

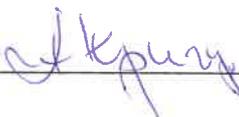
МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

КУРСОВОЕ ПРОЕКТИРОВАНИЕ
Разработка информационной системы для управления
небольшим многоквартирным домом
Б-ПИ-21-01-155915.8850-с.03.000.КП

Студент
гр. БПИ-21-МП1


_____ А.С. Бахарева

Руководитель
канд. физико-
математических
наук, доцент


_____ А. Б. Кригер

Владивосток 2025

Аннотация

Курсовая работа по разработке информационной системы для управления многоквартирным домом состоит из введения, трех глав, заключения, списка использованных источников и приложений. Вторая глава включает в себя 4 пункта. Общий объем работы составляет 54 страниц. Работа содержит 5 таблиц и 16 рисунков.

Список литературы включает в себя источники из научной и технической литературы, профессиональных баз данных, а также профессиональные интернет-ресурсы.

Ключевыми словами, определяющими направление работы, являются: информационная система, управление многоквартирным домом, автоматизация, база данных.

Первая глава посвящена анализу предметной области управления многоквартирным домом. В ней рассматриваются задачи и возможности самоуправления жилым многоквартирным домом, а также

Во второй главе исследуются процессы и задачи «Управдома» и выделение процессов, требующих автоматизации.

В третьей главе представлен обзор и сравнительный анализ прикладного программного обеспечения, автоматизирующего функции "управляющего ТСЖ" и/или старшего по дому, а также разработка технического задания.

В четвертой главе будет рассмотрена архитектура решения, предназначенного для управления многоквартирным домом, а также проектирование базы данных и макета веб-интерфейса.

Практическая значимость исследования специфики разработки информационной системы для управления многоквартирным домом заключается в том, что на основе проведенного анализа предметной области, законодательных требований и технологических возможностей появляется возможность создать более эффективную и функциональную систему.

Данная система будет ориентирована как на старшего по дому, который играет ключевую роль в организации взаимодействия между жителями и старшим по дому или ТСЖ, так и обычных пользователей (жильцов). Основной целью разработки является упрощение этого взаимодействия, что будет достигнуто за счет предоставления доступной и понятной информации о платежах, услугах и текущих делах дома.

Упрощение использования современных технологий также станет важным аспектом. Система будет интуитивно понятной, что позволит жителям различных возрастных групп

легко освоить ее функционал. Это, в свою очередь, минимизирует трудности, связанные с использованием технологий, и сделает повседневную жизнь жителей более комфортной.

Таким образом, разработка информационной системы для управления многоквартирным домом не только облегчит работу старшего по дому, но и создаст условия для более активного участия жителей в управлении своим домом.

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Индивидуальное задание
на курсовое проектирование

Студенту гр. БПИ 21-мп1 Бахарева Анастасии Сергеевне

1. Тема: Разработка информационной системы для управления небольшим многоквартирным домом.

1.1 Номер приказа 8850-с от 03.10.2024г.

1.2 Дата утверждения: 15.11.2024

2. Срок сдачи работы: 16.01.2025

3. Техническое задание

3.1 Цель

Цель работы – анализ функций существующих информационных систем для управления многоквартирными домами и разработка технического задания новой системы, ориентированной на потребности жильцов.

3.2 Технические требования

1. Описание назначения:

- Информационная система с веб-интерфейсом предназначена для управления многоквартирным домом, где находятся не более 30 квартир. Она упрощает процессы управления для старшего по дому.

2. Описание области применения:

- Информационная система для управления многоквартирным домом будет использоваться для автоматизации процессов учета и управления. Она обеспечивает эффективную коммуникацию между жильцами и старшим по дому, позволяет контролировать выполнение работ по обслуживанию и ремонту.

3. Функциональные требования:

- Регистрация и управление данными жильцов, включая контактную информацию и права доступа.

- Возможность жильцов создавать и отслеживать заявки на ремонт и обслуживание общих помещений.

- Платформа для размещения объявлений и уведомлений между жильцами и старшим по дому.

4. Нефункциональные требования:

- Конфиденциальная информация, включая пароли и персональные данные, должна быть зашифрована в базе данных для защиты от несанкционированного доступа.

- Доступ к системе и хранимым данным должен обеспечиваться надежной аутентификацией, включая механизмы логина/пароля и двухфакторную аутентификацию.

5. Интерфейс старшего по дому:

- Удобный и интуитивно понятный интерфейс, поддерживающий работу на различных устройствах.

- Возможность настройки пользовательских ролей и прав доступа.

4. Курсовой проект в обязательном порядке представляется:
- а) пояснительной запиской,
 - б) графическими материалами:
рекомендуются:
 - в) действующий макет проекта или его функционального узла.
5. Содержание пояснительной записки
- 1 Анализ предметной области.
 - 2 Анализ и моделирование рабочих процессов «старшего» по МЖК (управдома).
 - 2.1 Диаграммы процессов - верхний уровень.
 - 2.2 Перечень и спецификация документов, которые составляет старший по дому.
 - 3 Обзор и сравнительный анализ прикладного программного обеспечения для управления домом.
 - 3.1 Разработка технического задания.
 - 4 Проект.
 - 4.1 Архитектура информационной системы.
 - 4.2 Создание базы данных.
 - 4.3 Макет веб-интерфейса.

5 Срок сдачи отчета на кафедру: 16.01.2025

Руководитель
канд. физико-математических
наук, доцент



Кригер А.Б.

Задание получил:



Бахарева А.С.

Содержание

Введение	7
1 Анализ предметной области самоуправления многоквартирным домом	9
2 Процессы и задачи "Управдома"	11
2.1 Обзор процессов управления многоквартирным домом	11
2.2 Перечень и спецификация документов, которые составляет старший по дому	16
3 Обзор и сравнительный анализ прикладного программного обеспечения для управления домом.	18
3.1 Сравнительный анализ прикладного программного обеспечения	18
3.2 Разработка технического задания.	23
4 Проект	31
4.1 Архитектура информационной системы	31
4.2 Создание базы данных	34
4.3 Макет веб-интерфейса	38
Заключение	45
Список использованных источников	47
Приложение А	49

Введение

Разработка информационной системы для управления небольшим многоквартирным домом, в котором находится не более 30 квартир. Такие системы могут значительно упростить взаимодействие жильцов со старшим по дому и повысить прозрачность расчетов, что особенно важно для пожилых людей.

Интуитивно понятные интерфейсы позволят жителям легко пользоваться технологиями для управления своим жильем. Функции, направленные на безопасность и поддержку, создадут комфортные условия для жизни.

Оптимизация ресурсов и снижение затрат на содержание жилья являются важными аспектами для повышения качества жизни жильцов и эффективности управления домом. Растущий интерес к информационным системам управления домами связан с их возможностью оптимизировать рутинные процессы и улучшить обслуживание. Важно учитывать потребности всех пользователей, включая старшего дому и самих жильцов.

Анализ современного состояния информатизации управления небольшими многоквартирными домами позволяет выявить основные тенденции и востребованные функции систем. Автоматизация управления домом помогает управляющему быстрее реагировать на запросы жильцов и повышать качество их жизни.

Объектом исследования является современное состояние информатизации управления жилищными комплексами, что становится актуальным в свете необходимости повышения качества жизни жильцов. Предметом исследования выступают методы и технологии разработки и внедрения информационных систем для управления многоквартирными домами. Эти системы должны учитывать потребности жильцов, обеспечивая им удобный доступ к услугам и информации.

Целью данной работы является анализ функций существующих информационных систем для управления многоквартирными домами и разработка технического задания для новой системы, ориентированной на потребности жильцов. В результате выполнения работы будет создан макет веб-интерфейса и проект базы данных, которые будут учитывать выявленные требования и предпочтения пользователей.

Задачи:

– Изучить теоретические аспекты разработки информационных систем для управления многоквартирными домами, включая ключевые принципы, подходы и методы проектирования.

- Провести анализ отечественного опыта в разработке и внедрении информационных систем для управления многоквартирными домами, выявив успешные практики, а также возможные проблемы и препятствия.

- Исследовать существующие информационные системы, предназначенные для управления многоквартирными домами, с акцентом на их функциональные возможности, удобство использования и адаптацию для различных категорий пользователей.

- Спроектировать базу данных, которая будет эффективно хранить и обрабатывать информацию о жильцах, квартирах, платежах и запросах на обслуживание

- Разработать макет веб-интерфейса, обеспечивающий удобный доступ к функционалу системы для всех пользователей.

В процессе исследования были использованы различные интернет-ресурсы, веб-сайты, специализированные порталы и публикации, а также данные организаций, занимающихся разработкой и внедрением программного обеспечения для многоквартирных домов.

Для достижения поставленных целей будут использованы разнообразные источники данных, включая научные статьи, техническую документацию и статистические данные. Это позволит получить комплексное представление о состоянии информатизации управления многоквартирными домами и разработать эффективную концепцию информационной системы, отвечающей потребностям жителей.

1. Анализ предметной области самоуправления многоквартирным домом

Многоквартирный дом (МКД) – это не просто набор отдельных квартир, а целостная система, включающая в себя жилые помещения и общее имущество, предназначенное для обеспечения комфортного проживания всех собственников. Он представляет собой единое здание с общей территорией и инфраструктурой, требующей скоординированного управления и обслуживания. МКД, в отличие от жилого комплекса(ЖК), как правило, имеет более компактные размеры и ограниченный набор элементов, но, тем не менее, требует комплексного подхода к управлению [1].

Элементы МКД:

1. Квартиры: Основные жилые помещения, принадлежащие собственникам на праве частной собственности. Различаются по площади, планировке и количеству комнат.

2. Общее имущество: Помещения, предназначенные для обслуживания более чем одной квартиры (лестницы, лифты, коридоры, подвалы, чердаки), а также крыша, фасад, земельный участок, прилегающий к дому, и инженерные коммуникации. Общее имущество находится в общей долевой собственности всех собственников квартир.

3. Придомовая территория: Земельный участок, прилегающий к дому, включая газоны, тротуары, пешеходные дорожки, скамейки и элементы благоустройства. Эта территория предназначена для общего пользования собственников и должна поддерживаться в надлежащем состоянии.

4. Детские площадки (при наличии): Специально оборудованные зоны для детского отдыха и игр, которые являются частью общего имущества и требуют регулярного обслуживания и проверки безопасности.

5. Парковки (при наличии): Места для хранения транспортных средств, которые могут быть как наземными, так и подземными, и также являются частью общего имущества или принадлежат отдельным собственникам.

6. Инженерные сети: Комплекс коммуникаций, обеспечивающих функционирование МКД, включая электроснабжение, водоснабжение, водоотведение, отопление, газоснабжение, системы связи и интернет. Эти сети требуют регулярного обслуживания и ремонта.

7. Объекты инфраструктуры (при наличии): Коммерческие помещения, встроенные в МКД, такие как магазины или аптеки, которые могут оказывать услуги жильцам.

Старший по дому наделен особым правовым статусом по сравнению со всеми остальными жильцами дома, и ему предоставлены определенные полномочия [2].

Так, в обязанности старшего по дому входит:

- Осуществление на постоянной основе контроля за качеством и регулярностью уборки всех объектов общего пользования (лестниц, лифтов и т.д.) и придомовой территории, в том числе проверка качества уборки снега в зимний период;

- Контроль деятельности организаций, осуществляющих эксплуатацию жилого фонда, в вопросах исполнения ими обязательств, принятых на себя на основании договоров, — уборка, благоустройство и т.д.;

- Контроль за проведением текущего или капитального ремонта;

- Контроль за соблюдением стандартов в области гигиены, санитарии, экологии, противопожарной безопасности, над качеством проведения дезинфекция, и в том числе оценка условий содержания кошек, собак и прочих домашних животных;

- Принятие мер, направленных на устранение нарушений в использовании и содержании общего имущества дома;

- Предложение и внедрение идей и решений по поддержанию и обеспечению сохранности общедомового имущества;

- Разъяснение жильцам их правового статуса в вопросах содержания и использования жилых и нежилых помещений и общих мест;

- Доведение до сведения жильцов контактов эксплуатационных учреждений, которые могут понадобиться в чрезвычайной и иной нестандартной ситуации;

- Взаимодействие с органами пожарного надзора для обеспечения надлежащего противопожарного состояния дома;

- Общение с органами соцзащиты и здравоохранения при оказании помощи малоимущим и проведении профилактических мероприятий;

Хранение ключей от аварийных выходов, подсобок, служебных, технических помещений (исключение — электрощитов и пусковые), а ведение и хранение журналов с протоколами собраний собственников, учет обращений жильцов, паспортов на МКД и т.д. также входит в обязанности старшего по дому.

