

Тема: Разработка аналитического приложения для автоматизации учета арендных помещений арендодателя

4.1. Анализ области применения программного приложения

4.1.1. Описание предметной области

Программное средство автоматизирует учет арендных помещений арендодателя: ведение справочников, договоров, ежемесячные начисления в BYN с НДС, учет оплат, расчет задолженности и пени, а также построение аналитических отчетов и дашбордов.

Цель — снизить ручные операции, ошибки и задержки, повысить прозрачность показателей.

При росте фонда помещений и числа арендаторов традиционные способы (бумага, Excel) становятся недостаточными: возрастают операционные издержки, риски ошибок при расчетах НДС и индексаций, сложность контроля статусов договоров, просрочек, простоев и заполняемости. Это снижает качество сервиса и оперативность управленческих решений.

Процесс начинается с поступления запроса от потенциального арендатора или с активного предложения со стороны арендодателя. После квалификации запроса организуется просмотр объекта и сбор технических требований (площадь, мощность электросети, водоснабжение, потолки, загрузка, требования к пожарной безопасности). По итогам осмотра формируется коммерческое предложение с вариантами ставки (BYN/м²/мес или фикс) и условиями (НДС, депозит, срок оплаты, каникулы, индексация). На период переговоров помещение может резервироваться. После согласования условий готовится и подписывается договор аренды с приложениями (план БТИ, акт осмотра, схема помещения). В дату начала аренды подписывается акт приема-передачи (с фиксацией показаний счетчиков, комплектов ключей). Далее ежемесячно формируются счета, производятся оплаты, выполняются индексации согласно правилам, ведется переписка и акты сверок. По окончании срока — пролонгация или расторжение; при выезде — акт возврата помещения и закрытие обязательств.

Документация, используемая при сдаче. Коммерческое предложение/оферта, протокол согласования условий, договор аренды, дополнительные соглашения (об индексации, пролонгации, изменении ставки/площади), технический паспорт, акт приема-передачи (первичный и при возврате), акты выполненных работ (при комплексных услугах), счета и счета-фактуры (с НДС 20%), платежные документы, акты сверки, переписка по ключевым вопросам.

Для ритейла — схема витрины и требования к вывеске; для складов — допуски по нагрузкам и рампы; для офисов — класс отделки и доступ вСКУД.

Типы помещений и ключевые характеристики. Для офисов важны класс здания и отделки, планировка, доступ вСКУД, наличие переговорных, серверной, мощность электросети, кондиционирование/вентиляция, высота потолков. Для ритейла критичны фронт витрины и витринные окна, трафик, видимость, доступность для погрузки, режим работы ТЦ, требования к вывеске, возможность мокрой точки. Для складов учитываются тип (сухой/холодильный), температура, высота стеллажей, напольные нагрузки, наличие док-станций и рамп, охрана и видеонаблюдение, подъезд фур. Для парковок — тип места (крытое/открытое), ширина, доступ, охрана; для рекламных поверхностей — формат/площадь, размещение, сроки кампаний. Эти характеристики используются как атрибуты справочников и как измерения в аналитике (фильтрация, группировки).

Ключевые объекты предметной области.

- 1 Помещение :
 - a. единица арендуемой площади;
 - b. тип (офис/ритейл/склад/паркинг/реклама);
 - c. адрес/здание;
 - d. этаж;
 - e. площадь (м²);
 - f. базовая ставка (BYN/м²/мес или фикс);
 - g. статус (свободно/сдано/резерв);
 - h. доступность с даты.
- 2 Арендатор
 - a. юр/физ лицо(наименование, УНП, контакты, банковские реквизиты, договорной e-mail)
- 3 Договор аренды :
 - a. номер/дата;
 - b. период действия;
 - c. Помещение;
 - d. база тарификации (м²/фикс);
 - e. валюта (BYN), НДС (ставка, по умолчанию 20%);
 - f. депозит; правила пени;
 - g. срок оплаты.
- 4 Индексация/тариф —
 - a. правило пересмотра ставки ;

- б. дата вступления;
- с. история изменений.

5 Пени — санкции за просрочку (0.1% в день), база расчета, период просрочки.

Типы помещений (принятые в проекте).

- 1 Офисы (кабинеты), ставка BYN/м²/мес.
- 2 Ритейл (торговые площади), ставка BYN/м²/мес; опционально оборотная надбавка.
- 3 Склады ставка BYN/м²/мес.

Основные бизнес-процессы.

- 1 Ведение справочников помещений и арендаторов.
- 2 Заключение/продлонгация/расторжение договоров (контроль сроков).
- 3 Ежемесячные начисления (аренда, доп. услуги) в BYN с НДС.
- 4 Учет оплат, автоматическая сверка, расчет задолженности и пени.
- 5 Индексации ставок согласно правилам договора.
- 6 Аналитика: заполняемость фонда, доходность, дебиторка (aging), динамика ставок/индексаций, простой.

KPI и аналитика.

- 1 Заполняемость (% по фонду и по объектам/типам).
- 2 Доход/прибыль по периодам, по помещениям/арендаторам (с НДС/без НДС).
- 3 Дебиторка: остатки и просрочки 0–30/31–60/61–90/90+ дней, DSO.
- 4 Средняя ставка BYN/м² и динамика индексаций; средний простой помещения.

Участники процесса (роли). Со стороны арендодателя: менеджер по аренде (лид-менеджмент и переговоры), технический специалист (подготовка помещения, ввод показаний), бухгалтер/финансист (выставление счетов, учет оплат, сверки), юрист (правовая экспертиза договоров), руководитель (утверждение условий). Со стороны арендатора: инициатор запроса, финансовый представитель и подписант договора.

Бизнес-правила и ограничения. Базовая валюта расчетов — BYN; НДС по умолчанию — 20% (конфигурируемо). Периодичность начислений — ежемесячно; срок оплаты определяется договором (день месяца). Пени начисляются на просроченную сумму из расчета 0.1% в день (настраиваемо).

Добавлено примечание ([КИ1]):

Добавлено примечание ([КИ2]): Пока что это только набросок предметной области. Неплохой, не скрою, но надо больше информации о том как происходит сдача помещений, какие виды помещений имеются ввиду, какие виды характеристик используются для каждого вида, какая документация используется для сдачи в аренду и т.д.

Индексации применяются по событиям (дата вступления, коэффициент) и фиксируются в истории. Для каждого помещения поддерживается статус и календарь доступности; бронь ограничена по времени. Все документы и события жизненного цикла должны иметь сквозные идентификаторы и храниться в системе с возможностью аудита.

4.1.2. Обзор существующих аналогов программного приложения

Рассматривались покрытие доменной логики (помещения, договоры, тарифы/индексации, начисления, оплаты, пени), возможности аналитики и дашбордов, степень автоматизации ежемесячных процессов (BYN+НДС, график начислений, сверка оплат), интеграции и API, сложность внедрения и стоимость владения, расширяемость и локализация.

1С: Аренда и управление недвижимостью

Решение класса «учёт/документооборот» с глубокой локализацией. Поддерживает справочники помещений и контрагентов, договоры, выставление счетов и печатные формы, интеграции с бухгалтерским учетом на платформе 1С. Предусмотрены типовые отчеты и регламентные операции. При этом интерактивная аналитика и дашборды ограничены возможностями платформы, а серьёзные доработки требуют квалификации по 1С и ведут к «vendor lock-in». Веб-интерфейс и внешние REST-интеграции уступают современным открытым стекам.

