

Санкт-Петербургский государственный университет

ТОПУЗОВ Эдем Русланович

Выпускная квалификационная работа

***Разработка веб-ориентированной аналитической системы для обработки
данных медицинских научных исследований***

Уровень образования: магистратура

Направление *02.04.02 «Фундаментальная информатика и информационные
технологии»*

Основная образовательная программа *ВМ.5503.2020 «Технологии баз данных»*

Научный руководитель:

доцент кафедры, заведующий кафедрой
технологии программирования СПбГУ,
кандидат технических наук

Блеканов Иван Станиславович

Рецензент:

заведующий кафедрой неврологии и
психиатрии НМИЦ им. В. А. Алмазова,
доктор медицинских наук, доцент

Алексеева Татьяна Михайловна

Санкт-Петербург

2022

Содержание

Введение.....	3
Актуальность.....	3
Цель работы.....	5
Задачи работы.....	6
Практическая значимость работы.....	7
Глава 1. Обзор литературы.....	8
Глава 2. Обзор существующих решений.....	11
Глава 3. Материалы и методы.....	15
3.1. Требования к программному продукту.....	15
3.2. Описание архитектуры программного продукта.....	17
Глава 4. Результаты.....	23
4.1. Реализация архитектуры программного продукта.....	23
4.2. Тестирование и апробация.....	28
4.3. Внедрение в эксплуатацию.....	30
Заключение.....	31
Список сокращений.....	32
Список литературы.....	33

Введение

Актуальность

Технологические достижения в медицине в сочетании с новыми возможностями обработки масштабных объемов информации в области здравоохранения открывают как потенциал так и связанные с этим проблемы [2, 19].

Медицина традиционно была наукой о наблюдении и опыте [11]. И если когда-то исследования могли состоять из наблюдения лишь за несколькими пациентами и на основе полученных данных делались выводы о болезнях в целом, то сейчас необходимы достаточно крупные выборки пациентов, что привело к неуклонному увеличению объема исследовательского материала в каждой научно-исследовательской работе. Так например, в 1817 году английский врач Джеймс Паркинсон обессмертил свое имя, опубликовав знаменитое «Эссе о дрожательном параличе» со своими наблюдениями всего лишь за 6 больными с дрожательным параличом [22]. Парадигма современной доказательной медицины требует гораздо более обширных рандомизированных исследований, в ходе которых проводится сбор множества показателей с дальнейшей статистической обработкой результатов [14].

Цифровизация медицины позволяет отслеживать прогрессирование заболевания. Современные объемы медицинских данных огромны. Требуется упорная работа, чтобы собрать полезную информацию из этих данных и превратить ее в знания и действия. Для научных медицинских исследований обработка и анализ полученных данных должны значительно облегчить поиск подходящих участников, а также оценку осуществимости новых исследований. Исследователи нуждаются в новых вычислительных инструментах для улучшения доступа к данным, облегчения их анализа и эффективного принятия решений [11].

В результате проведенного анализа доступной литературы, выявлено, что на данный момент не существует систем, подходящих для индивидуальных исследований, например, диссертационных или в рамках государственных заданий, как в России, так и за рубежом.

Один из вариантов решения такой проблемы – разработка веб-ориентированной системы, которую можно использовать на различных этапах медицинских исследований: от ввода данных о пациентах до автоматического анализа показателей и предоставления прогностических данных.

В настоящее время актуальным является разработка российского программного обеспечения. Учитывая тот факт, что не существует аналогов таких систем в мире, создание подобной системы может являться первой, а следовательно, уникальной отечественной разработкой.

Цель работы

Создание веб-ориентированной аналитической системы для оптимизации существующего процесса обработки данных медицинских научных исследований.

Данная работа является разработкой программного продукта, заказчиком которого являются врачи-неврологи и сотрудники кафедры неврологии и психиатрии с клиникой ФГБУ НМИЦ им. В.А. Алмазова Минздрава России, выполняющие научные исследования.

Задачи работы

1. Проведение анализа существующего процесса обработки данных научных исследований в медицине.
2. Выявление недостатков существующей системы и разработка потенциальной стратегии ее оптимизации для увеличения удобства использования.
3. Проектирование архитектуры разрабатываемой системы.
4. Создание веб-приложения для хранения, анализа и обработки данных медицинских научных исследований согласно предъявляемым требованиям.
5. Тестирование и апробация разработанной системы.
6. Внедрение в эксплуатацию разработанного программного продукта.

Практическая значимость работы

Разработка и внедрение веб-ориентированной аналитической системы позволит оптимизировать процесс медицинских научных исследований путем сокращения времени на внесение и стандартизацию собранных данных, увеличения удобства создания и использования баз данных (БД), предоставления возможности членам рабочей группы научного исследования получить совместный удаленный доступ к информационной системе для чтения, пополнения, анализа и обработки информации.

Глава 1. Обзор литературы

В настоящее время существующие интеллектуальные системы медицинского назначения, как правило, подходят для узкого круга задач, например, направлены на диагностику или лечение конкретных заболеваний [5, 18].

Разработка интеллектуальных медицинских экспертных систем связана с трудно формализуемыми и трудноразрешимыми проблемами [18].

При анализе организма необходимо решать огромное количество различных взаимосвязанных задач. Сложность обработки определяется как самим функционированием организма, так и разнообразием и сложностью выполнения отдельных задач. Поэтому необходим комплексный подход к такому анализу [1, 7, 18].

Стоит отметить, что исходные данные в предметной области могут быть неоднозначны, противоречивы. На настоящий момент не существует единого и оптимального алгоритмического решения этой проблемы [7, 18].

