

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

Проектирование медицинской информационной системы

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы
направление 09.03.03 — Прикладная информатика

механико-математического факультета

Бахарева Сергея Николаевича

Научный руководитель
к.ю.н., доцент

подпись, дата

Р.В. Амелин

Зав. кафедрой
д.ф.-м.н., профессор

подпись, дата

Л.Ю. Коссович

Саратов 2024

Введение. Экстренные хирургические операции играют критическую роль в спасении жизней пациентов, чье состояние требует немедленного медицинского вмешательства. Их важность заключается в том, что такие операции могут спасти жизнь пациента, предотвратить осложнения и последствия тяжелых состояний, а также улучшить прогнозы выздоровления. Экстренные операции могут быть необходимы в случаях травм, острой недостаточности органов, кровотечения, инфекций и других состояний, требующих немедленного вмешательства хирурга.

Для планирования экстренной хирургической операции врачам нужно знать, в каком состоянии находится пациент, имеются ли у него осложнения после проведения других процедур или в результате заболевания. Также необходимо учитывать и такие данные, как группа крови пациента и резус-фактор. Для определения некоторых параметров требуется продолжительное время, которого может не хватать, поэтому собирать их нужно заранее на приеме в поликлинике. Хранить подобную информацию можно в базе данных.

Целью настоящей работы является создание базы данных для хранения информации о параметрах пациента, необходимых именно для предоперационного планирования экстренных хирургических операций.

В связи с поставленной целью необходимо решить следующие задачи:

1. Изучить предметную область: найти параметры, необходимые для предоперационного планирования экстренных хирургических операций.
2. Спроектировать и создать базу данных для рассмотрения необходимых для предоперационного планирования параметров.
3. Создать интерфейс для чтения данных необходимых для предоперационного планирования параметров, и для работы с общими данными пациента.

Структура работы. Основная часть состоит из 3 разделов:

- Подбор и изучение параметров, необходимых для проведения предоперационного планирования экстренного вмешательства;

- Обоснование выбора технологий, проектирование и реализация базы данных;
- Обоснование выбора технологий, проектирование и реализация интерфейса для работы с базой данных.

В первом разделе была рассмотрена предметная область и изучены параметры, необходимые для планирования.

В качестве параметров были выбраны следующие анализы и обследования: общий анализ крови, общий анализ мочи, коагулограмма, группа крови и резус-фактор пациента, наличие или отсутствие ранее перенесенных гемотрансфузий, наличие аллергических реакций на медицинские препараты, наличие хронических заболеваний внутренних органов, последствия ранее перенесенных хирургических вмешательств, наличие вредных привычек, информация о телосложении пациента (включая индекс массы тела), данные о липидном профиле, наличие инфекционных заболеваний и их последствий. Было изучено, что представляют собой вышеназванные параметры, что в них включается, в каких единицах измеряются, в пределах каких значений находится их норма. Каждый из параметров будет либо входить в состав таблицы базы данных, либо образовывать новую.

Во втором разделе описывается выбор технологий, процессы проектирования и реализации базы данных.

Для проектирования ER-диаграммы и диаграммы прецедентов был выбран ресурс Draw.io – инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Сервис распространяется на бесплатной основе с открытым исходным кодом. Draw.io обладает богатым набором функций для визуализации большинства задач пользователя. Одними из функций сервиса являются возможность создания ER-диаграмм и диаграмм прецедентов, позволяющих описывать концептуальные схемы предметной области. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Достоинствами Draw.io можно

назвать доступность, возможность построения диаграмм с помощью элементов интерфейса, быстрая скорость работы, удобный импорт и экспорт

Построение ER-диаграмм, с одной стороны, процесс не особо долгий, однако часто приходится возвращаться к уже созданным схемам. В таком случае Draw.io позволяет делать это достаточно быстро и без лишних неудобств.

В качестве средства проектирования базы данных была выбрана pgAdmin3 – графический клиент, инструмент управления, работающий с базой данных PostgreSQL.

