Слайд 2 “Введение”.

Маршрутизаторы играют важную роль в современном мире обеспечивая передачу данных между различными сегментами сети. Их правильная настройка позволяют эффективно управлять трафиком, оптимизировать скорость передачи данных и обеспечивать защиту сети от внешних угроз.

Реализованный проект представляет из себя систему управления маршрутизатором, в неё входит настройка аккаунта, настройка интернета и уведомление о важных событиях. В нашей реализации нет доступа к реальным устройствам, этот проект рассматривается как интерфейс для дальнейшего внедрения в системы маршрутизаторов.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 3 "Используемые технологии для работы над проектом".

Для ведения учёта задач мы использовали Trello, систему управления проектами в стиле доски канбан. Мы выделили 4 колонки. Это задачи, которые нужно сделать, задачи в работе, выполненные задачи и опциональные(необязательные) задачи. Для хранения кода мы использовали репозиторий на GitHub. В качестве среды разработки - IntelliJ Idea.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 4 “База данных”.

Для подключения к СУБД PostgreSQL использовалось расширение Spring Data JPA (Java Persistence API). База данных содержит 7 основных сущностей и 3 вспомогательных, выделенных в процессе приведения базы данных к нормальной форме.

Сущность “Основные настройки” инкапсулирует в себе информацию о панели роутера: данные администратора и банковского счета, настройки проводной и беспроводной сети, подключенные устройства и список пользователей, которым на электронную почту будут присылаться уведомления, касающиеся работы сети.

В процессе проектирования базы данных использовались типы связей один к одному, один ко многим, многие ко многим.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 5 “Spring Security”.

Для защиты ресурсов и API приложения и ограничения к ним доступа использовалась технология Spring Security. При попытке перейти на любой URL-адрес кроме входа (login) и выхода (logout) произойдет перенаправление на форму ввода логина и пароля. В первый раз пользователь заходит на панель роутера с помощью данных, которые задаются производителем (логин и пароль), что моделируется соответствующей записью в таблице базы данных. После входа эти данные можно изменить.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 6 "Проводной интернет ".

Для настройки интернета мы выделили два вида интернет соединения, это проводной и беспроводной.

Проводной интернет содержит настройку IP-адреса, маски подсети и основного шлюза. Есть возможность выбрать тип IP-адреса, это динамический и статический. Используя динамический тип, данные IP-адреса, маски подсети и шлюза, заполняются автоматически. Используя статический тип, данные необходимо ввести вручную.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 7 "Беспроводной интернет".

Беспроводной интернет содержит настройку имени сети, пароля и режима wifi-соединения. Каждый режим отвечает за определённые скорость передачи данных и диапазон частот, в котором будет работать wifi. Это имеет важное значение при одновременном использовании очень старых и новых устройств, поддерживающие разный диапазон частот.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 8 "Работа с устройствами".

При подключении к интернету устройств как по проводу, так и по wifi-сети, информация о них сохраняется в базу данных. В административной панели есть возможность просмотра всех подключенных устройств и их отключения от сети. Это может быть полезным для отключения неиспользуемых устройств или тех, которые получили несанкционированный доступ.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 9 “Уведомление получателей”.

Наше приложение предоставляет функциональность уведомления пользователей о перебоях в сети интернет и рассылки информации о списании средств за оплату услуг.

Для этого администратор в соответствующем пункте на главной странице должен внести информацию об электронной почте, на которую будут приходить уведомления.

Реализация данного функционала осуществляется посредством JavaMail API.

Для отправки писем используется протокол SMTP, в приложении указываются настройки подключения к почтовому серверу gmail. На основе настроек создается сессия, производится аутентификация, формируется и затем отправляется письмо.

Ситуация перебоев в сети интернет моделируется посредством перехода по эндпоинту “/disconnect-internet/id\_администратора”.

Уведомление пользователей о списании средств за оплату услуг интернета осуществляется раз в месяц и реализовывается посредством планировщика Quartz Scheduler.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Слайд 10 “Дальнейшее развитие проекта”.

Поскольку текущий проект представляет из себя только интерфейс взаимодействия с административной панелью wifi-роутера, дальнейшее его развитие может быть следующим:

1. Улучшение визуального отображения всех компонентов системы.
2. Интеграция в существующие административные панели для возможности выбора из нескольких видов понравившийся.
3. Сотрудничество с производителями маршрутизаторов для использования в качестве основной административной панели.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Информация для демонстрации кода.

Наше приложение имеет многоуровневую (многослойную) архитектуру.

В её рамках были выделены следующие уровни:

1. model
2. repository
3. services
4. controller
5. configuration.

Пакет model содержит сущности приложения, каждая из которых помечена аннотацией @Entity и соответствует таблице базы данных.