Важно принять во внимание несколько моментов. Медицинские данные очень разнородны. Существует проблема качества медицинских данных, что усложняет анализ, диагностику и прогноз [17]. Дополнительная сложность связана с отсутствием единого стандарта хранения и передачи медицинской информации, а также разнообразием типов обрабатываемой информации [12].

Модули для обработки различных форматов данных не должны зависеть друг от друга, имея в целом только сущность пациента. Отсутствие такой зависимости является ключевым фактором, поскольку оно позволяет сервисам развиваться независимо. Кроме того, сервисы могут быть свободно заменены при необходимости [29]. Представляется необходимым при проектировании использовать информационные технологии, позволяющие

построить архитектуру таким образом, чтобы она обладала свойствами модифицируемости и масштабируемости [12].

Медицинские аналитические системы имеют различные подходы. Так, простые экспертные системы, основанные на заданных правилах, выдают результат в виде предиката: например, наличие или отсутствие синдрома, они не сообщают о вероятности в виде низкой, средней, высокой и т.д. Экспертные системы на основе нечеткой логики также известны как интеллектуальные нечеткие системы, содержат нечеткие правила и функции принадлежности. Неточные лингвистические определения при этом формализуются в виде нечетких множеств с определенной степенью принадлежности к нему. Проблема в такой системе заключается в сложном ручном формировании правил и функций принадлежности. Как компромисс могут быть использованы гибридные нейро-нечеткие модели [28].

Но опять же существует проблема в очень узкой специализации таких систем. На практике существует потребность в сложной системе интеллектуального анализа данных, которая могла бы проектировать и анализировать разнообразную информацию, поступающую из различных источников [18]. Отсутствие единого подхода к проектированию архитектуры является основным недостатком разрабатываемых медицинских информационных систем, что значительно усложняет процесс их модификации и масштабирования [7, 18].

Таким образом, возникает проблема разработки методов и инструментов интеллектуальной медицинской системы. Одна из задач для разрешения этой проблемы – создание оптимальной системы анализа медицинских данных [18].

Вариантов решения такой проблемы является разработка веб-ориентированной системы, которую можно использовать на различных этапах медицинских исследований: от ввода данных о пациентах до

автоматического анализа показателей и предоставления прогностических данных [1, 18].

Для построения системы используется структурно-алгоритмический метод проектирования биомедицинского объекта, заключающейся в декомпозиции исходной задачи проектирования на иерархически связанные проблемы и задачи [1, 18].

При анализе требований для разработки такой системы учитываются основные ограничения на проектируемую систему, на определенные характеристики системы (время обработки информации, надежность и т.д.), указываются основные критерии, по которым необходимо оптимизировать проектируемую систему [18].

Конфиденциальность – еще одна серьезная сложность общественного здравоохранения. В связи с этим существует необходимость в дополнительной защите информации в этой области, особенно в связи с растущим числом киберпреступлений [17].

Глава 2. Обзор существующих решений

Обзор литературы показал, что большинство существующих программных решений, связанных с обработкой данных результатов медицинских исследований сводятся или к созданию электронных медицинских карт, или к системам, которые на основе клинико-лабораторных анализов прогнозируют диагноз.

Такие решения направлены преимущественно для ведения клинической деятельности и подразумевают создание медицинских информационных систем (МИС). Например, в России наибольшее распространение получила qMS (q Медицинская Система, q Medical System) [9].

Такие системы включают в себя автоматизацию деятельности лечебно-профилактических учреждений, в которых объединены система поддержки принятия медицинских решений, электронные медицинские записи о пациентах, данные медицинских исследований в цифровой форме, данные мониторинга состояния пациента с медицинских приборов, средства общения между сотрудниками, финансовая и административная информация [3, 4, 6].

Для ведения научно-исследовательских работ такие решения не подходят, т.к. каждая из них подразумевает создание собственного дизайна, включающего определенный набор показателей, а также группировку, сортировку по пациентам и полученным данным с последующей статистической их обработкой, что имеющиеся медицинские информационные системы не позволяют сделать.

В таблице 1 представлена сравнительная таблица различных способов хранения и обработки данных медицинских научных исследований.

Текущее решение, используемое врачами-исследователями сводится к следующему: исследователи в начале создают таблицу Excel, в которой присутствуют колонки с названиями показателей. Затем заполняют в строках данные пациентов.

Таблица 1

**Сравнение различных способов хранения и
обработки данных медицинских научных исследований**

Критерий	МИС (на примере qMS)	Таблицы Excel, Google Spreadsheets	Разрабатываемая система
Подбор пациентов для исследования из общей базы данных	+	-	+
Хранение текстовых приложений с дополнительной информацией о пациентах	+	-	+
Возможность создания дизайна конкретного исследования	-	+	+
Возможность хранения данных для конкретного исследования	-	+	+
Возможность автоматизированной индивидуальной группировки данных по показателям согласно дизайну конкретного исследования	-	-	+
Возможность автоматизированного статистического анализа данных конкретного исследования по заданным группам	-	-	+
Возможность выгрузки данных в виде таблиц для последующей статистической обработки	-	+	+
Возможность автоматизированной группировки данных для последующей статистической обработки	-	-	+
Гибкое ограничение прав пользователей для конкретного исследования	-	-	+
Хранение данных и информации группы контроля (здоровые и добровольцы)	-	+	+

В ходе исследований могут существовать несколько групп пациентов, например, основная группа, группа контроля и другие. В зависимости от требований создаются новые таблицы, в которых присутствуют либо все пациенты, либо каждая из групп в новой таблице.

Также исследователи регулярно патентуют текущее состояние своей БД для каждого такого исследования. При этом с одной стороны данные должны обезличиваться, с другой стороны сохраняется необходимость идентифицировать данные исследования с конкретным пациентом. Также одно из требований при получении патента – указание типов данных и кодирование некоторых групп данных (например, 0 – женский пол, 1 – мужской пол и т.п.).

