

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)

Разработка социальных и экономических информационных систем

(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка информационной системы учета объектов и предоставляемых услуг

Обучающийся

Н.А. Нестеров

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Рогова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

канд. пед. наук, доцент А.В. Егорова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

В выпускной квалификационной работе рассматривается разработка информационной системы учета объектов и предоставляемых услуг. Сервис представляет собой программное решение, разработанное для упрощения и автоматизации процесса учета и проверки данных о потреблении коммунальных ресурсов и подключенных услугах.

Цель данного сервиса состоит в обеспечении удобного и безопасного механизма передачи показаний приборов учета непосредственно ответственному сотруднику компании, а также автоматизации процесса подсчёта этих данных для обеспечения точности и достоверности информации о потреблении коммунальных услуг.

Во введении данной работы обозначены цели и задачи, определена актуальность исследования, а также выделены объект и предмет исследования.

В первой главе анализируется и проводится общая характеристика предметной области «ЖКХ» и организации «Квартплата 24». Моделируется рассматриваемый бизнес-процесс.

Во второй главе представлено логическое проектирование информационной системы с помощью различных диаграмм и их описания.

В третьей главе была продемонстрирована реализация приложения по учету объектов и предоставляемых услуг, так же была проведена спецификация выбранных технологий реализации, обоснован выбор библиотек и наборов инструментов, проведено тестирование.

Abstract

The title of the graduation work is Development of an Information System for Managing Properties and Provided Services.

The graduation work consists of an explanatory note on 46 pages, introduction, including 17 figures, 4 tables, and the list of 20 references including 5 foreign sources.

The object of the graduation work is the activity of the "Housing and Utilities" sector and the company "Kvartplata 24".

The subject of the graduation work is an application that automates managing properties and provided services in the field of housing and utilities.

The key issue of the thesis is the development of a functional application for a company in order to automate business processes and increase effectiveness of the employees.

The graduation work may be divided into several connected parts which are: analysis of existing business processes within the framework of the «AS IS» model, analysis and modeling of business processes within the framework of the «TO BE» model, analysis of competitors and setting the task for the development of the application, development of requirements for the application, the definition of development tools and the formation of a logical and physical model of the application.

Finally, in the practical part we consider the development and conduct testing of the application for managing properties and the services provided. We conduct testing to check if the application works as intended.

In conclusion we'd like to stress this work is relevant in solving the problem of the business process automation for the company «Kvartplata 24» in the managing properties and the services provided, as well as business process template of model «TO BE» can be applied to the entire field.

Оглавление

Введение.....	6
Глава 1 Функциональное моделирование информационной системы учета объектов и представленных услуг.....	8
1.2 Концептуальное моделирование информационной системы учета объектов и представленных услуг.....	11
1.3 Анализ существующих разработок.....	13
1.4 Разработка и анализ модели бизнес-процесса «как должно быть» .	15
1.5 Постановка задачи на разработку информационной системы учета объектов и предоставляемых услуг.....	16
Глава 2 Логическое проектирование информационной системы учета объекта и предоставляемых услуг.....	19
2.1 Выбор технологии логического моделирования.....	19
2.2 Разработка логических моделей информационной системы учета объекта и предоставляемых услуг.....	19
2.3 Разработка логической и физической модели данных.....	22
Глава 3 Физическое проектирование информационной системы учета объектов и предоставляемых услуг.....	27
3.1 Выбор технологий разработки информационной системы.....	27
3.2 Спецификация выбранных технологий.....	30
3.3 Инициализация главного графического окна.....	31
3.4 Инициализация функций управления данными.....	34
3.5 Контрольный пример реализации информационной системы учета объекта и предоставляемых услуг.....	36

3.6 Требования к аппаратно-программному обеспечению приложения	41
3.7 Тестирование информационной системы.....	42
Заключение	44
Список используемой литературы и используемых источников.....	45

Введение

В современном обществе одним из ключевых аспектов социально-экономических изменений является модернизация и улучшение жилищно-коммунальной сферы, которая создаёт все необходимые условия для обеспечения достойного проживания людей. Основные составляющие этой сферы – управление жилищным фондом и предоставление коммунальных услуг, которые играют важную роль в обеспечении жильем и коммунальными услугами конечных пользователей.

Развитие информационных технологий в секторе жилищно-коммунального обслуживания приобретает первостепенное значение в рамках государственной политики, как в Российской Федерации, так и во всём мире. Процессы реформирования в сфере коммунального хозяйства привели к изменениям в структуре собственности муниципального жилищного фонда. Новая стратегия формирует новый класс собственников недвижимости, что порождает проблемы в управлении и обслуживании.

Таким образом, тема выпускной квалификационной работы представляет актуальную и перспективную область разработки, требующую внимания к практическим аспектам учета и контроля платежей за жилищно-коммунальные услуги.

Объект исследования данной работы является процесс учета объектов и предоставляемых услуг в сфере ЖКХ.

Предметом исследования является автоматизация процесса учета объектов и предоставляемых услуг в сфере ЖКХ.

Целью выпускной квалификационной работы является разработка информационной системы учёта объектов и предоставляемых услуг, которая позволит автоматизировать процессы, выполняемые вручную, и эффективно вести учёт объектов и услуг в сфере ЖКХ.

Для достижения поставленной цели требуется решить следующие задачи:

- смоделировать и провести анализ бизнес-процесса;
- описать обоснование необходимости автоматизированного варианта информационной системы и сформировать требования к новой технологии;
- провести анализ существующих разработок на предмет соответствия сформулированным требованиям;
- разработать логическую модель информационной системы и описать её;
- разработать информационную систему с учетом сформированных требований.

Выпускная квалификационная работа состоит из 46 страниц и содержит 17 рисунков, 4 таблицы, 20 источников.

Глава 1 Функциональное моделирование информационной системы учета объектов и представленных услуг

1.1 Технико-экономическая характеристика организации

Жилищно-коммунальное хозяйство (ЖКХ) – это сектор экономики, который заботится о поддержании функциональности и работоспособности зданий, чтобы люди могли проживать и находиться в них в безопасности и комфорте [4].

«Квартплата 24» — ИТ-компания, эффективно решающая задачи автоматизации процессов расчета и учета платы за жилищно-коммунальные услуги, приема и распределения платежей, востребования долгов в полном соответствии с законодательством для товариществ собственников жилья, управляющих и ресурсоснабжающих организаций, информационно-расчетных центров на всей территории РФ.

Организационная структура компании «Квартплата 24» приведена на рисунке 1.

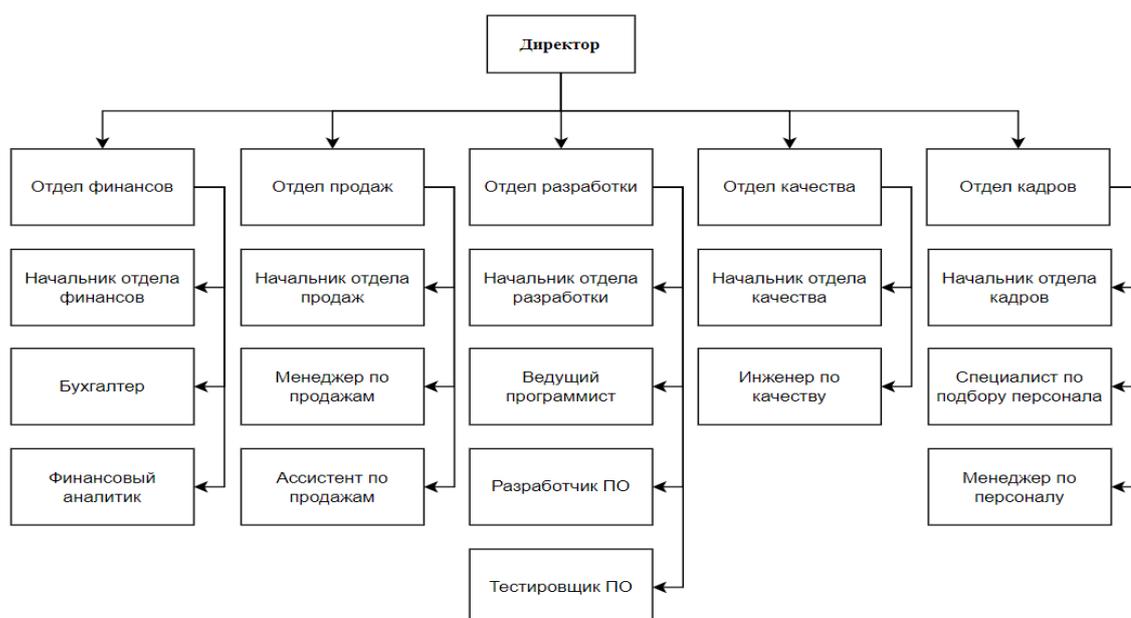


Рисунок 1 – Организационная структура компании «Квартплата 24»

Организационная структура компании «Квартплата 24» подразумевает, что директору подчиняются начальники отделов. В свою очередь начальникам отделов подчиняются сотрудники соответствующих отделов.

«Квартплата 24» предоставляет сервисы по расчету и приему платежей ЖКУ. Экосистема сервисов «Квартплата 24» представляет собой более 10 тесно интегрированных между собой облачных сервисов.

На рисунке 2 представлена диаграмма сервисов организации «Квартплата 24».

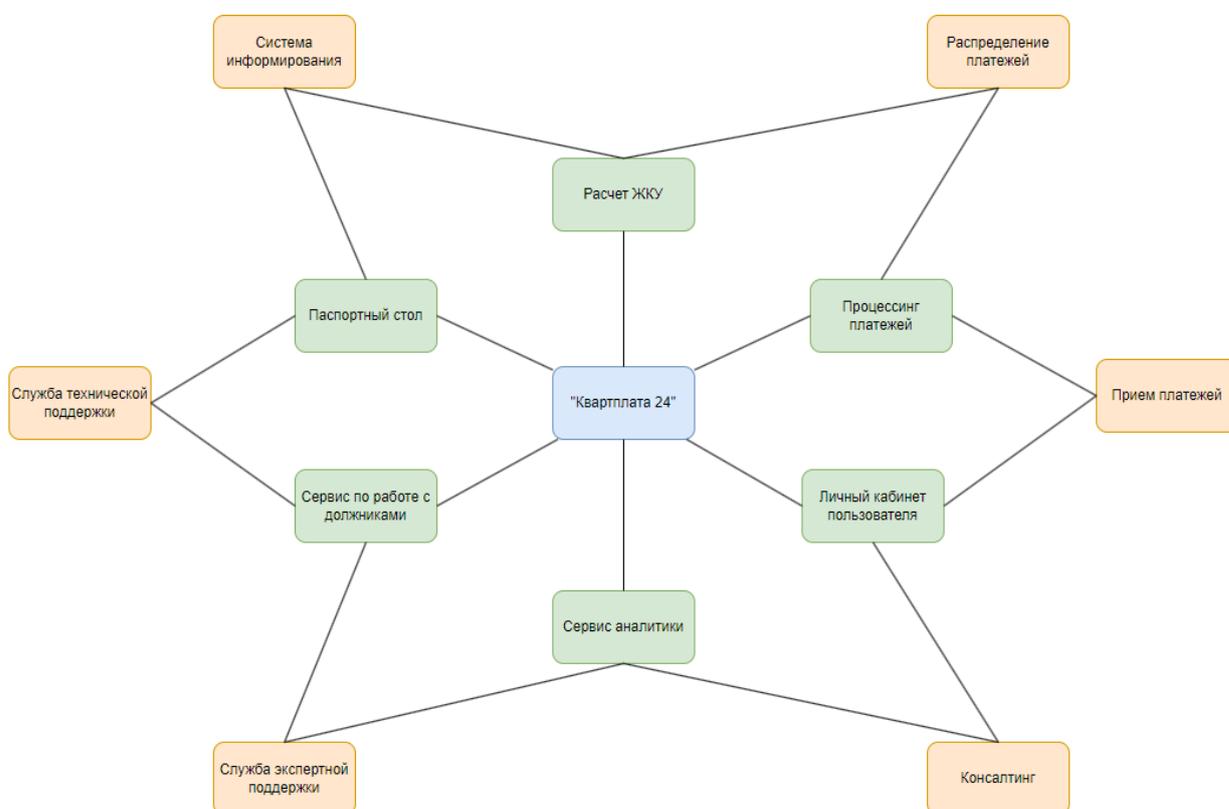


Рисунок 2 – Диаграмма сервисов «Квартплата 24»

Инновационные технологии. В основе работы сервиса лежат высокотехнологичные IT-решения, которые позволяют оптимизировать процессы учета и расчетов между потребителями и поставщиками услуг. Это

включает в себя использование машинного обучения для анализа платежных потоков и управления дебиторской задолженностью.

