

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Цифровая трансформация бизнеса

(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка подсистемы расчета стоимости коммунальных
услуг»

Обучающийся А.Д. Кульминский

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н. , Н.В. Хрипунов

(ученая степень, звание, И.О. Фамилия)

Консультант

М.В. Дайнеко

(ученая степень, звание, И.О. Фамилия)

Тольятти 2025

Аннотация

Данная бакалаврская работа посвящена созданию веб-подсистемы расчета стоимости коммунальных услуг как части системы управления объектами недвижимости. Основной целью проекта стало повышение точности вычислений, автоматизация процессов и улучшение взаимодействия пользователей с системой.

Работа состоит из введения, трех глав, заключения и приложений. В исследовании применены современные методы проектирования информационных систем, включая объектно-ориентированный анализ и модульный подход к разработке. Особое внимание уделено вопросам масштабируемости системы и её интеграции с внешними сервисами.

Ключевой задачей исследования является разработка функциональной и экономически целесообразной подсистемы, которая отвечает современным требованиям автоматизации и гибкости. Это особенно актуально в связи с текущими неэффективностями ручных методов расчета, ограниченной адаптируемостью тарифов и недостаточной интеграцией существующих решений с платежными системами. Система позволяет оперативно реагировать на изменения нормативной базы и тарифной политики.

В первой главе проводится анализ предметной области, рассматриваются бизнес-процессы компании «Квартплата 24», а также приводится сравнительный анализ аналогичных решений на рынке. Особое внимание уделено исследованию проблемных мест существующих систем. Также определены функциональные и бизнес-требования к новой системе, сформулированные на основе интервью с сотрудниками компании.

Во второй главе представлена концептуальная модель системы, включающая UML-диаграммы (вариантов использования, последовательности, классов) и ERD-модели баз данных. Эти модели позволяют четко представить структуру системы, взаимодействие компонентов и организацию хранения данных. Разработана нормализованная

структура базы данных, обеспечивающая целостность и непротиворечивость информации.

Третья глава посвящена реализации подсистемы. Описаны выбранные технологии — HTML, CSS, JavaScript для клиентской части и SQLite Online для серверной части. Подробно рассмотрены этапы разработки интерфейса, структуры базы данных и основных функций для управления объектами недвижимости, услугами и расчетами. Приведен контрольный пример расчета стоимости коммунальных услуг, а также выполнена оценка экономической эффективности проекта, подтвердившая его рентабельность.

В заключении сделан вывод о том, что разработанная подсистема позволяет полностью автоматизировать процесс расчета стоимости коммунальных услуг, повысить точность вычислений и удобство использования сервиса для пользователей. Внедрение данной системы позволит заказчику — ООО «Квартплата 24» — снизить операционные издержки на 25-30%, минимизировать ошибки при расчетах и повысить уровень удовлетворенности клиентов. Проект экономически обоснован и окупится менее чем за 1,5 года. Полученные результаты могут быть применены для модернизации других учетных систем в сфере ЖКХ.

Abstract

This bachelor's thesis presents the development of a comprehensive web-based subsystem for utility cost calculation within a real estate management framework. The project's primary objectives focus on enhancing computational accuracy, streamlining operational processes through automation, and optimizing user-system interaction to achieve greater efficiency and service transparency.

The research comprises an introductory section, three substantive chapters, concluding remarks, and supplementary appendices. The study employs contemporary system design methodologies, including object-oriented analysis and modular development principles, with particular emphasis on scalability and third-party service integration capabilities.

The core research challenge involves creating a functionally robust and economically sustainable subsystem that satisfies modern automation standards while maintaining operational flexibility. This investigation proves especially timely given the persistent shortcomings of manual calculation approaches, rigid tariff adjustment mechanisms, and inadequate payment system interoperability found in conventional solutions. The developed system demonstrates exceptional responsiveness to regulatory changes and tariff policy updates.

The opening chapter conducts an extensive domain analysis, examining the business workflows of "Kvartplata 24" while identifying pain points in their existing infrastructure. A detailed market comparison of alternative solutions precedes the formulation of comprehensive system requirements, derived through structured interviews with company personnel.

Chapter two establishes the conceptual foundation through detailed UML modeling (incorporating use case, sequence, and class diagrams) complemented by ERD database schemas. These visual representations effectively communicate the system architecture, component interactions, and data organization principles. The implemented database structure ensures optimal data integrity through rigorous normalization processes.

The implementation phase (Chapter three) documents the technological stack selection - HTML, CSS, and JavaScript for client-side development paired with SQLite Online for server operations. The discussion encompasses interface design considerations, database implementation specifics, and core functionality development for property management, service administration, and automated billing. Validation through practical calculation scenarios confirms operational reliability, while financial projections indicate a 19-month return on investment.

The concluding analysis confirms the subsystem's effectiveness in automating utility cost computations while improving result accuracy and user experience. Deployment at "Kvartplata 24" is projected to reduce operational expenditures by 25-30%, minimize calculation discrepancies, and significantly enhance customer satisfaction metrics. The proven economic viability and technical adaptability of the solution suggest potential applications for modernizing various utility accounting platforms across the housing sector.

Оглавление

Введение	8
Глава 1 Анализ предметной области.....	9
1.1 Характеристика деятельности компании	9
1.2 Анализ диаграммы верхнего уровня.	10
1.3 Внесение изменений в структуру ТО-ВЕ по сравнению с AS-IS.	15
1.4 Анализ существующих разработок для решения задачи	17
1.5 Функциональные требования проекта.	20
1.6 Бизнес-цели и требования проекта.....	22
Глава 2 Концептуальное моделирование системы для аналитической обработки корпоративной информации.....	24
2.1 Классы и формализация пользователей системы для аналитической обработки корпоративной информации.....	24
2.2 Описание функциональных требований системы для расчета коммунальных платежей.....	27
2.3 Разработка ERD-модели для подсистемы расчета коммунальных платежей.	31
Глава 3. Архитектура системы веб-приложения для расчета коммунальных платежей и особенности реализации	37
3.1. Системная архитектура веб-приложения для расчета коммунальных платежей	37
3.2 Информационная модель и ее описание	39
3.3 Разработка конечного продукта.....	49
3.4 Контрольный пример расчета коммунальных услуг	57
3.5 Расчет экономической эффективности	61
Заключение	63
Список используемой литературы и используемых источников	64

Введение

В ходе выпускной квалификационной работы по теме: «Разработка подсистемы расчета стоимости коммунальных услуг для системы управления объектами недвижимости». Основное внимание уделено созданию функционала, способствующего повышению точности расчетов и улучшению взаимодействия пользователей с системой.

Объектом исследования выступает процесс расчета стоимости коммунальных услуг на основе данных о потреблении ресурсов и тарифах компаний. Целью практики стало проектирование и разработка подсистемы расчета, обеспечивающей автоматизацию вычислений, отображение результатов и внедрение дополнительных инструментов для упрощения взаимодействия пользователей с системой.

Для достижения поставленной цели решены следующие задачи:

- изучение предметной области и существующих бизнес-процессов расчета коммунальных услуг;
- анализ текущей модели процессов расчета (AS-IS) и разработка целевой модели (TO-BE);
- создание прототипа подсистемы расчета с ключевой функциональностью;
- проведение анализа аналогичных решений для учета лучших практик;
- оценка затрат на разработку и определение сроков окупаемости проекта.

Разработанная подсистема позволяет автоматизировать процесс расчета стоимости коммунальных услуг, повысить прозрачность расчетов и удобство использования сервиса для пользователей.

управляющие элементы [20]. С помощью данной методологии была построена модель AS-IS, которая позволила детализировать текущие процессы.

1.2 Анализ диаграммы верхнего уровня.

Диаграмма верхнего уровня представляет общую модель процесса расчета коммунальных платежей.

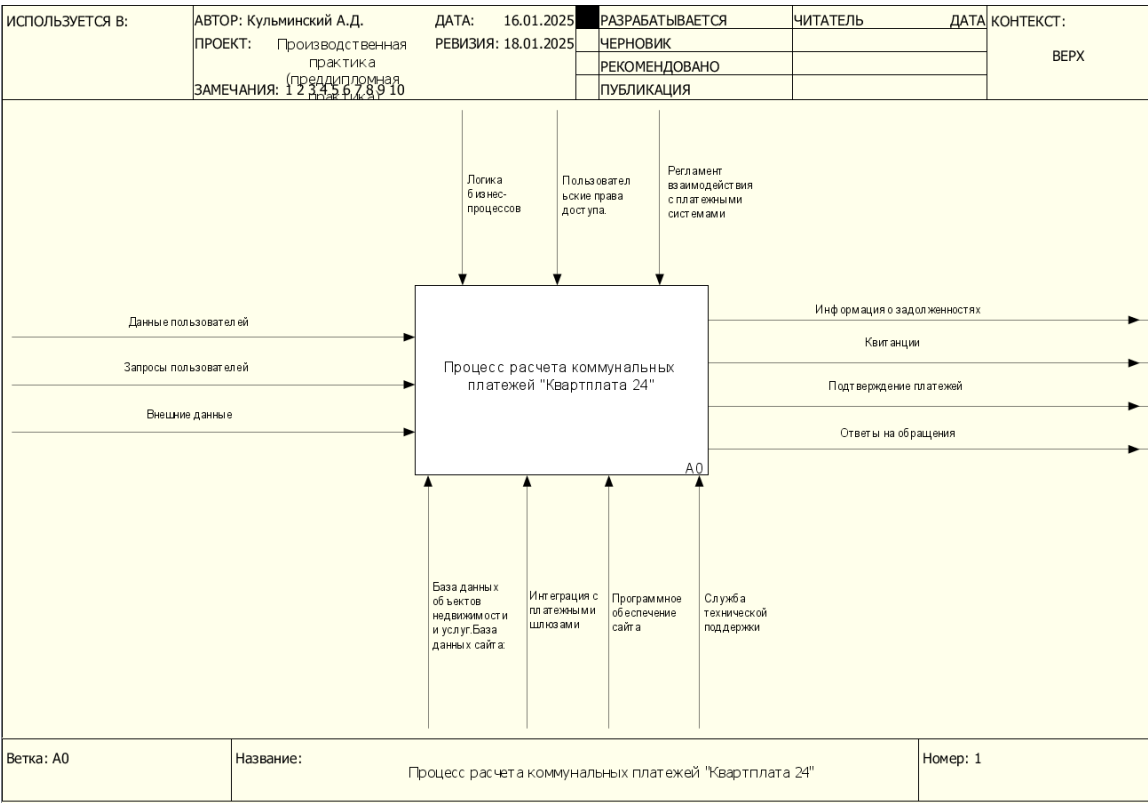


Рисунок 2- Контекстная диаграмма бизнес-процесса по расчету коммунальных платежей AS-IS

Контекстная диаграмма, предоставленная на рисунке 2, отражает взаимодействие системы расчёта коммунальных платежей с внешними сущностями. На входе представлены данные о пользователях, запросы пользователей, справочная информация, данные о состоянии объекта недвижимости и платежах. На выходе формируются отчёты, квитанции, подтверждения платежей и информационные данные о задолженности. Управление осуществляется через политики доступа и правила проведения

расчетов. В качестве механизма задействованы базы данных, интеграции с платёжными системами и службы поддержки. Диаграмма демонстрирует общую структуру процесса, но не детализирует взаимодействия между отдельными функциональными блоками.

Рассмотрим диаграмму более детально:

Входными потоками для бизнес-процесса по расчету коммунальных платежей являются:

- данные пользователей (информация о зарегистрированных лицах, объектах недвижимости, потребленных ресурсах);
- запросы пользователей (запросы на расчет платежей или загрузку квитанций);
- внешние данные (нормативные или тарифные данные от поставщиков коммунальных услуг).

Выходными потоками для бизнес-процесса по расчету коммунальных платежей являются:

- информация о задолженностях;
- квитанции;
- подтверждения платежей;
- ответы на обращения.

В своей работе компания руководствуется следующими документами:

- логику бизнес-процессов (например, правила расчета платежей, обработки запросов и работы с данными);
- пользовательские права доступа, которые определяют роли и допустимые действия пользователей;
- регламенты взаимодействия с платёжными системами.

Механизмами для бизнес-процесса по расчету коммунальных платежей являются:

- базы данных объектов недвижимости и услуг;
- программное обеспечение сайта «Квартплата 24»;

- интеграция с платежными шлюзами;
- служба технической поддержки.

На этом уровне система представлена как целостный процесс, требующий дальнейшей декомпозиции.

Диаграмма, показанная на рисунке 3 наглядно детализирует процесс работы с квитанциями. А также более подробно показывает структуру процесса.

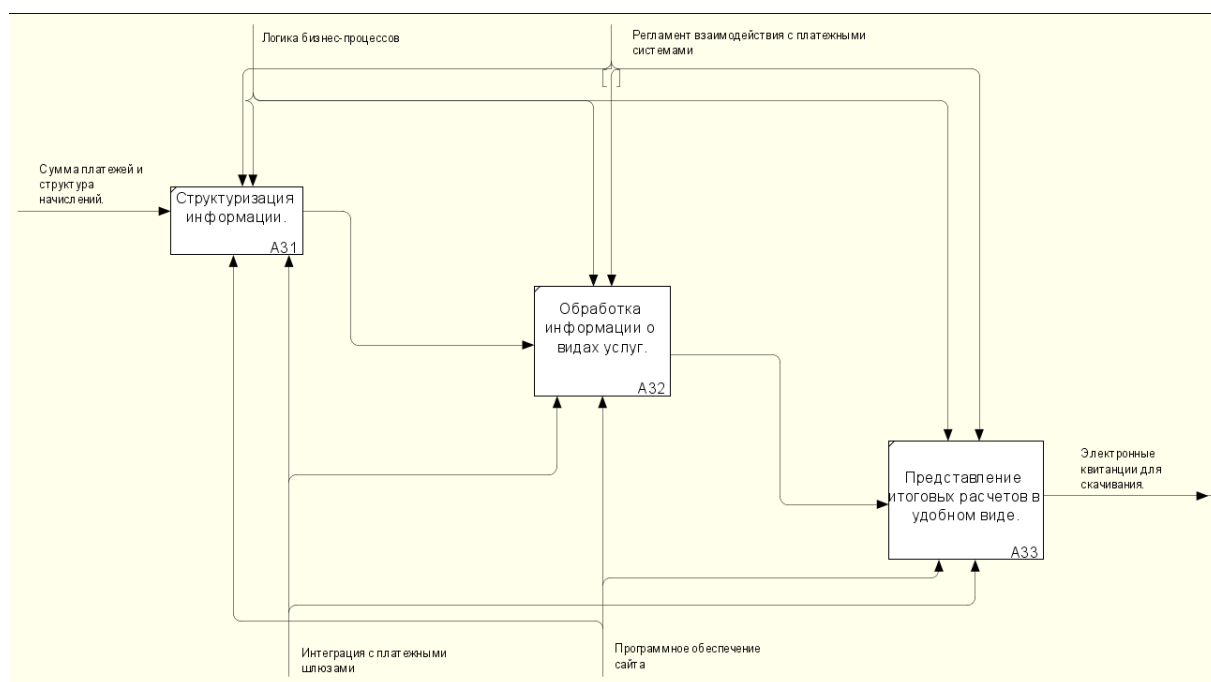


Рисунок 3 – Процесс «Работа с квитанциями AS-IS»

Процесс работы с квитанциями – это элемент системы расчета коммунальных платежей, который охватывает сбор данных, расчет стоимости услуг, формирование итогового документа и его предоставление пользователю. Основой этого процесса является корректная и своевременная обработка информации, связанной с характеристиками недвижимости, объемами потребляемых ресурсов и привязанными услугами.