Кроме того, данные исследований выгружаются в программы Statistica и SPSS (статистический пакет для общественных наук, Statistical Package for the Social Sciences) для статистической обработки. При этом данные не обезличиваются, но показатели также кодируются. Часть данных, например, контактная информация пациента, в такие таблицы не выгружается. К тому же, для статистической обработки выделяется признак, по которому также происходит группировка, сортировка и кодирование. Например, по возрасту: 0 – до 45 лет, 1 – от 46 до 60 лет, 2 – старше 60 лет и т.п.

Таким образом, в ходе исследований члены рабочей группы вручную создают множество таблиц на основе исходной. Причем для разных целей содержимое и порядок данных в таких таблицах должны быть разными. Это занимает большое количество времени, а по сути является технической работой, которую можно было бы автоматизировать и тем самым существенно увеличить удобство работы с данными пациентов и облегчить обработку полученной информации. Данная работа по созданию веб-ориентированной аналитической системы обработки данных медицинских научных исследований поможет оптимизировать существующий процесс путем сокращения времени на внесение и стандартизацию собранных

данных, увеличения удобства создания и использования БД, предоставления возможности членам рабочей группы научного исследования получить совместный удаленный доступ к информационной системе для чтения, пополнения, анализа и обработки информации.

Идея создания такой системы была основана на запросе практикующих врачей-неврологов, которые ведут различные научные проекты на кафедре неврологии и психиатрии с клиникой ФГБУ НМИЦ им. В.А. Алмазова Минздрава России. Однако созданная система будет универсальна не только для врачей-исследователей различных специальностей, но и для любых научных исследований, подразумевающих сбор, хранение и обработку данных от объектов (или пациентов).

Глава 3. Материалы и методы

3.1. Требования к программному продукту

Основным требованием заказчика является создание веб-приложения для возможности внесения, изменения, хранения, систематизации и анализа медицинских данных пациентов, получаемых в ходе научных исследований.

В приложении необходимо реализовать следующий функционал, обеспечивающий возможность:

- создания дизайна научного исследования;
- внесения данных, в т.ч. согласно группам и подгруппам, выделенным в дизайне;
- предоставления разных уровней доступа к системе (полные права, ограниченные права) для обеспечения безопасности данных;
- одновременного доступа к системе до 100 пользователей с возможностью масштабирования;
- контроля со стороны главного исследователя изменений в БД, сделанными разными пользователями;
- регулярного резервного копирования и восстановления (при необходимости) копий БД;
- проведения анализа содержимого БД, внесенных в систему, путем поиска и просмотра данных по заданным параметрам;
- максимально быстрой (не более 1 с) выгрузки внесенных данных в таблицы Excel;
- кодирования данных, внесенных в систему с присвоением кодов для регистрации БД и подготовки к использованию в программах статистической обработки;
- группировки данных с выгрузкой в таблицы Excel для проведения статистической обработки, основанное на выделении определенных группирующих показателей (как одиночных, так и множественных);

- проведения статистического анализа в созданной системе согласно перечню статистических методов.

Перечень статистических методов:

1. Описание количественных переменных следующими параметрами: число пациентов, среднее значение, стандартное отклонение, дисперсия мода, медиана, 25-ый и 75-ый процентиля.

2. Для сравнения двух групп по количественному признаку:

- параметрический метод: t-критерий Стьюдента (t-test);

- непараметрический метод: критерий Уилкоксона.

Перечень статистических методов может расширяться и пополняться в зависимости от требований заказчика.

3.2. Описание архитектуры программного продукта

Прежде чем приступить непосредственно к разработке необходимо определиться с технологическим стеком проекта. Прежде всего был проработан основной функционал сервиса для того, чтобы иметь основу для принятия решений о выборе технологий.

Перед разработкой приложения были сформулированы следующие требования к системе:

1. Масштабируемость. Возможность повышать пропускную способность сервиса и добавлять новый функционал без необходимости сильно менять существующий код.

2. Безопасность. В основе технологического стека должны лежать продукты, которые имеют встроенную защиту. Вокруг этих продуктов должны быть живые сообщества разработчиков, поддерживающих продукт, чтобы вновь выявляемые уязвимости в безопасности могли быть оперативно определены и исправлены.

Для удобства работы была использована система контроля версий – Git.

В качестве хранилища кодовой базы была выбрана открытая платформа GitHub.

В качестве веб-фреймворка был выбран Ruby on Rails [25]. Он позволяет вести быструю разработку, выводя продукт в рабочий прототип [30]. Кроме того, фреймворк хорошо масштабируется и успешно используется в таких высоконагруженных сервисах как GitHub, GitLab, AirBnB [21], Сбермаркет [8] и другие.

Архитектура приложений Ruby on Rails определяется как фулстек фреймворк, поскольку он может работать со всеми возможными частями веб-приложения. К таким частям можно отнести: веб-сервисы, управление базами данных, генерация HTML (языка гипертекстовой разметки, HyperText Markup Language), сборка CSS (каскадных таблиц стилей, cascading style sheets) и JavaScript файлов, маршрутизация, кеширование, аутентификация,

авторизация, локализация и многое другое.

Одно из преимуществ фреймворка, влияющее на скорость разработки – принцип, в основе которого лежит факт того, что соглашение важнее конфигурации, это позволяет избегать большого количества повторяющегося кода, что ускоряет разработку [15].