По итогам года управдом отчитывается о проделанной работе с представлением соответствующих документов общему собранию жильцов.

2. Процессы и задачи "Управдома"

2.1. Обзор процессов управления многоквартирным домом

Многоуровневый подход к построению диаграмм, начиная с верхнего уровня, необходим для эффективного моделирования сложных систем, таких как управление многоквартирным домом. Верхний уровень обеспечивает общее видение, определяет границы системы, приоритизирует процессы и служит основой для декомпозиции. Декомпозиция, в свою очередь, позволяет детализировать процессы, выявить взаимосвязи, точно настроить каждый подпроцесс, сделать модель более понятной и подготовить систему к автоматизации. Такой подход, основанный на иерархии и четких стандартах, обеспечивает структурированное и понятное представление о системе, необходимое для ее анализа и оптимизации.

Процессы в деятельности "Управдома" представляют собой последовательность взаимосвязанных действий, направленных на достижение конкретной цели. Эти процессы можно представить в нотации IDEF0 как наборы функций, которые принимают на вход определенные ресурсы (информацию, материалы, финансы), преобразуют их и выдают результат. Например, процесс "Приём и обработка обращений жильцов" включает в себя: получение заявки, ее регистрацию, назначение исполнителя, контроль выполнения и уведомление заявителя о завершении. Каждый процесс характеризуется набором входов, выходов, механизмов (ресурсов) и контролей (правил и стандартов). Анализ процессов в формате IDEF0 помогает выявить узкие места, неэффективные операции и зоны, где автоматизация может принести наибольшую пользу [3].

Представленная далее диаграмма IDEF0 описывает основные процессы деятельности старшего по дому. Она демонстрирует взаимосвязь функций, входов, выходов, механизмов и управляющих воздействий, что позволяет анализировать текущие процессы и выявлять области для оптимизации и автоматизации [4].



Рисунок 1 – Процесс деятельности старшего по дому

Далее идет подробное описание представленной диаграммы IDEF0, разбор основных процессов деятельности старшего по дому, их входов, выходов, механизмов и контрольных воздействий. Этот анализ поможет лучше понять текущую деятельность и выявить области для улучшений.

Описание элементов схемы:

1. A0: Деятельность старшего по дому – основной процесс, который включает все действия старшего по дому.

2. Входы:

- Обращения жильцов – запросы и проблемы, с которыми обращаются жильцы.
- Информация от служб – данные о выполненных работах и техническом состоянии дома.

3. Выходы:

- Документы для учета и хранения – набор документов, предназначенных для систематизации, учета и долговременного хранения информации, необходимой для управления и контроля деятельности организации. Включает в себя отчеты, акты, договоры и другие важные материалы.

- Акты для старшего по дому – документация о проделанной работе.

- Информация для жильцов – уведомления о результатах и планах.

4. Управление:

- Нормативные документы – официальные акты, регулирующие управление многоквартирными домами. К ним относятся законы, постановления и правила, которые

определяют права и обязанности жильцов и управляющих компаний, а также условия содержания жилых помещений.

– Регламенты и инструкции по документообороту – внутренние документы, которые определяют порядок создания, обработки, хранения и передачи документов, связанных с управлением многоквартирными домами. Они описывают процедуры работы с документами, включая их регистрацию, согласование, архивирование и уничтожение.

5. Механизм:

- Старший по дому – ключевая фигура, осуществляющая управление процессом.
- Подрядчики – организации, выполняющие работы по заявкам.
- Аварийные службы – службы, реагирующие на неотложные обращения.
- Информационная система – интегрированная платформа, которая автоматизирует и упрощает деятельность старшего по дому, обеспечивая эффективное взаимодействие с жильцами и старшего по дому/ТСЖ.

Для более детального изучения процессов деятельности старшего по дому, представленных на предыдущей диаграмме IDEF0, проведем декомпозицию. Это позволит разбить процесс на более мелкие и понятные составляющие, что значительно упростит их анализ и выявление возможностей для автоматизации.

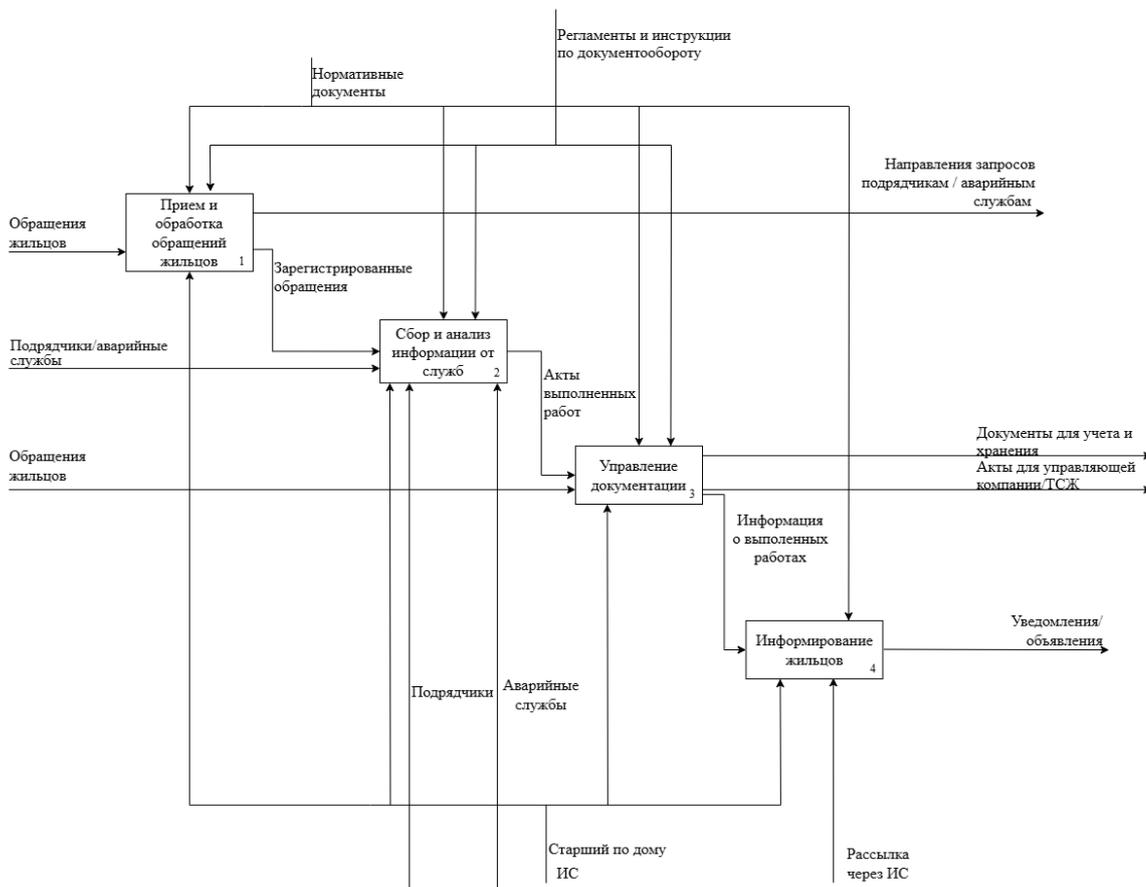


Рисунок 2 – Декомпозиция диаграммы A0

Описание подфункций (элементов) декомпозированной IDEF0:

Каждая подфункция будет охватывать определенный аспект работы старшего по дому, а также взаимодействие с жильцами и другими службами.

A0: Деятельность старшего по дому

A1: Приём и обработка обращений жильцов

Входы:

– Обращения жильцов (запросы, проблемы, жалобы)

Выходы:

– Зарегистрированные обращения (внутренний документ)

– Направления запросов подрядчикам / аварийным службам (при необходимости)

Управление:

– Нормативные документы (сроки реагирования, порядок рассмотрения)

– Регламенты и инструкции по документообороту (регистрация, учет)

Механизм:

– Старший по дому

– Информационная система

A2: Взаимодействие с подрядчиками и аварийными службами

Входы:

– Зарегистрированные обращения

Выходы:

– Акты выполненных работ от подрядчиков

– Информация о выполненных работах

– (При необходимости) Акты о выполненных работах для старшего по дому

Управление:

– Нормативные документы (договоры с подрядчиками, гарантии)

– Регламенты и инструкции по документообороту (оформление актов, приемка работ)

Механизм:

– Старший по дому

– Подрядчики

– Аварийные службы

A3: Управление документацией

Входы:

– Акты выполненных работ

– Обращения жильцов

Выходы:

- Документы для учета и хранения (отчеты, акты, договоры)
- Акты для старшего по дому/ТСЖ

Управление:

- Нормативные документы (сроки хранения, требования к документам)
- Регламенты и инструкции по документообороту (архивирование, ведение реестров)

Механизм:

- Старший по дому
- Информационная система

A4: Информирование жильцов

Входы:

- Информация о выполненных работах

Выходы:

- Информация для жильцов (уведомления, объявления)

Управление:

- Нормативные документы (требования к информированию)

Механизм:

- Старший по дому
- Доска объявлений, чат дома, рассылка через ИС

Также выделим общие замечания по данной диаграмме:

- Управление: Нормативные документы, Регламенты и инструкции по документообороту. Эти факторы являются общими для всех подфункций и определяют рамки их выполнения.

- Механизмы: Старший по дому, Информационная система. Эти ресурсы используются для выполнения всех подфункций. Только Аварийные службы и подрядчики используются для второго элемента схемы.

- Поток информации: Выходы одной подфункции могут быть входами для других, демонстрируя последовательность действий.

На основе этой диаграммы IDEF0, можно выделить следующие области для автоматизации:

- A1: Прием обращений: Онлайн-система регистрации, автоматическая классификация, уведомления.

- A2: Взаимодействие с подрядчиками: Система управления заявками, отслеживание статуса, автоматическое формирование актов.

- A3: Управление документацией: Электронный документооборот, автоматические отчеты, интеграция с бухгалтерией.

– А4: Информирование жильцов: Рассылка уведомлений, интеграция с чатом.

Общие: Интегрированная информационная система, аналитика, управление доступом.

2.2. Перечень и спецификация документов, которые составляет старший по дому

Роль старшего по дому требует ведения определенной документации. В данном разделе рассмотрим, какие именно документы входят в его обязанности и какие требования предъявляются к их составлению. Это позволит четко определить роль старшего по дому в системе управления многоквартирным домом [5].

Старший по дому ведет следующую документацию:

1. Журнал заявок/обращений:

Описание: Записи обо всех обращениях от жильцов (проблемы, запросы, предложения).

Содержание: Дата обращения, ФИО заявителя, описание проблемы, статус рассмотрения, назначенное решение, дата исполнения, отметка о выполнении.

2. План собраний на период (месяц/квартал/год):

Описание: План собраний представляет собой структурированный график проведения собраний собственников помещений (жильцов) для обсуждения и утверждения ключевых вопросов, связанных с обслуживанием, ремонтом и управлением многоквартирным домом.

Содержание: Перечень работ, сроки выполнения, ответственные, бюджет, фактическое исполнение.

3. Акты выполненных работ (по каждой работе):

Описание: Документ, подтверждающий факт выполнения работы, с описанием выполненных действий и объемов.

Содержание: Дата выполнения, вид работы, перечень выполненных действий, объем выполненных работ, ответственные, подписи сторон.

4. Отчет о проделанной работе (периодический):

Описание: Отчет о деятельности "Управдома" за определенный период времени.

Содержание: Тематика собрания, дата/время, место проведения, ответственные.

5. Протоколы собраний жильцов:

Описание: Записи о решениях, принятых на общих собраниях жильцов.

Содержание: Дата и время собрания, перечень участников, повестка дня, принятые решения.

6. Уведомления жильцам:

Описание: Информация для жильцов о предстоящих работах, изменениях, важных объявлениях.

Содержание: Тема, текст уведомления, дата отправки/публикации.

7. Договора с обслуживающими организациями:

Описание: Официальные соглашения с компаниями, предоставляющими услуги.

