Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных технологий

Кафедра инженерной психологии и эргономики

Дисциплина: управление информационными проектами

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

Разработка информационной системы «Автоматизация учета лекарств для сети аптек»

Студент: гр. 910903 Логвин В. В.

Руководитель: Борисик М. М.

Минск 2022

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерного проектирования

УТВЕРЖДАЮ

Заведующий кафедрой ИПиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ T.В. Казак

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 года

ЗАДАНИЕ

по курсовому проектированию

Студенту(–ке) Логвину Владислав Валерьевичу*–––––––––––*

1. Тема проекта Разработка информационной системы «Автоматизация учета лекарств для сети аптек»

2. Срок сдачи студентом законченного проекта––––1 *03.05.2022 –*

3. Исходные данные к проекту *Основные характеристики информационной системы и базы данных (БД): 1) не менее 10 сущностей; 2) 3–я нормальная форма; 3) назначение – в соответствии темой; 4) БД разрабатывается и реализуется с помощью СУБД PostgreSQL; 5) клиентская часть информационной системы разрабатывается на объектно–ориентированном языке программирования Java и реализуется с использованием средств доступа к данным Spring Data и Hibernate; 6) клиентская часть информационной системы должна обеспечивать возможность добавления выборки, поиска и модификации данных в БД (добавление, удаление, обновление); 7) должно быть реализовано два режима работы с БД: режим администратора и пользователя (в режиме администратора доступны все функции манипулирования данными, в режиме пользователя функции манипулирования данными ограничены в соответствии с предметной областью БД); 8) предусмотреть возможность сохранения резервной копии БД по требованию пользователя и автоматически.*

4. Содержание расчетно–пояснительной записки (перечень вопросов, которые подлежат разработке)

*Введение. 1. Постановка задачи. 1.1 Описание предметной области. 1.2* *Цели и задачи на проектирование 2. Проектирование.2.1 Проектирование схемы данных. 2.2* *Обоснование выбора технологии. 2.3 Описание среды реализации. 3. Программная реализация. 3.1 Физическая структура базы данных. 3.2 Структура информационной системы. 3.3 Тестирование системы. 3.4 Описание применения. Заключение. Список использованных источников. Приложение (листинг программы)*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*1. Структура информационной системы*

*2. Схема базы данных*

*3. Схема алгоритма работы пользователя/администратора*

*4. Диаграмма вариантов использования*

*5. Диаграмма взаимодействия*

6. Консультант по проекту *M.М. Борисик –––*

7. Дата выдачи задания  *–*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*раздел 1: 01.03.2022 (20 %); \_\_\_\_\_––––––––––––––––––––––––––––––––*

*раздел 2: 29.03.2022 (40 %); ––––––\_\_\_\_\_\_\_\_\_––––––––––––––––––––––*

*раздел 3: 26.04.2022 (30 %); \_\_\_–––––––––––––––––––––––––––––––––*

*оформление пояснительной записки и графического материала к 02.05 – 10 %*

РУКОВОДИТЕЛЬ*– М.М. Борисик*

(подпись)

Задание принял к исполнению *–––––––\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_––*

(дата и подпись студента)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc102769395)

[1 ПОСТАНОВКА ЗАДАЧИ 7](#_Toc102769396)

[1.1 Описание предметной области 7](#_Toc102769397)

[1.2 Цели и задачи на проектирование 9](#_Toc102769398)

[2 ПРОЕКТИРОВАНИЕ 11](#_Toc102769399)

[2.1 Проектирование схемы данных 11](#_Toc102769400)

[2.2 Обоснование выбора технологий 12](#_Toc102769401)

[2.3 Описание среды реализации 13](#_Toc102769402)

[3 ПРОЕКТИРОВАНИЕ 14](#_Toc102769403)

[3.1 Физическая структура базы данных 14](#_Toc102769404)

[3.2 Структура информационной системы 19](#_Toc102769405)

[3.3 Тестирование системы 20](#_Toc102769406)

[3.4 Описание применения 26](#_Toc102769407)

[ЗАКЛЮЧЕНИЕ 28](#_Toc102769408)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 29](#_Toc102769409)

[ПРИЛОЖЕНИЕ А 30](#_Toc102769410)

[ПРИЛОЖЕНИЕ Б 31](#_Toc102769411)

[ПРИЛОЖЕНИЕ В 32](#_Toc102769412)

[ПРИЛОЖЕНИЕ Г 33](#_Toc102769413)

[ПРИЛОЖЕНИЕ Д 34](#_Toc102769414)

# **ВВЕДЕНИЕ**

Скорость развития технологий в наш век рекордно большая для всей истории человечества. Мир движется к удобствам, экономии ресурсов, автоматизации и оптимизации всех процессов. В контексте информационных технологий во многом это удается благодаря разработке эффективных сервисов, которые направлены на облегчение работы людей, занимающихся рутинной и кропотливой деятельностью, которая, благодаря сервисам, теперь может выполняться практически без человеческого участия или контроля.