Фреймворк Ruby on Rails, уделяет большое внимание безопасности. Многие механизмы встроены – это экранирование запросов, выполняемых на SQL (языке структурированных запросов, structured query language) для защиты от SQL-инъекций, защита от межсайтовой подделки запроса (Cross-Site Request Forgery, CSRF-атак), защита от межсайтового скриптинга (cross-site scripting, XSS-атак), фильтрация чувствительных данных при логировании, очистка данных из базы данных при выводе страницы в браузер, встроенное кодирование данных при сохранении в БД и многое другое.

В качестве системы управления базами данных (СУБД) выбрана PostgreSQL. Выбор пал на реляционную СУБД в связи с тем, что в приложении будет множество связанных между собой сущностей со строгой системой атрибутов. PostgreSQL позволяет осуществлять многопользовательский доступ к базе данных, обладает всеми свойствами ACID (atomicity – атомарностью, consistency – согласованностью, isolation — изолированностью). При необходимости позволяет безболезненно масштабировать приложение с помощью распределенных СУБД, совместимых с PostgreSQL, таких как CockroachDB, YugabyteDB и других.

Во время разработки были созданы две БД – непосредственно для разработки и для тестирования.

Изменения в схеме БД поддерживались с помощью миграций, которые позволяют поддерживать актуальное состояние схемы базы данных, гибко добавлять, изменять, удалять таблицы, индексы, триггеры, функции и прочие объекты СУБД, согласовывать текущее состояние с помощью системы

контроля версий Git.

Для предотвращения проблемы N+1 и неоправданного использования нетерпеливой загрузки во время разработки и тестирования использовалась библиотека bullet.

Для сбора данных о выгружаемых исследователями отчетов используется СУБД MongoDB. Эти данные представляют интерес для дальнейшего исследования и разработки. Необходимо накопить достаточно большой объем данных, чтоб впоследствии можно было реализовать функцию подсказки или предсказания выгружаемых параметров. Данная СУБД является документоориентированной, не требует описания схемы таблиц. Получаемые данные о выгружаемых отчетах разнородны, слабо структурированы, поэтому хорошо вписываются в базу данных без схемы.

В качестве менеджера пакетов Ruby был использован сборщик bundler, который скачивает библиотеки, выстраивает дерево зависимостей между ними, не допуская конфликтов.

Для аутентификации использована библиотека Devise. Для выделений ролей и авторизации было использовано сочетание библиотек RoleModel [23] и ActionPolicy [10]. Последняя обладает гибкостью для управления правами на различные действия в веб-приложении, в том числе ограничение на создание, чтение, обновление, удаление сущностей, ограничение списка сущностей для просмотра и т.д.

Одно из требований – контроль произведенных изменений (создание, изменение, удаление записей) в БД, сделанными разными пользователями. Данный функционал реализован с помощью библиотеки PaperTrail, которая позволяет проводить аудит изменений, сделанных разными пользователями. Инструментарий позволяет отслеживать какие изменения каким пользователем были осуществлены, кроме того, при необходимости производить откат и восстановление данных.

Экспорт данных в Excel осуществляется с помощью библиотеки

RubyXL [26]. Библиотека позволяет создавать Excel-файлы, внутри них создавать листы, заполнять ячейки, стилизовать их. Также при помощи этой библиотеки произведен импорт имеющихся БД, хранящихся в Excel-файлах.

С одной стороны таблицы Excel удобны и привычны для работы исследователей, с другой – позволяют масштабировать систему в связи с тем, что максимальные ширина и высота на каждом листе – 1048576 и 16384 ячеек соответственно, что означает возможность внесения данных более 1 миллиона пациентов по более, чем 16000 показателям.

Для статистической обработки использованы библиотеки `descriptive_statistics` и `ruby-statistics`.

Для тестирования был использован фреймворк RSpec [24] – наиболее популярное средство для тестирования приложений, написанных на Ruby. RSpec позволяет наглядно с помощью предметно-ориентированного языка (*domain specific language, DSL*) писать тесты так, что их можно читать, как документацию (*spec – спецификация*), при этом существует опция, которая в стандартном выводе в терминал в структурированном виде указывает что делает тестируемый класс и метод. Тестирование с помощью RSpec с одной стороны позволит в дальнейшем развивать приложение, масштабируя его, с другой – при необходимости быстро погружаться разработчикам в проект.

Для вычисления степени покрытия кодовой базы тестами была использована библиотека SimpleCov [27].

Чтобы сделать код единообразным, удобным для восприятия, в т.ч. для дальнейшей поддержки и развития приложения, используется линтер Rubocop, который автоматически вызывается перед каждым Git-коммитом с помощью утилиты `lefthook`.

Эта же утилита вызывает перед отправкой Git-коммитов в удаленной репозиторий задачу `bundle-audit`, которая проверяет имеющиеся в проекте библиотеки на уязвимости, используя базу данных общеизвестных уязвимостей (*Common Vulnerabilities and Exposures, CVE*). Это позволяет

поддерживать безопасность проекта, вовремя реагируя на выявляемые угрозы, обновляя библиотеки.

В качестве шаблонизатора был использован `slim` – быстрый легковесный движок, который по производительности быстрее используемого по умолчанию встроенного в Ruby on Rails `ERB` (встроенный Ruby, `embedded Ruby`). Другим преимуществом является отсутствие открывающих и закрывающих `HTML`-тегов, что с одной стороны ускоряет разработку, с другой – уменьшает вероятность ошибок.

Для динамической отрисовки элементов, например, добавления, изменения, скрытия полей ввода в веб-формах и т.д., в зависимости от действий пользователей, был использован `JavaScript`.

Для ввода дат и диапазона дат используется отечественный легкий, быстрый календарь с открытым исходным кодом, написанный на чистом `JavaScript`, без сторонних зависимостей – `Air DatePicker`.

В качестве менеджера `JavaScript` библиотек использовался `uagn`, который скачивает их, выстраивает зависимости между ними, предотвращая конфликты несовместимых версий.

