

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

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

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е.Н. Скуратенко
подпись инициалы, фамилия
« _____ » _____ 2019г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 – Прикладная информатика
код – наименование направления

Автоматизация передачи сведений о диспансерном наблюдении
тема
застрахованных лиц из РИАМС «ПроМед» в ИС ТФОМС РХ

Руководитель _____ ст. преподаватель В.И. Кокова
подпись, дата должность, ученая степень инициалы, фамилия

Выпускник _____ И.М Прохорович
подпись, дата инициалы, фамилия

Консультанты по
разделам:

Экономический _____ Е.Н. Скуратенко
наименование раздела подпись, дата инициалы, фамилия

Нормоконтролер _____ В.И. Кокова
подпись, дата инициалы, фамилия

Абакан 2019

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

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

институт

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
 Е.Н. Скуратенко
подпись инициалы, фамилия
« ____ » _____ 2019 г

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

Студенту Прохорович Ивану Михайловичу

фамилия, имя, отчество

Группа 55-1 (ХБ 15-04) Направление (специальность) 09.03.03

номер

код

Прикладная информатика

наименование

Тема выпускной квалификационной работы Автоматизация передачи сведений о диспансерном наблюдении застрахованных лиц из РИАМС «ПроМед» в ИС ТФОМС РХ

Утверждена приказом по институту № 279 от 18.04.2019 г.

Руководитель ВКР В.И. Кокова, ст. преподаватель, ХТИ – филиал СФУ
инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР общие требования ГКУЗ РХ «РМИАЦ» для разработки АИС передачи сведений о диспансерном наблюдении застрахованных лиц в ИС ТФОМС РХ

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.

2. Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

3. Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

Перечень графического материала —

Руководитель ВКР

подпись

В.И. Кокова

инициалы и фамилия

Задание принял к исполнению

подпись,

И.М. Прохорович

инициалы и фамилия студента

« 19 » апреля 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа (ВКР) по теме «Автоматизация передачи сведений о диспансерном наблюдении застрахованных лиц из РИАМС «ПроМед» в ИС ТФОМС РХ» содержит 68 страниц текстового документа, 12 формул, 12 таблиц, 15 рисунков, 18 использованных источников.

ГКУЗ РХ «РМИАЦ», ТФОМС РХ, АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗЫ ДАННЫХ, МИКРОСЕРВИСНАЯ АРХИТЕКТУРА, ПЕРЕДАЧА ДАННЫХ, ФОРМАТ ДАННЫХ.

Объект ВКР: процесс преобразования данных.

Целью ВКР является сокращение времени обработки и передачи сведений о диспансерном наблюдении застрахованных лиц с помощью разработки автоматизированной информационной системы. Для достижения поставленной цели были решены следующие задачи: осуществлён поиск информации по организационной характеристике предметной области; проведено ознакомление с основной деятельностью ГКУЗ РХ «РМИАЦ»; составлена характеристика ИТ-инфраструктуры ГКУЗ РХ «РМИАЦ»; сформулированы цель и задачи разработки проекта; обоснован выбор средств проектных решений; построены модели DFD и IDEF3 для реализации автоматизированной информационной системы (АИС) передачи сведений о диспансерном наблюдении застрахованных лиц из РИАМС «ПроМед» в ИСТФОМС РХ; разработана АИС для передачи данных из РИАМС «ПроМед» в ИСТФОМС РХ; проведена оценка экономической эффективности АИС.

В итоге была разработана АИС для передачи сведений о диспансерном наблюдении застрахованных лиц из РИАМС «ПроМед» в ИС ТФОМС РХ, преобразованных из формата XML в формат JSON.

SUMMARY

The theme of the graduation thesis is «Automation of data transfer on regular medical check-ups of the insured from RIAMS «ProMed» (Regional Information Analytics Medical System) to IS of MHIF RKH». It contains 68 pages, 12 formulae, 12 charts, 15 drawings, 18 reference items.

SHCPI RKH «RMCIA» (State Healthcare Public Institution of Republic of Khakassia «Republican Medical Center for Information and Analysis»), MHIF RKH (Mandatory Health Insurance Fund of Republic of Khakassia), AUTOMATED INFORMATION SYSTEM, DATABASES, MICROSERVICE ARCHITECTURE, DATA TRANSFER, DATA FORMAT.

The object of the graduation thesis is data conversion process.

The purpose of the graduation thesis is to reduce the processing and information transfer time on regular medical check-ups of the insured via the development of an automated information system. To achieve the present purpose the following objectives have been performed: the subject area information has been searched for; the mainstreaming of SHCPI RKH «RMCIA» has been analyzed; the IT infrastructure of SHCPI RKH «RMCIA» has been characterized; the purpose and the objectives of the project have been defined; the selection of design concept tools has been proved; DFD and IDEF3 models have been created to implement an automated information system (AIS) for information transfer on regular medical check-ups of the insured from RIAMS «ProMed» to IS (Information System) of MHIF RKH; the AIS has been developed for data transfer from RIAMS «ProMed» to IS of MHIF RKH; the AIS cost-effectiveness has been calculated.

Results: the AIS has been developed for data transfer from RIAMS «ProMed» to IS of MHIF RKH, and being converted from XML format into JSON format.

English language supervisor

(signature, date)

(full name)

СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области. Выбор средств проектных решений	10
1.1 Организационная характеристика ГКУЗ РХ «РМИАЦ».....	10
1.2 Программные продукты, имеющиеся в распоряжении ГКУЗ РХ «РМИАЦ»	13
1.3 Защищенная сеть передачи данных	15
1.4 Описание Территориального Фонда обязательного медицинского страхования Республики Хакасия	16
1.5 Постановка цели и задач проектирования	17
1.6 Обоснование выбора средств для разработки автоматизированной информационной системы передачи сведений о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.....	18
1.6.1 Выбор языка программирования для разработки.....	20
1.6.2 Выбор системы управления базами данных для разработки	24
Выводы по разделу «Анализ предметной области. Выбор средств проектных решений»	27
2 Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ	28
2.1 Описание модели реализации автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении из РИАМС «ПроМед» в ИС ТФОМС РХ по нотациям DFD и IDEF3.....	28
2.2 Описание БД шины для получения данных из РИАМС «ПроМед» и передачи данных в ИС ТФОМС РХ.....	37
2.3 Разработка автоматизированной информационной системы для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ.....	39
2.3.1 Разработка микросервиса принятия информации из базы данных РИАМС «ПроМед»	39
2.3.2 Разработка микросервиса преобразования данных из формата XML в JSON.....	40

2.3.3 Разработка микросервиса загрузки преобразованных данных в БД шины	43
2.3.4 Разработка микросервиса передачи данных из БД шины в ИСТФОМС РХ	45
2.4 Тестирование	45
Выводы по разделу «Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ»	46
3 Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ	47
3.1 Технические характеристики проекта.....	47
3.2 Расчет затрат реализации проекта разработки ИС	48
3.2.1 Капитальные затраты	49
3.2.2 Эксплуатационные затраты	53
3.2.3 Прямые затраты.....	55
3.3 Экономическая эффективность	56
3.4 Оценка риска при реализации проекта разработки АИС	62
3.4.1 Меры по предотвращению или снижению риска	63
Вывод по разделу «Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ»	63
Заключение	65
Список использованных источников	67
Приложение А	69
Приложение Б.....	73
Приложение В	75
Приложение Д	79
Приложение Ж	83
Приложение И	84

ВВЕДЕНИЕ

Для нормального функционирования различных организаций, а особенно государственных, первостепенной задачей является использование существующего, создание своего программного обеспечения (ПО) для обработки и обмена данными.

ГКУЗ РХ «РМИАЦ» осуществляет свою деятельность с целью формирования единой информационной системы здравоохранения Российской Федерации путем организации на базе современных компьютерных технологий межотраслевой системы сбора, обработки, хранения и представлений информации.

На данный момент в ГКУЗ РХ «РМИАЦ» нет информационной системы, которая позволила бы автоматически передавать данные о застрахованных лицах, стоящих на диспансерном наблюдении, в Территориальный Фонд обязательного медицинского страхования Республики Хакасия (ТФОМС РХ). Данная автоматизированная информационная система нужна для того, чтобы связать потоки данных в системах, которые используют разные форматы данных. Таким образом, актуальность выбранной темы связана с необходимостью разработки автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из Региональной информационно-аналитической медицинской системой (РИАМС) «ПроМед» в ИС ТФОМС РХ.

Целью ВКР является сокращение времени обработки и передачи сведений о диспансерном наблюдении застрахованных лиц с помощью разработки автоматизированной информационной системы.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Осуществить поиск информации по организационной характеристике предметной области.

2. Ознакомиться с основной деятельностью ГКУЗ РХ «Республиканский медицинский информационно-аналитический центр».
3. Составить характеристику IT-инфраструктуры ГКУЗ РХ «Республиканский медицинский информационно-аналитический центр».
4. Сформулировать цель и задачи разработки проекта.
5. Обосновать выбор средств проектных решений.
6. Построить модели DFD и IDEF3 для реализации АИС передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении из РИАМС «ПроМед» в ИС ТФОМС РХ.
7. Описать БД шины для получения данных из РИАМС «ПроМед» и передачи данных в ИС ТФОМС РХ.
8. Разработать автоматизированную информационную систему для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ.
9. Оценить экономическую эффективность АИС.

1 Анализ предметной области. Выбор средств проектных решений

1.1 Организационная характеристика ГКУЗ РХ «РМИАЦ»

ГКУЗ РХ «РМИАЦ» осуществляет свою деятельность с целью формирования единой информационной системы здравоохранения Российской Федерации путем организации на базе современных компьютерных технологий межотраслевой системы сбора, обработки, хранения и представлений информации, обеспечивающей динамическую оценку здоровья и информационную поддержку принятия решений, направленных на его улучшение.

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

Учреждение выполняет следующие основные виды деятельности:

- Разработка концепций и программ информатизации 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, обработки, хранения и передачи информации.

— 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.

— Осуществление контроля над ведением в учреждениях здравоохранения медицинской документации, удостоверяющей случаи смерти.

— Обеспечение учреждений здравоохранения формами медицинской учетной и отчетной документации, утвержденными приказами Министерства здравоохранения Российской Федерации и постановлениями Госкомстата России.

— Изучение и прогнозирование процессов и явлений, связанных со здоровьем человека.

— Выработка управленческих решений по повышению эффективности деятельности учреждений здравоохранения.

— Организация и контроль над состоянием статистического учета и отчетности учреждений здравоохранения.

— Координация деятельности службы медицинской статистики Республики Хакасия.

— Представление в Министерство здравоохранения сводных государственных и отраслевых медицинских статистических отчетов в соответствии с установленным порядком.

— Формирование и сопровождение государственной статистической отчетности Российской Федерации.

— Анализ медико-статистической информации о состоянии здоровья населения и состоянии здравоохранения Республики Хакасия.

— Повышение эффективности использования информационной инфраструктуры здравоохранения Российской Федерации.

— Обеспечение достоверности сведений в учетной и отчетной медицинской документации.

— Внедрение новых технологий сбора и обработки медико-статистической информации.

— Создание и поддержание единого мониторинга здоровья населения;

— Подготовка и публикация аналитических обзоров о состоянии здоровья и здравоохранения, сборников основных показателей по разделам здравоохранения.

— Проведение мероприятий по повышению деловой и профессиональной квалификации работников медицинской статистики и технического обеспечения при организации семинаров, распространение передового опыта и оказание организационно-медицинской помощи.

Структурная схема ГКУЗ РХ «РМИАЦ» представлена на рисунке 1.1.

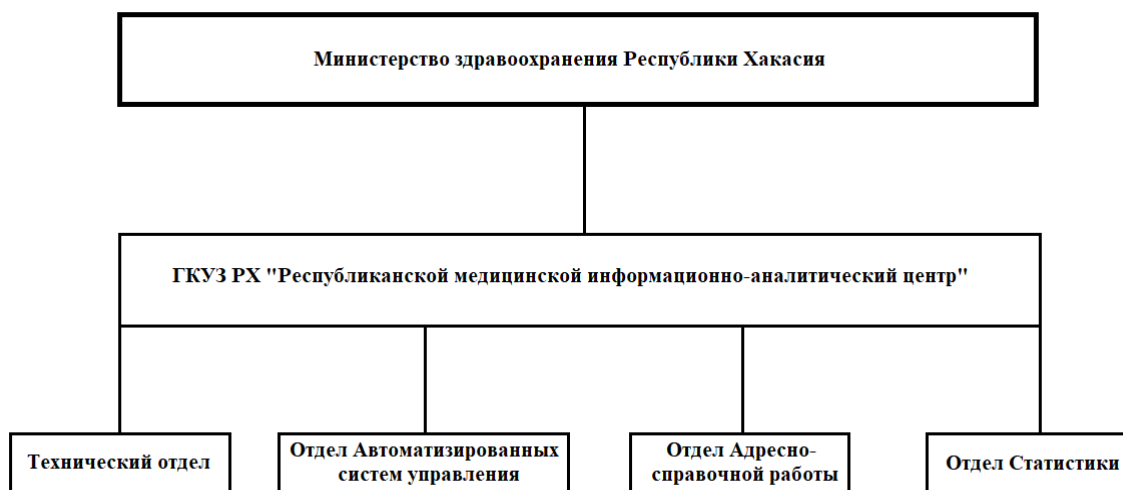


Рисунок 1.1 – Структурная схема ГКУЗ РХ «РМИАЦ»

В ГКУЗ РХ «РМИАЦ» существуют следующие отделы: технический, автоматизированных системы управления (АСУ), адресно-справочной работы (АСР), статистики.

Отдел АСУ сопровождает медицинскую систему РИАМС«ПроМед» и сайт «кврачу19.рф», принимает звонки от медицинских организаций. Если есть ошибки с этих сервисов, то сообщается о них либо разработчикам в ООО «Сван», либо техническому отделу.

Отдел АСР занимается проектом «Электронная Регистратура», и в этом отделе есть операторы, которые принимают звонки от граждан и записывают их на прием к врачам.

Отдел статистики работает с медицинской статистикой: обрабатывает информацию про больницы, про количество заболевших и т.п.

Технический отдел занимается развитием информационных технологий, работой над аппаратным обеспечением, информационными системами и внедрением софта. Также технический отдел занимается сбором информации от медицинских образований, поддержкой медицинских образований, которые касаются сервисов ГКУЗ РХ «РМИАЦ». Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ будет проводиться в техническом отделе.

1.2 Программные продукты, имеющиеся в распоряжении ГКУЗ РХ «РМИАЦ»

Основные информационные системы, используемые для решения задач в сфере здравоохранения Республики Хакасия, представлены в таблице 1.1.

Таблица 1.1 – Информационные системы, используемые для решения задач в сфере здравоохранения Республики Хакасия

№ п/ п	Название программного продукта	Краткое описание программного продукта
1	Единый медицинский портал Республики Хакасия	Предоставление государственной услуги «Заявка (запись) на прием к врачу» в электронном виде
2	Глобальная навигационная спутниковая система (ГЛОНАСС) "АвтоГРАФ"	Система спутникового мониторинга и контроля транспорта, которая позволяет видеть точное местоположение санитарного транспорта на карте и измерить простой и пробег, выбрать целесообразный маршрут движения
3	Телемедицинская система Республики Хакасия	Комплекс взаимосвязанных информационных систем для обеспечения дистанционного взаимодействия медицинских работников в целях организации медицинской помощи
4	Почтовый сервис РМИАЦ	Система, которая позволяет защитить принятие и передачу писем (?)
5	Джаббер сервер	Система для быстрого обмена сообщениями и информацией
6	ИПРА-Веб	Получение и направление в Медико-социальную экспертизу исполненных карт реабилитации инвалидов
7	Сервис АХД	Сбор информации об административно-хозяйственной деятельности медицинских организаций

В данной таблице не указан РИАМС «ПроМед», так как с помощью него будет реализована автоматизированная информационная система и данный программный комплекс лучше раскрыть более подробно.