Подобные сервисы способствуют более эффективному распределению кадров: компаниям требуется меньше работников, чтобы обеспечивать деятельность предприятия, и, соответственно, появляется большая денежная выгода, благодаря которой можно также обеспечить более высокие зарплаты сотрудникам, повышая таким образом их мотивацию и продуктивность. Кроме того, если функции, выполняемые сервисами, занимали определенный процент обязанностей работника, то вместе с автоматизацией этих функций данный сотрудник получает возможность перенаправить свое время и усилия на более важные аспекты своей работы.

Автоматизация производства – это процесс в развитии машинного производства, при котором функции управления и контроля, ранее выполнявшиеся человеком, передаются приборам и автоматическим устройствам. Введение автоматизации на производстве позволяет значительно повысить производительность труда, обеспечить стабильное качество выпускаемой продукции, сократить долю рабочих, занятых в различных сферах производства [1].

До внедрения средств автоматизации замещение физического труда происходило посредством механизации основных и вспомогательных операций производственного процесса. Интеллектуальный труд долгое время оставался не механизированным (ручным). В настоящее время операции физического и интеллектуального труда, поддающиеся формализации, становятся объектом механизации и автоматизации [1].

Внедрение программы для автоматизации учета лекарств для сети аптек является технологией абсолютной необходимости. Обусловлено это многими факторами. Во–первых, подобный сервис обеспечивает разумное распределение ресурсов и кадров аптеки, при этом повышая эффективность и оперативность работы. Во–вторых, информация часто нуждается в своевременной коррекции, реализовать которую без помощи программного сервиса в разы дольше и сложнее. В–третьих, экономия времени в современном мире является приоритетной, особенно когда речь идет о таких процессах, как покупка и просмотр информации о различной продукции. Данная программная система позволит централизованно и структурированно хранить, и обрабатывать данные о лекарствах, а также удобно осуществлять поиск и сортировку даже в самых объемных базах данных.

# **ПОСТАНОВКА ЗАДАЧИ**

# **Описание предметной области**

Аптека – особая специализированная организация системы здравоохранения, занимающаяся изготовлением, фасовкой, анализом и продажей лекарственных средств. Аптеку традиционно рассматривают как учреждение здравоохранения, а её деятельность формулируют как «оказание фармацевтической помощи населению». Фармацевтическая помощь включает в себя процедуру консультирования врача и пациента с целью определения наиболее эффективного, безопасного и экономически оправданного курса лечения [2].

Аптека осуществляет следующие функции:

* основной задачей аптеки является обеспечение населения, учреждений здравоохранения и других учреждений, предприятий и организаций лекарственными средствами, разрешенными к медицинскому применению в странах СНГ, изделиями медицинского назначения и другими предметами аптечного ассортимента;
* приготовление, контроль и отпуск лекарственных средств населению и лечебно–профилактическим учреждениям по оформленным в соответствии с установленными правилами рецептам и требованиям;
* реализация населению готовых лекарственных средств, разрешенных к медицинскому применению на территории стран СНГ и подлежащих отпуску по рецептам и без рецептов врача, а также изделий медицинского назначения и других предметов аптечного ассортимента;
* обеспечение прикрепленных на мед–обеспечение лечебно–профилактических и других учреждений, предприятий и организаций по оформленным в соответствии с установленными правилами требованиям лекарственными средствами и изделиями медицинского назначения. Прикрепление к аптеке конкретных учреждений, предприятий и организаций осуществляется вышестоящей организацией в порядке подчиненности;
* организация заготовки лекарственного растительного сырья;
* распространение среди населения санитарно–гигиенических знаний и проведение санпросвет работы;
* информация медицинских работников об имеющихся в аптеке и временно отсутствующих лекарственных средствах, поступлении новых лекарственных препаратов и способах их применения [3].

Аптека состоит из нескольких помещений: торговый зал, склад, бухгалтерия, включая кабинет директора. Аптека взаимодействует с поставщиками в лице менеджера, осуществляющими заказы на поставку лекарств, бытовой химии, медицинского оборудования и косметических средств. Каждый препарат имеет свой состав, наименование, артикул и показание к применению. Так же в аптеке есть свой штат сотрудников: фармацевты, бухгалтера, грузчики–водители, менеджеры. Клиент, приходя в аптеку, взаимодействует с фармацевтом, который, в свою очередь, осуществляет продажу. Операции с денежными средствами и отбивку чека осуществляет так же фармацевт. При покупке товара клиент получает чек с указанием товара и его стоимости. В аптеке ведется документация: кассовая книга (сумма проданного товара, которая отдается бухгалтеру, а затем передается в налоговую для отчетности), приходная накладная, ведется учет доходов и расходов аптеки.

