

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»,
профессор департамента программной
инженерии, канд. техн. наук

В.В. Шилов
« ____ » _____ 2018 г.

Выпускная квалификационная работа

на тему «Программа мониторинга ауторегуляции головного мозга у пациентов с
черепно-мозговой травмой»

по направлению подготовки 09.03.04 «Программная инженерия»

Научный руководитель

Руководитель отдела новых
технологий научно-исследовательского
института неотложной детской
хирургии и травматологии, кандидат
технических наук
Арсеньев Сергей Борисович

С.Б. Арсеньев
« ____ » _____ 2018 г.

Выполнил

Студент группы БПИ143
4 курса бакалавриата
образовательной программы
«Программная инженерия»
Мендеев Александр Петрович

А.П. Мендеев
« ____ » _____ 2018 г.

Консультант

Старший преподаватель, младший
научный сотрудник МНУЛ ИССА
Максименкова Ольга Вениаминовна

О.В. Максименкова
« ____ » _____ 2018 г.

Москва 2018

Аннотация

На сегодняшний день черепно-мозговая травма является одной из самых распространенных форм патологии в области нейрохирургии. Наиболее частыми причинами возникновения черепно-мозговой травмы являются дорожно-транспортные происшествия и бытовые травмы. Пациенты с черепно-мозговой травмой зачастую страдают от нарушения ауторегуляции мозгового кровообращения. Нарушенная ауторегуляция головного мозга может привести к смерти пациента после травмы головы. Мониторинг процесса ауторегуляции головного мозга позволяет врачу оценить состояние пациента и назначить соответствующее лечение. На данный момент не было выработано единой методологии оценки процесса ауторегуляции мозгового кровообращения. В данной работе используются индексы ауторегуляции мозга для оценки ауторегуляции головного мозга.

Целью данной работы является разработка многоплатформенного программного решения, которое позволит врачу непрерывно оценивать состояние ауторегуляции головного мозга у пациентов с черепно-мозговой травмой. Ожидается, что данное программное решение позволит врачам индивидуализировать лечение пациентов на основании данных, полученных в результате непрерывной оценки процесса ауторегуляции головного мозга.

Работа содержит 58 страниц, 3 главы, 26 рисунков, 18 источников и 6 таблиц.

Ключевые слова: *черепно-мозговая травма, ауторегуляция головного мозга, мониторинг, оценка, индексы ауторегуляции, java, javascript, spring framework, angular framework.*

Abstract

Traumatic brain injury (TBI) become one of the most spread pathologies form in the field of neurosurgery nowadays. The leading causes of traumatic brain injury are road accidents and home accidents. Patients diagnosed with traumatic brain injury often suffer from impairments in brain autoregulation. Disturbed autoregulation can be one of the causes of death after head injury. Monitoring of brain autoregulation enables doctor assessing patient's condition and give appropriate treatment. At the moment, there is no universal method for assessing brain autoregulation. This research focuses on the use of brain autoregulation indices in the assessment of brain autoregulation process.

The aim of this study is to develop multiplatform software solution which will allow doctors to continuously assess brain autoregulation process condition in patients with traumatic brain injury. It is expected that the software solution will help doctors to individualize treatment of patients with traumatic brain injury by using the monitoring data.

The paper contains 58 pages, 3 chapters, 26 illustrations, 18 references and 6 tables.

Keywords: traumatic brain injury, cerebral autoregulation, monitoring, assessment, autoregulation indices, java, javascript, spring framework, angular framework.

Оглавление

Аннотация	2
Abstract	3
Список использованных терминов и сокращений	6
Введение.....	7
Глава 1. Ауторегуляция головного мозга	9
1.1 Ауторегуляция мозгового кровообращения.....	9
1.2 Мониторинг ауторегуляции мозгового кровообращения	9
1.3 Способы оценки процесса ауторегуляции мозгового кровообращения.....	9
1.3.1 Индексы ауторегуляции мозга	10
1.3.1.1 Pressure Reactivity Index	10
1.3.1.2 Variation Reactivity Index	12
1.3.2 Существующее программное решение	13
1.4 Разработка программного решения.....	15
1.5 Задачи работы.....	15
1.6 Выводы по главе	16
Глава 2. Особенности реализации	18
2.1 Общая архитектура проектного решения.....	18
2.2 Обработка данных.....	20
2.2.1 Данные пациентов	20
2.2.2 Временной интервал между поступающими данными.....	20
2.2.3 Окно вычислений.....	21
2.2.4 Вычисление индексов PRx и VRx.....	21
2.2.5 Классификация значений индексов ауторегуляции PRx и VRx	22
2.2.6 Расчет относительных частот встречаемости значений разных классов индексов ауторегуляции PRx и VRx	23
2.2.7 Представление информации в программе.....	24
2.3 Программа мониторинга ауторегуляции головного мозга	26
2.3.1 Пользователи программы	26
2.3.1.1 Аутентификация.....	26
2.3.1.2 Роли пользователей.....	26
2.3.2 Варианты использования	27
2.3.2.1 Мониторинг пациента	28
2.3.2.2 Изменение индивидуальных настроек пациента	30
2.3.2.3 Составление отчётов (выгрузка данных).....	31

2.3.2.4	Администрирование	34
2.4	Выводы по главе	35
Глава 3. Технические детали реализации программного решения		37
3.1	Использованные инструменты разработки	37
3.1.1	Серверная часть программы	37
3.1.2	Клиентская часть программы	37
3.2	Использованные технологии	37
3.2.1	Серверная часть программы	37
3.2.2	Клиентская часть программы	39
3.3	База данных	40
3.4	Работа с данными пациентов	41
3.4.1	Источник данных пациентов	41
3.4.2	Клиническая база данных НИИ НДХиТ	42
3.4.3	Конфигурация соединения	42
3.4.4	Запрос данных	44
3.4.5	Обработка данных	45
3.4.6	Передача данных в клиентскую часть программы	47
3.5	Веб-сервисы	47
3.5.1	Аутентификация и авторизация доступа к ресурсам	50
3.6	Выводы по главе	53
Апробация результатов работы		54
Заключение		55
Список использованных источников		57
Приложение А		59
Приложение Б		80
Приложение В		115
Приложение Г		151

Список использованных терминов и сокращений

PRx – Pressure Reactivity Index

VRx – Variation Reactivity Index

АД – артериальное давление

ВЧД – внутричерепное давление

ЦПД – церебральное перфузионное давление

РШИГ – Расширенная шкала исходов Глазго

НИИ НДХиТ - Научно-исследовательский институт неотложной детской хирургии и травматологии

БД – База данных

СУБД – Система управления базами данных

CRUD (create, read, update, delete) – операции создания, чтения, изменения и удаления информации

Java EE (Java Enterprise Edition) – набор спецификаций (стандартов), описывающих создание серверных приложений и информационных систем на платформе Java

Сервлет – спецификация компонента серверного приложения Java, расширяющего возможности сервера

STOMP (Simple Text-Oriented Messaging Protocol) – протокол обмена текстовыми сообщениями

JSON (JavaScript Object Notation) – формат представления текстовых данных

HATEOAS (Hypermedia as the Engine of Application State) – принцип построения веб-сервиса, основанный на предоставлении данных о сложном составном ресурсе в виде гиперссылок на соответствующие подресурсы

Аутентификация – процесс проверки подлинности введенных пользователем данных и установления личности этого пользователя

ПИН-код – цифробуквенное сочетание символов, которое используется для прохождения аутентификации и позволяет однозначно идентифицировать пользователя программы

Введение

Несмотря на активное развитие медицинских технологий, на сегодняшний день черепно-мозговая травма является остается из важнейших проблем здравоохранения. Каждый год в мире около 1.5 млн. человек погибают в результате черепно-мозговой травмы [1]. Пациенты с тяжелой черепно-мозговой травмой зачастую страдают от нарушений ауторегуляции мозгового кровообращения. Ауторегуляция мозгового кровообращения – это способность организма поддерживать постоянство мозгового кровотока [2]. Исследования показывают, что от 49 до 87% пациентов с черепно-мозговой травмой демонстрируют нарушенную или отсутствующую ауторегуляцию мозгового кровообращения [3] [4]. Нарушенная ауторегуляция мозга может приводить к тяжелым последствиям у пациентов после травмы головы [5].

Наблюдение за состоянием ауторегуляции головного мозга позволяет врачу подбирать лечение для пациента индивидуально, учитывая текущее состояние пациента [5]. Мониторинг ауторегуляции головного мозга подразумевает непрерывную качественную оценку данного процесса в реальном времени. Проблема качественной оценки процесса ауторегуляции мозга до сих пор не имеет единого решения. На практике врачи используют различные методы для оценки ауторегуляции. Одним из методов оценки ауторегуляции головного мозга является использование индексов ауторегуляции – обычных чисел, позволяющих оценить эффективность процесса ауторегуляции головного мозга. Несмотря на усилия, прилагаемые мировым научным сообществом к изучению индексов ауторегуляции, а также высокую прогностическую значимость использования индексов ауторегуляции, на сегодняшний день в свободном доступе находится только одно программное решение, позволяющее производить непрерывную оценку процесса ауторегуляции при помощи вычисления индексов ауторегуляции. К сожалению, имеющееся программное обеспечение имеет некоторые недостатки, которые будут рассмотрены подробнее в ходе данной работы.

Целью работы является создание многоплатформенного программного решения, позволяющего производить непрерывную оценку процесса ауторегуляции головного мозга у пациентов с черепно-мозговой травмой с использованием индексов ауторегуляции. Ожидается, что данное программное обеспечение поможет врачам отслеживать состояние пациентов с травмой головы в реальном времени, что поможет им в назначении соответствующего лечения и принятии необходимых клинических решений.

Для решения поставленной цели были выделены следующие задачи:

1. Изучить современные способы оценки ауторегуляции головного мозга;
2. Провести обзор существующих решений и выявить их недостатки;
3. Определить методологию оценки ауторегуляции мозгового кровообращения;

4. Выявить требования к функциональным и нефункциональным характеристикам разрабатываемого решения;
5. Определить технологии, которые будут использованы при разработке программы;
6. Выявить требования к входным и выходным данным;
7. Разработать архитектуру программного решения;
8. Разработать структуру базы данных для хранения данных программы;
9. Разработать программу;
10. Разработать техническую документацию.

Данная работа состоит из трёх глав. В первой главе представлен обзор существующих методов оценки ауторегуляции мозга, а также краткий обзор существующего программного решения и его недостатков. В конце главы приводятся некоторые особенности проектируемого программного обеспечения в сравнении с существующим решением и приводится постановка задачи на разработку. Во второй главе данной работы содержится описание методологии оценки ауторегуляции мозгового кровообращения, используемой в данном проекте и основанной на вычислении индексов ауторегуляции. Также во второй главе описаны основные особенности программной реализации и варианты использования программы с точки зрения конечных пользователей. В третьей главе описаны технические детали реализации наиболее важных частей программы. В заключительной части работы представлены результаты, полученные в ходе выполнения данной работы, а также описание дальнейших направлений данного исследования. В приложениях к данной работе содержится техническая документация программной реализации данной работы.

Глава 1. Ауторегуляция головного мозга

Данная глава содержит обзор некоторых методов оценки ауторегуляции головного мозга, а также описание существующего программного решения и его недостатков.

1.1 Ауторегуляция мозгового кровообращения

Ауторегуляция головного мозга является сложным физиологическим процессом, направленным на поддержание адекватного кровоснабжения головного мозга. При патологических состояниях (черепно-мозговая травма, другие травмы головы) ауторегуляция головного мозга зачастую нарушается. При нарушенной ауторегуляции головного мозга, изменения церебрального перфузионного давления (уровня кровоснабжения головного мозга) на фоне колебаний артериального и внутричерепного давлений могут приводить к гипоксии (кислородному голоданию) или отёку головного мозга [6].

Исходя из этого, поддержание нормального состояния ауторегуляции мозга имеет большое практическое значение при лечении травм головы.

1.2 Мониторинг ауторегуляции мозгового кровообращения

Целью данной работы является разработка программы для мониторинга ауторегуляции головного мозга. Само понятие мониторинга в разных источниках выглядит по-разному. Одно из определений гласит, что мониторинг – это система сбора данных о некоем сложном явлении или процессе в целях оперативной диагностики состояния объекта исследования и оценки его в динамике [7]. Согласно определению Оксфордского словаря английского языка, глагол «to monitor» обозначает наблюдение и оценку прогресса или качества чего-либо во времени [8].

В данной работе под мониторингом ауторегуляции имеется в виду непрерывная оценка данного физиологического процесса во времени для предоставления врачу информации о состоянии ауторегуляции головного мозга пациента.

1.3 Способы оценки процесса ауторегуляции мозгового кровообращения

Мониторинг процесса ауторегуляции головного мозга подразумевает его непрерывную качественную оценку. Оценка ауторегуляции предоставляет врачу ценную информацию о состоянии регуляции мозгового кровообращения пациента. Поскольку ауторегуляция сама по себе является довольно сложным физиологическим процессом, на сегодняшний день вопрос качественной оценки ауторегуляции мозгового кровообращения остаётся открытым. Сегодня можно условно выделить два подхода к оцениванию ауторегуляции головного мозга:

- Методы статической оценки;

- Методы динамической оценки.

К методам статической оценки ауторегуляции традиционно относят такие методы как ультразвуковое исследование (транскраниальная доплерография) и компьютерная томография [5]. Хотя такие методы достаточно точны и позволяют оценить состояние кровоснабжения головного мозга в определенный момент времени, они также имеют некоторые недостатки. Во-первых, методы статической оценки не подходят для мониторинга (непрерывной оценки), так как связанные с ними процедуры (ультразвуковое исследование, томография) требуют времени на их проведение. Во-вторых, данные процедуры требуют наличия дорогостоящего оборудования и квалифицированных врачей, способных провести данные процедуры.

Методы динамической (непрерывной) оценки позволяют выявить не только текущее состояние кровоснабжения головного мозга, но и определять динамику состояния процесса ауторегуляции. Методы динамической оценки в основном основываются на непрерывном анализе временных рядов основных показателей, таких как среднее артериальное давление и внутричерепное давление.

1.3.1 Индексы ауторегуляции мозга

К методам динамической оценки ауторегуляции мозга относится и одно из ключевых понятий, используемых в данной работе – индексы ауторегуляции мозга. Непрерывная оценка ауторегуляции головного мозга с помощью вычисления индексов ауторегуляции является одним из методов многопараметрического мониторинга состояния пациента с травмой головы. Индексы ауторегуляции мозга могут служить в качестве индикаторов нарушений в работе ауторегуляции головного мозга.

1.3.1.1 Pressure Reactivity Index

Pressure Reactivity Index (PRx) впервые был описан в работе исследовательской группы из Кембриджского университета в 1997 году [9]. Согласно определению, данному учёными, Pressure Reactivity Index может быть описан как коэффициент корреляции Пирсона между временными рядами средних значений артериального и внутричерепного давлений в скользящем окне. Pressure Reactivity Index может быть использован в качестве своеобразного индикатора нарушений в работе кровеносных сосудов головного мозга [10].

С момента появления данного индекса было проведено множество исследований, которые доказали прогностическую значимость в лечении черепно-мозговой травмы.

Исследование [10], проведенное около 10 лет назад, и включавшее наблюдение над 398 пациентами с травмами головы, показало, что высокие значения индекса PRx сильно

коррелируют с процентом неблагоприятных исходов. Напротив, наиболее низкие значения PRx встречались у группы пациентов с низким процентом неблагоприятного исхода.

Другое исследование [11], включало в себя ретроспективный анализ данных за 6 лет. Выборка включала в себя данные 36 детей в возрасте от 6 месяцев до 16 лет. Данное исследование также подтверждает, что значения PRx были значительно выше в группах пациентов с неблагоприятным исходом. В ходе данного исследования удалось выяснить, что есть прямая связь между утерянной ауторегуляцией и смертельным исходом пациента после травмы головы, а также примерно определить условные границы значений PRx, которые бы лучше всего разделяли выборку на пациентов с благоприятным и неблагоприятным исходом. Согласно данным, полученным в результате данного исследования, условными границами PRx являются значения 0 и 0,25. Значения меньше нуля свидетельствуют о стабильной ауторегуляции головного мозга, значения выше 0,25 обозначают утрату ауторегуляции, значения между 0 и 0,25 – промежуточное состояние ауторегуляции (дестабилизация ауторегуляции). Эти граничные значения PRx схожи с теми граничными значениями, которые обычно выделяют в ходе других исследований. Поскольку граничные значения отличаются от исследования к исследованию в меньшую или большую сторону, в данной работе было принято решение использовать по умолчанию граничные значения, представленные в таблице 1.

Таблица 1

Таблица интерпретации значений индекса PRx

Значение PRx	Интерпретация
От -1 до 0	Стабильная ауторегуляция
От 0 до 0,2	Дестабилизация ауторегуляции (промежуточное состояние)
От 0,2 до 1	Утраченная ауторегуляция

Еще одно исследование, которое проводилось с 1997 по 2002 (практически сразу после того, как ученые из Кембриджского университета представили PRx), включало в себя непрерывное наблюдение над внутричерепным давлением, артериальным давлением и значениями PRx у 193 пациентов с травмой головы [12]. Результаты данного исследования также выявили, что в группах пациентов с неблагоприятным исходом наблюдались повышенные значения PRx. Более того, при превышении значения PRx в 0,3 наблюдалось резкое повышение процента смертельного исхода у пациентов (с 20% до 70%).

Данный обзор включает описание лишь некоторых исследований, задействовавших измерение PRx для оценки ауторегуляции головного мозга, поскольку рассмотрение всех таких

исследований не является целью данной работы. Практическая значимость PRx также отражена во многих других исследованиях, не рассмотренных в данной работе.

Несмотря на свои очевидные преимущества, такие как простота вычисления и прогностическая значимость, использование индекса PRx также имеет свои особенности. Во-первых, поскольку для вычисления единицы PRx используются временные ряды средних значений артериального и внутричерепного давлений в скользящем окне, то возникает необходимость в предобработке поступающих данных, поскольку любые статистические выбросы в данных могут исказить результат вычислений. Во-вторых, получаемый в результате вычислений график PRx довольно «шумный» и требуется некоторое сглаживание.

Принимая во внимание результаты рассмотренных в данном разделе исследований, можно сказать, что использование индекса PRx показало свою прогностическую значимость в лечении больных с травмой головы. Значения PRx могут служить в качестве индикатора нарушений в системе кровоснабжения головного мозга.

1.3.1.2 Variation Reactivity Index

Исходя из описанных выше недостатков PRx возникла идея о разработке индекса, который бы отражал зависимость между вариациями внутричерепного и артериального давлений. Индекс, описанный впервые в работе Д. Москалевой, получил рабочее название Variation Reactivity Index [13]. Основной идеей создания индекса VRx была модификация индекса PRx с целью уменьшить чувствительность индекса к статистическим выбросам в исходных данных. Для достижения данной цели было принято решение об использовании скользящих коэффициентов вариации временных рядов артериального и внутричерепного давлений, вместо средних значений при вычислении коэффициентов корреляции, как это происходит в случае с PRx.

Работа Д. Москалевой была посвящена исследованию прогностической значимости данного индекса. Для достижения данной цели были сформулированы и выполнены следующие задачи: изучение влияния взаимодействия внутричерепного и артериального давлений на исход заболевания, соотнесение полученных в результате исследования значений индекса VRx со значениями расширенной шкалы исходов Глазго (далее – РШИГ), исследование пороговых значений индекса VRx, а также частоты встречаемости пороговых значений и соотнесение этих данных с фактическим состоянием пациентов [13].

Анализ данных проводился по архивным данным 34 пациентов, пребывавших в отделении интенсивной терапии НИИ НДХиТ. Результаты соотнесения вычисленных индексов VRx со значениями РШИГ показали, что при значениях VRx больше 0,6 вероятность неблагоприятного исхода у пациентов равнялась 100%. При значениях индекса VRx от 0,4 до

0,6 вероятность неблагоприятного исхода была несколько ниже – 41,6%. Значения VRx меньше 0,4 были в группе пациентов с вероятностью неблагоприятного исхода 15,7% [13].

Исходя из данных, полученных в результате исследования Д. Москалевой, было принято решение построить таблицу интерпретации значений для индекса VRx (аналогично таблице для индекса PRx). Таблица 2 содержит интервалы значений индекса VRx и их интерпретацию.

Таблица 2

Таблица интерпретации значений индекса VRx

Значение VRx	Интерпретация
От -1 до 0,4	Стабильная ауторегуляция
От 0,4 до 0,6	Дестабилизация ауторегуляции (промежуточное состояние)
От 0,6 до 1	Утерянная ауторегуляция

Несмотря на то, что в результате работы Д. Москалевой использование индекса VRx показало свою прогностическую значимость в определении состояния пациентов с травмами головы, на данный момент данный индекс все еще является малоизученным – одного исследования явно недостаточно. Требуется проводить больше исследований с использованием индекса VRx для подтверждения его достоверности в прогнозировании состояния пациентов с травмами головы.

1.3.2 Существующее программное решение

Несмотря на усилия, прилагаемые исследователями по всему миру к изучению индексов ауторегуляции, на данный момент в свободном доступе находится лишь одно программное решение, использующее вычисление индексов ауторегуляции для мониторинга состояния пациентов с травмами головы.

Разработка программного обеспечения ICM+ (Intensive Care Monitor) началась еще в 1980-1990-х годах прошлого столетия и продолжается до сих пор [14]. Разработка программы ведется той же исследовательской группой, которая предложила индекс PRx в 1997 году. Первая версия программы была публично представлена учёными из Кембриджского университета в 2004 году в рамках конференции ICP 12 в Гонконге. На данный момент программа ICM+ предоставляет широкие возможности по мониторингу основных показателей, таких как внутричерепное давление, артериальное давление, уровень насыщенности кислородом головного мозга, температура тела и т.д. На рисунке 1 представлен снимок экрана с запущенной программой ICM+.

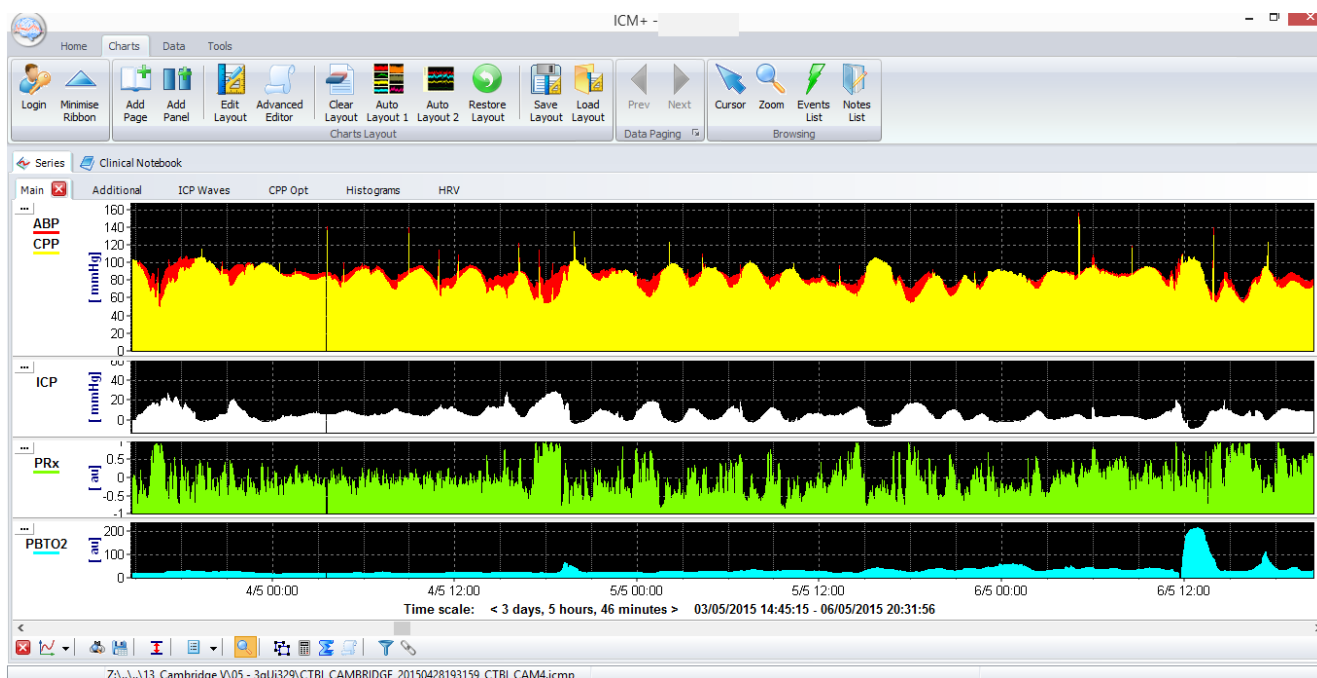


Рисунок 1 – скриншот программы ICM+

Несмотря на широкие возможности, которые предоставляет данное программное обеспечение, оно также имеет и некоторые недостатки.

По всей видимости, самым серьезным недостатком ICM+ является то, что данная программа является зависимой от целевой платформы: программа ICM+ представлена только в версии для операционной системы Microsoft Windows. Тогда, в начале 2000-ых, когда была представлена первая версия ICM+, это было обычным явлением, но сейчас, с развитием технологий и повышением уровня доступности мобильных устройств, многие отказываются от использования стационарных рабочих станций в пользу планшетных компьютеров с мобильной операционной системой. По данным Интернет-ресурса Statista, мировой объем поставок планшетных компьютеров на конец четвертого квартала 2017 года составил почти 60 миллионов единиц [15]. Согласно этим же данным, большую часть составили планшетные компьютеры на базе операционных систем Google Android и Apple iOS – 34,8 млн. и 13,2 млн. устройств соответственно. Доля устройств на базе Microsoft Windows составила всего 7,3 млн. устройств.

Другой особенностью ICM+ является высокая стоимость. Согласно данным с официального сайта программы, лицензия на использование данного программного обеспечения может стоить от 4000 до 7500 фунтов стерлингов [16].

Сравнение ICM+ с данным решением представлено в таблице 3.

Таблица 3

Сравнительная таблица ICM+ и данного программного решения

Критерий для сравнения	ICM+	Данное решение
Вычисление индексов PRx	Да	Да
Вычисление индексов VRx	Нет	Да
Стоимость программы	От 4000 до 7500 фунтов стерлингов	Бесплатно
Поддержка русского языка	Нет	Да
Целевая платформа	Microsoft Windows	Любая (требуется веб-браузер с поддержкой JavaScript)

Подводя итог данному разделу, можно сказать, что несмотря на то, что ICM+ обладает широкими возможностями для мониторинга ауторегуляции головного мозга, на текущий момент ICM+ имеет некоторые недостатки и существует потребность в многоплатформенном программном решении, которое позволяло бы непрерывно оценивать состояние ауторегуляции головного мозга у пациентов с помощью индексов ауторегуляции.

1.4 Разработка программного решения

Исходя из рассмотренных в данной главе особенностей индексов ауторегуляции, а также недостатков имеющегося программного обеспечения, можно сделать вывод о том, что необходимо: во-первых, реализовать вычисление обоих рассмотренных индексов ауторегуляции, и, во-вторых – разработать многоплатформенное программное решение.

Реализация индекса PRx позволит внедрить в программу хорошо изученный способ оценки ауторегуляции головного мозга. Использование индекса VRx может помочь избавиться от чувствительности к статистическим выбросам в исходных данных и лучше изучить прогностическую значимость данного индекса на практике.

Как уже обсуждалось ранее, носимые устройства с мобильными операционными системами становятся всё популярнее. Разработка многоплатформенного решения позволит врачам не зависеть от рабочего места и иметь доступ к информации о состоянии пациента в любое время в любом месте.

1.5 Задачи работы

В ходе выполнения работы были выделены следующие задачи:

Для решения поставленной цели были выделены следующие задачи:

1. Изучить современные способы оценки ауторегуляции головного мозга;
2. Провести обзор существующих решений и выявить их недостатки;
3. Определить методологию оценки ауторегуляции мозгового кровообращения;
4. Выявить требования к функциональным и нефункциональным характеристикам разрабатываемого решения;
5. Определить технологии, которые будут использованы при разработке программы;
6. Выявить требования к входным и выходным данным;
7. Разработать архитектуру программного решения;
8. Разработать структуру базы данных для хранения данных программы;
9. Разработать программу;
10. Разработать техническую документацию.

1.6 Выводы по главе

Ауторегуляция мозгового кровообращения является жизненно важным физиологическим процессом, направленным на поддержание адекватного кровотока в области головного мозга при условии колебаний внутричерепного и артериального давлений. Нарушенная ауторегуляция у пациентов после травмы головы может приводить к неблагоприятным последствиям. Непрерывная оценка процесса ауторегуляции у пациентов с травмой головы имеет большую практическую значимость, поскольку позволяет врачу оперативно реагировать на изменяющееся состояние пациента.

В условиях отсутствия единой методологии оценки ауторегуляции мозгового кровообращения индексы ауторегуляции могут служить в качестве индикаторов состояния кровоснабжения головного мозга и общего состояния пациента после травмы головы.

В данном разделе были рассмотрены два индекса ауторегуляции: PRx и VRx. Оба рассмотренных индекса ауторегуляции обладают как преимуществами, так и недостатками. К преимуществам использования индекса PRx можно отнести высокую прогностическую значимость, подтвержденную многочисленными исследованиями, к недостаткам – высокую чувствительность индекса к статистическим выбросам в исходных данных, что обуславливает высокую «шумность» получающихся графиков. Индекс VRx был предложен как решение части недостатков PRx. Хотя исследование, проведенное с использованием индекса VRx, показало его прогностическую значимость, требуется провести больше исследований для более тщательного изучения свойств данного индекса.

Имеющееся программное обеспечение, позволяющее производить оценку ауторегуляции с помощью индексов ауторегуляции, обладает некоторыми существенными недостатками.

Разрабатываемое мультиплатформенное программное решение позволит врачам непрерывно отслеживать динамику состояния ауторегуляции головного мозга пациентов с травмой головы. Полученная в результате использования программы информация может быть использована врачами для принятия клинических решений и назначения соответствующего лечения.

Глава 2. Особенности реализации

В данной главе подробно рассматриваются особенности реализации данного программного решения, описаны варианты использования программы с точки зрения конечных пользователей.

2.1 Общая архитектура проектного решения

Архитектура программы представляет собой классическое клиент-серверное решение. Использование клиент-серверной модели позволяет перенести всю работу по вычислениям на отдельную рабочую станцию (сервер). Учитывая тот факт, что клиентская часть программы может быть использована на мобильных устройствах, данный подход позволит сэкономить заряд аккумулятора и вычислительные ресурсы устройства, сведя использование вычислительных ресурсов исключительно к задачам по отображению поступающей от сервера информации и взаимодействию с серверной частью. Еще одним преимуществом клиент-серверной модели является снижение уровня связанности разных компонентов программы, что может быть полезно, например, при локализации и устранении возникающих ошибок. Данное программное решение состоит из следующих частей:

1. Серверная часть
2. Клиентская часть
3. База данных

Серверная часть программы отвечает за получение данных пациента, их обработку и последующую передачу информации в клиентскую часть программы. Также серверная часть программы предоставляет веб-сервис для взаимодействия с сервером из клиентской части программы. Клиентская часть программы содержит пользовательский интерфейс и предназначена для взаимодействия с серверной частью. Клиентская часть программы представляет собой веб-приложение, созданное с использованием технологий JavaScript. База данных служит для хранения некоторых настроек программы, внутренних картотек, а также для накопления данных пациентов за время работы программы. На рисунке 2 представлена общая схема взаимодействия компонентов программы.

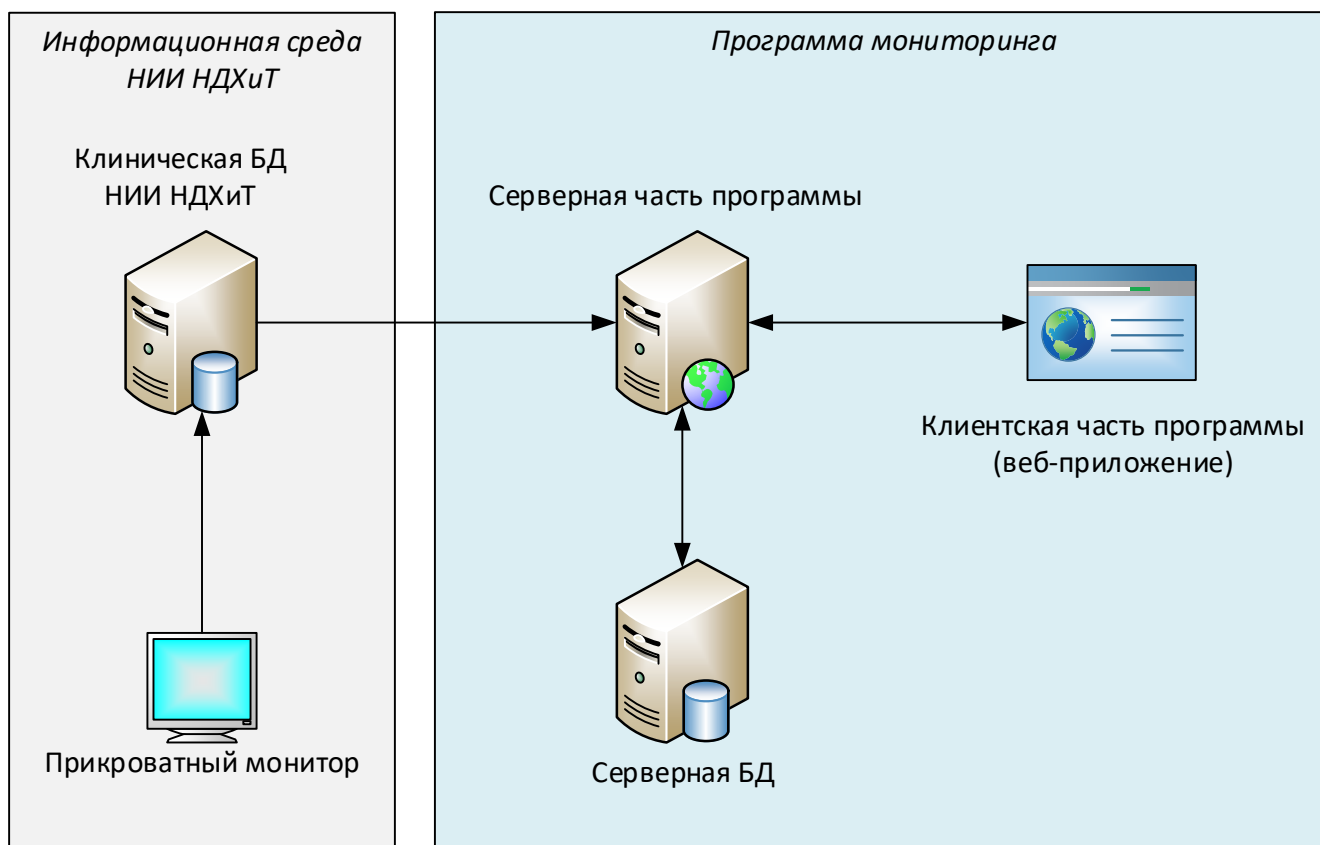


Рисунок 2 - Схема взаимодействия компонентов программы

Сервер получает из клинической базы данных НИИ НДХиТ следующую информацию о пациентах:

- Список пациентов, находящихся в отделении интенсивной терапии;
- Данные с прикроватных мониторов пациентов (артериальное давление, внутричерепное давление и др. параметры. Список получаемых из БД НИИ НДХиТ параметров может быть указан администратором в панели администрирования в клиентской части программы).

Часть данных, получаемых из БД НИИ НДХиТ, используется при вычислении индексов ауторегуляции. Сервер постоянно производит обработку поступающей информации и отправляет в клиентскую часть как информацию о вычисленных данных, так и необработанные данные с прикроватных мониторов пациентов, полученные из БД НИИ НДХиТ.

Серверная база данных используется для хранения внутренних данных программы, таких как:

- Информация о пользователях программы;
- Роли пользователей программы;
- Список соответствия пользователей ролям;

- Список пациентов;
- Настройки для каждого пациента;
- Список параметров, данные по которым получает сервер из БД НИИ НДХиТ;
- Значения необработанных данных пациентов;
- Относительные частоты встречаемости классов значений индексов ауторегуляции PRx и VRx.

2.2 Обработка данных

Серверная часть программы обрабатывает данные пациентов при их поступлении в программу.

2.2.1 Данные пациентов

Источником данных пациентов является клиническая база данных НИИ НДХиТ, в которую поступают оцифрованные данные с прикроватных мониторов, установленных в отделении интенсивной терапии НИИ НДХиТ. Данные пациентов описываются несколькими атрибутами:

- Порядковый номер наблюдения (ID);
- Время наблюдения;
- Идентификатор пациента;
- Идентификатор наблюдаемого параметра (показателя);
- Значение.

Список пациентов также хранится в базе данных НИИ НДХиТ. Серверная часть программы взаимодействует с клинической БД для получения списка пациентов и числовых данных пациентов.