Содержание: Реквизиты сторон, перечень услуг, стоимость, сроки, ответственность.

Краткое описание документов:

- Все документы должны быть понятными, точными и содержать необходимую информацию.

- Документы должны вестись аккуратно и храниться в порядке.

- Важно отслеживать текущую ситуацию по каждой записи в журналах и актах.

- Отчетность должна быть прозрачной и понятной для жильцов.

- Протоколы должны фиксировать все решения, принятые на собраниях.

Перечень и спецификация документов, приведенные выше, представляют собой основу для организации и эффективного управления многоквартирным домом. Четкое ведение, своевременное обновление и правильное использование этой документации позволяет старшему по дому эффективно выполнять свои обязанности и обеспечивать комфортное проживание для всех жильцов.

3. Обзор и сравнительный анализ прикладного программного обеспечения для управления домом. Разработка ТЗ

3.1 Сравнительный анализ прикладного программного обеспечения

В современном мире эффективное управление жилищными комплексами и товариществами собственников жилья (ТСЖ) становится все более актуальной задачей. С ростом числа многоквартирных домов и увеличением требований со стороны жильцов и государственных органов, управляющим компаниям необходимо использовать современные инструменты для оптимизации своей деятельности.

Рассмотрим управление многоквартирными домами на примере пяти популярных программных решений (таблица 1) [6]. Каждое из них обладает уникальными функциями и возможностями, которые могут существенно упростить работу управляющих компаний и повысить уровень удовлетворенности жильцов.

Таблица 1 – Описание программных обеспечений

Название	Описание
Контур ЖКХ	Это облачное решение для управления жилищным фондом. Оно предоставляет инструменты для автоматизации расчетов, учета платежей, формирования отчетности и взаимодействия с жильцами. Платформа включает функции для работы с документами, управления задачами и контроля за выполнением обязательств перед жильцами.
РеестрДома	Это программа, ориентированная на автоматизацию процессов управления многоквартирными домами. Она позволяет вести учет жилого фонда, проводить расчеты по коммунальным услугам и формировать отчетность. Программа также включает инструменты для взаимодействия с жильцами и управления финансами.
ДомиЛенд	Это программное обеспечение для управления жилыми комплексами и многоквартирными домами. Оно предлагает функционал для учета платежей, формирования отчетов, а также для работы с обращениями жильцов. Платформа обеспечивает возможность интеграции с другими системами и поддерживает автоматизацию различных процессов.
Домучет	Это система, предназначенная для автоматизации учета и управления многоквартирными домами. Она включает в себя функционал для расчетов по коммунальным услугам, учета платежей, формирования отчетности и взаимодействия с жильцами. Система также поддерживает интеграцию с различными сервисами и позволяет управлять документооборотом.
СБЕР ЖКХ	Это платформа от Сбербанка, разработанная для управления жилищным фондом. Она предлагает широкий спектр функций, включая автоматизацию расчетов, учет платежей, формирование отчетности и взаимодействие с жильцами через мобильное приложение. Платформа ориентирована на повышение прозрачности и удобства для пользователей.

Составлено автором по [7,8,9,10,11]

В следующей таблице представлены ключевые функции различных программных решений для управления жилищным фондом (таблица 2).

Таблица 2 – Описание функций

Название	Функции
КОНТУР.ЖКХ	<ul style="list-style-type: none"> – Интеграция с ГИС ЖКХ для автоматизации обмена данными. – Возможность создания личных кабинетов для граждан и управляющих компаний. – Три основных модуля: прием платежей, паспортный учет и облачный сервис. – Упрощение работы с отчетностью и статистикой. – Высокий уровень безопасности данных и удобство использования.
РеестрДома	<ul style="list-style-type: none"> – Учет технического состояния и характеристик домов – Управление документацией по домам и квартирам – Интеграция с ГИС ЖКХ для автоматизации обмена данными. – Возможность формирования отчетов и аналитики – Поддержка взаимодействия с управляющими компаниями
ДомиЛенд	<ul style="list-style-type: none"> – Онлайн-платформа для управления многоквартирными домами – Автоматизация расчетов по коммунальным услугам – Учет платежей и задолженности жильцов – Ведение базы данных собственников и арендаторов – Инструменты для организации собраний и голосований
Домучет	<ul style="list-style-type: none"> – Учет всех видов расходов на содержание жилья – Формирование квитанций для жильцов – Автоматизация процессов сбора платежей и отчетности – Ведение учета технического состояния домов – Поддержка работы с обращениями жильцов
СБЕР ЖКХ	<ul style="list-style-type: none"> – Удобный интерфейс для оплаты коммунальных услуг – Учет платежей и задолженности жильцов – Интеграция с другими сервисами Сбербанка – Возможность подачи заявок на услуги и обращений в управляющую компанию – Аналитика по расходам и платежам

Составлено автором по [7,8,9,10,11]

В представленном списке описаны пять различных инструментов для управления многоквартирными домами, каждый из которых предлагает уникальные функции и возможности. Контур ЖКХ фокусируется на учете и управлении домами, автоматизации расчетов с жильцами и формировании отчетности. РеестрДома акцентирует внимание на техническом состоянии зданий, управлении документацией и интеграции с ГИС ЖКХ. ДомиЛенд представляет собой онлайн-платформу, которая автоматизирует расчеты по коммунальным услугам и предоставляет инструменты для организации собраний. Домучет ориентирован на учет расходов на содержание жилья, формирование квитанций и работу с обращениями жильцов. СБЕР ЖКХ предлагает удобный интерфейс для оплаты коммунальных услуг, учет платежей и интеграцию с другими сервисами Сбербанка.

Далее идет сравнение по показателям (таблица 3) [12].

Таблица 3 – Сравнение по функциональности, удобству, стоимости, поддержке и репутации

	Функциональность	Удобство использования	Стоимость	Поддержка	Репутация и отзывы
КОНТУР.ЖКХ	Учет и управление домами, расчеты с жильцами, отчетность, обращения, база данных по услугам	Веб-интерфейс с возможностью доступа с разных устройств. Есть обучение в виде вебинаров и обучающие материалы. Навигация стандартная, требует времени для освоения.	Подписка, зависит от количества домов и функционала. Диапазон: От 5000 до 20000 руб./мес. Внедрение и настройка могут оплачиваться отдельно.	Телефон, электронная почта, чат. Время отклика от нескольких часов до 1 рабочего дня. Персональные менеджеры для крупных клиентов.	Средняя, с некоторыми жалобами на техподдержку. Опыт пользователей разный, от положительного до отрицательного.
РеестрДома	Учет тех. состояния, управление документацией, интеграция с ГИС ЖКХ, отчеты, взаимодействие с УК.	Веб-интерфейс, удобен для работы с реестрами. Есть обучение: Руководство пользователя, возможно обучение по запросу. Интуитивная навигация, но может быть сложной при работе с большим объемом данных.	Подписка, зависит от количества домов и функционала. Диапазон: От 3000 до 15000 руб./мес. Интеграция с ГИС ЖКХ может требовать дополнительных затрат.	Телефон, электронная почта, форум. Время отклика от 1 до 2 рабочих дней. База знаний и FAQ.	Хорошая, с положительными отзывами о функционале. Опыт пользователей в целом положительный, но с некоторыми оговорками.
Домиленд	Онлайн-управление домами, расчеты по услугам, учет платежей, база собственников, собрания, голосования.	Онлайн-платформа с адаптивным дизайном для ПК и мобильных устройств. Есть обучение: Онлайн-руководство, FAQ. Простая и понятная навигация для пользователей всех уровней.	Подписка, тарифные планы в зависимости от размера УК. Диапазон: От 1000 до 10000 руб./мес. Могут быть дополнительные расходы на настройку и обучение.	Телефон, электронная почта, чат. Время отклика от нескольких минут до нескольких часов. Оперативная поддержка в рабочее время.	Положительная, с хорошими отзывами об удобстве и функциональности. Опыт пользователей в целом положительный, но может быть дорого для некоторых компаний.
Домучет	Учет расходов на содержание жилья, квитанции, сбор платежей, отчетность, тех. состояние,	Десктопное приложение, веб-версия может быть ограничена в функциональности.	Лицензия или подписка, зависит от количества домов и функционала.	Телефон, элек. почта. Время отклика от 1 до 3 рабочих дней. Техническая	Смешанная, с отзывами о сложности в освоении. Успешность использования "Домучет"

Продолжение таблицы 3

	обращения.	Есть обучение: Руководство пользователя и техническая документация. Навигация может показаться сложной для пользователей без опыта работы с подобными программами.	Диапазон: От 2000 до 12000 руб./мес. Могут быть дополнительные расходы на обучение и техническую поддержку.	документация	зависит от того, насколько компания готова инвестировать время и ресурсы в обучение персонала и настройку системы.
СБЕР ЖКХ	Оплата услуг, учет платежей, интеграция со Сбербанком, заявки, аналитика по расходам и платежам.	Мобильное приложение и веб-версия с акцентом на простоту оплаты и подачи заявок. Интуитивно понятный интерфейс, минимальное обучение. Простая и понятная навигация для пользователей, ориентированная на быстрое выполнение основных операций.	Бесплатная базовая версия для жильцов, платная версия для УК и ТСЖ. Диапазон: От 0 до 7000 руб./мес. Зависит от дополнительных интеграций и функционала.	Телефонная линия поддержки, электронная почта, чат. Время отклика от нескольких минут до 1 рабочего дня. Поддержка от специалистов Сбербанка.	В целом положительная, с акцентом на удобство для жильцов. Опыт пользователей в основном положительный среди жильцов, но для УК может быть ограничен.

Составлено автором по [7,8,9,10,11]

В заключение, анализ представленных решений для управления ЖКХ выявил, что каждое из них имеет свои сильные и слабые стороны, а выбор оптимального варианта зависит от приоритетов и потребностей конкретного старшего по дому или ТСЖ. "Контур ЖК" выделяется функциональностью и интеграцией, но может оказаться сложным в освоении и поддержке. "РеестрДома" предлагает полноту информации и соответствие законодательству, но иногда может иметь технические проблемы. "ДомиЛенд" отличается удобством использования и функциональностью онлайн-платформы, но может быть дорогим для некоторых компаний. "Домучет" характеризуется широким функционалом и гибкостью, но сложен в освоении и настройке. "СБЕР ЖКХ" предоставляет удобную оплату и интеграцию с сервисами Сбербанка, но может иметь ограниченный функционал для самих управляющих компаний. Таким образом, при выборе необходимо учитывать не только функциональные возможности, но и удобство использования, стоимость, качество

поддержки, а также репутацию и отзывы пользователей, чтобы подобрать наиболее подходящее решение для конкретных условий.

Также отдельно рассмотрим технические показатели и их сравнение (таблица 4) [12].

Таблица 4 – Сравнение технических показателей

	Надежность	Безопасность	Производительность	Масштабируемость	Интеграция
Контур ЖКХ	Зависимость от серверов разработчика, возможны сбои при пиковой нагрузке. Резервные копии данных, но может быть не полностью отказоустойчивым.	Стандартная: SSL/TLS шифрование, защита от DDoS-атак, аутентификация, соответствие ФЗ-152. Зависит от настроек сервера.	Может работать медленно при большой нагрузке, время отклика зависит от объема данных.	Ограниченная, вертикальное масштабирование, зависит от инфраструктуры Контура. Горизонтально ограничено возможностью увеличения нагрузки на сервер.	Интеграция с продуктами Контур, интеграция с ГИС ЖКХ через API. Возможна интеграция с другими системами по API, но требует настройки.
РеестрДома	Зависимость от серверов, возможны сбои, но в целом стабильная работа. Стандартные меры резервирования.	Стандартная: SSL/TLS шифрование, защита от DDoS-атак, аутентификация, соответствие ФЗ-152. Доп. меры: Ограничение доступа.	Зависит от нагрузки на сервер, может замедляться при большом объеме данных.	Ограниченная, вертикальное масштабирование, сложно масштабировать для крупных организаций. Горизонтально: Ограничена архитектурой системы.	Интеграция с ГИС ЖКХ через API, интеграция с УК, интеграция с другими системами ограничена. Возможна интеграция с другими системами через API.
ДомиЛенд	Облачная архитектура, масштабируемые серверы, резервирование данных. Автоматическое резервное копирование, отказоустойчивость.	Высокая: SSL/TLS шифрование, защита от DDoS-атак, аутентификация, соответствие ФЗ-152, регулярные аудиты безопасности. Защита от SQL-инъекций, XSS.	Оптимизированные базы данных, масштабируемая архитектура, быстрое время отклика. Не требует сложной настройки производительности.	Высокая, горизонтальное и вертикальное масштабирование, легко увеличивать ресурсы при необходимости. Горизонтально: Автоматическое масштабирование при увеличении нагрузки.	Интеграция с ГИС ЖКХ через API, интеграция с платежными системами, широкие возможности интеграции через API. Легкая интеграция с другими сервисами через API.