Основным аналогом данной информационной системы является компьютерная программа «1С для аптек».

Преимуществами данного аналога являются:

* надёжность работы: за годы работы программа получила все необходимые обновления, которые оптимизировали стабильность этого ПО для аптек;
* бесплатная поддержка 6 месяцев – при приобретении лицензионного программного обеспечении для аптек «1С», вы получаете полгода бесплатных консультаций специалистов фирмы «1С»;
* большое число специалистов - вы легко найдёте программиста «1С», который за обговоренную плату сделает необходимую оптимизацию.

К недостаткам компьютерной программы можно отнести:

* необходимость доработки: программа не оптимизирована полностью под все необходимые для аптек функции;
* дороговизна;
* сложность понимания – интерфейс и функционал программы не простой, не всему персоналу может быть легко понятна работа в программе;
* справочник наименований (номенклатура) – вносится вручную: rнесколько рабочих дней вам будет необходимо руками набивать начальный ассортимент тысяч медпрепаратов.

Таким образом, проанализировав аналоги, можно прийти к выводу, что необходим продукт, который будет прост в использовании, иметь привлекательный дизайн и будет способен автоматизировать процесс управления лекарствами.

# **Цели и задачи на проектирование**

Целью курсовой работы является разработка программного продукта, предназначенного для автоматизации учета лекарств для аптек, а также систематизация и углубление знаний и навыков, выработанных в процессе изучения дисциплины «Управление информационными проектами».

Программный продукт должен быть интуитивным в использовании, понятным и удобным для рядовых пользователей. Все основные действия с базой должны осуществляться путем взаимодействий с интерфейсом приложения, вследствие чего вся сложность реализации будет скрыта от пользователя.

Программа должна осуществлять запросы на чтение, изменение, добавление или удаление данных, хранящихся в БД, в зависимости от соответствующих действий пользователя.

Для достижения поставленных целей необходимо решить следующие задачи:

* проанализировать предметную область;
* разработать структурную схему программы;
* разработать алгоритм работы программного продукта;
* разработать схему базы данных;
* разработать интерфейс пользователя;
* провести тестирование программного продукта.

В информационной системе функциональные требования предъявляются для фармацевта, для владельца аптеки и для администратора.

Функциональные требования для фармацевта:

* оформление заказа для покупателя;
* просмотр информации о лекарствах;
* просмотр информации об аптеке и её складах;
* просмотр истории заказов;
* просмотр отчётов.

Функциональные требования предъявляемые к владельцу аптеки:

* регистрация аптеки;
* регистрация и удаление фармацевтов в системе;
* создание и удаление складов аптек;
* отправка запросов поставщикам на получение лекарств;
* генерация и удаление отчётов.

Функциональные требования администратора:

* удаление аптеки;
* создание резервной копии базы данных.

# **ПРОЕКТИРОВАНИЕ**

# **Проектирование схемы данных**

В качестве концепции взаимодействия была выбрана модель клиент–сервер. Клиент–сервер – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер – это программное обеспечение [4]. Общение между клиентом и сервером происходит посредством HTTP протокола с использованием GET, POST, PUT, DELETE видами запросов.

Архитектура курсового проекта трехуровневая – это хорошо зарекомендовавшая себя архитектура программных приложений, которая объединяет приложения в три логических и физических вычислительных уровня: уровень представления или пользовательский интерфейс; уровень приложения, на котором обрабатываются данные; и уровень данных, где данные, связанные с приложением.

Архитектура веб-приложения построена с использованием паттернов MVC и Redux. Model–View–Controller – схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо. Redux представляет собой контейнер для управления состоянием приложения.

Курсовой проект будет соответствовать принципам REST – архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

Для описания функциональности разрабатываемой системы, обозначения рамок проекта и отображения пользовательских требований была разработана диаграмма вариантов использования (см. Приложение Г). В данной диаграмме представлено 3 вида пользователей: фармацевт, владелец аптеки и администратор. При входе в систему в первый раз, владелец аптеки должен зарегистрировать аптеку. После этапа регистрации, владелец может создать аккаунты для своих работников, сгенерировать отчёты по заказам, изменять информацию об аптеке и её складах, отправлять запросы на получение лекарств поставщикам, а также всё то, что может фармацевт. Фармацевт – вид пользователя, который создаёт владелец аптеки. Фармацевт может оформить заказ для покупателя, просмотреть историю заказов, информацию об аптеке, складах, лекарствах, отчётов. Администратор – пользователи, которые могут быть назначены только напрямую через базу данных. Администраторы имеют доступ к удалению аптеки, а также созданию резервного копирования базы данных.