2.2.2 Временной интервал между поступающими данными

Данные пациентов, поступающие в БД НИИ НДХиТ от прикроватных мониторов пациентов разделены между собой некоторым временным интервалом. На момент написания данной работы, сигналы с прикроватных мониторов оцифровывались каждые 5 секунд.

Временной интервал между поступающими данными пациентов является одним из ключевых параметров данной программы. Эта информация используется сервером при определении размера окна вычислений (количества наблюдений в целевой выборке). Данный

параметр хранится в серверной базе данных и может быть изменен администратором через панель администрирования в клиентской части программы.

2.2.3 Окно вычислений

Для вычисления индексов ауторегуляции PRx и VRx требуется производить оценку корреляции внутричерепного и артериального давлений в скользящем окне средних значений или коэффициентов вариаций соответственно. Окно вычислений в терминах данной программы – это некий контейнер ограниченного размера, содержащий в себе данные пациентов. При поступлении новых данных пациентов, они добавляются в начало контейнера. В случае если контейнер уже заполнен, данные из конца контейнера удаляются, а остальные данные смещаются, освобождая место в начале контейнера для нового наблюдения.

Одна из возможностей данной программы – определять размер скользящего окна для вычисления индексов ауторегуляции индивидуально для каждого пациента. Пользователь задает данное значение в виде количества секунд, используя специальный раздел в клиентской части программы. Для определения необходимого размера окна вычисления (кол-ва наблюдений в выборке) в виде размера контейнера используется следующая формула (1):

$$N = \frac{T_1}{T_2} \quad (1)$$

где N – это размер окна вычислений (количество наблюдений в выборке), T_1 – это заданный пользователем размер окна вычисления индекса в секундах, T_2 – это временной интервал (разрыв) между поступающими данными – глобальный параметр программы, указываемый администратором (см. предыдущий пункт). Результат деления T_1 на T_2 округляется в меньшую сторону. Полученное в итоге целое число и будет определять, какое количество наблюдений сможет вместить в себя окно вычислений при заданных параметрах.

Для примера: предположим, что для некоего пациента установлен размер окна для вычисления индекса PRx в 50 секунд, а временной разрыв между поступающими данными составляет 5 секунд. Тогда, подставив данные значения в формулу 1, нетрудно подсчитать, что окно вычисления индекса PRx в данном случае будет состоять из 10 таких 5-секундных наблюдений.

2.2.4 Вычисление индексов PRx и VRx

Процесс вычисления индексов ауторегуляции состоит из двух этапов:

- Этап предварительной обработки поступивших данных;
- Этап вычисления индекса.

На этапе предварительной обработки данных производится анализ временных рядов артериального и внутричерепного давлений. Для значений внутричерепного и артериального давлений в некотором временном промежутке (окне вычислений) вычисляются их средние значения. Средние значения артериального и внутричерепного давлений образуют собой два массива. На следующем этапе по ранее полученным массивам средних значений артериального и внутричерепного давлений рассчитывается коэффициент корреляции Пирсона (2)

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - M_x) \times (y_i - M_y)}{\sqrt{\sum_{i=1}^n (x_i - M_x)^2 \times (y_i - M_y)^2}} = PRx \quad (2)$$

где n – размер окна вычислений (кол-во наблюдений), $x_i \dots x_n$ – средние значения внутричерепного давления в окне вычислений, $y_i \dots y_n$ – средние значения артериального давления в окне вычислений, M_x, M_y – математические ожидания средних значений артериального и внутричерепного давлений внутри окна вычислений.

Полученное в результате вычислений значение и является индексом PRx. В данном случае подразумевается, что процесс вычисления индексов непрерывный, и, при поступлении в программу новых значений внутричерепного и артериального давлений, окно вычислений смещается, включая в себя новые данные. Процесс вычисления повторяется с использованием новых данных.

Вычисление индекса VRx производится аналогичным способом, за исключением того, что, в случае вычисления индекса VRx, используются не средние значения артериального и внутричерепного давлений в скользящем окне, а их коэффициенты вариации. То есть, $x_i \dots x_n$ и $y_i \dots y_n$ будут являться числовыми последовательностями, составленными из коэффициентов вариаций внутричерепного и артериального давлений.

2.2.5 Классификация значений индексов ауторегуляции PRx и VRx

В предыдущей главе на основании рассмотренных исследований была приведена интерпретация значений индексов ауторегуляции PRx и VRx.

Исходя из этой информации, значения индексов PRx и VRx можно условно разделить на три класса:

- Значения, обозначающие стабильный процесс ауторегуляции;
- Значения, обозначающие дестабилизацию ауторегуляции (промежуточное состояние);
- Значения, обозначающие утерянную ауторегуляцию.

Вынесение суждения о принадлежности полученного значения индекса PRx или VRx к тому, или иному классу происходит на основании числовых интервалов, которые определяют

диапазоны значений для каждого класса. Интервалы значений для интерпретации индексов ауторегуляции PRx и VRx, приведенные в таблицах 1 и 2 в первой главе, весьма условны и разнятся от исследования к исследованию. Именно поэтому, в данной программе реализована возможность задавать данные интервалы для интерпретации значений индексов ауторегуляции вручную. Программа, помимо расчета значений индексов ауторегуляции пациента, производит классификацию этих значений на три вышеприведенных класса значений.

2.2.6 Расчет относительных частот встречаемости значений разных классов индексов ауторегуляции PRx и VRx

Следующим этапом обработки данных является вычисление относительной частоты встречаемости вышеприведенных классов значений индексов ауторегуляции PRx и VRx за прошедший период. Иными словами, необходимо вычислить, сколько раз за прошедшее время полученные значения PRx или VRx попадали в «хороший», «промежуточный» или «плохой» диапазоны значений.

Данный процесс также является непрерывным (повторяющимся) и похож на вычисление значения индексов ауторегуляции PRx и VRx. Здесь также необходимо применить скользящее окно вычислений. В данном случае мы используем термин временной буфер. Пользователь программы может задать размер данного временного буфера отдельно для каждого пациента в секундах. В самой программе размер данного временного буфера – это (как и для окна вычислений индексов PRx и VRx) целое число, представляющее необходимый размер выборки. Вычисление размера выборки для временного буфера происходит аналогично вычислению размера скользящего окна вычислений индексов PRx и VRx – делением размера временного буфера, установленного для данного пациента и выраженного в секундах, на интервал между поступающими данными, заданный в глобальных настройках программы. Так, например, если временной интервал между поступающими данными составляет 5 секунд, а размер временного буфера для вычисления частот встречаемости классов значений индекса PRx задан 50 секунд, то, разделив 50 на 5 получаем 10. Это и будет размер выборки для вычисления частот встречаемости классов значений индекса PRx.

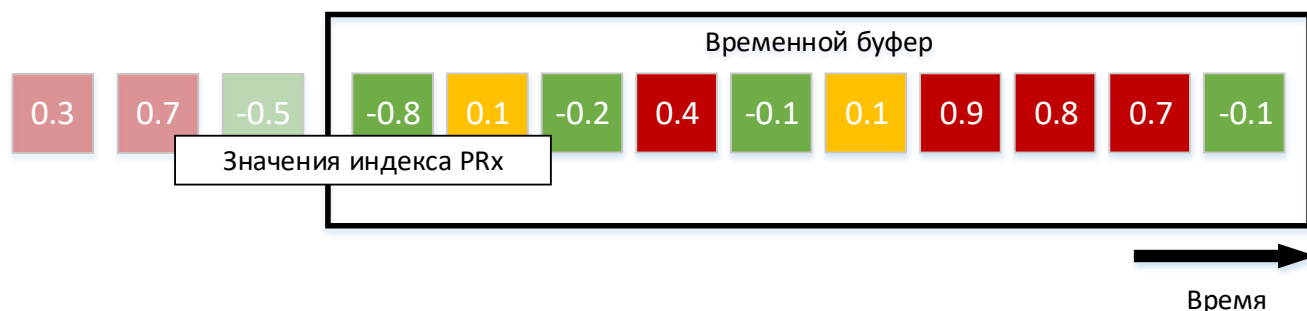


Рисунок 3 – временной буфера для вычисления частот встречаемости классов значений индекса PRx (пример)

Частота встречаемости значений определенного класса значений индексов PRx или VRx вычисляется отдельно для каждого класса значений по формуле (3).

$$\omega = \frac{N_A}{N} \quad (3)$$

где ω – это относительная частота встречаемости данного класса значений в пределах данного временного буфера, N_A – количество значений, отнесённых к данному классу в выборке, N – размер выборки (временного буфера).

Результатом данных вычислений станет частота встречаемости значений каждого из трех классов для индекса PRx или VRx.

2.2.7 Представление информации в программе

Основной информацией, которая используется для оценки состояния ауторегуляции головного мозга, в данной программе являются частоты встречаемости каждого из трёх ранее описанных классов значений индексов ауторегуляции PRx/VRx. В программе данная информация представляется в виде двух круговых диаграмм (для PRx и VRx). Каждая круговая диаграмма имеет три сектора разных цветов. Каждый цвет соответствует определенному классу значений индексов ауторегуляции PRx и VRx. В таблице 4 приведено соответствие.

Таблица 4

Соответствие цветов секторов круговых диаграмм классам значений индексов PRx и VRx

Цвет сектора	Класс значений
Зелёный	Стабильная ауторегуляция
Жёлтый	Дестабилизация ауторегуляции (промежуточное, неопределённое состояние)
Красный	Утраченная ауторегуляция

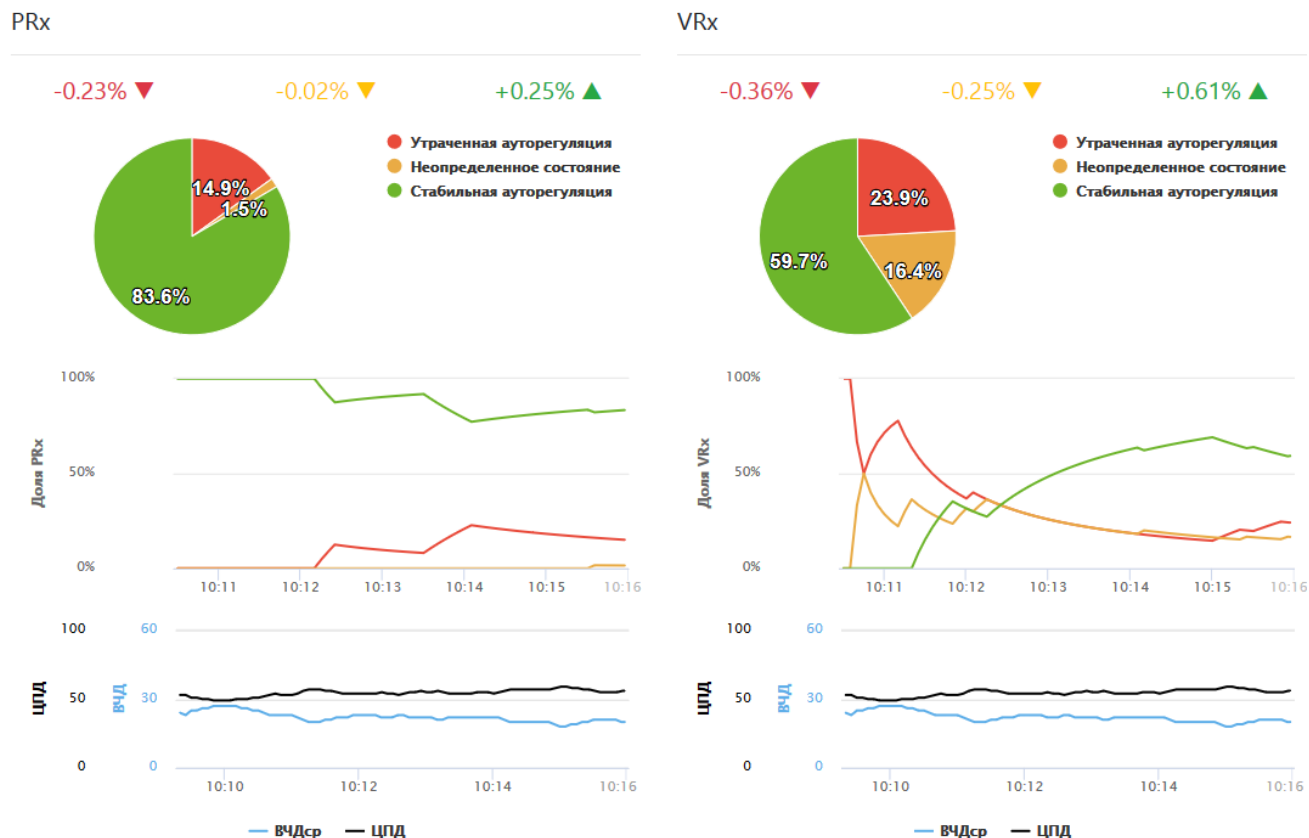


Рисунок 4 – информация об относительных частотах значений разных классов индексов PRx и VRx за прошедший период

Под каждой круговой диаграммой также находится её линейная репрезентация с разверткой по времени и графики с информацией о внутричерепном давлении и церебральном перфузионном давлении. Поскольку процесс вычисления относительных частот встречаемости классов значений индексов PRx/VRx непрерывный (повторяющийся), над круговыми диаграммами располагается информация об изменении удельной доли того или иного класса значений в выборке по сравнению с предыдущими показателями. Данная информация позволяет врачу понять динамику изменений в состоянии ауторегуляции.

Инициатором обновления информации является сообщение от серверной части программы, содержащее новые данные.

Графики необработанных значений индексов ауторегуляции могут быть достаточно «шумными», как это уже было упомянуто в первой главе. Использование классификации и вычисления относительных частот встречаемости тех или иных классов значений индексов PRx и VRx позволяет предоставить врачу более наглядную информацию как о состоянии ауторегуляции головного мозга на текущий момент, так и о динамике изменения состояния ауторегуляции за прошедшее время.

2.3 Программа мониторинга ауторегуляции головного мозга

В данном разделе рассматриваются основные возможности программы. Использование программы осуществляется с помощью веб-браузера с поддержкой технологии JavaScript.

2.3.1 Пользователи программы

2.3.1.1 Аутентификация

Аутентификация в программе – это механизм установления личности пользователя. В данной программе пользователь проходит аутентификацию, указывая свой ПИН-код в поле ввода на форме входа (рис. 5).

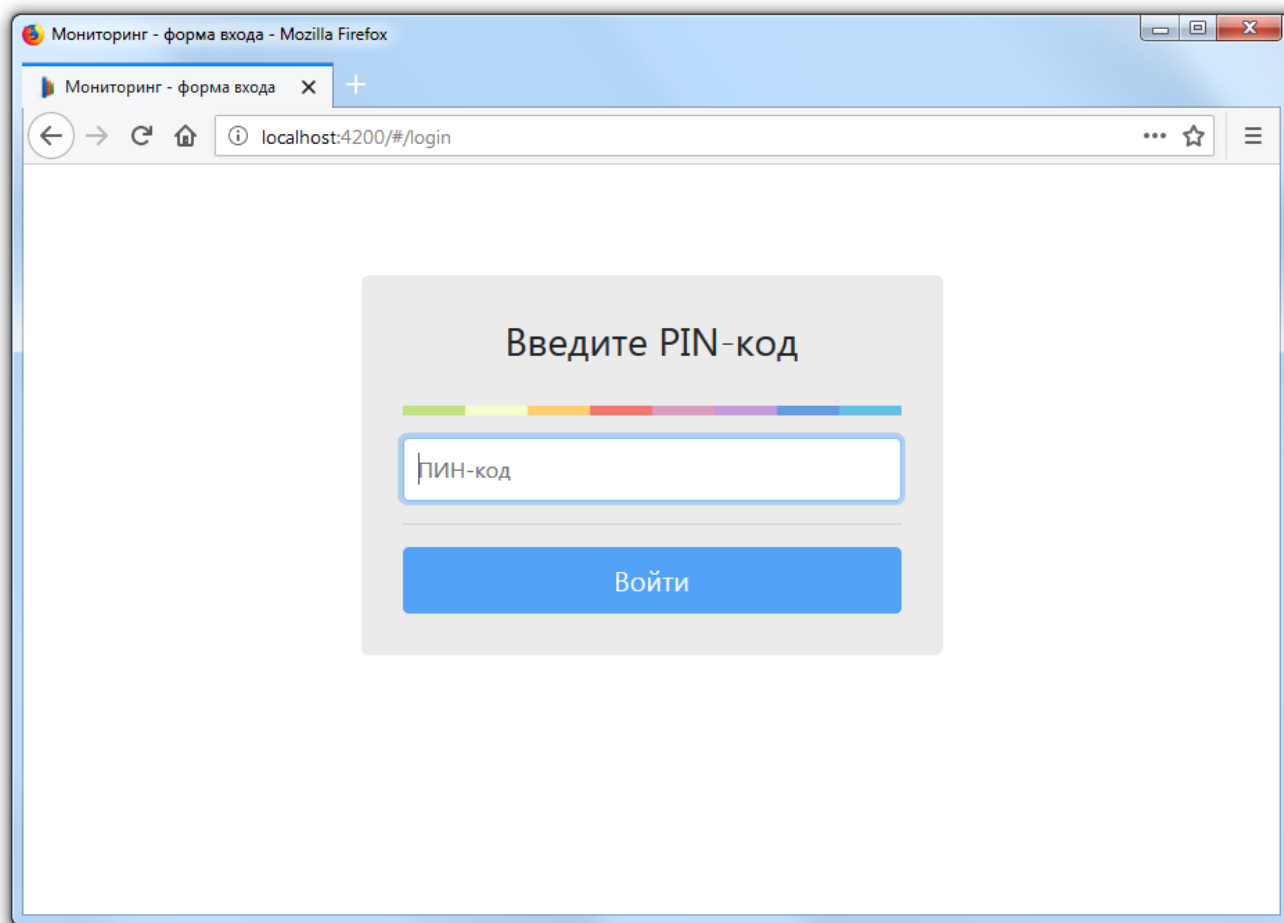


Рисунок 5 – форма входа в клиентской части программы

ПИН-код – это цифробуквенный набор символов, по которому осуществляется идентификация пользователей программы.

2.3.1.2 Роли пользователей

В программе существует разделение пользователей по ролям. Каждой роли

соответствует определенный набор прав (возможностей) на совершение различного рода действий в программе. В таблице 5 представлено описание типов пользователей в соответствии с ролями, используемыми в программе.

Таблица 5

Описание типов пользователей в программе в соответствии с ролями

Тип пользователя	Описание
Пользователь без роли	Пользователь без роли может проходить аутентификацию, имеет доступ к списку пациентов, главному окну мониторинга с информацией о пациенте, разделу загрузки данных (составление отчетов).
Врач	Имеет все те же права, что и пользователь без роли. Может изменять настройки вычисления индивидуально для каждого пациента, переключать видимость тех или иных показателей на странице мониторинга пациента.
Администратор	Имеет все те же права, что и пользователь без роли и врач. Имеет доступ к административной панели, с помощью которой может: создавать, редактировать, удалять пользователей, менять роли пользователей, определять список параметров, данные по которым будут запрашиваться из БД НИИ НДХиТ, изменять глобальные настройки программы.

2.3.2 Варианты использования

В данном подразделе подробно описываются основные варианты использования программы. На рисунке 6 представлена Use-Case диаграмма (диаграмма вариантов использования программы).

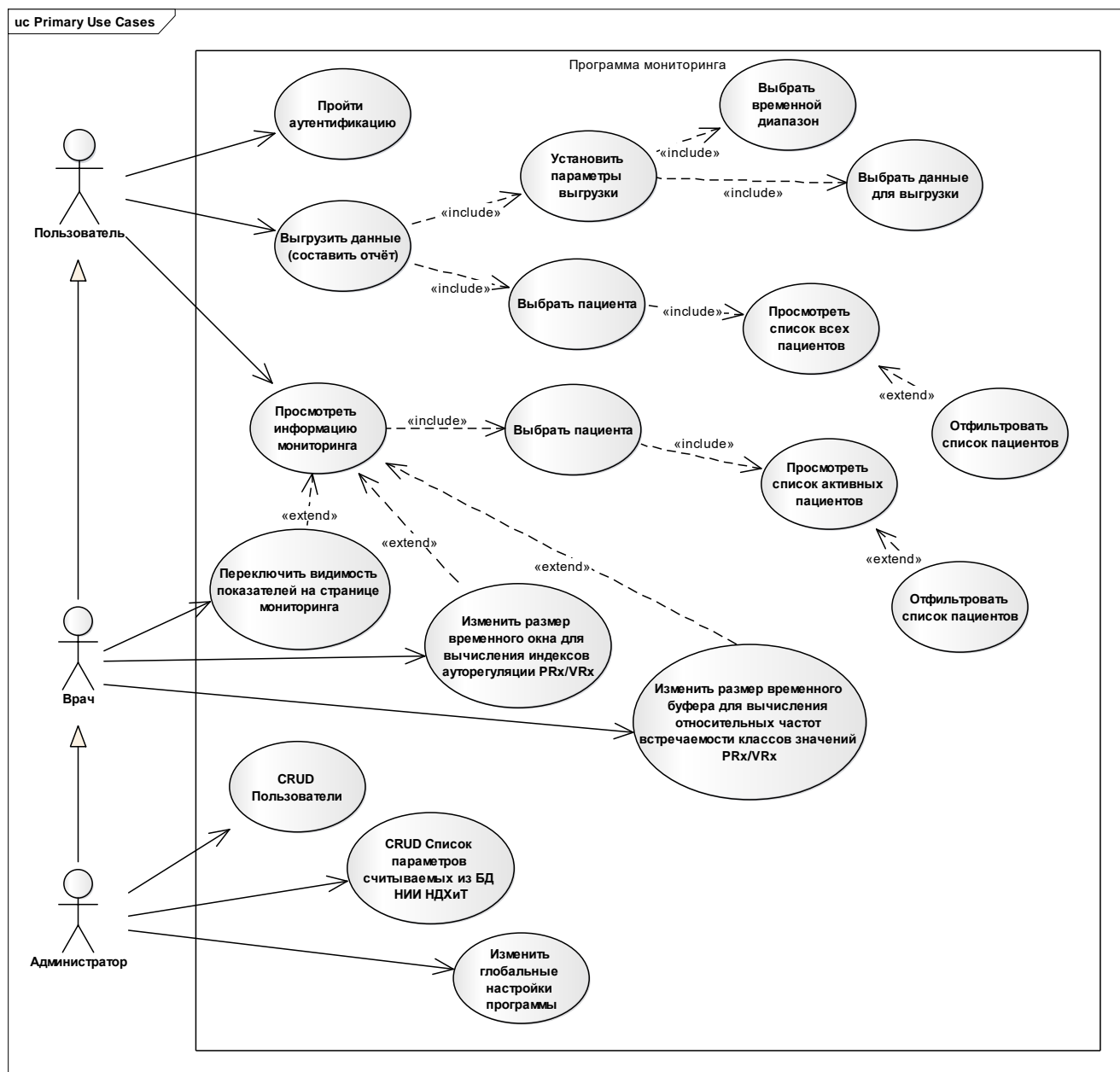


Рисунок 6 – диаграмма вариантов использования программы

2.3.2.1 Мониторинг пациента

Страница мониторинга пациента является основным разделом данной программы. В данном разделе отражаются сведения о текущем состоянии пациента в виде круговых диаграмм и линейных графиков, отображающих относительные частоты встречаемости тех или иных классов значений индексов PRx или VRx за прошедший период. Также здесь находится таблица, содержащая необработанные данные по ряду параметров. На рисунке 7 представлен снимок экрана со страницей мониторинга, открытой в веб-браузере.

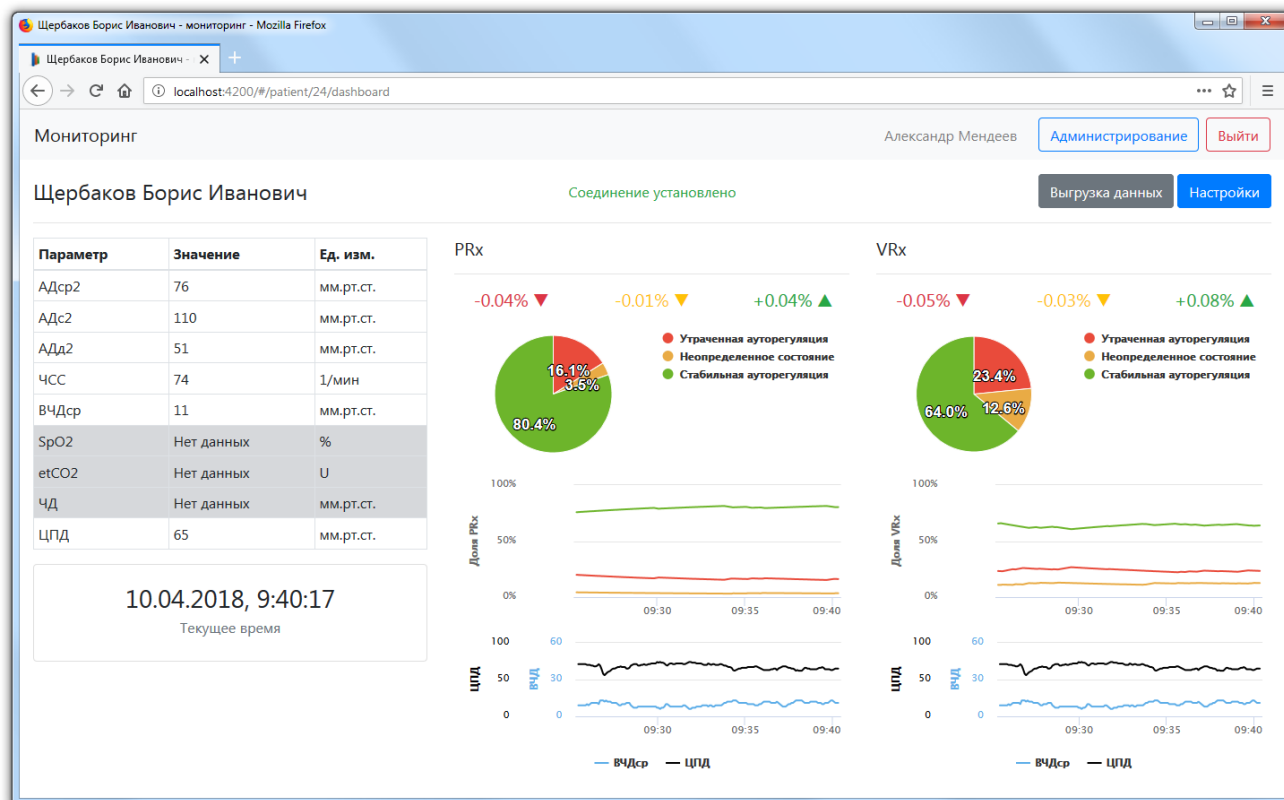


Рисунок 7 – страница мониторинга пациента

Слева сверху отображается ФИО выбранного пациента, чуть ниже – таблица основных показателей и их последних значений. Еще ниже – текущее время и дата. Текущее время определяется временем последних поступивших наблюдений. На изображении можно обнаружить, что по некоторым показателям данных не было. Это происходит тогда, когда пациент, размещенный в отделении интенсивной терапии НИИ НДХиТ, подключен не ко всем измерителям прикроватного монитора. Правую часть экрана занимает информация об относительных частотах встречаемости значений разных классов индексов PRx и VRx за прошедший период. Сверху информация изображена в виде круговых диаграмм, а снизу – в виде линейной развертки по времени. Еще ниже находятся графики церебрального перфузионного и внутричерепного давлений. Справа от ФИО пациента находится индикатор соединения с серверной частью. Еще правее – кнопка, ведущая на страницу выгрузки данных по данному пациенту, и кнопка, ведущая в раздел настроек вычисления для данного пациента. Наличие тех или иных элементов управления в интерфейсе клиентской части программы зависит от наличия или отсутствия ролей у пользователя. Например, пользователь, не имеющий роли врача, не увидит кнопку «Настройки» на данной странице.

Данные на этой странице обновляются автоматически без необходимости перезагружать страницу.

2.3.2.2 Изменение индивидуальных настроек пациента

Одной из важнейших особенностей данного решения является то, что пользователь может изменять настройки вычисления как отдельных единиц PRx и VRx, так и настройки для вычисления относительных частот встречаемости классов значений этих индексов. Для вычисления индексов ауторегуляции пользователь может задать величину (ширину) окна вычисления индекса в виде количества секунд. Аналогичным образом можно задать величину временного буфера для расчета относительных частот встречаемости классов значений индексов PRx и VRx. Данные настройки устанавливаются отдельно для каждого пациента в разделе настроек пациента (рис. 8).

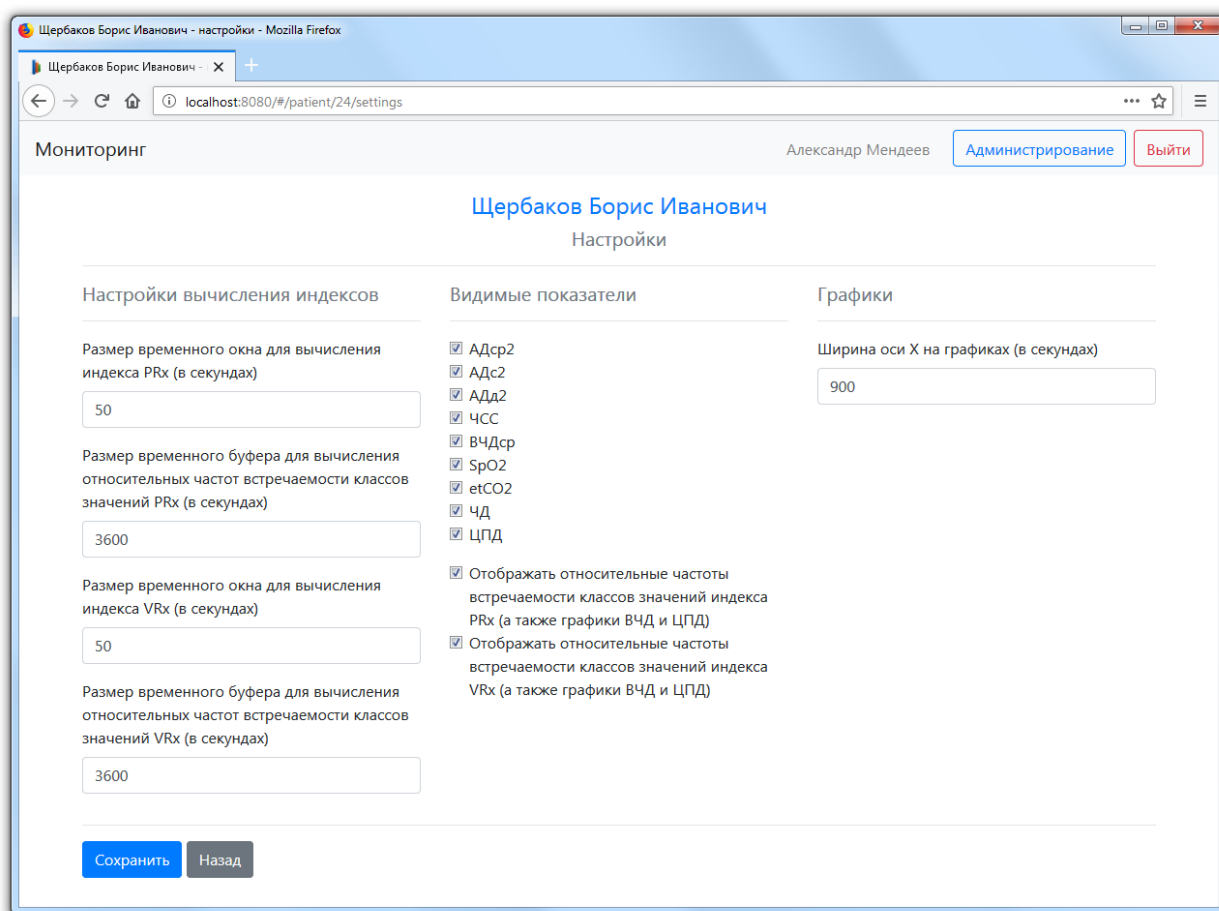


Рисунок 8 – раздел настроек пациента

Здесь же пользователь может переключить видимость тех или иных показателей на странице мониторинга (к примеру, отображать диаграмму относительных частот встречаемости классов значений только для PRx или для VRx), изменить ширину оси абсцисс на линейных графиках.

После сохранения настроек данные передаются в серверную часть программы и изменения вступают в силу.

Изменение настроек вычисления индексов ауторегуляции позволяет более точно настроить этот инструмент измерения ауторегуляции. Задав, например, более «широкое» окно для вычисления индексов, можно использовать более длинные волны значений внутричерепного и артериального давлений при вычислении индексов. Настройка размера временного буфера для вычисления относительных частот встречаемости классов значений PR_x и VR_x позволит гибко настроить их как для отображения более полной картины произошедших изменений в состоянии пациента (широкий временной буфер), так и для наблюдения за самыми последними изменениями в состоянии пациента (узкий временной буфер).

2.3.2.3 Составление отчётов (выгрузка данных)

В программе, помимо мониторинга текущего состояния пациента, есть также возможность составить отчёт, содержащий архивную информацию о пациенте. Во время своей работы программа не только считывает и обрабатывает данные, но также сохраняет их в серверной базе данных. Информация обо всех пациентах, которые были в программе также хранится в базе. Пользователь может выгрузить данные за определенный период по отдельному пациенту. Выгрузка осуществляется через специальный раздел в клиентской части программы. Чтобы попасть в раздел выгрузки данных, пользователь сначала переходит в раздел, содержащий список всех пациентов, о которых есть данные в программе (рис. 9).

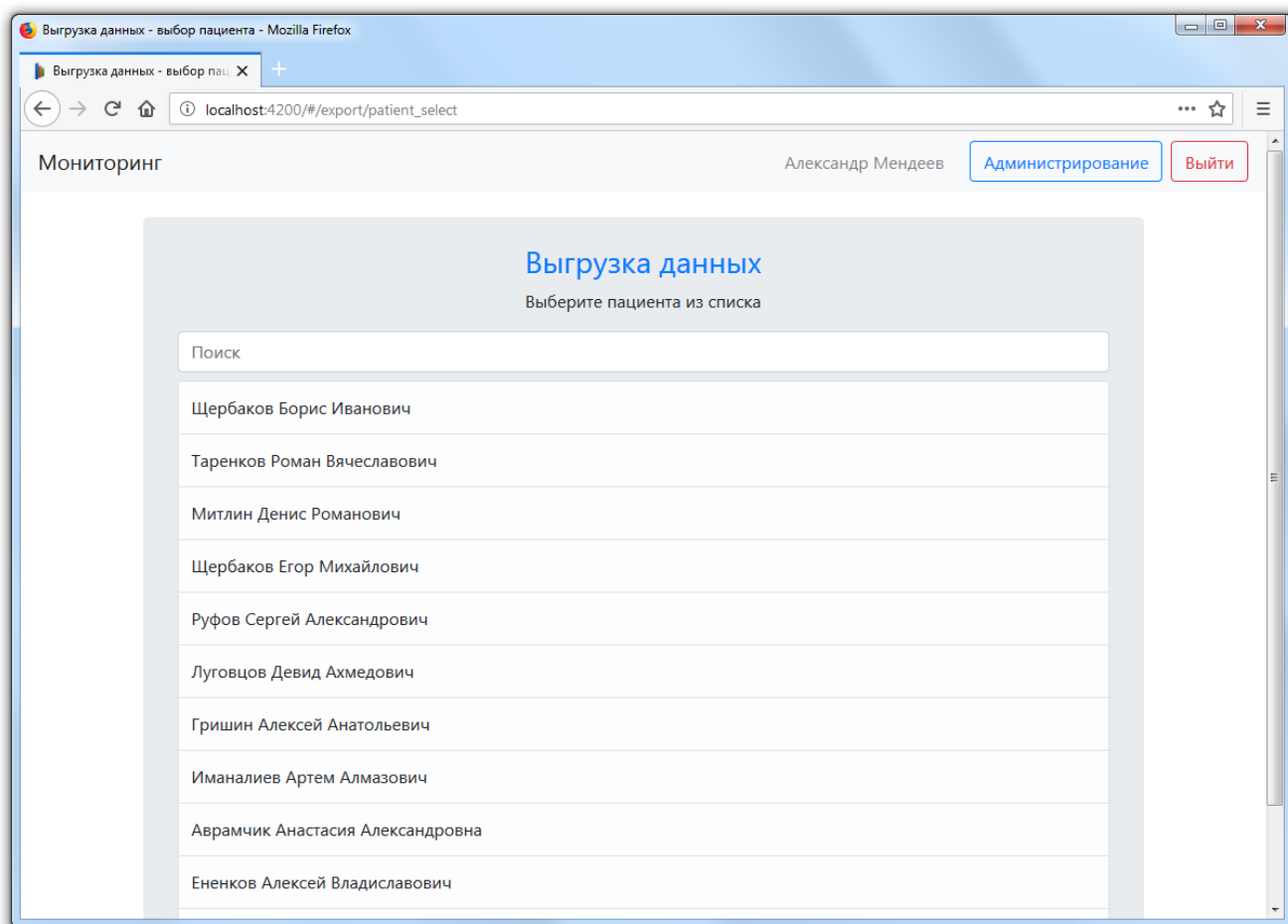


Рисунок 9 – список всех пациентов

Выбор пациента из списка инициирует переход пользователя к следующему шагу составления отчёта – настройке параметров отчёта (рис. 10).

