

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е. Н. Скуратенко
 подпись
« ___ » _____ 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

Разработка информационной системы «Паспорт цифровизации ГБУЗ РХ
«РМИАЦ»

Руководитель _____ доцент, канд. техн. наук Е. Н. Скуратенко
подпись, дата

Выпускник _____ В. А. Озол
подпись, дата

Консультанты
по разделам:

Экономический _____ Е. Н. Скуратенко
подпись, дата

Нормоконтролер _____ В. И. Кокова
подпись, дата

Абакан 2022

РЕФЕРАТ

Выпускная квалификационная работа (ВКР) по теме «Разработка информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»» содержит 88 страниц текстового документа, 7 таблиц, 86 иллюстраций, 9 формул, 15 использованных источников.

ИНФОРМАЦИОННАЯ СИСТЕМА, ПАСПОРТ ЦИФРОВИЗАЦИИ, ГБУЗ РХ «РМИАЦ», PYTHON, DJANGO, РИСКИ, ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ, АВТОРИЗАЦИЯ.

Объект выпускной квалификационной работы: деятельность сотрудников ГБУЗ РХ «РМИАЦ» по взаимодействию с гражданами.

Предмет ВКР: процесс автоматизации работы с информацией о медицинских организациях Республики Хакасия.

Цель ВКР: разработка информационной системы «Паспорт цифровизации ГБУЗ РХ РМИАЦ».

В первом разделе ВКР описана основная деятельность ГБУЗ РХ «РМИАЦ». Разработаны модели IDEF0, Use-case, DFD, описывающие требуемый функционал веб-приложения. Выбран язык и вспомогательные средства разработки.

Во втором разделе описан процесс разработки информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ».

В третьем разделе ВКР произведена оценка экономической эффективности информационной системы, проанализированы эксплуатационные риски, представлены возможные действия по их решению.

В результате была разработана информационная система «Паспорт цифровизации ГБУЗ РХ «РМИАЦ» для ГБУЗ РХ «РМИАЦ».

SUMMARY

The theme of the graduation thesis is «IT System Development: Digitalization Passport of State Budgetary Health Institution of Republic of Khakassia, Republican Medical Information and Analytical Center» (SBHI RKh «RMIAC»). It contains 88 pages, 7 charts, 86 figures, 7 formulae, 15 references.

IT SYSTEM, PASSPORT OF DIGITALIZATION, SBHI RKh «RMIAC», PYTHON, DJANGO, RISKS, ECONOMIC EFFICIENCY, AUTHORIZATION.

The object of the thesis: the framework of employees of SBHI RKh «RMIAC» on interaction with citizens.

The subject of the thesis: the process of information automating with medical organizations of the Republic of Khakassia.

The purpose of the thesis: to develop the IT system of Digitalization Passport for State Budgetary Health Institution of Republic of Khakassia, Republican Medical Information and Analytical Center.

The first section of the thesis describes the framework of State Budgetary Health Institution of Republic of Khakassia, Republican Medical Information and Analytical Center. The models of IDEF0, Use-case, and DFD have been developed that describe the required functionality of a web application. The language and development tools have been selected.

The second section describes the process of developing the IT system «Digitalization Passport for State Budgetary Health Institution of Republic of Khakassia, Republican Medical Information and Analytical Center».

The third section of the thesis assesses the economic efficiency of the IT system, analyzes operational risks, and presents possible actions to address them.

As a result, the IT system «Digitalization Passport for State Budgetary Health Institution of Republic of Khakassia, Republican Medical Information and Analytical Center» has been developed.

English language supervisor _____ N.V. Chezybaeva

СОДЕРЖАНИЕ

Введение	5
1 Анализ предметной области для создания информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»	8
1.1 Краткое описание деятельности ГБУЗ РХ «РМИАЦ»	8
1.2 По ширине Обоснование необходимости разработки системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ». Актуальность разработки	10
1.3 Бизнес-процессы предметной области	16
1.4 Характеристика IT-инфраструктуры ГБУЗ РХ «РМИАЦ»	21
1.5 Цель и задачи разработки проекта	21
1.6 Анализ типовых проектных решений и выбор инструментальных средств разработки	23
1.7 Выводы по разделу «Анализ предметной области для создания информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»	27
2 Разработка информационной системы	27
2.1 Установка необходимого ПО	27
2.2 Создание стартового проекта на Django.....	30
2.3 Создание приложений, основного html шаблона и главного администратора системы	33
2.4 Создание шаблонов html и маршрутов для отображения информации из базы данных	46
2.5 Настройка административной панели, разграничение прав доступа к информации из базы данных	60
2.6 Создание отчётов для выгрузки информации из базы данных в excel файл.....	72
2.7 Выводы по разделу «Разработка информационной системы».....	76

3 Расчёт экономической эффективности проекта «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»	77
3.1 Экономическая эффективность реализации проекта.....	77
3.2 Эксплуатационные риски.....	83
3.3 Выводы по разделу «Расчёт экономической эффективности проекта «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»	84
Заключение	85
Список использованных источников	86

ВВЕДЕНИЕ

Цифровизация деятельности органов власти является приоритетной задачей в рамках реализации проекта «Цифровая экономика Российской Федерации». В рамках реализации данной программы, необходимо обеспечить современным ПО и оборудованием отрасли социальной и промышленной сферы.

Государственное бюджетное учреждение здравоохранения республики Хакасия "Республиканский медицинский информационно-аналитический центр" занимается реализацией данной национальной программы в сфере здравоохранения. У ГБУЗ РХ «РМИАЦ» имеется потребность в оперативном получении достоверной информации о состоянии информационно-коммуникационной инфраструктуры сферы здравоохранения в разрезе региона.

По заказу ГБУЗ РХ «РМИАЦ» проделана работа по созданию информационной системы для сбора, хранения информации о цифровизации медицинских организаций в республике Хакасия.

Практически каждый день происходит изменение данных о оснащенности информационно-коммуникационной инфраструктуры медицинских организаций, обслуживаемых ГБУЗ РХ «РМИАЦ». В настоящее время сбор данной информации осуществляется посредством сохранения на внешний носитель. Работнику ГБУЗ РХ «РМИАЦ» приходится лично посещать более 30 медицинских организаций для получения данной информации.

Данный способ является неудобным и затратным. Обновление информации происходит в определенные периоды, это означает, что она не всегда актуальна.

Создание инструмента, позволяющего оперативно собирать, хранить и обрабатывать информацию об оснащенности медицинских организаций республики Хакасия является актуальной задачей, позволяющей решить вышеуказанную проблему.

Целью выпускной квалификационной работы является создание веб-приложения для сбора, хранения и обработки информации о медицинских организациях республики Хакасия.

Разработка информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ» выполнена совместно со студентом группы 58-1 Сагояковым Александром Дмитриевичем.

Во время разработки веб-приложения решены следующие задачи:

- 1 Изучена основная деятельность предприятия ГБУЗ РХ «РМИАЦ».
- 2 Обоснована необходимость разработки информационной системы.
- 3 Поставлена цель разработки программы, сформулированы задачи подлежащих решению.
- 4 Разработано веб-приложение.
- 5 Дана оценка экономической эффективности внедрения разработанного веб-приложения.

В первом разделе ВКР описана основная деятельность ГБУЗ РХ «РМИАЦ». Разработаны модели IDEF0, Use-case, DFD, описывающие требуемый функционал веб-приложения. Выбран язык и вспомогательные средства разработки.

Во втором разделе описан процесс разработки информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ».

В третьем разделе ВКР произведена оценка экономической эффективности информационной системы, проанализированы эксплуатационные риски, представлены возможные действия по их решению.

1 Анализ предметной области для создания информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»

1.1 Краткое описание деятельности ГБУЗ РХ «РМИАЦ»

Разработка паспорта цифровизации медицинских организаций, осуществляется для государственного бюджетного учреждения здравоохранения Республики Хакасия «Республиканский медицинский информационно-аналитический центр».

Республиканский медицинский информационно-аналитический центр основан в 2000 году для достижения важных целей:

- создание и обслуживание автоматизированных систем управления здравоохранением Республики Хакасия;
- координация деятельности службы медицинской статистики и медицинского статистического обеспечения органов управления здравоохранением;
- обеспечение безопасности автоматизированных информационных систем здравоохранения Республики Хакасия;
- формирование единой информационной системы здравоохранения Российской Федерации на базе современных компьютерных технологий с применением новых технологий сбора и анализа статистической информации;
- обработка, хранение и предоставление информации, обеспечивающей динамическую оценку здоровья граждан информационную поддержку принятия решений, направленных на его улучшение.

Рассмотрим структуру ГБУЗ РХ «РМИАЦ». На рисунке 1 отображены отделы, которые функционируют в учреждении.

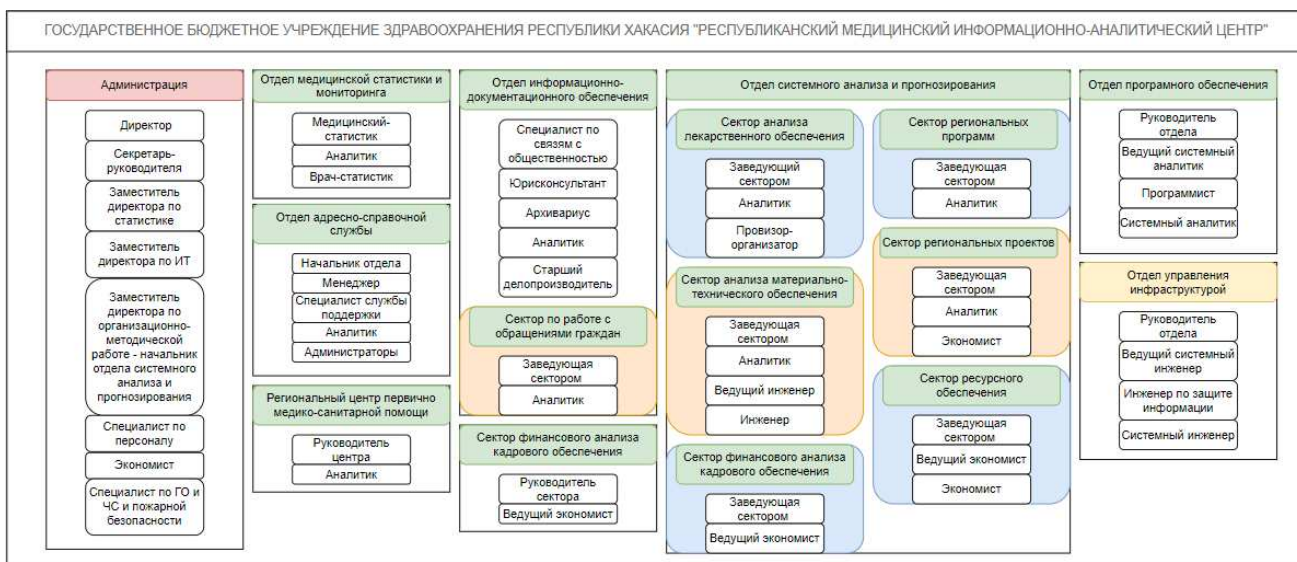


Рисунок 1 – Структурная схема ГБУЗ РХ «РМИАЦ»

Для обеспечения большого круга задач, стоящих перед ГБУЗ РХ «РМИАЦ» в организации существуют 7 отделов и 14 различных секторов в данных отделах, выполняющие различные функции. Данные отделы напрямую подчиняются администрации организации. Администрация выполняет самую важную задачу – оптимизация деятельности всей организации.

Разрабатываемая информационная система будет использоваться отделом управления инфраструктурой. Отдел управления инфраструктурой занимается сбором информации о цифровизации всех медицинских организаций республики. В собираемую информацию входят данные о медицинских организациях республики Хакасия, связанные с оснащением их компьютерной техникой, средствами телекоммуникации их характеристиками данных цифровых устройств, кроме того, требуется информация о направлениях деятельности организации.

Отдел управления инфраструктурой собирает информацию о характеристиках оборудования в медицинских организациях для того, чтобы своевременно модернизировать средства информационной поддержки их деятельности.

1.2 Обоснование необходимости разработки системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ». Актуальность разработки

ГБУЗ РХ «РМИАЦ» курирует и обслуживает множество медицинских организаций республики Хакасия, на данный момент курируется 39 организаций. У каждой из вышеуказанных организаций существует потребность в применении информационно-коммуникационных технологий, систем и оборудования. Для управления ими необходим «Паспорт цифровизации ГБУЗ РХ «РМИАЦ», позволяющий своевременно получать актуальную информацию.

Далее предоставлен список (таблица 1) медицинских организаций, обслуживаемых ГБУЗ РХ «РМИАЦ».

Таблица 1 – Список медицинских организаций

№	Наименование
1	ГБУЗ РХ «Абазинская городская больница»;
2	ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»;
3	ГБУЗ РХ «Абаканская межрайонная клиническая больница»;
4	ГБУЗ РХ «Аскизская межрайонная больница»;
5	ГБУЗ РХ «Бейская районная больница»;
6	ГБУЗ РХ «Белоярская районная больница»;
7	ГБУЗ РХ «Боградская районная больница»;
8	ГБУЗ РХ «Копьевская районная больница»;
9	ГБУЗ РХ «Республиканская детская клиническая больница»;
10	ГБУЗ РХ «Республиканская клиническая больница им. Г.Я. Ремишевской»;
11	ГБУЗ РХ «Республиканская клиническая инфекционная больница»;
12	ГБУЗ РХ «Республиканская клиническая офтальмологическая больница имени Н. М. Одежкина»;
13	ГБУЗ РХ «Республиканская клиническая психиатрическая больница»;
14	ГБУЗ РХ «Республиканская клиническая стоматологическая поликлиника»;
15	ГБУЗ РХ «Республиканский клинический кожно-венерологический диспансер»;
16	ГБУЗ РХ «Республиканский клинический наркологический диспансер»;
17	ГБУЗ РХ «Республиканский клинический онкологический диспансер»;
18	ГБУЗ РХ «Республиканский клинический перинатальный центр»;
19	ГБУЗ РХ «Республиканский клинический противотуберкулезный диспансер»;
20	ГБУЗ РХ «Республиканский медицинский информационно-аналитический центр»;

Продолжение таблицы 1

21	ГБУЗ РХ «Республиканский центр профилактики и борьбы со СПИД»;
22	ГБУЗ РХ «Саяногорская городская станция скорой медицинской помощи»;
23	ГБУЗ РХ «Саяногорская городская стоматологическая поликлиника»;
24	ГБУЗ РХ «Саяногорская межрайонная больница рабочего поселка Майна»;
25	ГБУЗ РХ «Саяногорская межрайонная больница»;
26	ГБУЗ РХ «Сорская городская больница»;
27	ГБУЗ РХ «Таштыпская районная больница»;
28	ГБУЗ РХ «Усть-Абаканская районная больница»;
29	ГБУЗ РХ «Черногорская городская стоматологическая поликлиника»;
30	ГБУЗ РХ «Черногорская межрайонная больница»;
31	ГБУЗ РХ «Черногорская межрайонная детская больница»;
32	ГБУЗ РХ «Черногорский межрайонный родильный дом»;
33	ГБУЗ РХ «Ширинская межрайонная больница»;
34	ГКУЗ РХ «Республиканский центр крови»;
35	ГКУЗ РХ «Республиканский центр медицины катастроф»;
36	ГКУЗ РХ «Республиканский центр общественного здоровья и медицинской профилактики»;
37	ГКУЗ РХ «Республиканское клиническое бюро судебно-медицинской экспертизы»;
38	ГКУ РХ «Учреждение централизованного обслуживания Министерства здравоохранения Республики Хакасия и подведомственных ему учреждений»;
39	Минздрав Хакасии.

Вышеперечисленные организации расположены в разных субъектах республики, из-за чего становится невозможным собирать актуальную информацию старым способом – при помощи личного посещения организации. Без применения веб-технологий невозможно получать объемную и актуальную информацию из разных мест.

ГБУЗ РХ «РМИАЦ» требуется знать о каждой из вышеуказанных организаций: общую информацию о медицинской организации; информацию об информационном системе организации; информацию о каждом структурном подразделении; информацию об информационных структурах каждого подразделения.

В общую информацию о медицинской организации входит:

- Наименование медицинской организации.
- Юридический адрес.

- OIDMO (уникальный номер медицинской организации).
- Уровень медицинской организации.

Об информационной системе медицинской организации требуется знать:

- Наименование ИС.
- Тип ИС.
- Разработчик ИС.
- Наличие модели угроз.
- Наличие аттестата соответствия ФТСЭК (аттестат соответствия

федеральной службы по техническому и экспортному контролю).

- ГИС (геоинформационная система).
- Объект КИИ (критическая информационная инфраструктура).
- Категория КИИ (критическая информационная инфраструктура).

Информация о территориально-выделенном структурном подразделении(ТВСП) состоит из основных сведений о подразделении, информации о каждом рабочем месте, о всех созданных сетевых инфраструктур и информации о диагностическом оборудовании.

Общая информация о ТВСП состоит из:

- Наименование.
- Тип.
- Адрес.
- OID ТВСП (уникальный номер ТВСП).
- Координаты ТВСП.
- Этажность.
- Форма собственности.
- Состояние здания.
- Планируется ли реорганизация.

Сведения о подразделениях ТВСП:

- Амбулаторная помощь.
- Оказание ПМСП (первичная медико-санитарная помощь).

- Наличие регистратуры.
- Стационарная помощь.
- КДЛ (клиническая информационная инфраструктура).
- СМП (скорая медицинская помощь).
- НМП (неотложная медицинская помощь).
- Инструментальная диагностика.
- ЛЛО (льготное лекарственное обеспечение).
- Направление на МСЭ (медико-социальная экспертиза).
- Оформление листов временной нетрудоспособности.
- Отделение для работы с ССЗ (сердечно-сосудистые заболевания).
- Онкология.

Информация о рабочем месте содержит в себе данные о подключенной клавиатуре, компьютерной мыши, мониторе, системном блоке и программном обеспечении.

Данные о системном блоке состоят из:

- Год приобретения.
- Инвентарный номер.
- Модель процессора.
- Модель ОЗУ.
- Объем ОЗУ.
- Система хранения.
- Объем Хранения.
- Модель системны хранения.
- Операционная система.
- Кабинет.

Данные о мониторе состоят из:

- Модель.
- Диагональ.
- Инвентарный номер.

- Год приобретения.
- Клавиатура.
- Модель.
- Инвентарный номер.
- Год приобретения.

Данные о компьютерной мыши состоят из:

- Модель.
- Инвентарный номер.
- Год приобретения.

Данные о программном обеспечении состоят из:

- Наименование.
- Разработчик.
- Наличие в реестре отечественного ПО.
- Версия.
- Стоимость.
- Год приобретения.

Информация о сетевой инфраструктуре ТВСП состоит из данных о локальной сети, о протоколе защиты сети, о данных для доступа к сети интернет и диагностическом оборудовании.

Информация о сетевой инфраструктуре включает в себя информацию, связанную с локальной сетью, её защитой и наличием доступа к сети интернет:

Информация о доступе к Интернет:

- Статус подключения к Интернет.
- Технология подключения к Интернет.
- Пропускная способность канала.
- Пропускная способность по тарифу.
- Поставщик услуг.
- Стоимость услуг в месяц.
- Участие в ФП (федеральный проект).