РИАМС«ПроМед» – это специализированный программный комплекс, позволяющий автоматизировать процессы сбора, обработки и хранения медицинской, экономической и статистической информации в системе здравоохранения региона.

РИАМС «ПроМед» функционирует по модели SaaS (Software as a Services – программное обеспечение как услуга) на едином центре обработки данных для неограниченного числа пользователей. В центре обработки данных консолидируется вся информация, связанная с персонифицированным учетом оказанной медицинской помощи и управлением ресурсами здравоохранения региона. «ПроМед» обеспечивает информационный обмен между медицинскими учреждениями, органами управления здравоохранением, ТФОМС РХ, страховыми медицинскими организациями и аптечными учреждениями, участвующими в реализации программы дополнительного лекарственного обеспечения [1].

В 2007 году медицинская информационная система Пермского края была переведена на облачные технологии, в результате чего все участники здравоохранения региона получили возможность работать с единой базой данных в режиме реального времени. Тогда же началось тиражирование РИАМС в другие регионы. Из этого следует, что РИАМС «ПроМед» не является разработкой ГКУЗ РХ «РМИАЦ». Он был разработан в Перми. ГКУЗ РХ «РМИАЦ» только сопровождает РИАМС «ПроМед».

Именно из РИАМС «ПроМед» будет браться информация, которая нужна для передачи в ИСТФОМС РХ. Использование данной системы требует дополнительной автоматизации.

1.3 Защищенная сеть передачи данных

Данные, которые передаются, являются персональными, а это значит, что при передаче данных нужно придерживаться Федерального закона РФ от 27 июля 2006 года № 152-ФЗ «О персональных данных». В данном проекте

для защиты информации используется технология VPN, как со стороны ГКУЗ РХ «РМИАЦ», так и со стороны ТФОМС РХ.

VPN (англ. Virtual Private Network – виртуальная частная сеть) – это безопасное, зашифрованное подключение между двумя сетями или между отдельным пользователем и сетью. Сети VPN позволяют пользоваться Интернетом, сохраняя конфиденциальность.

Технология VPN шифрует все действия в интернете. Все, что отправляется и получается. Если входить только через VPN, виден неподлинный источник подключения, а один из многочисленных VPN-маршрутизаторов. Существует три основных вида шифрования: хеширование, симметричное и асимметричное шифрование. У каждого вида свои преимущества и недостатки, но все они шифруют данные так, что в чужих руках они будут бесполезными.

Дополнительный уровень защиты, который есть у большинства служб VPN – их собственная система DNS. DNS (система доменных имен)– это "телефонная книга интернета", в которой текстовые URL-адреса отождествлены с соответствующими IP-адресами. Киберпреступники могут наблюдать за запросами DNS, чтобы отслеживать действия в интернете, но система DNS в службах VPN разработана так, чтобы с помощью дополнительного шифрования помешать им [2].

1.4 Описание Территориального Фонда обязательного медицинского страхования Республики Хакасия

ТФОМС РХ создан для реализации государственной политики в сфере обязательного медицинского страхования Республики Хакасия. ТФОМС РХ управляет средствами обязательного медицинского страхования (ОМС), предназначенными для обеспечения гарантий бесплатного оказания застрахованным лицам медицинской помощи в рамках программы ОМС, обеспечивает финансовую устойчивость системы ОМС, контролирует

деятельность страховых медицинских организаций, качество оказания медицинской помощи застрахованным лицам, а также целевое использование средств обязательного медицинского страхования медицинскими организациями на территории Республики Хакасия [3].

У ТФОМС РХ имеется своя информационная система и формат записи данных (JSON) не совпадает с форматом данных у РИАМС «ПроМед» (XML), что создает дополнительные трудности, т.к. нужно будет преобразовать данные из формата XML в формат JSON.

1.5 Постановка цели и задач проектирования

На данный момент в ГКУЗ РХ «РМИАЦ» нет информационной системы, которая позволила бы автоматически передавать данные о застрахованных лицах, стоящих на диспансерном наблюдении, в ИСТФОМС РХ. Данная АИС нужна для того, чтобы связать потоки данных в системах, которые используют разные форматы данных. Таким образом, актуальность выбранной темы ВКР связана с необходимостью разработки автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

Целью ВКР является сокращение времени обработки и передачи сведений о диспансерном наблюдении застрахованных лиц с помощью разработки автоматизированной информационной системы.

1. Для достижения поставленной цели необходимо решить следующие задачи:

2. Осуществить поиск информации по организационной характеристике предметной области.

3. Ознакомиться с основной деятельностью ГКУЗ РХ "Республиканский медицинский информационно-аналитический центр".

4. Составить характеристику IT-инфраструктуры ГКУЗ РХ "Республиканский медицинский информационно-аналитический центр".

5. Сформулировать цель и задачи разработки проекта.

6. Обосновать выбор средств проектных решений.

7. Построить модели DFD и IDEF3 для реализации АИС передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении из РИАМС «ПроМед» в ИС ТФОМС РХ.

8. Описать БД шины для получения данных из РИАМС «ПроМед» и передачи данных в ИС ТФОМС РХ.

9. Разработать автоматизированную информационную систему для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ.

10. Оценить экономическую эффективность АИС.

1.6 Обоснование выбора средств для разработки автоматизированной информационной системы передачи сведений о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ

Автоматизированная информационная система (АИС) – совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации [4].

По способу организации информационные системы подразделяются на следующие классы:

- системы на основе архитектуры файл-сервер;
- системы на основе архитектуры клиент-сервер;
- системы на основе микросервисной архитектуры.

В архитектуре файл-сервер сетевое разделение компонентов диалога отсутствует, а компьютер используется для функций отображения, что облегчает построение графического интерфейса. Файл-сервер только

извлекает данные из файлов, так что дополнительные пользователи и приложения лишь незначительно увеличивают нагрузку на центральный процессор. Каждый новый клиент добавляет вычислительную мощность к сети.

Архитектура клиент-сервер предназначена для разрешения проблем файл-серверных приложений путем разделения компонентов приложения и размещения их там, где они будут функционировать наиболее эффективно. Особенностью архитектуры клиент-сервер является наличие выделенных серверов баз данных, понимающих запросы на языке структурированных запросов (Structured Query Language, SQL) и выполняющих поиск, сортировку и агрегирование информации.

Микросервисная архитектура – принципиальная организация распределенной системы на основе микросервисов и их взаимодействия друг с другом и со средой по сети, а также принципов, направляющих проектирование архитектуры, её создание и эволюцию [5]. Суть данной архитектуры в том, что каждый созданный сервис должен решать одну задачу.

Микросервисная архитектура содержит следующие принципы:

– Сервисы должны быть сфокусированы. Таким образом, из данного принципа следует, что каждый сервис выполняет свою единственную задачу.

– Сервис должен быть небольшим. Из этого принципа следует, что разработка сервиса не требует множества людей.

– Сервисы должны быть малосвязанными. В этом принципе описано то, что изменения в одном сервисе не должно влиять на другой.

Данная АИС будет разрабатываться с помощью микросервисной архитектуры, т.к. деление на небольшие сервисы значительно упростит разработку АИС.

Разрабатываемая АИС будет представлять из себя веб-приложение. Веб-приложение – это клиент-серверное приложение, в котором клиентом выступает браузер, а сервером – веб-сервер. Логика веб-приложения

распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами[6].

1.6.1 Выбор языка программирования для разработки

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. В данной разработке будет применяться серверный язык, т.к. работа таких программ полностью зависима от сервера, на котором расположена. Важная сторона работы серверных языков – возможность организации непосредственного взаимодействия с системой управления базами данных.

По данным сайта [octoverse.github](https://octoverse.github.io/), одними из самых популярных серверных языков являются Python, PHP, JavaScript. Охарактеризуем каждый из них и обоснуем выбор языка программирования для разработки АИС по передаче данных о застрахованных лицах, стоящих на диспансерном наблюдении, в ИС ТФОМС РХ. Рейтинг языков программирования представлен на рисунке 1.2.

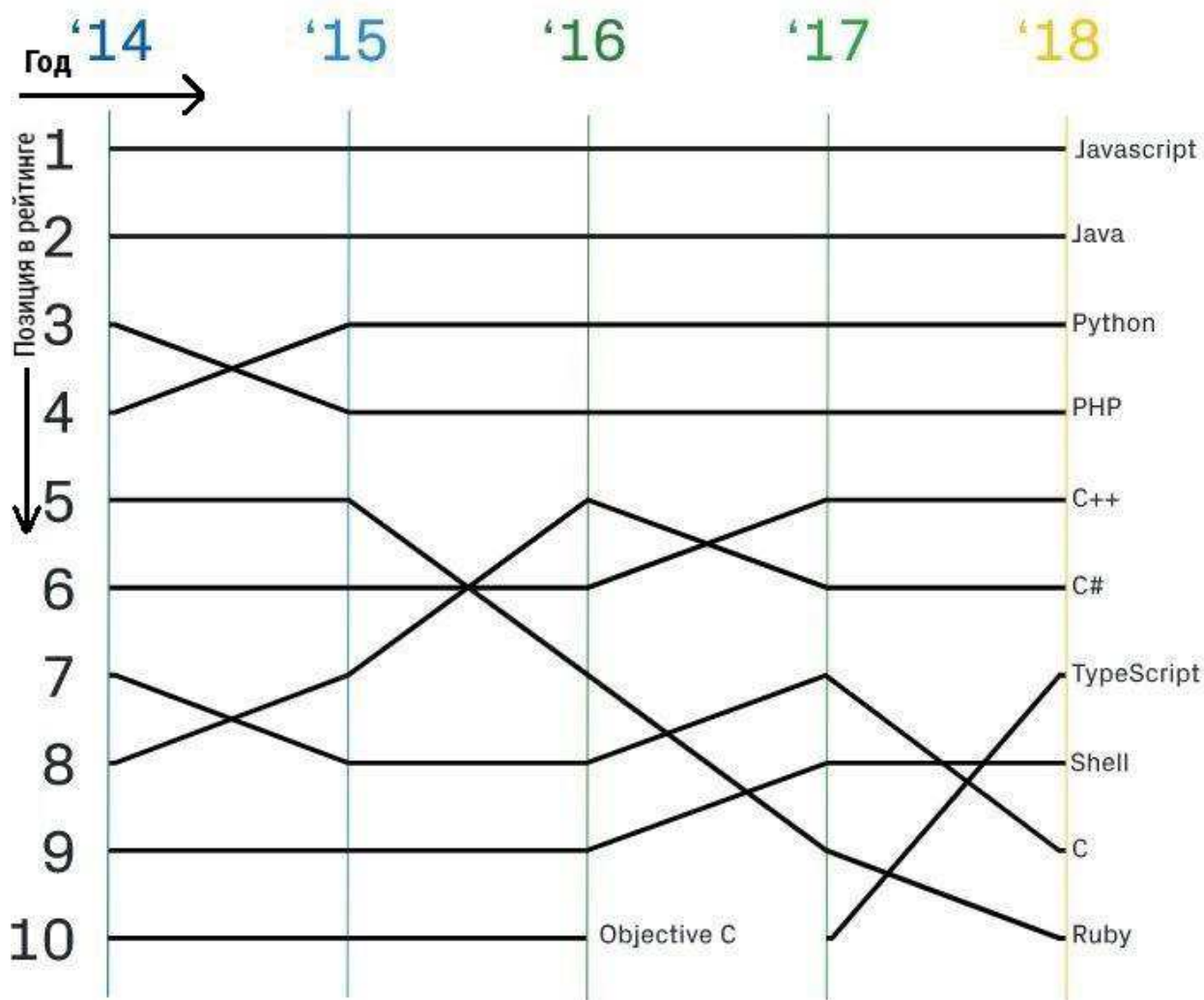


Рисунок 1.2 – Рейтинг языков программирования

Python – интерпретируемый язык программирования. С одной стороны, это позволяет значительно упростить отладку программ, с другой – обуславливает сравнительно низкую скорость выполнения. Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных [7].

Преимущества языка программирования Python:

– кроссплатформенность и бесплатность;

– простой синтаксис и богатые возможности позволяют записывать программы очень кратко, но в то же время понятно;

– по простоте освоения язык сравним с бейсиком, но куда более богат возможностями и значительно более современен;

– богатая стандартная библиотека, возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т.д.).

Недостатки языка программирования Python:

– ограниченность средств разработки с базами данных;

– низкое быстродействие;

– ограниченность средств для работы с базами данных.

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML.

Скрипты, написанные на языке PHP, обычно хранятся в файлах с расширением php, которые содержат в себе смесь обычных HTML-тэгов со специальной разметкой: открывающим тэгом `<?php` и закрывающим `?>`[7].

Преимущества языка программирования PHP:

– поддержка многих систем управления базами данных (СУБД) (MySQL, PostgreSQL, Sybase, Informix и других);

– доступность для большинства операционных систем (Linux, Unix, Microsoft Windows, MacOS и других);

– большое количество библиотек и расширений языка;

– прост в освоении.

Недостатки языка программирования PHP:

– не подходит для создания десктопных приложений или системных компонентов;

– веб-приложения, написанные на PHP, зачастую имеют проблемы с безопасностью.

JavaScript – язык сценариев, или скриптов. Скрипт представляет собой программный код – набор инструкций, который не требует предварительной обработки (например, компиляции) перед запуском. Код JavaScript интерпретируется движком браузера во время загрузки веб-страницы. Интерпретатор браузера выполняет построчный анализ, обработку и выполнение исходной программы или запроса [8].

Изначально JavaScript можно использовать только, как средство для клиентской стороны, однако с помощью Node.js JavaScript можно использовать и для серверной стороны. Node.js — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения[9].

Преимущества языка программирования JavaScript с использованием Node.js:

- возможность работы как с клиентской стороны, так и с серверной стороны;
- очень высокая производительность;
- встроенная система управления микросервисами и зависимостями;
- прост в освоении.

Недостатки языка программирования JavaScript с использованием Node.js:

- молодость системы сказывается на том, что поддержка со стороны пользователей пока существенно ниже, чем у более старших серверных «братьев»;
- асинхронность кода может сильно его усложнить.

В итоге, исходя из сделанного сравнения, можно сказать, что Python, несмотря на множество плюсов, очень сильно проигрывает в

производительности рассмотренным в данном разделе языком программирования. Исходя из этого, Python не будет использоваться в разработке, автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ. Переходя к PHP и JavaScript стоит сказать, что несмотря на то, что Node.js довольно молодая платформа, в создании АИС будет использоваться именно она по причинам того, что PHP проигрывает в безопасности и производительности. Таким образом, для разработки АИС был выбран язык программирования JavaScript.

1.6.2 Выбор системы управления базами данных для разработки

Одним из самых важных элементов при создании данной АИС будет являться база данных, которая будет хранить в себе передаваемые данные. Поэтому следует выяснить, что такое системы управления базами данных (СУБД).

СУБД позволяют структурировать, систематизировать и организовывать данные для их компьютерного хранения и обработки. Именно базы данных являются основой практически любой информационной системы.

Взаимодействие пользователя с БД обеспечивает специальный язык запросов – Structured Query Language (SQL) (Язык структурированных запросов).

Существует множество СУБД для управления реляционными базами данных. Каждая из них обладает своими достоинствами и недостатками. Рассмотрим несколько СУБД, которые широко используются в интернете: SQLite, PostgreSQL, MariaDB.

SQLite – это, легко встраиваемая в приложения, база данных. Так как эта система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При

работе с этой СУБД обращения происходят напрямую к файлам (в этих файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная, благодаря технологиям обслуживающих библиотек [10].

Преимущества SQLite:

- вся база данных хранится в одном файле, что облегчает перемещение;

- стандартный функционал SQL.