Пользовательский интерфейс и мобильное приложение. «Квартплата 24» предлагает удобный пользовательский интерфейс и мобильное приложение, которые делают процесс оплаты ЖКУ простым и доступным с любого устройства. Пользователи могут управлять своими платежами, просматривать историю транзакций и получать уведомления о счетах и платежах.

Консультационные и поддерживающие услуги. Сервис предоставляет круглосуточную поддержку клиентов, что включает телефонные звонки, чаты в приложении и электронные запросы. Команда поддержки помогает решать вопросы, связанные с платежами и обслуживанием счетов, а также консультирует по вопросам экономии на коммунальных услугах.

Широкая сеть партнеров. «Квартплата 24» сотрудничает с широким кругом финансовых учреждений, интеграторами систем безопасности и поставщиками коммунальных услуг, чтобы предложить потребителям лучшие условия и сервис [13].

Экологическая инициатива. Сервис активно внедряет экологические инициативы, направленные на снижение использования бумаги за счет электронных платежей и выписок, что способствует защите окружающей среды.

Безопасность и конфиденциальность. «Квартплата 24» придает особое значение защите данных клиентов. Все платежные транзакции и пользовательские данные защищены современными методами шифрования и соответствуют международным стандартам безопасности.

1.2 Концептуальное моделирование информационной системы учета объектов и представленных услуг

Модель «как есть» представляет собой состояние бизнес-процесса, не имеющего информационной системы для упрощения передачи показаний приборов учета. Для моделирования бизнес-процесса было принято решение использовать нотацию BPMN.

Нотация BPMN (Business Process Model and Notation) — это стандарт для графического представления бизнес-процессов в виде бизнес-диаграмм. Она разработана для обеспечения понимания бизнес-процессов всеми заинтересованными сторонами, включая бизнес-аналитиков, технических разработчиков и менеджеров. BPMN предоставляет единый графический язык, что позволяет описывать процессы в деталях, обеспечивая легкость интерпретации и реализации.

BPMN поддерживает создание различных типов диаграмм, в том числе диаграмм процессов, диаграмм сотрудничества и диаграмм диалогов, каждая из которых обеспечивает различные перспективы на процессы и их взаимодействия. Это делает BPMN мощным инструментом для документирования, анализа и оптимизации бизнес-процессов [7].

Преимущества использования BPMN заключаются в стандартизации представления бизнес-процессов, улучшении коммуникации между различными сторонами, участвующими в проекте, и повышении эффективности управления процессами. Эта нотация широко используется в бизнес-анализе и управлении проектами и признана многими организациями как стандарт для описания бизнес-процессов. Её главной задачей является возможность смоделировать бизнес-процесс так, чтобы он был прост в понимании для аналитиков или разработчиков [11].

Диаграмма бизнес-процесса обработки заявок изображена на рисунке 3.

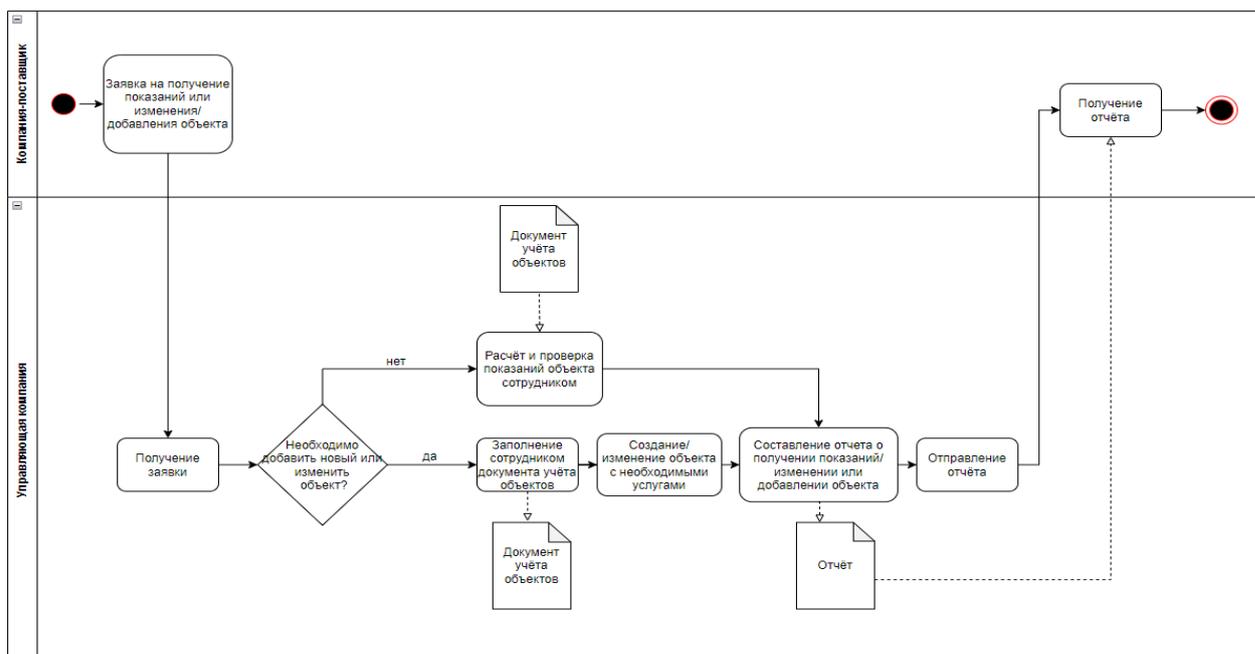


Рисунок 3 – Диаграмма бизнес-процесса учета объектов и предоставляемых услуг «как есть»

Процесс передачи показаний начинается с заявки сотрудника ресурсоснабжающей компании на передачу показаний по конкретному объекту.

Далее сотрудник управляющей компании вручную проверяет достоверность и точность показаний, рассчитывает стоимость оказанных услуг, составляет и отправляет отчёт компании-поставщику.

Заключением этих процессов является получение сотрудником компании-поставщика отчета с расчётами оказанных услуг, и их стоимостью, либо отчета о постановке на учет нового объекта.

Из диаграммы видно, что сотруднику приходится вручную заполнять документ учёта объектов, составлять отчет, что на дистанции сильно замедляет процесс обработки заявок.

В данной модели процесс сбора и передачи данных выполняется вручную или с применением устаревших методов и инструментов. Отсутствие автоматизации влечет за собой ряд проблем и ограничений:

- возникают ошибки в данных из-за ручного ввода информации, подверженного человеческим ошибкам, и невозможности быстрой проверки. Это может привести к неточным данным и недостоверным отчетам о потреблении ресурсов;
- процесс замедлен из-за того, что ручной сбор и передача данных, или их изменение, требуют дополнительного времени и усилий для обработки информации;
- аналитика ограничена отсутствием централизованной системы сбора и анализа данных, что затрудняет анализ потребления ресурсов и выявление тенденций.

Таким образом, данная модель бизнес-процесса несовершенна из-за отсутствия автоматизации процессов передачи, добавления новых объектов или услуг, и обработки данных от счетчиков, что замедляет процесс обработки информации. Этот недостаток мы решаем с помощью внедрения новой информационной системы.

1.3 Анализ существующих разработок

В современном мире есть несколько существующих решений для реализации информационных систем в сфере ЖКХ [4]. Проведем их анализ и выявим соответствие сформулированным требованиям.

Alphaletz – является британским сервисом для учета объектов недвижимости и управления бизнесом. Сервис позволяет пользователям организовать учёт своих объектов недвижимости и связанных с ними финансовых операций. Минусы этого сервиса – невозможность на данный момент его использования в Российской Федерации, высокая цена.

Rent-manager – это программное обеспечение для управления недвижимостью, предназначенное для помощи владельцам, менеджерам и арендодателям в управлении их портфелями недвижимости. Из недостатков

данного программного обеспечения – его так же нельзя использовать на территории Российской Федерации, а интерфейс этого приложения требует длительного изучения и опыта работы в нём, для достижения эффективного использования приложения.

Buildium – облачное программное обеспечение для ведения учета объектов недвижимости, управления маркетингом, публикацией объявлений об аренде и т.д. Этот сервис тоже не доступен в России, а его функционал хоть и является очень обширным, но не решает основную поставленную задачу.

Intrum CRM - платформа, которая объединяет в себе передовые IT разработки в сфере недвижимости, а также программу обучения руководителей и рядовых агентов. Позволяет вести учет объектов недвижимости. Сервис доступен в России, но является очень большим по своему масштабу, а также требует относительно длительного обучения.

Сравнительный анализ рассмотренных программных продуктов проведен ниже в таблице 1.

Таблица 1 - Сравнительный анализ рассмотренных программных продуктов

Функция	Alphaletz	Rent-manager	Buildium	Intrum CRM
Управление арендаторами	+	+	+	+
Управление собственностью	+	+	+	+
Бухгалтерский учет	+	+	+	+
Обслуживание	+	+	+	+
Отчетность	+	+	+	+
Интеграции	Ограничено	Средне	Широко	Широко
Цена	Средняя	Высокая	Высокая	Высокая
Поддержка клиентов	Средняя	Хорошая	Отличная	Отличная

В результате анализа аналоговых программ, стало ясно, что большинство самых популярных сервисов по учёту объектов недвижимости являются зарубежными. Сейчас их невозможно, либо же очень сложно использовать на территории Российской Федерации. Так же, большинство

таких сервисов и приложений являются обширными, и мощными инструментами для решения множества задач, в том случае, если нам необходимо лишь учитывать показатели счётчиков объекта недвижимости, такие приложения будут слишком большими и не эффективными, с точки зрения выполняемой ими работы и финансовых затрат на них. Кроме того, обширный функционал таких приложений, может усложнить их интеграцию в другие системы.

