**ГОСДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ САРАТОВСККОЙ ОБЛАСТИ**

**«ВОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»**

Специальность: 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
|  | «Утверждаю»  И.О. Зам. директора по учебной работе  /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Игнатьева Л.В.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

**ЗАДАНИЕ**

НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА

студенту группы ИТ-21 Шляпникову Дмитрию Алексеевич

Тема курсового проекта: **«Разработка программного модуля городской аптечной сети»**

|  |
| --- |
| Содержание курсового проекта |

1. Введение………………………………………………………………………………………..2
2. Сбор требований программного обеспечения,………………………………………………4

3. Разработка и оформление ТЗ (технического задания)………………………………………..7

4. Реализация ……………………………………………………………………………………….8

4.1. Выбор языка программирования и обоснование IDE…………….………………………….8

4.2. Написание кода………………………………………………………………………………...11

4.3. Дизайн и интерфейс……………………………………………………………………………17

5. Создание сборки………………………………………………………………………………… 19

6. Интервью с разработчиком……………………………………………………………………....19

7. Известные аналоги………………………………………………………………………………..21

8. Заключение………………………………………………………………………………………..22

9. Список используемых источников ……………………………………………………...………26

Дата выдачи задания: \_\_.\_\_\_\_\_\_. 2024 г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_преподаватель Мифтахов Н.И.

**1.Введение**

Городской аптечной сети программный модуль представляет собой приложение, которое предназначено для поиска информации о лекарствах в городской аптечной сети. Оно позволяет пользователям быстро и удобно находить необходимые им препараты по их названию.

Приложение должно имеет простой и интуитивно понятный интерфейс, который позволяет легко вводить название лекарства в специальное поле и осуществлять поиск по базе данных.

Цель разработки данного программного модуля - обеспечить жителей города быстрым и удобным доступом к информации о лекарствах, что позволит им принимать более информированные решения при покупке и использовании препаратов.

**Актуальность**

Приложение для поиска лекарств является актуальным и необходимым инструментом в современном мире, поскольку оно обеспечивает быстрый и удобный доступ к информации о лекарствах. С развитием технологий и увеличением объема информации, поиск необходимых лекарств.

Приложение для поиска лекарств позволяет пользователям быстро найти информацию о нужном лекарстве, включая его описание. Это экономит время и усилия пользователей, а также снижает вероятность ошибок, связанных с неправильным использованием или покупкой неподходящих лекарств.

В целом, приложение для поиска лекарств представляет собой актуальное и необходимое решение для обеспечения доступности и безопасности медицинской информации, а также оптимизации процесса поиска .

Кроме того, такое приложение может также предоставлять пользователю дополнительную информацию о лекарствах, такую как инструкции по применению, побочные эффекты, взаимодействия с другими препаратами и прочие полезные сведения. Это помогает повысить уровень осведомленности пользователей о принимаемых ими лекарствах и способствует безопасному и правильному их использованию.

Удобство использования подобного приложения также может оказаться полезным для пожилых людей и тех, кто имеет ограниченные возможности по посещению аптек лично. Благодаря онлайн доступу к информации о лекарствах, такие пользователи могут получить необходимую помощь и консультацию прямо из дома.

Наконец, разработка и совершенствование подобных программных модулей способствует повышению доступности медицинской информации для широкого круга пользователей и способствует современному подходу к заботе о своем здоровье и безопасности.

Развитие технологий и цифровизация здравоохранения сегодня играют ключевую роль в обеспечении доступности медицинской информации. Приложение для поиска лекарств является одним из инструментов, который способствует улучшению качества обслуживания и удовлетворения потребностей пользователей.

Одним из важных аспектов такого приложения является возможность получения актуальной информации об ассортименте лекарств в городской аптечной сети. Это позволяет пользователям быть в курсе последних обновлений, новых поступлений и изменениях в наличии препаратов. Тем самым, люди могут принимать информированные решения при выборе лекарств и экономить время при их приобретении.

Благодаря возможности оперативного поиска информации о лекарствах, приложение также способствует повышению безопасности покупок и использования медикаментов. Пользователи могут легко проверить соответствие препарата их потребностям, изучить инструкцию по применению, ознакомиться с побочными эффектами и взаимодействиями с другими лекарствами. Такой подход минимизирует риски ошибок при самолечении или использовании неподходящих препаратов.

С точки зрения удобства использования, приложение для поиска лекарств может предлагать дополнительные функции, такие как сохранение избранных лекарств, создание списка необходимых медикаментов, уведомления о сроке годности или необходимости повторной покупки. Это позволяет пользователям эффективно управлять своими лекарственными средствами и поддерживать контроль над своим здоровьем.

С учетом того, что многие люди могут испытывать затруднения с доступом к аптекам, особенно в ситуациях ограниченной подвижности или пандемий, онлайн приложение для поиска лекарств становится важным инструментом для обеспечения им своевременной медицинской помощи. Пожилые люди, люди с ограниченными возможностями или те, кто находится в домашнем изоляции, могут воспользоваться удобством онлайн доступа к информации о лекарствах и заботиться о своем здоровье без необходимости посещения аптеки.

В целом, разработка и использование приложения для поиска лекарств демонстрирует эффективный способ обеспечения доступности медицинской информации, повышения осведомленности пользователей о лекарствах и содействия безопасному и правильному использованию медикаментов. Это пример синергии между медицинской практикой и современными технологиями, которая способствует улучшению качества жизни и заботе о собственном здоровье.

**2.Сбор требований программного обеспечения.**

1. Проведем интервью с различными пользователями для выявления их потребностей и проблем, связанных с учетом медицинских данных.
2. Проанализируем существующие аналоги программных модулей для учета медицинских данных и выявим их преимущества и недостатки.
3. Проведем совещание с экспертами из области медицины и информационных технологий, чтобы обсудить основные задачи, которые должен решать разрабатываемый модуль.
4. Составим документ с требованиями к функциональности и интерфейсу разрабатываемого модуля на основе полученных данных.
5. Подготовим техническое задание для разработчиков и начнем проектирование и разработку модуля с учетом всех выявленных требований и предпочтений пользователей и экспертов.

Важно учитывать потребности и удобство использования приложения как для обычных пользователей, так и для специалистов (например, медицинских работников), которые могут быть заинтересованы в поиске лекарств.

**Пользовательский интерфейс:**

* Интуитивно понятный пользовательский интерфейс.
* Наличие поисковой строки для быстрого поиска нужного лекарства.

**Функциональность:**

* Возможность просмотра подробной информации о найденных лекарствах
* Возможность оценки и оставления отзывов о лекарствах.

**Постановка задачи**

**Логика работы приложения**

Приложение для поиска лекарств будет работать следующим образом:

1. Пользователь вводит название лекарства в строку поиска.
2. Приложение отправляет запрос к базе данных, содержащей информацию о лекарствах.
3. База данных возвращает результаты поиска, соответствующие запросу пользователя.
4. Приложение отображает результаты поиска: список лекарств с названиями, характеристиками.

