**Оглавление**

[**Освоение методики документирования ПС** 3](#_Toc133579620)

[**Анализ предметной области** 3](#_Toc133579621)

[**Анализ существующих программных разработок** 10](#_Toc133579622)

[**ТЗ на создание программного продукта** 25](#_Toc133579623)

[**Проектирование функциональной структуры программного продукта** 29](#_Toc133579624)

[**Проектирование функциональной структуры программного продукта** 33](#_Toc133579625)

[**Проектирование Базы данных программной системы** 41](#_Toc133579626)

[**Разработки программного обеспечения ПС** 51](#_Toc133579627)

[**Тестирование программной системы** 56](#_Toc133579628)

# **Освоение методики документирования ПС**

## **Анализ предметной области**

Объект информатизации: городская поликлиника «Спасибо, Боже!»

Была построена в центре города Воронеж в 2020 году. Двухэтажное здание содержит в себе: регистратуру, дневной стационар, отделение платных услуг, несколько уборных, 60 кабинетов, из которых 44 – лечебно-профилактические, 6 – используемые администрацией поликлиники, 2 – процедурных, 1 – рентгенодиагностических, 1 – учета и мед статистики, 2 – кабинет доврачебного приема, 3 – административно-хозяйственной части, 1 – аптечный пункт. Основной поток пациентов в день в среднем составляет около 978 человек. Объем годовых продаж составляет от 1210 до 2750 препаратов, 7 млн.руб.

Основные отделения: терапевтическое, хирургическое, рентгенологическое, женской консультации, профилактики, дневной стационар, диагностическое.

Состав мед. персонала: главный врач, 3 замов глав. врача, 126 врача, 3 лаборанта, 4 мед-регистраторов, 3 бухгалтера, 7 санитаров, 62 медсестры, 5 работников отдела лекарственного обеспечения и фармацевтической деятельности, 1 главная сестра.

Система закупок делится следующим образом: ЖНЛВП (жизненно необходимые и важнейшие лекарственные препараты) приобретаются в экстренном порядке, временные – в определенный промежуток, зависящий от срока годности препаратов / даты поставки поставщиком или утвержденной главным врачом, офсетные контракты – согласованно поставленным условиям. Также присутствует льготная категория, которая предоставляет скидку (в случае покупки пациентом за свой счет), изменение срока получения препарата. Поликлиника поддерживает связь с рядом поставщиков, закупки препаратов у которых оплачиваются либо государством, либо частным лицом.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 1.1 – Бизнес-направления

Для информатизации выбирается процесс обработки заказа препаратов, запрошенных пациентом на основе рецепта врача, проводившего осмотр в данной поликлинике, подсчета конечной стоимости, оплачиваемой государством или больным, фиксацией продажи и выдачи в журнале отчета.

К действующим лицам данного бизнес-процесса относятся:

* Работник отдела лекарственного обеспечения и фармацевтической деятельности: оформляет продажу препарата пациенту с учетом данных о медикаменте, личных данных клиента и врача, выписавшего рецепт, и формирует накладную;
* Пациент: обращается с заявкой на покупку препарата и предоставляет личные данные;

На Рисунке 1.2 представлена модель «Черный ящик».



Рисунок 1.2 – Черный ящик

Рисунок 1.3 иллюстрирует декомпозицию бизнес-процесса

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.3 – Декомпозиция бизнес-процесса

Задачи:

* Задача ввода данных о пациенте и рецепте;
* Задача формирования заявки;
* Задача сохранения данных в базе;
* Задача записи о продаже в журнал учета;
* Задача формирования накладной;

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

Рисунок 1.4 – Модель вариантов использования UML для бизнес-процесса

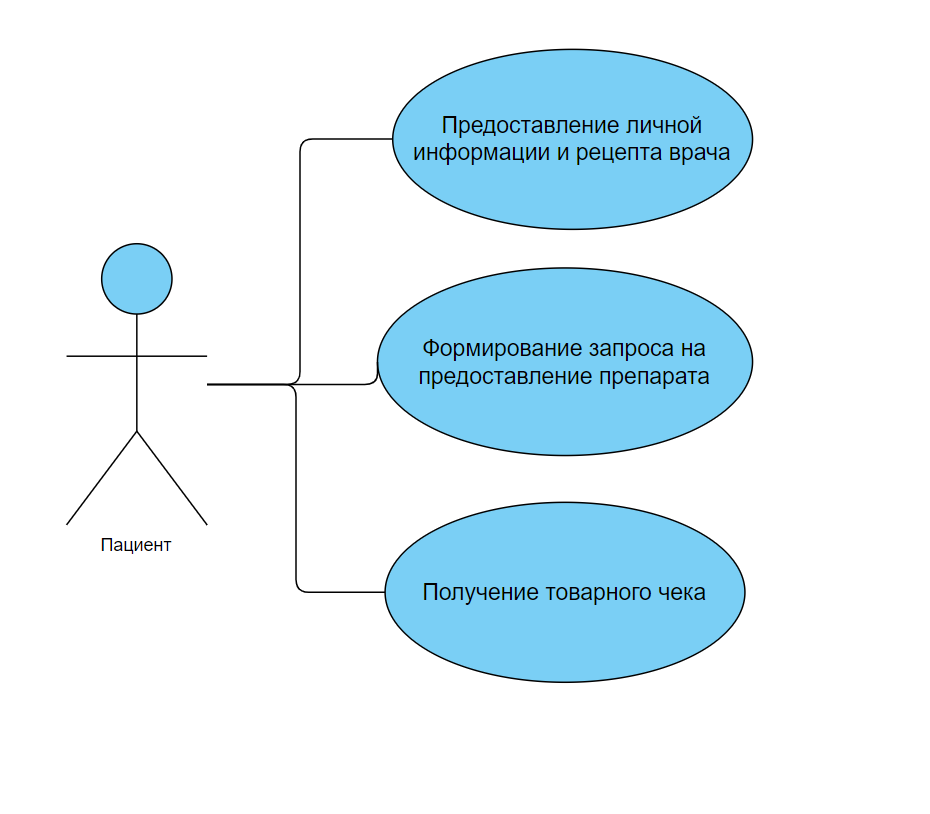


Рисунок 1.5 – Модель вариантов использования UML для бизнес-процесса

Таблица 1 – Реестр входных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование и назначение потока | Форма представления | Обработчик (кто обрабатывает) | Корреспондент (Откуда) | Характеристики обработки | |
| Трудозатраты, чел.ч | Периодичность, регламент |
| 1 | Данные о пациенте и рецепте | Документ | Работник отдела лекарственного обеспечения и фармацевтической деятельности | Пациент | 0,16 | По запросу |
| 2 | Информация об имеющемся препарате | Электронная | База данных | Работник отдела лекарственного обеспечения и фармацевтической деятельности | 0,06 | По запросу |
| 3 | Заявка на покупку | Электронная | Поставщик | База данных | 0,11 | По запросу |

Таблица 2 – Реестр выходных информационных потоков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование и назначение потока | Форма представления | Обработчик (кто обрабатывает) | Корреспондент (куда) | Характеристики обработки | |
| Трудозатраты, чел.ч | Периодичность, регламент |
| 1 | Товарная накладная | Документ | работник отдела лекарственного обеспечения и фармацевтической деятельности | Пациент | 0,41 | По запросу |
| 2 | Запись о проведенной продаже в журнал учета | Документ | Работник отдела лекарственного обеспечения и фармацевтической деятельности | Журнал | 0,1 | По запросу |

К правилам обработки информации и возможным ограничениям относятся:

* Данные пациента идентифицируются как уникальная запись в базе;
* Для начала оформления продажи товаров по заказу работнику отдела лекарственного обеспечения и фармацевтической деятельности необходим идентификатор заказа, предоставляемый пациентом;

К нормативно-справочной документации, регламентирующей бизнес-процесс, относятся:

* Инструкция по организации хранения в аптечных учреждениях различных групп ЛС и изделий медицинского назначения;
* Нормы СанПиН. Например, 2.1.3.2630-10 «Санитарно-эпидемиологические требования к организациям, осуществляющим медицинскую деятельность»;
* Приказы МЗ РФ. Например, «Об утверждении правил надлежащей аптечной практики лекарственных препаратов для медицинского применения»;
* Гражданский кодекс РФ;
* Договор поставки;

## **Анализ существующих программных разработок**

**1С:Розница:** универсальное решение для управления розничной торговлей одиночных магазинов и розничных сетей. Автоматизирует все бизнес-процессы магазина, в том числе оформление чеков в специализированном рабочем месте кассира и торговлю через Интернет.

 Возможности ПО "1С:Розница" включают:

* работу с нормативно-правовой информацией;
* управление программами лояльности;
* анализ спроса и ликвидности товаров, управление ценообразованием;
* учет товаров на складах;
* контроль запасов и управление закупками;
* управление персоналом;
* рабочее место кассира;
* детальную аналитику;
* обмен данными с "1С:Управление торговлей";
* поддержку и подключение оборудования.

На Рисунке 2.1 представлена описанная сверху информация в виде схемы, а на Рисунке 2.2 и Рисунке 2.3 – внешний вид программы.

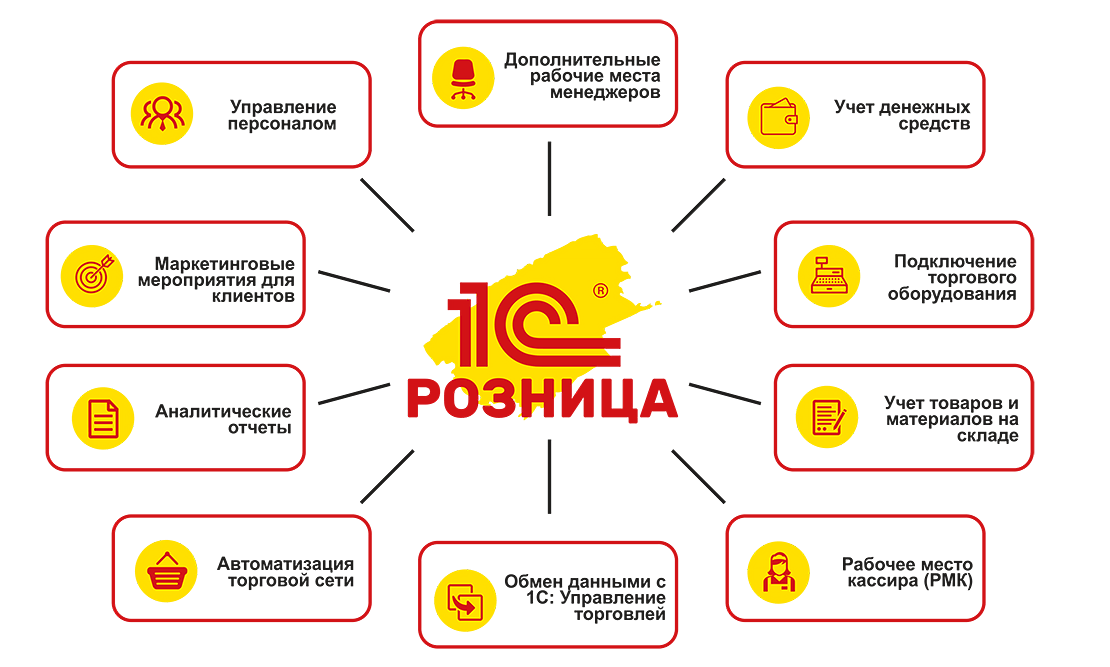


Рисунок 2.1 – Функциональные возможности "1С:Розница"

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.2 – Интерфейс "1С:Розница"-1