Главным недостатком SQLite для разрабатываемой системы является то, что данная СУБД не поддерживает формат данных XML. Это является критически важным недостатком данной системы, т.к. РИАМС «ПроМед» пересылает данные в формате XML.

PostgreSQL – свободно распространяемая СУБД с поддержкой объектно-ориентированного подхода к данным. Реализован принцип параллельной обработки. За счет использования встроенных хранимых процедур ускорено использование повторяемых процессов. СУБД отлично справляется и во многом ориентирована на работу со сложными структурами данных – по сравнению с другими свободно распространяемыми СУБД.

Достоинства PostgreSQL:

- открытое ПО, соответствующее стандарту SQL – PostgreSQL – бесплатное ПО с открытым исходным кодом;

- большое количество дополнений;

- расширения – существует возможность расширения функционала за счет сохранения своих процедур;

- объектность – PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования.

Недостатки PostgreSQL:

– производительность – при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов, таких как MariaDB;

– малоизвестность.

MariaDB – ответвление СУБД MySQL, разрабатываемое сообществом. Основная цель проекта MariaDB – создание полностью бинарно совместимой с оригинальной MySQL версии СУБД, которая при этом будет иметь значительное количество улучшений в коде, влияющих на производительность [11].

Преимущества MariaDB:

- высокая производительность и безопасность;
- может работать с действительно большими объёмами данных, и неплохо подходит для масштабируемых приложений;
- поддерживает большую часть функционала SQL;
- высокая стабильность.

Недостатки MariaDB:

– более низкая поддержка со стороны сообщества, в связи с небольшой популярностью.

Согласно проведенному анализу СУБД, создание БД будет осуществляться с помощью СУБД MariaDB, преимуществом которой является ее производительность и безопасность, а также простота в использовании.

Также в проекте будет использоваться HTML и CSS, которым не существует альтернатив.

Таким образом, для разработки автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, будет использоваться JavaScript, Node.js, СУБД MariaDB, а также HTML и CSS.

Выводы по разделу «Анализ предметной области. Выбор средств проектных решений»

В данном разделе проанализирована основная деятельность учреждений ГКУЗ РХ «РМИАЦ» и ТФОМС РХ, которая позволила выявить, что РИАМС «ПроМед» и ИС ТФОМС РХ используют разные форматы записи данных.

Данные будут передаваться по защищенной сети в соответствии с Федеральным законом РФ от 27 июля 2006 года № 152-ФЗ «О персональных данных».

Для реализации автоматизированной информационной системы передачи сведений о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ, были проанализированы наиболее популярные средства в среде современных IT-разработчиков:

- языки программирования: Python, PHP и JavaScript;
- СУБД: Microsoft SQL, SQLite, PostgreSQL и MariaDB.

Был выбран язык программирования JavaScript.

Согласно проведенному анализу СУБД создание БД будет осуществляться с помощью MariaDB, преимуществом которой является ее производительность и безопасность, а также простота в использовании. Также в проекте будут использоваться HTML и CSS, которым не существует альтернатив.

2 Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ

2.1 Описание модели реализации автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении из РИАМС «ПроМед» в ИС ТФОМС РХ по нотациям DFD и IDEF3

Бизнес-процесс – последовательность действий (подпроцессов), направленная на получение заданного результата, ценного для организации [12].

Для более лучшего понимания автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ в виде модели использовалась методология графического структурного анализа DFD, позволяющая описать внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. На данный момент в ГКУЗ РХ «РМИАЦ» нет информационной системы, которая позволила бы автоматически передавать данные о застрахованных лицах, стоящих на диспансерном наблюдении в ИС ТФОМС РХ.

Следует рассмотреть некоторые понятия из методологии DFD. Процесс (англ. Process), т.е. функция или последовательность действий, которые нужно предпринять, чтобы данные были обработаны. Это может быть создание заказа, регистрация клиента и т.д. В названиях процессов принято использовать глаголы, т.е. «Создать клиента» (а не «создание клиента») или «обработать заказ» (а не «проведение заказа»). Здесь нет строгой системы требований, как, например, в IDEF0, где нотации имеют жестко определенный синтаксис, так как они могут быть исполняемыми.

Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных. Это может быть человек, внешняя система, какие-либо носители информации и хранилища данных.

Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе. Поступившие данные перед обработкой и результат после обработки, а также промежуточные значения должны где-то храниться. Это и есть базы данных, таблицы или любой другой вариант организации и хранения данных. Здесь будут храниться данные о клиентах, заявки клиентов, расходные накладные и любые другие данные, которые поступили в систему или являются результатом обработки процессов.

Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме [13]. Синтаксис диаграмм DFD в двух вариантах представлен на рисунке 2.1.









Нотация	Юрдан и Коад	Гейн и Сарсон
Внешняя сущность		
Процесс		
Хранилище данных		
Поток данных		

Рисунок 2.1 – Синтаксис диаграммы DFD

В данном случае будет построена модель «как будет» в варианте Гейна-Сарсона.

Контекстная диаграмма процесса передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, в ИСТФОМС РХ представлена на рисунке 2.2.

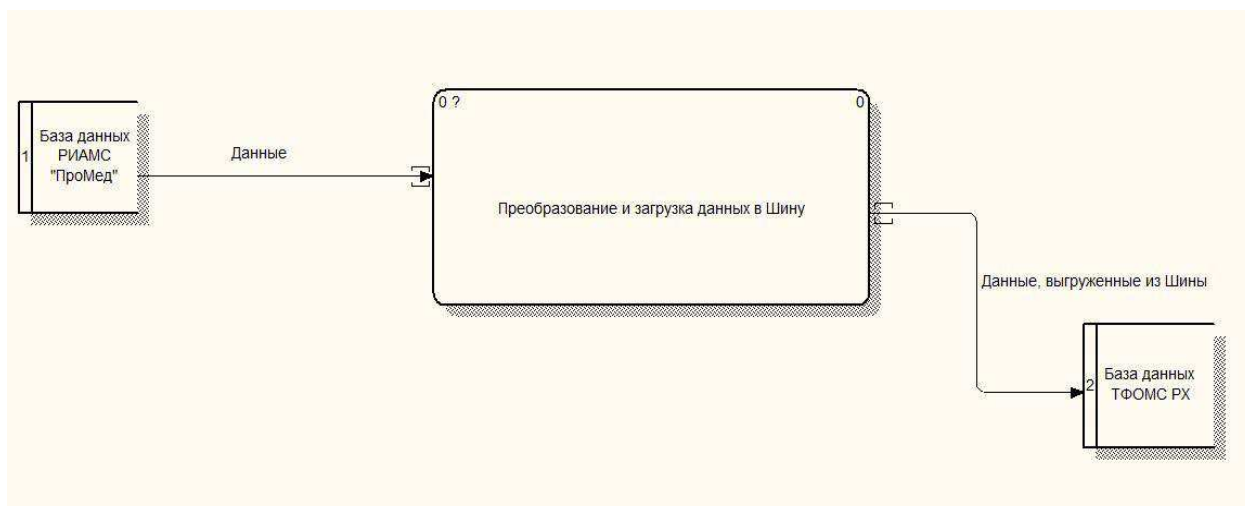


Рисунок 2.2 – Контекстная диаграмма передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, в ИСТФОМС РХ

Как видно из рисунка, контекстная диаграмма представляет из себя 2 внешних сущности (База данных РИАМС «ПроМед», База данных ТФОМС РХ) и 1 блок процесса (Преобразование и загрузка данных в Шину). Шина – это АИС, которая позволяет соединять поток данных, исходящих из РИАМС «ПроМед» в ИС ТФОМС РХ.

В результате выполнения декомпозиции блока процесса контекстной диаграммы были определены основные процессы передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, в ИС ТФОМС РХ, представленные на рисунке 2.3.

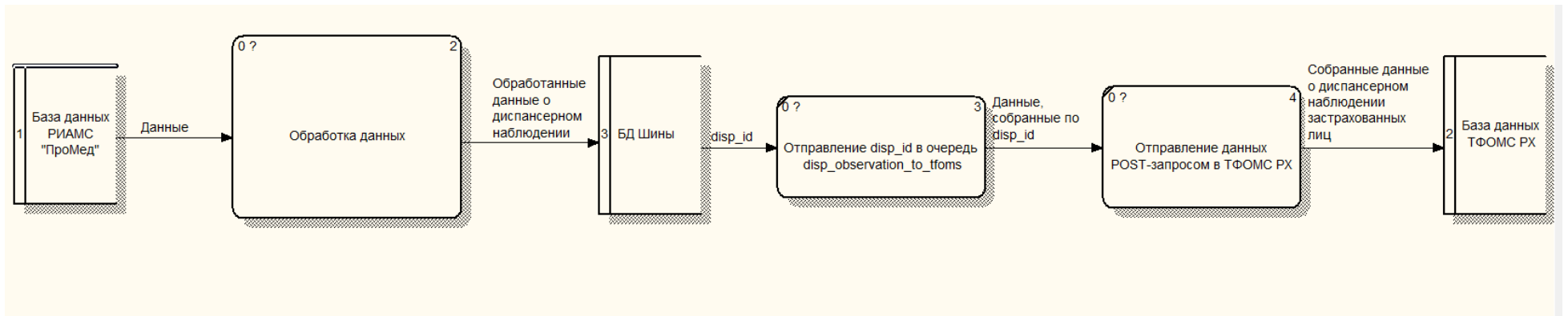


Рисунок 2.2 – Декомпозиция блока процесса «Преобразование и загрузка данных в Шину» по методологии DFD

Из рисунка видно, что из блока процесса «Преобразование и загрузка данных в Шину» появилась ещё одна внешняя сущность (Шина) и такие блоки процессов, как: обработка данных, отправление `disp_id` (атрибут, в котором указан идентификатор карты диспансерного наблюдения) в очередь `disp_observation_to_tfoms`, отправление данных POST-запросом в ИС ТФОМС РХ. Теперь нужно декомпозировать блок процесса «Обработка данных». Данная декомпозиция, которая показана на рисунке 2.3, будет представлена по методологии IDEF3.

IDEF3 – методология моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. IDEF3 дает возможность аналитикам описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.

Любая IDEF3-диаграмма может содержать работы, связи, перекрестки и объекты ссылок.

Работа (Unit of Work, activity) изображается прямоугольником с прямыми углами и имеет имя, выраженное отглагольным существительным, обозначающим процесс действия, одиночным или в составе фразы, и номер (идентификатор); другое имя существительное в составе той же фразы обычно отображает основной выход (результат) работы. Все стороны работы равнозначны. В каждую работу может входить и выходить ровно по одной стрелке.

Связи показывают взаимоотношения работ. Все связи в IDEF3 однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо.

Перекрестки (Junction). Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для

отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления [14].

Типы перекрестков представлены в таблице 2.1.

Таблица 2.1 – Типы перекрёстков

Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Асинхронное «И» (Asynchronous AND)	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Синхронное «И» (Synchronous AND)	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Асинхронное «ИЛИ» (Asynchronous OR)	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Синхронное «ИЛИ» (Synchronous OR)	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
	Исключающее «ИЛИ» XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

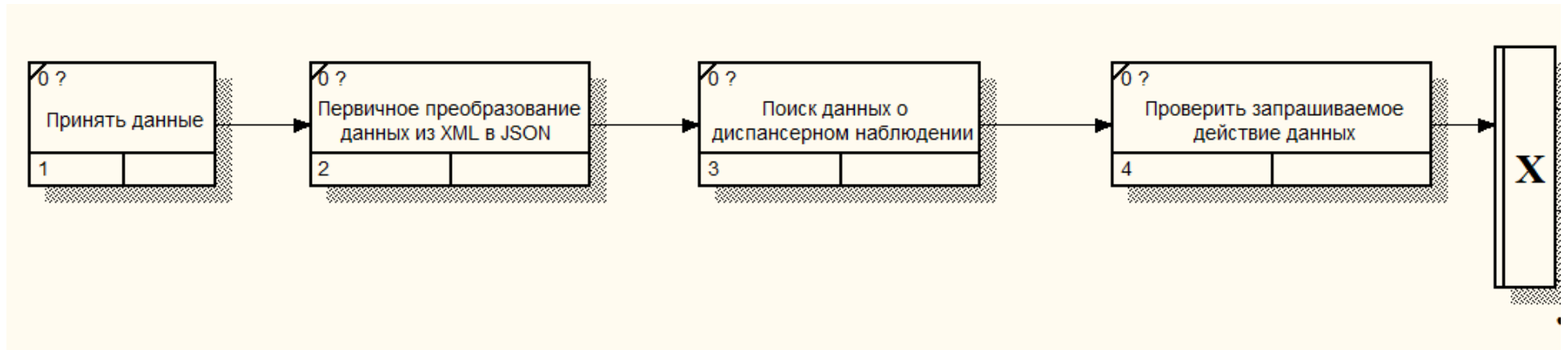


Рисунок 2.3 – Декомпозиция блока процесса «Обработка данных», лист 1

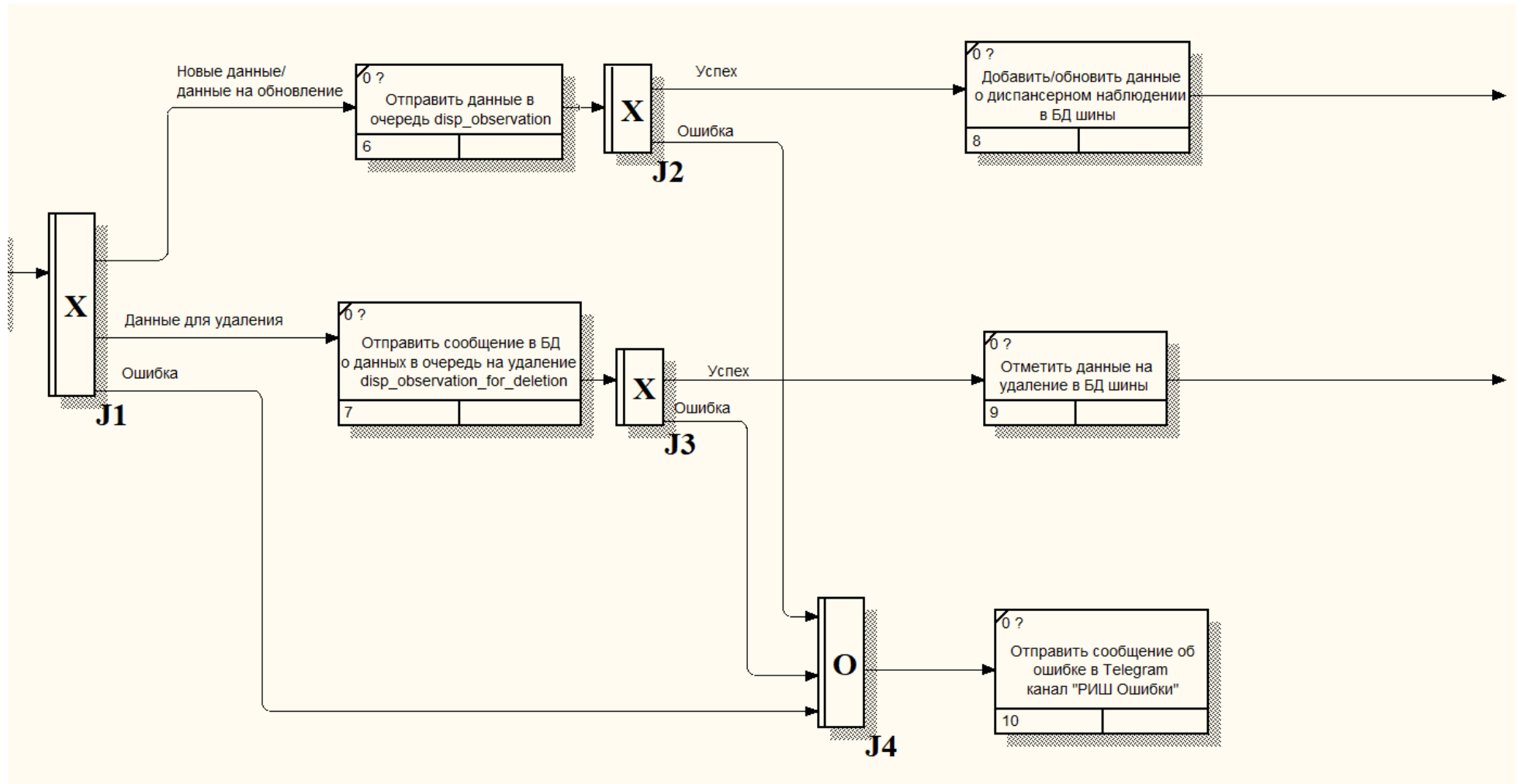


Рисунок 2.3, лист 2

Исходя из рисунка, в построенной диаграмме используются четыре перекрестка. Три исключают «ИЛИ» и одно асинхронное «ИЛИ». Следует раскрыть процесс передачи сведений подробнее.

