**Министерство науки и высшего образования Российской федерации**

**ФГБОУ ВО «Магнитогорский государственный технический университет им. Носова»**

Кафедра информатики и информационной безопасности

КУРСОВАЯ РАБОТА

по дисциплине «Методы проектирования защищенных распределенных информационных систем»

на тему: «Проектирование Автоматизированной Информационной Системы для поликлиники»

Исполнитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа допущена к защите "\_\_\_\_\_" \_\_\_\_\_\_\_\_\_ 2021г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена "\_\_\_\_\_" \_\_\_\_\_\_\_\_\_ 2021г. с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

Магнитогорск, 2020

**Министерство науки и высшего образования Российской федерации**

**ФГБОУ ВО «Магнитогорский государственный технический университет им. Носова»**

Кафедра информатики и информационной безопасности

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Тема: «Проектирование Автоматизированной Информационной Системы для поликлиники»

Студенту Шпак Виталию Алексеевичу

Исходные данные: Разработать проект локальной вычислительной сети как основы комплекса технических средств информационной системы поликлиники. Сконфигурировать и рассчитать локальную сеть Ethernet. Определить структуру, способ использования вычислительной сети кампуса и необходимый перечень услуг для информационной системы в заданной предметной области.

Срок сдачи: «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021г

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Магнитогорск, 2021

**Содержание**

[**Введение** 5](#_Toc62402420)

[**1** **Техническое задание на разработку АИС “Поликлиника”** 6](#_Toc62402421)

[1.1 Общие сведения 6](#_Toc62402422)

[1.2 Назначение и цели создания системы 6](#_Toc62402423)

[1.2.1 Назначение системы 6](#_Toc62402424)

[1.2.2 Цели создания системы 6](#_Toc62402425)

[1.3 Характеристика объектов автоматизации 7](#_Toc62402426)

[1.4 Требования к системе 7](#_Toc62402427)

[1.4.1 Требования к системе в целом 7](#_Toc62402428)

[1.4.2 Требования к функциям, выполняемым системой 11](#_Toc62402429)

[1.4.3 Требования к видам обеспечения 11](#_Toc62402430)

[1.5 Состав и содержание работ по созданию системы 13](#_Toc62402431)

[1.6 Порядок контроля и приемки системы 13](#_Toc62402432)

[1.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 14](#_Toc62402433)

[1.8 Источники разработки 14](#_Toc62402434)

[**2** **Выбор топологии сети для АИС “Поликлиника”** 15](#_Toc62402435)

[**3** **Сетевые протоколы, используемые в АИС** 16](#_Toc62402436)

[3.1 Rapid Spanning Tree Protocol 16](#_Toc62402437)

[3.2 VLAN Trunking Protocol 16](#_Toc62402438)

[3.3 Terminal Access Controller Access Control System plus 17](#_Toc62402439)

[3.4 Secure Shell v2 18](#_Toc62402440)

[3.5 HyperText Transfer Protocol Secure 19](#_Toc62402441)

[3.6 Dynamic Host Configuration Protocol 19](#_Toc62402442)

[**4** **Маршрутизация в компьютерной сети.** 20](#_Toc62402443)

[**5** **Анализ СЗИ, используемых в разрабатываемой компьютерной сети** 21](#_Toc62402444)

[5.1 Authentication Authorization and Accounting 21](#_Toc62402445)

[5.2 Access Control List 21](#_Toc62402446)

[5.3 Secure Shell 22](#_Toc62402447)

[5.4 Virtual Private Network 23](#_Toc62402448)

[5.5 Redundant Array of Independent Disks 24](#_Toc62402449)

[5.6 Virtual Local Area Network 25](#_Toc62402450)

[5.7 Iptables 26](#_Toc62402451)

[5.8 Simplified Mandatory Access Control Kernel 27](#_Toc62402452)

[5.9 StaffCounter 29](#_Toc62402453)

[**6** **Выбор программного обеспечения на серверах и АРМ** 30](#_Toc62402454)

[6.1 Операционная система 30](#_Toc62402455)

[6.2 Средство разработки и отладки программ 30](#_Toc62402456)

[6.3 Браузер 30](#_Toc62402457)

[6.4 Пакет офисных программ 31](#_Toc62402458)

[6.5 Приложение организации домена поликлиники 31](#_Toc62402459)

[**7** **Построение схемы сети. Выбор необходимого оборудования** 32](#_Toc62402460)

[**8** **Обзор разработанной ИС для обработки амбулаторных карт** 37](#_Toc62402461)

[**Заключение** 41](#_Toc62402462)

[**Список использованных источников** 42](#_Toc62402463)

[**Приложение А** 44](#_Toc62402464)

[**Приложение Б** 48](#_Toc62402465)

# **Введение**

В настоящее время практика использования информационных систем в производстве или в документообороте предприятий стала обыденной, перечень охватываемых ими задач постоянно расширяется. Благодаря этому постоянно растет объем и сложность обрабатываемой информации, что позволяет специалистам производить намного более точные операции, основываясь на достоверности данных выборочных совокупностей.

Вот только некоторые из преимуществ, которые дает использование вычислительной техники при работе организации:

* Возможность оперативного контроля за достоверностью информации;
* Уменьшение числа возможных ошибок при генерировании производных данных;
* Возможность быстрого доступа к любым данным;
* Возможность быстрого формирования отчетов;
* Экономия трудозатрат и затрат времени на обработку информации.

Все эти преимущества в данный момент оценены многими организациями, поэтому сегодня наблюдается процесс бурного развития специализированных информационных систем и внедрения их в работу различных учреждений. Фактически проблема комплексной автоматизации стала актуальной для каждого предприятия.

В ходе выполнения данной работы необходимо спроектировать защищенную распределенную информационную систему для поликлиники и разработать внутренний портал для служебного пользования, выполняющий функции ведения медицинской карты в электронном виде, содержащей данные о пациенте и состоянии его здоровья на основании проведенных врачебных осмотров, лабораторных и других видов исследований. Использование информационной системы позволяет централизованно хранить данные пациента, что дает возможность просматривать информацию его истории болезни различным специалистам лечебно-профилактических учреждений без передачи между ними бумажной амбулаторной карты.

Для выполнения поставленных целей нужно разработать и оформить техническое задание на проектирование АИС, выбрать и обосновать выбор аппаратного и программного обеспечения, используемого в информационной системе, проанализировать средства защиты информации, используемые в разрабатываемой компьютерной сети.

# **Техническое задание на разработку АИС “Поликлиника”**

## Общие сведения

Полное наименование системы: Автоматизированная Информационная Система “Поликлиника”. Краткое наименование системы: АИС, Система.

Работа выполняется на основании Договора “ПРОЕКТИРОВАНИЕ ЗАЩИЩЕННЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ. Методические указания по выполнению курсовой работы по дисциплине «Методы проектирования защищенных распределенных информационных систем» для студентов специальностей 09030307, 100503” от 17.11.2015 г.

Заказчик: ГАУЗ “Городская поликлиника №17”

Адрес фактический: г. Магнитогорск ул. Доменщиков, 3

Телефон / Факс: +7 (351) 9307311

Разработчик: ИП “Bot”

Адрес фактический: г. Магнитогорск ул. Пионерская, 419

Телефон / Факс: +7 (902) 4056944

Плановая дата начала работ по созданию системы: 09.12.2020

Плановая дата окончания работ по созданию системы: 20.01.2021

Источники и порядок финансирования указаны в договоре №583 от 17.11.2020

Работы по созданию АИС сдаются Разработчиком целиком в соответствии с календарным планом Проекта. По окончании работ по созданию Системы Разработчик сдает Заказчику соответствующие отчетные документы, состав которых определены Договором.

## Назначение и цели создания системы

### Назначение системы

Автоматизированная Информационная Система “Поликлиника” предназначена для автоматизации процесса ведения больничной документации (амбулаторные карты пациентов), аналитической обработки данных и представления полученных данных в форме самостоятельного отчета произвольной формы.

### Цели создания системы

АИС создается с целью:

* Обеспечения сбора и обработки информации о пациентах;
* Повышения качества (полноты, точности, достоверности, своевременности, согласованности, безопасности) информации;
* Обеспечение централизованного выхода в интернет;

В результате создания Системы должны быть улучшены значения следующих показателей:

* время сбора и первичной обработки исходной информации;
* время, затрачиваемое на информационно-аналитическую деятельность;

## Характеристика объектов автоматизации

Выделены следующие процессы в деятельности ГАУЗ “Городская поликлиника №17”, в рамках которых производится анализ информации и вынесены соответствующие выводы о возможности их автоматизации (табл. 1).

Таблица 1 - Решение об автоматизации структурных подразделений

|  |  |  |  |
| --- | --- | --- | --- |
| Структурное подразделение | Наименование процесса | Возможность автоматизации | Решение об автоматизации в ходе проекта |
| Руководящий отдел | Принятие решений в области развития ГАУЗ | Возможна | Не будет автоматизирован |
| Бухгалтерия | Управление финансовой деятельностью ГАУЗ | Возможна | Не будет автоматизирован |
| Отдел кадров | Управление кадровым составом ГАУЗ | Возможна | Не будет автоматизирован |
| Архив | Хранение информации | Возможна | Не будет автоматизирован |
| Технический отдел | Обеспечение работоспособности Системы | Возможна | Будет автоматизирован |
| Отдел работы с пациентами | Создание и редактирование информации о пациентах | Возможна | Будет автоматизирован |

## Требования к системе

### Требования к системе в целом

Система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище. АИС должна иметь трехуровневую архитектуру (первый - источник, второй - хранилище, третий - отчетность).

В Системе предлагается выделить следующие функциональные подсистемы:

* подсистема сбора, обработки и загрузки данных, которая предназначена для реализации процессов сбора данных из систем источников, приведения указанных данных к виду, необходимому для наполнения подсистемы хранения данных;
* подсистема хранения данных, которая предназначена для хранения данных в структурах, нацеленных на принятие решений;
* подсистема формирования и визуализации отчетности.

В качестве протокола взаимодействия между компонентами Системы на транспортно-сетевом уровне необходимо использовать протокол TCP/IP.

Для организации информационного обмена между компонентами Системы должны использоваться специальные протоколы прикладного уровня, такие как: HTTPS, NetBios/SMB, SSH.

Для организации доступа пользователей к отчетности должен использоваться протокол презентационного уровня HTTPS.

Источниками данных для Системы должны быть:

* База данных пользователей (СУБД SQLite 3).
* База данных пациентов и их амбулаторных карт (SQLite 3).

Система должна поддерживать следующие режимы функционирования:

* Основной режим, в котором подсистемы АИС выполняют все свои основные функции.
* Профилактический режим, в котором одна или все подсистемы АИС не выполняют своих функций.

В основном режиме функционирования Системы должна обеспечивать:

* работу пользователей в режиме – 12 часов в день, 6 дней в неделю (12х6);
* выполнение своих функций – сбор, обработка и загрузка данных; хранение данных, предоставление отчетности.

В профилактическом режиме Система должна обеспечивать возможность проведения следующих работ:

* техническое обслуживание;
* модернизацию аппаратно-программного комплекса;
* устранение аварийных ситуаций.

Общее время проведения профилактических работ не должно превышать 5% от общего времени работы системы в основном режиме (300 часов в месяц).

В состав персонала, необходимого для обеспечения эксплуатации АИС в рамках соответствующих подразделений Заказчика, необходимо выделение следующих ответственных лиц:

* Руководитель эксплуатирующего подразделения - 1 человек.
* Администратор подсистемы сбора, обработки и загрузки данных - 2 человека.
* Администратор подсистемы хранения данных - 2 человека.
* Администратор подсистемы формирования и визуализации отчетности - 1 человек.

Данные лица должны выполнять следующие функциональные обязанности.

* Руководитель эксплуатирующего подразделения - на всем протяжении функционирования АИС обеспечивает общее руководство группой сопровождения,
* Администратор подсистемы сбора, обработки и загрузки данных - на всем протяжении функционирования АИС обеспечивает контроль процессов, подготовку и загрузка данных из внешних источников в хранилище данных.
* Администратор подсистемы хранения данных - на всем протяжении функционирования АИС обеспечивает распределение дискового пространства, модификацию структур БД, оптимизацию производительности.
* Администратор подсистемы формирования и визуализации отчетности - на всем протяжении функционирования АИС обеспечивает поддержку пользователей.

К квалификации персонала, эксплуатирующего Систему, предъявляются следующие требования.

* Конечный пользователь - знание соответствующей предметной области;
* Администратор подсистемы сбора, обработки и загрузки данных - знание методологии проектирования хранилищ данных; знание СУБД.
* Администратор подсистемы хранения данных - глубокие знания СУБД; знание архитектуры «Звезда» и «Снежинка»; опыт администрирования СУБД; знание и навыки операций архивирования и восстановления данных; знание и навыки оптимизации работы СУБД.
* Администратор подсистемы формирования и визуализации отчетности - понимание принципов многомерного анализа; знание методологии проектирования хранилищ данных; знание и навыки администрирования приложения; знание инструментов разработки.

Персонал, работающий с Системой и выполняющий функции её сопровождения и обслуживания, должен работать в следующих режимах:

* Конечный пользователь - в соответствии с основным рабочим графиком подразделений Заказчика.
* Администратор подсистемы сбора, обработки и загрузки данных – двухсменный график, поочередно.
* Администратор подсистемы хранения данных – двухсменный график, поочередно.
* Администратор подсистемы формирования и визуализации отчетности – в соответствии с основным рабочим графиком подразделений Заказчика.

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств.

Надежность должна обеспечиваться за счет:

* применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;
* своевременного выполнения процессов администрирования Системы;
* соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;
* предварительного обучения пользователей и обслуживающего персонала.

Подсистема формирования и визуализации отчетности данных должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий следующим требованиям:

* интерфейсы подсистем должен быть типизированы;
* должно быть обеспечено наличие русскоязычного интерфейса пользователя;
* должен использоваться шрифт: Arial, размер 12 пунктов;
* цветовая палитра должна быть выполнена в синем цвете и его оттенках;
* в приложении должен использоваться логотип Заказчика.

Условия эксплуатации, а также виды и периодичность обслуживания технических средств Системы должны соответствовать требованиям по эксплуатации, техническому обслуживанию, ремонту и хранению, изложенным в документации завода-изготовителя (производителя) на них.

Технические средства Системы и персонал должны размещаться в существующих помещениях Заказчика, которые по климатическим условиям должны соответствовать ГОСТ 15150-69 «Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды» (температура окружающего воздуха от 5 до 40 °С, относительная влажность от 40 до 80 % при Т=25 °С, атмосферное давление от 630 до 800 мм ртутного столба). Размещение технических средств и организация автоматизированных рабочих мест должны быть выполнены в соответствии с требованиями ГОСТ 21958-76 «Система "Человек-машина". Зал и кабины операторов. Взаимное расположение рабочих мест. Общие эргономические требования».

