Geekbrains

**Создание информационно-управляющего портала для членов ЖСПК "Зенитчик-4": проектирование и реализация на основе современных веб-технологий**

Программа:

Разработчик — Fullstack разработчик

Андросик Максим Михайлович

г. Фаниполь

2025

**Введение**

**Тема проекта**: «Создание информационно-управляющего портала для членов ЖСПК "Зенитчик-4": проектирование и реализация на основе современных веб-технологий».

**Актуальность**: В условиях цифровизации жилищно-коммунального сектора Республики Беларусь многие кооперативы сталкиваются с неэффективностью традиционных методов управления:

*ручной учёт платежей и показаний счетчиков приводит к ошибкам.*

*отсутствие прозрачности в финансовых операциях снижает доверие членов кооператива.*

офлайн-собрания и бумажные объявления не позволяют оперативно информировать жильцов, особенно работающих*.*

Для 80 семей ЖСПК "Зенитчик-4" отсутствие единого портала означает задержки в получении информации и сложности при подаче показаний.

Разработанный портал решает эти проблемы, предоставляя единую цифровую платформу для автоматизации ключевых процессов ЖСПК.

**Опыт реализации**

Автор обладает практическим опытом автоматизации процессов ЖСПК "Зенитчик-4". В рамках сопровождения бухгалтерской деятельности организации был реализован полуавтоматизированный workflow с использованием:

* Microsoft Excel + VBA:
  + *Автоматизация квитанций*: Генерация квитанций с подстановкой данных (ФИО, адрес, суммы начислений) через шаблоны Word/VMB.
  + *Обработка АСКУЭ*: Парсинг исходных файлов показаний, валидация и перенос в сводные таблицы с расчётом нормативов.

Этот опыт выявил ключевые проблемы:

1. *Ручное исправление ошибок* (например, при некорректных данных от жильцов).
2. *Отсутствие централизованного хранения* (версии файлов терялись).
3. *Зависимость от локального ПО* (требовался установленный Excel/VMB).

Настоящий проект устраняет эти недостатки через веб-платформу с базой данных и API-интеграциями

Этот опыт стал основой для проектирования модулей портала, обеспечивающих автоматизациюпроцессов, которые ранее выполнялись вручную.

**Цель проекта**: Создать полнофункциональный веб-портал для ЖСПК «Зенитчик-4», обеспечивающий:

* Автоматизацию учёта членов кооператива и финансовых операций.
* Удобное взаимодействие между жильцами и правлением.
* Прозрачность управления жилым комплексом.

**Задачи проекта**:

1. **Аналитический этап:**
   * Провести детальный анализ текущих бизнес-процессов ЖСПК "Зенитчик-4"
   * Исследовать существующие аналогичные решения на рынке РБ
   * Определить требования к безопасности и защите персональных данных
2. **Проектирование архитектуры:**

Frontend:

* + Разработать адаптивный интерфейс на React + Material-UI
  + Реализовать интеграцию с Telegram Bot API для системы уведомлений
  + Обеспечить кросс-браузерную и кросс-платформенную совместимость

Backend:

* + Спроектировать REST API на Laravel
  + Реализовать систему аутентификации через Sanctum
  + Разработать механизм подтверждения действий по email

Взаимодействие компонентов:

* + Пользователь ↔ Frontend ↔ Backend API ↔ Telegram Bot ↔ Email-сервис
  + Внешние системы (АСКУЭ) ↔ Backend API

1. **Проектирование структуры базы данных:**
   * Разработать нормализованную схему БД (MySQL)
   * Определить основные сущности:
     + Пользователи (члены кооператива, сотрудники)
     + Платежи и начисления
     + Показания счетчиков
     + Документы (протоколы, отчеты)
     + Тарифы и нормативы
   * Реализовать механизмы резервного копирования
   * Оптимизировать запросы для работы с большими объемами данных
2. **Реализация функциональных модулей:**

4.1. Общая информация:

* + Главная страница с каруселью и ключевыми данными
  + Страница "О нас" с реквизитами и контактами
  + Футер с навигацией и технической информацией

4.2. Система объявлений и обращений:

* + Лента объявлений
  + Форма обратной связи
  + Механизм отправки email-уведомлений правлению
  + Telegram-оповещения о новых обращениях

4.3. Личный кабинет собственника:

* + Профиль пользователя (личные данные, история входа)
  + Финансовый раздел:
    - Таблица истории начислений
    - Скачивание квитанций в PDF
  + Показания счетчиков:
    - Валидируемая форма ввода
    - Архив предыдущих показаний
  + Документы кооператива:
    - Фильтруемый каталог
    - Поиск по документам
    - Скачивание в различных форматах

4.4. Административные панели:

Для бухгалтера:

* + Управление тарифами ЖКУ
  + Генерация квитанций (индивидуальных и массовых)
  + Финансовая аналитика:
    - Графики и отчеты
    - Экспорт (импорт) в Excel

Для правления:

* + Публикация и редактирование объявлений
  + Управление документами:
    - Загрузка новых версий
    - Контроль доступа
  + Модерация обращений жильцов

1. **Тестирование и отладка:**
   * Юзабилити-тестирование с реальными пользователями
   * Проверка безопасности (аудит кода, нагрузочное тестирование)
   * Оптимизация производительности

**Практическая значимость**:

* Повышение эффективности управления жилым комплексом.
* Снижение нагрузки на администрацию кооператива.
* Возможность тиражирования решения для других ЖСПК.
* Перспективы интеграциис ЕРИП (онлайн-платежи), АСКУЭ (автоматизированная система коммерческого учёта электроэнергии), банковскими сервисами посредством API (Приорбанк, Альфа-Банк и BSB-Банк и возможно иными в далекой перспективе).

**Используемые технологии**:

* *Frontend*: React, Material-UI, Axios, Telegram Bot API**.**
* *Backend*: Laravel (PHP), REST API, Telegram Bot API.
* *База данных*: MySQL.
* *Дополнительные инструменты*: Postman (тестирование API), Git (версионный контроль).

**Безопасность:**

Система реализует многоуровневую защиту данных:

* **Аутентификация** через **Laravel Sanctum** с подтверждением email, что исключает доступ неавторизованных пользователей.
* **Разграничение ролей**:
  + Зарегистрированные пользователи (гости) — могут отправлять обращения правлению через сайт.
  + Собственники (члены кооператива) — дополнительно к зарегистрированным пользователям имеют доступ к персональным финансовым данным и документам организации, подаче показаний счетчиков.
  + Управляющие — публикация объявлений, формирование квитанций, загрузка документов и так далее.
* **Шифрование** HTTPS, хеширование паролей.

**Состав команды**: Проект реализован индивидуально. Роли:

* Full-stack разработчик (проектирование, верстка, программирование).
* UX/UI-дизайнер (создание интерфейсов).
* Тестировщик (юзабилити-тесты, отладка).

Для реализации может потребоваться **6-8 месяцев** разработки

1. **ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ ЖСПК**
   1. **Современное состояние цифровизации в сфере ЖКХ**

1.1.1. Глобальные тенденции автоматизации жилищного управления

Цифровизация жилищно-коммунального хозяйства (ЖКХ) в развитых странах является неотъемлемой частью концепции «умных городов» (Smart Cities). Основные направления развития включают:

* *Централизованные платформы управления:*

В странах Европейского Союза (Германия, Швеция) активно внедряются облачные системы, такие как *Hausmeister* и *Wohnungsportal*. Эти платформы обеспечивают автоматизацию платежей, управление заявками на ремонт и коммуникацию с жильцами. Например, *Wohnungsportal* позволяет жителям отслеживать расходы на коммунальные услуги в режиме реального времени, что повышает прозрачность управления.