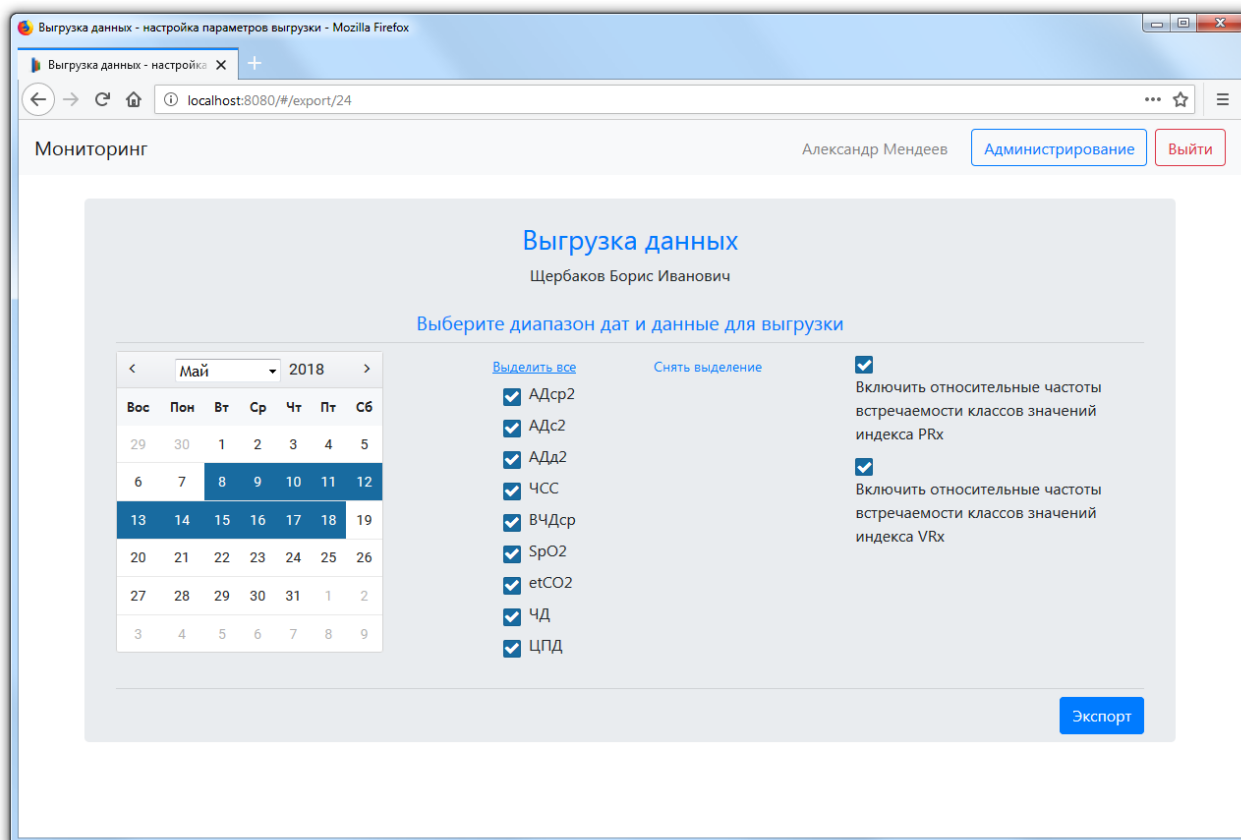


Рисунок 10 – настройка параметров формируемого отчёта

На данной странице пользователь выбирает за какой период требуется включить данные в отчёт, определяет какие данные следует включить. Всего пользователю доступно 3 вида данных для включения в отчёт:

- Необработанные данные пациента из БД НИИ НДХиТ;
- Данные об относительных частотах встречаемости классов значений индекса PRx за прошедший период;
- Данные об относительных частотах встречаемости классов значений индекса VRx за прошедший период.

Отчёт формируется в серверной части программы и, после формирования, отправляется в клиентскую часть программы. Браузер пользователя предлагает загрузить сформированный отчёт. Отчёт представляет собой таблицы с данными в виде файла книги Microsoft Excel (рис. 11).

The screenshot shows an Excel spreadsheet with the following data:

Дата и время	ЧСС	АДс2
2018-04-10 09:00:05.0	82	111
2018-04-10 09:00:10.0	83	110
2018-04-10 09:00:15.0	83	110
2018-04-10 09:00:20.0	84	109
2018-04-10 09:00:25.0	85	109
2018-04-10 09:00:30.0	85	108
2018-04-10 09:00:35.0	85	107
2018-04-10 09:00:40.0	85	107
2018-04-10 09:00:45.0	85	107
2018-04-10 09:00:50.0	85	107
2018-04-10 09:00:55.0	85	107
2018-04-10 09:01:00.0	85	107
2018-04-10 09:01:05.0	85	106
2018-04-10 09:01:10.0	84	105
2018-04-10 09:01:15.0	85	105

Рисунок 11 – пример сформированного отчёта в виде книги Microsoft Excel

Файл книги Excel содержит до трёх листов (в зависимости от выбранных параметров на этапе формирования отчёта). Первый лист книги содержит данные о времени и значениях необработанных данных, полученных из БД НИИ НДХиТ, второй и третий – данные об относительных частотах встречаемости разных классов значений индексов PRx и VRx в разные моменты времени.

2.3.2.4 Администрирование

Управление глобальными настройками программы, а также основными картотеками осуществляется администратором в панели администратора в клиентской части программы. Пользователь, имеющий роль администратора, может наблюдать кнопку «Администрирование», расположенную в верхней части окна программы (рис. 12).

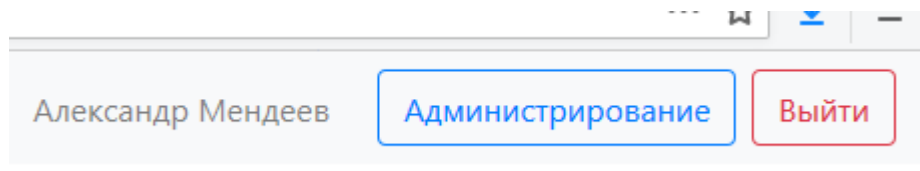


Рисунок 12 – кнопка перехода в административную панель программы

Административная панель состоит из двух частей: бокового меню, где пользователь выбирает раздел панели для просмотра, и рабочей области, в которой отображается выбранный раздел (рис. 13).

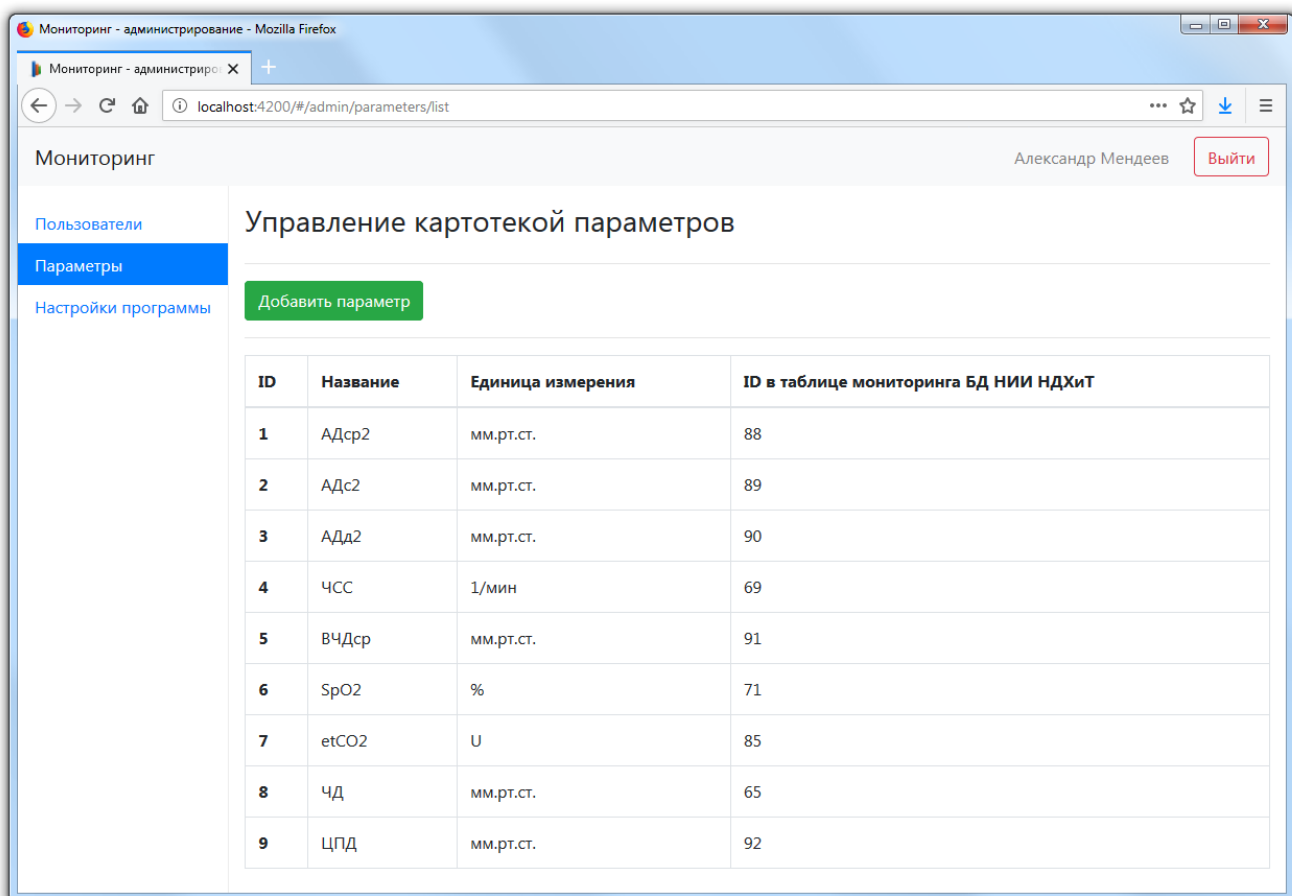


Рисунок 13 – административная панель

В разделе «Пользователи» администратор может добавлять новых пользователей, удалять пользователей и редактировать их (включая список ролей пользователя). В разделе «Параметры» находится картотека числовых параметров, данные по которым будут считываться серверной частью программы из БД НИИ НДХиТ. В данном разделе администратор вводит эти параметры, указывая также идентификатор соответствующего параметра из таблицы мониторинга клинической базы данных НИИ НДХиТ. Раздел «Настройки программы» содержит глобальные настройки программы, такие как: временной интервал между поступающими данными пациентов и соответствие параметров внутричерепного, артериального и перфузионного давлений записям в картотеке параметров. Это необходимо, чтобы предоставить программе информацию о том, какой из введенных в картотеку параметров, например, является артериальным давлением. Любые изменения в административной панели передаются в серверную часть программы и немедленно вступают в силу.

2.4 Выводы по главе

В данной главе были рассмотрены основные особенности реализации программного

решения. Помимо описания архитектуры данного решения, были описаны также подробности процесса вычисления индексов ауторегуляции головного мозга, был предложен способ оценки состояния ауторегуляции пациента на основании анализа частоты встречаемости значений индексов ауторегуляции и соотнесения данной информации с принятой классификацией значений индексов. Данный подход позволяет избавиться от одного из недостатков индексов ауторегуляции – шумности их графиков, и предоставить врачу более наглядную информацию как о текущем состоянии пациента, так и о динамике изменений состояния за прошедший период времени.

Были также рассмотрены варианты использования программы с точки зрения конечных пользователей. Представлено описание основных возможностей данной программы.

Глава 3. Технические детали реализации программного решения

Данная глава включает в себя описание технических деталей реализации, включая описание использованных программных средств и технологий.

3.1 Используемые инструменты разработки

3.1.1 Серверная часть программы

При разработке серверной части использовалась интегрированная среда разработки IntelliJ IDEA от компании JetBrains. Для работы с серверной базой данных, тестирования запросов SQL, был использован инструмент DataGrip от компании JetBrains. Для сборки проекта, а также для подключения необходимых библиотек, использовался инструмент Apache Maven. Для работы с клинической БД НИИ НДХиТ использовалось VPN соединение с сетью клиники.

3.1.2 Клиентская часть программы

Для разработки клиентской части была использована интегрированная среда разработки WebStorm от компании JetBrains. Отладка клиентской части программы производилась с помощью веб-браузеров Google Chrome и Mozilla Firefox, используя встроенный инструментальный разработчика этих программ. Для компиляции и сборки файлов клиентской части программы была использована утилита Angular CLI.

3.2 Используемые технологии

3.2.1 Серверная часть программы

Для разработки серверной части программы использовался фреймворк Spring Framework. данный фреймворк основан на реализации спецификаций Java EE, которые описывают архитектуру серверных приложений. Выбор платформы Java для реализации серверной части обусловлен, прежде всего, кроссплатформенностью данной технологии, что позволит запускать сервер на любом компьютере с установленной средой исполнения Java.

Для конфигурирования сервера используется надстройка фреймворка Spring – Spring Boot. Spring Boot упрощает развертывание серверной платформы, автоматизируя большую часть работы по конфигурации сервера. Полученное в результате сборки Spring приложение представляет архив, содержащий в себе Java EE приложение, которое может быть развернуто с использованием совместимого сервлет-контейнера, например, Apache Tomcat или Oracle Glassfish. Одной из особенностей Spring является то, что данный фреймворк позволяет собрать серверное Java EE приложение в один архив с расширением .jar, который далее может быть

запущен как обычное Java приложение. В данном программном решении применен именно этот подход, поскольку он позволяет конечному пользователю не устанавливать совместимый Java EE сервер или сервлет-контейнер, а ограничиться установкой Java Runtime Environment. При этом, разумеется, сервлет контейнер также используется, однако Spring встраивает его внутрь Java приложения. По умолчанию в качестве встроенного сервлет-контейнера используется Apache Tomcat. На рисунке 14 изображена структура приложения, получаемого при развертывании приложения Spring со встроенным сервлет-контейнером.

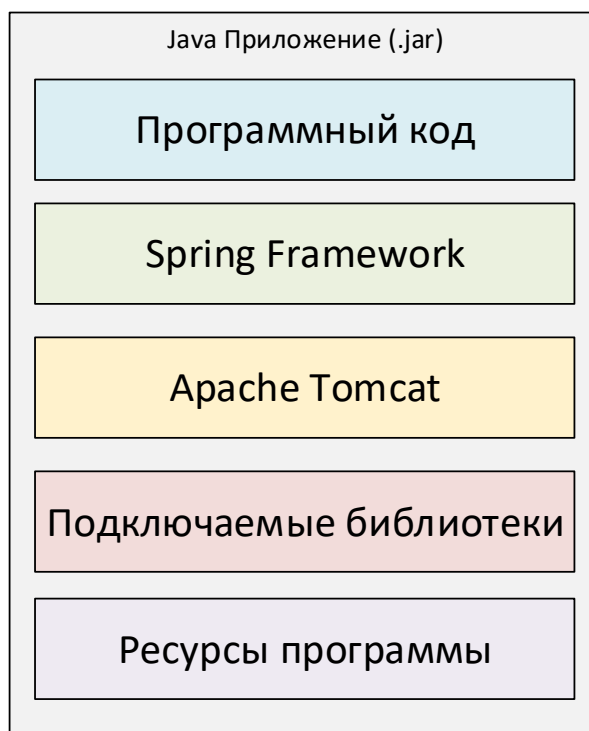


Рисунок 14 – структура собранного Spring приложения со встроенным контейнером сервлетов

Для взаимодействия с базой данных в данном проекте используется Spring Data и Hibernate. Spring Data основывается на спецификациях Java EE, а точнее – на спецификациях Java Persistence API (далее – JPA). JPA описывает программные интерфейсы для взаимодействия с реляционными базами данных при этом абстрагируя взаимодействие с конкретной базой данных. Спецификация JPA реализует концепцию ORM (от англ. Object-Relational Mapping), которая позволяет сопоставить ключевые абстракции программы, представленные в виде классов Java, с сущностями в базе данных.

Spring Data является в данном случае еще одной абстракцией более высокого уровня над JPA и позволяет, при минимальных усилиях программиста, реализовать, например, такой шаблон проектирования как репозиторий. В данной программе репозитории используются для взаимодействия с ключевыми картотеками программы. Для программной реализации спецификаций JPA используется библиотека Hibernate.

Для реализации немедленной реакции вычислительных компонентов программы на изменения в настройках программы, выполняемых пользователем в клиентской части программы, используется библиотека RxJava, позволяющая внедрить технологии реактивного программирования в Java приложение.

3.2.2 Клиентская часть программы

Клиентская часть программы была разработана с использованием фреймворка Angular, позволяющего создавать веб-приложения на основе технологий JavaScript. Готовое приложение Angular представляет собой набор файлов, содержащих программный код на языке JavaScript. Исполнение данных файлов JavaScript в веб-браузере приводит к генерации HTML кода страницы в реальном времени.

Веб-приложение, написанное с использованием фреймворка Angular, состоит из компонентов, каждый из которых представляет собой программный код, написанный на языке TypeScript. Каждый компонент содержит:

- Программный код, контролирующий поведение компонента, его реакцию на события (файл TypeScript);
- Файл HTML, содержащий визуальную разметку компонента;
- Файл CSS, описывающий стили, применяемые к визуальной разметке компонента.

Одной из ключевых особенностей системы компонентов Angular является поддержка иерархии компонентов. Так один компонент может быть вмещающим или дочерним для другого компонента. Вставка компонента осуществляется в тексте разметки с помощью тега HTML, состоящего из названия данного компонента. Например: “<component-a></component-a>”.

Связь данных контроллера компонента с его визуальным представлением достигается путем внедрения в текст разметки компонента специальных символьных конструкций. К примеру, текст “{{var1}}” в исходном коде страницы будет заменен на значение переменной var1, принадлежащей данному контроллеру. Фреймворк Angular во многом базируется на принципах реактивного программирования и использует компоненты библиотеки RxJS.

Для отрисовки графиков используется библиотека HighCharts. Также в реализации графического интерфейса были использованы некоторые компоненты библиотеки PrimeNG. В качестве таблицы стилей используется Twitter Bootstrap 4.

В качестве СУБД в данном решении используется Oracle MySQL Community Edition.

В базе данных хранятся основные картотеки программы, настройки программы, а также данные пациентов, собранные за время работы программы (необработанные данные основных показателей, запрашиваемых из БД НИИ НДХиТ, значения долей PRx и VRx). На рисунке 15 представлена диаграмма отношений сущностей серверной БД.

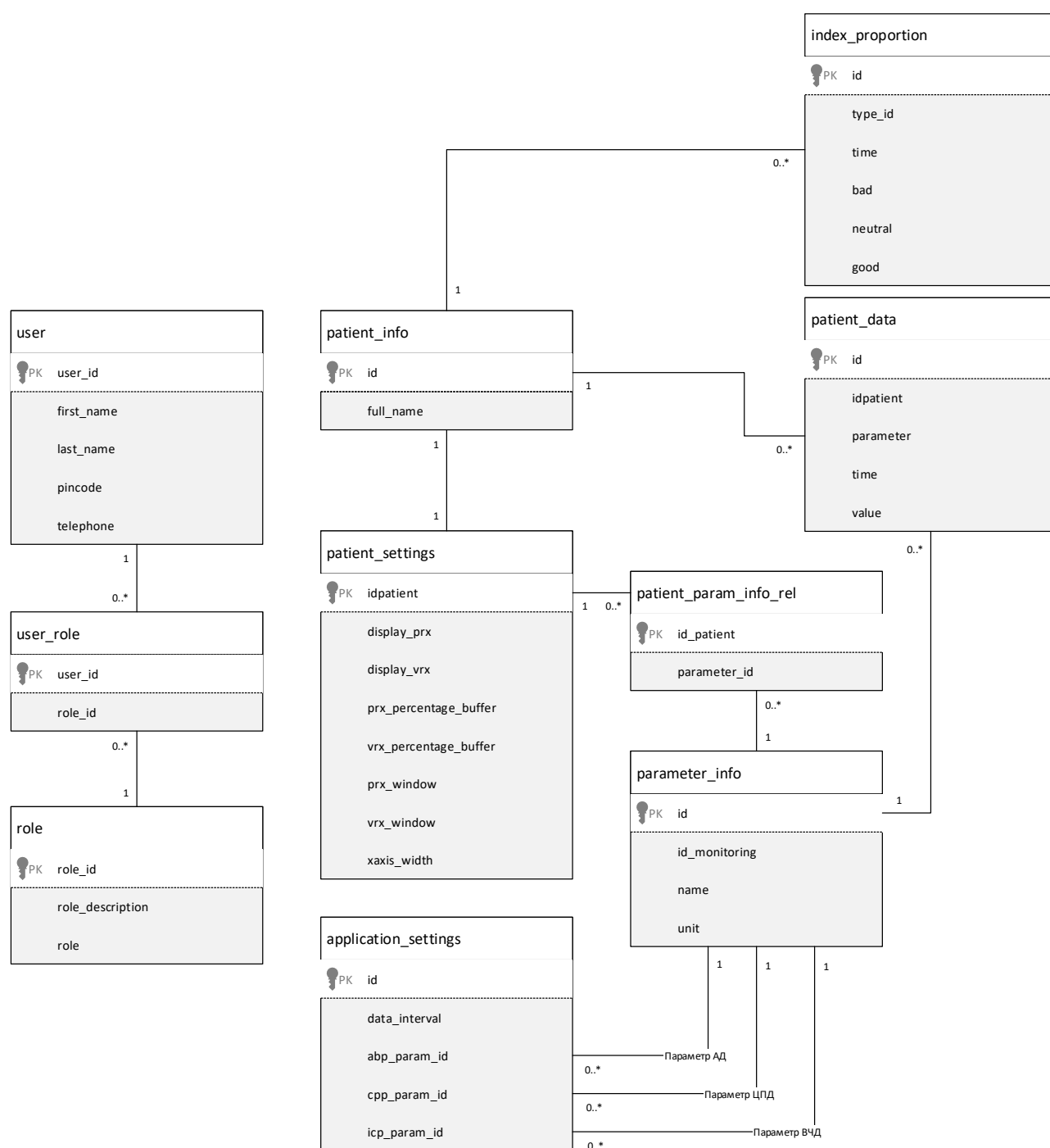


Рисунок 15 – диаграмма отношений сущностей серверной БД

В таблице ниже представлено описание сущностей БД.

Таблица 6

Описание сущностей серверной БД

Название сущности	Описание
user	Описывает сущность пользователя программы.
role	Описывает сущность роли пользователя.
user_role	Вспомогательная сущность, предназначенная для разнесения отношения «много ко многим» в связи сущностей пользователя и роли. Сущность сопоставляет идентификатор пользователя с идентификатором соответствующей роли.
patient_info	Описывает сущность пациента.
patient_settings	Описывает сущность настроек пациента, которые врач может менять через пользовательский интерфейс программы. Содержит параметры для вычисления индексов ауторегуляции, для вычисления относительных частот встречаемости классов значений индексов PRx и VRx и т.д.
parameter_info	Описывает сущность числового параметра, значения по которому будут считываться из БД НИИ НДХиТ (АД, ВЧД, ЦПД и т.д.).
patient_param_info_rel	Вспомогательная сущность, предназначенная для разнесения отношения «много ко многим» в связи сущностей настроек пациента и параметров. Набор этих сущностей определяет, какие параметры (показатели) будут отображаться в таблице на странице мониторинга пациента.
patient_data	Сущность для хранения необработанных данных пациента, считываемых из БД НИИ НДХиТ (значения параметров).
index_proportion	Сущность для хранения относительных частот встречаемости значений разных классов индексов PRx и VRx пациента.
application_settings	Сущность для хранения глобальных настроек программы, таких как временной интервал между поступающими данными пациентов и сопоставление параметров АД, ВЧД и ЦПД записям в картотеке параметров.

3.4 Работа с данными пациентов

3.4.1 Источник данных пациентов

Поскольку НИИ НДХиТ не предоставляет каких-либо унифицированных программных интерфейсов для доступа к данным, данные пациентов, содержащие значения жизненно важных

показателей (параметров) пациентов в определенный момент времени, поступают в программу из клинической базы данных НИИ НДХиТ.

3.4.2 Клиническая база данных НИИ НДХиТ

БД НИИ НДХиТ представляет собой СУБД Microsoft SQL Server. Данные в клиническую базу поступают от прикроватных мониторов пациентов. Также в данной базе хранится список пациентов, находящихся в отделении интенсивной терапии и полная картотека параметров клинических данных пациентов, по которым ведется наблюдение. На рисунке 16 представлен фрагмент модели отношений в БД НИИ НДХиТ на котором можно видеть основные сущности внешней БД, с которыми работает данная программа.



Рисунок 16 – фрагмент общей модели отношений сущностей в БД НИИ НДХиТ

Как уже было упомянуто, во внутренней БД программы также есть картотека параметров. Здесь важно упомянуть, что в картотеке параметров внутренней БД сущность параметра имеет атрибут, обозначающий идентификатор параметра в таблице мониторинга (“id_monitoring”). При считывании данных из БД НИИ НДХиТ данный параметр сопоставляется значению атрибута идентификатора параметра в сущности данных пациента, считываемой из клинической БД.

3.4.3 Конфигурация соединения

Данная программа взаимодействует одновременно с двумя базами данных: клинической БД НИИ НДХиТ и внутренней БД программы. При работе с обеими базами данных используются средства фреймворка Spring.

Конфигурация соединения с базами данных производится в файле конфигурации application.properties, где указываются параметры соединения. Параметры соединения затем используются в классе DBConfig (рис. 17), который производит настройку соединения с базами данных на основании значений конфигурации.

```

14  @Configuration
15  public class DBConfig {
16
17      @Bean(name = "smkm")
18      @ConfigurationProperties(prefix = "spring.smkm")
19      public DataSource smkmDataSource() { return DataSourceBuilder.create().build(); }
20
21      @Primary
22      @Bean(name = "sdb")
23      @ConfigurationProperties(prefix = "spring.sdb")
24      public DataSource sdbDataSource() { return DataSourceBuilder.create().build(); }
25
26
27
28
29      @Bean(name = "jdbcSMKM")
30      @Autowired
31      public JdbcTemplate smkmJdbcTemplate(@Qualifier("smkm") DataSource smkmDataSource) {
32          return new JdbcTemplate(smkmDataSource);
33      }
34
35      @Bean(name = "jdbcSDB")
36      @Autowired
37      public JdbcTemplate sdbJdbcTemplate(@Qualifier("sdb") DataSource sdbDataSource) {
38          return new JdbcTemplate(sdbDataSource);
39      }
40  }

```

Рисунок 17 – класс DBConfig

Помеченный аннотацией `@Configuration`, данный класс вносит изменения в конфигурацию сервера, создаваемую средствами Spring Boot. Также данная аннотация означает, что жизненным циклом объекта данного класса будет управлять контейнер фреймворка Spring. Использование управляемых компонентов позволяет запрашивать экземпляры этих классов из объектов других управляемых компонентов без необходимости создавать их вручную. В этом проявляется принцип внедрения зависимостей (Dependency Injection). Внедрение зависимостей является одним из принципов шаблона проектирования для построения слабосвязанных систем – IoC (от англ. Inversion of Control) [17]. Фреймворк Spring имеет встроенный контейнер IoC, который управляет управляемыми компонентами приложения. Внедрение зависимостей используется в данной программе в большинстве компонентов. Данный вариант соединения с БД наиболее предпочтителен, поскольку, в случае использования управляемых компонентов, фреймворк Spring самостоятельно управляет активными соединениями и их количеством (пул соединений).

Данный класс имеет 4 метода, аннотированных аннотацией `@Bean`, которая обозначает, что метод, помеченный данной аннотацией, является источником (правилом) создания управляемых Java Bean компонентов с указанным названием. Первые два метода создают компоненты типа `DataSource`, представляющие источник данных для соответствующей конфигурации соединения с БД. Два других метода используют данные соединения, уточняя название соответствующих управляемых компонентов в аннотации `@Qualifier`, для создания компонентов типа `JdbcTemplate`. Класс `JdbcTemplate` является частью фреймворка Spring и предоставляет методы для работы с БД посредством выполнения SQL запросов.

3.4.4 Запрос данных

Запрос данных пациентов из клинической БД НИИ НДХиТ происходит в классе DBPatientDataSource, который наследует абстрактный класс AbstractPatientDataSource. Класс AbstractPatientDataSource в свою очередь является абстрактным источником данных пациента и реализует обобщённый интерфейс PatientDataSource. Иерархия данных классов приведена на рисунке ниже.

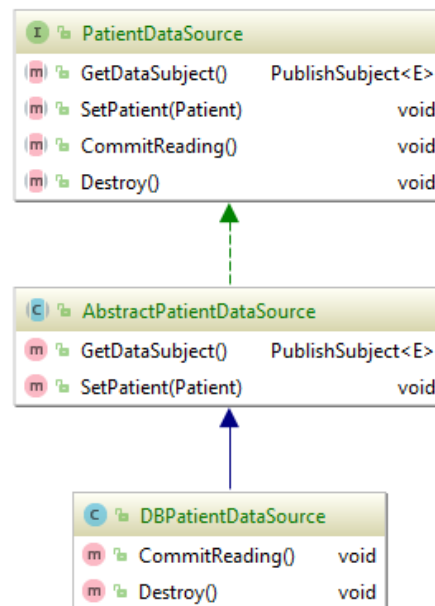


Рисунок 18 – иерархия классов, представляющих источник данных пациента

Класс DBPatientDataSource использует объект класса Timer из встроенного в Java пакета java.util для организации повторяющегося отложенного считывания данных из БД НИИ НДХиТ.

Алгоритм считывания новых данных пациента может быть описан следующим образом: изначально, после запуска программы, для каждого пациента считывается самая последняя запись о данных этого пациента. Затем, идентификатор считанной в последний раз записи используется программой для определения новых поступивших данных. Далее программой считываются только те данные, которые имеют идентификатор больший, чем считанный программой в последний раз. Выбор такого подхода обусловлен тем, что атрибут времени в таблице данных пациента клинической базы НИИ НДХиТ не имеет индекса. Следовательно, осуществлять выборку, задавая в качестве одного из критериев время измерения, было бы невозможно, поскольку такая выборка заняла бы слишком много времени из-за огромного количества записей в БД НИИ НДХиТ. Использование атрибута идентификатора записи позволяет решить эту проблему, поскольку данный атрибут имеет индекс в базе данных, а также принимает значения по возрастанию.

Считывание данных пациента происходит с помощью SQL запросов к клинической БД, используя объекты класса JdbcTemplate. Одна из перегрузок метода query класса JdbcTemplate подразумевает передачу объекта класса, реализующего обобщенный интерфейс ResultSetExtractor. Данный интерфейс служит для извлечения данных из результатов SQL запроса и приведения их к соответствующему типу. В программе есть класс PlainNumericDataResultSetExtractor, который реализует вышеупомянутый интерфейс и служит для обработки поступивших в результате запроса данных. Данный класс трансформирует набор поступивших в результате запроса данных в коллекцию данных типа PlainData, который представляет собой класс, описывающий необработанные данные пациента.

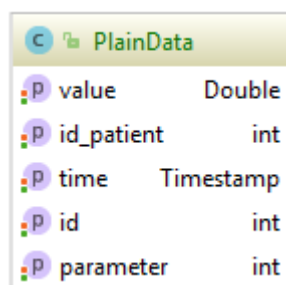


Рисунок 19 – класс PlainData

Класс PlainNumericDataResultSetExtractor также производит предварительную обработку поступающей информации, удаляя из считанного набора данных те записи, которые, например, содержат ошибки или строковые значения.

Класс DBPatientDataSource наследует атрибут dataSubject типа PublishSubject от класса AbstractPatientDataSource. Класс PublishSubject из состава библиотеки RxJava реализует сразу два шаблона проектирования: Observer (наблюдатель) и Observable (наблюдаемый). Объекты данного класса представляют собой некий «канал» сообщений определенного типа, на который могут подписываться другие объекты. Применительно к классу DBPatientDataSource поле dataSubject является каналом информации пациента. Считав данные из базы данных, класс DBPatientDataSource отправляет эти данные в dataSubject, уведомляя всех подписчиков данного канала о новых данных пациента.

3.4.5 Обработка данных

Основным модулем обработки данных пациента является класс Patient. Данный класс помечен аннотациями @Component и @Scope("prototype"). Первая аннотация помечает данный класс как управляемый компонент Spring (Spring Bean). Вторая аннотация указывает на то, что жизненный цикл данного компонента не ограничен одним экземпляром класса и при каждом

запросе экземпляра данного компонента IoC контейнер Spring будет создавать новый экземпляр данного класса.

Данный класс имеет поле типа `PatientDataSource<PlainData>`, которое представляет собой источник необработанных данных для пациента. Источник данных запрашивается классом `Patient` в его конструкторе у экземпляра класса `PatientDataSourceProvider`, который предоставляет соответствующий источник данных на основании конфигурации программы. На данный момент единственным источником данных пациента является экземпляр класса `DBPatientDataSource`. Получив источник данных, класс `Patient` подписывается на поле типа `PublishSubject<PlainData>` этого класса и предоставляет метод-обработчик для входящей информации.

Абстрактный класс `AbstractPatientDataProvider<E>` описывает обработчик данных пациента. В данном классе имеется поле, содержащее ссылку на объект класса `Patient` и поле типа `PublishSubject<E>`, представляющее собой канал выходных данных данного обработчика. Реализациями данного класса являются классы `PRxCalculator` и `VRxCalculator`, которые отвечают за вычисление индексов ауторегуляции `PRx` и `VRx` соответственно. Также к наследникам, реализующим данный класс, относятся классы `PRxProportionCalculator` и `VRxProportionCalculator`, которые отвечают за вычисление значений долей `PRx` и `VRx` за прошедший период. Классы `DeltaPRxCalculator` и `DeltaVRxCalculator` также являются наследниками класса `AbstractPatientDataProvider` и отвечают за вычисление разности между текущими удельными частотами встречаемости классов значений `PRx` и `VRx` и предыдущими показателями.

Для статистической обработки данных используется библиотека `Apache Commons Math` версии 3.6.1. Класс `DescriptiveStatistics` из данной библиотеки используется для реализации окна вычислений значений `PRx` и `VRx`, а также для временного буфера вычислений частот встречаемости классов их значений. Данный класс не только хранит числовые данные, но и позволяет задать количество хранимых данных. При нехватке места в контейнере хранения `DescriptiveStatistics`, старые данные удаляются, а текущие данные смещаются, освобождая место для новых. Данное свойство данного контейнера позволяет использовать его в качестве окна для вычислений. При этом, данный класс предоставляет методы для получения статистических данных о хранящихся в данном контейнере значениях. Для расчета коэффициента корреляции Пирсона используется метод `correlation` класса `PearsonsCorrelation`, входящего в состав данной библиотеки.

3.4.6 Передача данных в клиентскую часть программы

Для передачи данных пациента в клиентскую часть программы используется протокол передачи данных WebSocket совместно с протоколом STOMP. Конфигурация WebSocket соединения в серверной части программы производится в классе WebSocketConfig.

```
9  @Configuration
10 @EnableWebSocketMessageBroker
11 public class WebSocketConfig extends AbstractWebSocketMessageBrokerConfigurer {
12     @Override
13     public void registerStompEndpoints(StompEndpointRegistry stompEndpointRegistry) {
14         stompEndpointRegistry.addEndpoint( ...strings: "/data").setAllowedOrigins("*").withSockJS();
15     }
16 }
```

Рисунок 20 – содержимое класса WebSocketConfig

Передача данных через WebSocket происходит в объектах класса PatientDataWebSocketEmitter. Данный класс подписывается на канал информации отдельного пациента в классе Patient и отправляет данную информацию, используя метод convertAndSend объекта класса SimpMessagingTemplate, предоставляемого фреймворком Spring. Использование протокола STOMP предполагает наличие некоего «канала» или «очереди» сообщений. При отправке данных пациента, данные отправляются по каналу, соответствующему идентификатору данного пациента. При этом, в зависимости от данных, используются разные подканалы передачи информации. Так, например, канал информации, содержащий данные о внутричерепном давлении пациента с идентификатором 44, будет иметь следующий вид: “/topic/data/44/icp”. При передаче объектов Java по протоколу STOMP происходит их автоматическое приведение в текстовый вид в формате JSON.