SAP RE-FX

Модуль корпоративного уровня для управления недвижимостью, тесно интегрирован с SAP FI/CO и модулем активов. Богатая модель договоров, условий индексации, учёт по международным стандартам, развитые отчеты через SAP BW/BI. Подходит крупным холдингам. Недостатки для средних организаций: высокая стоимость лицензий и внедрения, длительные сроки проектов, избыточность функционала, зависимость от экосистемы SAP.

Локальные accounting-решения

Небольшие коробочные продукты/самописные системы, обеспечивающие первичный учёт: справочники, договоры, счета, простые отчеты. Преимущество — быстрый старт и низкий порог входа. Недостатки — фрагментация данных, слабые REST-интерфейсы и отсутствие

стандартизированной модели аналитики (aging, KPI), ограниченная масштабируемость, зависимость от конкретного поставщика/исполнителя.

BI-шаблоны (Power BI/Zoho и др.)

Готовые визуализации и панели на основе импортированных данных. Сильная сторона — интерактивные графики, фильтры, публикация. Однако отсутствует доменная логика аренды: нет автоматизации начислений, сверки оплат, расчета пени, индексаций. BI-шаблоны зависят от качества источников и не решают задачу единой транзакционной модели.

В таблице 1 будет продемонстрировано сравнение аналогов.

Таблица 1 – сравнение аналогов программного приложения

Критерий	1С	SAP RE-FX	Локальные	BI-шаблоны
Доменная логика аренды	Широкая, но жестко вшитая в платформу	Полная enterprise-модель	Базовая, разнородная	Отсутствует
Аналитика/дашборды	Ограниченная без внешних BI	Сильная через BW/BI (дорого)	Простые отчеты	Сильная визуализация без транзакций
Автоматизация (начисления, сверка, пени)	Есть, но гибкость ограничена	Полная (сложно/дорого)	Частичная/ручная	Нет
REST-API/интеграции	Ограничены экосистемой 1С	Сильные, но в рамках SAP	Слабые/разные	Импорт/коннекторы без доменной логики
Стоимость и сроки внедрения	Средние	Высокие	Низкие/средние	Низкие (без учета источников)

4.2. Разработка и анализ требований к программному приложению

4.2.1. Анализ и моделирование бизнес-процессов (BPMN)

Основной сквозной процесс. Базовый процесс предметной области — сдача в аренду с фиксацией статусов на всем жизненном цикле: обработка лида, осмотр и согласование условий, резерв помещения, подписание договора, акт

приема-передачи, ежемесячные начисления (BYN+НДС), прием и разнесение оплат, контроль задолженности и пени, индексации ставок, пролонгация/изменение условий либо расторжение, акт возврата и закрытие обязательств. Именно этот процесс определяет ключевые события и состояния (свободно → резерв → сдано → на расторжении → освобождено) и служит основой для аналитики.

Состав основных бизнес-процессов учета. В рамках системы выделяются: ведение справочников (помещения, арендаторы, тарифы/НДС), управление договорами и статусами, планирование и исполнение ежемесячных начислений, прием/импорт и разнесение оплат, расчет задолженности и пени, применение индексаций, формирование счетов/счетов-фактур и актов, ведение переписки и актов сверки, аналитика KPI (заполняемость, доход, дебиторка, DSO), подготовка отчетов/дашбордов и экспорт, импорт/валидация данных (CSV/Excel), а также закрытие периода. Эти процессы моделируются в BPMN как взаимосвязанные подпроцессы ядра «Сдача в аренду» и поддерживающего контура «Аналитика».

В рамках проектирования аналитического приложения мы рассмотрели два состояния процесса подготовки аналитической отчетности в предметной области: текущее (AS-IS) и целевое (TO-BE). Текущее состояние отражает практику, при которой запрос на отчет инициируется пользователем и приводит к сбору разрозненных источников: Excel-таблиц с оперативными данными и банковских выписок. Набор файлов передается аналитику по электронной почте либо иным неформальным каналам. Аналитик вручную проверяет полноту и корректность, выполняет очистку, фильтрацию и объединение данных, затем рассчитывает ключевые показатели с использованием формул и сводных таблиц. Финальный отчет верстается вручную и распространяется в виде Excel/PDF. Такой подход подвержен задержкам, ошибкам согласования, дублированию и снижает воспроизводимость расчетов.

Целевое состояние описывает процесс в архитектуре будущего решения. Запрос на отчет формализуется как запуск регламента: оператор импортирует справочные и оперативные наборы в систему, валидация выполняется по заданным правилам, после чего данные загружаются в единое хранилище (PostgreSQL) и актуализируют материализованные витрины. Система автоматически пересчитывает KPI — заполняемость фонда, доход (база/итог с НДС), дебиторскую задолженность с разбиением по срокам, DSO, среднюю ставку и простой — и формирует дашборд и типовые отчеты. Пользователь

получает уведомление о готовности, просматривает результаты с фильтрами и дрилл-дауном до договора и платежа, при необходимости экспортирует материалы в PDF/Excel. Исключительные ситуации (некорректный файл, несопоставленные записи) фиксируются и обрабатываются в отдельной ветке, не блокируя публикацию витрин. Такой процесс обеспечивает предсказуемость сроков, прозрачность методик и воспроизводимость аналитики.

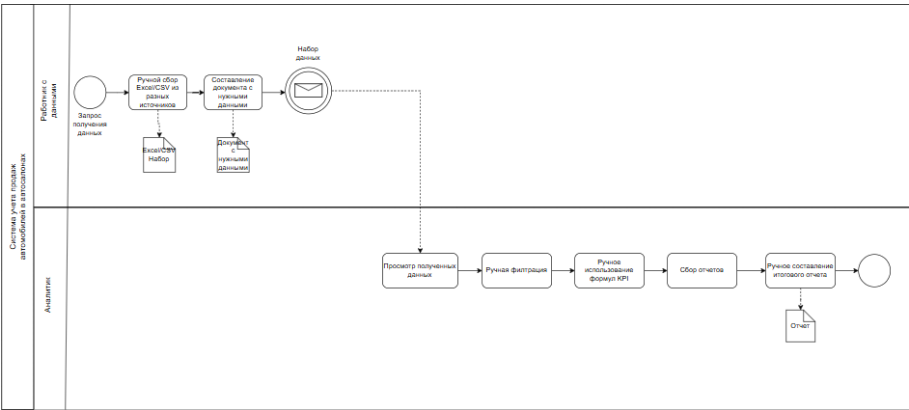


Рисунок 4.1 – диаграмма AS-IS

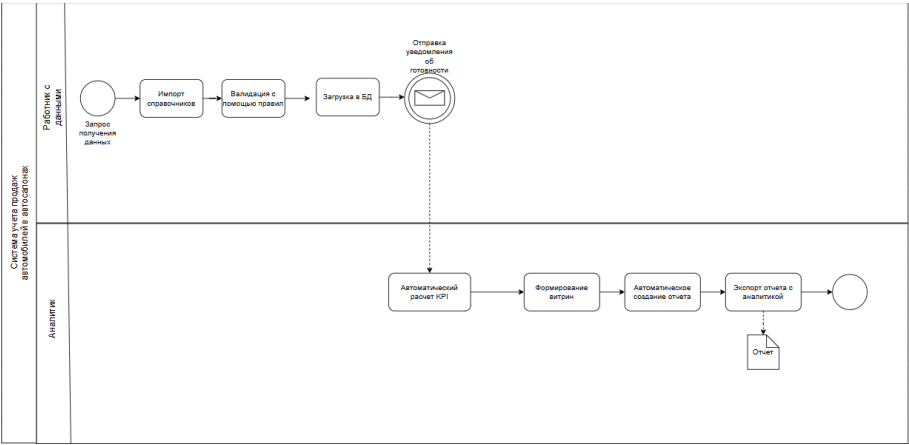


Рисунок 4.2 – диаграмма TO BE

Добавлено примечание ([КИЗ]): Нет такого участника бизнес-процесса как система.

