



Факультет Информационных технологий
кафедра Информатики и информационных технологий

направление подготовки 09.03.02 «Информационные системы и технологии»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРСКАЯ РАБОТА

Тема: Разработка информационной системы для автоматизации процессов
интернет -аптеки

Исполнитель Петров. В.С. _____
(Фамилия И.О.) (Подпись)

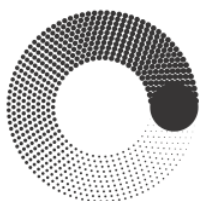
Руководитель Тигина М.С., доцент, к.н. _____
(Фамилия И.О., степень, звание) (Подпись)

«ДОПУЩЕНО К ЗАЩИТЕ»

Зав. кафедрой ИиИТ: **к.т.н. Е.В. Булатников** _____
(Подпись)

Москва

2025



**Факультет Информационных технологий
кафедра Информатики и информационных технологий**

направление подготовки 09.03.02 «Информационные системы и технологии»

УТВЕРЖДАЮ

Зав. каф., к.т.н. Е.В. Булатников

« ____ » _____ 2025 г. _____
(Подпись)

ЗАДАНИЕ НА БАКАЛАВРСКУЮ РАБОТУ

Студент Петров Вадим Станиславович **группа** 211- 723
(Фамилия, Имя, Отчество)

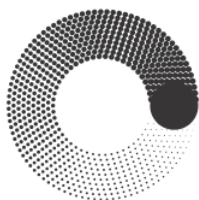
Тема Разработка информационной системы для автоматизации процессов интернет-аптеки

Утверждена приказом по Университету № 6711-УД **от «17» декабря 2024 г.**

- 1. Срок представления работы к защите** « 20 » июня 2025 г.
- 2. Исходные данные для выполнения работы:** методические рекомендации, интернет-ресурсы, техническая документация
- 3. Содержание бакалаврской работы:** введение, анализ предметной области проектирование информационной системы, программная реализация системы, заключение, приложения
- 4. Перечень графического материала** схемы рабочих процессов, диаграммы, примеры интерфейса
- 5. Дата выдачи задания:** « 15 » ноября 2024 г.

6. Руководитель: _____ / Тигина М.С. /
(Подпись) (И.О. Фамилия)

7. Задание к исполнению принял: _____ / Петров В.С. /
(Подпись) (И.О. Фамилия)



Факультет Информационных технологий
кафедра Информатики и информационных технологий

направление подготовки 09.03.02 «Информационные системы и технологии»

КАЛЕНДАРНЫЙ ПЛАН

Студент Петров Вадим Станиславович **группа 211-723**
(Фамилия, Имя, Отчество)

Тема ВКР: Разработка информационной системы для автоматизации
процессов интернет-аптеки

№ п/п	Наименование этапа ВКР	Срок выполнения этапа	Отметка о выполнении
1.	Детальное изучение требований	15.11.2024	выполнено
2.	Анализ предметной области	16.11.2024 – 31.12.2024	выполнено
3.	Разработка архитектурной модели	01.01.2025 – 15.01.2025	выполнено
4.	Проектирование и разработка базы данных	16.01.2025 – 01.02.2025	выполнено
5.	Реализация и тестирование базового API	02.02.2025 – 01.03.2025	выполнено
6.	Доработка функционала	02.03.2025 – 01.04.2025	выполнено
7.	Реализация интерфейса пользователей	02.04.2025 – 15.05.2025	выполнено
8.	Тестирование функционала	16.05.2025 – 24.05.2025	выполнено
9.	Доработки после тестирования	25.05.2025 – 10.06.2025	выполнено
10.	Итоговое тестирование всей системы	11.06.2025 – 20.06.2025	выполнено

Руководитель Тигина М.С., доцент, к.н.
(Фамилия И.О., степень, звание) (Подпись)

Зав. каф. ИиИТ /Е.В. Булатников/
(Подпись)

« ____ » _____ 2025 г.

Аннотация

Выпускная квалификационная работа посвящена разработке информационной системы, предназначенной для автоматизации ключевых процессов интернет-аптеки.

Работа включает такие разделы как введение, анализ предметной области, проектирование информационной системы, программная реализация системы и заключение.

В аналитической главе проводится анализ рынка электронных аптек, приводятся сведения о нормативно-правовых требованиях, рассматриваются ключевые бизнес-процессы, описываются существующие программные решения для автоматизации и выявляются их недостатки.

В главе проектирования описывается архитектура системы, ее компоненты и взаимодействие между ними. Разрабатываются структуры базы данных и алгоритмы, обеспечивающие функционирование.

В главе реализации описывается процесс программной работы системы. Разрабатываются основные модули. Особое внимание уделяется созданию удобного пользовательского интерфейса. Рассматриваются методы интеграции с внешними сервисами, например, электронной почтой.

В заключении подводятся итоги работы, анализируется соответствие достигнутых результатов поставленным целям. Оценивается эффективность разработанной системы и рассматриваются перспективы ее дальнейшего развития и внедрения.

Abstract

The final qualifying work is devoted to the development of an information system designed to automate key processes of an online pharmacy.

The work includes sections such as introduction, domain analysis, information system design, software implementation of the system, and conclusion.

The analytical chapter analyzes the e-pharmacy market, provides information on regulatory requirements, examines key business processes, describes existing automation software solutions and identifies their shortcomings.

The design chapter describes the architecture of the system, its components and the interaction between them. Database entities and algorithms that ensure their operation are being developed.

The implementation chapter describes the process of software operation of the system. The main modules are being developed. Special attention is paid to creating a user-friendly interface. The methods of integration with external services, for example, by e-mail, are considered.

In conclusion, the results of the work are summarized, and the compliance of the achieved results with the set goals is analyzed. The effectiveness of the developed system is evaluated and the prospects for its further development and implementation are considered.

Реферат

Тема: Разработка информационной системы для автоматизации процессов интернет-аптеки.

Объем пояснительной записки составляет N страниц. В работу включено N рисунков, N таблиц, N листингов, N приложений и N источников.

Ключевые слова: информационная система, аптека, архитектура системы, проект, функционал, алгоритм.

В работе обосновывается актуальность проекта, проводится анализ предметной области, рассматриваются особенности бизнес-процессов. Выполнен анализ аналогичных решений на рынке – рассмотрены функциональные возможности ведущих интернет-аптек. На основе анализа формулируются требования к разрабатываемой системе.

Представлено проектирование информационной системы, разработана архитектура клиент-серверного приложения. Уделено внимание вопросам масштабируемости и расширения системы, описаны основные алгоритмы процессов.

В программной реализации выделены ключевые модули системы, интерфейс веб-приложения, интеграция с внешними сервисами.

Оглавление

ВВЕДЕНИЕ	8
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1 Описание предметной области и нормативные требования.....	10
1.2 Текущие тенденции и цифровизация аптечного сегмента	13
1.3 Анализ существующих решений	14
1.4 Выявление проблемных моментов	17
1.5 Определение возможностей для автоматизации.....	18
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	21
2.1 Выбор технологий и инструментов.....	21
2.2 Проектирование архитектуры системы	23
2.3 Возможности расширения архитектуры системы	26
2.4 Интеграция с внешними сервисами и API.....	28
2.5 Разработка базы данных	31
2.6 Разработка алгоритмов работы системы	36
2.6.1 Регистрация и авторизация	36
2.6.2 Поиски фильтрация товаров	40
2.6.3 Корзина и оформление заказа	41
2.6.4 Оплата заказа	44
2.6.5 Администрирование.....	44
2.6.6 Электронная почта и уведомления.....	46
ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ	48
3.1 Реализация основных модулей	48
3.1.1 Модуль авторизации и регистрации.....	48
ЗАКЛЮЧЕНИЕ	52
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	53
ПРИЛОЖЕНИЕ А	54
ПРИЛОЖЕНИЕ Б.....	61

ВВЕДЕНИЕ

Развитие информационных технологий оказывает существенное влияние на здравоохранение и медицину. В связи с растущей потребностью населения в быстром и легком доступе к лекарственным средствам, а также повышением спроса на качественное обслуживание организация эффективного онлайн-сервиса для аптечных сетей становится все более актуальной. Сегодня интернет-аптеки занимают важное место на фармацевтическом рынке, предоставляя удобный способ заказа лекарств и медицинских изделий через веб-интерфейс.

Темпы роста онлайн оборота лекарств опережают большинство других направлений электронной коммерции. В 2022 году количество онлайн-заказов медицинских изделий выросло на 45% по сравнению с предыдущим годом, а средний чек в интернет-аптеке составил около 1680 рублей. Относительно низкий объем заказов обусловлен спецификой товара, однако общее число транзакций быстро растет. Фармацевтическая торговля продолжает набирать обороты, и все больше людей предпочитают заказывать лекарства удаленно.

Многие интернет-аптеки реализованы либо как агрегаторы с ограниченной функциональностью, либо на основе устаревших решений, которые не предоставляют должного удобства. Это открывает путь в разработке собственной информационной системы, способной интегрировать весь цикл работы интернет-аптеки. Такая система улучшит обслуживание клиентов, автоматизирует рутинные задачи сотрудников и обеспечит эффективное управление данными и бизнес-процессами.

Целью данной работы является разработка информационной системы, предназначенной для автоматизации процессов интернет-аптеки.

Для достижения цели были поставлены следующие задачи:

- Провести анализ предметной области и действующих нормативных актов;
- Выявить существующие проблемы и перспективы автоматизации;

- Изучить существующие аналогичные программные решения;
- Спроектировать архитектуру веб-приложения с учетом выбранного стека технологий;
- Разработать архитектуру базы данных для хранения медицинской информации;
- Реализовать модули системы, включающие авторизацию, каталог продукции, обработку заказов, личный кабинет, административный интерфейс и интеграцию с электронными сервисами.

Выполнение этих задач позволит создать полнофункциональную и масштабируемую веб-систему, которая обеспечит автоматизацию процессов интернет-аптеки и будет способна стать основой для внедрения в реальных условиях.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области и нормативные требования

Интернет-аптека – это электронный сервис, предназначенный для дистанционной продажи лекарственных средств и медицинских изделий. По сути, эта система сочетает в себе функции традиционной аптеки и интернет-магазина. Современные потребители ожидают от подобного сервиса удобства, полной информации о продуктах и ценах. Все больше людей предпочитают заказывать товары, в том числе лекарства, удаленно. По данным исследования Data Insight, сегмент интернет-аптек демонстрирует одни из самых высоких темпов роста в электронной коммерции. В 2022 году объем продаж увеличился на 45% по сравнению с 2021 годом, что показано на рисунке 1.1.1. При этом средний чек заказа относительно невелик (~1680 руб). Люди часто заказывают повседневные медикаменты и не собирают большие корзины.

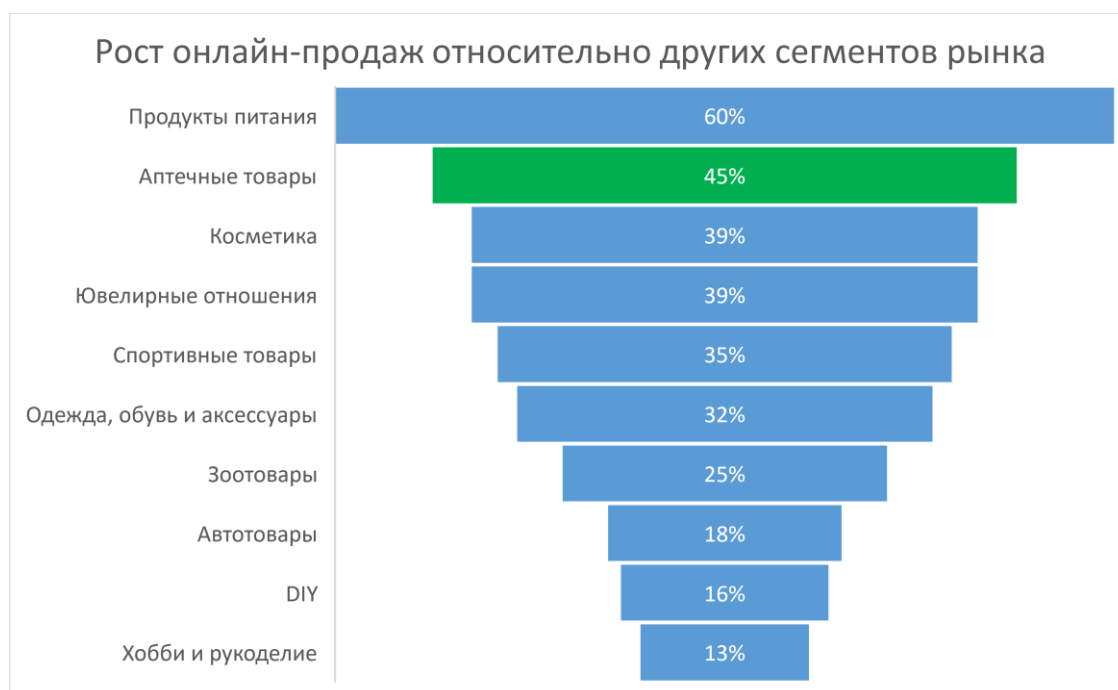


Рисунок 1.1.1 – Темпы роста аптечного сегмента в 2022 году в сравнении другими категориями

Интернет-аптеки имеют ряд особенностей, которые отличают их от классических интернет-магазинов. Во-первых, очень широкий и разнообразный ассортимент товаров. Типичный аптечный каталог содержит тысячи наименований лекарственных средств, медицинских изделий, витаминов, лечебной косметики и т.д. В данном секторе объем данных увеличивается в несколько раз – необходимо учитывать разные дозировки и формы выпуска одного препарата, наличие аналогов. В результате в базе данных товаров появляются десятки и сотни тысяч позиций. Кроме того, в фармацевтической отрасли цены и уровень запасов постоянно меняются: поступает новая партия препаратов, поставщики корректируют цены и т.д. Обновление данных происходит гораздо чаще, чем в обычной рознице. Объем и изменчивость данных предъявляют высокие требования к информационной системе интернет-аптеки: она должна эффективно обрабатывать большой каталог товаров, поддерживать сложную иерархию категорий, обеспечивать быстрый поиск по базе данных. Также необходимы гибкие механизмы фильтрации и маршрутизации, чтобы пользователь мог легко найти нужный товар среди большого ассортимента.