Продолжение таблицы 4

Домучет	Зависит от серверной инфраструктуры, возможны сбои. Резервирование локальное или на сервере, ручное создание копий.	Средняя: Зависит от настроек, межсетевой экран, разделение прав доступа. Зависит от настроек сервера, требуется ручная настройка.	Зависит от мощности сервера, объем данных влияет на скорость работы. Требуется настройка для работы с большим количеством данных и пользователей.	Ограниченна, зависит от ресурсов сервера, может быть ограничена при большом количестве данных. Горизонтально: Ограничено, возможно только вертикальное.	Интеграция с ГИС ЖКХ, но не всегда удобная, интеграция с другими системами ограничена. Возможна интеграция через API, но требует настройки.
СБЕР ЖКХ	Зависит от инфраструктуры Сбербанка, высокая доступность и надежность. Автоматическое резервное копирование, отказоустойчивость.	Высокая: SSL/TLS шифрование, защита от DDoS-атак, многофакторная аутентификация, строгие меры защиты данных, соответствие стандартам безопасности Сбербанка. Регулярные проверки безопасности.	Быстрая обработка транзакций, оптимизированная работа с платежными системами. Оптимизировано для работы в условиях высокой нагрузки.	Высокая, горизонтальное и вертикальное масштабирование, способна обслуживать большое количество пользователей и транзакций.	Интеграция с ГИС ЖКХ через API, полная интеграция с сервисами Сбербанка. Интеграция с другими системами через API, интеграция с платежными системами.

Эта таблица дает общее представление о технических характеристиках каждого из рассматриваемых решений. "ДомиЛенд" и "СБЕР ЖКХ" выделяются высокой надежностью, безопасностью и масштабируемостью за счет облачной архитектуры и инфраструктуры. "Контур ЖК" и "РеестрДома" показывают средние показатели, в то время как "Домучет" имеет больше ограничений в плане масштабируемости и требует более тщательной настройки.

3.2 Разработка технического задания

В рамках данного проекта будет разрабатываться информационная система (ИС). В контексте управления многоквартирными домами использование ИС позволяет значительно упростить и автоматизировать взаимодействие между жильцами и ТСЖ, повышая эффективность и прозрачность процессов.

Приступая к формированию технического задания на разработку информационной системы, необходимо сосредоточиться на минимально необходимом функционале, простом

интерфейсе, надежной работе и адекватной стоимости. Выбор в пользу ИС как основы для управления МКД обусловлен ее универсальной доступностью. ИС не требует установки на конкретные устройства и может быть использована с компьютера через обычный браузер, что особенно важно для обеспечения удобства всех жильцов, включая старшее поколение, которое может быть менее знакомо с мобильными приложениями, но активно использует компьютеры.

Первым этапом работы является составление технического задания для разработки информационной системы, предназначенной для автоматизации и оптимизации процессов управления многоквартирным домом. В ТЗ необходимо перечислить все необходимые требования, включая функциональные, нефункциональные и технические аспекты разработки [13,14].

Данная информационная система разрабатывается прежде всего для Товарищества собственников жилья или старшего по дому, а также для жильцов МКД. Ее основная цель – существенно облегчить и оптимизировать процессы взаимодействия и управления, снизив нагрузку на ТСЖ/старшего по дому. ИС станет единой точкой доступа к информации и инструментам, необходимым для эффективного управления МКД, обеспечивая прозрачность всех процессов и повышая качество предоставляемых услуг.

В частности, ИС будет способствовать сокращению бумажной работы, повышению оперативности решения проблем, информированности жильцов. ИС должна предоставлять ТСЖ и/или старшему по дому инструменты для эффективного управления МКД, включая управление данными о жильцах, квартирах, договорах и платежах, систему обработки заявок, возможность публикации новостей и объявлений, инструменты для формирования отчетов и коммуникационный канал для связи с жильцами.

Кроме того, ИС должна обеспечивать жильцам возможность просматривать информацию о своем доме и квартире, подавать заявки на обслуживание, оплачивать коммунальные услуги онлайн, участвовать в обсуждениях и получать уведомления о важных событиях.

В информационной системе также будет предусмотрен обучающий материал по новым введениям и процессам как для сотрудников ТСЖ/старшего по дому, так и для жильцов. Им будет открыт доступ к инструкциям, подробной документации и часто задаваемым вопросам, что поможет быстро ознакомиться с новыми процедурами, освоить функционал системы и повысить свою эффективность.

Все эти функции вместе помогут ТСЖ и/или старшему по дому более эффективно управлять МКД, сократить время на рутинные задачи, такие как прием и обработка заявок, информирование жильцов и подготовка отчетов. Это освободит время для решения более

важных задач и стратегического планирования, а также позволит улучшить качество предоставляемых услуг и повысить удовлетворенность жильцов. В свою очередь, жильцы смогут сократить время на подачу заявок, получение информации и оплату коммунальных услуг, что сделает их взаимодействие с ТСЖ/старшим по дому более удобным и комфортным.

Общие технические требования:

1. Общие требования к системе:

– Система должна быть реализована как веб-приложение, доступное через интернет и локальную сеть.

– Интерфейс должен быть интуитивно понятным и удобным для пользователей.

2. Технические характеристики сервера:

– Операционная система: Linux (рекомендуется Ubuntu или CentOS) или Windows Server.

– Процессор: минимум 4 ядра, рекомендовано 8 ядер.

– Оперативная память: минимум 8 ГБ, рекомендуется 16 ГБ.

– Дисковое пространство: минимум 100 ГБ (SSD для повышения производительности).

– Поддержка виртуализации (например, Docker или виртуальные машины).

3. База данных:

– Система должна использовать реляционную базу данных (например, PostgreSQL или MySQL).

– Минимальные требования к базе данных: 10 ГБ свободного места для хранения данных.

– Поддержка механизмов резервного копирования и восстановления данных.

4. Клиентские требования:

– Поддержка всех современных операционных систем (Windows, macOS, Linux).

5. Браузеры:

– Поддержка последних версий популярных браузеров: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari

6. Безопасность:

– Использование HTTPS для защиты передаваемых данных.

– Аутентификация пользователей с использованием безопасных методов (например, OAuth, JWT).

– Регулярные обновления системы безопасности и патчи.

7. Производительность:

- Время отклика системы не должно превышать 5 секунд для основных операций.
- Система должна поддерживать одновременную работу не менее 50 пользователей без заметного ухудшения производительности.

8. Интерфейс пользователя:

- Интерфейс должен быть многоязычным (как минимум поддержка русского и английского языков).
- Доступность интерфейса для людей с ограниченными возможностями (соответствие стандартам WCAG 2.1).

9. Логирование и мониторинг:

- Ведение логов операций пользователей и системных событий.
- Возможность мониторинга состояния системы в реальном времени (например, использование инструментов мониторинга).

10. Поддержка API:

- Предоставление RESTful API для интеграции с другими системами и приложениями.
- Документация по API должна быть доступна для разработчиков.

11. Резервное копирование и восстановление:

- Автоматическое резервное копирование данных не реже одного раза в сутки.
- Возможность восстановления данных из резервной копии должна быть доступна и тестироваться регулярно.

Цели и задачи:

Разработка данной информационной системы направлена на создание эффективной и удобной платформы для автоматизации процессов управления многоквартирными домами и улучшения взаимодействия между товариществами собственников жилья (ТСЖ), старшими по дому и жильцами. ИС предназначена для упрощения коммуникации, оптимизации рутинных задач, повышения прозрачности управления и обеспечения комфорта для всех участников процесса.

1. Автоматизация управления данными о многоквартирных домах (МКД):

- Разработать базу данных для хранения информации о жильцах, квартирах, платежах и договорах.

- Создать пользовательский интерфейс, который позволит старшему по дому удобно добавлять, редактировать и удалять данные.

- Внедрить меры защиты информации от несанкционированного доступа, включая шифрование данных и контроль доступа на основе ролей.

- Обеспечить аутентификацию пользователей и разграничение прав доступа для гарантии целостности данных и оперативного доступа к ним.

2. Упрощение подачи и обработки заявок:

- Создание формы для подачи заявок: Разработать удобную форму для жильцов, позволяющую им описывать проблемы, прикреплять фотографии и указывать уровень срочности.

- Интерфейс для обработки заявок: Создать интерфейс для старшего по дому, который позволит принимать заявки, назначать исполнителей, отслеживать статус и добавлять комментарии.

- Система уведомлений: Настроить автоматические уведомления о новых заявках, изменения статуса и других событиях, связанных с процессом обработки заявок.

- Обеспечение прозрачности процесса: Гарантировать быстрый процесс обработки заявок с возможностью отслеживания всех этапов и оперативного решения возникающих проблем.

3. Обеспечение эффективной коммуникации между участниками:

- Обеспечить возможность создания рассылок для информирования жильцов о важных событиях.

- Реализовать механизм отправки уведомлений о новых сообщениях, новостях, объявлениях и других важных событиях через электронную почту.

- Создать удобный интерфейс для ТСЖ/старших по дому, позволяющий быстро публиковать важные новости и объявления.

- Включить возможность редактирования и удаления опубликованной информации, а также настройки сроков видимости.

- Гарантировать, что все пользователи будут своевременно информированы о важных событиях через автоматизированные уведомления и рассылки.

4. Оптимизация управления финансовыми операциями:

- Обеспечить возможность онлайн-оплаты коммунальных услуг через популярные платежные системы.

- Реализовать безопасный процесс обработки платежей с использованием шифрования данных.

- Создать интерфейс для учета всех финансовых операций ТСЖ/старших по дому, включая поступления от жильцов и расходы на обслуживание.

- Включить возможность добавления комментариев к каждой операции для лучшего понимания финансовых потоков.

- Разработать инструменты для автоматического формирования отчетов о доходах и расходах за различные периоды.

- Внедрить многоуровневую систему защиты данных, включая двухфакторную аутентификацию для доступа к финансовым разделам.

- Предоставить пользователям доступ к истории транзакций и отчетам для повышения прозрачности.

5. Разработка пользовательского интерфейса (UI) и пользовательского опыта (UX) веб-сайта:

- Создать макеты и интерактивные прототипы пользовательского интерфейса с учетом потребностей различных категорий пользователей (ТСЖ, старшие по дому, жильцы).

- Создать современный и привлекательный дизайн, соответствующий актуальным стандартам веб-дизайна и фирменному стилю проекта.

- Использовать цветовые схемы и шрифты, которые способствуют легкости восприятия информации.

- Организовать тестирование интерфейса с участием реальных пользователей для выявления проблем в навигации и взаимодействии.

- Собрать отзывы пользователей и внести необходимые изменения для улучшения удобства использования.

- Внедрить элементы управления, которые будут интуитивно понятны для всех категорий пользователей, независимо от их технической подготовки.

Модули информационной системы с веб-интерфейсом:

1. Модуль аутентификации и авторизации:

- Регистрация новых пользователей, вход в систему с использованием логина/пароля, сброс пароля, разграничение доступа на основе ролей (ТСЖ/старший по дому, жилец).

- Безопасное хранение паролей, защита от перебора паролей, возможность использования различных методов аутентификации (например, двухфакторная).

2. Модуль управления данными о МКД:

- Управление информацией о домах, квартирах, жильцах, договорах, платежных реквизитах и других данных, необходимых для управления МКД.

- Обеспечение целостности и актуальности данных, возможность удобного добавления, редактирования и удаления данных.

3. Модуль обработки заявок:

- Подача заявок на обслуживание жильцами, управление заявками (назначение, отслеживание статуса) для ТСЖ/старшего по дому, уведомления о новых заявках и изменениях статуса.

- Возможность прикрепления файлов (фотографий), категоризация заявок, возможность комментирования заявок.

4. Модуль управления финансами:

- Просмотр начислений за коммунальные услуги.