Передача данных о застрахованных лицах, стоящих на диспансерном наблюдении, в ИС ТФОМС РХ начинается с внесения различных данных о пациентах, которые вносятся врачом в РИАМС «ПроМед».

После этого сообщение с данными отправляется в очередь «test_input». Они изымаются из очереди, преобразуются из формата XML в JSON, чтобы пройти сравнение. После попадают под фильтрацию, в ходе которой ведется поиск данных, связанных с диспансерным наблюдением. Также смотрятся, что с ними нужно сделать: удалить, добавить, обновить. Если возникает ошибка, то отправляется специальное оповещение в Telegram об ошибке. Данные с действием «удалить» отправляются в специальную очередь disp_observation_for_deletion. А после отмечаются данные в БД, которые нужно удалить. Если возникает ошибка, то отправляется специальное оповещение в Telegram об ошибке. Данные с действиями «добавить» или «обновить» отправляются в очередь disp_observation, с помощью которой данные отправляются в БД. При возникновении ошибки, также отправляется оповещение в Telegram.

После добавления/обновления данных в базу или отметки об удалении, идентификаторы (id) данных о диспансерном наблюдении застрахованных лиц отправляются в специальную очередь для отправки данных в ИС ТФОМС РХ. После этого микросервис disp-observation-totfoms просматривает эту очередь идентификаторов, преобразует сведения в JSON и отправляет запрос на их отправку. Микросервис disp-observation-totfoms предназначен для того, чтобы данные отправлялись в ИС ТФОМС РХ. В результате, данные о застрахованных лицах, стоящих на диспансерном наблюдении, пересылаются в ИС ТФОМС РХ.

Таким образом, исходя из построенной модели бизнес-процессов по методологиям DFD и IDEF3, можно понять, как именно будет происходить

передача данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

2.2 Описание БД шины для получения данных из РИАМС «ПроМед» и передачи данных в ИС ТФОМС РХ

На рисунке 2.4 представлено отношение «Диспансерное наблюдение».

Column	Type	Comment
id	int(11) <i>Auto Increment</i>	
bdz_id	int(11)	
updated_at	timestamp [<i>current_timestamp()</i>]	
DS	varchar(10)	
DS_DETECT	int(11) <i>NULL</i>	
DS_DETECTTYPE	int(11) <i>NULL</i>	
SNILS_VR	varchar(11) <i>NULL</i>	
DATE_OUT	date <i>NULL</i>	
RESULT_OUT	int(11) <i>NULL</i>	
DATES	varchar(2048) <i>NULL</i>	
CODE_MO	int(6)	
created_at	date	
promed_pacient_id	int(11)	
DISP_ID	int(11)	
DATE_IN	date <i>NULL</i>	
SEND_TO_TFOMS_STATUS_id	int(11) [0]	

Indexes

PRIMARY	<i>id</i>
INDEX	<i>bdz_id</i>

Рисунок 2.4 – Отношение «Диспансерное наблюдение»

Описание атрибутов сущностей отношения представлено в таблице 2.2.

Таблица 2.2 – Описание атрибутов сущностей

Атрибуты	Сущности	Примечание
CODE_MO	Код МО отправителя	Федеральный реестровый код МО (LPU_f003mcod) формы Паспорт МО
DISP_ID	Идентификатор карты диспансерного наблюдения	Person Disp_id
BDZ_ID	Идентификатор человека	Идентификатор человека в ТФОМСРХ
DATE_IN	Дата постановки на диспансерное наблюдение	Взят (PersonDisp_begDate) формы «Контрольные карты диспансерного наблюдения»
DS_DETECT	Диагноз установлен впервые или ранее	
DS_DETECTTYPE	Диагноз установлен в ходе лечения или при профосмотре	
SNILS_VR	СНИЛС ответственного врача	СНИЛС (Person_SNILS) из прикрепления пациента
DATE_OUT	Дата снятия с диспансерного наблюдения	Снят (PersonDisp_endDate) формы «Контрольные карты диспансерного наблюдения» Выгружается только в случае, если месяц и год снятия с диспансерного наблюдения не больше отчетных месяца и года (вводимых на форме «Выгрузка карт диспансерного наблюдения»)
DATES	Плановые даты визитов	
ID	Первичный ключ	
UPDATED_AT	Дата обновления	
RESULT_OUT	Причина снятия с диспансерного наблюдения	Причина снятия (DispOutType) формы «Контрольные карты диспансерного наблюдения» 1.Выздоровление – при выгрузке значение «1». 2.Переезд – при выгрузке значение «2». 3.Смерть – при выгрузке значение «3». 4.Перевод в другую группу – при выгрузке значение «4». 5.Не наблюдался – при выгрузке значение «4»

Окончание таблицы 2.2

PROMED_ PACIENT_ID	Номер пациента в базе данных РИАМС «ПроМед»	
CREATED_AT	Дата создания	
SEND_TO_ TFOMS_ STATUS_ID	Присвоенный номер для передачи в ИС ТФОМС РХ	

Исходя из данных, представленных в данном подразделе, можно сделать вывод о том, какие именно данные будут переданы в БД шины.

2.3 Разработка автоматизированной информационной системы для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ

Для разработки АИС передачи данных из РИАМС «ПроМед» в БД шины необходимо реализовать следующие микросервисы:

- принятие информации из базы данных РИАМС «ПроМед»;
- преобразование данных из формата XML в JSON;
- загрузка преобразованных данных в БД шины;
- отправление данных из БД шины в ИС ТФОМС РХ.

На рисунке Д.1 в приложении Д представлен рисунок с блок-схемой, созданной АИС

2.3.1 Разработка микросервиса принятия информации из базы данных РИАМС «ПроМед»

Для того, чтобы создать микросервис принятия информации из базы данных РИАМС «ПроМед», первое, что требуется сделать, это начать «слушать» очередь данных, поставляемых из РИАМС «ПроМед». После «прослушивания» идет принятие данных из РИАМС «ПроМед» в строковом

виде и преобразование этих данных из формата XML в JSON. Однако для преобразования данных требуется создать отдельный микросервис. Код микросервиса принятия информации из базы данных РИАМС «ПроМед» представлен в приложении Д.

2.3.2 Разработка микросервиса преобразования данных из формата XML в JSON

Для того, чтобы создать микросервис преобразования данных из формата XML в JSON, сначала нужно разобраться в том, что это за форматы данных.

XML-файлы представляют собой файлы, используемые в различных приложениях, поскольку они относятся к файлам, которые написаны в Extensible Markup Language. Это формат, который содержит теги и использует его для определения объекта, а также атрибуты. XML-файлы используются для хранения и передачи данных в Интернете, которые приходят из разных программ и могут быть отредактированы в любом текстовом редакторе, потому что они приходят в текстовых документах[15]. Пример XML-файла представлен на рисунке 2.5.

JSON (JavaScriptObjectNotation) –простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка программирования JavaScript, определенного в стандарте ECMA-262 3rd Edition–December 1999. JSON – текстовый формат, полностью независимый от языка реализации, но он использует соглашения, знакомые программистам C-подобных языков, таких как C, C++, C#, Java, JavaScript, Perl, Python и многих других. Эти свойства делают JSON идеальным языком обмена данными.


```

FIRST_DISP_OBSERVATION_XML = `<?xml version="1.0" encoding="utf-8"?>
<DISP>
  <HEADER>
    <OPERATIONTYPE>Insert</OPERATIONTYPE>
    <CODE_MO>190036</CODE_MO>
    <DATA>2019-05-05</DATA>
    <DISP_ID>3562395</DISP_ID>
  </HEADER>
  <BODY>
    <BDZ_ID>1</BDZ_ID>
    <ID_PAC>3562395</ID_PAC>
    <FAM>БОЛДЫРЕВ</FAM>
    <IM>ИГОРЬ</IM>
    <OT>ВИКТОРОВИЧ</OT>
    <W>1</W>
    <DR>1956-03-20</DR>
    <DATE_IN>2003-09-15</DATE_IN>
    <DS>Y52</DS>
    <SNILS_VR>03654778654</SNILS_VR>
    <DATES>
      <PLAN_DATE>2019-10-05</PLAN_DATE><PLAN_DATE>2019-12-15</PLAN_DATE>
    </DATES>
  </BODY></DISP>`

```

Рисунок 2.5 – XML-файл

JSON основан на двух структурах данных:

- Коллекция пар ключ/значение. В разных языках, эта концепция реализована как объект, запись, структура, словарь, хэш, именованный список или ассоциативный массив.

- Упорядоченный список значений. В большинстве языков это реализовано как массив, вектор, список или последовательность.

Это универсальные структуры данных. Почти все современные языки программирования поддерживают их в какой-либо форме. Логично предположить, что формат данных, независимый от языка программирования, должен быть основан на этих структурах [16]. Пример JSON-файла представлен на рисунке 2.6.

Данное преобразование необходимо в связи с тем, что форматы данных у РИАМС «ПроМед» (XML) и ТФОМС РХ (JSON) не совпадают. Для преобразования из формата данных XML в формат JSON, нужно подключить библиотеку `xml2json`. После этого объявить переменную, в которую будет записан текст в формате XML. После этого нужно создать переменную, в которой совершается преобразование данных из XML в JSON с помощью

функции `parser.toJson`. Таким образом, в итоге данные будут преобразованы из формата XML в формат JSON. На рисунке E.1 показана блок-схема алгоритма `xml2json` в приложении E.

```
FIRST_DISP_OBSERVATION_JSON
{
  "DISP": {
    "HEADER": {
      "OPERATIONTYPE": "Insert",
      "CODE_MO": "190036",
      "DATA": "2019-05-05",
      "DISP_ID": "3562395"
    },
    "BODY": {
      "BDZ_ID": "1",
      "ID_PAC": "3562395",
      "FAM": "БОЛДЫРЕВ",
      "IM": "ИГОРЬ",
      "OT": "ВИКТОРОВИЧ",
      "W": "1",
      "DR": "1956-03-20",
      "DATE_IN": "2003-09-15",
      "DS": "Y52",
      "SNILS_VR": "03654778654",
      "DATES": {
        "PLAN_DATE": [
          "2019-10-05",
          "2019-12-15"
        ]
      }
    }
  }
}
```

Рисунок 2.6 – JSON-файл

2.3.3 Разработка микросервиса загрузки преобразованных данных в БД шины

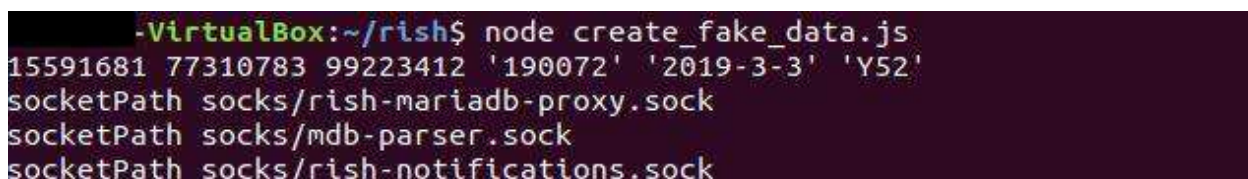
Микросервис загрузки преобразованных данных в БД шины будет показана на примере создания микросервиса фиктивных данных из РИАМС «ПроМед». Модуль генератора фиктивных данных из РИАМС «ПроМед» представлен в приложении Б.

Для создания микросервиса генератора фиктивных данных из РИАМС «ПроМед» необходимо реализовать следующие микросервиса:

- Генератор следующих атрибутов:BDZ_ID, DS, CODE_MO, CREATED_AT, PROMED_PACIENT_ID, DISP_ID;

- загрузка данных в БД шины.

Для того, чтобы запустить данный модуль нужно в терминале Linuxнаписать в репозитории, где хранится файл данного микросервиса, node create_fake_data.js. На рисунке 2.7 показана работа этого микросервиса.



```
-VirtualBox:~/rish$ node create_fake_data.js
15591681 77310783 99223412 '190072' '2019-3-3' 'Y52'
socketPath socks/rish-mariadb-proxy.sock
socketPath socks/mdb-parser.sock
socketPath socks/rish-notifications.sock
```

Рисунок 2.7 – Демонстрация работы микросервиса генератора ненастоящих преобразованных данных из РИАМС «ПроМед»

После этого следует зайти в базу данных, чтобы посмотреть прошла ли запись данных в базу. На рисунке 2.9 показана база данных со сгенерированными данными.

Далее данные из атрибута DISP_ID передаются в очередь на отправление в ИС ТФОМС РХ.

DISPS

Show structure Alter table New item

Search Sort Limit 50 Text length 100 Action Select

FROM `DISPS` LIMIT 50 (0.002 s) Edit

id	bdz_id	updated_at	DS	DS_DETECT	DS_DETECTTYPE	SNILS_VR	DATE_OUT	RESULT_OUT	DATES	CODE_MO	created_at	promed_pacient_id	DISP_ID	DATE_IN	SEND_TO_T
5	15591681	2019-04-09 01:44:10	Y52	NULL	NULL	NULL	NULL	NULL	NULL	190072	2019-03-03	77310783	99223412	NULL	0
6	27012468	2019-04-10 02:16:38	C02	NULL	NULL	NULL	NULL	NULL	NULL	190030	2019-03-04	3912219	41072993	NULL	0
7	24380539	2019-04-10 02:16:40	J09	NULL	NULL	NULL	NULL	NULL	NULL	190004	2019-03-04	32734942	47007434	NULL	0
8	35905501	2019-04-10 02:16:42	B80	NULL	NULL	NULL	NULL	NULL	NULL	190045	2019-03-04	32523390	25149384	NULL	0
9	87544019	2019-04-10 02:16:44	Z91	NULL	NULL	NULL	NULL	NULL	NULL	190037	2019-03-04	42727902	57227849	NULL	0

Рисунок 2.8 – База данных со сгенерированными данными

2.3.4 Разработка микросервиса передачи данных из БД шины в ИСТФОМС РХ

Для того чтобы создать микросервис передачи данных из БД шины в ИСТФОМС РХ нужно, чтобы началось «прослушивание» уже созданной в предыдущем подразделе очереди, состоящей из данных в атрибуте DISP_ID. После этого полученные данные отправляются POST-запросом на специальный URL-адрес. Написанный код микросервиса представлен в приложении В.

2.4 Тестирование

Тестирование системы состояло в том, чтобы введенные сведения о застрахованных лицах, стоящих на диспансерном наблюдении, проходили проверку на совпадение с уже вписанными в БД шины. В проверяемых данных должны быть такие записи, как: даты плановых посещений, диагноз, СНИЛС ответственного врача, диагноз установлен впервые или ранее; диагноз установлен в ходе лечения или при профпросмотре; дата постановки на диспансерное наблюдение; причина снятия с диспансерного наблюдения; код МО отправителя.