Во-вторых, немалое влияние оказывает нормативно-правовая среда. До недавнего времени законодательство накладывало строгие ограничения на дистанционную продажу лекарств. До апреля 2020 года онлайн-продажа лекарств была запрещена. Каждая аптека являлась приемным пунктом, и информация о ней должна была размещаться на сайте. С апреля 2020 правила были пересмотрены – принятие Федерального закона от 03.04.2020 №105-ФЗ внесло изменения в закон №61-ФЗ “Об обращении лекарственных средств”, официально разрешив продажу некоторых лекарственных средств через интернет. В частности, аптекам, которые имеют действующую лицензию, разрешено принимать онлайн-платежи и передавать заказы службам доставки. Но остаются ограничения: рецептурные лекарства, наркотические и психотропные средства, а также препараты, содержащие более 25% этанола, запрещены для дистанционной продажи. Все остальные одобренные лекарства

можно легально заказать онлайн. Эти законодательные изменения значительно повлияли на развитие отрасли, открыв новые возможности для автоматизации продаж медицинских изделий.

Важно отметить, что согласно новым правилам дистанционной торговли лекарствами, аптеки обязаны получать специальное разрешение Росздравнадзора для работы онлайн. Разрешение выдается только аптечным организациям с действующей лицензией не менее одного года и при соблюдении ряда условий: наличие более 10 стационарных аптечных пунктов, оборудованных мест хранения, регистрации доменного имени сайта в зоне .ru и пр. Росздравнадзор ведет электронный реестр выданных разрешений на онлайн торговлю лекарствами. И в случае нарушений ведомство может отозвать разрешение и заблокировать сайт. Регулирование интернет-аптек в РФ достаточно строгое и для законной работы онлайн-аптека должна соответствовать всем требованиям и отчитываться перед надзорными органами.

Помимо правил продажи, есть нормативные требования к хранению и обработке медицинских и персональных данных. Информация о здоровье клиентов относится к специальной категории персональных данных. В соответствии со ст.10 Федерального закона №152-ФЗ “О персональных данных”, обработка сведений о состоянии здоровья без явного согласия пользователя запрещена. Аптека должна обеспечить конфиденциальность данных, применять меры защиты и хранить данные только на серверах на территории РФ. Также действуют приказы Минздрава, которые регламентируют учет рецептов и хранение медицинской документации. С 1 июля 2020 года в России была введена обязательная цифровая маркировка лекарств. Каждый проданный препарат должен быть промаркирован уникальным кодом и его движение отслеживается в национальной системе мониторинга “Честный ЗНАК”. Благодаря этому интернет-аптеки и клиенты могут проверить подлинность и статус лекарства. В данном проекте основное внимание уделяется общей автоматизации онлайн-продаж, но архитектура

предусматривает возможность доработки под требования маркировки и других стандартов.

В итоге, предметная область интернет-аптек характеризуется:

- Большим объемом данных;
- Отраслевыми ограничениями;
- Повышенными требованиями к безопасности.

Эти особенности требуют необходимость комплексной информационной системы, способной эффективно управлять внутренними процессами онлайн-аптеки.

1.2 Текущие тенденции и цифровизация аптечного сегмента

В условиях смягчения законодательства и особенно после пандемии наблюдается значительный рост онлайн-сегмента фармацевтики. Как отмечалось ранее, уже в 2022 году объем продаж в интернете для аптечных товаров в России вырос на ~45% по сравнению с предыдущим годом. На этом рост не остановился и в 2024 году было реализовано лекарств на сумму 211,1 млрд руб., что на 24,9% больше, чем в 2023. Итак, онлайн-аптеки постепенно набирают долю в общем рынке.

Подобный быстрый рост имеет ряд причин:

Изменение предпочтений потребителей. Люди предпочитают заказывать лекарства онлайн из дома. По опросам 94% россиян, которые уже покупали товары аптеки онлайн, и дальше намереваются делать это.

Смещение аудитории в мобильные приложения. Современный пользователь чаще заказывает товары через смартфон. Например, у сервиса “СберЕаптека” около 68% посетителей заходят со смартфонов, поэтому большие лидеры рынка развивают мобильные приложения наряду с веб-сайтами.

Участие маркетплейсов в реализации лекарственных средств. К 2024 году крупнейшие интернет-площадки принимали до 80% от количества онлайн-

заказов, выступая посредниками между покупателями и аптечными сетями. Это ведет к тому, чтобы традиционным аптекам приходится адаптироваться и интегрироваться с этими площадками или развивать собственные сервисы.

В то же время происходит цифровизация смежных отраслей здравоохранения, что открывает новые возможности для электронных аптек. В настоящее время в России внедрилась система электронных рецептов: согласно Стратегии лекарственного обеспечения населения России все регионы должны были внедрить систему электронных рецептов до конца 2023 года. Некоторые регионы полностью отказались от бумажных рецептурных бланков и отдали предпочтение цифровым аналогам. Развитие электронных рецептов и интеграция медицинских информационных систем позволит интернет-аптекам принимать и обрабатывать рецепты напрямую от врачей. Это существенно повысит удобство клиентов, которым больше не придется вручную переносить назначения. Рецептурные препараты смогут заказывать онлайн, а в перспективе и полностью отпускать дистанционно по электронным рецептам. В общем, развивается тенденция к глубокой цифровой интеграции аптек в систему здравоохранения, и в ближайшие годы ожидается дальнейший рост онлайн-продаж лекарств, а также появление новых услуг на базе интернет-аптек.

1.3 Анализ существующих решений

Рассмотрим существующие решения на рынке, близкие по функционалу к разрабатываемой системе. Были выбраны три крупных российских интернет-аптеки: Apteka.ru, Еаптека (Sber Еаптека) и Asna.ru. Их анализ поможет отметить сильные стороны и недостатки.

Apteka.ru – это один из первых и самых крупных онлайн-сервисов для заказа лекарств в России. Фактически это аптечный агрегатор, объединяющий есть аптек-партнеров по всей стране. Сервис позволяет покупателям выбрать необходимые лекарственные препараты через сайт (или мобильное приложение), проверить их наличие и цену, а также оформить заказ на доставку

или самовывоз. Пользователь выбирает подходящую аптеку, резервирует там товар, а затем может получить и оплатить его на месте. Из-за своей партнерской базы Apteka.ru покрывает почти все регионы РФ. По состоянию на 2022 год сервис занял первое место по объему онлайн-продаж. На Apteka.ru пришлось около трети (31%) всей части электронных аптек.

Возможности Apteka.ru включают в себя огромный каталог из десятков тысяч записей препаратов, удобный поиск в автодополнением, фильтрацию по характеристикам и категориями, выбор пункта выдачи на интерактивной карте, уведомление о статусе заказа. К недостаткам, которые отмечали в ранние годы работы, можно отнести функционал личного кабинета: изначально в нем было только отслеживание статуса заказа, без расширенных возможностей (история покупок, избранное и др.). Но со временем платформа обросла и добавила элементы полноценного интернет-магазина, однако остается проблема достаточно перегруженного интерфейса.

Eapteka.ru (после интеграции в систему Сбербанком известна как СберЕаптека) – крупная интернет-аптека, основанная в 2000 году. На сегодняшний день это один из лидеров рынка, конкурирующий с Apteka.ru по охвату аудитории. Изначально сервис развивался как классическая интернет-аптека: компания имела собственные склады и фирменные пункты выдачи. Ассортимент Еаптеки очень широк. По заявлениям компании, в него входит более 45 тыс. позиций товаров. Сервис ориентирован на удобство пользователя и в нем реализованы функция корзины покупок, различные способы получения заказов, поддержка онлайн-оплаты. Еаптека имеет собственное мобильно приложение – около 68% посетителей заходят на сайты аптек со смартфонов.

Функциональные возможности не уступают Apteka.ru. Они включают в себя каталог с подробной страницей каждого товара, поиск по заболеваниям и симптомам, рекомендации по похожим товарам, личный кабинет с историей заказов, скидки. Также доступны консультации по телефону или в чате, работает горячая линия техподдержки. Еаптека представляет собой пример полноценной информационной системы. Данный сервис очень поможет в

разработке собственной системы, т.к. уже имеет необходимый в реализации функционал.

И последний участник анализа – Asna.ru. Это крупнейший в России аптечный интернет-агрегатор, который по собственным данным объединяет более 13 тыс. аптек по всей стране. Проект АСНА (Ассоциация независимых аптек) предоставляет единый сервис для поиска и оформления заказа лекарств у разных компаний и сетей. Работа Asna.ru во многом схожа с Apteka.ru. Сначала пользователь вводит название лекарства, а система выводит список аптек-партнеров в выбранном городе с этим товаром в наличии. Затем клиент заказывает товар в выбранной аптеке и забирает самостоятельно. Одним из главных преимуществ сайта является широкий охват и функция сравнения цен и наличия в десятках аптечных сетей. Для аптек-участников Asna.ru создает дополнительный поток клиентов без необходимости разрабатывать собственный сервис.

Функционал Asna.ru включает поиск лекарств и их аналогов, фильтрацию результатов по местоположению, просмотр детальной информации препарата, а также личный кабинет для отслеживания статуса заказа и истории покупок. За последние годы в компании автоматизирована интеграция с системами учета лекарственных средств, чтобы информация о ценах и наличии обновлялась в режиме реального времени. Но, как и Apteka.ru, платформа не обязательно предлагает доставку или онлайн оплату – это агрегатор для заказа с дальнейшей офлайн-покупкой. С точки зрения собственной информационной системы опыт Asna показывает функционал сравнения товаров в разных аптеках, но и показывает ограничения агрегаторного подхода – отсутствие единой корзины покупок и доставка могут быть менее удобными, чем в случае с полноценными интернет-аптеками.

Из проанализированного выше можно сделать вывод, что успешная система должна иметь большой каталог, удобный поиск, отображение наличия и цен, поддержку различных способов оплаты. Важна интеграция с внешними

системами – обновление наличия и цен. Недостатки некоторых существующих решений указали на чем следует сосредоточиться в новой системе.

1.4 Выявление проблемных моментов

Выделим основные проблемы и ограничения, которые должна учитывать разрабатываемая информационная система.

Первое на что нужно обратить внимание это сложность управления данными. Огромный ассортимент продукции и постоянная смена этикеток затрудняют ручное ведение остатков и цен на складе. Это может вызвать проблему асинхронности, когда покупатель видит на сайте товар, которого в реальности уже нет. Проще говоря, без автоматизации управления данных в интернет-аптеке будет значительный объем ручной работы в совокупности с вероятностью ошибок и задержек.

Отсутствие единой структуры обработки онлайн-заказов. Во многих аптеках ввод онлайн-продаж было разным – заказы можно было получать по электронной почте или через отдельные программы, что требовало дополнительной ручной обработки. Это увеличивает риск ошибок и в принципе замедляет процесс выполнения заказа. Возникает нужда в едином автоматизированном управлении заказами. Заказ, который оформлен на сайте, попадает в систему, где его видит сотрудник аптеки. Сотрудник меняет статус заказа, а клиент получает уведомления. Алгоритм обработки должен быть единым для всех участников процесса.

