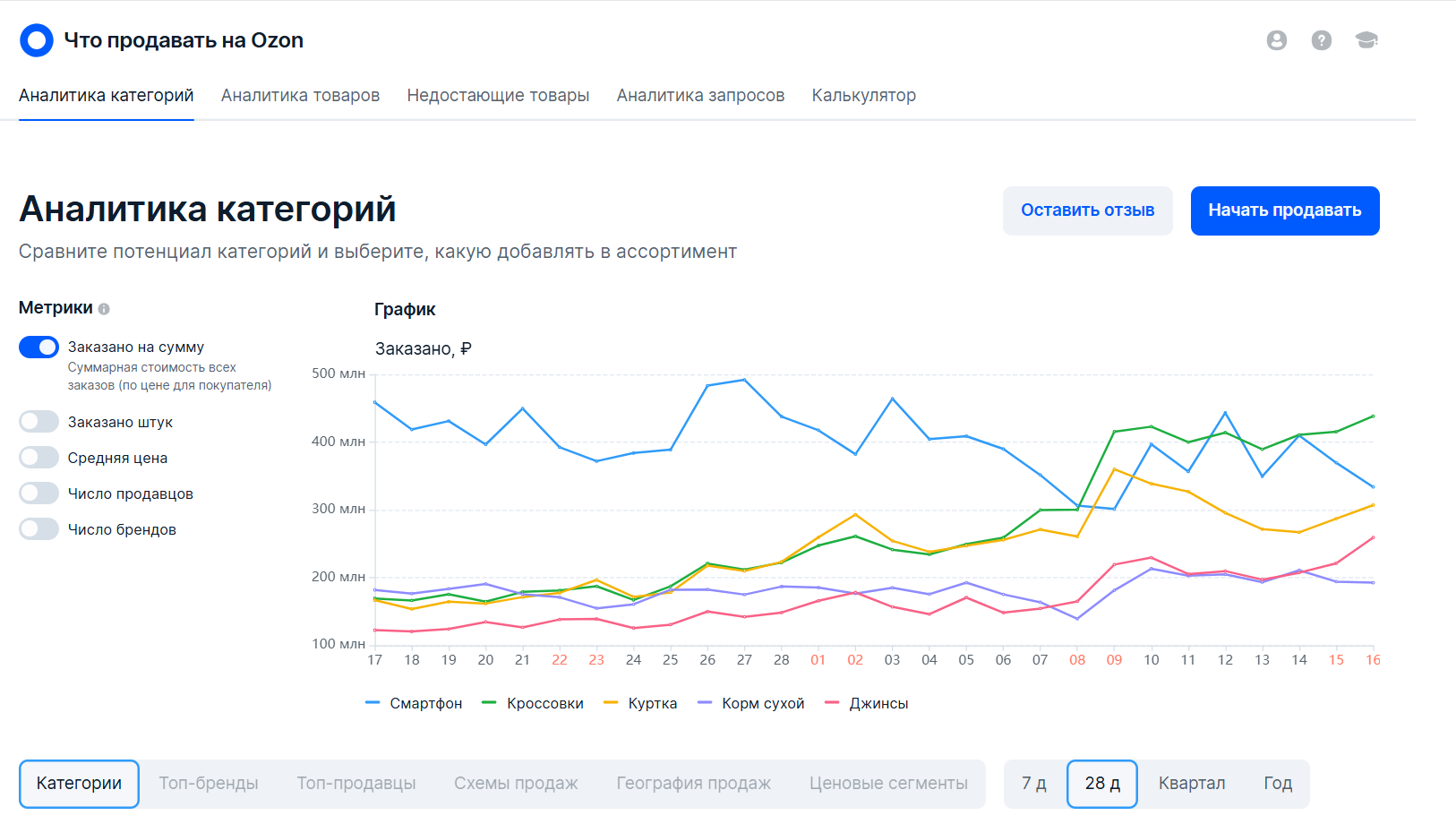


Рисунок 1.1 – Веб-интерфейс платформы Ozon Seller

* + 1. Moneyplace

Moneyplace [9] – это российская аналитическая платформа, предназначенная для продавцов маркетплейсов, включая Ozon, Wildberries и Яндекс.Маркет. Сервис предоставляет инструменты для анализа спроса, конкурентной среды и управления ассортиментом. Веб-интерфейс платформы ориентирован на визуализацию данных о продажах и прогнозах, что помогает пользователям принимать обоснованные решения о закупках и ценообразовании. Пример интерфейса показан на рисунке 1.2.