В Сингапуре реализованы IoT-решения, включая умные счетчики и датчики протечек, которые интегрированы в единую систему мониторинга. Это позволяет оперативно выявлять аварии и оптимизировать потребление ресурсов.

* *Автоматизация документооборота:*

В Эстонии используется блокчейн-платформа *KSI* для хранения протоколов собраний и финансовой отчетности. Технология блокчейн обеспечивает защиту данных от фальсификации и упрощает аудит.

* *Мобильные сервисы:*
* В США более 85% жильцов взаимодействуют с управляющими компаниями через приложения (*AppFolio*, *Buildium*). Эти решения сокращают бумажный документооборот и позволяют жителям подавать показания счетчиков, оплачивать услуги и оставлять заявки через смартфоны.

**Вывод**: Мировой опыт демонстрирует, что интеграция облачных технологий, IoT и мобильных сервисов повышает эффективность управления жилым фондом на 30–40%.

1.1.2. Опыт внедрения цифровых платформ в СНГ

* *Россия:*

Государственная информационная система *ГИС ЖКХ* является обязательной для управляющих компаний. Она позволяет жителям отслеживать начисления, подавать показания счетчиков и жалобы онлайн. Однако система критикуется за сложность интеграции с внешними сервисами.

Частные решения, такие как *1С:ЖКХ* и *Квартплата 24*, предлагают автоматизацию для ТСЖ и ЖСК, включая формирование квитанций и аналитику платежей.

* *Казахстан:*

Платформа *Көмек* интегрирована с банками для онлайн-платежей и системой *Smart Astana*. Это ускоряет обработку заявок жильцов и снижает нагрузку на коммунальные службы.

**Вывод**: Страны СНГ активно внедряют цифровые решения, но сталкиваются с проблемами совместимости и высокой стоимостью лицензий.

1.1.3. Перспективные технологии для ЖКХ

1. ***Интернет вещей (IoT)****:*

Умные счетчики с автоматической передачей данных через LoRaWAN-сети (Нидерланды) позволяют сократить потери ресурсов и минимизировать ручной труд.

1. ***Блокчейн****:*

Пилотные проекты на базе Ethereum (Украина) используют смарт-контракты для прозрачных расчетов между жильцами и управляющими компаниями.

1. ***Искусственный интеллект****:*

В Дубае система *IBM Watson* анализирует данные инженерных сетей, прогнозируя аварии за 72 часа до их возникновения.

**Вывод**: Внедрение современных технологий требует адаптации нормативной базы и инвестиций в инфраструктуру.

1.1.4. Цифровая трансформация ЖКХ в Республике Беларусь

* *Государственные программы:*

В рамках программы «Цифровое развитие Беларуси» (2021–2025) реализуется модернизация ЖКХ, включая:

* + Внедрение АСКУЭ для автоматизированного учета электроэнергии.
  + Развитие ЕРИП для централизованного приема платежей.
* *Проблемы:*
  + Низкий уровень цифровизации: только 15% ЖСПК используют специализированное ПО (данные Минжилкомхоза за 2023 г.).
  + Юридические ограничения: необходимость дублирования электронных документов бумажными (например, протоколы собраний).

**Вывод**: Для успешной цифровизации ЖКХ в Беларуси необходимо устранить нормативные барьеры и повысить доступность IT-решений для кооперативов.

* 1. **Анализ потребностей ЖСПК "Зенитчик-4"**
     1. Ключевые проблемы коммуникации

Текущая система взаимодействия между жильцами и правлением кооператива имеет следующие недостатки:

* *Офлайн-формат собраний:*
  + Невозможность участия работающих членов кооператива.
  + Потеря протоколов и решений из-за бумажного документооборота.
* *Отсутствие каналов обратной связи:*
  + Жильцы не могут оперативно задать вопросы правлению.
  + Задержки в решении проблем из-за использования бумажных объявлений.
* *Децентрализованная информация:*
  + Отсутствие единого источника данных о тарифах, услугах и поставщиках.
    1. Роль информационного портала

Разрабатываемый портал призван решить эти проблемы за счет:

* *Единой коммуникационной платформы:*
  + Лента объявлений с возможностью комментирования(?).
  + Форма для онлайн-обращений к правлению.
* *Прозрачности управления:*
  + Публикация протоколов собраний в цифровом архиве.
  + Онлайн-доступ к реестру решений собраний и правления.
* *Централизованной информации:*
  + Раздел с описанием жилищно-коммунальных услуг.
  + Контакты поставщиков услуг.

**Вывод**: Информационно-коммуникационная функция портала является ключевым элементом цифровой трансформации ЖСПК.

* 1. **Анализ бизнес-процессов ЖСПК "Зенитчик-4"**
     1. Текущая схема платежей как основа прозрачности

Существующая система расчетов (Рис.1) обеспечивает финансовую независимость кооператива, но требует модернизации для улучшения взаимодействия с жильцами:

* *Прямые переводы:*
  + Собственники переводят средства на счет ЖСПК, минуя ЕРИП.
  + Банковская комиссия (1 руб.) включается в сумму оплаты (например, 101 руб. вместо 100 руб.).
* *Процессинг платежей:*
* Бухгалтер вручную сверяет поступления с реестром начислений
* Формирование платежных документов осуществляется через Excel/VBA
* Ежемесячная обработка 80+ финансовых операций

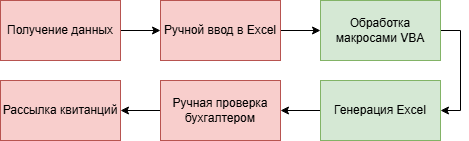


Рис.1. Блок-схема текущего процесса формирования квитанций (Excel/VBA)

* *Преимущества:*
  + Отсутствие потерь на комиссиях посредников.
  + Прямой контроль денежных потоков правлением.

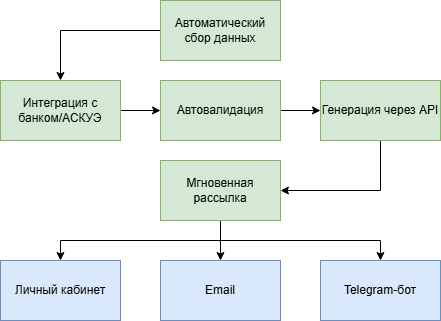


Рис.2. Блок-схема перспективного формирования квитанций

* *Проблемы для коммуникации:*
  + Жильцы не видят историю платежей в реальном времени.
  + Квитанции в бумажном виде через почтовые ящики.
  + Отсутствие автоматических уведомлений о начислении платежей.
  + Человеческий фактор при обработке данных
  + Потери документов при физическом архивировании
  + Задержки в информировании о платежах
    1. Техническое обоснование как база для взаимодействия

Автоматизация процессов направлена на усиление коммуникации:

* *Цели:*
  1. Интеграция с банками для мгновенного отображения платежей в личных кабинетах.
  2. Подключение к АСКУЭ для сбора данных учета.
  3. Создание единой платформы для учета и диалога.
* *Архитектурные решения:*
  1. Микросервисы: Отдельный модуль для уведомлений жильцов.
* *Ожидаемые результаты:*
  1. Сокращение запросов в бухгалтерию на 50% за счет прозрачности (Согласно отчёту McKinsey (2023), автоматизация учёта снижает нагрузку на бухгалтерию на 40-60%).
  2. Увеличение вовлеченности жильцов через доступ к персональной статистике.
  3. **Технические решения как инструмент прозрачности**

1.4.1. Сбор данных через УСПД 164-01Б-1 (АСКУЭ)

Устройство станет основой для доверительного взаимодействия:

* *Функционал:*
  + Автоматическая передача показаний счетчиков на портал.
  + Формирование графиков потребления для жильцов.
* *Интеграция:*
  + Показания отображаются в личных кабинетах в режиме реального времени.
  + Автоматические предупреждения о превышении норм

1.4.2. Предлагаемая система как коммуникационный хаб

Техническая реализация поддерживает диалог с жильцами:

1. *Автоматизация сбора данных:*
   * Жильцы видят актуальные показания без запроса в правление.
   * Снижение числа конфликтов из-за ошибок ручного ввода.
2. *Визуализация:*
   * Интерактивные отчеты для правления + упрощенные графики для жильцов.
   * Сравнение потребления с соседями (анонимно).