**Документация по сбору требований для программы по поиску лекарств должна включать следующие пункты:**

1. Название программы: название должно четко отражать функциональность программы и быть привлекательным для пользователей.
2. Цель программы: опишите в общих чертах, для чего будет использоваться программа по поиску лекарств. Например, облегчение поиска нужных медикаментов в различных аптеках.
3. Описание пользователей: определите целевую аудиторию программы - врачи, пациенты, фармацевты и т.д. Опишите их основные потребности и ожидания от программы.
4. Функциональные требования: опишите основные функции, которые должны быть реализованы в программе. Например, поиск лекарств по названию, активному веществу, производителю, наличие в аптеках и т.д.
5. Нефункциональные требования: опишите такие требования, как удобство использования, быстрота поиска, надежность работы программы, безопасность данных и т.д.
6. Платформа и технологии: определите платформу, на которой должна работать программа (например, мобильное приложение, веб-приложение) и используемые технологии (языки программирования, базы данных и т.п.)
7. Дополнительные требования: опишите любые другие требования, которые могут быть важны для успешной реализации программы по поиску лекарств.
8. Важность требований: оцените важность каждого требования, чтобы определить их приоритеты в разработке программы.

Документация по сбору требований должна быть подробной и понятной для разработчиков, чтобы они могли точно определить все функциональности и особенности программы.

**Прототипирование:**

Для прототипирования программы по поиску лекарств можно воспользоваться специальными программами или онлайн-сервисами для создания прототипов интерфейсов. Следует разработать прототипы основных экранов приложения, чтобы проиллюстрировать функциональности и интерфейс программы.

Примеры экранов, которые могут быть включены в прототип:

1. Начальный экран с формой поиска лекарств.
2. Экран с результатами поиска: список лекарств с подробной информацией.
3. Экран с подробной информацией о выбранном лекарстве.
4. Экран с возможностью оценки и оставления отзывов о лекарстве.
5. Экран с настройками пользователя или другими дополнительными функциями.

Прототипирование поможет участникам проекта лучше представить, как будет работать приложение, и обсудить его основные характеристики до начала активной разработки. Это также позволит провести тестирование и получить обратную связь от пользователей на ранних этапах проекта.

После создания прототипов и получения обратной связи можно внести необходимые изменения в дизайн и функционал программы перед началом полноценной разработки.

Кроме сбора требований и прототипирования, важным этапом процесса разработки программного обеспечения является тестирование. Тестирование поможет проверить работоспособность и качество программы перед ее внедрением. Вот некоторые методы тестирования, которые могут быть применены для приложения по поиску лекарств:

Модульное тестирование: проверка отдельных модулей программы на корректность работы.

Интеграционное тестирование: проверка взаимодействия между разными модулями программы.

Системное тестирование: проверка работы системы в целом для удостоверения, что все функции выполняются правильно.

Пользовательское тестирование: тестирование программы конечными пользователями для оценки удобства использования, обнаружения ошибок и сбора обратной связи.

Также важно уделить внимание управлению изменениями и обновлениями программного обеспечения. После выпуска приложения важно следить за обратной связью пользователей, вносить улучшения на основе полученных данных и регулярно обновлять программу для обеспечения актуальности и безопасности.

Процесс сбора требований программного обеспечения для разработки модуля по учету медицинских данных включает важные этапы, описанные выше. Для успешной реализации проекта также требуется строгое управление проектом, создание плана разработки, учёт рисков и эффективное взаимодействие с командой разработчиков.

Управление проектом: • Составление и контроль выполнения плана проекта, включая определение этапов работы, выделение ресурсов, установление сроков и контроля прогресса. • Организация коммуникации между всеми участниками проекта для эффективного обмена информацией и решения проблем. • Мониторинг качества работы и соответствия результатов разработки установленным требованиям.

Управление рисками: • Идентификация рисков и разработка стратегий их управления. • Регулярное обновление и анализ реестра рисков для предотвращения негативных последствий и обеспечения успешного завершения проекта.

Взаимодействие с командой разработчиков: • Активное вовлечение членов команды во все этапы процесса разработки. • Поддержка команды в уточнении требований, обсуждении методов реализации и принятии ключевых решений.

Обращение к стандартам и моделям управления проектами, таким как PRINCE2 (Projects in Controlled Environments), Agile, Scrum и другим, может помочь оптимизировать процесс разработки и обеспечить высокое качество результата.

Также стоит обратить внимание на важность обратной связи с пользователями и постоянное совершенствование проекта после выпуска. Анализ данных использования, отзывов и предложений пользователей поможет определить направления для улучшений и обновлений продукта, что способствует удовлетворению потребностей пользователей и повышению конкурентоспособности программного продукта.

Непрерывное обучение и развитие для команды разработчиков также важны для повышения квалификации и эффективности работы. Курсы повышения квалификации, тренинги по новым технологиям и методикам разработки помогут команде оставаться в курсе последних тенденций и лучших практик в сфере программирования.

**3.Разработка и оформление ТЗ (технического задания).**

1.Данные о заказчике:

Название организации: Городская аптечная сеть "Здоровье"

Адрес: ул. Ленина, д. 10, г. Москва

Телефон: +7 (495) 123-45-67

ФИО представителя: Иванова Ольга Петровна

2. Данные о разработчике:

Название организации: Разработка программ

Адрес: ул. Советская, д. 32, г. Пугачев

Телефон: +7 (927) 139-28-99

ФИО представителя: Шляпников Дмитрий Алексеевич

3. Сроки разработки:

дату начала разработки:

дату окончания разработки:

4. Городской аптечной сети программный модуль представляет собой приложение, которое предназначено для поиска информации о лекарствах в городской аптечной сети. Оно позволяет пользователям быстро и удобно находить необходимые им препараты по их названию.

4.1. Название информационной системы: MedicineSearchApp

4.2. Описание функций, реализуемых системой:

Возможность ввода пользователем названия лекарства для поиска в базе данных.

Кнопка "Найти", которая по введенному пользователем названию лекарства ищет соответствующую информацию в базе данных.

Кнопка "Сбросить", которая сбрасывает поисковый запрос и отображенные результаты.

Отображение найденной информации о лекарствах (название и описание) на экране.

4.3. Название файла приложения, работающего с информационной базой: medicines.db

4.4. Название файла информационной базы: medicines.db

4.5. Описание архитектуры информационной системы (схема):

В информационной базе medicines.db используются две таблицы: search\_history и medicines.

search\_history: хранит историю поисковых запросов (search\_query)

medicines: хранит информацию о лекарствах, включая название (name) и описание (description)

4.6. Описание системных требований и конфигурации технических устройств:

Для запуска приложения необходим Python и библиотека PyQt5

Для работы с базой данных используется модуль sqlite3

В базе данных medicines.db хранится информация о лекарствах и история поисковых запросов

4.7. Описание интерфейса приложения для реализации функции заполнения личных данных Работника:

Пользователь вводит название лекарства в поле QLineEdit

По нажатию кнопки "Найти" (search\_button) происходит поиск лекарства в базе данных medicines.db

По нажатию кнопки "Сбросить" (reset\_button) пользователь может сбросить введенный запрос

Результаты поиска отображаются в виде QLabel (results\_label) ниже ввода запроса

База данных medicines.db заполняется некоторыми тестовыми данными о лекарствах: Аспирин, Парацетамол, Ибупрофен, Нурофен , и. т

**4.Реализация.**

**4.1 Выбор языка программирования и обоснование IDE.**

Python является одним из самых популярных и простых языков программирования, который используется для создания различных приложений. Он обладает большим количеством библиотек и инструментов, которые позволяют быстро и эффективно разрабатывать программы.