Ограниченный функционал для клиентов. К потенциальным проблемам можно отнести неинтуитивный пользовательский интерфейс поиска лекарств, отсутствие уведомлений об изменении статуса заказа и ограниченные возможности обратной связи. Если клиент испытывает трудности с поиском товара или не знает, на каком этапе находится его заказ, то это снижает доверие к сервису. Многие классические аптечные сети при первом выходе в интернет не сразу внедряют функционал персонализации. Например, список избранного

или история заказов, которые уже стали обычным явлением в интернет-магазинах.

Производительность и масштабируемость. По мере увеличения количества пользователей и запросов могут возникнуть проблемы с производительностью, например, длительная загрузка страниц, зависания и т.д. Система должна быть способна обрабатывать сотни запросов в секунду (RPS) без ухудшения производительности. И для этого требуется принимать соответствующие архитектурные решения (подробнее в Главе 2), например, оптимизация запросов к базе данных и другие меры по обеспечению стабильной работы.

1.5 Определение возможностей для автоматизации

Для решения проблем, обозначенных выше, необходимо внедрить систему, которая автоматизирует основные процессы интернет-аптеки.

Единая платформа для данных и операций. Создание такой информационной системы позволит объединить все нужные функции в одном месте для разных ролей пользователей – администраторов, сотрудников аптеки и клиентов. Информация о продуктах, заказах и клиентах будет храниться в общей базе данных и обновляться в режиме реального времени. В панели администратора можно будет управлять каталогом товаров, контролировать заказы и все что с ними связано. Такой подход повысит эффективность работы и облегчит обновление данных.

Автоматизация процесса заказа. Информационная система должна поддерживать весь путь обработки заказа, от момента появления товара в корзине до оформления и оплаты с последующим получением заказа на руки. Каждый шаг должен быть автоматизирован. Клиент подтверждает заказ нажатием кнопки на сайте, после чего система создает заказ и уведомляет сотрудника аптеки об этом. Сотрудник видит задачу в своем интерфейсе, и ему остается только менять статусы заказа. После сборки заказа, он отмечает его как готовый к получению, а клиенту приходит уведомление, что заказа можно

забрать. Рутинные процессы заменяются электронными транзакциями, что ускоряет процесс и снижает ошибки.

Безопасность и соответствие требованиям. Как уже было отмечено ранее, при обработке персональных данных клиентов и информации о лекарствах система должна соблюдать требования Федерального закона №152-ФЗ. Проблема может заключаться в недостаточной защите данных. Система должна обеспечить высокий уровень безопасности информации. Например, хранение паролей в зашифрованном виде (хеширование), использование защищенных соединений, разграничение прав доступа между ролями.

Улучшение лояльности клиентов. Автоматизация предоставляет инструменты для лучшего взаимодействия с клиентами. Система сможет отправлять пользователям уведомления на каждом этапе: подтверждение заказа, смена статуса заказа, электронный чек и т.д. Личный кабинет станет единым местом, где пользователь получает всю необходимую информацию о своих персональных данных, текущем статусе заказов, истории покупок. При реализации “Избранного”, пользователь сможет формировать список товаров, которые он часто приобретает и быстро находить их для создания повторных заказов. Также можно внедрить личные рекомендации на основе предыдущих покупок.

В результате автоматизация процессов интернет-аптеки позволит не допустить большинство выявленных проблем. Она будет соответствовать современным требованиям к онлайн-сервисам и позволит избавиться от рутинных операций. Также перечисленные моменты в целом определяют технические и функциональные требования к разрабатываемой системе.

В ходе анализа были изучены особенности работы интернет-аптек, текущие тенденции, нормативные ограничения и существующие решения. Рынок активно развивается, но многие системы не охватывают весь цикл заказа товаров. Также были выделены основные проблемы и сложности процессов. Сравнение ведущих сервисов показывает, что для привлечения пользователей важно обеспечить богатый функционал и простоту использования. Анализ

продемонстрировал потребность в комплексной, гибкой и безопасной системе. Это подчеркивает актуальность разработки собственного программного решения, способного автоматизировать основные процессы электронных аптек.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Выбор технологий и инструментов

Для реализации программной части проекта выбран современный технологический стек, который обеспечивает надежность, производительность и соответствие поставленным требованиям.

Платформа .NET 9 – как основа для серверной части (бекенд). Последняя версия .NET 9 обеспечивает улучшенную производительность и поддерживает кроссплатформенную разработку. C# – объектно-ориентированный язык программирования, идеально подходящий для корпоративных приложений. Объединение .NET + C# позволяет реализовать надежный веб-сервис (в рамках ASP.NET Core), отвечающий за бизнес-логику интернет-аптеки. Выбор .NET обусловлен его большой и разнообразной экосистемой библиотек и инструментов, встроенными средствами для создания веб-API, а также высоким уровнем безопасности и масштабируемости. Кроме того, большое сообщество и хорошая документация облегчат разработку и поддержку вашего кода.

PostgreSQL – как система управления базами данных. Это мощная объектно-реляционная система управления базами данных с открытым исходным кодом, известная своей надежностью и соответствием стандартам SQL. Она поддерживает транзакции (ACID), имеет широкие возможности по хранению и манипулированию данными (включая хранимые процедуры, триггеры, поля JSON и т. д.), что полезно при выполнении сложных запросов. PostgreSQL хорошо адаптирован к растущим объемам информации, что напрямую касается баз данных продуктов и заказов в интернет-аптеках. Немаловажно и то, что сообщество PostgreSQL активно развивается – существуют многочисленные инструменты резервного копирования и репликации версий, обеспечивающие надежное хранение фармацевтических данных.

Vue.js – это фреймворк для разработки клиентской части веб-приложения. Для реализации интерфейса интернет-аптеки был выбран Vue.js из-за его

легкости и гибкости. Он реализует реактивный подход к построению пользовательского интерфейса, что означает, что изменения данных автоматически отражаются в интерфейсе, не требуя перезагрузки страницы. Vue.js прост в использовании и легко интегрируется, позволяя быстро создавать интерактивные компоненты, например, динамический поиск лекарств с предложениями, мгновенное обновление корзины, фильтры каталога и многое другое. Кроме того, Vue.js имеет хорошо развитую экосистему (Pinia для управления глобальным состоянием и Vue Router для маршрутизации страниц приложений), что будет полезно при масштабировании функциональности. Альтернативами могли бы стать React или Angular, но Vue.js часто хвалят за сочетание простоты и производительности, что важно для этой системы.

Cloudflare R2 – облачное хранилище для изображений товаров. Оно используется в связке с S3-совместимым API, что позволяет управлять изображениями через бекенд без использования внешнего CDN или FTP. Хранилище устойчиво к сбоям, а отсутствие платы за исходящий трафик позволяет экономить.

Gmail SMTP – сервис для отправки уведомлений по почте. Используется SMTP-сервер Google, который обеспечивает доставку писем. Подключение реализовано через защищенный TLS протокол, а данные SMTP авторизации хранятся в конфигурации сервера.

ЮKassa – сервис для приема онлайн-оплаты. Произведена интеграция с платежным сервисом, который полностью обрабатывает платеж на своей стороне и возвращает уведомление о статусе операции. Подробнее об интеграции с внешними сервисами расписано в разделе 2.4.

JetBrains Rider – интегрированная среда разработки (IDE), используемая для написания кода. Rider был выбран в качестве основного инструмента разработки по нескольким причинам. Во-первых, он поддерживает готовые технологии .NET/C#, обеспечивая удобство рефакторинга, автодополнения кода, устранения неполадок и запуска веб-приложений. Во-вторых, Rider является кроссплатформенным – разработка может осуществляться на Windows, Linux или macOS с одинаковой легкостью (в отличие от Visual Studio,

которая ограничена Windows для разработки .NET). Rider также умеет работать с фронтенд-технологиями, подсвечивает синтаксис, подсказывает ошибки, что удобно при создании единого решения.

Выбранные технологии хорошо работают вместе. .NET для серверной логики и PostgreSQL для хранения данных обеспечат надежный бэкэнд, а Vue.js обеспечит современный интерактивный фронтенд. Использование этих инструментов позволит разрабатывать качественные системы. Кроме того, все компоненты либо бесплатны, либо имеют открытые лицензии.

2.2 Проектирование архитектуры системы

Структура разрабатываемой информационной системы основана на принципе многоуровневого веб-приложения с четким разделением на клиентскую и серверную части (архитектура клиент-сервер), а также базу данных. Ниже описано взаимодействие этих компонентов.

Клиент здесь представляет собой одностраничное веб-приложение на базе Vue.js. Оно работает в браузере пользователя. Этот уровень отвечает за отображение интерфейса, сбор данных от пользователя и взаимодействие с сервером через API. Фронтенд обрабатывает основные сценарии, которые включают просмотр каталога, наполнение корзины, оформление заказа, ввод персональных данных и т.д. При выполнении какого-либо действия клиент отправляет асинхронные запросы на сервер и обновляет интерфейс на основе ответа. Все визуальные компоненты являются частью интерфейса. Интерфейс динамически реагирует на действия пользователя, например, при добавлении товара в корзину соответствующий счетчик будет обновлен немедленно, без необходимости перезагрузки всей страницы.

Серверный слой – это приложение на платформе ASP.NET, которое работает на удаленном сервере. Он предоставляет API для интерфейса и обрабатывает всю бизнес-логику. Эта часть выполняет следующие функции:

- Аутентификации и авторизации пользователей. Проверка логина и пароля, управление токенами JWT;
- Обработка CRUD-операций с данными;
- Применение бизнес-правил;
- Валидация запросов и данных;
- Интеграция с внешними сервисами. Отправка уведомлений по электронной почте.

Архитектурно бэкенд разделен на подуровни. Эндпоинты принимают запросы API и возвращают ответы. Сервисы инкапсулируют бизнес-логику и уровень доступа к данным (ORM и репозитории для взаимодействия с базой данных). Серверное приложение, написанное на C#, компилируется и размещается на веб-сервере в среде Kestrel и Nginx для .NET Core. Взаимодействие клиента и сервера осуществляется по стандартному протоколу HTTPS, который обеспечивает шифрование передаваемых данных. Каждый запрос связан с определенным методом на сервере. Такая организация позволяет отделить клиентскую логику от серверной.

Слой хранения данных – это централизованное хранилище данных на основе СУБД PostgreSQL. В базе данных хранятся основные сущности предметной области (структура подробно рассмотрена в разделе 2.5). Серверное приложение взаимодействует с БД через SQL-запросы, которые написаны либо вручную, либо сгенерированы самой ORM. Хранилище обычно размещено на отдельном сервере или контейнере, доступ к которому имеет только бекенд, а прямого доступа из интернета нет ни у кого. Для защиты данных разграничиваются права, а именно приложение подключается к БД от имени пользователя с ограниченными правами. Также использование PostgreSQL обеспечивает работу при высоких нагрузках, поддерживает индексы для поиска записей, хранимые процедуры и др. Данная СУБД полностью соответствует принципам ACID по надежности транзакции, и это важно для обработки данных.

Для наглядности архитектура может быть представлена в виде диаграммы развертывания на рисунке 2.2.1.

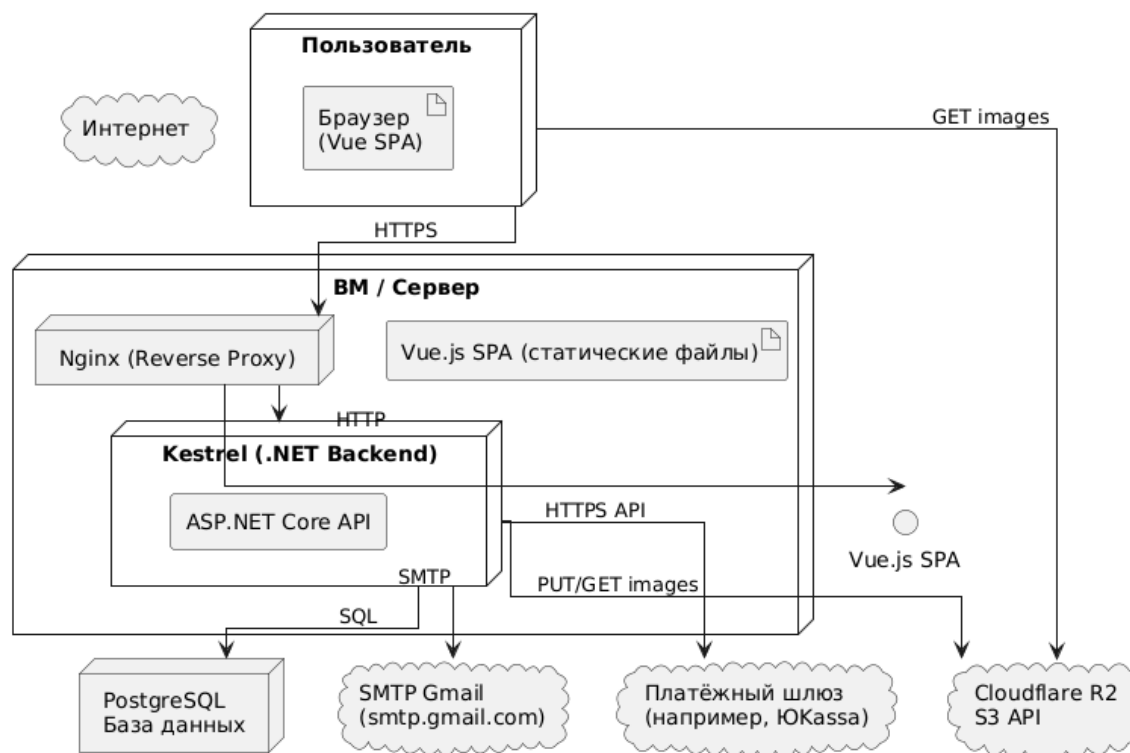


Рисунок 2.2.1 – Диаграмма развертывания архитектуры системы

Система построена по традиционной клиент-серверной архитектуре. Клиентом является браузер, в котором загружается одностраничное приложение (SPA), которое реализовано на Vue.js. Статические файлы размещаются на сервере и отдаются пользователю через Nginx.

