

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем  
Направление подготовки 09.04.04 – Программная инженерия  
Направленность (профиль) образовательной программы Управление  
разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов  
« \_\_\_\_ » \_\_\_\_\_ 2025 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему: Проектирование и реализация информационной системы  
здравоохранения

Исполнитель студент группы 3105-ом2	_____	Фан Цзинчао
	(подпись, дата)	
Руководитель доцент, канд. техн. наук	_____	А.В. Бушманов
	(подпись, дата)	
Руководитель научного содержания программы магистратуры профессор, доктор техн. наук	_____	И.Е. Еремин
	(подпись, дата)	
Нормоконтроль инженер кафедры	_____	В.Н. Адаменко
	(подпись, дата)	
Рецензент старший преподаватель, канд. техн. наук	_____	Д.В. Фомин
	(подпись, дата)	

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Институт компьютерных и инженерных наук  
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

\_\_\_\_\_ А.В. Бушманов  
« \_\_\_\_ » \_\_\_\_\_ 2024 г.

**З А Д А Н И Е**

К магистерской диссертации студента группы 3105-ом2 Фан Цзинчао

1.Тема магистерской диссертации: Проектирование и реализация  
информационной системы здравоохранения

(Утверждено приказом от 06.03.2025 № 609-уч)

2.Срок сдачи студентом законченной работы (проекта): 10.06.2025

3.Исходные данные к магистерской диссертации: документация  
разработчиков, интернет ресурсы, учебная литература

4.Содержание магистерской диссертации (перечень подлежащих  
разработке вопросов): Здравоохранение, проектирование алгоритма  
решения задачи, разработки системы управления здравоохранением

5.Перечень материалов приложения (наличие чертежей, таблиц, графиков,  
схем, программных продуктов, иллюстративного материала и т.п.):

6.Рецензент магистерской диссертации: Фомин Д.В., старший  
преподаватель, канд. техн. наук

7.Дата выдачи задания 29.01.2024

Руководитель выпускной квалификационной работы: А.В. Бушманов,  
доцент, канд. техн. наук

(фамилия, имя, отчество, должность, уч.степень уч.звание)

Заявление принял к исполнению \_\_\_\_\_

## РЕФЕРАТ

Магистерская диссертация содержит 80 страниц, 20 рисунков, 7 таблицы  
30 источников

### ОНЛАЙН-ПЛАТФОРМА ЗДОРОВЬЯ, СИСТЕМА JAVA ЗДОРОВЬЯ, АНАЛИЗ СПРОСА, SPINGBOOT

В условиях стремительного развития современных технологий персонал уделяет здоровью все больше внимания, прежняя единая офлайн-консультация в больнице не позволяет населению своевременно понять быструю эффективность состояния, поэтому внедрение единой онлайн-платформы управления здоровьем может быть более удобным для населения в управлении телом. В данной статье представлены конкретные функции системы управления сокровищами здоровья. Прежде всего, через анализ пользовательского спроса, чтобы понять спрос пользователей на программное обеспечение сокровища здоровья и для этого проектирования и реализации функций системы. В данной работе используется блок-схема для окончательной доработки и разработки структуры каждого функционального модуля системы и соответствующего кода. Система управления здоровьем на базе Java может повысить корректность и эффективность доступа пользователя к медицинской информации, что позволит сэкономить время, затрачиваемое пользователем на поиск информации, и сэкономить человеко-часы ручного вмешательства. В этой статье, бэк-энд SpringBoot структуры, так что процесс его работы является чрезвычайно удобным, значительно сократив время, необходимое для разработки системы обслуживания и модернизации также очень удобно, проект также встроенный jar-пакеты объединены Java инструменты управления проектами для управления. После пробного запуска платформа может работать стабильно, но также имеет определенное значение.

## СОДЕРЖАНИЕ

Введение	6
1 Анализ систем здравоохранения и контроля здоровья	7
1.1 Текущее состояние отечественных и международных исследований	7
1.1.1 Современное состояние отечественных исследований	7
1.1.2 Текущее состояние зарубежных исследований	9
1.2 Содержание исследования	12
1.3 Инструменты разработки и внедрение технологий	13
1.4 Техничко-экономический анализ функций системы	17
1.4.1 Техническая реализуемость	17
1.4.2 Экономическая целесообразность	18
1.4.3 Операционная целесообразность	19
1.4.4 Юридическая целесообразность	19
1.5 Анализ системных требований	20
2 Разработка и внедрение систем управления здравоохранением	22
2.1 Дизайн функций системы	22
2.2 Тестирование системного каркаса	23
2.3 Проектирование базы данных	25
2.3.1 Концептуальный дизайн базы данных	25
2.3.2 Логическая структура базы данных	26
2.4 Каталог структур проекта	33
2.5 Функциональный дизайн и реализация	35
2.5.1 Разработка и реализация интерфейса входа и регистрации	35
2.5.2 Разработка и внедрение управления корзиной покупок	38
2.5.3 Разработка и внедрение системы управления товарами	39
2.5.4 Разработка и реализация экспертных рекомендаций	41
3 Реализация и оценка системы мониторинга здоровьем	45
3.1 Обзор дизайна системы	45

3.2 Дизайн функциональных модулей	48
3.3 Витрина интерфейсов	53
3.4 Архитектура системы и реализация модулей	64
3.5 Тестирование системы	66
3.5.1 Аппаратная среда	68
3.5.2 Программная среда	68
3.5.3 Тестовый пример	68
3.5.4 Краткое описание теста	73
3.6 Будущие улучшения и расширения	74
Заключение	76
Библиографический список	77

## ВВЕДЕНИЕ

В современных социальных условиях, в связи с ростом возраста людей, сопротивляемость организма будет снижаться с каждым днем, единая оффлайн консультация больницы уже не удобна для населения, чтобы понять быструю эффективность состояния своевременной консультации, внедрение единой онлайн-платформы управления здоровьем может достичь людей в жизни в момент может проконсультироваться, чтобы определить состояние организма, если организм имеет проблемы могут быть даны планы лечения. И превращение сервиса в основную функциональную систему - это главное, что нужно довести до совершенства на данном этапе, поэтому необходимо срочно использовать полученные знания для ее развития. В процессе создания управленческого приложения HealthPower в различных медицинских компаниях существуют отзывчивые системы управления. Однако их запутанная среда приводит к проблеме высокой сложности использования самой системы. В сочетании со сложностью различных регионов отечественные медицинские магазины трудно поддаются комплексному контролю, в результате чего во многих случаях не удастся осуществлять комплексное управление. Часть сайтов системы не способна достичь всеобъемлющего охвата. Кадровые службы находятся в состоянии блокады, что приводит к невозможности более качественно обслуживать население. Основная цель системы управления сокровищами здоровья на базе Java - решить проблему поиска пользователями знаний и услуг, через форму науки и техники обслуживать пользователей, чтобы они понимали соответствующие знания, через знание собственного состояния. Чтобы помочь пользователю онлайн-операции и эксперт онлайн стыковки, пользователь может быть основан на соответствующих знаний, чем физическое состояние, но и может купить медицинские заказы, администраторы могут контролировать процесс обслуживания, и пластичность системы, очень сильный очень удобно добавлять новые функции впоследствии.

# 1 АНАЛИЗ СИСТЕМ ЗДРАВООХРАНЕНИЯ И КОНТРОЛЯ ЗДОРОВЬЯ

## 1.1 Текущее состояние отечественных и международных исследований

### 1.1.1 Современное состояние отечественных исследований

Сяни Хэ отметил, что с ростом популярности Интернета различные онлайн-сервисы значительно облегчили повседневную жизнь людей. Современные цифровые технологии, особенно в сфере Интернета и мобильных приложений, радикально меняют многие аспекты общественной жизни, и здравоохранение здесь не является исключением. В последние годы наблюдается стремительный рост онлайн-сервисов, таких как виртуальные клиники и платформы медицинского консалтинга, которые позволяют пациентам не только удобно записываться на приём, но и проводить видеоконсультации, получать электронные рецепты, просматривать свои медицинские карты и общаться с врачами в режиме. Это повышает доступность медицинской помощи для жителей удалённых регионов, людей с ограниченной мобильностью или занятых работой, а также способствует более эффективному распределению ресурсов. Дополнительную пользу приносит интеграция с устройствами дистанционного мониторинга: данные о состоянии здоровья автоматически передаются медикам, позволяя оперативно реагировать на изменения и предлагать персонализированные рекомендации, что ведёт к повышению качества и экономической эффективности медицинских услуг.

Кроме того, с развитием технологий онлайн-и мобильных платежей появилась новая потребность возможность оплаты медицинских услуг через Интернет. Мобильные приложения предоставляют пациентам удобство оплачивать медицинские услуги в любое время и в любом месте, исключая необходимость посещать кассу или специализированные отделы для проведения оплаты. Это значимо улучшает пользовательский опыт, ускоряет процесс обслуживания и значительно сокращает очереди в медицинских

учреждениях. Более того, система онлайн-оплаты снижает нагрузку на медицинский персонал, позволяя быстрее и проще обрабатывать финансовые транзакции, что ведет к оптимизации процессов и повышению качества обслуживания в целом.

Таким образом, сочетание различных онлайн-приложений для управления здоровьем пациентов имеет огромный потенциал для решения множества проблем, включая более лёгкий доступ к медицинским консультациям, улучшение контроля за состоянием здоровья и своевременный доступ к медицинской помощи. Эти приложения могут включать не только медицинские рекомендации, но и функции мониторинга состояния здоровья, напоминания о приёме лекарств, а также интеграцию с носимыми устройствами и фитнес-трекерами. Благодаря таким функциям пользователи могут контролировать свои параметры здоровья в реальном времени, что способствует более раннему выявлению возможных рисков и позволяет медицинским специалистам своевременно предоставлять персонализированные рекомендации. В конечном итоге такие технологии способствуют улучшению качества жизни и эффективному решению множества проблем, связанных с поддержанием и улучшением здоровья.

С трёх аспектов управления услугами – персонала, пользователей и экспертов — крайне важно создавать различные сервисные платформы, которые могут удовлетворить потребности всех участников. Для специалистов это означает создание удобных и функциональных инструментов для удалённой диагностики, мониторинга состояния здоровья пациентов и эффективного взаимодействия с ними. Благодаря таким инструментам специалисты смогут более точно и своевременно оценивать здоровье пациента и предоставлять персонализированные рекомендации. Для пациентов ключевым является обеспечение быстрого и качественного обслуживания через онлайн-каналы, что позволит им получить доступ к медицинской помощи в любое время и в любом месте. Для администраторов в свою очередь необходимо обеспечить наличие

эффективных инструментов для управления всеми процессами внутри системы, включая контроль за медицинским обслуживанием, обработку запросов и своевременную реакцию на возникающие проблемы, что способствует повышению общей эффективности работы платформы.

### 1.1.2 Текущее состояние зарубежных исследований

Фолсом Лисал отметил, что современный образ жизни требует от человека активности. Однако, несмотря на высокоразвитыми здравоохранительными системами в некоторых странах, проблемы в их функционировании всё ещё остаются актуальными. Хотя в ряде развитых стран медицина является бесплатной и доступной для большинства граждан, инфраструктурные и организационные проблемы, такие как перегрузка больниц, нехватка медицинского персонала и большие очереди, затрудняют доступ к качественной медицинской помощи для всех социальных слоёв населения. В результате, несмотря на доступность медицины, многие группы, особенно из бедных слоёв общества, сталкиваются с трудностями при попытке воспользоваться медицинскими услугами. Это оказывает негативное влияние на их здоровье, поскольку они часто оказываются в ситуации, когда не могут своевременно получить необходимое лечение. Решение этих проблем требует изменений в организации и доступности медицинских услуг для всех слоёв населения.

Кроме того, в таких странах, как США, Великобритания и другие, национальные переписи населения не являются достаточно полными и не охватывают всех граждан. Это приводит к отсутствию актуальных данных о нуждах населения в медицинских услугах, что препятствует эффективному планированию и распределению ресурсов в здравоохранении. Без точных данных о потребностях граждан органы здравоохранения не могут правильно организовать и распределить медицинские ресурсы, что негативно сказывается на качестве и доступности медицинского обслуживания для различных слоёв населения. Также важно отметить, что не все передовые онлайн-сервисы

здравоохранения доступны для широкой аудитории, особенно для пожилых людей и тех, кто не имеет доступа к современным цифровым технологиям. Это создаёт дополнительные барьеры в использовании этих сервисов, ограничивая их потенциал и эффективность. Чтобы эти сервисы могли в полной мере раскрыть свой потенциал, необходимо преодолеть существующие цифровые барьеры и обеспечить доступность технологий для всех категорий граждан.

Для того чтобы система управления здравоохранением стала действительно технологичной и адаптированной к рыночным условиям, необходимо создать такие условия, при которых поставщики медицинских услуг будут постоянно улучшать свою компетентность и смогут своевременно и качественно обслуживать все группы населения. Это возможно только при интеграции современных технологий с системой управления персоналом, улучшении организации работы медицинских учреждений и обеспечении персонала необходимыми инструментами для эффективной работы. Важно не только внедрить высокие технологии в рабочие процессы, но и обеспечить медицинских работников необходимыми ресурсами и инструментами для выполнения их задач. Эффективная работа системы здравоохранения требует оптимизации всех её составляющих, включая процессы управления и взаимодействия между персоналом, а также обеспечение доступности и качества медицинских услуг для каждого пациента.

Важным шагом в развитии здравоохранения является также активное использование онлайн-сервисов для улучшения доступа к медицинской помощи. Создание таких онлайн-платформ для медицинского консультирования, записи на приём и мониторинга состояния здоровья может значительно повысить доступность и качество медицинских услуг, особенно в отдалённых и сельских районах. С помощью таких платформ пациенты могут легко и быстро получать консультации, записываться на приём к врачу и следить за своим здоровьем без необходимости тратить время на поездки в медицинские учреждения. Эти онлайн-сервисы позволяют врачам не только отслеживать изменения в

состоянии здоровья пациентов, но и предоставлять своевременные рекомендации, что особенно важно для людей, живущих в удалённых районах, где медицинские ресурсы ограничены. Таким образом, онлайн-платформы могут значительно улучшить качество медицинского обслуживания и расширить доступность для всех граждан, вне зависимости от их местоположения.

В связи с этим, в Китае текущая система консультирования и лечения сталкивается с проблемами, связанными с длительным временем ожидания для пользователей, а также отсутствием ряда важных функций, таких как онлайн-услуги самопомощи и возможность предварительной записи на прием к врачу. Эти проблемы значительно увеличивают время ожидания и затрудняют доступность медицинских услуг для граждан. В зарубежных странах аналогичные проблемы возникают из-за недостаточной полноты переписи населения, что приводит к нехватке актуальной информации о нуждах населения в медицинских услугах. К тому же, отсутствует необходимая инфраструктура, способная улучшить качество обслуживания и обеспечить своевременное обновление данных о медицинском персонале и его доступности. Это в свою очередь негативно сказывается на эффективности работы системы здравоохранения и доступности медицинских услуг для граждан.

На основе анализа текущих проблем в здравоохранении, как в стране, так и за рубежом, становится очевидной необходимость разработки нового подхода. Внедрение передовых технологий и использование полноценных демографических данных может стать основой для создания системы управления здоровьем, которая будет не только удобной и доступной для пациентов, но и эффективно функционирующей с точки зрения оптимизации медицинского персонала и обеспечения качественного обслуживания. В связи с этим, разработка сайта системы управления здоровьем на основе Java представляет собой актуальную задачу, сочетающую передовые технологические возможности и нужды современного общества.

## 1.2 Содержание исследования

Анализ основной бизнес-информации системы управления здоровьем на базе Java должен охватывать широкий спектр факторов, которые влияют на её успешную разработку и внедрение. Одним из ключевых аспектов является определение функциональных модулей системы, которые должны быть интегрированы для обеспечения эффективного и бесперебойного функционирования. Это включает в себя модули для записи на прием к врачу, получения консультаций, мониторинга состояния здоровья, управления медицинской информацией, а также взаимодействия с пациентами и специалистами через онлайн-платформы.

Кроме того, важно тщательно проанализировать требования пользователей, чтобы система могла удовлетворить различные потребности разных групп пользователей, включая врачей, администраторов и пациентов. Это потребует проведения глубокой работы по анализу сценариев применения системы, включая все возможные пути взаимодействия пользователей с системой, различные варианты использования и потребности в интерфейсах для каждого типа пользователей.

После тщательного анализа этих факторов можно приступать к оценке целесообразности создания системы. Важным шагом на этом пути будет анализ текущих решений, как существующих, так и развивающихся в сфере здравоохранения, а также обобщение особенностей проектирования системы, которые позволят эффективно интегрировать различные функциональные модули и обеспечить устойчивость и масштабируемость системы.

