|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №5-8**

по дисциплине «Системная и программная инженерия»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-20-22 | Шумахер М.Е.  Муллаянов А.И.  Исаков Л.А.  Туганов Н.А.  Фридрих А.В. |
| **Проверил:**  Ст. преподаватель | Михайлова Е.К. |

2025 г.

СОДЕРЖАНИЕ

[Практическая работа №5 3](#_Toc194263270)

[Практическая работа №6 4](#_Toc194263271)

[Практическая работа №7 8](#_Toc194263272)

[Практическая работа №8 11](#_Toc194263273)

**Практическая работа №5**

На рисунке 5.1 представлена диаграмма классов.

Рисунок 5.1 – Диаграмма классов

На рисунке 5.2 представлена диаграмма объектов.

Рисунок 5.2 – Диаграмма объектов

Нотация IDEF0.

На рисунке 5.3 представлена контекстная диаграмма в нотации IDEF0.

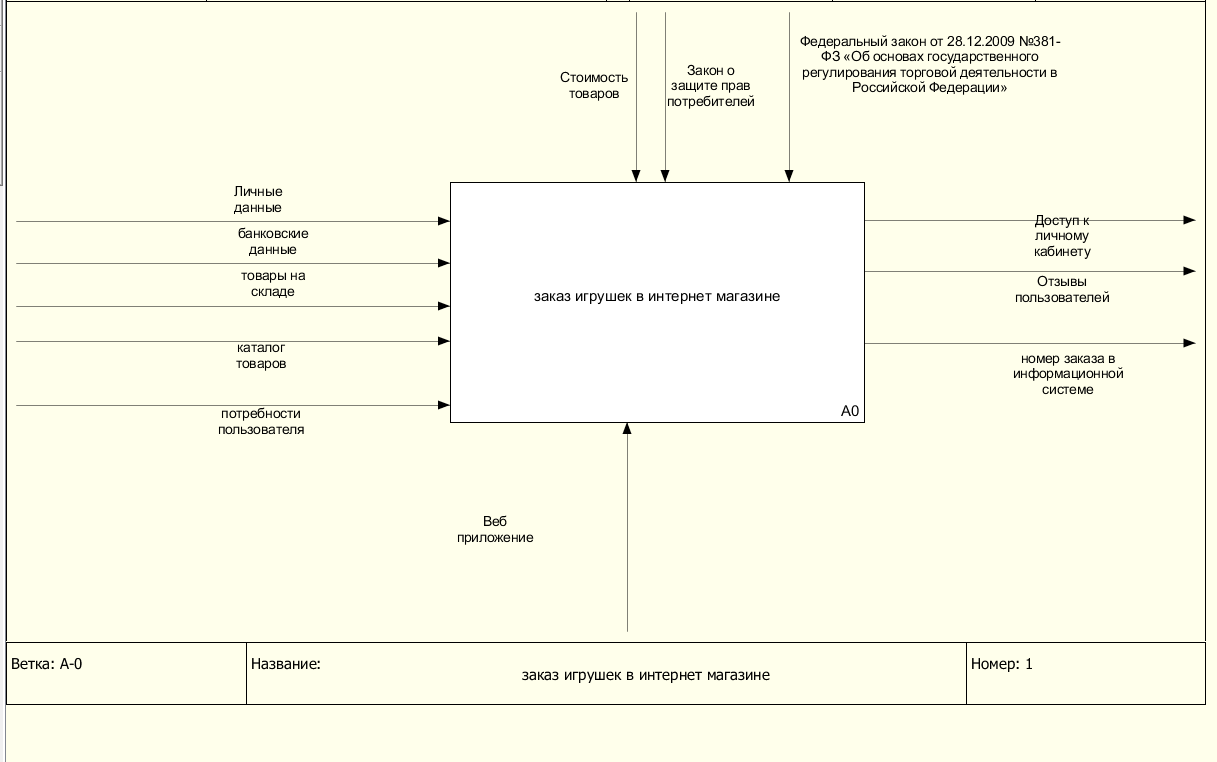


Рисунок 5.3 – Контекстная диаграмма

На рисунке 5.4 представлена декомпозиция диаграммы.

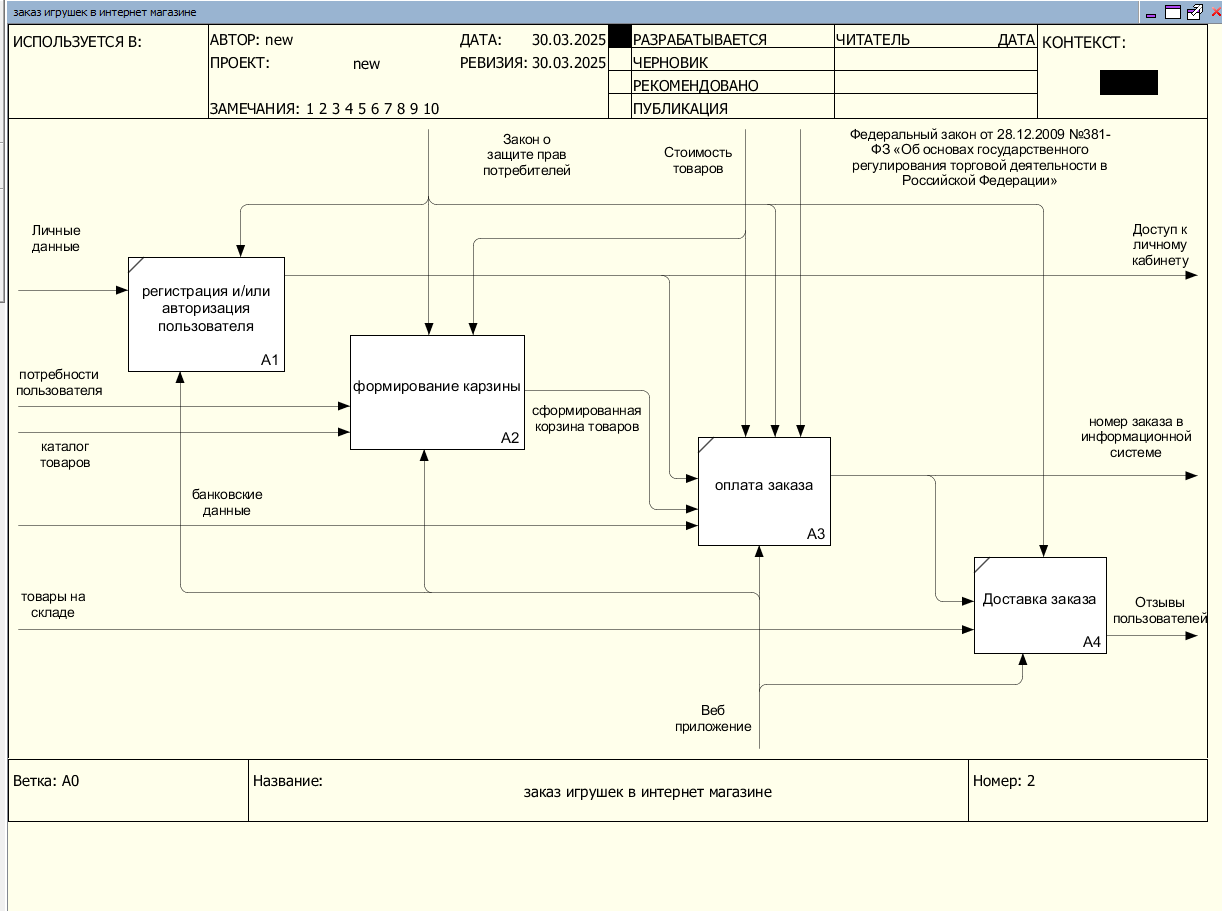


Рисунок 5.4 – Декомпозиция диаграммы

На рисунке 5.5 представлена декомпозиция функционального блока “Формирование корзины”.