5. Модуль управления пользовательскими профилями:

- Просмотр и редактирование личных данных, управление настройками уведомлений.

- Разграничение прав доступа к личным данным других пользователей.

6. Модуль отчетов:

- Генерация отчетов о заявках, платежах и других параметрах.

- Возможность экспорта данных в различные форматы, настройка параметров отчетов.

Дополнительные требования:

- Информационная система должна обеспечивать возможность загрузки и скачивания документов в различных форматах, включая (но не ограничиваясь): PDF, DOC/DOCX, XLS/XLSX, CSV.

- Должна быть возможность предпросмотра документов.

- В модуле управления шаблонами документов (и, возможно, в модуле управления данными о МКД) должна быть реализована функция поиска по названию, описанию и метаданным документов.

- Должна быть реализована фильтрация документов по типам, датам загрузки, автору и т.д.

- Внедрить систему автоматического резервного копирования данных ИС, включая базу данных и файлы, для защиты информации.

- Регулярность резервного копирования должна быть настраиваемой (например, ежедневно, еженедельно) в зависимости от потребностей ТСЖ/старшего по дому.

- Должна быть обеспечена возможность восстановления данных из резервной копии в случае сбоя.

- Обеспечить возможность восстановления данных из резервной копии в случае сбоя или потери информации.

- Хранение резервных копий должно быть безопасным и осуществляться в отдельном от основного сервера месте для защиты от потери данных.

Требования к производительности:

- ИС должна обеспечивать быструю загрузку интерфейса и документов, даже если они имеют большой размер.

– Время отклика при выполнении любых операций (поиск, загрузка, отправка данных) должно быть минимальным.

Требования к безопасности:

– Конфиденциальная информация, включая пароли и персональные данные, должна быть зашифрована в базе данных для защиты от несанкционированного доступа.

– Доступ к системе и хранимым данным должен обеспечиваться надежной аутентификацией, включая механизмы логина/пароля и двухфакторную аутентификацию.

– Разграничение прав доступа на основе ролей пользователей.

Требования к поддержке и обновлениям:

– Разработчики должны предоставлять регулярные обновления ИС для исправления ошибок, добавления новой функциональности и повышения безопасности.

– Должна быть предусмотрена возможность отката к предыдущей версии в случае проблем.

– Должна быть предусмотрена система технической поддержки для пользователей в случае возникновения проблем.

4. Проект

4.1. Архитектура информационной системы

Цель информационной системы – это обеспечение удобного и эффективного управления основными процессами в небольшом многоквартирном доме, включая учет жильцов, коммунальных услуг, платежей и заявок.

Для проектирования информационной системы по управлению кадровыми данными (УМКД) для старшего по дому с акцентом на легкость и простоту, можно использовать архитектурный подход, который включает выбор структурных элементов, их интерфейсы, поведение и соединение. Рассмотрим это более подробно [15].

1. Архитектурный стиль.

Архитектурный стиль: Микросервисная архитектура. Этот стиль позволяет разрабатывать независимые модули, которые могут взаимодействовать друг с другом через API, обеспечивая гибкость и масштабируемость системы.

2. Структурные элементы и их интерфейсы

Основные структурные элементы системы:

2.1 Модуль аутентификации и авторизации:

– Интерфейс: REST API

– Методы:

- POST /auth/login — вход пользователя
- POST /auth/logout — выход пользователя
- POST /auth/register — регистрация нового пользователя
- GET /auth/refresh — обновление токена

2.2 Модуль управления данными о МКД:

– Интерфейс: REST API

– Методы:

- GET /buildings — получение списка МКД
- POST /buildings — добавление нового МКД
- PUT /buildings/{id} — обновление данных о МКД
- DELETE /buildings/{id} — удаление МКД

2.3 Модуль обработки заявок:

– Интерфейс: REST API

– Методы:

- GET /requests — получение списка заявок
- POST /requests — создание новой заявки

- PUT /requests/{id} — обновление статуса заявки
- DELETE /requests/{id} — удаление заявки

2.4 Модуль управления пользовательскими профилями:

– Интерфейс: REST API

– Методы:

- GET /profiles/{userId} — получение профиля пользователя
- PUT /profiles/{userId} — обновление профиля пользователя

2.5 Модуль управления финансами:

– Интерфейс: REST API

– Методы:

- GET /finances — получение финансовых данных
- POST /finances — добавление новой финансовой операции
- PUT /finances/{id} — обновление финансовой операции
- DELETE /finances/{id} — удаление финансовой операции

2.6 Модуль отчетов:

– Интерфейс: REST API

– Методы:

- GET /reports — получение доступных отчетов
- POST /reports/generate — генерация отчета

Выбор REST API для разработки информационной системы (ИС) обусловлен его простотой и легкостью использования, так как он использует стандартные HTTP-методы (GET, POST, PUT, DELETE) и форматы (JSON, XML), что упрощает интеграцию. REST API поддерживает микросервисную архитектуру, позволяя каждому модулю работать независимо, что облегчает разработку и развертывание [16].

REST API обеспечивает высокую масштабируемость благодаря возможности развертывания сервисов на отдельных серверах и использованию кэширования на уровне HTTP для повышения производительности. Он также поддерживает кросс-платформенность, что позволяет обслуживать различные клиенты — от веб-приложений до мобильных устройств и IoT.

Статус stateless архитектуры в REST упрощает управление состоянием, так как каждый запрос независим от предыдущих. Наконец, широкое распространение REST API гарантирует наличие обширной документации и ресурсов, что делает его идеальным выбором для создания современных веб-приложений и микросервисов.

3. Поведение структурных элементов

Каждый модуль будет иметь определенное поведение и взаимодействовать с другими модулями:

- Модуль аутентификации и авторизации будет проверять права доступа пользователей к различным модулям.
- Модуль управления данными о МКД будет предоставлять информацию для других модулей, таких как обработка заявок и управление финансами.
- Модуль обработки заявок будет взаимодействовать с модулями управления МКД и пользовательскими профилями для обработки запросов.
- Модуль управления финансами будет генерировать отчеты об оплатах и начислениях, которые могут быть использованы в модуле отчетов.
- Модуль отчетов будет собирать данные из всех остальных модулей для формирования отчетов по запросу.

4. Соединение элементов структуры и поведения.

Элементы структуры будут соединены через API Gateway, который будет служить единой точкой входа для всех запросов к системе. Это обеспечит:

- Централизованное управление доступом.
- Упрощение взаимодействия между клиентом и модулями.
- Легкость в интеграции новых модулей.

5. Архитектурная схема



Рисунок 3 – Архитектурная схема

Описание архитектуры ИС с веб-интерфейсом (Рисунок 3):

5.1 Клиентская часть:

– Веб-приложение: Это основной интерфейс, через который пользователи взаимодействуют с системой. Он может включать различные страницы, такие как главная, профиль пользователя, управление задачами и т.д.

– Веб-приложение будет отправлять запросы к серверу через API.

5.2 API:

– Этот компонент будет обрабатывать запросы от веб-приложения и взаимодействовать с серверной частью. API будет реализован с использованием REST.

5.3 Сервер:

– Сервер будет обрабатывать бизнес-логику, аутентификацию пользователей, управление сессиями и обработку ошибок.

– Он будет взаимодействовать с базой данных для получения и сохранения данных.

5.4 База данных:

– База данных будет хранить всю необходимую информацию, такую как данные пользователей, задачи, события и сообщения.

– Можно использовать реляционную базу данных (например, PostgreSQL или MySQL) или NoSQL (например, MongoDB), в зависимости от требований.

6. Масштабируемость и безопасность

- Масштабируемость: каждый модуль может быть развернут независимо и масштабирован по мере необходимости.

- Безопасность: реализация аутентификации и авторизации на уровне API Gateway, использование HTTPS для защиты данных и токенов (например, JWT) для защиты взаимодействия между клиентом и модулями.

Предложенная архитектура микросервисной системы для старшего по дому включает модули аутентификации и авторизации, управления данными о МКД, обработки заявок, управления пользовательскими профилями, управления финансами и отчетности. Каждый модуль имеет четкие интерфейсы и поведение, что позволяет легко интегрировать новые функции и адаптироваться к изменяющимся требованиям.

4.2. Создание базы данных

При разработке системы управления многоквартирными домами была поставлена задача создать эффективную и надежную базу данных, способную обрабатывать информацию о жильцах, управляющего, квартирах, запросах и коммунальных услугах. В результате анализа требований проекта и доступных технологий было принято решение использовать систему управления базами данных MySQL [17].

MySQL была выбрана благодаря своей высокой производительности, надежности и простоте в использовании. Она предоставляет мощные возможности для работы с реляционными данными, поддерживает сложные запросы и обеспечивает безопасность хранения информации. Кроме того, MySQL является открытым программным обеспечением, что позволяет сократить затраты на лицензирование и обеспечивает гибкость в дальнейшем развитии системы.

Структура базы данных была спроектирована с учетом всех необходимых сущностей и их взаимосвязей. В результате была разработана схема, включающая таблицы для хранения информации о управляющих, зданиях, квартирах, жильцах, статусах запросов и коммунальных услугах. Каждая таблица содержит ключевые поля, которые обеспечивают целостность данных и позволяют эффективно управлять информацией (Рисунок 4).

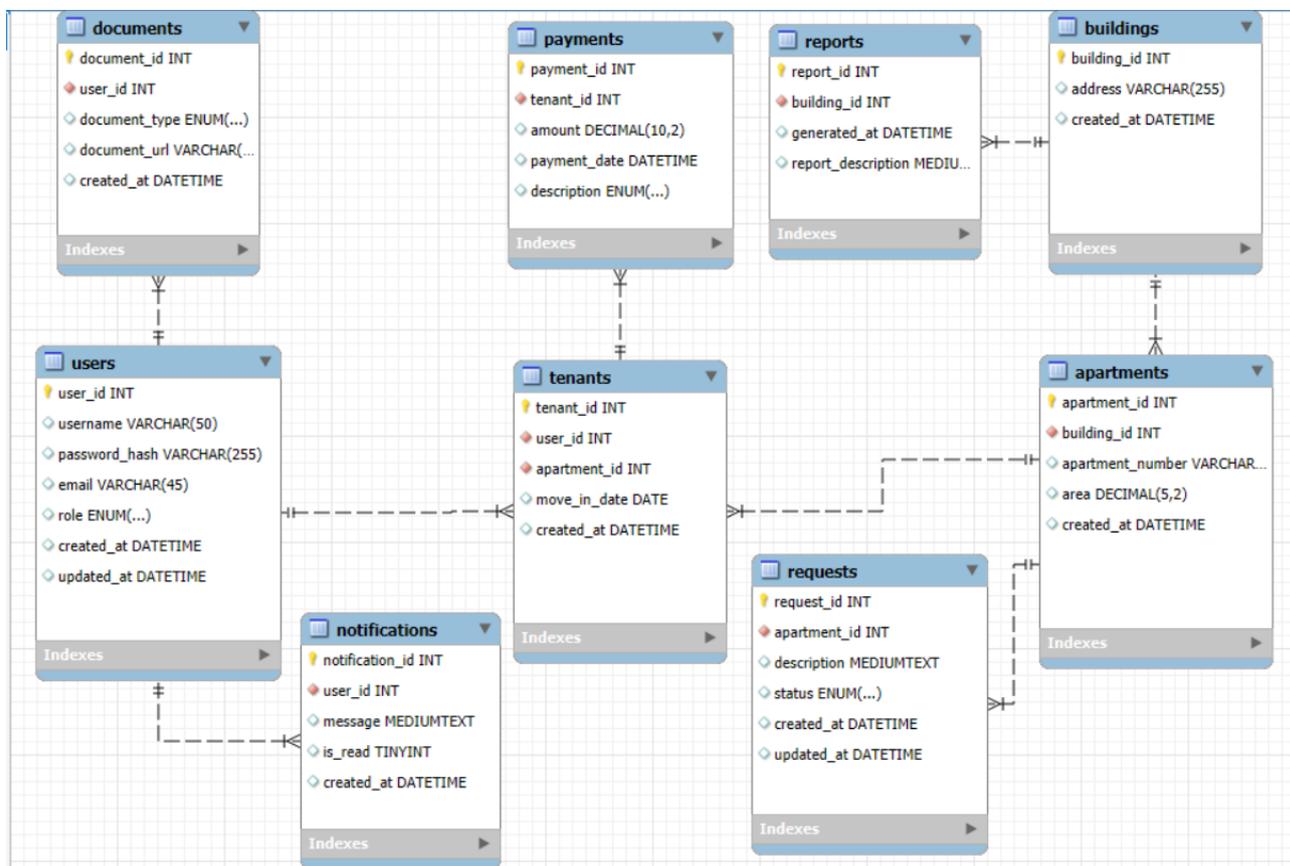


Рисунок 4 – Схема логической организации базы данных

Далее идет описание логической организации базы данных (Таблица 5).