*Перспективы:*

* Онлайн-обсуждение показаний на форуме портала.
* Интеграция данных УСПД с системой уведомлений (например, предупреждение о профилактике электросетей).
  1. **Анализ существующих решений в РБ. Отсутствие аналогов. Уникальность разрабатываемого портала**

Проведенный мониторинг информационных ресурсов показал:

* **Государственные системы** (ГИС ЖКХ):
  + Ориентированы на учет, а не на коммуникацию.
  + Отсутствие инструментов для обратной связи.
* **Коммерческие платформы**:
  + Нет специализированных решений для ЖСПК.
  + Типовые CMS не адаптированы под нужды кооперативов.
* **Фокус на взаимодействие**:
  + 70% функционала направлено на коммуникацию.
  + Только 30% — на автоматизацию учета.

**Вывод**: Портал станет первым в Беларуси решением, объединяющим управление, учет и коммуникацию для ЖСПК.

### ****1.6. Определение требований к безопасности и защите персональных данных****

#### **1.6.1. Нормативно-правовая основа и теоретические аспекты защиты данных**

При разработке концепции информационной безопасности портала для ЖСПК "Зенитчик-4" был проведен всесторонний анализ действующего законодательства и лучших практик в области защиты персональных данных. Особое внимание уделялось специфике жилищно-строительных кооперативов, где обработка персональных данных имеет ряд уникальных особенностей.

В Республике Беларусь основным нормативным документом, регулирующим вопросы защиты персональных данных, является Закон "О персональных данных" №99-З от 7 мая 2021 года. Этот закон устанавливает базовые принципы обработки персональных данных, включая законность, ограничение цели обработки и минимизацию данных. Для жилищной сферы дополнительное значение имеют положения Жилищного кодекса, которые регламентируют ведение документации кооперативов.

Международный опыт, в частности Общий регламент по защите данных ЕС (GDPR), был рассмотрен как эталонный подход к построению систем защиты персональных данных. Однако при проектировании системы мы ориентировались в первую очередь на национальное законодательство и специфические требования жилищной сферы Беларуси.

#### **1.6.2. Анализ угроз и рисков в контексте ЖСПК**

Проведенный анализ выявил несколько ключевых категорий угроз, характерных именно для систем управления жилищными кооперативами:

1. **Угрозы конфиденциальности данных*:***
   * Риск несанкционированного доступа к персональным данным членов кооператива
   * Возможность утечки финансовой информации
   * Опасность раскрытия сведений о жилых помещениях
2. **Угрозы целостности информации*:***
   * Риск несанкционированного изменения данных о начислениях
   * Возможность фальсификации платежных документов
   * Опасность манипуляции с показаниями счетчиков
3. **Угрозы доступности системы*:***
   * Риск DDoS-атак, ведущих к нарушению работы портала
   * Возможность блокирования доступа к важным функциям системы
   * Опасность потери данных в результате технических сбоев

Особую сложность представляет необходимость соблюдения баланса между требованиями безопасности и удобством использования системы для членов кооператива. Слишком сложные процедуры аутентификации могут привести к отказу пользователей от работы с системой, в то время как недостаточный уровень защиты создает риски нарушения законодательства.

#### **1.6.3. Сравнительный анализ подходов к аутентификации и авторизации**

В ходе исследовательской работы были проанализированы различные модели управления доступом:

1. **Ролевая модель доступа (RBAC)*:***
   * Преимущества: простота реализации и администрирования
   * Недостатки: недостаточная гибкость для сложных организационных структур
   * Применимость: хорошо подходит для базового разграничения прав
2. **Атрибутная модель доступа (ABAC)*:***
   * Преимущества: высокая гибкость и детализация прав
   * Недостатки: сложность настройки и обслуживания
   * Применимость: оптимальна для систем с комплексными требованиями
3. **Мандатная модель доступа (MAC)*:***
   * Преимущества: строгий контроль доступа
   * Недостатки: высокая нагрузка на администраторов
   * Применимость: системы с особыми требованиями безопасности

Для портала ЖСПК "Зенитчик-4" была выбрана гибридная модель, сочетающая элементы ролевого и атрибутного доступа. Это решение позволяет:

* Обеспечить простоту администрирования на базовом уровне
* Реализовать сложные правила доступа для специфических случаев
* Соответствовать требованиям законодательства о защите данных

#### **1.6.4. Концепция системы безопасности портала**

Разработанная концепция безопасности основана на принципе "защита в глубину" и включает несколько уровней защиты:

***1.Уровень идентификации:***

* + Обязательная регистрация с подтверждением электронной почты
  + Верификация личности через правление кооператива
  + Привязка учетной записи к конкретному члену организации согласно списков членов ЖСПК