Для электропитания технических средств должна быть предусмотрена трехфазная четырехпроводная сеть с глухо заземленной нейтралью 380/220 В (+10-15)% частотой 50 Гц (+1-1) Гц. Каждое техническое средство запитывается однофазным напряжением 220 В частотой 50 Гц через сетевые розетки с заземляющим контактом.

Для обеспечения выполнения требований по надежности должен быть создан комплект запасных изделий и приборов (ЗИП).

Обеспечение информационное безопасности АИС должно удовлетворять следующим требованиям:

* Защита Системы должна обеспечиваться комплексом программно-технических средств и поддерживающих их организационных мер.
* Защита Системы должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.
* Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики Системы (надежность, быстродействие, возможность изменения конфигурации).
* Разграничение прав доступа пользователей и администраторов Системы должно строиться по принципу "что не разрешено, то запрещено".

### Требования к функциям, выполняемым системой

Разрабатываемая система должна обеспечивать:

* Формирование последовательности выполнения процессов сбора, обработки и загрузки данных
* Запуск процедур сбора данных из систем источников, загрузка данных в область временного, постоянного хранения
* Обработка и преобразование извлеченных данных
* Ведение журналов результатов сбора, обработки и загрузки данных
* Оперативное извещение пользователей о всех нештатных ситуациях в процессе работы подсистемы

### Требования к видам обеспечения

Структура хранения данных в АИС должна состоять из следующих основных областей:

* область временного хранения данных;
* область постоянного хранения данных;
* область витрин данных.

Области постоянного хранения и витрин данных должны строиться на основе многомерной модели данных, подразумевающей выделение отдельных измерений и фактов с их анализом по выбранным измерениям.

Многомерная модель данных физически должна быть реализована в реляционной СУБД по схеме «звезда» и/или «снежинка».

Система не должна быть закрытой для смежных систем и должна поддерживать возможность экспорта данных в смежные системы через интерфейсные таблицы или файлы данных. Система должна обеспечить возможность загрузки данных, получаемых от смежной системы.

Для реализации подсистемы хранения данных должна использоваться промышленная СУБД SQLite. Информация в базе данных системы должна сохраняться при возникновении аварийных ситуаций, связанных со сбоями электропитания.

Система должна иметь бесперебойное электропитание, обеспечивающее её нормальное функционирование в течение 15 минут в случае отсутствия внешнего энергоснабжения, и 5 минут дополнительно для корректного завершения всех процессов.

Резервное копирование данных должно осуществляться на регулярной основе, в объемах, достаточных для восстановления информации в подсистеме хранения данных.

Система должна протоколировать все события, связанные с изменением своего информационного наполнения, и иметь возможность в случае сбоя в работе восстанавливать свое состояние, используя ранее запротоколированные изменения данных.

К хранению данных предъявляется требования хранения исторических данных в системе не более чем за 5 (пять) предыдущих лет. По истечению данного срока данные должны переходить в архив;

К обновлению и восстановлению данных предъявляются следующие требования:

* для сервера сбора, обработки и загрузки данных необходимо обеспечить резервное копирование его бинарных файлов (Home) раз в 2 недели и хранение копии на протяжении 2-х месяцев;
* для сервера базы данных необходимо обеспечить резервное копирование его бинарных файлов раз в 2 недели и хранение копии на протяжении 2-х месяцев;

При реализации системы должны применяться следующие языки высокого уровня: SQL, Python, JavaScript. При реализации системы должны применяться следующие языки и стандарты взаимодействия сервера со смежными системами, и пользователей: должны использоваться встроенные средства диалогового взаимодействия; JavaScript; HTML; CSS; Для реализации алгоритмов манипулирования данными в ХД необходимо использовать стандартный язык запроса к данным SQL. Для организации диалога системы с пользователем должен применяться графический оконный пользовательский интерфейс.

К обеспечению качества программной среды (ПС) предъявляются следующие требования:

* функциональность должна обеспечиваться выполнением подсистемами всех их функций.
* надежность должна обеспечиваться за счет предупреждения ошибок - не допущения ошибок в готовых ПС;
* легкость применения должна обеспечиваться за счет применения покупных программных средств;
* эффективность должна обеспечиваться за счет принятия подходящих, верных решений на разных этапах разработки ПС и системы в целом;
* сопровождаемость должна обеспечиваться за счет высокого качества документации по сопровождению, а также за счет использования в программном тексте описания объектов и комментариев; использованием осмысленных (мнемонических) и устойчиво различимых имен объектов; избеганием создания фрагментов текстов программ с неочевидным или скрытым смыслом.

Система должна быть реализована с использованием специально выделенных серверов Заказчика.

## Состав и содержание работ по созданию системы

Работы по созданию системы выполняются в три этапа:

* Проектирование. Разработка эскизного проекта. Разработка технического проекта (продолжительность — 1 месяц).
* Разработка рабочей документации. Создание и адаптация программ (продолжительность — 1 месяц).
* Ввод в действие (продолжительность — 1 месяц).

Конкретные сроки выполнения стадий и этапов разработки и создания Системы определяются Планом выполнения работ, являющимся неотъемлемой частью Договора на выполнение работ по настоящему Частному техническому заданию.

## Порядок контроля и приемки системы

Прием работ по разработке АИС “Поликлиника” проводится на стадии приемочных испытаний на территории Заказчика, с 30.12.2020 по 29.01.2021 с участием организаций Заказчика и Разработчика. На этом этапе происходят: проведение приемочных испытаний, фиксирование выявленных неполадок в Протоколе испытаний, принятие решения о возможности передачи АИС в эксплуатацию, составление и подписание Акта о завершении приемочных испытаний и передаче АИС в промышленную эксплуатацию, оформление Акта завершения работ.

## Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для создания условий функционирования АИС, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в настоящем техническом задании, и возможность эффективного ее использования, в организации Заказчика должен быть проведен комплекс мероприятий:

* приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ;
* составление списка изменений, которые необходимо осуществить в объекте автоматизации;
* создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ;
* создание необходимых для функционирования системы подразделений и служб.

## Источники разработки

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

* Договор “ПРОЕКТИРОВАНИЕ ЗАЩИЩЕННЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ. Методические указания по выполнению курсовой работы по дисциплине «Методы проектирования защищенных распределенных информационных систем» для студентов специальностей 09030307, 100503” от 17.11.2015 г.
* ГОСТ 24.701-86 «Надежность автоматизированных систем управления».
* ГОСТ 15150-69 «Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды».
* ГОСТ 21958-76 «Система "Человек-машина". Зал и кабины операторов. Взаимное расположение рабочих мест. Общие эргономические требования».

# **Выбор топологии сети для АИС “Поликлиника”**

Компьютеры и другие компоненты локальной сети могут соединяться между собой различными способами. Используемая схема физического расположения сетевых компонентов называется топологией. На данный момент существует несколько топологии построения локальной сети:

1. **Полносвязная топология.** В такой топологии для каждой пары должна быть выделена независимая линия, каждый компьютер должен иметь столько коммуникационных портов сколько компьютеров в сети. По этой причине полносвязная топология не подходит для АИС “Поликлиника”.
2. **Ячеистая топология.**  Не подходит для создания АИС по тем же причинам, что и полносвязная топология.
3. **Общая шина.** Такая топология очень простая и дешевая, но возникновении поломки в сети блокирует всю сеть, а локализация неисправности сложна. Также данная топология не подходит по причине прямой зависимости между производительностью сети и количеством рабочих станций.
4. **Кольцо.** Недостатки топологии “Кольцо” те же, что и у топологии “Шина”.
5. **Звезда.** При такой топологии подключение новых рабочих станций не вызывает особых затруднений. “Звезда” позволяет проводить мониторинг сети и централизованное управления сетью. Также эта топология обеспечивает хорошую расширяемость и модернизацию сети, но для создания распределенной ИС не подходит.
6. **Снежинка.** Эта топология наиболее предпочтительна для построения больших распределенных ИС, т.к. она обладает всеми преимуществами топологии типа “Звезда”, позволяет разделять сегменты сети на отдельные рабочие группы и сохраняет свою работоспособность при отключении от сети рабочих групп по причине поломки.

Таким образом в проектируемой ИС будет использоваться смешанная топология, созданная на основе топологий типа “Снежинка” и “Кольцо”. Использование “Снежинки” обуславливается необходимостью централизации и расширяемости сети. Топология “Кольцо” обеспечит отказоустойчивость сети при межфилиальном общении за счет создания более одного маршрута для сетевого трафика.

# **Сетевые протоколы, используемые в АИС**

## Rapid Spanning Tree Protocol

Для обеспечения отказоустойчивости сетевого взаимодействия между филиалами поликлиники необходимо обеспечить избыточность сетевых путей. Так, в данном случае, крайние филиалы соединены оптоволоконным проводом не только с центральным филиалом, но и с ближайшими филиалами. Такие топологии подвержены возникновению широковещательного шторма из-за образовавшихся петель из лишних соединений. Для избегания проблем с маршрутизацией в получившемся кольце используется Rapid Spanning Tree Protocol.

Rapid Spanning Tree Protocol — сетевой протокол, работающий на втором уровне модели OSI. Основан на алгоритме Spanning Tree Protocol, разработчиком которого является Радья Перлман.

Основной задачей RSTP является приведение сети Ethernet с множественными связями к древовидной топологии, исключающей циклы пакетов. Происходит это путем автоматического блокирования ненужных в данный момент для полной связности портов [1].

Для определения кратчайшего пути в сети выбирается один корневой мост.

Далее каждый, отличный от корневого, мост просчитывает кратчайший путь к корневому. Соответствующий порт называется корневым портом. У любого не корневого коммутатора может быть только один корневой порт.

После этого для каждого сегмента сети просчитывается кратчайший путь к корневому порту. Мост, через который проходит этот путь, становится назначенным для этой сети. Непосредственно подключенный к сети порт моста — назначенным портом.

Далее на всех мостах блокируются все порты, не являющиеся корневыми и назначенными. В итоге получается древовидная структура с вершиной в виде корневого коммутатора.

## VLAN Trunking Protocol

Распределенная ИС поликлиники насчитывает 36 коммутаторов, на каждом из которых нужно создать несколько VLAN’ов. Для облегчения работы сетевого администратора можно использовать протокол, служащий для обмена информацией о VLAN между сетевыми устройствами, такой как VTP или GVRP. Так как в данном случае в поликлинике используется сетевое оборудование CISCO, очевидно использование проприетарного протокола VTP.

Протокол формирования магистральных каналов виртуальной локальной сети (VTP) представляет собой удобное дополнение к средствам управления виртуальными локальными сетями. Он позволяет автоматически присваивать определения виртуальных локальных сетей сразу нескольким коммутаторам в сети.

Программное обеспечение VTP может функционировать в коммутаторе в одном из трех режимов: клиентском, серверном и прозрачном.

**Клиентский режим**. В этом режиме коммутатор принимает и распространяет анонсы VTP, которые относятся к его домену управления. С учетом этих анонсов коммутатор вносит изменения в свою конфигурацию виртуальной локальной сети. До тех пор, пока коммутатор находится в клиентском режиме, в нем не могут быть непосредственно внесены изменения в конфигурацию виртуальной локальной сети. Поэтому изменения в конфигурацию виртуальной локальной сети коммутатора, находящегося в клиентском режиме, могут быть внесены только с помощью протокола VTP.

**Серверный режим**. В этом режиме коммутатор также принимает и распространяет анонсы VTP, которые относятся к его домену управления, но наряду с этим формирует новые анонсы. Этот режим позволяет модифицировать информацию виртуальной локальной сети непосредственно в самом коммутаторе в том числе добавлять и удалять виртуальные локальные сети из домена управления. Модификация конфигурации VTP домена управления вызывает обновление номера версии конфигурации (а также номера версии базы данных VTP). Такое обновление вынуждает все коммутаторы в домене управления обновить свои конфигурации VTP с учетом новой информации.

Прозрачный режим. В прозрачном режиме информация VTP перенаправляется, но данные о конфигурации виртуальных локальных сетей, содержащиеся в этих анонсах, игнорируются. В противной режиме разрешается непосредственно вносить изменения в конфигурацию виртуальных локальных сетей в коммутаторе, но такие изменения в конфигурации относятся только к этому локальному коммутатору [2].

## Terminal Access Controller Access Control System plus

В данной АИС используется технология ААА. AAA (Authentication Authorization and Accounting) — система аутентификации авторизации и учета событий, встроенная в операционную систему Cisco IOS, служит для предоставления пользователям безопасного удаленного доступа к сетевому оборудованию Cisco. Она предлагает различные методы идентификации пользователя, авторизации, а также сбора и отправки информации на сервер [[3](https://habr.com/ru/company/pt/blog/192668/)].

В ААА используется два основных протокола:

* RADIUS
* TACACS+

Основные различия между этими протоколами указаны в таблице 2.

Таблица 2 - Сравнение протоколов RADIUS и TACACS+

|  |  |  |
| --- | --- | --- |
| Наименование параметра сравнения | Сравниваемые протоколы | |
| RADIUS | TACACS+ |
| Протокол | UDP  1812/1645 authentication  1813/1646 accounting | TCP/49 |
| Сервисы | Объединяет аутентификацию и авторизацию | Разные процессы для аутентификации, авторизации и учета |
| Шифрование | Шифруется только пароль | Шифруется все тело пакета |
| Основное назначение | Доступ к сети | Администрирование устройств |
| Особенности | Открытый стандарт | Проприетарная разработка CISCO |

Как видно из таблицы, протокол TACACS+ предпочтительнее для использования в данной сети, т.к. он позволит сохранить безопасность работы сети даже в случае перехвата злоумышленником пакетов авторизации.

## Secure Shell v2

Для обеспечения безопасности передачи данных нужно шифровать весь пользовательский трафик в локальной сети поликлиники. Для шифрования трафика обычно используется SSH. Помимо передачи данных через SSH можно его использовать для настройки сетевого оборудования, вместо telnet.

SSH делает удалённое управление операционной системой безопасным, так как шифрует весь трафик, включая и передаваемые пароли. При этом возможен выбор различных алгоритмов шифрования.

Кроме удаленного управления, SSH позволяет безопасно передавать в незащищенной среде практически любой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по шифрованному каналу звуковой поток или видео (например, с веб-камеры), производить работу с базами данных и другими хранилищами, а также использовать любые другие протоколы.

SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем, в дистрибутивах операционных систем семейства Linux клиент и сервер SSH обычно предустановлены [[4](https://web-creator.ru/articles/ssh)].