Таблица 5 – Описание структуры.

Таблица	Поле/Тип данных	Описание
users	user_id/INT (PK)	Уникальный идентификатор пользователя
	username/VARCHAR(50)	Имя пользователя
	password_hash/VARCHAR(255)	Хэш пароля
	email/VARCHAR(45)	Электронная почта
	created_at/DATETIME	Дата и время создания пользователя

Продолжение таблицы 5

buildings	building_id/ INT (PK)	Уникальный идентификатор здания
	address/ VARCHAR(255)	Адрес здания
	created_at/DATETIME	Дата и время создания записи
apartments	apartment_id/ INT (PK)	Уникальный идентификатор квартиры
	building_id/ INT (FK)	Идентификатор здания
	apartment_number/ VARCHAR(10)	Номер квартиры
	Area/ DECIMAL(5,2)	Площадь квартиры
	created_at/ DATETIME	Дата и время создания записи
tenants	tenant_id/ INT (PK)	Уникальный идентификатор жильца
	apartment_id/ INT (FK)	Идентификатор квартиры
	user_id/ INT (FK)	Идентификатор пользователя
	move_in_date/ DATE	Дата въезда жильца в квартиру
	created_at/ DATETIME	Дата и время создания записи
requests	request_id/ INT (PK)	Уникальный идентификатор заявки
	apartment_id/ INT (FK)	Идентификатор квартиры
	description/ TEXT	Описание заявки
	status/ENUM("новый", "в_прогнесе", "завершен", "отклонен")	Статус запроса
	created_at/ DATETIME	Дата и время создания запроса
	updated_at/ DATETIME	Дата и время обновления запроса
payments	payment_id/ INT (PK)	Уникальный идентификатор платежа
	tenant_id/ INT (FK)	Идентификатор жильца
	amount/ DECIMAL(10, 2)	Сумма платежа
	payment_date/DATETIME	Дата и время платежа
	description/ ENUM("Электричество", "Холодная вода", "Горячая вода", "Квартплата")	Описание оплаты
reports	report_id/ INT (PK)	Уникальный идентификатор отчета
	building_id/ INT (FK)	Идентификатор здания
	report_description/ TEXT	Описание отчета
	generated_at/DATETIME	Дата и время создания отчета
documents	document_id/ INT (PK)	Уникальный идентификатор документа
	user_id/ INT (FK)	Идентификатор пользователя
	building_id/ INT (FK)	Идентификатор здания
	document_type/ ENUM("договор", "счет", "отчет")	Тип документа
	document_url/ VARCHAR(255)	URL документа
	created_at/ DATETIME	Дата и время создания документа
notifications	notification_id/ INT (PK)	Уникальный идентификатор уведомления
	user_id/ INT (FK)	Идентификатор пользователя
	message/ TEXT	Текст уведомления
	is_read/ BOOLEAN	Флаг чтения уведомления
	created_at/ DATETIME	Дата и время создания уведомления

Каждая таблица имеет свои уникальные поля и связи с другими таблицами, что позволяет организовать данные о пользователях, зданиях, квартирах, жильцах, запросах на обслуживание и платежах.

Опишем также связи:

1. Таблица users (Пользователи)

– Один ко многим с таблицей tenants: Один пользователь может быть связан с несколькими жильцами. Это означает, что один пользователь может арендовать несколько квартир или быть зарегистрированным в системе как несколько жильцов.

– Один ко многим с таблицей notifications (если такая таблица присутствует): Один пользователь может получать несколько уведомлений.

2. Таблица buildings (Здания)

– Один ко многим с таблицей apartments: В одном здании может находиться несколько квартир. Каждая квартира будет связана с одним конкретным зданием.

– Один ко многим с таблицей reports (если такая таблица присутствует): В одном здании могут быть созданы несколько отчетов о состоянии или событиях.

3. Таблица apartments (Квартиры)

– Многие к одному с таблицей buildings: Каждая квартира принадлежит одному зданию.

– Один ко многим с таблицей tenants: В одной квартире может проживать несколько жильцов (например, в случае совместной аренды).

– Один ко многим с таблицей requests: В одной квартире может быть создано несколько запросов на обслуживание.

4. Таблица tenants (Жильцы)

– Многие к одному с таблицей apartments: Каждый жилец привязан к одной квартире.

– Многие к одному с таблицей users: Каждый жилец является пользователем системы. Это позволяет идентифицировать жильца как зарегистрированного пользователя.

– Один ко многим с таблицей payments: Один жилец может делать несколько платежей за аренду или услуги.

5. Таблица requests (Заявки)

– Многие к одному с таблицей apartments: Каждый запрос на обслуживание относится к одной квартире. Это позволяет отслеживать запросы, поступившие от жильцов определенной квартиры.

6. Таблица payments (Платежи)

– Многие к одному с таблицей tenants: Каждый платеж относится к одному жильцу. Это позволяет отслеживать все платежи, сделанные конкретным жильцом.

7. Таблица reports (Отчеты)

– Многие к одному с таблицей users: Каждый документ может быть связан с одним пользователем. Это позволяет отслеживать документы, принадлежащие конкретному пользователю.

– Многие к одному с таблицей buildings: Каждый документ может быть связан с одним зданием. Это позволяет организовать документы по зданиям.

8. Таблица documents (Документы)

– Многие к одному с таблицей users: Каждый документ может быть связан с одним пользователем. Это позволяет отслеживать документы, принадлежащие конкретному пользователю.

9. Таблица notifications (Уведомления)

– Многие к одному с таблицей users: Каждое уведомление предназначено для одного пользователя. Это позволяет отправлять уведомления конкретным пользователям системы.

В Приложении А представлена структура базы данных (физическая реализация).

4.3. Макет веб-интерфейса

Разработка веб-интерфейса является ключевым этапом в создании любого цифрового продукта, определяющим характер взаимодействия пользователей с приложением. В данном контексте, Figma выступает в качестве мощного инструмента для дизайнеров, обеспечивая возможность визуализации и прототипирования интерфейсов с высокой степенью детализации. В рамках настоящей работы рассматривается процесс создания макета веб-интерфейса с использованием Figma. Целью является обеспечение положительного пользовательского опыта и повышение эффективности взаимодействия с разработанным продуктом [18].

На главной странице расположены ключевые элементы, которые помогут старшим по дому быстро ориентироваться в системе (Рисунок 5). В верхней части страницы находится логотип системы и название, например, «Управление МКД». Под ним размещено навигационное меню с основными разделами: «Главная» и «Личный кабинет», где пользователи могут войти в систему или зарегистрироваться. Основной контент включает блок с новостями, где представлена информация о плановых работах. Важные уведомления, такие как предстоящие собрания, выделены отдельным блоком для удобства пользователей.



Рисунок 5 – Скриншот окна главной страницы

Окно входа в личный кабинет оформлено в простом и интуитивно понятном стиле (Рисунок 6). В верхней части размещен заголовок «Личный кабинет». Пользователи вводят логин или адрес электронной почты и пароль в соответствующие поля, при этом пароль отображается в виде точек с возможностью его временного показа. Также имеет выбор ролей такие как admin/user. Под полями расположена кнопка «Войти», а также ссылки «Забыли пароль» для восстановления доступа, отметка «Запомнить», чтобы не вводить каждый раз данные заново и «Зарегистрироваться» для создания нового аккаунта. В случае ввода неверных данных появляется сообщение об ошибке.

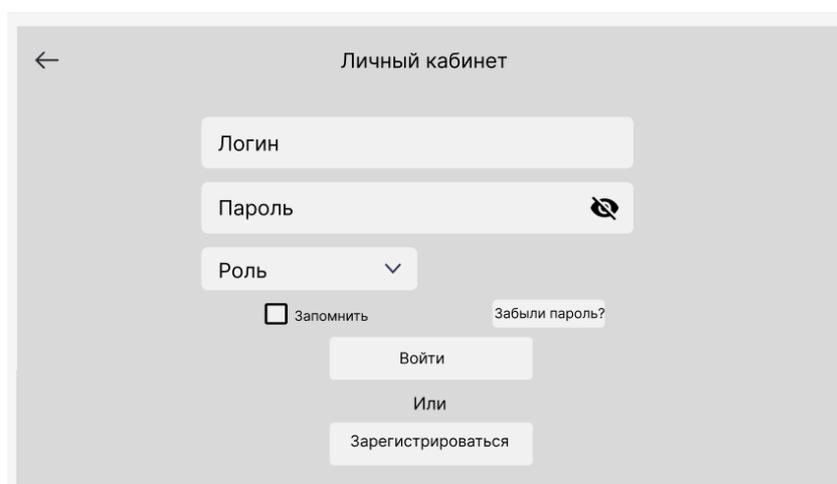


Рисунок 6 – Скриншот окна входа в личный кабинет

Рассмотрим макет со стороны пользователя «Старшего по дому» (admin). Личный кабинет предоставляет доступ к важной информации и функциям (Рисунок 7). Панель навигации включает разделы: «Личный профиль» с информацией о старшем по дому, «Управление заявками» для обработки и создания новых заявок, «Учет платежей» с данными о платежах жильцов, а также «Документы», где доступны договора и акты. Основной

контент кабинета отображает текущие задачи и статус заявок, график собраний и мероприятий.

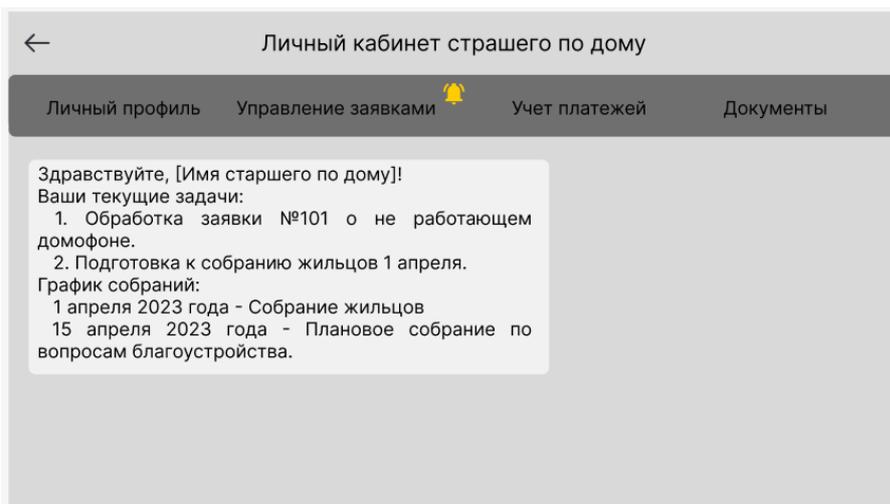


Рисунок 7 – Скриншот окна главной станицы личного кабинета старшего по дому

В этом разделе представлен список заявок в виде списка, где каждая заявка отображается в виде карточки. Каждая карточка содержит следующую информацию: номер заявки, дату создания, краткое описание проблемы, статус, номер квартиры, автора и дополнительные комментарии. Старший по дому может сортировать заявки по статусу с помощью фильтров. При выборе конкретной заявки открывается модальное окно с детальной информацией о проблеме и комментариями от старшего по дому. Также предусмотрена возможность добавления собственных комментариев к заявке (Рисунок 8).