4.2.2. Требования к программному приложению

Функциональные требования

Актеры. Администратор (управление доступом и настройками); Менеджер/Юрист (договора и статусы); Бухгалтер/Оператор (начисления, счета, платежи); Аналитик (витрины, отчеты, KPI); Руководство (просмотр дашбордов).

UC-01. Ведение справочных источников данных

Цель: поддерживать актуальные измерения для аналитики и расчетов. Первичный

актер: Администратор/Оператор. Предусловия: пользователь авторизован с правом CRUD. Триггер: изменение фонда/контрагента/тарифов.

Основной поток: Актер создает/редактирует записи «Помещение», «Арендатор», «Тариф/индексация», «НДС-настройки»; Система валидирует обязательные поля (тип, площадь, УНП и пр.), обеспечивает уникальность кодов/УНП, сохраняет изменения с аудитом и актуализирует справочные измерения для витрин. Постусловия: данные доступны для договоров, расчетов и аналитики.

UC-02. Управление договорами и статусами

Цель: оформить и сопровождать договор аренды в жизненном цикле. Актер: Менеджер/Юрист. Предусловия: в справочниках есть помещение и арендатор. Триггер: согласованные условия аренды.

Основной поток: Актер создает карточку договора (период, база тарификации, ставка в BYN, НДС=20% по умолчанию, срок оплаты, депозит, правила пени/индексации); Система присваивает номер, валидирует даты и связи; по подписанию меняет статус помещения на «сдано», договора — «действует». Допускаются операции редактирования, пролонгации, расторжения с фиксацией историй и актов. Исключения: конфликт дат/пересечение площадей — система блокирует сохранение и указывает причину.

UC-03. Ежемесячные начисления и счета

Добавлено примечание ([КИ4]): В таком формате конечно функциональные требования не принимаются.
Непонятно вообще кто чего делает.

Цель: рассчитать обязательства арендаторов. Актер: Бухгалтер/Оператор. Предусловия: есть действующие договоры. Триггер: таймер «ежемесячно» либо ручной запуск периода.

Основной поток: Система генерирует график начислений; рассчитывает базу ($m^2 \times \text{ставка}$ либо фикс) и НДС (20% по умолчанию, конфигурируемо); создает документы «Начисление» и «Счет»; предоставляет рассылку счетов. При изменении ставки/индексации выполняется пересчет текущего/следующих периодов с протоколированием. Постусловия: сформирован список счетов к оплате, доступный для сверки и аналитики дохода.

UC-04. Обработка платежей

Цель: фиксировать поступления и сопоставлять их с начислениями. Актер: Бухгалтер/Оператор. Предусловия: сформированы счета. Триггер: импорт банковской выписки (CSV/Excel) или ручной ввод.

Основной поток: Актер импортирует файл; Система валидирует формат, пытается авторазнести платежи по номеру счета/договору/сумме/дате; поддерживает частичные оплаты, переплаты (в аванс), возвраты/перерасчеты; нерешенные записи попадают в очередь ручного сопоставления. Постусловия: остатки по начислениям обновлены; журнал оплат пополнен.

UC-05. Дебиторская задолженность и пени

Цель: контролировать просрочку и начислять санкции. Актер: Бухгалтер/Аналитик. Триггер: ежедневный регламент.

Основной поток: Система рассчитывает aging (0–30/31–60/61–90/90+) и DSO; начисляет пени на просроченную сумму из расчета 0.1%/день (настраиваемо) с записью в журнал пени; формирует уведомления/эскалации по порогам. Постусловия: витрины задолженности и пени обновлены, доступны отчеты.

UC-06. Аналитика и отчеты

Цель: предоставить управленческую аналитику. Актер: Аналитик/Руководство. Предусловия: обновлены витрины. Триггер: запрос пользователя.

Основной поток: Система рассчитывает KPI (заполняемость, доход base/total, дебиторка, DSO, средняя ставка, простой, пени), строит отчеты и визуализации; пользователь применяет фильтры (период, объект, тип

помещений, арендатор), выполняет дрилл-даун до договора/помещения/платежа; доступен экспорт PDF/Excel. Альтернативы: публикация через REST API /analytics/* для внешних панелей.

UC-07. Дашборд KPI

Цель: получить свод ключевых показателей в реальном времени. Актер: Руководство/Аналитик. Основной поток: Система отображает KPI-панель с сохраненными представлениями и пресетами фильтров; обеспечивает автообновление согласно SLA обновления витрин.

UC-08. Управление доступом и аудит

Цель: безопасная эксплуатация. Актер: Администратор. Основной поток: создание пользователей и ролей (Администратор, Бухгалтер/Оператор, Аналитик); назначение прав на CRUD и отчеты; ведение журнала входов и аудита действий по договорам, начислениям и оплатам.

UC-09. Импорт/экспорт и шаблоны

Цель: стандартизировать обмен данными. Актер: Оператор/Аналитик. Основной поток: генерация валидируемых CSV/Excel-шаблонов для справочников и оплат; импорт с протоколом ошибок; экспорт отчетов и витрин.

UC-10. Прием лида и квалификация

Цель: зафиксировать обращение и оценить релевантность. Актер: Менеджер по аренде. Предусловия: доступ к модулю лидов. Триггер: входящее обращение (звонок/почта/форма).

Основной поток: создание карточки лида (контакты, требуемая площадь/тип, сроки); присвоение статуса («новый», «в работе», «закрыт»); планирование показа; фиксация источника. Постусловия: лид привязан к потенциальным помещениям.

UC-11. Показ и сбор требований

Цель: уточнить потребности арендатора. Актер: Менеджер/Тех.специалист. Триггер: согласованная дата показа.

Основной поток: фиксация параметров (мощность, мокрые точки, высота, доступ, график работ); прикрепление фото/плана БТИ; отметка результата

(«интерес подтвержден/отказ»). Постусловия: данные доступны для коммерческого предложения.

UC-12. Резервирование помещения

Цель: временно закрепить помещение. Актер: Менеджер. Предусловия: подтвержден интерес. Триггер: договоренность о брони.

Основной поток: установка статуса помещения «резерв» с датой истечения; генерация подтверждения брони; уведомление юриста/бухгалтерии. Исключения: пересечение с действующим договором — блокировка. Постусловия: бронь учитывается в отчетах доступности.

UC-13. Подписание договора

Цель: оформить договор и перевести статусы. Актер: Менеджер/Юрист. Предусловия: заведены арендатор и помещение. Триггер: согласованы условия.

Основной поток: генерация проекта договора и приложений; согласование; е-подпись или загрузка подписанного; перевод договора в «действует», помещения — в «сдано». Постусловия: доступен запуск акта приема-передачи.

UC-14. Акт приема-передачи

Цель: зафиксировать передачу помещения. Актер: Менеджер/Тех.специалист. Триггер: дата начала аренды.