К числу преимуществ pgAdmin3 можно отнести: интуитивный интерфейс, кросс-платформенность, возможность создавать и редактировать SQL-запросы с помощью графического редактора.

Таким образом, pgAdmin3 является вполне удобным средством создания и заполнения базы данных.

PostgreSQL – это объектно-реляционная система баз данных с открытым исходным кодом. Объектная, потому что поддерживает некоторые технологии объектно-ориентированного подхода, когда программа представлена в виде совокупности объектов, относящихся каждый к своему классу, а классы выстроены в иерархию наследования. Объекты, классы и наследование реализованы и в структуре баз данных. Помимо традиционных типов данных также поддерживает неструктурированные и перечисляемые.

К достоинствам PostgreSQL можно причислить следующие особенности: надежность и целостность данных, наличие богатого набора функций, открытый исходный код проекта, достаточно высокую популярность.

Таким образом, PostgreSQL является отличным выбором для использования в различных проектах.

Если говорить о самой системе, то база данных будет представлять собой электронную карту пациента, состоящую из двух частей: общей и специальной. В первую часть будет записываться общая информация о пациенте: ФИО, дата рождения, адрес места жительства и так далее. Вторая часть, медицинская,

будет содержать именно те медицинские показания и параметры, которые должны представлять интерес для оперирующих хирургов в том случае если пациент будет срочно нуждаться в хирургическом вмешательстве после какого-либо несчастного случая, повлекшего за собой срочную госпитализацию. При проектировании базы данных электронная карта будет представлять собой множество таблиц.

Отличием создаваемой системы от уже существующих является сфокусированность на параметрах, необходимых для предоперационного планирования именно экстренных вмешательств. Такие системы, как "МедЭксперт", "ИнфоМед", "1С: Медицина" и "Медиум", выдают информацию о результатах осмотров, не имеющих значения для операций, например, у оториноларинголога. В случаях экстренных вмешательств важно действовать быстро, а наличие чересчур сложного интерфейса может этому воспрепятствовать, поэтому было решено оставить только результаты тех ранее проведённых осмотров или операций, которые играют важную роль в конкретной ситуации.

Всех людей, использующих базу данных, можно поделить на 3 роли, различающиеся между собой: Регистратор, Врач и Общий хирург (рисунок 1). Разница между ними будет заключаться в том, как и с какими данными они смогут взаимодействовать.