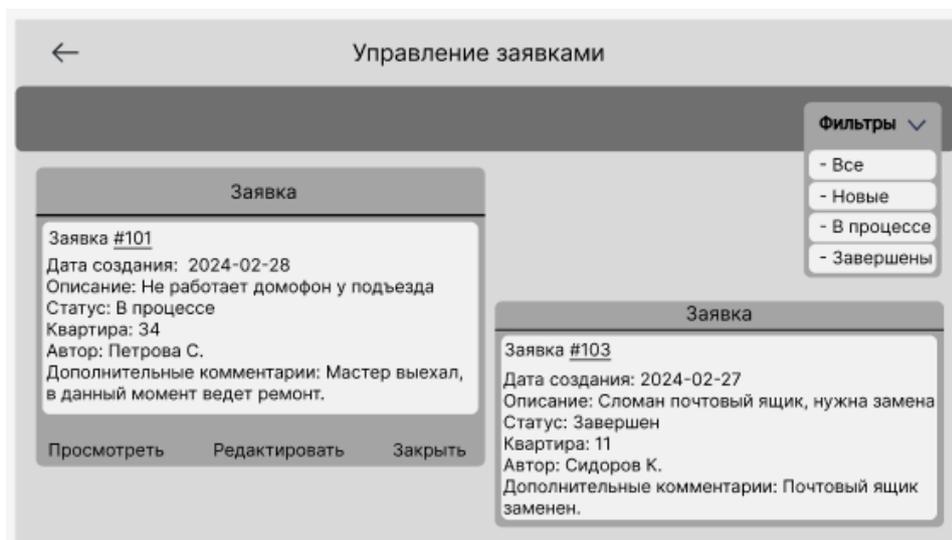


Рисунок 8 – Скриншот окна «Управления заявками»

В этом разделе представлен список документов, организованный по категориям: договора, акты выполненных работ и протоколы собраний. Старший по дому имеет возможность загружать и скачивать документы в формате PDF, что обеспечивает легкий доступ к важной информации (Рисунок 9).

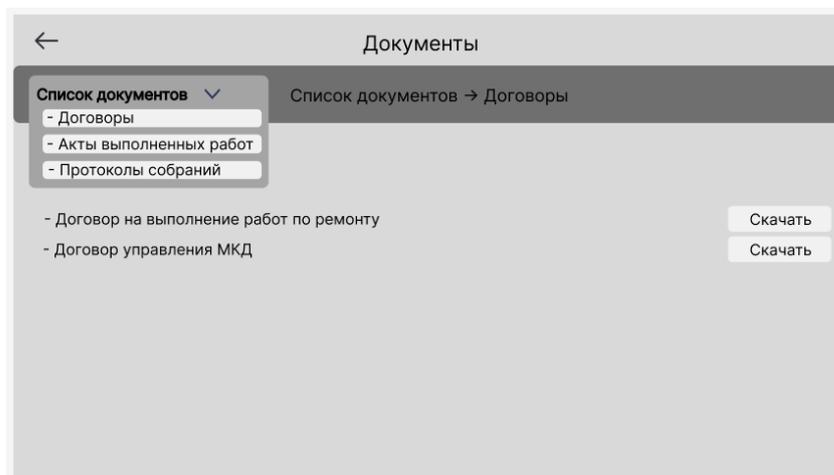


Рисунок 9 – Скриншот окна «Документы»

Раздел учета платежей жильцов представлен в виде таблицы с колонками: ФИО жильца, вид коммунальных услуг, сумма и статус (оплачено/не оплачено). Для удобства пользователей предусмотрена возможность скачивания отчетов по платежам за определённый период, что упрощает контроль финансов (Рисунок 10).

ФИО жильца	Коммунальные услуги	Сумма	Статус
Иванов И.И.	Электричество	2500 руб	Оплачено
Петрова А.А.	Холодная вода	2500 руб	Не оплачено
Сидоров С.С.	Холодная вода	7000 руб	Оплачено
Смирнова Е.Е.	Электричество	4500 руб	Не оплачено
...			

Рисунок 10 – Скриншот окна «Учет платежей»

Теперь рассмотрим макет со стороны пользователя «Жилец» (user). Личный кабинет пользователя в информационной системе управления многоквартирными домами (МКД) представляет собой удобный интерфейс, который позволяет жильцам управлять своими коммунальными услугами и получать актуальную информацию о состоянии своего аккаунта. На главной странице ЛК пользователь приветствуется по имени, что создает персонализированный опыт. Здесь также размещается краткая информация о текущем состоянии аккаунта, включая уведомления о задолженностях и предстоящих платежах (Рисунок 11).

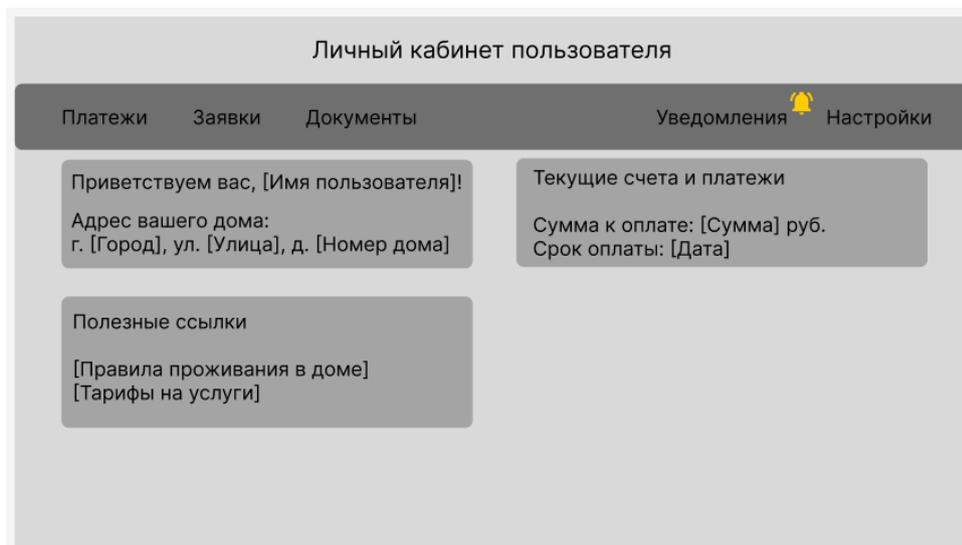


Рисунок 11 – Скриншот окна «Личный кабинет пользователя»

В Личном кабинете пользователя важную роль играют раскрывающиеся уведомления, которые информируют жильцов о ключевых событиях и изменениях (Рисунок 12). Они напоминают о сроках платежей, сообщают о статусах заявок на ремонт, обновлениях документов и персонализированных предложениях. Эти уведомления обеспечивают мгновенный доступ к актуальной информации, удобство использования без необходимости перехода на другие страницы и возможность настройки под индивидуальные предпочтения, что значительно улучшает взаимодействие пользователей с сервисом.

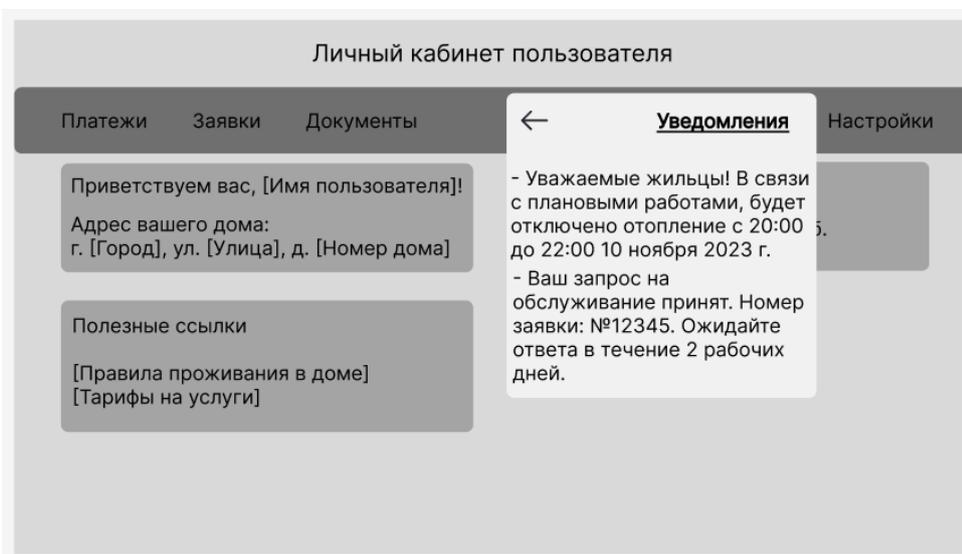


Рисунок 12 – Скриншот «Раскрывающееся уведомление»

В разделе "Платежи" для пользователей доступна история платежей, где можно увидеть список недавних транзакций с указанием дат и сумм. Также показываются текущие счета, где пишется сумма к оплате и срок оплаты (Рисунок 13).



Рисунок 13 – Скриншот окна «Платежи»

Раздел "Заявки" предоставляет пользователям возможность отслеживать и управлять заявками на ремонт или обслуживание (Рисунок 14). На главной странице отображается информация о текущих заявках, включая их статусы. Пользователи могут быстро создавать новые заявки, заполняя необходимые поля, что упрощает процесс обращения за помощью. Эта функция позволяет жильцам быть в курсе статуса своих запросов.



Рисунок 14 – Скриншот окна «Заявки»

Раздел "Документы" предлагает пользователям доступ к важным документам, связанным с управлением многоквартирным домом (Рисунок 15). Здесь можно найти копии договоров, акты выполненных работ, а также другие юридические документы. Пользователи могут легко загружать и просматривать необходимые файлы, что упрощает процесс получения информации.

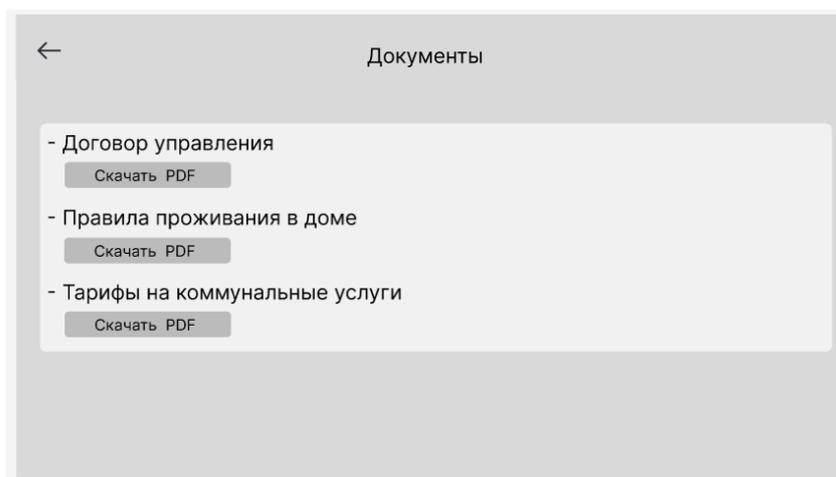


Рисунок 15 – Скриншот окна «Документы»

Раздел "Настройки" позволяет пользователям настраивать параметры своего аккаунта в соответствии с личными предпочтениями. Здесь можно изменить контактные данные, настроить уведомления и управлять доступом к различным функциям Личного кабинета. Удобный интерфейс позволяет легко находить необходимые настройки и адаптировать работу системы под свои нужды, что делает взаимодействие с информационной системой более комфортным и эффективным.



Рисунок 16 – Скриншот окна «Настройки»

В заключение, разработанный макет веб-интерфейса представляет собой интуитивно понятное и функциональное решение, которое отвечает потребностям пользователей. Удобная навигация, четкая структура и визуальная привлекательность способствуют улучшению пользовательского опыта. Внедрение раскрывающихся уведомлений обеспечит оперативное информирование жильцов о важных событиях и изменениях, что повысит уровень взаимодействия с сервисом. Данный интерфейс станет надежным инструментом для управления жилым пространством и удовлетворения потребностей пользователей.

Заключение

В заключение, в рамках данной работы был разработан проект по реализации архитектурного решения, созданию базы данных и макету веб-интерфейса. Основной целью данного проекта стало проведение анализа функций существующих информационных систем для управления многоквартирными домами и разработка технического задания для новой системы, ориентированной на потребности жильцов.

Для достижения этой цели была выполнена комплексная работа, включающая несколько ключевых этапов. На начальном этапе был проведен анализ предметной области управления многоквартирными домами, в ходе которого были рассмотрены ключевые понятия, такие как структура дома, общедомовое имущество и участники процесса управления. Особое внимание уделялось роли старшего по дому как ключевого субъекта управления, ответственного за взаимодействие с жильцами, подрядчиками и старшим по дому. Этот анализ позволил сформировать четкое понимание контекста функционирования будущей информационной системы.

