

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е.Н.Скуратенко  
подпись

«\_\_\_\_\_» \_\_\_\_\_ 2020г.

### **БАКАЛАВРСКАЯ РАБОТА**

09.03.03 – Прикладная информатика

Тема: Разработка системы учета уровня автоматизации и оснащенности  
информационно-техническим оборудованием медицинских учреждений  
Республики Хакасия

Руководитель \_\_\_\_\_ зав. кафедрой, доцент, к.т.н. Е. Н. Скуратенко  
подпись, дата

Выпускник \_\_\_\_\_ А. В.Петров  
подпись, дата

Консультанты по разделам:

Экономический \_\_\_\_\_ Е. Н. Скуратенко  
подпись, дата

Нормоконтролер \_\_\_\_\_ В. И. Кокова  
подпись, дата

Абакан 2020

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е.Н.Скуратенко  
подпись

«\_\_\_\_\_» \_\_\_\_\_ 2020г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Петрову Артему Васильевичу

Группа 56-1(ХБ 16-03)

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка системы учета уровня автоматизации и оснащенности информационно-техническим оборудованием медицинских учреждений Республики Хакасия

Утверждена приказом по институту № 216 от 06.04.2020 г.

Руководитель ВКР Е.Н. Скуратенко, зав. кафедрой, доцент, к.т.н., ХТИ – филиал СФУ

Исходные данные для ВКР: информация о текущем состоянии работ по созданию отчетов об уровне автоматизации и информатизации медицинских учреждений, технические характеристики разрабатываемого в ВКР программного продукта.

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки системы для учета уровня автоматизации и информатизации медицинских учреждений Республики Хакасия с возможностью создания отчетов.
3. Оценка экономической эффективности веб-приложения для учета уровня автоматизации и информатизации медицинских учреждений Республики Хакасия.

Перечень графического материала: нет

Руководитель ВКР

\_\_\_\_\_ Е. Н. Скуратенко  
подпись

Задание принял к исполнению

\_\_\_\_\_ А. В. Петров  
подпись

«06» апреля 2020 г.

## **РЕФЕРАТ**

Бакалаврская работа по теме «Разработка системы учета уровня автоматизации и оснащенности информационно-техническим оборудованием медицинских учреждений Республики Хакасия» содержит 53 страницы текстового документа, 30 формул, 8 таблиц, 17 рисунков, 20 использованных источников.

**ГКУЗ РХ «РМИАЦ», ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗЫ ДАННЫХ, IDEF, NODE.JS, VUE.JS, MONGODB, ПРЯМЫЕ ЗАТРАТЫ, ЭКСПЛУАТАЦИОННЫЕ ЗАТРАТЫ, КАПИТАЛЬНЫЕ ЗАТРАТЫ.**

Целью бакалаврской работы является сокращение времени на создание отчетов об уровне автоматизации и информатизации медицинских объектов с помощью разработки веб-приложения. Для достижения поставленной цели были решены следующие задачи:

- 1) осуществлен поиск информации об организационной характеристике предметной области;
- 2) построены модели IDEF0, IDEF3;
- 3) обоснован выбор средств проектных решений;
- 4) спроектирована документно-ориентированная файловая база данных, при помощи MongoDB;
- 5) разработано веб-приложение на языке программирования JavaScript для доступа к базе данных;
- 6) проведена оценка экономической эффективности.

В результате было разработано веб-приложение, которое позволяет сократить время, затрачиваемое на составление отчетов об уровне автоматизации и информатизации медицинских организаций Республики Хакасия.

## **SUMMARY**

The theme of the Bachelor's thesis is «Filing System Development of Automation Level and Availability of IT-equipment in Medical Institutions of the Republic of Khakassia». It comprises 53 pages, 30 formulae, 8 charts, 17 drawings, 20 references items.

**STATE PUBLIC HEALTHCARE INSTITUTION OF THE REPUBLIC OF KHAKASSIA «MEDICAL CENTER FOR INFORMATION AND ANALYSIS», IT SYSTEM, DATABASES, IDEF, NODE.JS, VUE.JS, MONGODB, DIRECT COSTS, OPERATING COSTS, CAPITAL COSTS.**

The purpose of the thesis is to reduce time spent on writing reports due to automation and IT support of medical facilities by developing a web application. To achieve this goal the following objectives have been introduced:

- 1) to compile information on the characteristics of the subject area;
- 2) to design IDEF0, IDEF3 models;
- 3) to give reasons for design solutions provided;
- 4) to design a document-oriented file database using MongoDB;
- 5) to develop a web application in the JavaScript programming language for accessing the database;
- 6) to present the project's economic efficiency.

A web application has been developed to reduce time spent on compiling reports due to automation and IT support of medical institutions in the Republic of Khakassia.

English language supervisor

---

(signature, date)

N.V. Chezybaeva

## **СОДЕРЖАНИЕ**

Введение .....	7
1   Анализ предметной области .....	9
1.1 Моделирование информационной системы.....	12
1.2 Выбор средств проектных решений .....	16
1.3 Выбор системы управления базами данных для разработки .....	19
1.4 Выводы по разделу “Анализ предметной области” .....	21
2   Разработка информационной системы .....	21
2.1 Подготовка среды разработки .....	21
2.2 Создание веб-приложения .....	24
2.3 Выводы по разделу “Разработка информационной системы” .....	33
3   Оценка экономической эффективности разработки информационной системы .....	33
3.1 Капитальные затраты .....	33
3.2 Эксплуатационные затраты .....	41
3.3 Расчет экономической эффективности ИС .....	44
3.4 Оценка риска при реализации информационной системы .....	47
3.5 Выводы по разделу “Оценка экономической эффективности разработки информационной системы” .....	49
Заключение .....	50
Список использованных источников .....	52

## **ВВЕДЕНИЕ**

Одной из приоритетных задач современной России является цифровизация процессов производственной и общественной сферы. Перспективное функционирование различных организаций, особенно государственных, связано с использованием существующего и созданием своего программного обеспечения(ПО) для обработки данных.

ГКУЗ РХ "РМИАЦ" осуществляет свою деятельность, связанную с формированием единой информационной системы здравоохранения Республики Хакасия путем организации на базе современных компьютерных технологий межотраслевого сбора, обработки, хранения и предоставления информации.

На данный момент в ГКУЗ РХ "РМИАЦ" нет информационной системы, которая позволила бы получить доступ к информации об автоматизации и информатизации подконтрольных медицинских организаций, составлять отчеты об уровне оснащенности этих организаций. Данная информационная система нужна для того, чтобы связать потоки данных в системах. Таким образом, актуальность выбранной темы связана с необходимостью разработки информационной системы для объединения данных в один поток, упрощения создания отчетности.

Целью проекта является сокращение времени обработки и передачи сведений об автоматизации и информатизации в медицинских организациях с помощью веб-приложения.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Осуществить поиск информации с целью определения технических характеристик разработки.
2. Построить модели процессов предметной области.
3. Обосновать выбор средств проектных решений.

4. Спроектировать базу данных для хранения сведений об автоматизации и информатизации медицинских организаций Республики Хакасия.

5. Разработать веб-приложение для доступа к спроектированной базе данных.

6. Оценить экономическую эффективность внедрения разработанной информационной системы.

## **1      Анализ предметной области**

Заказчиком разработки является Республиканский медицинский информационно-аналитический центр, он нуждается в системе для быстрого и удобного поиска информации об уровне автоматизации и информатизации объектов, так как эта информация требуется в однообразных отчетах по всем объектам, а предоставляется в разных формах и её необходимо получать из разных систем.

**Организационная характеристика ГКУЗ РХ "РМИАЦ".**

Полное наименование – Государственное казенное учреждение здравоохранения Республики Хакасия «Республиканский медицинский информационно-аналитический центр».

Краткое наименование – ГКУЗ РХ "РМИАЦ". Юридический адрес учреждения: ул.Крылова 72, Абакан, Республика Хакасия, 655003. Контактный телефон: 8(390)229-50-13.

Учреждение выполняет следующие основные виды деятельности:

- Разработка концепций и программ информационной системы здравоохранения Республики Хакасия.
- Координация работ по созданию единой информационной системы здравоохранения Республики Хакасия.
- Формирование единой системы учета и отчетности медико-статистической информации с применением новых технологий её обработки.
- Прием и обработка статистических отчетов от учреждений здравоохранения.
- Разработка показателей, характеризующих деятельность учреждений здравоохранения Республики Хакасия, состояние здравоохранения в рамках утвержденной статистической отчетности.
- Разработка, внедрение и сопровождение автоматизированных систем сбора, хранения и передачи информации [1].

- Анализ полученной информации с привлечением главных штатных и внештатных специалистов Министерства здравоохранения Республики, организационно-методических отделов Республики, больниц и диспансеров.

- Осуществление взаимодействия с территориальным фондом обязательного медицинского страхования, территориальными органами государственной статистики, образовательными и научными учреждениями и другими сторонними организациями.

- Осуществление контроля над использованием в работе учреждений здравоохранения международных классификаций при ведении медицинской документации.

- Выработка управленческих решений по повышению эффективности деятельности учреждений здравоохранения.

- Анализ медико-статистической информации о состоянии здоровья населения и состояния здравоохранения Республики Хакасия [1].

Одним из приоритетных направлений работы ГКУЗ РХ "РМИАЦ" является формирование единой информационной системы здравоохранения Республики Хакасия путем организации на базе современных компьютерных технологий межотраслевой системы сбора, обработки, хранения и предоставления информации, обеспечивающей динамическую оценку здоровья и информационную поддержку принятия решений, направленных на его улучшение.

Министерство здравоохранения регулирует работу ГКУЗ РХ "РМИАЦ", которое состоит из таких отделов как: отдел медицинской статистики и мониторинга, отдел автоматизированных систем управления, технический отдел, отдел адресно-справочной работы (Рисунок 1).