Основной поток: заполнение акта (показания счетчиков, ключи, состояние); прикрепление фото; подписание; сохранение в архив договора. Постусловия: старт ежемесячных начислений.

UC-15. Пролонгация и изменение условий

Цель: продлить срок или скорректировать параметры. Актер: Менеджер/Юрист. Триггер: приближение даты окончания или инициатива сторон.

Основной поток: подготовка допсоглашения; внесение изменений (ставка, площадь, срок, индексация); пересчет будущих начислений; уведомление заинтересованных ролей. Постусловия: история версий обновлена.

UC-16. Инициирование расторжения

Цель: корректно завершить договор. Актер: Менеджер/Юрист. Триггер: заявление или окончание срока.

Основной поток: перевод договора в «на расторжении», установка даты T, расчет финальных обязательств, планирование акта возврата. Постусловия: сформирован чек-лист закрытия.

UC-17. Акт возврата и закрытие обязательств

Цель: зафиксировать возврат помещений и закрыть финансы. Актер: Менеджер/Бухгалтер. Триггер: дата выезда.

Основной поток: оформление акта возврата; финальный расчет (начисления/пени/депозит); погашение/возврат депозита; перевод статусов: договор «закрыт», помещение «освобождено». Постусловия: договор архивирован, отчеты обновлены.

UC-18. Уведомления и синхронизация статусов

Цель: обеспечить своевременные действия участников. Актер: Система (сервис уведомлений), все роли — получатели. Триггер: бизнес-события и таймеры.

Основной поток: рассылка уведомлений о брони (истечение), счетах, просрочке, актах, индексациях, расторжении; синхронизация статусов помещения и договора по правилам раздела

Нефункциональные требования

1. Архитектура:
 - a. клиент-сервер;
 - b. REST API (JSON); бизнес-логика только на сервере.
2. Совместимость:
 - a. Windows 8+
 - b. Android 10+
 - c. iOS 13+ (веб/PWA).
3. Данные:
 - a. реляционная БД;
 - b. 3НФ;
 - c. разделение справочных и оперативных данных.
4. Производительность:
 - a. медиана ответа $API \leq 500$ мс;

- б. отчет ≤ 5 с на наборах до 100 тыс. строк;
- с. фоновые задачи не блокируют UI.

5. Безопасность:

- а. авторизация (JWT/куки);
- б. хэширование паролей;
- с. роли;

журнал входов.

Локализация:

русский UI и подписи на диаграммах;
BYN и НДС 20% по умолчанию (конфигурируемо).

Технологические решения и паттерны

Паттерны: Repository + Unit of Work; Strategy (пени/индексации); Factory (создание начислений/счетов); Adapter (импорт CSV/Excel); Specification (фильтрация отчетов/KPI).

Интеграции: импорт банковских выписок (CSV/Excel); e-mail-уведомления (опц.).

Отчеты

Заполняемость по объектам и типам помещений (структура, динамика).

Доходность по арендаторам/помещениям/периодам (с НДС/без НДС).

Дебиторская задолженность: свод по aging с детализацией до договора/помещения.

История ставок/индексаций: изменения и влияние на доход.

Аналитические требования

KPI: заполняемость (%), доход (base/total), дебиторка, DSO, средняя ставка BYN/м², простой, пени.

Витрины данных.

API аналитики.

Дашборды: KPI-панель, графики заполняемости/доходности, отчеты по дебиторке и пеням, экспорт PDF/Excel.

4.2.3. Постановка задачи на разработку программного приложения

Разработать веб-приложение учета аренды для арендодателя с ежемесячными начислениями в BYN+НДС, учетом оплат, расчетом задолженности/пени и аналитикой (дашборд, отчеты). Обеспечить клиент-серверную архитектуру с REST API (JSON), авторизацию и RBAC, хранение данных в реляционной БД (ЗНФ), генерацию и импорт CSV/Excel шаблонов, документацию по установке и эксплуатации.

4.3. Разработка программного приложения

4.3.1. Проектирование интерфейса программного приложения

Общий каркас. Лейаут с фиксированным верхним баром (логотип, название, быстрый поиск, иконки уведомлений/профиля) и боковым меню.

Основная область — контент со вторичной панелью фильтров. Цветовая тема светлая с акцентами; поддержка темной темы.

Навигация. Главная (Дашборд), Справочники (Помещения, Арендаторы, Тарифы/Индексации), Договоры, Начисления/Счета, Платежи, Отчеты (Заполняемость, Доход, Дебиторка, Индексации), Документы (Акты, Договора), Администрирование (Пользователи/Роли, Настройки, Аудит), Центр уведомлений.

Общие паттерны. Таблицы с серверной пагинацией, сортировкой и многостолбцовыми фильтрами; формы с пошаговыми мастерами; модальные окна подтверждений; тост-уведомления; пустые состояния с подсказками; статус-бейджи; экспорты в Excel/PDF; ролевые ограничения на действия.

Экран: Дашборд (Главная)

Состав виджетов. KPI-карточки (Заполняемость %, Доход base/total, Дебиторка, DSO, Пени); график «Доход по месяцам»; диаграмма «Заполняемость по типам»; матрица Aging; график «Средняя ставка и индексации» (Line с маркерами событий); таблица «Топ-N должников»; карта статусов помещений.

Фильтры. Период, Объект/здание, Тип помещения, Арендатор. Действия. Сохранить пресет, Экспорт, Дрилл-даун до договора/помещения/платежа.

Состояния. Загрузка скелетонами; «нет данных» с инструкциями; отображение времени последнего обновления витрин.

Экран: Справочник «Помещения»

Список. Таблица: Код, Адрес/Объект, Этаж, Площадь (м²), Тип (офис/ритейл/склад/паркинг/реклама), Базовая ставка (BYN/м² или фикс), Статус (свободно/резерв/сдано/на расторжении/освобождено), Доступно с даты. Фильтры по типу, статусу, площади, объекту. Массовые операции: экспорт, смена статуса (для свободно↔резерв).

Карточка. Вкладки: Профиль (атрибуты), Фото/план БТИ (галерея), История статусов, Связанные договоры, Показатели (мини-KPI по доходу/простою). Валидации. Уникальность кода; обязательные поля тип/площадь/адрес; запрет смены на «сдано» вне контекста договора.

Экран: Справочник «Арендаторы»

Список. Наименование, УНП, Контакты, Банковские реквизиты, Активные договоры, Долг. Фильтры по УНП/названию/статусу. Карточка. Реквизиты, Контакты, Договоры, Платежи, Акты сверки, Просрочки.

Экран: «Тарифы/Индексации»

Список. Объект применения (договор/тип/помещение), Тип ставки (м²/фикс), Значение, Правило индексации (коэффициент/новая ставка), Дата вступления, История. Действия. Добавить/изменить, Просмотреть историю индексаций, Сгенерировать план индексаций.

Экран: «Договоры»

Список. Номер, Арендатор, Помещение, Период, База тарификации, Ставка, НДС, Срок оплаты, Депозит, Статус (черновик/на подписи/действует/на расторжении/закрыт). Фильтры по статусу, арендатору, помещению, срокам.

Мастер создания договора. Шаги: 1) Выбор помещения и арендатора; 2) Параметры (ставка, НДС, срок оплаты, депозит, пени, индексации); 3) Документы (генерация/загрузка); 4) Подписание; 5) Итог. Валидации. Пересечение периодов на одно помещение; полнота обязательных полей; согласованность НДС.

