

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)
Разработка программного обеспечения
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка программного обеспечения информационной
системы управления промышленными поставками химических продуктов

Обучающийся Л.Е. Яцкаер (Инициалы Фамилия) _____ (личная подпись)
Руководитель канд. техн. наук, Н.В. Хрипунов
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема выпускной квалификационной работы – «Разработка программного обеспечения информационной системы управления промышленными поставками химических продуктов».

Актуальность работы определена необходимостью автоматизации процессов управления поставками продукции организации ООО Торговый Дом «Профиль-Снаб». Необходимость обусловлена повышенными затратами на издержки, связанные с количеством затрачиваемого времени на обработку заказов, а также числом совершаемых ошибок при ручном заполнении документации.

Целью исследования является создание автоматизированной информационной системы, оптимизирующей горизонтальную коммуникацию в компании, повышающей эффективность работы персонала и улучшающей качество и скорость обслуживания клиентов.

Объект исследования – процесс управления промышленными поставками химической продукции.

Предмет исследования – автоматизация управления поставками, включающая обработку заказов и формирование сопроводительной документации.

Исследование основано на методах системного анализа, проектирования баз данных, а также, функционального и логического моделирования.

Результаты работы включают в себя описание разработанной информационной системы, основанной на клиент-серверной архитектуре. Система создана при помощи языка программирования Python и использует PostgreSQL в качестве СУБД.

Структура работы включает в себя введение, три главы, заключение и список используемой литературы.

Объем выполненной работы составляет 60 страниц текста, 22 рисунка и 2 таблицы.

Содержание

Введение.....	5
1 Функциональное моделирование предметной области.....	7
1.1 Техничко-экономическая характеристика предметной области	7
1.2 Описание ИТ-инфраструктуры предприятия	10
1.3 Концептуальное моделирование предметной области.....	13
1.3.1 Разработка и анализ модели бизнес-процесса «как есть».....	13
1.3.2 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии	14
1.4 Анализ существующих разработок на предмет соответствия	15
сформулированным требованиям	15
1.4.1 Определение критериев анализа.....	15
1.4.2 Сравнительная характеристика существующих разработок	16
1.5 Постановка задачи на разработку проекта создания/внедрения ИС.....	19
2 Логическое проектирование АИС	21
2.1 Разработка модели бизнес-процесса «Как будет».....	21
2.2 Выбор технологии логического моделирования АИС	22
2.3 Информационное обеспечение АИС	24
2.3.1 Используемые классификаторы и системы кодирования.....	24
2.3.2 Характеристика нормативно-справочной и входной оперативной информации	25
2.3.3 Характеристика выходной информации.....	26
2.4 Проектирование базы данных АИС.....	27
2.4.1 Выбор технологии проектирования БД АИС.....	27
2.4.2 Разработка концептуальной модели данных АИС	28

2.4.3	Разработка логической модели данных АИС.....	30
2.5	Требования к аппаратно-программному обеспечению АИС	32
3	Физическое проектирование АИС.....	35
3.1	Выбор архитектуры АИС.....	35
3.2	Выбор технологии разработки программного обеспечения АИС.....	36
3.3	Разработка физической модели данных АИС	38
3.4	Разработка программного обеспечения АИС	40
3.4.1	Схема взаимосвязи модулей приложения АИС	40
3.4.2	Описание модулей приложения АИС	42
3.5	Описание функциональности АИС.....	48
3.6	Тестирование программного проекта как неотъемлемая часть разработки программного обеспечения.....	50
3.6.1	Выбор методов тестирования программного продукта	50
3.6.2	Описание программного кода тестирования АИС	51
3.6.3	Результаты тестирования АИС	54
	Заключение	56
	Список используемой литературы	58

Введение

В современных условиях стремительного развития экономики, оптимизация бизнес-процессов становится одним из ключевых требований для развития организаций, работающих в различных отраслях промышленности. Предприятиям, специализирующимся на производстве, поставке и транспортировке химических продуктов особенно важна организация эффективного управления логистическими цепочками, необходимая для обеспечения конкурентоспособности и устойчивого развития.

Данная выпускная квалификационная работа (ВКР) посвящена разработке автоматизированной информационной системы (АИС) управления промышленными поставками химических продуктов для организации ООО Торговый Дом «Профиль-Снаб».

Актуальность выбранной темы обусловлена тем, что применяемые в компании процессы управления транспортными перевозками и взаимодействие между подразделениями, в частности, транспортным и тендерным отделами, характеризуются низким уровнем автоматизации. Это может являться причиной ряда проблем, негативно сказывающихся на эффективности всей организации, а именно:

- риски потери информации;
- низкая эффективность;
- отсутствие централизации.