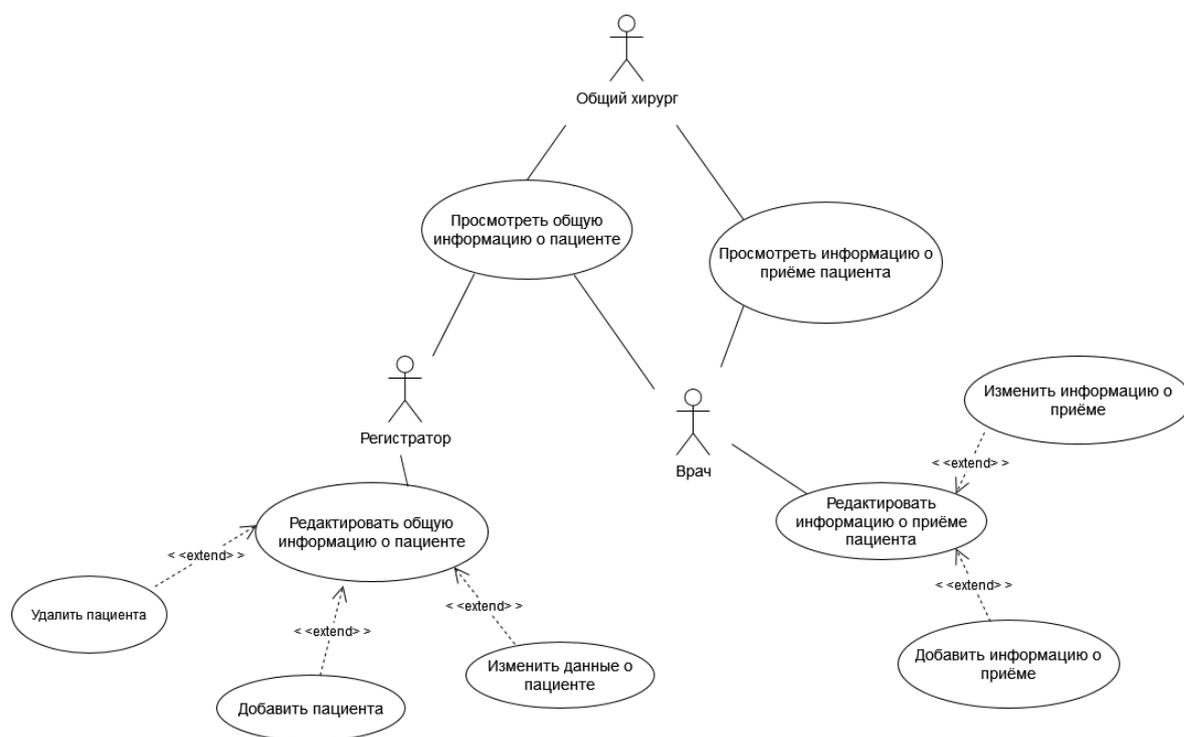


Рисунок 1 – Диаграмма прецедентов

«Регистратор» может работать с первой частью электронной карты пациента, с его данными. «Врач» - специалист, который будет взаимодействовать с данными при приёме или операции. «Общий хирург» – это действующее лицо, которое планирует и проводит экстренные операции. Именно на этого актора и ориентирована база данных.

В базе данных будет реализовано множество таблиц (рисунок 2), представляющих собой электронную карту пациента.