Одним из главных аспектов в проектировании системы является её функциональная составляющая, которая будет определять основные возможности системы, такие как доступность и точность медицинской информации, возможность удаленной диагностики и консультаций, а также удобство для пользователей. Эта часть разработки будет основываться на результатах тщательных исследований и анализе потребностей клиентов, с

использованием существующих научных трудов и технологических решений, чтобы выявить преимущества и недостатки текущих подходов и предложить улучшенные, более эффективные решения.

Тестирование системы на основе Java будет важнейшей частью процесса разработки. Включение различных инструментов для отладки функциональности и самотестирования системы позволит значительно улучшить качество её работы, минимизировать количество ошибок и повысить стабильность. Использование рамочных технологий, таких как JUnit для тестирования компонентов системы и различных аспектов её работы, даст возможность автоматизировать процесс проверки и ускорить внесение изменений в случае обнаружения багов. Важно проводить всестороннюю отладку, чтобы система функционировала в реальных условиях, удовлетворяя потребности пользователей и обеспечивая своевременный доступ к медицинской помощи.

Кроме того, для качественного тестирования будет важно провести всеобъемлющие испытания, охватывающие как функциональные тесты (проверка работы всех заявленных функций системы), так и тесты производительности, безопасности, а также тесты на удобство и восприятие интерфейса системой конечными пользователями. Это поможет выявить возможные слабые места системы и устранить их еще на этапе разработки, что обеспечит успешную реализацию системы и её долгосрочную эффективность в реальной эксплуатации.

### **1.3 Инструменты разработки и внедрение технологий**

Средства разработки:

IDE (Среда разработки): IdeaIU

IntelliJ IDEA (IDEA) является одной из самых популярных и мощных сред разработки для Java, предоставляя разработчикам широкий спектр возможностей для повышения производительности и эффективности работы. Это интегрированная среда разработки, которая включает умные

автодополнения, позволяющие ускорить процесс написания кода и минимизировать количество ошибок, а также статический анализ кода, который помогает выявлять потенциальные проблемы на ранних этапах разработки. Кроме того, IDEA поддерживает интеграцию с системами контроля версий, такими как Git и SVN, что упрощает работу в команде и управление версиями кода. Встроенные инструменты тестирования делают процесс написания и запуска тестов простым и быстрым, способствуя улучшению качества программного обеспечения. IDEA особенно привлекательна как для новичков, так и для опытных разработчиков, поскольку она сочетает в себе интуитивно понятный интерфейс с мощными функциональными возможностями, такими как поддержка множества современных технологий и фреймворков, включая Spring, Hibernate и другие. Эти особенности делают IDEA идеальной средой разработки для широкого спектра задач, от создания небольших проектов до масштабных корпоративных приложений, обеспечивая разработчикам гибкость и удобство на всех этапах разработки. Система сборки: Maven 3.5.2

Maven – это система управления проектами и сборки, которая автоматизирует многие процессы, такие как зависимость, сборка, тестирование и развертывание. В частности, версия Maven 3.5.2 предоставляет улучшенную производительность, а также более стабильную работу с большими проектами и зависимостями. Использование Maven позволяет упростить процесс управления зависимостями и улучшить совместимость между различными библиотеками и фреймворками.

Язык программирования: JDK 1.8

Java Development Kit (JDK) 1.8 – это ключевая версия Java, в которой были внедрены многие важные новшества, такие как лямбда-выражения, Stream API и улучшения в производительности. JDK 1.8 также продолжает поддерживать платформу Java SE, что делает его идеальным выбором для разработки корпоративных приложений, особенно тех, которые требуют высокую производительность и надежность.

## Система управления базами данных: MySQL 5.7

MySQL является одной из самых популярных реляционных баз данных с открытым исходным кодом. Это высокопроизводительное решение для хранения и обработки данных, которое используется во множестве крупных веб-приложений. MySQL 5.7 предоставляет улучшенные возможности для работы с данными, включая улучшенную поддержку JSON, расширенные индексы и оптимизацию запросов. Он идеально подходит для проектов, которые требуют быстрого доступа к данным и поддерживают большой объем пользователей.

Техническое введение:

### Язык программирования Java

Java в настоящее время является одним из самых распространенных и широко используемых языков программирования в мире. С момента своего создания Java зарекомендовала себя как язык, который легко освоить и использовать. Особенно он популярен среди начинающих разработчиков, так как для того, чтобы начать работу с Java, достаточно освоить базовые синтаксические конструкции и принципы объектно-ориентированного программирования (ООП). Он также поддерживает обширную экосистему библиотек и фреймворков, что значительно ускоряет процесс разработки.

Java выделяется своей платформенной независимостью, что означает, что написанное на Java приложение может работать на различных операционных системах без необходимости модификации кода. Это стало возможным благодаря Java Virtual Machine (JVM), которая является основой Java-программ. Платформа Java предоставляет разработчикам богатые возможности для создания сложных и масштабируемых приложений, поддерживающих многозадачность, работу с большими объемами данных и сетевые коммуникации.

Одним из важнейших преимуществ Java является наличие множества открытых библиотек и пакетов (JAR), которые можно использовать для

решения разнообразных задач, таких как работа с базами данных, сетевыми соединениями, графическими интерфейсами и т.д. Эти библиотеки значительно сокращают время разработки, позволяя фокусироваться на реализации логики приложения.

### Spring Framework

Spring Framework – это один из самых популярных фреймворков для разработки на Java. Он представляет собой универсальную платформу, которая предоставляет широкие возможности для построения современных корпоративных приложений. Основная цель Spring – упростить разработку, предложив инструменты для решения типичных задач, таких как инъекция зависимостей (IOC), работа с базами данных, управление транзакциями и создание RESTful сервисов.

IOC (Inversion of Control) – это ключевая концепция Spring, которая помогает решить проблему жесткой связанности компонентов приложения. Вместо того чтобы напрямую создавать экземпляры объектов, Spring позволяет конфигурировать зависимости через XML-конфигурацию или аннотации, что улучшает тестируемость и расширяемость системы.

Кроме того, Spring предоставляет множество расширений для различных технологий, таких как Spring Boot (для упрощенной настройки и развертывания приложений), Spring Data (для работы с базами данных), Spring Security (для обеспечения безопасности), что делает его идеальным выбором для создания современных многослойных приложений.

### MySQL

MySQL – это мощная система управления базами данных (СУБД), которая использует реляционную модель для хранения данных. Она известна своей высокой производительностью, простотой использования и широкими возможностями для обработки больших объемов информации. MySQL является наиболее распространенной СУБД в веб-разработке благодаря своей способности эффективно работать с многими пользователями одновременно и

обеспечивать быстрые запросы к данным.

В версии MySQL 5.7 были добавлены новые возможности для работы с данными, такие как улучшенная поддержка JSON и оптимизация производительности, что делает ее идеальным инструментом для обработки сложных и многозадачных приложений. Эта версия также поддерживает различные способы индексации данных, что позволяет ускорить выполнение запросов и уменьшить нагрузку на систему.

#### **1.4 Технико-экономический анализ функций системы**

В условиях современной социальной ситуации, когда наблюдается рост продолжительности жизни и старение населения, необходимо учитывать, что с возрастом сопротивляемость организма человека снижается. Это приводит к увеличению частоты заболеваний и необходимости в более быстром и доступном получении медицинской помощи. Офлайн-консультации в больницах, несмотря на их важность, становятся все менее удобными, так как они требуют значительного времени на ожидание и посещение медицинского учреждения, что в свою очередь затрудняет своевременную диагностику и лечение. В этих условиях необходимость внедрения эффективных и доступных онлайн-систем здравоохранения становится особенно актуальной.

Технико-экономический анализ системы управления здоровьем проводится для оценки целесообразности её разработки, внедрения и эксплуатации. Он служит основой для определения возможных проблем и рисков, а также позволяет на раннем этапе разработки системы выявить её сильные и слабые стороны. В этом анализе особое внимание уделяется технической, экономической, эксплуатационной и юридической целесообразности системы.

##### **1.4.1 Техническая реализуемость**

Основной технологией, используемой в данной системе, является Web-технология. В последние годы веб-разработка стала одной из самых популярных и востребованных сфер в IT-индустрии. Существуют различные

фреймворки и инструменты, которые позволяют быстро разрабатывать и внедрять решения. Java и её популярные фреймворки, такие как Spring, широко используются для создания масштабируемых, высокоэффективных и безопасных приложений, что идеально подходит для задач, связанных с управлением здоровьем.

Разработка системы была выполнена мной самостоятельно, начиная с этапа проектирования и заканчивая развертыванием и тестированием. Благодаря использованию современных технологий и инструментов, таких как IntelliJ IDEA, Maven и MySQL, процесс разработки был упрощён. В будущем система будет масштабироваться для удовлетворения потребностей увеличивающегося числа пользователей. Это обеспечит не только успешную работу системы, но и положительный экономический эффект, что делает её технически целесообразной. Также, благодаря применению облачных технологий и возможности работы с большими объемами данных, система будет способна эффективно функционировать с постоянно растущим количеством пользователей.

#### 1.4.2 Экономическая целесообразность

С точки зрения экономической целесообразности, проект является весьма выгодным. Все этапы разработки, включая проектирование, тестирование и развертывание, проводятся мной самостоятельно без привлечения дорогостоящих внешних подрядчиков, что существенно снижает затраты на проект. Весь используемый софт имеет открытый исходный код и доступен бесплатно, что еще больше сокращает первоначальные затраты.

Используемые инструменты, такие как MySQL, Spring и Maven, также бесплатны и широко распространены, что способствует снижению стоимости разработки. К тому же, активность сообщества разработчиков этих технологий очень высока, что позволяет получить поддержку и решать возникающие проблемы без необходимости платить за профессиональные консультации.

Экономическая эффективность системы управления здоровьем будет

расти по мере её внедрения. С развитием системы и увеличением числа пользователей, её рынок будет расширяться, что обеспечит неограниченные экономические выгоды. Система сможет приносить доход за счет внедрения премиум-услуг и дополнительных функций, таких как персонализированные медицинские консультации, мониторинг здоровья в реальном времени, а также платёжные сервисы для онлайн-оплаты медицинских услуг.

Таким образом, с экономической точки зрения, разработка системы управления здоровьем вполне обоснована и принесет значительную выгоду как в краткосрочной, так и в долгосрочной перспективе.

#### 1.4.3 Операционная целесообразность

Для пользователей система предоставляет множество удобных функций, таких как консультации с экспертами, заказы медицинских товаров, комментарии к консультациям и другие возможности, что делает её удобной в использовании. Интерфейс системы спроектирован так, чтобы пользователи могли легко начать работать с приложением, не тратя много времени на освоение функционала.

Для администраторов система предоставляет удобные инструменты для управления пользователями, публикации новостей, а также мониторинга и проверки экспертных знаний. Это позволяет эффективно организовывать работу системы, обеспечивать качество предоставляемых услуг и поддерживать высокий уровень пользовательского опыта. Разработанная система также обеспечит простоту обслуживания, что значительно уменьшит нагрузку на персонал и обеспечит стабильность работы всей платформы.

В целом, система спроектирована таким образом, чтобы пользователи могли быстро и без проблем начать взаимодействовать с ней, а администраторы-эффективно управлять её функционированием, что подтверждает высокую операционную целесообразность.

#### 1.4.4 Юридическая целесообразность

Разработчикам необходимо удостовериться, что содержание системы не

нарушает права интеллектуальной собственности других лиц, включая авторские права, торговые марки, патенты и другие объекты интеллектуальной собственности. Также важно обеспечить защиту собственных прав интеллектуальной собственности разработчиков. Кроме того, система должна быть в полном соответствии с законодательством о защите личных данных пользователей. Для этого необходимо внедрить соответствующие технические и организационные меры, такие как шифрование данных, контроль доступа и регулярные проверки безопасности, чтобы гарантировать конфиденциальность, целостность и доступность персональной информации пользователей. В случае утечки или компрометации данных разработчики обязаны немедленно уведомить пользователей и принять все необходимые меры для устранения последствий.

### **1.5 Анализ системных требований**

#### Модуль администратора

–администратор входит в учетную запись и может просматривать информацию об администраторе в личном центре;

–администратор может просматривать информацию о пользователях и экспертах в интерфейсе управления пользователями;

–администратор может публиковать информацию об объявлениях через управление объявлениями;

–статьи знаний, опубликованные пользователями-экспертами, должны быть проверены администратором через управление статьями, и проверенные статьи публикуются на стене знаний, которая будет доступна для просмотра обычным пользователям;

–администратор просматривает информацию о знаниях, опубликованную экспертами;

–администратор может добавлять информацию о продукте, изменять права доступа к продукту, просматривать информацию о продукте и помещать продукт на полку. Администратор может просматривать информацию о заказе

лекарств и изменять статус заказа;

–после завершения заказа администратор может добавить дополнительные заказы и комментарии к товару.

Модуль пользователя

–регистрация пользователя, проверка личной информации и изменение пароля;

–на главной странице можно просмотреть информацию об объявлениях, опубликованных администратором;

–советы по здоровью, публикуемые экспертами, могут быть просмотрены пользователями после проверки и одобрения администратором;

–пользователи могут общаться с экспертами, чтобы получить ответы на свои вопросы;

–пользователи могут просматривать информацию о продуктах, опубликованную администратором, и в соответствии с информацией о продукте выбирать необходимое количество продуктов для заказа или добавлять их в корзину;

–пользователи могут просматривать выполненный заказ, вы можете комментировать выполненный заказ.

Модуль Эксперт

–пользователи-эксперты могут зарегистрировать личный аккаунт и просматривать личную информацию;

–пользователи-эксперты могут публиковать собственные статьи и работы в управлении стеной знаний, а также показывать их обычным пользователям после прохождения проверки администратором;

–эксперты могут просматривать вопросы пользователей и давать на них ответы через пользовательское общение;

–обычные пользователи могут комментировать статьи, опубликованные экспертами, через область комментариев, а эксперты могут добавлять комментарии после комментариев.

## 2 РАЗРАБОТКА И ВНЕДРЕНИЕ СИСТЕМ УПРАВЛЕНИЯ ЗДРАВООХРАНЕНИЕМ

### 2.1 Дизайн функций системы

Система управления здоровьем на базе Java состоит из трех основных ролей, каждая из которых имеет свой собственный функциональный модуль: администратор, обычный пользователь и эксперт. Модуль администратора включает в себя управление пользователями (регистрация, модификация данных, управление ролями), управление контентом (товары, статьи, объявления) и аналитику (управление заказами, анализ комментариев и отзывов). Администратор имеет возможность изменять настройки системы, управлять правами доступа и контролировать активность пользователей.

Обычные пользователи взаимодействуют с платформой через персонализированный профиль. Они могут просматривать товары, добавлять их в корзину и оформлять заказы. Кроме того, пользователи имеют доступ к «Стене знаний», где могут изучать материалы, оставлять комментарии и участвовать в обсуждениях. Этот модуль позволяет пользователям обмениваться мнениями и получать полезную информацию.

Эксперты создают образовательный контент, включая статьи и руководства, а также предоставляют профессиональные консультации пользователям. Они отвечают на комментарии и вопросы, помогают пользователям в процессе выбора товаров и услуг, а также взаимодействуют с данными для интеграции и обновления информации.

Архитектура системы, представленная на Рисунке 1, организована таким образом, чтобы обеспечить четкое разделение прав доступа и функционала. Это позволяет обеспечить гибкое взаимодействие между модулями и централизованное управление основными компонентами системы: пользователями, товарами, заказами и образовательными ресурсами.



Рисунок 1 – Диаграмма функциональных модулей системы управления здоровьем на базе Java

## 2.2 Тестирование системного каркаса

Архитектура системы представляет собой многоуровневую структуру, которая условно делится на четыре основные слоя: клиентский слой, прикладной слой, слой кэширования и слой данных. Каждый из этих слоев играет важную роль в обеспечении эффективной работы системы, и их взаимодействие позволяет создать гибкую и масштабируемую архитектуру.

Клиентский слой – это интерфейс, через который пользователи взаимодействуют с системой. Он включает мобильные приложения, веб-браузеры и другие устройства, обеспечивающие доступ к системе. Основная задача этого слоя заключается в предоставлении удобного и интуитивно понятного способа взаимодействия с пользователем. Клиентский слой также выполняет роль "моста" между системой и пользователем, обеспечивая передачу данных и взаимодействие с прикладной логикой.

Прикладной слой выполняет роль центрального компонента системы, обеспечивая обработку всех бизнес-процессов. Он включает три ключевых подсистемы. Слой управления отвечает за организацию процессов и взаимодействие между различными частями системы. Слой бизнес-логики непосредственно обрабатывает данные, выполняя основные операции, необходимые для выполнения функционала системы. Базовый сервисный слой предоставляет вспомогательные функции, такие как обработка запросов и управление сервисами.

Слой кэширования представляет собой промежуточный уровень для хранения данных, которые могут быть временно сохранены для ускорения доступа и уменьшения нагрузки на базу данных. Он играет важную роль в обеспечении производительности системы, поскольку позволяет значительно сократить время отклика при повторных запросах к данным. Этот слой способствует более эффективному использованию ресурсов и повышению общей скорости работы системы.