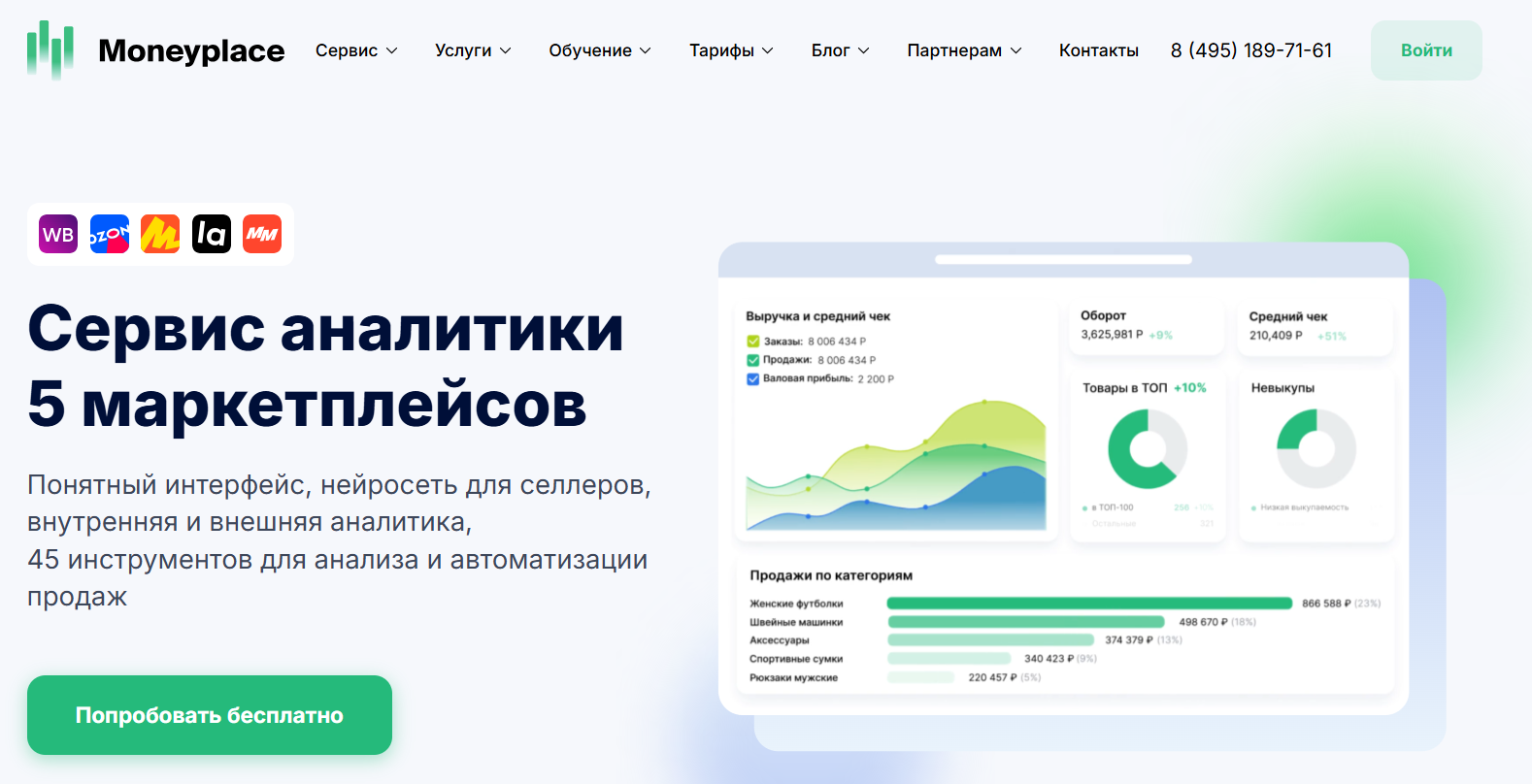


Рисунок 1.2 – Веб-интерфейс платформы 1.1.2 Moneyplace

* + 1. MPStats

MPStats [10] — это аналитическая платформа, предназначенная для продавцов на маркетплейсах Wildberries, Ozon и Яндекс.Маркет. Сервис предоставляет инструменты для анализа продаж, конкурентов, оптимизации рекламных кампаний и исследования товаров. Веб-интерфейс платформы позволяет пользователям получать доступ к детальным отчетам о продажах, выручке, ценах, рейтингах и других ключевых показателях. Кроме того, MPStats предлагает расширение для браузера Chrome, которое упрощает доступ к аналитическим данным непосредственно на страницах маркетплейсов. Пример интерфейса показан на рисунке 1.3.