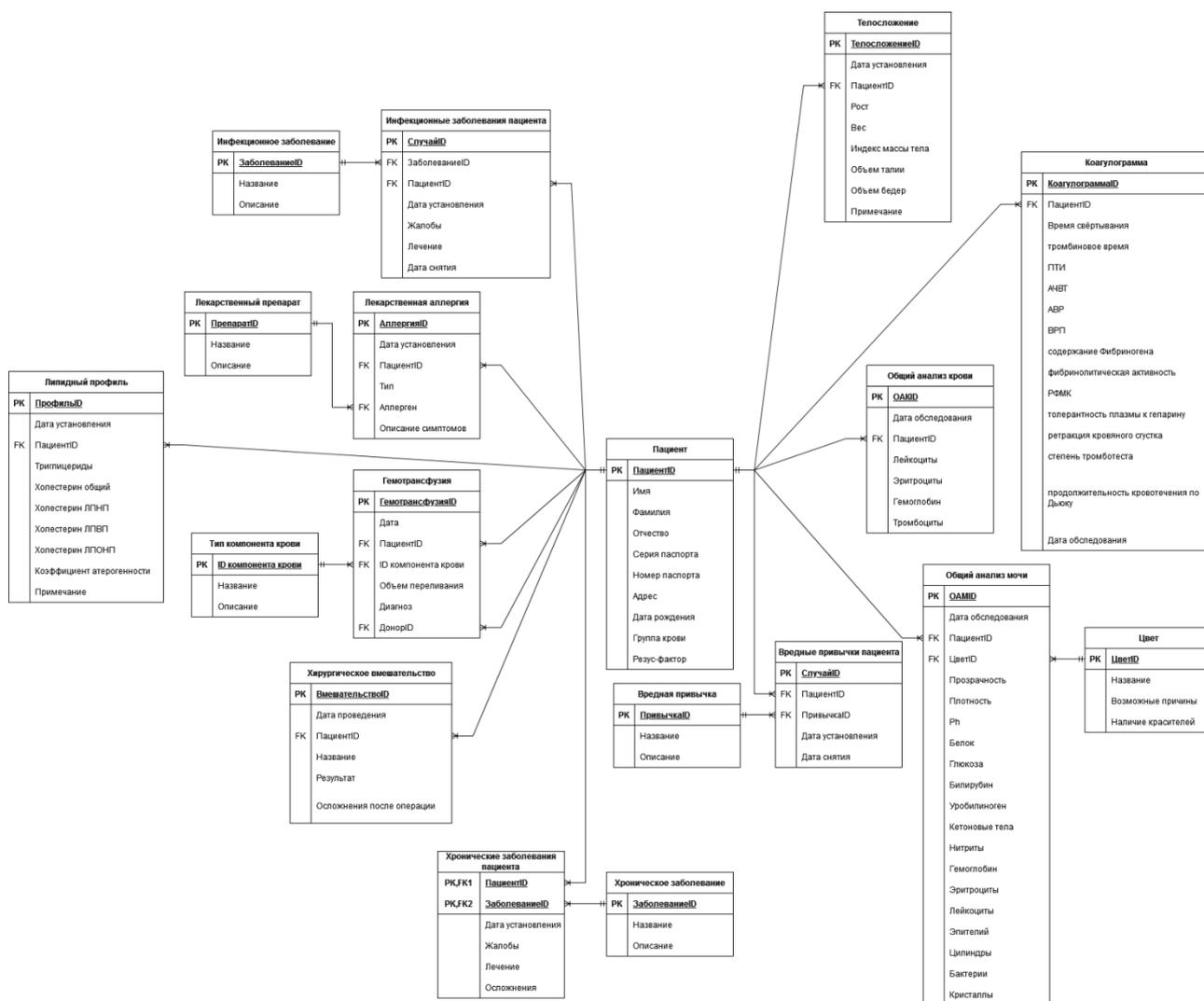


Рисунок 2 – ER-диаграмма

Далее в работе рассматривается каждая таблица с точки зрения построения базы данные: выбора типов данных и некоторых параметров.

Третий раздел описывает этапы разработки графического интерфейса для базы данных: выбор технологии, создание классов, программирование.

Диаграмма классов была создана с помощью интернет сервиса PlantUML – универсального инструмента, позволяющего быстро и просто создавать широкий спектр диаграмм. В отличие от Draw.io, PlantUML создаёт схемы с помощью использования UML. Это позволяет легче строить более сложные диаграммы, на построение которых с помощью графического интерфейса понадобилось бы слишком много лишних движений.

Достоинства PlantUML: доступность, быстрое изменение деталей достаточно сложных схем, удобный экспорт, автоматическое размещение

таблиц.

Таким образом, можно сделать вывод: PlantUML позволяет быстро построить и исправить недостатки в таблицах, однако внешний вид из-за автоматизации может быть не очень приятным.

Для создания интерфейса к базе данных был использован язык программирования Python – интерпретируемый высокоуровневый язык программирования с динамической строгой типизацией и автоматическим управлением памятью.

Преимуществами Python можно назвать: простоту и читаемость кода, наличие множества библиотек, кросс-платформенность, удобную интеграцию с SQL и активное сообщество.

Для реализации интерфейса был использован модуль PyQt5 – библиотека, которая позволяет использовать платформу Qt framework в Python. Сам Qt изначально был написан на C++, что позволило создавать приложения намного быстрее, не жертвуя при этом значительной частью производительности.

Преимущества PyQt5: широкая документация и сообщество, удобная интеграция с PostgreSQL, поддержка множества элементов интерфейсов и хорошая производительность, делающая его подходящим для приложений, требующих быстрого обновления пользовательского интерфейса.

Для реализации некоторых запросов (удаление, изменение, добавление данных) был выбран модуль psycopg2 – один из самых популярных адаптеров PostgreSQL для языка Python.

Преимущества psycopg2: стабильная работа с PostgreSQL, высокая производительность при отправке запросов в базу данных, хорошая документация.