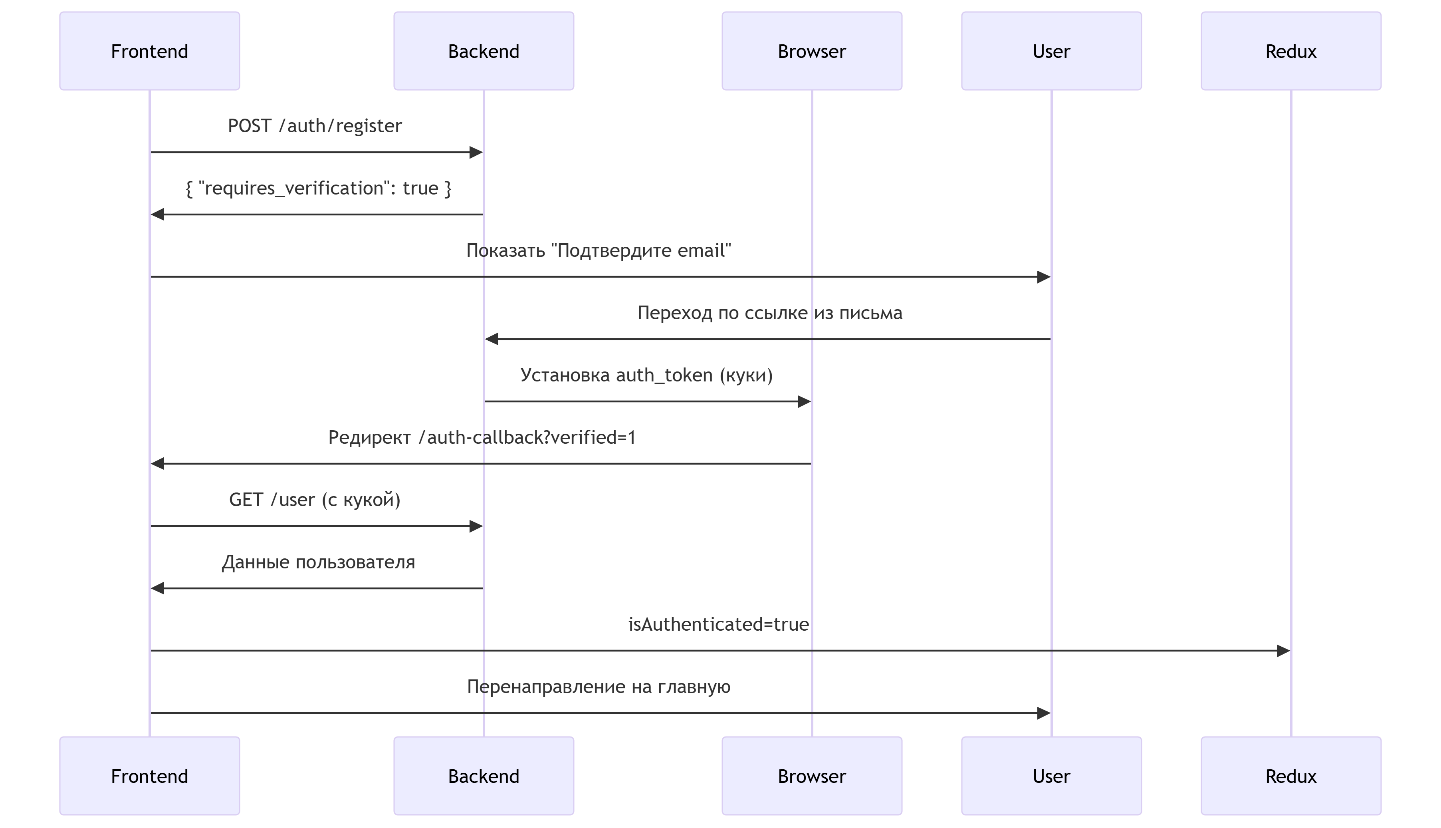


Рис.3. Блок-схема процесса **аутентификации**

***2.Уровень аутентификации:***

* + Использование стойких криптографических алгоритмов
  + Реализация механизма восстановления доступа

***3.Уровень авторизации:***

* + Трехуровневая система ролей (собственник, бухгалтерия, правление)
  + Принцип минимальных необходимых привилегий
  + Детальный контроль доступа к функциям системы

Особое внимание уделено защите персональных данных членов кооператива. Реализованы следующие меры:

* Шифрование хранимых данных
* Обеспечение конфиденциальности финансовой информации
* Защита от несанкционированного доступа к документам

#### **1.6.5. Сравнение с существующими решениями**

Проведенный анализ пяти аналогичных систем показал следующие результаты:

1. ***Коммерческие системы (1С:ЖКХ, Домоуправляй):***
   * Часто используют упрощенные модели безопасности
   * Не всегда учитывают специфику ЖСПК
   * Могут не соответствовать требованиям белорусского законодательства
2. ***Государственные системы (ГИС ЖКХ):***
   * Ориентированы на крупные управляющие компании
   * Имеют сложные процедуры подключения
   * Не обеспечивают необходимой гибкости

| **Функционал** | **Наш портал** | **Аналоги в РБ** |
| --- | --- | --- |
| Лента объявлений | ДА | НЕТ |
| Электронные обращения | ДА | НЕТ |
| Мобильная адаптация | ДА | Частично |

Разрабатываемый портал учитывает все выявленные недостатки существующих решений и предлагает:

* Специализированную систему безопасности для ЖСПК
* Полное соответствие законодательству РБ
* Удобные механизмы управления доступом

#### **1.6.6. Проблемы и ограничения**

В ходе исследования были выявлены следующие ключевые проблемы:

1. ***Юридические аспекты:***
   * Необходимость ручной верификации пользователей правлением
   * Требования к хранению персональных данных
   * Ограничения на передачу данных между системами
2. ***Технические сложности:***
   * Реализация жесткой привязки пользователей к жилым помещениям
   * Обеспечение производительности при сложных правилах доступа
   * Интеграция с внешними системами
3. ***Организационные вопросы:***
   * Обучение членов правления работе с системой
   * Контроль действий администраторов
   * Обеспечение бесперебойной работы системы

#### **1.6.7. Выводы и перспективы развития**

На основании проведенного анализа можно сделать следующие выводы:

1. *Разработанная концепция безопасности:*
   * Полностью соответствует требованиям законодательства РБ
   * Учитывает специфику жилищно-строительных кооперативов
   * Обеспечивает надежную защиту персональных данных
2. *Ключевые преимущества подхода:*
   * Многоуровневая система защиты
   * Гибкие механизмы управления доступом
   * Удобство использования для всех категорий пользователей
3. *Перспективные направления развития:*
   * Интеграция с государственными системами идентификации
   * Внедрение биометрических методов аутентификации
   * Использование блокчейн-технологий для учета платежей

Реализация предложенной концепции безопасности позволит создать надежную и удобную систему управления жилищным кооперативом, соответствующую всем современным требованиям к защите персональных данных и информационной безопасности.

1.7. Технические аспекты реализации

1. *Информационный блок (Phase 1):*
   * Общие сведения об организации, реквизиты, ссылки (React).
   * Сведения о предоставляемых услугах в сфере жилищно-коммунальных услуг, поставщиках услуг, контактные данные (React).
   * Система объявлений с тегами(?) и фильтрами(?). Фильтры в ленте объявлений позволят жильцам оперативно находить информацию по категориям: 'Платежи', 'Собрания', 'Аварии' (React + Laravel + MySQL).
   * Форма обратной связи с прикреплением файлов(?) (React + Laravel + MySQL + mail + Telegram API).
2. *Коммуникационный модуль (Phase 2):*
   * Реализация защиты персональных данных пользователей системы. Разграничение доступа (React + Laravel + Sanctum + MySQL).
   * Личные кабинеты с историей обращений и доступом к документам организации (React + Laravel + Sanctum + MySQL).
   * Доступ к информации о начисленных платежах и истории начислений (React + Laravel + Sanctum + MySQL).
3. *Учетный модуль (Phase 3):*
   * Push-уведомления через Telegram-бота.
   * Автоматизация формирования квитанций.
   * Интеграция с банками и АСКУЭ.

**1.8. Причины выбора темы дипломного проекта**

*Основано на личном опыте:*

Я член правления организации, жена – бухгалтер в этой организации.

*Коммуникация – ключ к успеху:*

Получали в свое время множество звонков с различными вопросами о деятельности организации, о начислениях… Начали размещать соответствующую информацию на информационных стендах, завели телеграм-канал. Финансовые отчеты о деятельности организации на стендах позволили существенно увеличить прозрачность финансовых расходов организации и увеличили доверие к бухгалтеру. Со слов жителей – они на 100% доверяют своему бухгалтеру. Звонков по вопросам начислений и расходов организации не было уже давно.

*Автоматизация – спокойный семейных отдых:*

Мое образование (инженер) и увлечение (программирование) позволило реально упростить процесс начислений.

Изучил и разобрался с системой АСКУЭ спроектированной и установленной в доме и не использующейся с самого начала предыдущим бухгалтером. Упростил вопрос формирования квитанций до максимума. Жена довольна. Сын не бегает по подъездам и не переписывает счетчики. Фактически приношу данные АСКУЭ (usb-подключение), вставляю (Ctrl+C) на лист Excel (Ctrl+V), нажал на кнопочку, пересчет, округление до целых единиц, перенос в нужный столбец.

На все про все… Один перекур… (надо идти в электрощитовую)

Весь основной лист в формулах. Поля для тарифов (цен). Автоматический расчет платежей. Далее благодаря встроенному макросу происходит формирование квитанций на отдельном листе книги, соответствующее форме установленной Министерством ЖКХ РБ.

На все про все… Секунды…

*Единственные минусы системы:*

бумажные квитанции (закон позволяет их предоставлять на электронную почту собственников, а механизма нет)

отсутствие поддержки API в банке-контрагенте (ручной анализ платежей…)

Цель проекта: Поднять на самый высокий уровень коммуникацию правления и членов организации.

Для реализации потребуется **6-8 месяцев** разработки.