Карточка договора. Вкладки: Сводка, Условия, Документы (договор, допсоглашения), Акты (ПП/возврата), Начисления/Счета, Платежи, Статусы, История изменений. Действия: Пролонгация; Изменение условий; Расторжение; Экспорт.

Экран: «Начисления и Счета»

Список начислений. Период, Договор, База, НДС, Итог, Статус (черновик/утвержден/выставлен), Счет №/дата. Фильтры по периоду/договору/статусу. Действия. Сгенерировать за период, Пересчитать (при индексации), Сформировать/отправить счета, Экспорт реестра.

Экран: «Платежи (импорт и разнесение)»

Импорт. Загрузка CSV/Excel, выбор шаблона, предпросмотр, протокол ошибок. Авторазнесение. Таблица сопоставления: Платеж → Счет/Договор/Назначение/Сумма/Дата; статусы «разнесен/частично/требуется внимания»; очередь ручного сопоставления; операции возврата/перерасчета.

Экран: «Отчеты — Заполняемость»

Визуализации. Динамика оссупансу по месяцам; столбцы по объектам/типам; карта статусов помещений. Фильтры. Период, Объект, Тип, Статус. Действия. Экспорт PDF/Excel; переход к помещениям.

Экран: «Отчеты — Доход»

Визуализации. Линии base/total, разбиение по арендаторам/помещениям. Детализация. Таблица начислений и счетов с переходом в договор.

Экран: «Отчеты — Дебиторка/Aging»

Визуализации. Матрица aging (0–30/31–60/61–90/90+), тренд DSO. Таблица. Остатки по арендаторам/договорам, пени, статусы просрочки. Действия: рассылка уведомлений должникам (опц.).

Экран: «Индексации — История и план»

Список событий. Дата, Основание, Коэффициент/Ставка, Договор/Тип. Действия: применить с даты, откат (по правилам), пересчет будущих начислений.

Экран: «Документы и Акты»

Реестр. Договора, Допсоглашения, Акты ПП/Возврата, Акты сверки. Поля: Номер, Дата, Контрагент, Объект, Статус подписи. Действия: загрузка/скачивание, е-подпись (если предусмотрено), привязка к договору.

Экран: «Администрирование — Пользователи/Роли»

Пользователи. Логин, ФИО, Роли, Статус, Последний вход. Действия: создать/деактивировать, сброс пароля, назначить роли. Роли/права. Матрица CRUD и доступов к отчетам/импорту.

Экран: «Администрирование — Настройки»

Параметры. НДС (по умолчанию 20%), Ставка пени (0.1%/день), Срок оплаты (по умолчанию), Шаблоны импорта, Почтовые настройки. Валидации диапазонов; тест-отправка писем.

Экран: «Аудит и Журналы»

Журнал действий. Время, Пользователь, Объект, Действие, Результат. Фильтры по актеру/типу/дате. Экспорт. Журнал уведомлений. Тип, Получатель, Статус доставки, Ретраи.

Экран: «Центр уведомлений»

Лента событий. Бронь истекает, Счета готовы, Просрочка, Индексация, Расторжение, Импорт (ошибки). Действия: перейти к объекту, отметить прочитанным, настройка предпочитаемых уведомлений.

Глобальный поиск

Поисковые области. Договоры, Помещения, Арендаторы, Счета, Платежи, Документы. Подсказки по мере ввода, быстрые переходы.

Состояния, валидация, доступность

Состояния загрузки/ошибок. Скелетоны, retry, сообщения об ошибках с кодами и ссылкой на журнал. Валидации форм. Обязательные поля, маски для УНП/IBAN, проверки дат и пересечений. Доступность. Горячие клавиши, контрастность, aria-атрибуты, навигация с клавиатуры.

4.3.2. Выбор и обоснование архитектуры программного приложения

Архитектура нацелена на: надежность (устойчивость к сбоям импорта/пересчета), масштабируемость (рост договоров/платежей до 100k строк/месяц), наблюдаемость (логирование, метрики, трассировки), безопасность (RBAC, аудит, хранение секретов), изменяемость (конфигурирование НДС/пени/индексаций), переносимость (Windows-ориентированное развёртывание, возможен Linux).

Выбранный стиль. Клиент-сервер с REST API и четким разделением слоев: UI (SPA на React), API (ASP.NET Core), слой доменных сервисов (правила начислений/индексаций/пеней), доступ к данным через EF Core (Repository +

Unit of Work), БД PostgreSQL (3НФ). Долгие операции (импорт, пересчет KPI, регламенты) вынесены в фоновые сервисы.

Состав компонентов. API-шлюз (контроллеры REST), Доменные сервисы (Начисления, Индексации, Пени, Договоры/Статусы), Модуль импорта (CSV/Excel + адаптеры), Модуль сопоставления платежей, Модуль уведомлений (e-mail), Слой аналитики (SQL-представления и materialized views для KPI/aging/revenue/occupancy).

Потоки данных. Пользователь → UI → REST API → доменные сервисы → EF Core → PostgreSQL (OLTP-схема). Регламент (таймер) → доменные сервисы → расчеты/записи.

Импорт CSV/Excel → валидация → staging-таблицы → сопоставление → факты.

Развёртывание. Вариант по умолчанию — монолитный бэкенд (API + фоновые задания) как единый сервис; фронтенд на статическом хостинге. БД — отдельный сервер PostgreSQL. Возможность контейнеризации (Docker) и развёртывания в on-prem/облаке. Горизонтальное масштабирование API через балансировщик; фоновые задания сконфигурированы как единственный исполнитель, чтобы исключить гонки.

Масштабирование и производительность. Кеширование справочников на уровне API. Индексация ключевых полей (договор, помещение, периоды). Пакетная обработка импорта. Материализация витрин KPI уменьшает нагрузку на отчеты (цели: отчет ≤ 5 с на 100k строк). Профилирование SQL, планы выполнения, лимиты на размер импорта.

Безопасность. JWT/куки-аутентификация, роли (Администратор, Бухгалтер/Оператор, Аналитик). Аудит CRUD и важных событий (подписание, начисления, оплаты). Валидация входных данных, ограничение размеров загружаемых файлов, хранение секретов вне кода. Резервные копии БД по расписанию; контроль доступа на уровне БД (схемы/роли).

Наблюдаемость. Централизованное логирование (уровни INFO/WARN/ERROR, корреляционные ID запросов), метрики (время ответа API, число импортированных записей, время пересчета витрин), алерты по SLA (просрочка обновления KPI, ошибки импорта), трассировки.

Конфигурация и CI/CD. Параметры (НДС 20%, ставка пени 0.1%/день, дедлайны) — в конфигурации среды с возможностью изменения через

админ-UI. Сборка: фронтенд (Vite) и бэкенд (.NET) отдельными пайплайнами; миграции БД (EF Migrations); автоматические smoke-тесты после деплоя.

Паттерны и их применение. Repository + Unit of Work (изоляция доступа к данным), Strategy (вариации расчета пеней/индексаций), Factory (создание начислений/счетов), Adapter (импорт CSV/Excel разных форматов), Specification (фильтры отчетов/поиска), Saga/Outbox (опц.) для гарантированной доставки уведомлений.

Альтернативы и отказ от них. Микросервисы — отвергнуты на первом этапе из-за избыточной сложности (оркестрация, распределенные транзакции); возможны позже при росте нагрузки (выделение импорта/аналитики). Бэкенд Node/Java — .NET выбран ради зрелых инструментов на Windows-целевой среде и Npgsql/EF экосистемы. MySQL — уступает PostgreSQL по возможностям materialized views и аналитических запросов.