Итоговый результат показан на рисунках 2.9 и 2.10.

```
В базе должно быть
✓ Даты плановых посещений (DATES) = DatesEq
✓ Диагноз (DS) = I11.2
✓ СНИЛС ответственного врача (SNILS_VR) = 063-956-107 11
✓ Дата снятия с диспансерного наблюдения (DATE_OUT) = null
✓ Диагноз установлен впервые или ранее (DS_DETECT) = null
✓ Диагноз установлен в ходе лечения или при профосмотре (DS_DETECTTYPE) = null
✓ Дата постановки на диспансерное наблюдение (DATE_IN) = 27.08.2001
✓ Причина снятия с диспансерного наблюдения (RESULT_OUT) = null
✓ Код МО отправителя (Код МО отправителя) = 190006
```

Рисунок 2.9 – Пройденная проверка данных, которые должны быть в базе

```
Должны измениться
✓ Даты плановых посещений (DATES) = EqDates
✓ Диагноз (DS) = I11.3
✓ СНИЛС ответственного врача (SNILS_VR) = 063-956-107 21
✓ Дата снятия с диспансерного наблюдения (DATE_OUT) = null
✓ Диагноз установлен впервые или ранее (DS_DETECT) = 1
✓ Диагноз установлен в ходе лечения или при профосмотре (DS_DETECTTYPE) = 1
✓ Дата постановки на диспансерное наблюдение (DATE_IN) = 27.08.2001
✓ Причина снятия с диспансерного наблюдения (RESULT_OUT) = 3
✓ Код МО отправителя (Код МО отправителя) = 190010
```

Рисунок 2.10 – Пройденная проверка данных, которые должны измениться

В итоге, все введенные данные прошли проверку на совпадение с уже вписанными в БД данными.

Выводы по разделу «Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ»

В проектном разделе была построена модель по методологиям моделирования DFD и IDEF3 для того, чтобы понять, как будут передаваться данные о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

Была разработана АИС для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ, которая включила в себя создание таких микросервисов, как:

- принятие информации из базы данных РИАМС «ПроМед»;
- преобразование данных из формата XML в JSON;
- загрузка преобразованных данных в БД шины;
- отправление данных из БД шины в ИС ТФОМС РХ.

Также был разработан модуль генератора преобразованных фиктивных данных из РИАМС «ПроМед».

После этого были проведены тесты на совпадение сведений с уже вписанными в БД.

3 Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ

3.1 Технические характеристики проекта

Для того, чтобы провести расчет затрат реализации автоматизированной информационной системы для обмена информацией между базами данных ТФОМС РХ и РИАМС «ПроМед», для начала, нужно разобраться с тем, кто именно будет связан с создаваемой системой и с помощью каких устройств он будет взаимодействовать с этой системой, а также какое программное обеспечение (ПО) необходимо для разработки АИС.

Чтобы показать кто связан с системой, нужно построить таблицу 3.1, в которой будет описана деятельность персонала.

Таблица 3.1 – Персонал, связанный с проектом

Должность	Заработная плата, руб.	Продолжительность работы в проекте
Инженер-программист	13 050	1,5 месяца

Исходя из таблицы видно, что только один человек связан с системой, поэтому для данного человека нужно определить стоимость оборудования для работы, которая представлено в таблице 3.2. Также этот человек будет сопровождать автоматизированную информационную систему для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ после внедрения.

Таблица 3.2 – Стоимость оборудования

Наименование оборудования	Стоимость, руб.	Срок эксплуатации
Компьютер (IntelCorei3 3460, 8 GBDDR3, 120gbSSD)	18 750	5 лет
Монитор 23.8" AOC I2480SX	7 899	5 лет
Дополнительное оборудование	1000	5 лет
Итого	27 649	5 лет

В АИС для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ используется такое ПО, как: операционная система Linux, Node.js, MariaDB, Git, RabbitMQ, Docker. Всё ПО распространяется бесплатно.

3.2 Расчет затрат реализации проекта разработки ИС

Полученные данные из подраздела 3.1 дают возможность рассчитать ТСО (совокупную стоимость владения). Для расчета затрат реализации проекта была выбрана методика ТСО, которая рассчитывается по формуле:

$$ТСО = DE + IC_1 + IC_2, \quad (1)$$

где DE – прямые затраты;

IC_{1,2} – косвенные расходы первой и второй группы, но в данном проекте они отсутствуют.

3.2.1 Капитальные затраты

Расчет капитальных затрат происходит по следующей формуле:

$$K = K_{\text{пр}} + K_{\text{тс}} + K_{\text{лс}} + K_{\text{по}} + K_{\text{ио}} + K_{\text{об}} + K_{\text{оэ}}, \quad (2)$$

где $K_{\text{пр}}$ – затраты на разработку информационной системы;

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

$K_{\text{лс}}$ – затраты на создание линий связи, а также интернет соединений;

$K_{\text{по}}$ – затраты на программные средства для использования готового программного продукта;

$K_{\text{ио}}$ – затраты на формирование информационной базы.

$K_{\text{об}}$ – заработная плата работника, обучающего персонал работать с новым программным продуктом;

$K_{\text{оэ}}$ – состав затрат, соответствующих эксплуатационным затратам.

Расчет затрат на разработку информационной системы ($K_{\text{пр}}$) происходит по следующей формуле:

$$K_{\text{пр}} = K_{\text{зп}} + K_{\text{ипс}} + K_{\text{свт}} + K_{\text{проч}}, \quad (3)$$

где $K_{\text{зп}}$ – затраты на заработную плату инженера;

$K_{\text{ипс}}$ – затраты на инструментальные программные средства для проектирования;

$K_{\text{свт}}$ – затраты на средства вычислительной техники для проектирования;

$K_{\text{проч}}$ – прочие затраты на разработку;

Затраты на заработную плату разработчиков.

Для расчета заработной платы проектировщиков ($K_{\text{зп}}$) нужно составить таблицу заработной платы и результат умножить на продолжительность работы

в проекте и умножить получившуюся сумму на ФОТ (Фонд оплаты труда), который составляет 32%. Расчет зарплаты проектировщиков показан на таблице 3.3.

Районный и северный коэффициенты в сумме равны 60%, т.к. разработка ведётся в Республике Хакасия. Рабочая норма инженеров в месяц составляет 22 дня.

Таблица 3.3 – Расчет заработной платы разработчика

Код	Начисление/Удержание	Начислено	Удержано
003	Оклад	13 050	
181	Районный коэффициент + северный коэффициент	7 830	
202	НДФЛ		2 714
Итого		20880	2 714

Итого: $K_{зп} = 20880 \cdot 1,302 \cdot 1,5 = 40\,778$ (руб.)

Затраты на инструментальные ПО для проектировщика ($K_{ипс}$), исходя из раздела 3.1 составляют 0 руб.

Затраты на средства вычислительной техники для проектирования ($K_{свт}$), ввиду наличия всего необходимого оборудования для разработки системы на рабочем месте проектировщика, включают лишь сумму амортизационных отчислений за период создания проекта ($A_{пэвм}$) и равняются произведению амортизационных отчислений в день на количество дней эксплуатации компьютера при создании системы.

$$A_{г} = C_{б} \cdot N_{ам}, \quad (4)$$

где $A_{г}$ – сумма годовых амортизационных отчислений;

C_6 – балансовая стоимость компьютера, р./шт.;

$N_{ам}$ – норма амортизации, %.

$$A_r = 27\,649 \cdot 0,20 = 5\,530 \text{ (руб.)}$$

$$A_{пэвм} = (5\,530 \cdot 45)/264 = 943 \text{ (руб.)}$$

$$\text{Итого: } K_{свт} = 943 \text{ (руб.)}$$

Прочие затраты на разработку ($K_{проч}$) обычно представляет из себя резерв на непредусмотренные расходы, который составляет 3% от общей суммы затрат на разработку. Такие расходы формируются из плат за свет, интернет.

$$\text{Итого: } K_{проч} = (40\,778 + 943) \cdot 3\% = 1\,252 \text{ (руб.)}$$

Затраты на разработку, которые рассчитываются по формуле (3) информационной системы равны:

$$K_{пр} = 40\,778 + 943 + 1\,252 = 42\,973 \text{ (руб.)}$$

Затраты на технические средства управления ($K_{тс}$) отсутствуют, потому что разрабатываемая АИС функционирует отдельно от других систем, использующихся на предприятии.

Затраты на создание линий связи локальных сетей ($K_{лс}$) исключаются, так как на предприятии функционирует собственная локальная сеть, удовлетворяющая новым требованиям с учетом работы разработанной системы.

Затраты на программные средства ($K_{по}$) не учитываются, так как все используемые для разработки системы программные средства являются свободно распространяемыми.

Затраты на формирование информационной базы ($K_{но}$) включают в себя заработную плату проектировщика за время создания информационной базы (5 дней – 1 рабочая неделя), а также стоимость программного обеспечения для её создания. Программное обеспечение для создания информационной базы не

учитывается, так как оно является свободно распространяемым. Таким образом данный вид затрат составляет:

Итого: $K_{\text{ио}} = 20880/4 = 5\,220$ (руб.)

Затраты на обучение персонала ($K_{\text{об}}$) не учитываются, т.к. автоматизированная информационная система для обмена данными о застрахованных лицах, стоящих на диспансерном наблюдении между базами данных ТФОМС РХ и РИАМС «ПроМед» в будущем будет сопровождаться самим разработчиком.

Затраты на опытную эксплуатацию ($K_{\text{оэ}}$) включают в себя заработную плату проектировщика за время тестирования информационной системы (5 дней – 1 рабочая неделя). Таким образом данный вид затрат составляет:

Итого: $K_{\text{оэ}} = 20880/4 = 5\,220$ (руб.)

Список капитальных затрат показан в таблице 3.4 и рисунке 3.1.

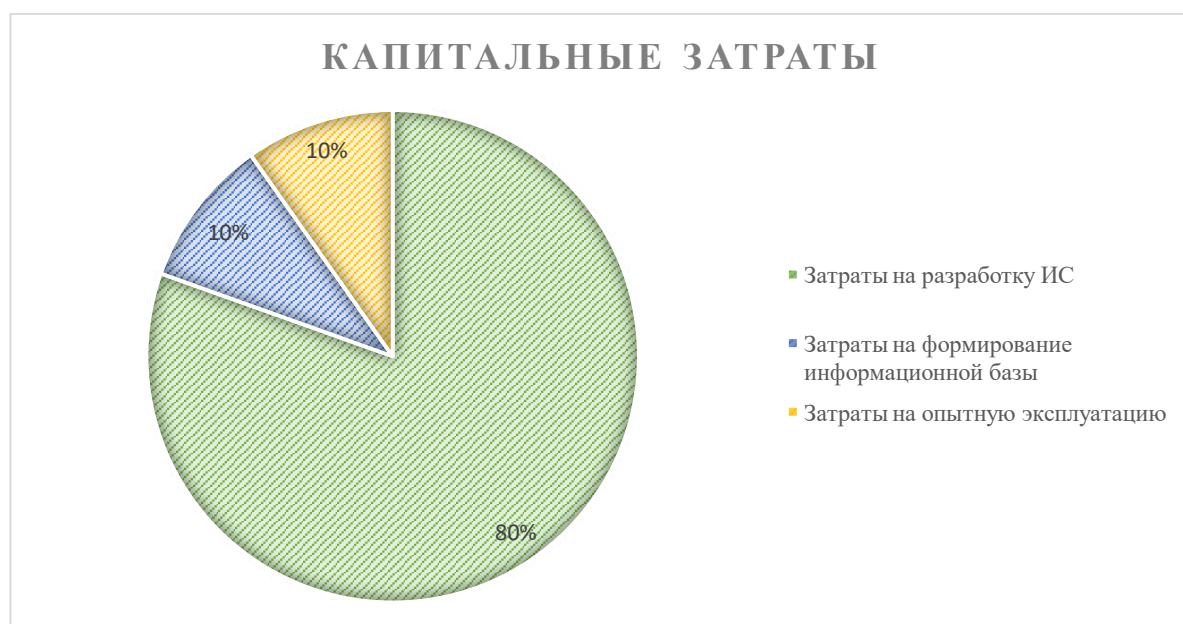


Рисунок 3.1 – Диаграмма капитальных затрат

Таблица 3.4 – Список капитальных затрат

Затраты	Состав затрат	Планируемая сумма, руб.
Затраты на разработку ИС	Затраты на заработную плату разработчика	40 778
	Затраты на инструментальные программы средства для проектирования	0
	Затраты на средства вычислительной техники для проектирования	943
	Прочие затраты на разработку	1 252
Затраты на технические средства управления		0
Затраты на создание линий связи локальных сетей		0
Затраты на программные средства		0
Затраты на формирование информационной базы		5 220
Затраты на обучение персонала		0
Затраты на опытную эксплуатацию		5 220

Капитальные затраты, которые рассчитываются по формуле (2), равны:

$$K = 43\,942 + 5\,220 + 5\,220 = 54\,382 \text{ (руб.)}$$

3.2.2 Эксплуатационные затраты

Расчет эксплуатационных затрат происходит по следующей формуле:

$$C = C_{\text{зп}} + C_{\text{ао}} + C_{\text{то}} + C_{\text{лс}} + C_{\text{ни}} + C_{\text{проч}}, \quad (5)$$

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

$C_{\text{ао}}$ – амортизационные отчисления;

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

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

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

$C_{проч}$ – прочие затраты.

Зарплата управленческого персонала ($C_{зп}$), работающего с использованием ИС(пользователей ИС) не учитывается, так как ввод в эксплуатацию разработанной АИС не влияет существенным образом на их работу.

Амортизационные отчисления ($C_{ао}$) состоят из стоимости используемого для работы системы оборудования (в данном случае – сервера) и не будут учитываться, т.к. система уже размещена на сервере виртуальной машины.

Затраты на техническое обслуживание ($C_{то}$) состоят из заработной платы администратора системы за год. Администратору предстоит тратить на обслуживание разработанной АИС около 1 часа рабочего времени в неделю. Таким образом в год будет уходить 48 часов на обслуживание АИС, а это 6 рабочих дней (0,2 от заработной платы в месяц), а именно:

$$C_{то} = 20\ 880 \cdot 0,2 = 4\ 176 \text{ (руб.)}$$

Затраты, связанные с использованием глобальных вычислительных сетей ($C_{лс}$), не учитываем ввиду использования на предприятии безлимитного интернета и несущественного трафика, создаваемого системой, по сравнению с общим трафиком предприятия.

Затраты на носители информации ($C_{ни}$) исключаются, так как система располагается на сервере и не требует дополнительных носителей информации.

Прочие затраты ($C_{проч}$) составляют 3 % от общей суммы эксплуатационных затрат.

$$C_{проч} = 4\ 176 \cdot 3\% = 125 \text{ (руб.)}$$

Список эксплуатационных затрат показан в таблице 3.5.

Таблица 3.5 – Список эксплуатационных затрат

Состав затрат	Планируемая сумма рублей в год, руб.
Затраты на заработную плату персонала	0
Затраты на амортизацию отчисления	0
Затраты на техническое обслуживание	4 176
Затраты на использование глобальных сетей	0
Затраты на носители информации	0
Прочие затраты	125

Таким образом, сумма годовых эксплуатационных затрат, рассчитываемых по формуле (5), составляет:

$$C = 4\,176 + 125 = 4\,301 \text{ (руб.)}$$

3.2.3 Прямые затраты

Прямые затраты равны:

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \quad (6)$$

где DE_1 – Капитальные затраты;

DE_2 – Расходы на управление информационными технологиями;

DE_3 – Расходы на техническую поддержку;

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;

Капитальные затраты (DE_1), рассчитанные ранее, составляют 54 964(руб.)

Расходы на управление ИТ (DE_2) в соответствии с эксплуатационными затратами не учитываются.

Расходы на техническую поддержку АО и ПО (DE_3) составляют 4 176 руб.