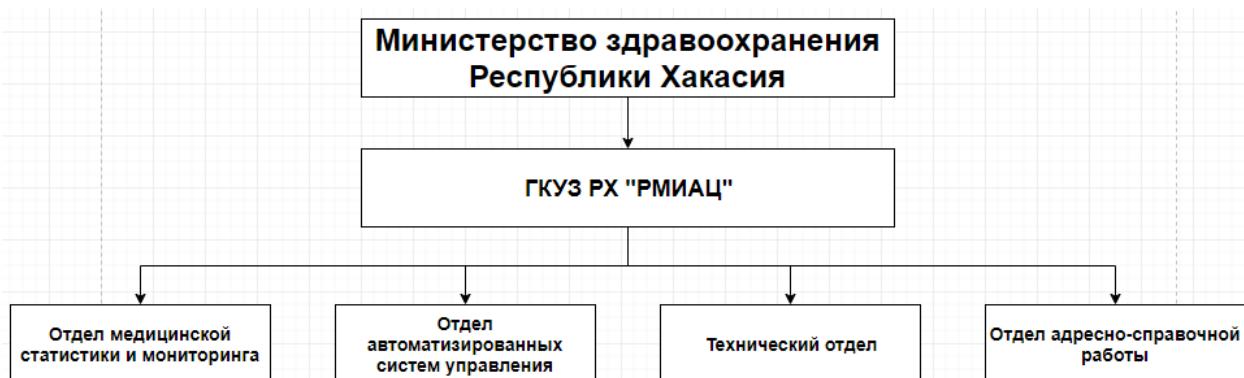


Рисунок 1 –Структурная схема ГКУЗ РХ "РМИАЦ"

На данный момент в ГКУЗ РХ "РМИАЦ" функционирует автоматизированный обмен с медицинскими организациями через e-mail и облачное хранилище.

Сопровождением программного обеспечения занимается технический отдел.

Отдел автоматизированных систем управления занимается формированием отчетности об уровне автоматизации и информатизации медицинских учреждений.

Для быстрого и удобного поиска и анализа актуальной информации в центре используются современные программные решения для мониторинга. Основные информационные системы, используемые для формирования информационной базы по медицинским учреждениям Республики Хакасия:

- региональная информационно-аналитическая медицинская система (РИАМС) "ПроМед";
- единая государственная информационная система в сфере здравоохранения Республики Хакасия (ЕГИСЗ РХ).

Региональная информационно-аналитическая медицинская система (РИАМС) "ПроМед" – это специализированный программный комплекс, позволяющий автоматизировать процессы сбора, обработки и хранения медицинской, экономической и статистической информации в системе здравоохранения. РИАМС "ПроМед" функционирует на едином центре

обработки данных для неограниченного числа пользователей. В центре обработки данных консолидируется вся информация, связанная с персональным учетом оказанной медицинской помощи и управлением ресурсами здравоохранения региона. "ПроМед" обеспечивает информационный обмен между медицинскими учреждениями, органами управления здравоохранением.

Единая государственная информационная система в сфере здравоохранения (ЕГИСЗ) представляет собой совокупность информационно-технологических и технических средств, обеспечивающих информационную поддержку методического и организационного обеспечения деятельности участников системы здравоохранения. Региональный фрагмент единой государственной информационной системы здравоохранения Республики Хакасия является частью программы модернизации здравоохранения.

Главной функцией разрабатываемой информационной системы предполагается предоставление справочной информации об объектах медицинских учреждений Республики Хакасия, на тему их уровня автоматизации и информатизации, упрощение составления отчетов. На данный момент информация в неполном виде хранится в системе "ПроМед" и ЕГИСЗ, для формирования отчета необходимо брать информацию о разных подразделениях одного медицинского учреждения из разных систем и объединять в один документ.

## **1.1 Моделирование информационной системы**

Моделирование информационной системы основывалось на методологии функционального моделирования IDEF0.

Так как данная система создается с целью упростить и ускорить существующий процесс был проведен анализ процесса получения информации и выведения её в отчет до внедрения системы (Рисунок 2). В системе РМИАЦ находятся 43 медицинских учреждения, у каждого из

которых в ответственности находится несколько объектов. Необходимо получить информацию об уровне информатизации и автоматизации каждого объекта.

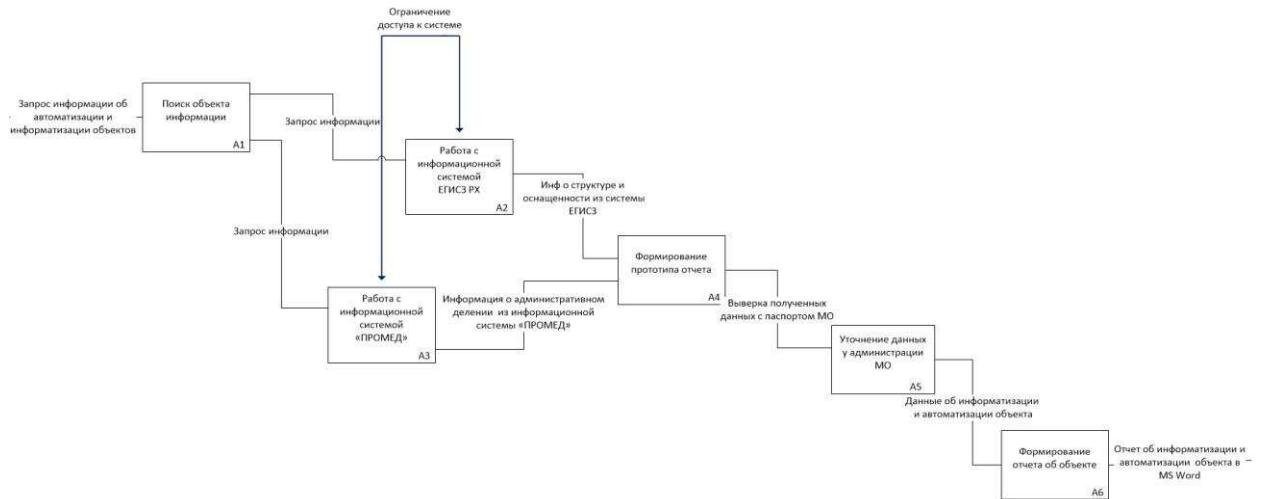


Рисунок 2 – Функциональная модель проекта до внедрения системы

На вход поступает запрос информации об автоматизации и информатизации определенного объекта. Например, о здании в таблице будут храниться данные о типе здания, наличии персональных компьютеров, сети, количестве и типе отделений, где постройки, а также о юридических лицах, связанных с этим зданием. Для получения данной информации сотрудник ГКУЗ РХ "РМИАЦ" обращается в две информационные системы, в информационной системе "ПроМед" можно получить информацию об административном делении, а в системе ЕГИСЗ находится информация о структуре и оснащенности объекта. Далее он формирует прототип отчета об объекте, проводит выверку полученных данных с паспортом МО. На следующем этапе уточняет данные у администрации МО. Далее из данных формируется отчетная форма и предоставляется в электронном или бумажном виде.

Исходя из этой схемы очевидно наличие этапов, таких как работа с информационными системами "ПроМед" и ЕГИСЗ, а также уточнение данных у администрации медицинского учреждения, от которых можно

избавиться созданием общей базы данных о медицинских объектах. Специфика веб-приложения позволит медицинским организациям удаленно заполнять и получать информацию (Рисунок 3).

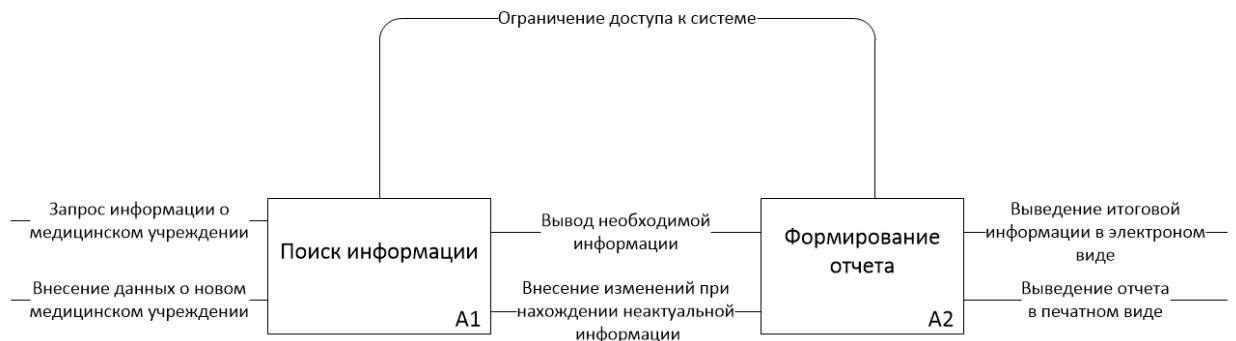


Рисунок 3 – Функциональная модель проекта после внедрения системы

Основанием для проведения разработки является проведенное исследование процесса создания отчета до внедрения системы, текущее состояние системы предполагает подключение к двум информационным системам (ЕГИСЗ РХ и РИАМС "ПроМед") и последующую выверку с администрацией МО, а также ручное составление отчета при помощи MS Word, которые являются избыточными шагами, приложение позволит упростить эти процессы. При работе в разработанном приложении имеется возможность удаленного доступа в систему, например, для внесения актуальной информации от специалиста, находящегося в медицинском учреждении.

Методология IDEF3 является одним из стандартов семейства IDEF и довольно широко используется при декомпозиции моделей IDEF0 для моделирования процессов более низкого уровня, поскольку с его помощью можно смоделировать технологические процессы, происходящие на предприятии, т.е. описать возможные сценарии реализации процессов, в рамках которых происходит последовательное изменение свойств объекта. Данная методология позволяет показывать возможные разветвления в процессе. Например, когда результат одного действия может инициировать

запуск нескольких действий или наоборот, чтобы начать какое-то действие, необходимо завершить несколько предыдущих действий, на рисунке 4 изображена модель работы в информационной системе.