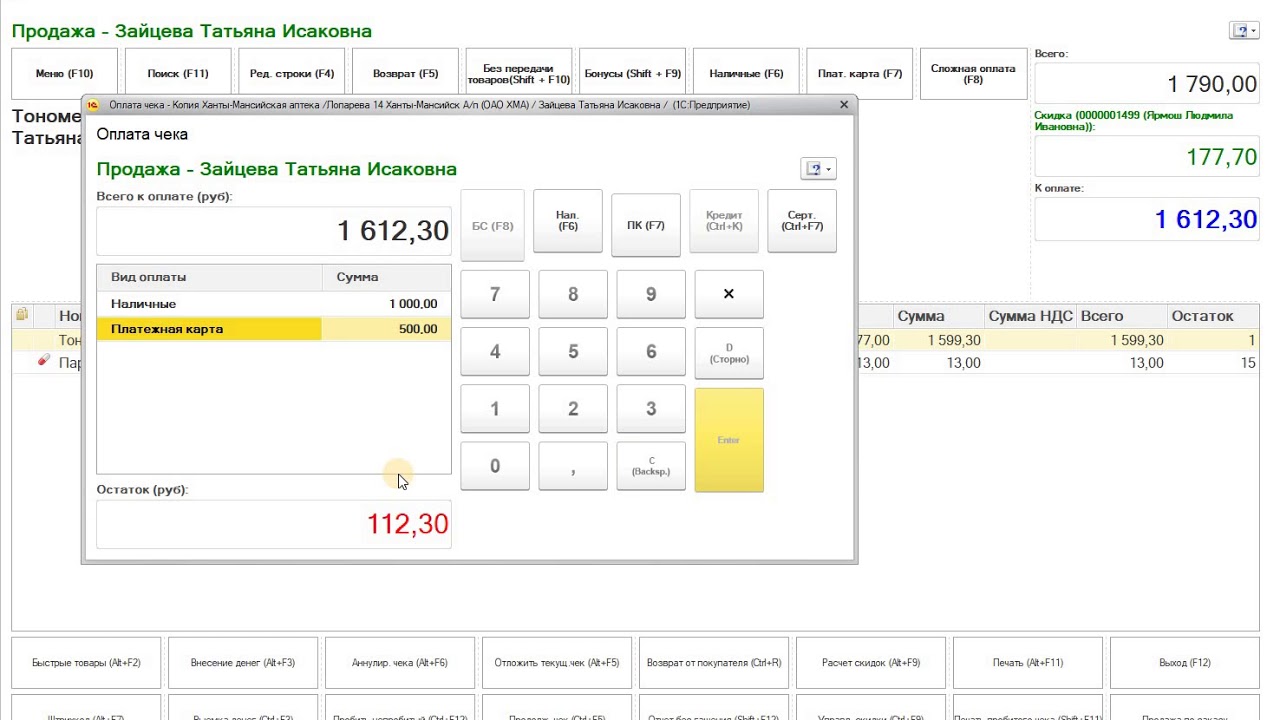


Рисунок 2.3 – Интерфейс "1С:Розница"-2

С помощью указанного продукта не составит труда:

* Управлять ценовой политикой и формировать ассортиментный ряд. Проведя анализ покупательских потребностей и ликвидности товарных позиций, можно определить оптимальный стоимостной диапазон и выставить наиболее привлекательные предложения. Разумеется, опираясь на расположение точки, а также ЦА.
* Обрабатывать нормативную и справочную документацию. Пользоваться информацией, предоставляемой российскими классификаторами.
* Задействовать комплекс маркетинговых мероприятий для создания продолжительных связей с клиентами (программа лояльности). Условия начисления бонусов, поддерживаемые продуктом, способствуют развитию повторных связей.
* Вести учет запаса товаров на складе. Отслеживать позиции с истекающим сроком годности.
* Осуществлять контроль по закупкам, проверять остатки. По результатам мониторинга составлять заявки поставщикам. Здесь же можно следить за изменением стоимости, времени доставки, наличием сопроводительной документации.
* Проводить настройку рабочего места сотрудника-кассира. ПО позволяет определить объем персональных продаж.
* Систематизировать данные по персоналу, планировать занятость людей, проводить анализ эффективности их работы.
* Собирать аналитические и статистические сведения, необходимые для функционирования компании.

Данный программный комплекс используется для управления продажами в разных сферах торговли и бизнеса любого масштаба. Имеет понятный интерфейс, удобный функционал, дает возможность эффективно управлять ассортиментом, ценообразованием, действующими системами скидок. При помощи данного программного продукта можно с легкостью обрабатывать документацию, управлять ценовой политикой, бонусной системой, контролировать закупки товаров и их остатки на складах, а также проводить анализ эффективности работы сотрудников.

Пользователям доступна техподдержка и сопровождение. Кроме этого, к "1С:Розница" можно подключать дополнительные сервисы, а также осуществлять интеграцию с системой "1С:Бухгалтерия".

##### В 1С:Розница присутствует гибкая система оперативного управления, гибкость лицензирования, отчетов конфигурации, механизмов ценообразования. Программа отдает приоритет поставке продуктов с программной защитой. Все основные поставки, за исключением бюджетных, и дополнительные места поставляются с программной защитой. Однако файловые базы 1С являются наиболее уязвимые к физическому воздействию. Связано это с особенностями архитектуры такого типа баз – необходимостью держать открытыми (с полным доступом) все файлы конфигурации и самих файловых баз для всех пользователей операционной системы При наличии несанкционированного доступа к серверному оборудованию сотрудники компании или третьи лица могут использовать этот доступ для кражи или порчи информации.

В новом обновлении программы 1С:Розница появились не только новые функции программы, но и возможность изменения внешнего оформления интерфейса. Можно настроить автоматическое создание резервной копии.

К основным недостаткам относятся:

1. Дополнительный рабочий персонал, который в любом случае придется нанять;
2. В программе не работает расчет личных премий сотрудников;
3. Резервное копирование базы возможно только под ником администратора. Если у вас база стоит на кассовом компьютере, после завершения кассиром рабочей смены резервное копирование не выполняется, что очень неудобно;
4. Недоработанная функция расчета прибыли предприятия;
5. Сложное обновление (даже в автоматическом режиме) для клиента;
6. Руководства недостаточно для полноценного пользования системой. Нужно разбираться самостоятельно или прибегать к помощи специалистов;

**Cloudshop:** Программа позволяет автоматизировать работу магазина в части управления жизненным циклом товара и продаж. Программный продукт CloudShop от одноименной компании – онлайн-сервис предназначенный для автоматизации бизнес-процесса и учета розничной торговли.

Функции сервиса CloudShop:

* Автоматизация работы кассира с учётом требований 54-ФЗ;
* Учёт продаж;
* Складской учет: закупки, инвентаризации и списания;
* База товаров;
* Финансовый учёт;
* Поддержка дисконтных карт и база покупателей;
* Отчётность;

На Рисунке 2.4 представлена описанная сверху информация в виде схемы, а на Рисунке 2.5 и Рисунке 2.6 – внешний вид программы.



Рисунок 2.4 – Функциональные возможности "CloudShop"

Изображение выглядит как Веб-сайт

Автоматически созданное описание

Рисунок 2.5 – Интерфейс "CloudShop"-1

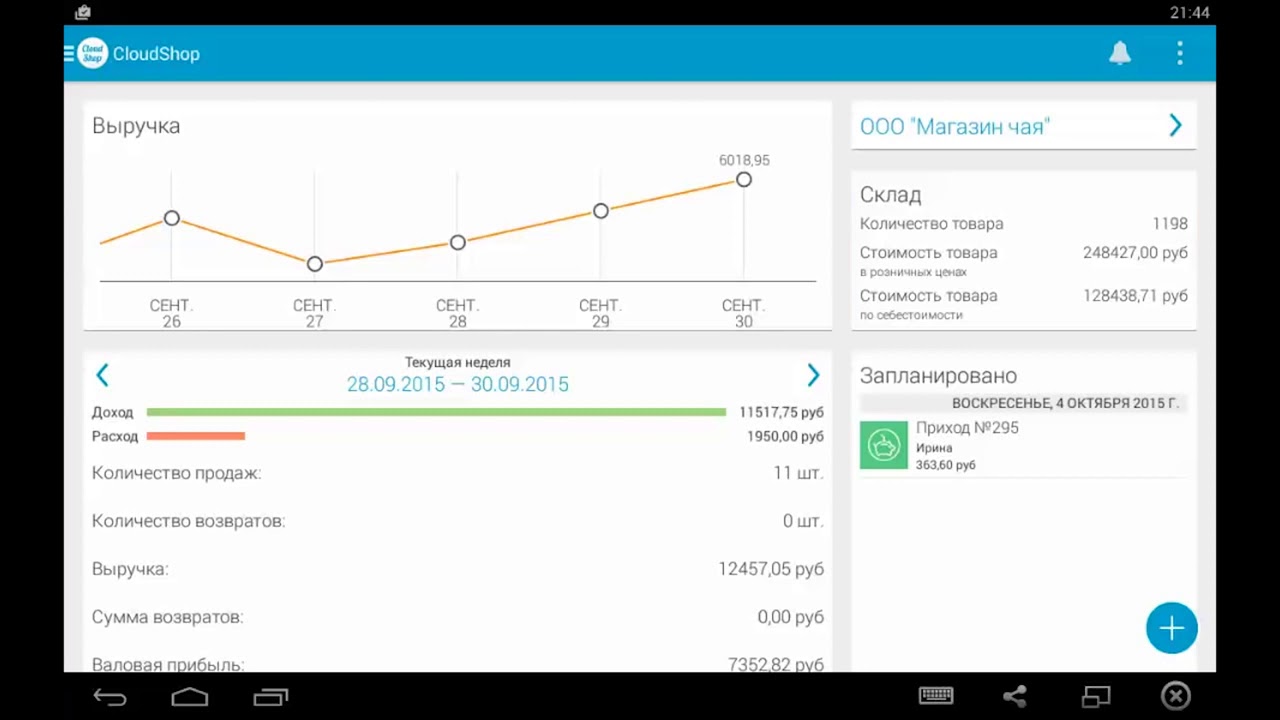


Рисунок 2.6 – Интерфейс "CloudShop"-2

Выбирая облачный сервис для учета продаж, товаров и клиентов CloudShop, вы выбираете высокую скорость выполнения операций, сохранность и безопасность вашей информации и мобильность.  Все передаваемые данные шифруются и передаются по защищенным каналам.

К числу основных преимуществ сервиса CloudShop перед основными конкурентами можно отнести:

1. Наличие полностью бесплатной версии (в пределах разрешенного количества работников, которые могут пользоваться сервисом, и функционала);
2. Наличие функций, позволяющих управлять ценами и скидками «вручную»;
3. Скорость внедрения (как правило, не превышает 1 недели, в то время как типичный для многих других товароучетных систем срок — 3-4 недели).
4. Передача данных по шифрованным каналам, резервное копирование;
5. Наличие возможности работать офлайн при отсутствии подключения к интернету (недоступны лишь отдельные функции — например, связанные с подготовкой отчетности);

К недостаткам относятся:

1. Плохая служба поддержки;
2. Интерфейс кассиров и основного окна программы не очень читабельный, тип и размер шрифтов;
3. Длинное название номенклатуры и чего всё не помещается в одну строку, часть характеристики товара переносится, что может привести к ошибкам списания товаров;
4. Небольшой функционал;
5. Веб-версия не всегда стабильно работает;
6. Отсутствие аналитики по кассирам и ассортиментным группам;
7. Отсутствует возможность доработки программ под конкретного клиента;

Таблица 3 – Сравнительная таблица

|  |  |  |
| --- | --- | --- |
| Параметр | 1С:Розница | CloudShop |
| Подходит для | Средний бизнес, ИП, Малый бизнес | Средний бизнес, ИП, Малый бизнес |
| Основная категория | [Системы управления торговой точкой (POS](https://soware.ru/categories/point-of-sale-systems/)) | [Системы управления торговой точкой (POS](https://soware.ru/categories/point-of-sale-systems/)) |
| Вид продукта | Система | Программа |
| Развёртывание | Персональный компьютер, Сервер предприятия | Мобильное устройство, Персональный компьютер |
| Тарификация | Единовременная оплата | Бесплатно, Ежемесячная оплата, Ежегодная оплата |
| Пробная версия | - | + |
| Функция администрирования | + | - |
| Импорт/ экспорт данных | + | + |
| Комбинированная оплата | + | \_ |
| Многопользовательский доступ | + | + |
| Отчетность и аналитика | + | + |
| Программа лояльности | + | + |
| Сканирование штрих-кодов | + | + |
| Управление персоналом | + | - |
| Управление чеками | + | + |
| Цена наименьшего тарифа | от 499 ₽/мес.\* | от 0 ₽/мес.\* или от 999 ₽/мес.\* |
| Цена наибольшего тарифа | от 3 999 ₽/мес.\* | от 8 999 ₽/мес.\* |

Таблица 4 – Сравнительная таблица по определенным показателям

|  |  |  |
| --- | --- | --- |
| Показатель | 1С:Розница | CloudShop |
| Назначение | Готовое решение для автоматизации всех основных бизнес-процессов розничной торговли. | Программа Cloudshop позволяет автоматизировать работу магазина в части управления жизненным циклом товара и продаж. |
| Эффективность | + | - |
| Гибкость | + | + |
| Защищенность | - | + |
| Живучесть | - | - |
| Надежность | + | + |
| Открытость | + | - |
| Оптимальность использования ресурсов | + | + |
| Удобство пользовательского интерфейса системы | - | + |
| Стоимость системы | Не указана | Не указана |
| Эргономичность | - | + |

**1С:Розница:** один из примеров цен на внедрение данной программы предоставлен на Рисунке 2.7 и Рисунке 2.8.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 2.7 – Стоимость проекта в целом

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2.8 – Переменная часть, исходя из кол-ва магазинов

В среднем, программу можно внедрить на крупное предприятие за 3–5 дней при помощи специалистов.