Расходы на разработку прикладного ПО внутренним силам (DE_4) включены в капитальные затраты.

Расходы на аутсорсинг (DE_5) не учитываются, так как услуги аутсорсеров не использовались.

Командировочные расходы (DE_6) не учитываются ввиду отсутствия командировок.

Расходы на услуги связи (DE_7) исключаются, так как данная статья расходов не относится к проекту.

Другие группы расходов (DE_8) включают в себя прочие эксплуатационные затраты и составляют 125 руб.

Таким образом, сумма прямых расходов, рассчитываемые по формуле (6), составляет:

$$DE = 54\,382 + 4\,176 + 125 = 58\,683 \text{ (руб.)}$$

Итого, совокупная стоимость владения информационной системой, рассчитываемая по формуле (1), составляет:

$$TCO = 59\,265 \text{ (руб.)}$$

3.3 Экономическая эффективность

Автоматизированная информационная система для обмена данными о застрахованных лицах, стоящих на диспансерном наблюдении между базами данных ТФОМС РХ и РИАМС «ПроМед» создаётся с целью сокращения

времени обработки данных. В отличие от того, что, если бы врачам пришлось передавать данные по электронной почте, созданная АИС позволяет сделать передачу данных в несколько раз быстрее.

Значимость технических решений (ЗТР) вычисляется по следующей формуле:

$$\text{ЗТР} = k_a \cdot k_{\text{п}} \cdot k_c + k_m \cdot k_o \cdot k_{\text{ш}}, \quad (5)$$

где k_a – коэффициент актуальности;

$k_{\text{п}}$ – коэффициент соответствия программам важнейших работ научно-технического прогресса;

k_c – коэффициент сложности;

k_m – коэффициент места использования;

k_o – коэффициент объема использования;

$k_{\text{ш}}$ – коэффициент широты охвата охраняемыми мероприятиями;

В таблице 3.6 приведены коэффициенты и ЗТР базового и разрабатываемого вариантов проекта.

Таблица 3.6 – Коэффициенты и ЗТР базового и разрабатываемого варианта

Коэффициенты	Базовый вариант	Разрабатываемый вариант
k_a	1	2
$k_{\text{п}}$	1	1
k_c	1	1
k_m	1	1
k_o	1	2
$k_{\text{ш}}$	1	2
ЗТР	2	6

Таким образом, из данной таблицы видно, что разрабатываемая АИС имеет более высокий показатель эксплуатационно-технического уровня по

сравнению с базовым вариантом. Вычисляем коэффициент эксплуатационно-технического уровня $k_{эту}$ по формуле:

$$k_{эту} = \frac{ЗТР_{пр}}{ЗТР_{баз}}, \quad (6)$$

где $ЗТР_{пр}$ и $ЗТР_{баз}$ – значимость технического решения для проекта и для базового варианта соответственно.

$$k_{эту} = \frac{6}{2} = 3$$

$k_{эту} > 1$, следовательно, разработка проекта является оправданной с технической точки зрения.

Вычислим комплексный показатель качества проекта по группе показателей $I_{эту}$ по формуле:

$$I_{эту} = \sum b_i * X_i, \quad (7)$$

где b_i – коэффициент весомости i го показателя;

X_i – относительный показатель качества, устанавливаемый экспертным путем по выбранной шкале оценивания.

Для оценки $I_{эту}$ рекомендуется пятибалльная шкала оценивания.

В таблице 3.7 приведен расчет показателя качества.

Таблица 3.7 – Расчет показателя качества

Показатель качества	Весовой коэффициент, b_i	Оценка, X_i	
		Разраб. проект	Базовый проект
Удобство работы(пользовательский)	0,2	5	1

Окончание таблицы 3.7

Надежность (защита данных)	0,2	4	2
Функциональные возможности	0,1	1	1
Временная экономичность	0,4	4	1
Время обучения персонала	0,1	1	1
Комплексный показатель качества $I_{эту}$		3,4	1,3

Коэффициент технического уровня:

$$k_T = \frac{I_{этупр}}{I_{этубаз}} \quad (8)$$

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

$$k_T = \frac{3,4}{1,3} = 2,6$$

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

$$Z_i = C_i + E_i * Z_{ппi}, \quad (9)$$

где C_i – текущие эксплуатационные затраты единицы i го вида работ, р.

$Z_{ппi}$ – суммарные затраты, связанные с внедрением проекта;

$E_n = 0,33$ – нормативный коэффициент экономической эффективности;

Для базового варианта:

$$Z_{\text{баз}} = 47\,314 + 0,33 * 0 = 47\,314 \text{ (руб.)}$$

Для проекта:

$$Z_{\text{пр}} = 4\,301 + 0,33 * 54\,964 = 22\,439 \text{ (руб.)}$$

Экономический эффект от использования разрабатываемой системы определяется по формуле:

$$\mathcal{E} = (Z_{\text{баз}} \cdot k_{\text{T}} - Z_{\text{пр}}) \cdot V, \quad (10)$$

где $Z_{\text{баз}}$, $Z_{\text{пр}}$ – приведенные затраты на единицу работ, выполняемых с помощью базового и проектируемого вариантов процесс обработки информации, р.; k_{T} – коэффициент эксплуатационно-технической эквивалентности;

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

Экономический эффект от использования разрабатываемой системы:

$$\mathcal{E} = (47\,314 \cdot 2,6 - 22\,439) \cdot 1 = 100\,577$$

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

$$T_{\text{ок}} = \frac{Z_{\text{пп}}}{\mathcal{E}}, \quad (11)$$

где $Z_{\text{пп}}$ – единовременные затраты на разработку проекта, р.;

\mathcal{E} – годовая эффективность, р.

Рассчитываемый срок окупаемости затрат на разработку продукта:

$$T_{\text{ок}} = \frac{54\,964}{100\,577} = 0,55$$

Таким образом, срок окупаемости составляет примерно полгода.

Фактический коэффициент экономической эффективности разработки (E_{ϕ}):

$$E_{\phi} = \frac{1}{T_{ок}} \quad (12)$$

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

Рассчитаем фактический коэффициент экономической эффективности разработки (E_{ϕ}):

$$E_{\phi} = \frac{1}{0,55} = 1,81$$

Так как $E_{\phi} = 1.81 > E_n$, 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 3.8 приведены сводные данные экономического обоснования.

Таблица 3.8 – Сводные данные экономического обоснования.

Показатель	Величина
Затраты на разработку проекта	22 314
Общие эксплуатационные затраты	4 301
Экономический эффект	100 702
Коэффициент экономической эффективности	1,85
Срок окупаемости	6 м.

Рассчитанная экономическая эффективность дает возможность оценить риски в данном проекте.

3.4 Оценка риска при реализации проекта разработки АИС

Наиболее сильно влияющими на реализацию проекта рисками являются:

Организационный риск. Этот риск имеет высокий уровень влияния на проект, так как может существенно увеличить стоимость разработки и внедрения проекта из-за того, что может увеличиться время разработки. Однако вероятность данного риска минимальна, так как между ГКУЗ РХ «РМИАЦ» и ТФОМС РХ заключён договор.

Лимитированное время разработки. Данный риск имеет высокий уровень влияния на проект, потому что, если разработка проекта не уложится в срок, то это повлечет за собой дополнительные капитальные расходы. При этом вероятность риска средняя, так как этот проект является не основной задачей разработчика на работе, что может повлечь за собой задержки при разработке.

Неучтенные дополнительные расходы. Этот риск имеет средний уровень влияния на проект, так как это может затормозить разработку проекта и увеличить расходы. При этом вероятность риска средняя, так как учесть все дополнительные расходы почти не возможно.

Реализационный риск. Этот риск имеет средний уровень влияния на проект, так как может увеличить стоимость разработки из-за возможного добавления нового функционала. Вероятность риска низкая, потому что список требований, которым должна соответствовать ИС, оглашен в договоре и вряд ли будет изменён.

Дополнительные расходы на доработку проекта. Этот риск имеет низкий уровень влияния на проект, так как он немного повышает операционные расходы на разработку проекта. При этом вероятность риска высокая, потому что все ошибки при разработке найти почти невозможно.

Таблица Ж.1 со списком рисков представлена в приложении Ж.

3.4.1 Меры по предотвращению или снижению риска

Шанс срабатывания организационного риска в данном проекте невероятно мал благодаря тому, что между организациями заключён договор.

Лимитированное время разработки. Тщательное продумывание плана перед разработкой поможет более четко сформулировать временные границы для разрабатываемого проекта, а это существенно снизит вероятность данного риска.

Неучтенные дополнительные расходы. Дополнительное соглашение к договору, т.е. увеличение бюджета на дополнительные расходы. Это значительно снизит вероятность риска.

Реализационный риск. Создать дополнительное соглашение к договору, где будут максимально расписаны, возможные непредвиденные расходы. Это существенно снизит шанс срабатывания риска.

Дополнительные расходы на доработку проекта. Регулярные тесты могут значительно снизить вероятность данного риска, однако это повлечет за собой увеличение стоимости проекта.

Вывод по разделу «Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ»

В данном разделе были выявлены технические характеристики проекта, также были посчитаны капитальные затраты, которые составляют 54 964 (руб.), и эксплуатационные затраты, которые составляют 4 301 (руб.), что позволило рассчитать совокупную стоимость владения информационной системой (ТСО), сумма которого составляет 59 265 (руб.). Так как фактический коэффициент

экономической эффективности разработки $1.81 > 0,33$, то разработка и внедрение разрабатываемого продукта являются эффективными, т. е. эффект от использования данной системы окупает все затраты, связанные с проектированием и эксплуатацией. Срок окупаемости составляет шесть месяцев.

ЗАКЛЮЧЕНИЕ

В разделе «Анализ предметной области. Выбор средств проектных решений» проанализирована основная деятельность учреждений ГКУЗ РХ «РМИАЦ» и ТФОМС РХ, которая позволила выявить то, что РИАМС «ПроМед» и ИС ТФОМС РХ используют разные форматы записи данных.

Данные будут передаваться по защищенной сети в соответствии с Федеральным законом РФ от 27 июля 2006 года № 152-ФЗ «О персональных данных».

Для реализации автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ, были проанализированы наиболее популярные средства в среде современных IT-разработчиков:

- языки программирования: Python, PHP и JavaScript;
- СУБД: Microsoft SQL, SQLite, PostgreSQL и MariaDB.

Был выбран язык программирования JavaScript.

Согласно проведенному анализу СУБД создание БД будет осуществляться с помощью MariaDB, преимуществом которой является ее производительность и безопасность, а также простота в использовании. Также в проекте будут использоваться HTML и CSS, которым не существует альтернатив.

В разделе «Разработка автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ» была построена модель по методологиям моделирования DFD и IDEF3 для того, чтобы понять, как будут передаваться данные о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ.

Была разработана АИС для передачи данных из РИАМС «ПроМед» в ИС ТФОМС РХ, которая включила в себя создание таких микросервисов, как:

- принятие информации из базы данных РИАМС «ПроМед»;

- преобразование данных из формата XML в JSON;
- загрузка преобразованных данных в БД Шину;
- отправление данных из БД Шину в ИС ТФОМС РХ.

Также был разработан модуль генератора преобразованных фиктивных данных из РИАМС «ПроМед» для того, чтобы показать как данные передаются в БД шины.

После этого были проведены тесты на совпадение сведений о застрахованных лицах, стоящих на диспансерном наблюдении, с уже вписанными сведениями в БД шины.

В разделе «Оценка экономической эффективности автоматизированной информационной системы для передачи данных о застрахованных лицах, стоящих на диспансерном наблюдении, из РИАМС «ПроМед» в ИС ТФОМС РХ» были выявлены технические характеристики проекта, были посчитаны капитальные затраты, которые составляют 54 964 (руб.), и эксплуатационные затраты, которые составляют 4 301 (руб.), что позволило рассчитать совокупную стоимость владения информационной системой (ТСО), сумма которого составляет 59 265 (руб.). Срок окупаемости составляет шесть месяцев, а это показывает, что разработанная АИС экономически эффективна, т.к. количество затрат существенно снизится. Были определены риски и возможность их предотвращения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сван. Информационные технологии для здравоохранения [Электронный ресурс]: Региональная информационно-аналитическая медицинская система (РИАМС) «ПроМед». – Режим доступа: <http://swan.perm.ru>
2. Avast Blog [Электронный ресурс]: Что такое VPN и как это работает. – Режим доступа: <https://blog.avast.com>
3. ТФОМС РХ [Электронный ресурс]: О ТФОМС РХ. – Режим доступа: <http://tfomsrh.ru>
4. Словари и энциклопедии на Академике [Электронный ресурс]: Автоматизированная информационная система. – Режим доступа: <https://dic.academic.ru>
5. Хабр [Электронный ресурс]: Просто о микросервисах. – Режим доступа: <https://habr.com>
6. QA evolution [Электронный ресурс]: Особенности тестирования web-приложений. – Режим доступа: <https://qaevolution.ru>
7. Сайтостроение от А до Я [Электронный ресурс]: PHP, Ruby, Python – краткая характеристика трёх языков программирования. – Режим доступа: <http://www.internet-technologies.ru>
8. Ubuntu [Электронный ресурс]: Программная платформа Node.js. – Режим доступа: <https://help.ubuntu.ru/wiki/javascript>
9. HTML5 BOOK [Электронный ресурс]: Основы JavaScript. – Режим доступа: <https://html5book.ru/osnovy-javascript/>
10. DEVACADEMY – обучение современному программированию онлайн [Электронный ресурс]: SQLite vs MySQL vs PostgreSQL. – Режим доступа: <http://devacademy.ru/>
11. Национальная библиотека им. Н. Э. Баумана [Электронный ресурс]: MariaDB. – Режим доступа: <https://ru.bmstu.wiki>

12. BusinessStudio [Электронный ресурс]: Понятие бизнес-процесса. – Режим доступа: <https://www.businessstudio.ru>
13. Что такое DFD (диаграммы потоков данных) [Электронный ресурс]: Понятие бизнес-процесса. – Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/>
14. Построение диаграммы декомпозиции в нотации IDEF3 [Электронный ресурс]: Понятие бизнес-процесса. – Режим доступа: <http://khpriip.mipk.kharkiv.edu/library/technpgm/labs/lab05.html>
15. RevolverSoft [Электронный ресурс]: .XML – Расширение файла. – Режим доступа: <https://www.reviversoft.com>
16. Введение в JSON [Электронный ресурс]. – Режим доступа: <https://www.json.org/json-ru>
17. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 «Прикладная информатика» [Электронный ресурс]. – метод. указания / сост. Е. Н. Скуратенко, В. И. Кокова, И. В. Янченко; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Электрон. текстовые, граф. дан. (0,71 МБ). – Абакан : ХТИ – филиал СФУ, 2017. – 1 файл.
18. СТО 4.2–07–2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности, Красноярск: ИПК СФУ, 2014. – 60 с.