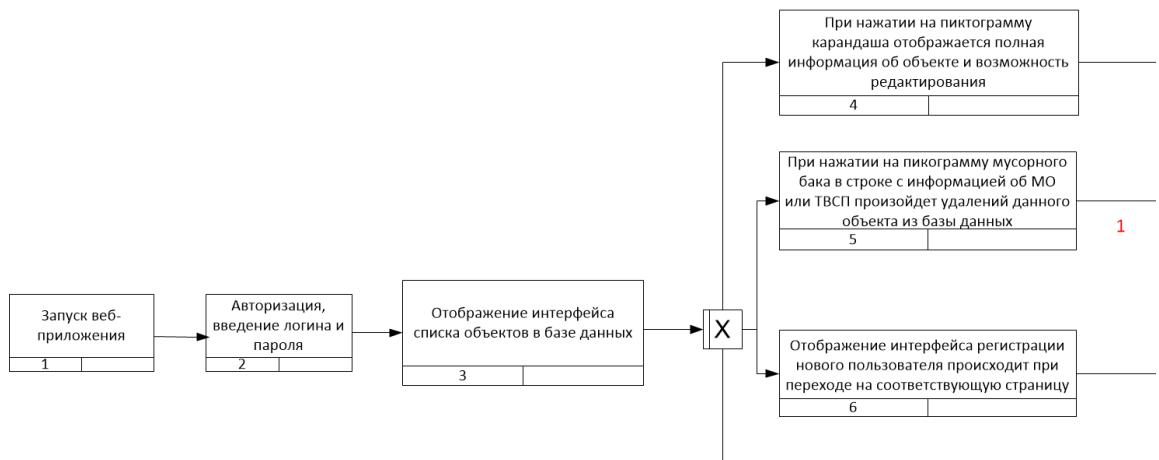


Рисунок 4 – Модель работы системы, лист 1

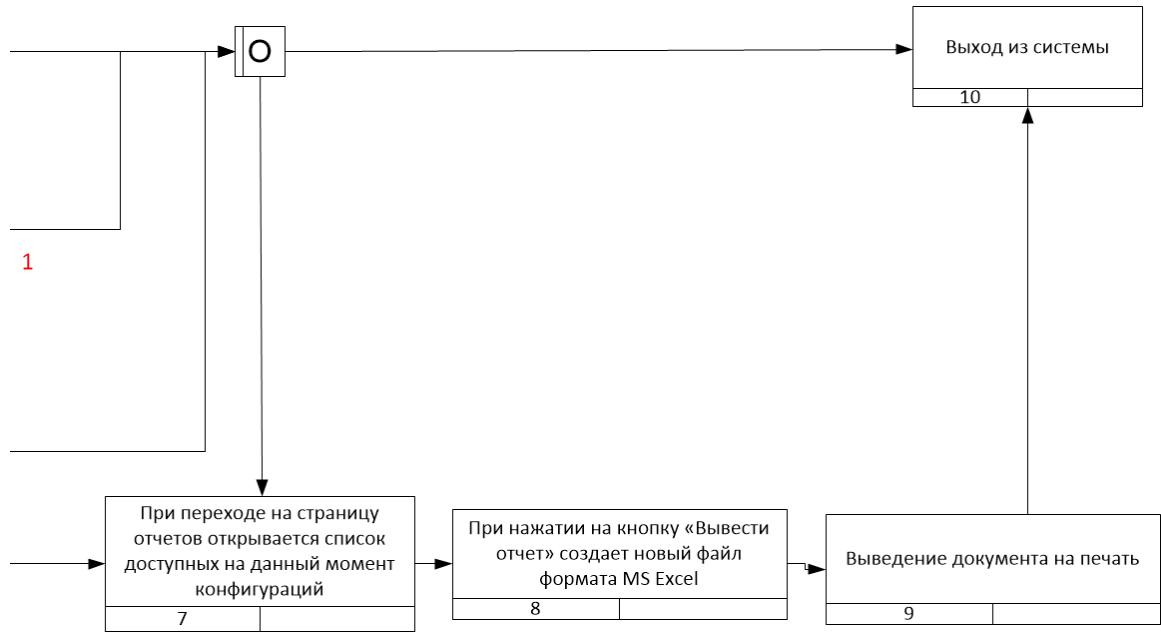


Рисунок 4, лист 2

Схема IDEF3 показывает процессы внутри системы с момента входа в систему до выхода из нее. После запуска приложения пользователю необходимо внести свой логин и пароль, используя базу данных пользователей происходит определение прав данного пользователя. Интерфейс приложения представляет страницу с переключением между таблицами подразделений и организаций, в каждой строке в крайней правой колонке в виде изображений указаны действия с этим объектом, изображение карандаша при нажатии переносит на страницу редактирования данного объекта, а изображение мусорного бака удаляет данный объект. Создание отчетов представляет формирование файла формата `xlsx` по заданным конфигурациям, в зависимости от данных в базе.

## 1.2 Выбор средств проектных решений

Для разработки веб-приложений используются языки программирования, которые в основном предназначены для работы с веб-технологиями. Языки программирования для веб-приложений делятся на клиентские и серверные. Важная сторона работы серверных языков – возможность организации непосредственного взаимодействия с системой управления базами данных. Клиентская часть приложения обрабатывается на стороне пользователя, как правило в браузере.

PHP – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML [2].

Преимущества PHP:

- поддержка многих систем управления базами данных (MySQL, PostgreSQL, Sybase, Infomix и других);
- доступность для большинства операционных систем (Linux, Unix, Microsoft Windows, Mac OS и других);
- большое количество библиотек и расширений;

- прост в освоении.

Недостатки PHP:

- не подходит для создания десктопных приложений или системных компонентов;
- веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью.

NodeJS – программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера.

Преимущества языка программирования JavaScript с использованием Node.js:

- возможность работы, как с клиентской стороны, так и с серверной стороны;
- очень высокая производительность;
- строенная система управления модулями зависимости;
- прост в освоении.

Недостатки языка программирования JavaScript с использованием Node.js:

- асинхронность кода может сильно его усложнить.

Обе технологии имеют хороший выбор редакторов, интегрированных сред разработки, отладчиков, валидаторов и других инструментов. Node.js имеет прекрасный инструмент, npm – менеджер пакетов, с его помощью можно управлять модулями и зависимостями. У PHP есть свой менеджер пакетов, разработанный под влиянием npm – Composer. Однако, если npm встроен по умолчанию, то Composer придется встраивать самостоятельно.

Веб-разработчикам часто нужно создавать приложения, которые относятся не только к web, например, разработка онлайн-сервиса, сценарии преобразования данных и т.д. На PHP можно разрабатывать десктопные приложения или консольные утилиты, но в основном PHP нужен на стороне сервера и редко выходит за пределы этой границы. Несколько лет назад, JavaScript использовался исключительно для браузера. С приходом Node.js появилась возможность писать десктопные и мобильные приложения, а также можно программировать и микроконтроллере. Node.js расширил границы JavaScript.

Express.js, или просто Express, фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js. Автор фреймворка описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом VueJS, React или AngularJS для frontend'a.

Немаловажным аргументом в сторону выбора Node.js являлось то, что он удобен для специалистов из ГКУЗ РХ "РМИАЦ", так как уже имеются приложения, созданные при помощи этой технологии и предполагается внесение изменений в программную часть приложения после реализации.

Исходя из этого, для реализации проекта был выбран Node.js, как серверный язык и встроенные в него библиотеки.

Для разработки клиентской части приложения было проведено сравнение фреймворков на JavaScript, которое представлено в таблице 1.

Таблица 1 – Сравнение фреймворков для клиентской части

Название	Размер файла библиотеки	Время загрузки	Уровень сложности	Гибкость использования	Скорость разработки
React	100 Кб	Среднее	Средний	Высокая	Средний
Angular	500 Кб	Высокое	Высокий	Средняя	Низкая
Vue	50 Кб	Низкое	Низкий	Низкая	Высокая

Исходя из данного сравнения, Angular не подходит для данной работы ввиду низкой скорости разработки и высокого порога входления для разработчика. React превосходит Vue в плане создания сложных интерфейсов, но для решения относительно простых задач использование Vue выглядит не только оправданным, но и предпочтительным.

### 1.3 Выбор системы управления базами данных для разработки

Система управления базами данных (СУБД) – это общий термин, относящийся ко всем видам абсолютно разных инструментов, от компьютерных программ до встроенных библиотек. Эти приложения управляют или помогают управлять наборами данных. Так как эти данные могут быть разного формата и размера, были созданы разные виды СУБД.

СУБД основаны на моделях баз данных – определённых структурах для обработки данных. Каждая СУБД создана для работы с одной из них с учётом особенностей операций над информацией.

Реляционные системы управления базами данных берут своё название от реализуемой модели – реляционной. Сейчас они остаются, да и ещё какое-то время будут, самым популярным выбором для надёжного, безопасного и производительного хранения данных. Схемы очень похожи на таблицы, столбцы которых отражают порядковый номер и тип информации в каждой записи, а строки — содержимое этих записей.

NoSQL – СУБД не используют реляционную модель структуризации данных. Существует много реализаций, решающих этот вопрос по-своему, зачастую весьма специфично. Эти бессхемные решения допускают неограниченное формирование записей и хранение данных в виде ключ-значение.

В отличие от традиционных РСУБД, некоторые базы данных NoSQL, например, MongoDB, позволяют группировать коллекции данных с другими базами данных. Такие СУБД хранят данные как одно целое. Эти данные могут представлять собой одиночный объект наподобие JSON и вместе с тем корректно отвечать на запросы к полям.

NoSQL базы данных и СУБД не подразумевают внутренних связей. Они не основываются на одной модели, каждая база данных в зависимости от целей использует различные модели. Добавление и удаление полей в базе данных, а также связи между таблицами устанавливаются на программном уровне, такой подход удобен для сотрудников ГКУЗ РХ "РМИАЦ", так как предполагается внесение изменений после внедрения информационной системы.

Существует несколько различных моделей и функциональных систем для NoSQL баз данных:

- хранилище ключ-значение;
- распределенное хранилище;
- документо-ориентированные;
- БД на основе графов.

Для данной работы необходима документо-ориентированная база данных, так как будет создана сложная структура данных по медицинскому учреждению, которая будет помещена в один документ и отдельная схема для его подразделений.

MongoDB – документо-ориентированная СУБД с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как

NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++.

## **1.4 Выводы по разделу “Анализ предметной области”**

В данном разделе проанализирована основная деятельность учреждения ГКУЗ РХ "РМИАЦ", которая позволила выявить необходимость веб-приложения, ускоряющего работу по поиску информации об автоматизации и информатизации медицинских учреждений и формированию отчетов.

Для создания веб-приложения были выбраны наиболее популярные средства в среде современных IT-разработчиков:

- В качестве программного решения был выбран Node.js.
- СУБД–MongoDB.