Далее был осуществлён детальный анализ процессов и задач, выполняемых старшим по дому, включая все этапы его работы: от приема и обработки обращений жильцов до взаимодействия с подрядными организациями и ведения документооборота. Были идентифицированы ключевые процессы, требующие автоматизации, такие как регистрация обращений, планирование работ, учет выполненных задач и формирование отчетов. Также был проведен анализ перечня и спецификации документов, составляемых старшим по дому, что позволило определить требования к структуре данных и отчетности будущей системы.

В третьей главе был представлен обзор отечественного опыта в разработке информационных систем для управления многоквартирными домами. Рассмотрены различные типы систем, их функциональные возможности, преимущества и недостатки, что позволило выявить наиболее успешные практики.

На основе проведенного анализа предметной области, процессов и задач старшего по дому, а также обзора отечественного опыта были сформулированы ключевые требования к функциональности, пользовательскому интерфейсу и техническим характеристикам будущей информационной системы. Определены основные модули системы, перечень необходимых функций, принципы взаимодействия с пользователями (жильцами, старшим по дому, подрядчиками и старшим по дому), а также требования к безопасности и надежности системы. Таким образом, разработанный проект закладывает основу для создания эффективной информационной системы, способствующей улучшению управления многоквартирными домами и удовлетворению потребностей жильцов. В заключение можно

отметить, что в результате проделанной работы был проведен комплексный анализ, позволивший сформировать четкое представление о потребностях и требованиях к информационной системе управления МКД. Данный анализ послужил основой для разработки технического задания, которое содержит описание функциональных и нефункциональных требований к системе, её архитектуру, а также описание всех необходимых модулей. Разработанное ТЗ является руководством для создания эффективной системы, ориентированной на потребности жильцов и упрощающей работу старшего по дому. Прделанная работа обеспечивает понимание предметной области и является важным шагом к созданию более удобного и прозрачного инструмента для управления многоквартирными домами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое МКД: признаки, структура [Электронный ресурс] // МКДЭКСПЕРТ.
– URL: <https://mkdexpert.com/publikatsii/chto-takoe-mkd-priznaki-struktura-i-otlichiya-ot-chastnogo-doma/> (Дата обращения – 01.12.2024)
2. Старший по дому: права, обязанности, зарплата [Электронный ресурс] // Дзен.
– URL: <https://dzen.ru/a/XKIT9RNNZwCypHOR> (Дата обращения – 01.12.2024)
3. IDEF0: Знакомство с нотацией и пример использования [Электронный ресурс]
// Тринион-Блог. – URL: <https://www.trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya> (Дата обращения – 05.12.2024)
4. Школа «Управдом» [Электронный ресурс] // Владивосток, официальный сайт администрации. – URL: <https://www.vlc.ru/city-environment/housing-and-utilities/housing-news/> (Дата обращения – 05.12.2024)
5. Электронный фонд нормативно-технической и нормативно-правовой информации Консорциума «Кодекс» [Электронный ресурс] // АО «Кодекс», официальный сайт. – URL: <https://docs.cntd.ru/document/494500938/titles/2K7083N> (Дата обращения – 05.12.2024)
6. Топ-15 лучших ЖКХ-программ для УК и ТСЖ для управление многоквартирными домами [Электронный ресурс] // Электронная платформа. Проект. – URL: <https://vc.ru/links/1468566-top-15-luchshih-zhkh-programm-dlya-uk-i-tszh-dlya-upravlenie-mnogokvartirnymi-domami> (Дата обращения – 10.12.2024)
7. Программа для ТСЖ и УК [Электронный ресурс] // КонтурЖКХ, официальный сайт. – URL: <https://kontur.ru/kontur-jkh-kvartplata> (Дата обращения – 17.12.2024)
8. Удобная и понятная система управления многоквартирными домами России от reestrdoma.ru [Электронный ресурс] // РЕЕСТР ДОМА, официальный сайт. – URL: <https://reestrdoma.ru/?r=MTc0OTU> (Дата обращения – 17.12.2024)
9. Сервисы ЖКХ от Domyland [Электронный ресурс] // Domyland, официальный сайт. – URL: <https://domyland.ru/gkh-services> (Дата обращения – 17.12.2024)
10. Домучет.онлайн. ДомУЧЕТ - программа для ТСЖ и УК [Электронный ресурс] // ДомУЧЕТ, официальный сайт. – URL: <https://domuchet.online/> (Дата обращения – 17.12.2024)
11. СберБизнес. ЖКХ [Электронный ресурс] // Сбербанк, официальный сайт. – URL: https://www.sberbank.com/ru/s_m_business/businessapps/hcs (Дата обращения – 17.12.2024)

12. Основные критерии выбора информационных систем [Электронный ресурс] // Студенческий научный форум - 2019, официальный сайт. – URL: <https://scienceforum.ru/2019/article/2018016733> (Дата обращения – 17.12.2024)
13. Трошина С.М., Мордвинов М. К вопросу о техническом задании [Электронный ресурс] / Трошина С.М., Мордвинов М. // научная электронная библиотека «eLIBRARY.ru». – 2017. – URL: <https://www.elibrary.ru/item.asp?id=29750560> (Дата обращения – 27.12.2024)
14. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс] // официальный сайт Росстандарта. – URL: <https://protect.gost.ru/document1.aspx?control=31&id=241754> (Дата обращения – 27.12.2024)
15. Habr - Архитектура ИТ решений [Электронный ресурс] // официальный сайт Habr – URL: <https://habr.com/ru/articles/347204/> (Дата обращения – 27.12.2024)
16. Habr – Введение в REST API – RESTful веб-сервисы [Электронный ресурс] // официальный сайт Habr – URL: <https://habr.com/ru/articles/483202/> (Дата обращения – 27.12.2024)
17. MySQL – Скачиваемое программное обеспечение // официальный сайт MySQL. – URL: <https://www.mysql.com/> (Дата обращения – 27.12.2024)
18. Figma: The Collaborative Interface Design Tool // официальный сайт Figma. – URL: <https://www.figma.com/design/F7Jt5LQLBhDYOC6jz8zyqI/Untitled?node-id=0-1&p=f&t=PqcMnzStCajoH8kf-0> (Дата обращения – 27.12.2024)

Приложения А
(Обязательное)

Структура базы данных (физическая реализация)

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,N
O_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----
```

```
-- Schema mydb
```

```
-----
```

```
DROP SCHEMA IF EXISTS `mydb` ;
```

```
-----
```

```
-- Schema mydb
```

```
-----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

```
-----
```

```
-- Table `mydb`.`users`
```

```
-----
```

```
DROP TABLE IF EXISTS `mydb`.`users` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`users` (
  `user_id` INT NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(50) NULL,
  `password_hash` VARCHAR(255) NULL,
  `email` VARCHAR(45) NULL,
  `role` ENUM("admin", "user") NULL,
  `created_at` DATETIME NULL,
  `updated_at` DATETIME NULL,
```

```

PRIMARY KEY (`user_id`),
UNIQUE INDEX `username_UNIQUE` (`username` ASC) VISIBLE,
UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`buildings`
-----

```

```

DROP TABLE IF EXISTS `mydb`.`buildings` ;

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`buildings` (
  `building_id` INT NOT NULL AUTO_INCREMENT,
  `address` VARCHAR(255) NULL,
  `created_at` DATETIME NULL,
  PRIMARY KEY (`building_id`))
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`apartments`
-----

```

```

DROP TABLE IF EXISTS `mydb`.`apartments` ;

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`apartments` (
  `apartment_id` INT NOT NULL AUTO_INCREMENT,
  `building_id` INT NOT NULL,
  `apartment_number` VARCHAR(10) NULL,
  `area` DECIMAL(5,2) NULL,
  `created_at` DATETIME NULL,
  PRIMARY KEY (`apartment_id`),
  INDEX `fk_apartments_buildings1_idx` (`building_id` ASC) VISIBLE,
  CONSTRAINT `fk_apartments_buildings1`
  FOREIGN KEY (`building_id`)
  REFERENCES `mydb`.`buildings` (`building_id`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`requests`
-----

DROP TABLE IF EXISTS `mydb`.`requests` ;

CREATE TABLE IF NOT EXISTS `mydb`.`requests` (
  `request_id` INT NOT NULL AUTO_INCREMENT,
  `apartment_id` INT NOT NULL,
  `description` MEDIUMTEXT NULL,
  `status` ENUM("новый", "в_прогессе", "завершен", "отклонен") NULL,
  `created_at` DATETIME NULL,
  `updated_at` DATETIME NULL,
  PRIMARY KEY (`request_id`),
  INDEX `fk_requests_apartments1_idx` (`apartment_id` ASC) VISIBLE,
  CONSTRAINT `fk_requests_apartments1`
    FOREIGN KEY (`apartment_id`)
    REFERENCES `mydb`.`apartments` (`apartment_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`reports`
-----

DROP TABLE IF EXISTS `mydb`.`reports` ;

CREATE TABLE IF NOT EXISTS `mydb`.`reports` (
  `report_id` INT NOT NULL AUTO_INCREMENT,
  `building_id` INT NOT NULL,

```

```

`generated_at` DATETIME NULL,
`report_description` MEDIUMTEXT NULL,
PRIMARY KEY (`report_id`),
INDEX `fk_reports_buildings1_idx` (`building_id` ASC) VISIBLE,
CONSTRAINT `fk_reports_buildings1`
  FOREIGN KEY (`building_id`)
  REFERENCES `mydb`.`buildings` (`building_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`tenants`
-----

DROP TABLE IF EXISTS `mydb`.`tenants` ;

CREATE TABLE IF NOT EXISTS `mydb`.`tenants` (
  `tenant_id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NOT NULL,
  `apartment_id` INT NOT NULL,
  `move_in_date` DATE NULL,
  `created_at` DATETIME NULL,
  PRIMARY KEY (`tenant_id`),
  INDEX `fk_tenants_users1_idx` (`user_id` ASC) VISIBLE,
  INDEX `fk_tenants_apartments1_idx` (`apartment_id` ASC) VISIBLE,
  CONSTRAINT `fk_tenants_users1`
    FOREIGN KEY (`user_id`)
    REFERENCES `mydb`.`users` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_tenants_apartments1`
    FOREIGN KEY (`apartment_id`)
    REFERENCES `mydb`.`apartments` (`apartment_id`)
    ON DELETE NO ACTION

```

```

    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`payments`
-----

DROP TABLE IF EXISTS `mydb`.`payments` ;

CREATE TABLE IF NOT EXISTS `mydb`.`payments` (
  `payment_id` INT NOT NULL AUTO_INCREMENT,
  `tenant_id` INT NOT NULL,
  `amount` DECIMAL(10,2) NULL,
  `payment_date` DATETIME NULL,
  `description` ENUM("Электричество", "Холодная вода", "Горячая вода", "Квартплата")
  NULL,
  PRIMARY KEY (`payment_id`),
  INDEX `fk_payments_tenants1_idx` (`tenant_id` ASC) VISIBLE,
  CONSTRAINT `fk_payments_tenants1`
  FOREIGN KEY (`tenant_id`)
  REFERENCES `mydb`.`tenants` (`tenant_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`documents`
-----

DROP TABLE IF EXISTS `mydb`.`documents` ;

CREATE TABLE IF NOT EXISTS `mydb`.`documents` (
  `document_id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NOT NULL,
  `document_type` ENUM("договор", "счет", "отчет") NULL,

```

```

`document_url` VARCHAR(255) NULL,
`created_at` DATETIME NULL,
PRIMARY KEY (`document_id`),
INDEX `fk_documents_users1_idx` (`user_id` ASC) VISIBLE,
CONSTRAINT `fk_documents_users1`
  FOREIGN KEY (`user_id`)
  REFERENCES `mydb`.`users` (`user_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mydb`.`notifications`
-----

DROP TABLE IF EXISTS `mydb`.`notifications` ;

CREATE TABLE IF NOT EXISTS `mydb`.`notifications` (
  `notification_id` INT NOT NULL,
  `user_id` INT NOT NULL,
  `message` MEDIUMTEXT NULL,
  `is_read` TINYINT NULL DEFAULT 0,
  `created_at` DATETIME NULL,
  PRIMARY KEY (`notification_id`),
  INDEX `fk_notifications_users1_idx` (`user_id` ASC) VISIBLE,
  CONSTRAINT `fk_notifications_users1`
    FOREIGN KEY (`user_id`)
    REFERENCES `mydb`.`users` (`user_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```