Формирование квитанций начинается с того, что система собирает данные о потребленных ресурсах и параметрах недвижимости. Затем эти данные используются в расчетах по установленным формулам. В итоге

получается структурированная квитанция, где указаны все предоставленные услуги, их стоимость и итоговая сумма к оплате. Процесс полностью автоматизирован, что снижает риск ошибок, но требует точного ввода данных.

После того как квитанция сформирована, она отправляется пользователю. Возможны разные способы доставки: через электронную почту, в личный кабинет на сайте или в распечатанном виде. Такой вариант удобен тем, что каждый пользователь может выбрать наиболее подходящий способ получения. Кроме того, система фиксирует платежи, обновляя информацию о задолженностях.

Важно, чтобы процесс начисления был прозрачным. Пользователь может просмотреть все расчеты, что снижает количество вопросов и повышает доверие к системе.

Система также позволяет адаптировать расчеты – добавлять новые услуги или изменять формулы. Однако при крупных изменениях это может вызвать сложности, особенно если они затрагивают структуру всей системы.

Переход от текущей модели (AS-IS) к новой (TO-BE) направлен на устранение недостатков и улучшение расчетов коммунальных платежей. Сейчас система выполняет основные функции, но ей не хватает удобства и гибкости. Например, сложно вносить изменения в параметры недвижимости и услуг, а недостаток автоматизации увеличивает время работы и вероятность ошибок. Анализ по методике IDEF0 позволил выявить эти проблемы и разработать более эффективную модель.

В новой версии система становится удобнее и функциональнее. Возможность редактирования данных, изменения формул и добавления новых услуг делает процесс расчетов проще и точнее. Итоговая квитанция будет содержать все необходимые расчеты и суммы, что повысит прозрачность начислений и удобство для пользователей.

Текущая модель AS-IS не отвечает современным требованиям управления коммунальными платежами. Среди основных недостатков – ограниченная гибкость при изменении расчетных параметров, зависимость от

статических данных и сложность внесения корректировок в начисления. Из-за жестких связей между процессами их сложно адаптировать под новые тарифы, законодательные изменения и потребности пользователей.

Кроме того, существующая система недостаточно интегрирована с внешними сервисами и платежными системами, что затрудняет обработку платежей, увеличивает нагрузку на администраторов и снижает удобство для жильцов. Из-за неэффективности обработки данных и слабой связи между процессами возникают задержки в расчетах, ошибки в начислениях и дополнительные трудозатраты на их исправление.

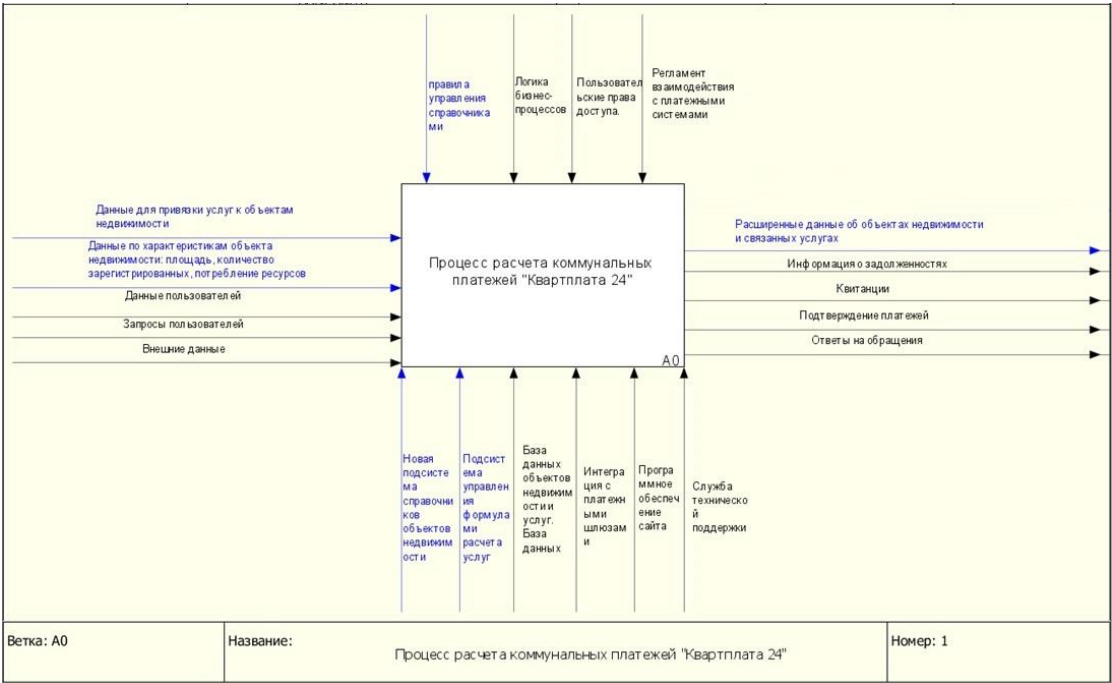


Рисунок 4 – «Диаграмма процесса по расчету коммунальных платежей ТО-ВЕ»

Для решения этих проблем предлагается новая структура ТО-ВЕ показанная на рисунке 4, в которой переработана логика взаимодействия между элементами системы. В обновленной модели процесс расчета коммунальных платежей становится более модульным и гибким, что позволяет легко изменять формулы расчета, обновлять данные об объектах и услугах, а также управлять начислениями в режиме реального времени.

Кроме того, ТО-ВЕ обеспечивает интеграцию с платежными шлюзами, что упрощает проведение платежей и их автоматическое подтверждение. Улучшенный механизм управления контентом сайта позволяет оперативно обновлять информацию о задолженностях и платежах, а новая система обратной связи делает взаимодействие пользователей с системой более эффективным.

Таким образом, новая структура ТО-ВЕ значительно повышает прозрачность, удобство и скорость работы системы, решая ключевые проблемы текущей модели и обеспечивая современный уровень автоматизации расчетов коммунальных платежей.

1.3 Внесение изменений в структуру ТО-ВЕ по сравнению с AS-IS.

В ходе модернизации системы расчета коммунальных платежей «Квартплата 24» были проведены структурные изменения, направленные на повышение гибкости, автоматизацию процессов и улучшение взаимодействия пользователей с системой. Ключевые изменения, внесенные в новую модель ТО-ВЕ по сравнению с исходной AS-IS, выделены синим цветом на диаграмме.

На рисунке 5 предоставлена диаграмма декомпозиции процесса IDEF0 расчета коммунальных платежей.

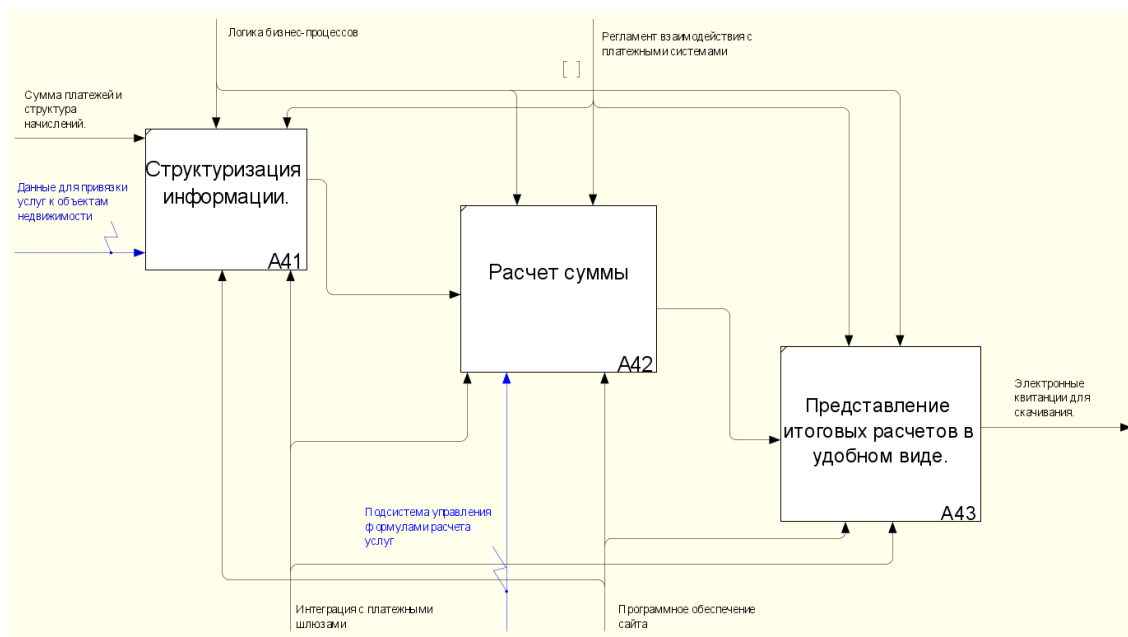


Рисунок 5 – «Диаграмма процесса по расчету коммунальных платежей ТО-ВЕ»

В новой модели добавлены расширенные данные об объектах недвижимости и связанных услугах, что позволяет системе учитывать больше характеристик недвижимости, таких как площадь, количество зарегистрированных лиц, объем потребляемых ресурсов и другие параметры. В AS-IS система использовала ограниченный набор характеристик, что сказывалось на точности расчетов. Теперь же возможна привязка услуг к объектам с учетом индивидуальных условий, что минимизирует ошибки и делает расчеты более прозрачными.

В структуре ТО-ВЕ реализована новая подсистема управления справочниками и расчетными формулами, которая позволяет администраторам оперативно изменять алгоритмы начислений без вмешательства в код системы. В AS-IS расчет выполнялся по фиксированным формулам, что ограничивало возможности гибкого управления начислениями. Теперь же возможно динамическое обновление данных в зависимости от тарифов, скидок, льгот и других факторов.

Ранее взаимодействие с платежными шлюзами было недостаточно автоматизировано. В новой модели ТО-ВЕ предусмотрена глубокая

интеграция с платежными шлюзами, что позволяет пользователям получать актуальные данные о платежах в режиме реального времени. Также автоматизировано подтверждение оплаты и обработка транзакций, что снижает нагрузку на службу поддержки и повышает удобство пользователей.

Ранее AS-IS формировала квитанции в статическом формате, что усложняло их обработку. В новой модели реализован механизм динамического формирования квитанций с учетом расчетных формул и параметров объектов, что делает их более информативными и наглядными. Теперь пользователи получают прозрачный расчет, в котором отображаются все начисления и итоговые суммы.

ТО-ВЕ включает в себя доработанный модуль управления контентом, который позволяет оперативно обновлять информацию о задолженностях, платежах и изменениях в расчетах. Также добавлена интерактивная система обратной связи, обеспечивающая оперативное предоставление ответов на запросы пользователей. Это повышает уровень обслуживания и снижает количество обращений в техническую поддержку.

1.4 Анализ существующих разработок для решения задачи

На основе анализа существующих решений было выбрано оптимальное направление для реинжиниринга системы расчета коммунальных платежей. Предложено заменить текущую линейную модель расчета и обработки данных на гибкую модульную систему с возможностью автоматического обновления параметров, интеграции с платежными шлюзами и расширенной системой управления справочниками. Новая структура обеспечит гибкость в настройке формул, прозрачность расчетов и удобство для пользователей, а также оптимизирует взаимодействие с внешними системами и минимизирует риск ошибок.

Были изучены платформы и инструменты, а также проведен анализ существующих решений в сфере автоматизации расчетов коммунальных

платежей. В качестве объектов исследования выбраны три популярных системы:

«1С: ЖКХ»

Программный продукт от компании 1С, предназначенный для автоматизации учета, начислений и расчетов коммунальных платежей в управляющих компаниях.

Критерии оценки:

- автоматизация расчетов (5/5): система поддерживает гибкие настройки тарифов, расчет по нормативам, учет льгот и перерасчет задолженностей;
- интеграция с внешними системами (4/5): возможна работа с ГИС ЖКХ, банковскими системами, ФНС, но требуется дополнительная настройка;
- простота использования (3/5): интерфейс сложен для неподготовленного пользователя, требуется обучение;
- гибкость настройки (4/5): формулы можно изменять, но для сложных расчетов нужны дополнительные модули;
- стоимость и доступность (3/5): высокая стоимость лицензии, требует внедрения и сопровождения.

Итоговая оценка: 19/25

«Биллинг ЖКХ»

Облачное решение для управляющих компаний и ТСЖ, предлагающее автоматизацию учета коммунальных услуг и интеграцию с платежными сервисами.

Критерии оценки:

- автоматизация расчетов (4/5): поддерживаются стандартные тарифные планы, но ограничена кастомизация формул;
- интеграция с внешними системами (5/5): Подключение к банковским шлюзам, онлайн-оплате, SMS-уведомлениям;

- простота использования (5/5): интуитивно понятный интерфейс, доступ с любого устройства через веб;
- гибкость настройки (3/5): пользователь может изменять только основные параметры, сложные тарифы требуют доработки;
- стоимость и доступность (4/5): подписная модель снижает первоначальные затраты, но есть зависимость от сервиса.

Итоговая оценка: 21/25

«Домопульт»

Платформа для управления жилищно-коммунальным хозяйством, рассчитанная на ТСЖ и управляющие компании.

Критерии оценки:

- автоматизация расчетов (3/5): основные функции присутствуют, но ограничены базовыми тарифами;
- интеграция с внешними системами (4/5): есть связь с платежными системами, но нет интеграции с ГИС ЖКХ;
- простота использования (5/5): удобный интерфейс, мобильное приложение, поддержка уведомлений;
- гибкость настройки (3/5): формулы изменить сложно, требуется поддержка разработчика;
- стоимость и доступность (5/5): доступные тарифные планы, возможность бесплатного тестирования.

Итоговая оценка: 20/25

Анализ существующих платформ и систем для расчета коммунальных платежей показал, что современные решения зачастую не обладают достаточной гибкостью, что создает неудобства как для управляющих компаний, так и для конечных пользователей. Во многих системах отсутствуют механизмы автоматического обновления данных, сложна настройка расчетных формул, а взаимодействие с платежными шлюзами требует значительных доработок. Это приводит к увеличению количества

ошибок в расчетах, задержкам в обработке платежей и низкому уровню прозрачности расчетов для пользователей.

На основе проведенного анализа можно сделать вывод, что оптимизация расчетной системы путем внедрения автоматизированного управления справочниками, гибкой настройки формул и интеграции с платежными шлюзами является перспективным направлением. Такой подход обеспечит прозрачность расчетов, снижение операционных затрат и повышение удобства взаимодействия с системой.