## HyperText Transfer Protocol Secure

Из-за того, что данная ИС обрабатывает данные, относящиеся не только к персональным данным, но и к врачебной тайне, использование шифрования для web трафика обязательно. В настоящее время общепринятым протоколом для шифрования интернет-трафика является HTTPS.

Обеспечение безопасной передачи данных необходимо на сайтах, где вводится и передается конфиденциальная информация (личные данные пользователей, детали доступа, реквизиты платежных карт) — на любых сайтах с авторизацией, взаимодействием с платежными системами, почтовыми сервисами. Шифрование таких данных позволит предотвратить их получение и использование третьими лицами.

Для реализации передачи данных посредством HTTPS на веб-сервере, обрабатывающем запросы от клиентов, должен быть установлен специальный SSL-сертификат. Есть сертификаты, защищающие только один домен. А есть сертификаты, которые обеспечивают защиту информации на всех поддоменах, и это Wildcard SSL. Шифрование происходит в обе стороны — как данных, полученных клиентом, так и данных, отправленных на сервер [5].

## Dynamic Host Configuration Protocol

DHCP — протокол прикладного уровня модели TCP/IP, служит для назначения IP-адреса клиенту. Это следует из его названия — Dynamic Host Configuration Protocol. IP-адрес можно назначать вручную каждому клиенту, то есть компьютеру в локальной сети. Но в больших сетях это очень трудозатратно, к тому же, чем больше локальная сеть, тем выше возрастает вероятность ошибки при настройке. Поэтому для автоматизации назначения IP был создан протокол DHCP.

Получение адреса проходит в четыре шага. Этот процесс называют DORA по первым буквам каждого шага: Discovery, Offer, Request, Acknowledgement.

Учитывая, что в ГАУЗ “Городская поликлиника №17” могут одновременно работать до 150 АРМ, использование DHCP обязательно.

# **Маршрутизация в компьютерной сети.**

Для разграничения доступа в сети было создано 6 виртуальных локальных сетей, подробные настройки которых указаны в таблице 3.

Таблица 3 - Настройки АРМ и серверов по VLAN’ам

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| VLAN | Филиал | Назначение | Пул IP адресов | Шлюз по умолчанию | Файловый сервер | Контроллер домена |
| 10 | Руководство | Разграничение сетей | 192.168.10.0-192.168.10.50 | 192.168.10.254 (L3 Switch) | 192.168.10.250 | 192.168.50.253 |
| 20 | Бухгалтерия | Разграничение сетей | 192.168.20.0-192.168.20.50 | 192.168.20.254 (L3 Switch) | 192.168.20.250 | 192.168.50.253 |
| 30 | Архив | Разграничение сетей | 192.168.30.0-192.168.30.50 | 192.168.30.254 (L3 Switch) | 192.168.30.250 | 192.168.50.253 |
| 40 | Поликлиника | Разграничение сетей | 192.168.40.0-192.168.40.50 | 192.168.40.254 (L3 Switch) | 192.168.40.250 | 192.168.50.253 |
| 50 | Центральный | Разграничение сетей | 192.168.50.0-192.168.50.50 | 192.168.50.254 (L3 Switch) | 192.168.50.250 | 192.168.50.253 |
| 100 | - | Разграничение сетей, выделение ip адреса для сотрудников, подключенных через VPN | 192.168.100.0-192.168.100.50 | 192.168.100.254 (L3 Switch) | 192.168.100.250 | 192.168.50.253 |

VLAN позволяет нескольким сетям работать практически как одна локальная сеть. Одним из наиболее полезных элементов VLAN является то, что он устраняет задержку в сети, что экономит сетевые ресурсы и повышает эффективность сети. Кроме того, VLAN созданы для обеспечения сегментации и поддержки в таких вопросах, как безопасность, управление сетью и масштабируемость. Трафик также можно легко контролировать с помощью VLAN.

# **Анализ СЗИ, используемых в разрабатываемой компьютерной сети**

## Authentication Authorization and Accounting

При строительстве и эксплуатации сети важно иметь строгий контроль над теми, кто имеет доступ к ее устройствам. Если сеть небольшая и число ее пользователей невелико, то следить за доступом и безопасностью не составляет труда, и часто за это отвечает один сетевой администратор. Но если организация имеет масштабную корпоративную сеть или это сеть оператора связи с большим числом абонентов, то отслеживать вручную, кто к чему имеет доступ, становится невозможно. В этом случае сетевыми администраторами используется AAA – механизм, который позволяет осуществлять аутентификацию, авторизацию и учет пользователей, то есть контролировать доступ и записывать производимые действия.

Принцип AAA можно описать так: для совершения какого-либо действия в сети AAA сервер должен проследить, кто инициирует это действие (authentication), имеет ли он право на выполнение этого действия (authorization) и что в журнал записаны все действия, которые он совершил (accounting).

Есть два основных типа AAA для сетей:

* Администрирование сетевых устройств. Осуществляет управление теми, у кого есть доступ для входа в консоль сетевого устройства, Telnet-сессии, SSH-сессии и т.д.
* Идентификация пользователя до того, как ему будет предоставлен доступ к сети или устройству (к примеру АРМ).

В современных сетях используют два основных решения для AAA: Remote Authentication Dial-In User Service (RADIUS) и Cisco’s Terminal Access Controller Access-Control System Plus (TACACS+) протоколы. Существует еще и третий AAA-протокол, известный как DIAMETER, но он обычно используется только мобильными операторами [6].

## Access Control List

В основном применение списков доступа рассматривают с точки зрения пакетной фильтрации, то есть пакетная фильтрация необходима в тех ситуациях, когда у вас стоит оборудование на границе Интернет и вашей частной сети и нужно отфильтровать ненужный трафик.

Функционал ACL состоит в классификации трафика, нужно его проверить сначала, а потом что-то с ним сделать в зависимости от того, куда ACL применяется. ACL применяется везде, например:

* На интерфейсе: пакетная фильтрация
* На линии Telnet: ограничения доступа к маршрутизатору
* VPN: какой трафик нужно шифровать
* QoS: какой трафик обрабатывать приоритетнее
* NAT: какие адреса транслировать

Применительно к пакетной фильтрации, ACL размещаются на интерфейсах. Сами правила создаются независимо, а уже потом их назначают на интерфейс. Как только правило назначено на интерфейс, маршрутизатор начинает просматривать трафик. Маршрутизатор рассматривает трафик как входящий и исходящий. Тот трафик, который входит в маршрутизатор называется входящим, тот который из него выходит — исходящий. Соответственно ACL размещаются на входящем или на исходящем направлении.

ACL представляет собой набор текстовых выражений, в которых написано permit (разрешить) или deny (запретить), и обработка ведется строго в том порядке, в котором заданы выражения. Соответственно, когда пакет попадает на интерфейс он проверяется на первое условие, если первое условие совпадает с пакетом, дальнейшая его обработка прекращается. Пакет либо перейдет дальше, либо уничтожится. В каждом конце списка стоит неявный deny any (запретить весь трафик).

ACL разделяются на два типа:

* Стандартные (Standard): могут проверять только адреса источников
* Расширенные (Extended): могут проверять адреса источников, а также адреса получателей, в случае IP ещё тип протокола и TCP/UDP порты

Обозначаются списки доступа либо номерами, либо символьными именами. ACL также используются для разных сетевых протоколов [7].

## Secure Shell

Сетевой протокол SSH используется для обмена информацией между двумя компьютерами по зашифрованному каналу с высокой степенью безопасности соединения, а также имеет ряд других возможностей.

В настоящий момент известны две рабочие версии протокола SSH:

* SSHv1. Версия 1995 года, сейчас фактически не используется из-за наличия серьезных недостатков в реализации, наиболее подвержена взлому злоумышленниками.
* SSHv2. Запущена спустя год после первой версии и является современной модификацией сетевого протокола, устойчивой к агрессивным атакам "man-in-the-middle", благодаря которой злоумышленники не смогут «прослушивать» трафик пользователя. Еще один уровень защиты второй версии — максимальная степень устойчивости к несанкционированному доступу путем «серединного» подсоединения (например, session hijacking), а также к приему подмены серверных имен (DNS spoofing).

Из вышеизложенного можно сделать вывод, что использование сетевого протокола SSH-2 является единственно приемлемым с точки зрения безопасности и сохранности конфиденциальной информации.

Аутентификация происходит либо с использованием пароля, либо с помощью специального SSH-ключа. Доступ по паролю считается небезопасным методом из-за возможности автоматического подбора, поэтому специалисты рекомендуют более безопасный способ — создание SSH-ключей.

Помимо использования ключей для более безопасной работы рекомендуется также изменить и ряд других настроек некоторых параметров серверной части протокола:

* Подключение с правами суперпользователя (root) и подключение с пустым паролем должны быть запрещены.
* Изменение стандартного 22 порта SSH-сервера на любой нестандартный. Также хорошей защитой от злоумышленников является закрытие порта (port knocking) — технология, при которой все порты сервера закрыты для подключения, а доступ осуществляется путем установленной последовательности пересылки IP-пакетов.
* Определение сетевых адресов, а также конкретных пользователей или групп пользователей, которым разрешено подключение, и запретить доступ заведомо неблагонадежным адресам, а также некоторым пользователям и группам пользователей [8].

## Virtual Private Network

VPN (Virtual Private Network) – виртуальная частная сеть – технология организации в сети с передачей пакетов защищенного выделенного канала связи. Технология VPN позволяет решить проблему передачи защищенной информации по общедоступным сетям за счет организации защищенного соединения.

Существует несколько механизмов организации VPN соединений: межсетевой экран, шифрование, использование протокола IPSec, применение на сети специального сервера AAA.

**Firewall** представляет собой барьер, который ограничивает сеть пользователя от остальной публичной сети. Он преграждает доступ извне для пользователей, не имеющих на это право, а разрешает только по заранее установленному списку. Таким образом, Firewall позволяет разграничивать доступ и тем самым создавать виртуальную сеть с другими пользователями и сетями, у которых на вход в общедоступную сеть также стоит подобная защита. Обычно Firewall используется совместно с другими средствами защиты.

**Шифрование** – это еще один способ организации VPN, при котором все участники информационного обмена обладают ключом и программным средством, которое позволяет шифровать передаваемую в виртуальной сети информацию. Для повышения безопасности применяются сложные алгоритмы шифрования, постоянная смена ключей, несколько ступеней шифрования. Однако такой способ создания VPN обычно используется для сетей, где передаются небольшие объемы данных.

Использование протокола **Internet Protocol Security (IPSec)** также позволяет организовывать VPN с многочисленными настройками и большим числом элементов в сети. IPSec – это разновидность протокола IP, предусматривающего процедуру шифрования заголовка пакета и передаваемой информации, а также описывающий политику распределения ключей шифрования между участниками VPN. В зависимости от конкретной реализации данные могут шифроваться между роутером и ПК, роутером и роутером, Firewall и роутером, роутером и сервером. Главное требование для нормального функционирования VPN с помощью IPSec – это поддержка пограничных устройств сети данного протокола.

Использование **AAA сервера**. В этом случае участники VPN перед тем, как получить доступ в сеть должны пройти процедуры аутентификации, авторизации и для него должен быть создан аккаунт. Таким образом, всех участников VPN можно разделить по правам доступа и обеспечить максимально тонкую настройку сети. Вместе с тем этот способ наиболее затратный, т.к. требует установку отдельного устройства и поддержание его работы, однако в крупных сетях это оказывается оправданным.

Таким образом, независимо от способа организации VPN – это эффективный способ создания защищенной телекоммуникационной сети, не требующий инвестирования значительных финансовых средств и предоставляющий все преимущества выделенной независимой сети [9].

## Redundant Array of Independent Disks

RAID - это акроним от Redundant Array of Independent Disks. Дисковые массивы могут обеспечивать избыточное хранение данных, с тем чтобы данные не были потеряны в случае выхода из строя одного из дисков. В зависимости от уровня RAID, проводится или зеркалирование или распределение данных по дискам.

**RAID уровня 0** (data striping). С применение этой технологии, информация разбивается на куски (фиксированные объемы данных, обычно именуемые блоками); и эти куски записываются на диски и считываются с них в параллель. С точки зрения производительности это означает два основных преимущества:

* повышается пропускная способность последовательного ввода/вывода за счет одновременной загрузки нескольких интерфейсов.
* снижается латентность случайного доступа; несколько запросов к различным небольшим сегментам информации могут выполняться одновременно.

Недостаток: уровень RAID 0 предназначен исключительно для повышения производительности, и не обеспечивает избыточности данных. Поэтому любые дисковые сбои потребуют восстановления информации с резервных носителей.

**RAID уровня 1** (disk mirroring). Уровень RAID 1 хорошо подходит для приложений, которые требуют высокой надежности, низкой латентности при чтении, а также если не требуется минимизация стоимости. RAID 1 обеспечивает избыточность хранения информации, но в любом случае следует поддерживать резервную копию данных, т.к. это единственный способ восстановить случайно удаленные файлы или директории.

Технология **RAID уровней 2 и 3** предусматривает параллельную работу всех дисков. Отличие RAID 3 от RAID 2 состоит только в том, что RAID 2 использует для хранения битов четности несколько дисков, тогда как RAID 3 использует только один. Если происходит сбой одного диска с данными, то система может восстановить его содержимое по содержимому остальных дисков с данными и диска с информацией четности.

**RAID 4** исправляет некоторые недостатки технологии RAID 3 за счет использования больших сегментов информации, распределяемых по всем дискам, за исключением диска с информацией четности. При этом для небольших объемов информации используется только диск, на котором находится нужная информация. Это означает, что возможно одновременное выполнение нескольких запросов на чтение. Однако запросы на запись порождают блокировки при записи информации четности. RAID 4 используется крайне редко.

Технология **RAID 5** очень похожа на RAID 4, но устраняет связанные с ней блокировки. Различие состоит в том, что информация четности распределяется по всем дискам массива. В данном случае возможны как одновременные операции чтения, так и записи. Данная технология хорошо подходит для приложений, которые работают с небольшими объемами данных, например, для систем обработки транзакций [10].

## Virtual Local Area Network

Один из самых эффективных способов защитить подразделения, работающие с важной информацией, от риска заражения — разбить корпоративную сеть на несколько автономных подсетей. Сегментация позволяет изолировать отдельные компьютеры или группы компьютеров от других устройств.

Для сегментирования сетей используются виртуальные сети VLAN. Чаще всего технологию VLAN используют для объединения в одну подсеть компьютеров, подключенных к разным физическим маршрутизаторам (например, машины, находящиеся в разных офисах). Однако с точки зрения информационной безопасности у нее тоже много преимуществ. Она не только позволяет обезопасить устройство одной подсети от несанкционированного доступа из другой, но и облегчает управление политиками безопасности, позволяя применять эти политики к целой подсети, а не к отдельным устройствам.