Причины, от которых зависит цена внедрения:

1. Уникальность бизнес-процессов, которые необходимо  
   автоматизировать в результате проекта;
2. Качество проработки и регламентации бизнес-процессов, которые следует автоматизировать;
3. Интеграция с другими системами;
4. Вовлеченность сотрудников Заказчика в проект внедрения (как функциональных пользователей, так и сотрудников IT службы);
5. Объём и состав документации, требуемой на проекте;

Объем дискового пространства занимаемого платформой 1С зависит от версии платформы, его релиза, устанавливаемых компонентов, языка интерфейса и типа установленного клиента ("толстый клиент" или "тонкий клиент"). Например, при релизе 2.2.8.29 размер программы составляет 774 МБ для файлового режима работы в операционной системе Windows.

Также на Рисунке 2.9 представлен один из примеров цены на программное обеспечение программы:



Рисунок 2.9 – Цены на пакеты

Отзыв одного из покупателей: «Выручка в 2021 году выросла (на начало декабря, по сравнению с 2020 годом): магазин Вега — на 50%, магазин Глобус — на 40%, магазин Экономыч — на 25%. На 50% ускорилось получение управленческой отчетности. На 20% увеличилась скорость обработки заказов. На 15% сократились административные издержки. На 10% выросла оборачиваемость складских запасов и сократились трудозатраты персонала. И, главное: магазины работают стабильно.»

**CloudShop:** в связи работы данный программы на небольших предприятиях, нет никакой конкретной информации о предоставлении услуг о ее внедрении.

На Рисунке 2.10 представлен пример цен на тариф одного из сайтов:

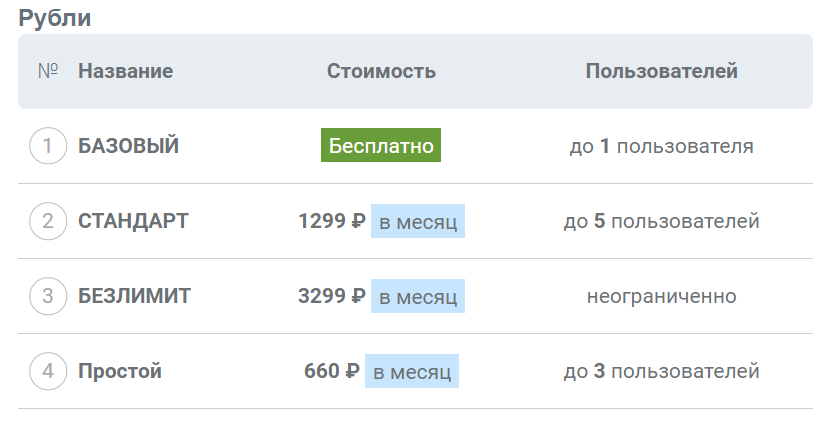


Рисунок 2.10 – Тарифы

Программа имеет собственную базу данных с готовыми штрих-кодами на 13 млн. товарных позиций.

Возможна установка специалистом примерно за 40000 тнг. В тарифе безлимит внедрение входит в стоимость.

Отзыв одного из покупателей: «Открывали второй магазин и пробовали сложные программы для учёта в магазине одежды. Большая часть функций оказалась не нужна. Система CloudShop полностью решила нашу задачу. Работать так же просто, как с таблицами Excel. Запускается на дешевых Android-планшетах. Отличный вариант для старта.»

Приложением пользуются более 3000 предпринимателей из 32 стран.

На Рисунке 2.11 представлено сравнение всех программ, созданное на основе ответов пользователей.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2.11 – Топ программ

Как итог, можно сказать следующее: из рассмотренных ранее программ, 1С:Розница и CloudShop, наиболее подходящей в использовании на выбранном объекте информатизации является первая.

1С:Розница подходит для больших предприятий, обладает обширным функционалом и возможностью выбора различных услуг по ее внедрению. К существенным недостаткам относится сложность в использовании пользовательского интерфейса, отсутствие руководства по работе с данной программы, цена тарифов (относительно CloudShop)

CloudShop проста в использовании, обладает стандартным функционалом и бесплатным тарифом. Однако подходит только для небольших предприятий, не имеет хорошей технической поддержки и возможности изменения программы под клиента. В отличие от первой, не такая востребованная на рынке.

## **ТЗ на создание программного продукта**

Для информатизации выбирается процесс обработки заказа препаратов, запрошенных пациентом на основе рецепта врача, проводившего осмотр в данной поликлинике, подсчета конечной стоимости, оплачиваемой государством или больным, фиксацией продажи и выдачи в журнале отчета. Объектом автоматизации могут являться поликлиники, аптечные пункты, больницы, фармацевтические предприятия.

Задачи:

* Задача ввода данных о пациенте и рецепте;
* Задача формирования заявки;
* Задача сохранения данных в базе;
* Задача записи о продаже в журнал учета;
* Задача формирования накладной;

Функциональные требования:

Требования к задаче «Задача ввода данных о пациенте и рецепте»: все необходимые для работы системы должны иметь некоторую базу, в которую включены обязательные поля, хранящие личную информацию об пациенте, враче и рецепте.

Требования к задаче «Задача формирования заявки»: на этапе формирования заявки необходимо внести всю информацию, полученную из прошлой задачи, а также сформировать индивидуальный номер заказа.

Требования к задаче «Задача сохранения данных в базе»: все необходимые для работы системы должны сохраняться в базе данных. Обязательно наличие возможности редактирования, удаления и экспорта произвольной информации из БД.

Требования к задаче «Задача записи о продаже в журнал учета»: все произведенные операции должны быть записаны не только в базу данных, но и в бумажную версию журнала учета.

Требования к задаче «Задача формирования накладной»: в накладной должны присутствовать наименование препарата, его цена, количество, номер заказа, а также реквизиты передающей и принимающей сторон.

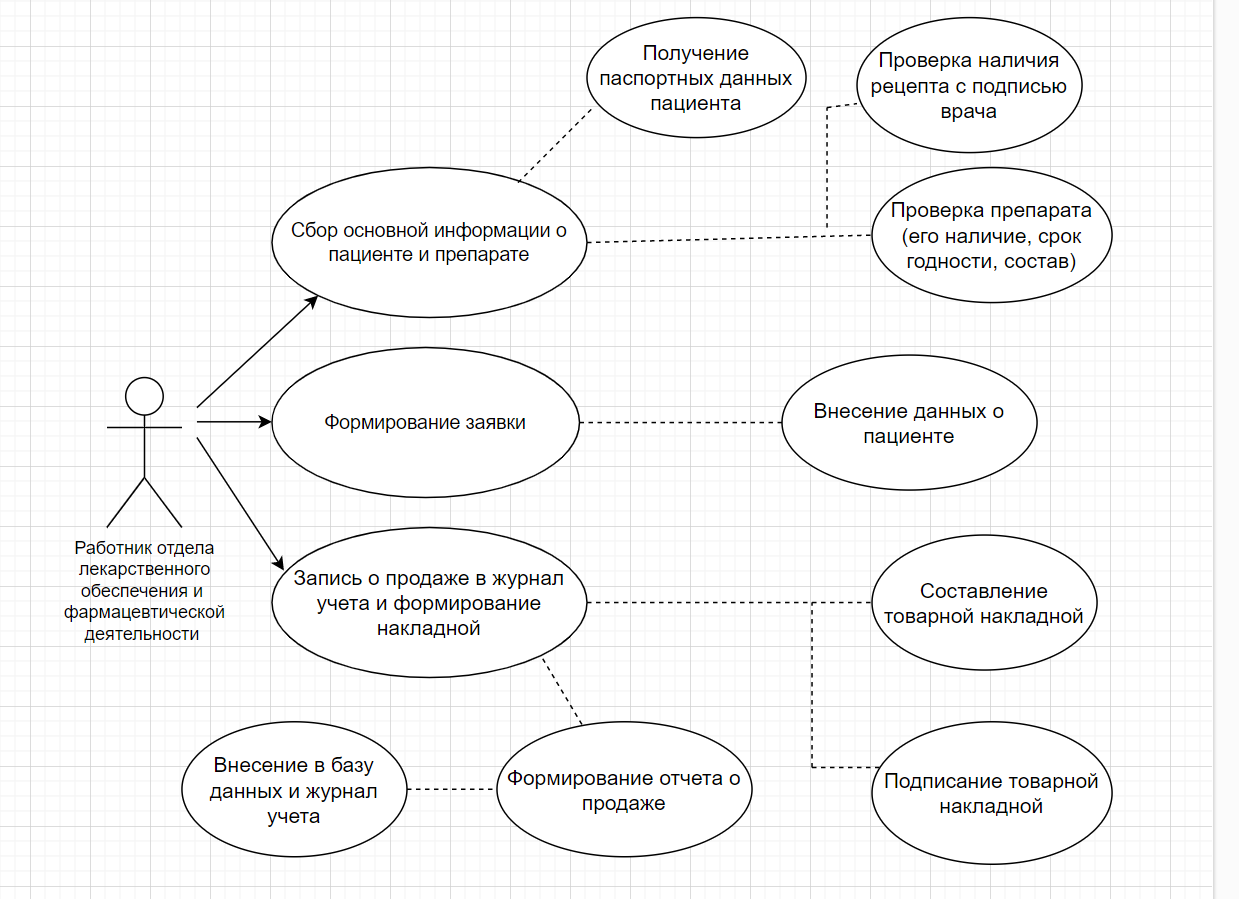


Рисунок 3.1 – Диаграмма требований UML-1

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 3.2 – Диаграмма требований UML-2

База данных проектируемой программной системы должна иметь следующие характеристики:

* Отсутствие дублирования информации;
* Низкая стоимость хранения использования данных;
* Защита от несанкционированного доступа к данным;

Требования к инструментальному программному обеспечению:

* Необходимо обеспечить поддержку операционных систем Windows 10 и выше;
* Наличие 1,5 ГБ свободного места на жестком диске;
* Visual Studio 2022 или более поздняя версия, C#;
* Разрешение экрана не менее 1024x768 пикселей;

Нефункциональные требования к ПО:

1. База данных для разрабатываемой системы не должна терять данные в случае аварийного завершения системы;
2. Для получения информации о заказе каждому клиенту будет предоставлен индивидуальный код с целью обезопасить данные;
3. Для удобного использования приложения необходима проработка дизайна и цветовой гаммы программы.
4. Программа может быть скачана из интернета либо передана с помощью внешнего источника;
5. Должно быть уточнение даты и времени всех проведенных операций (принятие заявки, отгрузка и доставка препарата и т.п) во время поставки;