Разработка собственного приложения даст возможность легко интегрировать его с другими информационными системами и составит меньше финансовых затрат.

1.4 Разработка и анализ модели бизнес-процесса «как должно быть»

Для автоматизации учета объектов и предоставляемых услуг в сфере ЖКХ была внедрена информационная система, которая обеспечивает более эффективную работу в организации. На рисунке 4 отображена модель «как должно быть» бизнес-процесса учета объектов и предоставляемых услуг.

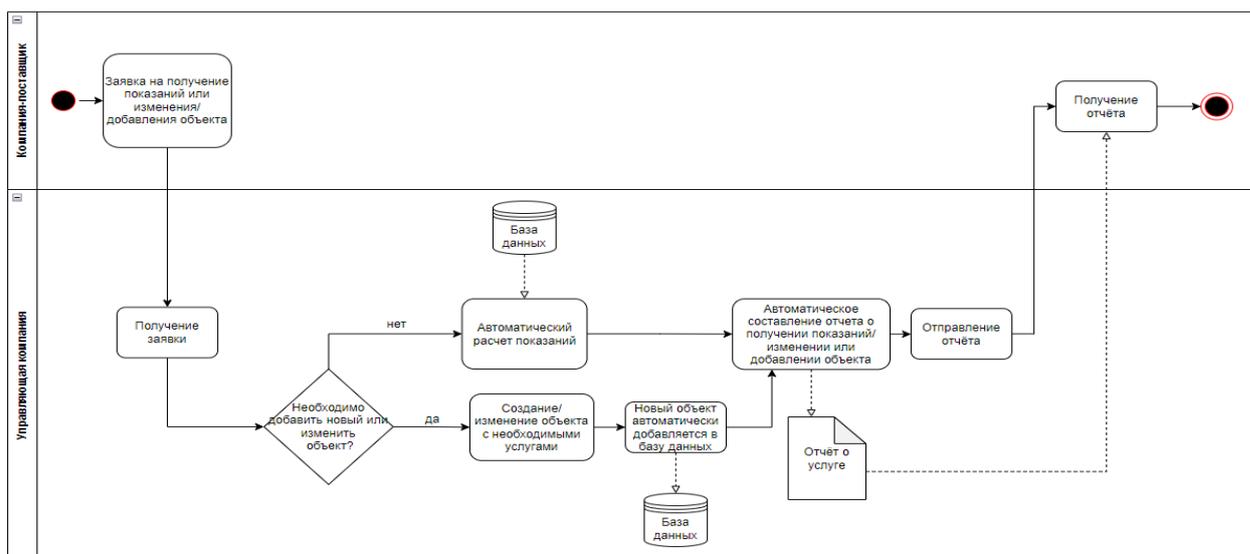


Рисунок 4 – Диаграмма бизнес-процесса учета объектов и предоставляемых услуг «как должно быть»

В данной модели процесс передачи показаний начинается с заявки сотрудника ресурсоснабжающей компании, но теперь вся работа, в которой сотрудник управляющей компании мог допустить ошибки в расчётах, передана на обработку интегрируемому приложению. Добавлена база данных, для более структурированного хранения данных, а также доступа к данным, с любого компьютера, который имеет доступ к базе данных. Приложение само рассчитывает показания объектов, стоимость оказываемых для них услуг, а затем выводит их ответственному сотруднику. В случае добавления на учёт нового объекта, сотруднику необходимо будет внести данные для этого объекта, и затем они автоматически добавятся в общую базу данных объектов.

Сотруднику управляющей компании в данном случае достаточно лишь отправить отчёт для компании-поставщика, на основе данных, которые ему предоставит информационная система.

В усовершенствованной модели, сотрудник ресурсоснабжающей компании инициирует заявку на получение показаний, либо изменение объекта, но все расчёты теперь выполняет интегрированное приложение. Оно автоматически обрабатывает данные, рассчитывает стоимость услуг и предоставляет итоги ответственному сотруднику в виде автоматически создаваемого отчёта. При добавлении нового объекта сотрудник вносит начальные данные, которые автоматически интегрируются в базу данных. Задача сотрудника управляющей компании сводится к отправке отчёта поставщику.

1.5 Постановка задачи на разработку информационной системы учета объектов и предоставляемых услуг

Развитие технологий в области сбора и анализа данных позволит оптимизировать учет потребления ресурсов и повысить эффективность использования энергии. Для этого было решено разработать приложение,

которое позволит сотрудникам управляющих компаний легко отслеживать данные показаний счетчиков.

Для учёта объектов недвижимости, а также предоставляемых им услуг, необходимо разработать специализированное, узконаправленное приложение, которое будет легко интегрироваться в другие внешние сервисы, обладать интуитивно понятным интерфейсом, и обеспечивать надёжность вычислений.

Задачи, поставленные на разработку информационной системы:

- разработать пользовательский интерфейс для просмотра показаний счётчиков по объекту;
- создать базу данных для хранения информации по объектам и предоставляемым услугам;
- реализовать возможность добавления новых объектов и новых услуг;
- реализовать возможность привязки нескольких услуг к объекту недвижимости;
- реализовать возможность создания отчетов по объектам и предоставляемым услугам.

Требования к разрабатываемой системе:

- круглосуточный режим функционирования;
- возможность расширения системы;
- интерфейс приложения должен быть понятным и удобным;
- обеспечение точности расчётов данных;
- показания должны учитывать услуги электроэнергии, горячей воды, холодной воды, газоснабжения, отопления;
- на главной странице должен быть представлен список всех объектов, поставленных на учёт;
- по каждому объекту необходимо выводить информацию о каждой подключённой услуге, их индивидуальную и суммарную стоимость;
- в каждой услуге должна быть формула, по которой производится расчёт финансовой стоимости услуги;

- формула должна быть редактируема, чтобы при изменении тарифов не приходилось создавать новую услугу.

Поставленные задачи и требования помогут эффективно решить имеющиеся проблемы в бизнес-процессе путем разработки и внедрения ИС.

Выводы по главе 1

В первой главе была представлена структура компании. Создана модель бизнес-процесса учета объектов и предоставляемых услуг для автоматизации решения задач. Затем была разработана модель «как есть» и проведен анализ этой модели для определения требований к информационной системе. Также создана модель «как должно быть» текущего бизнес-процесса и поставлены задачи на разработку информационной системы.

Глава 2 Логическое проектирование информационной системы учета объекта и предоставляемых услуг

2.1 Выбор технологии логического моделирования

Под логическим моделированием информационной системы подразумевается описание решения со стороны разработчика, создания её структуры, синтаксиса и взаимодействия модулей в системе. В основном для логического моделирования используется нотация UML, которая позволяет графически описать систему на стадиях проектирования и разработки [1].

Нотация UML является стандартом в области моделирования информационных систем и широко используется в различных областях, включая разработку ПО, проектирование БД, а также моделирование бизнес-процессов [9]. Данная нотация предоставляет различные наглядные диаграммы для логического моделирования.

Таким образом, выбор нотации UML для логического моделирования обеспечивает эффективное отображение моделей, с целью облегчения дальнейшей разработки информационной системы.

Далее разрабатываются диаграммы логической модели посредством методологии на стандартах нотациях языка UML.

2.2 Разработка логических моделей информационной системы учета объекта и предоставляемых услуг

2.2.1 Диаграмма вариантов использования

Диаграммы вариантов использования отражает функционал информационной системы в виде отношения актёров и прецедентов. Актёр – человек, который взаимодействует с системой. Прецедент – действие, которое выполняет актёр в системе. Диаграмма вариантов использования помогает

лучше понять, как система будет использоваться в реальном мире и какие функциональные возможности она должна предоставлять.

На рисунке 5 представлена диаграмма вариантов использования, описывающая функционал информационной системы.

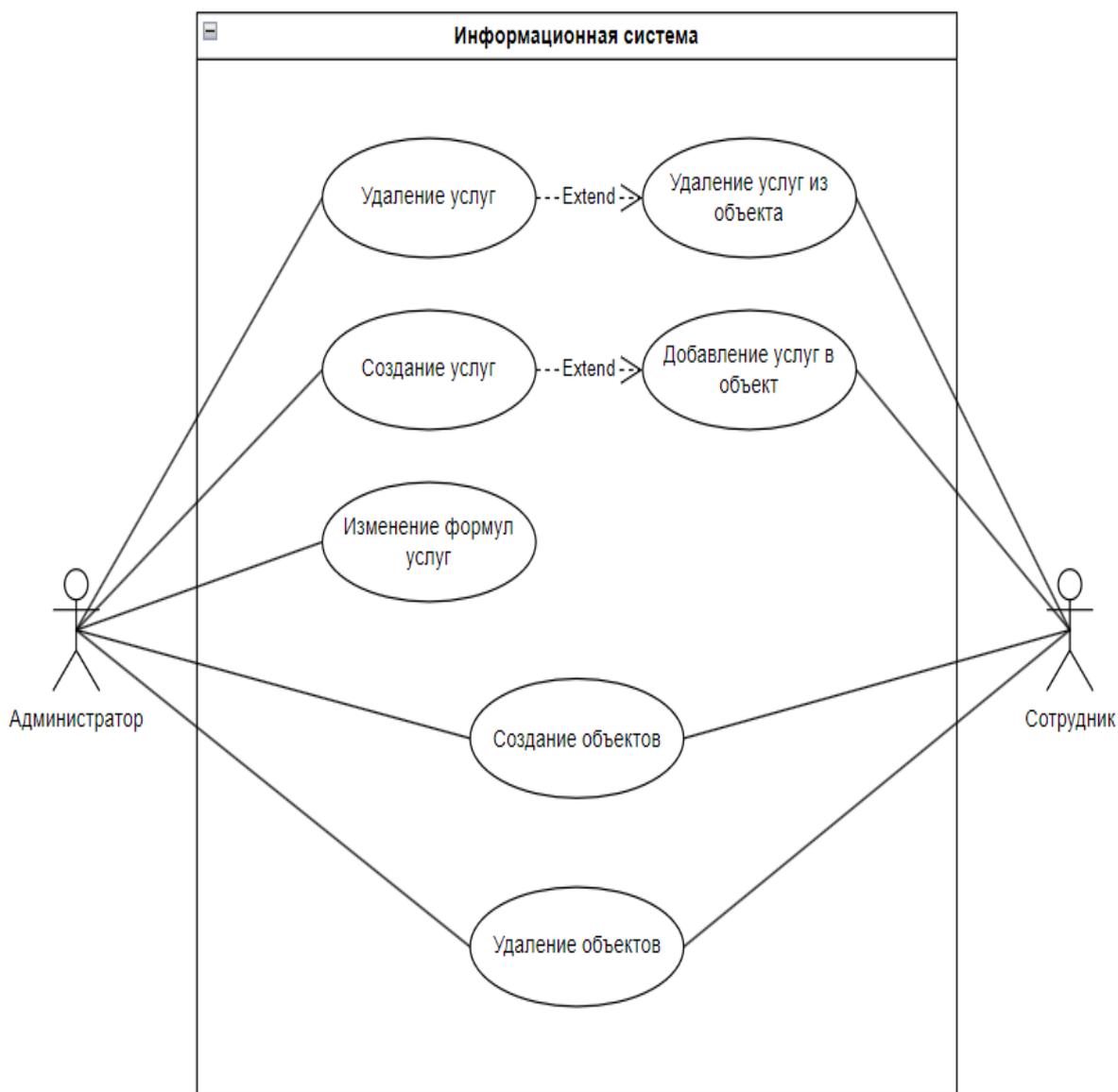


Рисунок 5 – Диаграмма вариантов использования

На представленной диаграмме вариантов использования представлены два актёра — «Администратор» и «Сотрудник» — взаимодействуют с системой для управления услугами и объектами недвижимости.