## **2 Разработка информационной системы**

### **2.1 Подготовка среды разработки**

Для создания информационной системы была подготовлена среда разработки, т.е. установлено и настроено программное обеспечение(ПО). Visual Studio Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как редактор кода для кроссплатформенной разработки веб-и облачных приложений [3].

Официальный сайт nodejs.org предоставляет возможность скачать инсталляционный файл, включающий в себя компоненты Node. Далее в терминале, который можно запускать напрямую из Visual Studio Code установлен npm (node.js package manager). С помощью npm можно устанавливать пакеты, которые отображаются в файле package.json и хранятся в каталоге node\_modules.

MongoDB для Windows предоставляется разработчиком в виде архива, содержимое которого представлено на рисунке 6.

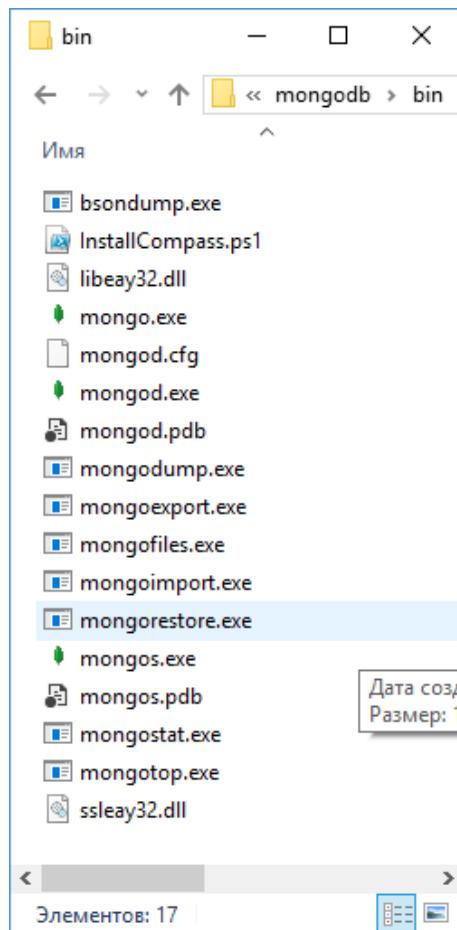


Рисунок 5 – Содержимое архива MongoDB

Рассмотрим функцию каждого файла из архива:

- Bsondump – считывает содержимое BSON-файлов и преобразует их в читабельный формат, например, в JSON;
- mongo – представляет консольный интерфейс для взаимодействия с базами данных, своего рода консольный клиент;
- mongod – сервер баз данных MongoDB. Он обрабатывает запросы, управляет форматом данных и выполняет различные операции в фоновом режиме по управлению базами данных;
- mongodump – утилита создания бэкапа баз данных;

- mongoexport – утилита для экспорта данных в форматы JSON, TSV или CSV;
- mongofiles – утилита, позволяющая управлять файлами в системе GridFS;
- mongoimport – утилита, импортирующая данные в форматах JSON, TSV или CSV в базу данных MongoDB;
- mongorestore – позволяет записывать данные из дампа, созданного – mongodump, в новую или существующую базу данных;
- mongos – служба маршрутизации MongoDB, которая помогает обрабатывать запросы и определять местоположение данных в кластере MongoDB;
- mongorestat – представляет счетчики операций с базой данных;
- mongotop – предоставляет способ подсчета времени, затраченного на операции чтения-записи в базу данных.

В ОС Windows по умолчанию MongoDB хранит базы данных по пути C:\data\db. Для запуска сервера необходимо запустить файл mongod.exe. Как он выглядит во время запуска, демонстрируется на рисунке 7.

```
C:\Windows\system32\cmd.exe - mongod
c data capture with directory 'C:/data/db/diagnostic.data'
2020-04-17T16:59:48.084+0700 I SHARDING [LogicalSessionCacheRefresh] Marking collection config.system.sessions as collection version: <unsharded>
2020-04-17T16:59:48.084+0700 I SHARDING [LogicalSessionCacheReap] Marking collection config.transactions as collection version: <unsharded>
2020-04-17T16:59:48.086+0700 I NETWORK [listener] Listening on 127.0.0.1
2020-04-17T16:59:48.086+0700 I NETWORK [listener] waiting for connections on port 27017

2020-04-17T16:59:49.008+0700 I SHARDING [ftdc] Marking collection local.oplog.rs as collection version: <unsharded>
```

Рисунок 6 – Запуск сервера MongoDB

Командная строка отображает ряд служебной информации, например, что сервер запускается на localhost на порту 27017.

Таким образом, среда разработки представлена Node.js внутри редактора Visual Studio Code и сервером MongoDB.

## 2.2 Создание веб-приложения

Разработка информационной системы была разделена на следующие этапы:

- формирование базы данных для хранения информации;
- создание серверной части приложения с функционалом добавления, редактирования и удаления записей в базе данных;
- создание клиентской части веб-приложения.

База данных включает два основных объекта: медицинское учреждение и территориально-выделенное структурное подразделение (ТВСП), а также модель пользователей, для ограничения доступа к системе. Медицинское учреждение включает в себя информацию о его основных сведениях и информационных системах, ТВСП имеет следующие данные: основные сведения, АРМ, сетевая инфраструктура, диагностическое оборудование. NodeJS позволяет всю структуру базы данных прописывать внутри программы, все недостающие элементы в виде полей и таблиц будут создаваться автоматически при помощи библиотеки mongoose.

Серверная часть представляет механизм обработки запросов из клиентской части к базе данных. Для работы с объектами базы данных потребуется 5 функций: создание, редактирование, удаление, выведение на экран всех объектов одного типа, поиск и выведение на экран определённого объекта.

Клиентская часть веб-приложения построена на фреймворке Vue.js и отвечает за отображение действий пользователя, обращаясь к серверу позволяет взаимодействовать с базой данных.

Формирование базы данных при использовании MongoDB моделей, в данном случае создается 3 модели: медицинские учреждения, ТВСП и

пользователи. Код для создания модели пользователя представлен на рисунке 7.

```
const mongoose = require('mongoose'),
bcrypt = require('bcrypt');

const Schema = mongoose.Schema({
  username: {
    type: String,
    unique: true,
    required: true
  },
  password: {
    type: String,
    required: true
  }
});
Schema.pre('save', function (next) {
  const user = this;
  if (this.isModified('password') || this.isNew) {
    bcrypt.genSalt(10, (error, salt) => {
      if (error) return next(error);
      bcrypt.hash(user.password, salt, (error, hash) => {
        if (error) return next(error);
        user.password = hash;
        next();
      });
    });
  } else {
    return next();
  }
});
Schema.methods.comparePassword = function (password, callback) {
  bcrypt.compare(password, this.password, (error, matches) => {
    if (error) return callback(error);
    callback(null, matches);
  });
};
mongoose.model('User', Schema);
```

Рисунок 7 – Создание модели пользователя

После создания модели необходимо создать api(application programming interface) – описание процедур, при помощи которых взаимодействует клиентская и серверная часть. Процедурами для модели пользователей будут являться авторизация и регистрация. Код файла user.js, находящегося в папке api, представлен на рисунке 8.

```

const mongoose = require('mongoose');

const api = {};

api.signup = (User) => (req, res) => {
  if (!req.body.username || !req.body.password) res.json({ success: false, message: 'Пожалуйста, введите логин и пароль.' });
  else {
    const user = new User({
      username: req.body.username,
      password: req.body.password
    });

    user.save(error => {
      if (error) return res.status(400).json({ success: false, message: 'Данный логин не может быть использован.' });
      res.json({ success: true, message: 'Пользователь успешно зарегистрирован.' });
    });
  }
}

module.exports = api;

```

Рисунок 8 – Процедуры модели пользователей

Заключительным этапом на пути от сервера к клиенту является настройка адресов, по которым необходимо переходить, чтобы задействовать ту или иную функцию, файл отвечающий за роутинг модели пользователей представлен на рисунке 9.

```

const models = ...require('@DiplomManager/app/setup');

module.exports = (app) => {
  const api = app.API.app.api.user;

  app.route('/api/v1/signup')
    .post(api.signup(models.User));
}

```

Рисунок 9 – Настройка роутинга модели пользователей

Создание клиентской части начинается с создания папки и установки в неё фреймворка Vue.js при помощи команд:

```

npm i --g vue-cli;
vue init webpack;

```

Данные команды инициализируют создание структуры папок стандартного набора фреймворка, создают стартовую страницу с

информацией о нем, после очистки от лишних созданных таким образом файлом структура клиентской части выглядит так, как показано на рисунке 10.

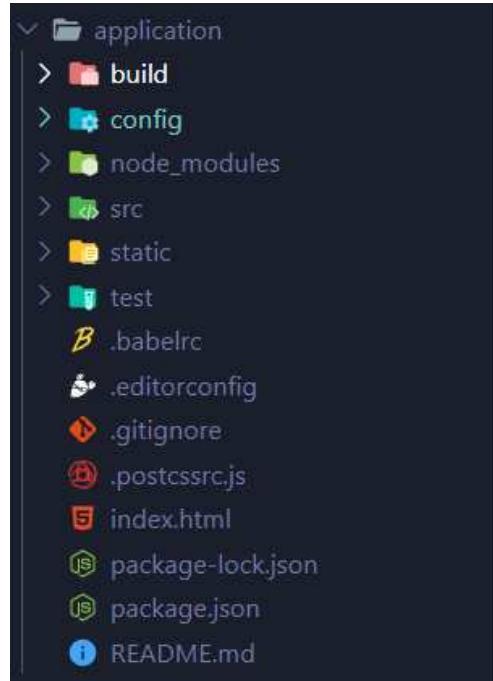


Рисунок 10 – Стандартная структура папок фреймворка Vue.js

Из элементов фронтенда приложения удобно показать создание страницы авторизации, для этого были созданы 2 файла в папке src/pages/Authentication, первый, а именно Authentication.vue будет отвечать за визуальное представление страницы при помощи библиотеки vuety, отвечающей за расположение элементом страницы и scss, описывающим стили данных элементов, фрагмент кода структуры объектов на странице из файла представлен на рисунке 11.

```

<template>
  <div class="l-auth-container">
    <div class="l-auth">
      <v-form v-model="validLogin">
        <v-text-field label="Логин"
          v-model="credentials.username"
          prepend-icon="account_box"
          :rules="rules"
          required
          color="light-blue lighten-1">
        </v-text-field>

        <v-text-field label="Пароль"
          v-model="credentials.password"
          prepend-icon="lock"
          :rules="rules"
          :append-icon="loginPasswordVisible ? 'visibility' : 'visibility_off'"
          :append-icon-cb={() => (loginPasswordVisible = !loginPasswordVisible)}
          :type="loginPasswordVisible ? 'text' : 'password'"
          color="light-blue lighten-1"
          required>
        </v-text-field>

        <v-btn color="Light-blue lighten-1" @click.native="submitAuthentication()">Войти</v-btn>
      </v-form>
    </div>