Алгоритм работы пользователя описан в приложении В.

# **Обоснование выбора технологий**

Для написания курсового проекта для написания серверной части был выбран язык программирования Java с использованием Spring Boot. Java – строго типизированный объектно–ориентированный язык программирования общего назначения. Преимуществами языка являются:

* Использование в корпоративных приложениях, Java способен поддержать строительные блоки системы или различные библиотеки, с их помощью создают необходимые функции;
* Запуск приложений в «песочнице» с устранением распространённых, уязвимых объектов в соответствии с политикой безопасности;
* Независимое положение от платформ;
* Повышенная производительность труда благодаря встроенному механизму, чтобы совместно использовать данные программы на нескольких компьютерах;
* Возможность автоматического управления памятью с одновременным независимым запуском потоков.

Преимуществами Spring Boot являются:

* Быстрая и легкая разработка приложений на основе Spring.
* Автоконфигурация всех компонентов для приложения Spring производственного уровня.
* Готовые встроенные серверы, обеспечивающие ускоренное и более продуктивное развертывание приложений.
* HTTP end–points, позволяющие вводить внутренние функции приложения, такие как показатели, состояние здоровья и другие.

В качестве языка, отвечающего за front–end сайта, были выбраны HTML, CSS, JS с использованием React Framework. Плюсы использования React:

* Виртуальная объектная модель документа;
* Повторное применение компонентов;
* Нисходящий поток данных.

Для работы с базами данных была выбрана СУБД PostgreSQL – свободная объектно–реляционная система управления базами данных. СУБД отличается высокой надёжностью и хорошей производительностью. PostgreSQL поддерживает транзакции (ACID), репликация реализована встроенными механизмами. При этом система расширяемая — можно создавать свои типы данных и индексов, а также расширять поведение при помощи языков программирования.

Серверная часть для взаимодействия с базой данных использует Hibernate – библиотека для языка программирования Java, предназначенная для решения задач объектно–реляционного отображения (ORM), самая популярная реализация спецификации JPA. Библиотека не только решает задачу связи классов Java с таблицами базы данных (и типов данных Java с типами данных SQL), но и также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL– и JDBC–кода. Hibernate автоматизирует генерацию SQL–запросов и освобождает разработчика от ручной обработки результирующего набора данных и преобразования объектов, максимально облегчая перенос (портирование) приложения на любые базы данных SQL.

# **Описание среды реализации**

Для написания серверной и клиентской частей приложения будет использоваться IntelliJ IDEA – интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, разработанная компанией JetBrains с использованием Java версии 17.

Для взаимодействия с PostgreSQL будет использоваться pgAdmin 4 – бесплатная программа с графическим интерфейсом для разработки баз данных PostgreSQL.

1. **ПРОЕКТИРОВАНИЕ**
2. **Физическая структура базы данных**

Исходя из предметной области приложения, были выделены следующие сущности:

1. Пользователи (users);
2. Лекарства (drugs);
3. Аптеки (pharmacies);
4. Заказы (orders);
5. Чеки (receipts);
6. Отчёты (reports);
7. Поставщики (vendors);
8. Склады (storages);
9. Роли (roles);
10. Права доступа (authorities).

В сущности «users» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о зарегистрированных аккаунтов в системе, представлено в таблице 3.1.

Таблица 3.1 – Описание отношения «users»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| email | Обязательное | character varying (255) |
| encrypted\_password | Обязательное | character varying (64) |
| name | Обязательное | character varying (64) |
| surname | Обязательное | character varying (64) |
| address | Необязательное | character varying (512) |
| is\_deleted | Обязательное | boolean |

В сущности «drugs» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о лекарствах, представлено в таблице 3.2.

Таблица 3.2 – Описание отношения «drugs»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| name | Обязательное | character varying (255) |
| description | Необязательное | character varying (1024) |
| quantity | Обязательное | integer |
| price | Обязательное | numeric (18,2) |
| image | Обязательное | bytea |
| storage\_id | Обязательное (внешний ключ) | bigint |
| vendor\_id | Обязательное (внешний ключ) | bigint |
| is\_deleted | Обязательное | boolean |

В сущности «pharmacies» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о зарегистрированных аптеках в системе, представлено в таблице 3.3.