Администратор обладает полномочиями для выполнения разнообразных операций, таких как создание и удаление услуг, изменение формул расчета услуг, а также добавление и удаление объектов недвижимости. Сотрудник занимается управлением услугами в рамках объектов недвижимости, выполняя такие задачи, как добавление и удаление услуг и объектов. Основные операции для администратора включают удаление и создание услуг, изменение формул услуг. Для сотрудника к ключевым задачам относятся удаление услуг из объекта, добавление услуг в объект, а также добавление и удаление объектов недвижимости. Эта диаграмма иллюстрирует функции, выполняемые каждым из актёров в процессе управления услугами и объектами, подчеркивая их роли и обязанности в системе, что помогает понять, какие задачи являются частью их повседневных обязанностей.

2.2.2 Диаграмма классов

Диаграмма классов в UML — это один из видов структурных диаграмм, который используется для моделирования статической структуры системы. Она представляет классы, их атрибуты, методы и взаимосвязи между классами. Диаграммы классов используются во время разработки программного обеспечения для визуализации архитектуры системы, что помогает разработчикам и аналитикам лучше понимать и организовывать структуру приложения или системы. Диаграмма классов представлена на рисунке 6.

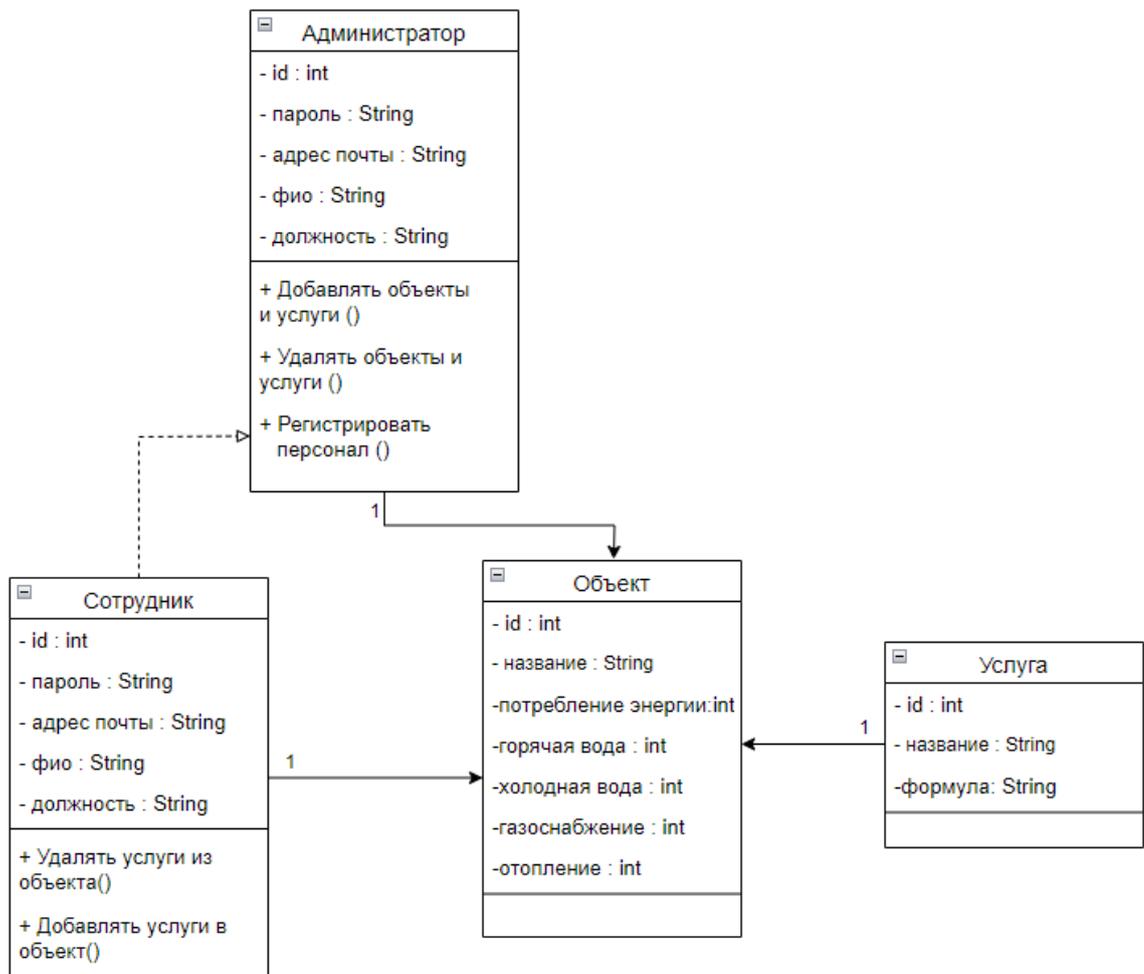


Рисунок 6 – Диаграмма классов ИС

Эта диаграмма описывает структуру информационной системы. Такой подход помогает систематизировать процесс реализации ИС.

2.3 Разработка логической и физической модели данных

Логические модели демонстрируют процесс выделения группы объектов с их атрибутами.

Для того чтобы все компоненты системы функционировали правильно, необходима их четкая структуризация. В системе должна присутствовать хорошо продуманная организационная логика для оптимизации данных в базе.

Для проектирования логической модели базы данных необходимо было выбрать технологию логического моделирования. В процессе логического моделирования рекомендуется применять методы и технологии объектно-ориентированного анализа и проектирования, которые основываются на стандартных нотациях UML. Но методология IDEF1X, в свою очередь, позволяет создавать семантические модели данных, способствующие управлению данными как ресурсом, интеграции информационных систем и разработке баз данных. Этот стандарт входит в группу методологий IDEF и также применяется для моделирования реляционных баз данных [12].

В таблице 2 представлен сравнительный анализ методологий UML и IDEF1X.

Таблица 2 – Сравнительный анализ методологий UML и IDEF1X

Критерий	UML	IDEF1X
Назначение	Общее проектирование систем, включая программное обеспечение и бизнес-процессы.	Специализированное моделирование реляционных баз данных.
Особенности	Включает разнообразные типы диаграмм для разных аспектов системы.	Фокус на семантическом моделировании данных и нормализации баз данных.
Преимущества	Гибкость, многогранное применение, поддержка множества инструментов.	Оптимизация и четкое управление структурами данных в базах данных.
Применение	Широкий спектр инженерных задач в разных областях.	Особенно полезен для проектирования, анализа и интеграции баз данных.
Визуализация	Множество диаграмм для структурного и поведенческого моделирования.	Ограниченное количество диаграмм, в основном фокусируется на отношениях между данными.

На основании информации, полученной с помощью таблицы 2, можно сделать вывод, что методология IDEF1X в данном случае подойдет лучше, чем UML. Это обусловлено тем, что необходимо спроектировать логическую модель данных, и IDEF1X специализируется конкретно на базах данных [16].

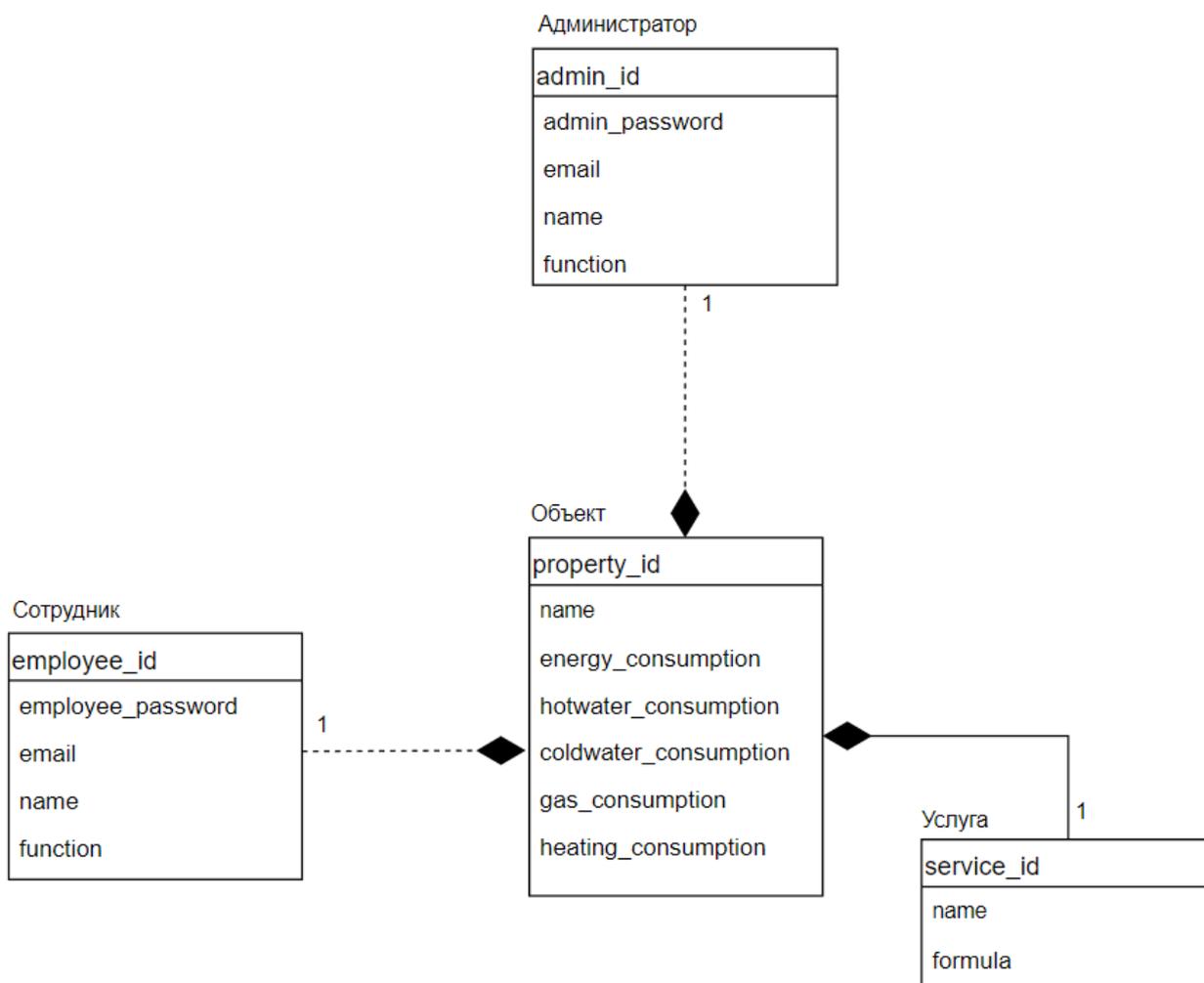


Рисунок 7 –Логическая модель данных

Диаграмма логической модели данных отображает структуру базы данных без привязки к физическим аспектам хранения данных. Логическая модель данных фокусируется на том, как данные взаимосвязаны и организованы в пределах предметной области.