Применение виртуальных локальных сетей не решает всех проблем безопасности, они позволяют минимизировать шансы заражения критических узлов. Сами отделы «зоны риска» это никак не защищает. Поэтому дополнительно нужно:

* Повышать грамотность сотрудников в сфере информационной безопасности и регулярно напоминать им, что стоит с осторожностью относиться к подозрительным письмам.
* Регулярно обновлять программное обеспечение на рабочих станциях, сетевых и других устройствах, чтобы злоумышленники не могли проникнуть в инфраструктуру через известные уязвимости.
* Использовать надежные защитные решения для рабочих станций и серверов, распознающие и обезвреживающие вредоносные программы и ресурсы [11].

## Iptables

Iptables — это утилита для настройки программного межсетевого экрана linux, которая предустанавливается по умолчанию во все сборки Linux, начиная с версии 2.4. Ее настройка производится в командной строке, с помощью правил iptables можно разрешать или блокировать прохождение трафика. Когда происходит попытка установления соединения с текущей машиной, iptables просматривает список правил в списке, чтобы понять, как нужно поступить в этом случае. Если правила нет, то выполняется действие по умолчанию. Существует три типа правил iptables — input, forward и output.

**Input** — Такие цепочки используются для контроля поведения входящих соединений. К примеру, если пользователь попробует подключиться к серверу по SSH, то iptables сравнит его IP-адрес со своим списком, чтобы разрешить или запретить доступ.

**Forward** — Правила этого типа используются для обработки входящих сообщений, конечный пункт назначения которых не является текущим сервером. К примеру, в случае маршрутизатора, к нему подключаются многие пользователи и приложения, но данные не посылаются на сам маршрутизатор, они лишь передаются ему, чтобы он мог перенаправить их адресату.

**Output** — Такие цепочки используются для исходящих соединений. К примеру, если пользователь пытается отправить запрос ping к сайту 1cloud.ru, iptables изучит цепочку правил, чтобы понять, что нужно делать в случае пинга этого сайт, и только потом разрешит или запретит соединение.

По умолчанию все три цепочки разрешают прием трафика. Чаще всего такое поведение более предпочтительно. После настройки поведения межсетевого экрана по умолчанию, можно переходить к созданию правил обработки трафика, чтобы iptables понимал, что делать с конкретным соединением. В iptables есть три основных вида действий с соединениями:

* Accept — разрешить соединение;
* Drop — игнорировать соединение. Действие подходит для случаев, когда нужно сделать так, чтобы источник запроса не узнал о его блокировке.
* Reject — заблокировать соединение и отправить в ответ сообщение об ошибке. Действие подходит для тех случаев, когда владелец сервера хочет дать понять, что соединение заблокировано файрволлом.

При работе с iptables важно помнить, что многие протоколы передачи данных требуют двусторонней коммуникации. Поэтому нужно настраивать правила соответствующим образом [12].

## Simplified Mandatory Access Control Kernel

Smack (полное название: Simplified Mandatory Access Control Kernel) — это модуль безопасности ядра Linux, который защищает данные и взаимодействие процессов от злонамеренных манипуляций с помощью набора настраиваемых правил обязательного контроля доступа (MAC) с простотой в качестве основной цели разработки. Smack предназначен для обеспечения мандатного контроля доступа.

В отличие от AppArmor и Tomoyo Linux, эта система разграничения прав доступа использует подход, основанный на присваивании уникальных меток объектам защищаемым объектам операционной системы. Smack предоставляет полноценный механизм защиты процессов и данных операционной системы.

Для перехвата обращений и контроля доступа на уровне ядра Linux, Smack использует механизм Linux Security Modules (LSM). Возможности системы безопасности покрывают контроль файловых операций и передачи сетевых пакетов на уровне приложений.

Система обеспечения мандатного контроля доступа Smack состоит из трех основных компонент:

* компонент ядра, который реализован как модуль Linux Security Modules;
* скрипт загрузки и вспомогательные утилиты, предназначенные для загрузки базовых настроек, проверки корректности атрибутов Smack для отдельных файлов устройств и т.п.;
* набор исправлений к пакету GNU Core Utilities, благодаря которому некоторые стандартные утилиты (например ls) могут оперировать расширенными атрибутами файлов, используемых системой безопасности.

Правила мандатного контроля доступа, как и положено «упрощенной» системе, отличаются простотой: они записываются в файл /etc/smack/accesses в виде <метка субъекта> <метка объекта> <тип доступа>.

Здесь субъект представляет собой исполняемый файл системы, объект — некоторый ресурс, к которому необходим доступ (например, файл или сетевое соединение). Smack, в основном, придерживается традиционной Linux-модели разделения прав доступа на запись (w), выполнение (x) и чтение (r), дополняя эту модель правом добавления или дозаписи (a).

Право добавления особенно важно для безопасности при работе с лог-файлами: системные процессы могут записывать свои сообщения в лог-файлы, но не должны иметь возможности уничтожать сообщения, записанные ранее. Передача сигнала от объекта к субъекту и передача сетевого пакета от объекта к субъекту рассматриваются как операция записи.

Права доступа субъекта к объекту определяются установленными пользователем правилами. Однако следующие аксиомы являются главенствующими и всегда определяют взаимоотношения субъекта и объекта:

* разрешен любой доступ объекта к субъекту, если они имеют одинаковые метки;
* разрешен тот тип доступа, который явно указан в правилах;
* если объект и субъект имеют различные метки, ни один из них не имеет служебной метки, и правила доступа для таких меток не заданы, то любой доступ объекта к субъекту запрещен.

Для выполнения администраторских функций применительно к системе обеспечения мандатного контроля доступа, в Smack используются понятия возможности (capability) и привилегии (privilege): процесс, обладающий возможностью, может нарушать установленные правила мандатного контроля доступа системы Smack (какие правила — зависит от типа возможности). Соответственно, процесс, обладающий одной или более возможностями, считается привилегированным, остальные процессы — непривилегированные [13].

## StaffCounter

StaffCounter — система контроля рабочего времени и анализа эффективности работы сотрудников. Платформа позволяет не только подсчитывать время, фактически затраченное персоналом на работу, но также анализировать и сравнивать продуктивность персонала.

Программу можно установить на компьютер на базе Windows, MacOS и Linux. Также существуют приложения для мобильных устройств Андроид и iOS. Кроме того, есть возможность использования системы без установки на устройство в виде браузерного приложения.

Функционал системы StaffCounter:

* контроль работы сотрудников;
* отправка руководителю снимков экрана с установленной периодичностью;
* отслеживание посещаемых сайтов и используемых приложений;
* подсчет продуктивности работы;
* составление списка бесполезных интернет-ресурсов;
* установка запрета на подключение к компьютеру USB накопителей, DVD дисков, bluetooth устройств, смартфонов;
* мониторинг поисковых запросов;
* отслеживание перемещений сотрудника;
* ведение журнала активности.

Все отчеты и статистическая информация хранятся на облачном диске. Срок хранения данных составляет 1 месяц [14]. Подробные автоматические отчеты, созданные в любом разрезе данных, позволяют проанализировать статистику в целом по предприятию, найти новые способы оптимизации труда персонала. Есть возможность контролировать офисы и фрилансеров удаленно и получать автоматические уведомления при снижении показателей ниже определенного уровня.

# **Выбор программного обеспечения на серверах и АРМ**

## Операционная система

В качестве ОС для пользователей будет использоваться Manjaro Linux, основанный на ArchLinux. Репозиторий Manjaro хорошо организован и имеет намного меньше нерабочих или устаревших пакетов, что делает его более удобным в использовании. Все обновления тестируются перед тем, как попасть в официальный репозиторий, поэтому шанс того, что система сломается во время обновления, очень мал.

Также на данный момент Manjaro позволяет очень легко выбрать любую из доступных версий ядра, начиная с 3.10, заканчивая последней экспериментальной, что позволяет менять ядро при обнаружении в нем критических уязвимостей. Для этого существует специальный пункт меню в менеджере настроек Manjaro.

На серверах будет установлен CentOS — серверная операционная система с открытым исходным кодом на базе Red Hat Enterprise Linux. В отличие от Red Hat Enterprise Linux, который выпускается с платной поддержкой клиентов, CentOS не является коммерческим решением и полностью бесплатен для конечного пользователя. Отличительными чертами CentOS является ее высокая стабильность и длительные сроки поддержки, которые, по заявлениям разработчиков, могут доходить до 10 лет.

## Средство разработки и отладки программ

Для запуска разработанной ИС для обработки амбулаторных карт пациентов на WEB-сервере поликлиники нужно будет установить Python и Django.

Django — это веб-фреймворк Python высокого уровня, который позволяет быстро разрабатывать безопасные веб-сайты. Он бесплатный и с открытым исходным кодом, имеет активное сообщество, отличную документацию и множество вариантов бесплатной и платной поддержки.

Django по умолчанию обеспечивает защиту от многих уязвимостей, включая SQL-инъекции, межсайтовые сценарии, подделку межсайтовых запросов и кликджекинг [15].

## Браузер

Учитывая ограничения, накладываемые наличием или отсутствием определенных программ в выбранной операционной системе, а также исследованиями в области безопасности пользовательских данных в браузерах [16] было выбрано приложение Brave.

Создатели утверждают, что Brave блокирует трекеры сайтов и удаляет интернет-рекламу, повышает уровень приватности в интернете и уменьшает количество передаваемых о потребителях рекламы данных.

## Пакет офисных программ

Для создания отчетности по проделанной работе сотрудники будут пользоваться пакетом офисных программ Libreoffice, который является ответвлением OpenOffice. Этот офисный пакет предустановлен на множестве дистрибутивов и в своем составе имеет все необходимые инструменты.

Он поддерживает большинство форматов MS Office, включая doc, xlsx и т д. Также поддерживаются многие открытые форматы, например, odf. OpenOffice может преобразовать любой документ в PDF формат. Еще одним преимуществом OpenOffice является то, что он предустановлен в Manjaro Linux.

## Приложение организации домена поликлиники

Для создания контроллера домена на сервере и клиентов домена на АРМ в АИС поликлиники будет установлен пакет программ Samba 4. Устанавливая Samba и базовые сетевые службы (DNS, NTP, Kerberos и т.д.) на один из Linux-дистрибутивов администратор безопасности получает следующую функциональность:

1. Контроллер домена Active Directory
2. DNS-сервер на базе BIND, обеспечивающий безопасную динамическую регистрацию имен.
3. Файловый сервер.
4. Сервер печати.

Использование SAMBA сервера в информационной системе позволяет снизить риски, связанные с использованием импортного программного обеспечения (для госучреждений это актуально с января 2016 года) и снизить совокупную стоимость владения информационной системой.

# **Построение схемы сети. Выбор необходимого оборудования**

Для построения сети для ГАУЗ “Городская поликлиника №17” воспользуемся методом декомпозиции. Декомпозиция цели — это её детализация, наглядное разделение комплексной объемлющей цели на многоуровневую иерархию небольших взаимосвязанных задач. Целью декомпозиции будет являться построение защищенной распределенной локальной вычислительной сети для поликлиники.

На первом уровне декомпозиции нужно выделить основные сущности, поставленные в задаче и указать связи между ними. Как показано на рис. 1, в данной задаче участвуют 6 сущностей: здание руководства, здание бухгалтерии, здание поликлиники, здание архива, здание центрального филиала (на рисунке серверная) и выход в интернет.

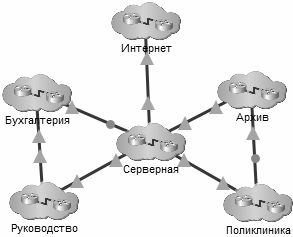


Рисунок 1 - Схема подключения филиалов поликлиники

При таком логическом подключении филиалов образуются две петли. Сделано это для обеспечения отказоустойчивости сети. В таком случае при обрыве сетевого соединения, информация, отправляемая между филиалами, не будет потеряна, а будет перенаправлена по запасному маршруту.

На следующем уровне декомпозиции нужно определить функциональную нагрузку каждого филиала с точки зрения сетевого взаимодействия и в соответствии с ней подобрать соответствующее сетевое оборудование (рис. 2). На схеме:

* «C» – медный S/UTP кабель;
* «F» – оптоволоконный SMF кабель;
* «КТ» – номер контрольной точки.

Начиная от наименее важного к самому важному в этой сети, на одном уровне находятся бухгалтерия, поликлиника и руководство, так как эти филиалы только пользуются инфраструктурой. Для оборудования каждого из этих филиалов необходимы только 1 коммутатор второго уровня с возможностью улучшения его двумя оптоволоконными модулями для передачи трафика на межфилиальном уровне, 1 сервер, совмещающий в себе функции файлового и DHCP сервера, несколько коммутаторов второго уровня для подключения к сети от 10 до 30 АРМ, расположенных на шести этажах здания. Отдельный DHCP сервер для каждого филиала нужен для обеспечения его работоспособности независимо от остальной сети.

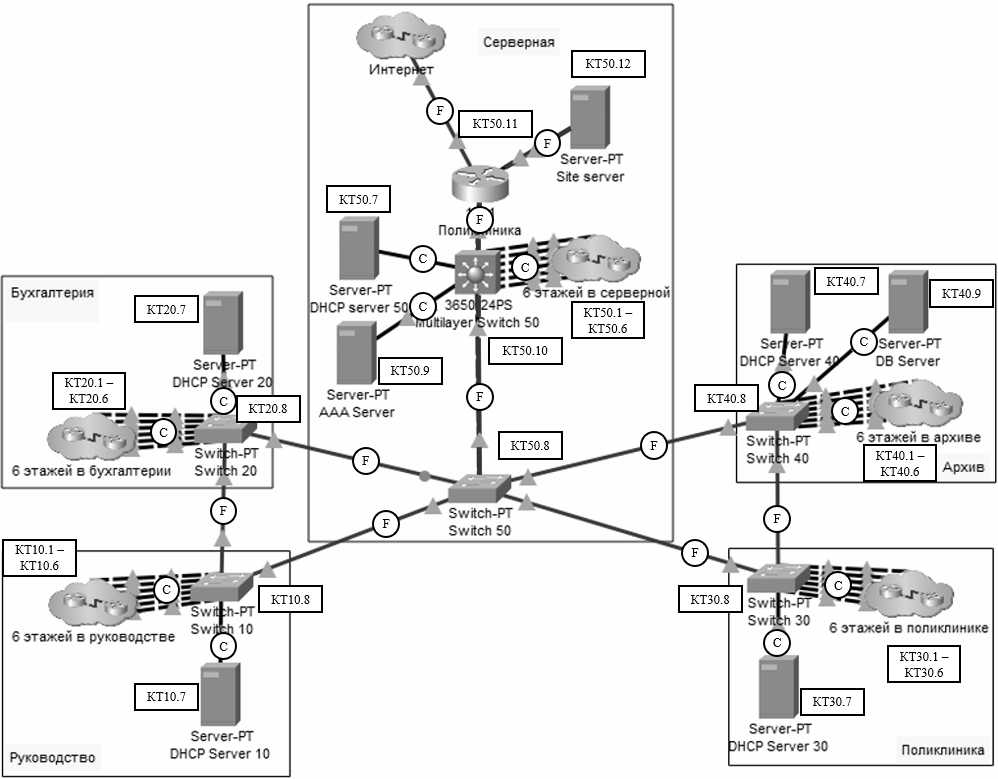


Рисунок 2 - Сетевое оборудование информационной системы

Аналогичное оборудование нужно для архива, но с добавлением отдельного сервера, хранящего в себе базу данных амбулаторных карт. Для предотвращения потери базы данных сервер должен быть обеспечен источником бесперебойного питания и RAID не ниже уровня 2.