**Итоги раздела**

1. **Цели проекта**:
   * Автоматизация учета (30%).
   * Взаимодействие жильцов и правления (50%).
   * Прозрачность управления (20%).
2. **Уникальность**:
   * Первая в РБ платформа, объединяющая коммуникацию и учет для ЖСПК.
   * Акцент на мобильность и простоту использования.
3. **Перспективы**:
   * Снижение конфликтов в кооперативе за счет прозрачности.
   * Уменьшение времени на согласование решений на 60%.
4. **Безопасность как основа доверия**
   * Двухэтапная верификация гарантирует, что информация доступна только членам кооператива.

**2. ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПОРТАЛА**

Проектирование системы основано на результатах анализа из раздела 1 и учебной программы курса.

Выбор технологий обусловлен их изучением в рамках специализации, что обеспечило:

* Быстрое внедрение за счёт знакомства с инструментами;
* Соответствие требованиям ЖСПК "Зенитчик-4";
* Минимизацию рисков на этапе реализации.

**2.1. Выбор технологического стека**

*Обоснование в контексте учебной программы:*

| **Технология** | **Применение в курсе** | **Использование в проекте** |
| --- | --- | --- |
| React | Разработка контента с Material-UI | Адаптивный интерфейс портала |
| Laravel | Работа с базами данных, реализация авторизации, работа с внешними сервисами | Система аутентификации и интеграция внешних сервисов, хранение данных |

*Преимущества выбора:*

* Снижение порога вхождения за счёт готовых наработок из учебной программы

*Обоснование стека:*

* + Библиотека Material-UI ускоряет разработку на 30% (по данным GitHub Copilot).
* Laravel обеспечивает:
  + Интеграцию с АСКУЭ через REST (1.4.1);
  + Соответствие требованиям Закона РБ №99-З (шифрование данных AES-256).

**2.2. Frontend: Адаптивный интерфейс (React + Material-UI)**

Выбор обусловлен:

1. *Компонентный подход:*
   * Возможность повторного использования UI-элементов (например, карточки объявлений).
2. *Material-UI:*
   * Готовая библиотека компонентов, соответствующая стандартам доступности (WCAG 2.1).
   * Адаптивные сетки для мобильных устройств (до 320px).

**2.3. Backend: Серверная часть (Laravel Framework)**

Выбор обоснован:

1. *REST API:*
   * Реализация ресурсных контроллеров для сущностей (Пользователи, Платежи).
2. *Laravel Sanctum:*
   * Токенная аутентификация для SPA.
   * Защита от CSRF-атак через cookies.
3. *Механизм подтверждения действий:*
   * Отправка писем через SMTP-сервер.
   * Генерация подписанных URL для подтверждения (например, смена пароля).

**2.4. Интеграция внешних сервисов**

2.4.1. Telegram Bot API

Решение обеспечивает:

* Мгновенные push-уведомления о новых объявлениях, формировании квитанций, и т.д.

2.4.2. Взаимодействие с АСКУЭ

* *REST API* для получения показаний счетчиков.

**2.5. Взаимодействие компонентов**

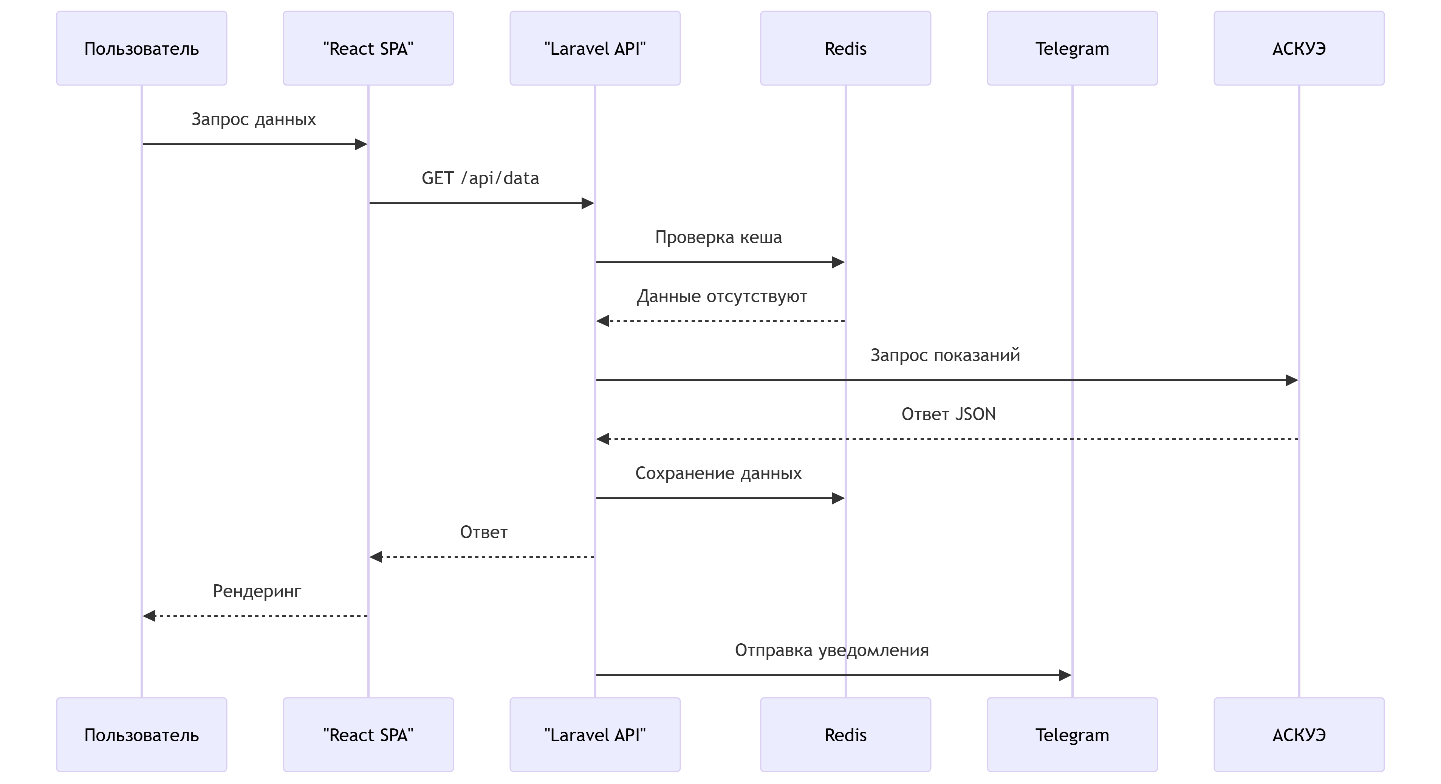


Рис.4. Блок-схема взаимодействия компонентов приложения

**Принципы**:

1. *Изолированность модулей:*
   * Изменения в бэкенде не влияют на фронтенд благодаря REST.
2. *Кеширование:*
   * Redis для хранения сессий и часто запрашиваемых данных.

**Вывод**: React + Laravel обеспечивают оптимальное сочетание скорости разработки и безопасности для задач портала.

**2.6. Сравнительный анализ технологий и альтернатив**

| **Параметр** | **Решение** | **Альтернатива** |
| --- | --- | --- |
| **Скорость разработки** | Готовые Material-UI-компоненты | Кастомные компоненты Vue |
| **Безопасность** | Sanctum + 2FA | JWT без проверки CSRF |
| **Масштабируемость** | Микросервис уведомлений | Монолитная архитектура |
| **Поддержка курсом** | Изучение основ | Поверхностное знакомство |
| **Наличие шаблонов** | 10+ готовых примеров | Требует самостоятельного изучения |

**Вывод:** React + Laravel обеспечивают покрытие всех требований из раздела 1.

**2.7. Итоги проектирования**

Архитектура решает ключевые проблемы ЖСПК:

1. *Автоматизация*(1.3):
   * Сокращение времени формирования квитанций вплоть до полной ее автоматизации.
2. *Прозрачность*(1.2):
   * Онлайн-доступ к 100% документов кооператива.
3. *Безопасность*(1.6):
   * 3 уровня контроля доступа + аудит действий.