- Информационная инфраструктура.
- Защищенная сеть передачи данных.
- Статус подключения к ЗСПД (защищенная сеть передачи данных).
- Технология подключения к ЗСПД (ViPNet Coordinator, ViPNetHW1000, ViPNetHW100, ViPNetClient.).

Информация о локальной вычислительной сети (ЛВС):

- Количество портов ЛВС.
- Сетевое оборудование.
- IP-адресация.
- Наличие серверного помещения.
- Площадь.
- Место размещения.
- Соответствие требования к серверным помещениям.
- Наличие схемы ЛВС.

Диагностическое оборудование состоит из таких данных как:

- Наименование оборудования.
- Тип оборудования.
- Год выпуска.
- Место размещение оборудования.
- Наличие подключения к ЛВС.
- Количество исследований в год.
- Средний объем одного исследования.
- Поддержка DICOM 3.0. (медицинского стандарта создания, хранения, передачи и визуализации цифровых медицинских изображений и документов обследованных пациентов)
- Признак активации.
- Наличие АРМ (автоматизированное рабочее место) диагноста.

Информация о медицинских организациях республики Хакасия, указанная выше, включает 13 блоков, содержащих в себе от 3 до 13 элементов данных.

Требуется постоянно работать с большим объемом данных, которые должны быть актуальными, а также иметь возможность управления ими, поэтому наиболее подходящим инструментом для достижения цели будет являться веб-приложение для хранения и взаимодействия с данными.

1.3 Бизнес-процессы предметной области

В каждой медицинской организации работает человек, выполняющий обязанности системного администратора, который отвечает за сбор, хранение, передачу информации по своей организации в ГБУЗ РХ «РМИАЦ», который в свою очередь использует данную информацию для модернизации.

Сотрудник отдела управления инфраструктурой ГБУЗ РХ «РМИАЦ» будет являться администратором разрабатываемого приложения. У администратора будет полный доступ к элементам системы, доступна фильтрация информации, а также отчетность о наполненности данных в базе (рисунок 2).

Для разработки информационной системы использовались моделирование бизнес-процессов в нотациях IDEF0, DFD и USE-CASE. С помощью моделей определено функциональное содержание системы, роли пользователей, потоки данных.

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. В нашем случае – какой функционал доступен каждому из администраторов.

IDEF0 — графическая нотация, предназначенная для формализации и описания бизнес-процессов, протекающих в системе.

DFD — графическая нотация, предназначенная для описания потоков данных в проектируемой информационной системе.

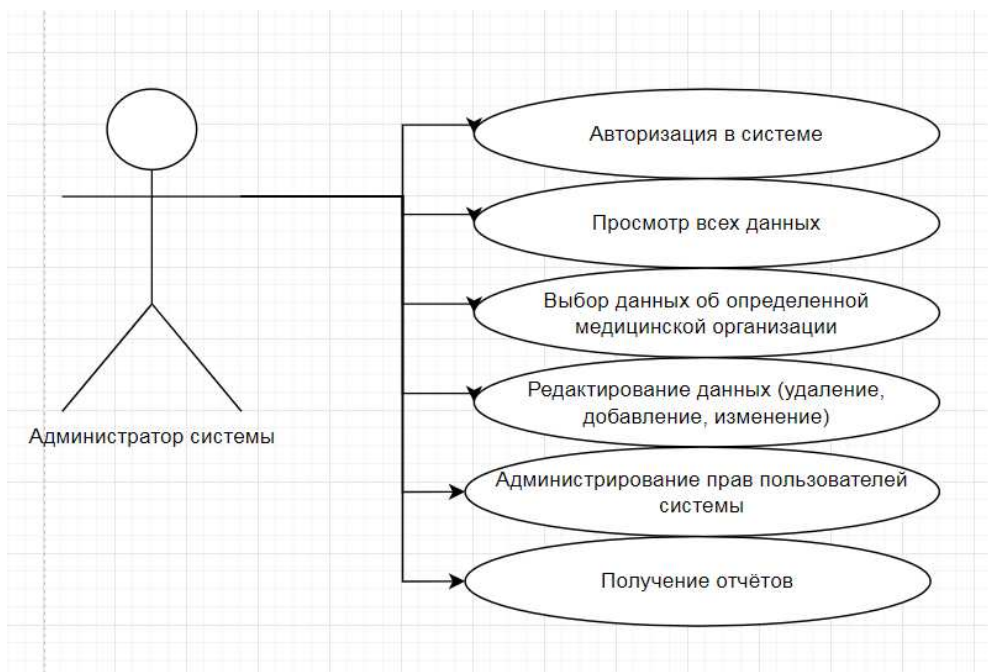


Рисунок 2 – Функционал администратора системы

Администратор системы осуществляет авторизацию пользователей в системе и назначает права доступа к данным, имеет доступ к данным по всем медицинским организациям республики Хакасия, может редактировать их и получать отчёты.

Администратор медицинской организации должен иметь доступ только к данным о своей организации (рисунок 3).

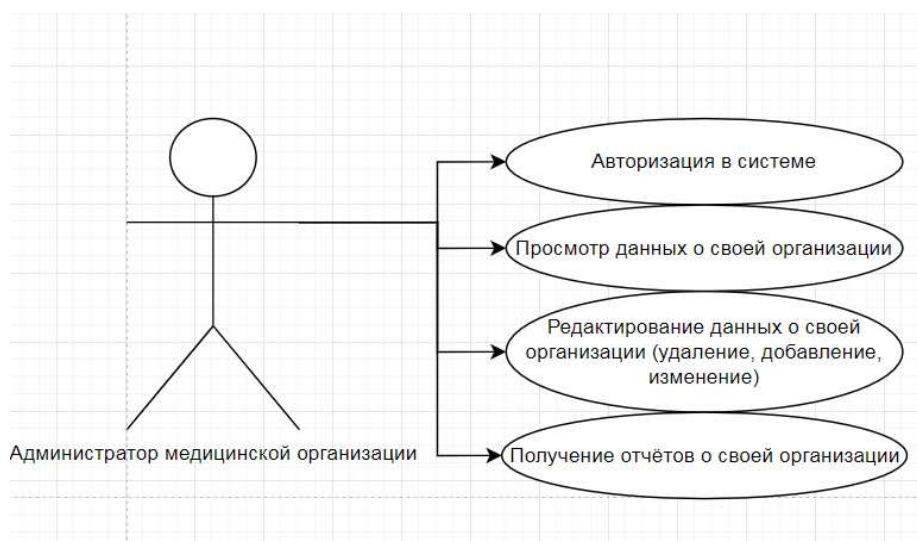


Рисунок 3 – Функционал администратора медицинской организации

Администратор медицинской организации после назначения ему прав доступа к информации по определенной медицинской организации имеет возможность просмотра, редактирования её и получения отчёта по ней.

На рисунке 4 изображена модель в нотации IDEF0 разрабатываемой системы. В создаваемую систему должна поступать информация о пользователе в виде логина и пароля, а также информация о медицинской организации. Пользователями системы являются главный администратор (сотрудник ГБУЗ РХ «РМИАЦ») и администратор медицинской организации. Контроль заполнения, отображения информации отвечает система управления данными. В итоге информационная система сохраняет внесенную информацию, выводит отчёт о медицинской организации или организациях, а также отображает данные для их просмотра.

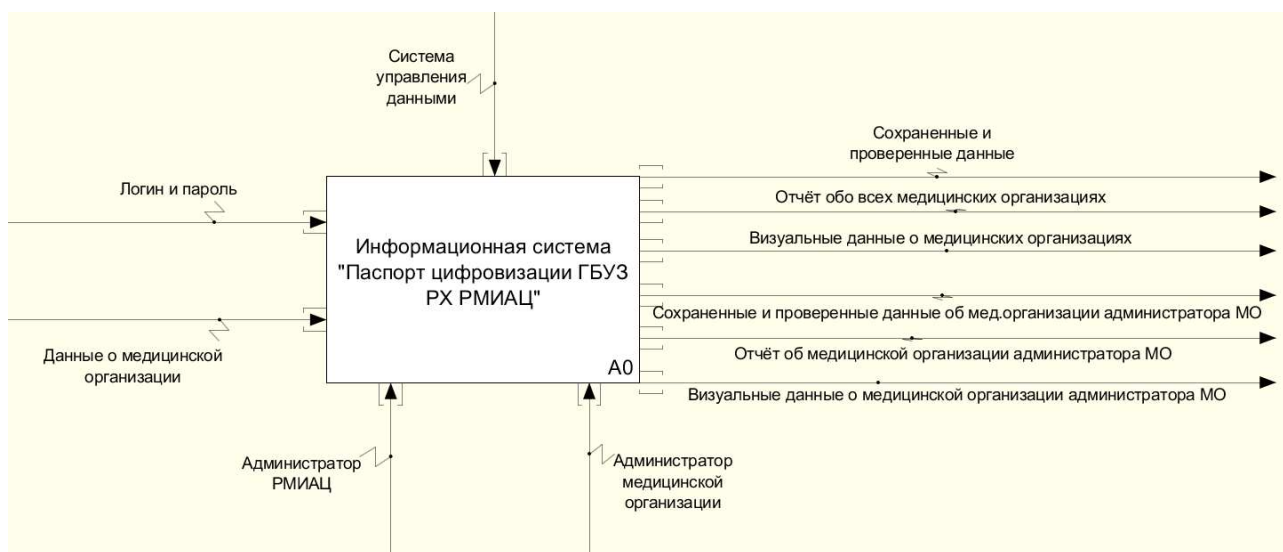


Рисунок 4 –Функциональная модель системы

На рисунке 5 изображена декомпозиция модели в нотации IDEF0. Она отображает последовательность процессов в системе. В самом начале пользователь обязан авторизоваться в системе, чтобы система смогла определить роль пользователя. Если пользователь является главным администратором, то он сможет взаимодействовать со всеми данными, находящимися в базе данных. Если пользователь является администратором

медицинской организации, то система отобразит ему данные только о ней и позволит взаимодействовать с этими данными. Каждому из администраторов доступна функция формирования отчётов, просмотра и редактирования данных. Разница лишь в том, как указывалось ранее, главный администратор работает сразу со всей информацией, находящейся в базе данных.

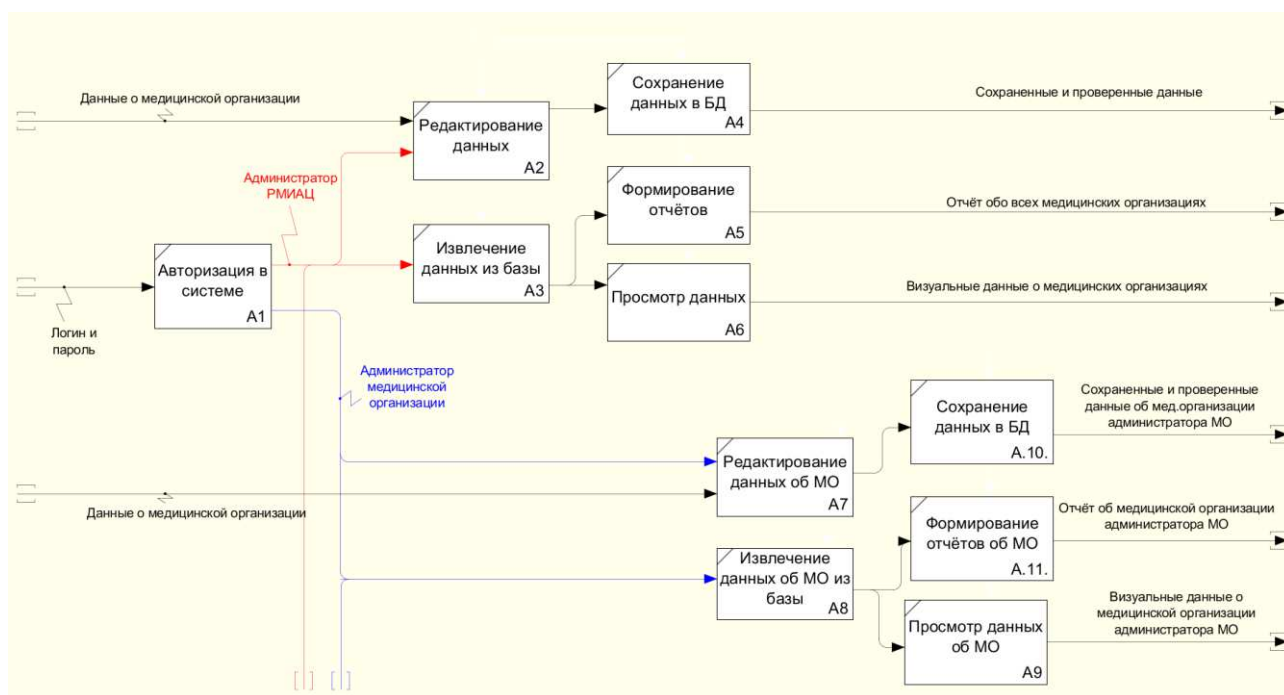


Рисунок 5 – Декомпозиция функциональной модели

Графическая нотация DFD отображающая информационные потоки в системе, изображена на рисунке 6.

Основной целью информационной системы является сбор данных о медицинских организациях, соответственно основными информационными потоками будут являться процессы передачи данных в систему и получения данных из системы. Взаимодействовать с данными будут: главный администратор, информационная система и администратор медицинской организации.

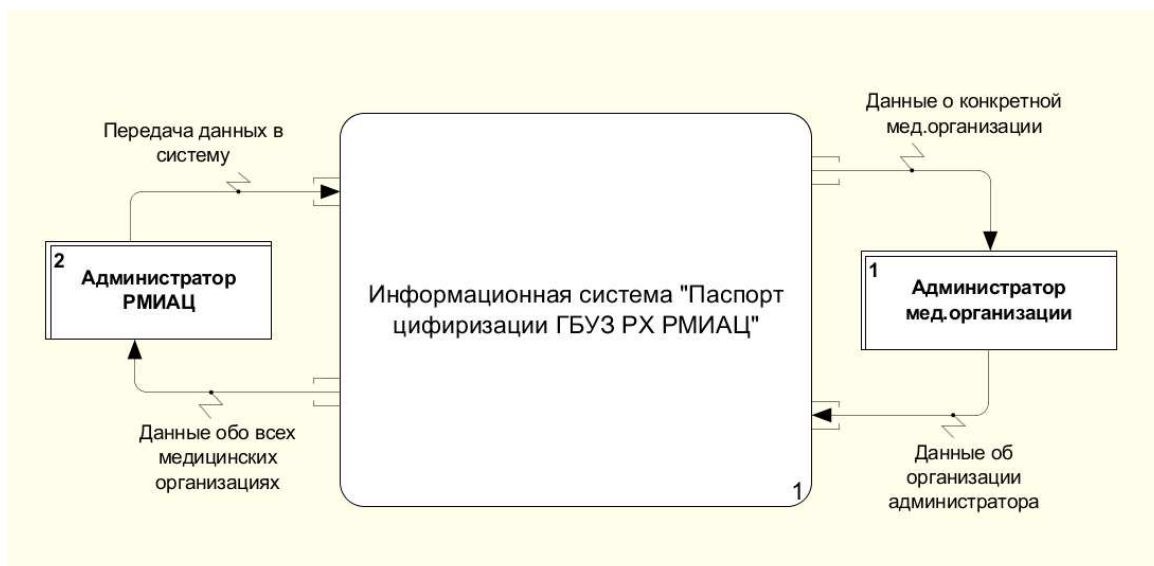


Рисунок 6 – Контекстная диаграмма в нотации DFD

Рассмотрим декомпозицию модели в нотации DFD (рисунок 7). Первым делом администратор должен авторизоваться в системе. Во время авторизации пользователя происходит сравнение введенных логина и пароля с имеющимися в базе данных зарегистрированных пользователей. Затем система отображает пользователю информацию о медицинских организациях, находящихся в базе. При необходимости администратор вносит изменения в базу данных, а также может сформировать отчет.

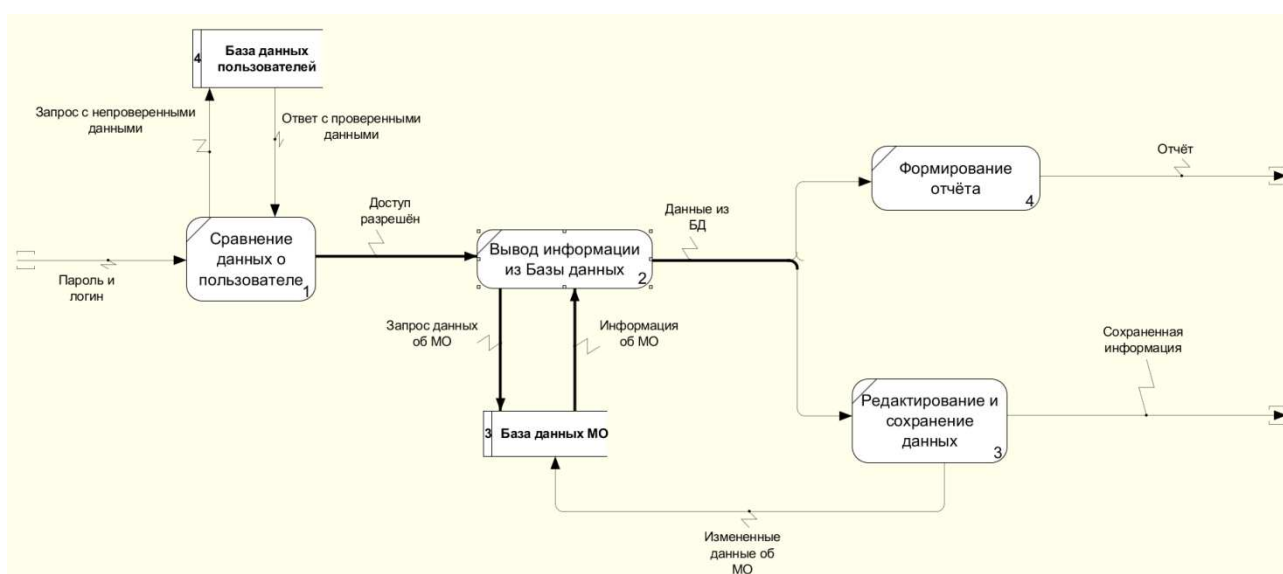


Рисунок 7 – Декомпозиция контекстной диаграммы в нотации DFD

1.4 Характеристика IT-инфраструктуры ГБУЗ РХ «РМИАЦ»

Определив необходимый функционал системы, требуется рассмотреть IT-инфраструктуру организации, где будет расположена система. ГБУЗ РХ «РМИАЦ» основана с целью осуществления процесса информатизации и цифровизации медицинских организаций республики Хакасия. Для осуществления этой цели в ГБУЗ РХ «РМИАЦ» расположено множество компьютеров и серверов на основе Linux, поддерживающих любые языки программирования. Среднестатистический ПК сотрудников ГБУЗ РХ «РМИАЦ» обладает следующими характеристиками: 4-х ядерный процессор для работы с 8-ю потоками данных, операционная система Linux или Windows 10; оперативная память 8Гб, запоминающее устройство 215Гб; имеется доступ к сети интернет, подключён в локальную сеть ГБУЗ РХ «РМИАЦ» для доступа к серверу.