На серверной стороне используется сочетание Nginx и Kestrel. Веб-сервер Nginx является обратным прокси. Он принимает входящие HTTPS-запросы и перенаправляет их в зависимости от пути. Основная серверная часть реализована на ASP.NET Core и развернута как отдельный процесс. Она обрабатывает входящие запросы от клиента, обращается к базе данных PostgreSQL для выполнения операций с данными.

Дополнительно показаны интеграции с внешними сервисами. Cloudflare R2 используется как S3-совместимое хранилище изображений. SMTP-сервер

Gmail нужен для отправки электронных писем, взаимодействие с которым осуществляется по протоколу SMTP с шифрованием. А платежный шлюз обрабатывает онлайн-платежи.

Итак, спроектированная архитектура является трехслойной веб-системой, где за отображение отвечает Vue.js, за бизнес-логику и интеграцию с внешними сервисами – ASP.NET Core, а за хранение данных – PostgreSQL.

2.3 Возможности расширения архитектуры системы

Разработанная архитектура системы интернет-аптеки является масштабируемой и модульной, поэтому при необходимости позволяет внедрять новые компоненты и улучшать производительность.

При увеличении числа активных пользователей потребуется масштабирование серверной части. Например, бекенд может быть развернут в виде нескольких экземпляров приложения ASP.NET Core за балансировщиком нагрузки, т.к. архитектура не хранит состояние сессий в памяти сервера. Балансировщик распределяет запросы между этими экземплярами, позволяя обрабатывать большое количество запросов в секунду.

Для снижения нагрузки на базу данных и ускорение отклика возможно применение кэша Redis. Он хранит в себе часто запрашиваемые данные, результаты запросов, которые не требуют мгновенного обновления, например, список категорий и подкатегорий товаров, персональная информация пользователя и т.д.

Для отслеживания технического состояния системы и диагностики ошибок возможна интеграция с системами логгирования и мониторинга. Например, связка Elasticsearch, Kibana и Prometheus, Grafana будет хорошо себя показывать в реальных условиях.

Система построена по принципу разделения на слои. Это позволяет в будущем перейти к микросервисной архитектуре. Каждый модуль может быть вынесен в отдельный сервис со своим хранилищем и API.

Для обеспечения устойчивости системы возможно добавление репликации базы данных и регулярного резервного копирования.

Некоторые операции, которые в данный момент не являются критическими, выполняются асинхронно с помощью брокера сообщений. Например, отправка уведомлений по электронной почте. Вместо того, чтобы отвлекать пользователя, эти задачи помещаются в очередь (RabbitMQ или Kafka) и обрабатываются в фоновом режиме. В итоге, основное веб-приложение не тратит ресурсы на длительные задачи.

Ниже на рисунке 2.3.1 представлена возможная диаграмма развертывания, которая включает блоки, расширяющие систему.

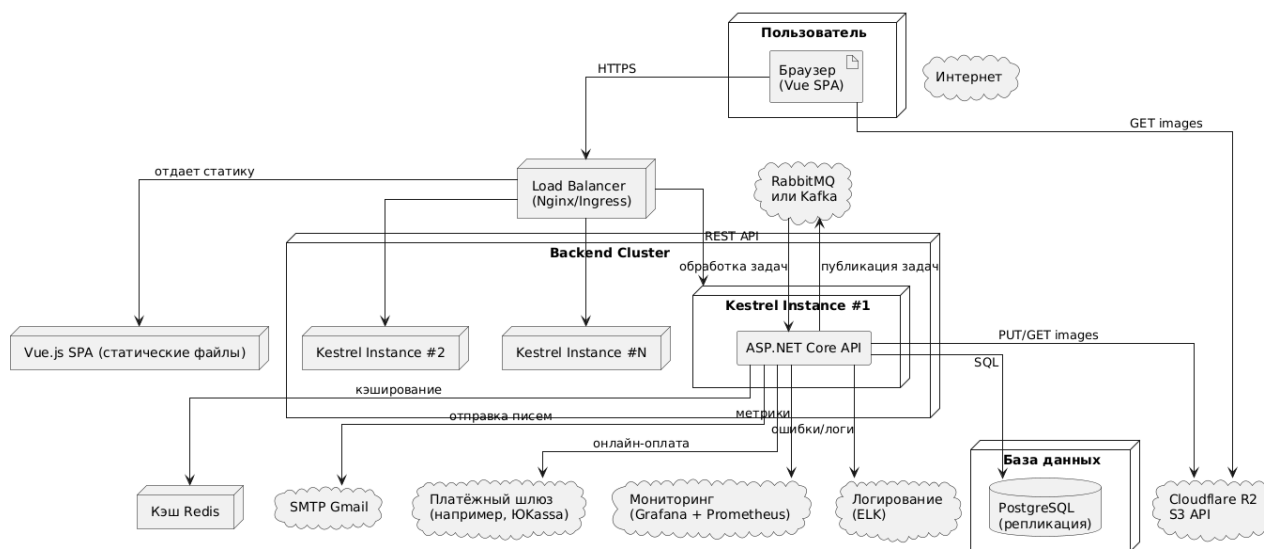


Рисунок 2.3.1 – Диаграмма развертывания архитектуры системы с учетом масштабируемости

Сервер приложений разворачивается по схеме N+1, т.е. всегда есть хотя бы два экземпляра, которые работают на разных физических узлах. И при выходе из строя одного, трафик переходит на остальные стабильные.

Важно учитывать масштабируемость по самой разработке. Код спроектирован модульно, это облегчает добавление новых функций. Внедрен механизм миграций базы данных для безопасного обновления схемы данных без простоя сервисов.

Все эти меры предполагают, что система сможет обслуживать рост нагрузок от нескольких заказов в день до нескольких заказов в секунду. Однако на текущем этапе проекта достаточно масштабируемого монолита с возможностью горизонтального роста.

2.4 Интеграция с внешними сервисами и API

Современная информационная система интернет-аптеки не существует сама по себе. Для расширения функциональности необходимо интегрировать ряд внешних сервисов. В проекте предусматриваются следующие ключевые направления интеграции:

Подключение к справочникам лекарственных средств. Для поддержания актуальности базы данных необходимо использовать открытые данные из государственных реестров и специализированных систем. Например, Минздрав ведет Государственный реестр лекарственных средств (ГРЛС), а Росздравнадзор ведет реестры лицензирования и зарегистрированных цен. Эти данные можно получить через API. Существуют готовые решения, которые объединяют информацию из ГРЛС, ЖНВЛП, МДЛП и других источников, предоставляя единый справочник лекарственных препаратов. Как пример, одним из таких решений является сервис Pharm Portal, который собирает данные и позволяет получать их через свое API. В текущей системе предполагается синхронизация части справочных данных с такими источниками. Данный подход избавит от ручного заполнения информации и гарантирует соответствие товаров с официальными данными.

Интеграция с платежными системами. Для обработки электронных платежей был выбран подход на основе внешнего платежного агрегатора. Прием платежей связан с высокими требованиями безопасности и технической сложностью, поэтому предпочтительнее использовать специализированный сервис, например, ЮKassa (YooMoney) или аналогичный. ЮKassa – самая популярная платежная система в России. На 2018 год его использовали более

106 тыс. сайтов, что составляет около 30% рынка онлайн-платежей. Сервис поддерживает все основные способы оплаты: банковские карты, электронные кошельки, мобильные платежи и даже наличные через терминалы. В контексте проектирования это означает, что клиентская часть не будет самостоятельно обрабатывать введенные данные карты. Вместо этого при оформлении заказа система инициирует платежную сессию, и пользователь либо перенаправляется на платежную страницу, либо видит встроенный платежный виджет от внешнего сервиса. Данные вводятся строго на стороне ЮKassa, что обеспечивает их шифрование и безопасное хранение. После успешной оплаты платежная система возвращает нашей системе уведомление (callback) с результатом, после чего происходит смена статуса платежа. В проекте описанная интеграция реализуется через API ЮKassa. Был зарегистрирован мерчант-аккаунт, получены закрытый ключ и идентификатор магазина, настроен webhook для уведомлений о платеже. Данный подход сильно упрощает процесс разработки и повышает доверие пользователей, поскольку они видят привычный платежный интерфейс. В дальнейшем возможна параллельная интеграция других платежных методов, например, с Системой быстрых платежей (СБП), но на текущем этапе возможностей ЮKassa вполне хватает для обработки транзакций.

Работа с адресами и геосервисами. Для интернет-аптеки важна точность и удобство ввода адреса доставки или самой аптеки. Ошибки в адресе могут привести к потере заказов или задержке доставки. Чтобы избежать таких неприятностей, проект интегрируется с открытыми бесплатными сервисами на базе OpenStreetMap (OSM). Ниже на рисунке 2.4.1 показан пример каким образом пользователь получает данные об адресах.

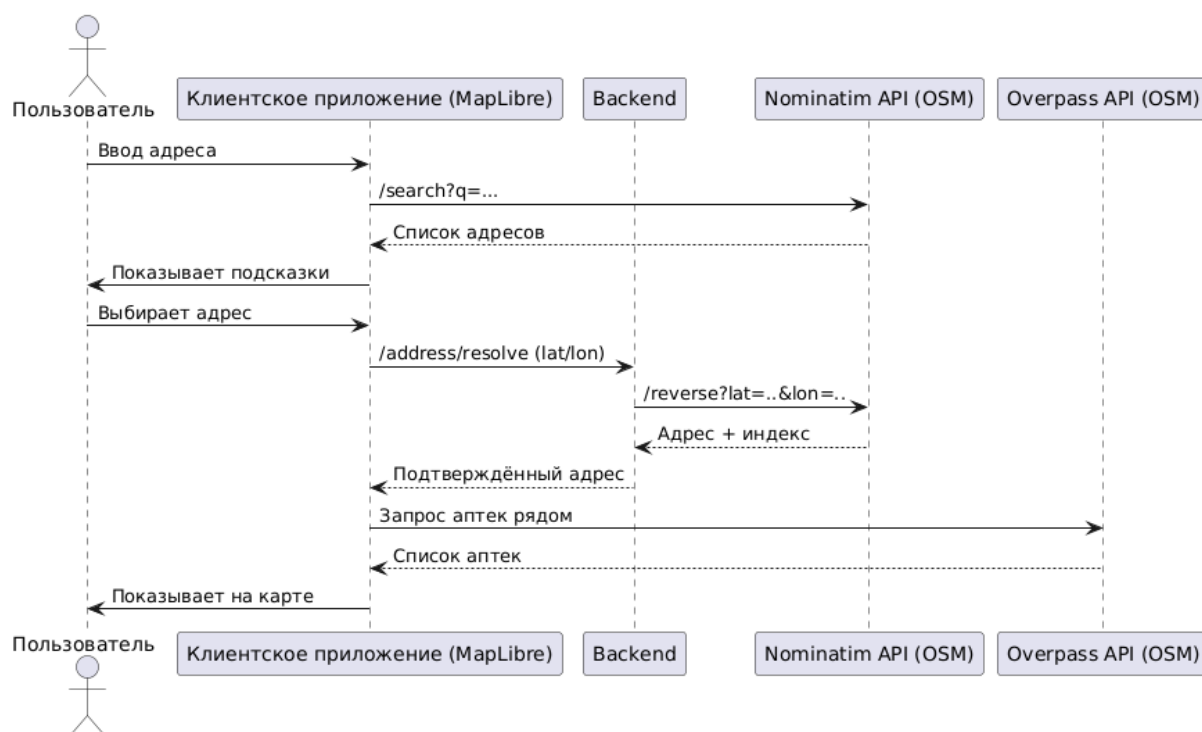


Рисунок 2.4.1 – Процесс отображения данных на карте

В качестве компонента отображения карты используется MapLibre GL JS. Это современная бесплатная альтернатива коммерческим решениям, например, Яндекс Картам. С ее помощью пользователи могут выбирать адрес на карте, видеть расположение аптек и в принципе получать визуальную обратную связь. Для подсказок и автозаполнения адресов используется Nominatim – сервис геокодирования, позволяющий находить адреса, вводя частичный текст. При выборе подходящего адреса на карте Nominatim возвращает все его компоненты, включая город, улицу, номер дома, почтовый индекс и координаты (широту и долготу). Интеграция реализована с помощью запросов к API сервиса. При каждом изменении поля адреса отправляется запрос, и пользователь видит полученную информацию. Кроме того, для получения информации об определенных элементах карты, а конкретно – аптек, используется API Overpass. Это инструмент для запросов к базе данных OpenStreetMap. Он позволяет получить все аптеки в административных границах города, района или ближайшие аптеки к введенному адресу. Также

через получение координат введенного адреса можно более точно оценивать время и стоимость доставки.