## **Проектирование функциональной структуры программного продукта**

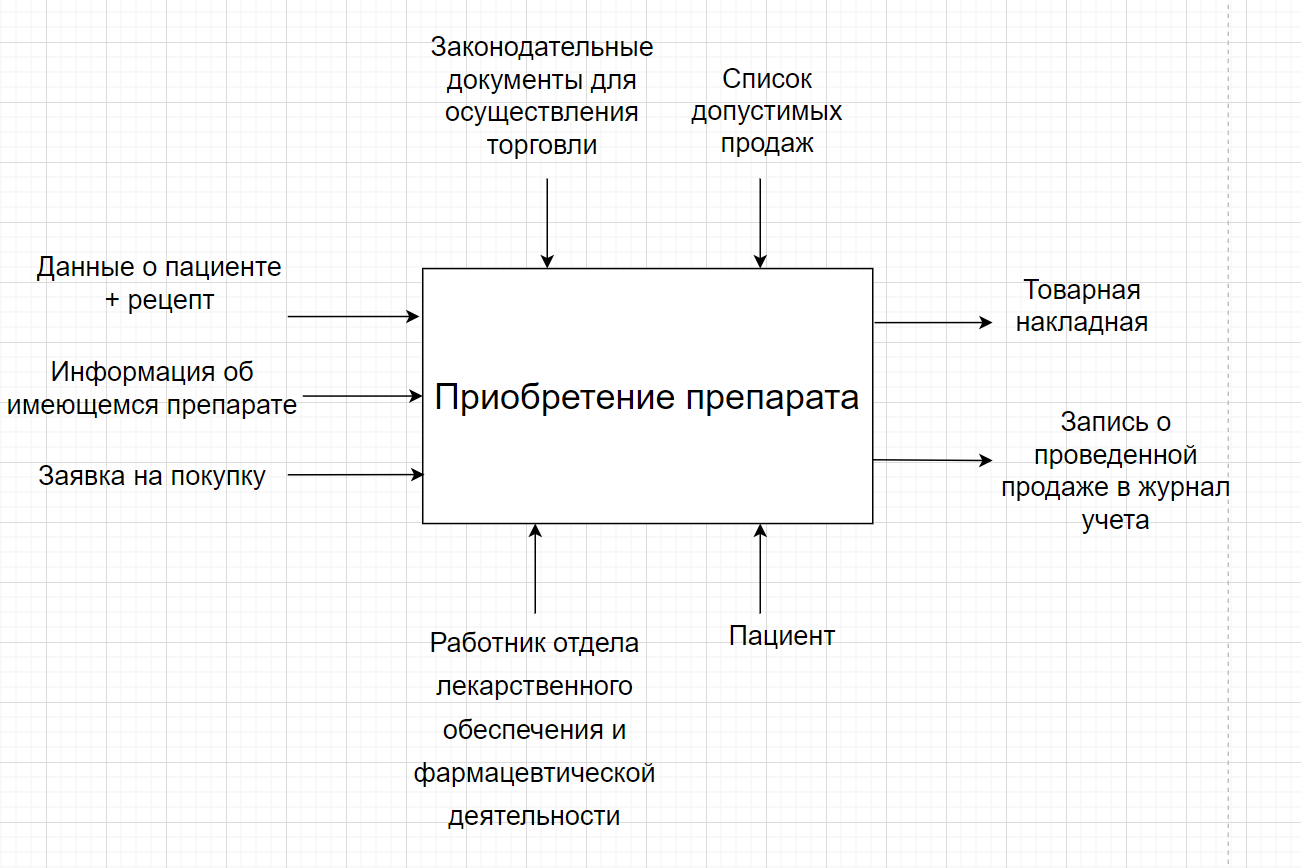


Рисунок 4.1 – Диаграмма в нотации IDEF0 – модель «Черный ящик»

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 4.2 – Диаграмма декомпозиции А0

Таблица 5 – Описание элементов функциональной модели

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование стрелки | Источник стрелки | Тип стрелки источника | Приемник стрелки | Тип стрелки приемника |
| Данные о пациенте + рецепте | Внешняя граница | Input | Заявка пациента | Input |
| Заявка на покупку | Внешняя граница | Input | Заявка пациента | Input |
| Информация об имеющимся препарате | Внешняя граница | Input | Информация о препарате | Input |
| Проверка акций | Информация о препарате | Output | Заявка пациента | Input |
| Наличие препарата | Информация о препарате | Output | Заявка пациента | Input |
| Проверка состава, срока годности | Информация о препарате | Output | Заявка пациента | Input |
| Формирование отчета о продаже | Заявка пациента | Output | Запись в журнал | Input |
| Формирование отчета о продаже | Заявка пациента | Output | Формирование накладной | Input |
| Внесение в базу данных | Запись в журнал | Output | Внешняя граница | Output |
| Накладная | Формирование накладной | Output | Внешняя граница | Output |
| Работник отдела лекарственного обеспечения и фармацевтической деятельности | Внешняя граница | Mechanism | Информация о препарате | Input |
| Работник отдела лекарственного обеспечения и фармацевтической деятельности | Внешняя граница | Mechanism | Заявка пациента | Input |
| Работник отдела лекарственного обеспечения и фармацевтической деятельности | Внешняя граница | Mechanism | Запись в журнал | Input |
| Работник отдела лекарственного обеспечения и фармацевтической деятельности | Внешняя граница | Mechanism | Формирование накладной | Input |

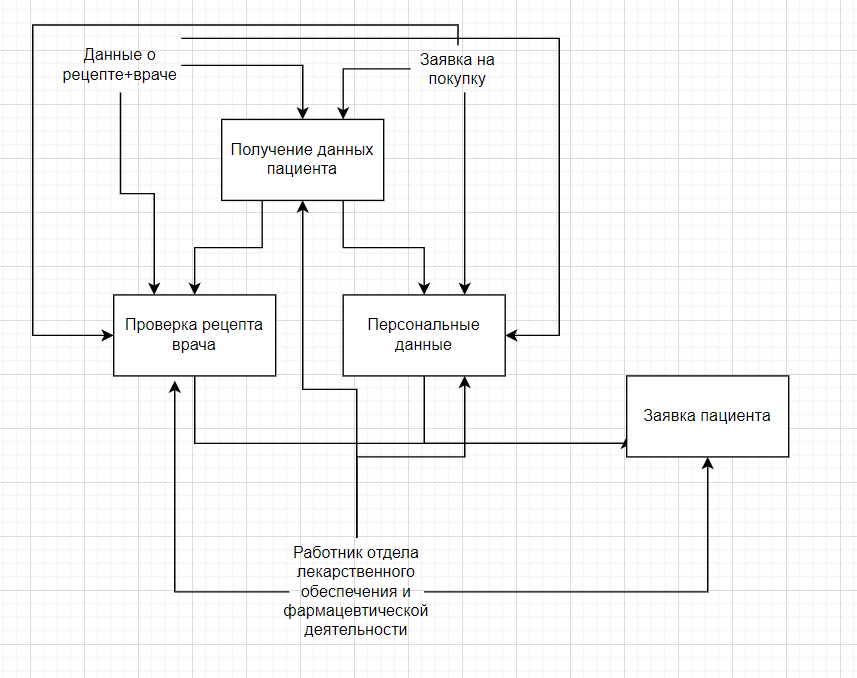


Рисунок 4.3 – Формирование заявки пациента

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 4.4 – Формирование выхода из блока информации о препарате

## **Проектирование функциональной структуры программного продукта**

Таблица 6 – Абстракции подсистемы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Тип | Описание |
| 1 | Работник отдела лекарственного обеспечения и фармацевтической деятельности | сущность | Пользователь системы, взаимодействующий с системой |
| 2 | Добавление/удаление препарата в заказе | поведение | Проведение работником поликлиники определенных действий (добавление/ удаление/ изменение) с препаратом в заказе |
| 3 | Выписка врачом рецепта | поведение | Выписка врачом рецепта, содержащего в себе диагноз пациента и необходимый препарат для его лечения |
| 4 | Пациент | сущность | Пользователь, обращающийся к работнику поликлиники с целью приобретения препарата, выписанного врачом |
| 5 | Внесение/изменение личной информации о пациенте | поведение | Внесение работником полученной от пациента личной информации с возможностью дальнейшего ее редактирования |
| 6 | Информация о препарате | сущность | Получение работником поликлиники основной информации о препарате, выписанном врачом для пациента, а именно: названия, состава, срока годности, стоимости, скидки |
| 7 | Проверка рецепта | поведение | Проверка рецепта на заполнения всех полей, необходимых для получения препарата |
| 8 | Формирование счета | поведение | Формирование итогового счета после получения и заполнения необходимой информации |
| 9 | Врач | сущность | Дополнительный пользователь, выписывающий рецепты пациентам |
| 10 | Формирование заказа | поведение | Формирование на основе полученной информации заказа |
| 11 | Проверка/изменение статуса заказа | поведение | Возможность получения и изменения работником поликлиники статуса выполнения заказа |
| 12 | Данные из рецепта | сущность | Получение основной информации о препарате, враче и диагнозе пациента |
| 13 | Формирование накладной | поведение | Формирование первичного документа, используемого при передаче товарно-материальных ценностей от одного лица к другому |
| 14 | Данные о пациенте | сущность | Предоставление пациентом личной информации, такой как: ФИО, дата рождения, уникальный номер рецепта и наличие льготы |
| 15 | Запись о проведенной продаже в журнал/БД | поведение | Внесение работником поликлиники соответствующей записи о продаже заказа в журнал/БД |
| 16 | Информация о враче | сущность | Предоставление базовой информации о враче, выписывающего рецепт |
| 17 | Счет | сущность | Формирование данных, описывающих итоговую стоимость заказа |
| 18 | Заказ | сущность | Сбор основной информации о проведенном заказе, построенного на основе рецепта пациента |
| 19 | Статус изменения заказа | сущность | Присвоение заказу определенного статуса в зависимости от этапа его выдачи |
| 20 | Запись в журнал | сущность | Создание записи о продаже с целью сохранения данных о заказе |
| 21 | Накладная | сущность | Предоставление возможности формирования накладной |

Таблица 7 - Классификация абстракции

|  |  |  |
| --- | --- | --- |
| № | Класс | Список абстракций |
| 1 | Люди | Работник отдела лекарственного обеспечения и фармацевтической деятельности, пациент, врач |
| 2 | Предметы | Информация о препарате, данные из рецепта, данные о пациенте, информация о враче, выдающего рецепт, счет, заказ, статус изменения заказа, запись в журнал, накладная |
| 3 | События | Добавление/удаление препарата в заказе, выписка врачом рецепта, проверка рецепта, предоставленного пациентом, внесение/изменение личной информации о пациенте, формирование счета, построенного на основе заказа, формирование заказа, проверка/изменение статуса заказа, формирование накладной, запись о проведенной продаже в журнал/БД |

Таблица 8 – Абстракции подсистемы и их поведение

|  |  |  |  |
| --- | --- | --- | --- |
| № | Абстракция | Поведение | Описание поведения |
| 1 | Работник отдела лекарственного обеспечения и фармацевтической деятельности | Добавить, удалить,  изменить, проверить, запросить, сформировать, записать | Добавление, удаление препаратов, указанных в заранее проверенном рецепте, предоставленном пациентом, данные которого запрашиваются при обращении; формирование заказа, счета, накладной; ведение записей в журнале/БД; проверка/изменение статуса заказа |
| 2 | Добавление/удаление препарата в заказе | Добавить, удалить | Добавление и удаление препаратов в заказе работников поликлиники |
| 3 | Выписка врачом рецепта | Выписать | Выписка врачом рецепта после проведения осмотра пациента |
| 4 | Пациент | Предоставить, оплатить | Предоставление пациентом личной информации для оформления и последующей оплаты заказа |
| 5 | Внесение/изменение личной информации о пациенте | Внести, изменить | Внесение или изменение работником поликлиники личной информации пациента в случае необходимости |
| 6 | Информация о препарате | Проверить | Проверка работником поликлиники информации о препарате, указанном в рецепте |
| 7 | Проверка рецепта | Проверить | Проверка наличия и подлинности рецепта пациента |
| 8 | Формирование счета | Сформировать | Формирование счета работником поликлиники после оплаты пациентом заказа |
| 9 | Врач | Выписать | Выписка рецепта пациенту |
| 10 | Формирование заказа | Сформировать | Формирование заказа пациента на основе предоставленной информации |
| 11 | Проверка/изменение статуса заказа | Проверить, изменить | Проверка и изменение статуса заказа работником поликлиники |
| 12 | Данные из рецепта | Предоставить, проверить | Предоставление пациентом рецепта врача и последующая проверка его работником поликлиники |
| 13 | Формирование накладной | Сформировать | Формирование накладной работником поликлиники после оформления и оплаты заказа |
| 14 | Данные о пациенте | Предоставить | Предоставление личной информации пациента работнику поликлиники |
| 15 | Запись о проведенной продаже в журнал/БД | Записать | Запись работником поликлиники продажи заказа в журнал/БД |
| 16 | Информация о враче | Проверить | Проверка работником поликлиники информации о враче, указанном в рецепте |
| 17 | Счет | Предоставить | Предоставление итогового счета пациенту и работнику поликлиники |
| 18 | Заказ | Предоставить | Предоставление заказа пациенту |
| 19 | Статус изменения заказа | Сформировать | Сформировать после создания заказа статус его изменения |
| 20 | Запись в журнал | Записать | Запись в журнал работником поликлиники о сформированном заказе |
| 21 | Накладная | Предоставить | Предоставление итоговой накладной работнику поликлиники |