Для работы были также использованы две дополнительные библиотеки: simple-settings и sys.

Первая представляет собой простой способ управления настройками проекта. Она была вдохновлена системой настроек Django. Simple-settings была использована для обеспечения интерфейса соединением. Все настройки

соединения хранились в отдельном файле, при запуске программы они считывались с помощью этой библиотеки. Стоит отметить её стабильную работу.

Модуль `sys` обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором Python. Является необходимым элементом при разработке приложения с использованием PyQt5, с помощью него приложение запускается. Для этого используется `sys.argv`, передающий список аргументов командной строки сценарию Python. Модуль `sys` является встроенным, соответственно, стабильная его работа гарантирована.

В качестве среды для разработки был выбран PyCharm 2021.2.1 Community edition. Является бесплатной IDE с открытым исходным кодом, разработанной компанией JetBrains. PyCharm включает в себя множество возможностей: помощь в написании кода (автодополнение текста, выделение неиспользуемых подключенных модулей, подсказки, автоматическое переформатирование файла), быстрая навигация и поиск, работа с Git и GitHub, древовидная структура отображения проекта и многое другое.

В качестве преимуществ PyCharm можно отметить: удобное взаимодействие с кодом, автоматическое создание виртуальной среды вокруг проекта, возможность импортировать модули используя интерфейс IDE и не прибегая к командной строке, возможность настройки интерфейса, удобный импорт и экспорт, возможность создания и записи обычных текстовых файлов прямо внутри IDE, возможность использования отладчика для пошагового анализа кода.

Таким образом, PyCharm является весьма мощным средством для разработки приложений на языке Python.

При разработке приложения была создана следующая архитектура классов (рисунок 3).