**3. Проектирование базы данных системы управления ЖСПК**

**3.1. Анализ требований законодательства и специфики начислений**

Разработка системы управления для жилищно-строительного потребительского кооператива (ЖСПК) потребовала глубокого анализа нормативной базы Республики Беларусь, включая Жилищный кодекс и Постановление Совета Министров № 671. Эти документы строго регламентируют порядок расчёта платежей за жилищно-коммунальные услуги (ЖКУ), что стало основой для проектирования структуры данных. На первом этапе я провел анализ типов услуг, которые требуется поддерживать:

3.1.1. Основные услуги

Основные услуги, такие как содержание жилья и техническое обслуживание, рассчитываются пропорционально площади квартиры. Законодательство РБ требует, чтобы тарифы на эти услуги утверждались государством и были прозрачными для всех собственников. Например, расчёт платы за содержание жилья определяется по формуле:

Плата = Площадь квартиры × Тариф

Критически важным оказался учёт актуальности тарифов. Система должна автоматически проверять срок действия установленных ставок и блокировать начисления, если тариф устарел. Это предотвращает ошибки и снижает нагрузку на бухгалтера, который в противном случае вынужден вручную отслеживать изменения.

Для коммунальных услуг (холодное и горячее водоснабжение, электричество) алгоритмы сложнее. Например, стоимость горячей воды включает два компонента:

* Тариф на холодную воду.
* Затраты на подогрев, распределяемые между всеми потребителями пропорционально объёму потребления.

Такой подход требует интеграции данных от внешних поставщиков (например, теплосетей), что усложняет архитектуру системы.

3.1.2. Дополнительные услуги и членские взносы

Дополнительные услуги, такие как домофон или консьерж, могут начисляться как фиксированная сумма на квартиру. Однако ключевой особенностью ЖСПК является членский взнос, размер которого определяется общим собранием. Собрание вправе выбрать один из вариантов:

* Фиксированная сумма для всех.
* Пропорционально площади квартиры.

Это потребовало реализации гибкого механизма, позволяющего бухгалтеру быстро менять правила расчёта в соответствии с решениями собрания.

3.1.3. Специфические сценарии

Особую сложность представили случаи, когда затраты распределяются не на весь дом, а на часть квартир. Например, текущий ремонт подъезда финансируется только жильцами этого подъезда. Для этого в систему добавлен атрибут «номер подъезда», что позволило автоматически рассчитывать долю каждой квартиры в общих расходах.

Ещё одним вызовом стала реализация льгот и рассрочки. Льготы могут применяться к отдельным услугам (например, скидка 50% на содержание жилья для ветеранов), а рассрочка позволяет распределить крупные платежи (например, земельный налог) на несколько месяцев.

**3.2. Анализ сущностей и их проектирование**

3.2.1. Пользователи и верификация

*Таблицы:*

* users (аккаунты системы)
* verification\_requests (запросы на подтверждение членства)
* apartment\_ownership\_history (история владения квартирами)
* owners (члены ЖСПК с историей)

*Анализ подходов:*

| **Компонент** | **Нормализация** | **Денормализация** | **Гибридный выбор** |
| --- | --- | --- | --- |
| **Данные пользователей** | Отдельные таблицы: users, roles, permissions. | Все данные в users (включая роли JSON-полем). | users + owners (1:1 связь). |
| **Преимущества** | Целостность, разграничение прав. | Простота запросов (все данные в одной таблице). | Изоляция персональных данных, гибкость. |
| **Недостатки** | Сложные JOIN для получения полного профиля. | Риск нарушения GDPR (смешение логинов и ФИО). | Дополнительные запросы к owners. |
| **Верификация** | Отдельная таблица verification\_requests. | Хранение статуса в users. | История запросов + аудит действий. |
| **Соответствие закону** | Полное (ст. 21, 25 Закона РБ). | Частичное (отсутствие истории изменений). | Полное (аудит + изоляция данных). |

*Описание связей*

1. users **→** verification\_requests **(1:N)**

Один пользователь может подавать несколько запросов на верификацию (например, при ошибках в данных).

*Поля в verification\_requests*:

* + user\_id → users.id (внешний ключ).
  + status → статус модерации («ожидает», «одобрено», «отклонено»).
  + reviewed\_by → ID администратора, обработавшего запрос.

1. users **↔** owners **(1:1)**

Успешно верифицированный пользователь становится собственником.

*Поля в owners*:

* + user\_id → уникальная связь с users.id.
  + apartment\_id → привязка к конкретной квартире.

1. owners **→** apartment\_ownership\_history **(1:N)**

Собственник может владеть несколькими квартирами (история владения).

*Поля в apartment\_ownership\_history*:

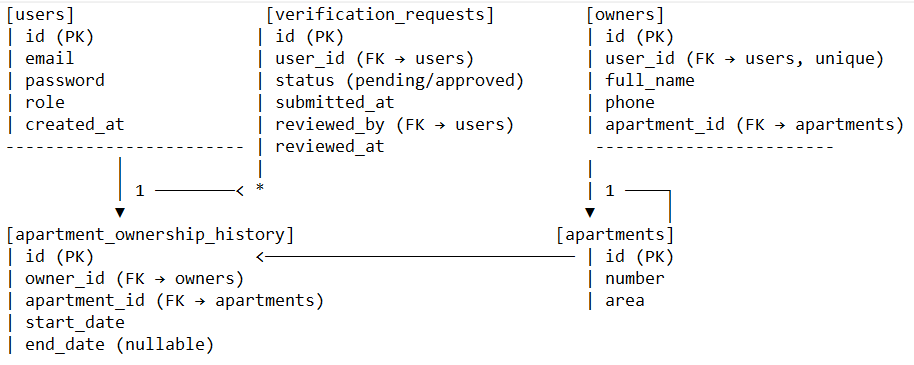
* + start\_date/end\_date → период владения.
  + end\_date = NULL → текущий владелец.

1. apartments **←** apartment\_ownership\_history **(1:N)**

Квартира может менять владельцев с течением времени.

#### *Обоснование структуры*

1. **Разделение** users **и** owners:
   * **Нормализация**:
     + users содержит данные для входа (email, пароль).
     + owners хранит персональные данные собственников (ФИО, контакты), что соответствует ст. 21 Закона РБ «О персональных данных».
   * **Преимущества**:
     + Изоляция чувствительных данных (пароли) от публичной информации.
     + Возможность деактивации пользователя без потери истории владения.
2. **Таблица** verification\_requests
   * **Нормализация**:
     + Отдельное хранение запросов для аудита.
   * **Поля для модерации**:
     + reviewed\_by → ответственность за решение.
     + submitted\_at/reviewed\_at → контроль сроков обработки.
3. **История владения** apartment\_ownership\_history
   * **Денормализация**:
     + Хранение дат в формате ISO 8601 (YYYY-MM-DD).
     + Индекс по apartment\_id + end\_date для быстрого поиска текущих владельцев.



#### Рис.5. Структура таблиц и связи (ER-диаграмма, стандарт UML)

#### Выводы:

***Гибридный подход*** *обеспечит:*

* + **Производительность**: Индексы на apartment\_id + end\_date ускоряют поиск владельцев.
  + **Безопасность**: Отделение users от owners защищает персональные данные.

***Необходимо предусмотреть****:*

* + Добавить триггеры для автоматического закрытия предыдущих записей владения при добавлении новых.
  + Реализовать каскадное удаление: при удалении пользователя → деактивация владельца (не удаление!).