ИС «Паспорт цифровизации ГБУЗ РХ «РМИАЦ» является клиент-серверным приложением, поэтому для него необходим сервер для размещения. Веб-приложение будет расположено на сервере ГБУЗ РХ «РМИАЦ», чтобы был доступен удаленный доступ к приложению из разных медицинских организаций.

У пользователя (администратора) системы, должен быть ПК с возможностью работы с веб-браузером. Для комфортной работы с веб-браузером подойдет ПК со следующими характеристиками: 2-х ядерный процессор, 4Гб оперативной памяти, запоминающее устройство на 124ГБ, обязательно наличие интернета.

1.5 Цель и задачи разработки проекта

В настоящее время у ГБУЗ РХ «РМИАЦ» отсутствует система, которая удаленно осуществляет сбор информации о медицинских организациях. Таким образом, актуальность данной темы связана с разработкой веб-приложения,

которое способное выполнять функции сбора, хранения, обработки данных. Перед разработкой информационной системы составлены требования к системе, которые отображены в таблице 2.

Таблица 2 – Требования к системе

№	Требования	Приоритет
1	Взаимодействие с системой через сеть интернет	Высокий
2	Возможность подключения к системе через ПК	Высокий
3	Возможность подключения к системе через мобильное устройство	Низкий
4	Хранение информации в базе данных	Высокий
5	Формирование отчётов	Высокий
6	Наличие авторизации	Высокий
7	Наличие ролей у приложения: главный администратор, администратор медицинской организации	Средний
8	Возможность администратору РМИАЦ удалять, редактировать, добавлять все данные в системе.	Высокий
9	Возможность администратору медицинской организации удалять, редактировать, добавлять данные только своей системы.	Средний
10	Возможность администратору РМИАЦ изменять настройки веб-приложения	Высокий

Название системы:

– Информационная система «Паспорт цифровизации ГБУЗ РХ «РМИАЦ».

Назначение системы:

– Сбор информации, связанной с оснащением медицинских организаций компьютерной техникой, средствами телекоммуникации и характеристиками данных цифровых устройств, направлениями деятельности организации.

Цель разработки системы:

- автоматизация сбора информации о каждом устройстве в муниципальных учреждениях.

Задачи разработки системы:

- Провести анализ предметной области.
- Провести сравнительный анализ и сделать выбор инструментальных средств разработки.
- Разработать веб-приложение.
- Рассчитать экономическую эффективность приложения.

1.6 Анализ типовых проектных решений и выбор инструментальных средств разработки

На рынке цифровых продуктов уже существуют системы анализа данных, системы хранения и сбора информации. Google Data Studio – программа для работы со сквозной аналитикой данных. TotalNetworkInventory – система для сбора и хранения характеристик устройств, подключенных к локальной сети. У существующих систем реализована только часть требуемого функционала, нет возможности использования собственной базы данных.

Разрабатываемая система отличается от существующих тем, что она создаётся под специализированную задачу – для сбора, хранения информации не только об информационных устройствах, используемых в медицинских организациях, но и хранения данных о выполняемых организацией функциях, характеристик организации.

В целях обеспечения информационной безопасности, информация о характеристиках сетей муниципальных учреждений обязана храниться на внутренних серверах ГБУЗ РХ «РМИАЦ» и быть доступна определенному кругу лиц.

Поскольку программы аналоги не имеют требуемый функционал, и информация, собираемая ГБУЗ РХ «РМИАЦ», предназначена для внутреннего

использования –необходимо разработать требуемое веб-приложение собственными силами.

ИС «Паспорт цифровизации ГБУЗ РХ «РМИАЦ» будет представлять собой веб-серверное приложение, которое будет храниться на серверах ГБУЗ РХ «РМИАЦ». Для того, чтобы ПК, находящиеся в медицинских организациях, смогли подключаться к приложению по протоколу HTTPS, необходимо выбрать язык программирования, который позволит производить обработку большого количества данных. Для реализации этого подходят языки программирования PHP, JavaScript и Python.

Все предложенные языки программирования могут быть использованы при реализации сбора, обработки, вывода и получения информации.

PHP-скриптовый язык, который используется только при создании скриптов (функций) веб-приложений. Данный язык не может быть интегрирован в компьютерное или мобильное приложение, не имеет модулей расширения.

JavaScript используется для разработки веб-сайтов, веб-серверных приложений, компьютерных игр и многих других приложений, совместим с другими языками программирования. В основном используется на стороне пользователя, в отличие от Python и PHP.

Python – это универсальный язык программирования, позволяет реализовать как интерфейс взаимодействия, так и запросы для работы с базой данных. Поддерживает множество дополнительных библиотек для работы с данными и фреймворков. Например, на Python реализованы приложения Nasa, YouTube, а также Google Search. Python предоставляет функции для обработки неструктурированной информации, которая включает в себя голосовые, текстовые и графические данные. Python также может обрабатывать данные в разных файлах, таких как CSV, XML, HTML, SQL, JSON и т.д.

Поскольку язык программирования Python универсален и предназначен для обработки большого количества данных, он подходит для создания информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»

Чтобы уменьшить количество ошибок во время создания программы, было принято решение использовать фреймворк, который бы не позволил нарушить логику создания веб-приложения.

Фреймворк – набор программ, правил, которые ускорят разработку больших и нагруженных проектов. Обычно любой фреймворк содержит в себе базовые программные модули, а все остальные компоненты реализуются на основе шаблонов этих модулей.

Так как для проекта будет использоваться язык программирования Python, то фреймворки также должны поддерживать данный язык разработки.

Для Python существуют следующие фреймворки для разработки веб-приложений: Django, Pyramid, TurboGears, Flask

Django является бесплатным веб-фреймворком для разработки сложных веб-систем. При разработке использует архитектуру MVT (*Model-View-Templates*). Model - отвечает за доступ к данным из базы данных. View – отвечает за бизнес-логику приложения, определяет какие данные будут отправлены определенному шаблону. Templates – отвечает за шаблоны отображения данных.

Django реализован по принципу DRY (don't repeat yourself). Данный принцип разработки служит для снижения одинаковой информации. Django поощряет свободное связывание и строгое разделение частей приложения. Это позволяет без ущерба вносить изменения в отдельную часть программы. В функциях созданного приложения важно отделить бизнес-логику от логики отображения с помощью шаблонной системы, также необходимо поступить и с доступом к данным, чтобы обеспечить корректное функционирование.

Чаще всего Django применяют для разработки платформ управления документооборотом, платформ анализа данных. Для Django создано множество дополнительных библиотек, упрощающих работу с данными. Присутствует встроенный интерфейс автоматического администрирования.

Фреймворк Flask в отличие от Django используется для небольших проектов. Данный фреймворк предоставляет всего две возможности:

возможность работать с шаблонами страниц и возможность отладки программы в реальном времени.

Фреймворк TurboGearsиспользуется для работы с базами данных так же, как и Django использует модель MVT. Не имеет четких инструкций по созданию приложения в отличие от Django. Отсутствуют стандартные настройки шаблонов. Возможны случаи (при одинаковом наименовании шаблона и модели), когда из шаблона отображения можно случайно изменить информацию в базе данных. Плюсы данного фреймворка в том, что на нём можно реализоватьнебольшие проекты, например сайт учёта выдачи товаров.

Фреймворк Pyramid является полным аналогом Django, но также как и TurboGears не следует чётким правилам разработки. Имеется возможность менять шаблоны работы фреймворка.

В таблице 3 отображено сравнение характеристик данных фреймворков.

Таблица 3 – Сравнение характеристик фреймворков

Характеристика	Django	Flask	TurboGears	Pyramid
Использует архитектуру MVT	Да	Нет	Да	Да
Реализован по принципу DRY (don'trepeatyourself)	Да	Нет	Нет	Да
Масштабируется при помощи добавления новых модулей	Да	Нет	Нет	Да
Поддерживает множество языков отображения	Да	Нет	Нет	Да
Возможность отладки кода в реальном времени	Да	Да	Нет	Да
Встроенные шаблоны отображения	Да	Нет	Нет	Да
Встроенная панель администратора	Да	Нет	Нет	Нет

Объектно-реляционное отображение (ORM)	Да	Нет	Нет	Да
---	----	-----	-----	----

После анализа написанной информации было принято решение разрабатывать веб-приложение при помощи фреймворка Django, так как в отличие от других фреймворков он более устойчив к работе с большими данными, имеет в наличии готовый шаблон панели администратора, при создании проекта автоматически генерирует защиту веб-приложения от стороннего воздействия на проект. Имеет возможность загрузки дополнительных модулей разработки, что в дальнейшем позволит модернизировать, масштабировать и улучшать созданную информационную систему.

1.7 Выводы по разделу «Анализ предметной области для создания информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»»

Вданном разделе ВКР описана деятельность ГБУЗ РХ «РМИАЦ». Разработана функциональная модель IDEF0, диаграмма USECASE, контекстная диаграмма в нотации DFD для информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ». Описаны основные функции разрабатываемой информационной системы. Выбраны средства разработки ИС «Паспорт цифровизации ГБУЗ РХ «РМИАЦ».

2 Разработка информационной системы

2.1 Установка необходимого ПО

Для разработки веб-приложения установлена среда разработки PyCharm и интегрированная среда разработки IDLEPython совместно с менеджером пакетов pip.

Pip – это система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python. Лучший способ установить IDLE Python и pip на Windows — это скачать официальный установщик с сайта python.org. Чтобы сделать это, нужно открыть браузер и перейти по ссылке <https://python.org/>. (рисунок 8).

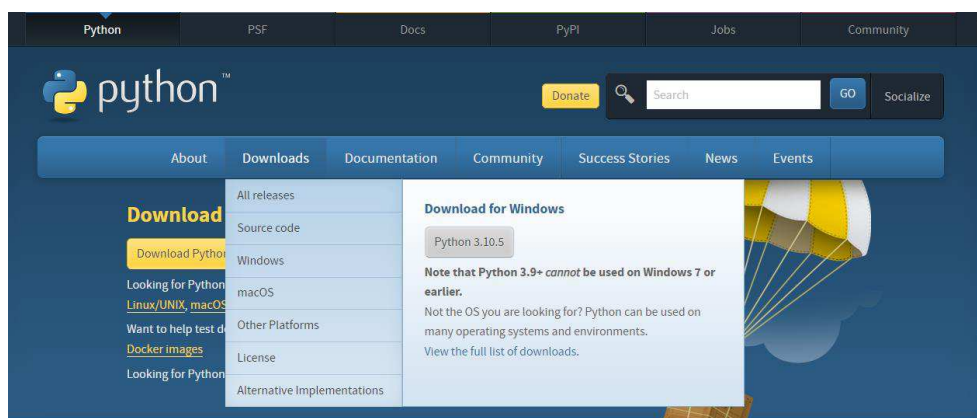


Рисунок 8 – Страница загрузки Python

После завершения загрузки установщика выбираем «AddPythontoPATH» (рисунок 9). Если не нажать «Add Python to PATH», то язык программирования не будет иметь доступа к работе с командной строкой. После чего запущен процесс установки IDLE Python

Установка IDLE Python позволяет устранить ошибки при работе Python в командной строке или консоли разработчика.



Рисунок 9— Установка IDLE Python

Среда разработки PyCharm установлена с сайта <https://www.jetbrains.com/ru-ru/pycharm/> (рисунок 10). Professional PyCharm— это платная версия с полным набором функций (позволяет работать с html, javascript, css и т.д). Она подходит для профессиональной разработки. CommunityPyCharm бесплатная и обладает базовыми возможностями.

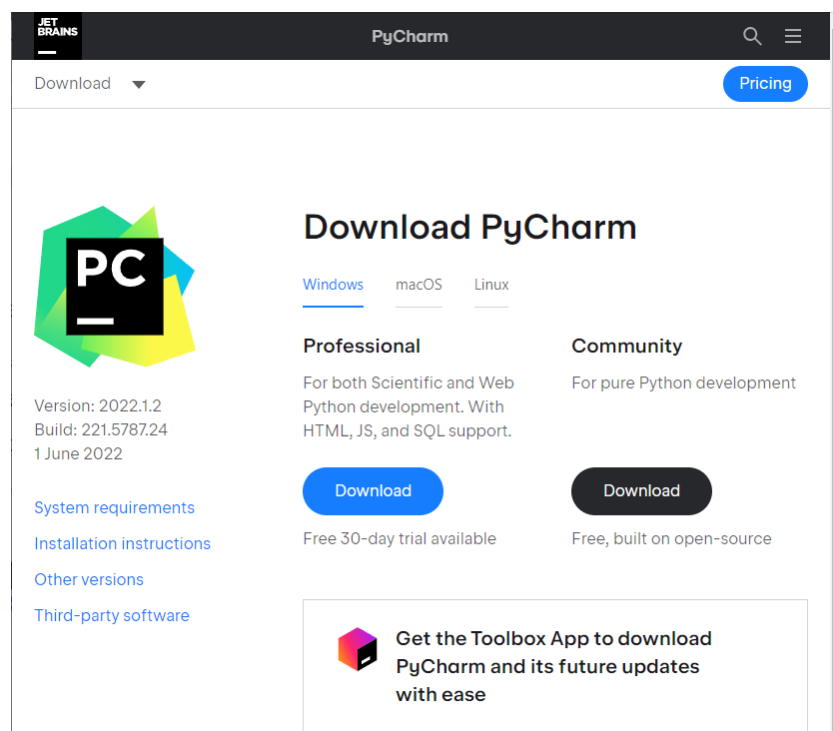


Рисунок 10 – Страница загрузки PyCharm

Рекомендуется устанавливать PyCharm по рекомендуемому пути “C:\ProgramFiles\JetBrains\PyCharm”. Это следует делать для того, чтобы избежать ошибок во время работы приложения, а именно во время создания локальных окружений для проектов.

При первом запуске версии ProfessionalPyCharm система потребует авторизоваться, чтобы проверить наличие лицензии.

После авторизации и активации в окне взаимодействия создаётся новый проект (рисунок 11).

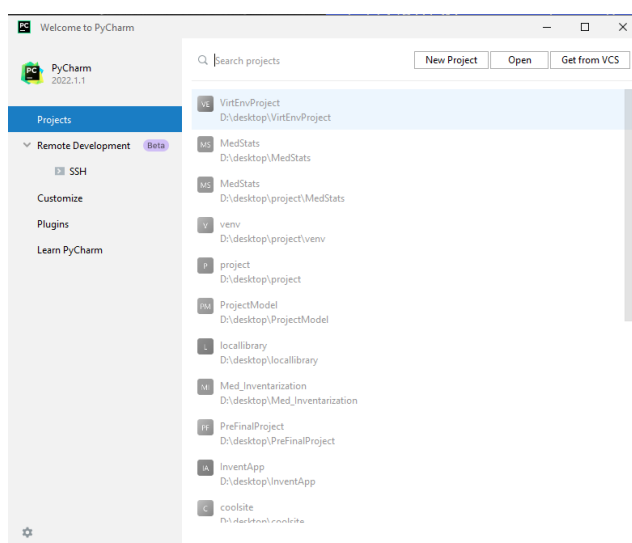


Рисунок 11 – Окно создание проектов в PyCharm

2.2 Создание стартового проекта на Django

Завершив установку необходимых программ, приступаем к разработке проекта. Создаём проект «NewProject» и определяем расположение проекта (рисунок 12). Среда разработки PyCharm автоматически создаёт виртуальное окружение для проекта и загружает туда основные файлы для работы с фреймворком Django, никакие сторонние файлы не смогут влиять на разрабатываемый проект (рисунок 13).

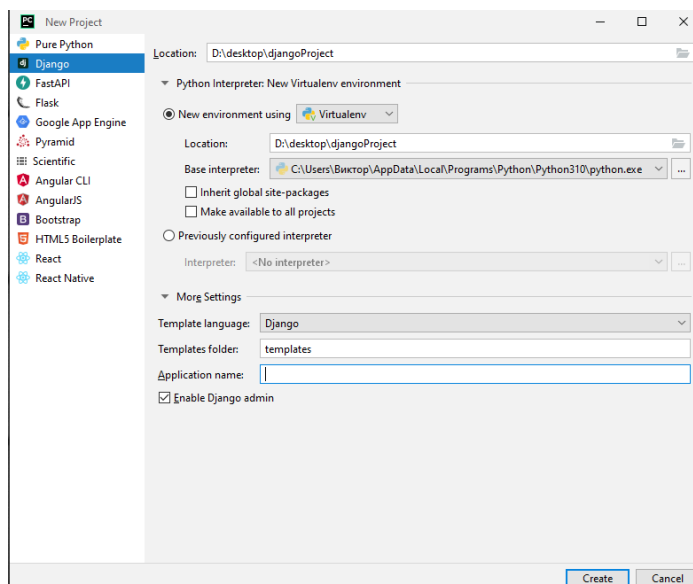


Рисунок 12 – Выбор расположения проекта Django

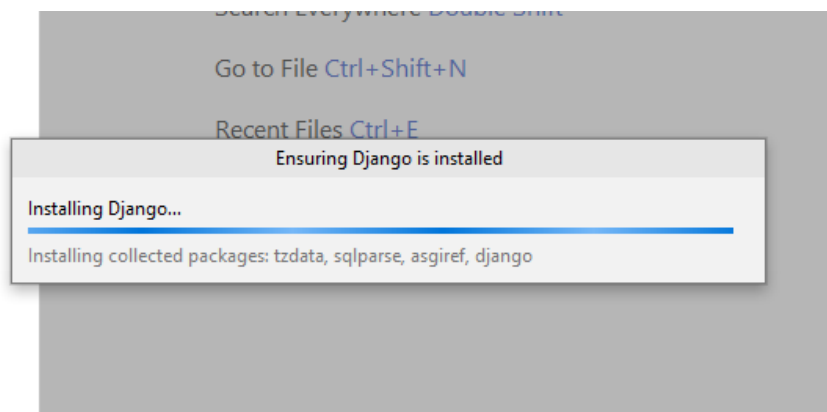


Рисунок 13– Автоматическая установка фреймворка Django

После установки основных файлов фреймворка Django создано основное приложение, включающее в себя стандартные настройки веб-приложения. Затем создана папка для размещения проекта (рисунок 14). Правильность установки проверена запуском проекта. Для запуска проекта из расположения файла `manage.py` введена команда: `Python manage.py runserver`.