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

Рисунок 5.1 – Диаграмма классов

Изображение выглядит как в помещении

Автоматически созданное описание

Рисунок 5.2 – Диаграмма состояния рецепта

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 5.3 – Диаграмма состояния заказа Изображение выглядит как текст, седзи

Автоматически созданное описание

Рисунок 5.4 – Диаграмма состояния статуса заказа

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 5.5 – Диаграмма деятельности

|  |  |  |  |
| --- | --- | --- | --- |
| № п.п | Прецедент | Объект | Описание объекта |
| 1 | Оформить продажу товара пациенту по заказу | Модуль формирования продажи | Часть системы, ответственная за оформление продажи |
| 2 | Подсчет стоимости товаров | Модуль подсчета стоимости товаров | Часть системы, ответственная за подсчет общей стоимости товаров в заказе |

Изображение выглядит как диаграмма

Автоматически созданное описание

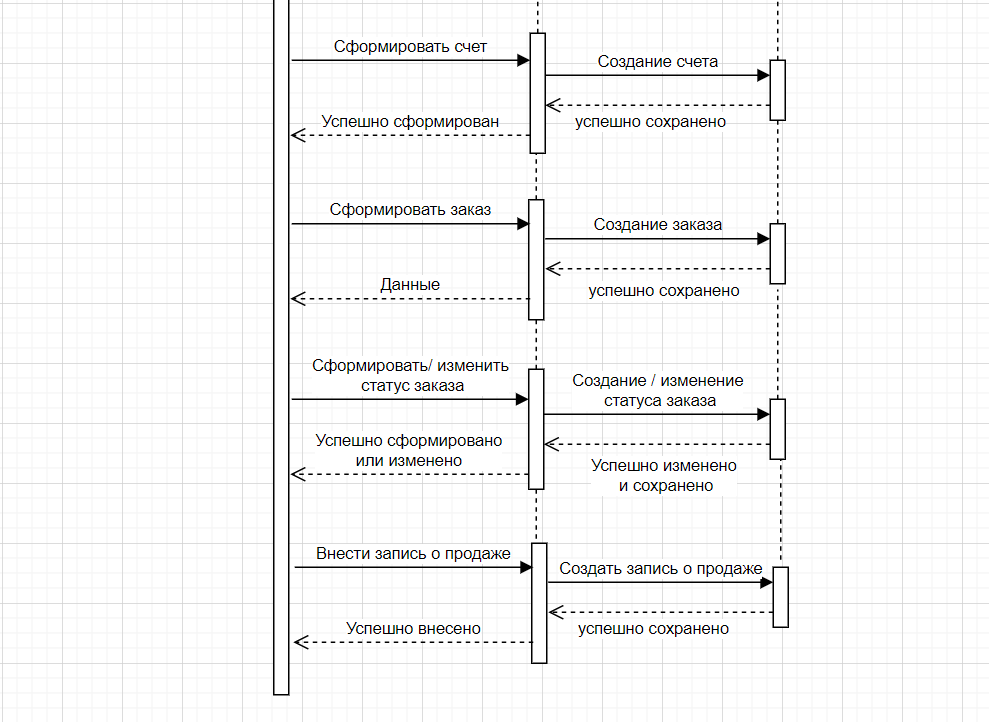


Рисунок 5.6 – Диаграмма последовательности

## **Проектирование Базы данных программной системы**

Изображение выглядит как диаграмма, схематичный

Автоматически созданное описание

Рисунок 6.1 – модель классов UML

Таблица 9 – идентификация сущностей информационной базы ПС

|  |  |
| --- | --- |
| Документ | Сущность |
| Перечень препаратов и их описание | Препарат |
| Перечень препаратов в заказе | Рецепт |
| Данные о работнике поликлиники | Работник поликлиники |
| Запись о пациенте в журнал | Пациент |
| Заявка на приобретение препарата | Заявка пациента |
| Данные о нахождении заказа пациента | Статус заказа |
| Данные о работнике, пациенте и заказе | Заказ |
| Данные о враче, выписывающем рецепт | Врач |

Таблица 10 – Сущности физической модели данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущность | Атрибуты | Описание |
| Справочные | | | |
| 1 | Работник | ID работника, Фамилия, Имя, Отчество, дата рождения | Информация о работниках |
| 2 | Препарат | ID препарата, название, состав, срок годности, наличие на складе, стоимость, скидка | Информация о препарате |
| 3 | Врач | ID врача, ФИО, дата рождения, специальность | Информация о враче |
| Оперативные | | | |
| 4 | Рецепт | ID рецепта, ФИО врача, диагноз, необходимый препарат | Информация о рецепте пациента |
| 5 | Пациент | ИНН, Фамилия, Имя, Отчество, Дата рождения, ID рецепта, наличие льгот | Информация о пациенте |
| 6 | Заявка пациента | ID заявки, ИНН пациента, время подачи заявки | Информация о заявках на покупку |
| 7 | Статус заказа | ID статуса, ID заявки, ID работника, статус, время изменения | Информация о статусе заказа |
| 8 | Заказ | ID препарата, ИНН пациента, ID заявки, ID работника | Информация о заказе |

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 6.2 – ER-диаграмма логической модели данных

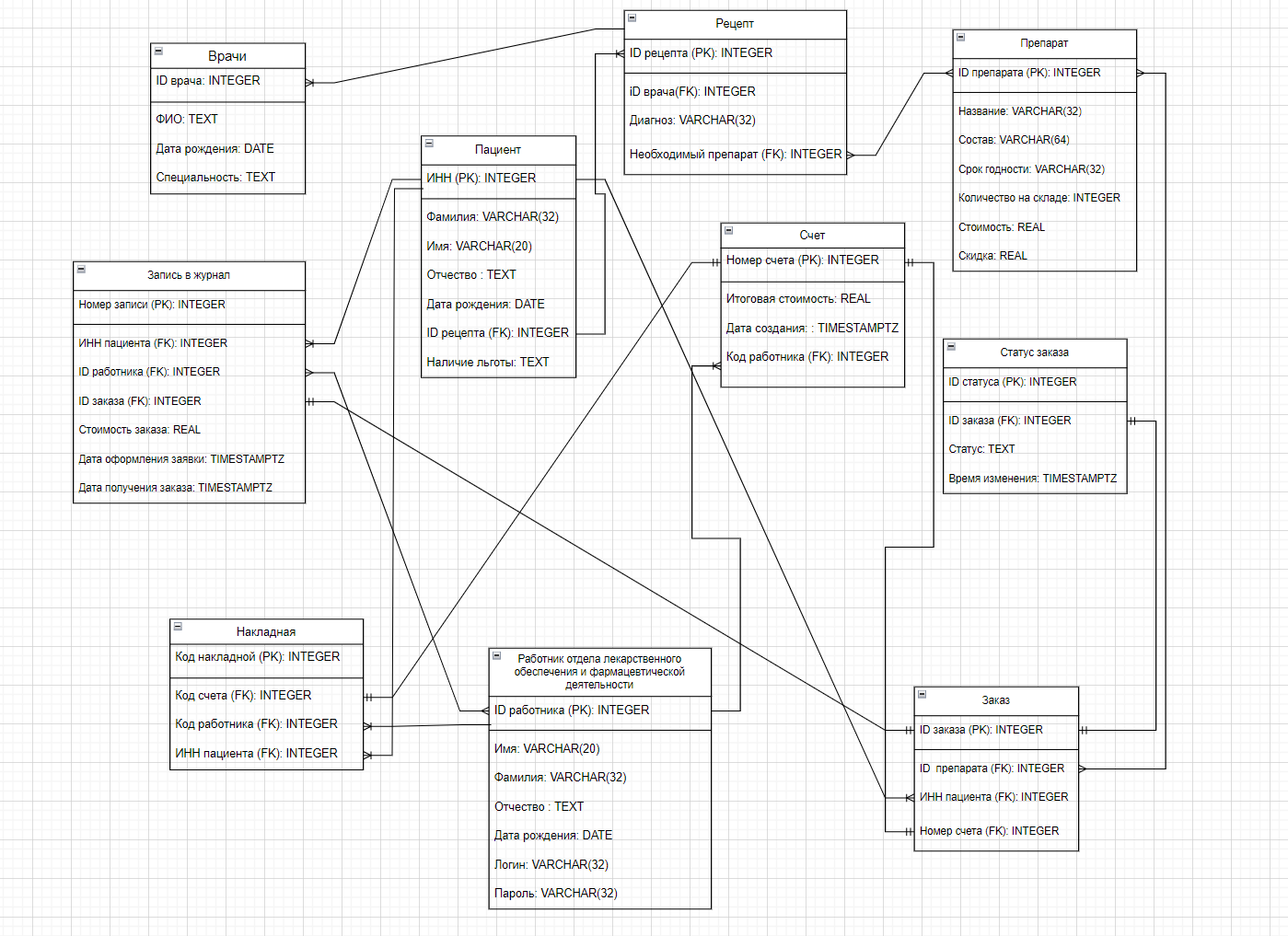


Рисунок 6.3 – ER-диаграмма физической модели данных

Сравнительный анализ с диаграммой классов показал на верную проектировку будущей БД. Существует вероятность дублирования информации, которую можно избежать, проведя нормализацию. Также существуют атрибуты, которые могут быть постоянно заполнены Null.

Для разработки ПС была выбрана СУБД PostgreSQL по следующим причинам:

* Бесплатная объектно-реляционная СУБД с открытым исходным кодом
* Гибкий доступ к базам данных, их организация и хранение
* Поддержка множества типов данных
* Работа с большими объемами
* Поддержка сложных запросов
* Одновременная модификация базы
* Высокая мощность и широкая функциональность
* Наличие удобного интерфейса при работе с БД

Далее представлена генерация SQL-кода для создания реляционной базы данных для ПС:

CREATE TABLE doctor (

id\_doctor BIGSERIAL NOT NULL PRIMARY KEY,

FIO VARCHAR(80) NOT NULL,

date\_of\_b DATE,

specialization TEXT NOT NULL

);

CREATE TABLE recipe (

id\_recipe BIGSERIAL NOT NULL PRIMARY KEY,

id\_doctor INTEGER REFERENCES doctor (id\_doctor),

diagnos VARCHAR(32) NOT NULL,

id\_prep INTEGER REFERENCES preparat (id\_prep)

);

CREATE TABLE preparat (

id\_prep BIGSERIAL NOT NULL PRIMARY KEY,

name VARCHAR(32) NOT NULL,

sostav VARCHAR(64) NOT NULL,

srok\_god VARCHAR(32) NOT NULL,

kol INTEGER DEFAULT 0,

price REAL NOT NULL,

discount REAL DEFAULT 0

);

CREATE TABLE patient (

id\_patient BIGSERIAL NOT NULL PRIMARY KEY,

surname VARCHAR(32) NOT NULL,

name VARCHAR(20) NOT NULL,

otchestvo TEXT,

date\_of\_bir DATE NOT NULL,

id\_recipe INTEGER REFERENCES recipe (id\_recipe),

plyclinic(# lgot TEXT DEFAULT 0

);

CREATE TABLE worker (

id\_worker BIGSERIAL NOT NULL PRIMARY KEY,

surname VARCHAR(32) NOT NULL,

name VARCHAR(20) NOT NULL,

otchestvo TEXT,

date\_of\_bir DATE NOT NULL,

login VARCHAR(32) NOT NULL,

parol VARCHAR(32) NOT NULL

);

CREATE TABLE chet (

id\_chet BIGSERIAL NOT NULL PRIMARY KEY,

price REAL NOT NULL,

date TIMESTAMPTZ NOT NULL,

id\_worker INTEGER REFERENCES worker (id\_worker)

);

CREATE TABLE zakaz (

id\_zakaz BIGSERIAL NOT NULL PRIMARY KEY,

id\_prep INTEGER REFERENCES preparat (id\_prep),

id\_patient INTEGER REFERENCES patient (id\_patient),

id\_chet INTEGER REFERENCES chet (id\_chet)