Самый важный филиал с точки зрения топологии сети - центральный филиал. В нем так же, как и в предыдущих филиалах нужны коммутатор второго уровня для передачи данных между филиалами, 6 коммутаторов второго уровня для подключения АРМ, 1 файловый/DHCP сервер.

Помимо описанного оборудования этот филиал должен иметь:

* 1 ААА сервер для упрощения настройки безопасности сетевого оборудования и логирования сессий. Он же выполняет функции контроллера домена.
* 1 коммутатор третьего уровня, который исполняет функции VTP сервера и перенаправления трафика между виртуальными локальными сетями (для каждого филиала существует свой VLAN + один VLAN для рабочих, подключенных через Remote Access VPN). Для
* 1 WEB сервер, доступный как из локальной сети, так и из сети интернет
* 1 роутер, предназначенный для обеспечения всей ГАУЗ “Городская поликлиники №17” возможностью выхода в интернет (при желании можно ограничить доступ некоторым отделам).

Продолжая декомпозировать локальную сеть, необходимо отдельно выделить структуру сети с внешней стороны (рисунок 3).

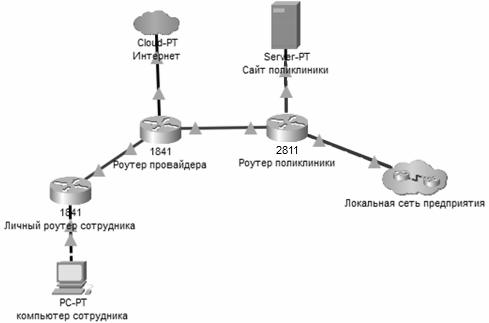


Рисунок 3 - Доступ к ЛВС извне

Выявив необходимые требования к выбору аппаратного обеспечения поликлиники, можно выбрать устройства, которые будут использованы в ИС (Таблица 4).

Таблица 4 - Выбор необходимого оборудования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функциональное описание оборудования (контрольная точка) | Количество | Наименование выбираемого оборудования, характеристики | | | Выбранный пункт, обоснование выбора |
| 1 | 2 | 3 |
| Роутер (КТ50.11) | 1 | Cisco 2811 2 порта Fast Ethernet (10/100BASE-T) , 2 порта USB 1.1, 4 слота HWIC/WIC/VIC/VWIC, 1 слот NM/NME, 2 слота PVDM2, 2 слота AIM 12 213 ₽ | Cisco 2821 2 порта Gigabit Ethernet (10/100/1000BASE-T), 2 порта USB 1.1, 1 слот NM/NME/NME-X, 4 слота HWIC/WIC/VIC/VWIC, 1 слот NM/NME/NME-X, 3 слота PVDM2, 2 слота AIM 12 321 ₽ | MikroTik RB4011IGS+RM 10 x Ethernet 10/100/1000 Мбит/сек, 1 дополнительный слот, 14 378₽ | Выбран пункт 1, т.к. он обладает необходимым количеством HWIC слотов и дешевле других вариантов |
| Сервер сайта (КТ50.12) | 1 | Сервер Dell PowerEdge R240 (up to 4 x 3.5″ HDD/SSD) rack 1U / Intel Xeon E-2224 (3,4 Ghz, cache 8 Mb, 4 cores, 71W) / 2 x 8Gb PC4-21300(2666MHz) DDR4 ECC Unbuffered DIMM / 1 x 1TB 7.2k SATA 6Gbps HDD HS 3.5″ / PERC S140 SATA RAID(0,1,5,10) Broadcom 5720 2x1Gb Integrated card / Power Supply, 250W / no OS / 3Y NBD 74 686 ₽ | Dell PowerEdge R640 (up to 4 x 3.5″ HDD/SSD) rack 1U / 1 x Intel Xeon Bronze 3104 (1.70 GHz, 6 cores, 8.25 MB L3, 2133 MHz, 85W) / PERC H330 RAID(0,1,5,10,50) Controller 12Gb/s / no DVD / iDRAC 9 Enterprise / Broadcom 5720 4x1Gb Network Interface Card Daughter Card / 1 x Power Supply, 750W, Hot-plug / no OS / 3Y Prosupport NBD 119 912 ₽ | Dell PowerEdge R740XD (up to 12 x 3.5″ + 4 x 2.5″ HDD/SSD) rack 2U / 1 x Intel Xeon Bronze 3104 (1.70 GHz, 6 cores, 8.25 MB L3, 2133 MHz, 85W) / PERC H730p RAID(0,1,5,6,10,50,60) / no DVD / iDRAC 9 Enterprise / Broadcom 5720 4x1Gb Network Interface Card Daughter Card / 1 x Power Supply, 750W, Hot-plug / no OS / 3Y Prosupport NBD 324 952 ₽ | Выбран пункт 1, т.к. он дешевле остальных вариантов |
| ААА сервер/ контроллер домена (КТ50,9), Файловый/ DHCP сервер (КТ10.7 – КТ50.7) | 6 | ACER Aspire XC-895, Intel Core i3 10100, DDR4 8ГБ, 1000ГБ, Intel UHD Graphics 630 24 990 ₽ | IRU Office 225, AMD Ryzen 5 3400G, DDR4 8ГБ, 240ГБ(SSD), AMD Radeon RX Vega 11, Free DOS 28 050 ₽ | DELL Vostro 3671, Intel Core i3 9100, DDR4 4ГБ, 1000ГБ, NVIDIA GeForce GT730 - 2048 Мб, DVD-RW, CR, Linux Ubuntu 28 990 ₽ | Выбран пункт 1, т.к. он дешевле остальных вариантов |
| Сервер БД (КТ40.9) | 1 | Dell PowerEdge R240 (up to 4 x 3.5″ HDD/SSD) rack 1U / Intel Xeon E-2224 (3,4 Ghz, cache 8 Mb, 4 cores, 71W) / 1 x 8Gb PC4-21300(2666MHz) DDR4 ECC Unbuffered DIMM / 3 x 1TB 7.2k Near Line SAS 12Gbps HDD HS 3.5″ / PERC S140 SATA RAID(0,1,5,10) / no DVD / iDRAC 9 Express / Broadcom 5720 2x1Gb Integrated card / Power Supply, 250W / no OS / 3Y NBD 107 534₽ | Dell PowerEdge R640 (up to 4 x 3.5″ HDD/SSD) rack 1U / 1 x Intel Xeon Bronze 3104 (1.70 GHz, 6 cores, 8.25 MB L3, 2133 MHz, 85W) / PERC H330 RAID(0,1,5,10,50) Controller 12Gb/s / no DVD / iDRAC 9 Enterprise / Broadcom 5720 4x1Gb Network Interface Card Daughter Card / 1 x Power Supply, 750W, Hot-plug / no OS / 3Y Prosupport NBD 119 912 ₽ | Dell PowerEdge R740XD (up to 12 x 3.5″ + 4 x 2.5″ HDD/SSD) rack 2U / 1 x Intel Xeon Bronze 3104 (1.70 GHz, 6 cores, 8.25 MB L3, 2133 MHz, 85W) / PERC H730p RAID(0,1,5,6,10,50,60) / no DVD / iDRAC 9 Enterprise / Broadcom 5720 4x1Gb Network Interface Card Daughter Card / 1 x Power Supply, 750W, Hot-plug / no OS / 3Y Prosupport NBD 324 952 ₽ | Выбран пункт 1, т.к. он дешевле остальных вариантов |

Продолжение таблицы 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Коммутатор 3 уровня (КТ50.10) | 1 | D-link DGS-3630-28PC, 20 портов 10/100/1000Base-T, 4 комбо‑порта 100/1000Base-T/SFP и 4 порта 10GBase-X SFP+ 164 859 ₽ | Cisco Catalyst WS-C3650-24TS-L 24 порта Ethernet 1 Гбит/с, 4 uplink/стек/SFP 1 Гбит/с 161 307 ₽ | Cisco Catalyst C9300-48T-E 48 портов Ethernet 1 Гбит/с 260 413 ₽ | Выбран пункт 2, т.к. он обладает 24-мя портами, 4-мя встроенным оптоволоконными GigabitEthernet портами и он дешевле |
| Коммутатор 2 уровня с оптоволоконными интерфейсами (КТ10.8 – КТ50.8) | 5 | Cisco Catalyst WS-C4900M 8 фиксированных портов 10G (X2), 2 слота под half slot модули 63 047 ₽ | Cisco WS-C2960-8TC-S 8 портов Ethernet 100 Мбит/с, 1 uplink/стек/SFP 1 Гбит/с 32 290 ₽ | Cisco WS-C2960CX-8PC-L 8 портов Ethernet 1 Гбит/с, 4 uplink/стек/SFP 1 Гбит/с 67 997 ₽ | Выбран пункт 3, поскольку это самый дешевый коммутатор с 4-мя встроенными SFP портами |
| Коммутатор 2 уровня без оптоволоконных интерфейсов  (КТ10-1 – КТ50.6) | 30 | Catalyst Cisco WS-C2960L-8PS-LL 8 портов Ethernet 1 Гбит/с, 2 uplink/стек/SFP 1 Гбит/с 39 836 ₽ | Cisco SG110D-08HP, 8 портов Ethernet 1 Гбит/с 9 357 ₽ | Cisco SG250-08 8 портов Ethernet 1 Гбит/с 7 442 ₽ | Выбран пункт 3, т.к. этот коммутатор дешевле остальных и поддерживает технологию 802.1q |
| АРМ (-) | 118 | MSI Pro 16T 10M-020XRU, 15.6", Intel Celeron 5205U, 4ГБ, 1000ГБ, Intel UHD Graphics 27 980 ₽ | HP 200 G3, 21.5", Intel Pentium Silver J5005, 4ГБ, 128ГБ SSD, Intel UHD Graphics 605, DOS 30 830 ₽ | IRU Office P2121, 21.5", Intel Pentium Gold G5400, 4ГБ, 120ГБ SSD, Intel UHD Graphics 610, Windows 10 Home 35 120 ₽ | Выбран пункт 1, т.к. он дешевле остальных вариантов |

Разработанная ЛВС имеет доступ к своему сайту и ко всем ресурсам индексируемой сети, за исключением запрещенных в StaffCounter сайтов. Внешний пользователь, благодаря настроенным на роутере ACL, напрямую имеет доступ только к сайту поликлиники. Но благодаря Remote Access VPN внешний пользователь может получить доступ к локальной сети предприятия, при этом используя отдельный VLAN. Настройка сетевого оборудования представлена в приложении А.

# **Обзор разработанной ИС для обработки амбулаторных карт**

Разработанная ИС предназначена для внутреннего использования и позволяет врачам добавлять в базу данных новых пациентов, добавлять им диагноз с развернутым комментарием, производить выборку на основе фильтров. Полный листинг программы указан в приложении Б.

В дополнение к возможности использования сайта только из локальной сети поликлиники, добавлена дополнительная ступень защиты, запрещающая просмотр контента страницы неавторизованным пользователям.

Проверка на авторизацию производится на сервере и, если анонимный пользователь пытается получить доступ к ресурсам, на АРМ приходит страница с просьбой авторизоваться, в ином случае возвращаемая страница содержит необходимый пользователю контент.

Авторизованный пользователь может просматривать список всех врачей в поликлинике, с возможностью фильтровать их по квалификационным категориям или по специальности (рис 4).

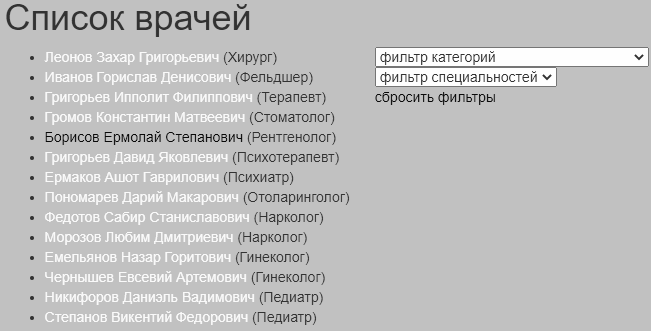


Рисунок 4 - Список всех врачей

Далее, из списка врачей пользователь по ссылке может перейти к подробной информации о враче (рис. 5), в которой указаны его категория и история всех его пациентов по датам.

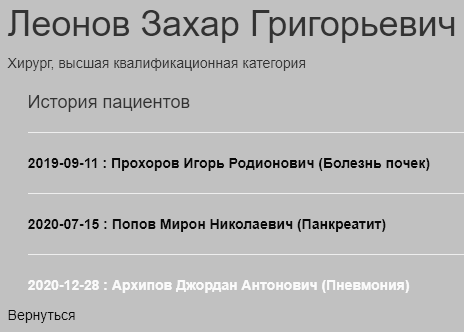


Рисунок 5 - Подробная информация о враче

Также врач может просматривать список всех пациентов, в котором указаны ФИО пациента и его текущее состояние, фильтровать его так же, как и на странице со всеми врачами, а также добавлять нового пациента (рис. 6). Если пациентов в базе данных достаточно много, на страницу они выводятся по двадцать.



Рисунок 6 - Список всех пациентов

Перейдя по ссылке, пользователь попадает в подробное описание пациента (рис. 7), в котором помимо уже известных ФИО и текущего состояния, отображаются его дата рождения, дата смерти (если есть), социальный статус и вся история болезней. Из этой страницы врач может попасть на форму редактирования информации о пациенте.