Таким образом, было принято решение о разработке обновленной системы расчета коммунальных платежей, в которой:

- будут автоматизированы процессы управления объектами недвижимости и услугами;
- появится возможность гибкого редактирования формул и параметров расчета;
- будет усилена интеграция с внешними платежными сервисами;
- реализуется расширенная отчетность по начислениям и платежам.

1.5 Функциональные требования проекта.

Основные функциональности:

- автоматизированное управление справочниками объектов недвижимости и услуг;
- возможность гибкого редактирования формул расчета коммунальных платежей;
- интеграция с платежными шлюзами для автоматической обработки платежей;
- формирование прозрачных расчетов с возможностью просмотра формул начисления;
- расширенная отчетность о начислениях, задолженностях и оплатах.

Пользовательские роли:

- администратор: управляет системой, настраивает формулы расчетов, добавляет объекты недвижимости и услуги;
- пользователь: просматривает начисления, оплачивает счета, получает уведомления о задолженностях;
- бухгалтер / сотрудник УК: контролирует начисления, корректирует расчеты, формирует отчетность.

Технические требования:

- разработка веб-платформы с возможностью интеграции с платежными шлюзами;
- поддержка масштабирования системы при увеличении количества объектов и пользователей;
- возможность автоматического обновления справочников и формул расчета.

Определим требования к веб-сайту для расчета коммунальных платежей в методологии FURPS+ и отобразим их в таблице 1 [2].

Таблица 1 – Требования к веб-сайту для расчета коммунальных платежей в методологии FURPS+

Вид требований	Содержание требований
F. Функциональные требования	- возможность ввода данных о жилье и коммунальных услугах;
	- автоматический расчет платежей;
	- формирование и скачивание квитанций;
U. Удобство использования	- адаптивный дизайн для разных устройств;
	- интуитивно понятный интерфейс;
	- наличие справочной информации по тарифам и услугам.
R. Надежность	- защита данных пользователей (шифрование, авторизация);
	- резервное копирование базы данных.
P. Производительность	- быстрая обработка расчетов;
	- оптимизированное взаимодействие с сервером.

Продолжение таблицы 1

Вид требований	Содержание требований
S. Поддерживаемость	- возможность обновления тарифов без изменения структуры сайта;
+. Ограничения	- поддержка стандартных форматов квитанций и платежных документов;
	- соответствие требованиям законодательства по обработке персональных данных.

Разработанная система для расчёта коммунальных платежей соответствует методологии FURPS+, обеспечивая удобный интерфейс, надёжность, производительность и гибкость в управлении начислениями, что делает её эффективным инструментом для пользователей и администраторов.

1.6 Бизнес-цели и требования проекта

Бизнес-цели:

- повышение точности и прозрачности расчетов коммунальных платежей за счет автоматизации и гибкой настройки системы;
- снижение операционных затрат за счет уменьшения ручного ввода данных и интеграции с платежными шлюзами;
- упрощение управления объектами недвижимости и услугами путем автоматизированного обновления справочников;
- увеличение удобства для конечных пользователей за счет возможности просмотра детализированных расчетов и получения актуальных данных о задолженностях.

Обновленные IT-требования:

- создание системы, обеспечивающей автоматизированный расчет коммунальных платежей с возможностью гибкого редактирования формул и параметров;
- обеспечение стабильной работы платформы при одновременном

доступе большого количества пользователей, включая управляющие компании и конечных клиентов;

- разработка удобного пользовательского интерфейса для всех ролей: администраторов, бухгалтеров, пользователей;
- интеграция с платежными системами для автоматического учета оплат и задолженностей.

Выводы по главе 1

Проведенное исследование предметной области позволило получить комплексное представление о современных подходах расчетов коммунальных платежей. В ходе детального анализа деятельности компании были выявлены проблемы существующей системы. Особое внимание было уделено сравнительному анализу моделей AS-IS и TO-BE, который наглядно продемонстрировал ключевые узкие места в текущей системе расчетов - недостаточную гибкость тарификации и ограниченные возможности адаптации к изменяющимся условиям. Всесторонний обзор существующих на рынке решений выявил существенный разрыв между предлагаемыми продуктами и реальными потребностями пользователей, что подчеркнуло необходимость разработки принципиально нового подхода к автоматизации расчетов коммунальных платежей. Полученные результаты легли в основу технического задания на разработку системы, определив ключевые требования к ее функциональности и архитектуре.

Глава 2 Концептуальное моделирование системы для аналитической обработки корпоративной информации

2.1 Классы и формализация пользователей системы для аналитической обработки корпоративной информации

В ходе проектирования информационной системы по теме ВКР «Разработка информационной системы для расчета коммунальных платежей» определено, что объем данных, поступающих в систему, является значительным. Для выпускной работы выделен процесс, который отвечает за расчет и анализ коммунальных платежей по объектам недвижимости, зарегистрированным в системе. Этот процесс охватывает все ключевые этапы взаимодействия пользователей с системой, от внесения данных об объектах недвижимости и тарифах до расчета стоимости услуг и их отображения в интерфейсе.

Для проектирования логической структуры системы используется технология моделирования UML (Unified Modeling Language).

UML (Unified Modeling Language) — это язык, позволяющий описывать архитектуру и поведение системы в виде диаграмм. UML является стандартом в индустрии программного обеспечения и широко применяется для проектирования информационных систем.

Одним из ключевых инструментов моделирования является диаграмма вариантов использования (Use Case Diagram). Данная диаграмма представляет функциональность системы на основе взаимодействия пользователей (актеров) с системой. Диаграмма прецедентов позволяет описать, какие функции доступны различным категориям пользователей.

Диаграмма вариантов использования для информационной системы расчета коммунальных платежей для компании «Квартплата 24» представлена на рисунке 6.

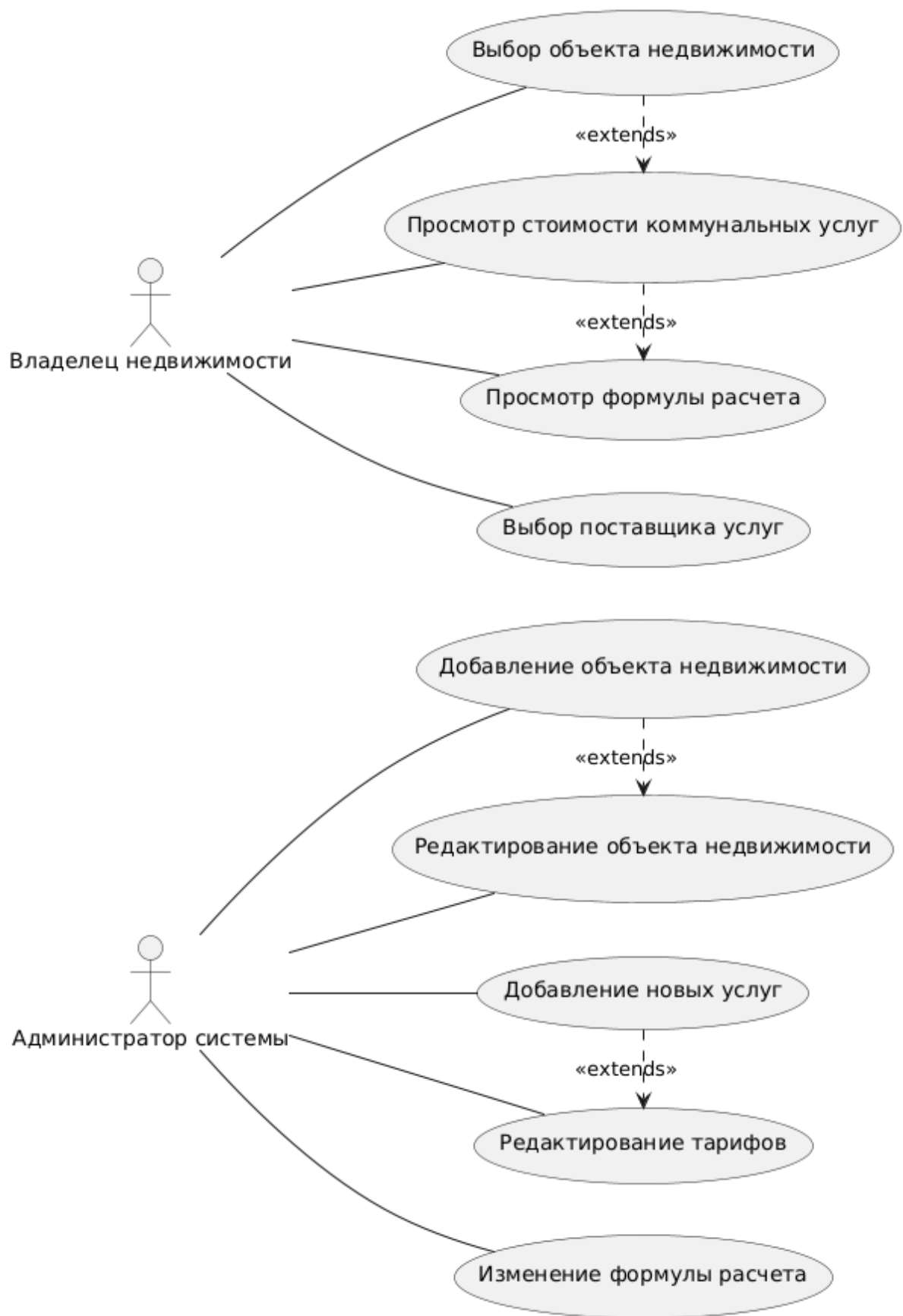


Рисунок 6 – UML диаграмма вариантов использования клиент-серверного приложения

Основные пользователи системы

Работа администратора в системе

1. Добавление нового объекта недвижимости

Администратор заходит в систему и открывает раздел «Объекты недвижимости», где представлена таблица всех зарегистрированных объектов.

Шаги для добавления нового объекта:

Вводится следующая информация:

- адрес;
- площадь (м^2);
- количество жильцов;
- потребление горячей и холодной воды (м^3);
- объем потребления газа (м^3);
- электроэнергия ($\text{кВт} \cdot \text{ч}$).

После внесения информации в поля нужно нажать на кнопку «Добавить объект», и система проверит данные и сохранит их в базе [16].

Дополнительные возможности:

- редактирование: администратор выбирает объект, вносит изменения и сохраняет их;
- удаление: система запрашивает подтверждение перед удалением данных.

2. Добавление поставщиков коммунальных услуг

Администратор переходит в раздел «Поставщики услуг», где отображается список всех ресурсоснабжающих компаний.

Шаги для добавления нового поставщика:

- вводится информация;
- название компании;
- район обслуживания;
- тарифы на воду, газ, электричество, отопление;
- региональный коэффициент (при наличии).

После заполнения формы нажимается «Добавить компанию», и данные сохраняются в базе.

Дополнительные возможности:

- редактирование: администратор может изменить информацию о поставщике и сохранить обновленные данные.

3. Настройка формул расчета коммунальных платежей

Администратор открывает раздел «Настройки расчета», где отображается текущая формула начислений.

Текущая формула: $(\text{Холодная вода} * \text{объем}) + (\text{Горячая вода} * \text{объем}) + (\text{Газ} * \text{объем}) + (\text{Электричество} * \text{объем}) + (\text{Отопление} * \text{площадь}) * \text{Количество жильцов} \times \text{Региональный коэффициент}$ [13].

После внесения изменений нажимается «Сохранить», и система обновляет параметры расчетов.

2.2 Описание функциональных требований системы для расчета коммунальных платежей.

Для описания работы системы будет использоваться диаграмма последовательности. Она помогает наглядно представить, как взаимодействуют компоненты системы при выполнении различных операций пользователем.

Основные действия в системе расчетов коммунальных платежей:

- просмотр информации о недвижимости
- выбор услуг и расчет их стоимости по тарифам
- оплата начисленных сумм
- просмотр детализации платежей

Диаграмма последовательности пользователя предоставлена на рисунке

7.

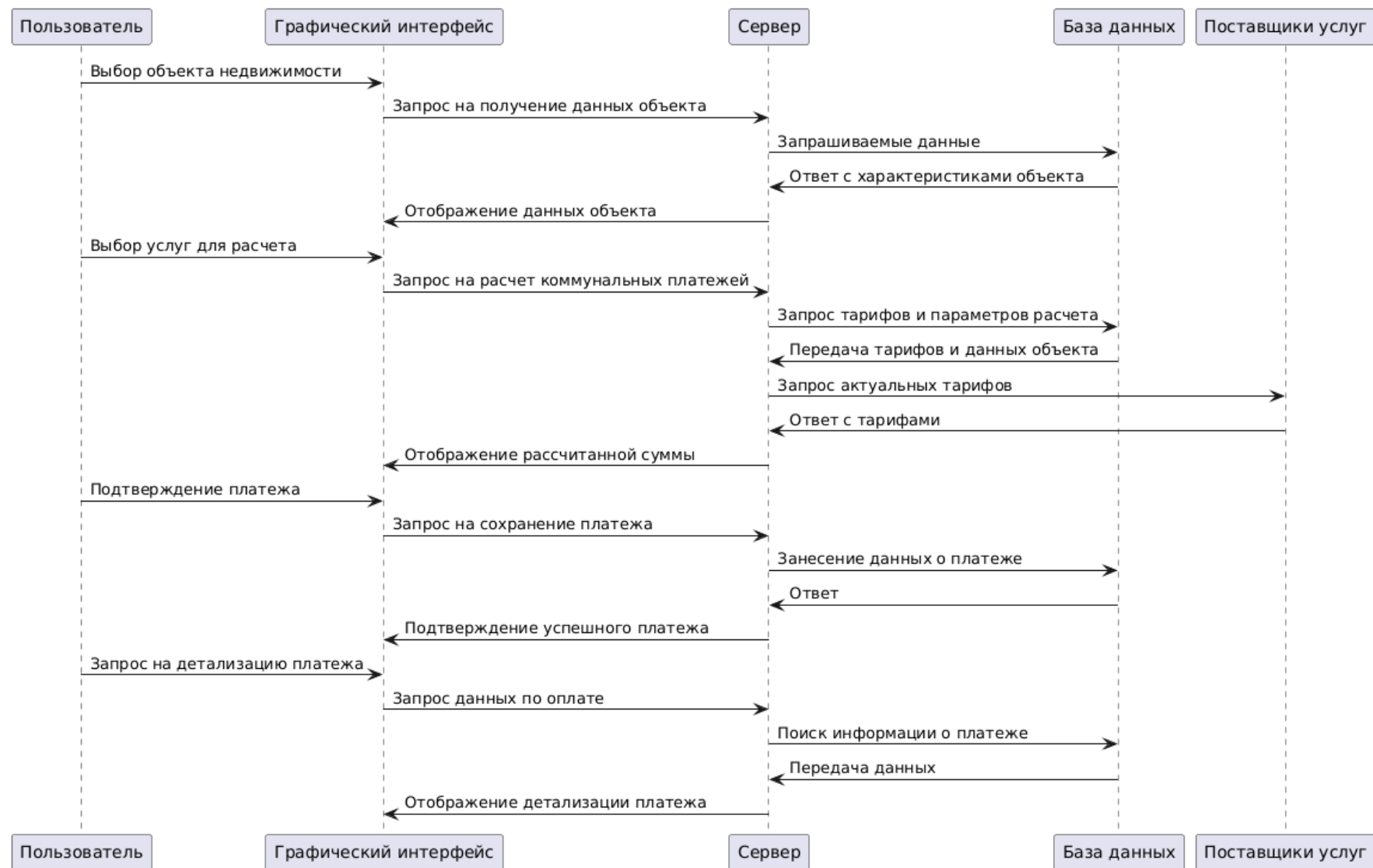


Рисунок 7 – UML диаграмма последовательности для серверной части

Алгоритм работы клиентской части приложения предоставляет пользователю (собственнику жилья, управляющей компании или поставщику услуг) удобный интерфейс для взаимодействия. На рисунке 8 предоставлена UML диаграмма последовательности клиентской части.