Электронная почта. Для отправки писем с кодами подтверждения, уведомлениями о заказах и др. необходима интеграция с почтовым сервисом через SMTP. В проекте была реализована такая интеграция с Gmail. Используется стандартный SMTP-протокол, предоставляемый Google. Это один из самых надежных и доступных способов отправки электронных писем.

Объектное хранилище. В системе реализовано внешнее хранилище для изображений товаров, аптек и брендов с помощью Cloudflare R2. Это S3-совместимое объектное хранилище, не требующее плату за исходящий трафик. Преимуществом такой интеграции является то, что изображения не хранятся в базе данных или на сервере приложения и ссылки на эти изображения доступны по публичным URL, которые можно напрямую использовать в клиентской части.

Текущая архитектура позволяет производить замену одних сервисов, на другие. Например, вместо Nominatim можно подключить DaData или Google Places API, ЮKassa можно заменить другой платежной системой, а в случае роста нагрузки вместо Gmail несложно использовать профессиональные почтовые сервисы, такие как Yandex360. Для этого весь интеграционный код изолирован в отдельных модулях.

Итак, система готова к взаимодействию с внешними сервисами от получения данных о лекарственных препаратах до обеспечения удобного клиентского пути при оформлении заказа.

2.5 Разработка базы данных

Для хранения данных интернет-аптеки разработана структура базы данных, включающая основные сущности предметной области и связи между ними. База данных построена в соответствии с нормализованной структурой, что обеспечивает удобство хранения, актуальность, целостность данных и

масштабируемость. Ниже на рисунке 2.4.1 представлена диаграммы базы данных.

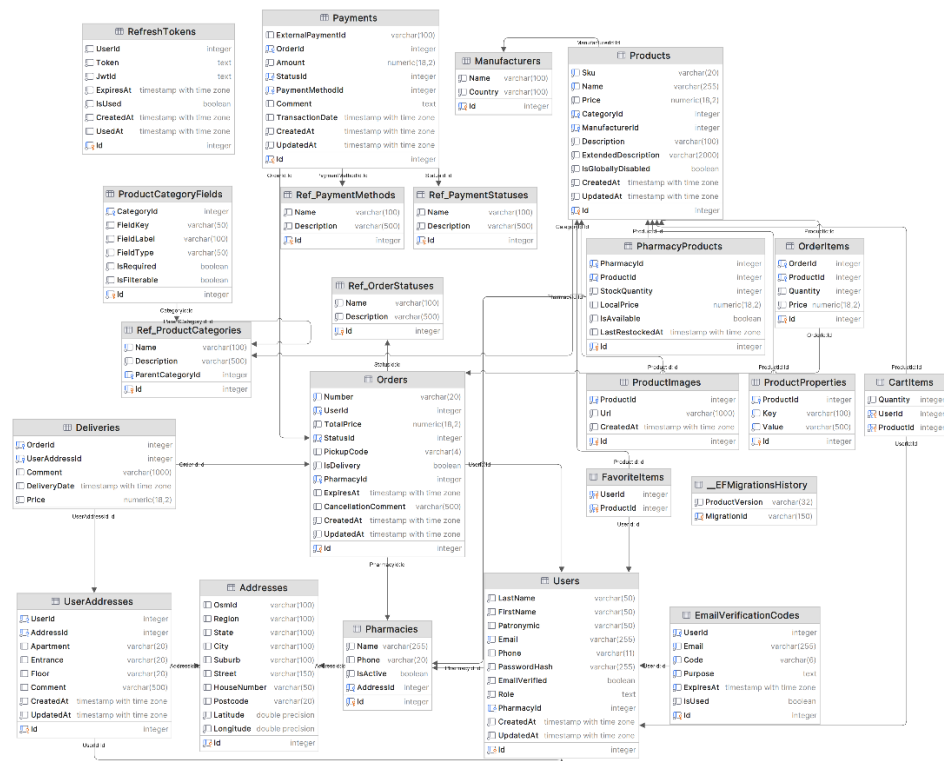


Рисунок 2.4.1 – Диаграмма базы данных

Основные сущности и их предназначения будут перечислены ниже.

Users – основная таблица пользователей системы. Здесь содержатся данные клиентов, сотрудников аптек и администраторов. Основные поля: фамилия, имя, отчество, контактные данные, а также хэш пароля (PasswordHash) и роль пользователя (RoleId). Имеются служебные поля CreatedAt и UpdatedAt для фиксации времени создания учетной записи и ее последнего изменения. Поле RoleId связано внешним ключом с таблицей ролей Ref_Roles.

RefreshTokens – таблица для хранения рефреш токенов авторизации, которые будут использоваться при истечении срока основного токена доступа для его обновления.

Ref_Roles – справочник ролей пользователей. Он содержит список возможных ролей и их описание. Например, роли “User”, “Admin”, “Employee”. Таблица позволяет разграничивать права доступа к функционалу. Клиенты могут оформлять заказы, сотрудники – подтверждать и выдавать, администраторы – управлять данными.

EmailVerificationCodes – таблица для хранения кодов подтверждения электронной почты. Она используется при регистрации пользователей или восстановления пароля, а также при смене почты. Генерируется уникальный шестизначный код, который отправляется на почту. Сохраняется запись с этим кодом, датой истечения и признаком использования. Когда пользователь вводит код, система сверяет его с существующими и отмечает его как использованный. Соответственно таблица реализует механизм верификации почты.

Products – основная таблица товаров (лекарственных средств и прочей продукции). Здесь хранятся названия, специальные индивидуальные номера, описание, цена, а также ссылки на категорию CategoryId и производителя ManufacturerId. Поле ExpirationDate указывает на срок годности товара.

Ref_Categories – справочник категорий товаров. Она содержит в себе название категории и описание. Поле родительской категории ParentCategoryId может ссылаться на саму себя. Категории используются для каталога и фильтрации.

ProductCategoryFields – таблица для индивидуальных характеристик каждой категории. Например, категория “Противовоспалительные и обезболивающие” может иметь обязательное поле “Дозировка”, в то время как “Витамины и минералы” не имеет. Имеет поля CategoryId для связи с категорией, FieldKey, FieldLabel, FieldType, IsRequired и IsFilterable.

ProductProperties – таблица со значениями индивидуальных характеристик товара. Содержит в себе поля ProductId, Key и Value. Key равен полю FieldKey в таблице ProductCategoryFields.

Manufacturers – таблица производителей. Содержит информацию о производителях лекарственных средств и товаров. Таблица связана с таблицей

Products. Это позволяет фильтровать товары по производителю и стране производства.

Orders – основная таблица заказов. Каждая запись представляет один заказ, оформленный клиентом. Каждый заказ имеет свой уникальный номер, статус заказа, временные метки, ссылку на пользователя, общую сумму заказа. Один пользователь может иметь несколько заказов, UserId – внешний ключ на Users.

Ref_OrderStatuses – справочник статусов заказа. Содержит список возможных статусов и их описание. Варианты статусов могут включать: “Ожидает оплаты”, “Ожидает обработки”, “В обработке”, “Готов к получению”, “Получен”, “Отменен”, “Возврат средств”, “Передан в доставку”, “Доставлен”. Это позволяет гибко изменять названия статусов без изменения основной таблицы заказов.

OrderItems – таблица состава заказа (позиции заказа). Она реализует связь многие-ко-многим между заказами и товарами. Каждый заказ может включать несколько товаров. А один и тот же товар может присутствовать в разных заказах. Дополнительные поля Quantity и Price показывают количество данного товара в заказе и цену на момент заказа. Хранение цены в OrderItems очень важно, поскольку цена товара может меняться со временем, а в составе конкретного заказа должна оставаться актуальная на момент оформления информация.

Payments – таблица для фиксации информации об оплате заказов. Поскольку возможны различные способы оплаты, например, онлайн и при получении, данная таблица универсальная для учета фактов оплаты. Таблица содержит в себе уникальный номер платежа, ссылку на заказ, сумму платежа, статус, способ оплаты и дату транзакции. Статус платежа показывает успешно ли прошла оплата, ожидается ли подтверждения, произошла ли ошибка и т.д.

Ref_PaymentMethods – справочник методов оплаты. Содержит варианты “Картой онлайн” и “При получении”. Используется вместе с таблицей Payments.

Ref_PaymentStatuses – справочник статусов оплаты. Содержит такие варианты, как “В ожидании”, “Завершено”, “Ошибка”, “Отменено”, “Не оплачено”, “Возвращено”.

CartItem – таблица для хранения содержимого корзины пользователя. По сути, это временные данные, которые превращаются в заказ при оформлении. Содержит UserId и ProductId, которые образуют составной первичный ключ, а также количество Quantity.

FavoriteItems – таблица списка желаний пользователя, его “избранное”. Поля UserId и ProductId формируют составной ключ.

ProductImages – таблица, которая хранит ссылки на изображения товаров, расположенные в облачном хранилище.

Pharmacies – основная таблица для хранения информации о самих аптеках. В ней находится информация об аптеке, в которой оформляется заказ и самовывоз или из которой будет произведена доставка.

Addresses – таблица, в которой хранятся данные об адресах. Она используется для обеспечения полноценной информации об аптеках и адресах пользователей для доставки. Имеет в себе идентификатор внешней системы, из которой получена информация об адресе, а также необходимые поля, такие как город, регион, улица, почтовый индекс, координаты и т.д.

UserAddresses – таблица для хранения адресов пользователей, которые используются для доставки заказа. Связана с таблицей Adressess и имеет собственные поля: подъезд, этаж, квартира.

Deliveries – таблица доставок. В ней хранятся данные по доставке заказа. Включает в себя идентификатор заказа и адреса пользователя, а также цену и фактической дату доставки.

При проектировании базы данных был использован нормализованный подход. Справочные данные вынесены в отдельные таблицы, а основные таблицы хранят только ссылки на них. У всех ключевых таблиц предусмотрены поля CreatedAt и UpdatedAt для хранения времени создания и последнего изменения записи. Индексы созданы по полям, которые часто используются в

условиях фильтрации и выборок, например, индекс по Products.Name для быстрого поиска товара по названию. Также в схеме базы данных предусмотрены внешние ключи с каскадными действиями и различные ограничения полей. Одним из главных ограничений является уникальность электронной почты в таблице Users.

Архитектура базы данных достаточно гибкая, чтобы при необходимости добавлять новые сущности, например реестр разрешений, чтобы хранить номер разрешения Росздравнадзора и онлайн-торговлю и дату получения для юридического лица.

По итогу, разработанная база данных обеспечивает хранение всей необходимой информации для работы интернет-аптеки. Ее структура оптимизирована под типовые операции: быстрый поиск товара, фильтрация по категориями, оформление заказа, авторизация пользователя и т.д.

2.6 Разработка алгоритмов работы системы

Перед реализацией отдельных модулей были спроектированы алгоритмы работы подсистем интернет-аптеки. Логика приложения разбита на отдельные вертикальные модули. Это означает, что для каждой операции есть свои эндпоинты, которые обрабатывают запрос и формируют ответ.

Исходя из функционала интернет-аптеки можно выделить основные алгоритмы работы информационной системы. Далее описаны эти процессы, участники и последовательность шагов выполнения.

2.6.1 Регистрация и авторизация

Алгоритм регистрации пользователя включает создание учетной записи и проверку адреса электронной почты через отправку кода подтверждения. Алгоритм регистрации обеспечивает, что в системе будут только подтвержденные учетные записи. Использование электронной почты в качестве точки входа повышает надежность и безопасность. Отправка писем реализована

через отдельный сервис уведомлений. Ниже на рисунке 2.6.1.1 приведена верхнеуровневая диаграмма последовательности для процесса регистрации.