Слой данных является основой всей системы, поскольку здесь хранятся все данные, с которыми работает система. Он включает в себя базы данных, системы хранения и все компоненты, связанные с управлением данными. Этот слой обеспечивает надежность, целостность и доступность данных для всех других слоев системы.

Такой многоуровневый дизайн архитектуры позволяет достичь множества важных целей, таких как децентрализованное внимание, когда каждый слой фокусируется на своей специфической задаче, что улучшает управление и организацию работы системы. Свободная связь между слоями позволяет независимым частям системы взаимодействовать без жесткой привязки. Логическое повторное использование повышает эффективность разработки и снижает избыточность кода. Четкое разделение модулей обеспечивает гибкость и улучшение поддерживаемости системы. Легкость в координации между модулями способствует более быстрому развитию и уменьшению сложности взаимодействия между командами разработки.

Такой подход также создает благоприятные условия для разработки, где каждый разработчик может работать над отдельными частями системы, обеспечивая высокую степень кооперации и эффективное разделение труда. В будущем такая архитектура будет способствовать поддержке и масштабированию системы с минимальными усилиями.

Диаграмма использования данной системы, которая объединяет различные модули, представлена на рисунке 2. В системе выделены три

основные роли: администратор, пользователь и эксперт, каждая из которых выполняет специфические действия. Администратор управляет пользователями, публикует объявления, проверяет знания, а также занимается управлением заказами и товарами. Пользователь может зарегистрироваться, войти в систему, просматривать объявления и товары, управлять заказами и оставлять комментарии. Эксперт отвечает на запросы пользователей, публикует знания и управляет комментариями.

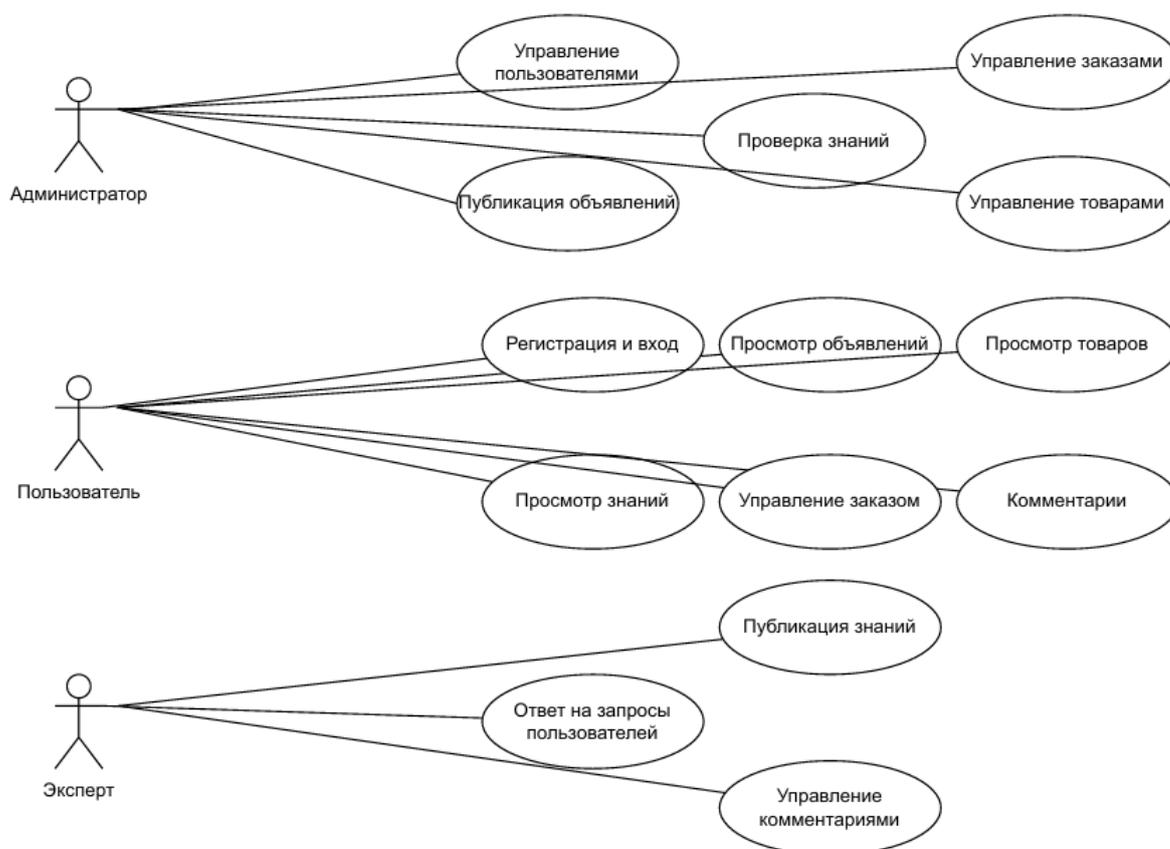


Рисунок – 2 Диаграмма вариантов использования

## 2.3 Проектирование базы данных

### 2.3.1 Концептуальный дизайн базы данных

Эта система в первую очередь включает три ключевых роли: администратора, пользователя и эксперта, каждая из которых выполняет свои специфические функции. В контексте этих ролей создается основная структура базы данных, включающая таблицы сущностей и их связи. Каждая таблица базы данных имеет уникальный идентификатор, который используется в

качестве первичного ключа с автоинкрементом для обеспечения уникальности данных. Для оптимизации работы системы применяется разделение труда с использованием различных типов параметров, что позволяет эффективно управлять данными. В таблице ассоциаций идентификаторы каждой таблицы играют ключевую роль в обеспечении уникальности и правильного доступа к базе данных, что гарантирует правильную организацию и использование данных в системе.

### 2.3.2 Логическая структура базы данных

Таблица пользователей, как показано в таблице 1, описывает ключевые данные, связанные с каждым пользователем. Одним из основных полей является идентификатор пользователя (`id`), который представляет собой уникальный числовой код для идентификации пользователя в системе. Это поле является первичным ключом, тип данных – `int`, длина составляет 20 символов, что обеспечивает уникальность записи.

Имя пользователя (`name`) – это строковое поле, в котором хранится имя пользователя. Оно имеет тип данных `varchar`, длина 20 символов, и используется для отображения имени пользователя в интерфейсе.

Пароль (`password`) – зашифрованное строковое поле, предназначенное для хранения пароля пользователя. Тип данных – `varchar`, длина 60 символов, что позволяет хранить достаточно длинные и сложные пароли для повышения безопасности.

Статус включения (`valid`) – целочисленное поле, которое показывает активность пользователя в системе. Если значение этого поля равно 1, это означает, что пользователь активен, а если 0 – пользователь заблокирован. Тип данных – `int`, длина поля составляет 6 символов.

Номер мобильного телефона (`phone`) – строка, предназначенная для хранения номера мобильного телефона пользователя. Оно имеет тип данных `varchar`, длина 15 символов, что позволяет учитывать международные форматы номеров.

Номер электронной почты (email) – строка, в которой хранится адрес электронной почты пользователя. Тип данных – varchar, длина 25 символов. Это поле используется для отправки уведомлений пользователю и для связи с ним.

Эти поля составляют основную структуру таблицы пользователей, обеспечивая хранение и управление данными о пользователях с высокой степенью безопасности и удобства в системе.

Таблица 1 – Таблица пользователей

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	Пользователь id	int	20	первичный ключ
2	name	имя пользователя	varchar	20	
3	password	криптовалюта	varchar	60	
4	valid	Состояние включения	int	6	
5	phone	номер мобильного телефона	varchar	15	
6	email	номер почтового ящика	varchar	25	

Таблица стены знаний, как показано в таблице 2, описывает статьи, опубликованные экспертами системы. Основными полями этой таблицы являются:

ID стены знаний (id) – уникальный идентификатор записи, являющийся первичным ключом. Это поле имеет тип данных int длиной 20 символов, что обеспечивает уникальность каждой статьи в системе.

Содержание (content) – строковое поле, в котором хранится сам текст статьи. Это поле имеет тип данных varchar с длиной до 255 символов, что

позволяет сохранять достаточно большие объемы текстовой информации.

Описание (description) – дополнительное описание статьи, которое может содержать аннотацию или краткое резюме. Также имеет тип данных varchar, длина 255 символов.

ID пользователя (userId) – поле, которое хранит идентификатор пользователя, связанного с данной статьей. Это поле используется для указания эксперта, который опубликовал статью. Тип данных – int, длина 20 символов.

Положение дел (valid) – целочисленное поле, которое указывает на актуальность или состояние статьи. Это поле используется для отслеживания статуса публикации (например, активная или удаленная статья). Тип данных – tinyint, длина 10 символов.

Эта таблица играет важную роль в хранении и управлении статьями, опубликованными экспертами, и обеспечивает возможность их эффективного поиска и обновления в системе.

Таблица 2 – Оглавление стены знаний

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	стена знаний id	int	20	первичный ключ
2	content	элемент	varchar	255	
3	description	описания	varchar	255	
4	userId	Пользователь id	int	20	
5	valid	положение дел	tinyint	10	

Таблица комментариев, как показано в таблице 3, хранит информацию о комментариях, связанных с заказами, что позволяет улучшить взаимодействие между клиентами и администраторами. Основными полями являются ID

комментария, который является первичным ключом, и ID заказа, связывающий комментарий с конкретным заказом. Статус (valid) указывает на актуальность комментария, помогая администратору быстро оценить его важность и необходимость дальнейшей обработки. ID клиента (user\_id) и ID администратора (technical\_id) связывают комментарий с конкретным пользователем и администратором, что позволяет легко отслеживать и управлять всеми комментариями в системе. В поле Отзыв клиента (user\_content) хранится текст отзыва клиента, а Ответ администратора (technical\_content) – текст ответа администратора на этот отзыв, что способствует более оперативному решению вопросов и улучшению качества обслуживания. Эта таблица помогает эффективно управлять комментариями и их статусом, обеспечивая прозрачность и качественную обработку обратной связи.

Таблица 3 – Форма комментариев к заказу

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	Комментарии id	int	20	первичный ключ
2	goods_order_id	Приказы id	int	20	
3	valid	Статус	int	10	
4	user_id	Клиент id	int	20	
5	technical_id	Администратор id	int	20	
6	user_content	Отзывы	varchar	255	
7	technical_content	Ответ администратора	varchar	255	

Таблица консультаций экспертов, как показано в таблице 4, предназначена для хранения информации об обмене запросами и ответами между пользователями и экспертами, обеспечивая организованный и

систематизированный способ управления консультациями. Основными полями этой таблицы являются ID запроса (id), являющийся уникальным идентификатором для каждой консультации, который позволяет легко отслеживать и ссылаться на конкретные запросы, и ID пользователя (userId), который связывает консультацию с конкретным пользователем, обеспечивая точную идентификацию инициатора консультации. Также в таблице присутствует поле ID эксперта (teacherId), которое указывает на эксперта, предоставившего консультацию, что позволяет связать каждый запрос с соответствующим специалистом и обеспечивать правильную идентификацию экспертов, участвующих в консультациях. Эти данные помогают эффективно отслеживать и управлять всей информацией о взаимодействиях между пользователями и экспертами, а также обеспечивают возможность анализа и улучшения качества консультационных услуг. Кроме того, в таблице содержится Содержание консультации (userContent) – текст, предоставленный пользователем, в котором изложен его вопрос или запрос. Ответ эксперта (teacherContent) – строковое поле, которое хранит ответ эксперта на запрос пользователя. Эта таблица помогает эффективно отслеживать и управлять консультациями между пользователями и экспертами, обеспечивая удобное хранение и доступ к информации.

Таблица 4 – Форма для консультаций с экспертами

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	Запрос id	int	20	первичный ключ
2	userId	Пользователь id	int	20	
3	teacherId	Эксперт id	int	20	
4	userContent	Содержание консультации	varchar	255	
5	teacherContent	Отзывы	varchar	255	

Таблица товаров, как показано в таблице 5, хранит информацию о товарах, представленных в системе. Эта таблица играет ключевую роль в управлении данными о товарах, обеспечивая их идентификацию и описание. Основные поля включают ID товара (id), который является уникальным идентификатором для каждой позиции, что позволяет однозначно различать товары в системе и облегчает их поиск. Также важным полем является Название товара (order\_name), которое используется для обозначения наименования товара, позволяя пользователям и системам легко идентифицировать его по имени. В таблице могут быть и другие поля, такие как описание товара, категория, цена, количество на складе и другие, которые дополнительно помогают в управлении товарными запасами и улучшении пользовательского опыта. Цена (price) – это стоимость товара в установленной валюте. Поле использует тип данных decimal(10,2), чтобы точно хранить цены с двумя знаками после запятой.

ID пользователя (user\_id) – поле, которое связывает товар с конкретным пользователем или владельцем. Это позволяет идентифицировать, кто создал или разместил товар в системе.

Изображение (img) – строковое поле для хранения адреса изображения товара. Это позволяет отображать графическое представление товара в интерфейсе.

Статус (valid) – поле, которое указывает на текущий статус товара, например, доступность или наличие на складе. Тип данных — int, где возможные значения могут быть использованы для отслеживания состояния товара.

Эта таблица представляет собой основу для управления товарами, их состоянием и связями с пользователями, обеспечивая эффективное хранение, доступ к информации и поддержку всех операций с товаром в системе, что способствует улучшению управления запасами и повышению удобства для пользователей.

Таблица 5 – Информация о товарах

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	Товар id	int	20	первичный ключ
2	oder_name	Название продукта	varchar	255	
3	price	Цена единицы продукции	decimal(10,2)	20	
4	user_id	Создано id	big	20	
5	img	Адрес изображения	varchar	255	
6	valid	Статус	Int	10	

Таблица заказов, как показано в таблице 6, хранит информацию о заказах, являясь ключевым элементом для управления процессом заказов на платформе. Она включает в себя ID заказа (id), который служит уникальным идентификатором для каждого заказа, что позволяет легко отслеживать и управлять заказами в системе. ID продавца (business\_id) связывает заказ с конкретным продавцом, что помогает определить, кто является ответственным за выполнение данного заказа. Кроме того, в таблице указан ID товара (seller\_id), который связывает заказ с определенным товаром, позволяя точно отслеживать, какие товары были заказаны.

Статус (status) отображает текущее состояние заказа, например, выполнен, в процессе или отменен, что позволяет администраторам и пользователям оперативно следить за движением заказов и предпринимать необходимые действия в случае необходимости. ID клиента (user\_id) связывает заказ с клиентом, который его разместил, предоставляя полную информацию о заказчике и его запросах. Эта таблица позволяет эффективно управлять

заказами, обеспечивая прозрачность всех процессов: от размещения заказа до его выполнения. Она также помогает отслеживать состояние каждого заказа, связывать его с соответствующими пользователями и продавцами, что способствует улучшению качества обслуживания и повышению общей эффективности работы платформы.

Таблица 6 – Форма заказа

серийный номер	Английское название	Русское имя	тип данных	длина	ключевые слова
1	id	Заказ id	int	20	первичный ключ
2	business_id	Торговец id	int	20	
3	seller_id	Товар id	int	20	
4	status	Статус	int	10	
5	user_id	Клиенты id	int	20	

## 2.4 Каталог структур проекта

Данный дизайн относится к веб-части проекта, который работает в серверном режиме через браузер. Для доступа к домашней странице достаточно ввести URL в браузере. Основным инструментом разработки для этой системы является IntelliJ IDEA, которая разделена на два блока. Первый блок — это проект на языке Java, структура которого представлена в основном в виде файлов .java, содержащих код для реализации серверной части и взаимодействия с браузером с помощью гипертекстового языка. Эти файлы отвечают за логику backend-части системы.

Второй блок проекта связан с JVM и содержит скомпилированные файлы .class, которые находятся в начале папки. Эти файлы декомпилируются из .java файлов и используются для запуска серверной службы и ее работы в

реальном времени. Структура проекта, как показано на рисунке 3, разделена по папкам, что упрощает организацию кода и его поддержание.

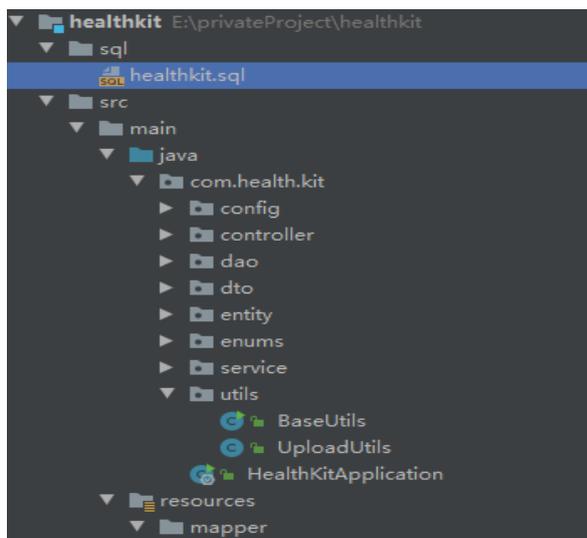


Рисунок 3 – Код на стороне сервера

Фреймворк Thymeleaf представляет собой мощный и современный инструмент для создания шаблонов веб-страниц в Java-приложениях. Он полностью заменяет устаревший подход использования JSP, обеспечивая более гибкую и простую разработку интерфейсов. Thymeleaf позволяет разработчикам эффективно взаимодействовать с серверной частью, предоставляя простой синтаксис для вставки данных и динамических элементов в HTML-страницы. В отличие от JSP, который ориентирован на серверную сторону и требует дополнительных манипуляций для работы с HTML, Thymeleaf позволяет легче обрабатывать шаблоны на стороне клиента и интегрировать их с сервером.