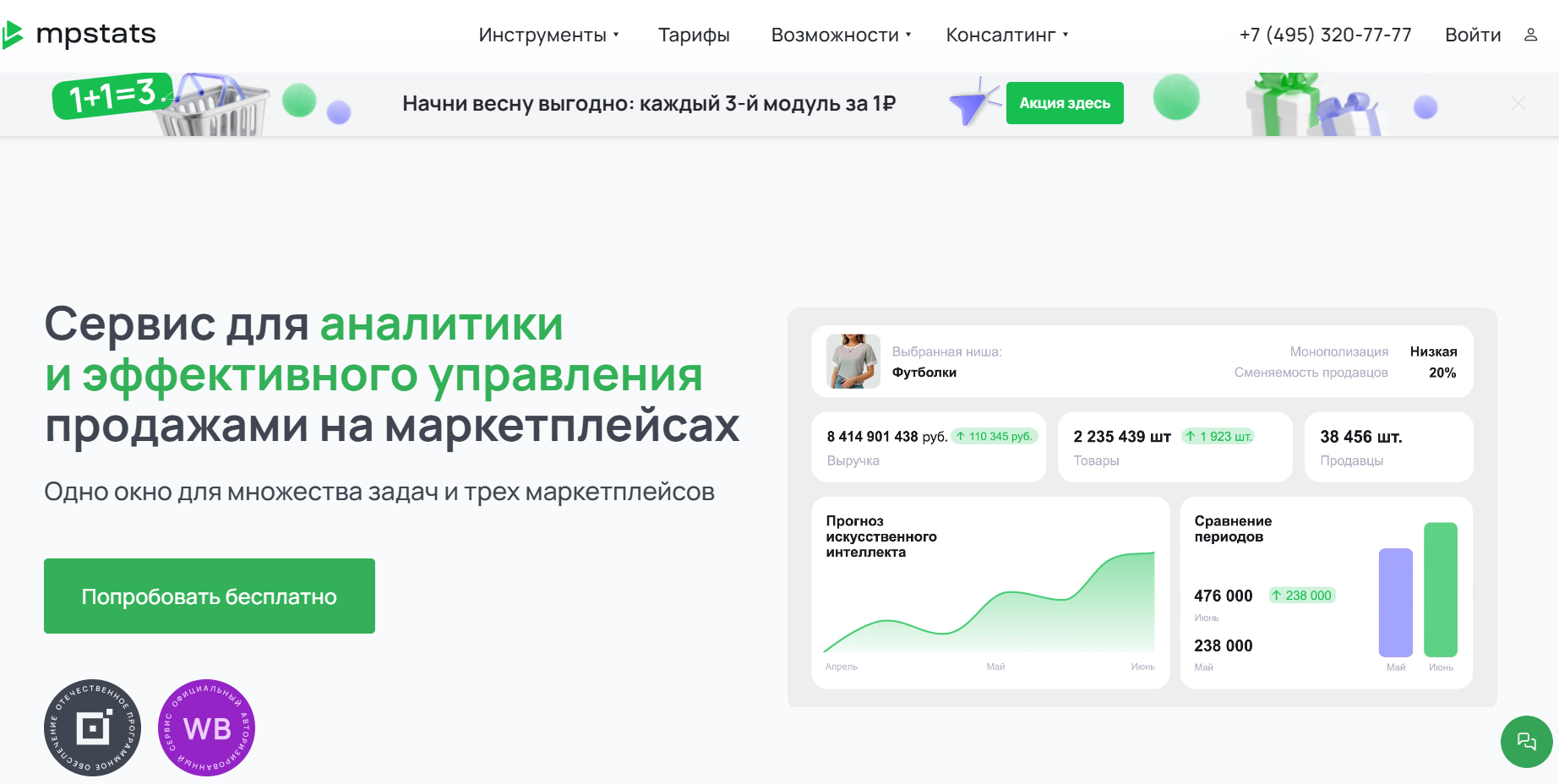


Рисунок 1.3 – Веб-интерфейс платформы MPStats

* + 1. Результаты сравнительного анализа

По итогу сравнительного анализа существующих аналогов разрабатываемой ИС составлена таблица 1.1.

Таблица 1.1 – Сравнительный анализ аналогов ИС

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий** | **Ozon Seller** | **Moneyplace** | **MPStats** |
| Функциональность | Отчеты о продажах, анализ конкурентов, рекомендации по закупкам | Анализ спроса, сравнение цен, прогноз остатков | Аналитика продаж, анализ конкурентов, оптимизация рекламы, исследование товаров |
| Интеграция | REST API для доступа к данным | API для получения аналитических данных | API для получения аналитических данных |
| Гибкость | Ограничена платформой Ozon | Поддержка нескольких маркетплейсов | Поддержка Wildberries, Ozon, Яндекс.Маркет |
| Аналитика и статистика | Статистика продаж и конкурентов | Детализированная аналитика по товарам | Подробные отчеты по продажам, выручке, ценам, рейтингам |

Продолжение таблицы 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Дополнительные возможности | Простота интерфейса | Анализ данных с разных платформ | Расширение для браузеров, инструменты для продавцов |

* + 1. Вывод и требования к разрабатываемой системе