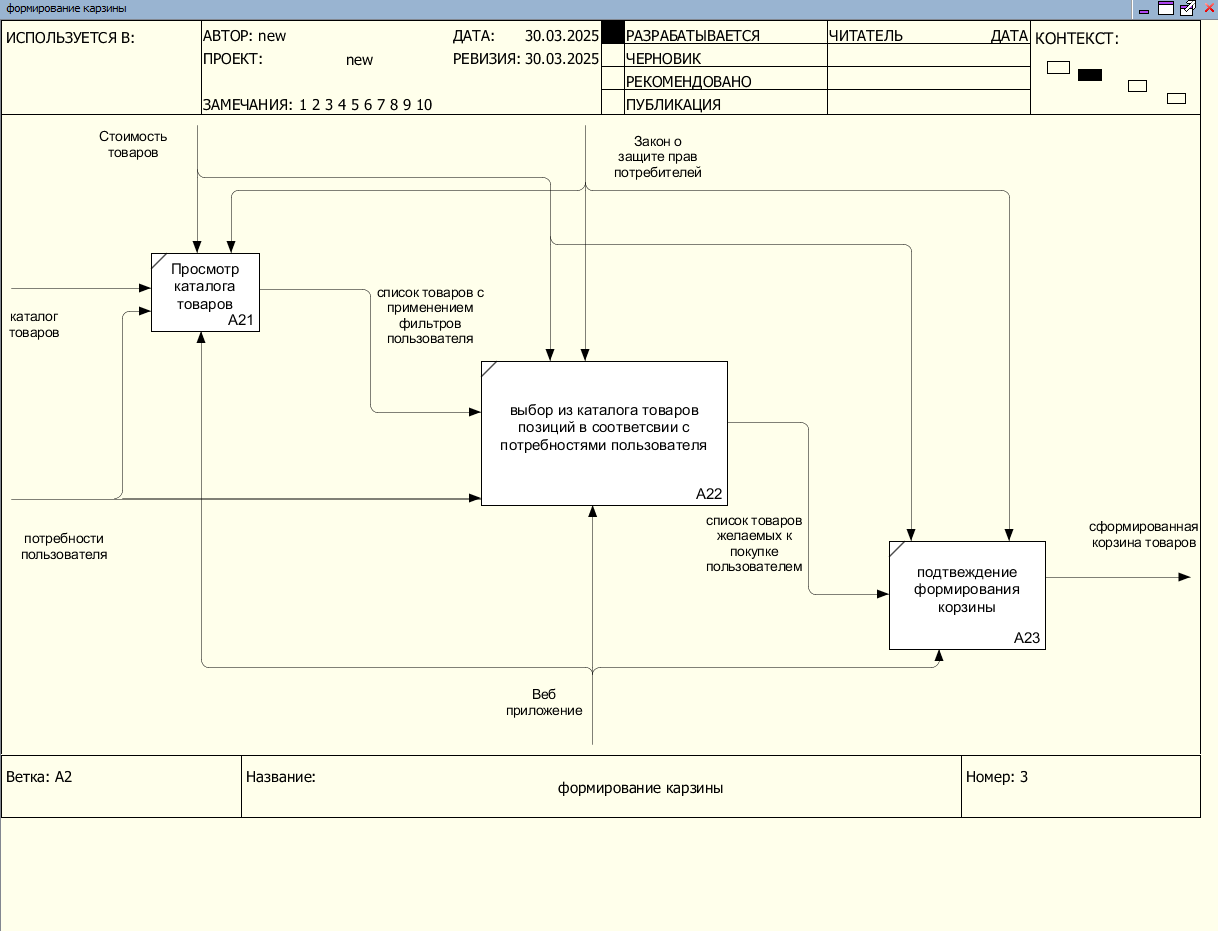


Рисунок 5.5 – Декомпозиция функционального блока “Формирование корзины”

**Практическая работа №6**

DFD — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе, источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Диаграмма потоков данных (data flow diagram, DFD) — один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.

На рисунке 6.1 представлен контекстный уровень DFD диаграммы.

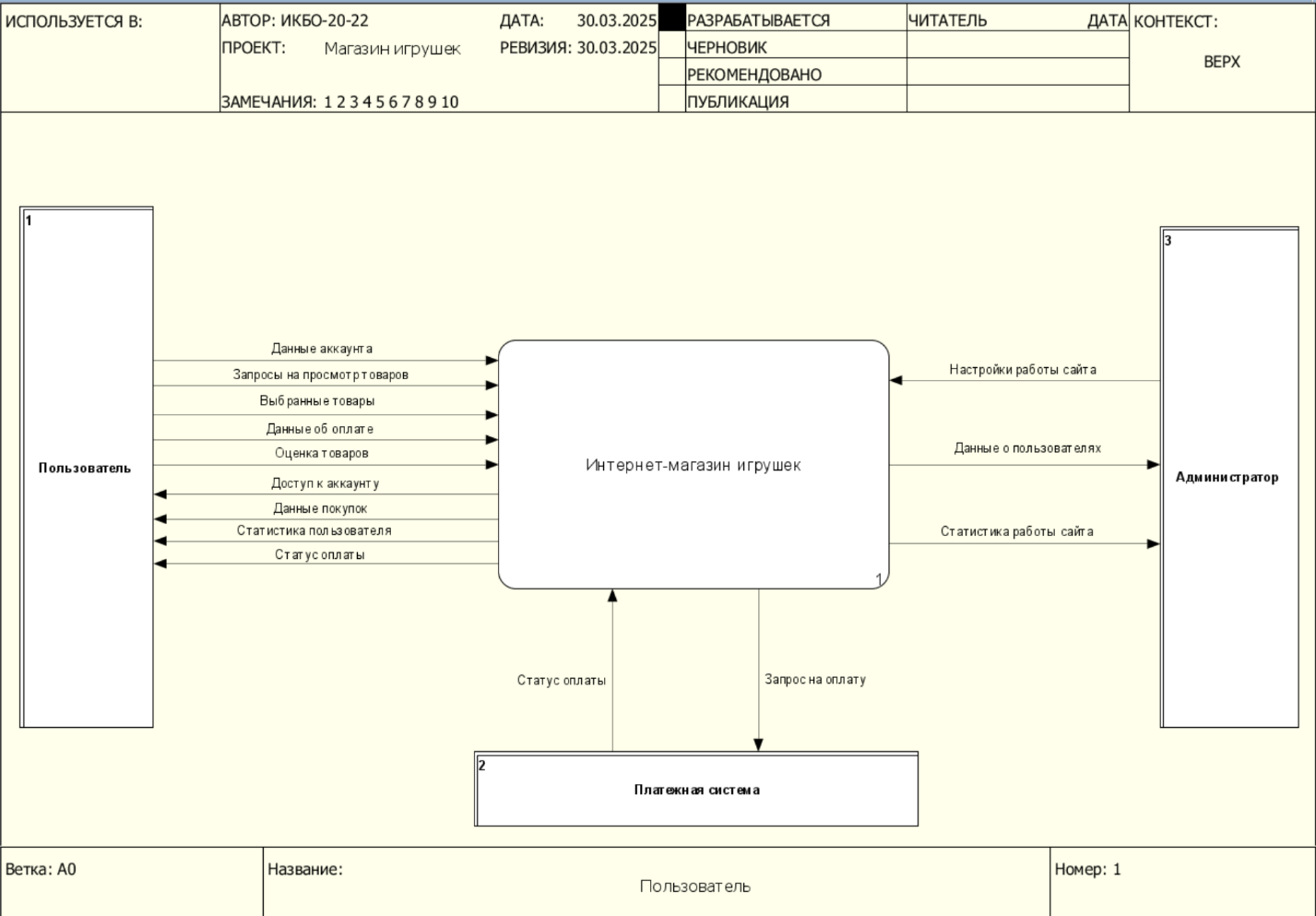


Рисунок 6.1 – Контекстный уровень DFD диаграммы

На рисунке 6.2 показана декомпозиция контекстного уровня DFD диаграммы.