Физическая модель данных представляет собой спецификацию данных для конкретной системы управления базами данных (СУБД). В процессе физического моделирования необходимо создать все требуемые таблицы и столбцы, а также установить первичные ключи для СУБД.

Модель физического представления базы данных является ключевым элементом для корректной разработки и реализации информационной системы с базой данных.

Диаграмма физической модели данных представлена на рисунке 8.

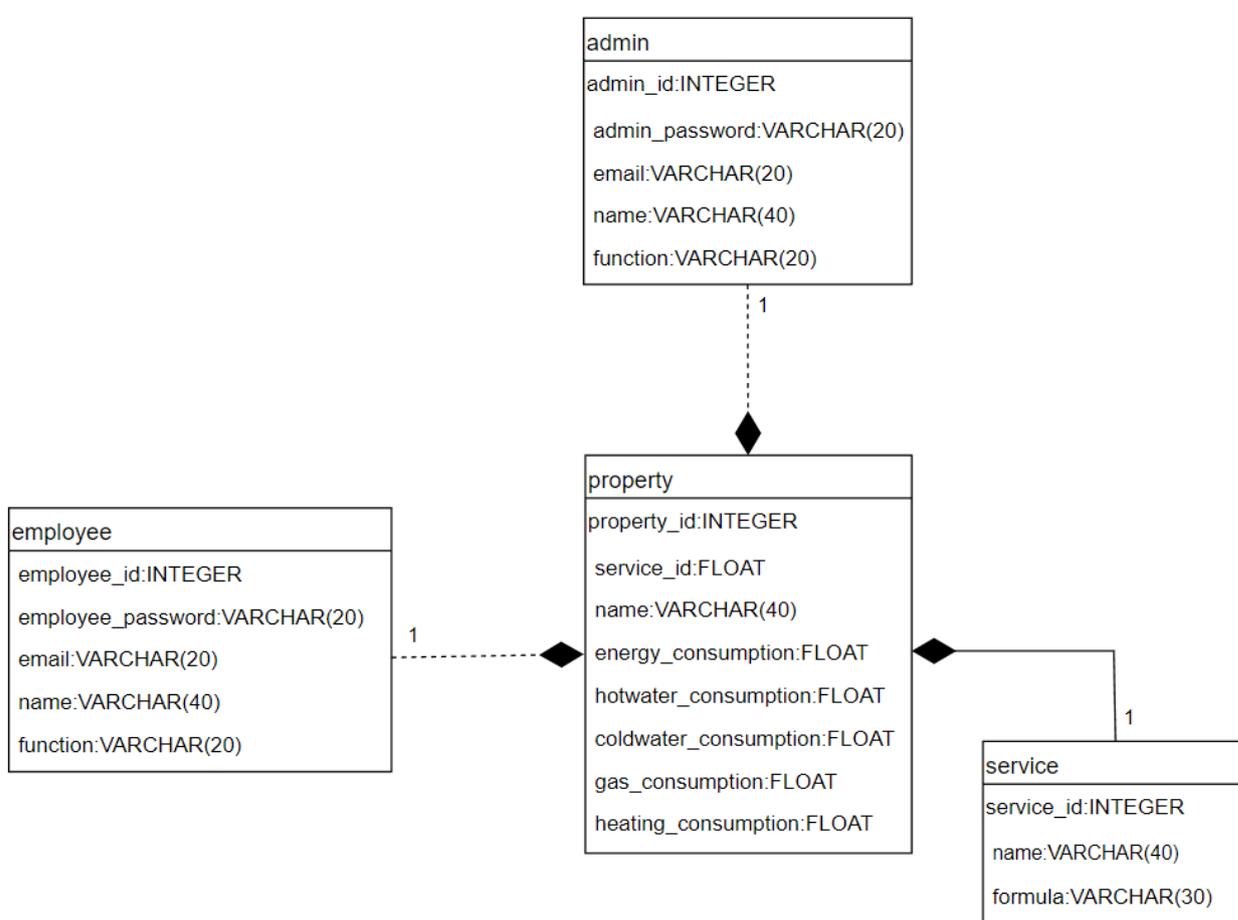


Рисунок 8 – Диаграмма физической модели базы данных

Это физическая модель данных, которая содержит все необходимые для СУБД таблицы и столбцы, типы передаваемых данных, первичные и внешние ключи, основаны на логических данных.

Основной сущностью является объект недвижимости (property).

Существует отдельная сущность услуги (service), которая может быть связана с объектом. Каждая услуга имеет уникальный идентификатор, название и формулу.

Сущность администратор (admin) содержит информацию об администраторах, которые могут управлять объектами недвижимости. Каждый администратор имеет уникальный идентификатор, пароль, адрес электронной почты, имя и функцию.

Сущность сотрудник (employee) включает информацию о сотрудниках, которые также могут управлять объектами недвижимости. Каждый сотрудник имеет уникальный идентификатор, пароль, адрес электронной почты, имя и функцию.

Таким образом, физическая модель базы данных позволяет учитывать и управлять различными аспектами управления недвижимостью, включая потребление ресурсов и предоставление услуг, а также распределять эти задачи между администраторами и сотрудниками.

Выводы по главе 2

Во второй главе было проработано логическое проектирование информационной системы учета объектов и предоставляемых услуг. Представлена логическая модель информационной системы в виде диаграмм в нотации UML, а именно диаграммы вариантов использования и классов, а также логическая и физическая модель данных. Все разработанные диаграммы в данной главе являются основой логики в разрабатываемой ИС [5].

Глава 3 Физическое проектирование информационной системы учета объектов и предоставляемых услуг

3.1 Выбор технологий разработки информационной системы

Разработка приложений, в том числе с использованием баз данных, оказала огромное влияние на ИТ-индустрию. Рынок систем управления базами данных (СУБД) значительно вырос благодаря увеличению количества корпоративных данных и необходимости их анализа, а также благодаря повышению требований к безопасности и скорости обработки информации. Современные условия и возможности для разработки и эксплуатации баз данных значительно улучшились.

В разработке было решено использовать язык Python.

Сейчас Python широко используется во всех информационных сферах. Простота в изучении и использовании, разнообразие библиотек и фреймворков, делают его популярным выбором для решения задач в самых различных областях, в том числе и в области жилищно-коммунальных услуг.

Python может работать с другими языками программирования, что значительно упрощает интеграцию приложения в любую информационную систему. Он может работать с такими популярными инструментами и языками, как: C++, Java, MATLAB и другими [14]. Это позволяет решать самые разнообразные задачи.

Для выбора наиболее подходящей СУБД, необходимо провести анализ наиболее популярных СУБД. Сравнение приведено в таблице 3.

Таблица 3 – Сравнение популярных СУБД

СУБД	Преимущества	Недостатки
PostgreSQL	Предопределённые функции; Ряд интерфейсов; Масштабируемость.	Конфигурация может сбивать с толку.
Oracle	На основе конструкции программирования PL / SQL; База данных Oracle безопасна и гарантирует, что пользовательские данные не будут изменены путем оперативных обновлений	Высокая стоимость для небольших организаций; Требуются значительные ресурсы для установки.
Microsoft SQL Server	Отдельные привилегии безопасности; Поддерживает резервный сервер; Поддерживает язык SQL.	Проблема служб Integration Services для импорта файлов; Идеально подходит только для крупных организаций.
MySQL	Поддерживает большие базы данных, до 50 миллионов строк и более в таблице; Реализация различных пользовательских интерфейсов.	Нет встроенной поддержки XML или OLAP.

Для реализации базы данных была выбрана объектно-реляционная система управления базами данных – PostgreSQL [3]. Она подходит по всем нашим требованиям, основным из которых была возможность

масштабирования системы. Основными преимуществами PostgreSQL являются:

- открытый исходный код: PostgreSQL распространяется под лицензией открытого исходного кода, что означает, что он свободно доступен для использования, модификации и распространения без необходимости платить лицензионные сборы.
- мощные функциональные возможности: PostgreSQL поддерживает широкий спектр возможностей, включая сложные типы данных, триггеры, хранимые процедуры, функции, оконные функции, полнотекстовый поиск, географические запросы и многое другое [10]. Это делает его привлекательным выбором для различных типов приложений и использования.
- надежность и устойчивость: PostgreSQL известен своей надежностью и устойчивостью. Он обеспечивает ACID-совместимость (атомарность, согласованность, изолированность, долговечность), что гарантирует надежное хранение данных и поддержку транзакций [6].
- масштабируемость: PostgreSQL способен обрабатывать как маленькие базы данных для небольших проектов, так и крупные базы данных для крупных предприятий. Он поддерживает различные методы масштабирования, включая репликацию, разделение данных и партиционирование [8].
- поддержка расширений и экосистемы: PostgreSQL имеет активное сообщество разработчиков и поддерживается большим количеством расширений и дополнений, которые позволяют расширять его функциональность и интегрировать с другими технологиями и инструментами.
- кроссплатформенность: PostgreSQL поддерживается на различных операционных системах, включая Linux, Windows, macOS и другие,

что делает его гибким выбором для разработчиков и администраторов баз данных [17,18].

3.2 Спецификация выбранных технологий

Для лучшего понимания используемых технологий необходимо разобрать важные аспекты, описывающие принципы работы каждого компонента. В нашем случае основными технологиями являются библиотека Qt6 для работы с графическим интерфейсом, библиотека Psycopg2 для взаимодействия с PostgreSQL и PyQt6 как связующее звено между Python и Qt.

Qt6 – это мощная и широко используемая кроссплатформенная библиотека для разработки GUI-приложений. Qt6 позволяет создавать высокопроизводительные приложения с современным графическим интерфейсом, которые могут работать на различных операционных системах, таких как Windows, Linux и macOS. Особенностью Qt6 является использование C++ и QML для описания логики и интерфейса приложений, что обеспечивает высокую скорость выполнения и гибкость в дизайне [20].

PyQt6 – это набор инструментов, который связывает Qt6 с языком Python, позволяя разработчикам использовать все преимущества Python и мощь Qt6[18]. PyQt6 предоставляет возможность создания комплексных пользовательских интерфейсов, интеграции с сетевыми функциями и базами данных, а также обработки событий и выполнения многозадачности.

Psycopg2 – это адаптер PostgreSQL для языка программирования Python. Этот модуль позволяет выполнять операции с базами данных PostgreSQL, используя простой и понятный Python-синтаксис. Psycopg2 поддерживает большинство функций PostgreSQL, включая транзакции, вызовы процедур, буферизацию запросов и обработку исключений, что делает его незаменимым инструментом для работы с данными [19].

Работа приложения, использующего эти технологии, осуществляется через взаимодействие между Python-скриптами, пользовательским

интерфейсом на PyQt6 и обработкой данных в PostgreSQL через Psycopg2. Эта архитектура позволяет эффективно разделять логику приложения и интерфейс, а также обеспечивать надёжное управление данными.

Пример использования данных технологий.