Целью работы является разработка АИС для управления транспортными операциями и базой данных продукции и контрагентов, позволяющей автоматизировать ключевые бизнес-процессы, связанные с логистикой, повысить эффективность работы транспортного и тендерного отделов, а также оптимизировать рабочий процесс всех подразделений описываемой организации

Для достижения поставленной цели необходимо выполнить ряд комплексных задач [13]:

- анализ существующих бизнес-процессов, связанных с транспортными операциями в компании;
- разработка концептуальной модели процесса управления транспортом;
- выполнение логического моделирования базы данных системы;
- создание элементов пользовательского интерфейса;
- проведение расчета экономической эффективности разработки и внедрения АИС.

В работе применен ряд теоретических методов исследования, включающий в себя бизнес-анализ, моделирование информационных систем и баз данных [11], а также различные методы проектирования пользовательских интерфейсов [7].

ВКР содержит введение, три основные главы, заключение и список используемой литературы. Первая глава содержит анализ внутренних процессов организации, во второй главе рассматриваются архитектура и требования к аппаратно-программному обеспечению будущей системы, в третьей главе рассмотрены процесс разработки и оценка экономической эффективности внедрения готового программного продукта. В заключении изложены результаты проделанной работы, а также дана их оценка в контексте поставленных задач и целей.

1 Функциональное моделирование предметной области

1.1 Технико-экономическая характеристика предметной области

Организация ООО Торговый Дом «Профиль-Снаб» занимается поставкой и транспортировкой разнообразных материалов, оборудования для предприятий промышленного сектора. Продукция, поставляемая компанией, удовлетворяет запросы химической и сельскохозяйственной отраслей.

Организационная структура компании – это пакет официальных документов, отражающих иерархию и состав организации, а также функции, права и обязанности ее основных элементов. Рассмотрим организационную структуру компании ООО ТД «Профиль-Снаб».

Организационная структура описываемой организации линейно-функциональная. Часть прав и обязанностей в компании распределяются линейно, остальные входят в зону ответственности функциональных подразделений, главная задача которых – подготовка сведений, необходимых для принятия взвешенных решений по запросу линейных управляющих.

Структура описываемой организации выглядит следующим образом: руководителем, согласно уставу, является директор. Директор – лицо, принимающее ключевые решения в компании. В его непосредственном подчинении находятся руководители отделов продаж, транспорта, а также руководители юридического, тендерного и ИТ отделов. Организационная структура представлена на схеме, изображенной на рисунке 1.



Рисунок 1 – Организационная структура ООО ТД «Профиль-Снаб»

Транспортный отдел представляет собой важную структуру внутри компании. На него возложены ключевые обязанности по организации и координации транспортировки промышленной химии как на территории Российской Федерации, так и в странах СНГ. Осуществление грузоперевозок у выполняется с применением колесного транспорта и железнодорожных перевозок. Основной задачей отдела является обеспечение эффективности и своевременности поставок.

Также, важной частью работы транспортного отдела остается не только обеспечение своевременных поставок продукции, но и проведение комплексных логистических операций. В контексте этих задач, специалисты отдела планируют маршруты, занимаются мониторингом рынка транспортных услуг и осуществляют выбор оптимальных вариантов перевозок.

В зону ответственности транспортного отдела входит контроль работы водителей грузового транспорта. Необходимо осуществлять контроль за соблюдением норм трудового времени, отдыха и соблюдением правил безопасности дорожного движения. Поддержание высоких стандартов безопасности в перевозках является одним из основополагающих принципов компании.

Помимо прочего, отдел занимается регистрацией транспортных средств, вагонов и автоцистерн в регуляторных органах. Для этого необходимо готовить и предоставлять государственным службам нормативные документы согласно регламенту. Соблюдение законодательных требований и нормативов позволяет компании эффективно взаимодействовать с государственными структурами.

Путевые листы, составляемые для каждой поездки, являются основой документооборота логистических операций. Сотрудники отдела ведут учет маршрутов и фиксируют время и место загрузки и разгрузки продукции, что позволяет обеспечить прозрачность работы и контроль над выполнением перевозок.

Для взаимодействия с другими структурами организации, сотрудники транспортного отдела формируют отчетную документацию. Она предоставляет собой совокупность информации о выполненных перевозках и служит основой для анализа эффективности бизнес-процессов и выявления возможности оптимизации работы отдела.

Таким образом, транспортный отдел обеспечивает бесперебойное функционирование логистических цепочек компании и активно участвует в стратегическом развитии организации. Подразделение играет важную роль в комплексе действий по обеспечению высокого уровня обслуживания клиентов и выполнения контрактных обязательств, что положительно сказывается на репутации компании в бизнес-среде.