Для ускорения разработки решено использовать `CSS`-фреймворк `Bootstrap`, как наиболее популярный (по количеству звезд `Github` и скачиваний в составе `npm`-пакетов), в его основе лежит компонентный подход.

Сборка и компоновка `CSS` и `JavaScript` файлов осуществлялась с помощью библиотеки `webpack`. Для удобства использовалась `Ruby`-обертка – `webpacker`, которая позволяет удобно организовать код и имеет встроенные `Ruby`-методы для доступа к стилям, скриптам, изображениям, шрифтам и прочим объектам. Во время разработки сервер `webpack-dev-server` позволял вносить правки в стили и скрипты таким образом, что эти изменения подхватывались на лету без перезагрузки бэкенд сервера, что также ускоряло процесс разработки.

Проект использует средство контейнеризации с помощью `Docker`. Это

позволяет изолировать приложение от основной операционной системы, что с одной стороны повышает безопасность приложения, с другой позволяет легко настроить окружение, системные утилиты, настройки, не влияя на инфраструктуру хоста. Кроме того, это ускоряет разворачивание приложения, позволяет масштабировать его. Для организации взаимодействия контейнеров между собой используется утилита `docker-compose`. В дальнейшем при существенном расширении проекта возможен переход к средствам оркестрации `Docker Swarm` или `Kubernetes`.

Для обеспечения многопоточности и быстроты работы используется веб-сервер `Puma`. В качестве обратного прокси-сервера используется `nginx`.

Таким образом, технологический стек можно представить в виде таблицы 2.

Таблица 2

Технологический стек разрабатываемой системы

Параметр	Значение
Веб-фреймворк	Ruby on Rails
СУБД	PostgreSQL, MongoDB
Аутентификация	Devise
Авторизация	ActionPolicy, RoleModel
Контроль изменений в БД	PaperTrail
Работа со статистикой	ruby-statistics
Импорт и экспорт Excel	RubyXL
Средство контейнеризации	Docker
Веб-сервер	Puma
Обратный прокси-сервер	Nginx
Средства тестирования	RSpec, Simplecov
Линтер	Rubocop
Средство отслеживания уязвимостей	Bundle Audit
Сборка фронтенд	Webpack
Шаблонизатор	Slim
Система контроля версий (удаленный репозиторий)	Git (GitHub)

Глава 4. Результаты

4.1. Реализация архитектуры программного продукта

В результате проведенной работы разработана веб-ориентированная аналитическая система для обработки данных медицинских научных исследований.

Адрес публичного репозитория – <https://github.com/mechnicov/med-research-system>.

С учетом требований к проекту была создана схема базы данных, указанная на рисунке 1.



Рисунок 1. Схема базы данных проекта

Реализован функционал создания дизайна научного исследования, предусматривающий управление показателями, пациентами, группами пациентов, участвующих в нем, внесение, изменение, удаление данных.

Для этого созданы модели согласно указанной схеме БД, соответствующие контроллеры и представления для просмотра и редактирования их атрибутов.

Доступ для указанных действий разграничен. Так, например, создать новое исследование, организовав его дизайн может только главный исследователь, вносить полученные данные в ходе исследования может только исследователь, имеющий доступ к нему и т.д.

Главный исследователь может управлять ролями, выдавая и отзывая права, также может добавлять и удалять учетные записи для членов научной группы.

Все внесенные изменения версионизируются. Главным исследователям доступен функционал выгрузки отчетов Excel, в которых указано, кто когда и какие изменения вносил. Для этого реализован специальный сервис.

Кроме того, раз в сутки происходит резервное копирование базы данных. Процесс резервного копирования оканчивается сжатием с помощью утилиты Gzip, а затем шифрованием и подписыванием с помощью утилиты GNU Privacy Guard. Таким образом, с одной стороны возможно восстановление данных, с другой стороны резервные копии можно открыть только при наличии соответствующего ключа.

Для удобства работы с пациентами, их показателями и прочими сущностями организованы поисковые формы по различным показателям, позволяющие фильтровать данные по разным признакам. Между различными сущностями организована сквозная навигация, например, со страницы пациента можно перейти на страницу исследования, в котором он участвует, оттуда на используемые показатели и т.д.

Реализована возможность выгрузки данных конкретного исследования в Excel, с фильтрацией по группам пациентов, группам показателей. За формирование таких отчетов отвечает отдельный сервис.

Возможна выгрузка трех видов отчета:

- для исследователей – наиболее полная форма отчета, содержащая все данные о проведенных исследованиях с данными о пациентах, в т.ч. их контактными данными;

- для регистрации – форма для регистрации патента, содержит обезличенные данные, данные исследований закодированы, на отдельных листах – расшифровка кодированных данных и пациентов;

- для статистики – форма для выгрузки данных для дальнейшего использования в статистических программах Statistica и SPSS (рисунок 2), данные исследований закодированы, произведена группировка и сортировка данных по заданным исследователем параметрам, на отдельном листе – расшифровка кодированных данных.

Реализовано формирование отчетов с использованием описательной статистических методов с группировкой результатов по выбранным показателям (рисунок 3).

Также предусмотрена выгрузка отчетов для оценки различий между выборками по уровню количественного показателя с помощью параметрических и непараметрических критериев, как по вариантам признака (рисунок 4), так и по группам выборки (рисунок 5).

За формирование статистических отчетов также отвечает специально разработанный сервис.

В случае ошибок при использовании системы отрисовываются соответствующие страницы. Так, созданы страницы для ошибок клиента: 403 – для запрещенных действий, 404 – для отсутствующих страниц, для ошибок сервера: 500 – для внутренних ошибок сервера.