Основные сущности:

1. Аккаунт администратора панели роутера. Данная сущность нужна для получения доступа к панели и имеющимся там настройкам. Содержит поля для хранения логина и пароля.
2. Лицевой счёт. С администратором связана сущность лицевого счета для того, чтобы пользователь после входа в панель мог посмотреть баланс на его счете и смог оценить необходимость оплаты услуг интернета. Сущность лицевого счета содержит номер счета и баланс. Связь с сущностью администратора: один к одному.
3. Получатели. Эта сущность связана с администратором и содержит информацию об электронной почте. Используется для уведомления в случае оплаты услуг интернета или для информировании о перебоях в сети интернет.

Связь с сущностью администратора: многие ко многим.

1. Настройки проводного интернета. Данная сущность содержит информацию об ip-адресе, маске сети.
2. Настройки беспроводного интернета. Данная сущность содержит информацию об имени сети, пароле, режиме сети.
3. Устройства. Данная сущность отображает информацию о подключенных к wifi-роутеру устройствах. Содержит поля названия устройства, ip-адреса, mac-адреса и типа его подключения к роутеру.
4. Основные настройки. Эта сущность, которая инкапсулирует в себе основную информацию административной панели роутера. Связана с сущностью администратора связью один к одному, потому что к одному роутеру может иметь доступ только один администратор. Аналогично со связью один к одному сущность основных настроек связана с настройками проводного и беспроводного интернета. Также в основных настройках указываются подключенные устройства. Их может быть несколько, ноутбук, телефон, стационарный персональный компьютер, поэтому сущность основных настроек имеет связь с ними один ко многим.

Также в базе присутствуют вспомогательные таблицы, которые были выделены в процессе приведения базы данных к нормальной форме:

1. Тип соединения устройства. Относится к сущности устройства. Его соединение к роутеру может быть проводным или беспроводным.
2. Тип соединения интернета. Данная сущность относится к информации об ip-адресе проводного интернета, который может быть динамическим или статическим.

Динамический ip-адрес. Меняется после завершения каждого сеанса подключения к Интернету. Выдается на определенное время и меняется автоматически.

Статический ip-адрес. Это постоянный ip-адрес устройства в сети, не изменяется после перезагрузки системы/отключения от сети. Назначается администратором или предоставляется провайдером за дополнительную плату.

1. Режимы Wi-Fi. Относится к сущности беспроводного интернета. Существуют пять основных режимов работы Wi-Fi 802.11:b/g/n/ac/ax, которые отличаются максимальной скоростью передачи данных и диапазоном частот.

Пакет repository содержит интерфейсы для доступа к сущностям. Каждый интерфейс помечен аннотацией @Repository, которая используется для указания на то, что класс предоставляет механизм для хранения, извлечения, обновления, удаления и поиска объектов. Каждый интерфейс расширяет соответствующий модели JpaRepository, который содержит API для базовых операций CRUD и поддерживает автоматическую генерацию кастомных методов для извлечения данных.

Например, AdminAccountRepository содержит такой метод поиска аккаунта администратора по его логину.

Пакет services предоставляет бизнес-логику для взаимодействия с моделями и репозиториями. Основными методами, которые составляют бизнес логику и присутствуют практически в каждом из классов являются методы изменения сущности, получения по идентификатору, получение всех сущностей. Например, AccountService или AdminAccountSevice. Также для классов настроек проводного и беспроводного интернета реализованы методы генерации ip-адреса, маски подсети, шлюза по умолчанию (ConfigService) и генерации пароля, имени сети (WifiService).

Также в этом пакете расположен сервис уведомления пользователей.

Класс PaymentNotificationService содержит функционал по оповещению пользователей раз в месяц о списании денежных средств на оплату услуг интернета. Для того, чтобы включить планирование используется аннотация @EnableScheduling.

Метод getRouterServicesPayment помечен аннотацией @Scheduled с настройкой времени, когда он будет запущен. Сама дата содержится в переменной PAYMENT\_DATE\_TIME, значение которой прописывается в .env файле.

Указываются секунда, минута, час, день, месяц, день недели.

Example for PAYMENT\_DATE\_TIME: 0 30 19 26 \* ?

# Seconds Minutes Hours Day-of-Month Month Day-of-Week

# \* --- any day of the week

# ? --- every month

Класс SendEmailService используется для отправки сообщений пользователю. Сначала указываются настройки SMTP-соединения с почтовым сервером gmail с помощью пар ключ-значение, передаваемых в виде объекта класса Properties.

Далее создается объект класса наследника абстрактного класса Authenticator, переопределяется метод getPasswordAuthentication() и посредством полиморфизма созданный объект присваивается переменной с типом Authenticator.

Далее указывается электронная почта и пароль, причем чтобы приложение могло отправлять почту нужна дополнительная настройка в самой почте (gmail) и специальный пароль, который при этом выдается. Далее создается сессия, которая представляет собой сеанс работы с почтовыми серверами. Создается и отправляется письмо.

Пакет controller содержит классы, отвечающие за обработку поступающих в приложение запросов. Обработка состоит из обращения к соответствующему сервису, получение необходимых объектов и загрузка html-страницы с переданными данными. Классы контроллеры помечены аннотацией @Controller, что определяет класс как обработчика веб-запросов. Аннотация @RequestMapping используется для отображения запросов на методы контроллеров. Она имеет различные атрибуты для сопоставления по URL-адресу, HTTP-методу, параметрам запроса и так далее.