```

Рисунок 11 – Фрагмента кода из файла Authentication.vue

Вторым файлом в папке Authentication является index.js, в котором хранятся функции для проверки валидности введенных данных, помимо этого функция авторизации записывает токен пользователя в cookie браузера на срок в один день, это помогает избежать повторной авторизации при закрытии страницы с приложением, фрагмент кода из этого файла представлен на рисунке 12.

```

import Axios from 'axios'
import router from '@/router'
const API = `http://${window.location.hostname}:3001`

export default [
  user: { authenticated: false },
  authenticate (context, credentials, redirect) {
    Axios.post(`${API}/api/v1/auth`, credentials)
      .then(({data}) => {
        context.$cookie.set('token', data.token, '1D')
        context.$cookie.set('user_id', data.user._id, '1D')
        context.validLogin = true

        this.user.authenticated = true

        if (redirect) router.push(redirect)
      }).catch(({response: {data}}) => {
        context.snackbar = true
        context.message = data.message
      })
  },
]

```

Рисунок 12 – Фрагмент кода из файла Athentication/index.js

Также необходимо внести страницу авторизации в файл index.js из папки src/router, в нем хранится информация об адресе домашней страницы, дополнительные компоненты для страниц, а также защита от перехода на нее неавторизованных пользователей, при переходе на главную страницу, пользователь токен которого не записан в cookie переносится на страницу авторизации. Код данного файла представлен на рисунке 13.

```
import Home from '@/components/pages/Home'
import Authentication from '@/components/pages/Authentication/Authentication'

// Global components
import Header from '@/components/Header'
import List from '@/components/List/List'
import Create from '@/components/pages/Create'

// Register components
Vue.component('app-header', Header)
Vue.component('list', List)
Vue.component('create', Create)

Vue.use(Router)

const router = new Router({
  routes: [
    {
      path: '/',
      name: 'Home',
      meta: {
        requiredAuth: true
      },
      components: {
        default: Home,
        header: Header,
        list: List,
        create: Create
      }
    },
    {
      path: '/login',
      name: 'Authentication',
      component: Authentication
    }
  ]
})
```

Рисунок 13 – Фрагмент кода из файла router/index.js

Страница авторизации представляет собой блоки для введения логина и пароля, а также кнопку «Войти» для вызова функции проверки валидности введенной информации, страница авторизации представлена на рисунке 14.

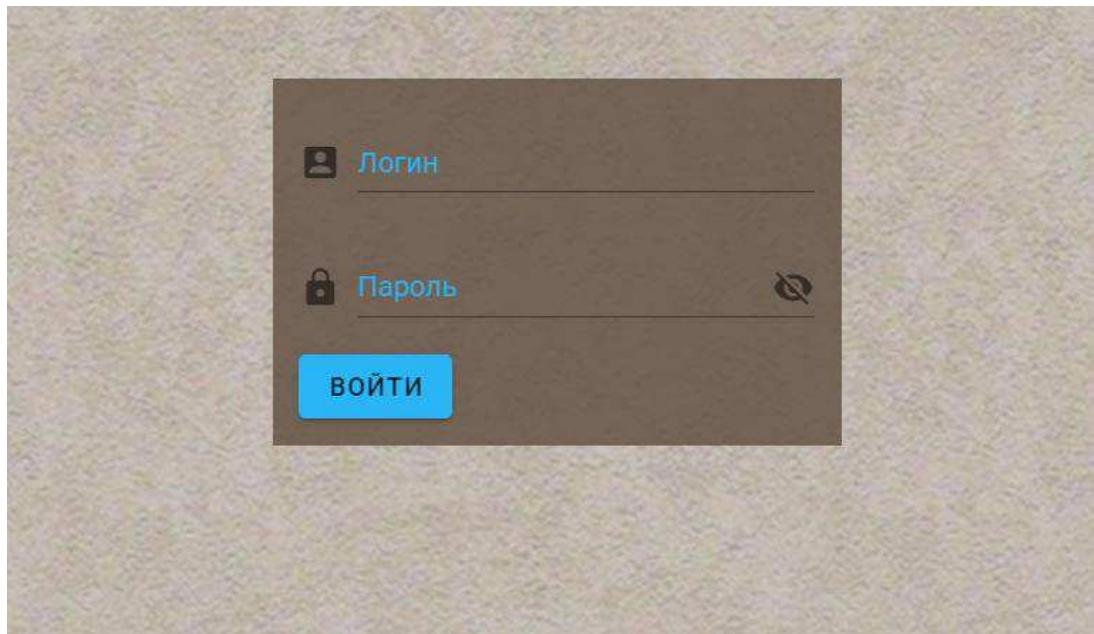


Рисунок 14 – Страница авторизации

После прохождения авторизации пользователь попадает на основную страницу приложения, на которой отображается таблица с данными. В верхней части страницы находится 4 объекта: поле поиска по столбцу “Медицинская организация”, фильтр организаций по уровню оказания медицинских услуг, кнопка переключения между отображением подразделений и организаций, а также кнопка выхода из системы. В правом нижнем углу находится кнопка с символом «+», при нажатии на нее появляется 6 объектов, направляющих пользователя на разные страницы (снизу-вверх): создание нового пользователя, создание нового ТВСП, создание новой МО, отображение таблицы ТВСП, отображение таблицы МО, страница создания отчетов. Основная страница приложения показана на рисунке 15.

Медицинская организация	ТВСП	Адрес	АРМ	Оператор	Скорость Мбит/с	Технология подключения	Действия
ГБУЗ РХЧ Черногорская межрайонная больница"	Паталогонавтоматическое отделение	Черногорск, Клубный 26	1	ООО Мегабит	1000	Оптика	
ГБУЗ РХЧ Черногорская межрайонная больница"	Здание хирургии	Черногорск, Мира 13 в.	54	ООО Мегабит	1000	Оптика	
ГБУЗ РХЧ Черногорская межрайонная больница"	Педиатрический кабинет	Пригорск, б	1	ООО Мегабит	1000	Оптика	
ГБУЗ РХЧ Черногорская межрайонная больница"	Стационар	Черногорск, Угольная 26 а	55	ООО Мегабит	1000	Оптика	
ГБУЗ РХЧ Черногорская межрайонная больница"	Поликлиника	Черногорск, Космонавтов 21	107	ООО Мегабит	1000	Оптика	
ГБУЗ РХЧ "Ибазинская городская больница"	Административный корпус	Абаза, Ленина 16 е	0	Ростелеком	100	Оптика	
ГБУЗ РХЧ "Абазинская городская больница"	Поликлиника	Абаза, Осиненко 17	63	Ростелеком	100	Оптика	
ГБУЗ РХЧ "Абазинская городская больница"	Туберкулезное отделение	Абаза, Больничная 9 г	1	Ростелеком	1	Оптика	
ГБУЗ РХЧ "Абазинская городская больница"	Хозяйственный корпус	Абаза, Осиненко 17	2	Ростелеком	1	Высокая пара	
ГБУЗ РХЧ "Абазинская городская больница"	Филиал поликлиники	Абаза, Энтузиастов 11	2	Ростелеком	100	Оптика	
Общее количество записей: 10 из 10							

Рисунок 15 – Основная страница приложения

Для создания и редактирования объектов базы данных используются разные макеты страницы, для регулирования отображения необходимого макета используется файл src/page/Create.vue. Структура файла представляет собой условное перенаправление, используется несколько переменных типа boolean и из сочетания этих переменных при нажатии на кнопку происходит перенаправление в файл, отвечающий за эту функцию, фрагмент кода из этого файла представлен на рисунке 16.

```

<template>
  <div class="l-create-page">
    <budget-creation v-if="budgetCreation && editPage && userCreation" slot="budget-creation" :clients="clients" :saveBudget="saveBudget"></budget-creation>
    <client-creation v-if="!budgetCreation && editPage && !usercreation" slot="client-creation" :saveClient="saveClient"></client-creation>
    <user-creation v-if="userCreation && editPage && !budgetCreation" slot="user-creation" :submitSignup="submitSignup"></user-creation>
    <reports-creation v-if="userCreation && editPage && budgetCreation" slot="reports-creation" :clients="clients" :processClient="processClient"></reports-creation>

    <budget-edit v-else-if="budgetEdit && editPage"
      slot="budget-creation"
      :clients="clients"
      :selectedBudget="budget"
      :fixClientNameAndUpdate="fixClientNameAndUpdate">
    </budget-edit>

    <client-edit v-else-if="!budgetEdit && editPage"
      slot="client-creation"
      :selectedClient="client"
      :updateClient="updateClient">
    </client-edit>
  </div>
</template>

<script>
import BudgetCreation from './Creation/BudgetCreation'
import ClientCreation from './Creation/ClientCreation'
import BudgetEdit from './Creation/BudgetEdit'
import ClientEdit from './Creation/ClientEdit'
import UserCreation from './Creation/UserCreation'
import ReportsCreation from './Creation/ReportsCreation'

export default [
  {
    'budgetCreation', 'clients', 'saveBudget', 'submitSignup', 'userCreation', 'processClient',
    'saveClient', 'budget', 'client', 'updateClient',
    'fixClientNameAndUpdate', 'editPage', 'budgetEdit'
  },
]

```

Рисунок 16 – Фрагмент кода файла page/Create.vue

Отчеты в данном приложении являются сконфигурированными на клиенте и выведенными файлами формата xlsx. Страница отчетов представлена на рисунке 17.