Для создания приложения для поиска лекарств Python подходит по следующим причинам:

1. Простота. Python имеет простой и понятный синтаксис, который позволяет разработчикам быстро освоить язык и начать создавать программы. Это особенно важно для начинающих разработчиков, которые только начинают свой путь в программировании.
2. Эффективность. Python обладает высокой производительностью и может использоваться для создания сложных приложений, таких как приложение для поиска лекарств. Он также имеет хорошую оптимизацию памяти, что позволяет создавать приложения, которые не занимают много ресурсов компьютера.
3. Большое количество библиотек. Для Python существует множество библиотек, которые облегчают разработку различных приложений. Например, для создания приложения для поиска лекарств можно использовать библиотеку Django, которая предоставляет инструменты для работы с базами данных и веб-разработки.
4. Открытый исходный код. Python является открытым программным обеспечением, что означает, что его исходный код доступен для всех желающих. Это позволяет разработчикам изучать код и использовать его для создания своих приложений.
5. Широкая поддержка сообщества Python имеет большое сообщество разработчиков, которое помогает новым пользователям освоить язык и решить возникающие проблемы. Это делает процесс разработки более комфортным и эффективным.
6. Кросс-платформенность. Python может работать на разных операционных системах, что делает его идеальным выбором для создания кросс-платформенных приложений, таких как приложение для поиска лекарств.
7. Гибкость. Python поддерживает различные стили программирования, такие как объектно-ориентированное, функциональное и процедурное программирование, что позволяет разработчикам выбирать подход, который лучше всего подходит для их проекта.
8. Многофункциональность. Python имеет широкий спектр возможностей и может использоваться для создания различных типов приложений, включая веб-сайты, игры, мобильные приложения, научные вычисления и многое другое. Это делает его подходящим инструментом для создания приложения для поиска лекарств с различной функциональностью.
9. Поддержка API. Python имеет хорошую поддержку для работы с API, что позволяет интегрировать различные веб-сервисы и базы данных в приложение для поиска лекарств, обеспечивая доступ к актуальным данным и информации.
10. Продвинутые инструменты. В Python существует множество инструментов для разработки и тестирования приложений, такие как Pytest, Flask и другие, которые помогают создавать качественный и стабильный софт.
11. Именно поэтому Python является отличным выбором для разработки приложения для поиска лекарств, который требует высокой производительности, эффективности и надежности.

**Выбор IDE**

PyCharm является одной из самых популярных интегрированных сред разработки (IDE) для языка программирования Python. Она предоставляет широкий спектр возможностей для разработки, такие как автодополнение кода, анализ кода на ошибки, отладка, визуализация данных и многие другие.

Выбор PyCharm для создания приложения для поиска лекарств обоснован следующими преимуществами:

* Удобство разработки: PyCharm имеет интуитивно понятный интерфейс, который упрощает процесс создания приложения. Она предлагает множество функций, которые ускоряют разработку, таких как автодополнение кода и анализ ошибок.
* Поддержка сообщества: PyCharm имеет активное сообщество пользователей и разработчиков, которые помогают решать возникающие проблемы и предлагают новые идеи для улучшения продукта.
* Широкий функционал: PyCharm поддерживает множество инструментов и библиотек для Python, что позволяет разработчикам легко создавать сложные приложения.
* Скорость работы: PyCharm работает быстро и эффективно, что важно для разработчиков, работающих над большими проектами.

Таким образом, PyCharm является отличным выбором для создания приложения для поиска лекарств, так как она предоставляет все необходимые инструменты для быстрой и эффективной разработки.

• Интеграция с другими инструментами: PyCharm легко интегрируется с другими инструментами разработки, такими как системы контроля версий (например, Git), виртуальные среды (например, Virtualenv) и менеджеры пакетов (например, pip), что обеспечивает удобство в работе и управлении проектом.

• Поддержка различных операционных систем: PyCharm доступен для различных операционных систем, таких как Windows, macOS и Linux, что делает его удобным для использования на разных платформах.

• Обучающие ресурсы: PyCharm предоставляет обширную документацию, видеоуроки и сообщества пользователей, что поможет новичкам быстро освоиться в IDE и начать разработку приложения.

• Поддержка разных версий Python: PyCharm поддерживает различные версии языка Python, что позволяет выбрать наиболее подходящую версию для разработки приложения.

• Бесплатная версия: PyCharm имеет бесплатную версию Community Edition, которая предоставляет все основные функции для разработки приложений на Python, что делает его доступным для широкого круга пользователей.

**4.2 Написание кода.**

Для начала необходимо импортировать нужные библиотеки для работы приложения такие как:

Для этого в Python библиотека импортируются с помощью ключевого слова import.

1. Библиотека sys в Python предоставляет функциональность для взаимодействия с интерпретатором Python и операционной системой.

Она используется для:

* Работы с аргументами командной строки: sys.argv позволяет получить список аргументов командной строки, переданных при запуске программы.
* Управления окружением выполнения: sys.path возвращает список директорий, в которых интерпретатор Python ищет модули при импорте, sys.modules содержит словарь уже импортированных модулей.
* Взаимодействия с операционной системой: sys.exit() завершает выполнение программы, sys.platform возвращает информацию об операционной системе, на которой выполняется код.
* Работы с потоками ввода-вывода: sys.stdin, sys.stdout и sys.stderr представляют стандартные потоки ввода-вывода и могут быть перенаправлены.
* Получения информации о текущем исполняемом коде: sys.executable указывает на путь к интерпретатору Python, sys.version возвращает информацию о версии интерпретатора.
* Управления памятью и выполнением гарбичной сборки: sys.getsizeof() позволяет получить размер объекта в байтах, sys.getrefcount() возвращает количество ссылок на объект.
* Это лишь некоторые из возможностей, предоставляемых модулем sys. Он является одним из стандартных модулей Python и широко используется для различных задач, связанных с интерфейсом между Python и операционной системой.

2. Библиотека PyQt5.QtWidgets предоставляет набор графических элементов пользовательского интерфейса (GUI) для создания приложений с помощью PyQt5. С ее помощью можно создавать окна, кнопки, поля ввода, списки, меню и другие элементы интерфейса, а также управлять их расположением на экране. Библиотека QtWidgets также позволяет обрабатывать события взаимодействия пользователя с интерфейсом, такие как нажатие кнопок или изменение значений полей. Кроме того, она поддерживает стилизацию виджетов с помощью каскадных таблиц стилей (CSS) и многоязыковую поддержку. В целом, библиотека PyQt5.QtWidgets является основной составляющей для создания графического пользовательского интерфейса в PyQt5.

3. Библиотека SQLite3 - это библиотека встраиваемой базы данных, которая позволяет создавать и управлять базами данных SQLite. SQLite3 является самодостаточной и независимой от сервера базой данных, что означает, что она не требует отдельного сервера баз данных для своего функционирования. Она обладает множеством преимуществ и широко используется в различных программных приложениях и встроена во множество операционных систем, таких как Android, iOS и многих других для хранения и доступа к данным.

**База данных для работы приложения**

SQLite - это мощная, компактная и самоуправляемая система управления базами данных (СУБД). Она представляет собой надежный и относительно быстрый способ организации хранения и извлечения данных. Основные особенности SQLite:

Компактность: SQLite представляет собой небольшую библиотеку, которую можно встроить в приложение или использовать как отдельный сервер базы данных. Это позволяет эффективно использовать ресурсы системы.