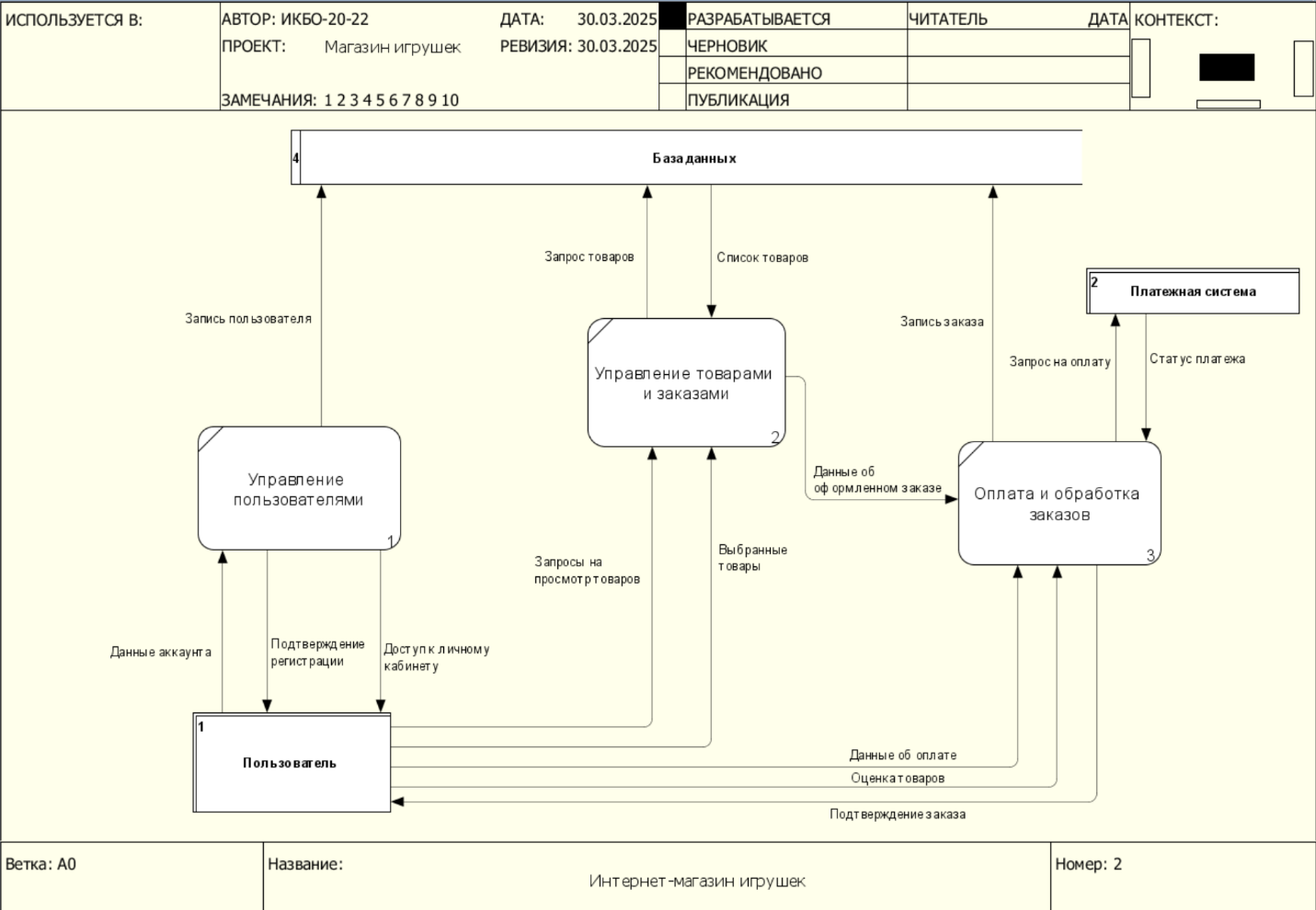


Рисунок 6.2 – Декомпозиция контекстного уровня DFD диаграммы

На рисунке 6.3 показана декомпозиция функционального блока “Управление товарами и заказами”.

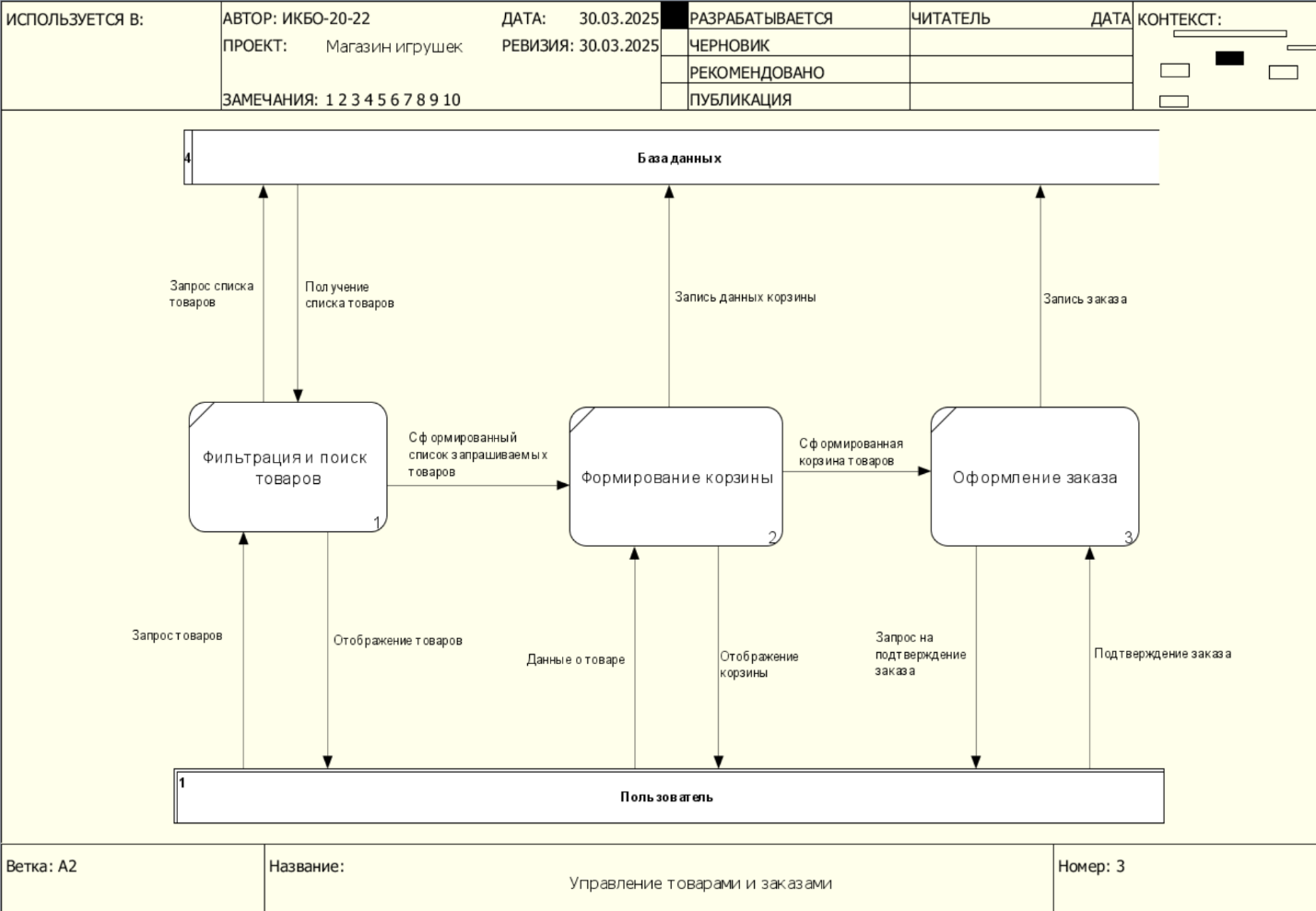


Рисунок 6.3 – Декомпозиция функционального блока “Управление товарами и заказами”