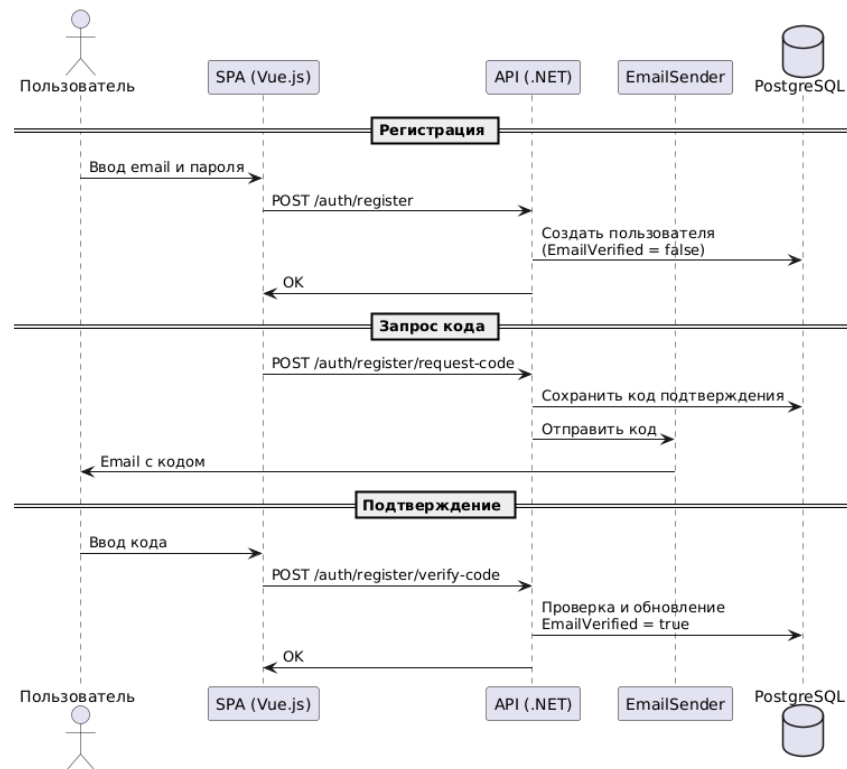


Рисунок 2.6.1.1 Диаграмма последовательности регистрации пользователя

Процесс можно разбить на подпроцессы:

1. Пользователь заполняет регистрационную форму, указывая необходимые данные. Бекенд получает эти данные через эндпоинт регистрации. Система проверяет валидность введенных данных и соответствие требованиями, например, что такая почта уже есть в базе. При обнаружении ошибок формируется ответ с описанием проблем.
2. Если данные верны, система хеширует пароль и создает новую запись в базе данных. На этом шаге аккаунт помечается как тот, что не прошел верификацию, поле `EmailConfirmed = false`. Такой функционал не дает ему полного доступа до тех пор, пока не будет подтвержден адрес электронной почты.

3. После создания пользователя генерируется шестизначный код подтверждения. Полученный код связывается с созданной учетной записью в специальной таблице базы данных.
4. Система генерирует электронное письмо для нового пользователя, которое содержит сгенерированный код подтверждения. Через настроенный SMTP-сервер почты отправляется письмо с кодом и просьбой ввести его в форму подтверждения на сайте.
5. После получения сообщения пользователь вводит полученный код подтверждения на веб-интерфейсе. Фронтенд передает код бэкенду. Система ищет ранее сохраненный код, связанный с указанной учетной записью, и сравнивает его с кодом, полученным от пользователя. Проверяется актуальность кода, возможно, истек срок действия кода. Если код совпадает и все еще действителен, система отмечает учетную запись как подтвержденную и устанавливает `EmailConfirmed=true`. Пользователь считается активированным и имеет полный доступ к системе. Сохраненный код подтверждения помечается как использованный и через некоторое время специальная фоновая задача очистит эту запись.

Модуль авторизации отвечает за проверку учетных данных при входе в систему и ограничение доступа к защищенным данным. В системе реализована авторизация JWT. При входе пользователя в систему генерируется специальный токен, содержащий зашифрованную информацию о пользователе и его правах. Токен затем используется для аутентификации в следующих запросах. Алгоритм для входа пользователя следующий:

1. Пользователь вводит адрес электронной почты и пароль на странице входа. Эти данные отправляются на бекенд через эндпоинт. Система находит пользователя с такой почтой в базе данных. Если пользователь не найден или пароль не совпадает, то возвращается ошибка аутентификации. На этом шаге проверяется

верификация почты. Если адрес не подтвержден, то будут повторяться шаги 3,4 и 5 как при регистрации.

2. Если адрес электронной почты и пароль верные и почта подтверждена, то генерируется токен JWT. Полученный токен возвращается клиенту в ответ на запрос авторизации.
3. После получения JWT фронтенд сохраняет его в локальном хранилище браузера. Этот токен будет автоматически прикрепляться в заголовке к каждому последующему запросу пользователя чтобы бекенд мог проверить личность вызывающей стороны.

Ниже на рисунке 2.6.1.2 представлена диаграмма последовательности, которая отражает процесс авторизации пользователя.

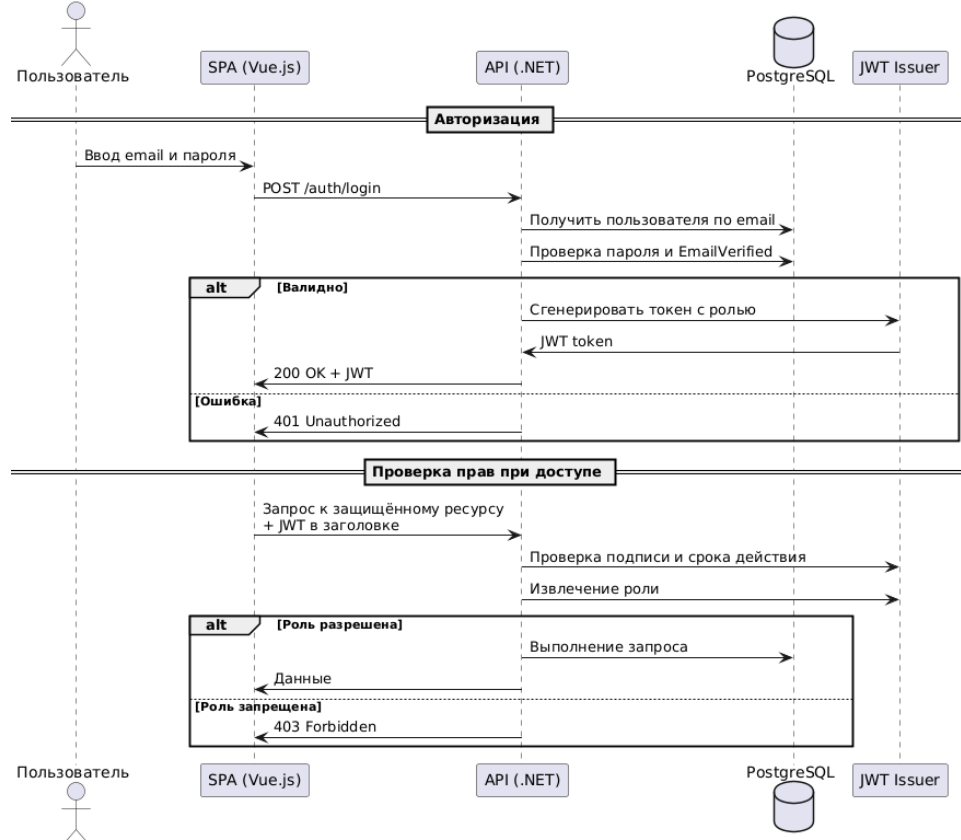


Рисунок 2.6.1.2 Диаграмма последовательности авторизации

Авторизация в системе основана на ролях пользователей. Для каждого защищенного процесса заранее определена роль, имеющая к нему доступ. Например, операции по управлению товарами и категориями разрешены только для роли администратора, тогда как операции по поиску продуктов и размещению заказов доступны авторизованным покупателям.

Таким образом, авторизация в системе двухуровневая. Сначала проверяется подлинность пользователя через JWT, а потом проверяются его права доступа на основе роли.

2.6.2 Поиски фильтрация товаров

Модуль поиска нужен для быстрого нахождения товаров по различным критериями. Алгоритм поиска реализован через эндпоинт, который принимает параметры фильтрации возвращает список товаров. Ниже на рисунке 2.6.2.1 показан процесс поиска товаров.

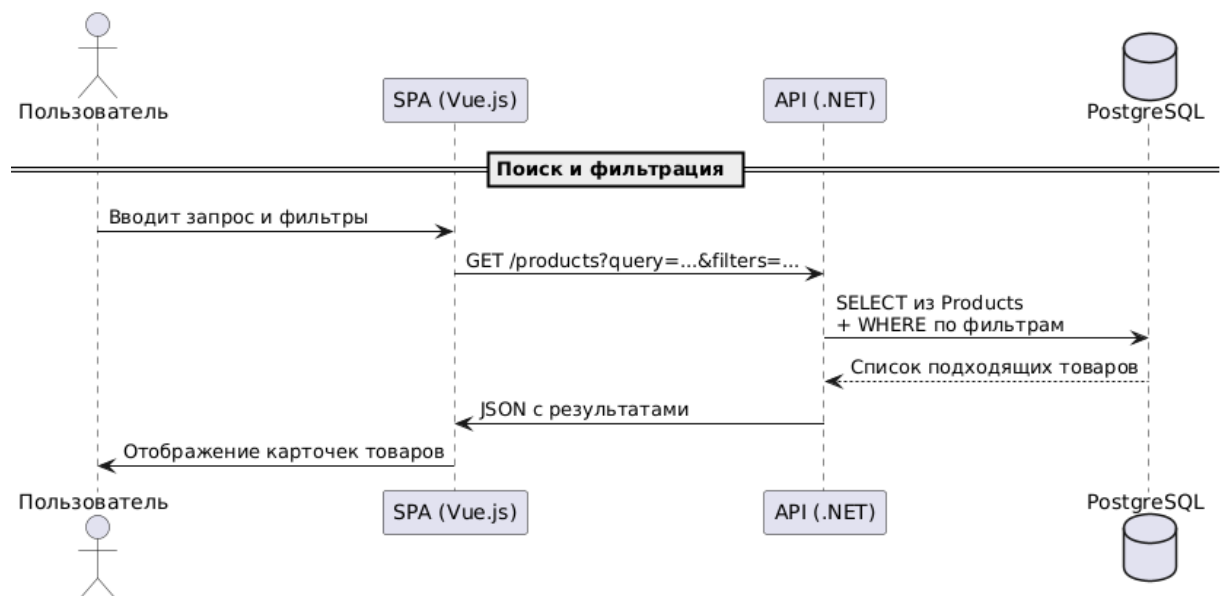


Рисунок 2.6.2.1 Диаграмма последовательности для поиска товара с фильтрацией

При поиске товаров с фильтрацией происходят следующие действия:

1. Клиент может указать строку поиска и дополнительные фильтры: категорию лекарства, производителя, индивидуальные поля и т.д. Эти параметры передаются в запросе (в строке запроса или в теле). Эндпоинт получает параметры и валидирует их. Если параметры неверны, будет возвращен ответ, содержащий ошибку и пояснение.
2. Если параметры валидны, то формируется запрос к базе данных. Если указано слово для поиска, то выполняется фильтрацию по названию. Если задан фильтр по категории, то добавляется условие на совпадение категории товара. Аналогично работает для всех параметров.
3. Для улучшения вывода при большом количестве результатов алгоритм поддерживает пагинацию – например, на первой странице он возвращает только 20 товаров при найденных 200. На второй странице он пропустит первые 20 и возьмет следующие 20 и отобразит их.
4. Эндпоинт возвращает клиенту объект JSON, содержащий найденные продукты. Если по указанным условиям не найдено ни одного товара, то будет возвращен пустой список или специальное сообщение об отсутствии результатов.

В процессе работы алгоритма поиска учитываются разные оптимизации. Например, для поиска строки по названию имеется индекс в базе данных по полю названия. Кроме того, поиск доступен даже неавторизованным пользователям, поскольку просмотр продуктов обычно не требует специальных прав.

2.6.3 Корзина и оформление заказа

Модуль корзины и оформления заказа отвечает за то, чтобы пользователь мог выбрать нужные товары, сохранить их для покупки, а затем оформить заказ. Алгоритм включает в себя операции с корзиной и создание заказа на

основе содержимого корзины. Также он гарантирует консистентность данных. Это означает, что все операции происходят как единое целое (транзакция). Если на каком-то шаге произошла ошибка, то транзакция откатывается, а все состояние возвращается в то, что было до нее. Ниже на рисунке 2.6.3.1 отображена диаграмма, показывающая данный процесс.