Нет необходимости в сервере: SQLite не требует установки и настройки отдельного сервера базы данных. Пользователь просто может создать файл базы данных и начать работу с ним.

Поддержка стандарта SQL: SQLite полностью соответствует стандарту SQL, включая большинство его расширений. Это означает, что вам доступны мощные возможности по запросам, фильтрации и сортировке данных.

Автономная работа: SQLite поддерживает полностью автономную работу, что означает, что приложение может продолжать работать даже при отсутствии подключения к базе данных.

Многопользовательская поддержка: SQLite поддерживает одновременный доступ к базе данных из нескольких потоков или процессов. Он использует механизмы блокировки, чтобы гарантировать целостность данных.

Кросс-платформенность: SQLite доступен для большинства платформ - Windows, macOS, Linux, iOS, Android и многих других.

Функционал SQLite включает в себя:

* Создание и управление базами данных с помощью языка SQL.
* Выполнение запросов к базам данных для получения, вставки, обновления и удаления данных.
* Поддержка индексов для ускорения выполнения запросов.
* Транзакционная обработка данных для обеспечения целостности и надежности.
* Поддержка различных типов данных: целые числа, числа с плавающей запятой, строки, даты и другие.
* Встроенный механизм шифрования для защиты данных.
* Поддержка хранимых процедур и триггеров для автоматизации выполнения определенных операций.
* Возможность работы с базами данных в оперативной памяти или на диске.
* Поддержка репликации данных для создания резервных копий или синхронизации данных между несколькими устройствами.

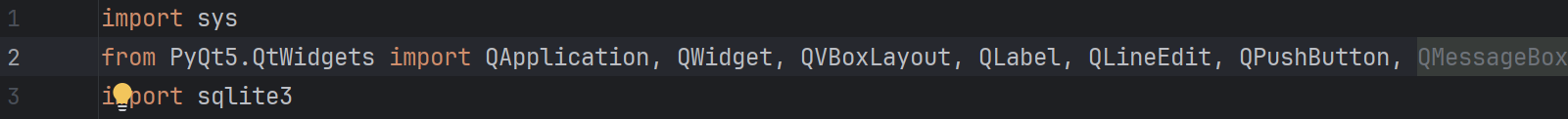
SQLite - это полноценная СУБД, которая предлагает множество возможностей для эффективной работы с данными в различных типах приложений.

SQLite также известен своей отличной производительностью при работе с небольшими и средними объемами данных. Его простота в использовании делает его популярным выбором для мобильных приложений, веб-сайтов, встроенных систем и других проектов, где требуется управление данными.

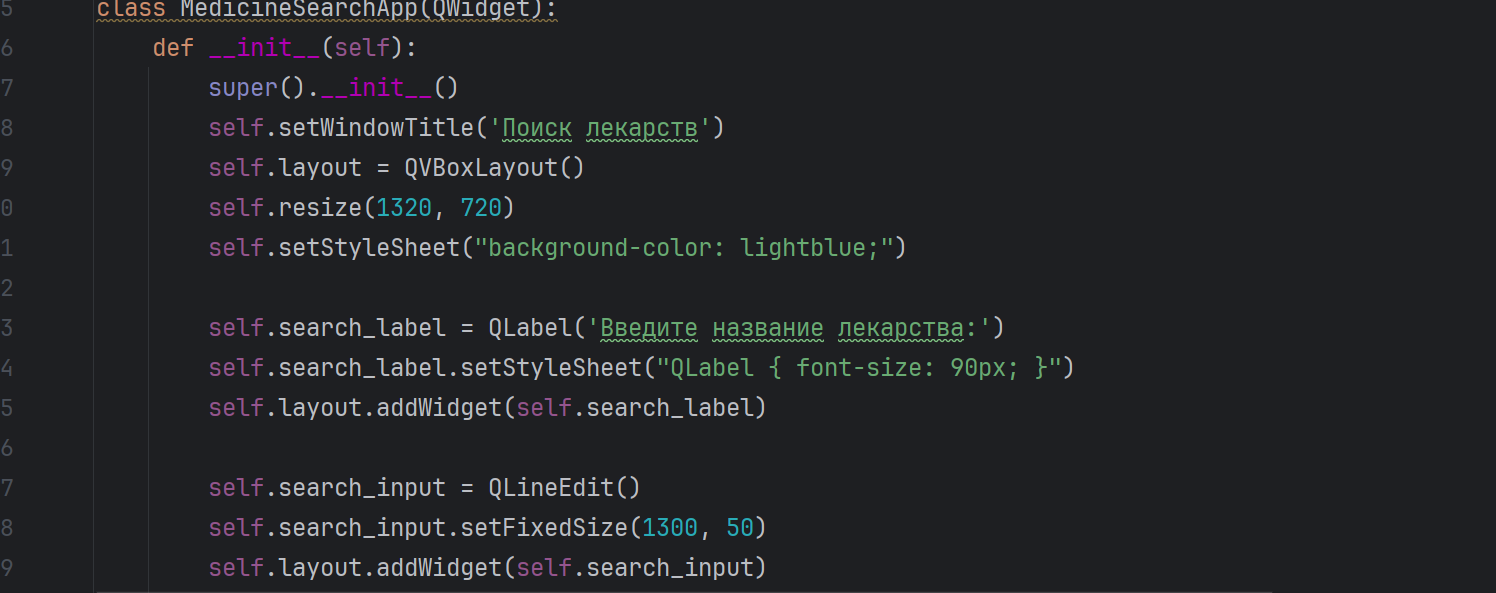
Кроме того, SQLite обладает хорошей поддержкой транзакций, что позволяет обеспечить целостность данных при выполнении сложных операций. Это делает его надежным инструментом для приложений, где важна безопасность и сохранность информации.