Таблица 3.3 – Описание отношения «pharmacies»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| name | Обязательное | character varying (255) |
| is\_deleted | Обязательное | boolean |

В сущности «orders» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о заказах, представлено в таблице 3.4.

Таблица 3.4 – Описание отношения «orders»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| quantity | Обязательное | integer |
| drug\_id | Обязательное (внешний ключ) | bigint |
| receipt\_id | Обязательное (внешний ключ) | bigint |
| is\_deleted | Обязательное | boolean |

В сущности «receipts» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о лекарствах, представлено в таблице 3.5.

Таблица 3.5 – Описание отношения «receipts»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| grand\_total | Обязательное | numeric (18,2) |
| date\_of\_purchase | Обязательное | date |
| solder\_id | Обязательное (внешний ключ) | bigint |
| is\_deleted | Обязательное | boolean |

В сущности «reports» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации об отчётах, представлено в таблице 3.6.

Таблица 3.6 – Описание отношения «reports»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| name | Обязательное | character varying (255) |
| start\_date | Обязательное | date |
| end\_date | Обязательное | date |
| is\_deleted | Обязательное | boolean |

В сущности «vendors» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о поставщиках, представлено в таблице 3.7.

Таблица 3.7 – Описание отношения «vendors»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| name | Обязательное | character varying (64) |
| address | Необязательное | character varying (512) |
| is\_deleted | Обязательное | boolean |

В сущности «storages» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о складах аптек, представлено в таблице 3.8.

Таблица 3.8 – Описание отношения «storages»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| pharmacy\_id | Обязательное | bigint |
| pharmacy\_address | Обязательное | character varying (512) |
| is\_deleted | Обязательное | boolean |

В сущности «roles» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о ролях пользователей, представлено в таблице 3.9.

Таблица 3.9 – Описание отношения «roles»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| role | Обязательное | character varying (255) |

В сущности «authorities» первичным ключом является атрибут «id». Описание сущности, предназначенной для хранения информации о правах доступа ролей, представлено в таблице 3.10.

Таблица 3.10 – Описание отношения «authorities»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| id | Обязательное (ключевое поле) | bigint |
| authority | Обязательное | character varying (255) |

Пользователь может иметь несколько ролей, как и роль может принадлежать различным пользователям. Из этого следует, что необходимо создать дополнительную сущность, которая будет обозначать связь многие ко многим. В сущности «m2m\_users\_roles» первичными ключами являются поля «user\_id» и «role\_id». Описание сущности представлено в таблице 3.11.

Таблица 3.11 – Описание отношения «m2m\_users\_roles»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| user\_id | Обязательное (ключевое поле) | bigint |
| role\_id | Обязательное (ключевое поле) | bigint |

Роль может иметь несколько прав доступа, как и права доступа могут принадлежать различным ролям. Из этого следует, что необходимо создать дополнительную сущность, которая будет обозначать связь многие ко многим. В сущности «m2m\_roles\_authorities» первичными ключами являются поля «authority\_id» и «role\_id». Описание сущности представлено в таблице 3.12.

Таблица 3.12 – Описание отношения «m2m\_roles\_authorities»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| authority\_id | Обязательное (ключевое поле) | bigint |
| role\_id | Обязательное (ключевое поле) | bigint |

Отчёт может иметь несколько чеков, как и чеки могут принадлежать различным отчётам. Из этого следует, что необходимо создать дополнительную сущность, которая будет обозначать связь многие ко многим. В сущности «m2m\_reports\_receipts» первичными ключами являются поля «report\_id» и «receipt\_id». Описание сущности представлено в таблице 3.13.

Таблица 3.13 – Описание отношения «m2m\_reports\_receipts»

|  |  |  |
| --- | --- | --- |
| Название атрибута | Заполнение | Тип данных |
| report\_id | Обязательное (ключевое поле) | bigint |
| receipt\_id | Обязательное (ключевое поле) | bigint |

Схема базы данных приложения описана в приложении Б.

1. **Структура информационной системы**

В качестве концепции взаимодействия была выбрана модель клиент–сервер. Клиент–сервер – вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер – это программное обеспечение [4]. Общение между клиентом и сервером происходит посредством HTTP протокола с использованием GET, POST, PUT, DELETE видами запросов.

Архитектура курсового проекта трехуровневая – это хорошо зарекомендовавшая себя архитектура программных приложений, которая объединяет приложения в три логических и физических вычислительных уровня: уровень представления или пользовательский интерфейс; уровень приложения, на котором обрабатываются данные; и уровень данных, где данные, связанные с приложением.