Интерфейс пользователя: создается с использованием PyQt6, который взаимодействует с Qt6 для отображения элементов GUI.

Обработка данных: выполняется через Psycopg2, который обеспечивает связь между Python и PostgreSQL, позволяя выполнить CRUD-операции.

Логика приложения: реализуется на Python, что делает программу гибкой и легко адаптируемой под различные требования [15].

В заключение, использование Qt6, PyQt6, и Psycopg2 позволяет создать мощное и гибкое приложение для учета объектов недвижимости и предоставления услуг, сочетая простоту Python и производительность Qt.

3.3 Инициализация главного графического окна

Начать разбор основных блоков кода было решено с функции, которая инициализирует главное графическое окно. Здесь определён класс `MainWindow`, который наследуется от класса `QMainWindow`. Это основное окно приложения с графическим интерфейсом пользователя [10].

Разберём его подробнее.

Конструктор (`__init__`). Конструктор класса начинается с вызова конструктора родительского класса `QMainWindow` для корректной инициализации окна. Далее устанавливаются размеры и позиция окна с помощью метода `setGeometry`, который обеспечивает начальное размещение окна на экране. Создается `QWidget`, который служит центральным виджетом окна. Этот виджет будет контейнером для всех других виджетов и слоёв, управляющих содержимым окна. В центральный виджет устанавливается вертикальный слой (`QVBoxLayout`), который позволяет удобно организовывать вложенные виджеты в виде столбца.

Панель выбора с кнопками. Для управления видом данных (свойства или услуги) создаётся горизонтальный слой (QHBoxLayout), на который добавляются две кнопки: одна для свойств (Properties), другая для услуг (Services). Каждая кнопка связана с событием (clicked), которое активирует определённый метод (display_properties или display_services) для отображения соответствующих данных. Эти методы будут изменять содержимое основного списка и других элементов интерфейса в зависимости от выбранного режима просмотра.

Список элементов. QListWidget используется для отображения списка элементов, который может включать в себя данные о свойствах или услугах, в зависимости от выбора пользователя. Список настраивается так, чтобы реагировать на двойной клик по его элементам (itemDoubleClicked), что запускает метод item_double_clicked, обеспечивающий более глубокую обработку выбранного элемента.

Боковая панель. Добавляется дополнительный вертикальный слой, предназначенный для размещения дополнительных элементов управления или информационных блоков, таких как детальная информация о выбранном объекте или дополнительные опции.

Инициализация и отображение. В конце конструктора происходит инициализация переменных для возможных диалоговых окон (property_dialog, service_dialog) и хранения последнего выбранного элемента (last_selected_item). Устанавливается начальный вид отображения данных (current_view установлено в properties), и вызывается метод display_properties, который загружает и отображает начальный набор данных в списке при первом запуске приложения.

Код этого функционального блока представлен ниже, на рисунке 9.

```

1 class MainWindow(QMainWindow):
9     def __init__(self):
10        super().__init__()
11        self.setGeometry(100, 100, 800, 600)
12        self.centralWidget = QWidget()
13        self.setCentralWidget(self.centralWidget)
14        self.main_layout = QVBoxLayout()
15        self.centralWidget.setLayout(self.main_layout)
16
17        self.list_selection_layout = QHBoxLayout()
18        self.properties_button = QPushButton("Properties")
19        self.services_button = QPushButton("Services")
20        self.properties_button.clicked.connect(self.display_properties)
21        self.services_button.clicked.connect(self.display_services)
22        self.list_selection_layout.addWidget(self.properties_button)
23        self.list_selection_layout.addWidget(self.services_button)
24        self.main_layout.addLayout(self.list_selection_layout)
25
26        self.main_list = QListWidget()
27        self.main_list.itemDoubleClicked.connect(self.item_double_clicked)
28        self.main_layout.addWidget(self.main_list)
29
30        self.side_panel = QVBoxLayout()
31        self.main_layout.addLayout(self.side_panel)
32        self.property_dialog = None
33        self.service_dialog = None
34        self.last_selected_item = None
35        self.current_view = 'properties'
36        self.setup_ui()
37        self.display_properties()

```

Рисунок 9 – Инициализация главного графического окна

Класс `MainWindow`, созданный с использованием библиотеки `PyQt6`, представляет собой структурированное графическое пользовательское окно, которое эффективно управляет интерфейсом приложения для взаимодействия с данными о свойствах и услугах. Он обеспечивает универсальную структуру с центральным виджетом и вертикальным слоем, что упрощает добавление и управление дополнительными элементами интерфейса, такими как кнопки, списки и панели. Эта универсальность повышает гибкость в отображении данных, позволяя пользователю легко переключаться между разными видами информации через интуитивно понятные кнопки управления и соответствующие методы обработки.

Интерактивность интерфейса подчеркивается через динамичное взаимодействие с пользователем, включая обработку событий от нажатий на кнопки и двойных кликов по элементам списка. Это взаимодействие улучшает пользовательский опыт, делая программу более удобной и понятной для использования.

Кроме того, архитектура MainWindow спроектирована таким образом, что обеспечивает легкость в добавлении дополнительных компонентов, таких как диалоговые окна и информационные панели. Это делает приложение масштабируемым и легко адаптируемым под разнообразные требования пользователей или изменения в данных.

Начальная настройка и загрузка данных о свойствах при запуске приложения также подчеркивают проработанность пользовательского интерфейса, что снижает начальное время взаимодействия пользователя с приложением и повышает его эффективность. В целом, MainWindow представляет собой комплексный и эффективный компонент управления пользовательским интерфейсом, предоставляющий широкие возможности для дальнейшего расширения и адаптации под конкретные нужды.

3.4 Инициализация функций управления данными

Так как в приложении используется база данных, необходимо её создать. За это отвечают функции управления данными, в качестве примера предоставлено три функции, подробнее на рисунке 10.

```
11 class DatabaseAdapter:
12     def __init__(self):
13         self.conn = psycopg2.connect(dbname=dbname, user=user, password=password, host=host, port=port)
14
15     def add_property(self, uuid, data):
16         s = f"INSERT INTO properties (uuid, value) VALUES ({str(uuid)}, '{data}');"
17         try:
18             cur = self.conn.cursor()
19             cur.execute(s)
20             self.conn.commit()
21         except Exception as e:
22             handle_exception(e)
23             self.conn.rollback()
24
25         finally:
26             cur.close()
27
28     def add_service(self, uuid, data):
29         s = f"INSERT INTO services (uuid, value) VALUES ({str(uuid)}, '{data}');"
30         try:
31             cur = self.conn.cursor()
32             cur.execute(s)
33             self.conn.commit()
34         except Exception as e:
35             handle_exception(e)
36             self.conn.rollback()
37
38         finally:
39             cur.close()
```

Рисунок 10 – Функции управления данными

Конструктор `__init__`. В конструкторе класса происходит инициализация подключения к базе данных. Для этого используется метод `psycopg2.connect`, который принимает параметры доступа к базе данных, такие как `dbname` (имя базы данных), `user` (пользователь), `password` (пароль), `host` (хост) и `port` (порт). Эти параметры необходимо предоставить при создании экземпляра класса, и они должны быть определены в коде до вызова или передаваться динамически.

Метод `add_property`. Этот метод предназначен для добавления записи о свойстве в таблицу `properties` базы данных. Аргументы метода: `uuid` (уникальный идентификатор объекта) и `data` (данные о свойстве). Строка SQL-запроса формируется с использованием этих аргументов. Производится попытка выполнить запрос с помощью курсора. При успехе изменения фиксируются в базе данных методом `commit`. В случае возникновения исключения (например, ошибки базы данных) исключение обрабатывается функцией `handle_exception`, а изменения откатываются с помощью `rollback`. Курсор закрывается в блоке `finally`, что гарантирует освобождение ресурсов независимо от результата операции.

Метод `add_service`. Метод аналогичен методу `add_property`, но он работает с таблицей `services` для добавления записей о предоставляемых услугах. Принцип работы и обработка исключений такие же, как и в методе `add_property`.

Этот класс `DatabaseAdapter` предоставляет базовый интерфейс для работы с базой данных, позволяя добавлять информацию о свойствах и услугах. Класс является фундаментальным компонентом для работы с информационными ресурсами в приложении.

3.5 Контрольный пример реализации информационной системы учета объекта и предоставляемых услуг

В данном параграфе представлен функциональный интерфейс разработанного программного обеспечения, ключевые аспекты его взаимодействия с пользователем, а также демонстрация основных возможностей системы. В современных условиях разработки программных продуктов особое внимание уделяется не только технической реализации функционала, но и удобству использования, интуитивно понятным и доступным интерфейсам, которые могут существенно повысить эффективность работы конечных пользователей.

Программное обеспечение, являясь сложным инструментом для решения специфических задач, требует тщательного подхода к проектированию интерфейса.

На рисунке 11 изображена вкладка «Недвижимость».

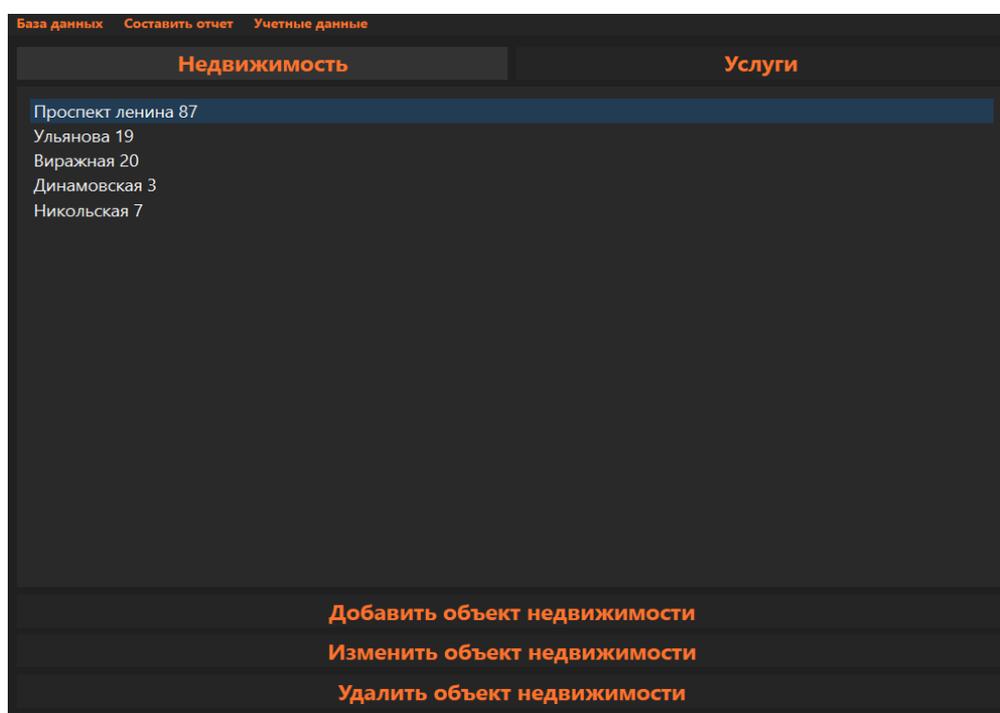


Рисунок 11 – Интерфейс вкладки «Недвижимость»

Перед пользователем предстаёт список объектов недвижимости по адресам. Так же, снизу окна, есть функциональные кнопки добавления объекта недвижимости, изменения и удаления.