Проведенный анализ показывает, что существующие решения, такие как Ozon Seller, Moneyplace и MPStats, ориентированы преимущественно на анализ данных внутри конкретных маркетплейсов и не обладают достаточной гибкостью для интеграции внешних источников или применения продвинутых методов прогнозирования. Разрабатываемая система должна преодолеть эти ограничения и соответствовать следующим требованиям:

* Гибкость настройки: Возможность подключения различных источников данных (внутренние системы, API, веб-скрапинг) и адаптации под нужды конкретного бизнеса.
* Интеграция: Предоставление REST API для легкой интеграции с существующими системами компаний и экспорта данных в удобных форматах (JSON, PDF, Excel).
* Аналитика: Использование ML-моделей для глубокого анализа и точного прогнозирования спроса, а также визуализация результатов через веб-интерфейс.

Таким образом, разрабатываемая система должна сочетать универсальность, высокую точность прогнозов и удобство использования, что обеспечит ее конкурентоспособность на рынке аналитических решений.

* 1. Определение требований к решению
     1. Формулирование требований к программной части системы

**Функциональные требования**

Система должна автоматизировать сбор данных об использовании продуктов и прогнозировать их спрос. Целевой аудиторией являются владельцы бизнесов.

Необходимо обеспечить:

* Сбор данных из внутренних систем (CRM, ERP), внешних API и веб-скрапинг.
* Обработка и нормализация данных для ML.
* Прогнозирование спроса с использованием ML-моделей.
* Предоставление API для интеграции с внешними системами.
* Визуализация прогнозов и аналитики через веб-интерфейс.