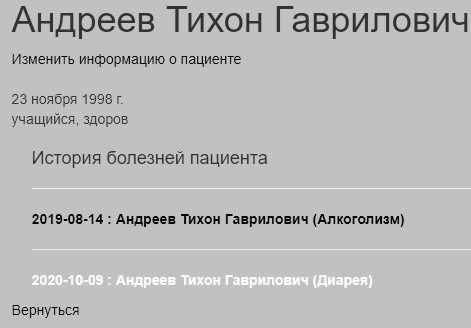


Рисунок 7 - Подробная информация о пациенте

На странице с подробным описанием болезни пациента (рис. 8) отображаются даты начала и конца лечения, имена пациента и лечащего врача, диагноз, симптомы, были ли назначены пациенту амбулаторное или диспансерное лечение. Врач с этой страницы может быстро попасть на форму редактирования болезни (уточнить диагноз, симптомы, поставить дату выписки).

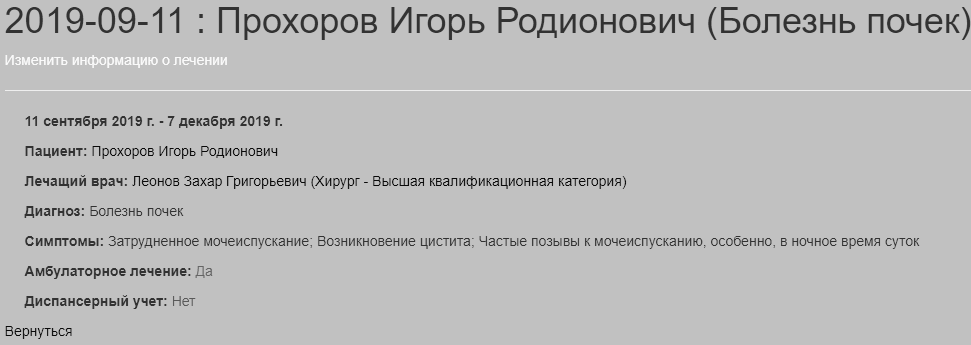


Рисунок 8 - Подробная информация о заболевании

Форма добавления нового пациента или редактирования информации об уже имеющемся представлена на рисунке 9.

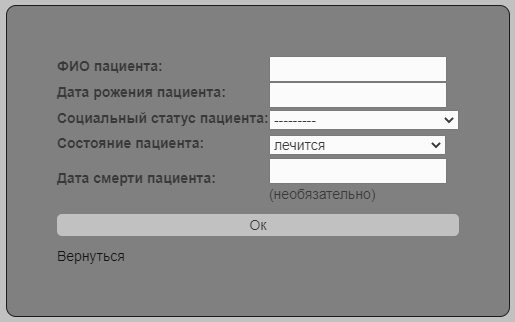


Рисунок 9 - Форма добавления пациента в базу данных

Форма добавления и редактирования заболеваний выглядит идентично. Новые диагнозы, социальные статусы и новых врачей в базе данных регистрирует администратор базы данных.

# **Заключение**

Разработка информационной системы, выполняется для вполне определенного предприятия. Особенности предметной деятельности предприятия, оказывают влияние на структуру информационной системы, но в тоже время структуры разных предприятий в целом схожи между собой. Каждая организация, независимо от рода её деятельности, состоит из ряда подразделений, непосредственно осуществляющих тот или иной вид деятельности компании и эта ситуация справедлива практически для всех организаций, каким бы видом деятельности они ни занимались.

Внедрение современных информационных технологий в инфраструктуру поликлиники позволяет избавиться от времени на передачу амбулаторных карт пациентов между врачами, исключить возможность появления ошибок в отчетах, добавить возможность оперативно посмотреть все оказанные пациенту услуги за время лечения в клинике. Это позволяет лечащему врачу ознакомиться с историей болезни пациента, посмотреть все назначения и заключения, которые были написаны другими врачами, а также сравнить результаты текущих и предыдущих анализов пациента.

Разумеется, для раскрытия всех потенциальных возможностей, которые несет в себе использование компьютеров, необходимо применять в работе на них комплекс программных и аппаратных средств, максимально соответствующий поставленным задачам. Поэтому в разработанной информационной системе применяются производительное, надежное и безопасное оборудование CISCO, позволяющее увеличить жизненный цикл ИС и в дальнейшем подключиться к ГИС, самые стабильные и безопасные репозитории Linux, браузеры, позволяющие не беспокоиться о передачи секретных данных третьим лицам.

Таким образом, система, на момент ее запуска и в течение всего времени эксплуатации, будет обеспечивать: требуемую функциональность и степень адаптации к изменяющимся условиям; требуемую пропускную способность; требуемое время реакции системы на запрос; безотказную работу системы в требуемом режиме (готовность и доступность системы для обработки запросов пользователей); простоту эксплуатации и поддержки системы; необходимую безопасность.

# **Список использованных источников**

1. Принцип работы протоколов STP/RSTP, функции защиты корня (root guard) [Электронный ресурс. – Электрон. текстовые дан.. – Новосибирск: НСК Коммуникации Сибири, 2009-2021. – Режим доступа: https://www.nsc-com.com/support/more/principle-of-protocols-stp-rstp, свободный.
2. VTP (VLAN Trunking Protocol) [Электронный ресурс. – Электрон. текст. дан. – Национальная библиотека им. Н. Э. Баумана, 01.02.2015. – Режим доступа: https://ru.bmstu.wiki/VTP\_(VLAN\_Trunking\_Protocol), свободный.
3. Аутентификация в Cisco IOS [Электронный ресурс] / Positive Technologies. – Электрон. текст. дан. – Хабр, 05.09.2013. – Режим доступа: https://habr.com/ru/company/pt/blog/192668/, свободный.
4. SSH — сетевой протокол для управления серверами [Электронный ресурс]. – Электрон. текст. дан. – Веб Креатор, 2004-2020. – Режим доступа: https://web-creator.ru/articles/ssh, свободный.
5. Разница между HTTP и HTTPS [Электронный ресурс]. – Электрон. текст. дан. – HOSTiQ, 2013-2021. – Режим доступа: https://hostiq.ua/wiki/http-https/, свободный.
6. Владимир Хазов. Аутентификация, авторизация и учет (AAA) [Электронный ресурс]: RADIUS или TACACS+ / Владимир Хазов. –Электрон. текст. дан. – Санкт-Петербург: VAS EXPERTS, 02.06.2017. – Режим доступа: https://vasexperts.ru/blog/autentifikaciya-avtorizaciya-i-uchet-aaa-radius-ili-tacacs/, свободный.
7. Николай Милинский. ACL [Электронный ресурс]: Списки контроля доступа в Cisco IOS / Николай Милинский. – Электрон. текст. дан. – Хабр, 15.06.2011. – Режим доступа: https://habr.com/ru/post/121806/, свободный.
8. SSH [Электронный ресурс]. – Электрон. текст. дан. – Москва: Atlex.ru, 2006-20021. – Режим доступа: https://www.atlex.ru/baza-znanij/tematicheskij-glossarij/ssh/, свободный.
9. Игорь Летов. VPN (Virtual Private Network) [Электронный ресурс]: Игорь Летов. – Электрон. текст. дан. –celnet.ru, 2010-2019. – Режим доступа: http://celnet.ru/vpn.php, свободный.
10. Краткий обзор технологии RAID [Электронный ресурс].– Электрон. текст. дан. – Москва: Лаборатория Параллельных информационных технологий НИВЦ МГУ, 1998-2020. – Режим доступа: https://parallel.ru/computers/reviews/raid-technology.html, свободный.
11. Леонид Грустный. VLAN как дополнительный уровень защиты [Электронный ресурс]: Леонид Грустный. –Электрон. текст. дан. – Москва: Kaspersky Daily, 16.08.2019. – Режим доступа: https://www.kaspersky.ru/blog/vlan-security/23402/, свободный.
12. Настройка Linux-файрвола iptables [Электронный ресурс]: Руководство для начинающих. – Электрон. текст. дан. – Санкт-Петербург: 1cloud.ru, 07.12.2018. – Режим доступа: https://1cloud.ru/help/linux/nastrojka\_linus-firewall\_iptables, свободный.
13. Евгений Ивашко. Часть 6 [Электронный ресурс]: Система мандатного контроля доступа Smack / Евгений Ивашко. – Электрон. текст. дан. – Санкт-Петербург: IBM Developer, 26.10.2010. – Режим доступа: https://www.ibm.com/developerworks/ru/library/l-apparmor-6/index.html, свободный.
14. StaffCounter [Электронный ресурс]. – Электрон. текст. дан. – Санкт-Петербург: GDE SAAS, 2018-2021. – Режим доступа: https://gde-saas.ru/applications/staffcounter/, свободный.
15. Django introduction [Электронный ресурс].– Электрон. текст. дан. – MDN Web Docs, 23.12.2020. – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction, свободный.
16. Douglas J. Leith. Web Browser Privacy: What Do Browsers SayWhen They Phone Home? / Douglas J. Leith // SCSS Technical Report. – 24.02.2020.

# **Приложение А**

(обязательное)

**Настройка Switch L2**

Current configuration : 862 bytes

!

version 12.1

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname Switch50

!

enable secret 5 $1$mERr$4bW9.cjKRYCH.vMYcINsT/

!

username admin secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0

!

spanning-tree mode pvst

spanning-tree extend system-id

!

interface FastEthernet1/1

switchport mode trunk

!

interface FastEthernet2/1

switchport mode trunk

!

interface FastEthernet3/1

switchport mode trunk

!

interface FastEthernet4/1

switchport mode trunk

!

interface GigabitEthernet5/1

switchport mode trunk

!

line con 0

login local

!

line vty 0 4

login local

line vty 5 15

login local

!

VTP Version : 2

Configuration Revision : 153

Maximum VLANs supported locally : 255

Number of existing VLANs : 11

VTP Operating Mode : Client

VTP Domain Name : vtp.domain

VTP Pruning Mode : Disabled

VTP V2 Mode : Enabled

VTP Traps Generation : Disabled

MD5 digest : 0x87 0x2D 0x10 0x8F 0x38 0xBE 0x08 0x87

Configuration last modified by 0.0.0.0 at 3-1-93 00:00:00

!

end

**Настройка Switch L3**

Current configuration : 3131 bytes

!

version 16.3.2

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname SwitchL3

!

enable secret 5 $1$mERr$GQRRaWhoZYwQxARUhN0G3.

!

aaa new-model

!

aaa authentication login default group tacacs+ local

!

aaa authorization exec default group tacacs+ local

!

aaa accounting exec default start-stop group tacacs+

!

no ip cef

ip routing

!

no ipv6 cef

!

username admin privilege 15 secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0

!

ip ssh version 2

ip domain-name mswitch

!

spanning-tree mode rapid-pvst

!

interface GigabitEthernet1/0/3

switchport access vlan 50

switchport mode access

switchport nonegotiate

!

interface GigabitEthernet1/0/5

switchport access vlan 50

switchport mode access

switchport nonegotiate

!

interface GigabitEthernet1/0/6

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/0/7

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/0/21

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/0/22

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/0/23

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/0/24

switchport trunk encapsulation dot1q

switchport mode trunk

switchport nonegotiate

!

interface GigabitEthernet1/1/1

switchport access vlan 100

switchport mode access

switchport nonegotiate

!

interface GigabitEthernet1/1/2

switchport trunk encapsulation dot1q

switchport mode trunk

!

interface Vlan10

mac-address 0040.0b1b.d001

ip address 192.168.10.254 255.255.255.0

!

interface Vlan20

mac-address 0040.0b1b.d002

ip address 192.168.20.254 255.255.255.0

!

interface Vlan30

mac-address 0040.0b1b.d003

ip address 192.168.30.254 255.255.255.0

!

interface Vlan40

mac-address 0040.0b1b.d004

ip address 192.168.40.254 255.255.255.0

!

interface Vlan50

mac-address 0040.0b1b.d005

ip address 192.168.50.254 255.255.255.0

!

interface Vlan100

mac-address 0040.0b1b.d006

ip address 192.168.100.254 255.255.255.0

!

router rip

network 172.16.0.0

network 192.168.10.0

network 192.168.20.0

network 192.168.30.0

network 192.168.40.0

network 192.168.50.0

network 192.168.100.0

!

ip classless

!

ip flow-export version 9

!

tacacs-server host 192.168.50.253

tacacs-server key cisco123

!

line con 0

!

line aux 0

!

line vty 0 4

transport input ssh

line vty 5 15

transport input ssh

!

VTP Version capable : 1 to 2

VTP version running : 2

VTP Domain Name : vtp.domain

VTP Pruning Mode : Disabled

VTP Traps Generation : Disabled

Device ID : 0001.6403.2200

Configuration last modified by 0.0.0.0 at 3-1-93 00:00:00

Local updater ID is 192.168.10.254 on interface Vl10 (lowest numbered VLAN interface found)

Feature VLAN :

--------------

VTP Operating Mode : Server

Maximum VLANs supported locally : 1005

Number of existing VLANs : 11

Configuration Revision : 153

MD5 digest : 0x87 0x2D 0x10 0x8F 0x38 0xBE 0x08 0x87 0xEB 0x83 0x32 0xD9 0x4F 0x46 0x8D 0x88

!

end

**Настройка Router**

Current configuration : 2421 bytes

!

version 12.4

no service timestamps log datetime msec

no service timestamps debug datetime msec

no service password-encryption

!

hostname pRouter100

!

enable secret 5 $1$mERr$mkwHtFiTZkAMGp.2X.qK8.