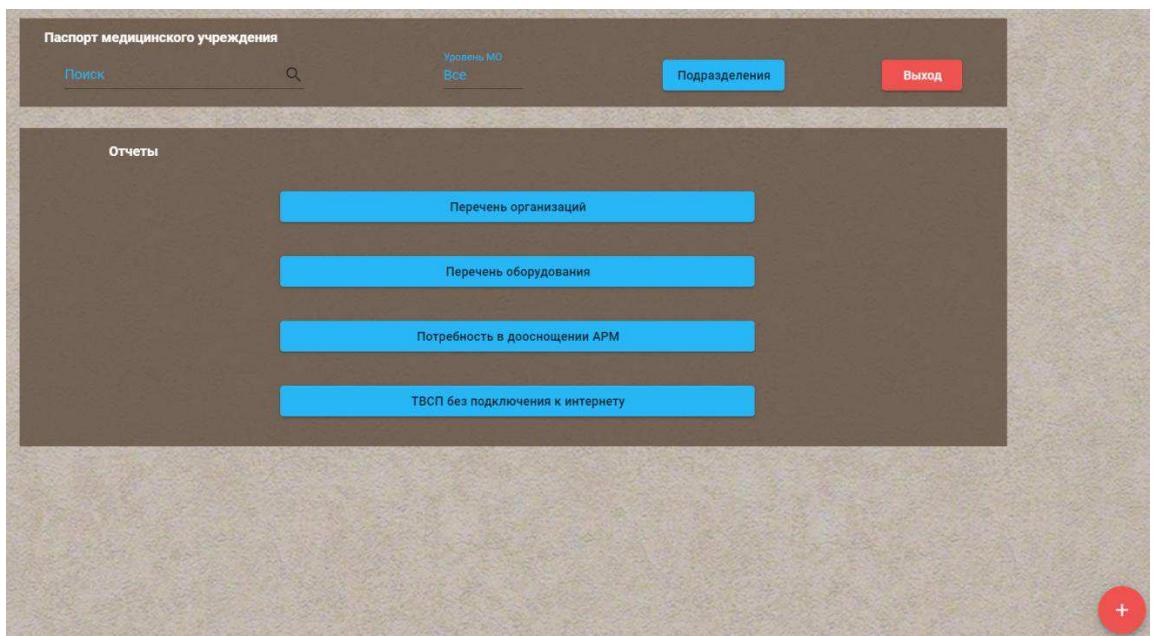


Рисунок 17 – Страница отчетов

При выборе пункта любого из пунктов будет создан и загружен файл расширения xlsx, представляющий набор данных, сформированный в зависимости от задачи перед отчетом, например при нажатии на кнопку «Потребность в дооснащении АРМ» создается отчет, в котором присутствуют основные данные о ТВСП, а также поле фактического количества АРМ, их планового количества и разница между ними, показывающая в каком здании, какой медицинской организации необходимо дооснащение, другая же кнопка «ТВСП без подключения к интернету» выводит список только тех подразделений, в которых на данный момент нет подключения к интернету.

## **2.3 Выводы по разделу “Разработка информационной системы”**

В данном разделе описывается поэтапная разработка приложения. В итоге имеется веб-приложение, состоящее из серверной части, написанной на Node.js, базы данных MongoDB и клиентской части, реализованной при помощи Vue.js. Веб-приложение включает экран просмотра таблиц ТВСП и МО, редактирование и создание отдельных элементов, а также экран выводов информации в Excel в виде отчетов.

### **3 Оценка экономической эффективности разработки информационной системы**

Для дальнейшей оценки проекта были установлены сроки создания информационной системы и ввода её в эксплуатацию:

1. Консультация с заказчиком, анализ предметной области 3 дня.
2. Разработка прототипа 10 дней.
3. Тестирование 2 дня.
4. Доработка 4 дня.
5. Введение в эксплуатацию 4 дня.

#### **3.1 Капитальные затраты**

Для оценки экономической эффективности разработки информационной системы была выбрана методика ТСО. Данный метод предлагает количественную оценку на внедрение и сопровождение программного обеспечения.

$$TCO = DE + IC_1 + IC_2, \quad (1)$$

где DE (direct expenses) – прямые расходы;

$IC_{1,2}$ (indirect costs) –косвенные расходы первой и второй групп.

Для данного проекта косвенные расходы не являются большими, поэтому  $TCO \approx DE$ .

Прямые затраты рассчитываются по формуле:

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (2)$$

где  $DE_1$  – капитальные затраты;

$DE_2$  – расходы на управление информационными технологиями;

$DE_3$  – расходы на техническую поддержку;

$DE_4$  – расходы на разработку прикладного внутреннего ПО внутренними силами;

$DE_5$  – расходы на аутсорсинг;

$DE_6$  – командировочные расходы;

$DE_7$  – расходы на услуги связи;

$DE_8$  – прочие расходы.

Капитальные затраты

Капитальные затраты вычисляются по следующей формуле.

$$K = K_{пр} + K_{тс} + K_{лс} + K_{пс} + K_{ио} + K_{об} + K_{оэ}, \quad (3)$$

где  $K_{пр}$  – затраты на проектирование ИС;

$K_{тс}$  – затраты на технические средства управления;

$K_{лс}$  – затраты на создание локальных связей;

$K_{пс}$  – затраты на программные средства;

$K_{ио}$  – затраты на формирование информационной базы;

$K_{об}$  – затраты на обучение персонала;

$K_{оэ}$  - затраты на опытную эксплуатацию.

$K_{tc} = 0$ , потому что ИС не взаимодействует с другими системами, которые используются на предприятии.

$K_{lc} = 0$ , потому что не надо проводить соединение между ПК пользователей, так как используется подключение через Интернет.

$K_{po} = 0$ , по причине нулевой стоимости ПО необходимого для получения доступа к ИС.

$K_{io} = 2000$  руб., заработка плата специалиста ГКУЗ РХ "РМИАЦ" за 2 дня, так как база данных приложения разработчиком заполняется только для представления возможностей.

$K_{ob} = 2000$  руб. на проведение вебинара по обучению специалистов в медицинских учреждениях.

$K_{oo} = 900$  руб., заработка плата разработчика за 1 день для внесения правок в проект в период эксплуатации, данные о зарплате разработчика представлены в таблице 2.

Затраты на проектирование информационной базы.

$$K_{pr} = K_{zp} + K_{ipc} + K_{cvt} + K_{proch}, \quad (4)$$

где  $K_{zp}$  – затраты на заработную плату проектировщика;

$K_{ipc}$  – на инструментальные ПО для проектировщика;

$K_{cvt}$  – на средства вычислительной техники для проектировщика;

$K_{proch}$  – прочие затраты.

Расчет зарплаты разработчика информационной системы представлен в таблице 2.

Таблица 2 – Расчетный листок

Расчетный листок за месяц		
Петров Артем Васильевич		
Подразделение: отдел разработки (Full-stack разработчик)		
Норма: 23 дня/138 часов		
Начисление/Удержание	Начислено, руб.	Удержано, руб.
Оклад	12000	
Премия	1000	
Районный коэффициент	3960	
Северный коэффициент	3960	
Итого	20920	
НДФЛ		2719
Итого на руки	18201	
Оплата 1 д. руб.	791	

Для оценки затрат на заработную плату необходимо учитывать обязательные отчисления во внебюджетные фонды, которые составляют 30,2% от начисленной заработной платы.

$$K_{зп} = 20920 * 1,302 = 27235 \text{ руб.} \quad (5)$$

Программное обеспечение для реализации и для работы разработанного приложения.

Затраты на инструментальные ПО для проектировщика в данном проекте рассчитываются из суммы затрат на программное обеспечение, с учетом возможного дальнейшего их использования на этом оборудовании для реализации других проектов, стоимость программ указана в таблице 3.

Таблица 3 – Стоимость программного обеспечения

Наименование	Назначение	Стоимость, руб.	Срок использования
Операционная система Microsoft Windows 10 Домашняя	Интерфейс для взаимодействия человека с машиной	8499	365+
Kaspersky Anti-Virus 2016	Для защиты данных от утечки	1299/год	365
Microsoft Office 2019	Для создания документации по проекту	3450	365+
Visual Studio Code	Программирование функционала приложения	бесплатно	365+
MongoDB	База данных	бесплатно	365+

ОС Windows и MicrosoftOffice будут актуальны ещё два года, за это время создается примерно 2 проекта в год, следовательно, общая стоимость делится 4, а стоимость Kaspersky Anti-Virus делится на 2.

$$K_{ипс} = (8499+3450)/4 + 1299/2=3635 \text{ руб.} \quad (6)$$

Стоимость оборудования используемого в процессе разработки веб-приложения для учета уровня автоматизации и информатизации медицинских учреждений представлена в таблице 4.

Таблица 4 – Стоимость оборудования

Наименование оборудования	Наименование элементов	Количество	Стоимость элемента, руб.	Стоимость всего, руб.
Компьютер для программиста и дизайнера	Процессор Intel Core i3-9100F BOX	1	7 299	7 299
	Видеокарта ASUS GeForce GT 730 [GT730-2GD5-BRK]	1	6199	6199
	Оперативная память Kingston Value RAM [KVR24N17S6/4] 4 ГБ	1	3099	3099
	Материнская плата GIGABYTE GA-H110N	1	4499	4499
	Корпус AeroCool Qs-182 черный	1	1899	1899
	500 ГБ Жесткий диск WD Black [WD5003AZEX]	1	4950	4950
	Блок питания InWin Powerman 600W [PM-600ATX-F]	1	1950	1950
	18.5" Монитор ASUS VS197DE [90LMF1001T02201C]	1	4699	4699
	Клавиатура DEXP K-5003BU	1	699	699
	Мышь проводная Jet.A OM-U54 серый	1	399	399
			Итого	35792

Так как компьютер для разработки уже имеется у разработчика и его не надо покупать, тогда расходы на технические средства разработки равны амортизации ПК и его комплектующих.

По статистике, средний срок службы компьютерных комплектующих составляет:

- Процессор – до 5 лет.
- Материнская плата – от 2 – до 5 лет.
- Жесткий диск (ЖД) – от 5 – до 7 лет (600 000 циклов start/stop).
- Модуль оперативной памяти (ОЗУ) – около 5 лет.
- Блок питания – от 3 – до 5 лет.
- Видеокарта – от 3 – до 5 лет.
- ЖК-монитор – от 4 – до 5 лет.
- Мыши и клавиатура – около 3-5 лет.