A0 - Интернет-магазин игрушек

Этот блок представляет собой всю систему интернет-магазина игрушек, включающую функциональность для пользователей и администраторов. В рамках него пользователи могут взаимодействовать с товарами, заказывать их, оплачивать, просматривать историю заказов и управлять своим аккаунтом. Администраторы имеют возможность управлять пользователями и товарами. Декомпозиция этого блока включает все основные функциональные элементы, обеспечивающие работу магазина

A1.1 - Управление пользователями

В этом блоке происходит регистрация новых пользователей, проверка введенных ими данных, отправка подтверждающих писем и создание учетной записи. При авторизации система сверяет введенный логин и пароль с базой данных, а затем создает сессию, позволяя пользователю оставаться в системе. В личном кабинете можно редактировать данные, изменять пароль, просматривать историю заказов. Администратор может просматривать список пользователей, блокировать учетные записи, менять их роли, если необходимо

A1.2 - Управление товарами и заказами

Администратор добавляет новые товары, заполняя название, описание, цену, категорию, изображения, характеристики. После публикации товар становится доступным пользователям для просмотра и покупки. Если товар нужно изменить, администратор может обновить описание, цену или статус наличия. При необходимости товар удаляется из каталога. Заказы пользователей также находятся под контролем администратора. Как только покупатель оформляет заказ, система фиксирует его данные, и администратор может просматривать все заказы, изменять их статус, уточнять детали, если это необходимо. Когда товар заканчивается, система обновляет его статус, скрывая его из доступных для покупки позиций

A1.3 - Оплата и обработка заказов

Когда пользователь выбирает товары и оформляет заказ, система запрашивает у него адрес доставки, контактные данные и желаемый способ получения товара. Затем предлагается выбор способа оплаты, который может включать оплату банковской картой, электронными деньгами или наличными при получении. Если выбирается онлайн-оплата, система перенаправляет пользователя на платежный шлюз, где происходит обработка транзакции. В случае успешной оплаты заказ регистрируется в базе данных и покупатель получает подтверждение. Система также отправляет уведомления о статусе заказа, включая подтверждение оформления, смену статуса на «в обработке», «отправлено», «доставлено»

A2.1 - Фильтрация и поиск товаров

Для удобного выбора товаров пользователю предоставляется возможность фильтрации и поиска. Можно выбрать интересующую категорию, отфильтровать товары по брендам.

A2.2 - Формирование корзины

Пользователь может добавлять товары в корзину, указывая нужное количество. Все добавленные товары остаются в корзине до момента оформления заказа, даже если пользователь выходит из системы. В корзине отображаются фотографии товаров, их цена, сумма заказа. Пользователь может изменять количество товаров или удалять ненужные позиции. Если он хочет отложить заказ, корзина сохраняется, и при следующем входе товары остаются доступными

A2.3 - Оформление заказа

Когда пользователь готов к покупке, он переходит к оформлению заказа. В этот момент система запрашивает его контактные данные, адрес доставки и предпочтительный способ получения товара. Пользователь выбирает удобный метод оплаты и подтверждает покупку. После этого заказ регистрируется в системе, а покупателю отправляется подтверждение. В личном кабинете он может следить за изменением статуса заказа, начиная с момента оформления и до доставки. Если заказ требует доработки, например, отсутствует товар или нужна замена, администратор может связаться с покупателем для уточнения деталей

На рисунка 6.4 представлена нормализованная логическая модель базы данных.

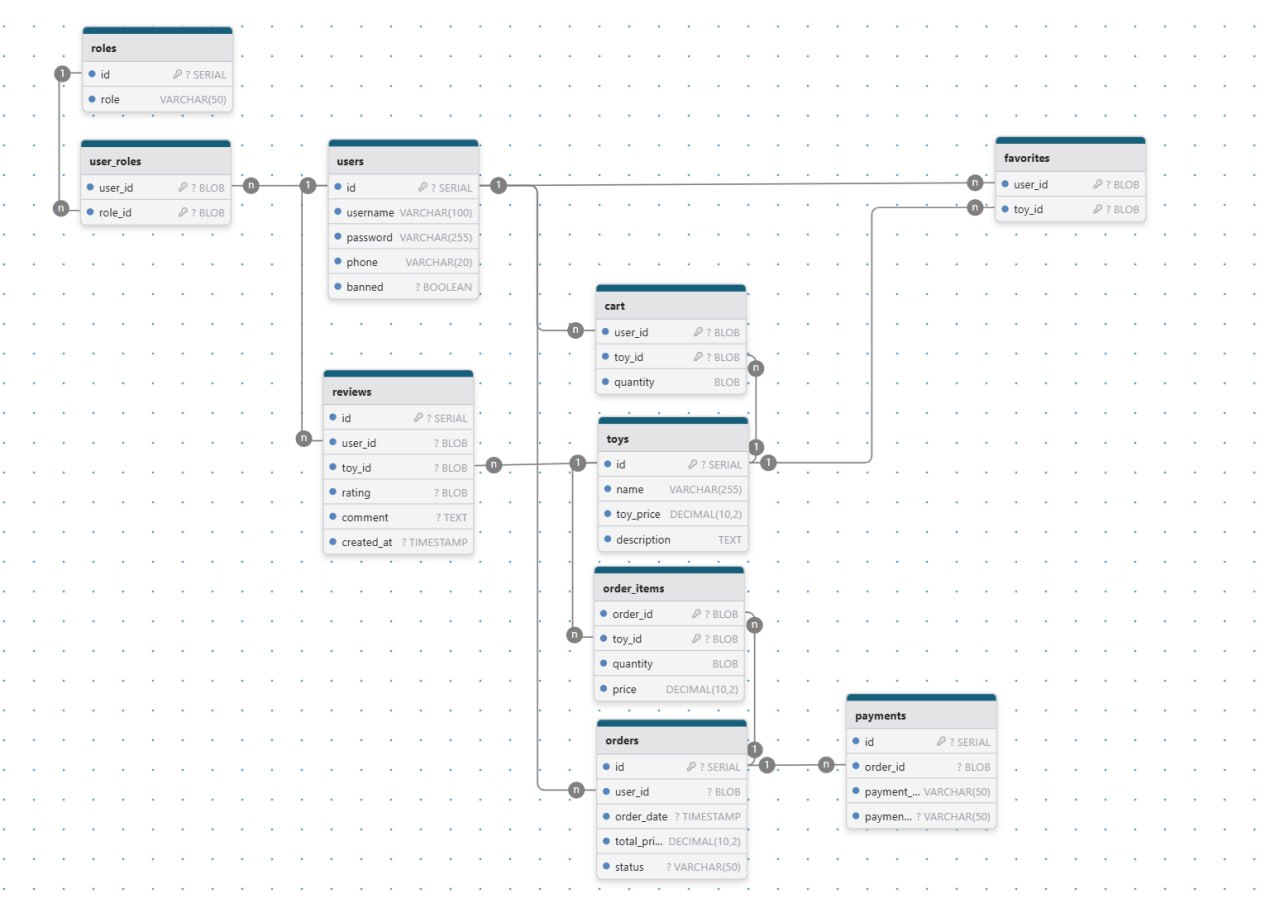


Рисунок 6.4 – Логическая модель базы данных

**Практическая работа №7**

**Используемое ПО и технологии**

В рамках данного проекта для разработки был выбран стек технологий, который обеспечивает высокую производительность, масштабируемость и удобство работы. Для серверной части приложения был использован Spring Boot. Этот фреймворк минимизирует конфигурацию, предоставляет готовую инфраструктуру для работы с базами данных и безопасностью, что ускоряет разработку и упрощает создание масштабируемых приложений.

Для разработки интерфейса был выбран Thymeleaf, который интегрируется с Spring и позволяет динамично генерировать HTML-страницы на серверной стороне. Это позволяет сохранять чистоту кода и удобно работать с данными, подставляемыми в шаблоны. Такой подход упрощает взаимодействие с пользователем и позволяет эффективно управлять содержимым веб-страниц.