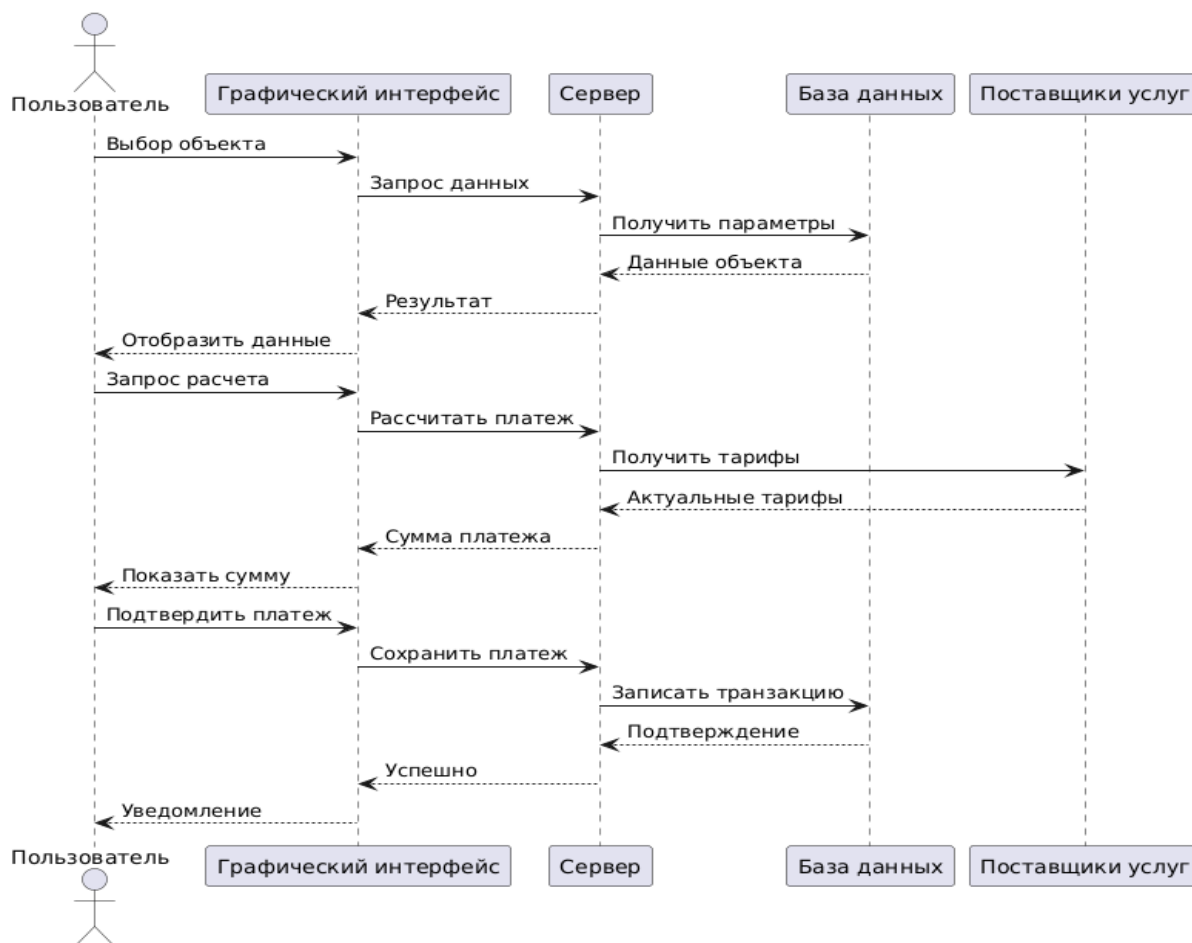


Рисунок 8 – UML диаграмма последовательности для клиентской части

После выделения объектов системы и определения основных функций и сценариев работы можно спроектировать диаграмму классов. Диаграмма классов — это структурная диаграмма языка моделирования UML, которая демонстрирует общую структуру иерархии классов системы, их атрибуты (поля), методы, интерфейсы и взаимосвязи между ними [21].

Диаграмма классов серверной части приложения для системы расчета коммунальных платежей представлена на рисунке 9.

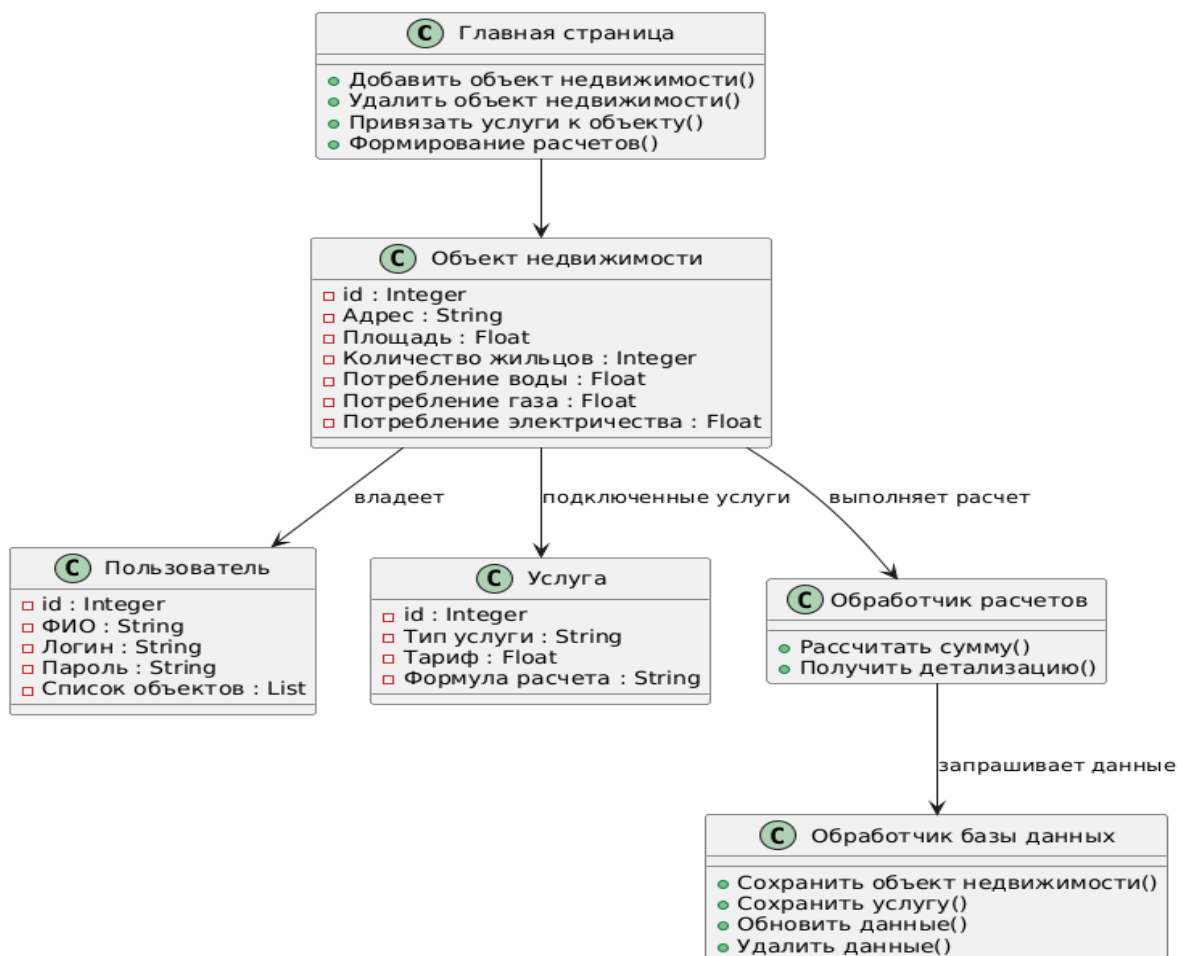


Рисунок 9 – UML диаграмма классов серверной части системы расчета коммунальных платежей

Диаграмма классов является ключевым инструментом для визуализации структуры клиент-серверного веб-приложения, предназначенного для расчета коммунальных платежей. Она отражает основные классы системы, их атрибуты, методы, а также связи между ними, что позволяет детально представить логику работы приложения, его функциональные компоненты и их взаимодействие.

2.3 Разработка ERD-модели для подсистемы расчета коммунальных платежей.

Одним из ключевых этапов моделирования информационной системы является разработка модели данных, которая определяет структуру хранения информации и взаимосвязи между различными элементами системы. В рамках данной выпускной квалификационной работы была выполнена разработка ERD-модели (Entity-Relationship Diagram — диаграммы «сущность-связь») для подсистемы расчета стоимости коммунальных услуг [9].

Рассмотрим логическую модель ERD расчета коммунальных платежей, которая представлена на рисунке 10.

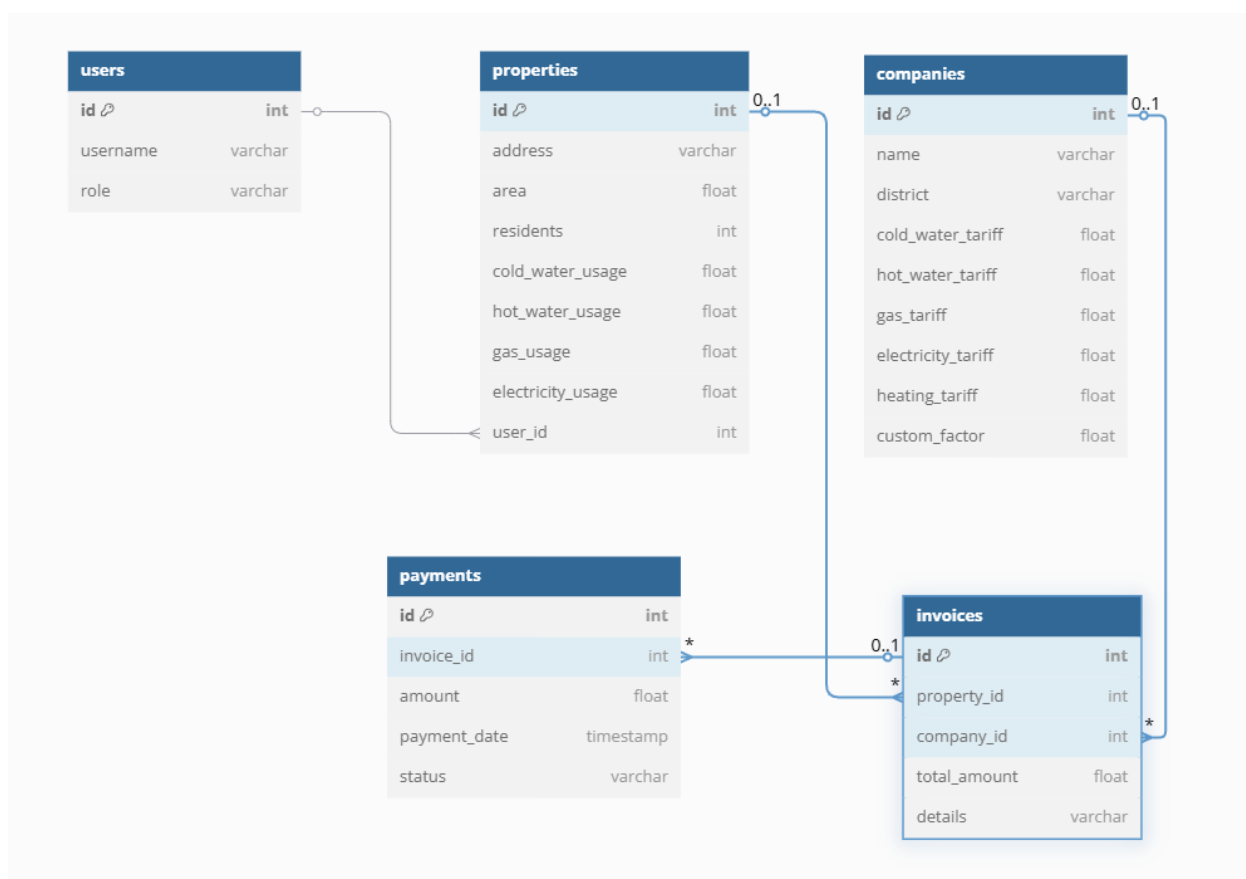


Рисунок 10 – логическая диаграмма ERD

Сущности (диаграммы ERD (логическая)):

users:

- id: Уникальный идентификатор пользователя (первичный ключ) ;
- username: Имя пользователя;
- role: Роль пользователя (например, администратор, пользователь, бухгалтер).

properties:

- id: Уникальный идентификатор объекта недвижимости (первичный ключ).
- address: Адрес объекта недвижимости;
- area: Площадь объекта (м²);
- residents: Количество прописанных жильцов;
- cold_water_usage: Потребление холодной воды (м³);
- hot_water_usage: Потребление горячей воды (м³);
- gas_usage: Потребление газа (м³);
- electricity_usage: Потребление электроэнергии (кВт·ч) ;
- user_id: Внешний ключ, связывающий объект недвижимости с пользователем.

companies:

- id: Уникальный идентификатор компании (первичный ключ) ;
- name: Название компании;
- district: Район обслуживания компании;
- cold_water_tariff: Тариф на холодную воду;
- hot_water_tariff: Тариф на горячую воду;
- gas_tariff: Тариф на газ;
- electricity_tariff: Тариф на электричество;
- heating_tariff: Тариф на отопление;
- custom_factor: Коэффициент района (для корректировки тарифов).

invoices:

- id: Уникальный идентификатор квитанции (первичный ключ);

- `property_id`: Внешний ключ, связывающий квитанцию с объектом недвижимости;
- `company_id`: Внешний ключ, связывающий квитанцию с компанией;
- `total_amount`: Итоговая сумма к оплате;
- `details`: Детали расчета (например, разбивка по услугам).

`payments`:

- `id`: Уникальный идентификатор оплаты (первичный ключ);
- `invoice_id`: Внешний ключ, связывающий оплату с квитанцией;
- `amount`: Сумма оплаты;
- `payment_date`: Дата оплаты;
- `status`: Статус оплаты (например, «ожидание», «оплачено», «ошибка»).