Как показано на рисунке 4, структура проекта использует папку templates, где хранятся все HTML-шаблоны, предназначенные для отображения данных. В этой папке можно найти шаблоны для различных страниц, таких как страницы оформления заказа, просмотра товаров, а также страницы личного кабинета пользователя. Эти шаблоны включают элементы, такие как формы для ввода данных, таблицы для отображения информации, а также механизмы для динамического обновления контента, обеспечивая удобство для пользователей и простоту для разработчиков.

Thymeleaf имеет множество преимуществ, среди которых стоит отметить его способность работать как в серверном режиме, так и в режиме работы с клиентскими приложениями. Он также поддерживает работу с различными расширениями, которые могут быть использованы для улучшения функциональности интерфейса. Например, возможность интеграции с другими технологиями, такими как Bootstrap, позволяет значительно улучшить внешний вид страниц без необходимости вручную прописывать весь HTML.

Кроме того, Thymeleaf предлагает высокую степень гибкости и расширяемости. Разработчики могут легко подключать и настраивать различные компоненты шаблонов в зависимости от специфики проекта. Благодаря этому, Thymeleaf идеально подходит для крупных проектов, где требуется интеграция с множеством данных и сложных пользовательских интерфейсов.

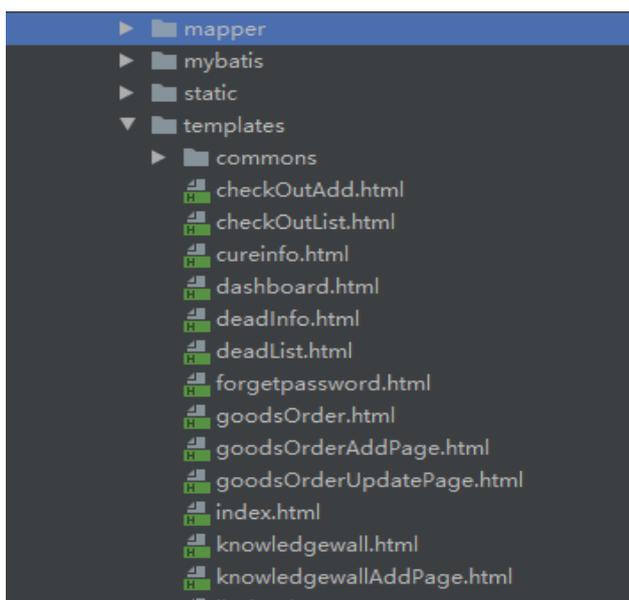


Рисунок 4 – Базовый код интерфейса

## 2.5 Функциональный дизайн и реализация

### 2.5.1 Разработка и реализация интерфейса входа и регистрации

Схема входа и регистрации, представленная на рисунке 5, описывает последовательность действий пользователя при попытке войти в систему или зарегистрироваться. Пользователь открывает страницу входа, вводит учетные

данные, которые передаются на сервер через HTTP-протокол для проверки. Если учетная запись существует и данные верны, пользователю предоставляется доступ. В противном случае доступ блокируется, и предлагается пройти процесс регистрации. При регистрации данные пользователя (имя, email, пароль) добавляются в базу данных с помощью метода add(), после чего пользователь перенаправляется на страницу входа для авторизации.

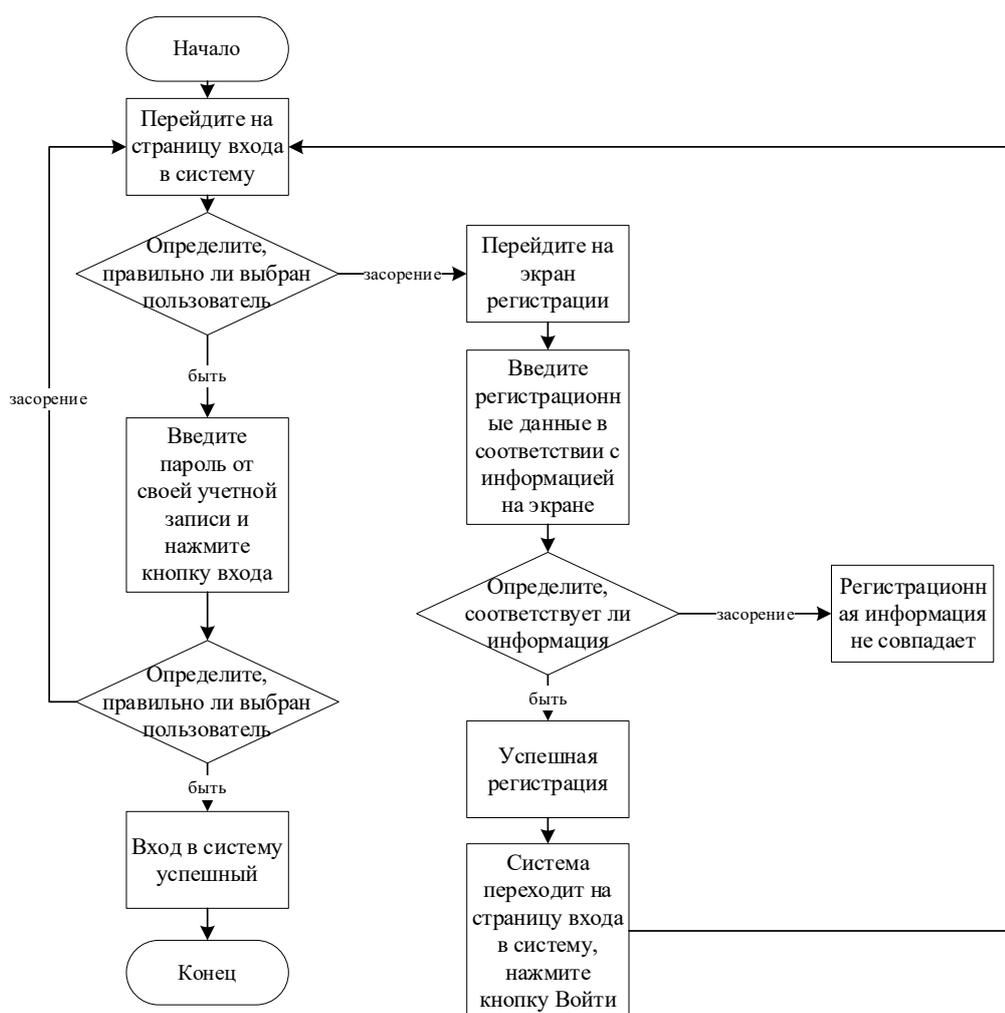


Рисунок 5 – Блок-схема регистрации входа

На рисунке 6 представлена диаграмма последовательности, которая описывает процесс аутентификации администратора и управления товарами. Процесс начинается с того, что администратор вводит свои учетные данные – логин и пароль. Система затем проверяет введенную информацию в базе данных. Если проверка проходит успешно, администратор получает доступ к

системе. В противном случае система отклоняет запрос на вход.

После успешной аутентификации администратор может приступить к управлению товарами. Для добавления информации о товаре администратор вводит необходимые данные, и система проверяет их корректность. Если информация верна, товар добавляется в систему, а если имеются ошибки, система сообщает об этом.

Диаграмма на рисунке 6 иллюстрирует последовательность действий, демонстрируя взаимодействие между администратором, системой и базой данных в процессе аутентификации и управления товарными записями.

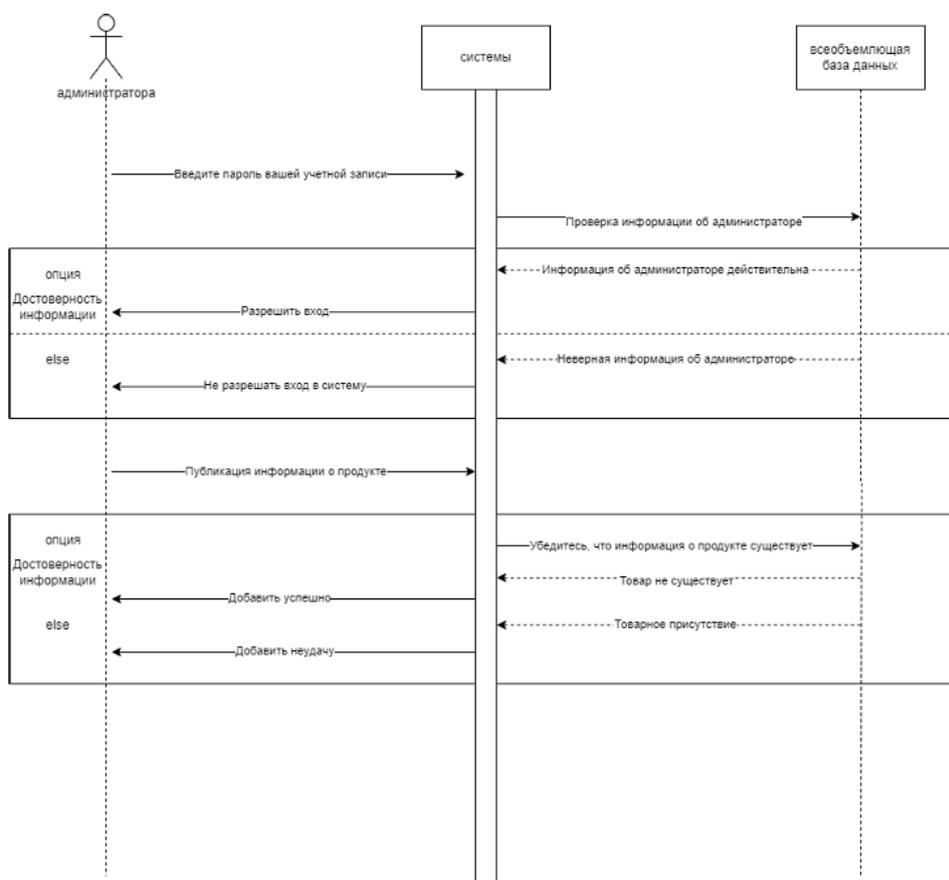


Рисунок 6 – Диаграмма последовательности

На рисунке 7 представлена диаграмма деятельности, которая описывает процесс проверки идентификационной информации пользователя. Пользователь начинает процесс, отправляя или изменяя свою идентификационную информацию. Эта информация затем поступает в процесс проверки, где ее анализирует администратор. Если проверка проходит успешно,

идентификационная информация помечается как проверенная, и она передается в систему. Если проверка не проходит, система уведомляет пользователя о необходимости повторной подачи или изменения своей идентификационной информации. Этот процесс разработан таким образом, чтобы обеспечить четкость и последовательность на всех этапах: от предоставления информации пользователем до ее проверки и обработки, гарантируя высокую степень точности и целостности данных. Таким образом, диаграмма отражает важный механизм обеспечения безопасности и достоверности данных, включая повторное представление информации, если это необходимо.

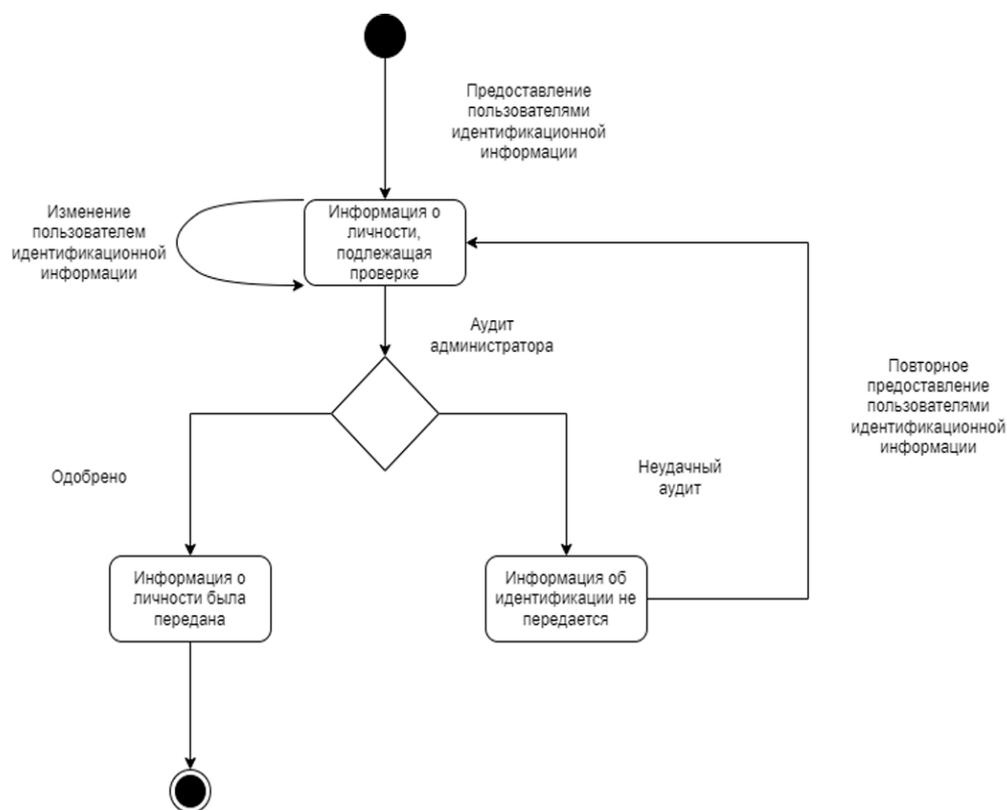


Рисунок 7 – Диаграммн деятельности

### 2.5.2 Разработка и внедрение управления корзиной покупок

После того как пользователи просматривают информацию о продукте, они могут добавить его в корзину для дальнейшего оформления заказа. При этом корзина позволяет не только управлять товарами, удаляя или добавляя их, но и запрашивать дополнительную информацию о каждом продукте. Также доступна

возможность отображать содержимое корзины, чтобы пользователи могли видеть все выбранные товары перед подтверждением заказа. На рисунке 8 представлена схема интерфейса корзины, где наглядно показано, как пользователь может взаимодействовать с корзиной, управлять товаром и продолжать процесс оформления заказа. Это упрощает процесс покупок, обеспечивая пользователю полный контроль над выбранными товарами.

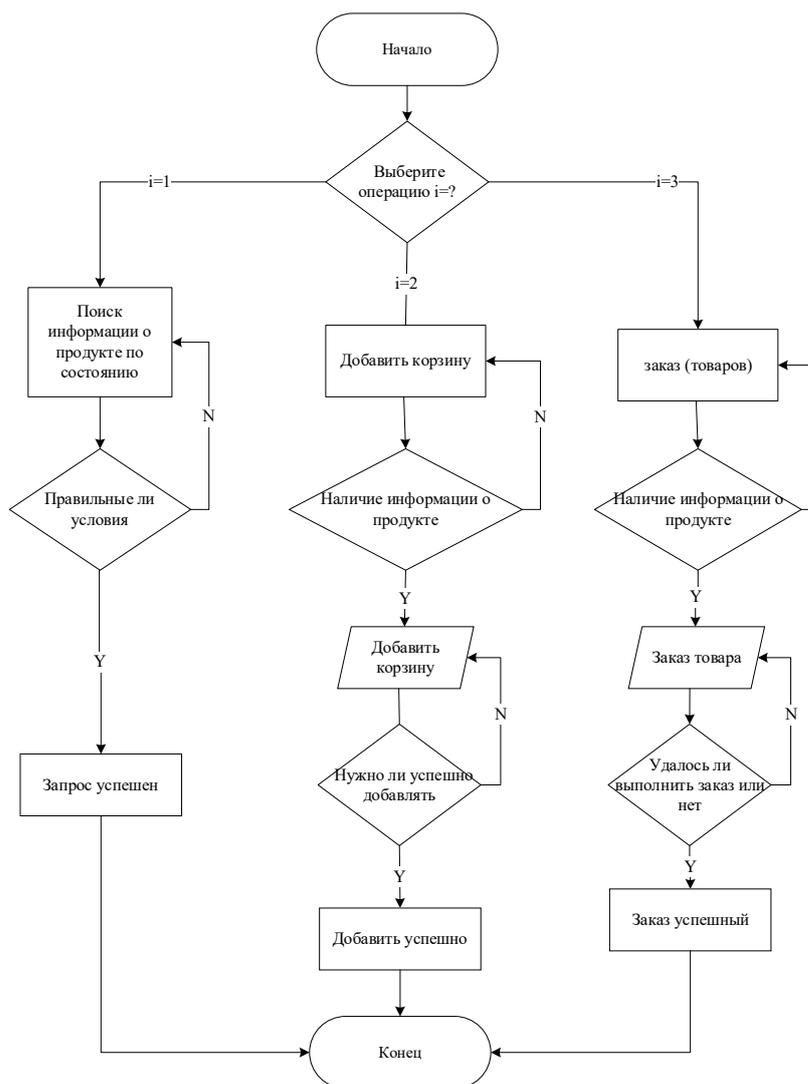


Рисунок 8 – Блок-схема управления корзиной покупок

### 2.5.3 Разработка и внедрение системы управления товарами

Администратор добавляет информацию о товарах через систему управления, после чего проводится проверка корректности и полноты данных. После успешной проверки товар получает статус "опубликован" и становится

доступным для просмотра пользователями на витрине магазина. Пользователи могут просматривать подробную информацию о товаре, такую как название, описание, цену и наличие на складе, а также использовать функции фильтрации и сортировки для упрощения поиска. После выбора товара пользователь может добавить его в корзину и оформить заказ. На этом этапе контроллер обрабатывает запросы от пользователя и передает их на уровень бизнес-логики, где реализуются функции, связанные с отображением товаров, пагинацией и фильтрацией данных. Администратор, в свою очередь, может отслеживать движение товаров через систему, включая их статус, продажи и изменения в запасах. Схема движения товаров показана на рисунке 9.