Для хранения данных в проекте использовалась база данных PostgreSQL. Эта реляционная система управления базами данных обеспечивает надежность, высокую производительность и поддержку сложных запросов. PostgreSQL является популярным решением для многих веб-приложений благодаря своей стабильности и доступности, а также широкому сообществу и поддержке.

Процесс разработки был значительно ускорен с помощью использования Lombok, который автоматизирует генерацию стандартных методов для классов, таких как геттеры, сеттеры и конструкторы. Это позволило сократить количество шаблонного кода и сосредоточиться на реализации бизнес-логики.

Для контроля версий был выбран Git, который является неотъемлемой частью разработки. Этот инструмент позволяет отслеживать изменения в коде, управлять ветками и работать над проектом совместно с командой, что особенно важно при масштабных проектах.

Для сборки проекта был использован Gradle. Этот инструмент автоматизирует процесс сборки, управления зависимостями и создания артефактов, что значительно упрощает процесс разработки и деплоя приложения. Gradle также поддерживает гибкость настройки и расширяемость, что делает его идеальным для использования в Java-проектах.

**Архитектура системы**

Монолитная архитектура имеет несколько преимуществ, таких как простота разработки и развертывания, так как все компоненты находятся в одном приложении. Это упрощает тестирование и отладку, ведь весь код доступен в одном месте. Также накладные расходы на инфраструктуру минимальны, так как приложение разворачивается как единое целое.

Однако с ростом приложения возникают сложности с масштабированием. При увеличении функциональности проект становится более сложным в обслуживании и поддержке. Добавление новых функций может повлиять на другие части системы, что затрудняет работу с кодом. Также деплой и обновления становятся сложными, особенно при крупных изменениях, которые затрагивают всё приложение.

Микросервисная архитектура позволяет легче масштабировать приложение и разрабатывать независимые сервисы, которые можно обновлять без воздействия на другие части системы. Она позволяет улучшить гибкость и делает систему более устойчивой к сбоям. Однако для реализации микросервисов требуются дополнительные ресурсы и более сложная инфраструктура. Множество сервисов требует аккуратного управления зависимостями, развертывания и мониторинга, что делает систему более сложной в разработке и поддержке.

В данном случае выбор в сторону монолита обоснован его простотой в разработке и развертывании. Для интернет-магазина, где основное внимание уделяется функциональности и скорости реализации, монолитная архитектура будет оптимальным решением, так как она позволит быстрее запустить проект с минимальными затратами на инфраструктуру.

На рисунке 7.1 представлена архитектурная диаграмма системы.