Исходя из этого можно предположить, что при нормальной работе срок службы компьютера составит 5 лет.

$$A_{\text{год}} = C_6 * H_{\text{ам}}, \quad (7)$$

где  $A_{\text{год}}$  – амортизация за год использования;

$C_6$  – балансовая стоимость;

$H_{\text{ам}}$  – норма амортизации.

$$A_{\text{пр}} = \frac{A_{\text{год}}}{K_{\text{рдг}}} * K_{\text{дэ}}, \quad (8)$$

где  $A_{\text{пр}}$  – проектная амортизация;

$K_{\text{рдг}}$  – количество рабочих дней в 2020 году;

$K_{\text{дэ}}$  – количество дней эксплуатации.

Так как срок эксплуатации ряда элементов может быть меньше 5 лет, нужно предусмотреть возможность их замены. Судя по информации, следует предусмотреть возможность замены процессора и материнской платы.

$$A_{год} = 27993 * 0.2 + (7299 + 4499) * 0.33 = 9138 \text{ руб.} \quad (9)$$

$$A_{пр} = \frac{9138}{248} * 23 = 847 \text{ руб.} \quad (10)$$

Так как возможна большая нагрузка на видеокарты и процессоры в следствие чего есть риск поломки данных комплектующих в ходе работы необходимо заложить 1000 рублей на обслуживание.

Прочие затраты: в прочие затраты входят: затраты на непредвиденные расходы. Принято, что на прочие затраты необходимо оставлять не меньше 3% от общих расходов на ПО и затраты на вычислительную технику. Так как на прочие расходы был взят минимальный порог отчислений равный 3% сумма затрат составила 951 руб.

В таблице 5 представлены затраты на проектирование информационной системы, они же капитальные затраты на проект.

Таблица 5 – Затраты на проектирование

Затраты	Состав затрат	Планируемая сумма
Затраты на заработную плату	Заработка разработчика	27235 руб.
Затраты на разработку информационной системы	Затраты на ПО	3635 руб.
	Амортизация ТО на период проекта	847 руб.
	Прочие затраты	951 руб.
Итого		33964 руб.

$$K_{пр} = 32668 + 0 + 0 + 0 + 2000 + 2000 + 900 = 37568 \text{ руб.} \quad (11)$$

Соотношение капитальных затрат на проект представлены на рисунке 13.



Рисунок 12 – Соотношение капитальных затрат

Наибольший удельный вес в капитальных затратах принадлежит затратам на проектирование.

### 3.2 Эксплуатационные затраты

Для работоспособности системы потребуется интернет, но так как он уже оплачен и работает по всей территории ГКУЗ РХ "РМИАЦ" для поддержания других систем, то оплачивать его заново не надо, поэтому затраты на Интернет = 0.

$$C = C_{зп} + C_{ao} + C_{to} + C_{lc} + C_{ни} + C_{проч}, \quad (12)$$

где  $C_{зп}$  – зарплата персонала, работающего с информационной системой;

$C_{ao}$  – амортизационные отчисления;

$C_{to}$  – затрата на техническое обслуживание;

$C_{\text{лс}}$  – затраты на использование глобальных сетей;

$C_{\text{ни}}$  – затраты на носители информации;

$C_{\text{проч}}$  – прочие затраты.

$C_{\text{зп}}$  – 300 руб/мес, так как предстоит работа по обновлению информации в базе данных и формированию отчетов.

$C_{\text{ao}}$  – 0, для работы базы данных необходим сервер. Ресурсы имеющего сервера позволяет без потери работоспособности разместить на нем базу данных, поэтому нет необходимости покупать новый.

$C_{\text{то}}$  – 1000 руб/год, так как в течение года возможна необходимость в доработке ИС.

$C_{\text{лс}}$  – 0, не потребуются глобальные вычислительные сети.

$C_{\text{ни}}$  – 0, так как вся информация передается через интернет и хранится на сервере.

$$C_{\text{проч}} = (300 * 12 + 1000 *) 0,03 = 138 \text{ руб./мес.} \quad (13)$$

$$C = 300 * 12 + 1000 + 138 = 4738 \text{ руб./год} \quad (14)$$

Эксплуатационные затраты значительно меньше капитальных, так как впускается готовый продукт, в котором необходимо только периодически обновлять данные.

$$TCO = DE + IC_1 + IC_2, \quad (15)$$

где  $DE$  – прямые расходы;

$IC_{1,2}$  – косвенные расходы первой и второй группы;

$IC_{1,2}$  – затраты на проведение вебинара и заполнение информационной базы внесены в капитальные, других косвенных затрат в проекте нет.

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (16)$$

где  $DE_1$  – капитальные затраты;

$DE_2$  – расходы на управление ИТ;

$DE_3$  – расходы на техническую поддержку АО и ПО;

$DE_4$  – расходы на разработку прикладного ПО внутренними силами;

$DE_5$  – расходы на аутсорсинг;

$DE_6$  – командировочные расходы;

$DE_7$  – расходы на услуги связи;

$DE_8$  – другие группы расходов.

$DE_1 = 37568$  руб.

$DE_2 = 0$ , так как на предприятии же есть опыт работы с подобными программными продуктами.

$DE_3 = 4738$  руб./год.

$DE_4 = 0$ , потому что нет необходимости осуществлять разработку внутренними силами.

$DE_5 = 0$ , потому что необходимости пользоваться аутсорсингом, так как проект не имеет сложного функционала.

$DE_6 = 0$ , так как нет необходимости отправлять работников в командировку.

$DE_7 = 0$ , по причине использования интернета уже используемого для поддержания других систем.

$DE_8 = 0$ , прочие расходы рассчитывались отдельно для каждой группы расходов.

$$DE = 37568 + 4738 = 42306 \text{ руб.} \quad (17)$$

$$\text{TCO} = 42306 \text{ руб.} \quad (18)$$

### 3.3 Расчет экономической эффективности ИС

Разработка систем учета уровня автоматизации и оснащенности информационно-техническим оборудованием медицинских учреждений Республики Хакасия создается с целью сокращения времени доступа к этой информации и упрощении создания отчетности, связанной с автоматизацией процессов в медицинских учреждениях.

Для оценки эффективности разрабатываемого продукта было проведено сравнение основных характеристик с текущим вариантом выполнения данной работы по пятибалльной шкале (таблица 6).

Таблица 6 – Расчет показателя качества

Показатель качества	Весовой коэффициент, $b_i$	Оценка $X_i$	
		Разраб. проект	Базовый проект
Удобство работы	0,1	4	1
Надежность	0,2	2	1
Функциональные возможности	0,2	3	1
Временная эффективность	0,4	5	1
Время обучения персонала	0,1	1	1
Комплексный показатель качества $I_{\text{эт}}$		3,5	1

Коэффициент технического ровня:

$$k_t = I_{\text{этуп}} / I_{\text{этубаз}}, \quad (19)$$

где  $I_{\text{этубаз}}$  и  $I_{\text{этуп}}$  – комплексные показатели качества, разрабатываемого и базового проектов.

$$k_t = 3 / 1 = 3,5 \quad (20)$$

Для расчета экономического эффекта рассчитаем приведенные затраты  $Z_i$  на единицу работ, выполняемых по базовом и разрабатываемому вариантам по формуле:

$$Z_i = C_i + E_h * Z_{ppi}, \quad (21)$$

где  $C_i$  – текущие эксплуатационные затраты единицы iго вида работ, руб.

$Z_{ppi}$  – суммарные затраты, связанны с внедрением проекта.

$E_h = 0,33$  – нормативный коэффициент экономической эффективности.

$$C_{баз} = 22000 / 21 * 3 * 4 = 12571 \text{ руб./месяц}, \quad (22)$$

где 22000 – оклад сотрудника, который занимается формированием отчетов об автоматизации и информатизации медицинских учреждений.

21 – количество рабочих дней в месяце в сотрудников

3 – количество дней, когда сотрудник формирует отчеты

4 – отчеты необходимо предоставлять 4 раза в год

Для базового проекта:

$$Z_{баз} = 12571 + 0,33 * 0 = 12571 \text{ руб.} \quad (23)$$

Для проекта:

$$Z_{пр} = 4738 + 0,33 * 37568 = 17 135,6 \text{ руб.} \quad (24)$$

Экономический эффект от использования разрабатываемо системы определяется по формуле:

$$\Theta = (Z_{баз} * k_T - Z_{пр}) * V, \quad (25)$$

где  $Z_{баз}$ ,  $Z_{пр}$  – приведенные затраты на единицу работ, выполняемых с помощью базового и проектируемого вариантов процессов обработки информации;

$k_t$  – коэффициент эксплуатационно-технической эквивалентности;

$V$  – объем работ, выполняемых с помощью разрабатываемого проекта, натуральные единицы.

Экономический эффект от использования разрабатываемой системы:

$$\mathcal{E} = (12571 * 3,5 - 17 \cdot 135,6) * 1 = 26\ 862,9 \quad (26)$$

Также необходимо рассчитать срок окупаемости затрат на разработку проекта о формуле:

$$T_{ок} = Z_{пп} / \mathcal{E}, \quad (27)$$

где  $Z_{пп}$  – единовременные затраты на разработку проекта;

$\mathcal{E}$  – готовая эффективность.

Рассчитываемый срок окупаемости затрат на разработку продукта:

$$T_{ок} = 37568 / 26862,9 = 1,44 \quad (28)$$

Таким образом, срок окупаемости составляет примерно два года.

Фактический коэффициент экономической эффективности разработки:

$$E_{\phi} = 1 / T_{ок} \quad (29)$$

Нормативное значение коэффициента эффективности капитальных вложений  $E_h = 0,33$ , если  $E_{\phi} > E_h$ , то делается вывод об эффективности капитальных вложений.

Рассчитаем фактический коэффициент экономической эффективности:

$$E_{\phi} = 1 / 1,44 = 0,69 \quad (30)$$

Так как  $E_{\phi} = 0,69 > E_n$ , то разработка и внедрение разрабатываемого продукта являются эффективными, т.е. эффект от использования данной системы окупает все затраты, связанные с проектированием и эксплуатацией.

В таблице 7 приведены сводные данные экономического обоснования.