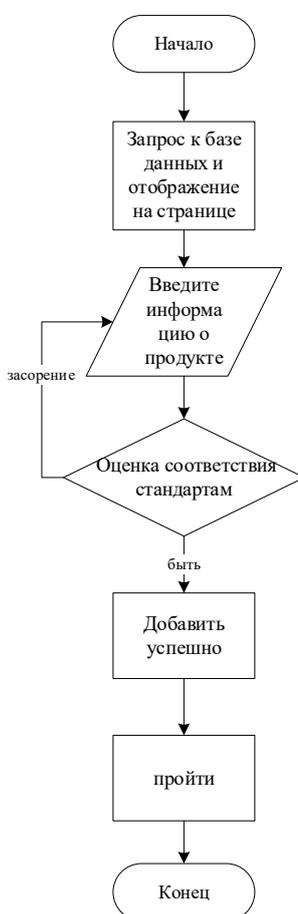


Рисунок 9 – Блок-схема списка продуктов

На следующем рисунке показан поток действий по оплате, который начинается с того, что пользователь подтверждает выбор товара. После этого система переходит на этап ожидания оплаты, где она проверяет информацию о транзакции. Если платеж прошел успешно, система завершает процесс, и

пользователю выводится сообщение «Успех покупки». Однако, если платеж не прошел, система сообщает об ошибке с сообщением «Ошибка при покупке» и предлагает пользователю ознакомиться с возможными причинами неудачи.

На данном этапе пользователь может принять решение о внесении изменений в данные товара или повторной попытке оплаты. В случае успешной попытки оплаты транзакция завершается, и пользователю подтверждается успешное завершение покупки. В случае неудачи система предоставляет дополнительные инструкции для исправления ошибки. Это отражает логику и результаты обработки платежной ссылки, как показано на рисунке 10, где каждый этап процесса строго организован и четко отображает последовательность действий и возможные результаты в случае разных вариантов развития событий.

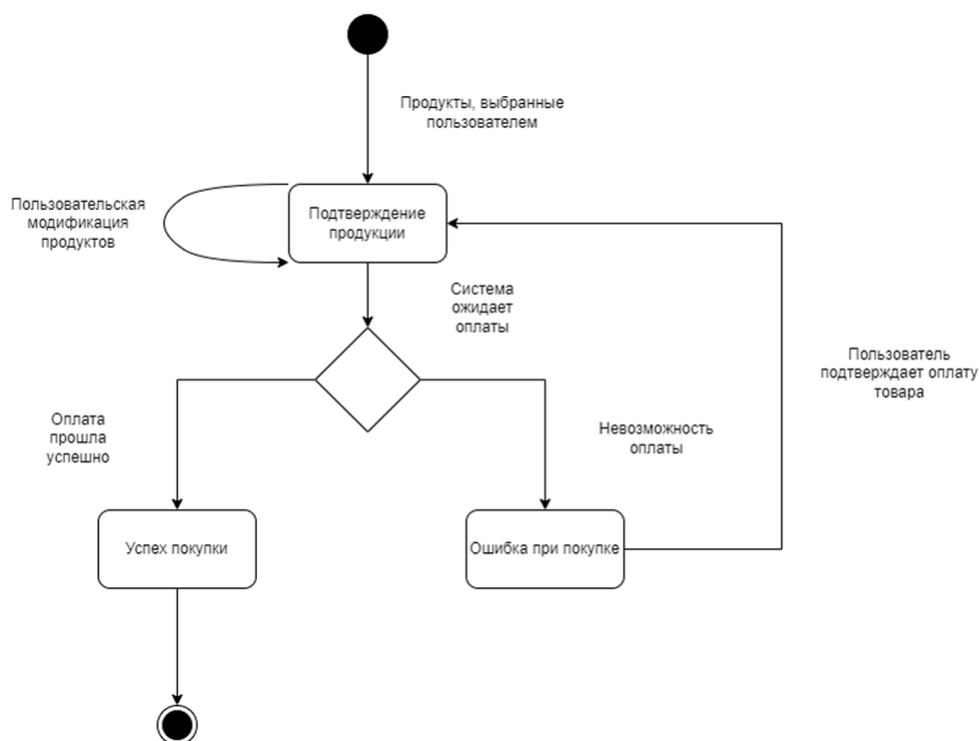


Рисунок 10 – Диаграммн деятельности

#### 2.5.4 Разработка и реализация экспертных рекомендаций

На основании содержимого изображения (Рисунок 11 – Блок-схема консультаций с экспертами) процесс взаимодействия экспертов, администраторов и пользователей реализован следующим образом. Эксперт

публикует статью в разделе «Стена знаний», после чего администратор отправляет материал на модерацию. При успешной аудиторской проверке («Успех аудита») статья становится доступной для просмотра всеми пользователями. Пользователи могут оставлять комментарии или советы под опубликованной статьей через функционал «Совет пользователя эксперту». Эксперты получают уведомления о новых комментариях и просматривают их через свой аккаунт. Для ответа на запросы эксперты используют опции «Посмотреть запрос» и «Дать совет», что позволяет предоставить персонализированные рекомендации. Администраторы отслеживают статусы заявок («Оправить запрос», «Закреть») и анализируют статистику взаимодействия («Посмотреть запрос», «Успехи»). При возникновении конфликтов или технических ошибок (например, «засорение» данных) система предусматривает механизмы коррекции. Схема на Рисунке 11 детализирует последовательность шагов: от инициализации консультации («Начинать») до завершения взаимодействия («Закреть»), обеспечивая прозрачность и контроль на всех этапах.

Кроме того, система предусматривает возможность отслеживания эффективности консультаций через аналитический модуль, который собирает и анализирует данные о количестве просмотров статей, активности комментариев, а также уровне удовлетворенности пользователей. Этот модуль предоставляет экспертам и администраторам подробные отчеты, которые помогают выявлять наиболее востребованные темы, оценивать реакцию аудитории на различные материалы и адаптировать контент с учетом предпочтений пользователей. Система позволяет оценивать эффективность отдельных статей и консультантов, выявляя области для улучшения.

Интеграция с базой данных играет ключевую роль в обеспечении быстрого и удобного доступа к истории взаимодействий пользователей с платформой. Это упрощает процесс анализа, так как все данные о предыдущих консультациях, оценках и комментариях хранятся в едином хранилище, что

ускоряет принятие обоснованных решений и позволяет своевременно вносить корректировки в работу системы. Благодаря такому подходу можно не только улучшать качество обслуживания пользователей, но и повышать общую эффективность работы консультационного сервиса.

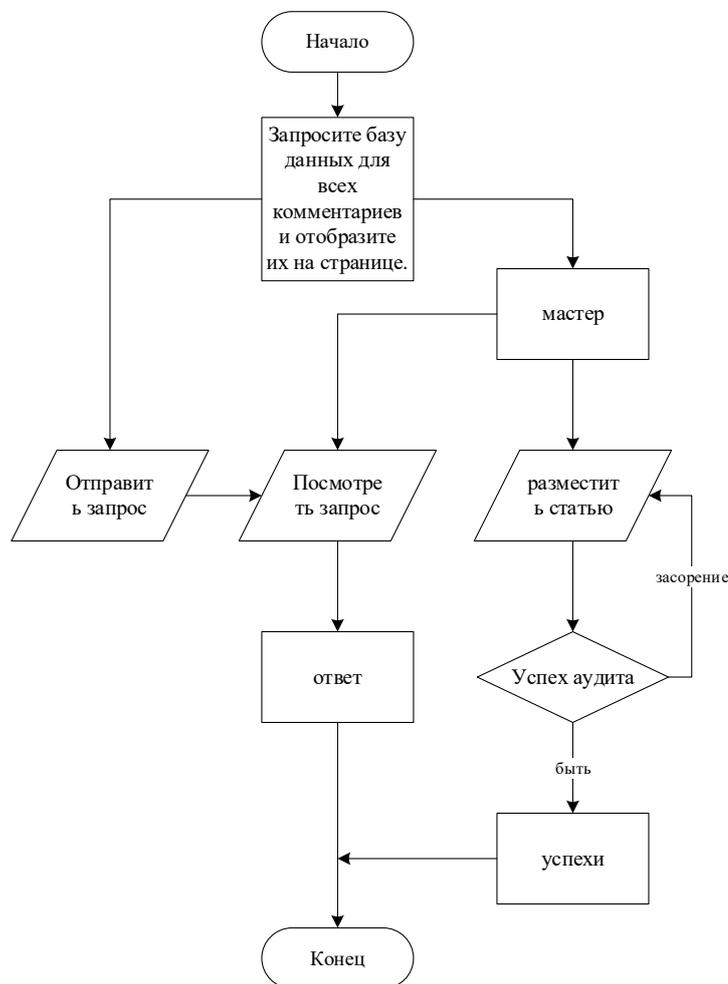


Рисунок 11 – Блок-схема консультаций с экспертами

На Рисунке 12 представлена компонентная структура системы управления здоровьем, где основной модуль HealthKitSystem.exe координирует все процессы, взаимодействуя с другими модулями. DatabaseModule.dll обеспечивает доступ к базе данных и обработку информации о пользователях, заказах и продуктах. BackendLogic.cpp реализует бизнес-логику системы, включая обработку запросов и интеграцию с внешними сервисами. Frontend.html отвечает за пользовательский интерфейс, предоставляя доступ к функционалу через веб-браузер, а HelpDocumentation.hlp предоставляет

руководство для пользователей и администраторов.

Система организована через четко определенные интерфейсы между модулями, что позволяет каждому компоненту выполнять свою задачу, не вмешиваясь в работу других. Это гарантирует гибкость и упрощает поддержку. Модульная структура системы позволяет легко интегрировать новые компоненты, адаптировать её под изменяющиеся требования и интегрировать новые технологии без необходимости значительных изменений в существующем функционале. Такая архитектура делает систему универсальным и масштабируемым решением для управления здоровьем, обеспечивая высокую производительность и безопасность при использовании в различных организациях и для разных пользователей.

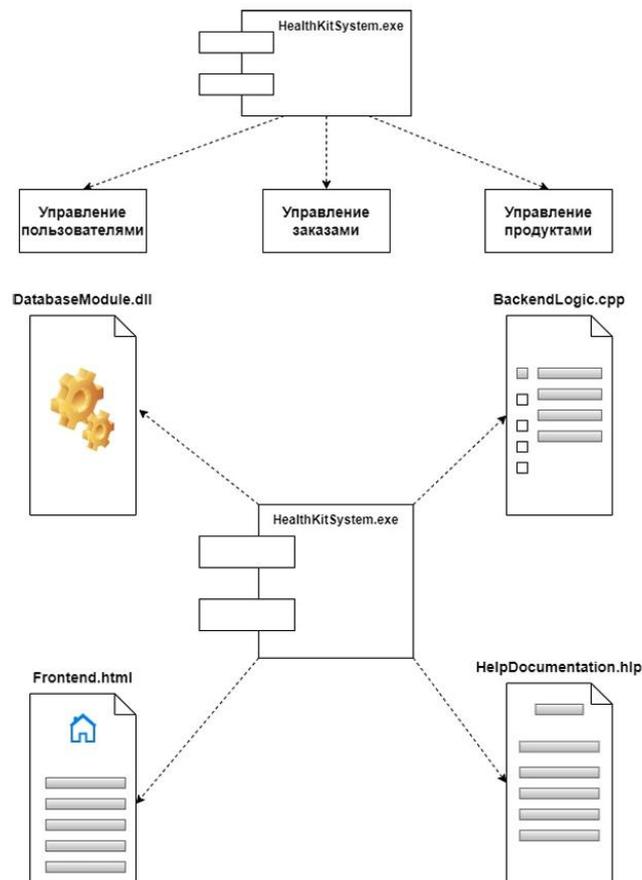


Рисунок 12 – Диаграмма компонентов

## 3 РЕАЛИЗАЦИЯ И ОЦЕНКА СИСТЕМЫ МОНИТОРИНГА ЗДОРОВЬЕМ

### 3.1 Обзор дизайна системы

С быстрым развитием информационных технологий и умных устройств традиционные методы управления здоровьем постепенно превращаются в более цифровые и персонализированные способы. Особенно в условиях глобальной пандемии, переход к онлайн-услугам и цифровым платформам стал ключевым для повышения эффективности медицинских ресурсов и оптимизации предоставления медицинских услуг. В ответ на этот растущий спрос, цель проектирования данной системы заключается в создании многофункциональной платформы для управления здоровьем, которая объединяет управление данными о здоровье, удаленную медицинскую помощь, систему бронирования, платежи и мониторинг здоровья пациента.

Основные пользователи системы – это пациенты, врачи и администраторы. Пациенты могут использовать платформу для управления своими медицинскими записями, записи на прием к врачу, участия в онлайн-консультациях, оплаты медицинских услуг и получения напоминаний о здоровье. Врачи могут использовать платформу для доступа к данным пациентов, организации записей на прием, предоставления консультаций онлайн и обновления медицинских данных пациентов. Администраторы отвечают за общую настройку системы, управление ролями пользователей, резервное копирование данных и другие операции по обслуживанию системы. Каждому типу пользователя предоставляются конкретные функции и доступ, что способствует эффективному и безопасному управлению системой.

Основные требования к системе можно сформулировать следующим образом. Во-первых, модуль управления пользователями должен обеспечивать поддержку различных типов пользователей, таких как пациенты, врачи и

администраторы, с возможностью назначения разных прав и уровней доступа для каждой роли. Во-вторых, система бронирования должна позволять пациентам записываться на прием к врачу, с возможностью подтверждения, отмены и изменения записи в реальном времени. В-третьих, управление данными о здоровье является ключевым функционалом системы, который включает в себя хранение и обработку медицинских данных, таких как отчеты о медицинских осмотрах, исторические записи о лечении и результаты анализов. Эти данные должны быть защищены и легко доступны для врачей, что позволит предоставлять персонализированные рекомендации и лечение. В-четвертых, платежная система должна поддерживать возможность онлайн-оплаты медицинских услуг с интеграцией популярных платежных систем, таких как WeChat Pay и Alipay, обеспечивая удобство и безопасность транзакций. В-пятых, система уведомлений и напоминаний должна автоматически информировать пациентов о предстоящих приемах, обновлениях медицинских данных и других важных событиях, используя SMS, электронную почту или push-уведомления.

Для реализации этих требований была выбрана следующая технологическая база. Для фронтенда использованы React.js и Ant Design. React.js – это один из самых популярных и эффективных фреймворков для создания динамичных и отзывчивых пользовательских интерфейсов. В сочетании с Ant Design, который предоставляет готовые компоненты для создания красивых и удобных интерфейсов, эти инструменты обеспечивают быструю разработку и высокое качество пользовательского опыта.

Для бэкенда использованы Java и Spring Boot. Java – это зрелый и мощный язык программирования, который широко применяется для разработки масштабируемых корпоративных приложений, благодаря своей производительности и поддержке многозадачности. Spring Boot является фреймворком, который упрощает настройку и запуск приложений, предлагая множество полезных функций для создания RESTful-сервисов, что идеально

подходит для построения системы на основе микросервисной архитектуры.

Для хранения данных был выбран MySQL – высокопроизводительная и надежная система управления базами данных, которая обеспечивает поддержку высоких объемов данных и эффективно справляется с большими запросами. В системе предусмотрено использование Redis в качестве кэширования для часто запрашиваемых данных, что позволяет снизить нагрузку на базу данных и ускорить обработку запросов.

Платежная система интегрирована с Alipay и WeChat Pay, что позволяет пользователям удобно оплачивать медицинские услуги через популярные платформы. Это улучшает удобство и безопасность финансовых транзакций в системе.