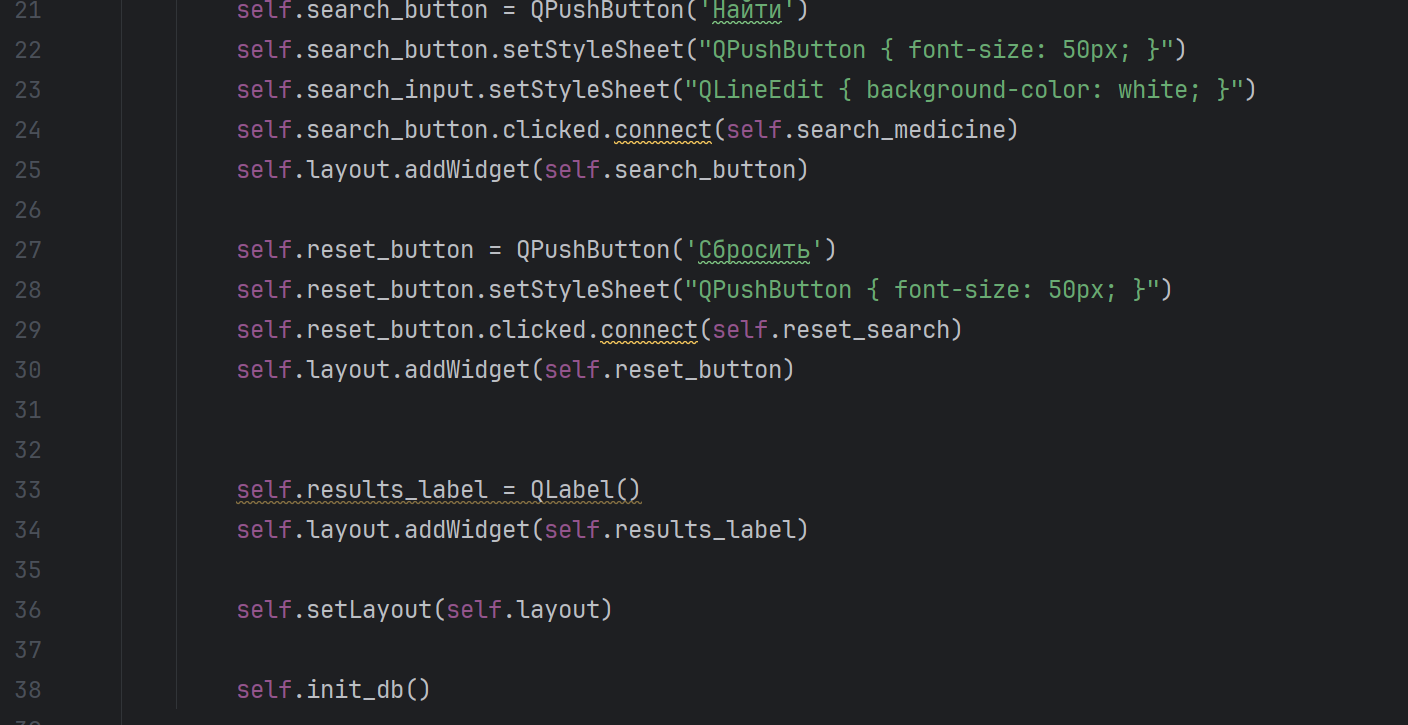
SQLite также обеспечивает удобные инструменты для резервного копирования и восстановления данных, что позволяет предотвратить потерю информации в случае сбоев или ошибок.

В целом, SQLite представляет собой отличное решение для разработчиков, которым нужна надежная и эффективная система управления базами данных для своих приложений.

****

После установки библиотек пишем код для приложения с графическим интерфейсом:



****

Этот код представляет собой графическое приложение для поиска лекарств. Класс MedicineSearchApp наследуется от QWidget и представляет основное окно приложения.

В конструкторе класса происходит инициализация оформления окна (заголовок, размер, цвет фона) и создание всех необходимых виджетов (метки, поле ввода, кнопки).

Метка search\_label отображает текст "Введите название лекарства" с использованием большого шрифта.

Поле ввода search\_input предназначено для ввода названия лекарства и имеет фиксированный размер.

Кнопка search\_button имеет надпись "Найти" и большой шрифт. При нажатии на эту кнопку вызывается метод search\_medicine, который выполняет поиск лекарства.

Кнопка reset\_button имеет надпись "Сбросить" и также вызывает связанный с ней метод reset\_search при нажатии.

Метка results\_label предназначена для отображения результатов поиска.

Компоновщик QVBoxLayout используется для упорядочивания и размещения всех виджетов в окне приложения.

Метод setLayout устанавливает компоновщик в качестве основного макета приложения.

Метод init\_db инициализирует базу данных, которая будет используется для поиска лекарств.

def init\_db(self):  
 conn = sqlite3.connect('medicines.db')  
 c = conn.cursor()  
 c.execute('''CREATE TABLE IF NOT EXISTS search\_history  
 (search\_query text)''')  
  
 c.execute('''CREATE TABLE IF NOT EXISTS medicines  
 (name text, description text)''')  
  
 c.execute("INSERT INTO medicines VALUES ('Аспирин', 'Противовоспалительное и обезболивающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Парацетамол', 'Обезболивающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Ибупрофен', 'Противовоспалительное и жаропонижающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Нурофен', 'Противовоспалительное и жаропонижающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Диклофенак', 'Противовоспалительное средство')")  
 c.execute("INSERT INTO medicines VALUES ('Но-шпа', 'Спазмолитик и обезболивающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Ибупрофен', 'Противовоспалительное и анальгетик')")  
 c.execute("INSERT INTO medicines VALUES ('Лоратадин', 'Противоаллергическое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Цетиризин', 'Противоаллергическое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Фенистил', 'Противоаллергическое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Ранитидин', 'Препарат против язвы желудка и двенадцатиперстной кишки')")  
 c.execute("INSERT INTO medicines VALUES ('Омепразол', 'Противоязвенное средство')")  
 c.execute("INSERT INTO medicines VALUES ('Панкреатин', 'Ферментный препарат для улучшения пищеварения')")  
 c.execute("INSERT INTO medicines VALUES ('Активированный уголь', 'Сорбент для отравлений и расстройств пищеварения')")  
 c.execute("INSERT INTO medicines VALUES ('Лоперамид', 'Противопроливное средство')")  
 c.execute("INSERT INTO medicines VALUES ('Линекс', 'Пробиотик для нормализации микрофлоры кишечника')")  
 c.execute("INSERT INTO medicines VALUES ('Амоксициллин', 'Антибиотик группы пенициллинов')")  
 c.execute("INSERT INTO medicines VALUES ('Азитромицин', 'Антибиотик группы макролидов')")  
 c.execute("INSERT INTO medicines VALUES ('Цефтриаксон', 'Антибиотик группы цефалоспоринов')")  
 c.execute("INSERT INTO medicines VALUES ('Активированный уголь', 'Сорбент для отравлений и расстройств пищеварения')")  
 c.execute("INSERT INTO medicines VALUES ('Лоперамид', 'Противопроливное средство')")  
 c.execute("INSERT INTO medicines VALUES ('Линекс', 'Пробиотик для нормализации микрофлоры кишечника')")  
 c.execute("INSERT INTO medicines VALUES ('Амоксициллин', 'Антибиотик группы пенициллинов')")  
 c.execute("INSERT INTO medicines VALUES ('Азитромицин', 'Антибиотик группы макролидов')")  
 c.execute("INSERT INTO medicines VALUES ('Цефтриаксон', 'Антибиотик группы цефалоспоринов')")  
 c.execute("INSERT INTO medicines VALUES ('Альфацин', 'Антибиотик группы тетрациклинов')")  
 c.execute("INSERT INTO medicines VALUES ('Флюконазол', 'Противогрибковое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Нистатин', 'Противогрибковое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Метронидазол', 'Противопротозойное и противомикробное средство')")  
 c.execute("INSERT INTO medicines VALUES ('Ацетилцистеин', 'Муколитическое средство')")  
 c.execute("INSERT INTO medicines VALUES ('Фенилэфрин', 'Сосудосуживающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Ксилометазолин', 'Сосудосуживающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Глицерин', 'Осмолитическое средство для очистки носа')")  
 c.execute("INSERT INTO medicines VALUES ('Нифедипин', 'Препарат для снижения артериального давления')")  
 c.execute("INSERT INTO medicines VALUES ('Метопролол', 'Бета-адреноблокатор для снижения артериального давления')")  
 c.execute("INSERT INTO medicines VALUES ('Атенолол', 'Бета-адреноблокатор для снижения артериального давления')")  
 c.execute("INSERT INTO medicines VALUES ('Эналаприл', 'Препарат для снижения артериального давления и лечения сердечной недостаточности')")  
 c.execute("INSERT INTO medicines VALUES ('Лизиноприл', 'Препарат для снижения артериального давления и лечения сердечной недостаточности')")  
 c.execute("INSERT INTO medicines VALUES ('Симастатин', 'Липидснижающее средство')")  
 c.execute("INSERT INTO medicines VALUES ('Феназепам', 'Препарат для снятия тревожности и нервного напряжения')")  
  