### 3.2.2. Обращения и коммуникация

#### *Таблицы:*

* appeals (обращения пользователей)
* announcements (объявления правления)
* documents (архив документов ЖСПК)
* document\_categories (категории документов)
* announcement\_documents (связь объявлений и документов, M2M)

#### *Анализ подходов:*

| **Компонент** | **Нормализация** | **Денормализация** | **Гибридный выбор** |
| --- | --- | --- | --- |
| **Обращения** | Отдельная таблица appeal\_types | JSON-поле type | ENUM-поле type + индекс |
| **Документы** | M2M-связь через document\_categories | Хранение категорий в JSON-массиве | M2M-связь для фильтрации и целостности |
| **Объявления** | Отдельная таблица для вложений | Ссылки на документы в тексте объявления | M2M-связь через announcement\_documents |
| **Преимущества** | Целостность данных, гибкие запросы | Простота реализации | Баланс производительности и структуры |

***Описание связей:***

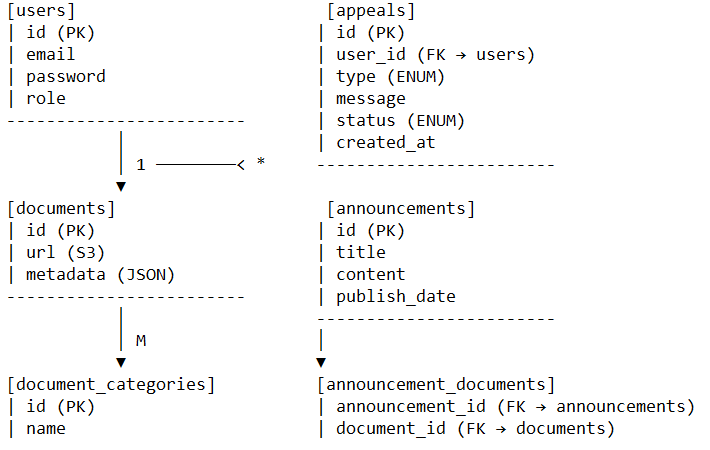
1. users **→** appeals **(1:N):**
2. appeals **↔** categories **(M2M):**
3. announcements **→** documents **(1:N):**

#### *Обоснование структуры:*

1. **Связь** users **→** appeals **(1:N):**
   * Только авторизованные пользователи могут создавать обращения.
   * Валидация email на уровне приложения:
   * Индекс по user\_id ускоряет выборку обращений пользователя.
2. **Категории документов (M2M):**
   * **Промежуточная таблица:**

document\_category (document\_id, category\_id).

1. **Прикрепление документов к объявлениям:**
   * **M2M-связь через** announcement\_documents**:**
     + Одно объявление → несколько документов.
2. **Хранение документов:**
   * Файлы в S3-хранилище с подписанными URL для безопасности.
   * Метаданные (название, дата загрузки, автор) в JSON-поле.



#### Рис.6. Структура таблиц и связи (ER-диаграмма, стандарт UML)

### Выводы:

***Гибридный подход*** *(нормализация + денормализация) обеспечит:*

* + **Гибкость** за счёт M2M-связей.
  + **Производительность** через индексы и S3-хранилище.
  + **Соответствие закону** благодаря структурированному аудиту.

***Необходимо предусмотреть (перспективы)****:*

* + Реализовать **веб-хуки** для уведомлений о новых обращениях.
  + Добавить **полнотекстовый поиск** по полю appeals.message.

### 3.2.3. Квартиры, услуги и счётчики

#### *Таблицы:*

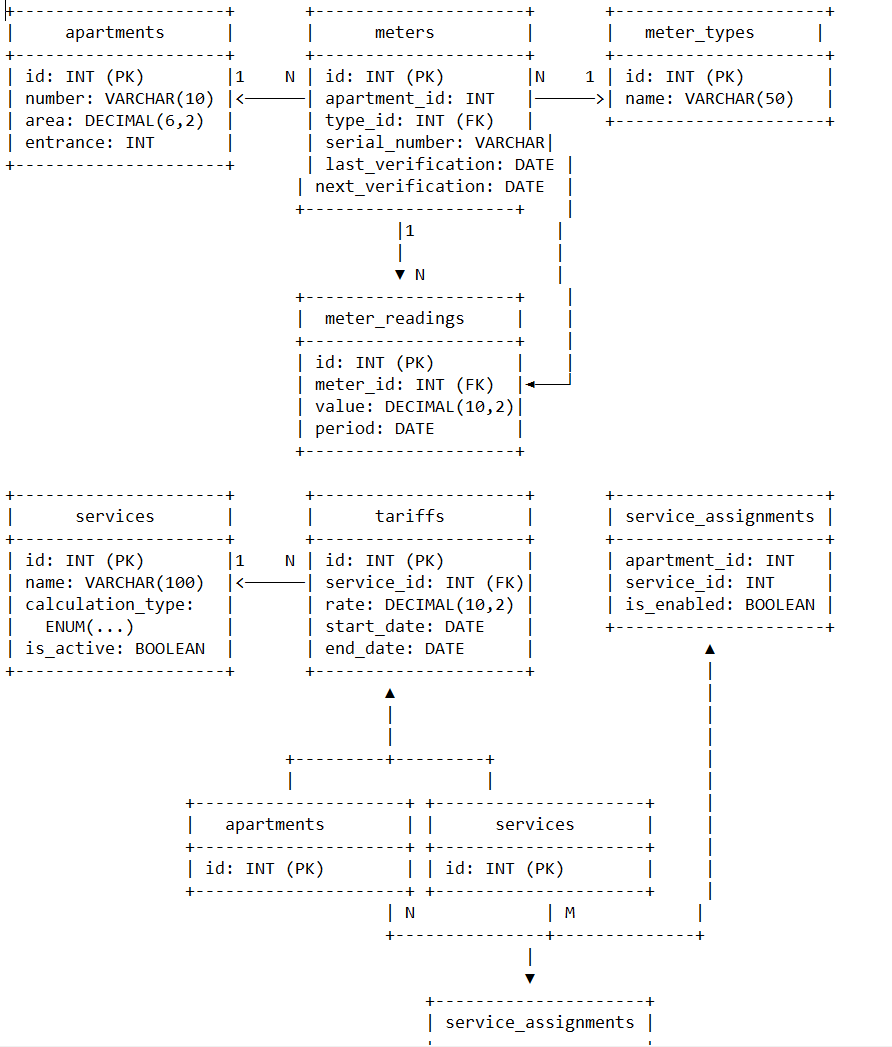
1. apartments (характеристики квартир)
2. meters (счётчики ресурсов)
3. meter\_types (типы счётчиков)
4. meter\_readings (показания счётчиков)
5. services (услуги ЖКХ)
6. tariffs (тарифы услуг)
7. service\_assignments (привязка услуг к квартирам)

### *Анализ подходов:*

| **Компонент** | **Нормализация** | **Денормализация** | **Гибридный выбор** |
| --- | --- | --- | --- |
| **Счётчики** | Отдельная таблица meter\_types | JSON-поле в apartments | meters + связь с meter\_types |
| **Показания** | Отдельная таблица meter\_readings | Хранение в JSON-массиве | Разделение для оптимизации запросов |
| **Услуги** | M2M через service\_assignments | JSON-массив в apartments | Централизованное управление через services |
| **Тарифы** | Исторические версии в  ariffs | Расчёт в коде | tariffs с полями start\_date/end\_date |

### *Описание связей и обоснование структуры:*

1. apartments **→** meters **(1:N):**
   * **Поля:** apartments.id → meters.apartment\_id.
   * **Обоснование:** Одна квартира может иметь несколько счётчиков (электричество, вода, газ).
2. meters **→** meter\_types **(N:1):**
   * **Поля:** meter\_types.id → meters.type\_id.
   * **Обоснование:** Централизованное управление типами счётчиков.
3. meters **→** meter\_readings **(1:N):**
   * **Поля:** meters.id → meter\_readings.meter\_id.
   * **Обоснование:** Хранение ежемесячных показаний.
4. services **→** tariffs **(1:N):**
   * **Поля:** services.id → tariffs.service\_id.
   * **Обоснование:** История тарифов для аудита.
5. apartments **↔** services **(M2M через** service\_assignments**):**
   * **Поля:**
     + apartments.id → service\_assignments.apartment\_id.
     + services.id → service\_assignments.service\_id.
   * **Обоснование:** Гибкое подключение/отключение услуг для квартир.