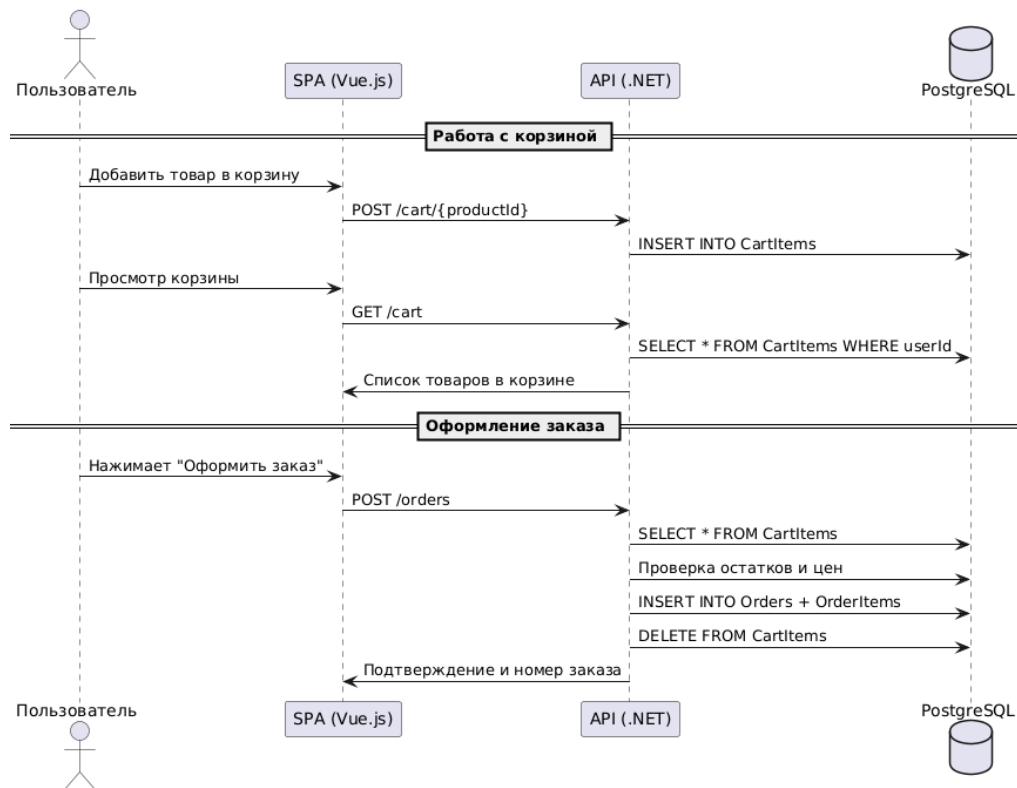


Рисунок 2.6.3.1 Диаграмма последовательности управления корзиной и оформления заказа

Корзина является временным местом хранения выбранных товаров пользователя. При реализации системы корзина привязана к его учетной записи. В текущем алгоритме работают следующие процессы:

1. Пользователь может добавить товар из каталога в корзину. На фронтенде это инициирует вызов эндпоинта добавления в корзину. Бекенд в обработчике этого заказа проверяет, авторизован ли пользователь (требуется токен JWT, в противном случае заказ будет отклонен), существует ли товар с таким идентификатором и доступен ли товар. Если проверки пройдены успешно, система

либо создает новую запись в корзине, либо обновляет количество, если товар уже находится там. После успешного добавления сервер возвращает обновленный список содержимого корзины или статус успеха.

2. В любой момент пользователь может запросить получение своей корзины. Эндпоинт берет все ожидающие товары для указанного пользователя из базы данных и возвращает их в виде списка с подробностями. Предусмотрены операции по изменению количества или удалению товара из корзины.

Когда пользователь заканчивает выбор товаров, он переходит к оформлению покупки. Алгоритм формирования заказа основан на данных из корзины:

1. Пользователь нажимает кнопку для оформления заказа. Сначала все товары в корзине покупок этого пользователя будут извлечены из базы данных. Если корзина пуста, будет возвращена ошибка.
2. Перед созданием заказа система еще раз проверяет корректность каждого товара. Эта проверка нужна, поскольку некоторые товары могли измениться с момента добавления в корзину. Если какой-либо товар отсутствует, то процесс оформления заказа прерывается, а пользователю возвращается ошибка.
3. Если все тесты пройдены, система формирует заказ. В базе данных создается новая запись, содержащая основную информацию заказа. Затем для каждого товара в корзине создаются записи `OrderItem`. Эти данные сохраняются в рамках транзакции. И либо сохраняются все элементы и заказ, либо запись не сохраняется.
4. После успешного создания заказа товары из корзины пользователя удаляются. Теперь корзина пуста.
5. После формирования заказа система возвращает ответ с подробностями выполненного заказа. Дополнительно отправляется уведомление по электронной почте.

2.6.4 Оплата заказа

2.6.5 Администрирование

Для предоставления возможностей администрирования в системе реализован отдельный модуль. Административные функции включают управление каталогом продукции, категориями, заказами, а также учетными записями пользователей. Ниже на рисунке 2.6.5.1 рассмотрены основные алгоритмы управления товарами.

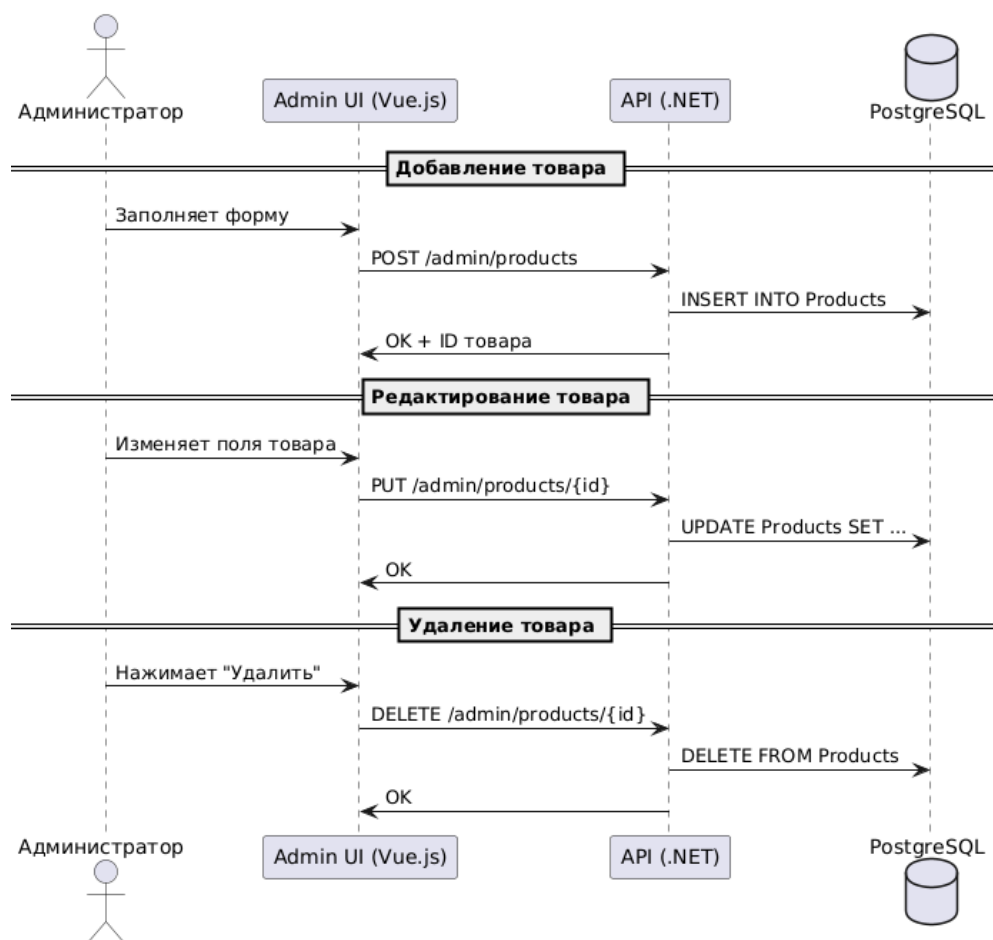


Рисунок 2.6.5.1 Диаграмма последовательности для управления товарами

Администратор может выполнять CRUD-операции с товарами. Создавать новые позиции, обновлять существующие и удалять их из ассортимента. Алгоритмы таких операций очень типичны:

- **Добавление товара.** Администратор вводит данные о новом продукте через панель управления. Нажимает “Добавить” и вызывает эндпоинт создания. Сервис валидирует введенные данные. Затем создается объект продукта, который сохраняется в базе данных. В случае успеха ответ будет содержать идентификатор добавленного продукта.
- **Редактирование товара.** Чтобы изменить существующий товар, администратор открывает форму редактирования, изменяет обязательные поля и отправляет запрос. Если товар не найден, то возвращается ошибка. Если обновление прошло успешно, ответ просто возвращает статус код 200.
- **Удаление товара.** Нажатие кнопки удаления отправляет запрос на удаление, в котором указывается идентификатор продукта. Эндпоинт проверяет, существует ли продукт с таким же идентификатором. Если да, он удаляет запись из базы данных. Ответ возвращает успех или сообщение об ошибке.

После любых изменений каталога панель администратора обновляет отображение списка товаров, показывая актуальные данные.

Администратор также имеет возможность просматривать список пользователей системы и при необходимости изменять их роли и не только. Описание алгоритмов следующее:

- **Просмотр пользователей.** Администратор может запросить список всех пользователей. Эндпоинт возвращает список учетных записей. Администратор может применять фильтры по роли или искать пользователя по его почте. Эти запросы обрабатываются аналогично поиску товаров с фильтрацией.
- **Изменение роли пользователя.** Администратор может расширить или ограничить права конкретного пользователя. Это делается через отдельный эндпоинт, который принимает идентификатор

пользователя и роль. Система проверяет, что было запрошено допустимое значение роли. Далее запись пользователя назначается новая роль и изменения сохраняются.

2.6.6 Электронная почта и уведомления

Модуль уведомлений по электронной почте отвечает за автоматическую отправку электронных писем пользователям при возникновении определенных событий в системе. Основные сценарии уведомлений по электронной почте такие:

- Как было показано выше, после регистрации на адрес электронной почты отправляется письмо с кодом подтверждения. Этот шаблон электронного письма содержит приветствие, инструкции по подтверждению, а также сам код.
- После создания заказа система генерирует подтверждающее письмо. Сообщение включает номер заказа, список приобретенных товаров, общую сумму и информацию о доставке. Это электронное письмо позволяет пользователю убедиться, что запрос зарегистрирован в системе. Пример такого письма приведен ниже на рисунке 2.5.6.
- Если пользователь запрашивает сброс пароля, ему будет отправлено электронное письмо, содержащее шестизначный код для сброса пароля.
- Если пользователь меняет почту, то на новую почту будет отправлено электронное письмо, которое содержит шестизначный код для смены почты.

Здравствуйте, Иванов Иван!

Ваш заказ **ORD-C48B766A** успешно оформлен.

Состав заказа:

Товар	Кол-во	Цена	Сумма
Амоксициллин 500 мг	2	120,00 Р	240,00 Р
Кальций-Д3 Никомед	1	399,00 Р	399,00 Р

Сумма заказа: 639,00 Р

Адрес аптеки: Москва, улица Плещеева, 3, 127560, Ваша №1

Способ оплаты: Картой онлайн

Статус заказа: Ожидает оплаты

Спасибо, что выбрали нашу аптеку!

Рисунок 2.5.6 Пример электронного письма с подтверждением заказа

Шаблоны писем могут храниться в виде встроенных строк. Например, для подтверждения по электронной почте шаблон содержит статический текст и поле для кода подтверждения. Затем вызывается SMTP-клиент и параметры сообщения передаются в систему. Передача осуществляется асинхронно.

Описанные алгоритмы работы показывают, как разработанная информационная система функционирует в типичных сценариях использования.

В данной главе были определены используемые технологии, спроектирована структура и архитектура базы данных. Предлагаемое решение основано на современном стеке. Трехуровневая архитектура разработана для масштабирования и легкой интеграции с внешними сервисами (ЮKassa, Cloudflare R2, SMTP и др.). Сложная структура базы данных охватывает все основные сущности. Рассмотрены механизмы масштабируемости. В конце были описаны основные алгоритмы, протекающие в информационной системе.

ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ

3.1 Реализация основных модулей

Серверное приложение ASP.NET Core разбито на ряд модулей в соответствии с функциональными областями. В проекте применяется библиотека FastEndpoints для организации REST API. Вместо традиционных контроллеров каждый маршрут описан отдельный классом-эндпоинтом, что повышает модульность. Для доступа к базе данных используется Entity Framework Core с провайдером Npgsql для PostgreSQL. Логика работы вынесена в сервисы, которые вызываются эндпоинтами. Благодаря этому достигается разделение на слои. Слой API отвечает за обработку запросов и валидацию, бизнес-логика инкапсулирована в сервисах, а слой управления данными через контекст EF Core. Ниже подробно рассмотрены ключевые модули системы.