На рисунке 11 показана физическая ERD диаграмма для расчета коммунальных платежей:

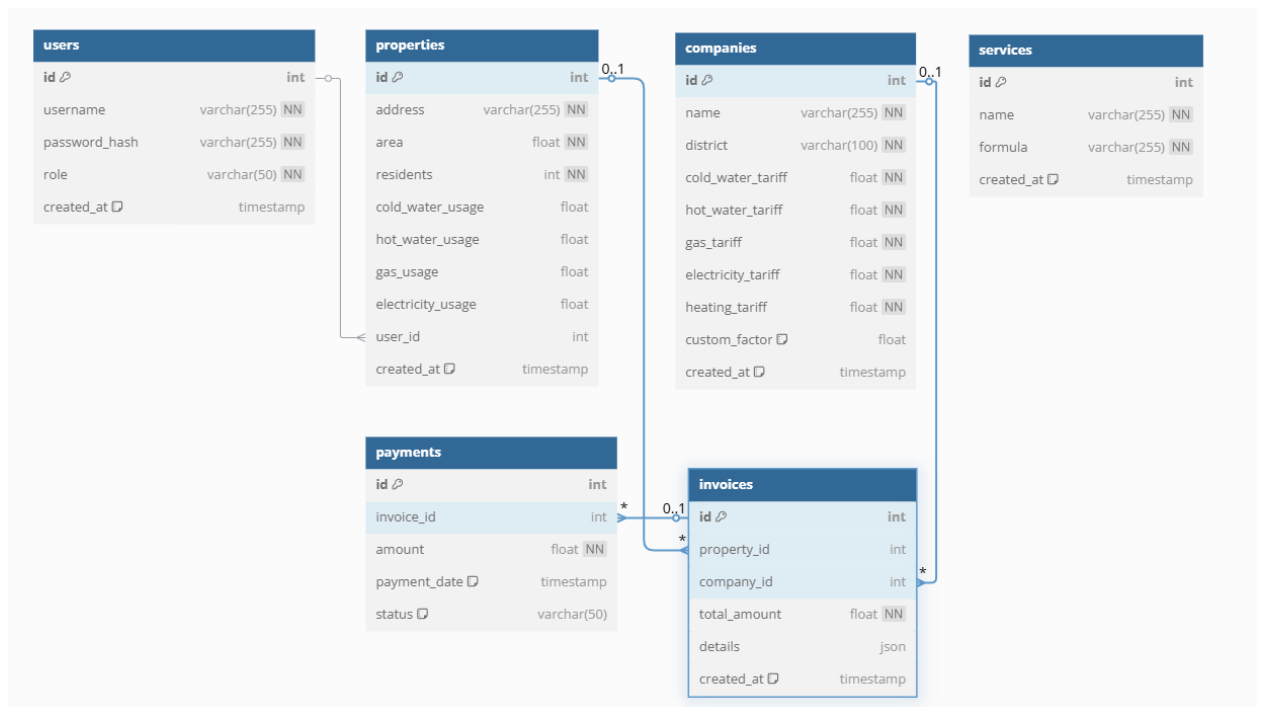


Рисунок 11 – ERD физическая диаграмма расчета коммунальных платежей.

Сущности (таблицы):

Users:

- id: Уникальный идентификатор пользователя (первичный ключ, int);
- username: Имя пользователя (varchar(255));
- password_hash: Хэш пароля (varchar(255));
- role: Роль пользователя (varchar(50));
- created_at: Дата создания записи (timestamp, значение по умолчанию).

Properties:

- id: Уникальный идентификатор объекта недвижимости (первичный ключ, int) ;
- address: Адрес объекта (varchar(255)) ;
- area: Площадь объекта (float);
- residents: Количество прописанных жильцов (int) ;
- cold_water_usage: Потребление холодной воды (float);
- hot_water_usage: Потребление горячей воды (float);
- gas_usage: Потребление газа (float);
- electricity_usage: Потребление электроэнергии (float);
- user_id: Внешний ключ, связывающий объект с пользователем (int);
- created_at: Дата создания записи (timestamp, значение по умолчанию).

companies:

- id: Уникальный идентификатор компании (первичный ключ, int);
- name: Название компании (varchar(255));
- district: Район обслуживания (varchar(100));
- cold_water_tariff: Тариф на холодную воду (float);
- hot_water_tariff: Тариф на горячую воду (float);
- gas_tariff: Тариф на газ (float);
- electricity_tariff: Тариф на электричество (float);
- heating_tariff: Тариф на отопление (float);

- custom_factor: Коэффициент района (float);
- created_at: Дата создания записи (timestamp).

services:

- id: Уникальный идентификатор услуги (первичный ключ, int);
- name: Название услуги (varchar(255));
- formula: Формула расчета стоимости услуги (varchar(255));
- created_at: Дата создания записи (timestamp, значение по умолчанию).

invoices:

- id: Уникальный идентификатор квитанции (первичный ключ, int);
- property_id: Внешний ключ, связывающий квитанцию с объектом недвижимости (int);
- company_id: Внешний ключ, связывающий квитанцию с компанией (int);
- total_amount: Итоговая сумма к оплате (float);
- details: Детали расчета (например, JSON с разбивкой по услугам);
- created_at: Дата создания записи (timestamp, значение по умолчанию).

payments:

- id: Уникальный идентификатор оплаты (первичный ключ, int);
- invoice_id: Внешний ключ, связывающий оплату с квитанцией (int);
- amount: Сумма оплаты (float);
- payment_date: Дата оплаты (timestamp, значение по умолчанию);
- status: Статус оплаты (varchar(50)) [15].

Выводы по главе 2

Во второй главе были рассмотрены вопросы концептуального моделирования системы для веб-приложения, предназначенного для расчета коммунальных платежей.

На функциональной схеме были выделены основные категории пользователей системы и их ключевые функции. Диаграмма последовательности позволила определить порядок взаимодействия пользователей с системой, включая ввод и обработку данных, а также генерацию итоговых расчетов.

Также были сформулированы основные требования к функциональности, надежности и удобству использования разрабатываемой системы, обеспечивающей автоматизацию расчетов коммунальных платежей, хранение информации о недвижимости и услугах, а также удобный интерфейс для пользователей.

Особое внимание было уделено проектированию структуры базы данных с использованием ERD-модели, что позволило обеспечить целостность данных и логическую организацию информации. Были определены ключевые сущности системы, такие как пользователи, объекты недвижимости, компании, квитанции и платежи, а также установлены связи между ними. Это стало важной основой для последующей реализации информационной системы.

Глава 3. Архитектура системы веб-приложения для расчета коммунальных платежей и особенности реализации

3.1. Системная архитектура веб-приложения для расчета коммунальных платежей

В данном разделе будут рассмотрены технологии и инструменты, использованные при разработке клиентской и серверной частей веб-приложения для расчета коммунальных платежей [17].

Для выбора стека технологий были проанализированы наиболее распространенные языки программирования и средства разработки, применяемые в веб-разработке. Так как разрабатываемая система является веб-приложением, основной акцент был сделан на использовании языков и технологий, обеспечивающих удобство создания интерактивных пользовательских интерфейсов и эффективное взаимодействие с сервером.

Ключевые требования к технологиям:

- простота и удобство разработки;
- кроссплатформенность и поддержка различных браузеров;
- высокая производительность и оптимизация работы с данными;
- возможность расширения и интеграции с другими сервисами.

Для реализации клиентской части веб-приложения был выбран HTML, CSS и JavaScript:

- HTML (HyperText Markup Language) – язык разметки, который используется для создания структуры веб-страницы;
- CSS (Cascading Style Sheets) – применяется для стилизации веб-интерфейса, позволяя управлять внешним видом и адаптивностью элементов;
- JavaScript – язык программирования, используемый для добавления интерактивности и динамических элементов на странице[1].

В качестве среды разработки была выбрана Visual Studio Code (VS Code).

Visual Studio Code – это мощный и удобный редактор кода, который обладает рядом преимуществ:

- поддержка множества языков программирования и плагинов;
- инструменты для работы с HTML, CSS и JavaScript;
- интегрированная система контроля версий Git;
- расширяемость благодаря множеству плагинов.

Для организации структуры веб-приложения было создано три HTML-файла, каждый из которых отвечал за отдельные разделы системы, а также использовались отдельные CSS и JavaScript файлы для стилизации и логики взаимодействия.

Для серверной части приложения был выбран SQLite online в качестве серверного фреймворка[6].

Для хранения данных была выбрана база данных SQLite Online, поскольку она является легковесной реляционной СУБД и обладает следующими преимуществами [22]:

- простота в использовании и настройке;
- не требует развертывания сервера, что упрощает интеграцию в веб-приложения;
- высокая скорость работы за счёт хранения данных в одном файле;
- полная поддержка стандартизированных SQL-запросов[10];
- подходит для небольших и средних проектов, а также для тестирования и прототипирования.

Выбранный стек технологий:

Клиентская часть (frontend) :

- HTML;
- CSS;
- JavaScript;
- Visual Studio Code.

Серверная часть (backend):

- SQLite online [23].

Выбранный стек технологий позволяет создать удобное, производительное и масштабируемое веб-приложение для расчета коммунальных платежей.

3.2 Информационная модель и ее описание

База данных — это способ хранения и организации информации, при котором данные структурируются в таблицы и связываются между собой для удобного доступа и обработки.

При проектировании базы данных важно учитывать несколько ключевых факторов:

- производительность. Если система должна работать с большим объемом данных и обеспечивать быстрый доступ, важно выбирать технологии, способные справляться с высокой нагрузкой при минимальных затратах ресурсов;
- безопасность. Если в базе хранятся конфиденциальные данные, необходимо предусмотреть надежные методы защиты информации;
- функциональность. Выбор инструментов зависит от задач, которые должна решать система, поэтому важно учитывать возможности используемой СУБД [24];
- скорость обработки запросов. Для комфортной работы пользователей база данных должна максимально быстро и эффективно обрабатывать запросы и предоставлять нужную информацию без задержек.

Диаграмма развертывания представляет собой специализированный тип UML-диаграммы, который демонстрирует архитектуру функционирования веб-системы [14]. Она позволяет наглядно показать физическое расположение компонентов программного обеспечения, взаимодействие серверной и клиентской частей, а также связи между различными узлами системы. Такая

диаграмма полезна при развертывании проекта, поскольку она дает четкое представление о том, какие аппаратные и программные ресурсы необходимы для корректной работы веб-приложения.

Диаграмма развертывания веб-системы расчета коммунальных платежей представлена на рисунке 12.

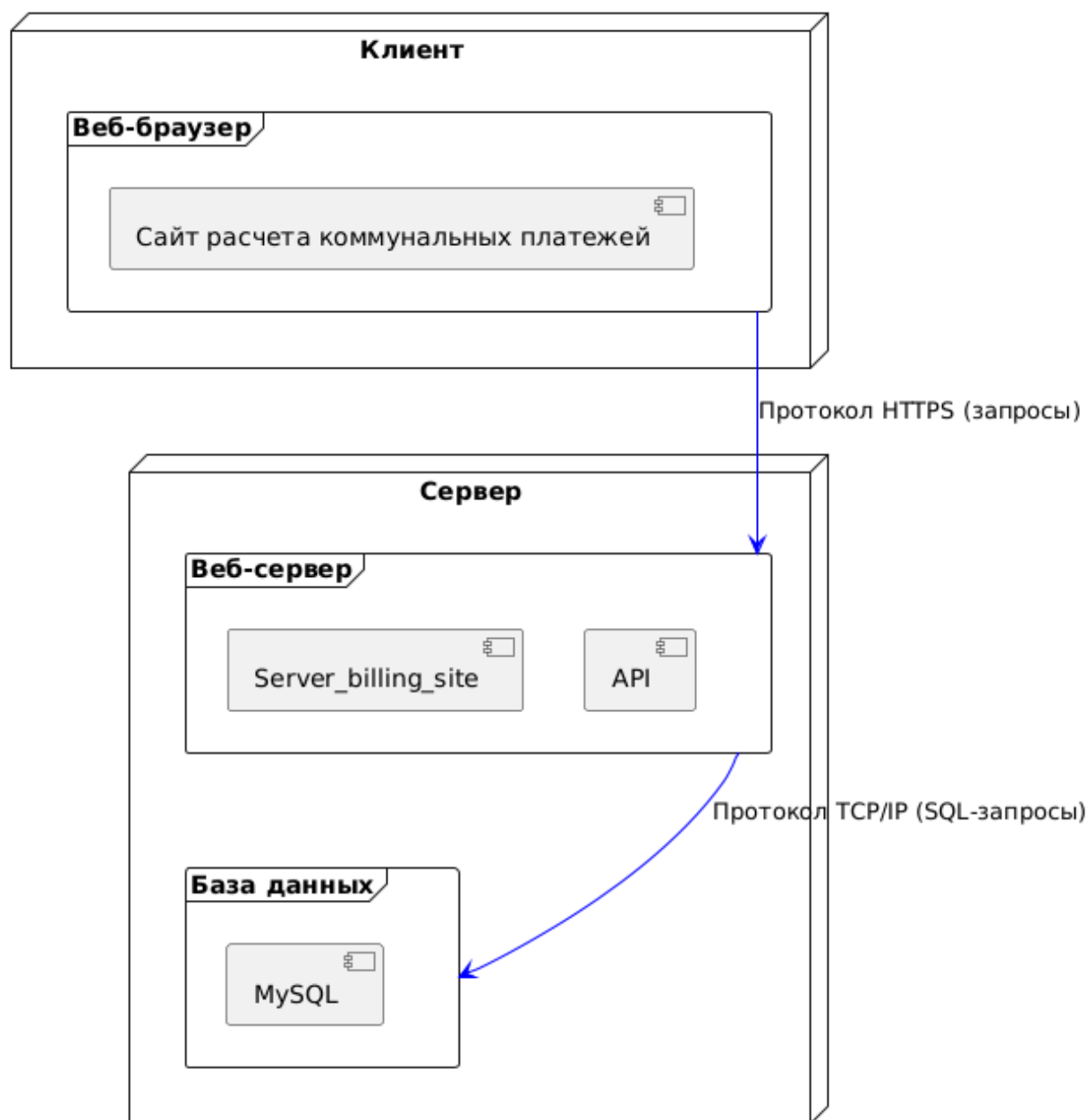


Рисунок 12 – UML диаграмма развертывания клиент-серверного веб-приложения для расчета коммунальных платежей

Данная диаграмма развертывания позволяет понять, как физически должна быть развернута система на аппаратном обеспечении. Клиент и сервер, представленные в виде кубов, являются физическими сущностями. Они связаны между собой протоколом HTTP [5].

Для системы расчета коммунальных платежей будет использоваться реляционная база данных. Прежде чем приступить к ее созданию, необходимо разработать логическую модель, которая определит основные элементы данных и их взаимосвязи. На рисунке 13 представлена логическая модель базы данных для данной системы [4].