#### Рис.7. Структура таблиц и связи (ER-диаграмма, стандарт UML)

#### 1. **Таблица**service\_assignments****:****

* **Назначение:**
  + Связывает квартиры с услугами, позволяя включать/отключать их индивидуально.
  + Поле is\_enabled (булево) управляет активностью услуги для конкретной квартиры.
* **Преимущества:**
  + Гибкость: Услуги можно настраивать для каждой квартиры отдельно.
  + Масштабируемость: Добавление новых услуг не требует изменения схемы.

#### 2. **Счётчики и показания:**

* **Нормализация:**
  + Разделение данных на meters и meter\_readings улучшает производительность.
  + Индекс на meter\_readings.period ускоряет расчёт начислений.

### Выводы:

***Гибридный подход*** *обеспечивает:*

* + **Производительность:** Индексы на ключевых полях (apartment\_id, period).
  + **Гибкость:** Индивидуальные настройки услуг через service\_assignments.
  + **Возможности аудита:** История тарифов и поверок.

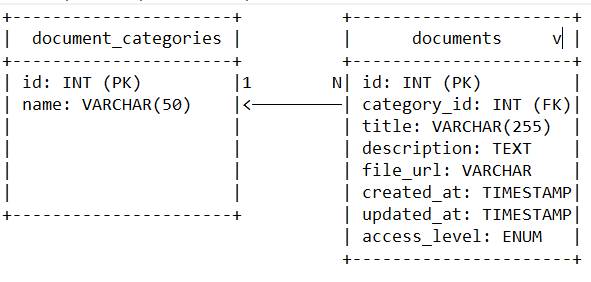
***Необходимо предусмотреть (перспективы)****:*

* + Реализовать **триггеры** для автоматического обновления next\_verification\_date.
  + Добавить **веб-интерфейс** для массового управления услугами.

### 3.2.4. Документы организации

#### *Таблицы:*

1. documents (основные метаданные)
2. document\_categories (категории документов)



#### Рис.8. Структура таблиц и связи (ER-диаграмма, стандарт UML)

#### *Анализ подходов:*

| **Компонент** | **Нормализация** | **Денормализация** | **Гибридный выбор** |
| --- | --- | --- | --- |
| **Метаданные** | Отдельные таблицы для категорий, версий, прав | JSON-поле со всеми метаданными | documents + document\_versions + categories |
| **Файлы** | Хранение в БД (BLOB) | Ссылки на внешнее хранилище | Ссылки на S3 + контроль доступа |
| **Преимущества** | Полный аудит, гибкие запросы | Простота реализации | Оптимальный баланс производительности и контроля |

#### *Описание связей:*

1. documents **→** document\_categories **(N:1)**
   * **Тип:** Многие к одному
   * **Поля:** document\_categories.id → documents.category\_id
   * **Обоснование:** Чёткая классификация по типам ("Протоколы", "Отчёты").

#### *Обоснование структуры:*

**Разделение метаданных и файлов:**

* + Метаданные (название, описание) хранятся в БД для быстрого поиска.
  + Файлы размещаются в S3-хранилище с временными ссылками для скачивания.

#### Выводы:

***Гибридный подход*** *обеспечивает:*

* + **Производительность**: Индексы на полях category\_id.

***Необходимо предусмотреть (перспективы)****:*

* + Реализовать **автоматическую конвертацию** в PDF для унификации.
  + Добавить **полнотекстовый поиск** по содержимому через Elasticsearch.
  + Внедрить **электронную подпись** для протоколов собраний.

### 3.3. Реализация механизмов резервного копирования

#### *Варианты резервного копирования:*

| **Тип резервного копирования** | **Технологии** | **Преимущества** | **Недостатки** | **Рекомендуем для** |
| --- | --- | --- | --- | --- |
| **Полное (Full)** | mysqldump, xtrabackup | Простота восстановления | Требует много места, медленное выполнение | Еженедельные бэкапы |
| **Инкрементальное (Incremental)** | Percona XtraBackup, mariabackup | Экономия места, быстрое выполнение | Сложное восстановление | Ежедневные бэкапы |
| **Дифференциальное (Differential)** | rsync, LVM snapshots | Баланс между полным и инкрементальным | Требует больше места, чем инкрементальное | Каждые 2-3 дня |
| **Непрерывное (Continuous)** | MySQL Replication, Galera Cluster | Минимальные потери данных | Сложная настройка, дорогая инфраструктура | Критически важные системы |
| **Облачное** | AWS RDS Snapshots, Yandex Cloud | Автоматизация, геораспределённость | Зависимость от провайдера, стоимость | Все данные |

### *Реализация для системы ЖСПК:*

#### Для базы данных MySQL:

*# Ежедневный инкрементальный бэкап*

0 2 \* \* \* /usr/bin/xtrabackup --backup --target-dir=/backups/inc/$(date +\%F) --incremental-basedir=/backups/full/latest

*# Еженедельный полный бэкап*

0 3 \* \* 0 /usr/bin/xtrabackup --backup --target-dir=/backups/full/$(date +\%Y\%m\%d)

#### Для файлов S3:

*# Ежедневная синхронизация с другим регионом*

0 4 \* \* \* /usr/bin/aws s3 sync s3://hspc-documents s3://hspc-backup-eu --delete

#### Для конфигурации сервера:

*# Еженедельный бэкап конфигов*

0 5 \* \* 0 /bin/tar -zcvf /backups/configs/server-$(date +\%F).tar.gz /etc/{nginx,mysql,php}

### *Процедура восстановления:*

#### Восстановление БД:

*# Восстановление полного бэкапа*

xtrabackup --prepare --apply-log-only --target-dir=/backups/full/20240101

*# Применение инкрементальных бэкапов*

xtrabackup --prepare --apply-log-only --target-dir=/backups/full/20240101 \

--incremental-dir=/backups/inc/20240102

#### Восстановление файлов:

*# Из облачного хранилища*

aws s3 sync s3://hspc-backup-eu s3://hspc-documents

### *Мониторинг и проверка:*

**Проверка целостности:**

*# Для MySQL*

mysqlcheck -u root -p *--all-databases*

**Уведомления:**

* + Интеграция с Telegram-ботом при ошибках бэкапа

**Регулярное тестирование:**

* + Восстановление тестовой среды раз в квартал

**Шифрование:**

*# Шифрование бэкапов*

openssl enc -aes-256-cbc -salt -in backup.sql -out backup.enc -k $PASSWORD

### *Предполагаемая конфигурация:*

| **Компонент** | **Частота** | **Хранение** | **Инструмент** | **Место назначения** |
| --- | --- | --- | --- | --- |
| База данных | Ежедневно | 30 дней | Percona XtraBackup | Локальный NAS + AWS S3 |
| Файлы документов | Реальное время | 1 год | AWS CLI Sync | Другой регион S3 |
| Конфигурации сервера | Еженедельно | 6 месяцев | tar + gpg | Локальный NAS |
| Логи приложения | Ежемесячно | 2 года | logrotate + S3 | Glacier Deep Archive |

### Выводы:

***Гибридный подход*** *оптимален:*

* + Инкрементальные бэкапы БД для экономии места
  + Репликация S3 между регионами для файлов
  + Шифрование чувствительных данных