В верхней части окна находятся вкладки база данных, составление отчета и учетные данные. База данных отвечает за синхронизацию с удалённой базой данных после внесения необходимых изменений, составление отчёта позволяет выбрать объекты, по которым необходимо составить отчёт. Кнопка учетных данных позволит администратору системы ввести своё имя пользователя и пароль, для доступа к функционалу администратора.

На рисунке 12 изображен функционал кнопки «Составить отчет».

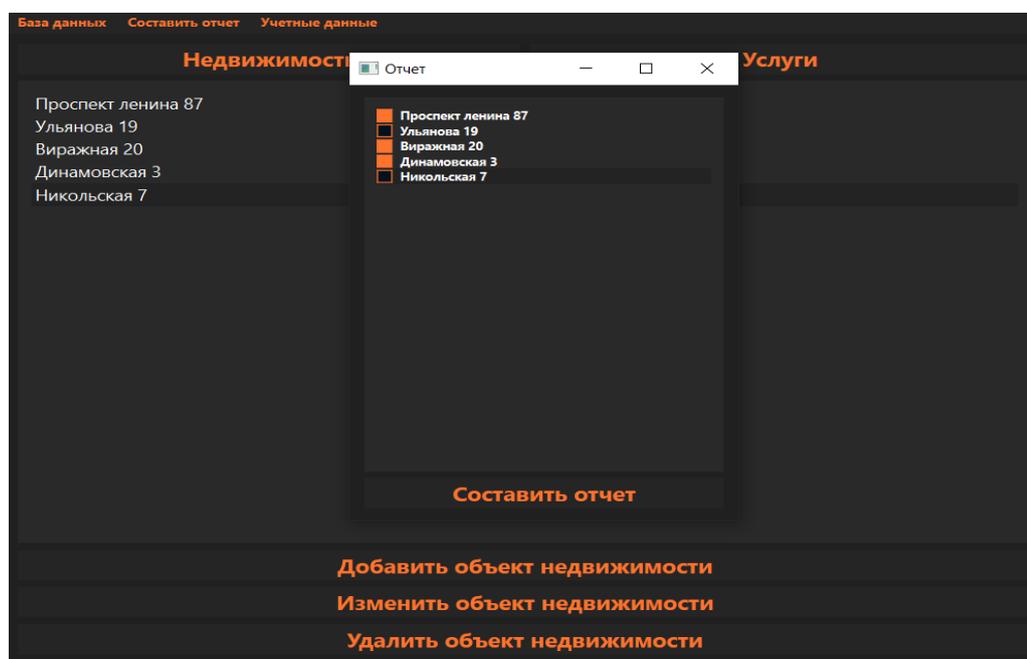


Рисунок 12 – Функционал вкладки «Составить отчет»

На данной вкладке пользователь имеет возможность выбрать необходимые объекты недвижимости, и приложение выгрузит отчет по их услугам и потреблению ресурсов в отдельный файл, для предоставления его компании-поставщику.

На рисунке 13 изображена вкладка «Учетные данные»

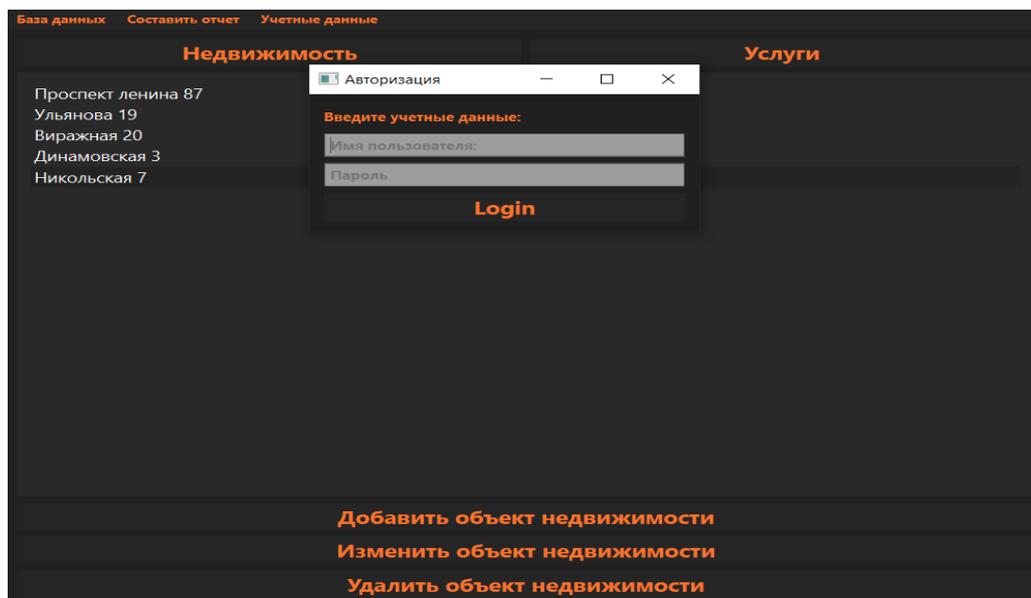


Рисунок 13 – Функционал вкладки «Учетные данные»

Данная вкладка позволяет администратору информационной системы войти под своими учетными данными, и разблокировать функции редактирования, а так же добавления или удаления услуг.

При нажатии на объект, откроется его подробное описание. Демонстрация на рисунке 14.

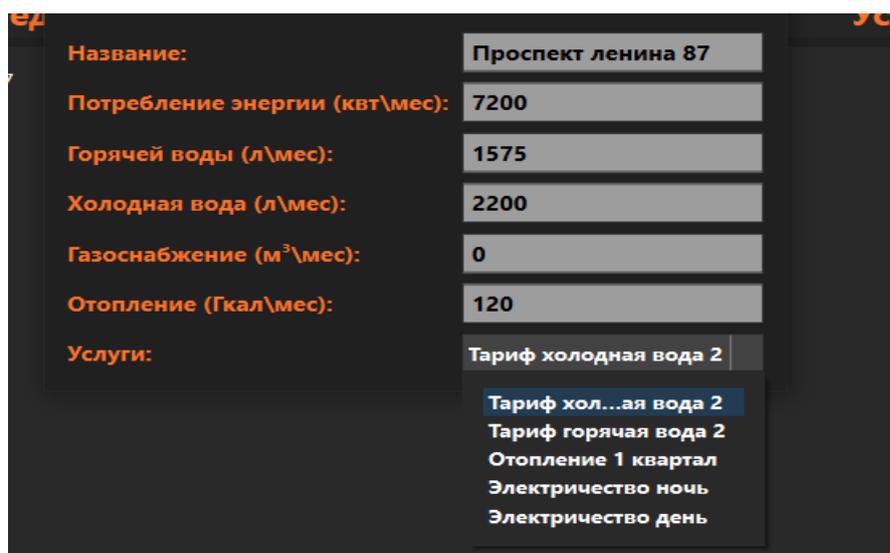


Рисунок 14 – Функционал списка объектов

На данной вкладке пользователь может просмотреть информацию по потреблению объектом ресурсов. Согласно требованиям, приведено потребление электроэнергии, горячей и холодной воды, а так же газоснабжения и отопления.

В выпадающем списке «Услуги» можно ознакомиться с услугами, которые подключены к данному объекту. На каждом объекте, если это необходимо, находятся разные тарифы услуг. Подробнее на рисунке 15.

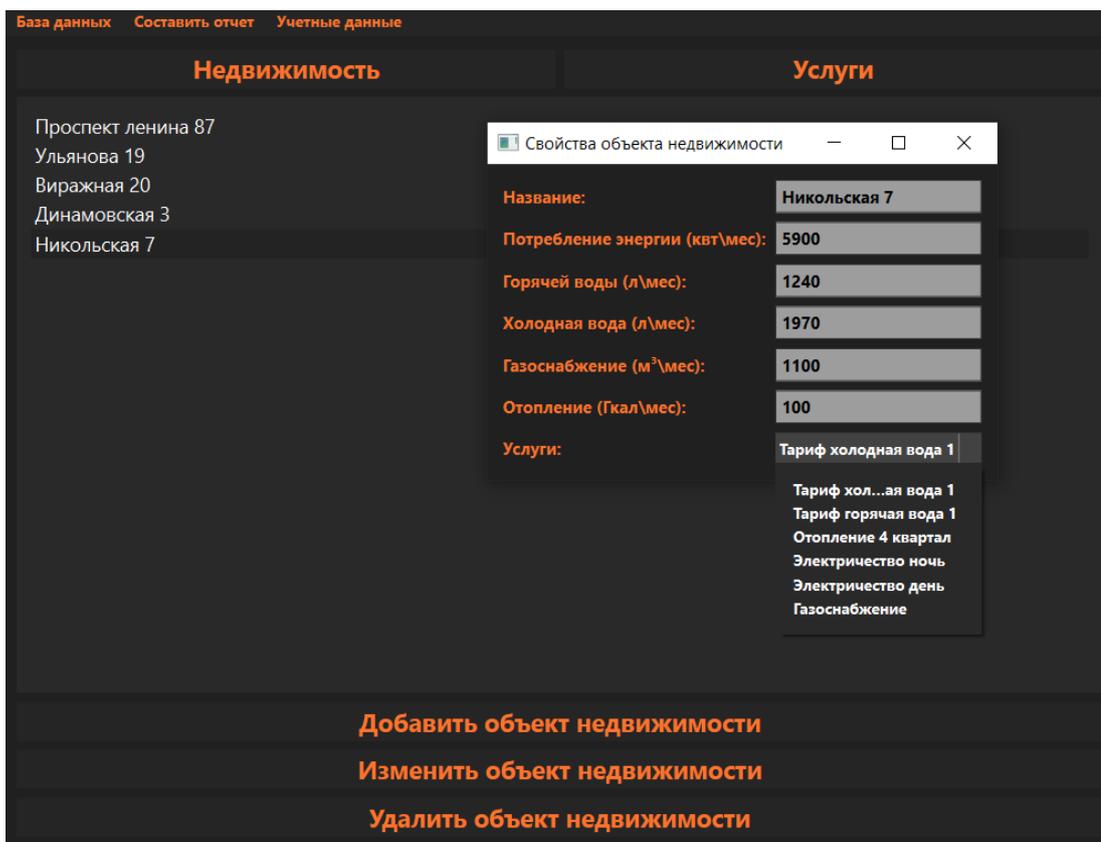


Рисунок 15 – Функционал объекта недвижимости

На рисунке изображён другой объект недвижимости. Он имеет свои собственные показатели потребления ресурсов, а так же свой набор услуг.

Вкладка со списком услуг представлена на рисунке 16.