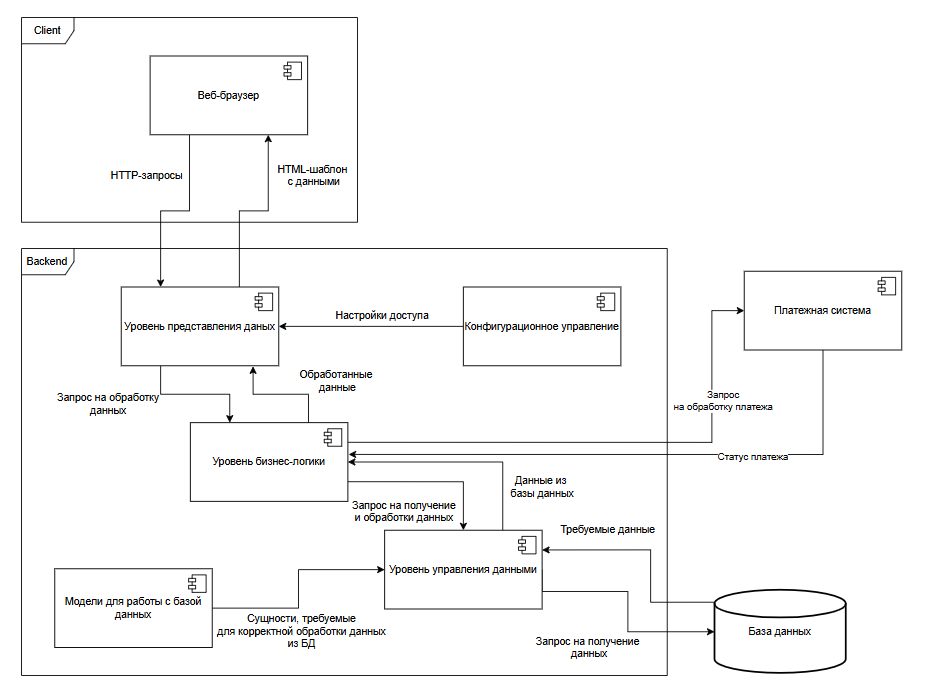


Рисунок 7.1 – Архитектурная диаграмма системы

Таблица 1 – Матрица требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Суть** | **Автор** | **Ссылки** | **Критерий проверки** | **Компоненты архитектуры** |
| 1 | Доступный пользователям веб-интерфейс приложения “Магазин игрушек” | | | | | |
| 1.1 | Регистрация пользователя | Пользователь должен иметь возможность зарегистрироваться указав email, пароль, и дополнительную информации | Муллаянов А.И. | https://habr.com/ru/articles/728072/ | Регистрация нового пользователя | Ввод данных в веб-интерфейсе, обработка в сервисе управления пользователями, хранение в БД пользователей |
| 1.2 | Авторизация пользователя | Пользователь должен иметь возможность войти в личный кабинет с использованием данных введенных при регистрации | Муллаянов А.И. | https://habr.com/ru/articles/728072/ | Вход пользователя в аккаунт | Ввод данных в веб-интерфейсе, обработка в сервисе авторизации, проверка в БД пользователей |
| 1.3 | Зайти в личный кабинет | Пользователь должен иметь возможность зайти в личный кабинет, где он может просмотреть свои прошлые покупки и информацию о доставках | Муллаянов А.И. | https://habr.com/ru/articles/728072/ | Открытие личного кабинета пользователя | Ввод данных в веб-интерфейсе, обработка в сервисе авторизации, запрос к сервису управления заказами, получение данных из БД заказов и БД пользователей |
| 1.4 | Открыть каталог товаров | Пользователь должен иметь доступ к каталогу товаров для добавления позиций в корзину и добавления товаров в избранное | Муллаянов А.И. | https://habr.com/ru/companies/trinion/articles/286188/ | Открытие каталога товаров | Отображение в веб-интерфейсе, обработка в сервисе каталога товаров, запрос к БД товаров, взаимодействие с сервисом управления корзиной и сервисом управления избранным |
| 1.5 | Формирование корзины товаров | Пользователь должен иметь возможность добавления товаров в корзину с указанием количества продукта | Муллаянов А.И. | https://habr.com/ru/companies/trinion/articles/286188/ | Сформированная корзина товаров с указанием общей стоимости товаров | Ввод данных в веб-интерфейсе, обработка в сервисе управления корзиной, хранение в БД корзины |
| 1.6 | Оплата сформированной корзины товаров | Пользователь должен иметь возможность выбора способа оплаты заказа | Муллаянов А.И. | https://habr.com/ru/companies/trinion/articles/286188/ | Подтверждение осуществления заказа, и отображения заказа в истории покупок | Ввод данных в веб-интерфейсе, обработка в сервисе оплаты, интеграция с платёжным шлюзом, хранение в БД заказов |
| 1.7 | Просмотр истории заказов | Пользователь должен иметь возможность ознакомится с историей своих заказов с указанием товаров, их количества и стоимости | Фридрих А.В. | https://habr.com/ru/companies/trinion/articles/286188/ | Отображение в личном кабинете истории заказов | Отображение в веб-интерфейсе, запрос к сервису управления заказами, получение данных из БД заказов |
| 1.8 | Добавление товара в избранное | Пользователь должен иметь возможность сохранения понравившегося товара в избранное | Фридрих А.В. | https://habr.com/ru/companies/trinion/articles/286188/ | Отображение в личном кабинете списка понравившихся товаров | Ввод данных в веб-интерфейсе, обработка в сервисе управления избранным, хранение в БД избранного |
| 1.9 | Фильтр товаров по категориям | Пользователь должен иметь возможность отображения товаров по категориям | Фридрих А.В. | https://habr.com/ru/companies/trinion/articles/286188/ | Страница отображения списка доступных позиций отфильтрованных по выбранной категории | Ввод данных в веб-интерфейсе, обработка в сервисе каталога товаров, запрос к БД товаров |
| 1.10 | Оценка товара | Пользователь должен иметь возможность оценки заказанного товара | Муллаянов А.И. | https://habr.com/ru/companies/trinion/articles/286188/ | Отображение популярных товаров в верху списка | Ввод данных в веб-интерфейсе, обработка в сервисе отзывов, хранение в БД отзывов |
| 1.11 | Кроссбраузерность | Веб-интерфейс должен одинакового функционировать во всех популярных браузерах | Муллаянов А.И. | https://habr.com/ru/articles/299846/ | Правильное отображение всех элементов в браузерах: Google Chrome 117, Mozilla Firefox 118, Opera 103, Yandex.Браузер 23.9, Microsoft Edge 117, Safari 17. | Реализация в веб-интерфейсе, тестирование на стороне клиента |
| 1.12 | Кроссплатформенность | Веб-интерфейс должен быть адаптирован под мобильные устройства (Samsung A51) | Муллаянов А.И. | https://habr.com/ru/articles/299846/ | Отображение всех элементов на мобильном устройстве в соответствии с обговоренным дизайном | Реализация в веб-интерфейсе (адаптивный дизайн), тестирование на стороне клиента |
| 1.13 | Лингвистическая универсальность | Приложение должно поддерживать русский язык, при необходимости английский. | Шумахер М.Е. | https://habr.com/ru/companies/trinion/articles/286188/ | Выбор языка веб-интерфейса | Реализация в веб-интерфейсе, обработка в сервисе локализации, хранение настроек в БД пользователей |
| 1.14 | Соответствие даты и времени региональным стандартам | Формат даты и времени должен соответствовать региональным стандартам пользователей (например, DD.MM.YYYY для России). | Шумахер М.Е. | https://habr.com/ru/articles/299846/ | Корректное отображение даты и времени | Реализация в веб-интерфейсе, форматирование в сервисе локализации |
| 1.15 | Локализация элементов | Локализация текстовых элементов интерфейса, включая кнопки, меню и уведомления. | Шумахер М.Е. | https://habr.com/ru/articles/299846/ | Отображение элементов в соответствии с региональным языком | Реализация в веб-интерфейсе, обработка в сервисе локализации |
| 2. | Панель администратора | | | | | |
| 2.1 | Просмотр аккаунтов клиентов | Администратор должен иметь доступ к списку зарегистрированных пользователей с возможностью просмотра их активности | Шумахер М.Е. | https://habr.com/ru/companies/sberdevices/articles/740894/ | Страница списка всех пользователей и истории активности одного из клиентов | Отображение в панели администратора, запрос к сервису управления пользователями, данные из БД пользователей |
| 2.2 | Блокировка аккаунта клиента | Администратор должен иметь возможность блокировки зарегистрированного пользователя за нарушение правил платформы | Шумахер М.Е. | https://habr.com/ru/companies/sberdevices/articles/740894/ | Блокировка одного из пользователей | Ввод данных в панели администратора, обработка в сервисе управления пользователями, обновление в БД пользователей |
| 2.3 | Размещение новых позиций | Администратор должен иметь возможность добавления новых товаров в ассортимент | Шумахер М.Е. | https://habr.com/ru/articles/299846/ | Добавление нового товара | Ввод данных в панели администратора, обработка в сервисе каталога товаров, хранение в БД товаров |
| 3 | Серверная часть приложения | | | | | |
| 3.1 | Скорость загрузки главной страницы | Скорость загрузки главной страницы веб-интерфейса должна составлять не более 3 секунд | Фридрих А.В. | https://habr.com/ru/articles/299846/ | Загрузка главной страницы за 3 секунды | Обработка в сервисе каталога товаров, кэширование, запрос к БД товаров |
| 3.2 | Скорость перемещения на страницу товара | Скорость открытия страницы товара должна составлять не более 3 секунд | Фридрих А.В. | https://habr.com/ru/articles/299846/ | Открытие страницы товара за 3 секунды | Обработка в сервисе каталога товаров, запрос к БД товаров |
| 3.3 | Скорость добавления товара в корзину | Добавление товара в корзину должно составлять не более 1 секунды | Фридрих А.В. | https://habr.com/ru/articles/299846/ | Добавление товара в корзину за 1 секунду | Обработка в сервисе управления корзиной, обновление в БД корзины |
| 3.4 | Трафик пользователей | Платформа должна поддерживать одновременное подключение как минимум 1000 пользователей | Фридрих А.В. | https://habr.com/ru/companies/sberdevices/articles/740894/ | Одновременное подключение 1000 пользователей к платформе | Балансировка нагрузки на серверной инфраструктуре, работа всех сервисов с БД PostgreSQL |
| 3.5 | Оптимизация базы данных | База данных должна быть оптимизирована для быстрого поиска и фильтрации товаров по категориям с помощью индексов. | Муллаянов А.И. | https://habr.com/ru/articles/349910/ | Отклик базы данных на запрос за 0.5 секунды | Оптимизация в БД товаров (индексы, кэширование), взаимодействие через серверные сервисы |
| 3.6 | Масштабируемость | Платформа должна иметь возможность расширения до 10 000 активных пользователей одновременно- | Фридрих А.В. | https://habr.com/ru/articles/349910/ | Приведенные разработчиками гарантии возможности масштабирования | Использование серверной инфраструктуры с контейнеризацией, масштабируемая БД PostgreSQL |
| 3.7 | Рабочее время | Время безотказной работы сервиса должно быть не менее 99,5% | Фридрих А.В. | https://habr.com/ru/articles/349910/ | Зафиксированное время работы сервиса по окончании работы в первый год | Мониторинг серверной инфраструктуры, устойчивость всех сервисов |
| 3.8 | Восстановление системы | Приложение должно корректно восстанавливаться в течении 5 минут после сбоя, сохраняя данные о корзине и избранном | Фридрих А.В. | https://habr.com/ru/companies/ruvds/articles/678364/ | Восстановление системы после сбоя | Резервное копирование на серверной инфраструктуре, восстановление из БД корзины |
| 3.9 | Резервное копирование | Регулярное резервное копирование данных для предотвращения потери информации каждую пятницу в 20:01 | Муллаянов А.И. | https://habr.com/ru/companies/ruvds/articles/678364/ | Корректное восстановление системы из резервного копирования | Реализация в сервисе резервного копирования, хранение копий в БД PostgreSQL |
| 3.10 | Защищенность соединения | Данные пользователей должны передаваться по защищенному соединению (HTTPS, TLS 1.2+). | Муллаянов А.И. | https://habr.com/ru/companies/vdsina/articles/503772/ | Отчет разработчиков о использованной технологии | Настройка HTTPS на серверной инфраструктуре, шифрование в сервисах |
| 3.11 | Защищенность данных | Использование протокола OAuth 2.0 для авторизации. | Муллаянов А.И. | https://habr.com/ru/companies/vdsina/articles/503772/ | Отчет разработчиков о использованной технологии | Реализация в сервисе авторизации, интеграция с OAuth 2.0 |
| 3.12 | Защищенность аккаунтов пользователей | Пароли пользователей должны храниться в зашифрованном виде с использованием алгоритма bcrypt. | Муллаянов А.И. | https://habr.com/ru/companies/vdsina/articles/503772/ | Отчет разработчиков о использованной технологии | Обработка в сервисе авторизации, хранение зашифрованных данных в БД пользователей |
| 3.13 | Операционная система сервера | Сервер должен быть развернут на машине с ОС Linux. | Муллаянов А.И. | https://habr.com/ru/companies/vdsina/articles/503772/ | Отчет разработчиков о использованной технологии | Развёртывание на серверной инфраструктуре с ОС Linux, поддержка всех сервисов и БД PostgreSQL |