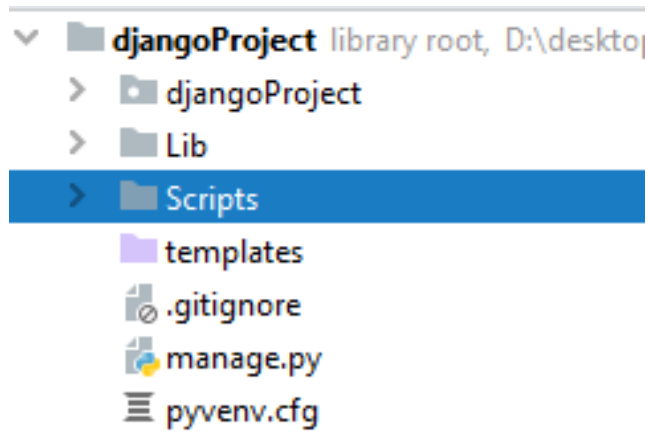


Рисунок 14 – Созданный проект Django

При установке без ошибок отображаются данные о том, что проект запущен на локальном сервере <http://127.0.0.1:8000/> (рисунок 15), и, пройдя по этому адресу, отобразилось окно с успешным запуском (рисунок 16).

```
Terminal: Local x + v
(djangoProject) PS D:\desktop\djangoProject> python manage.py runserver
python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s):
admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
June 07, 2022 - 03:56:41
Django version 4.0.5, using settings 'djangoProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Рисунок 15 – Сообщение о запуске Django



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

Рисунок 16 – Окно уведомления созданного проекта

2.3 Создание приложений, основного html шаблона и главного администратора системы

После корректной установки фреймворка созданы отдельные приложения: для работы с авторизацией; работой с базой данных; для формирования отчёта в файл excel; для работы с отчётами; для вывода информации из базы данных.

Приложения созданы путём ввода в консоль команды “django-admin createapp название приложения»” (рисунок 17).

```
(VirtEnvProject) PS D:\desktop\VirtEnvProject\medinentarization> django-admin createapp database
```

Рисунок 17 – Создание приложения database

Созданные приложения отображены на рисунке 18.

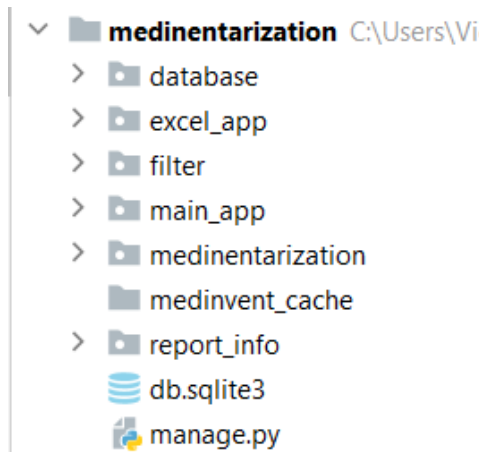


Рисунок 18– Созданные приложения

Созданное приложение database отвечает за настройку работы с базой данных, в этом приложении будут храниться шаблоны отображения данных.

Приложение excel_app отвечает за передачу данных из таблиц базы данных в таблицы excel.

Приложение `filter` отвечает за сортировку информации, полученной из базы данных.

Приложение `main_app` является основным и отвечает за хранение статических файлов, шаблонов отображения информации, за перенаправление на авторизацию.

Приложение `report_info` отвечает за отображение статистики заполняемости информации в базе данных.

Приложение `medinventarization` автоматически создано фреймворком Django. Оно хранит в себе общие настройки проекта, а именно: маршрутизацию между приложениями, настройки веб-безопасности, настройки отображения данных в браузере.

Активация созданных приложений произошла после того, как в приложении `medinventarization` в файле `settings` указали созданные приложения путем добавления их названий в список `INSTALLED_APPS` (рисунок 19) названий приложений `main_app` и `database`.

```
INSTALLED_APPS = [  
    'filter',  
    'excel_app',  
    'report_info',  
    'database',  
    'main_app',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

Рисунок 19 – Активация приложений

В файле настроек также совершена проверка, чтобы был включён режим разработки (рисунок 20) и подключена база данных (ВКР Сагояков Александр Дмитриевич) (рисунок 21).

```
# SECURITY WARNING: don't run with debug turned on in production!  
DEBUG = True
```

Рисунок 20 – Режим разработки

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

Рисунок 21 – Подключение базы данных

Приложению авторизации Django.contrib.auth объявлены действия после событий авторизации и выхода из приложения (рисунок 22).

```
LOGIN_REDIRECT_URL = 'main' #127.0.0.1:8000/main_app/  
LOGOUT_REDIRECT_URL = '/'
```

Рисунок 22 – Указание путей после авторизации и выхода

Закончив настройку приложения, объявлены маршруты между шаблонами и приложениями. Объявление маршрутов всегда происходит в файле urls.py (рисунок 23). Данный файл создан вручную во всех приложениях. Urls.py файл работает только с теми, приложениями, в которых он расположен. В файле urls.py, расположенном в главном приложении medinventarization, объявлены веб-маршруты до созданных приложений (рисунок 24).

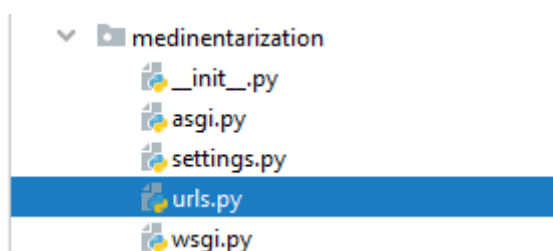


Рисунок 23 – Файл urls

Приложение для работы с панелью администратора создаётся автоматически по адресу “http/(название сервера)/admin/”, путем указания пути до модуля админ-панели - «path(‘admin/’,admin.site.urls)» (рисунок 24, строка кода 22)

```
21 urlpatterns = [  
22     path('admin/', admin.site.urls),  
23     path('', include("main_app.urls")),  
24     path('database/', include("database.urls")),  
25     path('accounts/', include('django.contrib.auth.urls')),  
26     path('report_info/', include("report_info.urls")),  
27     path('excel_app/', include("excel_app.urls")),  
28     path('filter', include("filter.urls")),
```

Рисунок 24 – Маршруты до приложений

Далее перейдя в приложение main_app, созданы маршруты в этом приложении. Создание маршрутов происходило в файле urls.py. (рисунок 25).

```
1 from django.urls import path, re_path  
2 from django.views.decorators.cache import cache_page  
3  
4 from . import views  
5  
6 urlpatterns = [  
7     path('', views.redirect_view),  
8     path(r'main_app/', views.index, name='main'),  
9     path(r'main_app/<int:pk>/', views.index, name='main'),  
10 ]
```

Рисунок 25– Настройка маршрутов main_app

Функция path сопоставляет шаблоны и адреса URL, которые указаны перед вызовом функции. На главном маршруте происходит вызов функции redirect_view (рисунок 26) из файла views. Данная функция при переходе на страницу “http://127.0.0.1:8000/” совершает автоматический переход на адрес авторизации “http://127.0.0.1:8000/acount/login/”.

Закончив с настройкой маршрутов в файле views.py объявлены функции, которые заранее обозначены в urls.py, после маршрутов.

```

def redirect_view(request):
    if request.user.is_authenticated == False:
        request = redirect('accounts/login')
    else:
        request = redirect('main')
    return request

```

Рисунок 26 – Файл views в папке main_app

Чтобы пользователь видел не пустую страницу, а страницу авторизации, создан шаблон отображения данной страницы. Все шаблоны создаются в фиксированном месте, в папке templates (шаблоны). При наличии двух одинаковых названий у шаблонов Django выдаёт ошибку, а чтобы этого не допустить, создана папка registrations в папке templates. В созданной папке объявлен шаблон для авторизации login.html.

Для того, чтобы избежать повторения одинаковой информации в html шаблонах, вначале создаётся главный html шаблон, где объявляется структура html шаблонов, в этом же шаблоне создаётся дизайн приложения.

В папке templates создана новая папка main, чтобы однозначно указать расположение шаблонов (рисунок 27).

После создания файла html для основного шаблона под названием base.html подключены CSS стили отображения данных. Чтобы в html были активны стили, необходимо ссылаться на css файл. Для возможности использования статических файлов (файлов, которые хранятся в Django приложении) необходимо создать и объявить в главных настройках Django расположение всех статических файлов static (рисунок 28). В статическом хранилище могут располагаться обычные файлы, картинки, скрипты и т.д.

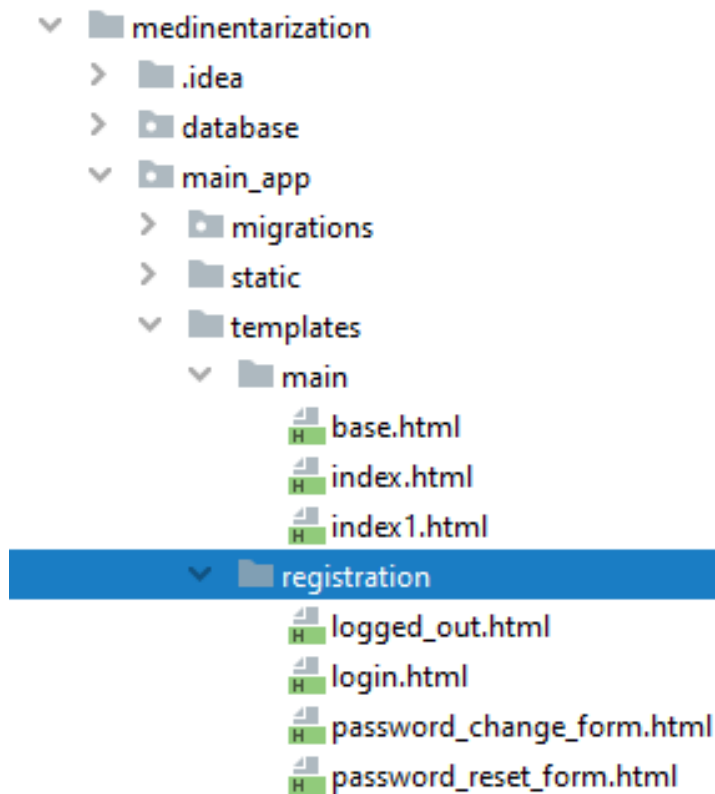


Рисунок 27 – Создание шаблонов

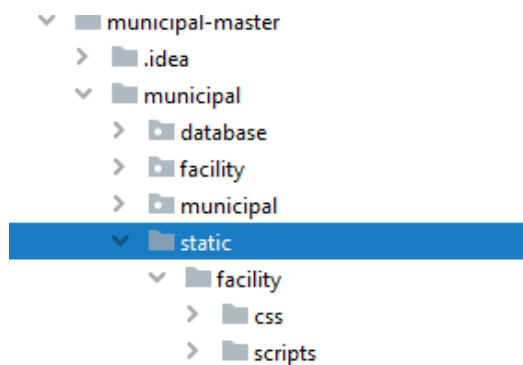


Рисунок 28 – Хранилище статических файлов

В переменной `Static_URL` указано наименование созданной папки (рисунок 29).

```
STATIC_URL = '/static/'  
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

Рисунок 29 – Объявление хранилища статических файлов

Далее началась работа с основным шаблоном `base.html` и `css`.

Первым делом этот файл объявили доступным для других шаблонов. Сделано это при помощи переменной `jinja2 {% loadstatic%}`. При помощи этой переменной другие файлы `html` получают возможность копировать структуру главного шаблона `base.html`.

После этого переданы в шаблон `css` файлы, собственное изображение, `bootstrap 5` и необходимые шрифты (строчки кода 6-9 на рисунке 30).

```
1 <!DOCTYPE html>
2 {% load static %}
3 <html lang="ru">
4 <head>
5   <meta charset="UTF-8">
6   <link rel="stylesheet" href="{% static '/css/bootstrap.css' %}">
7   <link rel="stylesheet" href="{% static '/css/bootstrap.min.css' %}">
8   <link rel="stylesheet" href="{% static '/css/sidebars.css' %}">
9   <link rel="stylesheet" href="{% static '/css/styles.css' %}">
10  <title>{% block title %}{% endblock %}</title>
11 </head>
```

Рисунок 30 – Head в основном шаблоне

После объявления всех стилей и изображений создано тело проекта. Для этого тэгом `<body>` указано начало тела шаблона, и после тэга размещены объекты. Шаблон содержит в себе `header` – верхнюю часть приложения (заголовок), часть `main` – где располагается основная информация (центральный блок), блок `sidebar` – содержит место для отображения функциональных кнопок, закрывающим элементом будет `footer` – нижняя часть проекта. Весь код основным шаблоном `base.html` изображен на рисунках 30-32. Описание кода характеристик основного интерфейса изображено на рисунке 33. Описание кода характеристик блока для расположения кнопок изображено на рисунке 34.

```

12 <body>
13 <div class="container">
14   <header class="d-flex flex-wrap align-items-center
15     justify-content-center justify-content-md-between
16     py-3 mb-4 border-bottom">
17     <a href="/" class="d-flex align-items-center col-md-3 mb-2 mb-md-0 text-dark text-decoration-none">
18       
19     </a>
20     <h4 class="card-text_t">Цифровой паспорт медицинских организаций</h4>
21     {% if request.user.is_authenticated %}
22     <ul class="nav col-12 col-md-auto mb-2 justify-content-center mb-md-0">
23       <li><a href="/" class="nav-link px-2 link-secondary">Главная</a></li>
24       <li><a href="{% url 'admin:index' %}"
25         class="nav-link px-2 link-dark">Профиль | {{user.username}}</a></li>
26       <li><a href="{% url 'filter' %}"
27         class="nav-link px-2 link-secondary">Отчёты</a></li>
28       <li><a href="{% url 'logout' %}"
29         class="nav-link px-2 link-dark">Выход</a></li>
30     </ul>
31     {% endif %}
32   </header>
33 </div>
34 <div class="container sidebar-and-content">
35   {% if request.user.is_authenticated %}
36   <div class="sidebar">
37     {% block SIDE_BAR %}
38     {% endblock SIDE_BAR %}
39   </div>
40   {% endif %}

```

Рисунок 31 –Верхняя и левая часть базового шаблона


```

37 <div class="content-main">
38   {% block header %} {% endblock %}
39   {% block content %} {% endblock %}
40 </div>
41 </div>
42
43 <div class="container footer-bottom">
44   <footer class="d-flex flex-wrap justify-content-between align-items-center py-3 my-4 border-top">
45     <div class="col-md-4 d-flex align-items-center">
46       <a href="/" class="mb-3 me-2 mb-md-0 text-muted text-decoration-none lh-1">
47         
48       </a>
49       <span class="text-muted">© {% now "Y" %} ГБУЗ "РМИАЦ" Хакасии</span>
50     </div>
51
52   </footer>
53 </div>
54 </body>
55 <script src="{% static 'facility/scripts/bootstrap.bundle.min.js' %}"></script>
56 <script src="{% static 'facility/scripts/sidebars.js' %}"></script>
57 </html>
58

```

Рисунок 32 – Центральная и нижняя часть шаблона

```

1  body{ display: flex; flex-direction: column;}
2  .sidebar-and-content{display: flex;flex-grow: 1;}
3  .content-main{flex-grow: 1;padding: 16px;}
4  .on-selected{background-color: #d2f4ea;}
5  .form-control {width: 300px}
6  ul li {list-style: none}
7  a {color: white; text-decoration: none }
8  .text_card{font-weight: bold;margin: 0 0 0 0;
9    padding: 0 0 0 0; display: inline;padding-right: 15px;}
10 .list-group-item {margin-bottom: 6px;}
11 p {display: inline;}
12 .sidebar{margin-right: 10px; margin-top: 20px;width: 170px;
13   align-items: center;text-align: center;}
14
15 .card_text{font-weight: bold;margin: 0 0 0 0;padding: 0 0 0 0;
16   display: inline;padding-right: 8px;}

```

Рисунок 33 – Основной css файл

```

1 body {min-height: 100vh;min-height: -webkit-fill-available;}
2 html {height: -webkit-fill-available;}
3 main {display: flex;flex-wrap: nowrap;height: 100vh;height: -webkit-fill-available;
4     max-height: 100vh;overflow-x: auto;overflow-y: hidden;}
5     .b-example-divder {flex-shrink: 0;width: 1.5rem;
6     height: 100vh;background-color: rgba(0, 0, 0, .1);
7     border: solid rgba(0, 0, 0, .15);border-width: 1px 0;
8     box-shadow: inset 0 .5em 1.5em rgba(0, 0, 0, .1), inset 0 .125em .5em rgba(0, 0, 0, .15);}
9     .bi {vertical-align: -.125em;pointer-events: none;fill: currentColor;}
10    .dropdown-toggle { outline: 0; }
11    .nav-flush .nav-link {border-radius: 0;}
12    .btn-toggle {display: inline-flex;align-items: center;
13    padding: .25rem .5rem;font-weight: 600;
14    color: rgba(0, 0, 0, .65);background-color: transparent;
15    border: 0;}
16    .btn-toggle:hover, .btn-toggle:focus {color: rgba(0, 0, 0, .85);background-color: #d2f4ea;}
17    .btn-toggle::before {width: 1.25em;
18    line-height: 0;content: url(data);
19    transition: transform .35s ease;transform-origin: .5em 50%;}
20
21    .btn-toggle[aria-expanded="true"] {color: rgba(0, 0, 0, .85);}
22    .btn-toggle[aria-expanded="true"]::before {transform: rotate(90deg);}
23
24    .btn-toggle-nav a {
25    display: inline-flex;padding: .1875rem .5rem;margin-top: .125rem;
26    margin-left: 1.25rem;text-decoration: none;}
27    .btn-toggle-nav a:hover,
28    .btn-toggle-nav a:focus {
29    background-color: #d2f4ea;
30    }
31    .scrollarea {overflow-y: auto;}
32    .fw-semibold { font-weight: 600; }
33    .lh-tight { line-height: 1.25; }

```

Рисунок 34 – CSS свойства side-bar блока

В блоках content-main, sidebar (рисунки 35-36) расположены динамически-изменяемые компоненты {% blockcontent %}, {% blockheader %}, {% blockSIDE_BAR %}. В данные компоненты поступает информация из других созданных шаблоновhtml. Функция {% ifrequest.user.is_authenticated %} позволяет проверить авторизацию пользователя, и, если он не авторизован, информация, находящаяся внутри функции, становится недоступной.

Динамически-изменяемые компоненты необходимы для того, чтобы не допустить избыточности данных при создании однотипных шаблонов.

Переменная {{user.username}}, указанная в строке кода 22 на рисунке 31 позволяет получать логин пользователя, который работает с системой.

После создания основного шаблона создан активный шаблон для показа информации с названиемlogin.html. Шаблон расположен в папке

templates/registrations и имеет название login. Данное название и расположение являются обязательными атрибутами шаблона авторизации. Это сделано для того, чтобы модуль авторизации Django заменил собственный стандартный шаблон авторизации на созданный. Созданный шаблон изображен на рисунке 35.

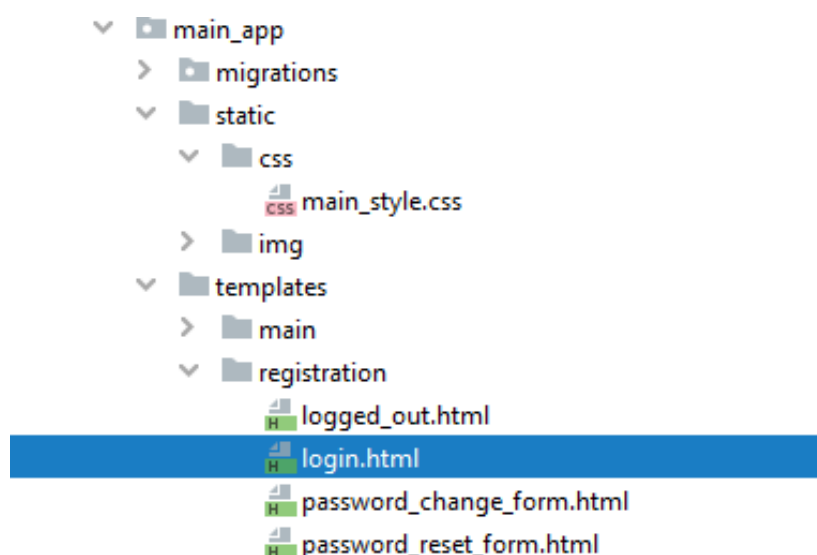


Рисунок 35– Шаблон авторизации

Структура файла login.html изображена на рисунках 36 и 37. В первой строке кода `{% extends «путь до шаблона» %}` указан основной шаблон, откуда дублируется структура для файла авторизации. Затем в динамически-изменяемых компонентах указана структура, которая отобразится в тех местах, где расположены эти элементы в основном шаблоне base.html. При авторизации или любом другом действии, влияющем на структуру сайта, вместе с файлами обязательно передаётся токен `{% csrf_token %}`. Данный токен содержит встроенную в Django функцию защиты от подделки межсайтовых запросов (CSRF). Подделка межсайтовых запросов – вид атак на посетителей веб-сайтов, использующий недостатки протокола HTTP. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер, осуществляющий некую вредоносную операцию. Для осуществления данной атаки жертва должна быть аутентифицирована на том сервере, на

который отправляется запрос, и этот запрос не должен требовать какого-либо подтверждения со стороны пользователя, которое не может быть проигнорировано или подделано атакующим скриптом [4].

Обычно токен добавляется в каждый элемент, в котором используются методы HTML-запросов POST, GET, PUT/DELETE, чтобы пользователь не был перенаправлен на другой сервер.

Вторым обязательным условием является прием данных в модуль авторизации, через формы Django - username и password. Данные формы вызваны в 13 и 20 строчках кода на рисунке 36.

Далее при нажатии на кнопку происходит отправка данных в форму авторизации Django, где происходит хеширование данных, а затем дальнейшая отправка этих данных на сервер. После происходит сравнение полученной информации о пользователе с информацией из базы данных пользователей.