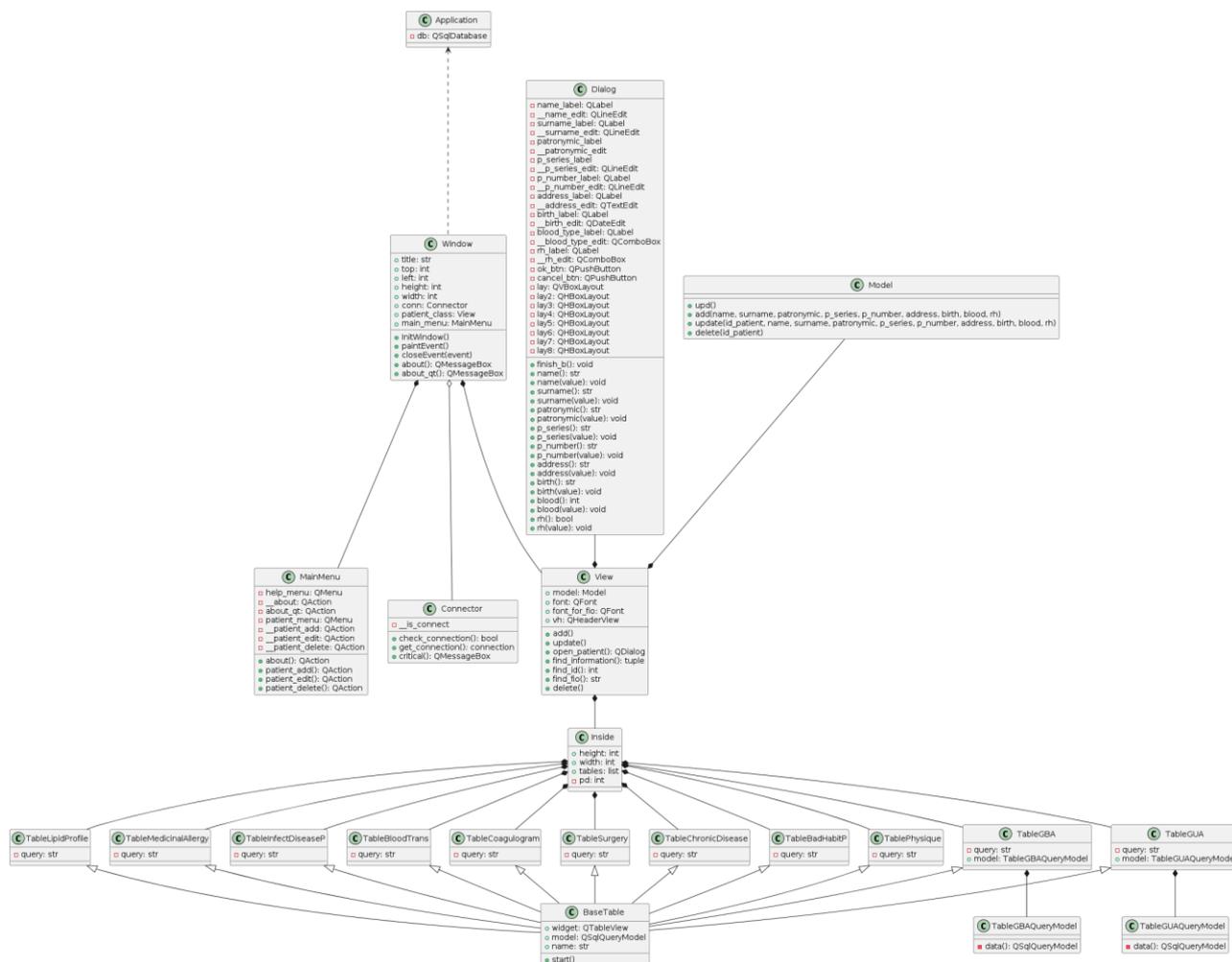


Рисунок 3 – Диаграмма классов

Диаграмма классов – это UML диаграмма, которая определяет типы классов системы, различного рода статические связи, которые существуют между ними, методы и атрибуты. Прямоугольники, разделенные на 3 секции, изображают классы, а стрелки между прямоугольниками указывают на связи между ними.

Далее в работе рассматривается каждый класс, описывается его функционал.

В качестве результата разработки интерфейса на рисунках 4 и 5 были изображены окна с таблицей пациентов и просмотром данных об анализах и приёмах у врача (конкретно – информация о результатах анализа крови).

База данных

Справка Пациент

	Имя	Фамилия	Отчество	Адрес	Дата рождения
1	Валерий	Добровольский	Андреевич	ул. Кирова, дом 7	08.01.1947
2	Сергей	Мельников	Вячеславович	ул. Советская, дом 2, кв. 4	20.07.2008
3	Валерий	Стрельцов		ул. Победная, дом 28	28.04.1982
4	Владимир	Иванов	Иванович		15.06.1936

Рисунок 4 – Окно с таблицей пациентов

Пациент

Мельников Сергей Вячеславович

Дата рождения: 2008-07-20

ОАК | Коагулограмма | Генотрансфузия | Хирургическая операция | Хроническое заболевание | Вредные привычки | ОАМ | Телосложение | Липидный профиль | Лекарственная аллергия | Инфекционные заболевания пациента

Пациент	Дата	Группа крови	Резус-фактор	Лейкоциты	Эритроциты	Тромбоциты	Гемоглобин
1 Мельников Сергей Вячеславович	20.01.2024	1	Отрицательный	3,7	4	200	154

Рисунок 5 – Внешний вид окна просмотра

Заключение. В результате проделанной работы было изучено, какие данные необходимы для предоперационного планирования экстренных хирургических операций – выполнена задача 1. Была спроектирована и создана база данных, содержащая в себе информацию, необходимую для предоперационного планирования экстренных хирургических операций – выполнена задача 2. Далее был создан интерфейс на языке Python, благодаря которому можно просматривать информацию, необходимую для предоперационного планирования экстренных хирургических операций – выполнена задача 3.

Таким образом, были решены все задачи, поставленные в начале работы, и достигнута конечная цель.