**Практическая работа №8**

Для разработки технического задания (ТЗ) на веб-приложение «Магазин игрушек» был выбран ГОСТ 19.201-78. Этот стандарт предназначен для создания технических заданий на разработку программного обеспечения и программных изделий, что полностью соответствует нашему проекту. Вот основные причины выбора данного ГОСТа:

1. Соответствие типу проекта – ГОСТ 19.201-78 ориентирован на программные продукты, включая веб-приложения, что делает его идеальным для нашего проекта. В отличие от ГОСТ 34.602-2020, который предназначен для автоматизированных систем (АС) и включает избыточные для нашего случая разделы (например, требования к патентной чистоте или транспортабельности), ГОСТ 19.201-78 фокусируется на ключевых аспектах разработки ПО.
2. Структура и содержание – стандарт включает разделы, которые полностью покрывают потребности:

* Введение: Краткое описание области применения и объекта,
* Основания для разработки: Документы и исходные данные,
* Назначение разработки: Функциональное и эксплуатационное назначение,
* Требования к программе: Функциональные характеристики, надежность, условия эксплуатации, совместимость и т.д.,
* Технико-экономические показатели: Эффективность и преимущества разработки,
* Стадии и этапы разработки: План работ и сроки,
* Порядок контроля и приемки: Критерии проверки.

1. Гибкость – ГОСТ 19.201-78 допускает уточнение содержания разделов, добавление новых или объединение существующих, что важно для адаптации под специфику проекта. Например, мы можем детализировать требования к интерфейсу или безопасности, не нарушая стандарт.
2. Акцент на программное обеспечение – В отличие от ГОСТ 34.602-2020, который охватывает широкий спектр систем (включая аппаратные компоненты), ГОСТ 19.201-78 фокусируется исключительно на ПО.

ГОСТ 19.201-78 был выбран благодаря его специализации на ПО, четкой структуре и гибкости. Он позволяет избежать избыточности, характерной для ГОСТ 34.602-2020, и сосредоточиться на ключевых аспектах веб-приложения. Это обеспечивает ясность, полноту и удобство использования ТЗ для всех участников проекта.

Техническое задание:

**1 ВВЕДЕНИЕ**

**1.1 Наименование программы**

Наименование программы – «Мир игрушек».

**1.2 Краткая характеристика области применения**

Система «Мир игрушек» предназначена для организации онлайн-продажи игрушек. Платформа предоставляет покупателям возможность просматривать каталог товаров, формировать корзину, оформлять заказы и отслеживать их статус. Для администраторов предусмотрены функции управления товарами, пользователями и заказами. Система включает две основные роли. Покупатель – просмотр товаров, оформление заказов, управление личным кабинетом. Администратор – управление ассортиментом, обработка заказов, контроль пользователей.

**2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**

Основанием для разработки является Приказ ректора РТУ МИРЭА №123 от 15.01.2025. Приказ утвержден Директором Института информационных технологий РТУ МИРЭА.

Наименование темы разработки: «Создание веб-приложения для онлайн-продажи игрушек».

Шифр темы: «ToysShop-01».

**3 НАЗНАЧЕНИЕ РАЗРАБОТКИ**

**3.1 Функциональное назначение**

Для покупателей:

* Просмотр каталога товаров с фильтрами (цена, категория, возрастная группа),
* Добавление товаров в корзину и избранное,
* Оформление заказов с выбором способа оплаты (карта, электронные деньги),
* Просмотр истории заказов и статуса доставки.

Для администраторов:

* Управление товарами (добавление, редактирование, удаление),
* Просмотр и блокировка пользователей.
  1. **Эксплуатационное назначение**
* Круглосуточный доступ для покупателей через веб-браузер или мобильное устройство.
* Работа администраторов через веб-интерфейс с ограниченным доступом.

**4 ТРЕБОВАНИЕ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ**

**4.1 Требования к функциональным характеристикам**

**4.1.1 Требования к составу выполняемых функций**

После запуска приложения пользователю отображается непосредственно каталог товаров, и кнопка регистрации или авторизации на сайте. В системе существует два типа пользователей: покупатель и администратор. Программа определяет тип пользователя и предоставляет соответствующий интерфейс и/или дополнительные возможности.

Для покупателя программа предоставляет следующие возможности:

* + Просмотр каталога товаров: пользователь может просматривать товары с возможностью фильтрации по категориям, цене и другим параметрам,
  + Формирование корзины: возможность добавлять товары в корзину с указанием количества,
  + Оформление заказов: выбор способа оплаты (карта, электронные деньги) и доставки,
  + Просмотр истории заказов: информация о текущих и завершенных заказах,
  + Оставление отзывов: возможность оценивать товары и оставлять комментарии,
  + Управление личным кабинетом: редактирование персональных данных, смена пароля,
  + Добавление товаров в избранное: создание списка предпочтительных товаров,
  + Получение уведомлений: информация о статусе заказов, акциях и новых поступлениях.

Для администратора программа предоставляет все функции покупателя, а также:

* + Добавление новых товаров с описанием и изображениями,
  + Редактирование информации о товарах,
  + Управление наличием и ценами,
  + Просмотр списка всех заказов,
  + Изменение статусов заказов,
  + Отмена заказов,
  + Просмотр списка зарегистрированных пользователей,
  + Блокировка/разблокировка аккаунтов,

**4.1.2 Требования к организации входных и выходных данных**

Данные о товарах, заказах и пользователях хранятся в реляционной базе данных PostgreSQL. Система обеспечивает разграничение прав доступа:

Покупатель имеет права только на чтение данных о товарах и запись своих заказов. Администратор имеет полные права на чтение и запись всех данных.

Валидация данных выполняется как на стороне клиента, так и на сервере. Основные требования к данным:

* + Дата и время должны быть в формате "ДД.ММ.ГГГГ ЧЧ:ММ",
  + Название товара - не более 100 символов,
  + Описание товара - не более 1000 символов,
  + Цена - числовое значение с двумя знаками после запятой,
  + Количество товара - целое положительное число.

**4.1.3 Требования к временным характеристикам**

После внесения изменений администратором.

* + Обновление информации о товарах должно отображаться у покупателей не позднее чем через 3 секунды,
  + Изменение статуса заказа должно отображаться у покупателя не позднее чем через 5 секунд,
  + Загрузка страницы каталога не должна превышать 2 секунд,
  + Обработка запроса на добавление товара в корзину - не более 1 секунды.

**4.2 Требования к надежности**

Вероятность безотказной работы системы должна составлять не менее 99,5% в год при условии исправности сетевого оборудования и стабильного интернет-соединения.

**4.2.1 Требования к обеспечению надежного (устойчивого) функционирования программы**

Для обеспечения надежной работы необходимо:

* + Организовать бесперебойное питание серверного оборудования,
  + Использовать лицензионное программное обеспечение,
  + Регулярно выполнять резервное копирование данных (не реже 1 раза в сутки),
  + Обеспечить защиту от вирусов и вредоносного ПО,
  + Реализовать систему мониторинга работоспособности.

**4.2.2 Время восстановления после отказа**

* + При программных сбоях - не более 5 минут,
  + При аппаратных сбоях - не более 2 часов,
  + В случае потери данных - восстановление из резервной копии не более 1 часа.

**4.2.3 Отказы из-за некорректных действий оператора**

Возможные отказы системы могут быть связаны с некорректными действиями оператора при работе с приложением, особенно если он выполняет операции, которые не соответствуют его правам или не предусмотрены в процессе эксплуатации. Чтобы избежать подобных ситуаций система должна:

* + Предотвращать выполнение операций, не соответствующих правам пользователя,
  + Обеспечивать откат транзакций при ошибках,
  + Ведение журнала всех значимых действий пользователей.