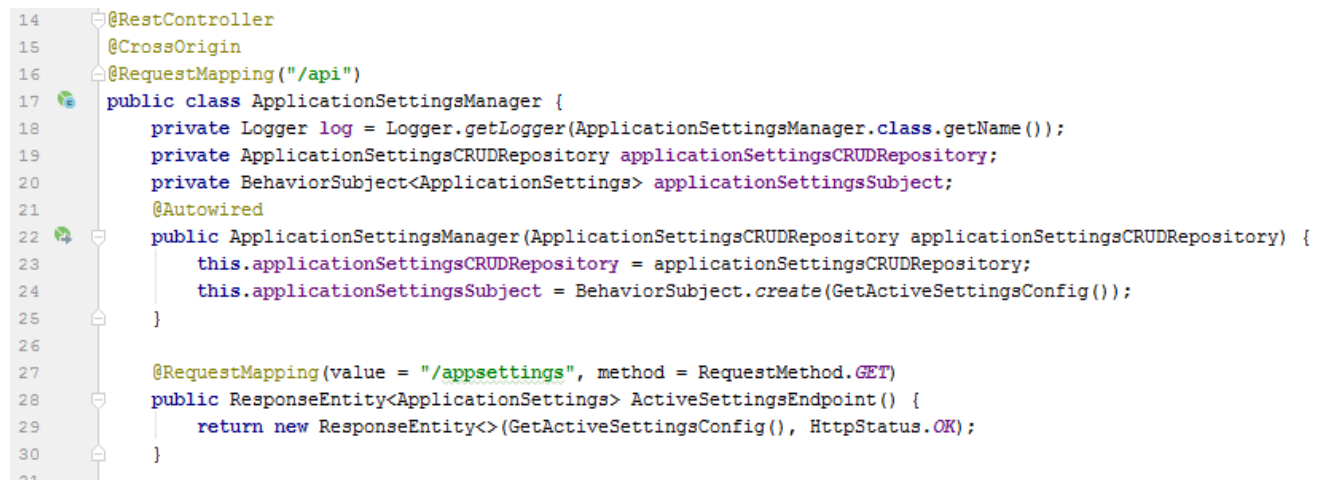
В клиентской части программы для получения данных через WebSocket соединение по протоколу STOMP используются библиотеки stompjs версии 2.3.3 и ng2-stomp-service версии 1.2.2.

3.5 Веб-сервисы

Одной из задач серверной части программы является предоставление веб-сервисов для клиентской части программы. Взаимодействие пользователя с программой через пользовательский интерфейс в клиентской части обеспечивается при помощи вызова соответствующих конечных точек веб-сервисов серверной части программы. Конечные точки веб-сервисов, предоставляемого серверной частью программы, представляют собой URL адреса, по которым направляются соответствующие HTTP запросы.

В серверной части программы за конечные точки веб-сервиса отвечают классы, помеченные аннотацией @RestController. Такие классы являются управляемыми компонентами

Spring приложения, аналогично классам, помеченным аннотацией `@Controller`. Каждая конечная точка представлена методом с аннотацией `@RequestMapping`. Параметры аннотации `@RequestMapping` указывают на относительный путь данной конечной точки, а также метод HTTP, по которому можно будет осуществить вызов данной конечной точки. Для примера на рисунке 20 представлен фрагмент исходного кода класса `ApplicationSettingsManager`



```

14  @RestController
15  @CrossOrigin
16  @RequestMapping("/api")
17  public class ApplicationSettingsManager {
18      private Logger log = Logger.getLogger(ApplicationSettingsManager.class.getName());
19      private ApplicationSettingsCRUDRepository applicationSettingsCRUDRepository;
20      private BehaviorSubject<ApplicationSettings> applicationSettingsSubject;
21      @Autowired
22      public ApplicationSettingsManager(ApplicationSettingsCRUDRepository applicationSettingsCRUDRepository) {
23          this.applicationSettingsCRUDRepository = applicationSettingsCRUDRepository;
24          this.applicationSettingsSubject = BehaviorSubject.create(GetActiveSettingsConfig());
25      }
26
27      @RequestMapping(value = "/appsettings", method = RequestMethod.GET)
28      public ResponseEntity<ApplicationSettings> ActiveSettingsEndpoint() {
29          return new ResponseEntity<>(GetActiveSettingsConfig(), HttpStatus.OK);
30      }
31  }

```

Рисунок 21 – фрагмент исходного кода класса `ApplicationSettingsManager`

Данные, возвращаемые веб-сервисами серверной части программы, представляются в виде текста в формате JSON.

Некоторые веб-сервисы серверной части для взаимодействия с внутренними картотеками программы построены по принципам HATEOAS. Данный принцип подразумевает, что данные о запрашиваемом ресурсе будут представлены в виде гиперссылок. То есть, если запрашиваемый объект представляет собой сложную структуру, составленную из других объектов, то в качестве значений данных объектов будут переданы ссылки на соответствующие ресурсы. На рисунке 22 представлен пример ответа на HTTP GET запрос к такому веб-сервису.



```

{
  "id" : 4,
  "pincode" : "1234",
  "firstName" : "Александр",
  "lastName" : "Мендеев",
  "telephone" : null,
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/crud/users/4"
    },
    "userInfo" : {
      "href" : "http://localhost:8080/api/crud/users/4"
    },
    "roles" : {
      "href" : "http://localhost:8080/api/crud/users/4/roles"
    }
  }
}

```

Рисунок 22 – пример ответа на запрос к веб-сервису, построенному по принципу HATEOAS

Как можно видеть на рисунке 22, объект, возвращаемый в результате запроса к HATEOAS веб-сервису по-прежнему представляет собой текст в формате JSON, однако, значения, представляющие собой некий объект или коллекцию (в данном случае это атрибут `roles`, представляющий собой список объектов ролей пользователя) представлены в виде ссылок.

Соответствующие операции CRUD над ресурсом осуществляются в виде вызова разных методов HTTP. Вызов метода GET вернет информацию о запрашиваемом ресурсе, вызов метода POST позволяет создать ресурс, вызов метода DELETE – удалить ресурс, PUT – обновить ресурс и т.д.

С точки зрения программирования реализация HATEOAS веб-сервиса осуществляется при помощи возможностей фреймворка Spring. Для каждого ресурса веб сервиса создается интерфейс, наследующий обобщенный интерфейс `Repository<T, ID>`, или один из его интерфейсов-потомков (`CrudRepository`, `PagingAndSortingRepository`). Данный интерфейс помечается аннотацией `@RepositoryRestResource`, в параметрах которой указывается относительный путь к соответствующему ресурсу. На рис. 23 представлен пример в виде интерфейса `ParameterInfoCRUDRepository`.

```
1 package com.alexander.controller.repository.parameter;
2
3 import com.alexander.model.ParameterInfo;
4 import org.springframework.data.repository.CrudRepository;
5 import org.springframework.data.rest.core.annotation.RepositoryRestResource;
6
7 @RepositoryRestResource(collectionResourceRel = "parameters", path = "parameters")
8 public interface ParameterInfoCRUDRepository extends CrudRepository<ParameterInfo, Integer> {
9 }
```

Рисунок 23 – исходный код интерфейса `ParameterInfoCRUDRepository`

При этом, важно отметить, что в данном случае от программиста не требуется реализации соответствующего интерфейса. Объекты типа данного интерфейса могут быть получены через механизм внедрения зависимостей, который уже обсуждался ранее. Во время исполнения соответствующая реализация данного интерфейса будет автоматически предоставлена фреймворком Spring через встроенные механизмы рефлексии языка Java. В данном случае реализуется шаблон проектирования «Заместитель» (Proxy).

В клиентской части программы отправка соответствующих HTTP запросов на конечные точки веб-сервисов осуществляется при помощи методов класса `HttpClient`, предоставляемого фреймворком Angular.

3.5.1 Аутентификация и авторизация доступа к ресурсам

Одним из требований при создании данной программы было обеспечение авторизованного доступа к различным функциям программы. Аутентификация в программе позволяет установить личность пользователя на основании введенных им данных. В данной программе аутентификация осуществляется путём ввода пользователем ПИН-кода – цифробуквенного сочетания символов, уникального для каждого пользователя.

Аутентификация в программе реализована по стандарту JWT (JSON Web Token). Данный стандарт используется для передачи данных авторизации между клиентом и сервером. Идея заключается в том, что клиентская часть программы отправляет запрос на аутентификацию, указывая необходимые данные. Серверная часть программы, в свою очередь, проверяет правильность введенных данных, и в случае, если введенные данные корректны, генерирует специальный токен доступа. Клиентская часть программы принимает токен, сгенерированный сервером в ответ на запрос аутентификации и сохраняет его, используя локальное хранилище браузера. В дальнейшем, клиентская часть программы, при отсылке запросов к веб-сервисам серверной части программы, добавляет значение сохранённого токена в заголовок HTTP запроса. Таким образом, серверная часть программы, получив запрос от клиентской части программы, содержащий токен авторизации в заголовке запроса, может проверить валидность переданного токена и определить пользователя, которому соответствует данный токен, а также перечень его прав (ролей). На рис. 24 представлено схематичное изображение процесса взаимодействия клиентской и серверной частей программы при использовании стандарта JWT.

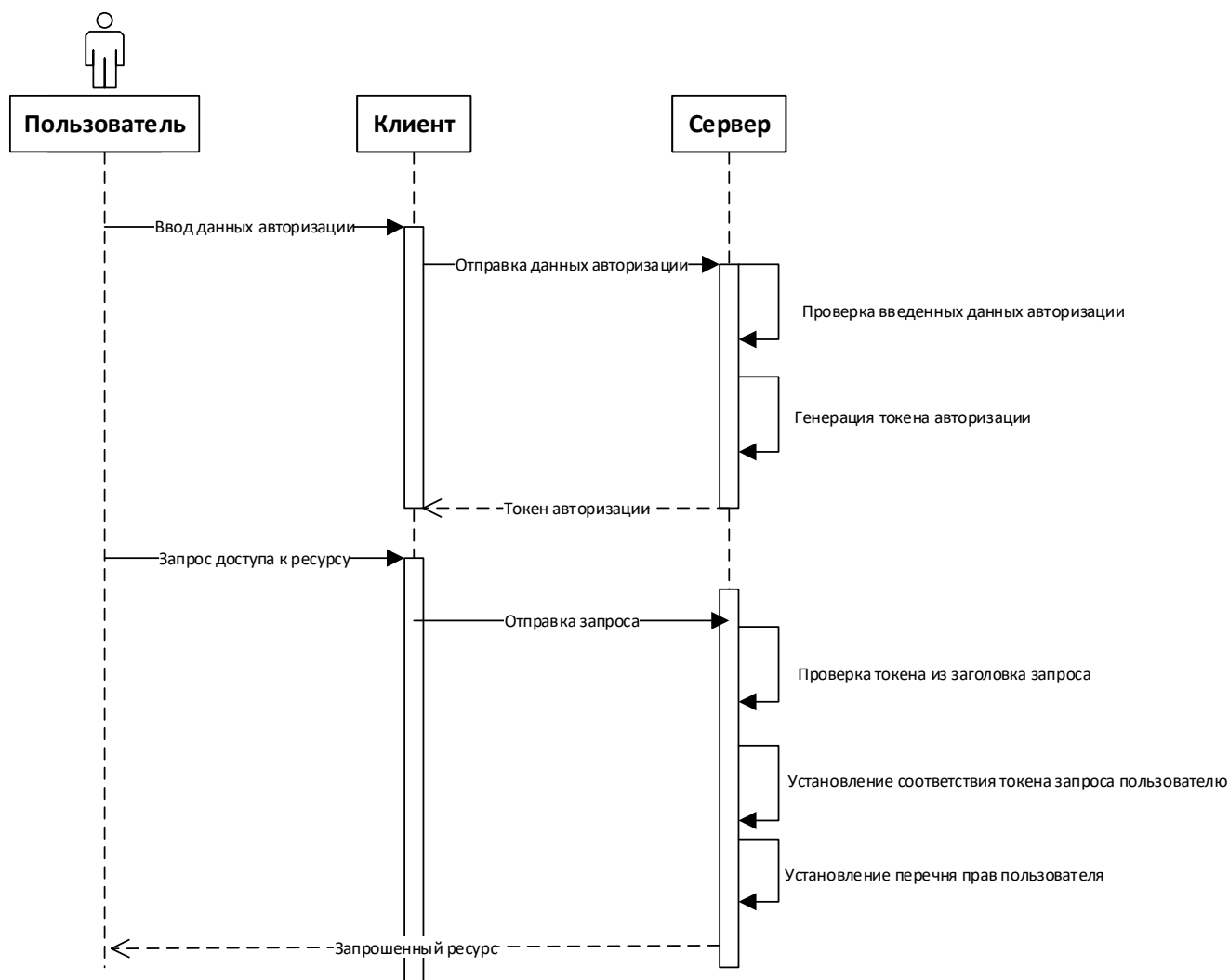


Рисунок 24 – схематичное изображение взаимодействия клиента и сервера при использовании стандарта JWT

В серверной части программы аутентификация и авторизация реализованы с помощью средств фреймворка Spring Security. Фильтрация всех поступающих запросов происходит в классе `JwtAuthenticationTokenFilter`, который наследует класс `OncePerRequestFilter`, предоставляемый Spring. Данный механизм фильтров в Spring основывается на одной из спецификаций Java EE – спецификации фильтров сервлетов. Фильтры сервлетов служат для фильтрации поступающих HTTP запросов. Использование класса `OncePerRequestFilter` гарантирует, что запрос будет обработан данным фильтром только один раз. Класс-фильтр `JwtAuthenticationTokenFilter` проверяет наличие заголовка запроса «Authorization», а также его значение. Класс `JwtAuthenticationTokenFilter` проверяет токен доступа и устанавливает личность пользователя, а также набор его прав. Генерация токена авторизации происходит в методе `generateToken` класса `JwtTokenUtil`. Для работы с JWT токенами используется библиотека `jjwt` версии 0.7.0.

Конфигурация доступа к ресурсам приводится в классе `ResourceServerConfig`. Данный класс наследует класс `WebSecurityConfigurerAdapter`, предоставляемый фреймворком Spring Security. Сама конфигурация производится в унаследованном методе `configure`. Параметром данного метода является объект класса `HttpSecurity`. При помощи вызова комбинации различных методов данного класса устанавливается свод правил безопасности для доступа к ресурсам приложения. На рис. 25 представлен фрагмент исходного метода `configure` класса `ResourceServerConfig`.

```
39  @Override
40  protected void configure(HttpSecurity http) throws Exception {
41      http.cors();
42      http.csrf().disable()
43          .exceptionHandling().authenticationEntryPoint(unauthorizedHandler)
44          .and()
45          .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS)
46          .and()
47          .authorizeRequests()
48          .antMatchers(...antPatterns: "/api/crud/roles/**").hasRole("ADMIN")
49          .antMatchers(...antPatterns: "/api/crud/users/**").hasRole("ADMIN")
50          .antMatchers(HttpMethod.POST, ...antPatterns: "/api/appsettings").hasRole("ADMIN")
51          .antMatchers(HttpMethod.POST, ...antPatterns: "/api/crud/parameters/**").hasRole("ADMIN")
52          .antMatchers(HttpMethod.PUT, ...antPatterns: "/api/crud/parameters/**").hasRole("ADMIN")
53          .antMatchers(HttpMethod.PATCH, ...antPatterns: "/api/crud/parameters/**").hasRole("ADMIN")
54          .antMatchers(HttpMethod.DELETE, ...antPatterns: "/api/crud/parameters/**").hasRole("ADMIN")
55          .antMatchers(HttpMethod.POST, ...antPatterns: "/api/crud/patient_settings/**").hasAnyRole(...roles: "ADMIN", "DOCTOR")
56          .antMatchers(HttpMethod.PUT, ...antPatterns: "/api/crud/patient_settings/**").hasAnyRole(...roles: "ADMIN", "DOCTOR")
57          .antMatchers(HttpMethod.PATCH, ...antPatterns: "/api/crud/patient_settings/**").hasAnyRole(...roles: "ADMIN", "DOCTOR")
58          .antMatchers(HttpMethod.DELETE, ...antPatterns: "/api/crud/patient_settings/**").hasAnyRole(...roles: "ADMIN", "DOCTOR")
59          .antMatchers(...antPatterns: "/api/**").authenticated()
60          .and()
61          .logout();
62      http.addFilterBefore(authenticationTokenFilterBean(), UsernamePasswordAuthenticationFilter.class);
63      http.headers().cacheControl();
64  }
65 }
```

Рисунок 25 – конфигурация безопасности ресурсов серверной части программы

В клиентской части программы полученный в результате запроса авторизации токен сохраняется в локальном хранилище веб-браузера. Добавление токена в заголовки исходящих запросов происходит при помощи использования интерфейса `HttpInterceptor`, предоставляемого фреймворком Angular. Классы, реализующие интерфейс `HttpInterceptor` выступают в роли своеобразных фильтров HTTP запросов. Класс `TokenInterceptorService` снабжает каждый запрос заголовком авторизации, содержащим значение токена авторизации, полученного ранее. На рис. 26 представлен исходный код класса `TokenInterceptorService`.

```
6 @Injectable()
7 export class TokenInterceptorService implements HttpInterceptor {
8   constructor(private injector: Injector) {
9   }
10  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
11    if(req.url.includes( searchString: 'auth')) {
12      return next.handle(req);
13    }
14    let authServ: AuthService = this.injector.get(AuthService);
15    const ch_req = req.clone( update: {
16      setHeaders: {
17        Authorization: `Bearer ${authServ.getBearerToken()}`
18      }
19    });
20    return next.handle(ch_req);
21  }
22 }
23
```

Рисунок 26 – исходный код класса TokenInterceptorService

Использование `HttpInterceptor` избавляет от необходимости каждый раз сопровождать токеном доступа запросы на доступ к ресурсам серверной части программы.

3.6 Выводы по главе

В данной главе были рассмотрены технические детали реализации некоторых частей программы. Были приведены использованные инструменты разработки, технологии реализации. Использование фреймворка Spring позволило с наименьшими усилиями создать многоплатформенное серверное приложение на платформе Java, а использование фреймворка Angular позволило создать клиентскую часть программы в виде веб-приложения, доступ к которому может быть осуществлен с любого современного веб-браузера с поддержкой технологии JavaScript. Была рассмотрена структура внутренней серверной базы данных, описана работа с данными пациентов в терминах программирования. В конце главы также было приведено описание реализации веб-сервисов, предоставляемых серверной частью программы и механизмов для их защиты.

Апробация результатов работы

Результат данной работы в виде демонстрации одного из первых прототипов программы был представлен на XX Конгрессе педиатров России с международным участием «Актуальные проблемы педиатрии» в рамках Российско-Австрийского симпозиума «Цифровая медицина и поддержка клинических решений». Тема доклада: «Цифровые технологии многопараметрического мониторинга в отделении анестезиологии-реанимации». 18 февраля 2018 года, г. Москва [18].

В мае 2018 года данное программное решение было внедрено в отделение анестезиологии-реанимации НИИ НДХиТ.

Заключение

Данная работа посвящена одной из самых значимых областей в области нейрохирургии – мониторингу ауторегуляции мозгового кровообращения. Целью данной работы явилась разработка многоплатформенного программного решения, позволяющего непрерывно отслеживать в реальном времени состояние ауторегуляции мозгового кровообращения у пациентов.

В рамках данной работы было разработано клиент-серверное решение, позволяющее производить непрерывную качественную оценку процесса ауторегуляции головного мозга пациентов при помощи вычисления индексов ауторегуляции мозга. Разработанное решение использует технологии Java и JavaScript. Использование языка Java в серверной части программы позволило расширить круг целевых программных платформ данного решения, а использование технологий JavaScript в клиентской части программы сделало возможным доступ к данной программе с носимых устройств, таких как, например, планшетные компьютеры. В ходе данной работы:

- 1) Были изучены принципы вычисления и особенности индексов ауторегуляции мозга PRx и VRx;
- 2) Был рассмотрен единственный существующий аналог данной программы – программное обеспечение ICM+. Были выявлены недостатки существующего программного обеспечения;
- 3) Была предложена и реализована методология оценки ауторегуляции мозгового кровообращения, основанная на вычислении индексов ауторегуляции мозга, их последующей классификации и подсчёте относительных частот встречаемости классов значений в пределах временного буфера;
- 4) На основании анализа существующего решения и его недостатков были выявлены ключевые функциональные и нефункциональные требования к разрабатываемой программе. Так, например, было принято решение о разработке многоплатформенного программного решения;
- 5) С учётом ключевых требований были выбраны технологии Java и JavaScript для серверной и клиентской частей программы соответственно;
- 6) Были выявлены требования к входным данным. В качестве источника входных данных было принято решение использовать записи клинической базы данных НИИ НДХиТ, содержащие данные пациентов по разным жизненно важным показателям. Выходными данными программы явились относительные частоты встречаемости классов значений индексов ауторегуляции PRx и VRx за прошедший период, представленные в виде круговых диаграмм и линейных графиков;

- 7) Было принято решение об использовании клиент-серверной архитектуры, поскольку данная архитектура позволяет сэкономить вычислительные ресурсы клиентских устройств, перенеся всю вычислительную работу на серверную часть программы. Это особенно важно, если учесть, что данное программное обеспечение может быть использовано врачами на планшетных компьютерах;
- 8) Разработана структура внутренней базы данных для хранения настроек программы и для аккумуляции данных пациентов, которые затем могут быть использованы для выгрузки и составления отчётов;
- 9) С помощью фреймворка Spring была разработана серверная часть программы;
- 10) С помощью фреймворка Angular была разработана клиентская часть программы в виде веб-приложения;
- 11) Была разработана техническая документация для программной реализации работы в соответствии с ГОСТ.

Ожидается, что данная программа предоставит врачам возможность непрерывного наблюдения за состоянием ауторегуляции мозгового кровообращения у пациентов с черепно-мозговой травмой. Информация, полученная в результате использования программы, может быть использована врачами для принятия соответствующих клинических решений и назначения подходящего лечения.

Основное направление для дальнейшей работы – это интеграция данного программного обеспечения с системой поддержки принятия клинических решений Arden Suite от компании Medexter. Планируется формализовать и реализовать в программном виде протокол лечения пациентов с черепно-мозговой травмой, ввести систему оповещений врачей с рекомендациями по принятию разного рода клинических решений. Данное направление работы требует более тесного взаимодействия с врачами для определения ключевых требований и задач.

Список использованных источников

1. Лихтерман Л.Б. Черепно-мозговая травма. Диагностика и лечение. ГЭОТАР-Медиа, 2014.
2. Ошоров А.В. Ауторегуляция, перфузия и ВЧД при поврежденном мозге. Москва: 2017. Презентация.
3. Bouma G.J., Muizelaar J.P., Bandoh K., Marmarou A. Blood pressure and intracranial pressure-volume dynamics in severe head injury: relationship with cerebral blood flow. // Journal of Neurosurgery, Vol. 77, No. 1, 1992. pp. 15-19.
4. Hlatky R., Valadka A.B., Robertson C.S. Intracranial pressure response to induced hypertension: role of dynamic pressure autoregulation // Journal of Neurosurgery, Vol. 57, No. 5, 2005. pp. 917-923.
5. Rangel-Castilla L., Gasco J., Nauta H.J., Okonkwo D.O., Robertson C.S. Cerebral pressure autoregulation in traumatic brain injury // Neurosurgical Focus, Vol. 25, No. 4, Oct 2008. P. E7.
6. Семенютин В.Б., Свистов Д.В. Методы оценки регуляции мозгового кровотока в нейрохирургии // Российская нейрохирургия. 2005. URL: http://www.neuro.neva.ru/ru/Articles_2005_1/semenyutin.shtml (дата обращения: 05.04.2018).
7. Ганеева Ж.Г. Определение понятия "мониторинг" в различных сферах его применения // Вестник Челябинского государственного университета, Т. 8, № 1, 2005. С. 30-33.
8. Definition of monitor in English by Oxford Dictionaries [Электронный ресурс] URL: <https://en.oxforddictionaries.com/definition/monitor> (дата обращения: 05.05.2018).
9. Czosnyka M., Smielewski P., Kirkpatrick P., Laing R.J., Menon D., Pickard J.D. Continuous assessment of the cerebral vasomotor reactivity in head injury // Journal of Neurosurgery, Vol. 41, No. 1, July 1997. pp. 11-17.
10. Zweifel C., Lavinio A., Steiner L.A., Radolovich D.K., Smielewski P., Timofeev I., Hiler M., Balestreri M., Kirkpatrick P.J., Pickard J.D., Hutchinson P.J., Czosnyka M. Continuous monitoring of cerebrovascular pressure reactivity in patients with head injury // Neurosurgical Focus, Vol. 25, No. 4, Oct 2008. P. E2.
11. Ducharme-Crevier L. Cerebrovascular Pressure Reactivity in Children with TBI // Pediatric Neurology Briefs, Vol. 29, No. 10, Oct 2015. P. 77.

12. Balestreri M., Czosnyka M., Steiner L.A., Hiler M., Schmidt E.A., Matta B.F., Menon D.K., Hutchinson P.J., Pickard J.D. Association between outcome, cerebral pressure reactivity and slow ICP waves following head injury // *Acta neurochirurgica*, Vol. 95, 2005. pp. 25-28.
13. Москалева Д.А. Исследование прогностической значимости показателей изменчивости внутричерепного и артериального давления у детей с тяжелой черепно-мозговой травмой, Кафедра медицинской кибернетики и информатики, Российский Национальный Исследовательский Медицинский Университет им. Н.И. Пирогова, Москва, Бакалаврская работа 2012.
14. Cambridge Enterprise ICM+ - About [Электронный ресурс] // University of Cambridge: [сайт]. URL: <https://icmplus.neurosurg.cam.ac.uk/home/about/> (дата обращения: 05.04.2018).
15. Global Tablet Shipments by Operating System 2010-2017 [Электронный ресурс] // Statista - The Statistics Portal: [сайт]. URL: <https://www.statista.com/statistics/273268/worldwide-tablet-sales-by-operating-system-since-2nd-quarter-2010/> (дата обращения: 06.04.2018).
16. ICM+ software for brain monitoring in neurological intensive care research [Электронный ресурс] // Cambridge Enterprise: [сайт]. URL: <https://www.enterprise.cam.ac.uk/opportunities/icm-software-for-brain-monitoring-in-neurological-intensive-care-research/> (дата обращения: 06.04.2018).
17. The Solution for Decoupling: Dependency Injection | Inversion of Control with the Managed Extensibility Framework (MEF) [Электронный ресурс] // InformIT: [сайт]. URL: <http://www.informit.com/articles/article.aspx?p=1635818&seqNum=2> (дата обращения: 04.14.2018).
18. XX КОНГРЕСС ПЕДИАТРОВ РОССИИ с международным участием «АКТУАЛЬНЫЕ ПРОБЛЕМЫ ПЕДИАТРИИ» [Электронный ресурс] // Российская академия педиатрии: [сайт]. [2018]. URL: <http://www.academypediatrics.ru/xx-kongress-2018> (дата обращения: 10.04.2018).

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"**

Факультет компьютерных наук, департамент программной инженерии

СОГЛАСОВАНО

Кандидат технических наук, руководитель
отдела новых технологий НИИ неотложной
детской хирургии и травматологии

_____ С.Б. Арсеньев
" __ " _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель ОП
"Программная инженерия" Национального
исследовательского университета "Высшая
школа экономики"

_____ В.В. Шилов
" __ " _____ 2018 г.

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.15-01 ТЗ 01-1

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Исполнитель

студент группы БПИ143

_____ /А.П. Мендеев /

« ____ » _____ 2018 г.

УТВЕРЖДЕН

RU.17701729.04.15-01 ТЗ 01-1-ЛУ

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Техническое задание

RU.17701729.04.15-01 ТЗ 01-1

Листов 21

<i>Инв. № подл</i>	
<i>Подп. и дата</i>	
<i>Взам. инв. №</i>	
<i>Инв. № дубл.</i>	
<i>Подп. и дата</i>	

СОДЕРЖАНИЕ

1. Введение	63
1.1. Наименование	63
1.2. Краткая характеристика области применения	63
2. Основания для разработки	64
2.1. Документ(ы), на основании которого(ых) ведется разработка	64
3. Назначение разработки.....	65
3.1. Функциональное назначение	65
3.2. Эксплуатационное назначение	65
4. Требования к программе	66
4.1. Требования к функциональной составляющей программы	66
4.1.1. Требования к составу выполняемых функций	66
4.1.2. Требования к входным данным.....	67
4.1.3. Требования к выходным данным.....	67
4.2. Требования к пользовательскому интерфейсу	69
4.3. Требования к надежности	70
4.4. Условия эксплуатации	70
4.4.1. Требования к квалификации пользователя	70
4.4.2. Требования к видам обслуживания	70
4.4.3. Требования к климатическим условиям.....	70
4.5. Требования к составу и параметрам технических средств.....	70
4.6. Требования к информационной и программной совместимости	71
4.6.1. Требования к программному обеспечению	71
4.6.2. Требования к исходным кодам и языкам программирования.....	71
4.6.3. Требования к защите информации	71
5. Требования к программной документации	72
5.1. Специальные требования к документации.....	72
5.2. Состав программной документации	72
6. Техничко-экономические показатели	73
6.1. Ориентировочная экономическая эффективность	73
6.2. Предполагаемая потребность	73

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. Стадии и этапы разработки.....	74
8. Порядок контроля и приемки	76
8.1. Виды испытаний	76
8.2. Общие требования к приемке работы	76
Приложение 1	77
Приложение 2	78

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование

«Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой».

1.2. Краткая характеристика области применения

Данное программное обеспечение позволяет непрерывно оценивать качество ауторегуляции мозгового кровообращения у пациентов с черепно-мозговой травмой на основе вычисления индексов ауторегуляции. Программное решение может быть использовано в отделении интенсивной терапии при лечении черепно-мозговой травмы и назначении соответствующего лечения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**2.1. Документ(ы), на основании которого(ых) ведется разработка**

Приказ Национального исследовательского университета «Высшая школа экономики»
№ 2.3-02/0805-04 от 08.05.2018.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Основным функциональным назначением данной разработки является предоставление врачу актуальной информации о состоянии ауторегуляции мозгового кровообращения у пациентов с черепно-мозговой травмой, позволяющей оценить как актуальное состояние, так и динамику изменений.

3.2. Эксплуатационное назначение

Данная программа предназначена медицинским персоналом в отделении интенсивной терапии. Главная цель эксплуатации данной программы – это непрерывная оценка состояния ауторегуляции головного мозга у пациентов с травмами головы. Программа также может быть использована в исследовательских целях при изучении прогностической значимости использования индексов ауторегуляции мозга в клинической практике.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональной составляющей программы

4.1.1. Требования к составу выполняемых функций

- 1) пользователь должен иметь возможность задать настройки соединения с клинической базой данных НИИ НДХиТ;
- 2) пользователь должен иметь возможность задать настройки соединения с базой данных, которая будет использоваться для хранения данных программы;
- 3) программа должна автоматически создавать необходимые сущности (таблицы) в базе данных, которая используется для хранения данных программы;
- 4) пользователь должен иметь возможность задать значения числовых интервалов для классификации значений индексов ауторегуляции PRx и VRx. Интервалы должны задаваться в виде граничных значений интервалов для всех трёх классов значений;
- 5) пользователь должен иметь возможность задать частоту (временной интервал) для операции запроса данных пациентов из БД НИИ НДХиТ в виде количества секунд;
- 6) пользователь должен иметь возможность проходить аутентификацию посредством ввода своего ПИН-кода;
- 7) пользователь должен иметь возможность выйти из текущей сессии (сбросить аутентификацию);
- 8) пользователь должен иметь доступ к списку активных пациентов;
- 9) пользователь должен иметь возможность поиска по списку активных пациентов;
- 10) пользователь должен иметь возможность перейти в раздел мониторинга для осуществления наблюдения за состоянием ауторегуляции, выбрав пациента из списка активных пациентов;
- 11) информация в разделе мониторинга должна обновляться автоматически без необходимости перезагружать страницу;
- 12) пользователь должен иметь возможность совершить выгрузку данных выбранного пациента за выбранный период дат;
- 13) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять размер временного окна для вычисления индекса PRx (в секундах);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 14) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять размер временного окна для вычисления индекса VRx (в секундах);
- 15) пользователь с ролью «Врач» должен иметь возможность изменять для каждого активного пациента размер временного буфера (в секундах) для вычисления частот встречаемости разных классов значений индекса PRx:
- 16) пользователь с ролью «Врач» должен иметь возможность изменять для каждого активного пациента размер временного буфера (в секундах) для вычисления частот встречаемости разных классов значений индекса VRx:
- 17) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять видимость группы показателей (элементов интерфейса) на странице мониторинга пациента, которые содержат данные об относительной частоте встречаемости разных классов значений индексов PRx/VRx и графики ВЧД/ЦПД;
- 18) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять видимость отдельных параметров (строк) в таблице необработанных значений, находящейся на странице мониторинга пациента;
- 19) пользователь с ролью «Администратор» должен иметь доступ в панель администрирования;
- 20) пользователь с ролью «Администратор» должен иметь возможность управления картотекой пользователей программы;
- 21) пользователь с ролью «Администратор» должен иметь возможность управления картотекой параметров, значения по которым будут запрашиваться программой из БД НИИ НДХиТ.

4.1.2. Требования к входным данным

Входными данными являются значения в таблицах клинической базы данных НИИ НДХиТ. Программа должна производить считывание данных пациентов из клинической базы данных, используя SQL запросы.

4.1.3. Требования к выходным данным

Выходными данными программы являются относительные частоты встречаемости классов значений индексов ауторегуляции PRx и VRx относительно общего количества

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

значений индексов в пределах заданного временного буфера. Программа должна для каждого активного пациента производить расчет и классификацию значений индексов ауторегуляции PRx и VRx. Классификация значений индексов ауторегуляции PRx и VRx должна производиться по заданным пользователем в конфигурационном файле интервалам значений. В процессе классификации значений индексов ауторегуляции PRx и VRx должны выделяться три класса значений:

- а. класс значений, обозначающих стабильную ауторегуляцию мозга;
- б. класс значений, обозначающих дестабилизацию ауторегуляции мозга (промежуточное состояние);
- в. класс значений, обозначающих утрату ауторегуляции мозга.

Программа должна производить расчет относительных частот встречаемости значений, отнесённых к тем или иным классам значений в пределах заданного для данного пациента временного буфера. Данные частоты встречаемости должны быть выражены в виде круговых диаграмм с секторами различных цветов в разделе мониторинга пациента. Доля значений класса «Стабильная ауторегуляция» должна быть представлена в виде сектора зелёного цвета, доля значений класса «Дестабилизация ауторегуляции» (неопределенное состояние) – в виде сектора жёлтого цвета, доля значений класса «Утраченная ауторегуляция» - в виде сектора красного цвета. Также данные круговые диаграммы должны иметь развертку по времени в виде линейных графиков с линиями разных цветов, соответствующих определенному классу значений. Помимо круговых диаграмм и линейных графиков, должна присутствовать информация о приросте или уменьшении доли соответствующего класса значений по сравнению с предыдущими показателями в виде надписи цвета, соответствующего цвету класса.

Помимо относительной частоты встречаемости разных классов значений индексов PRx и VRx, в программе должны быть представлены необработанные значения параметров, получаемые из клинической базы данных НИИ НДХиТ. Необработанные значения представляются в виде числовых значений в таблице.

Все перечисленные выше виды данных должны быть представлены в разделе мониторинга пациента.

При выгрузке данных пациентов, выгруженные данные должны быть представлены в виде файла формата Microsoft Excel, содержащего необходимые данные.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.2. Требования к пользовательскому интерфейсу

- 1) в программе должен присутствовать раздел аутентификации пользователя (вход в программу);
- 2) в программе должен присутствовать раздел со списком активных пациентов;
- 3) в разделе со списком активных пациентов должно присутствовать текстовое поле для ввода поискового запроса;
- 4) в программе должен присутствовать раздел мониторинга выбранного активного пациента;
- 5) в программе должен присутствовать раздел с настройками для выбранного активного пациента;
- 6) в программе должна присутствовать панель администрирования с разделами для управления картотекой параметров и картотекой пользователей;
- 7) в панели администрирования навигация должна осуществляться путем выбора соответствующего раздела в боковом меню. Выбранный раздел должен выделяться другим цветом;
- 8) в программе должен присутствовать элемент интерфейса, позволяющий сбросить текущую сессию пользователя (осуществить выход);
- 9) в программе должен присутствовать элемент интерфейса, содержащий имя текущего пользователя;
- 10) в разделе мониторинга пациента должна присутствовать надпись, содержащая ФИО наблюдаемого пациента;
- 11) в разделе мониторинга пациента должна присутствовать таблица, содержащая необработанные данные, получаемые из клинической базы данных НИИ НДХиТ;
- 12) в разделе мониторинга рядом с круговыми диаграммами частот встречаемости классов значений индексов PRx и VRx также должна находиться развертка данных диаграмм по времени в виде линейных графиков с линиями соответствующих цветов;
- 13) в разделе мониторинга рядом с круговыми диаграммами частот встречаемости классов значений индексов PRx и VRx также должны быть текстовые обозначения, отображающие прирост/уменьшение в процентах долей соответствующих классов значений по сравнению с прошлыми показателями;
- 14) в программе должен быть представлен раздел для выгрузки данных пациентов;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 15) в разделе для выгрузки данных пациентов должен быть элемент интерфейса в виде календаря, позволяющий выбрать диапазон дат для выгрузки;
- 16) в разделе для выгрузки данных должны быть представлены элементы интерфейса для множественного выбора типов данных, которые следует включить в выгрузку.