**Нефункциональные требования**

Основными метриками, которые необходимо обеспечить является:

* Время отклика API — не более 1 секунды при нагрузке до 100 пользователей.
* Время генерации прогноза — не более 5 минут.
* Безопасность: авторизация через JWT, шифрование данных.

Система должна запускаться в Docker’е для обеспечения кроссплатформенности.

Требования пользователей

Для удобства использования система должна иметь:

* Удобный интерфейс для настройки источников данных и просмотра прогнозов.
* Экспорт отчетов в PDF/Excel.
  + 1. Формулирование требований к аппаратной части системы

**Требования к серверной части системы**

Для серверной части, включающей микросервисы для сбора данных, обработки запросов и маршрутизации через API Gateway, выдвигаются следующие минимальные требования к аппаратному обеспечению:

* Процессор: 64-битный, с минимальной частотой 2.0 ГГц, 2 ядра (например, Intel Core i3 или аналогичный);
* Оперативная память: 4 ГБ для обеспечения одновременной работы нескольких микросервисов и брокера сообщений;
* Пропускная способность сетевого соединения: 100 Мбит/с для быстрой передачи данных между сервисами и внешними источниками;
* Хранилище: 50 ГБ SSD для хранения временных данных и логов.

Требования к оперативной памяти обусловлены необходимостью параллельной обработки запросов от множества пользователей и асинхронного взаимодействия между микросервисами через брокер сообщений. Высокая пропускная способность сети необходима для оперативного получения данных из внешних API и передачи их в систему для последующей обработки.

**Требования к модулям обработки данных и машинного обучения**

Для микросервисов, отвечающих за предобработку данных и выполнение прогнозов спроса с использованием моделей машинного обучения (ML Service), выдвигаются более строгие требования:

* Процессор: 64-битный с поддержкой аппаратной виртуализации (Intel VT-x или AMD-V), минимум 4 ядра (например, Intel Core i5 или аналогичный);
* Оперативная память: 8 ГБ (рекомендуется 16 ГБ для работы с большими наборами данных и сложными моделями);
* Пропускная способность сетевого соединения: 100 Мбит/с для передачи данных между сервисами и базой данных;
* Графический процессор (GPU): 2 ГБ видеопамяти (например, NVIDIA GTX 1050 или аналогичный) или 8 CPU ядер при отсутствии GPU;
* Хранилище: 100 ГБ SSD для хранения обработанных данных, моделей и временных файлов.

Необходимость в увеличенной оперативной памяти и GPU обусловлена высокими вычислительными нагрузками при обучении и использовании ML-моделей, таких как регрессии или нейронные сети. GPU значительно ускоряет выполнение операций с тензорами, что критично для работы с большими объемами данных в реальном времени. При отсутствии GPU нагрузка перекладывается на CPU, что требует дополнительных ядер для поддержания производительности.

* 1. Выбор инструментов и методов создания информационной системы
     1. Выбор технологий для серверной части

Для серверной части системы выбраны следующие технологии:

* Go (Gin): Высокопроизводительный язык программирования с фреймворком Gin для создания микросервисов. Go обеспечивает быструю обработку запросов, низкое потребление ресурсов и простоту работы с конкурентными задачами благодаря встроенным горутинам. Это особенно важно для микросервисов, таких как Data Collector и API Gateway, которые обрабатывают большие объемы входящих данных.
* gRPC и Protobuf: Используются для быстрого и эффективного взаимодействия между микросервисами. gRPC, основанный на Protocol Buffers, обеспечивает низкую задержку и высокую пропускную способность по сравнению с REST, что критично для передачи данных между сервисами в реальном времени.
* RabbitMQ: Брокер сообщений для асинхронной коммуникации между микросервисами. RabbitMQ позволяет декомпозировать процессы сбора, обработки и анализа данных, обеспечивая устойчивость системы при высоких нагрузках.