Например, класс MainController. Он содержит метод index, который загружает начальную веб-страницу, которая соответствует URL-адресу “\” (обратный слэш). В нем происходит получение объектов администратора, настроек проводной и беспроводной сети. Происходит их подготовка: если администратор заходит на панель роутера впервые, генерируются новые настройки сети. Иначе из базы данных загружаются существующие настройки панели роутера. Эти данные назначаются модели, которая выступает промежуточным звеном между контроллером и представлением. В качестве результата метод возвращает название html-страницы, которая будет загружена.

Также MainController содержит метод disconnectInternet, который служит для обработки ситуации перебоев в сети. Как говорилось ранее, моделирование происходит через эндпоинт /disconnect-internet/{id администратора}. В этом методе формируется сообщение, которое затем передается в сервис отправки сообщений.

Пакет configuration.

Содержит настройки Spring Security. Клиент отправляет запрос приложению, и контейнер создает цепочку фильтров. В нашем приложении цепочка фильтров содержит AuthenticationFilter. Для любого запроса, кроме входа (login) и выхода (logout), пользователь должен зарегистрироваться и указать данные в форме для получения доступа к панели роутера.

Для шифрования пароля администратора используется функция строгого хэширования BCrypt.

Пакет resources. Он содержит sql-код, который используется для инициализации таблиц данными. Также в нём содержится код на языке javascript для придания интерактивности веб-приложению. Он используется для переключения типа ip-адреса: с динамического на статический и наоборот. Подпакет templates содержит шаблоны веб-страниц.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дополнительная информация.

Java Persistence API — это спецификация, определяющая стандарт объектно-реляционного отображения (ORM) для хранения, доступа и управления Java-объектами в реляционной базе данных.

[Начало работы с Spring Data JPA (attacomsian.com)](https://attacomsian.com/blog/getting-started-spring-data-jpa)

b — это самый медленный режим. До 11 Мбит/с.

g — максимальная скорость передачи данных 54 Мбит/с

n — новый и скоростной режим. До 600 Мбит/c

802.11b (Wi-Fi 1) – работает в диапазоне 2.4 ГГц. Скорость до 11 Мбит/с.

802.11a (Wi-Fi 2) – это один из первых стандартов, работает в диапазоне 5 ГГц. Максимальная скорость 54 Мбит/c.

802.11g (Wi-Fi 3) – более современный и доработанный стандарт 802.11b. Работает так же в диапазоне 2.4 ГГц, скорость до 54 Мбит/с. Совместим с 802.11b.

802.11n (Wi-Fi 4) – самый популярный стандарт на сегодняшний день. Скорость до 600 Мбит/c в диапазоне 2.4 ГГц, совместимость с 802.11a/b/g.

802.11ac (Wi-Fi 5) – стандарт, который работает только в диапазоне 5 ГГц. Скорость передачи данных до 6,77 Гбит/с. Данный режим есть только на двухдиапазонных маршрутизаторах, которые могут транслировать сеть в диапазоне 2.4 ГГц и 5 ГГц.

802.11ax (Wi-Fi 6) – самый новый и быстрый стандарт Wi-Fi. На сегодняшний день его поддерживает большое количество роутеров и устройств.

[Режим работы Wi-Fi сети b/g/n/ac/ax. Что это и как сменить в настройках роутера? (help-wifi.com)](https://help-wifi.com/nastrojka-wi-fi-setej/rezhim-raboty-wi-fi-seti-b-g-n-ac-chto-eto-i-kak-smenit-v-nastrojkax-routera/?ysclid=lqazednqo3814911851)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Use Cases

1. При попытке попасть на страницу административной панели нас просят авторизоваться, необходимо ввести логин и пароль.
2. После входа отображается главная страница административной панели wifi-роутера, где есть кнопки для перехода в нужный раздел.
3. На странице аккаунта мы можем изменить логин и пароль администратора.
4. Настройка интернета.

По умолчанию стоит тип “динамический IP-адрес”, нажать на кнопку -> данные обновляются автоматически.

Переключиться на “статический IP-адрес”, ввести данные вручную и сохранить их.

1. Настройки беспроводного режима сети. Ввод имени сети, установка пароля и выбор режима wifi-соединения.
2. Для удобства есть возможность просмотра сразу всех настроек интернета.
3. В разделе устройства отображаются подключенные проводные и беспроводные устройства. Для демонстрации подключения мы сделали кнопку, которая имитирует подключение нового устройства. Если мы обнаружили, что есть неизвестное для нас устройство и подозрения о несанкционированном получении доступа к wifi-сети, мы можем его отключить, нажав на крестик.
4. В разделе получателей мы можем добавить емейл почты, на которые будут приходит уведомления о важных событиях. Добавленные записи можно изменить и удалить. Отправка письма о списании средств через Quartz. Отправка письма о перебоях в сети интернет через ендпоинт.