Архитектура веб-приложения построена с использованием паттернов MVC и Redux. Model–View–Controller – схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо. Redux представляет собой контейнер для управления состоянием приложения.

Взаимодействие между модулями системы происходит следующим образом:

1. Клиент переходит на необходимую ему веб-страницу.
2. Веб-браузер генерирует HTTP запрос на сервер.
3. Соответствующий контроллер принимает HTTP-запрос с параметрами, передаваемыми веб-браузером.
4. Контроллер вызывает необходимые методы сервисов.
5. Сервисы взаимодействуют с базой данных, получают необходимые данные, обрабатывают их и возвращают в контроллер.
6. Контроллер возвращает веб-браузеру необходимые данные.
7. Веб-браузер рендерит страницу, после чего клиент может продолжать работать с веб-приложением.

Структуру информационной системы можно посмотреть в приложении А.

1. **Тестирование системы**

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Для тестирования данного приложения используются тест-кейсы. Тест-кейс — набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определённой цели или тестового условия, таких как выполнения определённого пути программы или же для проверки соответствия определённому требованию.

В таблицах 3.14–3.18 приведены примеры тестов по тестированию программного средства по каждому из модулей.

Таблица 3.14 – Тестирование приложения, модуль «Регистрация»

|  |  |  |  |
| --- | --- | --- | --- |
| Название тест-кейса | Описание | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Регистрация владельца аптеки | 1. Заполнить поля формы «Name», «Surname», «Email», «Password», «Pharmacy Name»  2. Нажать кнопку «Register» | Переход на главную страницу с указанием имени владельца аптеки в шапке профиля | Выполнено |
| Регистрация с пустым полем «Email» | 1. Заполнить поля формы «Name», «Surname», «Email», «Password», «Pharmacy Name»  2. Нажать кнопку «Register» | Сообщение о пропущенном поле «Email» | Выполнено |
| Регистрация с пустым полем «Password» | 1. Заполнить поля формы «Name», «Surname», «Email», «Password», «Pharmacy Name»  2. Нажать кнопку «Register» | Сообщение о пропущенном поле «Password» | Выполнено |
| Регистрация владельца аптеки с существующим «Email» в системе | 1. Заполнить поля формы «Name», «Surname», «Email», «Password», «Pharmacy Name»  2. Нажать кнопку «Register» | Сообщение о том, что пользователь с таким «Email» уже зарегистрирован в системе | Выполнено |

Таблица 3.15 – Тестирование приложения, модуль «Авторизация»

|  |  |  |  |
| --- | --- | --- | --- |
| Название тест-кейса | Описание | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Авторизация пользователя | 1. Заполнить поля формы «Email», «Password»  2. Нажать кнопку «Login» | Переход на главную страницу с указанием имени пользователя в шапке профиля | Выполнено |
| Авторизация с пустым полем «Email» | 1. Заполнить поля формы «Password»  2. Нажать кнопку «Login» | Сообщение о пропущенном поле «Email» | Выполнено |
| Авторизация с пустым полем «Password» | 1. Заполнить поля формы «Email»  2. Нажать кнопку «Login» | Сообщение о пропущенном поле «Password» | Выполнено |
| Ввод неверных данных при авторизации | 1. Заполнить поля формы «Email», «Password»  2. Нажать кнопку «Register» | Сообщение о неверных данных при входе | Выполнено |

Таблица 3.16 – Тестирование приложения, модуль «Фармацевт»

|  |  |  |  |
| --- | --- | --- | --- |
| Название тест-кейса | Описание | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Оформление заказа | 1. Перейти по ссылке «Set Order»  2. Заполнить информацию о покупаемых лекарствах  3. Нажать на кнопку «Order» | Заказ оформлен | Выполнено |
| Просмотр информации о лекарствах | 1. Перейти по ссылке «Drugs» | Отображается страница с лекарствами | Выполнено |
| Просмотр информации об аптеке | 1. Перейти по ссылке «Pharmacy» | Отображается страница с информацией об аптеке | Выполнено |
| Просмотр информации о складах | 1. Перейти по ссылке «Storages» | Отображается страница с информацией о складах | Выполнено |
| Просмотр истории заказов | 1. Перейти по ссылке «Receipts» | Отображается страница со всеми заказами клиентов аптеки | Выполнено |
| Просмотр отчётов | 1. Перейти по ссылке «Reports» | Отображается страница со всеми созданными отчётами | Выполнено |