Экспорт

Уточнить параметры

Основная
 Контроль

Сахарный диабет в точке 1
 Размер очага в точке 1

Пораженный сосудистый бассейн в точке 1
 Наличие лейкоареоза в точке 1

Шкала NIHSS при поступлении в точке 1
 Шкала NIHSS при выписке в точке 1

Шкала Бартел при поступлении в точке 1
 Шкала Бартел при выписке в точке 1

Глюкоза в точке 1
 HCE в точке 2
 HCE в точке 1
 ГФКБ в точке 1

ГФКБ в точке 2
 NR2-антитела в точке 2
 NR2-антитела в точке 1

Для исследователей
 Для регистрации
 Для статистики

Рост в точке 1

18 44 Молодые -

45 60 Средний возраст -

61 До Название группы -

+

Выгрузить в XLS

Рисунок 2. Выгрузка отчета для статистики

Статистика

Уточнить параметры

Описательная статистика

Количество
 Среднее значение
 Медиана
 Мода
 Дисперсия
 Стандартное отклонение
 Минимальное значение
 Максимальное значение
 Первый квартиль
 Третий квартиль

Количественное сравнение

Основная
 Контроль

Показатели

Систolicеское артериальное давление в точке 1
 HCE в точке 1

Группировка

Наличие лейкоареоза в точке 1

Выгрузить в XLS

Рисунок 3. Выгрузка отчета с использованием описательных статистических методов

Статистика

[Уточнить параметры](#)

Описательная статистика

Количественное сравнение

Критерий Стьюдента По вариантам

Критерий Вилкоксона По группам

Основная × ▾

NR2-антитела в точке 1 × ▾

NR2-антитела в точке 2 × ▾

Выгрузить в XLS

Рисунок 4. Выгрузка отчета с оценкой критерия Стьюдента для сравнения в разных группах динамики изменения показателей

Статистика

[Уточнить параметры](#)

Описательная статистика

Количественное сравнение

Критерий Стьюдента По вариантам

Критерий Вилкоксона По группам

× Основная × Контроль ×

× НСЕ в точке 2 × НСЕ в точке 1 × ГФКБ в точке 1 × ГФКБ в точке 2 ×

Выгрузить в XLS

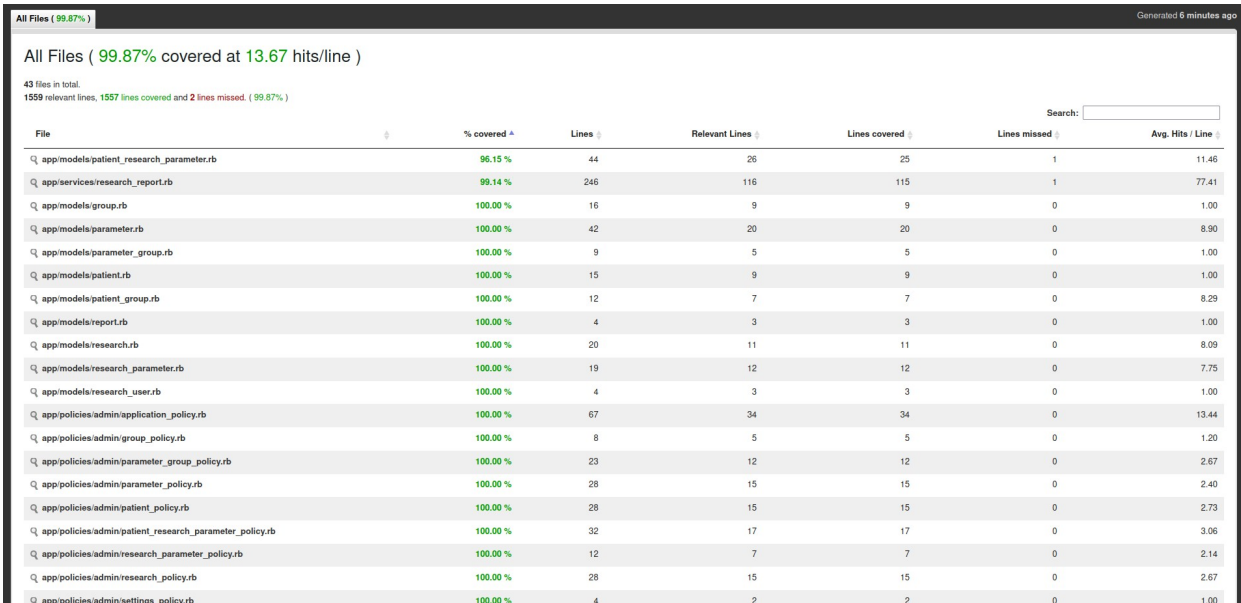
Рисунок 5. Выгрузка отчета с оценкой критерия Вилкоксона для сравнения различных показателей в двух группах

4.2. Тестирование и апробация

Разработанная система была покрыта тестами. Количество тестов – 318 (рисунок 6). Процент покрытия тестами – 99.87% (рисунок 7).

```
Finished in 5.05 seconds (files took 3.05 seconds to load)
318 examples, 0 failures
```

Рисунок 6. Количество тестов



The screenshot shows a code coverage report for 'All Files (99.87%)'. It indicates that 43 files in total were tested, with 1557 relevant lines, 1557 lines covered, and 2 lines missed. The overall coverage is 99.87% at 13.67 hits per line. A search bar is visible at the top right of the report area.