**4.3 Условия эксплуатации**

* + Клиентская часть: веб-браузеры Google Chrome 117, Mozilla Firefox 118, Opera 103, Yandex.Браузер 23.9, Microsoft Edge 117, Safari 17,
  + Серверная часть: Linux-сервер с установленным ПО (Java, PostgreSQL, Nginx),
  + Требования к сети: стабильное интернет-соединение со скоростью не менее 10 Мбит/с.

**4.3.1 Климатические условия эксплуатации**

Специальные климатические требования отсутствуют.

**4.3.2 Требования к видам обслуживания**

Программное обеспечение не требует постоянного обслуживания со стороны пользователей. Техническое обслуживание включает в себя периодическое обновление системы, исправление ошибок, а также резервное копирование данных.

**4.3.3 Требования к численности и квалификации персонала**

На этапе установки и настройки системы требуется участие системного администратора, который отвечает за развертывание серверной части, настройку базы данных и обеспечение безопасности системы.

**4.4 Требования к составу и параметрам технических средств**

Система должна функционировать на следующих аппаратных и программных платформах:

Клиентская часть (для покупателей и администраторов) – веб-браузеры Google Chrome 117, Mozilla Firefox 118, Opera 103, Yandex.Браузер 23.9, Microsoft Edge 117, Safari 17, мобильные устройства – поддержка мобильных браузеров на Android OS и IOS.

Серверная инфраструктура:

* + Аппаратные требования:
  + Процессор: 8 ядер, тактовая частота 2.4 ГГц и выше,
  + Оперативная память: 16 ГБ.
  + Хранилище данных:
  + SSD объемом не менее 500 ГБ,
  + Рекомендуется RAID-массив для отказоустойчивости,
  + Пропускная способность сети: не менее 1 Гбит/с.

Программное обеспечение:

* + Операционная система: Ubuntu Server 20.04 LTS или выше,
  + Веб-сервер: Nginx 1.18 или выше,
  + СУБД: PostgreSQL 13 или MySQL 8.0,
  + Сервер приложений: Java 11 (Spring Boot),
  + Кэширование: Redis 6.0 или выше.

**4.5 Требования к информационной и программной совместимости**

Архитектура системы:

* + Клиент-серверная архитектура с REST API,
  + Формат обмена данными: JSON,
  + Протокол передачи данных: HTTPS (TLS 1.2 и выше).

Интеграционные возможности. Поддержка стандартных протоколов:

* + OAuth 2.0 для авторизации,
  + OpenID Connect для аутентификации.

Совместимость с платежными системами:

* + API банковских карт (Visa, Mastercard, МИР),
  + Электронные кошельки (Qiwi, Яндекс.Деньги и др.),
  + Возможность интеграции с CRM и ERP системами.

**4.6 Требование к маркировке и упаковке**

Веб-приложение должно:

* + Должна быть доступна по уникальному доменному имени,
  + Должна иметь SSL-сертификат от доверенного центра сертификации,
  + Мета-теги и описание для поисковых систем.

**4.7 Требования к транспортированию и хранению**

Для серверного оборудования:

* + Транспортировка должна осуществляться с соблюдением мер защиты от механических повреждений,
  + Хранение должно осуществляться в сухом помещении с температурой от +10°C до +30°C,
  + Относительная влажность воздуха не должна превышать 80%.

**4.8 Специальные требования**

Специальных требований не предъявляется.

**5 ТРЕБОВАНИЕ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ**

Предварительный состав программной документации:

* + Техническое задание (настоящий документ),
  + Программа и методика испытаний,
  + Руководство системного администратора,
  + Руководство оператора (администратора магазина),
  + Руководство пользователя (покупателя),
  + Техническая документация для разработчиков,
  + Ведомость эксплуатационных документов.

**6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

Система "Магазин игрушек" ориентирована на владельцев бизнеса в сфере розничной торговли детскими товарами. Основное назначение - автоматизация процессов онлайн-продаж с минимальными затратами на внедрение.

Ключевые экономические преимущества:

* + Снижение операционных расходов на 30-40% за счет автоматизации процессов продаж и сокращения персонала,
  + Быстрая окупаемость (6-9 месяцев) благодаря оптимизации бизнес-процессов.

Рыночные перспективы:

* + Стабильный рост онлайн-продаж игрушек (15-20% ежегодно),
  + Высокий спрос среди молодых родителей,
  + Возможности интеграции с маркетплейсами и социальными сетями.

Конкурентные преимущества системы включают удобный интерфейс, мощные инструменты аналитики и поддержку мобильных устройств. Модель монетизации основана на лицензионных отчислениях с возможностью подключения дополнительных платных модулей.

**7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ**

Разработка веб-приложения "Магазин игрушек" включает следующие основные этапы:

Подготовительный этап (2 недели):

* + Анализ требований заказчика,
  + Разработка и согласование технического задания,
  + Планирование сроков и ресурсов,
  + Проектирование архитектуры системы.

Этап разработки (8 недель):

* + Создание интерфейсов для покупателей и администраторов,
  + Разработка серверной части и API,
  + Интеграция с платежными системами,
  + Реализация системы управления товарами и заказами,
  + Настройка базы данных.

Тестирование (2 недели):

* + Проверка функциональности всех модулей,
  + Нагрузочное тестирование,
  + Тестирование безопасности,
  + Исправление выявленных ошибок.

Внедрение (1 неделя):

* + Развертывание системы на серверах заказчика,
  + Настройка рабочей среды,
  + Обучение администраторов,
  + Запуск в эксплуатацию.

Сопровождение (постоянно):

* + Техническая поддержка,
  + Исправление критических ошибок,
  + Консультации пользователей,
  + Плановые обновления системы.

**8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**

Приемосдаточные испытания системы "Магазин игрушек" проводятся в следующем порядке:

Подготовка к испытаниям:

* + Исполнитель разрабатывает программу испытаний, включающую проверку всех функциональных требований, тестирование производительности, контроль безопасности системы,
  + Документ согласовывается с Заказчиком за 5 рабочих дней до начала испытаний.

Проведение испытаний:

* + Тестирование выполняется на тестовом стенде Исполнителя
  + Присутствие представителей Заказчика обязательно
  + Длительность этапа - не более 5 рабочих дней

Фиксация результатов:

* + Все испытания документируются в Протоколе испытаний
  + Протокол содержит результаты проверки каждого требования, выявленные несоответствия, рекомендации по устранению недостатков,
  + Документ подписывается обеими сторонами.

Устранение замечаний:

* + Исполнитель устраняет выявленные недостатки в течение 10 рабочих дней,
  + Проводится повторная проверка исправленных функций.

Завершение приемки:

* + При успешном прохождении испытаний подписывается Акт сдачи-приемки,
  + Акт подтверждает полное соответствие системы требованиям ТЗ, готовность к промышленной эксплуатации, выполнение всех условий договора.

Гарантийные обязательства:

* + Исполнитель предоставляет гарантию 3 месяца с момента подписания Акта,
  + В гарантийный период устраняются все выявленные критические ошибки.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. 1978. Режим доступа: <http://protect.gost.ru/document.aspx?control=7&id=155153>
2. ГОСТ 24.701-86. Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. М.: Издательство стандартов, 1987. – 17 с.
3. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

**Заключение по практическим работам №5-8**

В ходе выполнения практических работ были составлены диаграммы классов и объектов. Описана работа в нотации IDEF0. Также была построена диаграмма в нотации DFD для проекта. Была разработана и нормализирована логическая модель базы данных системы. Была описана архитектура системы и обоснован выбор определенных программных решений для реализации ее компонентов, а также была построена архитектурная диаграмма. В матрицу требований был добавлен столбец соответствия пункта с его компонентом архитектуры. В последней работе было написано ТЗ по ГОСТу 19.201-78.