!

aaa new-model

!

aaa authentication login VPN local

aaa authentication login default local none

!

aaa authorization exec default group tacacs+ local

aaa authorization network VPN local

!

aaa accounting exec default start-stop group tacacs+

!

no ip cef

no ipv6 cef

!

username Leonov.Z password 0 Leonov.Z

username admin secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0

!

crypto isakmp policy 10

encr aes 256

authentication pre-share

group 2

!

crypto isakmp client configuration group polyclinicVPN

key polyclinicSecret

pool VPN-CLIENTS

netmask 255.255.255.0

!

crypto ipsec transform-set key1 esp-3des esp-sha-hmac

!

crypto dynamic-map key2 10

set transform-set key1

reverse-route

!

crypto map key2 client authentication list VPN

crypto map key2 isakmp authorization list VPN

crypto map key2 client configuration address respond

crypto map key2 10 ipsec-isakmp dynamic key2

!

ip ssh version 2

ip domain-name pRouter

!

spanning-tree mode pvst

!

interface FastEthernet0/0

no ip address

duplex auto

speed auto

shutdown

!

interface FastEthernet0/1

no ip address

duplex auto

speed auto

shutdown

!

interface GigabitEthernet0/1/0

ip address 192.168.100.251 255.255.255.0

ip access-group FOR-NAT in

ip access-group FOR-NAT out

ip nat inside

!

interface GigabitEthernet0/2/0

ip address 172.16.20.21 255.255.0.0

!

interface GigabitEthernet0/3/0

ip address 172.15.10.10 255.255.0.0

ip nat outside

crypto map key2

!

interface Vlan1

no ip address

shutdown

!

router rip

network 172.15.0.0

network 172.16.0.0

network 192.168.0.0

network 192.168.100.0

!

ip local pool VPN-CLIENTS 192.168.100.1 192.168.100.50

ip nat inside source list FOR-NAT interface GigabitEthernet0/3/0 overload

ip classless

!

ip flow-export version 9

!

ip access-list standard FOR-NAT

permit 192.168.10.0 0.0.0.255

permit 192.168.20.0 0.0.0.255

permit 192.168.30.0 0.0.0.255

permit 192.168.40.0 0.0.0.255

permit 192.168.50.0 0.0.0.255

permit 192.168.100.0 0.0.0.255

permit 172.16.20.0 0.0.0.255

deny any

!

tacacs-server host 192.168.50.253

tacacs-server key cisco123

!

line con 0

!

line aux 0

!

line vty 0 4

transport input ssh

line vty 5 15

transport input ssh

!

end

# **Приложение Б**

(обязательное)

**./polyclinic/urls.py**

**from** django.contrib **import** admin

**from** django.urls **import** path

**from** django.urls **import** include

**from** django.views.generic **import** RedirectView

**from** django.conf **import** settings

**from** django.conf.urls.static **import** static

urlpatterns = [

path('admin/', admin.site.urls),

path('cardfile/', include('cardfile.urls')),

path('', RedirectView.as\_view(url='/cardfile/', permanent=True)),

path('accounts/', include('django.contrib.auth.urls')),

] + static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)

**./cardfile/models.py**

**from** django.db **import** models

**from** django.urls **import** reverse

**from** datetime **import** date

**from** django.contrib.auth.models **import** User

**import** uuid

**class** Doctors\_specialty(models.Model):

""" Модель, представляющая специальность врача (терапевт, невропатолог и т.п.) """

specialty = models.CharField(

verbose\_name='Специальность',

max\_length=200,

help\_text="Введите специальность врача (терапевт, невропатолог и т.п.)")

**class** Meta:

verbose\_name='Специальность врача'

verbose\_name\_plural = 'Специальности врачей'

**def** \_\_str\_\_(self):

**return** self.specialty

**class** Social\_status(models.Model):

""" Модель, представляющая социальный статус пациента (учащийся, работающий, временно неработающий, инвалид, пенсионер) """

status = models.CharField(

verbose\_name='Социальный статус:',

max\_length=200,

help\_text="Введите социальный статус пациента (учащийся, работающий, временно неработающий, инвалид, пенсионер)")

**class** Meta:

verbose\_name='Социальный статус пациента'

verbose\_name\_plural = 'Социальные статусы пациентов'

**def** \_\_str\_\_(self):

**return** self.status

**class** Diagnosis(models.Model):

""" Модель, представляющая диагноз (кариес, астма, пневмония и т.д.) """

diagnosis = models.CharField(

verbose\_name='Диагноз',

max\_length=200,

help\_text="Введите диагноз (кариес, астма, пневмония и т.д.)")

**class** Meta:

verbose\_name='Диагноз'

verbose\_name\_plural = 'Диагнозы'

**def** \_\_str\_\_(self):

**return** self.diagnosis

**class** Pacient(models.Model):

""" Модель, представляющая пациента """

name = models.CharField(

verbose\_name='ФИО пациента',

max\_length=200)

date\_of\_birth = models.DateField(

verbose\_name='Дата рождения пациента',

null=True,

blank=True)

soc\_status = models.ForeignKey(

'Social\_status',

verbose\_name='Социальный статус пациента',

on\_delete=models.SET\_NULL,

null=True)

PACIENT\_CONDITION = (

('t', 'лечится'),

('c', 'вылечился'),

('h', 'направлен в стационар'),

('d', 'умер'),

)

condition = models.CharField(

verbose\_name='Состояние пациента',

max\_length=1,

choices=PACIENT\_CONDITION,

blank=True, default='t')

date\_of\_death = models.DateField(

verbose\_name='Дата смерти пациента',

null=True,

blank=True,

help\_text='(необязательно)')

**class** Meta:

ordering = ['name']

verbose\_name='Пациент'

verbose\_name\_plural = 'Пациенты'

**def** get\_absolute\_url(self):

**return** reverse('pacient-detail', args=[str(self.id)])

**def** \_\_str\_\_(self):

**return** self.name

**class** Doctor(models.Model):

""" Модель, представляющая врача """

name = models.CharField(

verbose\_name='ФИО врача',

max\_length=200)

user = models.ForeignKey(

User,

verbose\_name='Username врача',

on\_delete=models.SET\_NULL,

null=True)

QUALIFICATION\_LEVEL = (

('2', 'Вторая квалификационная категория')***,***

('1', 'Первая квалификационная категория')***,***

('0', 'Высшая квалификационная категория'),

)

qualification = models.CharField(

verbose\_name='Квалификация врача',

max\_length=1,

choices=QUALIFICATION\_LEVEL,

blank=True,

default='2')

doc\_specialty = models.ForeignKey(

'Doctors\_specialty',

verbose\_name='Специальность врача',

on\_delete=models.SET\_NULL,

null=True)

**class** Meta:

ordering = ['doc\_specialty']

verbose\_name='Врач'

verbose\_name\_plural = 'Врачи'

permissions = (("can\_add\_doctor", "Can add doctor"),)

**def** get\_absolute\_url(self):

**return** reverse('doctor-detail', args=[str(self.id)])

**def** \_\_str\_\_(self):

**return** '{0} ({1} - {2})'.format(self.name, self.doc\_specialty, self.QUALIFICATION\_LEVEL[2-int(self.qualification)][1])

**class** Treatment\_history(models.Model):

""" Модель, представляющая историю лечений пациентов """

pac\_name = models.ForeignKey(

'Pacient',

verbose\_name='Пациент',

on\_delete=models.SET\_NULL,

null=True)

doc\_name = models.ForeignKey(

'Doctor',

verbose\_name='Лечащий врач',

on\_delete=models.SET\_NULL,

null=True)

diagnosis = models.ForeignKey(

'Diagnosis',

verbose\_name='Диагноз',

on\_delete=models.SET\_NULL,

null=True)

description = models.TextField(

verbose\_name='Симптомы',

max\_length=1000)

ambulatory = models.BooleanField(verbose\_name='Амбулаторное лечение')

dispensary = models.BooleanField(verbose\_name='Диспансерный учет')

start\_date\_of\_treatment = models.DateField(

verbose\_name='Дата начала лечения',

null=True,

blank=True)

end\_date\_of\_treatment = models.DateField(

verbose\_name='Дата окончания лечения',

null=True,

blank=True)

**class** Meta:

ordering = ['start\_date\_of\_treatment']

verbose\_name='История лечения'

verbose\_name\_plural = 'Истории лечений'

**def** get\_absolute\_url(self):

**return** reverse('treatment\_history\_detail', args=[str(self.id)])

**def** \_\_str\_\_(self):

**return** '{0} : {1} ({2})'.format(self.start\_date\_of\_treatment, self.pac\_name.name, self.diagnosis)ATIC\_URL, document\_root=settings.STATIC\_ROOT)

**./cardfile/urls.py**

**from** django.urls **import** path

**from** . **import** views

**from** django.conf.urls **import** url

urlpatterns = [

url(r'^$', views.index, name='index'),

url(r'^doctors/(?P<stub>[-\w]+)$', views.DoctorListView.as\_view(), name='doctors'),

url(r'^doctors/detail/(?P<pk>\d+)$', views.DoctorDetailView.as\_view(), name='doctor-detail'),

url(r'^pacients/(?P<stub>[-\w]+)$', views.PacientListView.as\_view(), name='pacients'),

url(r'^pacients/detail/(?P<pk>\d+)$', views.PacientDetailView.as\_view(), name='pacient-detail'),

url(r'^pacients/detail/(?P<pk>\d+)/update/$', views.PacientUpdate.as\_view(), name='pacient\_update'),

url(r'^pacients/create/$', views.PacientCreate.as\_view(), name='pacient\_create'),

url(r'^treatmenthistory/(?P<stub>[-\w]+)$', views.TreatmentHistoryListView.as\_view(), name='treatment\_history'),

url(r'^treatmenthistory/detail/(?P<pk>\d+)$', views.HistoryDetailView, name='treatment\_history\_detail'),

url(r'^treatmenthistory/detail/(?P<pk>\d+)/update/$', views.TreatmentHistoryUpdate.as\_view(), name='treatment\_history\_update'),

url(r'^treatmenthistory/create/$', views.TreatmentHistoryCreate.as\_view(), name='treatment\_create'),

]

**./cardfile/admin.py**

**from** django.contrib **import** admin

**from** .models **import** Doctors\_specialty, Social\_status, Diagnosis, Pacient, Doctor, Treatment\_history

**class** DoctorInline(admin.TabularInline):

model = Doctor

**class** TreatmentHistoryInline(admin.TabularInline):

model = Treatment\_history

**class** PacientInline(admin.TabularInline):

model = Pacient

@admin.register(Doctor)

**class** DoctorAdmin(admin.ModelAdmin):

list\_display = (

'name',

'user',

'qualification',

'doc\_specialty'

)

fields = [

'name',

'user',

(

'doc\_specialty',

'qualification'

)

]

@admin.register(Pacient)

**class** PacientAdmin(admin.ModelAdmin):

list\_display = (

'name',

'date\_of\_birth',

'soc\_status',

'condition',

'date\_of\_death'

)

fields = [

'name',

'soc\_status',

'condition',

(

'date\_of\_birth',

'date\_of\_death'

)

]

list\_filter = (

'soc\_status',

'condition'

)

@admin.register(Treatment\_history)

**class** TreatmentAdmin(admin.ModelAdmin):

list\_display = (

'pac\_name',

'doc\_name',

'diagnosis',

'ambulatory',

'dispensary',

'start\_date\_of\_treatment',

'end\_date\_of\_treatment'

)

list\_filter = (

'diagnosis',

'ambulatory',

'dispensary'

)

@admin.register(Doctors\_specialty)

**class** DoctorsSpecialtyAdmin(admin.ModelAdmin):

inlines = [DoctorInline]

@admin.register(Diagnosis)

**class** DiagnosisAdmin(admin.ModelAdmin):

inlines = [TreatmentHistoryInline]

@admin.register(Social\_status)

**class** SocialStatusAdmin(admin.ModelAdmin):

inlines = [PacientInline]

**./cardfile/vievs.py**

**from** django.shortcuts **import** render

**from** .models **import** Doctors\_specialty, Social\_status, Diagnosis, Pacient, Doctor, Treatment\_history

**from** django.views **import** generic

**from** django.contrib.auth.mixins **import** LoginRequiredMixin

**from** django.urls **import** reverse\_lazy

**from** django.views.generic.edit **import** CreateView, UpdateView, DeleteView

**import** datetime

**def** index(request):

**return** render(

request,

'index.html',

context={}

)

**class** DoctorListView(generic.ListView):

model = Doctor

paginate\_by = 20

**def** get\_queryset(self, queryset=None):

**try**:

**if** self.kwargs['stub'] == 'all':

**return** Doctor.objects.all()

**elif** self.kwargs['stub'][0] == 'k':

**return** Doctor.objects.filter(qualification\_\_exact=self.kwargs['stub'][1]).order\_by('name')

**elif** self.kwargs['stub'][0] == 's':

**return** Doctor.objects.filter(doc\_specialty\_\_exact=self.kwargs['stub'][1:]).order\_by('name')

**except** Doctor.DoesNotExist:

**raise** Http404("Такого фильтра не существует")

**class** DoctorDetailView(generic.DetailView):

model = Doctor

**def** doc\_detail\_view(self, request, pk):

**try**:

doc\_id=Doctor.objects.get(pk=pk)

**except** Doctor.DoesNotExist:

**raise** Http404("Doctor does not exist")

**return** render(

request,

'cardfile/doctor\_detail.html',

context={'doc':doc\_id,}

)

**class** PacientListView(generic.ListView):

model = Pacient

paginate\_by = 20

**def** get\_queryset(self, queryset=None):

**try**:

**if** self.kwargs['stub'] == 'all':

**return** Pacient.objects.all().order\_by('name')

**else**:

**return** Pacient.objects.filter(condition\_\_exact=self.kwargs['stub']).order\_by('name')

**except** Pacient.DoesNotExist:

**raise** Http404("Такого фильтра не существует")

**class** PacientDetailView(generic.DetailView):

model = Pacient

**def** pac\_detail\_view(self, request, pk):

**try**:

pac\_id=Pacient.objects.get(pk=pk)

**except** Pacient.DoesNotExist:

**raise** Http404("Pacient does not exist")

**return** render(

request,

'cardfile/pacient\_detail.html',

context={'pac':pac\_id,}

)

**class** TreatmentHistoryListView(generic.ListView):

model = Treatment\_history

paginate\_by = 20

**def** get\_queryset(self, queryset=None):

**try**:

**if** self.kwargs['stub'] == 'all':

**return** Treatment\_history.objects.all()

**elif** self.kwargs['stub'] == 'my':

**try**:

**return** Treatment\_history.objects.filter(doc\_name=Doctor.objects.filter(user=self.request.user)[0].id)

**except**:

**return** Treatment\_history.objects.all()

**else**:

**return** Treatment\_history.objects.filter(diagnosis\_\_exact=self.kwargs['stub'])

**except** Treatment\_history.DoesNotExist:

**raise** Http404("Такого фильтра не существует")

**def** HistoryDetailView(request, pk):

**try**:

hist\_id=Treatment\_history.objects.get(pk=pk)

**except** Treatment\_history.DoesNotExist:

**raise** Http404("Записи не существует")

**return** render(

request,

'cardfile/treatment\_history\_detail.html',

context={'hist': hist\_id ,}

)

**class** PacientCreate(CreateView):

model = Pacient

fields = '\_\_all\_\_'

**class** PacientUpdate(UpdateView):

model = Pacient

fields = '\_\_all\_\_'

**class** TreatmentHistoryCreate(CreateView):

model = Treatment\_history

fields = '\_\_all\_\_'

initial={'start\_date\_of\_treatment': datetime.date.today(),}

**class** TreatmentHistoryUpdate(UpdateView):

model = Treatment\_history

fields = '\_\_all\_\_'