4.3. Требования к надежности

Требования к надежности не предъявляются.

4.4. Условия эксплуатации

4.4.1. Требования к квалификации пользователя

Пользователь должен обладать базовыми навыками работы с компьютером.

4.4.2. Требования к видам обслуживания

На компьютере, где производится выполнение программы, необходимо обеспечить регулярные проверки обновлений ПО. Также необходимо обеспечить защиту компьютера от разного рода компьютерных вирусов.

4.4.3. Требования к климатическим условиям

- 1) Влажность от 10% до 80%;
- 2) Температура от -10°C до 55°C.

4.5. Требования к составу и параметрам технических средств

Для корректной работы программы требуется следующий минимальный состав технических средств:

- 1) Компьютер, оснащенный 32-разрядным или 64-разрядным процессором с тактовой частотой
1 ГГц и выше или аналогичный процессор;
- 2) 1 ГБ оперативной памяти или больше;
- 3) 128 МБ графической памяти;
- 4) жесткий диск объемом 40 ГБ с 1 ГБ свободного места;
- 5) компьютерная мышь;
- 6) клавиатура.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.6. Требования к информационной и программной совместимости**4.6.1. Требования к программному обеспечению**

- 1) Java Runtime Environment версии 8.0 или выше;
- 2) СУБД MySQL версии 5.7 или выше;
- 3) Веб-браузер с поддержкой технологий CSS, HTML, JavaScript.

4.6.2. Требования к исходным кодам и языкам программирования

Программа должна представлять собой веб-приложение, использующее технологии JavaScript.

4.6.3. Требования к защите информации

Требования к защите информации не предъявляются.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Специальные требования к документации

- 1) Документация к программе должна быть выполнена в соответствии с ГОСТ 19.106-78;
- 2) Документация должна быть сдана в печатном виде, при этом она должна быть подписана руководителем организации, утвердившей документ на разработку, руководителем разработки и исполнителем.

5.2. Состав программной документации

- 1) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Техническое задание;
- 2) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Руководство оператора;
- 3) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Программа и методика испытаний;
- 4) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Текст программы;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Ориентировочная экономическая эффективность

В рамках разработки данной программы расчет экономической эффективности не предусмотрен.

6.2. Предполагаемая потребность

Данная программа может служить вспомогательным инструментом при оценке состояния мозгового кровообращения пациентов с травмой головы, находящихся в отделении интенсивной терапии НИИ НДХиТ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии разработки	Этапы работ	Содержание работ
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи
		Сбор исходных материалов
	Научно-исследовательские работы	Определение структуры входных и выходных данных
		Определение требований к техническим средствам
		Обоснование принципиальной возможности решения поставленной задачи
	Разработка и утверждение технического задания	Определение требований к программе
		Определение стадий, этапов и сроков разработки программы и документации на неё
		Согласование и утверждение технического задания
2. Технический проект	Разработка технического проекта	Разработка алгоритма решения задачи
		Окончательное определение конфигурации технических средств
	Утверждение технического проекта	Разработка плана мероприятий по разработке программы
		Разработка пояснительной записки
3. Рабочий проект	Разработка программы	Программирование и отладка программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77
	Испытания программы	Разработка, согласование и утверждение порядка и методики испытаний.
4. Внедрение	Подготовка и защита программного продукта.	Подготовка программы и программной документации для презентации и защиты.
		Утверждение дня защиты программы
		Передача программы и программной документации в архив НИУ ВШЭ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1. Виды испытаний

Производится проверка функциональных требований и требований к пользовательскому интерфейсу.

8.2. Общие требования к приемке работы

Прием программного продукта происходит при полной работоспособности программы при различных входных данных, при выполнении указанных в пункте 4.1 данного документа функций, при выполнении требований, указанных в пункте 4.2 данного документа и при наличии полной документации к программе, указанной в пункте 5, выполненной в соответствии со специальными требованиями технического задания.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

- 1) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] URL: <http://fmi.asf.ru/library/book/Gost/19201-78.html> (дата обращения 18.04.2018);
- 2) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [Электронный ресурс] URL: http://www.rugost.com/index.php?option=com_content&view=article&id=53:19106-78&catid=19&Itemid=50 (дата обращения 18.04.2018).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗОВАННЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ

Ниже приведен список терминов и сокращений, используемых в данном документе.

ПО – программное обеспечение

PRx – Pressure Reactivity Index

VRx – Variation Reactivity Index

БД – база данных

СУБД – система управления базой данных

НИИ НДХиТ – Научно-исследовательский институт неотложной детской хирургии и травматологии

Активный пациент – пациент, находящийся на данный момент в отделении интенсивной терапии НИИ НДХиТ

Аутентификация – процесс проверки подлинности введенных пользователем данных и установления личности этого пользователя

ПИН-код – цифробуквенное сочетание символов, которое используется для прохождения аутентификации и позволяет однозначно идентифицировать пользователя программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"**

Факультет компьютерных наук, департамент программной инженерии

СОГЛАСОВАНО

Кандидат технических наук, руководитель
отдела новых технологий НИИ неотложной
детской хирургии и травматологии

_____ С.Б. Арсеньев
" __ " _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель ОП
"Программная инженерия" Национального
исследовательского университета "Высшая
школа экономики"

_____ В.В. Шилов
" __ " _____ 2018 г.

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.15-01 34 01-1-ЛУ

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Исполнитель

студент группы БПИ143

_____ /А.П. Мендеев /

« ____ » _____ 2018

УТВЕРЖДЕН

RU.17701729.04.15-01 34 01-1-ЛУ

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Руководство оператора

RU.17701729.04.15-01 34 01-1

Листов 35

<i>Инв. № подл</i>	
<i>Подп. и дата</i>	
<i>Взам. инв. №</i>	
<i>Инв. № дубл.</i>	
<i>Подп. и дата</i>	

СОДЕРЖАНИЕ

1. Назначение программы	84
1.1. Функциональное назначение	84
1.2. Эксплуатационное назначение	84
2. Условия выполнения программы	85
2.1. Минимальный состав аппаратных средств	85
2.2. Минимальный состав программных средств	85
3. Выполнение программы.....	86
3.1. Установка программы	86
3.2. Конфигурирование программы	86
3.3. Запуск программы.....	92
3.4. Запуск клиентской части программы.....	93
3.5. Выполнение клиентской программы	93
3.5.1. Вход в приложение (аутентификация)	93
3.5.2. Просмотр списка активных пациентов.....	94
3.5.3. Просмотр страницы мониторинга пациента	95
3.5.4. Внесение изменений в настройки для пациента.....	96
3.5.5. Выгрузка данных (составление отчётов)	98
3.5.6. Панель администрирования.....	101
3.5.6.1. Управление картотекой пользователей.....	102
3.5.6.1.1. Создание.....	102
3.5.6.1.2. Просмотр или изменение данных существующего пользователя.....	103
3.5.6.1.3. Удаление пользователя	104
3.5.6.2. Управление картотекой параметров.....	105
3.5.6.2.1. Добавление записи параметра.....	105
3.5.6.2.2. Просмотр или изменение записи параметра.....	106
3.5.6.2.3. Удаление записи параметра	107
3.5.6.3. Управление настройками программы	108
3.5.7. Выход из приложения (сброс аутентификации).....	109
3.6. Остановка программы	109
4. Сообщения оператору	110
Приложение 1	112

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Приложение 2	113
--------------------	-----

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Функциональное назначение

Основным функциональным назначением данной разработки является предоставление врачу актуальной информации о состоянии ауторегуляции мозгового кровообращения у пациентов с черепно-мозговой травмой, позволяющей оценить как актуальное состояние, так и динамику изменений.

1.2. Эксплуатационное назначение

Данная программа предназначена медицинским персоналом в отделении интенсивной терапии. Главная цель эксплуатации данной программы – это непрерывная оценка состояния ауторегуляции головного мозга у пациентов с травмами головы. Программа также может быть использована в исследовательских целях при изучении прогностической значимости использования индексов ауторегуляции мозга в клинической практике.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1. Минимальный состав аппаратных средств

Для корректной работы программы требуется следующий минимальный состав технических средств:

- 1) ПК, оснащенный 32-разрядным или 64-разрядным процессором с тактовой частотой 1 ГГц и выше или аналогичный процессор;
- 2) 1 ГБ оперативной памяти или больше;
- 3) 128 МБ графической памяти;
- 4) жесткий диск объемом 40 ГБ с 1 ГБ свободного места;
- 5) компьютерная мышь;
- 6) клавиатура.

2.2. Минимальный состав программных средств

- 1) Java Runtime Environment версии 8.0 или выше;
- 2) Веб-браузер с поддержкой технологий HTML, CSS, JavaScript;
- 3) СУБД MySQL версии 5.7 или выше.

Также для корректной работы серверной части программы требуется соединение с клинической БД НИИ НДХиТ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Установка программы

Программа поставляется в виде архива на флэш-накопителе, содержащего все необходимые файлы. Для начала использования программы необходимо распаковать содержимое данного архива в отдельную папку на компьютере.

3.2. Конфигурирование программы

Перед началом использования программы необходимо произвести конфигурацию сервера. Конфигурация производится в текстовом файле «application.properties». Данный файл поставляется в архиве вместе с программой. Файл должен располагаться в одной папке с программой (файлом DataFilter.jar). Конфигурация задается в виде набора параметров и соответствующих им значений. Каждая настройка конфигурации располагается на отдельной строке. В таблице ниже представлено описание параметров конфигурации.

Таблица 1

Таблица описания параметров конфигурации

Параметр	Описание	Значение по умолчанию
<i>Конфигурация подключения к серверной БД (MySQL)</i>		
spring.sdb.url	Строка подключения JDBC	jdbc:mysql://localhost:3306/dbname
spring.sdb.username	Имя пользователя	
spring.sdb.password	Пароль	
spring.sdb.driver-class-name	Полное имя класса-коннектора	com.mysql.jdbc.Driver
spring.jpa.hibernate.ddl-auto	Параметр, указывающий на способ автоматической проверки и генерации	По умолчанию используется значение «update» (обновить схему БД в соответствии с требованиями серверной части программы).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	структуры серверной БД	
<i>Конфигурация подключения к клинической БД НИИ НДХиТ</i>		
spring.smkm.url	Строка подключения JDBC	jdbc:sqlserver://monitoring-2.ndxt.ru;
spring.smkm.username	Имя пользователя	
spring.smkm.password	Пароль	
spring.smkm.driver-class-name	Полное имя класса-коннектора	com.microsoft.sqlserver.jdbc.SQLServerDriver
<i>Конфигурация параметров аутентификации</i>		
jwt.secret	Секретный ключ для шифрования токена авторизации JWT	
jwt.expiration	Время действия сессии авторизации (в секундах)	604800
<i>Настройки пациентов по умолчанию</i>		
prx.window.default	Размер окна вычисления индекса ауторегуляции PRx по умолчанию (в секундах)	50
vrh.window.default	Размер окна вычисления индекса ауторегуляции	50

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	VRx по умолчанию (в секундах)	
prx.percentage.buffer.default	Величина временного буфера для вычисления частоты встречаемости классов значений PRx (в секундах)	3600
vrх.percentage.buffer.default	Величина временного буфера для вычисления частоты встречаемости классов значений VRx (в секундах)	3600
<i>Настройки классификации значений индексов ауторегуляции PRx</i>		
prx.bad.from	Нижняя граница значений для отнесения к классу значений «Утраченная ауторегуляция». Включается в интервал.	0.2
prx.bad.to	Верхняя граница значений для отнесения к	1

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	классу значений «Утраченная ауторегуляция». Включается в интервал.	
prx.neutral.from	Нижняя граница значений для отнесения к классу значений «Неопределенное (промежуточное) состояние». Не включается в интервал.	0
prx.neutral.to	Верхняя граница значений для отнесения к классу значений «Неопределенное (промежуточное) состояние». Не включается в интервал.	0.2
prx.good.from	Нижняя граница значений для отнесения к классу значений «Стабильная ауторегуляция». Включается в интервал.	-1

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

prx.good.to	Верхняя граница значений для отнесения к классу значений «Стабильная ауторегуляция». Не включается в интервал.	0
<i>Настройки классификации значений индексов ауторегуляции VRx</i>		
vrx.bad.from	Нижняя граница значений для отнесения к классу значений «Утраченная ауторегуляция». Включается в интервал.	0.6
vrx.bad.to	Верхняя граница значений для отнесения к классу значений «Утраченная ауторегуляция». Включается в интервал.	1
vrx.neutral.from	Нижняя граница значений для отнесения к классу значений «Неопределенное (промежуточное)»	0.4

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	состояние». Не включается в интервал.	
vrх.neutral.to	Верхняя граница значений для отнесения к классу значений «Неопределенное (промежуточное) состояние». Не включается в интервал.	0.6
vrх.good.from	Нижняя граница значений для отнесения к классу значений «Стабильная ауторегуляция». Включается в интервал.	-1
vrх.good.to	Верхняя граница значений для отнесения к классу значений «Стабильная ауторегуляция». Не включается в интервал.	0.4
<i>Прочие настройки</i>		
read_speed	Временной интервал запроса	5

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

	данных из БД НИИ НДХиТ (в секундах)	
server.port	Номер порта, на котором будет работать сервер	80

Установка значений для всех параметров, за исключением параметра «server.port», является обязательной для корректной работы программы. При отсутствии значения у параметра «server.port», для сервера используется порт 8080.

3.3. Запуск программы

Запуск программы осуществляется с помощью командной строки. Необходимо запустить командную строку, сменить рабочий каталог на каталог, содержащий распакованные файлы, и запустить Java приложение DataFilter.jar с помощью команды “java -jar DataFilter.jar”. На рисунке 1 представлен пример запуска программы через командную строку.

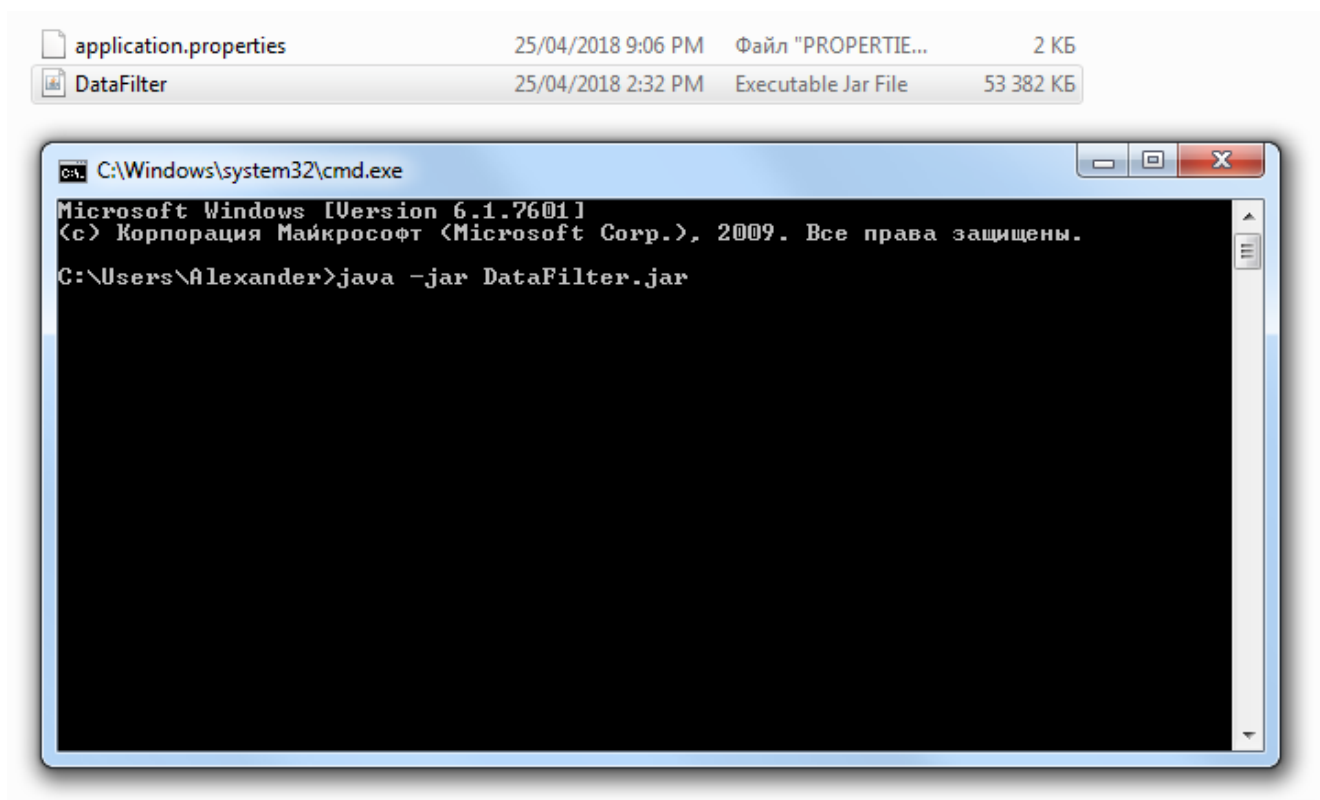


Рисунок 1 – запуск приложения через командную строку Windows

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

В случае, если в локальной базе данных нет пользователя-администратора (при первом запуске), сервером автоматически создается пользователь с правами администратора. ПИН-код сгенерированного пользователя с правами администратора сохраняется в файл password.txt в папку с программой. В дальнейшем, данный ПИН-код можно использовать для входа в клиентскую часть программы.

3.4. Запуск клиентской части программы

Основное взаимодействие пользователя с программой происходит через клиентскую часть программы. Клиентская часть программы запускается в окне веб-браузера посредством ввода в адресную строку веб-браузера адреса, соответствующего IP адресу компьютера, на котором запущена серверная часть программы.

3.5. Выполнение клиентской программы

3.5.1. Вход в приложение (аутентификация)

Вход в программу осуществляется посредством ввода ПИН-кода пользователя в специальное поле в форме входа (рис. 2).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

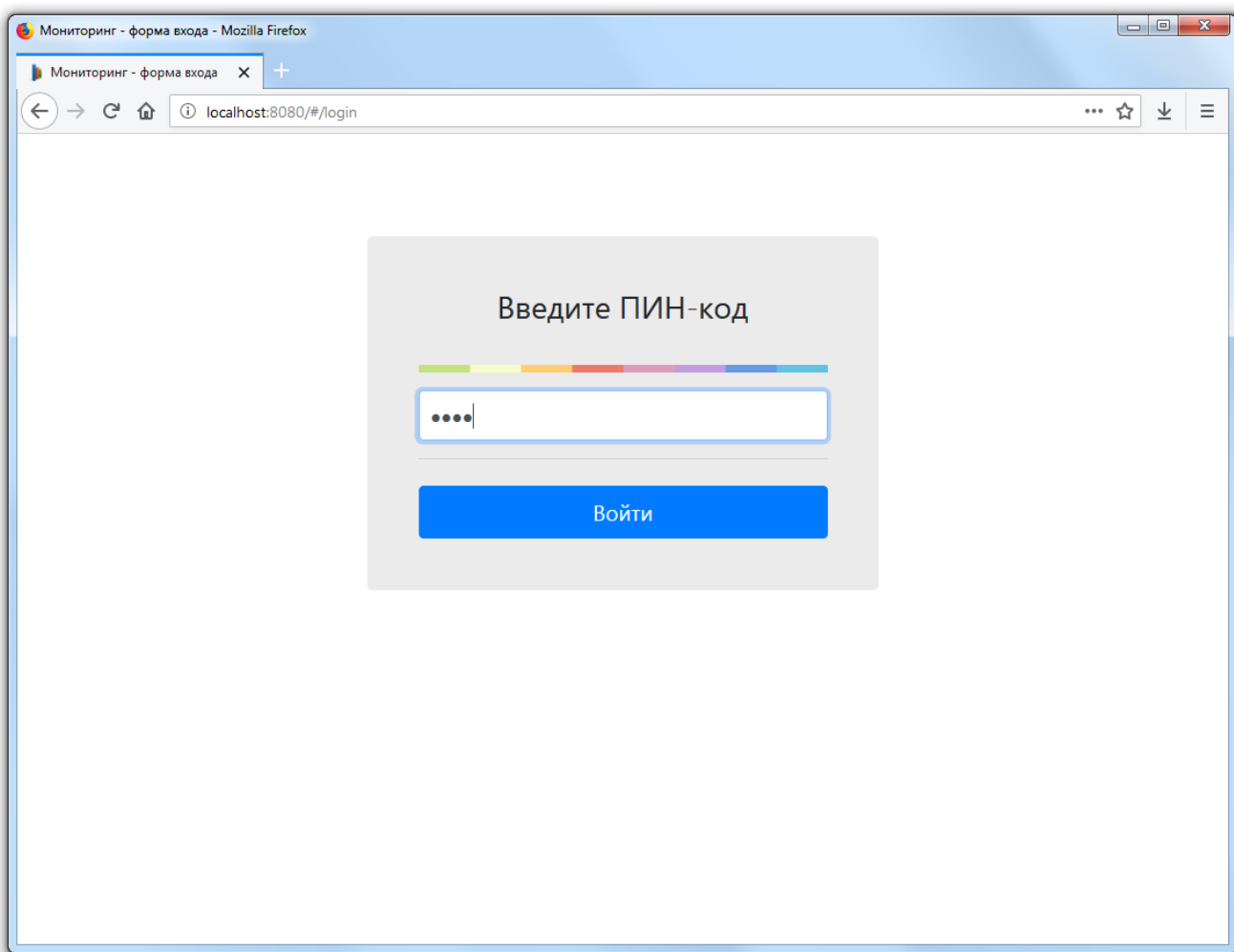


Рисунок 2 – форма входа (аутентификации) в приложение

После ввода ПИН-кода необходимо нажать на кнопку «Войти» или нажать на кнопку ввода на клавиатуре. При первом запуске можно использовать в качестве ПИН-кода сгенерированный ранее ПИН-код администратора.

3.5.2. Просмотр списка активных пациентов

После входа в приложение пользователь попадает на страницу, содержащую список активных пациентов (пациентов, находящихся в отделении интенсивной терапии НИИ НДХиТ).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

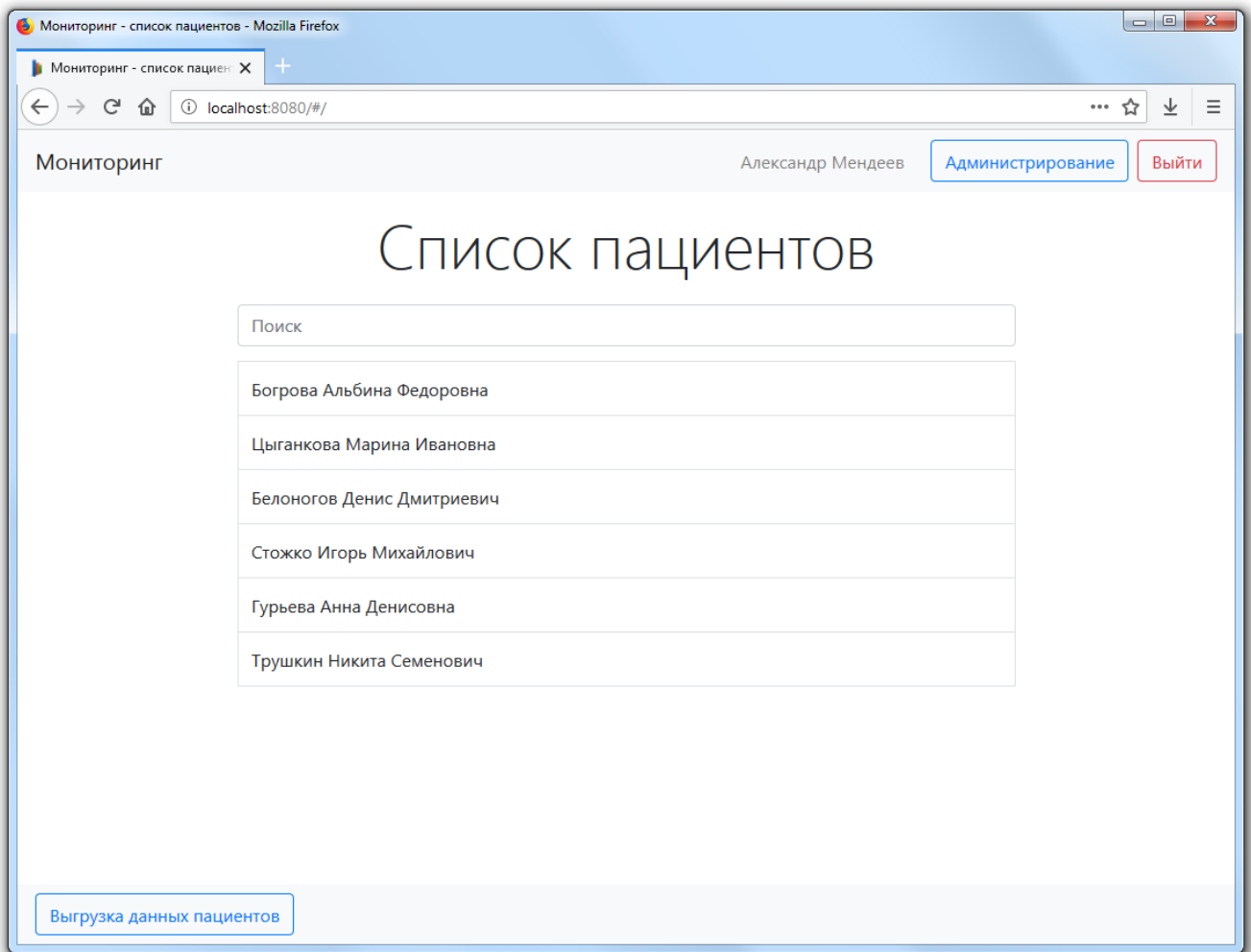


Рисунок 3 – список активных пациентов

На данном экране пользователь может перейти к странице мониторинга пациента, выбрав пациента из списка, или, воспользовавшись поиском. Поиск осуществляется вводом текста в поисковую строку (поле ввода), находящуюся над списком пациентов.

3.5.3. Просмотр страницы мониторинга пациента

Выбрав пациента из списка активных пациентов, пользователь может просмотреть страницу мониторинга пациента (рис. 4).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

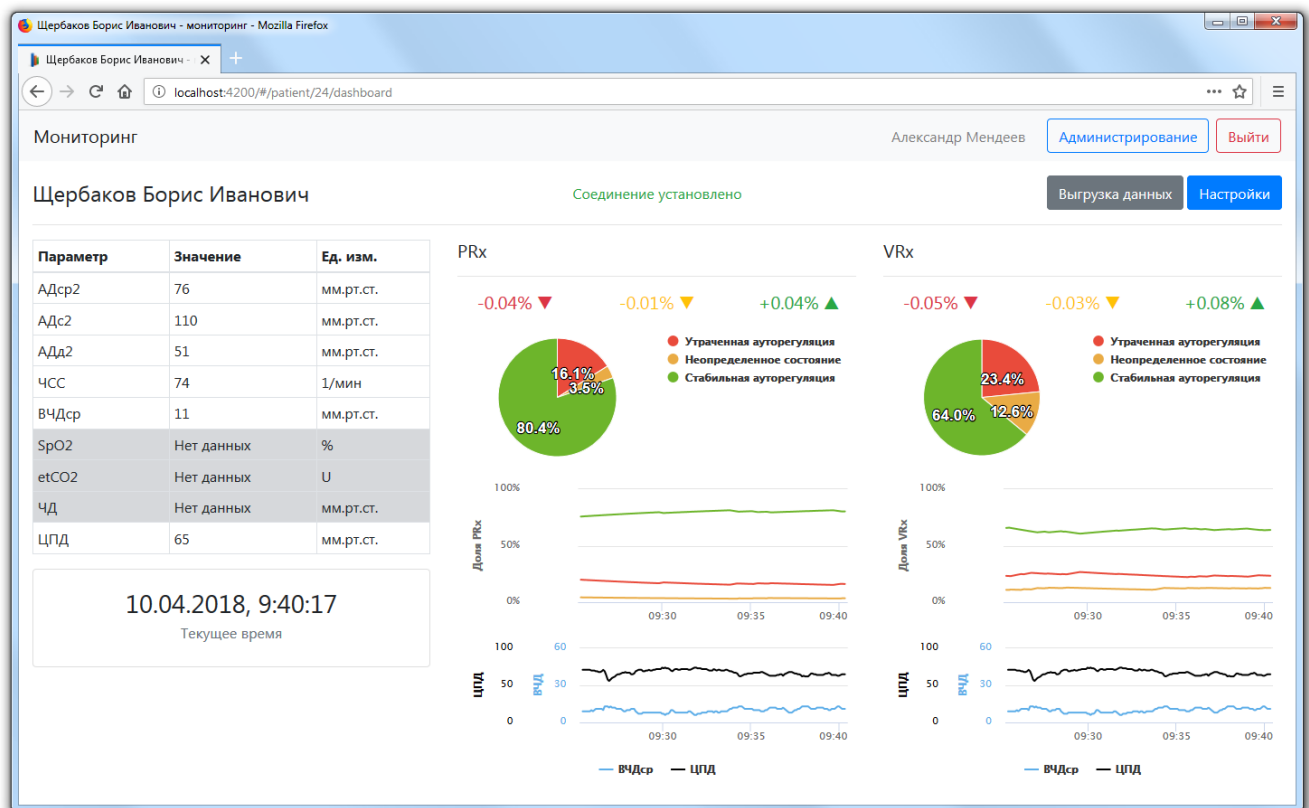


Рисунок 4 – страница мониторинга пациента

Данная страница состоит из нескольких частей: слева отображается таблица необработанных значений параметров, считываемых из клинической БД НИИ НДХиТ, а также текущие дата и время. В правой части страницы располагаются диаграммы и графики частот встречаемости разных классов значений индексов ауторегуляции PRx и VRx за прошедший период. Под графиками частот встречаемости классов значений PRx и VRx располагаются линейные графики значений ЦПД и ВЧД. Над круговыми диаграммами частот встречаемости классов значений индексов PRx и VRx располагается информация о приросте или уменьшении удельной доли данного класса значений по сравнению с предыдущими поступившими данными.

В верхней части страницы располагаются слева направо: ФИО пациента, индикатор WebSocket соединения с серверной частью программы, кнопки «Выгрузка данных» и «Настройки».

3.5.4. Внесение изменений в настройки для пациента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Пользователь с ролью «Врач» или «Администратор» может устанавливать настройки для пациента. Переход на страницу настроек пациента осуществляется при помощи нажатия на кнопку «Настройки», находящуюся на странице мониторинга пациента.

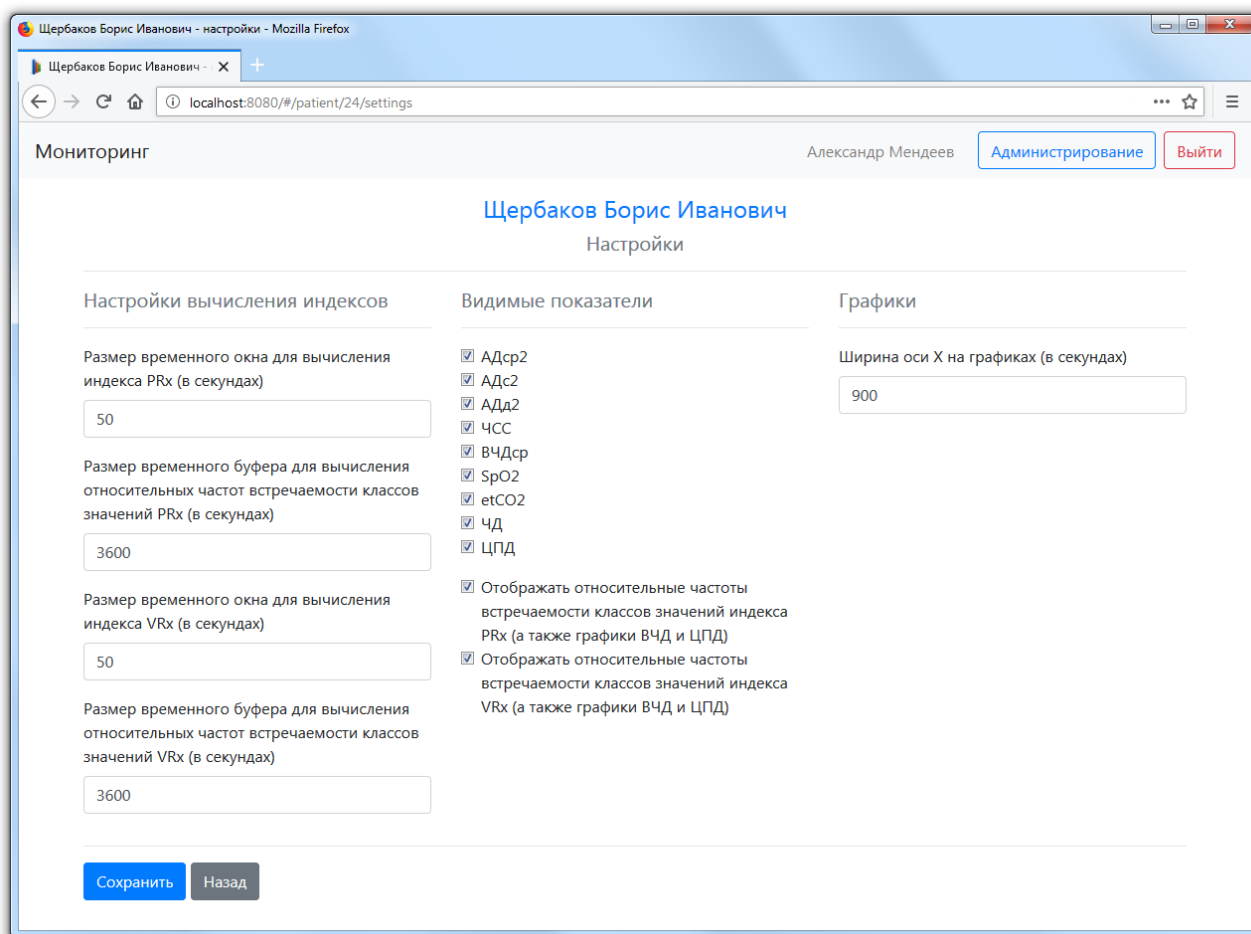


Рисунок 5 – страница настроек пациента

Данную страницу можно условно разделить на три части. Слева находятся настройки вычисления индексов ауторегуляции. Пользователь может задать: размер временного окна (в секундах) для вычисления индексов PRx и VRx, а также размер временного буфера для вычисления частот встречаемости разных классов значений PRx и VRx. Посередине находятся настройки видимости отдельных показателей на странице мониторинга пациента. Пользователь может переключать как видимость отдельных параметров в таблице, содержащей необработанные значения, так и видимость панелей, содержащих круговые диаграммы и графики с частотой встречаемости различных классов значений PRx и VRx. В правой части находится настройка, позволяющая задать ширину оси абсцисс (в секундах) на линейных графиках, находящихся на странице мониторинга пациента. Сохранение настроек

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

осуществляется при помощи нажатия на кнопку «Сохранить» в нижней части страницы.

Нажатие на кнопку «Назад» возвращает пользователя на страницу мониторинга пациента.

3.5.5. Выгрузка данных (составление отчётов)

Пользователь может выгрузить данные по выбранному пациенту, накопленные за время работы программы. Для инициирования выгрузки данных пользователь может нажать на кнопку «Выгрузка данных пациентов», находящуюся на странице со списком активных пациентов слева в нижней части окна (рис. 6).