ПРИЛОЖЕНИЕ А

Программный код файла index.js

```
const AMQP = require('amqplib')
const path = require('path')
const moment = require('moment')
const fs = require('fs')
const sharedFunctions = require(fs.existsSync(path.join(process.cwd(), 'services/rish-web-api/routes/shared.js'))
? path.join(process.cwd(), 'services/rish-web-api/routes/shared.js') : './shared.js')
const SERVICE_NAME = '[rmis-rabbitmq-poller]'
const consumeFromQ = sharedFunctions.consumeFromQ
const publishToQ = sharedFunctions.publishToQ

let localRMQConnectionString,
    rmisRMQConnectionString,
    rmisQData,
    rmisQAnswer,
    localDispObservationQ,
    localAttachmentQ,
    dispObservationDeletionQ

if (process.env.NODE_ENV === 'production') {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    rmisRMQConnectionString = process.env.RMQ_RMIS_CONNECTION_PRODUCTION
    rmisQData = process.env.RMQ_Q_RMIS_DATA_PRODUCTION
    rmisQAnswer = process.env.RMQ_Q_RMIS_ANSWER_PRODUCTION
    localDispObservationQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_PRODUCTION
    localAttachmentQ = process.env.RMQ_Q_LOCAL_ATTACHMENT_PRODUCTION
    dispObservationDeletionQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_DELETION_Q_PRODUCTION
} else {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    rmisRMQConnectionString = process.env.RMQ_RMIS_CONNECTION_TEST
    rmisQData = process.env.RMQ_Q_RMIS_DATA_TEST
    rmisQAnswer = process.env.RMQ_Q_RMIS_ANSWER_TEST
    localDispObservationQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_TEST
    localAttachmentQ = process.env.RMQ_Q_LOCAL_ATTACHMENT_TEST
    dispObservationDeletionQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_DELETION_Q_TEST
}

const localRMQConnection = AMQP.connect(localRMQConnectionString)
const rmisRMQConnection = AMQP.connect(rmisRMQConnectionString)

CODE_MO_HACK = {
    "190007": 190006,
    "190071": 190070,
    "190072": 190070
}

const processDispObservation = (json) => {
    const result = {
        meta: {
            OPERATIONTYPE: null
        },
        data: {
            bdz_id: null,
            DISP_ID: null,

```

```

    DS: null,
    DS_DETECT: null,
    DS_DETECTTYPE: null,
    SNILS_VR: null,
    DATE_OUT: null,
    RESULT_OUT: null,
    DATES: null,
    CODE_MO: null,
    promed_pacient_id: null,
    created_at: null
  }
}

if (json['DISP'] && json['DISP']['HEADER']) {
  //console.log('JSON', json)
  const header = json['DISP']['HEADER'][0]
  const body = json['DISP']['BODY'] ? json['DISP']['BODY'][0] : null
  console.log('OPERATIONTYPE', header.OPERATIONTYPE[0])
  if (body) {
    if (header.OPERATIONTYPE[0] === "Insert" || header.OPERATIONTYPE[0] === "Update") {
      if (header.OPERATIONTYPE&&header.OPERATIONTYPE[0]) {
        result.meta.OPERATIONTYPE = header.OPERATIONTYPE[0]
      }
      if (header.CODE_MO&&header.CODE_MO[0]) {
        const code_mo = CODE_MO_HACK[header.CODE_MO[0]] ? CODE_MO_HACK[header.CODE_MO[0]] :
header.CODE_MO[0]
        result.data.CODE_MO = code_mo
      }
      if (header.DISP_ID&&header.DISP_ID[0]) {
        result.data.DISP_ID = header.DISP_ID[0]
      }
      if (header.DATA&&header.DATA[0]) {
        result.data.created_at = header.DATA[0]
      }
      if (body.BDZ_ID&&body.BDZ_ID[0]) {
        result.data.bdz_id = body.BDZ_ID[0]
      }
      if (body.ID_PAC&&body.ID_PAC[0]) {
        result.data.promed_pacient_id = body.ID_PAC[0]
      }
      if (body.DATE_IN&&body.DATE_IN[0]) {
        result.data.DATE_IN = body.DATE_IN[0]
      }
      if (body.DS_DETECTTYPE&&body.DS_DETECTTYPE[0]) {
        result.data.DS_DETECTTYPE = body.DS_DETECTTYPE[0]
      }
      if (body.DS_DETECT&&body.DS_DETECT[0]) {
        result.data.DS_DETECT = body.DS_DETECT[0]
      }
      if (body.DS && body.DS[0]) {
        result.data.DS = body.DS[0]
      }
      if (body.SNILS_VR&&body.SNILS_VR[0] &&body.SNILS_VR[0] !== '0000000000') {
        result.data.SNILS_VR = body.SNILS_VR[0]
      }
      if (body.RESULT_OUT&&body.RESULT_OUT[0]) {
        result.data.RESULT_OUT = body.RESULT_OUT[0]
      }
      if (body.DATES&&body.DATES[0]) {
        if (body.DATES[0].PLAN_DATE) {
          result.data.DATES = body.DATES[0].PLAN_DATE.map(e => e)
        }
      }
    }
  }
}

```

```

else {
  console.error(`Ошибка с плановыми датами: ${JSON.stringify(body)}`)
  result.data.DATES = [ moment().format('YYYY-MM-DD') ]
  }
}
//console.log(localDispObservationQ, JSON.stringify(result))
}
else {
  const _message = `OPERATIONTYPE не найден: ${header.OPERATIONTYPE[0]}, DISP_ID:
${result.data.DISP_ID}`
  sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME, _message)
  }
}
else {
  if (header.OPERATIONTYPE[0] === 'Delete') {
    result.meta.OPERATIONTYPE = 'Delete'
    result.data.DISP_ID = header.DISP_ID[0]
  }
}
}
return result
}

const processAttachment = () => {

}

// слушаем очередь РМИС с данными
console.log(` Слушаем ${rmisQData}, NODE_ENV ${process.env.NODE_ENV}`)
consumeFromQ(rmisRMQConnection, rmisQData, async (ch, rawMsg) => {
  try {
    const content = rawMsg.content.toString()

    const parsedResult = await sharedFunctions.parseXML(content)
    console.log(parsedResult);
    if (parsedResult[1]) {
      console.log(parsedResult[1])
      const type = Object.keys(parsedResult[1])[0]
      if (type === "DISP") {
        const result = processDispObservation(parsedResult[1])
        console.log(`результат после обработки: ${JSON.stringify(result)}`)
        if (result.meta.OPERATIONTYPE === 'Insert' || result.meta.OPERATIONTYPE === 'Update') {
          publishToQ(localRMQConnection, localDispObservationQ, JSON.stringify(result), (err, ok) => {
            //console.log(err, ok)
          })
          if (!err) {
            ch.ack(rawMsg)
          }
        }
      }
      else if (result.meta.OPERATIONTYPE === 'Delete') {
        console.log(`DISP на удаление`)
        publishToQ(localRMQConnection, dispObservationDeletionQ, result.data.DISP_ID, (err, ok) => {
          if (!err) {
            ch.ack(rawMsg)
          }
        })
      }
      else {
        console.log(`Ошибка...`)
      }
    }
    elseif (type === "ATTACH") {

```

```

console.log('Прикрепление - отмечаем как обработанное...')
ch.ack(rawMsg)
}
else {
console.log('Неизвестный тип сообщения - отмечаем как обработанное...')
ch.ack(rawMsg)
console.error('Неизвестный тип: ${type}`')
}
}
else {
console.error('Ошибка при парсингеXML: ${parsedResult[0]}, данные: ${content}`')
process.exit(1)
}
} catch (e) {
const _message = `ошибка ${e.message}`
await sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME, _message)
process.exit(1)
}
})
console.log('Слушаем ${rmisQAnswer}, NODE_ENV ${process.env.NODE_ENV}`')
// слушаем очередь РМИС с ответами на обновление данных
consumeFromQ(rmisRMQConnection, rmisQAnswer, (ch, rawMsg) => {
console.log(rawMsg.content.toString())

})
.catch(e => console.warn(e))

```


ПРИЛОЖЕНИЕ Б

Программный код из файла create_fake_data.js

```
functiongetMoCode(){
    varmasCodeMo = ["190048", "190007", "190085", "190006", "190033", "190036", "190032", "190037",
"190038", "190008", "190001", "190013", "190004", "190002", "190003", "190009", "190010", "190022", "190087",
"190020", "190070", "190071", "190072", "190047", "190039", "190040", "190030", "190086", "190025", "190028",
"190031", "190045", "190113"];
    var result = masCodeMo[getRandomInt(0,masCodeMo.length-1)];
    return result;
}

functiongetDate(){
    var result;

    result = new Date();
    result = `${result.getFullYear()}-${result.getMonth()+1}-${result.getDay()+1}`;
    return result;
}

functiongetRandomInt(minRandom, maxRandom){
    var rand = minRandom - 0.5 + Math.random() * (maxRandom - minRandom + 1);
    rand = Math.round(rand);
    return rand;
}

functiongetDiag(){
    var result;
    vararr_EN = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
'Z'];
    vararr_chisl = ["00", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35",
"36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55",
"56", "57", "58", "59", "60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70", "71", "72", "73", "74", "75",
"76", "77", "78", "79", "80", "81", "02", "03", "04", "05", "06", "07", "08", "09", "90", "91", "92", "93", "94", "95",
"96", "97", "98", "99"];
    result = `${arr_EN[getRandomInt(0,arr_EN.length-1)]}`;
    result += `${arr_chisl[getRandomInt(0,arr_chisl.length-1)]}`;
    return result;
}

functiongetSampleDispObservation(){
    var bdz1 = getRandomInt(1, 99999999);
    varpromed = getRandomInt(1, 99999999);
    var disp1 = getRandomInt(1, 99999999);
    varcodeMO = getMoCode();
    varbirthDay = getDate();
    vardia = getDiag();
    console.log(bdz1, promed, disp1, codeMO, birthDay, dia);
    constdispObservation = {
        bdz_id: bdz1,
        DS: dia,
        CODE_MO: codeMO,
        created_at: birthDay,
        promed_pacient_id: promed,
        DISP_ID: disp1
    }
    returndispObservation
}
```

```
(async () => {
  try {
    const dispObservation = getSampleDispObservation()
    const sharedFunctions = require('./services/rish-web-api/routes/shared.js')
    const result = await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'insert', data:
dispObservation })
  }
  catch (e) {
    console.error(e) })})()
```

ПРИЛОЖЕНИЕ В

Программный код из микросервиса **disp-observation-to-tfoms**

```
console.log('Сервис по отправке диспансерных наблюдений в ТФОМС стартует...')
const AMQP = require('amqplib')
const fetch = require('node-fetch')
const SERVICE_NAME = '[send-to-tfoms-disp-observation]'
const fs = require('fs')
const path = require('path')
const sharedFunctions = require(fs.existsSync(path.join(process.cwd(), 'services/rish-web-api/routes/shared.js')) ? path.join(process.cwd(), 'services/rish-web-api/routes/shared.js') : './shared.js')
const consumeFromQ = sharedFunctions.consumeFromQ
const publishToQ = sharedFunctions.publishToQ

let localRMQConnectionString,
    localDispObservationToTfomsQ,
    tfoms_disp_observation_url

if (process.env.NODE_ENV === 'production') {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    localDispObservationToTfomsQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_TO_TFOMS_PRODUCTION
    tfoms_disp_observation_url = process.env.SEND_TO_TFOMS_DISP_OBSERVATION_URL_PRODUCTION
} else {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    localDispObservationToTfomsQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_TO_TFOMS_TEST
    tfoms_disp_observation_url = process.env.SEND_TO_TFOMS_DISP_OBSERVATION_URL_TEST
}

const connection = AMQP.connect(localRMQConnectionString)

console.log(`Начинаем слушать очередь ${localDispObservationToTfomsQ}, отправляем ${tfoms_disp_observation_url}`)

consumeFromQ(connection, localDispObservationToTfomsQ, async (ch, rawMsg) => {
    const DISP_ID = parseInt(rawMsg.content.toString())
    console.log(`Получен DISP_ID из очереди ${DISP_ID}`)
    const forTfoms = await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'get_by_DISP_ID', data: { DISP_ID: DISP_ID } })
    //console.log('forTfoms', forTfoms);
    const options = {
        method: 'POST',
        body: JSON.stringify(forTfoms),
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json'
        }
    }
    try {
        const response = await fetch(tfoms_disp_observation_url, options)
        const body = await response.json()
        //console.log(body)
        if (body.RESULT === "OK" && !(await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_tfoms_status', data: { send_to_tfoms_status_id: 2, DISP_ID: DISP_ID } })).error) {
            ch.ack(rawMsg)
        }
    }
})
```

```

    }
    else {
        (await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_tfoms_status',
data: { send_to_tfoms_status_id: 1, DISP_ID: DISP_ID } }))) && ch.ack(rawMsg)
        const _message = `[send-to-tfoms-disp-observation] ошибкаприобработкеответа:
${JSON.stringify(body)} DISP_ID: ${DISP_ID}`
        await sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME, _message)
    }
} catch (e) {
    const errorMessage = `ошибканасетевомуровнеприпередачкекартыс DISP_ID ${DISP_ID}
${e.message}`
    const _ok = await sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME,
errorMessage)
    await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_tfoms_status',
data: { send_to_tfoms_status_id: 3, DISP_ID: DISP_ID } })
    publishToQ(connection, localDispObservationToTfomsQ, DISP_ID.toString(), async (err, ok) => {
        if (!err) {
            ch.ack(rawMsg)
        }
        else {
            console.error(`${SERVICE_NAME} ошибкаприотправке DISP_ID ${DISP_ID} обратновочередь...`)
            process.exit(1)
        }
    })
}
})
}
})

```

ПРИЛОЖЕНИЕ Г

Программный код из микросервиса `disp-observation-to-ftoms`

```
console.log('Сервис по отправке диспансерных наблюдений в ТФОМС стартует...')
const AMQP = require('amqplib')
const fetch = require('node-fetch')
const SERVICE_NAME = '[send-to-ftoms-disp-observation]'
const fs = require('fs')
const path = require('path')
const sharedFunctions = require(fs.existsSync(path.join(process.cwd(), 'services/rish-web-api/routes/shared.js')) ? path.join(process.cwd(), 'services/rish-web-api/routes/shared.js') : './shared.js')
const consumeFromQ = sharedFunctions.consumeFromQ
const publishToQ = sharedFunctions.publishToQ

let localRMQConnectionString,
    localDispObservationToTfomsQ,
    tfoms_disp_observation_url

if (process.env.NODE_ENV === 'production') {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    localDispObservationToTfomsQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_TO_TFOMS_PRODUCTION
    tfoms_disp_observation_url = process.env.SEND_TO_TFOMS_DISP_OBSERVATION_URL_PRODUCTION
} else {
    localRMQConnectionString = process.env.RMQ_LOCAL_CONNECTION_PRODUCTION
    localDispObservationToTfomsQ = process.env.RMQ_Q_LOCAL_DISP_OBSERVATION_TO_TFOMS_TEST
    tfoms_disp_observation_url = process.env.SEND_TO_TFOMS_DISP_OBSERVATION_URL_TEST
}

const connection = AMQP.connect(localRMQConnectionString)

console.log(`Начинаем слушать очередь ${localDispObservationToTfomsQ}, отправляем в ${tfoms_disp_observation_url}`)

consumeFromQ(connection, localDispObservationToTfomsQ, async (ch, rawMsg) => {
    const DISP_ID = parseInt(rawMsg.content.toString())
    console.log(`Получен DISP_ID из очереди ${DISP_ID}`)
    const forTfoms = await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'get_by_DISP_ID', data: { DISP_ID: DISP_ID } })
    //console.log('forTfoms', forTfoms);
    const options = {
        method: 'POST',
        body: JSON.stringify(forTfoms),
        headers: {
            'Accept': 'application/json',
            'Content-Type': 'application/json'
        }
    }
    try {
        const response = await fetch(tfoms_disp_observation_url, options)
        const body = await response.json()
        //console.log(body)
        if (body.RESULT === "OK" && !(await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_ftoms_status', data: { send_to_ftoms_status_id: 2, DISP_ID: DISP_ID } })).error) {
            ch.ack(rawMsg)
        }
    } else {
```

```

        (await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_tfoms_status',
data: { send_to_tfoms_status_id: 1, DISP_ID: DISP_ID } })) && ch.ack(rawMsg)
        const _message = `[send-to-tfoms-disp-observation] ошибкаприобработкеответа:
${JSON.stringify(body)} DISP_ID: ${DISP_ID}`
        await sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME, _message)
    }
    } catch (e) {
        const errorMessage = `ошибканасетевомуровнеприпередачкартыс DISP_ID ${DISP_ID}
${e.message}`
        const _ok = await sharedFunctions.sendNotificationToDispObservationErrors(SERVICE_NAME,
errorMessage)
        await sharedFunctions.sendToMariadbProxy({ type: 'dispObservation', action: 'set_send_to_tfoms_status',
data: { send_to_tfoms_status_id: 3, DISP_ID: DISP_ID } })
        publishToQ(connection, localDispObservationToTfomsQ, DISP_ID.toString(), async (err, ok) => {
            if (!err) {
                ch.ack(rawMsg)
            }
            else {
                console.error(`${SERVICE_NAME} ошибкаприотправке DISP_ID ${DISP_ID} обратновочередь...`)
                process.exit(1)
            }
        })
    }
})
}
})