Таблица 3.17 – Тестирование приложения, модуль «Владелец аптеки»

|  |  |  |  |
| --- | --- | --- | --- |
| Название тест-кейса | Описание | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Регистрация фармацевта в системе | 1. Перейти по ссылке «Pharmacists»  2. Нажать на кнопку «Add Pharmacist»  3. Заполнить информацию о фармацевте | Фармацевт зарегистрирован | Выполнено |
| Удаление фармацевта из системы | 1. Перейти по ссылке «Pharmacists»  2. Напротив фармацевта, которого необходимо удалить, нажмите кнопку «Delete» | Фармацевт удалён | Выполнено |
| Добавление склада | 1. Перейти по ссылке «Storages»  2. Нажать на кнопку «Add Storage»  3. Заполнить информацию о складе | Склад создан | Выполнено |
| Удаление пустого склада | 1. Перейти по ссылке «Storages»  2. Напротив пустого склада, который необходимо удалить, нажмите кнопку «Delete» | Склад удалён | Выполнено |
| Удаление склада, в котором есть лекарства | 1. Перейти по ссылке «Storages»  2. Напротив пустого склада, который необходимо удалить, нажмите кнопку «Delete» | Сообщение об ошибке удаления из-за нахождения лекарств на складе | Выполнено |
| Генерация отчёта | 1. Перейти по ссылке «Reports»  2. Нажмите на кнопку «Generate Report»  3. Выберите начало и конец отчёта и дайте отчёту имя | Отчёт создан | Выполнено |
| Генерация отчёта без указанных дат | 1. Перейти по ссылке «Reports»  2. Нажмите на кнопку «Generate Report»  3. Введите отчёту имя | Сообщение об ошибки из-за отсутствия начала и конца |  |
| Удаление отчёта | 1. Перейти по ссылке «Reports»  2. Напротив отчёта, который необходимо удалить, нажмите кнопку «Delete» | Отчёт удален | Выполнено |

Таблица 3.18 – Тестирование приложения, модуль «Администратор»

|  |  |  |  |
| --- | --- | --- | --- |
| Название тест-кейса | Описание | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Удаление аптеки | 1. Перейти по ссылке «Pharmacies»  2. Напротив аптеки, которую необходимо удалить, нажмите кнопку «Delete»  3. Подтвердить удаление | Аптека удалена | Выполнено |
| Резервное копирование базы данных | 1. Нажать на кнопку «Save Database State» | База данных скачивается на устройство администратора | Выполнено |

Проведенное тестирование программного средства показало, что все функциональные требования были выполнены в полном объеме.

На рисунках 3.1–3.3 приведены примеры отображения ошибок.

Graphical user interface, application

Description automatically generated

Рисунок 3.1 – не валидный «Email»

Graphical user interface

Description automatically generated with low confidence

Рисунок 3.2 – Неверные данные для входа

Graphical user interface, text, application

Description automatically generated

Рисунок 3.3 – Недостаточно символов

1. **Описание применения**

Курсовой проект представляет собой веб-приложение с тремя различными ролями: фармацевт, владелец аптеки и администратор. Предназначен для автоматизации учета лекарств в аптеках.

При входе в систему, пользователю необходимо авторизоваться.

Авторизовавшись как фармацевт, клиент пересылает на «Home» страницу, где у него есть доступ к таким ссылкам как: «Drugs», «Pharmacy», «Storages», «Receipts», «Reports», «Set Order» и «Logout». Все ссылки, кроме «Set Order» и «Logout», позволяют только просматривать информацию, изменять её может только владелец аптеки.

Авторизовавшись в систему как владелец аптеки, пользователя пересылает на «Home» страницу, где у него есть доступ к таким ссылкам как: «Drugs», «Pharmacy», «Pharmacists», «Storages», «Receipts», «Reports», «Set Order» и «Logout», где, в отличие от клиента, у него есть доступ к изменению всех данных.

Авторизовавшись как администратор, пользователя пересылает на «Home» страницу, где у него есть доступ к таким ссылкам как: «Pharmacy» и «Save Database State». Администратор не имеет доступа к аптекам клиентов через UI.

Для входа в роли фармацевта можно использовать данные:

* Логин: [pharmacist@gmail.com](mailto:pharmacist@gmail.com)
* Пароль: pharmacist

Для входа в роли владельца аптеки можно использовать данные:

* Логин: pharmacy\_owner[@gmail.](mailto:testspec2@mail.)com
* Пароль: pharmacy\_owner

Для входа в роли администратора можно использовать данные:

* Логин: admin[@gmail.](mailto:testspec2@mail.)com
* Пароль: admin

# **ЗАКЛЮЧЕНИЕ**

В результате курсового проектирования был разработан программный продукт «Автоматизация учета лекарств для сети аптек» на языке Java и JavaScript с использованием технологий Spring и React.