Риски и меры. Риск больших импортов — пакетирование и лимиты; риск длительных пересчетов — ночные окна + инкрементальные refresh; риск несогласованности статусов — инварианты и доменные проверки; риск ошибок пользователей — строгие валидации и откаты операций, аудит.

4.3.3. Выбор и обоснование технологий для реализации

Frontend

React + TypeScript.

Библиотека для построения SPA и статическая типизация JS.

Обоснование: зрелая экосистема, компонентный подход, высокая продуктивность разработки, типобезопасность и автодополнение. Как применяем: модульная архитектура (страницы/виджеты/формы), типы для DTO API, хуки для состояния и эффектов.

Ant Design.

UI-компоненты (таблицы, формы, модальные окна, навигация).

Обоснование: богатый набор enterprise-компонентов, согласованный дизайн, поддержка таблиц с большими данными. Как применяем: таблицы с серверной пагинацией, Form с валидациями, Drawer/Modal для мастеров и карточек.

React Query. Что это: клиент кэширования и синхронизации данных. Почему: автоматическое кэширование, повтор запросов, статус загрузки, оптимистичные апдейты.

Как применяем: GET/POST к REST API, инвалидация кеша после мутаций, фоновое обновление KPI.

Recharts/ECharts.

Библиотеки визуализации графиков и диаграмм.

Обоснование: покрытие нужных типов (Line/Bar/Stacked/Pie/Heatmap), хорошая производительность и кастомизация.

Как применяем: дашборд и отчеты (доход, заполняемость, aging, DSO, индексации).

Vite.

Современный сборщик/дев-сервер.

Обоснование: быстрые сборки и HMR, простая конфигурация.

Как применяем: сборка SPA, отдельные конфиги для Dev/Prod, алиасы путей.

Backend

.NET 8 + ASP.NET Core Web API.

Платформа и фреймворк для высокопроизводительных REST сервисов. Обоснование: зрелость, кроссплатформенность, высокая производительность, обширные инструменты для Windows-среды.

Как применяем: контроллеры REST, middleware для логирования/авторизации, фильтры исключений.

Entity Framework Core (Npgsql).

ORM для работы с PostgreSQL.

Обоснование: ускоряет разработку, миграции БД, отслеживание изменений, LINQ. Как применяем: контекст с единицами работ (UoW), репозитории, миграции для схемы ЗНФ, профилирование запросов.

FluentValidation.

Декларативная валидация моделей.

Обоснование: отделение валидации от контроллеров, переиспользование правил. Как применяем: правила для DTO (УНП, даты, суммы, пересечения периодов), общий обработчик ошибок.

AutoMapper.

Сопоставление доменных моделей и DTO.

Обоснование: сокращение шаблонного кода, единообразие. Как применяем: профили для сущностей (договора, начисления, платежи).

База данных

PostgreSQL 16.

Реляционная СУБД.

Обоснование: строгая транзакционность, мощный SQL, индексы/partitions, материализованные представления.

Как применяем: схема ЗНФ (справочники/факты), индексы по договорам/периодам, внешние ключи, материализованные витрины KPI/aging/revenue/occupancy с refresh по регламенту.

Миграции БД (EF Migrations). Почему: управляемые изменения схемы с историей. Как применяем: версионирование таблиц и представлений, скрипты seed-данных.

Интеграции и файлы

CSV/Excel (ClosedXML/EPPlus).

Что это: библиотеки чтения/записи Excel.

Обоснование: типовой формат исходных данных; валидации на стороне сервера.

Как применяем: генерация шаблонов импорта (валидация, мастер-листы), парсинг загружаемых файлов, протокол ошибок.

Безопасность и доступ

JWT/куки-аутентификация.

Механизмы сессий и токенов.

Обоснование: совместимость с SPA и корпоративной политикой. Как применяем: выдача токенов после входа, хранение рефреш-токенов (опц.), роль-бэйд авторизация.

RBAC.

Разграничение прав.

Обоснование: разделение обязанностей (Админ, Бухгалтер/Оператор, Аналитик). Как применяем: политики доступа на эндпоинты/действия UI, матрица прав в админке.

Аудит.

Фиксация действий.

Обоснование: ответственность и трассируемость. Как применяем: журнал CRUD и ключевых событий (подпись, начисления, оплаты), идентификаторы корреляции.

4.3.4. Проектирование схемы информационной модели. Разработка справочников

Цель раздела. Определить ER-модель, справочники (измерения), операционные факты, связи и ограничения, индексы, правила историзации и состав аналитических витрин для устойчивой работы транзакционного контура и отчетности.

Сущности и атрибуты (ER-модель)

dim_premise (Справочник помещений): PremiseId (PK, UUID), Code (UK), Type (office/retail/warehouse/parking/ads), Address, Building, Floor, AreaM2 (NUM), BaseRate (NUM), RateType (PerM2|Fixed), VATRate (DEC, default 0.2), Status (free|reserved|leased|termination|released), AvailableFrom (DATE), CreatedAt/UpdatedAt.

Пояснение: центральный справочник фонда помещений. Используется при создании договоров, расчете ставки (м² или фикс), отображении статусов доступности и в аналитике заполняемости.

dim_tenant (Справочник арендаторов): TenantId (PK), Name, UNP (UK), Email, Phone, Address, BankName, IBAN, IsActive, CreatedAt/UpdatedAt.

Пояснение: хранит реквизиты контрагента (УНП, контакты, банковские данные). Нужен для договоров, выставления счетов, платежей и актов сверки.

dim_contract (Справочник договоров): ContractId (PK), ContractNo (UK), ContractDate, PremiseId (FK→dim_premise), TenantId (FK→dim_tenant), StartDate, EndDate, BillingBase (PerM2|Fixed), Rate (NUM), VATRate (DEC, default 0.2), Deposit (NUM), PaymentDueDay (1–31), PenaltyRatePerDay (DEC, default 0.001), Status (draft|signing|active|terminating|closed), CreatedAt/UpdatedAt.

Пояснение: определяет юридические и финансовые параметры аренды. Отсюда читаются ставка/НДС/срок оплаты/депозит для начислений и счетов; статус договора синхронизируется со статусом помещения.

dim_tariff (Правила тарифа/индексаций): TariffId (PK), ScopeType (Contract|PremiseType|Premise), ScopeRefId (nullable FK), RateType (PerM2|Fixed), RateValue (NUM), IndexationRule (None|Coeff|NewRate), IndexationValue (NUM), EffectiveFrom (DATE), EffectiveTo (DATE, nullable), IsActive, CreatedAt/UpdatedAt.

Пояснение: хранит версии тарифов и индексаций (SCD2). Позволяет применять новые ставки с даты вступления как точно по договору, так и по типу/объекту.

dim_calendar (Календарь): CalId (PK), Year, Month, DateStart, DateEnd, MonthKey (YYYY-MM) (UK).

Пояснение: стандартный календарь для привязки периодов отчетности, генерации графиков начислений, агрегирования KPI по месяцам.

fact_invoices (Счета): InvoiceId (PK), ContractId (FK), PeriodMonthKey (FK→dim_calendar.MonthKey), BaseAmount (NUM), VATAmount (NUM), TotalAmount (NUM), InvoiceDate, DueDate, Status (draft|sent|paid|partial|overdue), CreatedAt.