Преимущества данного подхода включают высокую скорость выполнения запросов, легкость масштабирования и надежность передачи данных, что соответствует требованиям микросервисной архитектуры.

* + 1. Выбор технологий для клиентской части

Клиентская часть системы реализована с использованием следующих технологий:

* React и TypeScript: React.js — библиотека JavaScript для построения динамических и адаптивных пользовательских интерфейсов с применением компонентного подхода.
* Tailwind CSS: Фреймворк для быстрого создания стилизованных интерфейсов. Tailwind предоставляет набор готовых классов, что ускоряет разработку и обеспечивает гибкость настройки дизайна под нужды пользователей.

Использование React с TypeScript и Tailwind CSS позволяет создать удобный, быстрый и адаптивный веб-интерфейс, обеспечивающий интерактивное взаимодействие с системой и визуализацию результатов анализа.

* + 1. Выбор СУБД

Для управления данными выбраны две системы:

* PostgreSQL: Основная реляционная база данных для хранения структурированных данных (пользователи, обработанные данные, прогнозы). PostgreSQL обеспечивает высокую производительность, надежность и поддержку сложных запросов, что делает её подходящей для аналитических задач.
* Redis: Используется как in-memory кэш для ускорения доступа к часто запрашиваемым данным, таким как последние прогнозы или аналитические сводки. Redis снижает нагрузку на PostgreSQL и улучшает время отклика серверной части.

Комбинация PostgreSQL и Redis обеспечивает баланс между долговременным хранением данных и быстрым доступом к ним, что критично для системы с высокой частотой запросов.

* + 1. Выбор технологий для создания ML-модуля

Модуль машинного обучения реализован с использованием:

* Python (scikit-learn, TensorFlow): Python выбран как основной язык для разработки ML-моделей благодаря богатому набору библиотек. Scikit-learn используется для базовых алгоритмов (например, линейной регрессии), а TensorFlow — для более сложных моделей, таких как нейронные сети для анализа временных рядов.
* ONNX: Формат для сохранения обученных моделей и их интеграции с Go через библиотеку onnxruntime-go. Это позволяет выполнять прогнозы непосредственно в микросервисе ML Service, минимизируя задержки.

Такой подход обеспечивает гибкость в обучении моделей и их эффективное использование в реальном времени.

* + 1. Выбор средств развертывания веб-приложения

Для удобного развертывания и масштабирования системы используется Docker. Этот инструмент позволяет запускать приложение в изолированных контейнерах, не требуя сложных настроек окружения, что значительно упрощает развертывание и масштабирование.

* + 1. Выбор среды разработки и вспомогательные инструменты

Для разработки системы выбраны следующие инструменты:

* GoLand: Интегрированная среда разработки (IDE) для Go, предоставляющая удобный интерфейс, автодополнение кода и встроенные плагины для работы с микросервисами и gRPC. Это ускоряет процесс написания серверной части.
* Visual Studio Code (VS Code): Легковесная IDE для разработки клиентской части на React и TypeScript. Поддержка расширений (например, для Tailwind CSS) делает её универсальным инструментом для фронтенда.
* PgAdmin: Графический интерфейс для администрирования PostgreSQL, упрощающий управление базой данных и выполнение запросов.

Эти инструменты выбраны за их популярность, функциональность и совместимость с технологическим стеком проекта.

* 1. Постановка задачи к проектированию и разработке информационной системы

Целью выпускной квалификационной работы является создание интеллектуальной системы, позволяющей пользователям анализировать рынок и эффективно прогнозировать спрос на товары.

Для реализации поставленной цели выделены следующие задачи к проектированию и разработке:

* Спроектировать информационную систему (архитектуру и взаимодействие элементов системы);
* Определить и обосновать информационные, технические, программные средства для разработки веб-приложения;
* Разработать серверную и клиентскую части информационной системы, предоставляющей пользователям заявленный функционал;
* Произвести тестирование разработанной информационной системы.