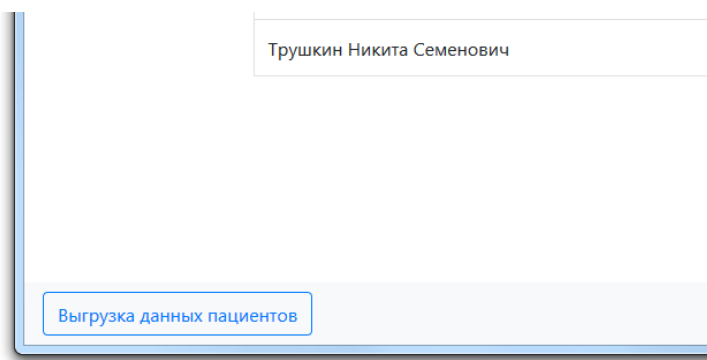


Рисунок 6 – кнопка перехода в режим выгрузки данных пациентов

Нажатие на кнопку выгрузки данных пациентов приводит к переходу на страницу со списком всех пациентов, данные по которым были сохранены программой. В данном списке находятся как активные, так и более неактивные пациенты, по которым были сохранены данные.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

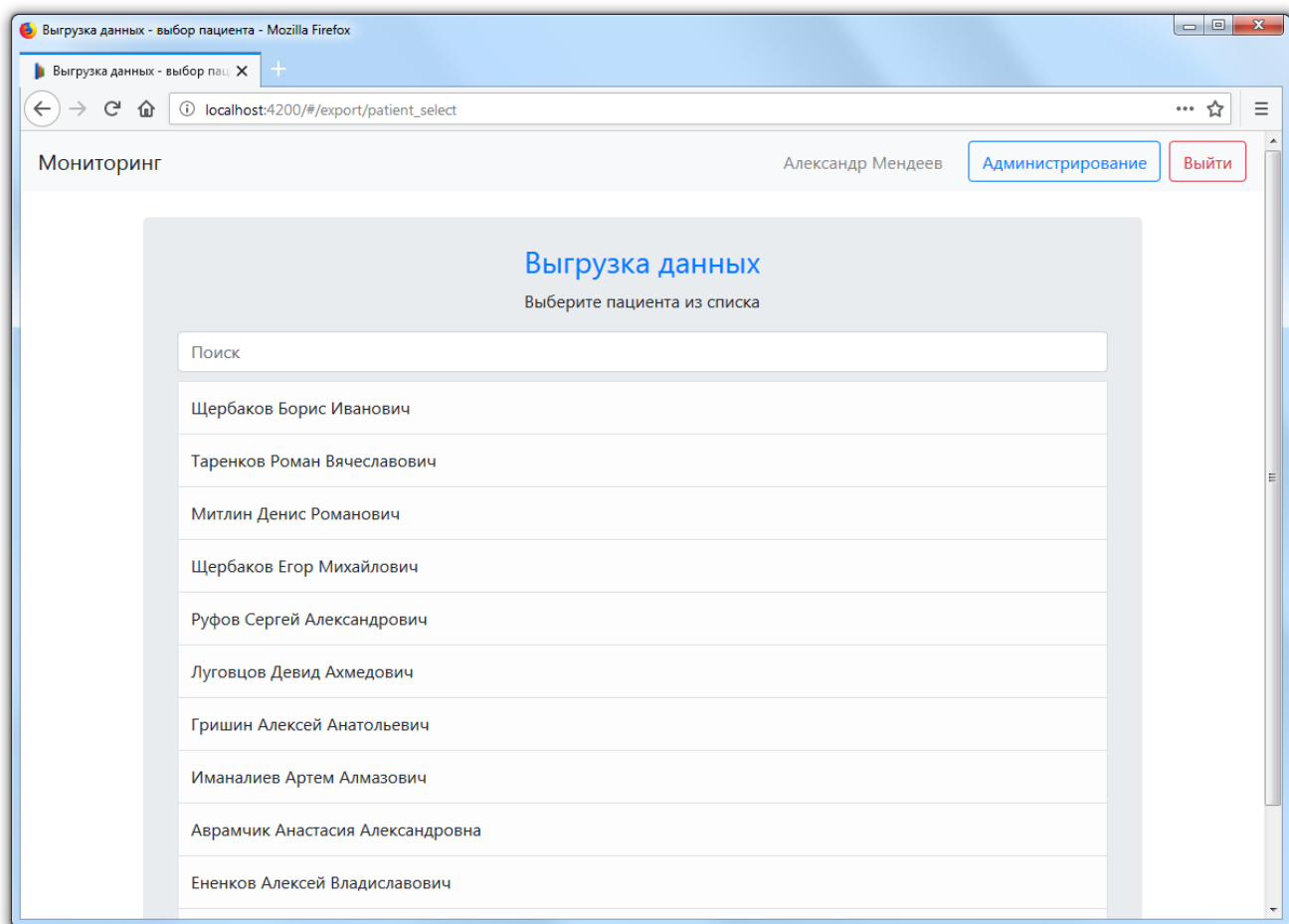


Рисунок 7 – список всех пациентов

Как и при выборе пациента для мониторинга, пользователь может воспользоваться поиском по списку, введя поисковый запрос в текстовое поле, находящееся над списком. Выбор пациента приводит к переходу на страницу с настройкой параметров выгрузки (рис. 8).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

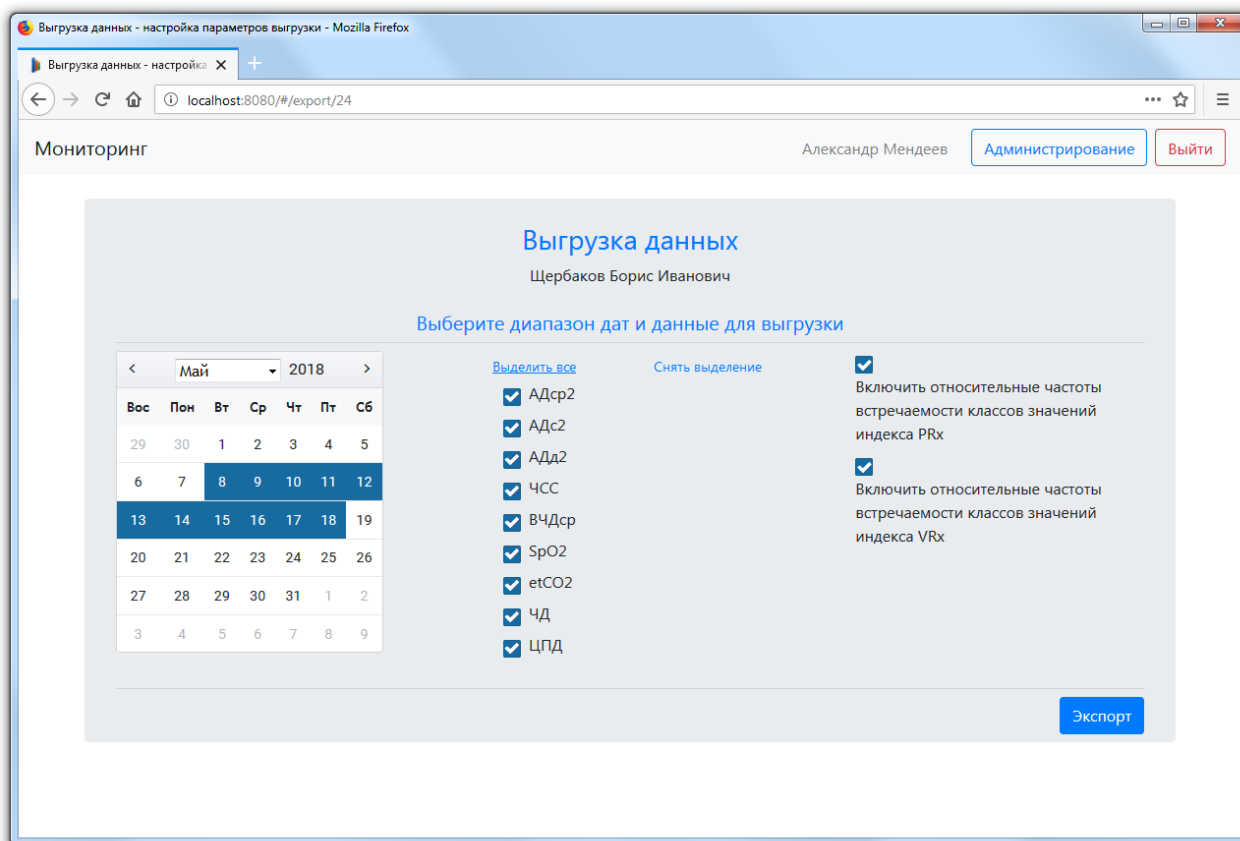


Рисунок 8 – страница настройки параметров выгрузки данных

На данной странице пользователь может выбрать диапазон дат для выгружаемых данных, а также данные для выгрузки. Выбор временного диапазона дат осуществляется двумя нажатиями на даты в календаре в левой части страницы: первое нажатие определяет начальную дату диапазона, а второе – конечную. Выбранный диапазон дат выделяется синим цветом. Диапазон выбирается, включая граничные даты. В центральной части страницы осуществляется выбор параметров, данные по которым будут включены в выгрузку. В правой части страницы можно указать, следует ли включать в выгрузку данных данные о частотах различных классов значений индексов ауторегуляции PRx и VRx, выраженные в удельных долях.

Нажатие на кнопку «Экспорт» приводит к отправке соответствующего запроса на выгрузку в серверную часть программы. После небольшого ожидания, веб-браузер предложит сохранить файл Excel. Данный файл будет содержать выгруженные данные.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

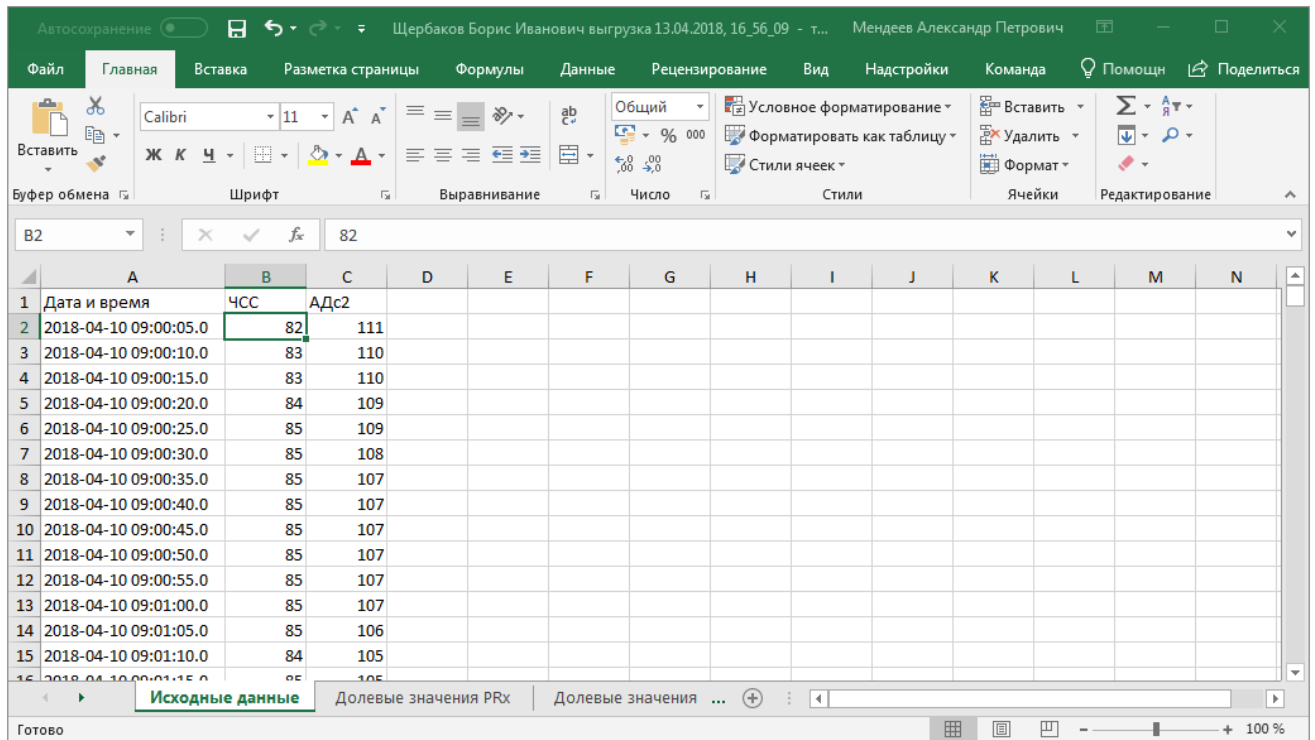


Рисунок 9 – пример составленного отчёта, открытого в программе Microsoft Excel

3.5.6. Панель администрирования

Пользователь с ролью «Администратор» может получить доступ в административную панель, где осуществляется управление внутренними картотеками и настройками программы. Доступ в панель администрирования осуществляется посредством нажатия на кнопку «Администрирование», расположенную в панели навигации в верхней части области страницы.

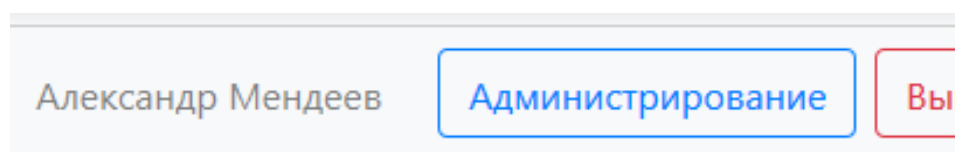


Рисунок 10 – кнопка перехода в панель администрирования

Панель администрирования условно разделена на две части: боковая панель и текущий раздел. В боковой панели пользователь выбирает раздел панели, который он хочет просмотреть. Выбранный раздел отображается в правой части страницы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

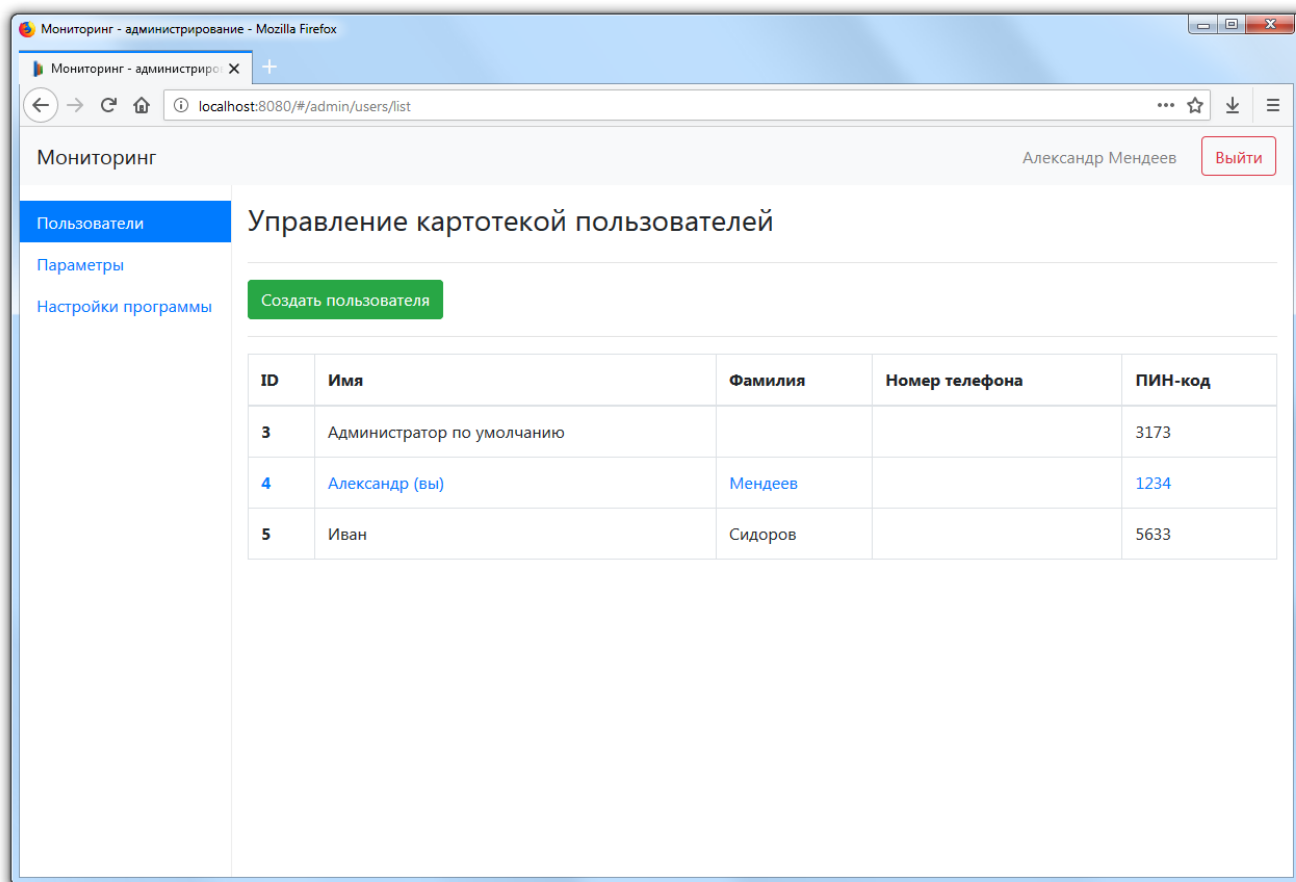


Рисунок 11 – панель администрирования

3.5.6.1. Управление картотекой пользователей

Администратор может совершать операции CRUD над картотекой пользователей программы. Управление картотекой пользователей осуществляется в разделе административной панели «Пользователи» (рис. 11). В данном разделе администратор может создать пользователя, изменить существующего пользователя или удалить пользователя.

3.5.6.1.1. Создание

Для инициирования процесса создания пользователя необходимо нажать на кнопку «Создать пользователя», находящуюся в разделе «Пользователи» административной панели (рис. 11). После нажатия на данную кнопку произойдет переход на страницу создания пользователя (рис. 12).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Рисунок 12 – страница создания пользователя

Необходимо заполнить поля формы создания пользователя, указав нужную информацию. После того, как необходимая информация будет введена, необходимо нажать на кнопку «Сохранить». Чтобы вернуться на предыдущий экран без сохранения изменений необходимо нажать на кнопку «Отмена» или выполнить переход на предыдущую страницу в веб-браузере.

3.5.6.1.2. Просмотр или изменение данных существующего пользователя

Для просмотра или изменения записи пользователя необходимо выбрать соответствующего пользователя из списка в разделе «Пользователи» административной панели (рис. 11). После выбора пользователя произойдет переход к странице редактирования пользователя (рис. 13). Можно просмотреть или изменить данные в соответствующих полях. Для сохранения введенных изменений необходимо нажать на кнопку «Сохранить». Чтобы вернуться на предыдущий экран без сохранения изменений необходимо нажать на кнопку «Отмена» или выполнить переход на предыдущую страницу в веб-браузере.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

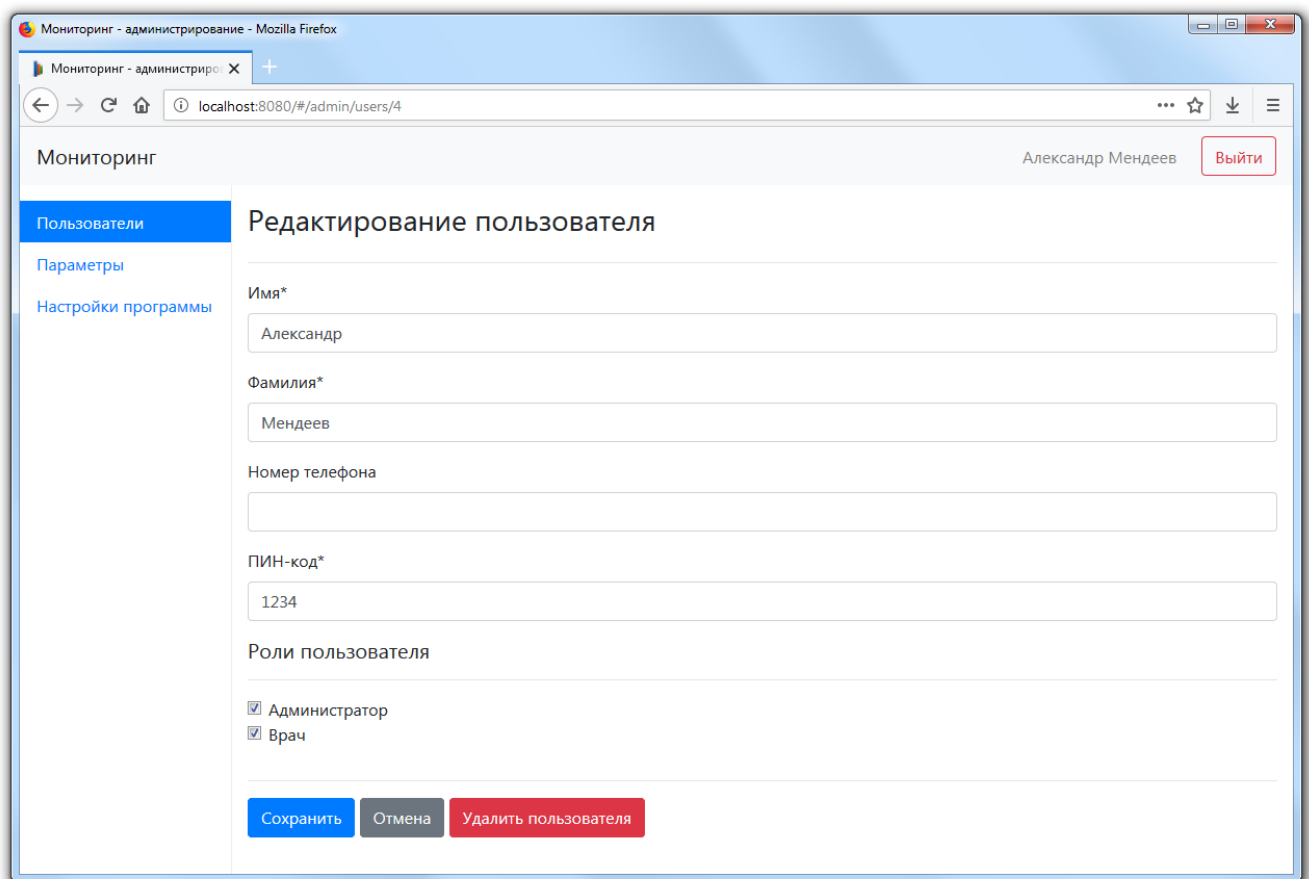


Рисунок 13 – страница редактирования пользователя

3.5.6.1.3. Удаление пользователя

Для удаления пользователя необходимо на странице редактирования пользователя (рис. 13) нажать на кнопку «Удалить пользователя». После этого, веб-браузер отобразит запрос подтверждения данного действия (рис. 14). После подтверждения удаления пользователя, запись пользователя будет удалена из программы.

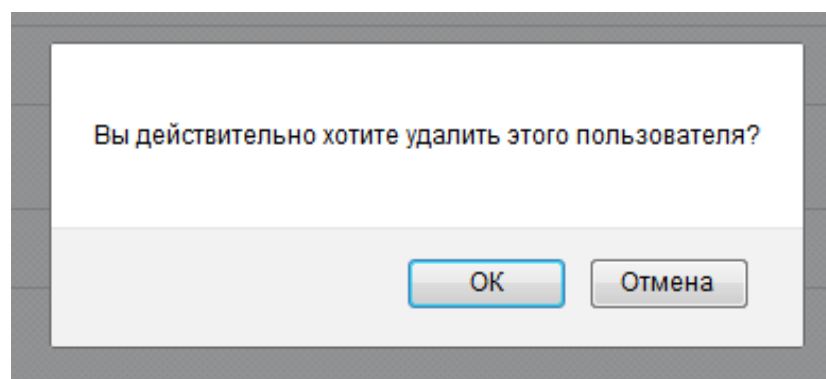


Рисунок 14 – запрос подтверждения удаления пользователя

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.5.6.2. Управление картотекой параметров

Администратор может управлять картотекой параметров, данные по которым будут считываться из БД НИИ НДХиТ. Данный список параметров определяет, по каким значениям будут запрашиваться данные из клинической базы данных НИИ НДХиТ. Для полноценной работы программы необходимо добавить записи параметров АД, ВЧД и ЦПД. Управление картотекой параметров осуществляется в разделе «Параметры» административной панели (рис. 15).

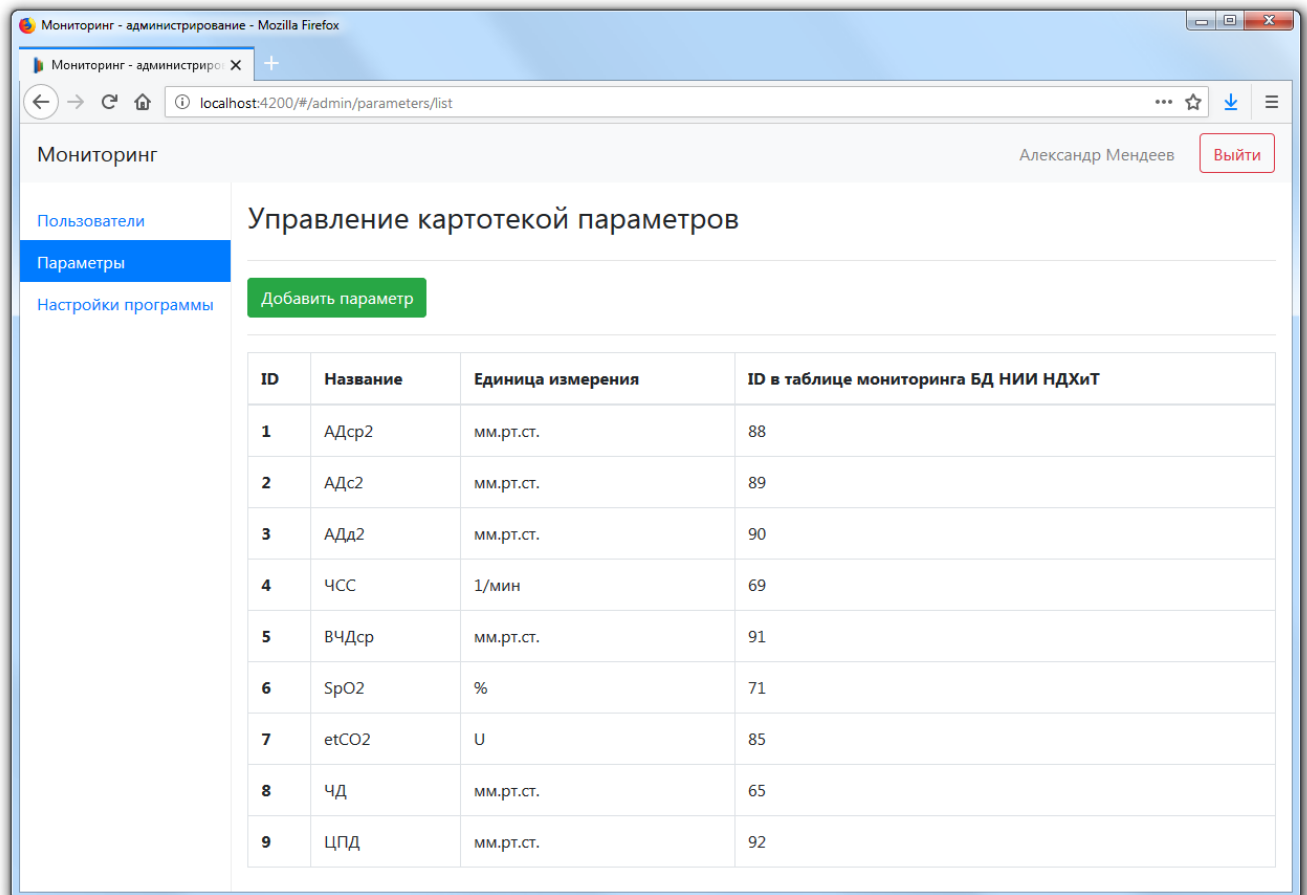


Рисунок 15 – управление картотекой параметров

3.5.6.2.1. Добавление записи параметра

Чтобы добавить новую запись параметра необходимо нажать на кнопку «Добавить параметр», находящуюся в разделе «Параметры» административной панели (рис. 15). После нажатия на кнопку добавления параметра произойдет переход на страницу создания записи параметра.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

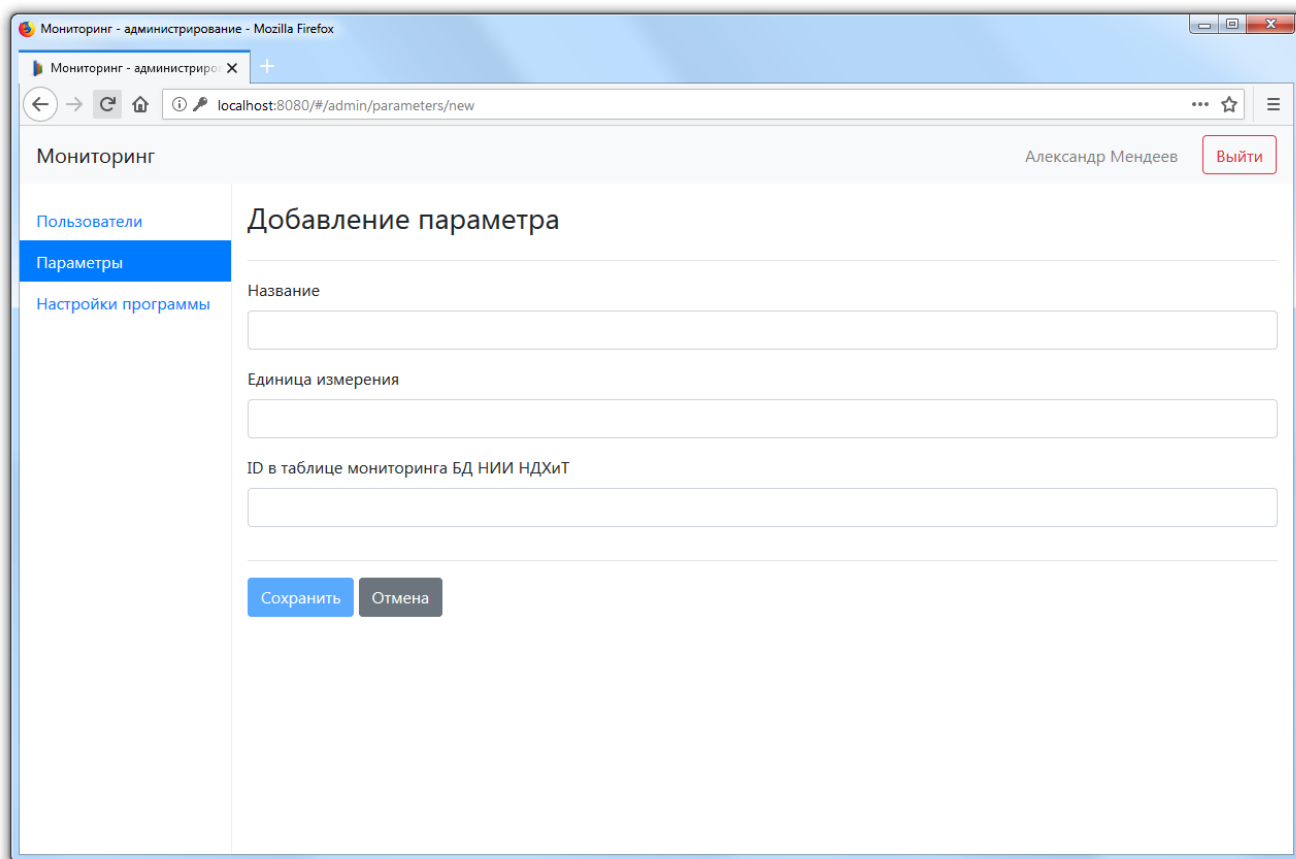


Рисунок 16 – страница добавления записи параметра

Для создания новой записи необходимо заполнить все поля данной формы и нажать на кнопку «Сохранить». После этого, запись с введенными данными будет сохранена и произойдет возврат к предыдущей странице. Чтобы вернуться на предыдущий экран без сохранения изменений необходимо нажать на кнопку «Отмена» или выполнить переход на предыдущую страницу в веб-браузере.

3.5.6.2.2. Просмотр или изменение записи параметра

Чтобы просмотреть существующую запись параметра или внести в неё изменения необходимо выбрать соответствующую запись параметра из списка в разделе «Параметры» административной панели (рис. 15). После выбора соответствующей записи из списка произойдет переход к странице редактирования выбранной записи параметра (рис. 17).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

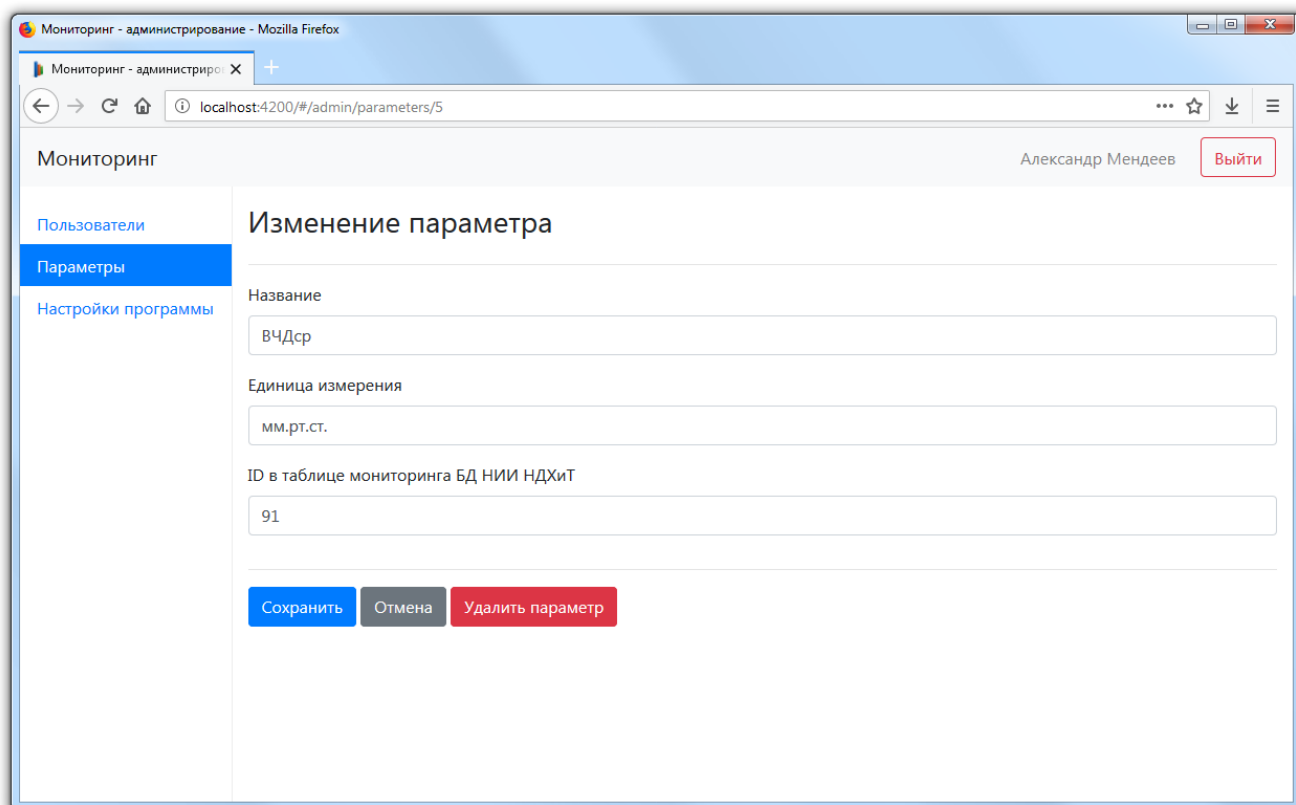


Рисунок 17 – страница изменения записи параметра

При создании или внесении изменений в запись параметра администратор должен указать: название параметра, единицу измерения и идентификатор параметра, соответствующий идентификатору данного параметра в таблице мониторинга в базе данных НИИ НДХиТ. Для сохранения изменений необходимо нажать на кнопку «Сохранить». После нажатия на кнопку «Сохранить» введенные изменения будут сохранены, а пользователь будет возвращен к предыдущей странице. Чтобы вернуться на предыдущий экран без сохранения изменений необходимо нажать на кнопку «Отмена» или выполнить переход на предыдущую страницу в веб-браузере.

3.5.6.2.3. Удаление записи параметра

Чтобы удалить запись параметра необходимо выбрать её из списка в разделе «Параметры» административной панели (рис. 15) и, после перехода на страницу редактирования параметра (рис. 17), нажать на кнопку «Удалить параметр». После этого должен появиться запрос подтверждения удаления (рис. 18).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

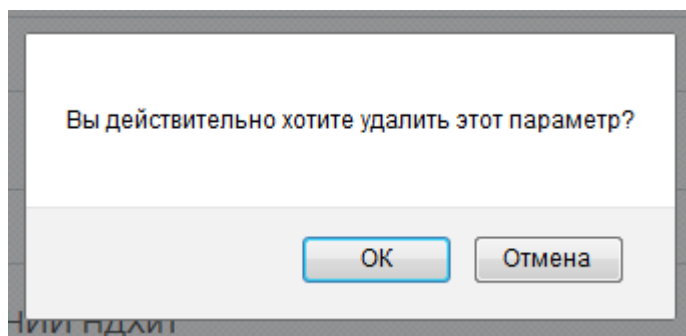


Рисунок 18 – запрос подтверждения удаления записи параметра

После подтверждения удаления запись будет удалена из базы данных.

3.5.6.3. Управление настройками программы

Раздел «Настройки программы» содержит глобальные настройки программы (рис. 19).