);

# CREATE TABLE state\_zakaz (

id\_state\_zakaz BIGSERIAL NOT NULL PRIMARY KEY,

id\_zakaz INTEGER REFERENCES zakaz (id\_zakaz),

state TEXT NOT NULL,

time TIMESTAMPTZ NOT NULL

);

CREATE TABLE naklad (

id\_naklad BIGSERIAL NOT NULL PRIMARY KEY,

id\_chet INTEGER REFERENCES chet (id\_chet),

id\_worker INTEGER REFERENCES worker (id\_worker),

id\_patient INTEGER REFERENCES patient (id\_patient)

);

CREATE TABLE write\_mag (

id\_write\_mag BIGSERIAL NOT NULL PRIMARY KEY,

id\_patient INTEGER REFERENCES patient (id\_patient),

id\_worker INTEGER REFERENCES worker (id\_worker),

id\_zakaz INTEGER REFERENCES zakaz (id\_zakaz),

price REAL NOT NULL,

date\_oform TIMESTAMPTZ NOT NULL,

date\_poluch TIMESTAMPTZ NOT NULL

);

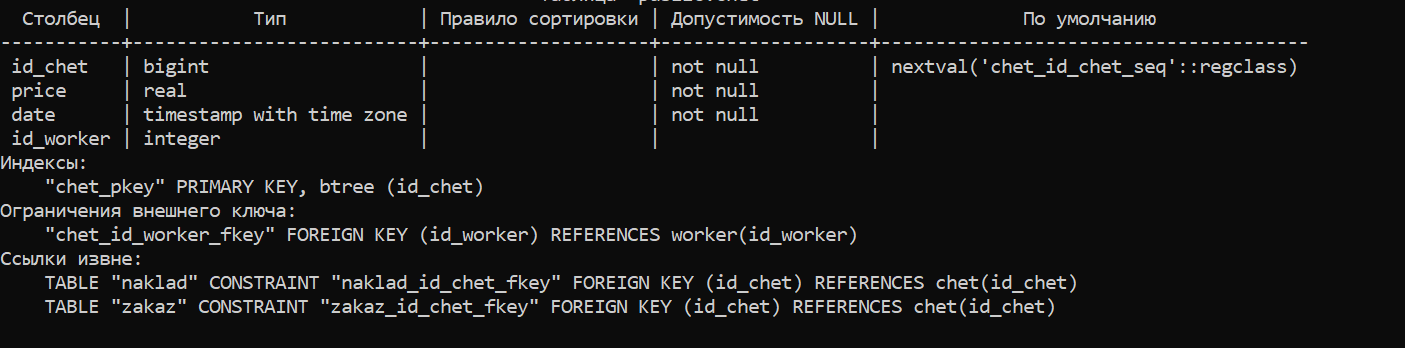


Рисунок 6.4.1 – Структура таблицы chet

Рисунок 6.4.2 – Изображение выглядит как текст, снимок экрана, монитор, в помещении

Автоматически созданное описаниеСтруктура таблицы doctor

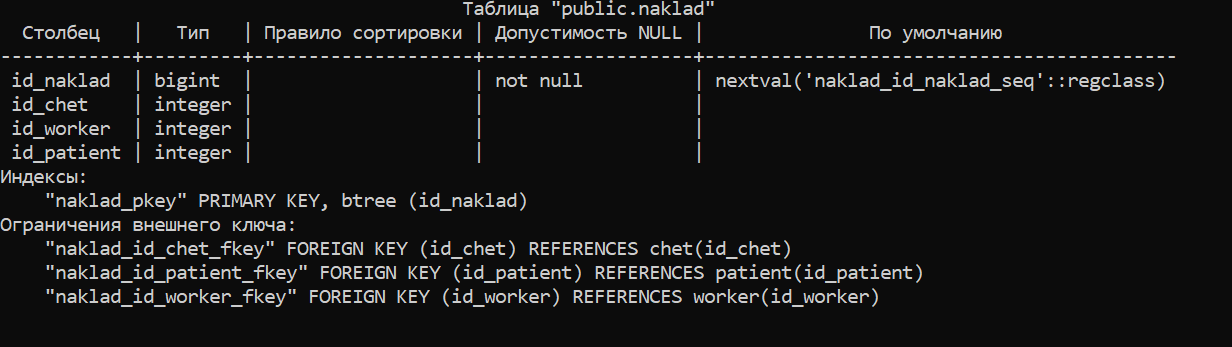


Рисунок 6.4.3 – Структура таблицы naklad

Изображение выглядит как текст

Автоматически созданное описание Рисунок 6.4.4 – Структура таблицы patient

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6.4.5 – Структура таблицы preparat