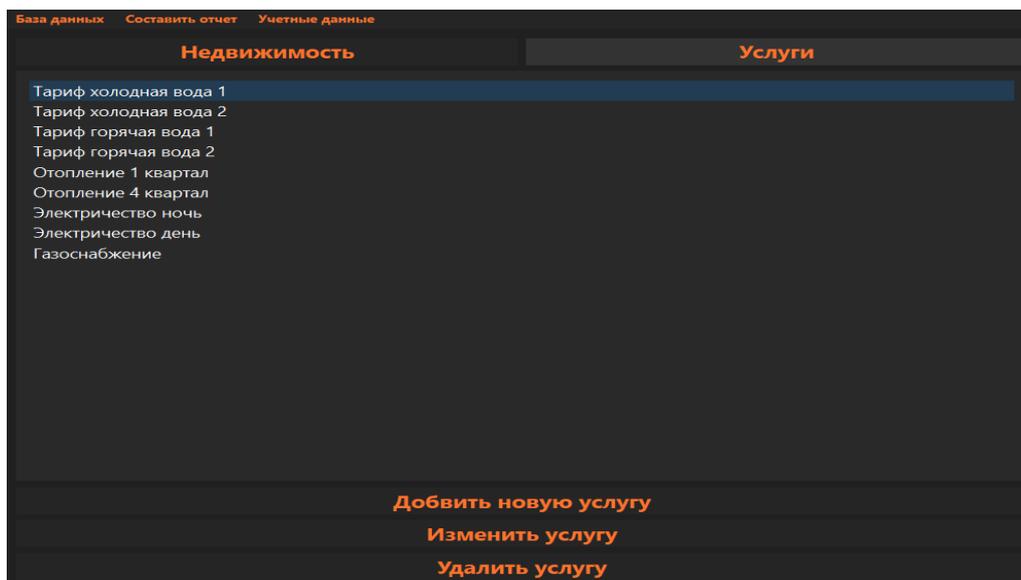


Рисунок 16 – Функционал вкладки «Услуги»

На данной вкладке пользователь может ознакомиться со списком услуг, которые были добавлены в базу данных администратором системы. Сотрудник без доступа администратора сможет добавлять созданные услуги к новым или уже существующим объектам, но их созданием занимается администратор.

Таким образом, интерфейс информационной системы соответствует всем предъявленным требованиям, является интуитивно понятным и отзывчивым по отношению к рядовому пользователю.

Пример отчета, который автоматически формируется программой, представлен на рисунке 17.

	A	B	C	D	E	F	G
1	Адрес	Потребление энергии	Потребление горячей воды	Потребление холодной воды	Газоснабжение	Отопление	
2	Проспект ленина 87	7200	1575	2200	0	120	
3	Ульянова 19	5350	1200	1800	1500	80	
4	Динамовская 3	3500	748	1100	0	75	
5							
6							
7							
8							

Рисунок 17 – Автоматически сформированный отчет

Отчет формируется в формате .xlsx, т.к. это наиболее часто встречающийся формат для работы с всевозможными отчетами, документами и статистическими данными. Для каждого из объектов представлены данные по потреблению всех видов услуг.

Это позволит предоставлять данные по объектам по запросу заказчиков.

3.6 Требования к аппаратно-программному обеспечению приложения

Приложение предназначено для работы с операционной системой Windows, Unix-подобными операционными системами.

Требования к аппаратному обеспечению:

- процессор: минимум Intel Core i3 или аналогичный от AMD;
- оперативная память: не менее 500 мб;
- жесткий диск: минимально необходимо 10 ГБ свободного пространства на диске для установки и хранения данных.

Требования к программному обеспечению:

- операционная система: Windows 10 или выше, Linux (Ubuntu 20.04 LTS или выше), MacOS Catalina или выше;
- Python: Версия 3.7 или выше;
- библиотека Qt6: установленная библиотека Qt6 для разработки графического интерфейса пользователя;
- СУБД PostgreSQL: PostgreSQL версии 12 или выше;
- дополнительные библиотеки Python: Установка таких библиотек, как psycopg2 для работы с PostgreSQL и библиотеки PyQt6 необходимы для функционирования приложения;
- поддержка сети: доступ в интернет для работы с базой данных.

Все компоненты должны быть совместимы друг с другом и с указанной операционной системой.

Рекомендуется регулярно проводить тестирование системы на совместимость после обновлений ПО и ОС.

Эти требования гарантируют стабильную работу приложения для учета объектов недвижимости и предоставления услуг, обеспечивая его эффективное функционирование и высокую производительность.

3.7 Тестирование информационной системы

Для тестирования и произведения испытаний информационной системы был определен метод «Черного ящика». Данный метод подразумевает собой исследование того, соответствует ли разработанная информационная система имеющимся требованиям [2]. В данном методе также используется проверка работы ИС при вводе ошибочных входных данных. Результат проведения тестирования методом «Черного ящика» представлен в таблице 3.

Таблица 4 – Проведение тестирования методом «Черного ящика»

Описание теста	Входные данные	Ожидаемый результат	Фактический результат	Статус
Полнота главного меню	Проверить наличие необходимых элементов	Все необходимые элементы в наличии и готовы к работе	Все необходимые элементы в наличии и готовы к работе	Пройден
Формы ввода данных	Проверить наличие форм для ввода данных	Формы в наличии и готовы к заполнению	Формы в наличии и готовы к заполнению	Пройден
Соответствие компонентов панели управления	Проверить соответствие компонентов панели управления их функциональной задаче	Все компоненты панели управления соответствуют своей функциональной задаче	Все компоненты панели управления соответствуют своей функциональной задаче	Пройден

Продолжение таблицы 4

Описание теста	Входные данные	Ожидаемый результат	Фактический результат	Статус
Проверка создания собственности	Проверить работоспособность создания новой собственности и её внесения в базу данных	Собственность с указанными параметрами создана и внесена в базу данных	Собственность с указанными параметрами создана и внесена в базу данных	Пройден
Проверить сохранение данных в БД после перезагрузки программы	Внести новую собственность в базу данных и выключить, а затем включить приложение	Собственность и все изменения сохранились	Собственность и все изменения сохранились	Пройден

Информационная система учёта объектов и предоставляемых услуг успешно прошла все пункты тестирования методом «черный ящик», что подтверждает её функционал. Информационная система полностью соответствует всем заявленным условиям и требованиям.

Выводы по главе 3

В третьей главе была продемонстрирована реализация информационной системы по учету объектов и предоставляемых услуг. Проведена спецификация выбранных технологий реализации, обоснован выбор библиотек и наборов инструментов. Так же продемонстрирована инициализация главного графического окна и функция управления данными, которая предоставляет базовый интерфейс для работы с базой данных. Был представлен пример работы информационной системы и её тестирование методом «Черного ящика».

Заключение

В рамках данной выпускной квалификационной работы была успешно разработана и внедрена информационная система, предназначенная для учёта объектов недвижимости и предоставляемых услуг. Основной целью было создание эффективного инструмента, который способствовал бы оптимизации процессов управления активами и услугами производственной компании, а также повышению общей эффективности учетных операций.

Для реализации информационной системы смоделированы бизнес-процессы компании, выявлены недостатки, которые требовали автоматизированного варианта решения. Это позволило точно определить требования к разрабатываемой системе и обеспечить её максимальную соответствие реальным потребностям пользователя.

Для реализации был проведен анализ современных технологий и инструментов разработки, что позволило выбрать наиболее подходящие и эффективные средства для создания системы. В результате разработки был создан конечный продукт, который включает в себя удобный пользовательский интерфейс и широкий функционал для управления данными о недвижимости и услугах. Созданный продукт удовлетворяет всем поставленным требованиям, таким как: круглосуточный режим функционирования, возможность расширения системы, интерфейс приложения является понятным и удобным, обеспечена точность расчётов данных, показания учитывают услуги электроэнергии, горячей воды, холодной воды, газоснабжения, отопления, на главной странице представлен список всех объектов, поставленных на учёт.

Для дальнейшего совершенствования информационной системы также имеется возможность добавить функционал, который будет применим к более узким особенностям производственной компании ООО «Квартплата 24», что дает большую эффективность по сравнению с другими продуктами.

Список используемой литературы и используемых источников

1. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. 2-е издание. Пер. с англ. СПб. : Символ-Плюс, 2007. 624 с.
2. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб. : Питер, 2004. 318 с.
3. Введение в реляционные базы данных и программирование на языке SQL [Электронный ресурс] URL: https://kpfu.ru/staff_files/F_307454421/Vvedenie_v_SUBD.pdf
4. Все о ЖКХ на 19 год: услуги, тарифы, платежи и сборы, основания не платить или платить меньше / сост. Е. Давыденко. – Москва: Издательство АСТ, 2018. – 224 с. (Справочник для населения)
5. Грекул В.И. Проектирование информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. - М.: Интернет-Уни-т информ. технол., 2005. - 304 с.
6. Джуба С. , Волков А. «Изучаем PostgreSQL 10» 2018 – 400 с.: ISBN: 978-5-97060-643-8 Дополнительная литература и учебные материалы.
7. Как начать моделировать бизнес-процессы в BPMN [Электронный ресурс] URL : <https://habr.com/ru/articles/697326/> (дата обращения 24.03.2024).
8. Курс «PostgreSQL для начинающих»: #1 — Основы SQL [Электронный ресурс] URL: <https://habr.com/ru/companies/tensor/articles/779698/>
9. Ларман К. Применение UML и шаблонов проектирование. 2-е издание. Пер. с англ. М. : Издательский дом Вильямс, 2004. 624 с.
10. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
11. Миндалёв И.В. Моделирование бизнес-процессов с помощью IDEF0, DFD, BPMN : учеб. пособие / И.В. Миндалёв; Краснояр. гос. аграр. ун-

т. - Красноярск, 2016. – 123 с. перераб. и доп. - Москва : Горячая линия - Телеком, 2012. - 516 с. : ил. - ISBN 978-5- 9912-0193-3. Дополнительная литература и учебные материалы.

12. Пискун А. – «Краткий руководитель по семейству нотаций IDEF» [Электронный ресурс] URL: https://www.antonpiskun.pro/kratkij-putevoditel-po-semejstvu-notaczij-idef/#IDEF1X_IDEF1_Extended

13. Производственная компания [Электронный ресурс]. URL: <https://www.kvp24.ru/> (дата обращения 13.03.2024).

14. Рамальо Л. Python. К вершинам мастерства [Электронный ресурс] URL: <https://lib.samtuit.uz/uploads/files/61a215578e5056.58184996.pdf> учебник. - Москва : ДМК Пресс, 2016. - 768 с. : ил. - ISBN 978-5-97060-315-8. Дополнительная литература и учебные материалы.

15. Франсуа Шолле «Глубокое обучение на Python» Глубокое обучение на Python. 2-е межд. издание. — СПб.: Питер, 2023. —576 с.: ил. - ISBN 978-5-4461-1909-7 [Электронный ресурс] URL: <https://coollib.net/b/651746-fransua-sholle-glubokoe-obuchenie-na-python/read>

16. Integration DEfinition for information modeling [Электронный ресурс] URL : <https://en.wikipedia.org/wiki/IDEF1X> (дата обращения 07.04.2024).

17. Documentation PostgreSQL [Электронный ресурс] URL: <https://www.postgresql.org/docs/>

18. Documentation PyQt6 [Электронный ресурс] URL: <https://www.riverbankcomputing.com/static/Docs/PyQt6/>

19. Documentation Psycopg – PostgreSQL database adapter for Python [Электронный ресурс] URL: <https://www.psycopg.org/docs/>

20. Documentation Qt6 cross-platform framework [Электронный ресурс] URL: <https://doc.qt.io/>