Проект разработан в полном объеме и полностью соответствует поставленной цели. В качестве возможного направления развития разработанного программного модуля можно рассматривать перевод в формат мобильного приложения и, если понадобится, паттернов проектирования для обеспечения намного лучшей масштабируемости и облегчения поддержки другими программистами данного программного продукта длительное время.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Автоматизация производства – Википедия [Электронный ресурс]. – Режим доступа – <https://ru.wikipedia.org/wiki/Автоматизация_производства> .
2. Аптека – Википедия [Электронный ресурс]. – Режим доступа – <https://ru.wikipedia.org/wiki/Аптека> .
3. Типовое положение об аптеке [Электронный ресурс]. – Режим доступа – [https://www.vsavm.by/veterinarnyj–nadzor/organizaciya–i–ekonomika–veterinarnoj–farmacii/tipovoe–polozhenie–o–xozraschetnoj–apteke/](https://www.vsavm.by/veterinarnyj-nadzor/organizaciya-i-ekonomika-veterinarnoj-farmacii/tipovoe-polozhenie-o-xozraschetnoj-apteke/) .
4. Клиент–сервер – Википедия [Электронный ресурс]. – Режим доступа – [https://ru.wikipedia.org/wiki/Клиент\_–\_сервер](https://ru.wikipedia.org/wiki/Клиент_—_сервер) .

# **ПРИЛОЖЕНИЕ А**

(Обязательное)

Структура информационной системы

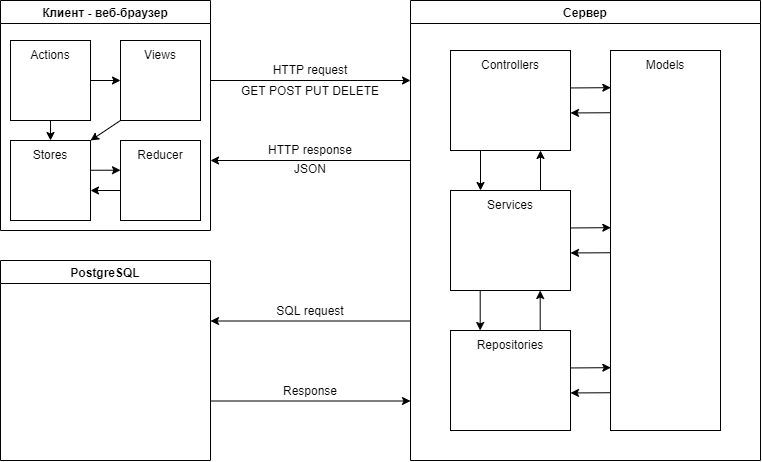


Рисунок А.1 – Структура информационной системы

# **ПРИЛОЖЕНИЕ Б**

(Обязательное)

Схема базы данных

Diagram, schematic

Description automatically generated

Рисунок Б.1 – Схема базы данных

# **ПРИЛОЖЕНИЕ В**

(Обязательное)

Схема алгоритма работы пользователя

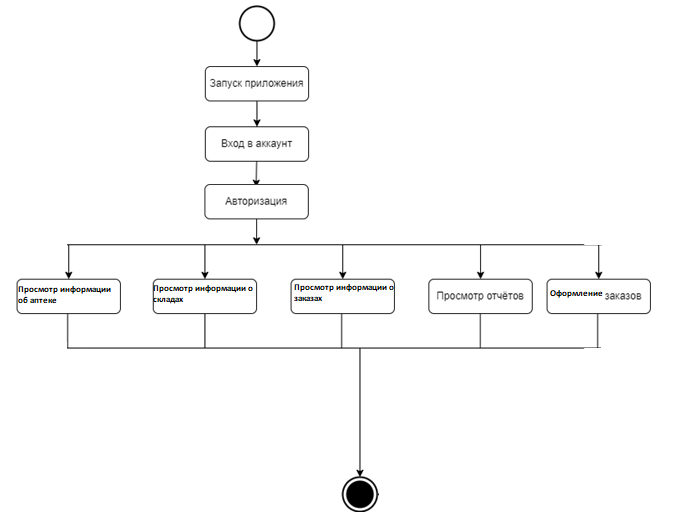


Рисунок В.1 – Схема алгоритма работы пользователя

# **ПРИЛОЖЕНИЕ Г**

(Обязательное)

Диаграмма вариантов использования

Diagram

Description automatically generated

Рисунок Г.1 – Диаграмма вариантов использования

# **ПРИЛОЖЕНИЕ Д**

(Обязательное)

Диаграмма взаимодействия

Diagram

Description automatically generated

Рисунок Д.1 – Диаграмма взаимодействия для варианта использования оформления заказа