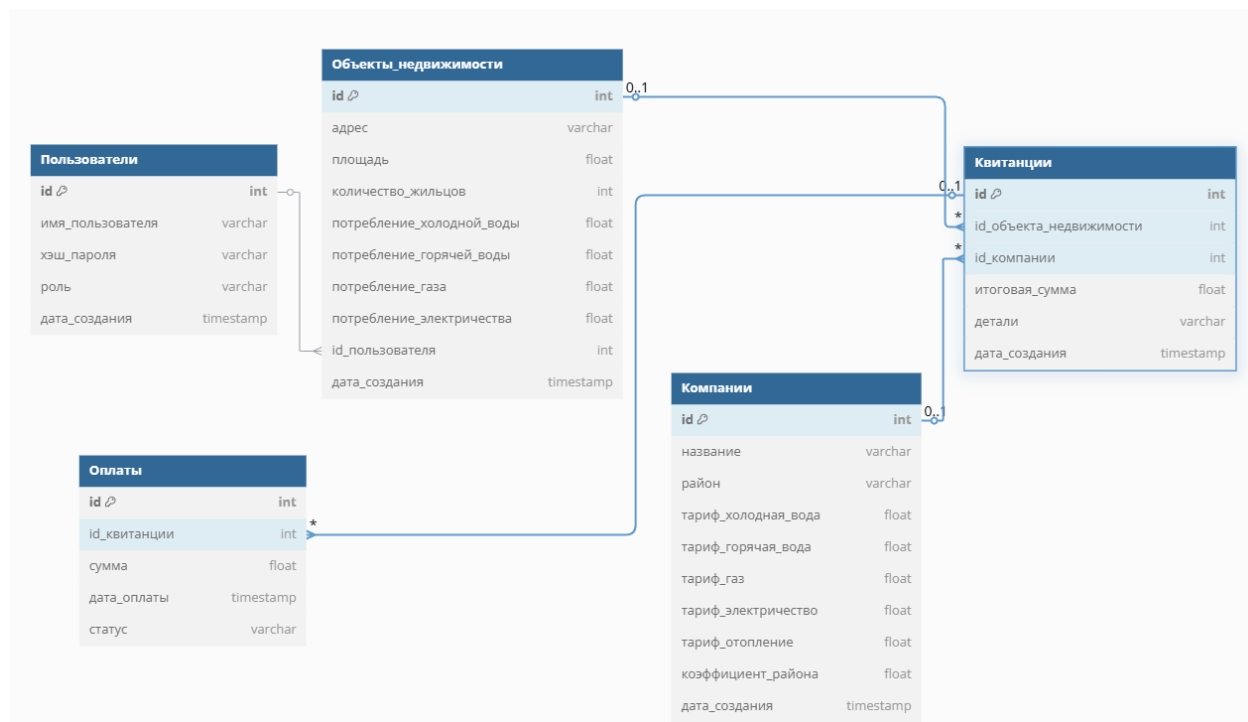


Рисунок 13 – логическая модель базы данных для системы расчета коммунальных платежей

Эта модель не привязана к конкретной СУБД и описывает структуру данных на концептуальном уровне.

Основные таблицы, которые будут использоваться в клиент-серверном приложении:

- «Пользователи» – содержит данные для авторизации и управления ролями (администратор, пользователь, бухгалтер);
- «Объекты недвижимости» – включает информацию о недвижимости: адрес, площадь, количество жильцов, объем потребленных ресурсов;
- «Компании» – хранит данные о поставщиках коммунальных услуг: название, обслуживаемый район, тарифы на услуги;
- «Квитанции» – фиксирует сформированные платежные документы, включая объект недвижимости, поставщика услуг, сумму и детали расчета;
- «Оплаты» – содержит информацию о произведенных платежах: связанная квитанция, сумма, дата и статус оплаты.

В таблицах 2 – 6 показаны атрибуты сущности логической модели базы данных:

Таблица 2 – Атрибуты сущности «Пользователи»

Поле	Тип данных	Ключ
id	int	PK
имя_пользователя	varchar	
хэш_пароля	varchar	
роль	varchar	
дата_создания	timestamp	

Таблица 3 – Атрибуты сущности «Объекты недвижимости»

Поле	Тип данных	Ключ
id	int	PK
адрес	varchar	
площадь	float	
количество_жильцов	int	
Потребление холодной воды	float	
потребление_горячей_воды	float	
потребление_газа	float	

Продолжение таблицы 3

потребление_электричества	float	
id_пользователя	int	FK
дата_создания	timestamp	

Таблица 4 – Атрибуты сущности «Компании»

Поле	Тип данных	Ключ
id	int	PK
название	varchar	
район	varchar	
тариф_холодная_вода	float	
тариф_горячая_вода	float	
тариф_газ	float	
тариф_электричество	float	
тариф_отопление	float	
коэффициент_района	float	
дата_создания	timestamp	

Таблица 5 – Атрибуты сущности «Квитанции»

Поле	Тип данных	Ключ
id	int	PK
id_объекта_недвижимости	int	FK
id_компании	int	FK
итоговая_сумма	float	
детали	varchar	
дата_создания	timestamp	

Таблица 6 – Атрибуты сущности «Оплаты»

Поле	Тип данных	Ключ
id	int	PK
id_квитанции	int	FK
сумма	float	
дата_оплаты	timestamp	
статус	varchar	

Разработка базы данных:

Разработка базы данных осуществлена с помощью языка SQL и SQLite Online [18]. Изначально было создано соединение с сервером базы данных, после чего выполнены следующие шаги:

- созданы таблицы с указанными атрибутами и связями;
- определены первичные (PK) и внешние ключи (FK) для обеспечения целостности данных;
- реализованы связи между таблицами (например, между «Пользователи» и «Объекты недвижимости», между «Квитанции» и «Оплаты»).

На рисунке 14 представлена иерархия структуры базы данных для системы расчета коммунальных платежей.

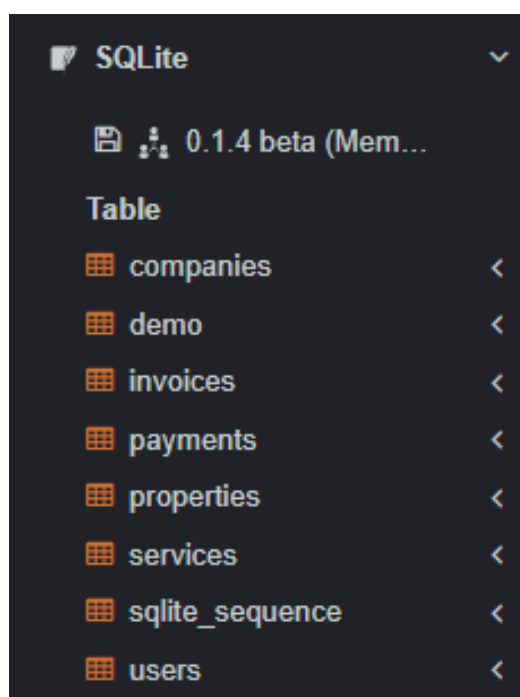
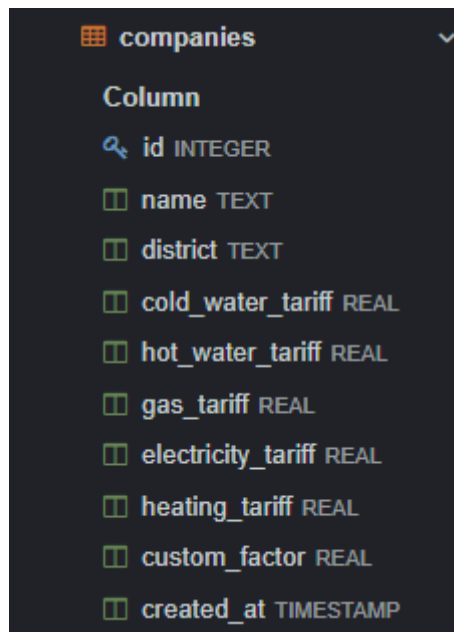


Рисунок 14 – Иерархия структуры БД

Для хранения информации о компаниях была разработана таблица «companies», которая представлена на рисунке 16.



Column
id INTEGER
name TEXT
district TEXT
cold_water_tariff REAL
hot_water_tariff REAL
gas_tariff REAL
electricity_tariff REAL
heating_tariff REAL
custom_factor REAL
created_at TIMESTAMP

Рисунок 15 – Структура таблицы «companies»

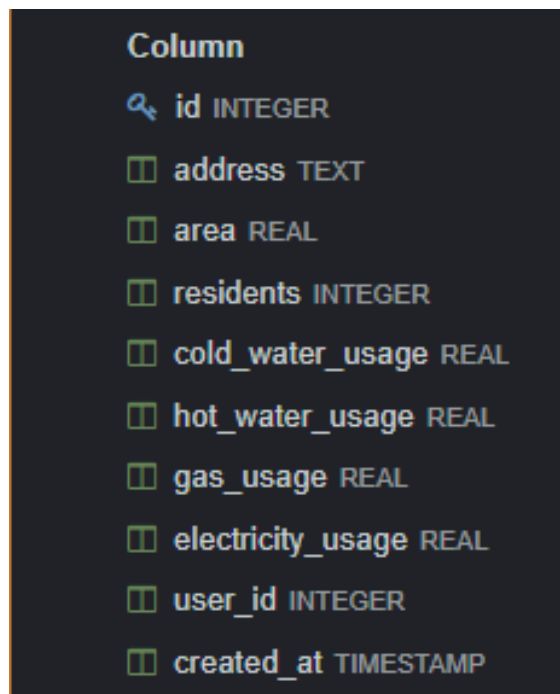
Перечень и описание полей:

- id (INTEGER) – уникальный идентификатор записи. Это поле является первичным ключом таблицы и используется для однозначной идентификации каждой компании;
- name (TEXT) – название компании. Поле предназначено для хранения текстового наименования компании;
- district (TEXT) – район, в котором расположена компания. Поле хранит информацию о географическом расположении компании;
- cold_water_tariff (REAL) – тариф на холодную воду. Поле содержит числовое значение, представляющее стоимость услуги;
- hot_water_tariff (REAL) – тариф на горячую воду. Аналогично предыдущему полю, хранит стоимость услуги;
- gas_tariff (REAL) – тариф на газ. Поле предназначено для хранения стоимости газоснабжения;
- electricity_tariff (REAL) – тариф на электроэнергию. Поле содержит информацию о стоимости электроэнергии;
- heating_tariff (REAL) – тариф на отопление. Поле хранит стоимость услуги отопления;

- custom_factor (REAL) – пользовательский фактор. Это поле может быть использовано для хранения дополнительных коэффициентов или параметров, необходимых для расчетов;
- created_at (TIMESTAMP) – временная метка создания записи. Поле автоматически фиксирует дату и время добавления записи в таблицу.

Данная структура позволяет эффективно хранить и управлять информацией о компаниях, включая их тарифы на различные услуги.

Для хранения информации о свойствах недвижимости или объектах была разработана таблица «properties», она предоставлена на рисунке 16.



Column
 id INTEGER
 address TEXT
 area REAL
 residents INTEGER
 cold_water_usage REAL
 hot_water_usage REAL
 gas_usage REAL
 electricity_usage REAL
 user_id INTEGER
 created_at TIMESTAMP

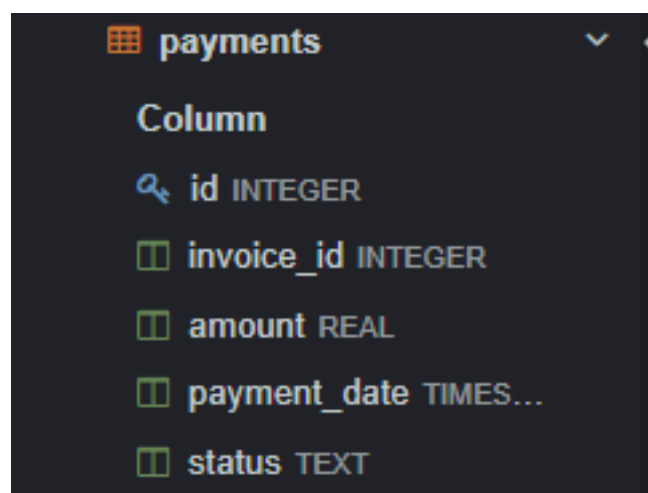
Рисунок 16 – Структура таблицы «properties»

Перечень и описание полей:

- id (INTEGER) – уникальный идентификатор записи. Это поле является первичным ключом таблицы и используется для однозначной идентификации каждого объекта;
- address (TEXT) – адрес объекта. Поле предназначено для хранения текстового описания местоположения объекта;

- area (REAL) – площадь объекта. Поле содержит числовое значение, представляющее площадь объекта в квадратных метрах;
- residents (INTEGER) – количество проживающих. Поле хранит информацию о числе людей, проживающих на объекте;
- cold_water_usage (REAL) – потребление холодной воды. Поле содержит числовое значение, представляющее объем потребления холодной воды;
- hot_water_usage (REAL) – потребление горячей воды. Аналогично предыдущему полю, хранит объем потребления горячей воды;
- gas_usage (REAL) – потребление газа. Поле предназначено для хранения объема потребления газа;
- electricity_usage (REAL) – потребление электроэнергии. Поле содержит информацию об объеме потребления электроэнергии;
- user_id (INTEGER) – идентификатор пользователя. Это поле является внешним ключом, который связывает объект с конкретным пользователем в системе;
- created_at (TIMESTAMP) – временная метка создания записи. Поле автоматически фиксирует дату и время добавления записи в таблицу.

Для управления информацией о платежах была разработана таблица «payments», которая представлена на рисунке 17.



Column
id INTEGER
invoice_id INTEGER
amount REAL
payment_date TIMES...
status TEXT

Рисунок 17 – Структура таблицы «payments»

Перечень и описание полей:

- a_id (INTEGER) – уникальный идентификатор записи. Это поле является первичным ключом таблицы и используется для однозначной идентификации каждого платежа.
- invoice_id (INTEGER) – идентификатор счёта. Это поле связывает платёж с конкретным счётом, выставляемым пользователю или компании;
- amount (REAL) – сумма платежа. Поле содержит числовое значение, представляющее сумму, которая была оплачена;
- payment_date (TIMESTAMP) – дата и время платежа. Поле фиксирует момент, когда был совершён платёж;
- status (TEXT) – статус платежа. Поле хранит текстовое описание текущего статуса платежа (например, «оплачено», «в обработке», «отклонено»).

Данная структура позволяет эффективно управлять информацией о платежах, включая их связь с выставленными счетами, суммы и статусы.

Чтобы показать, как разные «сущности» связаны между собой внутри системы, была использована EER диаграмма. Разработанная системы расчета коммунальных платежей представлена на рисунке 18.