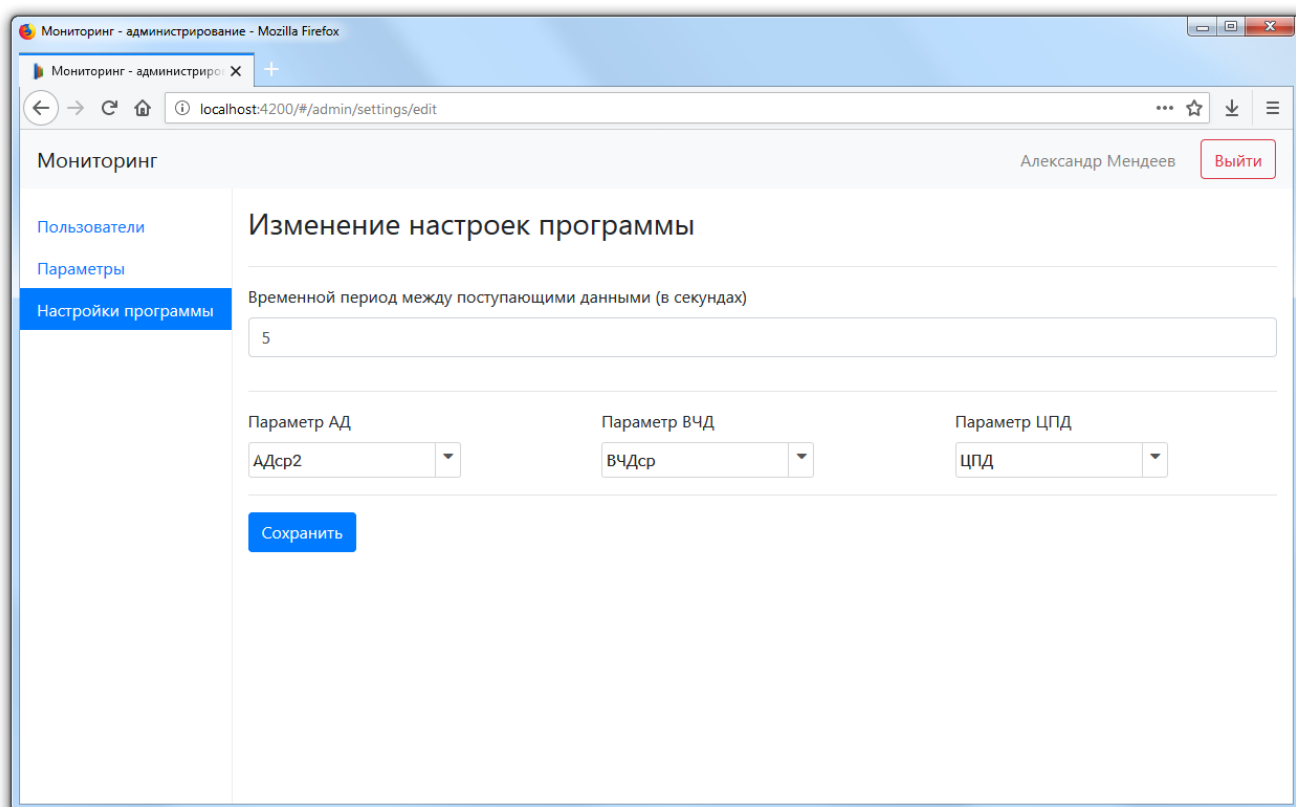


Рисунок 19 – раздел настроек программы

Временной период между поступающими данными определяет временной промежуток между получаемыми данными (частоту дискретизации данных). Данный параметр не следует путать с интервалом, который задается в конфигурационном файле в серверной части программы. Временной период между поступающими данными используется при вычислении индексов ауторегуляции PRx и VRx для вычисления размера окна вычислений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Также в данном разделе администратор задает соответствия параметров АД, ВЧД, ЦПД записям картотеки параметров, выбирая соответствующие записи параметров из выпадающего списка. Для сохранения изменений необходимо нажать на кнопку «Сохранить».

3.5.7. Выход из приложения (сброс аутентификации)

Выход из текущей сессии пользователя осуществляется посредством нажатия на кнопку «Выйти», расположенную в верхней части страницы справа.

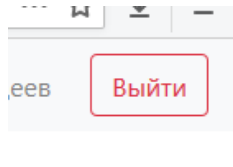


Рисунок 20 – кнопка выхода

После нажатия на данную кнопку веб-браузер запросит подтверждение данного действия. После подтверждения выхода текущая сессия пользователя будет сброшена, а пользователь будет возвращен на страницу входа.

3.6. Остановка программы

Остановка сервера осуществляется средствами операционной системы с помощью клавиатурных сочетаний, вводимых пользователем. Для операционных систем Windows и Linux таким клавиатурным сочетанием является одновременное нажатие клавиш Ctrl и C.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. СООБЩЕНИЯ ОПЕРАТОРУ

- 1) «Произошла ошибка при получении списка пациентов из БД». Возникает в серверной части программы при отсутствии соединения с клинической базой данных НИИ НДХиТ. Необходимо удостовериться в наличии соединения с БД НИИ НДХиТ, а также в правильности введенных данных конфигурации соединения;
- 2) «Соединение установлено». Отображается в виде надписи зеленого цвета на странице мониторинга пациента. Обозначает успешно установленное WebSocket соединение с серверной частью программы. Не требуется предпринимать никаких действий;
- 3) «Соединение потеряно. Выполняется повторное установление соединения...». Данное сообщение отображается в виде красной надписи на красном фоне на странице мониторинга пациента при потере WebSocket соединения с серверной частью программы. Необходимо удостовериться в корректной работе серверной части программы, а также в наличии соединения с сервером. В случае, если соединение не будет восстановлено автоматически спустя некоторый промежуток времени, следует перезагрузить страницу в веб-браузере;
- 4) «Неправильный ПИН-код». Данное сообщение возникает при попытке ввода недействительного ПИН-кода на странице входа. Следует убедиться в правильности вводимых данных;
- 5) «Запрашиваемый сервер не отвечает» или «Не удалось соединиться с запрашиваемым сервером». Возникает при отсутствии соединения с серверной частью программы. Необходимо убедиться в наличии соединения и сервером, а также в корректной работе серверной части программы. Затем следует повторить попытку;
- 6) «Вы не имеете доступа к запрашиваемому ресурсу». Данная ошибка возникает, когда пользователь, не имеющий соответствующих прав (ролей) пытается перейти на страницу, которая требует наличия данных прав. Следует обратиться к администратору для решения данного вопроса;
- 7) «Запись создана. Запись параметра с введенными данными успешно сохранена в базе данных». Данное сообщение свидетельствует о том, что запись параметра была успешно создана и сохранена в базе данных. Не требуется предпринимать никаких дополнительных действий;
- 8) «Запись создана. Запись пользователя с введенными данными успешно сохранена в базе данных». Данное сообщение свидетельствует о том, что запись пользователя была

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

успешно создана и сохранена в базе данных. Не требуется предпринимать никаких дополнительных действий;

- 9) «Запись удалена. Выбранная запись успешно удалена из базы данных». Данное сообщение свидетельствует о том, что выбранная запись была успешно удалена из базы данных. Не требуется предпринимать никаких дополнительных действий;
- 10) «Изменения сохранены. Введённые данные успешно сохранены». Данное сообщение свидетельствует о том, что выбранная запись была успешно изменена. Не требуется предпринимать никаких дополнительных действий;
- 11) «Настройки сохранены. Введённые данные успешно сохранены». Данное сообщение свидетельствует о том, что введенные настройки программы были успешно сохранены в базе данных. Не требуется предпринимать никаких дополнительных действий.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] URL: <http://fmi.asf.ru/library/book/Gost/19201-78.html> (дата обращения 05.05.2018);
- 2) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [Электронный ресурс] URL: http://www.rugost.com/index.php?option=com_content&view=article&id=53:19106-78&catid=19&Itemid=50 (дата обращения 05.05.2018).
- 3) ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению [Электронный ресурс] URL: http://www.rugost.com/index.php?option=com_content&view=article&id=65:19505-79&catid=19&Itemid=50 (дата обращения 05.05.2018);
- 4) Java SE Technologies - Database [Электронный ресурс] URL: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html> (дата обращения 05.05.2018);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2

СПИСОК ИСПОЛЬЗОВАННЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ

Ниже приведен список терминов и сокращений, используемых в данном документе.

PRx – Pressure Reactivity Index

VRx – Variation Reactivity Index

JDBC (Java Database Connectivity) – спецификация (стандарт) программного интерфейса для платформонезависимого соединения между программами на платформе Java и СУБД.

СУБД – система управления базами данных

БД – база данных

НИИ НДХиТ – Научно-исследовательский институт неотложной детской хирургии и травматологии

Активный пациент – пациент, находящийся в данный момент на лечении в НИИ НДХиТ в отделении интенсивной терапии

АД – артериальное давление

ВЧД – внутричерепное давление

ЦПД – церебральное перфузионное давление

CRUD (create, read, update, delete) – операции создания, чтения, изменения и удаления информации

Аутентификация – процесс проверки подлинности введенных пользователем данных и установления личности этого пользователя

ПИН-код – цифробуквенное сочетание символов, которое используется для прохождения аутентификации и позволяет однозначно идентифицировать пользователя программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 34 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"**

Факультет компьютерных наук, департамент программной инженерии

СОГЛАСОВАНО

Кандидат технических наук, руководитель
отдела новых технологий НИИ неотложной
детской хирургии и травматологии

_____ С.Б. Арсеньев
" __ " _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель ОП
"Программная инженерия" Национального
исследовательского университета "Высшая
школа экономики"

_____ В.В. Шилов
" __ " _____ 2018 г.

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.15-01 51 01-1-ЛУ

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Исполнитель

студент группы БПИ143

_____ /А.П. Мендеев /

« ____ » _____ 2018

УТВЕРЖДЕН

RU.17701729.04.15-01 51 01-1-ЛУ

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Программа и методика испытаний

RU.17701729.04.15-01 51 01-1

Листов 36

<i>Инв. № подл</i>	
<i>Подп. и дата</i>	
<i>Взам. инв. №</i>	
<i>Инв. № дубл.</i>	
<i>Подп. и дата</i>	

СОДЕРЖАНИЕ

1. Объект испытаний	119
1.1. Наименование.....	119
1.2. Область применения.....	119
2. Цель испытаний	120
3. Требования к программе	121
3.1. Требования к функциональной составляющей программы.....	121
3.1.1. Требования к составу выполняемых функций	121
3.1.2. Требования к входным данным.....	122
3.1.3. Требования к выходным данным	122
3.2. Требования к пользовательскому интерфейсу.....	123
3.3. Требования к надежности	125
3.4. Условия эксплуатации.....	125
3.4.1. Требования к квалификации пользователя	125
3.4.2. Требования к видам обслуживания	125
3.4.3. Требования к климатическим условиям.....	125
3.5. Требования к составу и параметрам технических средств	125
3.6. Требования к информационной и программной совместимости	125
3.6.1. Требования к программному обеспечению	125
3.6.2. Требования к исходным кодам и языкам программирования.....	126
3.6.3. Требования к защите информации	126
4. Требования к программной документации	127
4.1. Специальные требования к документации.....	127
4.2. Состав программной документации	127
5. Состав и порядок испытаний.....	128
5.1. Состав технических средств	128
5.2. Состав программных средств	128
5.3. Порядок проведения испытаний	128
6. Методы испытаний	129
6.1. Проверка выполнения функциональных требований	129
6.2. Проверка выполнения требований к пользовательскому интерфейсу	146

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6.3. Проверка выполнения требований к программной документации.....	146
Приложение 1	148
Приложение 2	149

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Наименование

«Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой».

1.2. Область применения

Данная программа предназначена для применения в отделении интенсивной терапии при диагностике состояния ауторегуляции мозгового кровообращения у больных с черепно-мозговой травмой.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ЦЕЛЬ ИСПЫТАНИЙ

Цель настоящих испытаний - проверить работоспособность всех заявленных функциональных характеристик, а также проверить на соответствие программы всем требованиям, предъявленным к ней в тексте технического задания.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1. Требования к функциональной составляющей программы

3.1.1. Требования к составу выполняемых функций

- 1) пользователь должен иметь возможность задать настройки соединения с клинической базой данных НИИ НДХиТ;
- 2) пользователь должен иметь возможность задать настройки соединения с базой данных, которая будет использоваться для хранения данных программы;
- 3) программа должна автоматически создавать необходимые сущности (таблицы) в базе данных, которая используется для хранения данных программы;
- 4) пользователь должен иметь возможность задать значения числовых интервалов для классификации значений индексов ауторегуляции PRx и VRx. Интервалы должны задаваться в виде граничных значений интервалов для всех трёх классов значений;
- 5) пользователь должен иметь возможность задать частоту (временной интервал) для операции запроса данных пациентов из БД НИИ НДХиТ в виде количества секунд;
- 6) пользователь должен иметь возможность проходить аутентификацию посредством ввода своего ПИН-кода;
- 7) пользователь должен иметь возможность выйти из текущей сессии (сбросить аутентификацию);
- 8) пользователь должен иметь доступ к списку активных пациентов;
- 9) пользователь должен иметь возможность поиска по списку активных пациентов;
- 10) пользователь должен иметь возможность перейти в раздел мониторинга для осуществления наблюдения за состоянием ауторегуляции, выбрав пациента из списка активных пациентов;
- 11) информация в разделе мониторинга должна обновляться автоматически без необходимости перезагружать страницу;
- 12) пользователь должен иметь возможность совершить выгрузку данных выбранного пациента за выбранный период дат;
- 13) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять размер временного окна для вычисления индекса PRx (в секундах);
- 14) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять размер временного окна для вычисления индекса VRx (в секундах);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 15) пользователь с ролью «Врач» должен иметь возможность изменять для каждого активного пациента размер временного буфера (в секундах) для вычисления частот встречаемости разных классов значений индекса PRx;
- 16) пользователь с ролью «Врач» должен иметь возможность изменять для каждого активного пациента размер временного буфера (в секундах) для вычисления частот встречаемости разных классов значений индекса VRx;
- 17) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять видимость группы показателей (элементов интерфейса) на странице мониторинга пациента, которые содержат данные об относительной частоте встречаемости разных классов значений индексов PRx/VRx и графики ВЧД/ЦПД;
- 18) пользователь с ролью «Врач» должен иметь возможность для каждого активного пациента изменять видимость отдельных параметров (строк) в таблице необработанных значений, находящейся на странице мониторинга пациента;
- 19) пользователь с ролью «Администратор» должен иметь доступ в панель администрирования;
- 20) пользователь с ролью «Администратор» должен иметь возможность управления картотекой пользователей программы;
- 21) пользователь с ролью «Администратор» должен иметь возможность управления картотекой параметров, значения по которым будут запрашиваться программой из БД НИИ НДХиТ.

пользователь с ролью «Администратор» должен иметь возможность управления картотекой параметров, значения по которым будут запрашиваться программой из БД НИИ НДХиТ.

3.1.2. Требования к входным данным

Входными данными являются значения в таблицах клинической базы данных НИИ НДХиТ. Программа должна производить считывание данных пациентов из клинической базы данных, используя SQL запросы.

3.1.3. Требования к выходным данным

Выходными данными программы являются относительные частоты встречаемости классов значений индексов ауторегуляции PRx и VRx относительно общего количества значений индексов в пределах заданного временного буфера. Программа должна для каждого

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

активного пациента производить расчет и классификацию значений индексов ауторегуляции PRx и VRx. Классификация значений индексов ауторегуляции PRx и VRx должна производиться по заданным пользователем в конфигурационном файле интервалам значений. В процессе классификации значений индексов ауторегуляции PRx и VRx должны выделяться три класса значений:

- а. класс значений, обозначающих стабильную ауторегуляцию мозга;
- б. класс значений, обозначающих дестабилизацию ауторегуляции мозга (промежуточное состояние);
- в. класс значений, обозначающих утрату ауторегуляции мозга.

Программа должна производить расчет частот встречаемости значений, отнесённых к тем или иным классам значений в пределах заданного для данного пациента временного буфера. Данные частоты встречаемости должны быть выражены в виде круговых диаграмм с секторами различных цветов в разделе мониторинга пациента. Доля значений класса «Стабильная ауторегуляция» должна быть представлена в виде сектора зелёного цвета, доля значений класса «Дестабилизация ауторегуляции» (неопределенное состояние) – в виде сектора жёлтого цвета, доля значений класса «Утраченная ауторегуляция» - в виде сектора красного цвета. Также данные круговые диаграммы должны иметь развертку по времени в виде линейных графиков с линиями разных цветов, соответствующих определенному классу значений. Помимо круговых диаграмм и линейных графиков, должна присутствовать информация о приросте или уменьшении доли соответствующего класса значений по сравнению с предыдущими показателями в виде надписи цвета, соответствующего цвету класса.

Помимо относительной частоты встречаемости разных классов значений индексов PRx и VRx, в программе должны быть представлены необработанные значения параметров, получаемые из клинической базы данных НИИ НДХиТ. Необработанные значения представляются в виде числовых значений в таблице.

Все перечисленные выше виды данных должны быть представлены в разделе мониторинга пациента.

При выгрузке данных пациентов, выгруженные данные должны быть представлены в виде файла формата Microsoft Excel, содержащего необходимые данные.

3.2. Требования к пользовательскому интерфейсу

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 1) в программе должен присутствовать раздел аутентификации пользователя (вход в программу);
- 2) в программе должен присутствовать раздел со списком активных пациентов;
- 3) в разделе со списком активных пациентов должно присутствовать текстовое поле для ввода поискового запроса;
- 4) в программе должен присутствовать раздел мониторинга выбранного активного пациента;
- 5) в программе должен присутствовать раздел с настройками для выбранного активного пациента;
- 6) в программе должна присутствовать панель администрирования с разделами для управления картотекой параметров и картотекой пользователей;
- 7) в панели администрирования навигация должна осуществляться путем выбора соответствующего раздела в боковом меню. Выбранный раздел должен выделяться другим цветом;
- 8) в программе должен присутствовать элемент интерфейса, позволяющий сбросить текущую сессию пользователя (осуществить выход);
- 9) в программе должен присутствовать элемент интерфейса, содержащий имя текущего пользователя;
- 10) в разделе мониторинга пациента должна присутствовать надпись, содержащая ФИО наблюдаемого пациента;
- 11) в разделе мониторинга пациента должна присутствовать таблица, содержащая необработанные данные, получаемые из клинической базы данных НИИ НДХиТ;
- 12) в разделе мониторинга рядом с круговыми диаграммами частот встречаемости классов значений индексов PRx и VRx также должна находиться развертка данных диаграмм по времени в виде линейных графиков с линиями соответствующих цветов;
- 13) в разделе мониторинга рядом с круговыми диаграммами частот встречаемости классов значений индексов PRx и VRx также должны быть текстовые обозначения, отображающие прирост/уменьшение в процентах долей соответствующих классов значений по сравнению с прошлыми показателями;
- 14) в программе должен быть представлен раздел для выгрузки данных пациентов;
- 15) в разделе для выгрузки данных пациентов должен быть элемент интерфейса в виде календаря, позволяющий выбрать диапазон дат для выгрузки;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

16) в разделе для выгрузки данных должны быть представлены элементы интерфейса для множественного выбора типов данных, которые следует включить в выгрузку.

3.3. Требования к надежности

Требования к надежности не предъявляются.

3.4. Условия эксплуатации

3.4.1. Требования к квалификации пользователя

Пользователь должен обладать базовыми навыками работы с компьютером.

3.4.2. Требования к видам обслуживания

На компьютере, где производится выполнение программы, необходимо обеспечить регулярные проверки обновлений ПО. Также необходимо обеспечить защиту компьютера от разного рода компьютерных вирусов.

3.4.3. Требования к климатическим условиям

- 1) Влажность от 10% до 80%;
- 2) Температура от -10°C до 55°C.

3.5. Требования к составу и параметрам технических средств

Для корректной работы программы требуется следующий минимальный состав технических средств:

- 1) Компьютер, оснащенный 32-разрядным или 64-разрядным процессором с тактовой частотой
1 ГГц и выше или аналогичный процессор;
- 2) 1 ГБ оперативной памяти или больше;
- 3) 128 МБ графической памяти;
- 4) жесткий диск объемом 40 ГБ с 1 ГБ свободного места;
- 5) компьютерная мышь;
- 6) клавиатура.

3.6. Требования к информационной и программной совместимости

3.6.1. Требования к программному обеспечению

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 1) Java Runtime Environment версии 8.0 или выше;
- 2) СУБД MySQL версии 5.7 или выше;
- 3) Веб-браузер с поддержкой технологий CSS, HTML, JavaScript.

3.6.2. Требования к исходным кодам и языкам программирования

Программа должна представлять собой веб-приложение, использующее технологии JavaScript.

3.6.3. Требования к защите информации

Требования к защите информации не предъявляются.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1. Специальные требования к документации

- 1) Документация к программе должна быть выполнена в соответствии с ГОСТ 19.106-78;
- 2) Документация должна быть сдана в печатном виде, при этом она должна быть подписана руководителем организации, утвердившей документ на разработку, руководителем разработки и исполнителем.

4.2. Состав программной документации

- 1) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Техническое задание;
- 2) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Руководство оператора;
- 3) «Программа мониторинга ауторегуляции головного мозга у пациентов с черепно-мозговой травмой». Программа и методика испытаний;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СОСТАВ И ПОРЯДОК ИСПЫТАНИЙ

5.1. Состав технических средств

- 1) Компьютер, оснащенный 32-разрядным или 64-разрядным процессором с тактовой частотой
1 ГГц и выше или аналогичный процессор;
- 2) 1 ГБ оперативной памяти или больше;
- 3) 128 МБ графической памяти;
- 4) жесткий диск объемом 40 ГБ с 1 ГБ свободного места;
- 5) компьютерная мышь;
- 6) клавиатура.

5.2. Состав программных средств

- 1) Java Runtime Environment версии 8.0 или выше;
- 2) СУБД MySQL версии 5.7 или выше;
- 3) Веб-браузер с поддержкой технологий CSS, HTML, JavaScript.

5.3. Порядок проведения испытаний

- 1) установление соединения с сетью НИИ НДХиТ;
- 2) установка, конфигурирование и запуск программы согласно руководству оператора;
- 3) прохождение аутентификации в программе;
- 4) добавление записей для параметров АД, ВЧД и ЦПД через панель администрирования программы путем выполнения инструкций, описанных в руководстве оператора;
- 5) установление соответствий параметров АД, ВЧД и ЦПД ранее добавленным записям через панель администрирования программы в разделе настроек путем выполнения инструкций, описанных в руководстве оператора;
- 6) проверка выполнения функциональных требований;
- 7) проверка требований к пользовательскому интерфейсу;
- 8) проверка требований к программной документации.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. МЕТОДЫ ИСПЫТАНИЙ

6.1. Проверка выполнения функциональных требований

6.1.1. Требование 1

Настройки соединения с клинической БД НИИ НДХиТ задаются в конфигурационном файле программы путем установления соответствующих значений следующим параметрам:

- 1) spring.smkm.url;
- 2) spring.smkm.driver-class-name;
- 3) spring.smkm.username;
- 4) spring.smkm.password.

Описания данных параметров представлены в руководстве оператора. Настройки соединения должны вступить в силу при следующем запуске сервера. При указании правильных настроек соединения сервер должен подключиться к заданной БД и запуститься без ошибок.

6.1.2. Требование 2

Настройки соединения с серверной БД MySQL задаются в конфигурационном файле программы путем установления соответствующих значений следующим параметрам:

- 1) spring.sdb.url;
- 2) spring.sdb.driver-class-name;
- 3) spring.sdb.username;
- 4) spring.sdb.password.

Описания данных параметров представлены в руководстве оператора. Настройки соединения должны вступить в силу при следующем запуске сервера. При указании правильных настроек соединения сервер должен подключиться к заданной БД и запуститься без ошибок.

6.1.3. Требование 3

За автоматическую генерацию сущностей в базе данных, которая используется для хранения данных программы, отвечает параметр конфигурации серверной части программы «spring.jpa.hibernate.ddl-auto». Установление значения «update» для данного параметра

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

приводит к проверке структуры БД и автоматической генерации необходимых таблиц при каждом запуске серверной части программы. Если параметру «spring.jpa.hibernate.ddl-auto» в конфигурационном файле программы присвоено значение «update», то после успешного запуска серверной части программы в серверной БД (группа настроек конфигурации «spring.sdb.*») должны появиться следующие таблицы:

- 1) application_settings;
- 2) index_proportion;
- 3) parameter_info;
- 4) patient_data;
- 5) patient_info;
- 6) patient_param_info_rel;
- 7) patient_settings;
- 8) role;
- 9) user;
- 10) user_role.

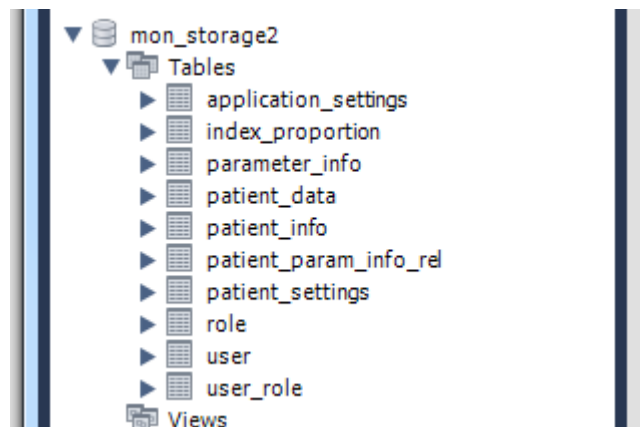


Рисунок 1 – автоматически сгенерированные таблицы в серверной БД (фрагмент снимка рабочего окна программы MySQL Workbench)

6.1.4. Требование 4

Настройки классификации значений индексов ауторегуляции PRx и VRx задаются в виде граничных значений числовых интервалов в конфигурационном файле программы. Необходимо присвоить соответствующие значения следующим параметрам:

- 1) prx.bad.from;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 2) prx.bad.to;
- 3) prx.neutral.from;
- 4) prx.neutral.to;
- 5) prx.good.from;
- 6) prx.good.to;
- 7) vrx.bad.from;
- 8) vrx.bad.to;
- 9) vrx.neutral.from;
- 10) vrx.neutral.to;
- 11) vrx.good.from;
- 12) vrx.good.to.

Подробное описание данных параметров приведено в руководстве оператора. Внесенные изменения должны вступить в силу при следующем запуске сервера.

6.1.5. Требование 5

Частота запроса (временной интервал) данных из клинической БД НИИ НДХиТ задается в виде количества секунд в конфигурационном файле серверной части программы путём установки значения параметра «read_speed». Изменения должны вступить в силу при следующем запуске сервера.

6.1.6. Требование 6

Для прохождения аутентификации необходимо ввести ПИН-код пользователя в форму аутентификации в клиентской части программы (рис. 2), после чего, нажать на кнопку «Войти» или клавишу ввода на клавиатуре. Если в программе отсутствуют пользователи (например, программа запускается в первый раз), можно воспользоваться ПИН-кодом администратора по умолчанию. ПИН-код администратора по умолчанию генерируется сервером автоматически в том случае, если в программе нет ни одного пользователя с ролью «Администратор». ПИН-код администратора по умолчанию после запуска программы должен находиться в текстовом файле password.txt, расположенном в одной папке с исполняемым файлом программы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

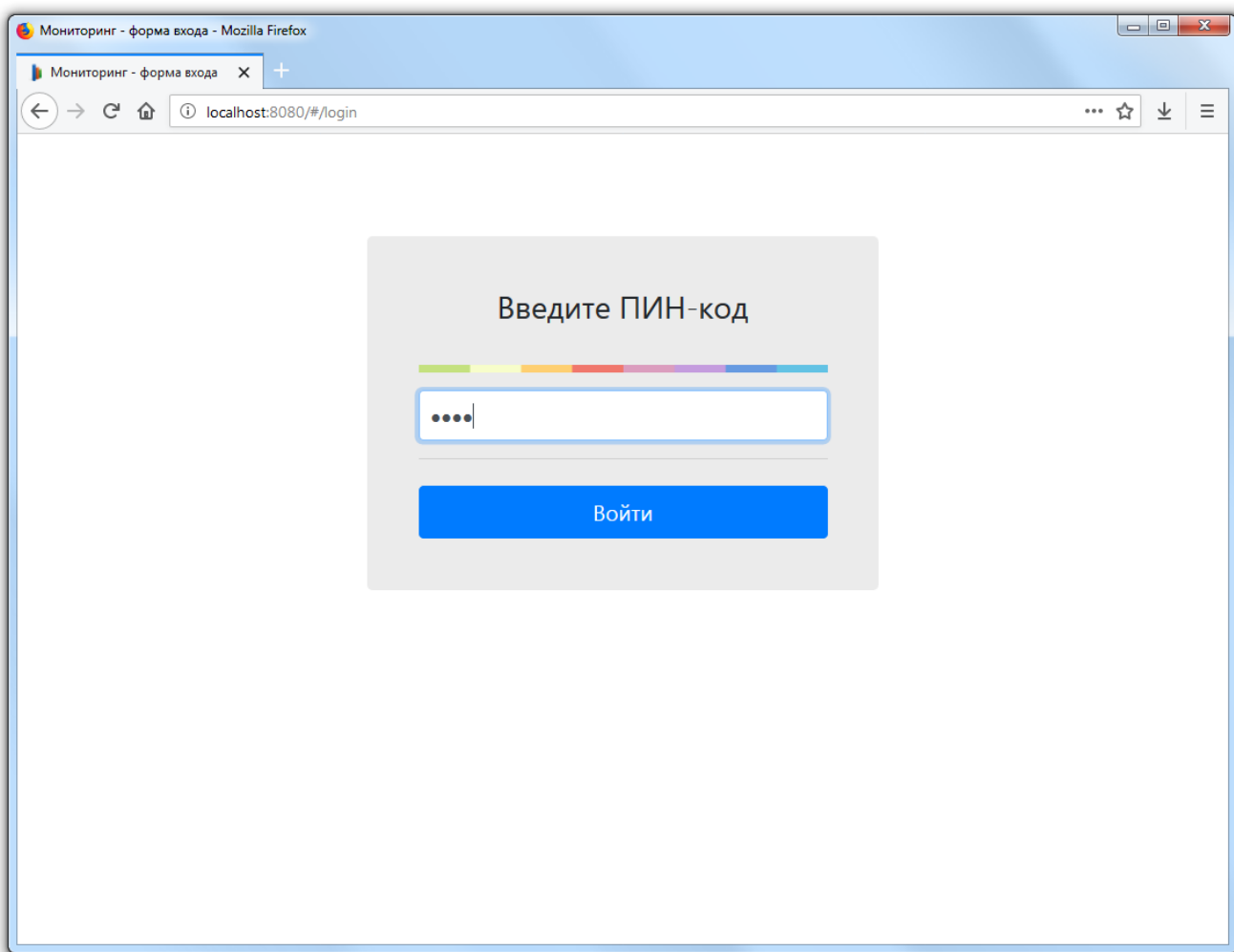


Рисунок 2 – форма аутентификации

При успешной аутентификации должен произойти автоматический переход в начальный раздел программы, содержащий список активных пациентов.

6.1.7. Требование 7

Выход пользователя из текущей сессии осуществляется посредством нажатия на кнопку «Выйти», расположенную в верхней части рабочей области программы (рис. 3).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

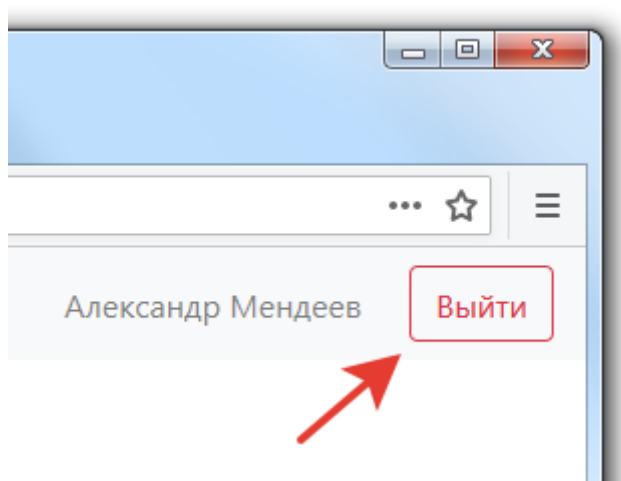


Рисунок 3 – кнопка выхода из текущей сессии

После нажатия на кнопку выхода должно появиться диалоговое окно веб-браузера с запросом подтверждения данного действия (рис. 4).

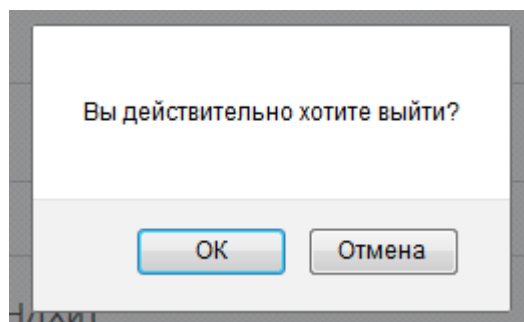


Рисунок 4 – запрос подтверждения выхода из текущей сессии

После подтверждения данного действия текущая сессия будет удалена из локального хранилища веб-браузера и должен произойти автоматический переход на страницу аутентификации (рис. 2).

6.1.8. Требование 8

Доступ к списку пациентов, находящихся в отделении интенсивной терапии НИИ НДХиТ осуществляется в начальном разделе программы (рис. 5) после прохождения пользователем аутентификации. В списке должны быть отображены ФИО пациентов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

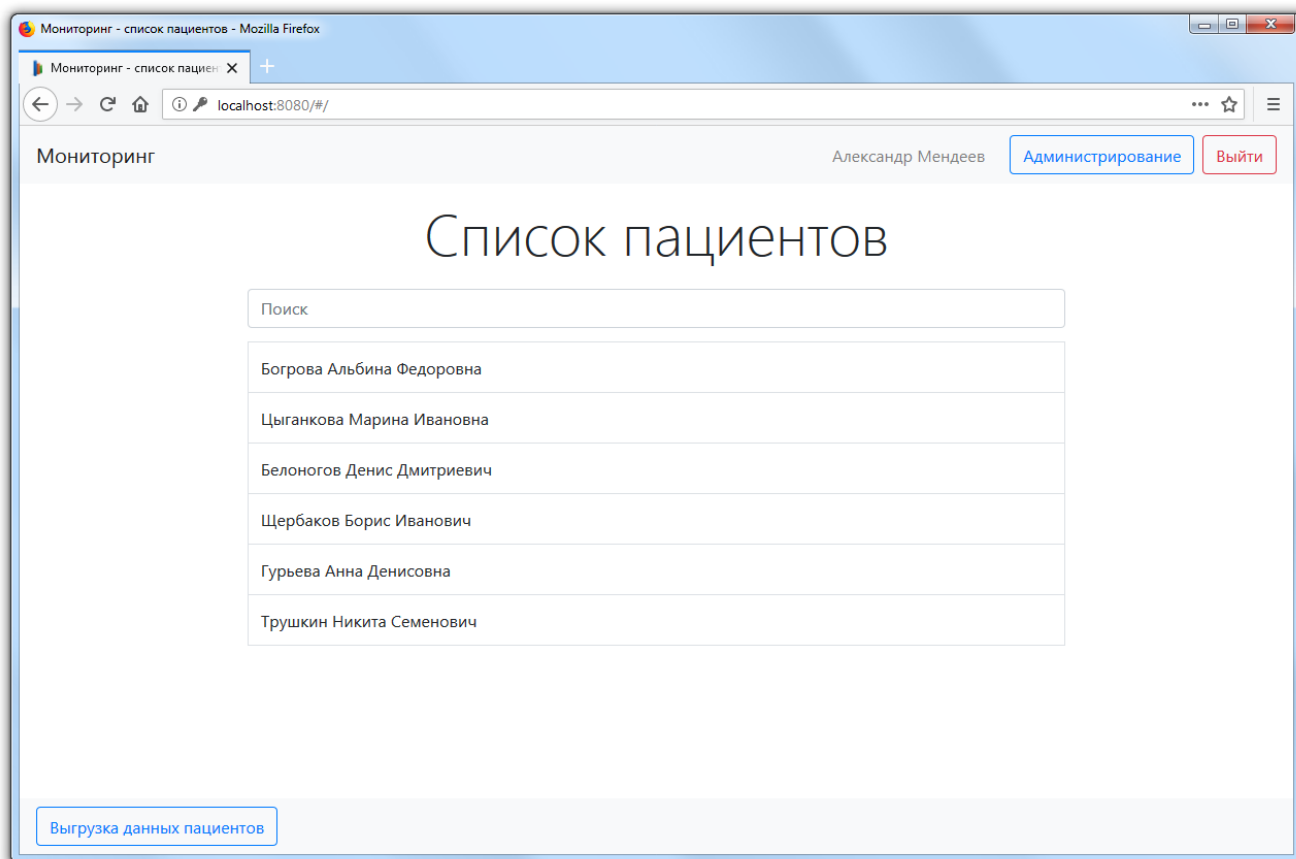


Рисунок 5 – список активных пациентов

6.1.9. Требование 9

Поиск по списку активных пациентов осуществляется посредством ввода соответствующего поискового запроса в текстовое поле, расположенное над списком активных пациентов в начальном разделе программы (рис. 6).