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/models/patient_research_parameter.rb	96.15 %	44	26	25	1	11.46
app/services/research_report.rb	99.14 %	246	116	115	1	77.41
app/models/group.rb	100.00 %	16	9	9	0	1.00
app/models/parameter.rb	100.00 %	42	20	20	0	8.90
app/models/parameter_group.rb	100.00 %	9	5	5	0	1.00
app/models/patient.rb	100.00 %	15	9	9	0	1.00
app/models/patient_group.rb	100.00 %	12	7	7	0	8.29
app/models/report.rb	100.00 %	4	3	3	0	1.00
app/models/research.rb	100.00 %	20	11	11	0	8.09
app/models/research_parameter.rb	100.00 %	19	12	12	0	7.75
app/models/research_user.rb	100.00 %	4	3	3	0	1.00
app/policies/admin/application_policy.rb	100.00 %	67	34	34	0	13.44
app/policies/admin/group_policy.rb	100.00 %	8	5	5	0	1.20
app/policies/admin/parameter_group_policy.rb	100.00 %	23	12	12	0	2.67
app/policies/admin/parameter_policy.rb	100.00 %	28	15	15	0	2.40
app/policies/admin/patient_policy.rb	100.00 %	28	15	15	0	2.73
app/policies/admin/patient_research_parameter_policy.rb	100.00 %	32	17	17	0	3.06
app/policies/admin/research_parameter_policy.rb	100.00 %	12	7	7	0	2.14
app/policies/admin/research_policy.rb	100.00 %	28	15	15	0	2.67
app/policies/admin/settings_policy.rb	100.00 %	4	2	2	0	1.00

Рисунок 7. Покрытие тестами

Проект контейнеризирован и развернут на виртуальном выделенном сервере по адресу <https://medbase.topuzov.dev/admin>.

Автоматический выпуск и продление сертификатов SSL (уровень защищенных сокетов, Secure Sockets Layer) центра сертификации Let's Encrypt осуществляются с помощью бота Certbot.

Для оценки масштабируемости системы было успешно проведено нагрузочное тестирование с количеством запросов до 10000 в минуту. Время ответа сервера не превышало 1 с, что соответствует требованиям к продукту.

Результаты работы продемонстрированы и протестированы на кафедре

неврологии и психиатрии с клиникой ФГБУ НМИЦ им. В.А. Алмазова Минздрава России.

Разработанный программный продукт полностью удовлетворяет текущим требованиям сотрудников кафедры, выполняющих научные исследования (таблица 3).

Таблица 3

Соответствие разработанного продукта заданным требованиям

Требование	Наличие
Возможность:	
- создания дизайна научного исследования	+
- внесения данных, в т.ч. согласно группам и подгруппам, выделенным в дизайне	+
- предоставления разных уровней доступа к системе (полные права, ограниченные права) для обеспечения безопасности данных	+
- одновременного доступа к системе до 100 пользователей с возможностью масштабирования	+
- контроля со стороны главного исследователя изменений в БД, сделанными разными пользователями	+
- регулярного резервного копирования и восстановления (при необходимости) копий БД	+
- проведения анализа содержимого БД, внесенных в систему, путем поиска и просмотра данных по заданным параметрам	+
- максимально быстрой (не более 1 с) выгрузки внесенных данных в таблицы Excel	+
- кодирования данных, внесенных в систему с присвоением кодов для регистрации БД и подготовки к использованию в программах статистической обработки	+
- проведения статистического анализа в созданной системе согласно перечню статистических методов	+

4.3. Внедрение в эксплуатацию

Веб-ориентированная аналитическая система для обработки данных медицинских научных исследований внедрена в эксплуатацию в ФГБУ НМИЦ им. В.А. Алмазова Минздрава России.

Продукт в течение месяца используется четырьмя научными группами в составе 10 человек, внесено более 5000 показателей исследований. За время эксплуатации система показала свою работу без сбоев, от пользователей получена положительная обратная связь, а также запросы для расширения функционала.

В дальнейшем планируется оформление патента на разработанный программный продукт.

Заключение

В ходе проделанной работы были исследованы возможные подходы к разработке веб-ориентированной аналитической системы для обработки данных медицинских научных исследований, спроектирована архитектура, подобраны подходящие для решаемой задачи инструменты, разработан, протестирован, апробирован и внедрен программный продукт. Процесс разработки и его результаты были в полной мере описаны и обоснованы. Также были продуманы и представлены дальнейшие варианты развития системы.

У разработанного продукта есть множество перспектив для развития. В дальнейшем возможна реализация предсказания выгружаемых отчетов, построение и экспорт диаграмм и графиков, расширение перечня поддерживаемых методов статистической обработки, необходимых для конкретных исследований, что в конечном счете позволит отказаться от использования дополнительных программных комплексов для статистического анализа.

Таким образом, задачи работы были решены, а поставленная цель достигнута.

Список сокращений

БД – база данных

МИС – медицинская информационная система

СУБД – система управления базами данных

ACID – атомарность (atomicity), согласованность (consistency),
изолированность (isolation), прочность (durability)

CSRF – межсайтовая подделка запроса (cross-site request forgery)

CSS – каскадные таблицы стилей (cascading style sheets)

CVE – база данных общеизвестных уязвимостей информационной
безопасности Common Vulnerabilities and Exposures

DSL – предметно-ориентированный язык (domain specific language)

ERB – встроенный Руби (embedded Ruby)

HTML – язык гипертекстовой разметки (HyperText Markup Language)

qMS – q Медицинская Система (q Medical System)

SPSS – статистический пакет для общественных наук (Statistical
Package for the Social Sciences)

SQL – язык структурированных запросов (structured query language)

SSL – уровень защищенных сокетов (Secure Sockets Layer)

XSS – межсайтовый скриптинг (cross-site scripting)