```
1 {% extends 'main/base.html'%}
2 {% block title %} Авторизация {% endblock %}
3 {% block header %} <h2>Страница авторизации</h2> {% endblock header%}
4 {% block content %}
5     <form method="post" action="{% url 'login' %}">
6         {% csrf_token %}
7         <div class = "form-row p-4">
8             <div class = "col mt-2">
9                 {{ form.username.label_tag }}
10                <input type = "text"
11                    class = "form-control"
12                    placeholder = 'Логин'
13                    {{form.username}}
14            </div>
15            <div class = "col mt-2">
16                {{ form.password.label_tag }}
17                <input type = "password"
18                    class = "form-control"
19                    placeholder = 'Пароль'
20                    {{form.password}}
21            </div>
22            <div class = "col-2 mt-2">
23                <button type="submit" class="btn btn-secondary">Войти</button>
24            </div>
25            <input type="hidden" name="next" value="{{ next }}" />
26        </div>
27    </form >
```

Рисунок 36 – Код файла Login

После проделанной работы командой «pythonmanage.py makemigrations» объявляется встроенная база данных пользователей, затем командой «pythonmanage.py migrate» подтверждаются внесенные изменения в модель приложения.

Для проверки работоспособности входа на сайт, командой создана «pythonmanage.py createsuperuser» учётная запись главного администратора. При вводе команды «pythonmanage.py createsuperuser» в систему введен пароль, логин главного администратора (рисунок 37). В дальнейшем данная учётная запись используется для создания новых пользователей системы.

```
(VirtEnvProject) PS D:\desktop\VirtEnvProject\medinentarization> python manage.py createsuperuser
Имя пользователя: admin
Error: That имя пользователя is already taken.
Имя пользователя: adminisrator
Адрес электронной почты:
Password:
```

Рисунок 37 – Создание главного администратора

Повторно запустив сервер командой pythonmanage.py runserver было отображено окно входа в систему (рисунок 38), где пользователь может ввести данные. Написав данные администратора, указанные при его создании, система отобразила шаблон выбора медицинских организаций (рисунок 39).



Страница авторизации

Имя пользователя:

Пароль:

Войти

Рисунок 38 – Окно входа



Главная страница

Медицинская организация

Вам необходимо выбрать муниципальное учреждение для получения информации о нём.

№1: ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»

№2: ГБУЗ РХ «Абазинская городская больница»

№3: ГБУЗ РХ «Белоярская районная больница»

Рисунок 39 – Выбор медицинских организаций

2.4 Создание шаблонов html и маршрутов для отображения информации из базы данных

Шаблонhtml-страницы для выбора медицинской организации создаётся на основе базового шаблона. Разница заключается в том, что в шаблон index в функции, привязанной к шаблону, происходит передача данных Data из

таблицы в базе данных. Получение данных в шаблон при помощи таблицы данных Data происходит в строке кода 13 на рисунке 40. `{% forel in data %}` это цикл, который выводит информацию из таблицы данных. `{{el.municipal_id}}` – передаёт id из поля `municipal_id`. Данное поле таблицы данных Data унаследовала данные от такого же поля в таблице медицинских организаций в базе данных. Передача таблицы данных в шаблон отображения html происходит в функции в файле `views.py`. Функция `index`, изображенная на рисунке 41, защищена от отображения неавторизованному пользователю при помощи декоратора `@login_required`. Данные из базы данных передаются в Data при помощи обращения к базе данных. Для главного администратора системы data получает всю информацию «`data = municipal_institution.object.all()`», так как после обращения к объектам в таблице стоит получение всех данных `all`. Если с приложением работать будет администратор медицинской организации, то данные будут отфильтрованы для определенного пользователя по `id_user`.

Строчка кода: `«data = municipal_institution.object.filter(user_id=request.user.id).all()»`

Если пользователь не будет авторизован, то его вернет на страницу авторизации.

```

1  {% extends 'main/base.html' %}
2  {% block title %} Главная страница {% endblock %}
3  {% block header %} <h1>Главная страница</h1> {% endblock header %}
4  {% block content %}
5  <h3>Медицинская организация</h3>
6  <p>Вам необходимо выбрать муниципальное учреждение для получения информации о нём.</p>
7  <ul class="list-group">
8      {% for el in data %}
9          <li>
10             <td> <a href="{% url 'database' el.municipal_id %}"
11                 class="list-group-item list-group-item-action list-group-item-secondary">
12                 №{{el.municipal_id}}: {{ el.name }}</a>
13             </td>
14         </li>
15     {% endfor %}
16 </ul>
17 {% endblock content %}

```

Рисунок 40 – Шаблон `Index.html`

```

36 @login_required
37 def index(request, pk=None):
38     if request.user.is_authenticated == True:
39         if request.user.is_superuser == True:
40             data = municipal_institutions.objects.all()
41         else:
42             data = municipal_institutions.objects.filter(user_id=request.user.id).all()
43     return render(request, 'main/index.html', {'data': data})

```

Рисунок 41 – Функция index в файле views.py

Выбрав любую из организаций, система перенаправит пользователя в приложение database и будет выводить шаблоны, созданные в данном приложении. Для отображения данных из таблиц необходимо для каждой таблицы, созданной в базе данных, создать свой шаблон отображения.

Созданные файлы шаблонов изображены на рисунке 42.

Создав шаблоны, необходимо обозначить маршруты, по которым эти шаблоны будут отображаться. Аналогичная операция совершена при создании шаблона авторизации и шаблона отображения медицинских организаций. Во-первых, в файле urls.py (рисунок 43) созданы пути, затем к каждому маршруту привязана функция из файла views, далее в файле views (рисунок 44) создана функция, которая вызывается при прохождении определенного пути.

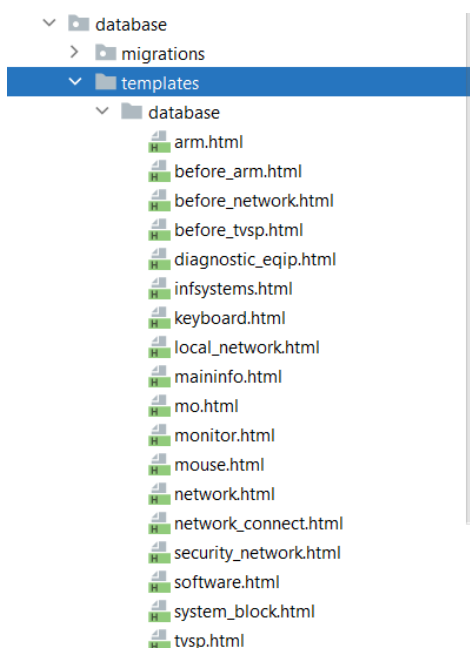


Рисунок 42 – Шаблоны для отображения информации из таблиц


```

6 urlpatterns = [
7     path('software/<int:armid>/', views.softwares, name = 'software'),
8     path('keyboard/<int:armid>/', views.keyboards, name = 'keyboard'),
9     path('mouse/<int:armid>/', views.mouses, name = 'mouse'),
10    path('monitor/<int:armid>/', views.monitors, name = 'monitor'),
11    path('system_block/<int:armid>/', views.system_block, name = 'system_block'),
12    path('arm/<int:networkid>/', views.arms, name = 'arm'),
13    path('before_arm/<int:networkid>/', views.before_arms, name = 'before_arm'),
14    path('local_network/<int:networkid>/', views.local_networks, name = 'local_network'),
15    path('security_network/<int:networkid>/', views.security_networks, name = 'security_network'),
16    path('network_connect/<int:networkid>/', views.network_connects, name = 'network_connect'),
17    path('network/<int:tvspid>/', views.networks, name = 'network'),
18    path('before_network/<int:tvspid>/', views.before_networks, name = 'before_network'),
19    path('diagnostic equip/<int:tvspid>/', views.diagnostic_eqips, name = 'diagnostic equip'),
20    path('tvsp/<int:mo_id>/', views.tvsp, name = 'tvsp'),
21    path('infsystems/<int:mo_id>/', views.infsystems, name = 'infsystems'),
22    path('maininfo/<int:mo_id>/', views.maininfo, name = 'maininfo'),
23    path('before_tvsp/<int:mo_id>/', views.before_tvsp, name = 'before_tvsp'),
24    path('<int:mo_id>/', views.database, name = 'database'),
25    path('', views.database, name = 'database'),
26 ]

```

Рисунок 43– Файл urls в приложении database

```

4 from .models import *
5 from django.contrib.auth.decorators import login_required
6
7 @login_required
8 def softwares(request, armid):
9     data = Software.objects.filter(Arm=armid).all()
10    return render(request, 'database/software.html', {'data':data})
11
12 @login_required
13 def keyboards(request, armid):
14     data = Keyboard.objects.filter(Arm=armid).all()
15    return render(request, 'database/keyboard.html', {'data':data})
16
17 @login_required
18 def mouses(request, armid):
19     data = Mouse.objects.filter(Arm=armid).all()
20    return render(request, 'database/mouse.html', {'data':data})
21
22 @login_required
23 def monitors(request, armid):
24     data = Monitor.objects.filter(Arm=armid).all()
25    return render(request, 'database/monitor.html', {'data':data})

```

Рисунок 44 – Файл views в database

Для каждой таблицы из базы данных созданы шаблоны html для отображения информации. Для каждого шаблона html созданы маршруты, для каждого маршрута созданы свои функции. Всего создано 19 шаблонов (таблица 4), 19 маршрутов (рисунок 43) и 19 функций для передачи данных в шаблон,

защищённый от неавторизированных пользователей, для передачи в шаблоны html данных из базы данных.

Таблица 4 –Перечень шаблонов и маршрутов для их отображения

№	Функция	Шаблон	Маршрут
1	Авторизация пользователя	Login.html	accounts/login
2	Вывод всех данных о медицинских организациях	maininfo.html	Database/maininfo
3	Вывод данных о выбранной медицинской организации	Mo.html	Database/database
4	Вывод данных о информационной системе медицинской организации	InfSystems.html	Database/infosystems

Продолжение таблицы 4

5	Вывод всех данных о территориально-выделенном структурном подразделении (ТВСП) медицинской организации	Before_tvsp.html	Database/before_tvsp
6	Вывод данных о выбранном ТВСП	tvsp.html	Database/tvsp
7	Вывод данных о диагностическом оборудовании выбранного ТВСП	Diagnostic equip.html	Database/diagnostic equip
8	Вывод данных о всех сетевых инфраструктурах выбранного ТВСП	Before_network.html	Database/before_network
9	Вывод данных о выбранной сетевой инфраструктуре	Network.html	Database/Network
10	Вывод данных о локальной сети ТВСП	Local_network.html	Database/Local_network
11	Вывод данных о способе подключения к сети интернет ТВСП	Network_connect.html	Database/Network_connect
12	Вывод данных о способе защиты сети ТВСП	Security_network.html	Database/Security_network
13	Вывод всех данных о каждом автоматизированном рабочем месте (АРМ) ТВСП	Before_arm.html	Database/before_arm
14	Вывод данных о выбранном АРМ	arm.html	Database/ arm
15	Вывод данных о мониторе АРМ	keyboard.html	Database/ keyboard
16	Вывод данных о клавиатуре АРМ	monitor.html	Database/ monitor
17	Вывод данных о компьютерной мыши АРМ	mouse.html	Database/ mouse

18	Вывод данных о системном блоке АРМ	System_block.html	Database/ System_block
19	Вывод данных о ПО АРМ	software.html	Database/ software

Шаблон вывода информации о медицинской организации изображен на рисунке 45. Данный шаблон аналогичен шаблону вывода списка медицинских организаций. Отличие состоит в том, что в шаблон поступила вся информация из базы данных о выбранной медицинской организации. В блок `{% blockside_bar %}` поставлены кнопки “ТВСП”, “ИНФОРМАЦИОННЫЕ СИСТЕМЫ”(рисунок 46), которые осуществляют переход на другой маршрут, где будет получен шаблон для вывода информации, получаемой по определённому маршруту.

```

1  {% extends 'main/base.html' %}
2  {% block SIDE_BAR %}
3  <ul class="list-group">
4  <li><input class="list-group-item list-group-item-action" type="button
5  value="Назад" onClick="javascript:history.go(-1);"></input></li>
6  <li><a href="{% url 'infsystems' mo_id %}"
7  class="list-group-item list-group-item-action">Информационные системы</a></li>
8  <li><a href="{% url 'before_tvsp' mo_id %}" class="list-group-item list-group-item-action">ТВСП</a></li>
9  </ul>
10  {% endblock SIDE_BAR %}
11  {% block title %} Медицинская организация {% endblock %}
12  {% block content %}
13  <h3>Медицинская организация</h3>
14  {% for el in data %}
15  <ul class = "list-group">
16  <li class = "list-group-item active"> <p>№ {{ el.municipal_id }}: {{ el.name }} <a></a></p></li>
17  <li class = "list-group-item ">
18  <p class="text_card"> ОИД учреждения: </p> {{ el.oid_municipal }}
19  <br> <p class="text_card">Адрес учреждения: </p> {{ el.address }}
20  <br> <p class="text_card">Уровень учреждения: </p> {{ el.level_municipal }}
21  </li>
22  </ul>
23  {% endfor %}
24  {% endblock %}

```

Рисунок 45 – Информация о медицинской организации



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

Назад	<h3>Медицинская организация</h3> <p>№ 1: ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»</p> <p>OID учреждения: Заполните Адрес учреждения: г. Абакан, ул.Тараса Шевченко, 83 Уровень учреждения: Заполните</p>
Информационные системы	
ТВСП	

Рисунок 46 – Страница с информацией о медицинской организации

Выбрав кнопку ТВСП, совершается переход на шаблон выбора информации о территориальных-выделенных структурных подразделениях выбранной медицинской организации (рисунок 47).



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

Назад	<h3>Выбор территориально-выделенного структурного подразделения (ТВСП)</h3> <p>№1: Администрация</p> <p>№3: Подстанция скорой медицинской помощи</p> <p>№4: Аптечный склад</p>
-------	--

Рисунок 47 – Страница выбора ТВСП

Выбрав структурное подразделение, осуществляется переход на страницу вывода информации о ТВСП (рисунок48).

Цифровой паспорт медицинских организаций [Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

Назад

Диагностическое оборудование

Сетевая инфраструктура ТВСП

Территориально-выделенное структурное подразделение (ТВСП)

№1: Администрация

Тип ТВСП: x
Адрес ТВСП: г. Абакан, ул.Тараса Шевченко, 83
ОIД ТВСП: x
Координаты ТВСП: 53.7251304, 91.4548107239726
Количество этажей ТВСП: x
Форма собственности: x
Состояние здания: исправное
Планируется реорганизация: нет
Амбулаторная помощь: False
Оказание ПМСП: False
Наличие регистратуры: True
Стационарная помощь: False
КДЛ: False
СМП: False
НМП: False
Инструментальная диагностика: False
ЛЛО: False
Направление на МСЭ: False
Оформление листов временной нетрудоспособности: False
ССЗ: False
Онкология: False

Рисунок 48 – Информация о выбранном ТВСП

Остальные страницы вывода информации созданы аналогично. В каждом шаблоне осуществлена передача кнопок страниц «Наименование страницы вывода информации» в главный шаблон. Осуществлено позиционирование данных при отображении в блоке {`%blockcontent%`}.

На рисунках46-61 показаны созданные 13 страниц вывода информации в соответствии с требованиями к системе (см. пункт 1.2).



[Назад](#)

Диагностическое оборудование

№1: Диагностическое оборудование1

Тип оборудовая: Диагностическое оборудование1
Год выпуска: 8 июня 2022 г.
Наличие подключения к ЛВС:
Место размещение оборудования: Диагностическое оборудование1
Наличие подключения к ЛВС: False
Количество исследований в год: Диагностическое оборудование1
Средний объем одного исследования (МБ): Диагностическое оборудование1
Признак активации DICOM 3.0: False
Наличие АРМ диагноста: False

Рисунок 49 – Страница диагностического оборудования



[Назад](#)

Выбор сетевой инфраструктуры ТВСП

№1: Администрация

№2: Администрация

Рисунок 50 – Страница выбора сетевой инфраструктуры ТВСП



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Сетевая инфраструктура

ID сетевых параметров ТВСП	id ТВСП
1	Администрация

Информация о доступе в интернет

Защищенная сеть передачи данных

Локальная вычислительная сеть

Оборудование (АРМ)

Рисунок 51 – Страница сетевой инфраструктуры ТВСП



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Доступ в интернет

№1: Заполните

Статус подключения к интернет: Заполните
Технология подключения к интернет: Заполните
Пропускная способность канала: Заполните
Поставщик услуг: Заполните
Стоимость услуг в месяц: Заполните
Участие в ФП: 22
Сетевая инфраструктура: Network object (1)



© 2022 ГБУЗ "РМИАЦ" Хакасии

Рисунок 52 – Страница с информацией о доступе в интернет



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Защищенная сеть передачи данных

№1: Отсутствует

CID защищенной сети передачи данных: 1
Статус подключения к ЗСПД: True
Технология подключения к ЗСПД: Отсутствует
ViPNet Coordinator: False
ViPNet HW1000: False
IPNet Client: False
Сетевая инфраструктура: Network object (1)



© 2022 ГБУЗ "РМИАЦ" Хакасии

Рисунок 53 – Страница с информацией о защищенной сети



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Локальная сеть ТВСП (ТВСП)

№1: Заполните

Количество портов ЛВС: 5
IP-адресация: Заполните
Площадь помещения: Заполните
Место размещения: Заполните
Соответствие требованиям к серверным помещениям: Заполните
Наличие схемы ЛВС: True
Сетевая инфраструктура: Network object (1)

Рисунок 54 – Страница с информацией о локальной сети



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Выбор автоматизированного рабочего места

№1 - Кабинет: 321

№2 - Кабинет: 333

Рисунок 55 – Страница выбора АРМ



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Автоматизированное рабочее место

Системный блок (АРМ)

Номер рабочего места: 1

Монитор(АРМ)

Кабинет АРМ: 321

Мышь(АРМ)

Сетевая инфраструктура: Network object (1)

Клавиатура(АРМ)

Программное обеспечение

Рисунок 56 – Страница информации о АРМ



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Системный блок

№: 312

Дата покупки: 7 июня 2022 г.

Модель процессора: Заполните

Модель ОЗУ: Заполните

Объем ОЗУ: Заполните

Тип системы хранения:

Заполните

Объем хранения: Заполните

Операционная система:

Заполните

АРМ: Arm object (1)

Рисунок 57 – Страница информации о системном блоке



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Монитор

LG

Инвентарный номер: Заполните

Диагональ: Заполните

Год покупки: 6 июня 2022 г.

АРМ: Arm object (1)

Рисунок 58 – Страница информации о мониторе



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Компьютерная мышь

Razen

Инвентарный номер: 444

Год покупки: 1 июня 2022 г.

АРМ: Arm object (1)

Рисунок 59 – Страница информации о компьютерной мыши



Цифровой паспорт медицинских организаций

[Главная](#) [Профиль | admin](#) [Отчёты](#) [Выход](#)

[Назад](#)

Клавиатура

Oklick

Инвентарный номер: 341

Год покупки: 2 июня 2022 г.

АРМ: Arm object (1)

Рисунок 60 – Страница информации о клавиатуре



Цифровой паспорт медицинских организаций

Главная | Профиль | admin | Отчёты | Выход

[Назад](#)

Программное обеспечение

Windows
Разработчик: Заполните
Наличие в реестре отечественного ПО: False
Версия: 10
Цена: Заполните
Год покупки: 4 июня 2022 г.
АРМ: Arm object (1)

Рисунок 61 – Страница информации о ПО

Закончив настройку отображения данных в системе, необходимо настроить разграничение прав доступа к данным, а также необходимо настроить окно администрирования приложения.

2.5 Настройка административной панели, разграничение прав доступа к информации из базы данных

Изначально при переходе к администрированию сайта, после нажатия на кнопку «профиль» (рисунок 64), ни главному администратору, ни администратору медицинской организации не доступны никакие действия с таблицами, так как их описание не объявлено в файле admin.py. Чтобы главный администратор увидел данные, достаточно написать в файле admin.py, в приложении database, строку кода «admin.site.register (“Название таблицы с данными”))». Для каждой таблицы должна быть своя запись, название таблицы не должно меняться. Поскольку с информационной системой должны работать

разные пользователи, то в этом же файле `admin.py` прописаны условия отображения данных для каждой таблицы (рисунок 63).

[Главная](#) [Профиль](#) | [admin](#) [Отчёты](#) [Выход](#)

Рисунок 62 – Кнопки для перехода по разделам сайта