В архитектуре системы используется многоуровневая модель. Фронтенд реализован с использованием React.js, отвечающий за взаимодействие с пользователями. Бизнес-логика реализована на базе Spring Boot, что позволяет централизованно обрабатывать все ключевые процессы системы, такие как управление пользователями, обработка данных о здоровье и управление записью на прием. Слой данных использует MySQL для хранения всех данных, включая информацию о пользователях, истории заболеваний и оплатах. Кэширование осуществляется через Redis, что улучшает производительность системы при работе с часто запрашиваемыми данными.

Кроме того, безопасность данных является одной из важнейших задач. Все передаваемые данные защищены с использованием HTTPS, а чувствительная информация, такая как пароли пользователей и данные о платежах, хранится в зашифрованном виде. Также реализована многоуровневая система контроля доступа, которая ограничивает возможность выполнения операций в зависимости от роли пользователя.

Система спроектирована таким образом, чтобы обеспечить максимальную гибкость и масштабируемость. Каждый функциональный модуль системы может быть независимо масштабирован и модифицирован без воздействия на

другие части системы. Это позволяет оперативно адаптироваться к изменениям в требованиях или к внедрению новых технологий.

Таким образом, разработка данной системы ориентирована на создание мощного и гибкого инструмента для удаленного управления здоровьем, который будет не только удовлетворять текущим потребностям пользователей, но и обеспечивать возможности для будущих улучшений и расширений.

### **3.2 Дизайн функциональных модулей**

#### **Модуль управления пользователями**

Модуль управления пользователями является одним из ключевых в системе, отвечая за управление различными ролями пользователей и обеспечивая доступ к функциям в пределах разрешенных прав. В системе предусмотрены три типа пользователей: пациенты, врачи и администраторы, каждый из которых имеет свои права и потребности.

Во-первых, регистрация и вход пользователей позволяют пользователям зарегистрироваться с помощью электронной почты, номера телефона или социальных сетей. Пациенты и врачи при регистрации вводят свои личные данные, такие как имя, пол, возраст и медицинская история. После регистрации пациенты могут просматривать свои медицинские записи, записываться к врачу и оплачивать услуги. Врачи могут управлять своим расписанием, просматривать записи пациентов и обновлять планы лечения. Администраторы управляют системой, включая права пользователей, резервное копирование данных и мониторинг системы.

Управление правами осуществляется через контроль ролей, гарантируя, что каждый пользователь может выполнять только те действия, которые разрешены его ролью. Пациенты могут только просматривать и редактировать свои личные данные, врачи могут просматривать и управлять данными своих пациентов, а администраторы имеют полный доступ к системе. Модуль управления правами основан на модели управления доступом на основе ролей (RBAC), что обеспечивает безопасность данных и правильность операций.

Хранение и обработка данных заключаются в том, что все данные пользователей, такие как информация о здоровье, записи о приемах, сохраняются в базе данных системы с использованием технологии шифрования для обеспечения безопасности конфиденциальных данных. Все данные хранятся в базе данных MySQL, и связаны через уникальные идентификаторы пользователей, что обеспечивает согласованность и эффективность запроса данных.

#### Модуль управления данными о здоровье

Модуль управления данными о здоровье является ключевым компонентом системы и играет решающую роль в обеспечении эффективного и безопасного использования медицинской информации. Он отвечает за сбор, хранение, обработку и управление медицинскими данными пациентов, включая отчеты о медицинских осмотрах, историю болезни, использование лекарств, результаты лабораторных анализов, данные о визитах к врачам и другие важные медицинские показатели. Этот модуль гарантирует, что все данные централизованы, структурированы и доступны в любой момент времени для врачей и медицинского персонала, что значительно улучшает качество ухода и ускоряет процесс принятия решений.

Кроме того, модуль управления данными о здоровье интегрируется с другими частями системы, такими как модули анализа и прогнозирования, что позволяет врачу получать на основе медицинских данных точные рекомендации по лечению, а также следить за состоянием пациента в реальном времени. Например, он может отслеживать и напоминать пациентам о необходимости принять лекарства, пройти очередной осмотр или выполнить рекомендации врача.

Эти данные также играют важную роль в предоставлении пациентам всестороннего управления здоровьем, позволяя им отслеживать свое состояние, получать полезные советы по улучшению образа жизни и профилактике заболеваний. В свою очередь, для врачей этот модуль становится ценным

инструментом для точного анализа состояния здоровья пациентов, а также для принятия обоснованных и персонализированных решений о лечении.

Сбор данных происходит двумя основными способами: через данные, которые вводят пациенты, такие как самонаблюдение за состоянием здоровья, описание симптомов и результаты медосмотров, и через данные, которые вводят врачи, например, результаты диагностики, лечебные планы и назначения. Все данные ассоциируются с личной картой здоровья пациента, что гарантирует полноту и точность информации.

Хранение данных осуществляется в базе данных MySQL, где каждый набор медицинских данных сохраняется в отдельных таблицах, например, таблица "Отчеты о медицинских осмотрах", "История болезни". Для повышения эффективности запросов между таблицами используются внешние ключи, что помогает поддерживать согласованность данных.

Обработка и анализ данных заключаются в том, чтобы на основе собранных данных автоматически генерировать отчеты о состоянии здоровья и предлагать персонализированные рекомендации пациентам. Также врачи могут использовать систему для принятия обоснованных решений на основе исторических данных пациентов.

#### Модуль онлайн-бронирования и консультаций

Модуль онлайн-бронирования и консультаций предоставляет пациентам удобный способ записываться на прием к врачу и получать основные онлайн-консультации, что помогает улучшить коммуникацию между пациентами и врачами.

Дизайн системы бронирования позволяет пациентам просматривать доступное время врачей и записываться на прием в выбранное время. Система автоматически обновляет доступность врача в реальном времени, что помогает избежать конфликта времени между пациентом и врачом. Пациенты могут выбирать врачей на основе их специализации, рейтинга или других критериев. Интерфейс системы был спроектирован так, чтобы сделать его простым и

удобным для пользователей.

Управление бронированиями и напоминаниями является важным элементом системы, который обеспечивает бесперебойное взаимодействие между пациентами и врачами, а также помогает минимизировать вероятность пропуска приемов. После успешного бронирования система автоматически отправляет уведомления как пациенту, так и врачу, подтверждая запись и предоставляя важную информацию о времени, месте и типе предстоящей консультации. Это позволяет обеим сторонам заранее подготовиться к встрече и быть уверенными в ее своевременности.

Кроме того, система активно отслеживает приближающееся время приема и отправляет напоминания за день или несколько часов до назначенной встречи. Это гарантирует, что ни пациенты, ни врачи не забудут о предстоящем визите. Напоминания могут быть отправлены через различные каналы, такие как электронная почта, SMS или push-уведомления, что обеспечивает максимальную доступность и надежность.

Если пациент решит отменить или изменить запись, система предоставляет ему возможность сделать это через личный кабинет, что значительно упрощает процесс управления записями и повышает удобство для пользователя. В случае изменения или отмены записи, система автоматически уведомляет врача о таких изменениях, чтобы он мог скорректировать свою расписание и принять необходимые меры. Такой механизм управления бронированиями и напоминаниями помогает повысить уровень организации и удовлетворенности как пациентов, так и медицинского персонала, а также снизить количество неявок и ошибок в расписании.

Онлайн-консультации в системе обеспечиваются с помощью текстовых сообщений. Пациенты могут задать врачу вопросы, описать свои симптомы, а врачи отвечают на них, предоставляя рекомендации и предварительные диагнозы. Хотя видео-консультации в этой версии системы не предусмотрены, текстовые консультации все равно являются важной частью общения для

решения несложных медицинских вопросов.

Запись и архивирование консультаций гарантирует, что все текстовые консультации сохраняются в медицинской карте пациента. Это помогает в будущем отслеживать изменения в здоровье пациента и предоставляет врачам полезную информацию для дальнейшего лечения.

#### Модуль интеграции платежной системы

Модуль интеграции платежной системы отвечает за обработку онлайн-платежей, создание счетов и выдачу электронных квитанций. Он интегрируется с популярными платежными системами, такими как Alipay и WeChat Pay, обеспечивая пациентам безопасный и удобный способ оплаты медицинских услуг.

Интеграция с платежными системами позволяет пациентам выбрать удобный способ оплаты – через мобильные приложения или банковские карты. Платежные операции выполняются через сторонние платежные платформы, что гарантирует безопасность транзакций и защиту персональных данных пользователей.

Запись о платежах и создание счетов происходит автоматически после успешной оплаты. Все платежные данные сохраняются в базе данных, а пациент получает электронный счет, который можно скачать и использовать для отчетности. Каждая операция включает информацию о сумме, времени оплаты и методе оплаты.

Уведомления и напоминания о платежах информируют пациентов о подтверждении оплаты или необходимости произвести оплату. Если платеж не был выполнен в установленный срок, система автоматически отправляет напоминания.

Безопасность и защита данных является одним из ключевых аспектов работы платформы, особенно в контексте обработки платежной информации. Все платежные транзакции проходят через защищенные каналы связи, использующие современные методы шифрования, такие как SSL/TLS (Secure

Socket Layer/Transport Layer Security), что обеспечивает высокий уровень безопасности при передаче данных. Эти технологии шифруют всю информацию, включая данные карт и личные сведения, гарантируя их конфиденциальность и предотвращая перехват данных в процессе передачи.

Платежная информация хранится в зашифрованном виде и доступна только для соответствующих авторизованных пользователей, таких как администраторы системы или служба поддержки, при этом доступ к данным строго ограничен и контролируется. Все операции с платежными данными логируются для обеспечения полной прозрачности и возможности аудита. Это помогает своевременно выявлять и предотвращать возможные попытки несанкционированного доступа или мошенничества.

Дополнительно, для защиты данных применяются многоуровневые системы безопасности, включая двухфакторную аутентификацию (2FA) и регулярные обновления системы безопасности, что повышает общий уровень защиты от внешних угроз и атак. Таким образом, система обеспечивает надежную защиту платежных данных пациентов, обеспечивая их конфиденциальность и безопасность на всех этапах обработки.

### **3.3 Витрина интерфейсов**

Рисунок 13 демонстрирует экран входа в систему для платформы управления здоровьем, который является первым шагом для пользователей, чтобы получить доступ к функционалу системы. На данном экране пользователи должны ввести свое имя пользователя и пароль, чтобы подтвердить свою личность. Поля для ввода данных четко обозначены, что позволяет избежать путаницы и ошибок при вводе. После ввода правильных данных, пользователь может нажать кнопку "Войти", чтобы попасть в свою учетную запись и начать использовать все возможности платформы.

Для новых пользователей предусмотрена кнопка "Регистрация", которая позволяет перейти на страницу создания учетной записи. На странице регистрации пользователи могут ввести необходимые данные, такие как имя,

адрес электронной почты, пароль и другие требуемые сведения, чтобы создать аккаунт и начать использовать систему. Это делает процесс подключения к платформе удобным и доступным для всех категорий пользователей, включая тех, кто ранее не имел учетной записи.

Интерфейс системы спроектирован таким образом, чтобы быть простым и интуитивно понятным. Все элементы управления, включая поля для ввода, кнопки и ссылки, расположены логично и легко доступны. Это позволяет пользователям быстро ориентироваться в системе, минимизируя количество шагов и усилий, необходимых для входа в систему. С таким интерфейсом процесс входа становится легким и удобным, что способствует положительному пользовательскому опыту с самого начала работы с платформой.



Рисунок 13 – Экран входа пользователя в систему

Рисунок 14 демонстрирует страницу регистрации пользователя. На этой странице пользователь должен ввести основную информацию, включая имя пользователя, пароль, номер телефона и адрес электронной почты. Также предусмотрен выбор типа учетной записи (например, обычный пользователь или администратор). После того как все необходимые поля будут заполнены, пользователю нужно нажать кнопку "Подтвердить", чтобы завершить процесс регистрации и создать новую учетную запись для доступа к платформе.

Страница регистрации имеет дружелюбный интерфейс с хорошо организованными полями, что помогает пользователям без труда пройти процесс регистрации. При этом на странице есть понятные подсказки, которые упрощают заполнение формы и делают процесс регистрации более удобным.



健康宝管理

账号注册

fjc001

...

13163516681

普通用户

951220109@qq.com

确定

Рисунок 14 – Экран регистрации пользователя

Этот скриншот демонстрирует интерфейс управления файлами платформы, который предназначен для администраторов и позволяет эффективно управлять и обрабатывать файлы, загруженные пользователями. Интерфейс выполнен в виде таблицы, которая отображает основную информацию о файлах, включая следующие ключевые параметры:

**Номер:** Уникальный идентификатор файла или товара, который позволяет администратору быстро находить нужную запись.

**Продавец:** Информация о продавце, который загрузил файл или товар. Это помогает администратору отслеживать источник данных.

**Название товара:** Название товара, связанного с файлом, что позволяет администратору легко идентифицировать содержимое файла.

Пользователь: Информация о пользователе, который загрузил файл или взаимодействовал с товаром. Это важно для отслеживания активности пользователей.

Время снятия товара с продажи: Временная метка, указывающая, когда товар был снят с продажи. Это помогает администратору управлять актуальностью данных.

Текущий статус: Статус файла или товара (например, "активен", "архивирован", "на модерации"), что позволяет администратору быстро оценить состояние записи.

Каждая строка таблицы представляет собой отдельную запись о файле или товаре, что позволяет администратору легко просматривать и управлять ими. Для каждой записи предусмотрена кнопка "Скачать", которая позволяет администратору загрузить соответствующие файлы для дальнейшего анализа, хранения или распространения. Эта функция значительно повышает эффективность работы администраторов, так как позволяет быстро получать доступ к необходимым данным без необходимости поиска вручную.

В нижней части таблицы реализована функция пагинации, которая отображает количество записей, видимых на текущей странице, и предоставляет возможность перехода между страницами. Это особенно полезно при работе с большими объемами данных, так как позволяет администраторам легко перемещаться между страницами, не перегружая интерфейс. Пагинация поддерживает чистоту и удобство интерфейса, обеспечивая быстрый доступ к нужным данным.

Кроме того, интерфейс поддерживает возможность сортировки по различным столбцам (например, по дате загрузки, статусу или названию товара), что позволяет администраторам быстро находить нужные записи и упорядочивать данные в соответствии с их потребностями. Также предусмотрена функция фильтрации, которая позволяет администраторам отображать только те записи, которые соответствуют определенным критериям

(например, файлы с определенным статусом или товары, загруженные конкретным продавцом).

Таким образом, данный интерфейс управления файлами обеспечивает администраторам удобный и эффективный инструмент для работы с загруженными пользователями данными. Он сочетает в себе простоту использования, функциональность и гибкость, что делает процесс управления файлами и товарами максимально удобным и продуктивным.

编号	商家	商品名称	客户	是否已下单	状态	操作
17	admin	VC银翘片	user	未下单	未下单	<a href="#">下单</a>
18	admin	999感冒灵	user	未下单	未下单	<a href="#">下单</a>
19	admin	连花清瘟颗粒	user	未下单	未下单	<a href="#">下单</a>
20	admin	羚羊感冒片	user	未下单	未下单	<a href="#">下单</a>
22	admin	连花清瘟颗粒	user	未下单	未下单	<a href="#">下单</a>
23	admin	VC银翘片	user	未下单	未下单	<a href="#">下单</a>

总共1页, 共6条数据, 每页 25 条

1 首页 上一页 下一页 尾页

Рисунок 15 – Управление корзиной покупок

Рисунок 16 демонстрирует интерфейс управления корзиной покупок, основная функция которого заключается в эффективном управлении товарами в корзине как пользователем, так и администратором. Интерфейс выполнен с упором на простоту и удобство, что позволяет пользователям легко и быстро выбрать и настроить информацию о товарах в корзине, а также обеспечить точность и актуальность данных.

В верхней части страницы расположен заголовок "Управление корзиной покупок", который четко указывает на текущую задачу интерфейса – управление содержимым корзины. Это помогает пользователю сразу понять, что он находится в соответствующем разделе платформы.

Основная часть интерфейса включает область с информацией о товаре,

где представлены два поля ввода: "Название товара" и "Выбор атрибутов товара". В поле "Название товара" пользователь может ввести наименование товара, и система автоматически предложит подходящие варианты на основе введенных данных. Это значительно ускоряет процесс поиска и выбора товаров. Кроме того, через выпадающее меню пользователь может выбрать атрибуты товара, такие как цвет, размер и другие параметры. Эта функция особенно полезна для товаров с множеством вариантов, например, одежды или аксессуаров, где различные комбинации параметров могут влиять на окончательное решение.

В нижней части интерфейса находятся две кнопки: "Сохранить" и "Вернуться". При нажатии на кнопку "Сохранить" система сохраняет выбранные товары и их атрибуты, добавляя их в корзину покупок. Эта функция гарантирует, что информация о товарах в корзине будет точно отражать выбор пользователя и доступна для дальнейшего использования, например, для оформления заказа. Кнопка "Вернуться" позволяет пользователю выйти из текущего режима и вернуться на предыдущую страницу управления корзиной, чтобы продолжить другие действия, такие как добавление новых товаров или изменения в заказе.