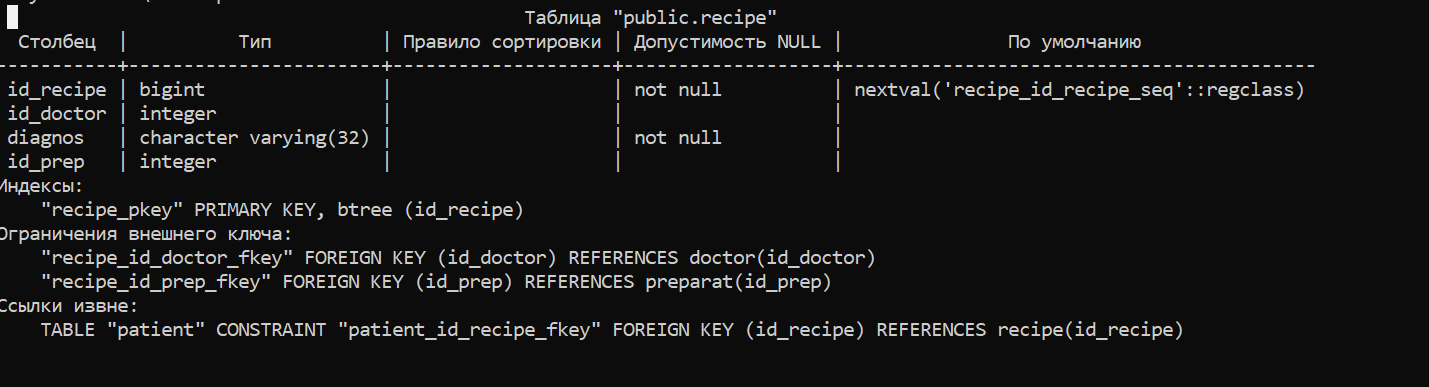


Рисунок 6.4.6 – Структура таблицы recipe

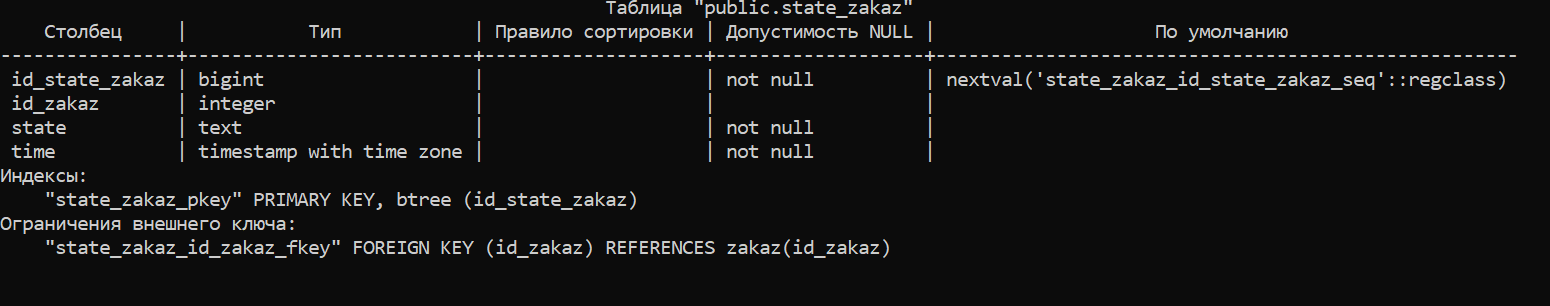


Рисунок 6.4.7 – Структура таблицы state\_zakaz

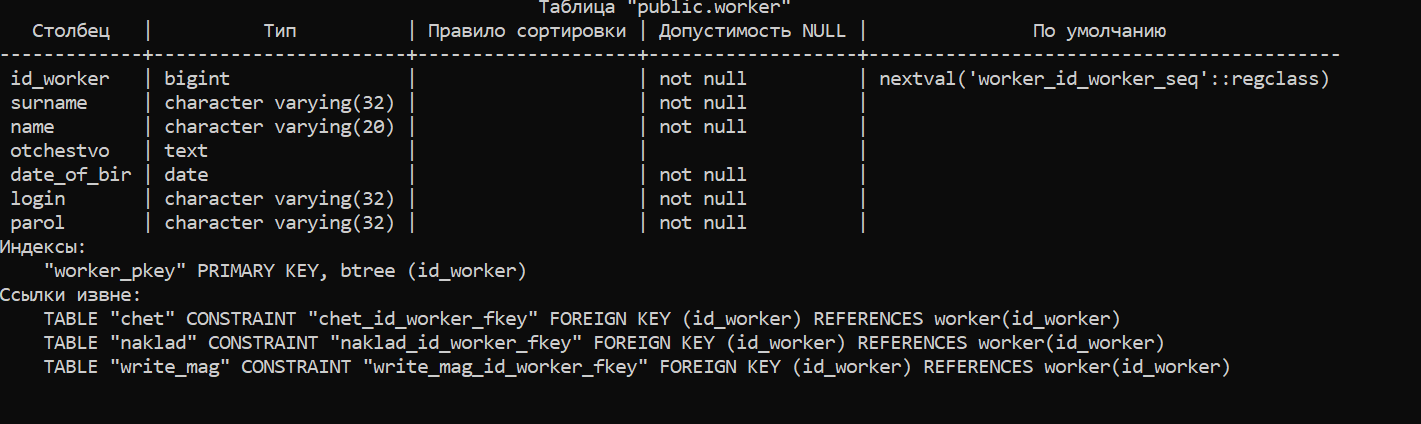


Рисунок 6.4.8 – Структура таблицы worker

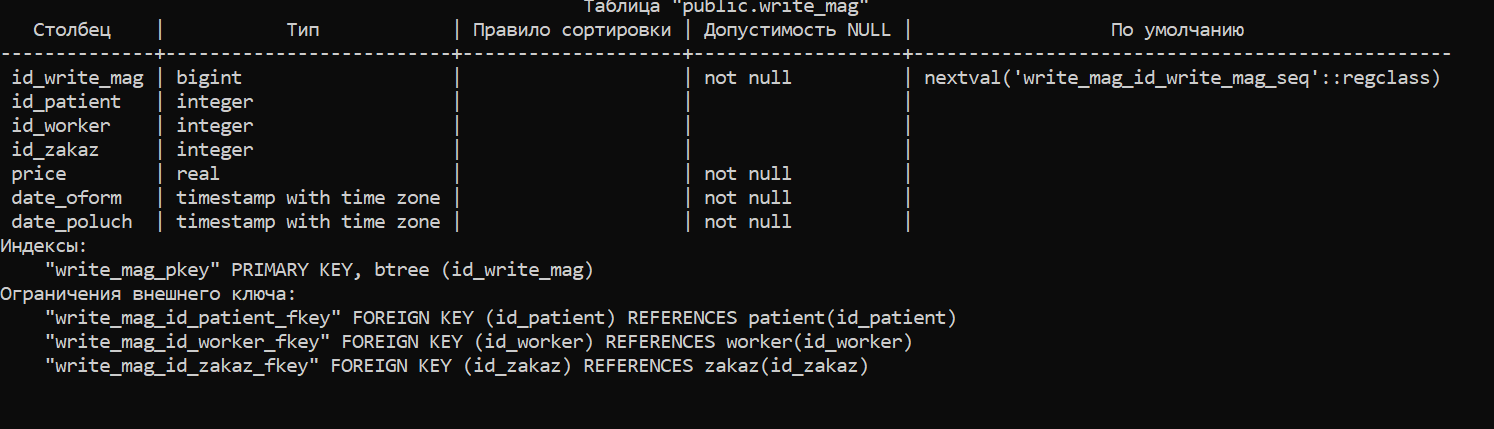


Рисунок 6.4.9 – Структура таблицы write\_mag

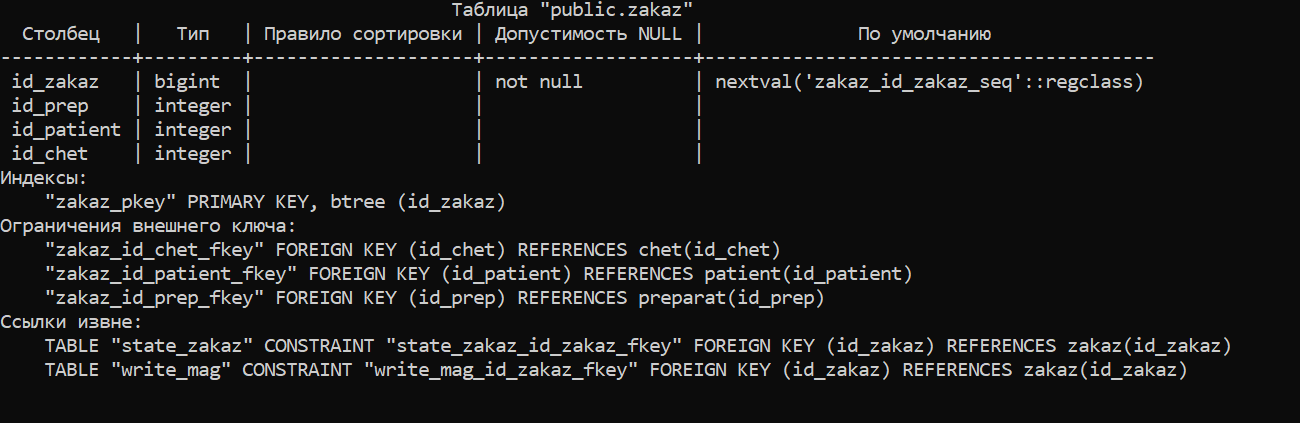


Рисунок 6.4.10 – Структура таблицы zakaz

Таблица 11 – Список разработанных таблиц

|  |  |  |
| --- | --- | --- |
| № | Имя таблицы | Описание |
| 1 | chet | Описывает сущность счета |
| 2 | doctor | Описывает сущность врача |
| 3 | naklad | Описывает сущность накладной |
| 4 | patient | Описывает сущность пациента |
| 5 | preparate | Описывает сущность препарата |
| 6 | recipe | Описывает сущность рецепта |
| 7 | state\_zakaz | Описывает сущность статуса заказа |
| 8 | worker | Описывает сущность работника поликлиники |
| 9 | write\_mag | Описывает сущность записи в журнал |
| 10 | zakaz | Описывает сущность заказа |

Таблица 12 – Связи между таблицами БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Родительская таблица | | Дочерняя таблица | | Тип связи |
| Название | Атрибут | Название | Атрибут |
| recipe | id\_doctor | doctor | id\_doctor | Один-ко-многим |
| patient | id\_recipe | recipe | id\_recipe | Один-ко-многим |
| recipe | id\_preparate | preparate | id\_preparate | Многие-ко-многим |
| naklad | id\_worker | worker | id\_worker | Один-ко-многим |
| write\_mag | id\_zakaz | zakaz | id\_zakaz | Один-к- одному |
| write\_mag | id\_worker | worker | id\_worker | Многие-ко-многим |
| naklad | id\_patient | patient | id\_patient | Один-ко-многим |
| naklad | id\_chet | chet | id\_chet | Один-к- одному |
| zakaz | id\_preparate | preparate | id\_preparate | Многие-ко-многим |
| chet | id\_worker | worker | id\_worker | Один-ко-многим |
| zakaz | id\_chet | chet | id\_chet | Один-к- одному |
| stat\_zak | id\_zakaz | zakaz | id\_zakaz | Один-к- одному |
| zakaz | id\_patient | patient | id\_patient | Один-ко-многим |
| write\_mag | id\_patient | patient | id\_patient | Один-ко-многим |

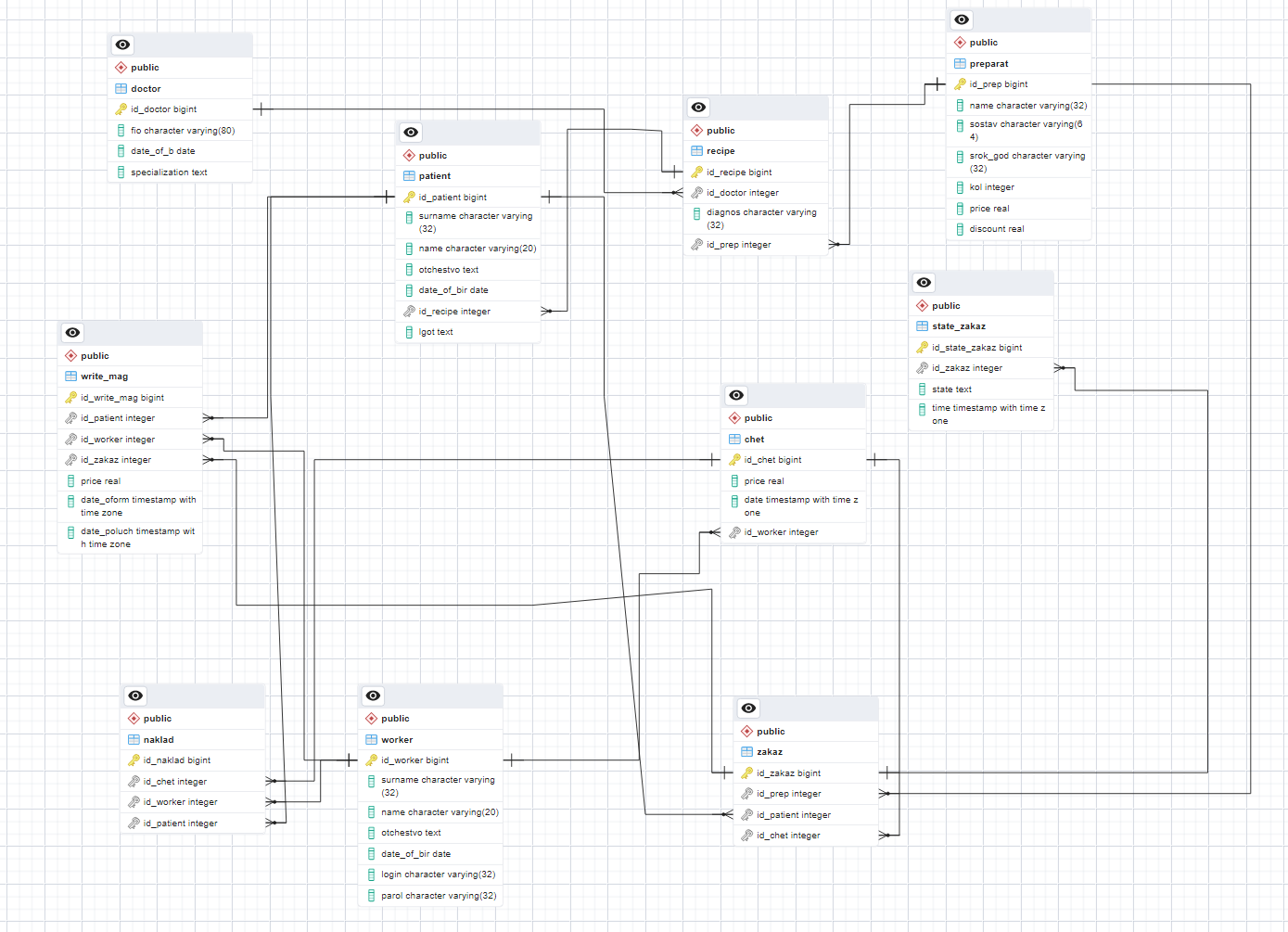


Рисунок 6. 5 – Структура БД в PostgreSQL

## **Разработки программного обеспечения ПС**

Структурная схема общесистемного программного обеспечения:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 7.1 – Структурная схема

Таблица 13 – перечень разработанных компонентов приложения

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | CRUD.cs | source | Файл, необходимый для подключения к базе данных PostgreSQL и дальнейшей с ней работой. |
| 2 | Form1.cs | source | В данном файле определяются все элементы управления, которые находятся на форме, а также их свойства и события. Кроме того, в нем определяются все методы, которые обрабатывают события элементов управления. |
| 3 | Form1.Designer.cs | source | Является вспомогательным файлом для работы с формой в проекте. Он содержит автоматически сгенерированный код для инициализации элементов управления и обработки событий. Является аналогом под номером 3, однако не предоставляет возможности изменения вручную |
| 4 | Form1.resx | file | Отвечает за хранение ресурсов формы. Он содержит все необходимые ресурсы, такие как изображения, строки, звуки и т.д., которые используются на форме. генерируется автоматически и не должен изменяться вручную. |
| 5 | OnlineStore.sln | Executable | В этом файле хранятся все проекты, которые входят в решение, а также их зависимости и настройки. |
| 6 | packages.config | file | содержит список всех пакетов NuGet, которые были установлены в проекте, а также их версии и зависимости. |
| 7 | Program.cs | source | Отвечает за запуск приложения и инициализацию главной формы. Он содержит метод Main, который является точкой входа в приложение. |
| 8 | App.config | file | Отвечает за хранение настроек приложения. Используется для хранения конфигурационных данных, таких как строки подключения к БД, настройки безопасности, настройки локализации и другие пользовательские настройки |

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 7.2 – Диаграмма компонентов

Анализ сложности кода:

В результате анализа кода сложности были получены следующие результаты:

* 26 функций и методов
* 16 переменных
* 3 оператора

Таблица 14 – Физические элементы программной системы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 1 | Рабочая станция | processor | Основная рабочая станция пользователя |

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 7.3 – Диаграмма развертывания