```
1 # -*- coding: utf-8 -*-
2 from django.contrib import admin
3 from .models import *
4
5 class municipal_institutions_A(admin.ModelAdmin):...
54 admin.site.register(municipal_institutions,municipal_institutions_A)
55
56 class InfSystems_A(admin.ModelAdmin):...
77 admin.site.register(InfSystems,InfSystems_A)
78
79 class tvsp_A(admin.ModelAdmin):...
103 admin.site.register(tvsp, tvsp_A)
104
105 class DiagnosticEquip_A(admin.ModelAdmin):...
133 admin.site.register(DiagnosticEquip, DiagnosticEquip_A)
134
135 class network_connect_B(admin.StackedInline):...
138
139 class security_network_B(admin.StackedInline):...
142
143 class local_network_B(admin.StackedInline):...
146
147 class Network_A(admin.ModelAdmin):...
171 admin.site.register(Network, Network_A)
172
173 class SystemBlock_b(admin.StackedInline):...
176
177 class Monitor_b(admin.StackedInline):...
180
181 class Keyboard_b(admin.StackedInline):...
184
185 class Mouse_b(admin.StackedInline):...
188
189 class Software_b(admin.StackedInline):...
192
193 class Arm_a(admin.ModelAdmin):...
217 admin.site.register(Arm, Arm_a)
```

Рисунок 63 – Файл `admin.py`

Для того, чтобы передать собственные настройки отображения в админ-панель, для каждой таблицы созданы классы данных с правилами отображения.

Для таблицы медицинских организаций (municipal_institutions) в классе municipal_institutions_A, созданы методы save_model, preprocess_list_display, preprocess_search_fields, queryset, fieldsets (рисунки 64-65).

Метод save_model отвечает за автозаполнение поля user (пользователь).

Метод preprocess_list_display добавляет колонку user в список объектов, если его просматривает главный администратор.

```
class municipal_institutions_A(admin.ModelAdmin):
    user_fieldsets = (
        (None, {
            'classes': ('wide',),
            'fields': ('name', 'address', 'oid_municipal', 'level_municipal')
        }),
    )
    list_display = ['name', 'user', 'municipal_id']
    raw_id_list_displayfields = ('user',)
    search_fields = ['name', 'user__username']

    def save_model(self, request, obj, form, change):
        if form.is_valid():
            if not request.user.is_superuser or not form.cleaned_data["user"]:
                obj.user = request.user
                obj.save()
            elif form.cleaned_data["user"]:
                obj.user = form.cleaned_data["user"]
                obj.save()

    def preprocess_list_display(self, request):
        if 'user' not in self.list_display:
            self.list_display.insert(self.list_display.__len__(), 'user')
        if not request.user.is_superuser:
            if 'user' in self.list_display:
                self.list_display.remove('user')
```

Рисунок 64 – Объявление класса с правилами для таблицы медицинских организаций

```

1 def preprocess_search_fields(self, request):
2     if 'user__username' not in self.search_fields:
3         self.search_fields.insert(self.search_fields.__len__(), 'user__username')
4     if not request.user.is_superuser:
5         if 'user__username' in self.search_fields:
6             self.search_fields.remove('user__username')
7
8     def changelist_view(self, request, extra_context=None):
9         self.preprocess_list_display(request)
10        self.preprocess_search_fields(request)
11        return super(municipal_institutions_A, self).changelist_view(request)
12
13    def get_queryset(self, request):
14        qs = super(municipal_institutions_A, self).get_queryset(request)
15        if request.user.is_superuser:
16            return qs
17        return qs.filter(user_id=request.user.id)
18
19    def get_fieldsets(self, request, obj=None):
20        if request.user.is_superuser:
21            return super(municipal_institutions_A, self).get_fieldsets(request, obj)
22        return self.user_fieldsets
23
24 admin.site.register(municipal_institutions, municipal_institutions_A)

```

Рисунок 65 – Регистрация правил в панели администратора для таблицы медицинских организаций

Метод `preprocess_search_fields` даёт возможность главному администратору не только видеть записи администратора медицинской организации, но и осуществлять поиск по его `username` (логину).

Метод `get_queryset` возвращает все записи для главного администратора и персональные записи администраторов медицинских организаций.

Изначально Django не знает, как выводить информацию и кому её отображать, поэтому в методе `get_queryset` происходит настройка фильтрации.

Вначале в методе происходит запрос всех данных из таблицы с данными, и, если пользователь является главным администратором (if `request.user.is_superuser:`), ему отобразится вся информация. Иначе произойдет поиск данных, закрепленных за определенным администратором (`qs.filter(user_id=request.user.id)`). Данный способ фильтрации работает только с

таблицей медицинских организаций, так как в данную таблицу уже занесены данные о пользователе, и, если данные авторизованного пользователя совпадут с данными в таблице, то произойдет вывод связанной с ним информацией.

Чтобы осуществить выбор информации на нижнем уровне базы данных, например в таблице диагностического оборудования, необходимо поменять условия отбора информации. Сначала нужно узнать, к какой медицинской организации прикреплен пользователь, а затем получить строку с данными этой организации. Далее идёт сравнение внешнего ключа таблицы диагностического оборудования с главным ключом таблицы муниципального образования. Если ключи одинаковые, то произойдёт вывод информации. Условия отбора информации для таблицы диагностического оборудования изображены на рисунке 66.

```
class DiagnosticEquip_A(admin.ModelAdmin):
    list_display = ['name', 'diagnostic_id', 'tvsp']
    raw_id_list_displayfields = ('tvsp',)
    search_fields = ['name', 'tvsp']
    user_fieldsets = (
        (None, {
            'fields': ('name', 'type', 'release_year', 'location', 'lvs_connect_capacity', 'research_year_number',
                       'average_size_research', 'sign_activation_discom', 'existence_armd_discom')
        }),
    )

    def get_queryset(self, request):
        qs = super(DiagnosticEquip_A, self).get_queryset(request)
        if request.user.is_superuser:
            return qs
        else:
            return qs.filter(tvsp__in=tvsp.objects.filter(municipal_institutions__in=
                                                         municipal_institutions.objects.
                                                         filter(user=request.user)))

    def get_fieldsets(self, request, obj=None):
        if request.user.is_superuser:
            return super(DiagnosticEquip_A, self).get_fieldsets(request, obj)
        return self.user_fieldsets

admin.site.register(DiagnosticEquip, DiagnosticEquip_A)
```

Рисунок 66 – Условия отбора информации для таблицы диагностического оборудования

После создания аналогичных условий выбора информации у всех таблиц данных, классы, содержащие в себе правила, отображены определенной таблицей с данными (рисунок 67).