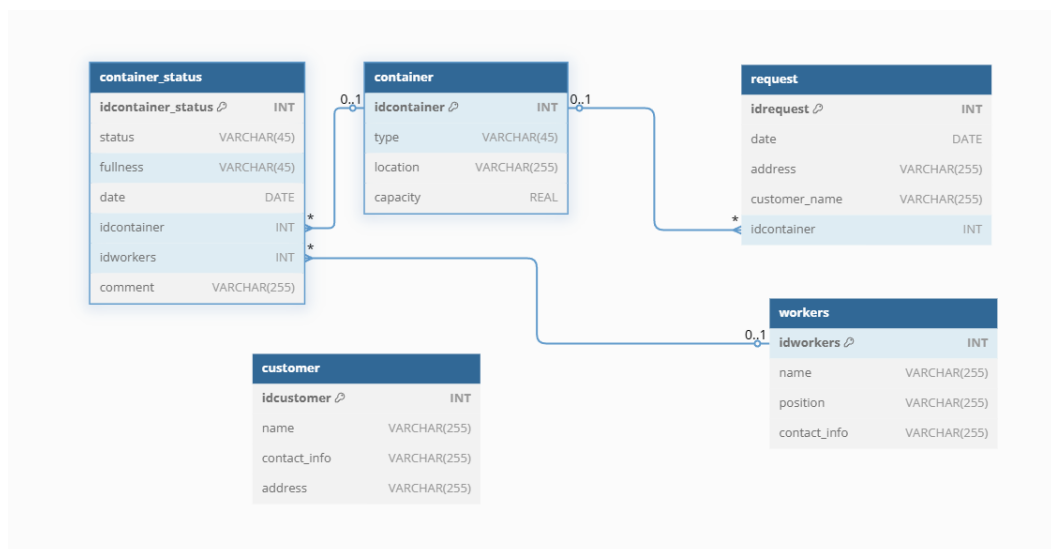


Рисунок 18 – Физическая схема базы данных

Эта схема представляет собой базу данных для системы управления контейнерами. Она включает таблицы для отслеживания статуса контейнеров (container_status), информации о самих контейнерах (container), заявок на обслуживание (request), данных о сотрудниках (workers) и клиентах (customer). Связи между таблицами позволяют контролировать заполненность контейнеров, назначение работников и обработку заявок.

3.3 Разработка конечного продукта.

Постановка задачи:

Основная цель проекта – создание удобного веб-интерфейса для пользователей, который позволит [12]:

- добавлять и редактировать объекты недвижимости;
- вносить информацию о поставщиках коммунальных услуг;
- рассчитывать стоимость услуг в зависимости от тарифов и характеристик недвижимости;
- хранить данные в базе данных SQLite,

Выбор технологий:

Для реализации проекта были выбраны следующие технологии:

- HTML, CSS – для создания пользовательского интерфейса [3];
- JavaScript – для обработки данных на клиентской стороне;
- SQLite online – в качестве базы данных для хранения объектов недвижимости и поставщиков услуг.

Разработка структуры базы данных

Для хранения данных используется база данных SQLite на рисунке 19 предоставлен код базы данных.

```

1 CREATE TABLE users (
2     id INTEGER PRIMARY KEY AUTOINCREMENT,
3     username TEXT NOT NULL UNIQUE,
4     password_hash TEXT NOT NULL,
5     role TEXT NOT NULL,
6     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
7 );
8
9 CREATE TABLE properties (
10    id INTEGER PRIMARY KEY AUTOINCREMENT,
11    address TEXT NOT NULL,
12    area REAL NOT NULL,
13    residents INTEGER NOT NULL,
14    cold_water_usage REAL,
15    hot_water_usage REAL,
16    gas_usage REAL,
17    electricity_usage REAL,
18    user_id INTEGER REFERENCES users(id),
19    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
20 );
21
22 CREATE TABLE companies (
23    id INTEGER PRIMARY KEY AUTOINCREMENT,
24    name TEXT NOT NULL,
25    district TEXT NOT NULL,
26    cold_water_tariff REAL NOT NULL,
27    hot_water_tariff REAL NOT NULL,
28    gas_tariff REAL NOT NULL,
29    electricity_tariff REAL NOT NULL,
30    heating_tariff REAL NOT NULL,
31    custom_factor REAL DEFAULT 1.0,
32    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
33 );
34
35 CREATE TABLE services (
36    id INTEGER PRIMARY KEY AUTOINCREMENT,
37    name TEXT NOT NULL,
38    formula TEXT NOT NULL,
39    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
40 );
41
42 CREATE TABLE invoices (
43    id INTEGER PRIMARY KEY AUTOINCREMENT,
44    property_id INTEGER REFERENCES properties(id),
45    company_id INTEGER REFERENCES companies(id),
46    total_amount REAL NOT NULL,
47    details TEXT,
48    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
49 );
50
51 CREATE TABLE payments (
52    id INTEGER PRIMARY KEY AUTOINCREMENT,
53    invoice_id INTEGER REFERENCES invoices(id),
54    amount REAL NOT NULL,
55    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
56    status TEXT DEFAULT 'pending'
57 );

```

Рисунок 19 – Код базы данных SQLite online

Разработка клиентской части

Интерфейс объектов недвижимости.

Пользовательский интерфейс реализован с помощью HTML, CSS и JavaScript [8].

На рисунках 20-21 предоставлен HTML код для страницы заполнения информации об объектах недвижимости.

```
1 <!DOCTYPE html>
2 <html lang="ru">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Объекты недвижимости</title>
8   <link rel="stylesheet" href="styles.css">
9 </head>
10
11 <body>
12   <div class="container">
13     <h2>Добавить объект недвижимости</h2>
14     <form id="propertyForm">
15       <!-- Адрес -->
16       <div class="form-group">
17         <label for="address">Адрес:</label>
18         <input type="text" id="address" placeholder="Введите адрес" required>
19       </div>
20
21       <!-- Район города -->
22       <div class="form-group">
23         <label for="district">Район города:</label>
24         <select id="district" required>
25           <option value="">Выберите район</option>
26           <option value="старый">Старый</option>
27           <option value="новый">Новый</option>
28         </select>
29       </div>
30
31       <!-- Площадь и количество прописанных -->
32       <div class="form-group">
33         <label for="area">Площадь (м²):</label>
34         <input type="number" id="area" placeholder="Площадь (м²)" min="1" required>
35       </div>
36       <div class="form-group">
37         <label for="residents">Кол-во прописанных:</label>
38         <input type="number" id="residents" placeholder="Кол-во человек" min="1" required>
39       </div>
40
41       <!-- Услуги -->
42       <div class="form-group row">
43         <label>Холодная вода</label>
44         <input type="checkbox" id="coldWaterEnabled" checked onclick="toggleField('coldWaterUsage')">
45         <input type="number" id="coldWaterUsage" placeholder="Количество м³" min="0">
46       </div>
47       <div class="form-group row">
48         <label>Горячая вода</label>
```

Рисунок 20 – Код HTML для заполнения информации о недвижимости

```

49     <input type="checkbox" id="hotWaterEnabled" checked onclick="toggleField('hotWaterUsage')">
50     <input type="number" id="hotWaterUsage" placeholder="Количество м³" min="0">
51 </div>
52 <div class="form-group row">
53     <label>Газ</label>
54     <input type="checkbox" id="gasEnabled" checked onclick="toggleField('gasUsage')">
55     <input type="number" id="gasUsage" placeholder="Количество м³" min="0">
56 </div>
57 <div class="form-group row">
58     <label>Отопление</label>
59     <input type="checkbox" id="heatingEnabled" checked onclick="toggleField('houseArea')">
60 </div>
61
62 <!-- Кнопки -->
63
64 <button type="submit" class="btn btn-save">Сохранить</button>|
65 <a href="companies.html" class="btn btn-legal">Перейти к юр. лицам</a>
66 </form>
67
68 </div>
69
70 <script src="script.js"></script>
71 </body>
72
73 </html>

```

Рисунок 21 – Код HTML интерфейса для заполнения информации о недвижимости

Интерфейс юридических компаний и их услуг.

На рисунке 22 предоставлен HTML код для страницы заполнения информации об юридических компаниях и их услугах

```

1  <!DOCTYPE html>
2  <html lang="ru">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Объекты недвижимости</title>
8      <link rel="stylesheet" href="styles.css">
9  </head>
10
11 <body>
12     <div class="container">
13         <h2>Добавить объект недвижимости</h2>
14         <form id="propertyForm">
15             <!-- Адрес -->
16             <div class="form-group">
17                 <label for="address">Адрес:</label>
18                 <input type="text" id="address" placeholder="Введите адрес" required>
19             </div>
20
21             <!-- Район города -->
22             <div class="form-group">
23                 <label for="district">Район города:</label>
24                 <select id="district" required>
25                     <option value="">Выберите район</option>
26                     <option value="старый">Старый</option>
27                     <option value="новый">Новый</option>
28                 </select>
29             </div>
30
31             <!-- Площадь и количество прописанных -->
32             <div class="form-group">
33                 <label for="area">Площадь (м²):</label>
34                 <input type="number" id="area" placeholder="Площадь (м²)" min="1" required>
35             </div>
36             <div class="form-group">
37                 <label for="residents">Кол-во прописанных:</label>
38                 <input type="number" id="residents" placeholder="Кол-во человек" min="1" required>
39             </div>
40
41             <!-- Услуги -->
42             <div class="form-group row">
43                 <label>Холодная вода</label>
44                 <input type="checkbox" id="coldWaterEnabled" checked onclick="toggleField('coldWaterUsage')>
45                 <input type="number" id="coldWaterUsage" placeholder="Количество м³" min="0">
46             </div>
47             <div class="form-group row">
48                 <label>Горячая вода</label>
49                 <input type="checkbox" id="hotWaterEnabled" checked onclick="toggleField('hotWaterUsage')>
50                 <input type="number" id="hotWaterUsage" placeholder="Количество м³" min="0">
51             </div>
52             <div class="form-group row">
53                 <label>Газ</label>
54                 <input type="checkbox" id="gasEnabled" checked onclick="toggleField('gasUsage')>
55                 <input type="number" id="gasUsage" placeholder="Количество м³" min="0">
56             </div>
57             <div class="form-group row">
58                 <label>Отопление</label>
59                 <input type="checkbox" id="heatingEnabled" checked onclick="toggleField('houseArea')>
60             </div>
61
62             <!-- Кнопки -->
63
64             <button type="submit" class="btn btn-save">Сохранить</button>
65             <a href="companies.html" class="btn btn-legal">Перейти к юр. лицам</a>
66         </form>
67     </div>
68
69     <script src="script.js"></script>
70 </body>
71 </html>

```

Рисунок 22 – Код HTML интерфейса для заполнения информации об юридических компаниях и их услугах

Интерфейс расчета стоимости коммунальных услуг.

На рисунке 23 предоставлен HTML код для страницы расчета стоимости коммунальных услуг.

```
1 <!DOCTYPE html>
2 <html lang="ru">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Расчет стоимости</title>
8   <link rel="stylesheet" href="styles.css">
9 </head>
10
11 <body>
12   <div class="container">
13     <h2>Расчет коммунальных услуг</h2>
14     <form id="calculateForm">
15       <label>Выберите объект недвижимости:</label>
16       <select id="selectedProperty"></select>
17
18       <label>Выберите поставщика услуг:</label>
19       <select id="selectedCompany"></select>
20
21       <button type="submit">Рассчитать</button>
22     </form>
23     <div id="result"></div>
24     <a href="index.html">Назад</a>
25   </div>
26
27   <script>
28     document.addEventListener("DOMContentLoaded", function () {
29       let properties = JSON.parse(localStorage.getItem("properties")) || [];
30       let companies = JSON.parse(localStorage.getItem("companies")) || [];
31
32       let propertySelect = document.getElementById("selectedProperty");
33       let companySelect = document.getElementById("selectedCompany");
34
35       properties.forEach((property, index) => {
36         let option = document.createElement("option");
37         option.value = index;
38         option.textContent = property.address;
39         propertySelect.appendChild(option);
40       });
41
42       companies.forEach((company, index) => {
43         let option = document.createElement("option");
44         option.value = index;
45         option.textContent = company.name;
46         companySelect.appendChild(option);
47       });
48     });
49
50     document.getElementById("calculateForm").addEventListener("submit", function (event) {
51       event.preventDefault();
52       let properties = JSON.parse(localStorage.getItem("properties")) || [];
53       let companies = JSON.parse(localStorage.getItem("companies")) || [];
54
55       let property = properties[document.getElementById("selectedProperty").value];
56       let company = companies[document.getElementById("selectedCompany").value];
57
58       if (!property || !company) {
59         alert("Выберите объект и компанию!");
60         return;
61       }
62
63       let coefficient = company.customFactor || 1; // Берем коэффициент района
64
65       let coldWaterCost = property.coldWater * company.tariffs.coldWater;
66       let hotWaterCost = property.hotWater * company.tariffs.hotWater;
67       let gasCost = property.gas * company.tariffs.gas;
68       let heatingCost = property.area * company.tariffs.heating;
69       let totalCost = (coldWaterCost + hotWaterCost + gasCost + heatingCost) * property.residents * coefficient;
70
71       document.getElementById("result").innerHTML = `
72         <h2>Итоговая стоимость: ${totalCost.toFixed(2)} Р</h2>
73       `;
74     });
75   </script>
76 </body>
77 </html>
```

Рисунок 23 – Код HTML интерфейса для расчета итоговой стоимости коммунальных услуг

Расчет стоимости коммунальных услуг.

Внесение информации об объекте недвижимости.

На рисунке 24 отображен пользовательский интерфейс внесения информации об объектах недвижимости.

The screenshot shows a mobile application interface with a title bar at the top that reads "Добавить объект недвижимости" (Add real estate object). Below the title bar, there are several input fields and checkboxes. The first field is labeled "Адрес:" (Address:) and contains the placeholder text "Введите адрес" (Enter address). The second field is labeled "Район города:" (City district:) and contains the placeholder text "Выберите район" (Select district) with a dropdown arrow. The third field is labeled "Площадь (м²):" (Area (m²):) and contains the placeholder text "Площадь (м²)". The fourth field is labeled "Кол-во прописанных:" (Number of registered:) and contains the placeholder text "Кол-во человек" (Number of people). Below these fields, there are four sections, each with a label, a checked checkbox, and a text input field. The first section is labeled "Холодная вода" (Cold water) and has a checked checkbox and a field labeled "Количество м³" (Quantity m³). The second section is labeled "Горячая вода" (Hot water) and has a checked checkbox and a field labeled "Количество м³". The third section is labeled "Газ" (Gas) and has a checked checkbox and a field labeled "Количество м³". The fourth section is labeled "Отопление" (Heating) and has a checked checkbox. At the bottom of the form, there are two green buttons: "СОХРАНИТЬ" (Save) and "ПЕРЕЙТИ К ЮР. ЛИЦАМ" (Go to legal entities).

Рисунок 24 – интерфейс внесения информации об объекте недвижимости

Внесение информации об компании и предоставляемых услугах.

На рисунке 25 предоставлен пользовательский интерфейс внесения информации об компаниях и предоставляемых ими услугах.