Таблица 7 – Сводные данные экономического обоснования

Показатель	Величина
Затраты на разработку системы	37 568 руб.
Общие эксплуатационные затраты	4 738 руб.
Экономический эффект	26862,9 руб.
Коэффициент экономической эффективности	0,69
Срок окупаемости	1 год 6 месяцев

Разработанная система повышает скорость получения актуальной информации об автоматизации и информационно-техническом оснащении медицинских учреждений, экономит время, поэтому экономическая эффективность очевидна.

### 3.4 Оценка риска при реализации информационной системы

Рассмотрим следующие риски, связанные с реализацией информационной системы:

Отсутствие необходимости в использовании информационной системы у сотрудников РМИАЦ. Заказчик может заблуждаться в необходимости системы в данный момент, большинство сотрудников привыкли делать

механический труд и могут воспринять переход на новую систему не так, как ожидается.

Ошибки в обновлении данных, неактуальные данные базе. Разрабатываемая система предполагает обновление изменение данных на актуальные, если некоторое время этого не делать, то отчеты могут содержать серьезные смысловые ошибки.

В таблице 8 представлены основные риски проекта.

Таблица 8 – Возможные риски

№	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
1	Риски, связанные с реализацией проекта	Отсутствие необходимости использования системы	Средний	Низкий	Уведомление о появлении новой программы, проведение вебинара по обучению пользования новой системой
2	Риски, связанные с выполнением проекта	Неактуальные, неверные данные в базе	Средний	Низкий	Установление правил пользования программой в каждом медицинском учреждении, объявление ответственных лиц за обновление информации о объектах

Чтобы избежать рисков, связанных с невостребованностью разрабатываемой ИС, следует вовлечь сотрудников в создание системы, прислушаться к их пожеланиям.

Чтобы избежать рисков, связанных с ошибками в обновлении данных в базе, необходимо на уровне руководства предприятия принять правила

использования данного продукта в медицинских учреждениях, назначить ответственных за обновление данных лиц.

### **3.5 Выводы по разделу “Оценка экономической эффективности разработки информационной системы”**

В результате экономических расчетов было установлено, что разрабатываемый проект имеет капитальные затраты, равные 37 568 руб. и эксплуатационные затраты, равные 4 738 руб. При этих показателях разрабатываемая ИС имеет коэффициент экономической эффективности равный 0,69. Это означает, что разработка и внедрение данной системы является эффективным и все затраты будут окуплены. Отсутствие большого списка рисков обусловлено технически не сложным функционалом ИС, поэтому система будет работать стабильно.

## **ЗАКЛЮЧЕНИЕ**

В разделе «Анализ предметной области» был проведен анализ основной деятельности ГКУЗ РХ «РМИАЦ» при помощи построения модели до внедрения системы и после по методологии моделирования IDEF0, который позволил выявить излишние затраты времени на создание отчетов об автоматизации и оснащённости информационно-техническим оборудованием медицинских учреждений Республики Хакасия. Для демонстрации функционала внутри системы была построена модель IDEF3 показывающая пошагово использование для создания отчетов, ограничение доступа к системе и администрирование.

Для реализации информационной системы, направленной на создание удобного пользовательского интерфейса для доступа к базе данных, были проанализированы наиболее популярные средства в среде современных ИТ-разработчиков и выбраны следующие:

- язык программирования: JavaScript;
- СУБД: MongoDB.

Язык программирования JavaScript, а точнее его расширение в виде NodeJS, представляет широкий функционал в проектировании и развёртывании на сервере веб-приложения любой сложности без необходимости использовать разные языки для разных частей.

Согласно проведенному анализу СУБД создание базы данных будет осуществляться с помощью MongoDB, эта NO-SQL документо-ориентированная база данных прекрасно взаимодействует с NodeJS и предоставляет функционал построения схемы взаимодействия объектов внутри кода программы.

В разделе «Разработка системы учета уровня автоматизации и оснащенности информационно-техническим оборудованием медицинских учреждений Республики Хакасия» была описана разработка

информационной системы в виде веб-приложения, представляющей пользовательский интерфейс в браузере, базу данных и сервер для их связи.

В разделе «Оценка экономической эффективности разработки информационной системы» были выявлены технические характеристики проекта, были посчитаны капитальные затраты, которые составляют 37 568 руб. и эксплуатационные затраты, равные 4 738 руб. При этих показателях разрабатываемая ИС имеет коэффициент экономической эффективности равный 0,69 и срок окупаемости 1 год и 6 месяцев, а это показывает, что разработка ИС экономически эффективна, т.к. количество затрат существенно снизится. Были определены риски и возможность их предотвращения.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Выпускная квалификационная работа. Методические указания для студентов [Электронный ресурс]. – Режим доступа: <http://elib.sfu-kras.ru/handle/2311/71001>
2. Введение в JSON [Электронный ресурс]. – Режим доступа: <https://www.json.org/json-ru>
3. ГКУЗ РХ «Республиканский медицинский информационно-аналитический центр» [Электронный ресурс]. – Режим доступа: <https://miac.mz19.ru/miac/>
4. Как анализировать бизнес-процессы с помощью IDEF0 [Электронный ресурс]. – Режим доступа: <https://zen.yandex.ru/media/id/597544a7b2d009201e3c2053/kak-analizirovat-biznesprocessy-s-pomoschiu-idef0-597f04b51410c389333ec29a>
5. Министерство здравоохранения Республики Хакасия [Электронный ресурс]. – Режим доступа: <https://mz19.ru/>
6. Начало работы с Node.js в Windows [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/windows/nodejs/beginners>
7. Об основах охраны здоровья граждан в Российской Федерации: федер. закон Рос. Федерации от 21 ноября 2011 г. №323-ФЗ: принят Гос. Думой 1 ноября 2011 г.: одобр. Советом Федерации 9 ноября 2011 г. // Рос. газ. – 2011. – 23 ноября.
8. Основы IDEF3 [Электронный ресурс]. – Режим доступа: <https://www.cfin.ru/vernikov/idef/idef3.shtml>
9. Основы JavaScript [Электронный ресурс]. – Режим доступа: <https://html5book.ru/osnovy-javascript/>
10. Положение о министерстве здравоохранения Республики Хакасия: Постановление правительства Республики Хакасия от 11 июня 2009 г. №260 (в ред. от 07.03.2017 N 93) // Вестник Хакасии. – 2009. – 23 июня.

11. Постановление правительства Республики Хакасия от 30 августа 2007 года – «О переименовании Государственного учреждения здравоохранения «Республиканское бюро медицинской статистики» в Государственное учреждение здравоохранения Республики Хакасия «Медицинский информационно-аналитический центр» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/819015519>

12. Разработка веб приложений [Электронный ресурс]. – Режим доступа: <https://infoshell.ru/blog/razrabotka-veb-prilozhenij/>

13. РИАМС ПроМед [Электронный ресурс]. – Режим доступа: [http://swan-it.ru/elektronnoe\\_zdravooohranenie/riams\\_promed](http://swan-it.ru/elektronnoe_zdravooohranenie/riams_promed)

14. Теория и методика обучения информатике: учебник / [М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер, М.И. Рагулина и др.]; под ред. М.П. Лапчика. – М.: Издательский центр «Академия», 2008. – 592 с.

15. Excel JS [Электронный ресурс]. – Режим доступа: <https://www.npmjs.com/package/exceljs>

16. PHP: Руководство по PHP[Электронный ресурс]. – Режим доступа: <https://www.php.net/manual/ru/intro-whatis.php>

17. VisualStudioCode – эволюция кроссплатформенного редактора кода [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/microsoft/blog/268837/>

18. Vue.js. Руководство. [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org/v2/guide/>

19. Wikipedia [Электронный ресурс]: Visual Studio Code. – Режим доступа: [https://ru.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://ru.wikipedia.org/wiki/Visual_Studio_Code)

20. Wikipedia [Электронный ресурс]: MongoDB. – Режим доступа: <https://ru.wikipedia.org/wiki/MongoDB>

Выпускная квалификационная работа выполнена мной самостоятельно.  
Использованные в работе материалы и концепции из опубликованной  
научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 20 наименований.

Экземпляр сдан на кафедру.

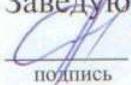
«\_\_\_\_» \_\_\_\_\_ 2020 г.

\_\_\_\_\_ Петров Артем Васильевич  
подпись

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

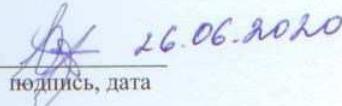
УТВЕРЖДАЮ  
Заведующий кафедрой  
  
E.N. Скуратенко  
подпись  
«26» 06.20 2020г.

**БАКАЛАВРСКАЯ РАБОТА**

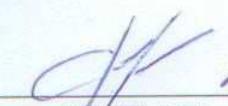
09.03.03 – Прикладная информатика

Тема: Разработка системы учета уровня автоматизации и оснащенности  
информационно-техническим оборудованием медицинских учреждений  
Республики Хакасия

Руководитель  зав. кафедрой, доцент, к.т.н. Е. Н. Скуратенко  
подпись, дата

Выпускник  А. В. Петров  
подпись, дата

Консультанты по разделам:

Экономический  26.06.20 Е. Н. Скуратенко  
подпись, дата

Нормоконтролер  В. И. Кокова  
подпись, дата

Абакан 2020

Студенту Петрову Артему Васильевичу

Группа 56-1(ХБ 16-03)

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка системы учета уровня автоматизации и оснащенности информационно-техническим оборудованием медицинских учреждений Республики Хакасия

Утверждена приказом по институту № 216 от 06.04.2020 г.

Руководитель ВКР Е.Н. Скуратенко, зав. кафедрой, доцент, к.т.н., ХТИ – филиал СФУ

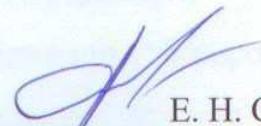
Исходные данные для ВКР: информация о текущем состоянии работ по созданию отчетов об уровне автоматизации и информатизации медицинских учреждений, технические характеристики разрабатываемого в ВКР программного продукта.

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки системы для учета уровня автоматизации и информатизации медицинских учреждений Республики Хакасия с возможностью создания отчетов.
3. Оценка экономической эффективности веб-приложения для учета уровня автоматизации и информатизации медицинских учреждений Республики Хакасия.

Перечень графического материала: нет

Руководитель ВКР

  
E. N. Скуратенко  
подпись

Задание принял к исполнению

  
A. V. Петров  
подпись

«06» апреля 2020 г.