```
admin.site.register(municipal_institutions, municipal_institutions_A)
```

Рисунок 67 – Регистрация правил и таблицы данных в панели администратора

По итогу объявлены в панели администратора таблицы, находящиеся в базе данных, а также к данным таблицам применены условия отбора.

Панель администратора с объявленными таблицами изображена на рисунке 68.

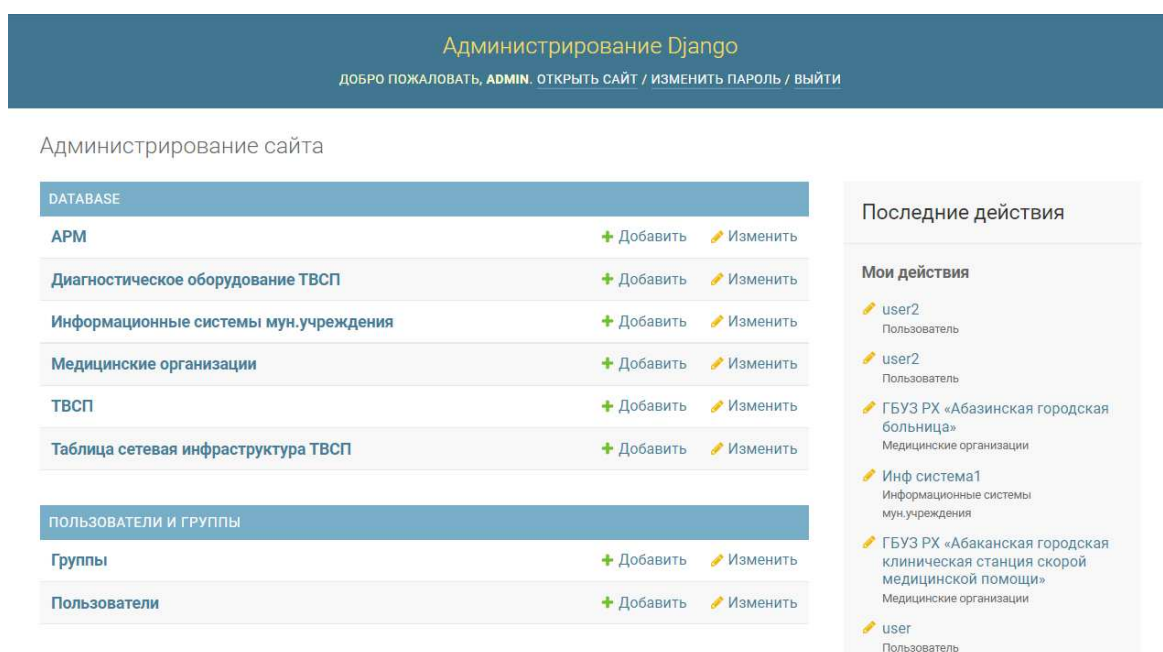


Рисунок 68 – Панель администратора с объявленными таблицами данных

Панель администратора Django позволяет редактировать, создавать записи в базе данных, не прибегая к сторонним программам.

Например, нужно в медицинской организации назначить администратора. Для осуществления этого действия главный администратор

выбирает определенную таблицу данных и нажимает кнопку «Изменить»(рисунок 69).

DATABASE		
АРМ	+ Добавить	✎ Изменить
Диагностическое оборудование ТВСП	+ Добавить	✎ Изменить
Информационные системы мун.учреждения	+ Добавить	✎ Изменить
Медицинские организации	+ Добавить	✎ Изменить
ТВСП	+ Добавить	✎ Изменить
Таблица сетевая инфраструктура ТВСП	+ Добавить	✎ Изменить

Рисунок 69 – Доступные таблицы для редактирования

Главный администратор попадает на страницу выбора медицинской организации (рисунок 70).

Выберите Муниципальные учреждения ДОБАВИТЬ МУНИЦИПАЛЬНЫЕ УЧРЕЖДЕНИЯ +

🔍 Найти

Действие: Выполнить

Выбрано 0 объектов из 3

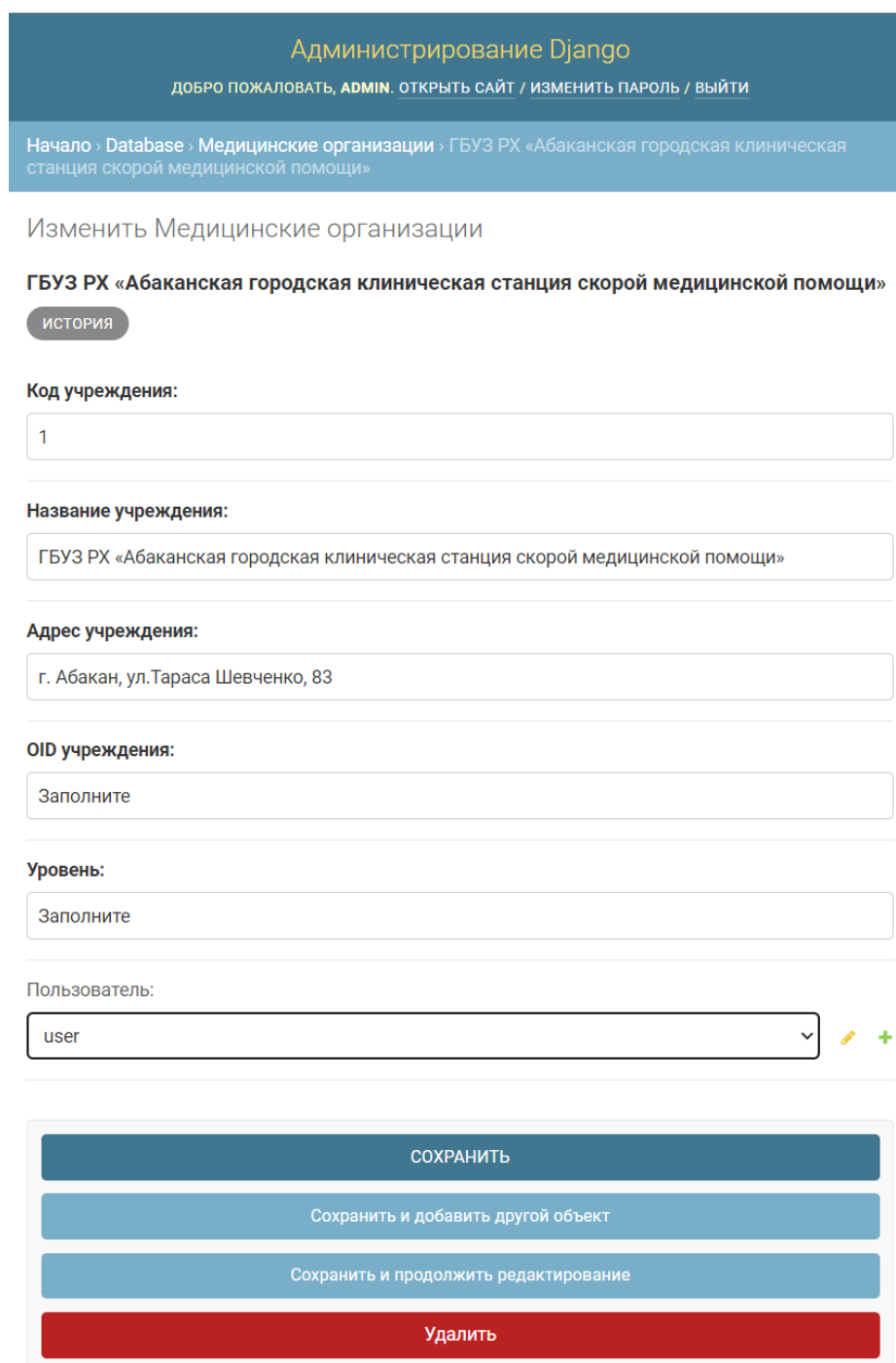
<input type="checkbox"/>	НАЗВАНИЕ УЧРЕЖДЕНИЯ	ПОЛЬЗОВАТЕЛЬ	КОД УЧРЕЖДЕНИЯ
<input type="checkbox"/>	ГБУЗ РХ «Белоярская районная больница»	-	3
<input type="checkbox"/>	ГБУЗ РХ «Абазинская городская больница»	user2	2
<input type="checkbox"/>	ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»	user	1

3 Муниципальные учреждения

Рисунок 70 – Страница выбора медицинской организации

После выбора медицинской организации открывается форма с полями, которые можно редактировать. Пример редактирования приведён на рисунке

71.У медицинской организации с кодом учреждения 1 назначен администратор - пользователь User.



Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало > Database > Медицинские организации > ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»

Изменить Медицинские организации

ГБУЗ РХ «Абаканская городская клиническая станция скорой медицинской помощи»

ИСТОРИЯ

Код учреждения:

Название учреждения:

Адрес учреждения:

OID учреждения:

Уровень:

Пользователь:

СОХРАНИТЬ

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

Удалить

Рисунок 71 – Страница изменения

После назначения пользователя-администратора медицинской организации, главный администратор назначает ему права пользователя для работы с БД и устанавливает «Статус персонала» (рисунки 72-73). «Статус

персонала» разрешит пользователю посещать административную часть приложения с данными его медицинской организации.

Изменить пользователь

ИСТОРИЯ

user

Имя пользователя:

user

Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @/./+/_/..

Пароль:

алгоритм: pbkdf2_sha256 итерации: 320000 соль: 8zDN1z*****
хэш: ХеZrHc*****

Пароли хранятся в зашифрованном виде, поэтому нет возможности посмотреть пароль этого пользователя, но вы можете изменить его используя эту форму.

Персональная информация

Имя:

Фамилия:

Адрес электронной почты:

Рисунок 72 – Страница изменения персональной информации

Права пользователя:

Доступные права пользователя

Фильтр

admin | запись в журнале | Can add log entry
admin | запись в журнале | Can change log entry
admin | запись в журнале | Can delete log entry
admin | запись в журнале | Can view log entry
auth | группа | Can add group
auth | группа | Can change group
auth | группа | Can delete group
auth | группа | Can view group
auth | право | Can add permission
auth | право | Can change permission
auth | право | Can delete permission
auth | право | Can view permission
auth | пользователь | Can add user

Выбрать все

Выбранные права пользователя

database | APM | Can add arm
database | APM | Can change arm
database | APM | Can delete arm
database | APM | Can view arm
database | Диагностическое оборудование ТВСП | Can add
database | Диагностическое оборудование ТВСП | Can cha
database | Диагностическое оборудование ТВСП | Can dele
database | Диагностическое оборудование ТВСП | Can view
database | Информационные системы мун.учреждения | C
database | Информационные системы мун.учреждения | C
database | Информационные системы мун.учреждения | C
database | Информационные системы мун.учреждения | C
database | Клавиатура | Can add keyboard
database | Клавиатура | Can change keyboard
database | Клавиатура | Can delete keyboard
database | Клавиатура | Can view keyboard

Удалить все

Рисунок 73 – Страница изменения прав пользователя

После окончания настройки прав пользователя проверена функция отбора информации. Авторизовавшись как user, осуществлен вход в

административную часть приложения. Пользователю недоступно редактирование информации о других пользователях, так как эти права не указаны в разрешениях. User получил права на изменения данных, связанных с его медицинской организацией (рисунок 74).

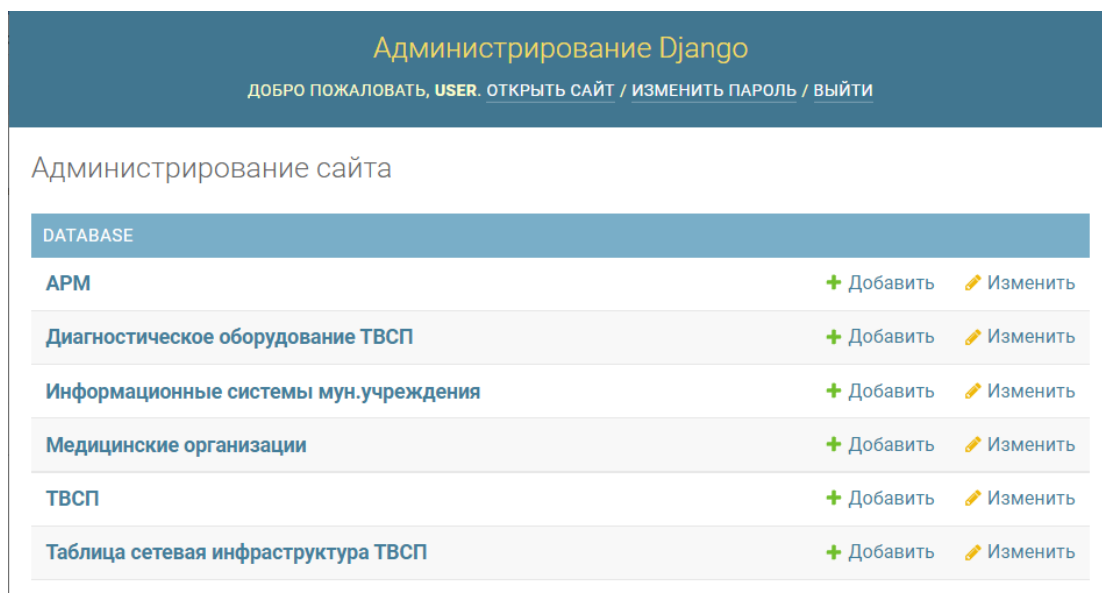


Рисунок 74 – Страница изменения данных

Совершив переход в таблицу медицинской организации, отображена информация только об организации, к которой прикреплен пользователь (рисунок 75).

Данные в таблице разрешается редактировать и просматривать, так как для этого объявлены соответствующие права (рисунок 76).

Совершив переход в таблицу ТВСП, отображена информация только об ТВСП, которые принадлежат медицинской организации пользователя (рисунок 77).

Данные в ТВСП разрешается удалять, добавлять, изменять и просматривать, так как для этого объявлены соответствующие права пользователя.

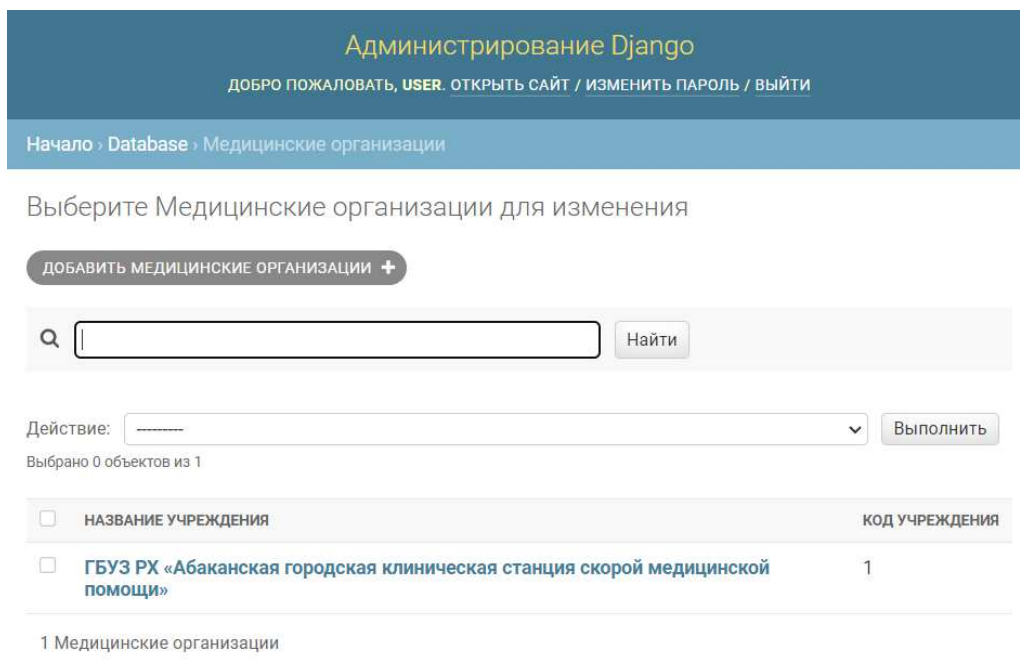


Рисунок 75 – Страница просмотра медицинских организаций

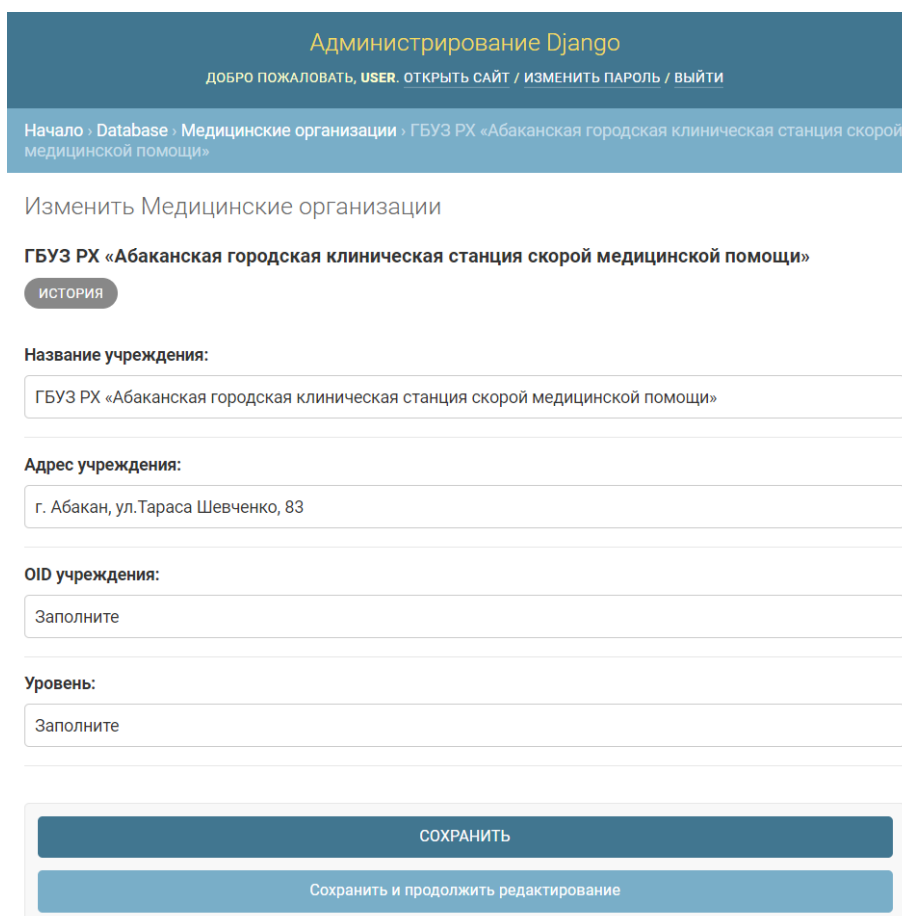


Рисунок 76 – Страница изменения данных в медицинской организации

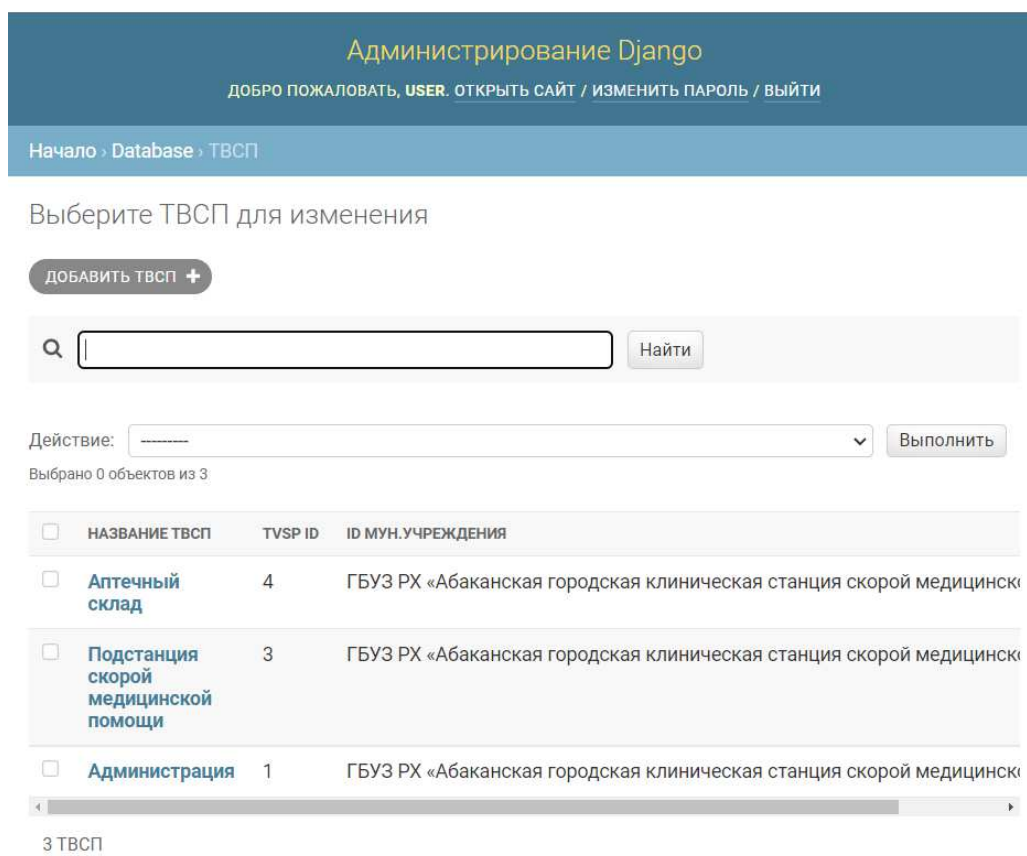


Рисунок 77 – Страница изменения данных в ТВСП

После настройки вывода информации и настройки прав пользователя в административной части веб-приложения, созданы аналогичные настройки вывода информации администратору медицинской организации в созданных html шаблонах приложения.

Для этого в файле views.py в приложении main_app добавляем условия фильтрации, такие же как объявлены в admin.py в приложении database.

Если пользователь обладает правами главного администратора, то он получает всю информацию. Иначе происходит отбор информации по id_user. Если за пользователем (администратором медицинской организации) закрепили медицинскую организацию, то он получит только свои данные, иначе окно отображения информации будет пустым (рисунок 78).

Так как user назначен в медицинской организации №1 в панели администратора, то и информацию в окне вывода он увидит только о медицинской организации №1 (рисунок 79).

```

@login_required
def index(request, pk = None):
    if request.user.is_authenticated == True:
        if request.user.is_superuser == True:
            data = municipal_institutions.objects.all()
        else:
            data = municipal_institutions.objects.filter(user_id=request.user.id).all()
    return render(request, 'main/index.html', {'data': data})

```

Рисунок 78 – Фильтр выводимой информации

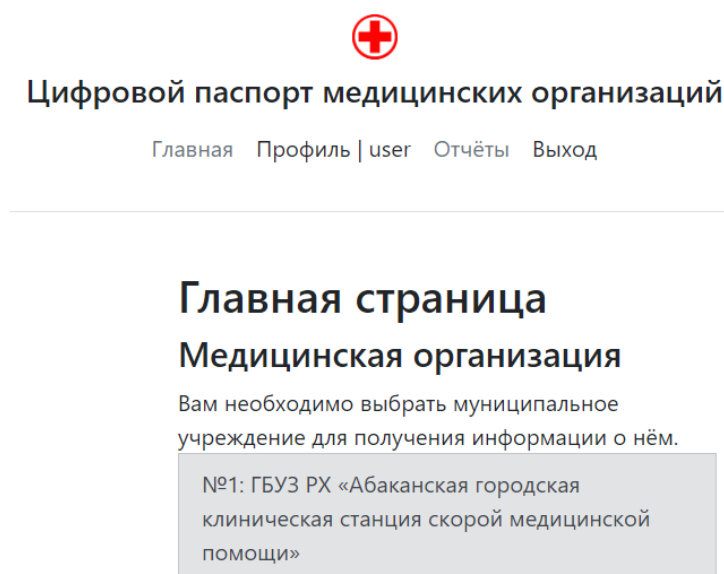


Рисунок 79 – Полученные данные

2.6 Создание отчётов для выгрузки информации из базы данных в excel файл

Разобравшись с выводом информации во всем приложении, приступаем к получению данных в excel файл. Поскольку система содержит несколько отчётов, то создаём шаблон для вывода отчётов на отдельную страницу (рисунок 80). После создаём отдельный путь (рисунок 81) и функцию отображения шаблона (рисунок 82) в приложении filter. На рисунке 83 отображена страница отчёта.

Чтобы попасть на данную страницу, достаточно нажать кнопку «Отчёты» на панели кнопок.

Процесс формирования excel-файла происходит в приложении excel_app в файле views.py. При формировании excel файла используется модуль xlwt. Модуль xlwt позволяет формировать образы таблиц excel в которые будут переданы данные из соответствующих таблиц базы данных.

```
urlpatterns = [
    path('', views.filter, name='filter'),
    path(r'filter/', views.filter, name='filter_1'), #127.0.0.1:8000/main_app/
    path(r'filter/<int:mo_id>/', views.filter, name='filter_2'), #127.0.0.1:8000/main_app/
    path(r'filter_infsystem/', views.filter_infsystems, name='filter_infsystem'), #127.0.0.1:8000/main_app/
```

Рисунок 80 – Маршруты приложения filter

```
{% extends 'main/base.html'%}
{% block title %} Отчёты {% endblock %}
{% block header %} <h1>Отчёты</h1> {% endblock header%}
{% block SIDE_BAR %}
<ul class="list-group">
<li><input class="list-group-item list-group-item-action" type="button" value="Назад" onClick="javascript:history.go(-1);"></input></li>
<li><a href="{% url 'report_info' %}" class="list-group-item list-group-item-action">Отчёт о заполненности данных в системе</a></li>
<li><a href="{% url 'export_xls' %}" class="list-group-item list-group-item-action">Отчёт в excel</a></li>
</ul>
{% endblock SIDE_BAR %}

{% block content %}
<p>Выберите мед. организацию</p>
<ul class="list-group">
<li><td> <a href="{% url 'filter_infsystem' %}" class="list-group-item list-group-item-action list-group-item-secondary">
Фильтр информационных систем</a></td></li>
</ul>
{% endblock content %}
```

Рисунок 81 – Шаблон страницы «Отчёты»

```
@login_required
def filter(request):
    if request.user.is_superuser == True:
        data = municipal_institutions.objects.all()

    else:
        data = municipal_institutions.objects.filter(user_id=request.user.id).all()
    return render(request, 'filter/filter.html', {'data': data})
```

Рисунок 82 – Функция вызова шаблона filter.html

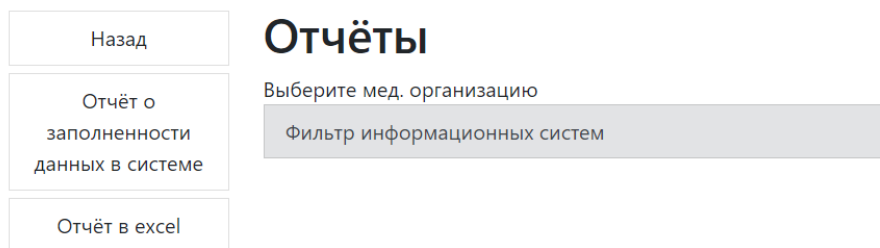


Рисунок 83 – Отображение страницы «Отчёты»

```
def export_mun_xls(request):  
    #Объявление веб-файла, который потом загрузится в excel  
    response = HttpResponse(content_type='application/ms-excel')  
    response['Content-Disposition'] = 'attachment; filename="med_password.xls"  
    wb = xlwt.Workbook(encoding='utf-8')  
  
    #-----Создание страницы Организаций-----  
    # Sheet  
    ws = wb.add_sheet('Медицинские организации')  
    # Sheet header, first row  
    row_num = 0  
    font_style = xlwt.XFStyle()  
    font_style.font.bold = True  
    columns = ['ID', 'Название', 'Адрес', 'OID', 'Уровень']  
    for col_num in range(len(columns)):  
        ws.write(row_num, col_num, columns[col_num], font_style)  
    # Sheet body, remaining rows  
    font_style = xlwt.XFStyle()  
  
    if request.user.is_superuser == True:  
        rows = municipal_institutions.objects.all().values_list('municipal_id',  
                                                                'name',  
                                                                'address',  
                                                                'oid_municipal',  
                                                                'level_municipal')  
    else:  
        rows = municipal_institutions.objects.filter(user_id=request.user.id).values_list('municipal_id',  
                                                                'name', 'address',  
                                                                'oid_municipal',  
                                                                'level_municipal')  
  
    for row in rows:  
        row_num += 1  
        for col_num in range(len(row)):  
            ws.write(row_num, col_num, row[col_num], font_style)  
    #wb.save(response)  
    # return response
```

Рисунок 84 – Создание excel файла и страницы медицинских организаций

На рисунке 84 изображен процесс создания excel файла, который потом будет отправлен пользователю. Первым делом создаётся сам файл excel. Затем при помощи библиотеки xlwt создаётся область для заполнения данными. Далее

переменной `ws` объявляется создание новой страницы в excel файле. После объявляются колонки, содержащие в себе названия выводимой информации, и при помощи цикла `for` происходит запись названий в файл excel.

Затем объявлен фильтр данных, если пользователь является главным администратором, то он получает все данные из таблицы, иначе происходит выбор данных, которые принадлежат определенному пользователю.

Далее при помощи цикла `for` происходит запись информации в ячейки.

Фильтрация данных при формировании excel файла такая же, как для отображения данных в административной части приложения, поэтому произошло заимствование условий вывода информации.

Чтобы получить excel файл со всеми таблицами из базы данных, в excel файле для каждой таблицы вручную создаётся страница, куда данные нужно передавать. На рисунке 85 изображен процесс создания и заполнения страницы “информационные системы”. Остальные 12 таблиц создаются аналогичным способом. На рисунках 85 и 86 это отображено.