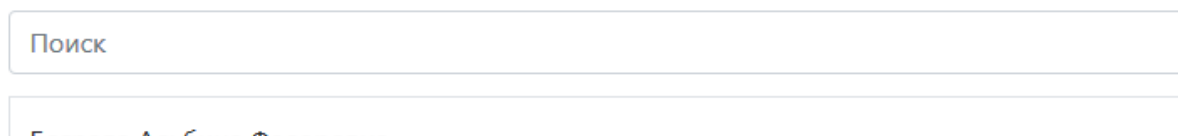


Рисунок 6 – поле ввода поискового запроса

Ввод текста в данное поле должен приводить к фильтрации записей на основании сравнения символов ФИО пациента и строки поискового запроса (рис. 7).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ПАЦИЕНТОВ

де

Белоногов Де нис Дмитриевич

Гурьева Анна Де нисовна

Рисунок 7 – отфильтрованный список активных пациентов

Совпадения фрагментов ФИО пациентов с текстом поискового запроса должны визуально выделяться светло-желтым цветом.

6.1.10. Требование 10

Переход в раздел мониторинга осуществляется путем выбора соответствующего пациента из списка активных пациентов (рис. 5) в начальном разделе программы. После выбора пациента должен произойти переход на страницу мониторинга (рис. 8).

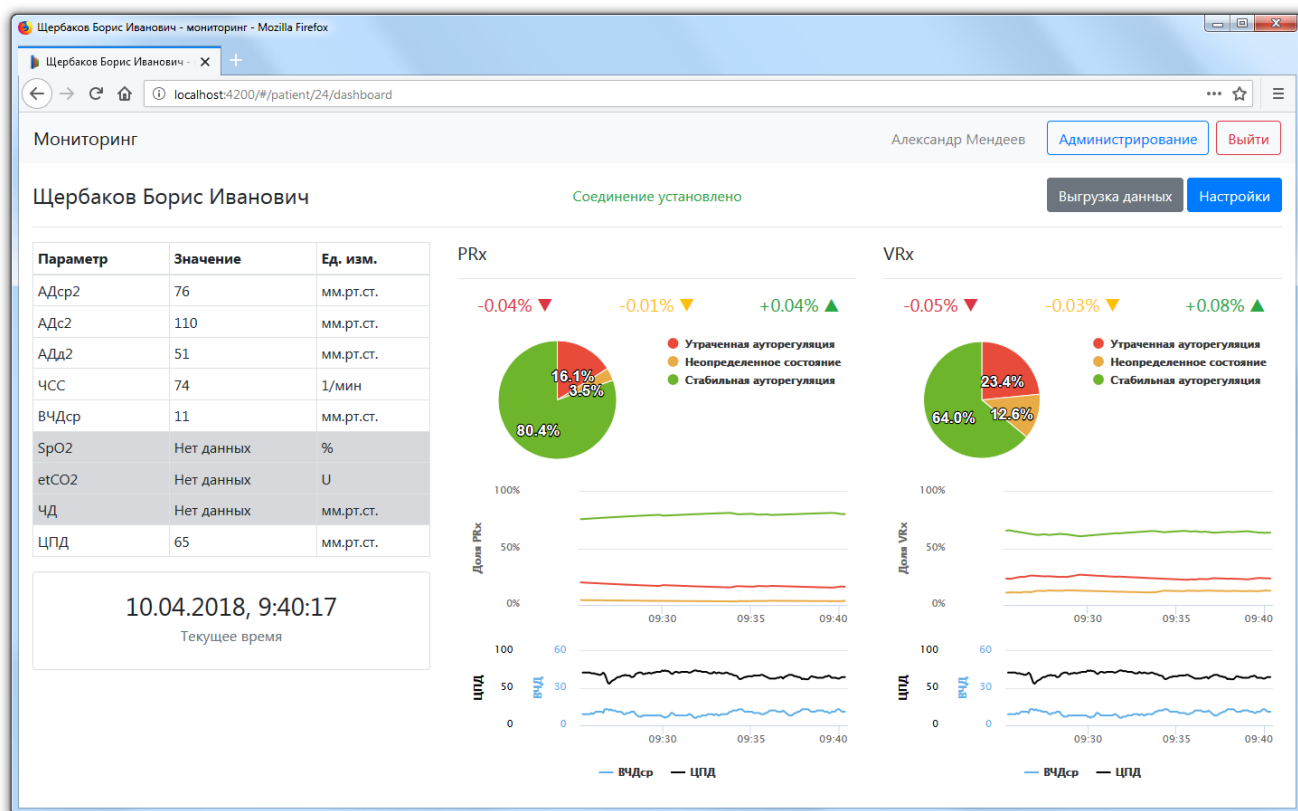


Рисунок 8 – страница мониторинга пациента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6.1.11. Требование 11

Проверка данного требования осуществляется путём просмотра страницы мониторинга пациента в течение некоторого времени. Круговые диаграммы относительных частот встречаемости классов значений индексов PRx и VRx, линейные графики, надписи, обозначающие прирост/уменьшение долей соответствующих классов значений, необработанные значения в таблице параметров и текущее время должны обновляться раз в несколько секунд (в зависимости от заданных настроек) автоматически без необходимости перезагружать страницу.

6.1.12. Требование 12

Для осуществления выгрузки данных пациентов необходимо в начальном разделе программы нажать на кнопку «Выгрузка данных пациентов», расположенную в нижней части рабочей области программы с левой стороны (рис. 9).

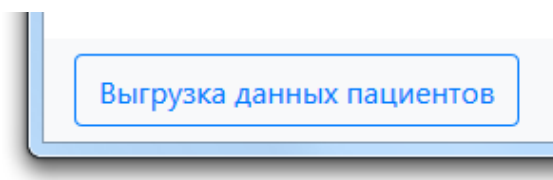


Рисунок 9 – кнопка выгрузки данных пациентов

После нажатия на данную кнопку, должен произойти автоматический переход на страницу выбора пациента для выгрузки данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

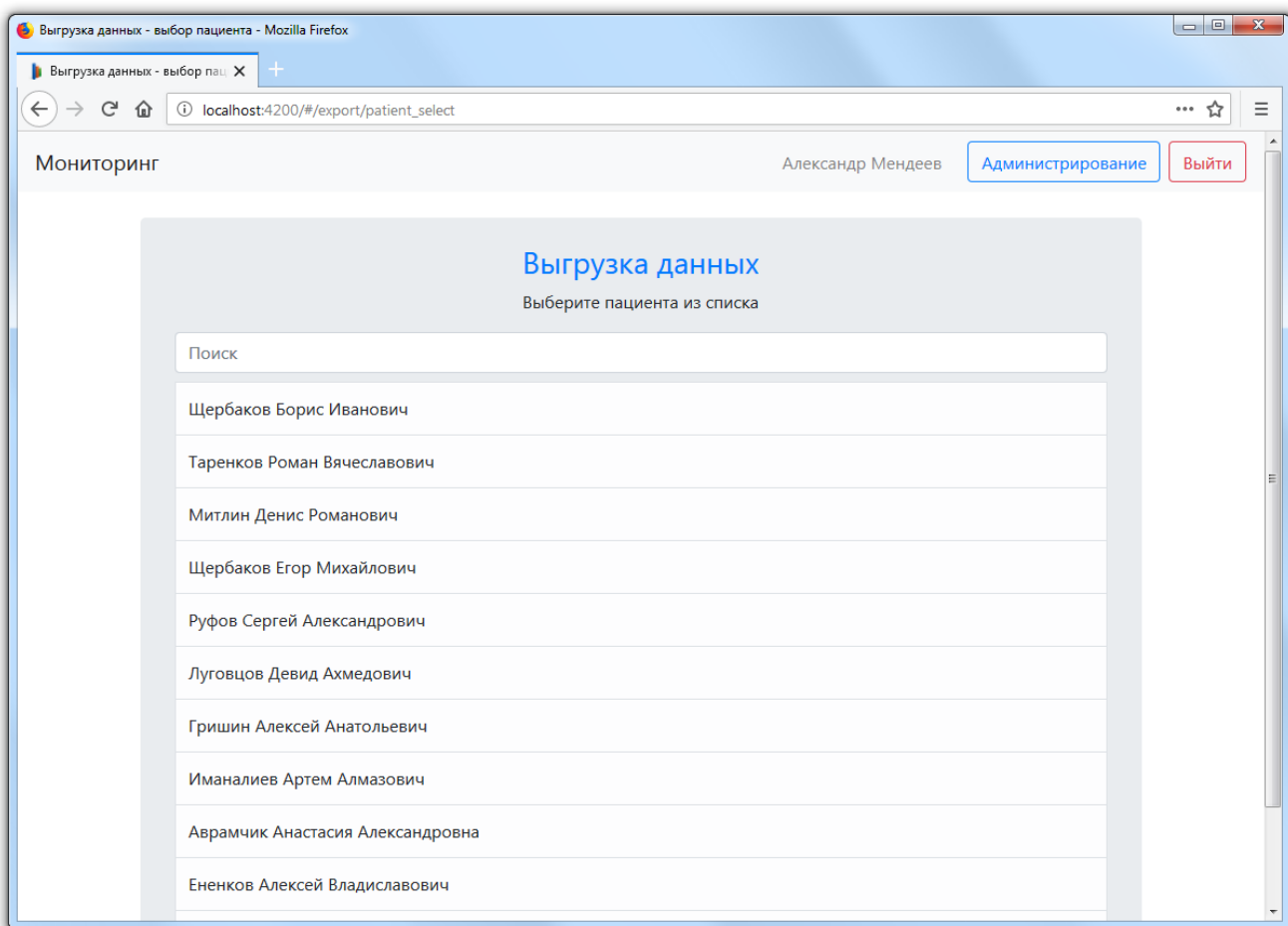


Рисунок 10 – страница выбора пациента для выгрузки данных

На данной странице необходимо выбрать нужного пациента из списка. Для фильтрации списка можно воспользоваться строкой поиска, расположенной сверху списка. После выбора пациента из списка должен произойти переход на страницу настройки параметров выгрузки (рис. 11).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

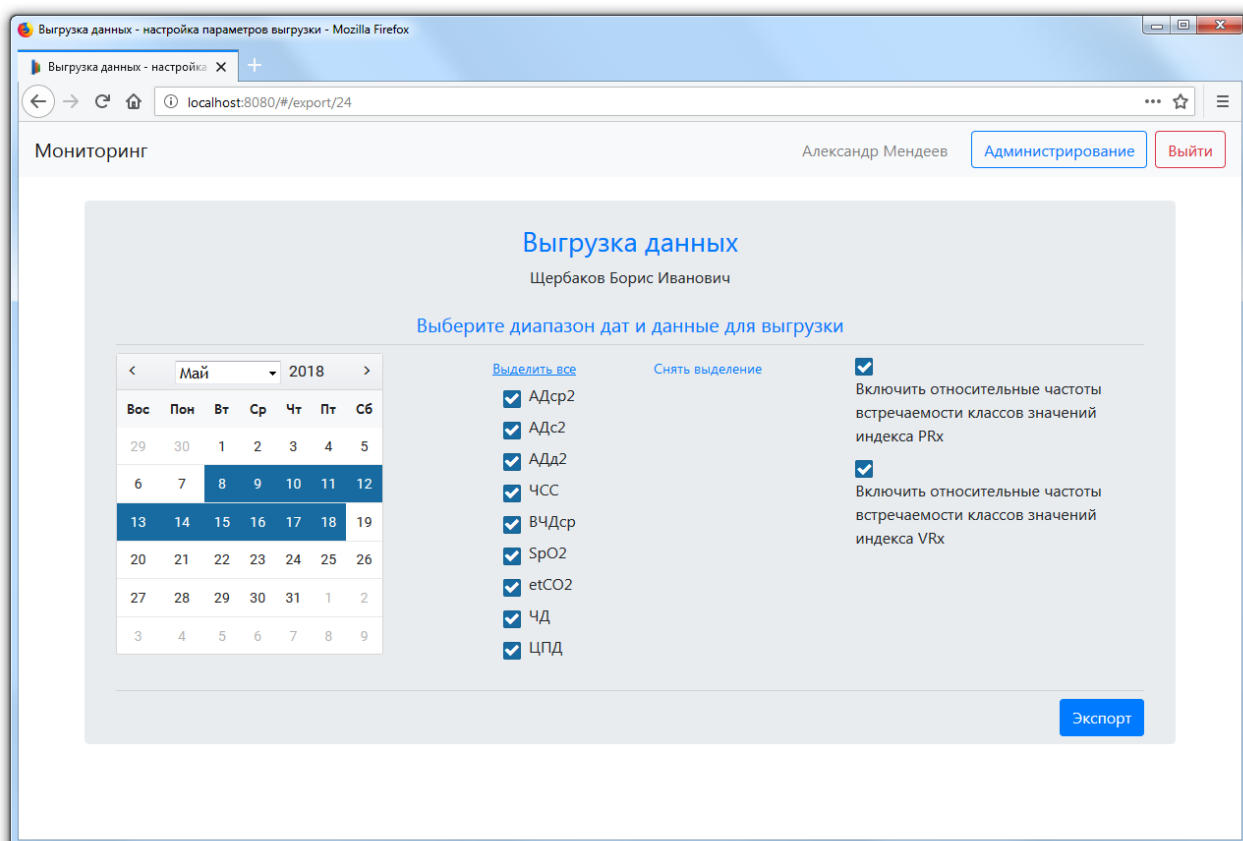


Рисунок 11 – страница настройки параметров выгрузки данных

На данной странице необходимо выставить диапазон дат для выгрузки данных, выбрав граничные даты с помощью соответствующего элемента интерфейса слева. Выбор диапазона дат осуществляется двумя нажатиями: сначала необходимо выбрать начальную дату, а затем – конечную дату диапазона. Выделенный диапазон дат должен выделиться темно-синим цветом. Затем необходимо отметить, какие данные необходимо выгрузить, с помощью элементов интерфейса для множественного выбора. После того, как необходимые параметры выгрузки были заданы, следует нажать на кнопку с надписью «Экспорт». После нажатия кнопки и непродолжительного ожидания веб-браузер должен инициировать загрузку готового файла Microsoft Excel.

Также выгрузку данных можно инициировать, нажав на соответствующую кнопку в разделе мониторинга пациента (рис. 12). После нажатия на данную кнопку должен произойти автоматический переход на страницу настройки параметров выгрузки данных выбранного пациента (рис. 11).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

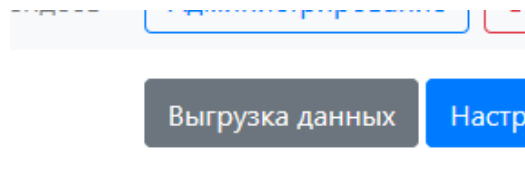


Рисунок 12 – кнопка выгрузки данных, расположенная на странице мониторинга пациента

6.1.13. Требование 13

Для изменения размера временного окна, которое используется при вычислении индекса ауторегуляции PRx, необходимо, будучи авторизованным пользователем с ролью «Врач», нажать на кнопку «Настройки», находящуюся в разделе мониторинга выбранного пациента. После нажатия на данную кнопку должен произойти переход на страницу настроек данного пациента (рис. 13).

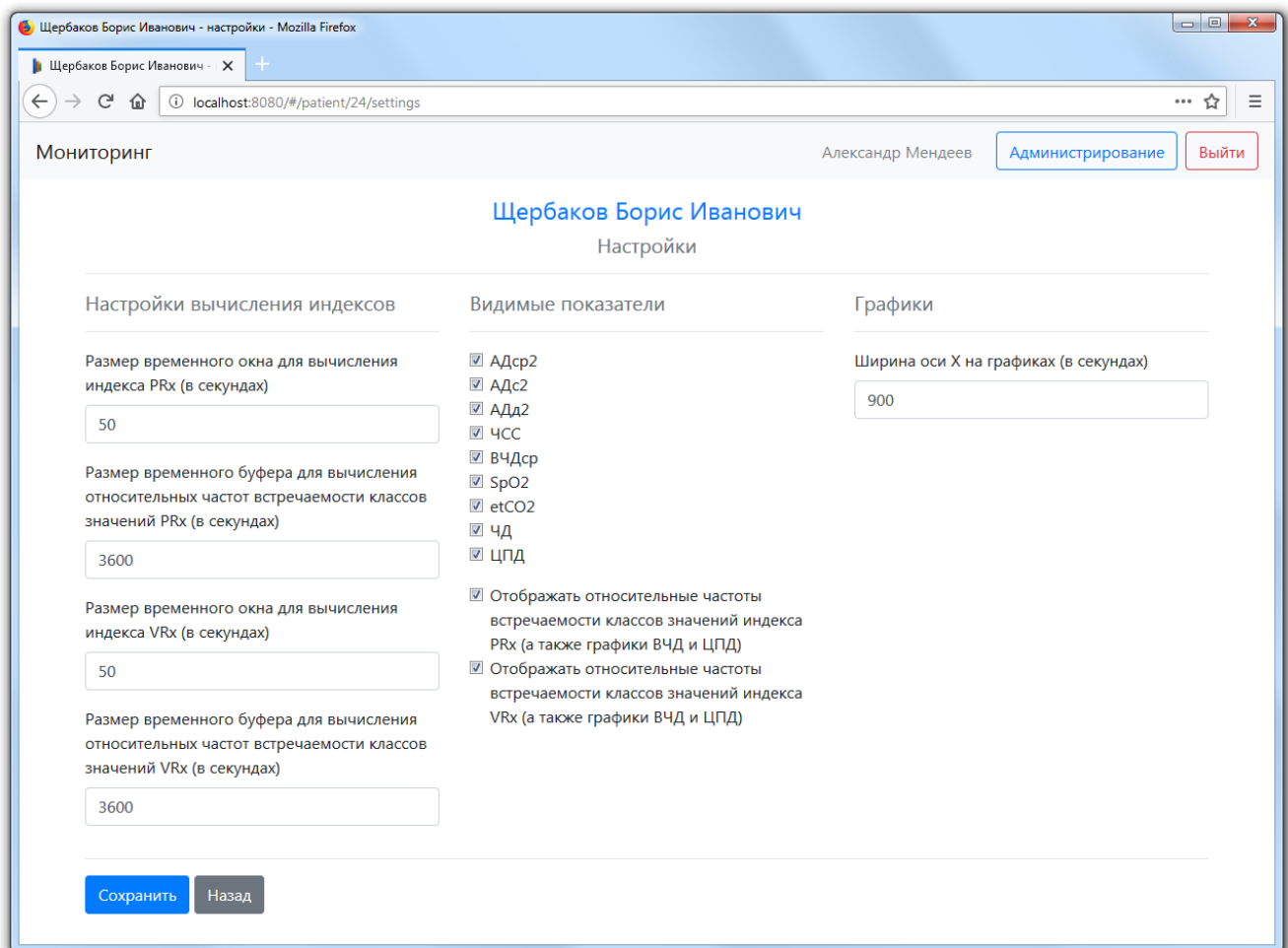


Рисунок 13 – страница настроек пациента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Необходимо ввести размер временного окна в виде количества секунд в текстовое поле с подписью «Размер временного окна для вычисления индекса PRx (в секундах)», а затем – нажать на кнопку «Сохранить». После нажатия на кнопку «Сохранить» должен произойти автоматический возврат на страницу мониторинга пациента. С данного момента настройки должны вступить в силу.

6.1.14. Требование 14

Проверка данного требования осуществляется аналогично проверке требования 13, за исключением того, что размер окна для вычисления индекса VRx в виде количества секунд необходимо ввести в текстовое поле с подписью «Размер временного окна для вычисления индекса VRx (в секундах)».

6.1.15. Требование 15

Для изменения размера временного буфера, который используется при вычислении относительных частот встречаемости разных классов значений индекса ауторегуляции PRx, необходимо, будучи авторизованным пользователем с ролью «Врач», нажать на кнопку «Настройки», находящуюся в разделе мониторинга выбранного пациента. После нажатия на данную кнопку должен произойти переход на страницу настроек данного пациента (рис. 13). Необходимо указать размер временного буфера в виде количества секунд, введя нужное значение в текстовое поле с подписью «Размер временного буфера для вычисления относительных частот встречаемости классов значений PRx (в секундах)», после чего – нажать на кнопку «Сохранить». Должен произойти автоматический возврат на страницу мониторинга данного пациента и настройки должны вступить в силу.

6.1.16. Требование 16

Проверка данного требования осуществляется аналогично проверке требования 15, за исключением того, что размер временного буфера, который используется при вычислении относительных частот встречаемости разных классов значений индекса ауторегуляции VRx, необходимо ввести в виде количества секунд текстовое поле с подписью «Размер временного буфера для вычисления относительных частот встречаемости классов значений VRx (в секундах)».

6.1.17. Требование 17

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Для переключения видимости групп показателей на странице мониторинга пациента, содержащих данные об относительной частоте встречаемости разных классов значений индексов PRx/VRx и графики ВЧД/ЦПД (рис. 14), необходимо, будучи авторизованным пользователем с ролью «Врач», изменить соответствующие настройки на странице настроек пациента (рис. 15).

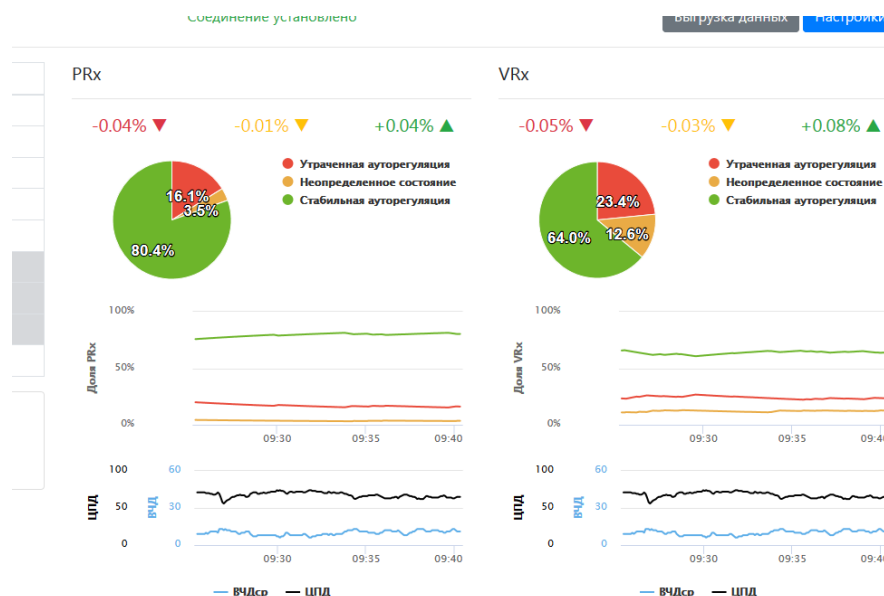


Рисунок 14 - группы элементов интерфейса на странице мониторинга пациента

Щербаков Борис Иванович
Настройки

Параметры вычисления индексов	Видимые показатели	Графики
<p>временного окна для вычисления а PRx (в секундах)</p> <input type="text"/>	<input checked="" type="checkbox"/> АДср2 <input checked="" type="checkbox"/> АДс2 <input checked="" type="checkbox"/> АДд2 <input checked="" type="checkbox"/> ЧСС <input checked="" type="checkbox"/> ВЧДср <input checked="" type="checkbox"/> SpO2 <input checked="" type="checkbox"/> etCO2 <input checked="" type="checkbox"/> ЧД <input checked="" type="checkbox"/> ЦПД	<p>Ширина оси X на графиках (в секундах)</p> <input type="text" value="900"/>
<p>временного буфера для вычисления тельных частот встречаемости классов ий PRx (в секундах)</p> <input type="text"/>	<input checked="" type="checkbox"/> Отображать относительные частоты встречаемости классов значений индекса PRx (а также графики ВЧД и ЦПД) <input type="checkbox"/> Отображать относительные частоты встречаемости классов значений индекса VRx (а также графики ВЧД и ЦПД)	
<p>временного окна для вычисления а VRx (в секундах)</p> <input type="text"/>		
<p>временного буфера для вычисления тельных частот встречаемости классов ий VRx (в секундах)</p> <input type="text"/>		

Рисунок 15 – переключатели видимости групп элементов интерфейса на странице мониторинга пациента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Затем необходимо нажать на кнопку «Сохранить». После нажатия на данную кнопку должен произойти автоматический возврат на страницу мониторинга пациента и настройки должны вступить в силу.

6.1.18. Требование 18

Для переключения видимости отдельных параметров (строк) в таблице необработанных значений, находящейся на странице мониторинга пациента, необходимо, будучи авторизованным пользователем с ролью «Врач» переключить видимость соответствующих параметров на странице настроек пациента, используя элементы интерфейса множественного выбора (рис. 16).

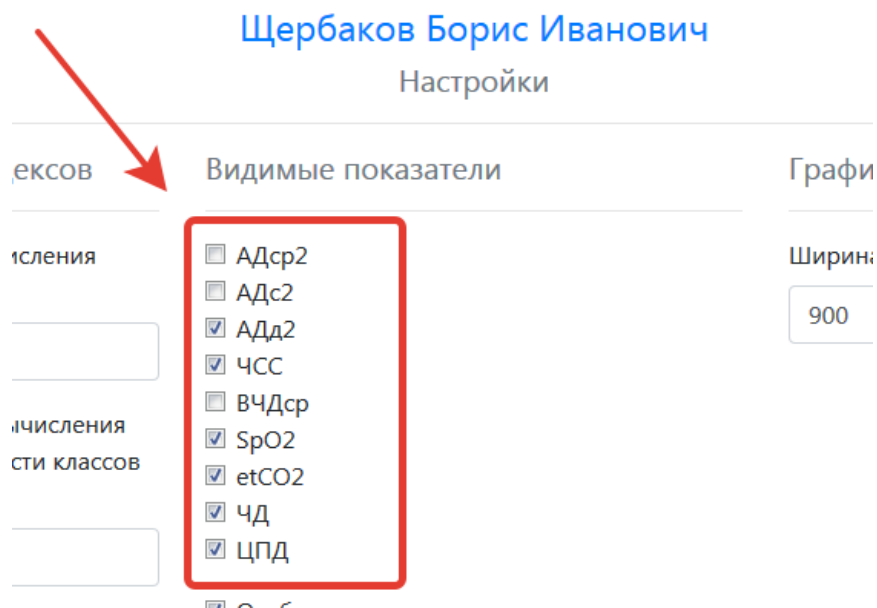


Рисунок 16 – переключатели видимости параметров таблицы необработанных значений

Затем необходимо нажать на кнопку «Сохранить». После нажатия на данную кнопку должен произойти автоматический возврат на страницу мониторинга пациента и настройки должны вступить в силу (рис. 17).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Параметр	Значение	Ед. изм.
Адд2	51	мм.рт.ст.
ЧСС	74	1/мин
SpO2	Нет данных	%
etCO2	Нет данных	U
ЧД	Нет данных	мм.рт.ст.
ЦПД	61	мм.рт.ст.

Рисунок 17 – таблица необработанных значений на странице мониторинга пациента

6.1.19. Требование 19

Для доступа в панель администрирования используется кнопка с надписью «Администрирование», расположенная в верхней части рабочей области программы (рис. 18).

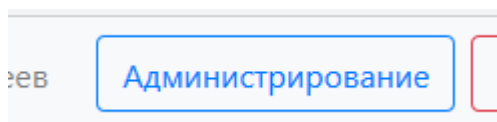


Рисунок 18 – кнопка перехода в панель администрирования

Данная кнопка должна отображаться только для пользователей с ролью «Администратор». После нажатия на данную кнопку должен произойти автоматический переход в панель администрирования (рис. 19).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

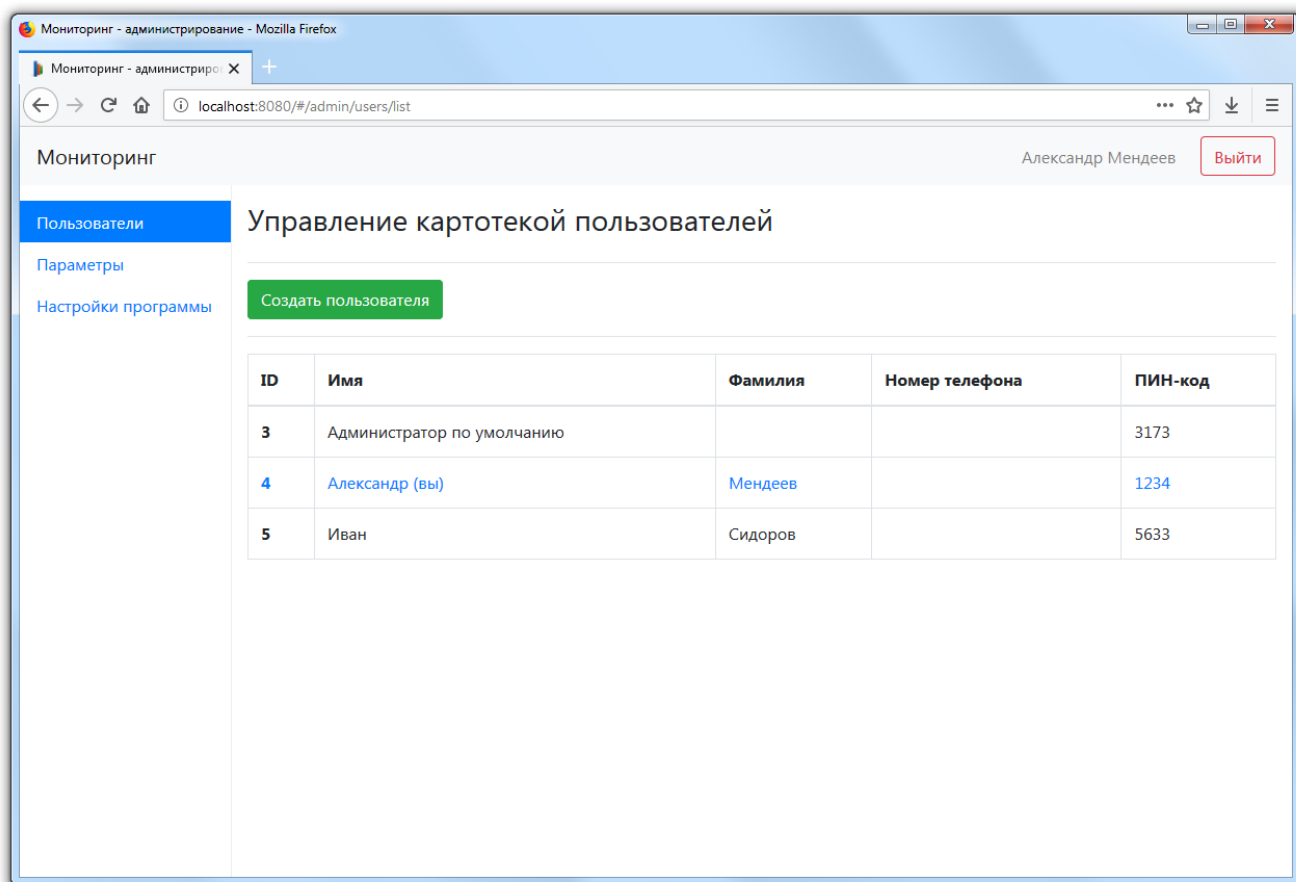


Рисунок 19 – панель администрирования

6.1.20. Требование 20

Для осуществления операций создания, просмотра, изменения и удаления над картотекой пользователей программы, необходимо, будучи авторизованным пользователем с ролью «Администратор», воспроизвести порядок действий, описанный в п. 3.5.6.1 руководства оператора.

При удалении пользователя или внесении изменений в его ПИН-код, любые попытки входа (аутентификации) в программу с использованием старых данных данного пользователя должны закончиться неудачно. При внесении изменений в список ролей пользователя, пользователь с измененным набором ролей должен приобрести или утратить возможность осуществления действий или доступа в соответствующие разделы программы.

При создании, изменении или удалении записей в картотеке пользователей программы, программа, в случае успешного осуществления данных действий, должна отображать соответствующие сообщения (рис. 20-22).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Запись создана

Запись пользователя с введёнными данными успешно сохранена в базе данных

Рисунок 20 – сообщение, уведомляющее пользователя об успешном создании записи в картотеке пользователей программы

Изменения сохранены

Введённые данные успешно сохранены

Рисунок 21 – сообщение, уведомляющее пользователя об успешном изменении записи в картотеке пользователей программы

Запись удалена

Выбранная запись успешно удалена из базы данных

Рисунок 22 – сообщение, уведомляющее пользователя об успешном удалении записи из картотеки пользователей программы

6.1.21. Требование 21

Для осуществления операций создания, просмотра, изменения и удаления над картотекой параметров, данные по которым будут запрашиваться из БД НИИ НДХиТ, необходимо, будучи авторизованным пользователем с ролью «Администратор», воспроизвести порядок действий, описанный в п. 3.5.6.2 руководства оператора.

При внесении изменений в картотеку параметров, изменения должны быть соответствующим образом отражены в таблице необработанных значений, находящейся на странице мониторинга пациента (рис. 17). То есть, при внесении изменений, соответствующие строки должны изменяться, при добавлении новых параметров – появляться, при удалении

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

параметров – исчезать из таблицы. Аналогичным образом изменения в картотеке параметров должны отражаться на списке доступных параметров в разделе настроек пациента (рис. 16).

При создании, изменении или удалении записей в картотеке параметров, программа, в случае успешного осуществления данных действий, должна отображать соответствующие сообщения (рис. 23-25).

Запись создана

Запись параметра с введёнными данными успешно сохранена в базе данных

Рисунок 23 – сообщение, уведомляющее пользователя об успешном создании записи в картотеке параметров

Изменения сохранены

Введённые данные успешно сохранены

Рисунок 24 – сообщение, уведомляющее пользователя об успешном изменении записи в картотеке параметров

Запись удалена

Выбранная запись успешно удалена из базы данных

Рисунок 25 – сообщение, уведомляющее пользователя об успешном удалении записи из картотеки параметров

6.2. Проверка выполнения требований к пользовательскому интерфейсу

Проверка выполнения требований к пользовательскому интерфейсу происходит одновременно с проверкой функциональных требований путём визуального ознакомления с пользовательским интерфейсом программы.

6.3. Проверка выполнения требований к программной документации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Проверка выполнения требований к программной документации происходит посредством визуального ознакомления с документацией и проверки её на соответствие описанным требованиям.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] URL: <http://fmi.asf.ru/library/book/Gost/19201-78.html> (дата обращения 11.05.2018);
- 2) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [Электронный ресурс] URL: http://www.rugost.com/index.php?option=com_content&view=article&id=53:19106-78&catid=19&Itemid=50 (дата обращения 11.05.2018).
- 3) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению [Электронный ресурс] URL: http://www.rugost.com/index.php?option=com_content&view=article&id=56:19301-79&catid=19&Itemid=50 (дата обращения 11.05.2018);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2**СПИСОК ИСПОЛЬЗОВАННЫХ ТЕРМИНОВ И СОКРАЩЕНИЙ**

Ниже приведен список терминов и сокращений, используемых в данном документе.

PRx – Pressure Reactivity Index

VRx – Variation Reactivity Index

СУБД – система управления базами данных

БД – база данных

НИИ НДХиТ – Научно-исследовательский институт неотложной детской хирургии и травматологии

Активный пациент – пациент, находящийся в данный момент на лечении в НИИ НДХиТ в отделении интенсивной терапии

АД – артериальное давление

ВЧД – внутричерепное давление

ЦПД – церебральное перфузионное давление

Аутентификация – процесс проверки подлинности введенных пользователем данных и установления личности этого пользователя

ПИН-код – цифробуквенное сочетание символов, которое используется для прохождения аутентификации и позволяет однозначно идентифицировать пользователя программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"**

Факультет компьютерных наук, департамент программной инженерии

СОГЛАСОВАНО

Кандидат технических наук, руководитель
отдела новых технологий НИИ неотложной
детской хирургии и травматологии

_____ С.Б. Арсеньев
" __ " _____ 2018 г.

УТВЕРЖДАЮ

Академический руководитель ОП
"Программная инженерия" Национального
исследовательского университета "Высшая
школа экономики"

_____ В.В. Шилов
" __ " _____ 2018 г.

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Текст программы

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.15-01 12 01-1-ЛУ

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Исполнитель

студент группы БПИ143

_____ /А.П. Мендеев /

« ____ » _____ 2018

УТВЕРЖДЕН

RU.17701729.04.15-01 12 01-1-ЛУ

**ПРОГРАММА МОНИТОРИНГА АУТОРЕГУЛЯЦИИ ГОЛОВНОГО МОЗГА У
ПАЦИЕНТОВ С ЧЕРЕПНО-МОЗГОВОЙ ТРАВМОЙ**

Текст программы

RU.17701729.04.15-01 12 01-1

Листов 5

<i>Инв. № подл</i>	
<i>Подп. и дата</i>	
<i>Взам. инв. №</i>	
<i>Инв. № дубл.</i>	
<i>Подп. и дата</i>	

СОДЕРЖАНИЕ

1. Текст программы.....	154
-------------------------	-----

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ТЕКСТ ПРОГРАММЫ

По причине большого объёма, исходный текст программы поставляется в электронном виде. Файлы, содержащие исходный код серверной и клиентской частей программы, располагаются в папке под названием «Исходный код программы» на флэш-накопителе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 12 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]