 conn.commit()  
 conn.close()  
  
 def search\_medicine(self):  
 medicine\_name = self.search\_input.text()  
 conn = sqlite3.connect('medicines.db')  
 c = conn.cursor()  
  
 c.execute("SELECT \* FROM medicines WHERE name=?", (medicine\_name,))  
 result = c.fetchone()  
 c.execute("INSERT INTO search\_history VALUES (?)", (medicine\_name,))  
 conn.commit()  
  
 if result:  
 self.results\_label.setText(f"Название: {result[0]}\nОписание: {result[1]}")  
 else:  
 self.results\_label.setText('Лекарство не найдено')  
  
 conn.close()  
  
 def reset\_search(self):  
 self.search\_input.clear()  
 self.results\_label.setText('')  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 window = MedicineSearchApp()  
 window.show()  
 sys.exit(app.exec\_())

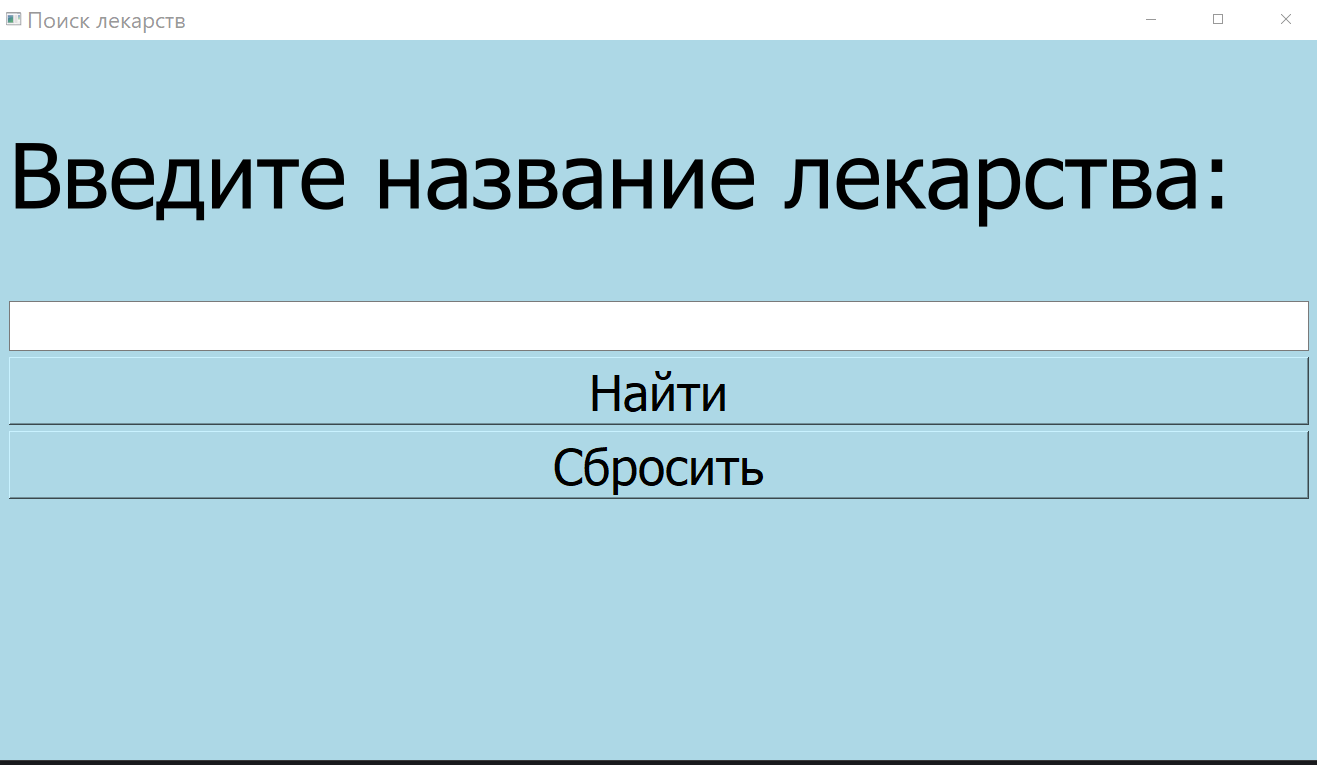
Этот код представляет собой простое приложение для поиска информации о лекарствах в базе данных.

1. Изначально в базу данных **medicines.db** добавляются данные о различных лекарствах с их описанием. В таблице **medicines** хранится информация о названии лекарства и его описании.
2. Пользователь может ввести название лекарства в текстовое поле и нажать кнопку поиска. Поиск лекарства происходит в базе данных, где происходит выборка данных по введенному названию лекарства.
3. Если лекарство найдено в базе данных, на экране отображается название и описание этого лекарства. Если лекарство не найдено, выводится сообщение "Лекарство не найдено".
4. Поиск лекарства также сохраняется в таблице **search\_history**, чтобы в дальнейшем можно было просмотреть историю поиска.
5. С помощью кнопки "Сбросить поиск" пользователь может очистить текстовое поле ввода и результаты поиска.
6. Приложение запускается с помощью кода в конце, который создает экземпляр приложения **MedicineSearchApp**, отображает его на экране и запускает цикл обработки событий приложения.

Таким образом, этот код позволяет пользователю искать информацию о лекарствах, сохранять историю поиска и удобно управлять поиском лекарств в базе данных.

**4.3. Дизайн и интерфейс.**

В верхней части окна располагается панель с названием "Поиск лекарств" и кнопками для управления окном (закрыть, развернуть или свернуть). Ниже находится заголовок "Введите название лекарства:", выполненный большими буквами. Под заголовком расположено поле для ввода текста, где пользователь может ввести название интересующего его лекарства. Под полем ввода находятся две кнопки: "Найти", с помощью которой можно начать поиск введенного лекарства, и "Сбросить", нажатие на которую скорее всего очистит поле ввода от введенного текста или сбросит результаты текущего поиска. Дизайн интерфейса минималистичный, цвета спокойные: синий, серый и белый.



**Плюсы данного дизайна и интерфейса включают в себя:**

1. Минимализм: Простота и минимализм дизайна делают интерфейс более понятным и удобным для пользователей, так как они могут легко сосредоточиться на основной задаче - поиске лекарств.
2. Ясная навигация: Расположение элементов, таких как панель управления, заголовок и кнопки, делает навигацию по приложению легкой и интуитивно понятной.
3. Хорошая видимость: Крупные буквы заголовка и понятные названия кнопок привлекают внимание пользователя, что помогает им быстро понять, что делать дальше.
4. Спокойные цвета: Использование спокойных цветов, таких как синий, серый и белый, способствует созданию приятной атмосферы и снижает нагрузку на глаза пользователя.
5. Понятные функции кнопок: Каждая кнопка имеет ясное название и предназначение, что позволяет пользователям легко понять, какую функцию они выполняют.
6. Удобство использования: Благодаря простоте и интуитивности дизайна, пользователи могут быстро и эффективно выполнять задачи по поиску лекарств без лишних препятствий.