```
#New Sheet Монитор-----
monit = wb.add_sheet('Мониторы')
# Sheet header, first row
row_num = 0
font_style = xlwt.XFStyle()
font_style.font.bold = True
columns = [ 'ID Монитора',
            'Модель',
            'Диагональ',
            'Инвентарный номер',
            'Год покупки',
            'Номер рабочего места (id_arm)']
for col_num in range(len(columns)):
    monit.write(row_num, col_num, columns[col_num], font_style)
# Sheet body, remaining rows
font_style = xlwt.XFStyle()
if request.user.is_superuser == True:
    rows = Monitor.objects.all().values_list('monitor_id',
                                             'model',
                                             'diagonal',
                                             'inventory_number',
                                             'purchase_year',
                                             'Arm_id')
else:
    network_tvsp = tvsp.objects.filter(
        municipal_institutions__in=municipal_institutions.objects.filter(user=request.user))
    rows = Monitor.objects.filter(
        Arm__in=Arm.objects.filter(Network__in=Network.objects.filter(tvsp__in=network_tvsp))).values_list('monitor_id',
                                                           'model',
                                                           'diagonal',
                                                           'inventory_number',
                                                           'purchase_year',
                                                           'Arm_id')
for row in rows:
    row_num += 1
    for col_num in range(len(row)):
        monit.write(row_num, col_num, row[col_num], font_style)
```

Рисунок 85 – Создание страницы с информацией о мониторах в АРМ в excel

```
#New Sheet Инф. системы-----
inf_sys = wb.add_sheet('Информационные системы')
# Sheet header, first row
row_num = 0
font_style = xlwt.XFStyle()
font_style.font.bold = True
columns = ['ID',
           'Название',
           'Тип',
           'Разработчик',
           'Наличие модели угроз',
           'Наличие аттестата соответствия ФТСЭК',
           'ГИС?',
           'Объект КИИ?',
           'Категория КИИ',
           'ID мед.образования']
for col_num in range(len(columns)):
    inf_sys.write(row_num, col_num, columns[col_num], font_style)
# Sheet body, remaining rows
font_style = xlwt.XFStyle()
if request.user.is_superuser == True:
    rows = InfSystems.objects.all().values_list('infsys_id', 'name', 'type', 'developer', 'treat_model_presence',
                                               'conformity_certificate', 'gis', 'object_cii', 'cii_category',
                                               'municipal_institutions_id')
else:
    id_mun = municipal_institutions.objects.filter(user_id=request.user.id).first()
    rows = InfSystems.objects.filter(municipal_institutions_id=id_mun.municipal_id).values_list(
        'infsys_id', 'name', 'type', 'developer', 'treat_model_presence',
        'conformity_certificate', 'gis', 'object_cii', 'cii_category',
        'municipal_institutions_id'
    )

for row in rows:
    row_num += 1
    for col_num in range(len(row)):
        inf_sys.write(row_num, col_num, row[col_num], font_style)
```

Рисунок 86 – Создание страницы информационных систем в excel

При нажатии на кнопку “отчёт в excel” (рисунок 83) происходит загрузка файла на устройство пользователя.

2.7 Выводы по разделу «Разработка информационной системы»

В данном разделе описаны все этапы создания информационной системы и установки необходимого программного обеспечения. Были описаны действия при разработке информационной системы и продемонстрирован их результат.

3 Расчёт экономической эффективности проекта «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»

3.1 Экономическая эффективность реализации проекта

Под эффективностью автоматизированного преобразования экономической информации понимают целесообразность применения средств вычислительной и организационной техники при форматировании, передаче и обработке данных. Обобщенным критерием экономической эффективности является минимум затрат живого и общественного труда. При этом установлено, что чем больше участников управленческих работ автоматизировано, тем эффективнее используется техническое и программное обеспечение. Для этого необходимо рассмотреть базовый вариант (до внедрения проекта) и проектный вариант (после внедрения проекта).

Значимость технических решений (ЗТР) вычисляется по следующей формуле

$$\text{ЗТР} = K_a * K_{\Pi} * K_c + K_M * K_o * K_{\text{ш}}, \quad (1)$$

где K_a – коэффициент актуальности;

K_{Π} – коэффициент соответствия программ 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 5 приведены коэффициенты и ЗТР базового и разрабатываемого вариантов проекта.

Таблица 5 – Коэффициенты и ЗТР базового и разрабатываемого варианта

Коэффициенты	Базовый вариант	Разрабатываемый вариант
K_a	1	1,5
K_{Π}	1	1,
K_c	1	1,2
K_M	1	1
K_o	1	1,5
$K_{\text{ш}}$	1	2
ЗТР	2	4,8

Актуальность заключается в том, что сотрудник ГБУЗ РХ «РМИАЦ» сможет обрабатывать больше данных за меньшее количество времени.

Сложность заключается в том, что согласно требованиям Заказчика ИС должна находиться на сервере ГБУЗ РХ «РМИАЦ» и быть клиент-серверной, ИС не должна иметь зависимости с системами из сети интернет.

Из данной таблицы видно, что разрабатываемый вариант имеет более высокий показатель эксплуатационно-технического уровня по сравнению с базовым вариантом.

Эксплуатационно-технический уровень (ЭТУ) разрабатываемого программного продукта – это обобщенная характеристика его эксплуатационных свойств, возможностей, степени новизны, являющихся основой качества продукта.

Коэффициент эксплуатационно-технического уровня $K_{\text{эту}}$ рассчитывается по формуле

$$K_{\text{эту}} = \text{ЗТР}_{\text{пр}} / \text{ЗТР}_{\text{баз}}, \quad (2)$$

где $\text{ЗТР}_{\text{пр}}$ и $\text{ЗТР}_{\text{баз}}$ – это значимость технического решения для разрабатываемого и базового вариантов соответственно.

$$K_{\text{эту}} = 4,8/2 = 2,4$$

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

Кроме того, для уверенности в обоснованности автоматизации можно использовать обобщающий индекс эксплуатационно-технического уровня $I_{\text{эту}}$ (комплексный показатель качества проекта по группе показателей), который рассчитывается по формуле

$$I_{\text{эту}} = \sum b_i X_i, \quad (3)$$

где b_i – коэффициент весомости i -ого показателя;

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

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

В таблице 6 представлены результаты расчета балльно-индексным методом.

Таблица 6 – Расчет балльно-индексный методом

Показатель качества	Весовой коэффициент, b_i	Оценка, X_i	
		Разрабатываемый проект	Базовый проект
Удобство работы	0,2	3	2
Надежность (защита данных)	0,2	3	3
Функциональные возможности	0,3	3	2
Временная экономичность	0,2	3	1
Время обучения персонала	0,1	2	2

Продолжение таблицы 6

Комплексный показатель качества $I_{эту}$	2,9	2
---	-----	---

Коэффициент технического уровня рассчитывается по формуле

$$K_T = I_{этупр} / I_{этубаз}, \quad (4)$$

где $I_{этупр}$ – это комплексный показатель качества для разрабатываемого варианта;

$I_{этубаз}$ — это комплексный показатель качества базового варианта.

$$K_T = 2,9 / 2 = 1,45$$

Для расчета экономического эффекта необходимо рассчитать приведенные затраты на единицу работ, выполняемых по базовому и разрабатываемому вариантам:

$$Z_i = C_i + E_H * Z_{ппi}, \quad (5)$$

где C_i – текущие эксплуатационные затраты -ого вида работ;

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

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

Для базового варианта текущие эксплуатационные затраты будут состоять из заработной платы сотрудника, который занимается сбором данных, за год. Средний оклад специалиста равен 13890 рублей. С учетом северного и районного коэффициентов, а также отчислений во внебюджетные фонды ЗП будет равняться: $(13890 * 1,6) * 1,302 = 28935,648$ рублей в месяц. Исходя из 24 рабочих дней и 8-часового рабочего дня зарплата в час будет равняться $28935,648 / 24 / 8 = 150,7$ руб./час. В среднем специалист тратит около 3 часов

своего рабочего времени на данную деятельность, в году 247 рабочих дней.
Таким образом эксплуатационные затраты базового варианта:

$$C_{\text{баз}} = 150,7 * 3 * 247 = 111668,7 \text{ рублей.}$$

$$Z_{\text{баз}} = 111668,7 + 0,33 * 0 = 111668,7 \text{ рублей.}$$

Для проектного варианта эксплуатационные затраты будут равняться:

$$C_{\text{пр}} = 150,7 * 1 * 247 = 37222,9 \text{ рублей.}$$

$$Z_{\text{пр}} = 37\,222,9 + 0,33 * 84924 = 65247,82 \text{ рублей.}$$

Экономический эффект от использования разрабатываемой системы определяется по формуле

$$\mathcal{E} = (Z_{\text{баз}} * K_{\text{T}} - Z_{\text{пр}}) * V, \quad (6)$$

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

K_{T} – коэффициент эксплуатационно-технической эквивалентности (формула (4));

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

Экономический эффект от использования разрабатываемой системы:

$$\mathcal{E} = (111668,7 * 1,45 - 65\,247,82) * 1 = 96670,795 \text{ рублей.}$$

Также необходимо рассчитать срок окупаемости затрат на разработку проекта по формуле

$$T_{ок} = Z_{пп} / Э, \quad (7)$$

где $Z_{пп}$ – единовременные затраты на разработку проекта;
 $Э$ – годовая эффективность.

Расчет срок окупаемости затрат на разработку проекта

$$T_{ок} = 84924 / 96670,795 = 0,87 \text{ года} = 10 \text{ месяца } 14 \text{ дней.}$$

Таким образом, срок окупаемости проекта составляет около десяти месяцев.

Фактический коэффициент экономической эффективности разработки

$$E_{ф} = 1 / T_{ок} \quad (8)$$

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

$$E_{ф} = 1 / 0,87 = 1,14$$

Так как $E_{ф} > E_{н}$, то разработка и внедрение разрабатываемого ИС является эффективной, т.е. эффект от использования данной системы окупает все затраты, связанные с проектированием и эксплуатацией. Ниже приведены сводные данные экономического обоснования разработки и внедрения проекта:

- Затраты на разработку проекта: 84924 рублей.
- Общие эксплуатационные затраты: 22396 рублей.
- Коэффициент экономической эффективности: 4,8.

– Срок окупаемости: 10 месяцев.

3.2 Эксплуатационные риски

Рассмотрим возможные риски после внедрения информационной системы в работу ГБУЗ РХ «РМИАЦ». Возможные риски отображены в таблице 7.

Таблица 7 – Эксплуатационные риски

№	Перечень рисков проекта	Вероятность возникновения риска	Степень воздействия риска на проект	Возможные решения риска
1	Перегрузка системы при одновременном изменении данных в базе данных	Низкая	Высокая	Репликация базы данных, создание очереди записи данных
2	Осуществлен несанкционированный доступ к данным	Низкая	Средняя	Создание истории действий пользователей. Каждую неделю создавать резервную копию базы данных
3	Изменение структуры системы или базы данных	Средняя	Высокая	Хранение резервной копии рабочей версии приложения

Системой будут пользоваться множество администраторов, при одновременном использовании может возникнуть риск перегрузки системы при изменении данных в системе. Риск имеет высокую степень возникновения и напрямую влияет на работоспособность системы. Чтобы нейтрализовать этот риск, необходимо создать очередность внесения данных с последующей синхронизацией изменений, поменять структуру базы данных на более облегченную, также чтобы точно сохранять информацию, то необходимо использовать репликацию данных.

Во время использования есть вероятность того, что к системе может получить доступ неизвестный пользователь, который захочет удалить данные. Вероятность возникновения риска крайне мала, так как информация, содержащаяся в базе данных, не обладает высокой стоимостью. Чтобы нейтрализовать данный риск, необходимо записывать действия пользователя с базой данных в историю. Также для возможности восстановления данных, требуется создавать резервную копию информации в базе данных.

Также в организации ГБУЗ РХ «РМИАЦ» могут захотеть добавить новые данные в базу данных или вовсе поменять структуру информационной системы для добавления новых возможностей. Данный риск имеет высокую вероятность возникновения, так как информационные процессы не стоят на месте, и требуется постоянно адаптировать системы к новым данным. Чтобы устранить данный риск, требуется хранить резервную копию работающей версии программы, чтобы в случае неудачного изменения структуры приложения, можно было не потерять рабочую версию приложения с данными.

3.3 Выводы по разделу «Расчёт экономической эффективности проекта «Паспорт цифровизации ГБУЗ РХ «РМИАЦ»»

В данном разделе были выполнены расчеты эксплуатационных затрат для реализации информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ». Также проведена оценка рисков, которые могут возникнуть за время эксплуатации информационной системы, определены мероприятия по снижению возникновения этих рисков в проекте.

ЗАКЛЮЧЕНИЕ

Результатом выпускной квалификационной работы является разработанная информационная система «Паспорт цифровизации ГБУЗ РХ «РМИАЦ».

Во время разработки информационной системы «Паспорт цифровизации ГБУЗ РХ «РМИАЦ» выполнены следующие задачи:

- 1 Изучена основная деятельность предприятия ГБУЗ РХ «РМИАЦ».
- 2 Обоснована необходимость разработки информационной системы.
- 3 Поставлена цель разработки программы, сформулированы задачи подлежащих решению.

- 4 Разработана информационная система.

- 5 Дана оценка экономической эффективности внедрения разработанного веб-приложения.

- 6 Рассмотрены эксплуатационные риски.

В будущем, имеется возможность усовершенствования или изменения структуры веб-приложения, так как фреймворк Django имеет возможность подключения новых шаблонов и функций, при этом не затрагивая старые функции приложения. Возможно создание на основе созданного приложения, нового с улучшенным функционалом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Django 4.0 на русском языке: официальный сайт. – 2022. – URL: <https://django.fun/docs/django/ru/4.0/> (дата обращения: 24.05.2022).
2. Django Documentation 4.0. The Django admin site: официальный сайт. – 2022. – URL: <https://django.fun/docs/django/ru/4.0/> (дата обращения: 24.05.2022).
3. JinjaDocumeentation: официальный сайт. – 2022. – URL: <https://jinja.palletsprojects.com/en/3.1.x/> (дата обращения: 24.05.2022).
4. Межсайтовая подделка запроса: официальный сайт. – 2022. – URL: https://ru.wikipedia.org/wiki/Межсайтовая_подделка_запроса (дата обращения: 24.05.2022).
5. Руководство Django Часть 8: Аутентификация и авторизация пользователя: официальный сайт. – 2022. – URL: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Authentication> (дата обращения: 24.05.2022).
6. ГБУЗ РХ «РМИАЦ»: официальный сайт. – 2021. – URL: <https://miac.mz19.ru/miac/> (дата обращения: 24.05.2022).
7. Request and response objects: официальный сайт. – 2022. – URL: <https://docs.djangoproject.com/en/4.0/ref/request-response/> (дата обращения: 24.05.2022).
8. How to manage static files (e.g. images, JavaScript, CSS): официальный сайт. – 2022. – URL: <https://docs.djangoproject.com/en/4.0/howto/static-files/> (дата обращения: 24.05.2022).
9. СТУ 7.5–07–2021 СТАНДАРТ УНИВЕРСИТЕТА : Система менеджмента качества : Общие требования к построению, изложению и оформлению документов учебной деятельности : сайт / Сибирский Федеральный Университет. – Красноярск : СФУ, 2021. – URL: <https://about.sfu-kras.ru/docs/8127/pdf/808588> (дата обращения: 05.04.2022).

10. ГОСТ Р ИСО 9001-2001 Системы менеджмента качества. Требования. Введ. 31.08.2001.
11. ГОСТ ISO/IEC/IEEE 29148:2018 Программная и системная инженерия. Процессы жизненного цикла. Разработка требований. Введ. 28.11.2018.
12. Выполнение и защита выпускной квалификационной работы по направлению 09.03.03 «Прикладная информатика» : метод. указания / сост. Е. Н. Скуратенко, В. И. Кокова, И. В. Янченко ; Сиб. федер. ун-т, ХТИ – филиал СФУ. – Абакан : ХТИ – филиал СФУ, 2017. – 78 с.
13. Министерство здравоохранения Республики Хакасия: официальный сайт. Медицинские организации подведомственные Министерству здравоохранения Республики Хакасия. – 2021. – URL: <https://mz19.ru/clinic/> (дата обращения: 24.05.2022).
14. Python : The official home of the Python Programming Language : официальный сайт. – 2022. – URL : <https://www.python.org/> (дата обращения: 12.04.2022).
15. PyCharm : The Python IDE for Professional Developers : официальный сайт. – 2022. – URL: <https://www.jetbrains.com/pycharm/> (дата обращения: 12.04.2022).

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 15 наименований.

Один экземпляр сдан на кафедру.

« ____ » _____ 2022 г.

_____ Озол Виктор Андреевич
подпись

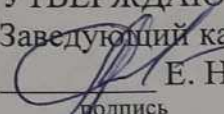
Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

Кафедра прикладной информатики, математики и естественно-научных
дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой


Е. Н. Скуратенко

подпись

« 17 » 06 2022 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.03 Прикладная информатика

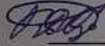
Разработка информационной системы «Паспорт цифровизации ГБУЗ РХ
«РМИАЦ»

Руководитель


17.06.22
подпись, дата

доцент, канд. техн. наук Е. Н. Скуратенко

Выпускник


17.06.2022
подпись, дата

В. А. Озол

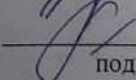
Консультанты
по разделам:

Экономический


17.06.22
подпись, дата

Е. Н. Скуратенко

Нормоконтролер


17.06.22
подпись, дата

В. И. Кокова

Абакан 2022