Список литературы

1. Арунянц Г.Г. Информационные технологии в медицине и здравоохранении: практикум / Г.Г. Арунянц, Д.Н. Столбовский, А.Ю. Калинин. – Ростов-на-Дону: Феникс, 2009. – 381 с.
2. Гусев А.В. Перспективы применения больших данных в российском здравоохранении // Московская медицина. – 2022. – № 1(47). – С. 26-30.
3. Гусев А.В., Плисс М. А., Левин М. Б. и др. Тренды и прогнозы развития медицинских информационных систем в России. // Врач и информационные технологии. – 2019. – № 2. – С. 38-49.
4. Карпов О.Э., Никуличев А.А., Пензин О.В. и др. Архитектура медицинских информационных систем нового поколения // Вестник Национального медико-хирургического Центра им. Н.И. Пирогова. – 2019. – Т. 14. – №. 3. – С. 126-134.
5. Луценко Е.В. Развитие медицинский информационных технологий в Российской Федерации // Вятский медицинский вестник. – 2017. – № 2(54). – С.73-77.
6. Приказ Министерства здравоохранения РФ от 24 декабря 2018 г. N 911н «Об утверждении Требований к государственным информационным системам в сфере здравоохранения субъектов Российской Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций» [Электронный ресурс]. – URL: <https://base.garant.ru/72217630>. (Дата обращения: 27.05.2022).
7. Пугачев П.С., Гусев А.В., Кобякова О.С. и др. Мировые тренды цифровой трансформации отрасли здравоохранения // Национальное здравоохранение. – 2021. – Т. 2. – №. 2. – С. 5-12.
8. «Руби мистически жив»: как в СберМаркете превращают разработчиков других стеков в рубистов. [Электронный ресурс]. URL:

<https://habr.com/en/company/sbermarket/blog/647937/>. (Дата обращения: 27.05.2022).

9. Шаханов А.С., Ушакова Е.В. Использование современных информационных технологий в государственном управлении // Трансформация бизнеса и общественных институтов в условиях цифровизации экономики. – 2020. – С. 199-211.

10. Action Policy. [Электронный ресурс]. URL: https://github.com/palkan/action_policy. (Дата обращения: 27.05.2022).

11. Auffray C., Balling R., Barroso I., et al. Making sense of big data in health research: towards an EU action plan // Genome medicine. – 2016. – 8(1) – P. 1-13.

12. Buldakova T. I., Lantsberg A. V., Suyatinov S. I. Multi-agent architecture for medical diagnostic systems // 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA). – IEEE, 2019. – P. 344-348.

13. Dhayne H., Haque R., Kilany R. et al. In search of big medical data integration solutions-a comprehensive survey // IEEE Access. – 2019. – V. 7. – P. 91265-91290.

14. Evidence-Based Medicine Working Group. Evidence-based medicine. A new approach to teaching the practice of medicine // JAMA. – 1992. – V. 268. – №. 17. – P. 2420-2425.

15. Fernández-Villamor J., Díaz-Casillas L., Iglesias C. A Comparison Model for Agile Web Frameworks // Proceedings of the 2008 Euro American Conference on Telematics and Information Systems (EATIS '08). – 2008. – ACM. New York, NY, USA, Article 14, 8 p.

16. Kumari N. M. J., Krishna K. K. V. Prognosis of diseases using machine learning algorithms: A survey // 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT). – IEEE, 2018. – P. 1-9.

17. Kotsiuba I. et al. Decentralized e-Health architecture for boosting

healthcare analytics // 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). – IEEE, 2018. – P. 113-118.

18. Magrupov T., Yusupov S., Talatov Y. et al. Intelligent Medical System of Designing Medical Technics and Technology // 2020 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2020. – P. 1-4.

19. Moutselos K. et al. Trustworthy data processing for health analytics tasks // 2018 IEEE International Conference on Big Data (Big Data). – IEEE, 2018. – P. 3774-3779.

20. Nazir S., Khan S., Khan H.U. et al. A comprehensive analysis of healthcare big data management, analytics and scientific programming // IEEE Access. – 2020. – V. 8. – P. 95714-95733.

21. Onishi I., Tsuji H., Irida M. A tool written in Scala for preparation and analysis in MD simulation and 3D-RISM calculation of biomolecules // Biophysics and physicobiology. – 2019. – T. 16. – С. 485-489.

22. Parkinson J. An essay on the shaking palsy // The Journal of neuropsychiatry and clinical neurosciences. – 2002. – V. 14. – №. 2. – P. 223-236.

23. RoleModel. [Электронный ресурс]. URL: https://github.com/martinrehfeld/role_model. (Дата обращения: 27.05.2022).

24. RSpec for Rails 5+. [Электронный ресурс]. URL: <https://github.com/rspec/rspec-rails>. (Дата обращения: 27.05.2022).

25. Ruby on Rails. [Электронный ресурс]. URL: <https://github.com/rails/rails>. (Дата обращения: 27.05.2022).

26. rubyXL. [Электронный ресурс]. URL: <https://github.com/weshatheleopard/rubyXL>. (Дата обращения: 27.05.2022).

27. SimpleCov. [Электронный ресурс]. URL: <https://github.com/simplecov-ruby/simplecov>. (Дата обращения: 27.05.2022).

28. Singh D., Verma S., Singla J. A comprehensive review of intelligent medical diagnostic systems // 2020 4th International Conference on Trends in

Electronics and Informatics (ICOEI)(48184). – IEEE, 2020. – P. 977-981.

29. Snegireva E., Khazankin G. R., Mikheenko I. System Architecture for Reading and Interpreting Physical Printouts of Medical Forms // 2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials (EDM). – IEEE, 2021. – P. 547-550.

30. Vassallo, K., Garg L., Prakash V. et al. Contemporary technologies and methods for cross-platform application development // Journal of Computational and Theoretical Nanoscience. – 2019. – V. 16. – №. 9. – P. 3854-3859.