## **Вывод по разделу 1**

В данном разделе проведен детальный сравнительный анализ существующих решений для оценки спроса, что позволило выявить ключевые требования и конкурентные преимущества разрабатываемой системы. На основе анализа определены функциональные и нефункциональные требования, а также выбраны оптимальные инструменты и методы разработки. Поставлены задачи к проектированию и разработке, которые охватывают создание архитектуры, реализацию системы и её тестирование. Этот подход обеспечивает прочную основу для дальнейшей работы над проектом, гарантируя соответствие системы современным стандартам программной инженерии.

1. ПРОЕКТНЫЙ РАЗДЕЛ
   1. Проектирование функциональной схемы

Для разработки веб-приложения потребовалось спроектировать функциональную схему в методологии IDEF0, на которой изображены бизнес-процессы необходимые для создания веб-приложения.

Реализация функциональной схемы в методологии IDEF0 представлена на рисунках 2.1 и 2.2.

[реализовать схему idef0]

* 1. Проектирование архитектуры информационной системы

Для разработки информационной системы была выбрана микросервисная архитектура с трехуровневой моделью.

Используется паттерн API Gateway, принимающий входящие запросы от клиентского приложения и распределяющий их между микросервисами.

[дописать архитектуру – описать клиентскую часть, описать сервисы, их взаиможействие между собой, построить схему архитектуры через мермейнд, построить диаграмму последовательности]

* 1. Проектирование клиентской части информационной системы

Для реализации клиентской части веб-приложения были использованы следующие технологии: JavaScript, React, Tailwind CSS.

React представляет собой библиотеку JS для создания пользовательских интерфейсов. Он реализует компонентный подход для использования шаблонных компонентов и позволяет обновлять интерфейс без необходимости полной перезагрузки страницы. Это повышает производительность, удобство работы и снижает время отклика и затраты на разработку.

Tailwind CSS — это CSS-фреймворк, который позволяет быстро и гибко создавать интерфейсы. Разработка интерфейсов становится быстрой и удобной, а также легко подстраивается под нужды проекта.

Использование данных технологий позволяет обеспечить высокую производительность клиентской части, удобный и адаптивный интерфейс, а также эффективное взаимодействие с серверной частью системы.

Схема связи страниц клиентской стороны ИС представлена на рисунке 2.2.

[переписать тескт главы и нарисовать схему связи страниц системы]

* 1. Проектирование серверной части информационной системы

При разработке серверной части веб-приложения был использован структурированный подход, основанный на архитектурном паттерне MVC (Model-View-Controller). Данный метод предполагает разделение функционала на три ключевые компонента. Model (Модель) – отвечает за управление данными, их обработку и взаимодействие с системой хранения. Модель представляет классы, которые непосредственно отражают таблицы базы данных и ложатся в основу Spring Data JPA репозиториев. Это позволяет работать с хранимой информацией на уровне объектно-ориентированного программирования, избегая написания сложных SQL-запросов и обеспечивая удобное взаимодействие между моделью данных и остальными компонентами системы. View (Представление) – отвечает за данные, представляющие собой ответ серверной части, передаваемый на клиентскую сторону. В вебприложении представление обычно работает с API, передавая данные в формате JSON. Controller (Контроллер) – управляет потоком данных между моделью и представлением. В данной системе для реализации слоя контроллера используется связка «Controller-Service-Repository», обеспечивающая разделение ответственности между компонентами на обработку запросов, бизнес-логику и взаимодействие с базой данных соответственно. Для взаимодействия клиентской и серверной частей используется API Gateway, который выступает единой точкой входа и маршрутизирует запросы к соответствующим микросервисам. Такой подход позволяет анализировать входящие запросы и направлять их в нужные обработчики, что улучшает производительность и упрощает масштабирование. Данные приложения хранятся и обрабатываются с использованием PostgreSQL, что гарантирует надежность и эффективность работы с информацией. Использование JPA и Hibernate позволяет организовать 19 удобный доступ к данным, автоматизируя процесс взаимодействия между объектами приложения и таблицами базы. Использование описанных паттернов позволяет достичь разделения ответственности между компонентами. Такой подход упрощает тестирование, ускоряет процесс разработки и обеспечивает масштабируемость, что особенно важно для микросервисной архитектуры. [переписать данный тест под свои технологии]