**./cardfile/templates/base\_generic.html**

<!DOCTYPE html>

<**html** lang="en">

<**head**>

{% load static %}

{% block title %}<**title**>Онлайн картотека</**title**>{% endblock %}

<**meta** charset="utf-8">

<**meta** name="viewport" content="width=device-width, initial-scale=1">

<**link** rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<**link** rel="shortcut icon" href="{% static 'krest.png' %}" type="image/png">

<**link** rel="stylesheet" href="{% static 'css/styles.css' %}">

</**head**>

<**body**>

<**div** class="container-fluid">

<**div** class="header">

{% if user.is\_authenticated %}

User: {{ user.get\_username }}

(<**a** href="{% url 'logout'%}?next={% url 'login'%}">Logout</**a**>)

{% else %}

<**li**><**a** href="{% url 'login'%}?next=/">Login</**a**></**li**>

{% endif %}

</**div**>

<**div** class="row">

<**div** class="col-sm-10">

{% block content %}{% endblock %}

{% block pagination %}

{% if is\_paginated %}

{% if user.is\_authenticated %}

<**div** class="pagination">

<**span** class="page-links">

{% if page\_obj.has\_previous %}

<**a** href="{{ request.path }}?page={{ page\_obj.previous\_page\_number }}">←</**a**>

{% endif %}

<**span** class="page-current">

страница {{ page\_obj.number }} из {{ page\_obj.paginator.num\_pages }}

</**span**>

{% if page\_obj.has\_next %}

<**a** href="{{ request.path }}?page={{ page\_obj.next\_page\_number }}">→</**a**>

{% endif %}

</**span**>

</**div**>

{% endif %}

{% endif %}

{% endblock %}

</**div**>

<**div** class="col-sm-2">

{% block sidebar %}

{% if user.is\_authenticated %}

<**ul** class="sidebar-nav">

<**center**><**a** href="{% url 'index' %}"><**img** src="{% static 'krest.png' %}" width="150" height="150"></**a**></**center**><**br**>

<**li**><**a** href="{% url 'index' %}">Домашняя страница</**a**></**li**>

<**li**><**a** href="{% url 'doctors' 'all' %}">Все врачи</**a**></**li**>

<**li**><**a** href="{% url 'pacients' 'all' %}">Все пациенты</**a**></**li**>

<**li**><**a** href="{% url 'treatment\_history' 'all'%}">Вся история лечения</**a**></**li**>

<**li**><**a** href="{% url 'treatment\_history' 'my'%}">Мои пациенты</**a**></**li**>

<**hr**>

<**li**><**a** href="{% url 'treatment\_create'%}">Новый случай</**a**></**li**>

<**li**><**a** href="{% url 'pacient\_create'%}">Новый пациент</**a**></**li**>

<**hr**>

{% if perms.cardfile.can\_add\_doctor %}

<**li**><**a** href="/admin">Админка</**a**></**li**>

{% endif %}

</**ul**>

{% endif %}

{% endblock %}

</**div**>

</**div**>

</**div**>

</**body**>

</**html**>

**./cardfile/templates/index.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if not user.is\_authenticated %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% else %}

<**h1**>Электронная картотека</**h1**>

<**strong**>Елена Петровская, директор Медицинского информационно-аналитического центра Челябинской области:</**strong**> «Внедрение этой системы до сих пор идет, это непрерывный процесс. Осуществляется он неравномерно: где-то быстро, где-то чуть медленнее. зависит это, конечно, от материально-технической базы и персонала, который в этой системе работает».<**br**><**br**>

Чтобы пользоваться системой, медицинские работники прошли обучение, но с момента ее внедрения прошло уже 6 лет: за это время менялись кадры, в системе появились новые возможности, поэтому процесс обучения идет непрерывно. Он может проходить как дистанционно, так и на базе Медицинского информационно-аналитического центра в Магнитогорске.<**br**><**br**>

<**strong**>Елена Петровская, директор Медицинского информационно-аналитического центра Челябинской области:</**strong**> «На самом деле это очень большая и мощная медицинская информационная система. Она дает возможность вести электронные медицинские карты и истории болезни». <**br**><**br**>

<**strong**>Елена Петровская, директор Медицинского информационно-аналитического центра Челябинской области:</**strong**> «Например, если человек лечился в Магнитогорск, а потом переехал в Челябинск, то врачи могут посмотреть, с чем он обращался в лечебное учреждение и какое медикаментозное лечение он получал. иногда пациенты просто забывают некоторые детали. даже если спросить, сколько лет человек страдает сахарным диабетом, многие затрудняются ответить».<**br**>

<**strong**></**strong**>

<**strong**></**strong**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/doctor\_detail.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>{{ doctor.name }}</**h1**>

{{doctor.doc\_specialty}},

{% if doctor.qualification == '0' %}

высшая квалификационная категория

{% elif doctor.qualification == '1' %}

первая квалификационная категория

{% elif doctor.qualification == '2' %}

вторая квалификационная категория

{% endif %}

<**div** style="**margin-left**:20px;**margin-top**:20px">

<**h4**>История пациентов</**h4**>

{% for copy in doctor.treatment\_history\_set.all %}

<**hr**>

<**p**><**strong**><**a** href="{% url 'treatment\_history\_detail' copy.pk %}">{{copy}}</**a**></**strong**></**p**>

{% endfor %}

</**div**>

<**a** href="{% url 'doctors' 'all' %}">Вернуться</**a**>

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next=/">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/doctor\_list.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>Список врачей</**h1**>

<**div** class="filter">

<**form**>

<**select** name="java-navigator" onchange="top.location.href = this.options[this.selectedIndex].value;">

<**option** value="">фильтр</**option**>

<**option** value="all">Все</**option**>

<**option** value="k2">Вторая квалификационная категория</**option**>

<**option** value="k1">Первая квалификационная категория</**option**>

<**option** value="k0">Высшая квалификационная категория</**option**>

<**option** value="s1">Хирург</**option**>

<**option** value="s2">Фельдшер</**option**>

<**option** value="s3">Терапевт</**option**>

<**option** value="s4">Стоматолог</**option**>

<**option** value="s5">Рентгенолог</**option**>

<**option** value="s6">Психотерапевт</**option**>

<**option** value="s7">Психиатр</**option**>

<**option** value="s8">Отоларинголог</**option**>

<**option** value="s9">Нарколог</**option**>

<**option** value="s10">Гинеколог</**option**>

<**option** value="s11">Педиатр</**option**>

</**select**>

</**form**>

</**div**>

{% if object\_list %}

<**ul**>

{% for doc in object\_list %}

<**li**>

<**a** href="{{ doc.get\_absolute\_url }}">{{ doc.name }}</**a**> ({{doc.doc\_specialty}})

</**li**>

{% endfor %}

</**ul**>

{% else %}

<**p**>Здесь нет ни одного врача</**p**>

{% endif %}

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/pacient\_list.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>Список пациентов</**h1**>

<**div** class="filter">

<**form**>

<**select** name="java-navigator" onchange="top.location.href = this.options[this.selectedIndex].value;">

<**option** value="">фильтр</**option**>

<**option** value="all">Все</**option**>

<**option** value="t">Лечится</**option**>

<**option** value="c">Вылечился</**option**>

<**option** value="h">Направлен в стационар</**option**>

<**option** value="d">Умер</**option**>

</**select**>

</**form**>

<**a** href="{% url 'pacient\_create' %}">Добавить пациента</**a**>

</**div**>

{% if object\_list %}

<**ul**>

{% for pac in object\_list %}

<**li**>

<**a** href="{{ pac.get\_absolute\_url }}">{{ pac.name }}</**a**> (

{% if pac.condition == 't' %} лечится {% endif %}

{% if pac.condition == 'c' %} вылечился {% endif %}

{% if pac.condition == 'h' %} направлен в стационар {% endif %}

{% if pac.condition == 'd' %} умер {% endif %} )

</**li**>

{% endfor %}

</**ul**>

{% else %}

<**p**>Здесь нет ни одного пациента</**p**>

{% endif %}

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/pacient\_detail.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>{{ pacient }}</**h1**>

<**a** href="{% url 'pacient\_update' pacient.pk %}">Изменить информацию о пациенте</**a**><**br**><**br**>

{{ pacient.date\_of\_birth }}

{% if pacient.date\_of\_death != None %}

- {{ pacient.date\_of\_death }}

{% endif %}<**br**>

{{ pacient.soc\_status }},

{% if pacient.condition == 't' %}

на лечении

{% elif pacient.condition == 'c' %}

здоров

{% elif pacient.condition == 'h' %}

направлен в стационар

{% elif pacient.condition == 'd' %}

мертв

{% endif %}

<**div** style="**margin-left**:20px;**margin-top**:20px">

<**h4**>История болезней пациента</**h4**>

{% for copy in pacient.treatment\_history\_set.all %}

<**hr**>

<**p**><**strong**><**a** href="{% url 'treatment\_history\_detail' copy.pk %}">{{copy}}</**a**></**strong**></**p**>

{% endfor %}

</**div**>

<**a** href="{% url 'pacients' 'all' %}">Вернуться</**a**>

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/pacient\_form.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**div** class="AAA">

<**form** action="" method="post">

{% csrf\_token %}

<**table**>

{{ form.as\_table }}

</**table**>

<**input** type="submit" value="Ок" />

</**form**>

<**a** href="{% url 'pacients' 'all' %}">Вернуться</**a**>

</**div**>

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/treatment\_history\_list.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>История лечений</**h1**>

<**div** class="filter">

<**form**>

<**select** name="java-navigator" onchange="top.location.href = this.options[this.selectedIndex].value;">

<**option** value="">фильтр</**option**>

<**option** value="all">Все</**option**>

<**option** value="my">Мои</**option**>

<**option** value="1">Алкоголизм</**option**>

<**option** value="2">Болезнь почек</**option**>

<**option** value="3">Варикоз</**option**>

<**option** value="4">Гайморит</**option**>

<**option** value="5">Гастрит</**option**>

<**option** value="6">Герпес</**option**>

<**option** value="7">Дерматит</**option**>

<**option** value="8">Диарея</**option**>

<**option** value="9">Панкреатит</**option**>

<**option** value="10">Пневмония</**option**>

<**option** value="11">Свинка</**option**>

<**option** value="12">Синдром дауна</**option**>

<**option** value="13">Хронический артрит</**option**>

<**option** value="14">Хронический стресс</**option**>

<**option** value="15">Хроническая стенокардия</**option**>

<**option** value="16">Язвенная болезнь</**option**>

</**select**>

</**form**>

<**a** href="{% url 'treatment\_create' %}">Добавить случай</**a**>

</**div**>

{% if object\_list %}

<**ul**>

{% for hist in object\_list %}

<**li**>

<**a** href="{{ hist.get\_absolute\_url }}">{{hist}}</**a**>

</**li**>

{% endfor %}

</**ul**>

{% else %}

<**p**>Здесь нет ни одной записи</**p**>

{% endif %}

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/treatment\_history\_detail.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**h1**>{{ hist }}</**h1**>

<**p**><**a** href="{% url 'treatment\_history\_update' hist.pk %}">Изменить информацию о лечении</**a**></**p**>

<**hr**>

<**div** style="**margin-left**:20px;**margin-top**:20px">

<**p**><**strong**>{{hist.start\_date\_of\_treatment}}

{% if hist.end\_date\_of\_treatment != None %}

- {{ hist.end\_date\_of\_treatment }}

{% endif %}</**strong**></**p**>

<**p**><**strong**>Пациент:</**strong**> <**a** href="{% url 'pacient-detail' hist.pac\_name.pk %}">{{ hist.pac\_name }}</**a**></**p**>

<**p**><**strong**>Лечащий врач:</**strong**> <**a** href="{% url 'doctor-detail' hist.doc\_name.pk %}">{{hist.doc\_name}}</**a**></**p**>

<**p**><**strong**>Диагноз:</**strong**> {{hist.diagnosis}}</**p**>

<**p**><**strong**>Симптомы:</**strong**> {{hist.description}}</**p**>

<**p**><**strong**>Амбулаторное лечение:</**strong**>

<**abbr** class="

{% if hist.ambulatory == True %}

text-success

{% else %}

text-danger

{% endif %}">

{% if hist.ambulatory == True %}

Да

{% else %}

Нет

{% endif %}

</**abbr**></**p**>

<**p**><**strong**>Диспансерный учет:</**strong**>

<**abbr** class="

{% if hist.dispensary == True %}

text-success

{% else %}

text-danger

{% endif %}">

{% if hist.dispensary == True %}

Да

{% else %}

Нет

{% endif %}

</**abbr**></**p**>

</**div**>

<**a** href="{% url 'treatment\_history' 'all'%}">Вернуться</**a**>

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/templates/cardfile/treatment\_history\_form.html**

{% extends "base\_generic.html" %}

{% block content %}

{% if user.is\_authenticated %}

<**div** class="AAA">

<**form** action="" method="post">

{% csrf\_token %}

<**table**>

{{ form.as\_table }}

</**table**>

<**input** type="submit" value="Ок" />

</**form**>

<**a** href="{% url 'treatment\_history' 'all' %}">Вернуться</**a**>

</**div**>

{% else %}

Вы не залогинены на этом сайте <**u**><**a** href="{% url 'login'%}?next={{request.path}}">залогиниться</**a**></**u**>

{% endif %}

{% endblock %}

**./cardfile/static/css/styles.css**

**body** {

**background-color**:lightsteelblue;

}

.container-fluid{

**width**: 100vw;

**height**: 100vh;

}

.header{

**margin-left**: -20px;

**margin-right**: -15px;

**padding-right**: 20px;

**padding-top**: 5px;

**background**: blue;

**text-align**: right;

**height**: 30px;

**color**: white;

**vertical-align**: middle

}

.row{

**width**: 100%;

**height**: 97%;

}

.col-sm-2{

**height**: 95%;

}

.sidebar-nav {

**margin-top**: 10%;

**padding**: 20px;

**list-style**: none;

**background**: #1976D2;

**opacity**: 0.95;

**border**: 1px solid blue;

**border-radius**: 10px;

**height**: 100%;

}

.col-sm-10{

**padding-left**: 20%;

**padding-top**: 3%;

}

.filter{

**position**: absolute;

**right**: 10px;

}

.AAA{

**padding**: 50px;

**border**: 1px solid blue;

**border-radius**: 10px;

**opacity**: 0.95;

**background**: #3c88d4;

**width**:fit-content;

**margin-left**: auto;

**margin-right**: auto;

}

**input**[type=submit] {

**width**: 100%;

**margin-top**: 10px;

**margin-bottom**: 10px;

**background**:lightsteelblue;

**border**:0 none;

**cursor**:pointer;

**border-radius**: 5px;

}

**a**{

**color**:floralwhite;

}

**a**:**hover**{

**color**: black;

}

**a**:**visited**{

**color**:navy;

}