Этот интерфейс представляет собой простое и интуитивно понятное решение, которое позволяет пользователю эффективно управлять товарами в корзине. Предоставление четких функций выбора товаров и конфигурации атрибутов улучшает удобство работы с корзиной покупок, ускоряя процесс выбора товаров и повышая точность данных. В свою очередь, это повышает эффективность работы платформы и удовлетворенность пользователей, обеспечивая удобный и быстрый процесс покупки.



The screenshot shows a web interface for shopping cart management. At the top, there is a header with the text "购物车管理" (Shopping Cart Management) and a link "添加购物车" (Add to Cart). Below the header, there is a section titled "商品信息" (Product Information). In this section, there is a label "品种名称" (Product Name) and a dropdown menu. The dropdown menu is currently open, showing the selected item "连花清瘟颗粒" (Lianhua Qingwen Granules). At the bottom right of the form, there are two buttons: a red button labeled "保存" (Save) and a blue button labeled "返回" (Return).

Рисунок 16 – Корзина Добавить

Рисунок 17 демонстрирует страницу управления товарами, предназначенную для администраторов платформы, с помощью которой осуществляется эффективное управление информацией о товарах. Интерфейс использует таблицу, которая наглядно отображает основную информацию о товарах, что облегчает администраторам процесс просмотра, редактирования и управления товарами. Каждая строка таблицы представляет собой отдельный товар и включает несколько столбцов, обеспечивающих более детальное ознакомление с характеристиками товара.

В первом столбце таблицы расположен номер товара, который уникально идентифицирует товар в системе. Это поле является важным для удобства управления товарами, так как позволяет быстро находить и отслеживать товар по уникальному идентификатору. Далее идет столбец "Продавец", в котором указано, кто загрузил данный товар — продавец или администратор. Это поле помогает администратору отслеживать источник товара и управлять правами доступа и ответственности.

Столбец "Название товара" отображает наименование каждого товара, что позволяет администраторам быстро и точно идентифицировать его. Для удобства восприятия и визуальной идентификации товара также присутствуют столбцы "Цена" и "Изображение товара". В столбце с ценой указана стоимость товара, что помогает администраторам следить за актуальностью ценовой политики, а в столбце с изображением товара отображается фото товара, что облегчает визуальное распознавание и подтверждение информации о товаре.

Столбец "Статус" информирует о текущем статусе товара, например, о том, находится ли он в продаже или снят с продажи. Это помогает администраторам оперативно реагировать на изменения, такие как снятие товара с продажи или добавление новых позиций. В крайнем столбце каждой строки находится кнопка действия, которая предоставляет администратору возможность редактировать информацию о товаре, изменять его цену, статус или удалять его из системы.

Для удобства навигации по большому объему данных предусмотрена система пагинации в нижней части страницы. Пагинация позволяет администраторам легко переходить между страницами с товарами, обеспечивая быстрый доступ ко всем данным, независимо от их количества. Это способствует эффективному управлению товарами и поддерживает своевременные обновления и обработку информации о товарах. В целом, интерфейс страницы управления товарами является удобным и функциональным инструментом для администраторов, обеспечивающим максимальное удобство при работе с товарами на платформе.

列表

添加 刷新

请输入商品名称 提交

编号	商家	商品名称	单价	图片	状态	操作
16	admin	VC银翘片	1.00		待审核	通过
17	admin	999感冒灵	25.00		待审核	通过
18	admin	连花清瘟颗粒	25.00		待审核	通过
19	admin	羚羊感冒片	7.00		待审核	通过
20	admin	白云山感冒灵颗粒	9.00		待审核	通过
21	admin	复方氨酚烷胺胶囊	19.00		待审核	通过

总共1页, 共6条数据。 每页 25 条

1 首页 上一页 下一页 尾页

Рисунок 17 – Управление товарами

Рисунок 18 демонстрирует страницу управления заказами, предназначенную для администраторов, что позволяет эффективно управлять заказами пользователей. Страница оформлена в виде таблицы, которая ясно отображает основную информацию о каждом заказе, включая уникальный номер заказа, продавца, название товара и пользователя, который разместил заказ. Также в таблице содержится информация о времени снятия товара с продажи и статусе заказа, что помогает администратору отслеживать текущий процесс выполнения заказов.

В конце каждой строки находится кнопка действия, с помощью которой администратор может изменять статус заказа, отменять его или удалять. В

нижней части страницы предусмотрена функция пагинации, которая позволяет легко переключаться между страницами заказов при большом объеме данных. Эта функция помогает улучшить управление заказами, ускоряя процесс поиска и обработки информации, что значительно повышает эффективность работы платформы.

编号	商家	商品名称	客户	是否已下单	状态	操作
12	admin	999感冒灵	user	已下单	已完成	
13	admin	白云山感冒灵颗粒	user	已下单	已完成	
14	admin	复方氨酚烷胺胶囊	user	已下单	待审核	
15	admin	999感冒灵	user	已下单	已完成	
16	admin	羚羊感冒片	user	已下单	待审核	

总共1页, 共5条数据, 每页 25 条

1 首页 上一页 下一页 尾页

Рисунок 18 – Диаграмма интерфейса пользовательского заказа

Рисунок 19 демонстрирует интерфейс управления консультациями экспертов, предназначенный для эффективного администрирования взаимодействия между пользователями и экспертами. Страница представлена в виде таблицы, которая предоставляет полную информацию по каждому случаю консультации, позволяя администратору удобно просматривать данные, выполнять необходимые действия и обеспечивать высокое качество обслуживания.

Первый столбец таблицы содержит уникальный номер, который идентифицирует каждую запись консультации внутри системы. Это поле облегчает поиск конкретных консультаций и обеспечивает точную идентификацию записи в базе данных. Далее идет столбец "Название документа", в котором указана суть запроса или тема консультации, что помогает понять, о каком вопросе шла речь, и какой именно помощи требовал пользователь. В столбце "Имя пользователя" указано, кто именно инициировал консультацию. Это важно для отслеживания источников обращения, а также для

корректной связи пользователя с его заявкой, что облегчает дальнейшее управление и анализ.

Столбец "Имя эксперта" показывает, какой специалист предоставил консультацию, что позволяет администратору контролировать распределение задач между экспертами и оценивать их работу. Это также помогает анализировать нагрузку на специалистов и их эффективность. Два столбца, "Оценка пользователя" и "Оценка эксперта", являются важными элементами обратной связи. Оценка пользователя отражает его удовлетворенность консультацией, что служит показателем качества предоставленной услуги. Оценка эксперта, в свою очередь, позволяет специалисту выразить мнение о взаимодействии с пользователем, его активности и корректности общения. Эти взаимные оценки создают двустороннюю систему обратной связи, что позволяет платформе анализировать процесс взаимодействия и работать над его улучшением.

В последнем столбце таблицы расположена кнопка "Оценить", которая предоставляет администратору возможность просмотреть или отреагировать на конкретную консультацию. Это дает возможность оперативно управлять качеством сервиса, решать возникшие вопросы и принимать меры по улучшению работы системы. Нижняя часть страницы включает систему пагинации, которая позволяет эффективно навигировать по записям консультаций, особенно при большом объеме данных. Пагинация делает процесс поиска и работы с информацией удобным и быстрым.

Благодаря такому дизайну интерфейса платформа обеспечивает прозрачность взаимодействия между экспертами и пользователями, улучшает контроль качества консультаций и способствует повышению общего уровня удовлетворенности пользователей. Это позволяет администраторам эффективно управлять данными, контролировать качество предоставляемых услуг и обеспечивать высокие стандарты работы платформы.

编号	文章名称	用户名称	专家名称	用户评价	专家反馈	操作
4	嗓子疼	user	专家	非常感谢	不客气	<a href="#">评价</a>
5	腰疼	user	专家	非常感谢	不客气	<a href="#">评价</a>

总共1页, 共2条数据。 每页  条

1 [首页](#) [上一页](#) [下一页](#) [尾页](#)

Рисунок 19 – Диаграмма консультаций экспертов

Рисунок 20 демонстрирует интерфейс управления меню "Стена знаний", предназначенный для публикации статей экспертами и их последующей модерации администраторами. Этот интерфейс представлен в виде таблицы, которая отображает информацию о публикациях, включая уникальные идентификаторы, заголовки, авторов, статус и другие важные данные, что облегчает администраторам процесс просмотра, модерации и управления статьями.

В таблице в первом столбце отображается номер каждой статьи, что позволяет администратору быстро найти и управлять различными статьями. Столбец содержание включает заголовок или краткое описание статьи, что помогает администратору понимать общую тему материала. В столбце ссылка указана ссылка на статью, по которой администратор может перейти, чтобы ознакомиться с полным текстом. Столбец автор показывает имя эксперта, опубликовавшего статью, что помогает администратору отслеживать источники материалов. В столбце статус отображается текущий статус статьи, например, "На рассмотрении", что указывает на то, что статья ещё не прошла проверку и ожидает одобрения.

В конце каждой строки статьи расположена кнопка действия, которая позволяет администратору перейти к процессу модерации и проверить, соответствует ли содержание статьи стандартам платформы. Если статья удовлетворяет требованиям, она проходит модерацию и становится доступной на Стене знаний для пользователей. Если статья не прошла проверку, администратор может отправить запрос на корректировку, предложив экспертам

внести необходимые изменения.

Этот интерфейс помогает администраторам эффективно управлять контентом, публикуемым экспертами, гарантируя, что все статьи на Стене знаний соответствуют стандартам платформы. Процесс модерации ясен и прозрачен, что позволяет платформе предоставить высококачественный контент, доступный пользователям на Стене знаний.



列表

添加

编号	内容	描述	作者	状态	操作
2	感冒	可以多喝热水吃感冒药	专家	审核通过	
3	新冠病毒	需要准备退烧药	专家	审核通过	
6	嗓子疼	可以吃点金嗓子喉片	专家	审核通过	
7	腰疼	可以吃点止痛药	专家	审核通过	

Рисунок 20 – Настенная диаграмма экспертных знаний

### 3.4 Архитектура системы и реализация модулей

#### Взаимодействие между модулями

Система использует одноуровневую архитектуру, состоящую из нескольких ключевых модулей, таких как модуль управления пользователями, модуль управления товарами и модуль управления заказами. Модули взаимодействуют между собой через базовые API-интерфейсы, что гарантирует эффективный обмен данными и бесперебойную работу системы.

Взаимодействие модуля управления пользователями с модулем управления товарами: при регистрации пользователь проходит проверку, и его данные сохраняются в системе. Модуль управления товарами отображает информацию о товарах, в зависимости от того, зарегистрирован ли пользователь. Когда пользователь просматривает товары, модуль управления товарами отображает информацию в соответствии с его правами доступа (например, для зарегистрированных и незарегистрированных пользователей).

Взаимодействие модуля управления заказами с модулем управления товарами: когда пользователь выбирает товары и оформляет заказ, модуль

управления заказами получает данные о товаре (цену, количество и другие параметры) от модуля управления товарами и создает заказ. После этого модуль управления заказами обновляет статус заказа, например, "Ожидает оплаты", "Оплачено" и т.д.

Через это взаимодействие между модулями система поддерживает согласованность данных и плавность процессов. Каждый модуль выполняет свою задачу, и данные передаются и обрабатываются эффективно и понятно.

#### Хранение и управление данными

Для хранения данных используется база данных MySQL. Дизайн базы данных включает следующие основные таблицы:

Таблица пользователей: хранит базовую информацию о пользователях, такую как имя пользователя, пароль, контактные данные и т.д., что позволяет управлять пользователями и проверять их учетные данные.

Таблица товаров: хранит информацию о товарах, такую как название товара, цена, количество на складе и другие параметры. Все изменения данных товаров выполняются через модуль управления товарами, что обеспечивает актуальность информации.

Таблица заказов: хранит информацию о заказах, включая номер заказа, данные о товаре, статус заказа и другие детали. Модуль управления заказами контролирует данные в этой таблице и следит за точностью и своевременностью обновлений.

Дизайн базы данных основан на принципах реляционной базы данных, все таблицы связаны с помощью внешних ключей, что позволяет поддерживать целостность и согласованность данных.

Для обеспечения безопасности данных система использует методы шифрования для хранения конфиденциальной информации, такой как пароли, и применяет базовые механизмы контроля доступа, чтобы только авторизованные пользователи могли изменять или просматривать данные.

### 3.5 Тестирование системы

Тестирование программных систем представляет собой комплексный процесс, который охватывает не только тестирование программного обеспечения (ПО), но и аппаратное и периферийное оборудование, от которого зависит функционирование ПО. Это также включает в себя определенные данные, вспомогательные программные системы и другие компоненты, которые могут повлиять на работу системы. Главной целью тестирования является подтверждение того, что конечная программная система соответствует требованиям продукта и системе дизайна, обеспечивая при этом необходимую производительность, надежность и функциональность.

Основными аспектами производительности, которые подлежат тестированию, являются быстродействие системы, время простоя, а также функциональность запросов к базе данных. В частности, система должна обеспечивать быстрые и эффективные запросы данных и высокую надежность работы, что критично для обеспечения стабильности работы в реальных условиях. Тестирование должно выявить потенциальные проблемы, такие как увеличение времени отклика системы при высоких нагрузках или проблемы с масштабируемостью при увеличении числа пользователей.

После проведения тестов на надежность, если система показывает сбои или ошибки, возможной причиной может быть переполнение пула соединений базы данных из-за слишком большого числа запросов на доступ. Когда пул соединений базы данных достигает максимального размера, новые запросы не могут быть выполнены, что приводит к таймаутам соединений и сбоям в работе приложения. Для предотвращения таких ошибок необходимо тщательно следить за нагрузкой на систему и правильно управлять пулом соединений, а также провести дополнительные тесты, чтобы удостовериться в его корректной работе при высоких нагрузках.

Одним из важных аспектов, который также следует учитывать, является безопасность системы. В случае возникновения ошибок или подозрительных

действий, связанных с доступом к базе данных, необходимо обеспечить механизм перехвата вредоносных попыток доступа. Это включает в себя анализ логов и установку фильтров для выявления разрушительных операций на определенных IP-адресах. Периодический мониторинг таких операций позволяет своевременно реагировать на угрозы безопасности и предотвращать потенциальные атаки на систему.

Таким образом, тестирование программных систем должно быть всесторонним и охватывать как производственные, так и безопасностные аспекты, что способствует обеспечению стабильности, производительности и безопасности работы конечной программной системы.

### 3.5.1 Аппаратная среда

Информационная система управления здравоохранением на базе Java была разработана и реализована в среде Lenovo Legion Y7000P, которая обеспечила отличную производительность для разработки программного обеспечения. Несмотря на выполнение сложных вычислительных задач, таких как обработка данных и многозадачность, данный компьютер продолжает показывать отличные результаты и не испытывает значительных задержек при выполнении программы. Благодаря мощному процессору и достаточному объему оперативной памяти, Lenovo Legion Y7000P эффективно справляется с требованиями программного обеспечения для данного проекта, обеспечивая стабильную работу системы и ускорение обработки данных, что крайне важно для бесперебойного функционирования системы управления здравоохранением. В целом, использование Lenovo Legion Y7000P для разработки системы является оптимальным выбором, так как оно обеспечивает успешную реализацию всех необходимых функций без задержек и ошибок. Высокая производительность Lenovo Legion Y7000P не только обеспечила плавное развитие проекта, но и стала надежной основой для стабильности и масштабируемости системы. По мере увеличения функционала проекта, этот компьютер продолжает работать эффективно, удовлетворяя все растущие

требования системы.

### 3.5.2 Программная среда

В данном аспекте тестирования программного обеспечения используется комплексная тестовая структура, которая начинается с разработки структуры тестирования для всей функциональности системы. Этот процесс начинается с анализа требований и создания детализированного тестового плана, который охватывает все ключевые аспекты работы системы. На первом этапе проводится тестирование основных функций системы, что позволяет удостовериться в их правильности и стабильности. Это включает в себя проверку всех критически важных процессов, таких как регистрация пользователей, обработка данных, взаимодействие с внешними сервисами и базами данных.

Далее, с использованием инструментов отладки в браузере, выполняются поэтапные субтесты для каждого бизнес-модуля. Это позволяет детально проверять работу отдельных компонентов, их функциональность и взаимодействие друг с другом. Например, для проверки корректности бизнес-логики могут быть проведены тесты на обработку входных данных, выполнение расчетов или передачу данных между различными модулями. Каждый субтест направлен на выявление проблем на уровне конкретных функциональных блоков, что обеспечивает более точную диагностику ошибок.

После выполнения тестов производится всесторонний анализ результатов. Этот этап включает в себя сравнение фактических выходных данных с ожидаемыми результатами, что позволяет выявить любые несоответствия или ошибки. Важно внимательно изучить как положительные, так и отрицательные результаты тестирования, так как это помогает не только устранить баги, но и улучшить архитектуру системы. В случае обнаружения ошибок анализируются возможные причины их возникновения, и при необходимости вносятся изменения в код.