Пояснение: документ начисленного к оплате за период. Используется для рассылки, контроля сроков и сопоставления платежей; служит основой отчета по доходу.

fact_charges (Начисления): ChargeId (PK), ContractId (FK), PeriodMonthKey (FK), BaseAmount (NUM), VATAmount (NUM), TotalAmount (NUM), Source (Regular|Recalc|Adjustment), CreatedAt.

Пояснение: внутренняя запись расчета обязательства (база/НДС/итог) до формирования счета. Позволяет хранить историю пересчетов (индексации/корректировки).

fact_payments (Платежи): PaymentId (PK), PayerTenantId (FK), BankRef (UK optional), PaymentDate, Amount (NUM), Currency (BYN), Purpose, Matched (BOOL), CreatedAt.

Пояснение: отражает поступления из банка (импорт/ввод). Используется для сверки с счетами, контроля дебиторки и расчета DSO.

fact_payment_match (Сопоставление платежей): MatchId (PK), PaymentId (FK), InvoiceId (FK), MatchedAmount (NUM), CreatedAt.

Пояснение: связующая таблица «многие-ко-многим» между платежами и счетами. Поддерживает частичные оплаты/переплаты и прозрачную историю сопоставлений.

fact_penalties (Пени): PenaltyId (PK), ContractId (FK), PeriodDate (DATE), BaseAmount (NUM), RatePerDay (DEC), Days (INT), PenaltyAmount (NUM), CreatedAt.

Пояснение: хранит начисленные пени по просрочкам для отчетности и уведомлений; связывается с aging для эскалаций.

fact_indexations (События индексации): IndexationId (PK), TariffId (FK), ContractId (nullable FK), EffectiveFrom, Rule (Coeff|NewRate), Value (NUM), AppliedBy (UserId), CreatedAt.

Пояснение: журнал применения правил индексации. Нужен для аудита, пересчета будущих начислений и объяснимости динамики ставок.

doc_registry (Документы): DocId (PK), DocType (Contract|Addendum|ActHandover|ActReturn|Reconciliation), ContractId (FK), Number, DocDate, FileUrl, SignStatus, CreatedAt.

Пояснение: реестр юридически значимых документов (договоры, допсоглашения, акты). Обеспечивает хранение ссылок на файлы, статусов подписи и связь с договором.

Ключи, связи и ограничения

1 FK: dim_contract.PremiseId→dim_premise;
dim_contract.TenantId→dim_tenant; fact_invoices/fact_charges→dim_contract;
fact_payments.PayerTenantId→dim_tenant;
fact_payment_match.PaymentId→fact_payments;
fact_payment_match.InvoiceId→fact_invoices;
fact_penalties.ContractId→dim_contract.

2 Уникальности: dim_premise.Code; dim_tenant.UNP;
dim_contract.ContractNo; dim_calendar.MonthKey.

3 Ограничения бизнес-правил: отсутствие пересекающихся по датам активных договоров на одно помещение; согласованность статусов Premise/Contract; PaymentDueDay ∈ [1;31]; VATRate ∈ [0;1]; PenaltyRatePerDay ≥ 0.

4 Каскады: запрет удаления `dim_premise/dim_tenant` при наличии связанных контрактов; мягкое удаление не используется — только деактивация.

Индексы и производительность

1 btree индекс по `fact_invoices(ContractId, PeriodMonthKey)`, `fact_charges(ContractId, PeriodMonthKey)` для отчетов по периодам.

2 btree по `fact_payments(PaymentDate)`, `(PayerTenantId)`, и `covering` индекс `(PaymentId, Amount)` для быстрых сопоставлений.

3 btree по `fact_payment_match(InvoiceId)`, `partial`-индекс по `fact_invoices(Status='overdue')` для выборок просрочек.

4 Гистограмма статистик по суммам для улучшения планов; лимиты на размер парсинга Excel.

Историзация (SCD) и изменения условий

`dim_tariff` ведется как SCD-Type 2: интервалы действия `EffectiveFrom..EffectiveTo` без пересечений (UK: `ScopeType+ScopeRefId+EffectiveFrom`). При изменении ставки/коэффициента создается новая версия; старая закрывается.

`dim_contract` хранит текущие параметры договора; изменения фиксируются допсоглашениями в `doc_registry` и событиями в `fact_indexations`. Пересчеты начислений помечаются источником Recalc в `fact_charges`.

Витрины аналитики (materialized views)

`mv_kpi_occupancy`: агрегирует статусные площади по датам/типам/объектам: `OccupiedM2`, `FreeM2`, `ReservedM2`; $Occupancy\% = \frac{OccupiedM2}{(OccupiedM2 + FreeM2)}$.

`mv_kpi_revenue`: сумма `BaseAmount/TotalAmount` по месяцам, арендаторам, помещениям; отдельные колонки для индексаций/перерасчетов.

`mv_kpi_aging`: остатки по счетам $= \Sigma(Invoice.Total) - \Sigma(Match.Matched)$ с разбиением 0–30/31–60/61–90/90+ на срезах «арендатор/договор/месяц».

`mv_kpi_penalties`: сумма `PenaltyAmount` по периодам и контрагентам; связь с `aging` для эскалаций.

Обновление витрин — ночное (`refresh concurrently`), при критических изменениях — инкрементальный пересчет.

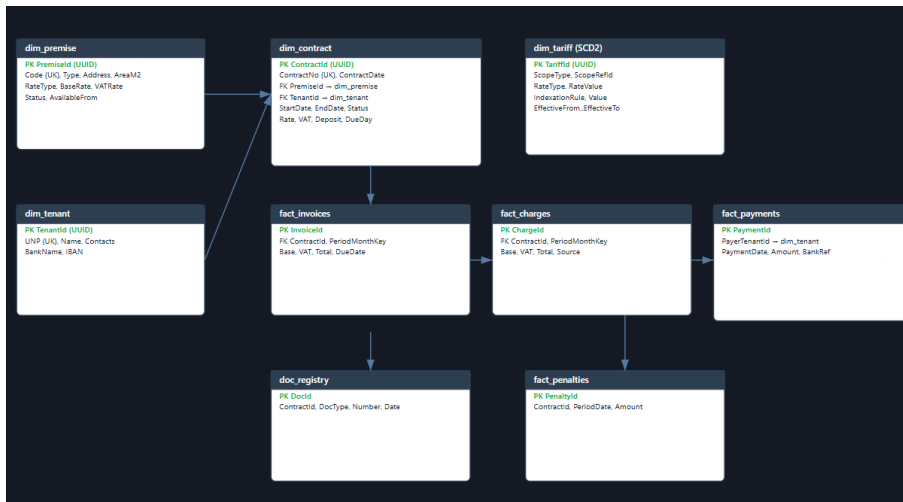


Рисунок 2 - схема информационной модели

4.3.5. Разработка и описание алгоритмов бизнес-логики

Ниже приведен единый алгоритм от появления заявки до закрытия договора и освобождения помещения.

Процесс начинается с поступления заявки арендатора. Менеджер выполняет подбор по критериям, после чего резервирует выбранное помещение на срок переговоров. В момент резерва создается черновик договора, где фиксируются базовые параметры будущих начислений и условия оплаты. Далее формируется проект договора; он проходит юридическое и коммерческое согласование и переходит в состояние подписания.

После подтверждения условий стороны подписывают договор: его статус становится «active», запись о документе попадает в реестр, а по помещению устанавливается статус «leased» с даты начала аренды.