```

ПРИЛОЖЕНИЕ Д

Алгоритм работы АИС в виде блок-схемы

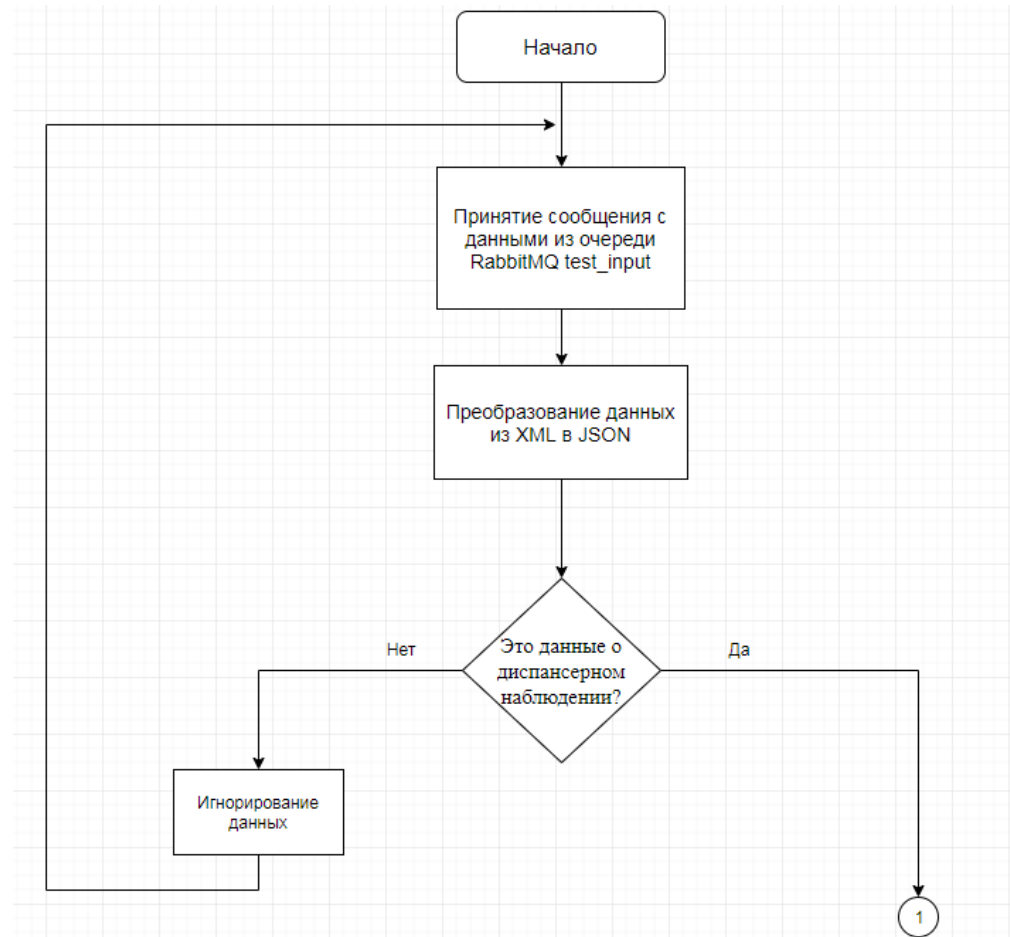


Рисунок Д.1– Алгоритм работы АИС в виде блок-схемы, лист 1

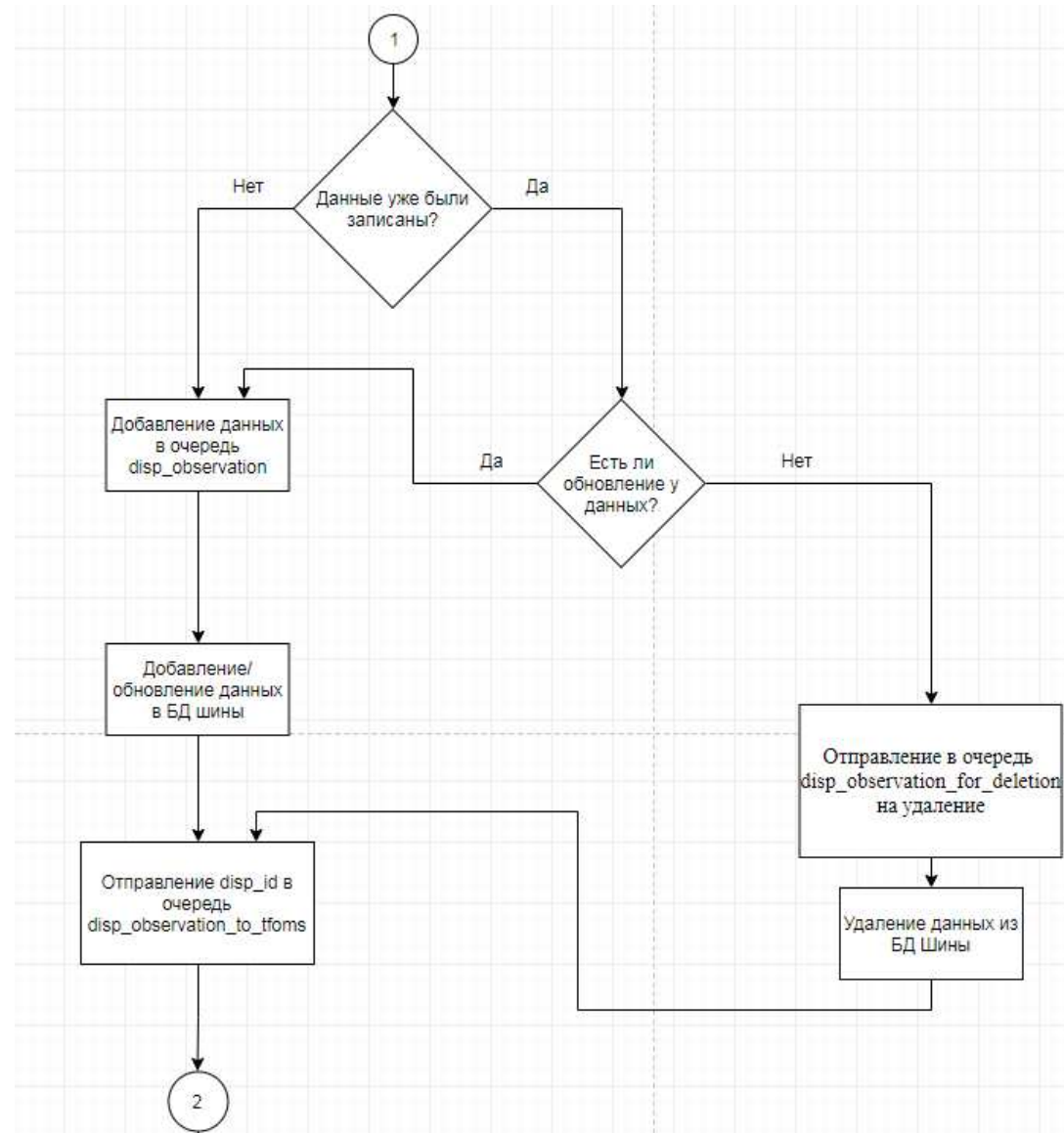


Рисунок Д.1, лист 2

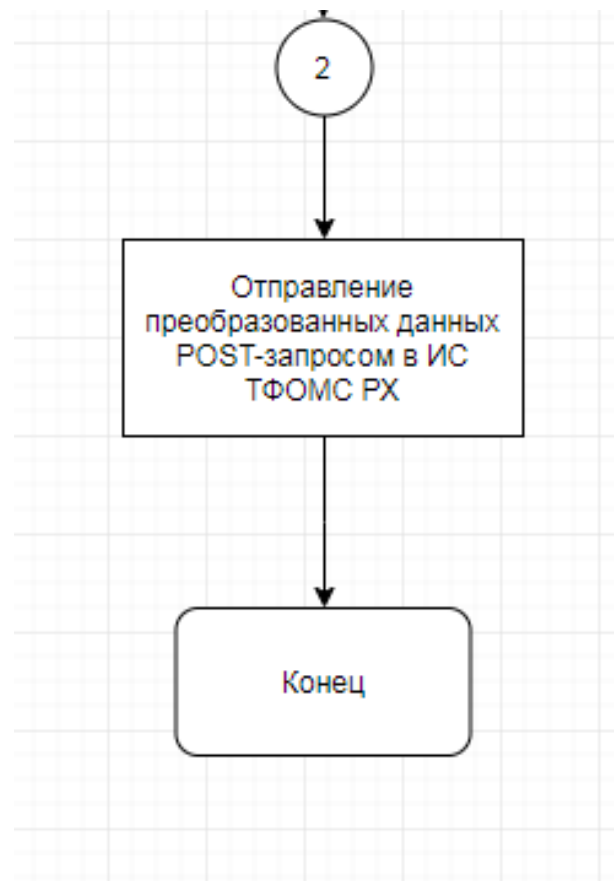


Рисунок Д.1, лист

ПРИЛОЖЕНИЕ Е

Алгоритм библиотеки xml2json в виде блок-схемы

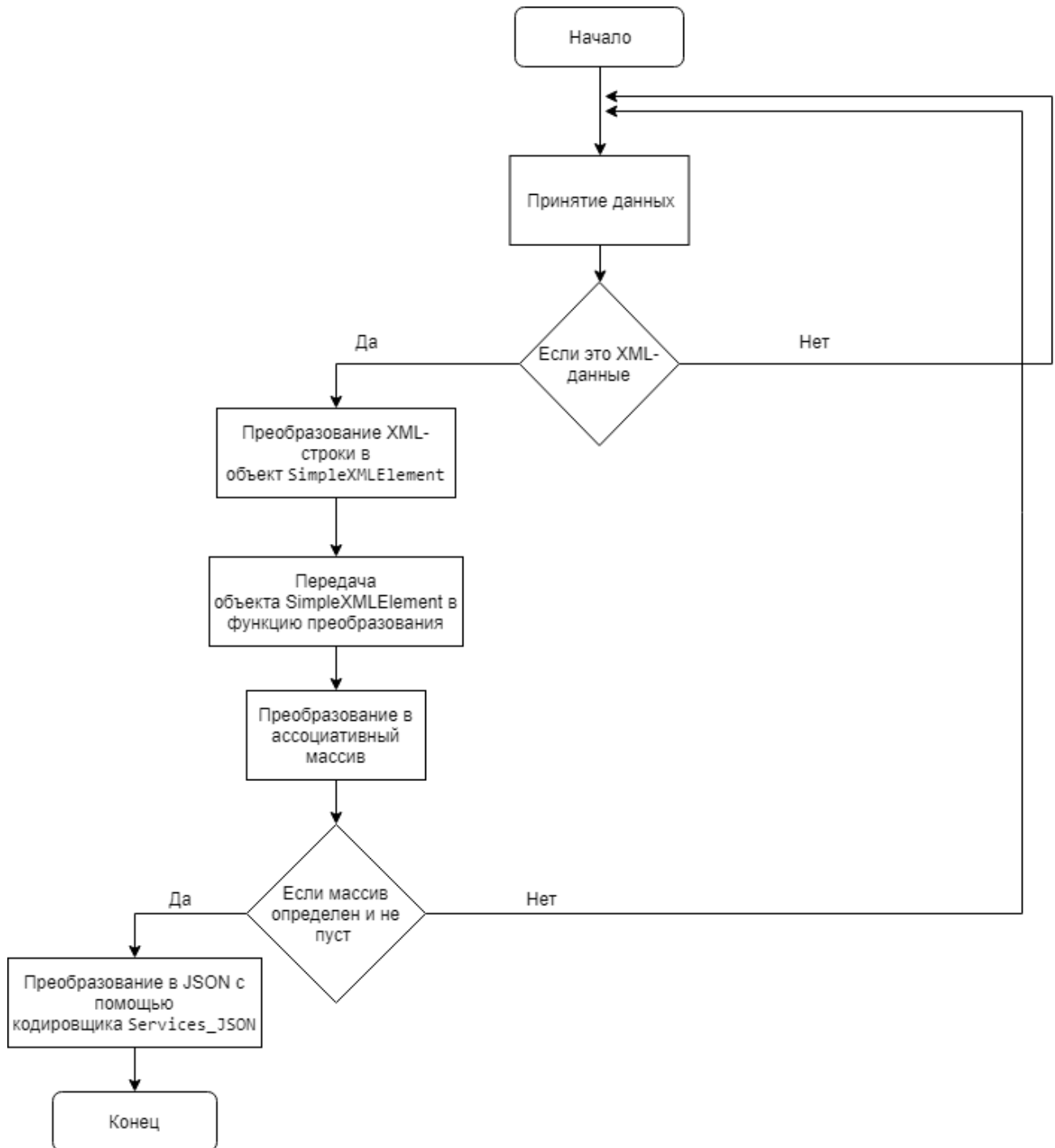


Рисунок Е.1 –Алгоритм библиотеки xml2json в виде блок-схемы

ПРИЛОЖЕНИЕ Ж

Список рисков

Таблица Ж.1 – Список рисков

№ проекта	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
1.	Риски, связанные с разработкой проекта	Организационный риск	Средний	Низкий	Обратная связь между организациями, проработка договора
2.		Лимитированное время разработки	Средний	Средний	Тщательное продумывание плана перед разработкой
3.		Неучтенные дополнительные расходы	Средний	Средняя	Дополнительное соглашение к договору
4.	Риски, связанные с внедрением проекта	Реализационный риск	Низкий	Низкий	Дополнительное соглашение к договору
5.		Дополнительные расходы на доработку проекта	Низкий	Высокая	Регулярные тесты

ПРИЛОЖЕНИЕ И

Содержимое Docker-файлов, прикрепленных к микросервисам

```
FROM alpine:3.8
ARG d_uid=0
ARG d_gid=0
WORKDIR /app
COPY ./ /app/
RUN apk update && \
    apk add git && \
    apk add nodejs && \
    apk add npm && \
    npm install

RUN chown -R $d_uid:$d_gid *

CMD          ./wait          &&          npm          run          start
```

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 18 наименований.

Один экземпляр сдан на кафедру.

« » июня 2019 г.
(дата)

(подпись)

И.М. Прохорович
(ФИО)

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

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

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
Е.Н. Скуратенко
подпись инициалы, фамилия
« 20 » 06 2019г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 – Прикладная информатика
код – наименование направления

Автоматизация передачи сведений о диспансерном наблюдении
тема
застрахованных лиц из РИАМС «ПроМед» в ИС ТФОМС РХ

Руководитель	<u>JK 20.06.19</u> подпись, дата	ст. преподаватель должность, ученая степень	<u>В.И. Кокова</u> инициалы, фамилия
Выпускник	<u>JY 20.06.19</u> подпись, дата		<u>И.М Прохорович</u> инициалы, фамилия
Консультанты по разделам:			
<u>Экономический</u> наименование раздела	<u>JK 20.06.19</u> подпись, дата		<u>Е.Н. Скуратенко</u> инициалы, фамилия
Нормоконтролер	<u>JK 20.06.19</u> подпись, дата		<u>В.И. Кокова</u> инициалы, фамилия

Абакан 2019