В заключение, выполняется нагрузочное тестирование, которое направлено на проверку производительности системы при высоких нагрузках.

Это тестирование имитирует реальную эксплуатацию системы с большим количеством пользователей или интенсивной обработкой данных, чтобы убедиться, что система сохраняет стабильность и эффективность в условиях пиковых нагрузок. Нагрузочное тестирование помогает выявить узкие места, такие как медленная обработка запросов, недостаточная производительность базы данных или проблемы с масштабируемостью, которые могут возникнуть в реальных условиях эксплуатации.

Эти шаги в совокупности помогают точно определить конечные результаты тестирования и удостовериться в готовности системы к полноценному использованию. Кроме того, комплексный подход к тестированию позволяет не только выявить ошибки, но и повысить общую производительность, надежность и безопасность системы, что в конечном итоге приводит к более качественному и стабильному продукту.

### 3.5.3 Тестовый пример

Для системы, вводимой в эксплуатацию, каждую функцию необходимо протестировать на конкретном экземпляре, чтобы убедиться в её целостности и корректной работе. Это критически важный этап, который позволяет не только выявить потенциальные ошибки в работе системы, но и убедиться в том, что все функции работают в соответствии с заданными требованиями. Тестирование должно охватывать все возможные сценарии использования, включая как стандартные операции, так и крайние случаи, которые могут возникнуть в реальных условиях эксплуатации. Это помогает избежать непредсказуемых сбоев и убедиться в стабильности системы при различных условиях работы.

Тестирование сценариев использования ориентировано на системные требования и функциональность, с учетом потребностей пользователей, администраторов и специалистов. Это значит, что каждый сценарий тестирования должен отражать реальные задачи и операции, которые могут быть выполнены в системе. Например, для пользователей тестируется

возможность регистрации, выполнения покупок, или получения консультаций, а для администраторов — управление данными, обработка запросов или настройка параметров системы. Для специалистов могут быть важны тесты на безопасность, производительность и интеграцию системы с другими сервисами.

Тестирование должно подтверждать соответствие системы ожиданиям заинтересованных сторон, включая проверку функциональности, удобства и эффективности для пользователей, а также устранение несоответствий с требованиями.

Таблица 7 приведена для демонстрации примеров тестирования системы в реальных условиях, включая сценарии регистрации пользователя, добавления товаров в корзину, операций с базой данных и работы при высокой нагрузке. Эти примеры помогают понять, как система будет функционировать в эксплуатации и удостовериться в её готовности к внедрению.

Таблице 7 – Сводка тестовых примеров

Тестовые функции	Тестовые примеры	Данные	Ожидаемые результаты	Фактически результаты	Заключение
Модуль входа в систему	Сообщение введено с неправильным номером счета или паролем.	Ошибка	Отказ	Неверная информация для входа в систему	Нормальный
Модуль входа в систему	Введите правильный номер счета и пароль	Правильный	Успешно	Успешный вход в систему и домашняя страница	Обычный

Модуль регистрации и	Ввод некорректного номера почтового ящика и мобильного телефона.	Введите номер телефона и почтовый ящик.	Подсказка о неправильном формате	Выдает неправильный формат и регистрация не проходит	Обычный
Запрос пользователей	Откройте интерфейс управления пользователями	Введите правильное имя пользователя	Успешный запрос	Нормальное отображение	Нормальный
Одобрено	Нажмите на кнопку Пройти	Нажмите кнопку Пройти	Пользователи могут войти в систему и выполнять операции.	Возврат успеха, обновление списка пользователей	Обычный
Добавление продуктов	Откройте управление продуктами и добавьте данные о продукте.	Введите только неправильное количество	Не удалось добавить продукт	Не удалось добавить, неверное количество	Обычный
Добавить продукт	Введите полную информацию	Введите корректные данные и количество	Успешно добавлено	Успешно добавлено	Обычный

Одобрено	Выберите продукт для проверки и нажмите кнопку	Товарный номер	Пользователи могут просматривать информацию о продукте	Изменение статуса проверки	Обычный
Одобрено	Нажмите кнопку Пройти	Номер заказа	Статус заказа изменен нормально	Информация о заказе изменена нормально, обновите список	Обычный
Завершение заказа	Нажмите на кнопку Завершить	Номер заказа	Изменение статуса заказа и создание комментария к заказу.	Изменение статуса заказа	Обычный
Обратная связь с администратором	Нажмите на кнопку комментарий	Номер заказанного комментария	Нормальный отзыв администратора	Успех обратной связи с комментарием к заказу	Обычный

Эксперты Публикуют стену знаний	Нажмите кнопку Добавить	Основная информация о стене знаний	Добавить как нормальный	Добавить успешно	Обычный
Обзор администратора	Нажмите кнопку Пройти	Выберите статью эксперта для просмотра и утверждения	Одобрено	Одобрено	Обычный

### 3.5.4 Краткое описание теста

В этой главе было проведено комплексное функциональное тестирование системы, которое включало проверку всех ключевых аспектов её работы. Тесты, выполненные в различных браузерах, подтвердили её работоспособность и корректное отображение всех элементов интерфейса, что гарантирует пользователям стабильную и удобную работу на платформе. Для более детального анализа и устранения возникающих ошибок были использованы инструменты отладки и протоколирования, которые позволили выявить потенциальные проблемы и оперативно внести необходимые коррективы в код, что в свою очередь способствовало его оптимизации.

Для проверки функциональности системы была использована платформа JUnit, а также другие инструменты тестирования, которые позволили системно и полноценно проверить каждую из функций системы. В результате тестирования было подтверждено, что все компоненты системы функционируют согласно установленным требованиям и ожидаемым результатам.

Кроме того, обновление содержимого программы производится автоматически, что позволяет сразу увидеть результаты изменений и убедиться

в их корректности. Этот процесс ускоряет проверку изменений и обеспечивает непрерывность работы системы. Все функции системы управления здоровьем, разработанные на базе Java, прошли успешное тестирование и функционируют стабильно, удовлетворяя потребности пользователей. Это тестирование подтвердило, что система полностью готова к внедрению и может быть использована для решения задач в реальных условиях эксплуатации.

### **3.6 Будущие улучшения и расширения**

В этой части рассматриваются направления будущего развития системы, обсуждаются возможные улучшения и расширения. С учетом постоянного технологического прогресса и изменения потребностей пользователей, система имеет значительный потенциал для улучшения и развития.

Одним из важных направлений будущего развития является расширение функционала системы. С учетом быстрого развития искусственного интеллекта и машинного обучения, система может интегрировать более интеллектуальные функции. Например, с использованием AI и машинного обучения система сможет проводить прогнозы здоровья, основываясь на исторических данных пользователя и результатах мониторинга в реальном времени. Это позволит не только предоставлять персонализированные рекомендации по здоровью, но и предсказывать потенциальные риски для здоровья, позволяя принимать превентивные меры заранее. Кроме того, можно добавить такие функции, как автоматическая диагностика и интеллектуальные рекомендации, что значительно повысит уровень интеллектуализации системы.

Важной частью будущего развития являются меры по оптимизации. С увеличением числа пользователей и расширением функциональности, производительность системы должна быть постоянно улучшена. Это может включать в себя оптимизацию алгоритмов, улучшение методов обработки данных, повышение способности системы к обработке множества запросов и сокращение времени отклика. Для улучшения пользовательского опыта система должна регулярно обновляться с учетом обратной связи от пользователей,

улучшая интерфейс и процессы взаимодействия, чтобы они более точно соответствовали потребностям пользователей.

Что касается рыночных приложений, то система обладает широкими перспективами и значительным потенциалом на рынке. В условиях растущего интереса людей к здоровью и увеличения потребности в услугах медицинского обслуживания, рыночные перспективы системы будут только расширяться. Система может быть ориентирована как на индивидуальных пользователей, предоставляя им персонализированные услуги по управлению здоровьем, так и на медицинские учреждения, такие как больницы и клиники, помогая им эффективно управлять пациентами, вести медицинские записи и анализировать данные о здоровье.

Кроме того, система имеет потенциал для интеграции с другими медицинскими платформами и сервисами, создавая возможности для обмена данными, сотрудничества и улучшения качества медицинского обслуживания. Такая интеграция значительно расширяет области применения системы и открывает новые рыночные возможности, включая подключение к более широким экосистемам здравоохранения.

В будущем бизнес-модели системы могут включать не только подписные услуги для пользователей и медицинских учреждений, но и различные аналитические решения, основанные на обработке данных о здоровье, что позволяет дополнительно монетизировать платформу. Кроме того, система может предоставлять индивидуализированные отчеты, консультации по медицинским вопросам, а также другие формы услуг, направленные на удовлетворение потребностей различных рынков. Это, в свою очередь, увеличивает гибкость и адаптивность системы, делая её привлекательной для множества потенциальных клиентов и пользователей в разных сегментах.

## ЗАКЛЮЧЕНИЕ

Выпускное проектирование - это важный этап начального освоения студентами общего метода инженерного проектирования, ключевой процесс трансформации базовых и профессиональных знаний в прикладные способности, а также важнейшее звено в развитии самостоятельных инженерных способностей студентов. Всестороннее знакомство с системой позволяет пользователям понять ее суть, лучше использовать систему для решения собственных задач, а также более четко и лаконично представить бизнес-процессы с помощью презентационного слоя. Кратко опишите дизайн системы данных для составления медицинских отчетов, перечислив некоторые блоки кода для основных функций. Перечислите функциональный дизайн и реализацию системы и дайте общий обзор основных функций, включенных в систему, используя функциональные тесты для общего представления результатов тестирования. На втором этапе система может быть оптимизирована, например, путем улучшения эстетики интерфейса. При дальнейшей разработке можно рассмотреть возможность добавления функций, связанных с безопасностью данных, таких как информация о номере мобильного телефона и адресе электронной почты, в виде фронтального десенсублизованного дисплея. Эта система не создана с использованием микросервисных фреймворков, которые обычно относятся к монолитной архитектуре, что имеет определенные недостатки для вторичного расширения обслуживающего персонала на более позднем этапе. Если эту проблему необходимо решить, ее нужно создавать с использованием современных основных микросервисных фреймворков, которые поддерживают субручное обслуживание на более позднем этапе. Будущая система управления здоровьем будет более интеллектуальной, персонализированной и комплексной, предоставляя пользователям более полные, точные и эффективные услуги по управлению здоровьем.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ван, Мэй. Цифровая медицинская платформа и её влияние на улучшение качества жизни / Мэй Ван // Журнал международного здравоохранения и медицины. – 2021. – № 25(4). – С. 78 – 83.

2. Ван, Тао; Сюй, Янь. Построение и оптимизация онлайн - медицинских сервисов в разных регионах / Тао Ван, Янь Сюй // Международный журнал медицины и системы управления здравоохранением. – 2020. – № 14(6). – С. 50 – 55.

3. Ван, Цзюань. Цифровая медицина и интеграция платформ управления здоровьем / Цзюань Ван // Журнал китайской цифровой медицины. – 2020. – № 9(4). – С. 70 – 75.

4. Ван, Цян; Чэнь, Фэй. Роль мобильных медицинских приложений в улучшении управления здоровьем / Цян Ван, Фэй Чэнь // Журнал китайской телемедицины. – 2020. – № 13(5). – С. 33 – 37.

5. Ван, Юй; Ли, Хуа. Проектирование и внедрение системы онлайн - оплаты медицинских услуг / Юй Ван, Хуа Ли // Журнал современной медицинской информационной системы. – 2020. – № 22(3). – С. 80 – 85.

6. Гао , Мин. Способы повышения эффективности медицинского обслуживания через цифровизацию / Мин Гао // Журнал китайской цифровой медицины. – 2021. – № 17(3). – С. 22 – 27.

7. Гао, Юй; Хан, Донг. Цифровизация в здравоохранении и её влияние на доступность услуг / Юй Гао, Донг Хан // Журнал китайской электронной науки и технологий. – 2021. – № 33(2). – С. 110 – 115.

8. Ли, Мин; Ван, Лэй. Интеграция удалённого мониторинга здоровья и онлайн - медицинских платформ / Ли Мин, Лэй Ван // Журнал китайской электронной науки и технологий. – 2021. – № 28(6). – С. 58 – 62.

9. Ли, Сюй. Проблемы безопасности данных в онлайн - медицине / Сюй Ли // Журнал китайской безопасности медицины. – 2021. – № 27(1). – С. 40 – 45.

10.Ли, Сяо.; Чжан, Ян. Облачные технологии в управлении медицинскими данными и безопасности данных / Сяо Ли, Ян Чжан// Журнал китайской медицины и здоровья данных. – 2021. – № 23(2). – С. 44 – 49.

11.Ли, Я; Чжан, Хао. Дизайн и внедрение мобильных платформ для управления здоровьем / Я Ли, Хао Чжан // Журнал китайского здоровья и медицинской науки. – 2021. – № 8(2). – С. 62 – 67.

12.Ли, Ли; Чэнь, Жуй. Будущее цифровых платформ для управления здоровьем / Ли Ли, Жуй Чэнь // Журнал китайской медицины и технологий. – 2021. – № 20(2). – С. 39 – 44.

13.Лю, Синь. Инновации и вызовы онлайн - медицинских услуг / Синь Лю // Журнал глобальной медицины. – 2021. – № 29(6). – С. 12 – 17.

14.Лю, Цян. Защита данных в мобильных медицинских приложениях / Цян Лю // Журнал китайской интернет - технологий. – 2020. – № 19(7). – С. 120 – 125.

15.Лю, Шань. Будущее развитие цифровых технологий в здравоохранении / Шань Лю // Международный журнал общественного здравоохранения. – 2021. – № 19(8). – С. 102 – 107.

16.Сунь, Хун. ; Ли, Ган. Применение смартфонов в управлении здоровьем / Хун Сунь, Ган Ли// Журнал китайской интеллектуальной медицины. – 2020. – № 9(1). – С. 51 – 56.

17.Сюй, Дан; Го, Ли. Влияние виртуальных медицинских платформ на улучшение медицинского обслуживания в сельских районах / Дан Сюй, Ли Го// Журнал китайской сельской медицины. – 2021. – № 18(7). – С. 45 – 50.

18.У, Линь. Онлайн - медицинские сервисы: проверка безопасности и доступности / Линь У // Журнал китайской медицинской безопасности. – 2021. – № 28(3). – С. 60 – 65.

19.Фолсом, Лисал. Современные проблемы здравоохранения в развитых странах: инфраструктурные и организационные вызовы / Лисал Фолсом // Международный журнал медицины и здравоохранения. – 2020. – № 11(4). – С.

45 – 50.

20.Хэ, Сянни. Роль онлайн - сервисов в улучшении здравоохранения в современных социальных условиях / Сянни Хэ // Журнал здравоохранения и технологий. – 2021. – № 15(3). – С. 25 – 30.

21.Чжан, Вэй; Ли, На. Применение и развитие онлайн - медицинских сервисов / Вэй Чжан, На Ли // Журнал китайской информационной медицины и здравоохранения. – 2019. – № 16(2). – С. 108 – 112.

22.Чжан, Цзе; У, Фэн. Цифровая платформа здравоохранения и её влияние на повышение качества жизни / Цзе Чжан, Фэн У // Журнал китайской медицинской информационной науки. – 2021. – № 26(4). – С. 134 – 139.

23.Чжан, Нин. Проектирование платформы управления медицинскими данными с использованием больших данных / Нин Чжан // Журнал китайской медицинской информационной науки. – 2021. – № 24(8). – С. 72 – 77.

24.Чжао, Мин; Ван, Хао. Применение искусственного интеллекта в здравоохранении / Мин Чжао, Хао Ван // Журнал китайского искусственного интеллекта. – 2020. – № 13(6). – С. 130 – 135.

25.Чжоу, Тин; Чжао, Лян. Применение мобильных платёжных систем в онлайн - медицине / Тин Чжоу, Лян Чжао // Журнал китайской мобильной связи. – 2020. – № 32(4). – С. 70 – 75.

26.Чжоу, Син. ; Пен, Сяо. Проблемы интеграции цифровых технологий в медицинские системы / Син Чжоу, Сяо Пен // Журнал китайской мобильной медицинской информационной технологии. – 2020. – № 12(5). – С. 98 – 103.

27.Чжоу, Хуа. Модели безопасности и защиты данных в цифровых платформах / Хуа Чжоу // Журнал китайской медицинской безопасности. – 2021. – № 23(6). – С. 45 – 50.

28.Чэнь, Фэн. Исследования и разработки системы онлайн медицинских платежей / Фэн Чэнь // Журнал китайских финансов и здравоохранения. – 2020. – № 34(8). – С. 21 – 26.

29.Чэнь, Хай; Чжао, Ся. Развитие платформ для обмена медицинской

информацией / Хай Чэнь, Ся Чжао // Журнал китайской интернет - медицины. – 2020. – № 15(5). – С. 101 – 106.

30.Чэнь Чэнь. Роль современных технологий в улучшении медицинских сервисов / Чэнь Чэнь // Журнал китайской информационной технологии и технологий. – 2019. – № 20(9). – С. 119 – 124.