С этого момента включается циклический контур: каждый расчетный месяц система рассчитывает начисления, формирует и отправляет счета, принимает банковские выписки, автоматически сопоставляет платежи и актуализирует статусы счетов. При возникновении просрочек ежедневно начисляются пени и обновляются показатели дебиторской задолженности. Если по договору вводятся индексации или допсоглашения, они регистрируются и применяются с указанной даты, влияя на будущие начисления.

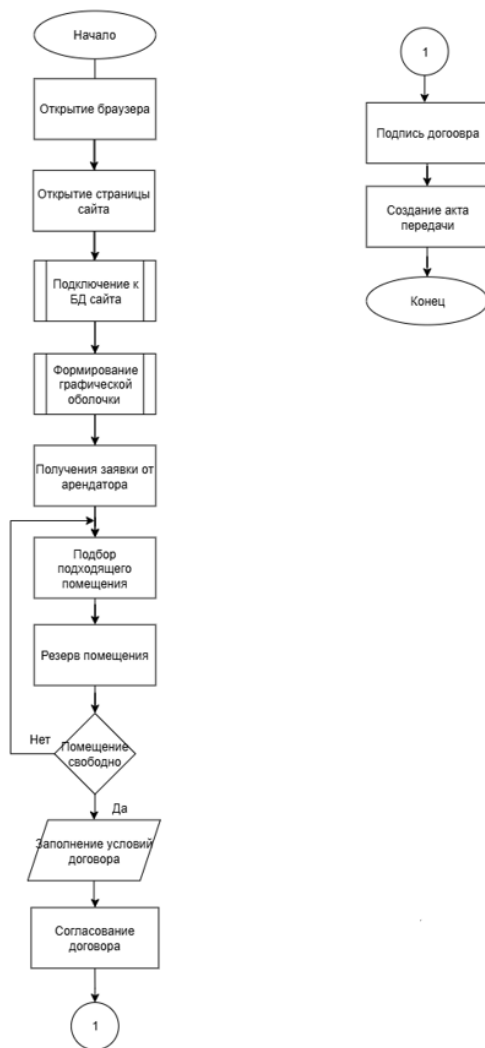


Рисунок 3 – схема алгоритма основного бизнес процесса

4.3.6. Визуализация данных

Дашборд служит центральной точкой наблюдения за фондом помещений и денежными потоками аренды. На одном экране пользователь видит текущую заполняемость, динамику выручки до и после НДС, распределение

дебиторской задолженности по интервалам просрочки и средний срок инкассации. По этой панели можно оперативно оценить риски и концентрацию просрочек, понять влияние индексаций на доходность, обнаружить сезонные колебания и сделать вывод об эффективности использования площадей в разрезе объектов и типов помещений.

В качестве входа дашборд получает агрегированные данные из материализованных витрин `mv_kpi_occupancy`, `mv_kpi_revenue`, `mv_kpi_aging` и `mv_kpi_penalties`, а также справочные срезы для фильтров. Обновление витрин выполняется по регламенту в ночной цикл с возможностью инкрементального пересчета при появлении новых начислений, оплат или изменений договоров. Пользователь управляет периодом анализа и разрезами (объект, тип помещения, арендатор); при отсутствии готовой витрины система корректно рассчитывает показатели на лету по представлениям без нарушения методики.

Методики расчета прозрачны и воспроизводимы. Заполняемость определяется как отношение занятой площади ко всей площади в выбранных границах. Выручка представляется в двух значениях — базовом (сумма начислений) и итоговом (с учетом НДС) с привязкой к календарным месяцам. Дебиторская задолженность разносится по интервалам 0–30, 31–60, 61–90 и свыше 90 дней на основе остатков по неоплаченным счетам относительно их срока оплаты, а показатель DSO рассчитывается как отношение дебиторки к обороту периода, умноженное на длину периода в днях. Пени суммируются за счет произведения просроченной суммы на дневную ставку пени и фактическое число дней просрочки.

Состав визуализаций включает карточки KPI для ключевых метрик, линейные графики динамики выручки, столбчатые диаграммы заполняемости по объектам и типам, матрицу или тепловую карту для распределения просрочек, круговые диаграммы для структуры фонда и таблицу с дрилл-дауном до договора и счета. Фильтры применяются интерактивно без перезагрузки страницы, по клику доступна детализация, предусмотрен экспорт в PDF/Excel и сохранение пользовательских пресетов для повторного использования. Такой набор обеспечивает одновременно быстрый мониторинг и углубленный анализ без выхода из дашборда.



Рисунок 4 – визуализация данных

6 ПЕРЕЧЕНЬ ГРАФИЧЕСКОГО МАТЕРИАЛА (С ТОЧНЫМ УКАЗАНИЕМ ОБЯЗАТЕЛЬНЫХ ЧЕРТЕЖЕЙ):

6.1. AS-IS и TO-BE модели процессов предметной области в нотации BPMN (плакат, 1 лист формата A3).

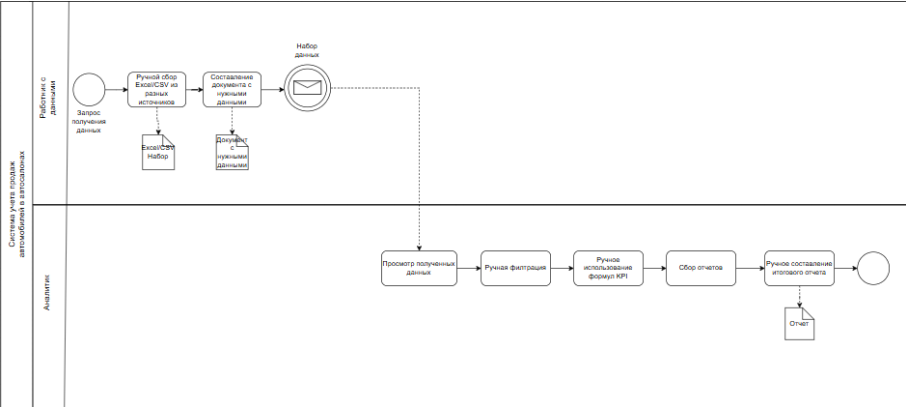


Рисунок 6.1 – диаграмма AS-IS

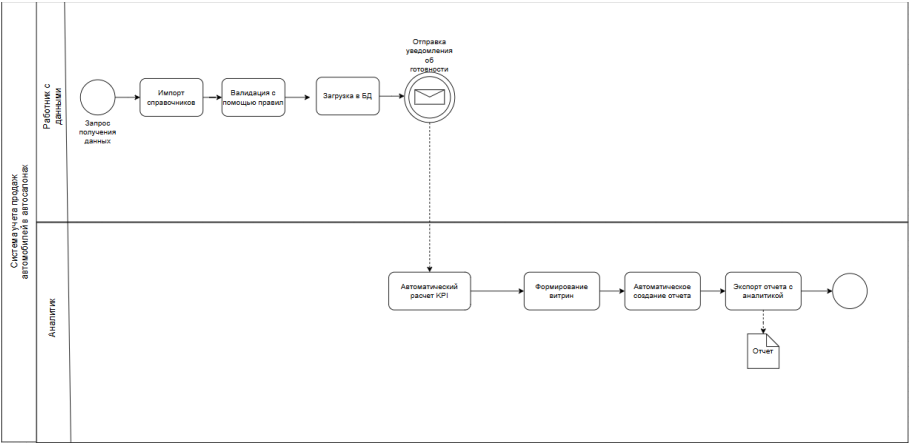


Рисунок 6.2 – диаграмма TO BE