Такой дизайн способствует удобству и эффективности использования приложения по поиску лекарств, что делает его привлекательным и функциональным для пользователей.

**5.Создание сборки.**

Для создания сборки вашего приложения с использованием PyInstaller более подробно следует выполнить следующие шаги:

Установка PyInstaller с помощью pip, если у вас его еще нет:

pip install pyinstaller

Перейдите в директорию с вашим скриптом Python, который содержит приложение для поиска лекарств.

Запустите PyInstaller, указав путь к вашему скрипту Python. Например:

pyinstaller --onefile your\_script.py

В данной команде:

--onefile означает, что будет создан одиночный исполняемый файл (один файл без дополнительных зависимостей) вашего приложения.

your\_script.py - это название вашего скрипта Python.

После выполнения команды PyInstaller создаст директорию dist, в которой будет находиться ваш исполняемый файл. Вы сможете найти его там и запустить на любой поддерживаемой платформе.

Таким образом, вы сможете удобно создать исполняемый файл вашего приложения для поиска лекарств, который можно запустить на других компьютерах без необходимости установки Python или других зависимостей.

**6.Интервью с разработчиком.**

**Можете ли вы объяснить цель этого приложения?**

Целью этого приложения является поиск информации о различных лекарствах, хранящейся в базе данных. Пользователи могут искать конкретное лекарство по его названию и получать подробную информацию о нем, включая описание.

**Как приложение хранит и получает информацию о лекарствах?**

Приложение использует базу данных SQLite под названием «medicines.db» для хранения информации о лекарствах. В базе данных есть две таблицы: «search\_history» для хранения поисковых запросов и «лекарства» для хранения названий и описаний лекарств. Когда пользователь ищет лекарство, приложение запрашивает таблицу «лекарства» на основе введенных данных и извлекает информацию.

**Можете ли вы объяснить функцию поиска более подробно?**

Когда пользователь вводит название лекарства в поле ввода поиска и начинает поиск, приложение запрашивает базу данных, чтобы найти совпадение. Если подходящее лекарство найдено, приложение отображает название и описание лекарства. Если совпадение не найдено, отображается сообщение о том, что лекарство не найдено.

**Что происходит, когда пользователь сбрасывает поиск?**

Когда пользователь сбрасывает поиск, приложение очищает поле ввода поиска и метку результатов, чтобы можно было выполнить новую операцию поиска. Это помогает предоставить пользователю понятный интерфейс для выполнения нового поиска.

**Как разрабатывается графический интерфейс пользователя (GUI)**

**приложения?**

Приложение использует платформу PyQt для создания графического интерфейса. Он определяет экземпляр QApplication, создает окно для MedicineSearchApp и настраивает необходимые компоненты, такие как поле ввода поиска, метка результатов и кнопки для функций поиска и сброса.

**Какие будущие улучшения можно внести в это приложение?**

Будущие улучшения могут включать добавление более продвинутых функций поиска, улучшение визуализации результатов поиска, реализацию операций CRUD для управления данными о лекарствах и интеграцию с внешними API для получения информации о лекарствах в режиме реального времени.

**Как вы обеспечиваете согласованность и надежность данных в приложении?**

Согласованность и надежность данных поддерживаются за счет использования транзакций для фиксации изменений в базе данных только после успешного выполнения запросов. Это помогает предотвратить повреждение данных и гарантировать, что история поиска и сведения о лекарствах точно сохраняются и извлекаются.

**Планируете ли вы в будущем расширить базу данных новыми лекарствами?**

Да, в будущем планируется расширить базу данных дополнительными лекарствами, чтобы предоставить пользователям более полную функциональность поиска. Это потребует добавления новых названий лекарств вместе с их описаниями для расширения информации, доступной в приложении.

**Как вы обрабатываете потенциальные ошибки или исключения в приложении?**

Приложение использует механизмы обработки ошибок, такие как блоки try-кроме, для обнаружения и обработки любых потенциальных ошибок, которые могут возникнуть во время операций с базой данных или взаимодействия с пользователем. Грамотная обработка исключений делает приложение более удобным для пользователя.

**Что вдохновило вас на создание этого приложения?**

Необходимость предоставить пользователям удобный способ поиска информации о лекарствах и расширить свои знания о различных лекарственных средствах вдохновила на создание этого приложения. Его цель — упростить процесс доступа к информации о лекарствах для людей, ищущих информацию, связанную со здоровьем.

**7.Известные аналоги.**

1. Приложение "Medscape" - предоставляет информацию о различных лекарственных препаратах, их назначении, дозировке, побочных эффектах и взаимодействии с другими препаратами.
2. Приложение "Drugs.com" - позволяет найти информацию о лекарственных препаратах, их инструкциях по применению, взаимодействии с другими препаратами и возможных побочных эффектах.
3. Приложение "Medicines Complete" - предоставляет информацию о различных лекарственных препаратах, их составе, назначении, дозировке и побочных эффектах.
4. Приложение "Epocrates" - содержит информацию о различных лекарственных препаратах, их назначении, дозировке, побочных эффектах и взаимодействии с другими препаратами.
5. Приложение "GoodRx" - помогает найти информацию о лекарственных препаратах, их ценах, ближайших аптеках и возможных скидках и купонах на покупку препаратов.
6. Приложение "1mg" - предлагает информацию о различных лекарственных препаратах, их назначении, составе, побочных эффектах и возможности заказать препараты онлайн.
7. Приложение "Medscape Reference" - предоставляет информацию о различных лекарственных препаратах, их инструкциях по применению, взаимодействии с другими препаратами и побочных эффектах.
8. Приложение "MediWell" - предоставляет информацию о лекарствах, включая их действие, показания к применению, противопоказания, дозировки и возможные взаимодействия.
9. Приложение "MediSafe" - помогает пациентам организовать прием лекарств, напоминает о приеме доз и предупреждает о возможных взаимодействиях между препаратами.
10. Приложение "Lexicomp" - содержит обширную базу данных по лекарствам, включая информацию о дозировках, побочных эффектах, фармакологии и взаимодействиях.
11. Приложение "Pill Identifier" - помогает идентифицировать лекарства по их внешнему виду, маркировке и цвету, предоставляя информацию о них.
12. Приложение "Prescription Checker" - позволяет проверить название лекарства и его детали, предоставляя информацию о применении, побочных эффектах и взаимодействиях.
13. Приложение "DrugStars" - поощряет пациентов за принятие лекарственных средств правильно, предоставляя информацию о препаратах и вознаграждения за соблюдение курса лечения.

**8.Заключение.**

Python имеет простый и понятный синтаксис, что облегчает разработку приложений.

Python поддерживает работу с базами данных, что позволит нам хранить информацию о лекарствах и осуществлять поиск по ней.

Python имеет обширную библиотеку для разработки графического интерфейса, что позволит нам создать удобный и интуитивно понятный интерфейс для пользователей.

Python имеет широкое сообщество разработчиков, что обеспечивает доступ к множеству ресурсов и поддержку.

Python является переносимым, т.е. приложение, разработанное на Python, может быть запущено на разных операционных системах.