* 1. Проектирование жизненного цикла информационной системы

[описать данную главу и сделать необходимую схему]

* 1. Проектирование схемы базы данных

Проектирование базы данных является одним из ключевых этапов для функционирование информационной системы и поддержания производительности при масштабировании.

Разработанная схема базы данных включает в себя сущности пользователей, компаний, игровых событий, магазина, транзакций и системы достижений. Такая схема позволяет гибко масштабировать систему, добавлять новые игровые механики и поддерживать персонализированный контент для различных организаций.

Для построения схемы разработанной базы данных, изображенной на рисунке 2.4, был использован dbdiagram.io [11].

[построить erd и переписать раздел под свой проект]

## **Вывод по разделу 2**

[написать вывод по разделу, примерно равный выводу 1]

ЗАКЛЮЧЕНИЕ

[написать заключение по аналогии с другими практиками]

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гвоздева Т. В., Баллод Б. А.. Проектирование информационных систем. Стандартизация: учебное пособие. - Санкт-Петербург: Лань, 2019. - 252 с.
2. Рочев К. В.. Информационные технологии. Анализ и проектирование информационных систем: учебное пособие. - Санкт-Петербург: Лань, 2019. - 128 с.
3. Вейцман В. М.. Проектирование информационных систем: учебное пособие. - Санкт- Петербург: Лань, 2019. - 316 с.
4. Остроух А. В., Суркова Н. Е.. Проектирование информационных систем: монография. - Санкт-Петербург: Лань, 2019. - 164 с.
5. Современные CASE средства проектирования систем: сайт. – URL: http://window.edu.ru/resource/616/73616/files/kulyabov-korolkova\_formal-methods.pdf (дата обращения 11.10.2024). – Текст: электронный.
6. HiHello – приложение для создания и управления электронными визитками: сайт. – URL: https://www.hihello.me (дата обращения 07.12.2024). – Текст: электронный.
7. CamCard – платформа для сканирования и хранения визиток: сайт. – URL: https://www.camcard.com (дата обращения 07.12.2024). – Текст: электронный.
8. LinkedIn – профессиональная социальная сеть с функцией обмена контактами: сайт. – URL: https://www.linkedin.com (дата обращения 07.12.2024). – Текст: электронный.
9. Типовые модели систем: сайт. – URL: http://kgau.ru/istiki/umk/pis/l37.htm (дата обращения 15.11.2024). – Текст: электронный.
10. Стандарты разработки: сайт. – URL: http://venec.ulstu.ru/lib/disk/2017/460.pdf (дата обращения 13.10.2024). – Текст: электронный.
11. Эксплуатация и сопровождение проекта: сайт. – URL: http://kgau.ru/istiki/umk/pis/l7.htm (дата обращения 09.11.2024). – Текст: электронный.
12. Тестирование и контроль программных систем: сайт. – URL: https://xreferat.com/33/2759-1-sushnost-i-osobennosti-ispol-zovaniya-instrumental-nogo-programmnogo-obespecheniya.html (дата обращения 24.11.2024). – Текст: электронный.
13. Контроль и корректировка кода: сайт. – URL: https://studfile.net/preview/2790134 (дата обращения 16.10.2024). – Текст: электронный.
14. Проектирование ER-диаграмм: сайт. – URL: http://citforum.ru/cfin/prcorpsys/infsistpr\_09.shtml (дата обращения 12.11.2024). – Текст: электронный.
15. Реляционные СУБД: сайт. – URL: https://compress.ru/article.aspx?id=10082 (дата обращения 11.10.2024). – Текст: электронный.