3.1.1 Модуль авторизации и регистрации

Данный модуль отвечает за управление учетными записями пользователей. Он включает создание новых пользователей, проверку учетных данных для входа и выдачу токенов JWT для аутентификации. При регистрации через эндпоинт `/authorization/register-initial` необходимые данные передаются и проверяются. Проверка выполняется с использованием встроенных атрибутов и отдельных классов проверки на основе FluentValidation (листинг 3.1.1). Пароль пользователя перед сохранением шифруется. Открытый пароль не сохраняется в базе данных, а только защищенный хэш. После успешной регистрации создается учетная запись с ролью пользователя по умолчанию и неподтвержденным адресом электронной почты. Выполняется функция входа в систему через эндпоинт `/authorization/login`. Здесь проверяется адрес электронной почты и пароль и в случае успеха генерирует токен JWT. Токен включает идентификатор пользователя и роль. Этот токен JWT используется клиентом для доступа к защищенным ресурсам. Во всех следующих запросах токен передается в заголовке и сервер автоматически проверяет аутентификацию. Модуль также

поддерживает выход из системы на стороне клиента, при котором фронтенд удаляет токен JWT из хранилища.

Листинг 3.1.1 Класс валидации данных при регистрации

```
public class RegisterRequestValidator : Validator<RegisterRequest>
{
    public RegisterRequestValidator()
    {
        RuleFor(x => x.Email)
            .NotEmpty()
            .EmailAddress();

        RuleFor(x => x.Password)
            .MinimumLength(6)
            .NotEmpty();

        RuleFor(x => x.FirstName)
            .NotEmpty()
            .MaxLength(50);

        RuleFor(x => x.LastName)
            .NotEmpty()
            .MaxLength(50);

        RuleFor(x => x.Patronymic)
            .MaxLength(50);

        RuleFor(x => x.Phone)
            .MaxLength(11)
            .Matches(@"^7\d{10}$")
            .When(x => !string.IsNullOrEmpty(x.Phone))
            .WithMessage("Введите номер в формате 7XXXXXXXXXX");
    }
}
```

Для повышения безопасности и надежности данных реализовано подтверждение адреса электронной почты пользователя. После регистрации система отправляет на электронную почту пользователя специальный шестизначный код подтверждения. Отправка осуществляется через SMTP-сервер Gmail с использованием библиотеки MailKit. Класс EmailSender реализует интерфейс отправки электронной почты: он составляет сообщение и устанавливает соединение с SMTP. Вот фрагмент кода, демонстрирующий, как

подключиться к почтовому серверу Gmail и отправить электронное письмо, содержащее код.

Листинг 3.1.2 Отправка сообщения через SMTP-сервер Google

```
var message = new MimeMessage();
message.From.Add(MailboxAddress.Parse(_configuration["SmtpSettings:Sender"]));
message.To.Add(MailboxAddress.Parse(to));
message.Subject = subject;

var builder = new BodyBuilder { HtmlBody = body };
message.Body = builder.ToMessageBody();

using var smtp = new SmtpClient();

await smtp.ConnectAsync(
    _configuration["SmtpSettings:Host"],
    int.Parse(_configuration["SmtpSettings:Port"]),
    SecureSocketOptions.StartTls);

await smtp.AuthenticateAsync(
    _configuration["SmtpSettings:Username"],
    _configuration["SmtpSettings:Password"]);

await smtp.SendAsync(message);
await smtp.DisconnectAsync(true);
```

При регистрации генерируется случайный проверочный код и сохраняется в БД. Письмо содержит этот код и инструкцию по подтверждению. Пользователь вводит код на странице подтверждения аккаунта, после чего эндпоинт `/verifications/confirm-code` проверяет соответствие кода и помечает учетную запись как подтвержденную. Модуль также включает восстановление пароля. Эта операция также проходит валидацию. Например, проверяется совпадение полей нового пароля и его подтверждения, срок действия кодов и т.д. Неподтвержденные аккаунты имеют ограниченный доступ к функционалу системы.

После входа в систему пользователь получает доступ к своему личному кабинету – интерфейсу, позволяющему управлять своими персональными данными. Модуль “Личный кабинет” реализован через набор эндпоинтов

Например, эндпоинт GET /users/profile позволяет получить информацию профиля текущего пользователя. Реализована возможность изменения некоторых данных – например, пользователь может обновить свое имя или пароль. История заказов пользователя также отображается в личном кабинете. Специальный эндпоинт возвращает список заказов для текущего пользователя, включая их состояние и конфигурацию. Пользователь может просмотреть детали каждого заказа. Данная функция обеспечивает связь модуля личного кабинета, модуля обработки заказов и каталога. Кроме того, реализована возможность ведения избранного. Пользователь может отмечать товары как “избранные”. Модуль “Избранное” похож на корзину покупок. Он имеет отдельную сущность базы данных и эндпоинты /users/favorites для получения списка избранных продуктов и их добавления или удаления. В итоге, личный кабинет объединяет в себе различные функции персонализации и .

ЗАКЛЮЧЕНИЕ

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

ПРИЛОЖЕНИЕ А

Техническое задание

1. Наименование и область приложения

Наименование темы работы: «Разработка информационной системы для автоматизации процессов интернет-аптеки». Разрабатываемая система предназначена для онлайн-продажи лекарственных товаров через веб-интерфейс. Сервис поддерживает оформление заказов, управление товарами и обработку платежей, а также содержит административные и пользовательские разделы.

2. Основания для разработки

Разработка ведётся в рамках дипломного проекта, утверждённого кафедрой «Информатика и информационные технологии» Московского политехнического университета.

3. Исполнитель

Исполнителем является студент кафедры «Информатика и информационные технологии» Московского политехнического университета группы 211-723, Петров Вадим Станиславович.

4. Назначение разработки

1.1 Назначение системы

Обеспечение пользователям возможности поиска и заказа аптечных товаров через единый веб-сервис.

1.2 Цели создания системы

- 1) Повышение доступности и удобства покупки лекарственных средств;
- 2) Упрощение учёта заказов, оплаты и доставки;
- 3) Предоставление административного интерфейса для управления ассортиментом, пользователями и заказами;
- 4) Предоставление интерфейса для администратора для управления заказами аптеки.

5. Технические требования к системе

1.1 Задачи, подлежащие решению

- 1) Анализ предметной области и имеющихся решений;
- 2) Проектирование архитектуры системы;
- 3) Реализация базы данных;
- 4) Разработка серверной части;
- 5) Разработка клиентской части;
- 6) Интеграция внешних сервисов;
- 7) Реализация панели администратора и сотрудника;
- 8) Тестирование и подготовка инструкции по развёртыванию;
- 9) Ввод системы в эксплуатацию.

1.2 Требования к функциональным характеристикам

Информационная системы должна предоставлять следующие функции:

- 1) Регистрация и авторизация пользователей;
- 2) Просмотр каталога товаров и поиск по фильтрам;
- 3) Добавление товаров в избранное и корзину;
- 4) Оформление заказа с выбором аптеки или адреса доставки;
- 5) Сохранение пользовательских адресов доставки;
- 6) Приём и обработка платежей;
- 7) Просмотр истории заказов и статусов их выполнения;
- 8) Функции администратора: управление пользователями, товарами, категориями товаров, заказами, аптеками и их складами;
- 9) Функции сотрудника: управление складом аптеки, статусами заказов и платежей.

1.3 Требования к входным и выходным данным

Информационная система принимает на вход следующие данные:

- 1) Данные для аутентификации и регистрации:
 - а) Адрес электронной почты;
 - б) Пароль;
 - в) Код подтверждения;

- г) ФИО пользователя;
 - д) Повтор пароля (для подтверждения при регистрации).
- 2) Поисковые и фильтрационные запросы:
- а) Ключевые слова;
 - б) Категории и подкатегории;
 - в) Производители;
 - г) Фильтры по свойствам;
 - д) Сортировка.
- 3) Данные корзины и оформления заказа:
- а) Идентификаторы и количество выбранных товаров;
 - б) Адрес доставки;
 - в) Способ оплаты;
 - г) Выбор существующего адреса или ввод нового.
- 4) Административные данные:
- а) Информация о товарах;
 - б) Категории и подкатегории товаров;
 - в) Информация об аптеках и их адресах;
 - г) Данные о пользователях.

Информационная система предоставляет следующие выходные данные:

- 1) Пользователю (клиенту):
- а) Список товаров с изображениями, ценами, описанием и фильтрацией;
 - б) Информация о конкретном товаре;
 - в) Информация о текущем состоянии корзины;
 - г) Детали и статус оформления заказа;
 - д) Подтверждение оплаты;
 - е) История заказов;
 - ж) Уведомления об успешных или неудачных действиях.
- 2) Администратору:
- а) Панель управления товарами;

- б) Список пользователей, заказов и аптек;
 - в) Возможность управления доступами и ролями сотрудников.
- 3) Сотруднику аптеки:
- а) Список заказов, оформленных на его аптеку;
 - б) Информация о наличии товаров на складе;
 - в) Статусы заказов и оплат.

1.4 Требования к составу программных компонентов

Система должна содержать следующие программные компоненты:

- 1) Клиентское приложение;
- 2) Серверное приложение;
- 3) База данных;
- 4) Объектное хранилище изображений;
- 5) Сторонние API.

1.5 Требования к компонентам программного продукта

Клиентское приложение:

- 1) Адаптивный и интуитивно понятный интерфейс;
- 2) Индикация ошибок и состояний загрузки;
- 3) Страницы должны быть логично разделены по функциональности;
- 4) Действия пользователя сопровождаются визуальной обратной связью.

Серверное приложение:

- 1) Валидация входящих запросов и данных;
- 2) Проверка авторизации и прав доступа;
- 3) Подробная документация API;
- 4) Безопасность API;
- 5) Поддержка внешних сервисов;
- 6) Производительность;
- 7) Мониторинг и логирование.

База данных:

- 1) Безопасность;

- 2) Целостность данных;
- 3) Производительность;
- 4) Масштабируемость;
- 5) Конфиденциальность.

1.6 Требования к составу и параметрам технических средств

1.6.1 Требования к составу программного обеспечения

- 1) Операционные системы:
 - а) Windows версии от 7 и выше;
 - б) macOS версии от 14 и выше;
 - в) Современные дистрибутивы Linux;
 - г) iOS версии от 14.5 и выше;
 - д) Android версии от 9.0 и выше.
- 2) Веб-браузеры:
 - а) Google Chrome версии 90 и выше;
 - б) Microsoft Edge версии 90 и выше;
 - в) Mozilla Firefox версии 90 и выше;
 - г) Safari версии 14.5 и выше;
 - д) Opera версии 80 и выше.
- 3) Прочие ПО:
 - а) Устройства ввода должны поддерживать работу с клавиатурой (физической или экранной);
 - б) Поддержка JavaScript и HTML5.

1.6.2 Требования к аппаратному обеспечению

- 1) Стабильное подключение к сети интернет;
- 2) Смартфон, планшет, ноутбук или стационарный ПК;
- 3) Разрешение экрана не менее 1280x720 пикселей;
- 4) Оперативная память от 2 ГБ;
- 5) Процессор с тактовой частотой от 1.4 ГГц;
- 6) Свободное место в хранилище от 100 Мб (для кэша и локальных данных браузера);

7) В случае стационарных компьютеров – мышь, клавиатура.

1.7 Условия эксплуатации

Веб-приложение может быть использовано на мобильных устройствах, стационарных компьютерах и ноутбуках. Пользователю необходимо иметь постоянное подключение к интернету, использовать поддерживаемый браузер. Без входа в систему доступны только функции добавления в корзину и избранное, просмотра продуктов и категорий. Возможность оформлять заказы, просматривать историю заказов и управлять продуктами и аптеками доступна только авторизованным пользователям, каждый из которых соответствует своей роли.

6. Стадии и этапы разработки

Название этапа	Даты реализации этапа
Анализ предметной области	16.11.2024 – 31.12.2024
Разработка архитектурной модели	01.01.2025 – 15.01.2025
Проектирование и разработка базы данных	06.01.2025 – 01.02.2025
Реализация и тестирование базового API	02.02.2025 – 01.03.2025
Доработка функционала	02.03.2025 – 15.04.2025
Реализация интерфейсов всех типов пользователей	16.04.2025 – 24.05.2025
Связь клиентской и серверной части	25.05.2025 – 10.06.2025
Итоговое тестирование всей системы	11.06.2025 – 20.06.2025

7. Техническая документация, предоставляемая по окончании работы

- 1) Схема архитектуры системы в целом;
- 2) Схема структуры базы данных;
- 3) Описание API серверного приложения;

- 4) Описание модулей клиентской и серверной частей;
- 5) Информация о процессе отладки и тестирования.

ПРИЛОЖЕНИЕ Б

Презентация