Таким образом, использование Python для разработки приложения для поиска лекарств позволит нам создать надежное, быстрое и удобное приложение для пользователей.

В заключение, разработка городской аптечной сети программного модуля для поиска информации о лекарствах является актуальной и необходимой задачей. Приложение обеспечивает быстрый и удобный доступ к информации о лекарствах, что помогает пользователям принимать более информированные решения при покупке и использовании препаратов.

Использование приложения позволяет сократить время и усилия пользователей при поиске необходимых лекарств, а также повысить безопасность использования препаратов. Подходящим языком программирования для реализации данного модуля является Python, благодаря его простоте и эффективности.

Таким образом, разработка городской аптечной сети программного модуля для поиска информации о лекарствах будет полезной и удовлетворит потребности пользователей в удобной и информативной системе поиска лекарств.

Разработка приложения для поиска лекарств в городской аптечной сети играет важную роль в обеспечении доступности и удобства получения информации о медикаментах для жителей города. Благодаря такому приложению пользователи могут быстро находить необходимые им препараты, получать подробную информацию о них и делать более обоснованные выборы при покупке и применении лекарств.

Внедрение такого приложения обеспечит оптимизацию процесса поиска лекарств, снижение ошибок при выборе препаратов и повышение уровня информированности пользователей. Технологии, такие как Python, SQLite и PyQt5, позволят создать надежное и производительное приложение с удобным интерфейсом.

Дальнейшее развитие приложения может включать расширение базы данных новыми лекарственными средствами, улучшение функциональности поиска, добавление возможности оценки и отзывов о препаратах, а также интеграцию с внешними API для обновления данных в реальном времени.

Создание городской аптечной сети программного модуля для поиска информации о лекарствах - это важный шаг в направлении улучшения доступности медицинской информации и повышения инновационности в здравоохранении.

Кроме того, развитие функционала приложения для поиска лекарств может также включать интеграцию с системой уведомлений для напоминания о приеме препаратов, возможность составления персонализированных списков лекарств для удобства хранения и заказа, а также добавление функции сравнения цен на медикаменты в разных аптеках.

Последующим шагом в развитии данного приложения может быть расширение его использования на уровень области или даже страны, чтобы обеспечить доступность и удобство поиска лекарств для большего количества пользователей.

Создание современного и инновационного приложения для поиска лекарств в городской аптечной сети демонстрирует важность использования технологий в сфере здравоохранения и стремление к повышению качества жизни горожан.

Дальнейшее развитие приложения также может включать интеграцию с системами электронной медицинской истории, что позволит пользователям получать более полную информацию о взаимодействии лекарств с их медицинской историей и поддерживать более актуальные записи о принимаемых препаратах.

Потенциал использования искусственного интеллекта в развитии приложения для поиска лекарств также является значительным. Можно рассмотреть возможность внедрения системы рекомендаций на основе персональных данных пользователей, что позволит предоставлять более индивидуальные и точные советы о лекарствах.

Одним из важных аспектов развития приложения для поиска лекарств будет обеспечение безопасности и конфиденциальности данных пользователей. Реализация механизмов шифрования и безопасной аутентификации поможет предотвратить утечку личной информации о пользователях и обеспечить защиту их данных.

Также важно уделить внимание разработке мобильной версии приложения, что позволит пользователям легко и удобно пользоваться функционалом поиска лекарств непосредственно с мобильных устройств. Мобильная версия приложения может стать удобным инструментом для пользователей в поиске и заказе лекарств даже вне дома.

Развитие функционала приложения также может включать интеграцию с системами доставки лекарств, что позволит пользователям быстро и удобно получать заказанные препараты непосредственно на дом. Это улучшит обслуживание и сделает процесс получения нужных лекарств еще более удобным для пользователей.

Важным шагом в развитии приложения можно также считать внедрение функционала мониторинга состояния здоровья пользователя на основе данных о принимаемых лекарствах и их воздействии на организм. Это поможет пользователям лучше понимать влияние лекарств на их организм и принимать более осознанные решения по их использованию.

Интеграция с системами медицинского консультирования или советов специалистов также может стать важным направлением развития приложения, что позволит пользователям получить профессиональные рекомендации и советы по применению лекарств и сохранению здоровья.

С учетом растущего интереса к здоровому образу жизни и самостоятельной заботе о здоровье, развитие приложения для поиска лекарств может способствовать более широкому распространению принципов самопомощи и бережливому отношению к собственному здоровью.

В целом, развитие и усовершенствование приложения для поиска лекарств в городской аптечной сети имеет большой потенциал для улучшения доступности медицинской информации, повышения уровня информированности пользователей и обеспечения комфортного и безопасного использования лекарственных средств.

Инновационные технологии в медицине являются ключевым фактором в улучшении качества оказания медицинской помощи и обеспечении лучших результатов лечения для пациентов. Развитие приложения для поиска лекарств в городской аптечной сети представляет собой важный шаг в направлении улучшения доступности медицинской информации и повышения качества услуг в медицинской сфере.

Одним из потенциальных направлений развития приложения является интеграция с системами искусственного интеллекта для анализа данных о потребностях пользователей и предоставления персонализированных рекомендаций по выбору лекарств. Такой подход позволит повысить эффективность использования приложения и удовлетворить индивидуальные потребности каждого пользователя.

Кроме того, возможным дальнейшим улучшением приложения является добавление функционала для мониторинга побочных эффектов лекарственных средств, что позволит пользователям будет получать информацию о возможных рисках и осложнениях при применении определенных препаратов. Эта функция поможет пользователям принимать информированные решения относительно своего лечения и минимизировать возможные негативные последствия.

Кроме того, такие инновации, как разработка возможности для онлайн консультаций с фармацевтами и врачами через приложение, могут увеличить доступность и качество медицинских консультаций для пользователей. Пользователи смогут получать квалифицированные рекомендации и советы по применению лекарств непосредственно через приложение, что существенно улучшит опыт пользования и повысит доверие к приложению.

Кроме того, интеграция с системами электронной медицинской истории может стать ключевым шагом для обеспечения безопасности и надежности данных о приеме лекарств. Пользователи смогут иметь доступ к своей полной медицинской истории через приложение, что обеспечит более информированные решения относительно своего здоровья и поможет взаимодействовать с врачами и фармацевтами на более осознанном уровне.

Таким образом, дальнейшее развитие приложения для поиска лекарств в городской аптечной сети включает в себя внедрение инновационных технологий, улучшение пользовательского опыта, обеспечение безопасности и конфиденциальности данных, а также расширение функционала для удовлетворения разнообразных потребностей пользователей.

Конечная цель развития такого приложения - обеспечение легкого доступа к информации о лекарствах, повышение информированности пользователей, обеспечение безопасного и эффективного использования лекарственных средств, а также улучшение качества здравоохранения в целом.

**9.Список используемых источников.**

.Библиотека PyQt5.QtWidgets - <https://pythonworld.ru/gui/pyqt5firstprograms.html#:~:text=PyQt5%20-%20это%20набор%20Python,можно%20с%20официального%20сайта%20библиотеки>

2.База данных sqlite - <https://timeweb.cloud/tutorials/sqlite/rukovodstvo-po-nastrojke-sqlite>

3.Библиотека sys - <https://all-python.ru/osnovy/sys.html>

4.Библиотека SQLite3 - <https://blog.skillfactory.ru/glossary/sqlite/>