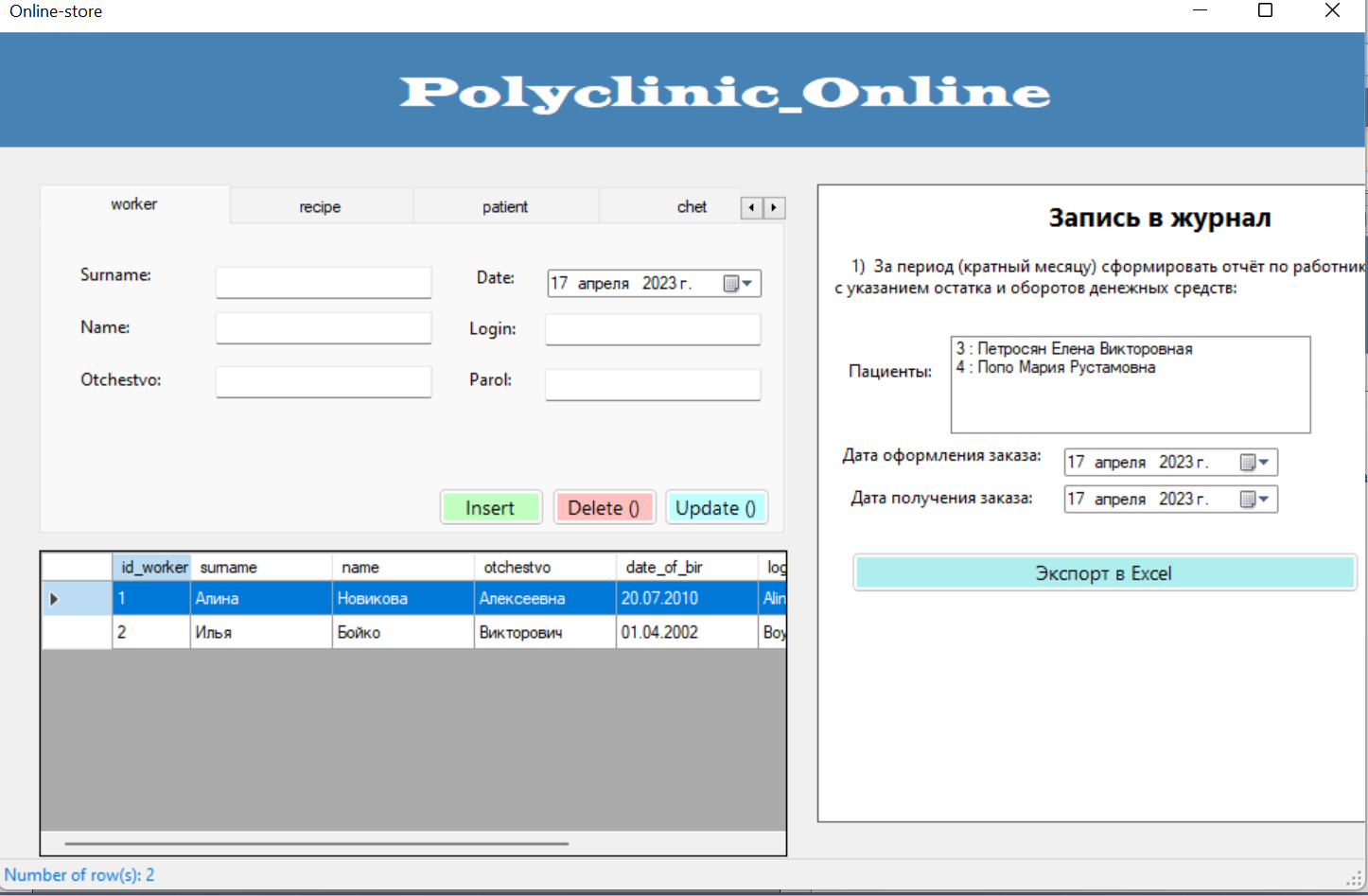


Рисунок 7.4 – Оконная форма

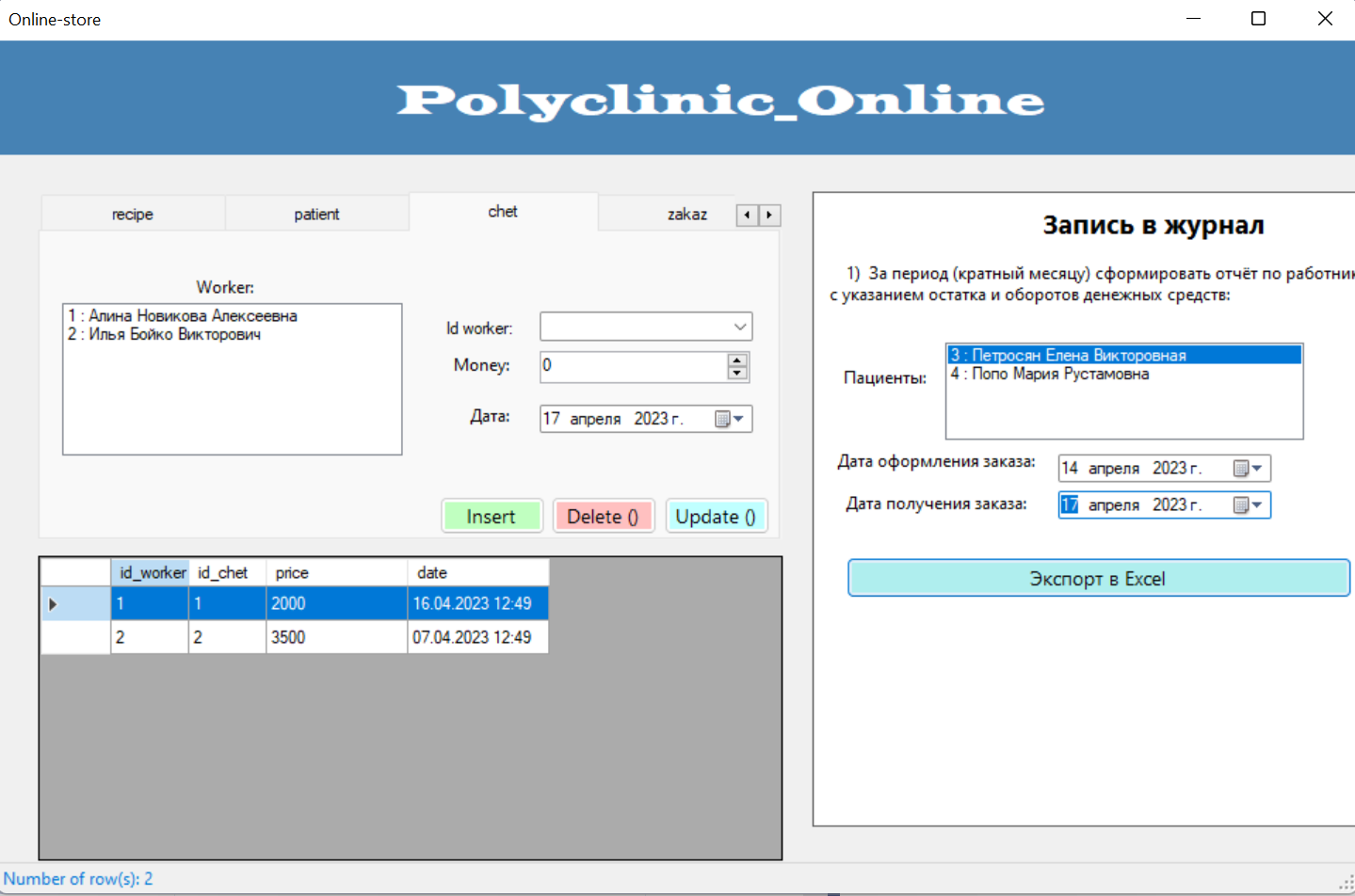


Рисунок 7.5 – Оконная форма ответов записи в журнал

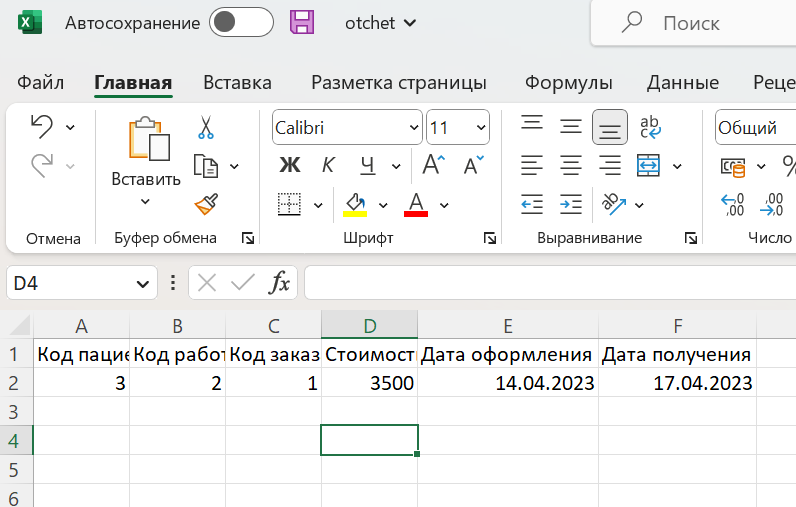


Рисунок 7.6 – Сохранение в Excel

## **Тестирование программной системы**

Были протестированы следующие функции системы:

* Добавление, изменение и удаление работников
* Добавление, изменение и удаление рецепта
* Добавление, изменение и удаление пациентов
* Добавление, изменение и удаление счета
* Добавление, изменение и удаление заказа
* Создание записи в журнал через Excel

В результате ручного тестирования были найдены следующие ошибки: 1. Отсутствие наличие ограничений в полях Даты рождения у Работника(worker), Пациента (patient). См. Рисунок 8.1

Последовательность для обнаружения: выбрать текущую дату в качестве значения. Внешнее проявление ошибки отсутствует. Способ ее устранения: введение триггеров, содержащих ограничение.

2. Отсутствие проверки логина и пароля у работника(worker). См. Рисунок 8.2

Последовательность для обнаружения: ввод полей login и parol. Внешнее проявление ошибки отсутствует. Способ ее устранения: введение БД, содержащее данные всех логинов и паролей различных работников, а также триггеров, осуществляющих проверку на наличие введенной информации.

3. Проведение записи в журнал только одного клиента в связи с записью только одной выбранной даты. См. Рисунок 8.3

Последовательность для обнаружения: выбор нескольких клиентов и ввод одной даты. Внешнее проявление ошибки: наличие одинаковых полей даты в excel для различных клиентов. Способ ее устранения: введение отдельной вкладки с полями для записи в журнал, после чего происходило бы формирование отчета для различных клиентов.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8.1 – Ошибка ввода даты

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8.2 – Ошибка проверки login, parol

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8.3 – Ошибка одинаковых дат

Технические характеристики машины:

* Процессор – 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
* Оперативная память – 8,00 ГБ
* Жесткий диск, на котором проводилось тестирование – HDD, 250 ГБ

Было проведено нагрузочное тестирование на следующих объемах случайных данных:

N – работников, N – пациентов, N – рецептов, N – счетов, N – заказов

Каждый из N работников записывает N пациентов с вероятностью 0.5 (в среднем N^2/2);

Каждый из N пациентов производят N заказов (в среднем 2\*N^2);

Каждый из N счетов включают в себя препараты из N рецептов с (в среднем 5\*N^2/2);

Для значений N= 10, 50, 100, 250, 700;

Были произведены замеры времени загрузки данных в БД следующих таблиц:

* Список всех докторов
* Список всех пациентов
* Список всех работников

По результатам нагрузочного тестирования, заметно меняется лишь время загрузки информации в БД работников, препаратов и записи в журнал. На Таблице 15 и Рисунке 8.4 приведены результаты.

Таблица 15 – Результаты нагрузочного тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| БД | N = 10 | N = 50 | N = 100 | N = 250 | N = 700 |
| worker | 0,029 | 0,037 | 0,019 | 0,049 | 0,044 |
| preparate | 0,008 | 0,04 | 0,027 | 0,111 | 0,2234 |
| write\_mag | 0,079 | 0,323 | 0,652 | 1,123 | 9,637 |

Рисунок 8.4 – Зависимость времени выполнения запроса от N

На Рисунке 8.5 представлен график зависимости затраченного времени от количества записей при отсутствии сторонней нагрузки на оборудование.

Рисунок 8.5 – Зависимость времени выполнения запроса от N (log)

Стрессовое тестирование проводилось на случайных данных, сгенерированных N = 1000, после чего было замерено время ответа БД каждой из вкладок программы. Результаты измерений таблиц приведены далее.

|  |  |
| --- | --- |
| Вкладка | Среднее время загрузки (секунды) |
| worker | 0,015856 |
| recipe | 0,12133 |
| patient | 1,0239 |
| chet | 0,9384 |
| zakaz | 0,00242 |
| doctor | 0,57894 |
| preparate | 1,49023 |
| state\_zakaz | 1,84832 |
| write\_mag | 0,4576 |
| naklad | 0,5234 |

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8.6 – Нагрузка ноутбука при помощи Furmark

Пустой проект Unit test:

[Test]  
public void TestPatientInsertButton\_Click()  
{  
    // Arrange  
    var form = new Form();  
    var namePatBox = new TextBox() { Text = "John" };  
    var surnamePatBox = new TextBox() { Text = "Doe" };  
    var otchestvoPatBox = new TextBox() { Text = "Smith" };  
    var bDateP = new DateTimePicker() { Value = new DateTime(1990, 1, 1) };  
    var idRecipeBox = new TextBox() { Text = "1" };  
    var lgotaBox = new TextBox() { Text = "Yes" };  
    form.Controls.Add(namePatBox);  
    form.Controls.Add(surnamePatBox);  
    form.Controls.Add(otchestvoPatBox);  
    form.Controls.Add(bDateP);  
    form.Controls.Add(idRecipeBox);  
    form.Controls.Add(lgotaBox);  
  
    var patientInsertButton = new Button();  
    var eventArgs = new EventArgs();  
    bool messageBoxShown = false;  
  
    // Act  
    patientInsertButton\_Click(patientInsertButton, eventArgs);  
  
    // Assert  
    if (string.IsNullOrEmpty(namePatBox.Text.Trim())  
        || string.IsNullOrEmpty(surnamePatBox.Text.Trim())  
        || string.IsNullOrEmpty(otchestvoPatBox.Text.Trim())  
        || string.IsNullOrEmpty(lgotaBox.Text.Trim()))  
    {  
        messageBoxShown = true;  
        Assert.IsTrue(messageBoxShown);  
    }  
    else if (string.IsNullOrEmpty(idRecipeBox.Text))  
    {  
        messageBoxShown = true;  
        Assert.IsTrue(messageBoxShown);  
    }  
}