The form is titled "Добавление компании" (Add company) and is set against a background image of a city at night. It contains the following fields and sections:

- Название компании:** A text input field.
- Район города:** A dropdown menu with "Старый город" (Old town) selected.
- Коэффициент района:** A text input field containing the value "0,6".
- Тарифы на услуги (₽)** (Service tariffs (₽)) section:
 - Холодная вода:** A text input field.
 - Горячая вода:** A text input field.
 - Газ:** A text input field.
 - Электричество:** A text input field.
 - Отопление:** A text input field.
- СОХРАНИТЬ КОМПАНИЮ** (Save company) button.
- Navigation buttons at the bottom:
 - ВЕРНУТЬСЯ НАЗАД** (Go back) button in green.
 - [ПЕРЕЙТИ К РАСЧЕТУ](#) (Go to calculation) link in purple.

Рисунок 25 – интерфейс внесения информации об компаниях и предоставляемых услугах

Выбор объекта недвижимости и компании. Итоговый расчет всех услуг.

На рисунке 26 предоставлен пользовательский интерфейс итогового расчета всех услуг.

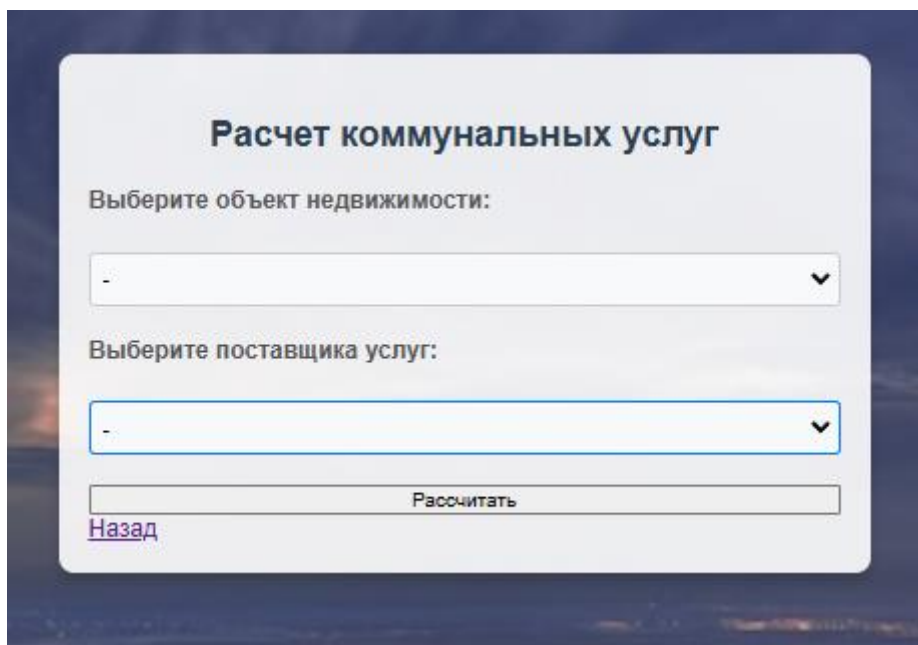


Рисунок 26 – интерфейс Выбор объекта недвижимости и компании.
Итоговый расчет всех услуг

Интерфейс для расчёта коммунальных платежей удобен и практичен: он позволяет быстро вводить данные, автоматически рассчитывать суммы к оплате и наглядно отображать результаты. Простое меню и понятная структура делают работу с системой комфортной как для пользователей, так и для сотрудников.

3.4 Контрольный пример расчета коммунальных услуг

Данный контрольный пример демонстрирует процесс расчета стоимости коммунальных услуг в разработанном веб-приложении. Он иллюстрирует удобство выбора объектов недвижимости и поставщиков, автоматическое применение тарифов и расчета итоговой суммы, что делает использование системы простым и интуитивно понятным.

Описание процесса

Пользователь ул. Пушкина, д. 25, заходит на сайт веб-приложения для расчета коммунальных услуг.

Выбор объекта и поставщика услуг

На открывшейся странице пользователь видит список принадлежащих ему объектов. Он выбирает свою квартиру в старом городе (Пушкина, д. 25), далее система предлагает ему указать поставщика коммунальных услуг. В списке доступных организаций он выбирает «ПетербургЖКХ», тарифы которого загружаются автоматически:

- холодная вода – 55 руб./м³;
- горячая вода – 160 руб./м³;
- газ – 25 руб./м³;
- отопление – 35 руб./м²;
- районный коэффициент – 1.2.

После выбора поставщика система автоматически загружает сохраненные данные о потребленных ресурсах за месяц, хранящиеся в базе данных (Рисунок 27) [11].

Добавить объект недвижимости

Адрес:

пушкина, д25

Район города:

Старый

Площадь (м²):

100

Кол-во прописанных:

4

Холодная вода

☒

10

Горячая вода

☒

5

Газ

☒

18

Отопление

☐

СОХРАНИТЬ

ПЕРЕЙТИ К ЮР. ЛИЦАМ

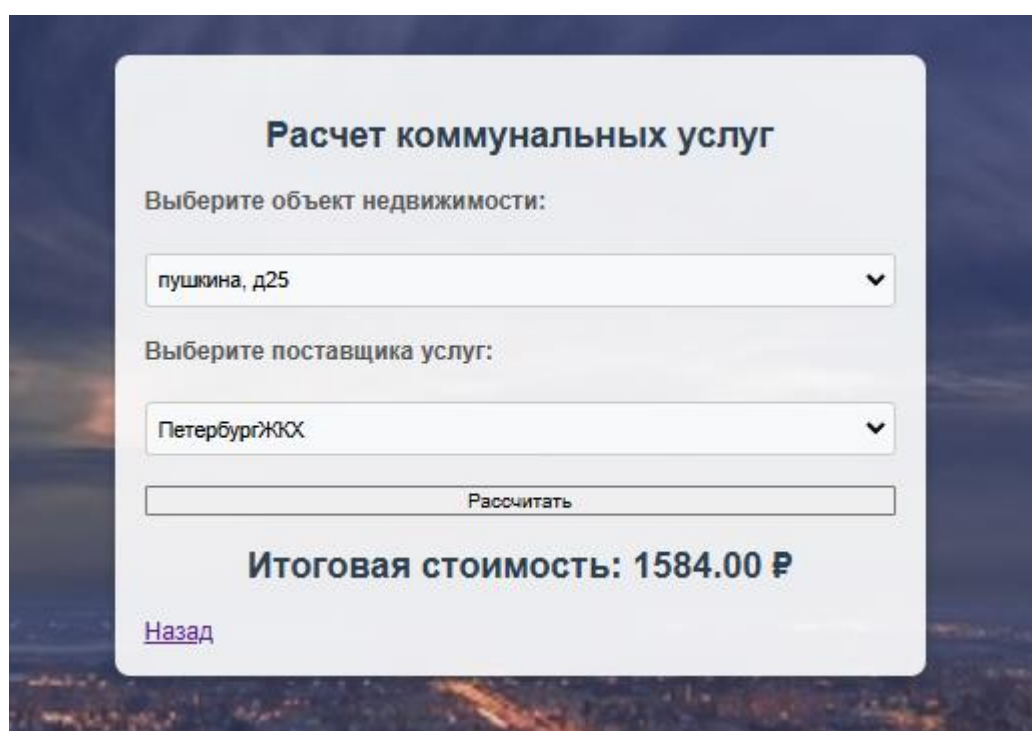
Рисунок 27 – пример внесенной информации об объекте недвижимости

Автоматический расчет стоимости услуг

На основании внесенных данных расчет стоимости услуг выглядит следующим образом:

- холодная вода: $(10 \times 55) \times 1.2 = 660$ руб.;
- горячая вода: $(5 \times 160) \times 1.2 = 384$ руб.;
- газ: $(18 \times 25) \times 1.2 = 540$ руб..

После выполнения всех расчетов система выводит сообщение: «Итоговая стоимость: 1584 руб.» (Рисунок 28).



Расчет коммунальных услуг

Выберите объект недвижимости:

пушкина, д25

Выберите поставщика услуг:

ПетербургЖКХ

Рассчитать

Итоговая стоимость: 1584.00 Р

[Назад](#)

Рисунок 28 – пример расчета коммунальных услуг

Приведенный контрольный пример показывает, как использование базы данных SQLite, автоматического расчета и удобного пользовательского интерфейса помогает быстро и безошибочно вычислить сумму коммунальных платежей. Система автоматически загружает данные о потребленных ресурсах, применяет тарифы и выводит итоговый расчет, что избавляет пользователей от необходимости делать сложные вычисления вручную []. Такой подход повышает удобство работы с сервисом и снижает вероятность

ошибок, делая процесс расчета коммунальных услуг более эффективным и доступным.

3.5 Расчет экономической эффективности

Для оценки экономической выгоды от внедрения системы расчетов коммунальных платежей были учтены основные статьи затрат:

- оплата труда специалистов, участвующих в разработке;
- амортизация оборудования, используемого в процессе работы;
- затраты на электроэнергию для работы оборудования.

Таблица 9 – Расчет затрат на оплату труда

Должность	Оплата за 1 час (руб.)	Трудозатраты (часы)	Общая сумма (руб.)
Программист	1100	140	154 000
Системный аналитик	850	90	76 500
UI/UX дизайнер	750	70	52 500
Тестировщик	650	100	65 000
Менеджер проекта	950	50	47 500
Итого:			395 500

Затраты на оплату труда специалистов составляют 395 500 руб.

С учетом налоговых отчислений (30,2%), сумма увеличивается до:

$$395\,500 \times 1,302 = 514\,041 \text{ руб.}$$

Амортизация оборудования

Срок службы оборудования – 5 лет (60 месяцев). Если стоимость техники составляет 180 000 руб., а разработка системы занимает 2 месяца, то амортизация за этот период составит:

$$18000060 \times 2 = 6000 \text{ руб.}$$

Расходы на электроэнергию:

Средняя мощность оборудования 0,6 кВт, время работы 160 часов, при тарифе 5,8 руб./кВт·ч, стоимость электроэнергии составит:

$$0,6 \times 160 \times 5,8 = 556 \text{ руб.}$$

Общий расчет затрат:

$$514041 + 6000 + 556 = 520597 \text{ руб.}$$

Предполагаемая экономия за счет внедрения системы составит 330 000 руб. в год.

Окупаемость проекта составит: $520597 / 330000 \approx 1,58$ года или 19 месяцев.

Вывод по главе 3

Таким образом, система полностью окупится менее чем за 1,5 года, после чего начнет приносить чистую экономическую выгоду.

Контрольный пример подтвердил корректность расчетов и удобство использования интерфейса, а экономический анализ показал, что проект является рентабельным и окупится менее чем за 1,5 года. Таким образом, внедрение данной подсистемы позволит компании ООО «Квартплата 24» повысить эффективность расчетов, снизить операционные издержки и улучшить качество обслуживания клиентов.

Заключение

В рамках выполнения выпускной квалификационной работы была разработана подсистема расчета коммунальных услуг для веб-приложения, позволяющего пользователям учитывать и анализировать платежи за жилищно-коммунальные услуги. Основное внимание уделено созданию удобного и функционального инструмента, который автоматизирует процесс расчетов и делает работу с системой более прозрачной и эффективной.

В ходе работы были выполнены следующие задачи:

Проведен анализ предметной области и изучены существующие методы расчета коммунальных платежей. Это позволило определить ключевые требования к функционалу системы.

Разработана структура базы данных на основе SQLite, обеспечивающая хранение информации об объектах недвижимости, поставщиках коммунальных услуг и потребляемых ресурсах.

Разработан интерфейс веб-приложения, включающий формы для ввода данных, автоматизированный расчет коммунальных платежей и систему отображения истории расчетов.

Проведено тестирование работы системы, в ходе которого были устранены ошибки и оптимизированы механизмы обработки данных.

В результате проделанной работы созданное приложение автоматизирует процесс расчета коммунальных услуг, упрощает взаимодействие пользователей с системой и исключает необходимость выполнения сложных расчетов вручную. Благодаря использованию базы данных SQLite обеспечивается надежное хранение информации.

Разработанная система способствует повышению точности вычислений, удобству использования и прозрачности учета коммунальных платежей, что делает ее полезным инструментом для владельцев недвижимости и управляющих компаний.

Список используемой литературы и используемых источников

1. Абрамов, С. А. JavaScript. Основы JavaScript / С. А. Абрамов. – Москва : МЦНМО, 2015. – 128 с.
2. Андреев А. В. Автоматизация бизнес-процессов: Учебное пособие. – М.: Инфра-М, 2020.
3. Бадюк, А. С. Web-программирование: учебное пособие / А. С. Бадюк, О. В. Дрокин. – Москва : Юрайт, 2023. – 195 с.
4. Васин, В. В. Базы данных: разработка, проектирование, сопровождение / В. В. Васин, А. П. Кузьменко. – Москва : ДМК Пресс, 2022. – 320 с.
5. Гаджиев, Г. И. Проектирование и разработка Web-приложений / Г. И. Гаджиев. – Москва : Лань, 2018. – 252 с.
6. Джейкобсон, А. SQL. Справочник / А. Джейкобсон. – Москва : ДМК Пресс, 2019. – 400 с.
7. Дёмин С. А., Кузнецова Е. В. Информационные технологии в управлении: Практическое руководство. – СПб.: Питер, 2021.
8. Зиновьев, Д. В. JavaScript. Энциклопедия / Д. В. Зиновьев. – Санкт-Петербург : Питер, 2017. – 320 с.
9. Карпова, И. П. Основы проектирования баз данных : учебное пособие / И. П. Карпова. – Санкт-Петербург : Университет ИТМО, 2016. – 123 с.
10. Кузнецов, М. В. SQL. Руководство разработчика / М. В. Кузнецов. – Москва : Бином-Пресс, 2020. – 480 с.
11. Лебедев, В. Н. JavaScript. Практика профессионального программирования / В. Н. Лебедев. – Москва : Эксмо, 2021. – 512 с.
12. Мокшин, И. В. Разработка Web-приложений на JavaScript / И. В. Мокшин. – Москва : ДМК Пресс, 2016. – 272 с.

13. Немцова, Т. И. Программирование на языке высокого уровня JavaScript : учебное пособие / Т. И. Немцова, О. Ю. Голова, А. В. Старых. – Москва : Форум, 2022. – 160 с.
14. Семенов, А. Н. Разработка Web-приложений / А. Н. Семенов. – Санкт-Петербург : БХВ-Петербург, 2019. – 768 с.
15. Степанов, А. Н. Базы данных / А. Н. Степанов. – Москва : КноРус, 2017. – 408 с.
16. Фленов, М. Е. JavaScript. Подробное руководство / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2020. – 512 с.
17. Харченко В. С. Проектирование и разработка программных систем. – Киев: Логос, 2019.
18. Холландер, Л. SQL. Полное руководство / Л. Холландер. – Москва : Вильямс, 2018. – 496 с.
19. Официальный сайт компании «Квартплата 24» [Электронный ресурс]. URL: <https://kvartplata24.ru/>
20. Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice (3rd ed.). Addison-Wesley Professional.
21. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of Business Process Management. Springer.
22. Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley Professional.
23. Kreibich, J. A. (2010). Using SQLite. O'Reilly Media.
24. Rozanski, N., & Woods, E. (2011). Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives (2nd ed.). Addison-Wesley Professional.