Постоянное взаимодействие транспортного отдела с другими подразделениями компании, тендерным, юридическим и ИТ отделами, обеспечивает слаженную работу организации. Сотрудничество отделов позволяет эффективно управлять всем жизненным циклом перевозок, от участия в тендерах и заключения контрактов, до успешной доставки заказанной продукции.

Тендерные специалисты после первого контакта с клиентом, передают транспортному отделу полученную необходимую информацию о выигранных конкурсах и полученных заказах на поставку продукции. Они обозначают детали о грузе и его объеме, особенностях маршрута, сроках поставки и прочих ключевых параметрах. После получения сообщения, транспортный отдел получает в распоряжение детальную и структурированную информацию, необходимую для разработки оптимальных планов перевозок и обеспечения их успешной реализации.

Юридический отдел, в свою очередь, занимается обеспечением правовой устойчивости транспортных операций. Юристы консультируют транспортный отдел по всем возникающим профильным вопросам, предоставляя правовую экспертизу контрактов и рекомендации по соблюдению нормативов в области транспортных услуг. Их работа позволяет

минимизировать риски и обеспечить соблюдение всех требований законодательства в сфере грузоперевозок.

ИТ отдел обеспечивает функционирование информационной структуры организации. ИТ специалисты поддерживают бесперебойную работу всех информационных систем организации. Их работа включает в себя мониторинг работы офисных компьютеров, внедрение и поддержание работы информационных систем, обслуживание офисной техники, а также решение возникающих вопросов.

Сотрудничество между отделами внутри компании создает сбалансированный механизм, который обеспечивает эффективное функционирование логистических процессов и способствует развитию глобальных стратегий компании. Взаимодействие и сотрудничество способствуют успеху компании в динамичной среде.

1.2 Описание ИТ-инфраструктуры предприятия

Информационная инфраструктура организации базируется на современных технических решениях. Она включает в себя серверное решение HP ProLiant DL360 в роли основного элемента ИТ-структуры. Этот сервер обеспечивает стабильную работу всех информационных систем, а также обеспечивает доступ к принтерам, обмену файлами и другим необходимым службам для работников организации.

Характеристики сервера:

- процессор: Intel Xeon E5 2620 (6C, 15Mb Cache L3, 2.10 GHz);
- оперативная память: 64 Gb (8x DDR3 DIMM 1600MHz 8 Гб);
- сетевой адаптер: Broadcom NetXtreme Gigabit Ethernet;
- жесткий диск: DELL PERC H710P SCSI Disk Device 1.2 TB.

Данный сервер выступает в роли многозадачного узла, обеспечивая выполнение множества функций внутри организации [12].

Функционал почтового сервера обеспечивает стабильную и безопасную работу корпоративной электронной почты, обеспечивая эффективный и быстрый обмен информацией между сотрудниками.

Программный пакет 1С также установлен на сервер и доступен каждому сотруднику благодаря использованию службы удаленных рабочих столов.

Серверная часть антивирусного пакета Pro32 Endpoint Security гарантирует защиту информационной среды предприятия от вредоносных программ, позволяет контролировать групповые политики безопасности и выполнять массовый мониторинг рабочих мест сотрудников.

Централизованное хранение и управление файлами в сетевых папках обеспечивает удобный доступ сотрудников к необходимой информации в локальной сети компании.

Система управления базами данных (СУБД) обеспечивает эффективное хранение и оперативный доступ к базам данных организации, обеспечивая бесперебойное выполнение бизнес-процессов.

В качестве серверной операционной системы была выбрана Windows Server 2012 R2. Это решение обеспечивает стабильность работы и совместимость с со всем актуальным ПО, используемым в корпоративных целях.

В качестве источника бесперебойного питания (ИБП) применяется APC Smart-UPS RT 1000VA в комбинации со стабилизатором электрической сети. Такая аппаратная связка гарантирует сохранность данных и стабильность работы сервера даже в случае скачков электрического напряжения или остановки подачи питания.

Каждый сотрудник имеет в личном распоряжении персональный компьютер, обладающий достаточной производительностью для решения типовых офисных задач.

Компонентная база типового офисного ПК выглядит следующим образом:

- процессор: Intel Core i3 (Gen 11 или выше)/AMD Ryzen 3 (Gen 5 или выше);
- оперативная память: 16 Gb (DDR4);
- жесткий диск: SATA/M2 NVME 512 Гб.

ИТ-инфраструктура компании представлена на схеме, изображенной на рисунке 2.

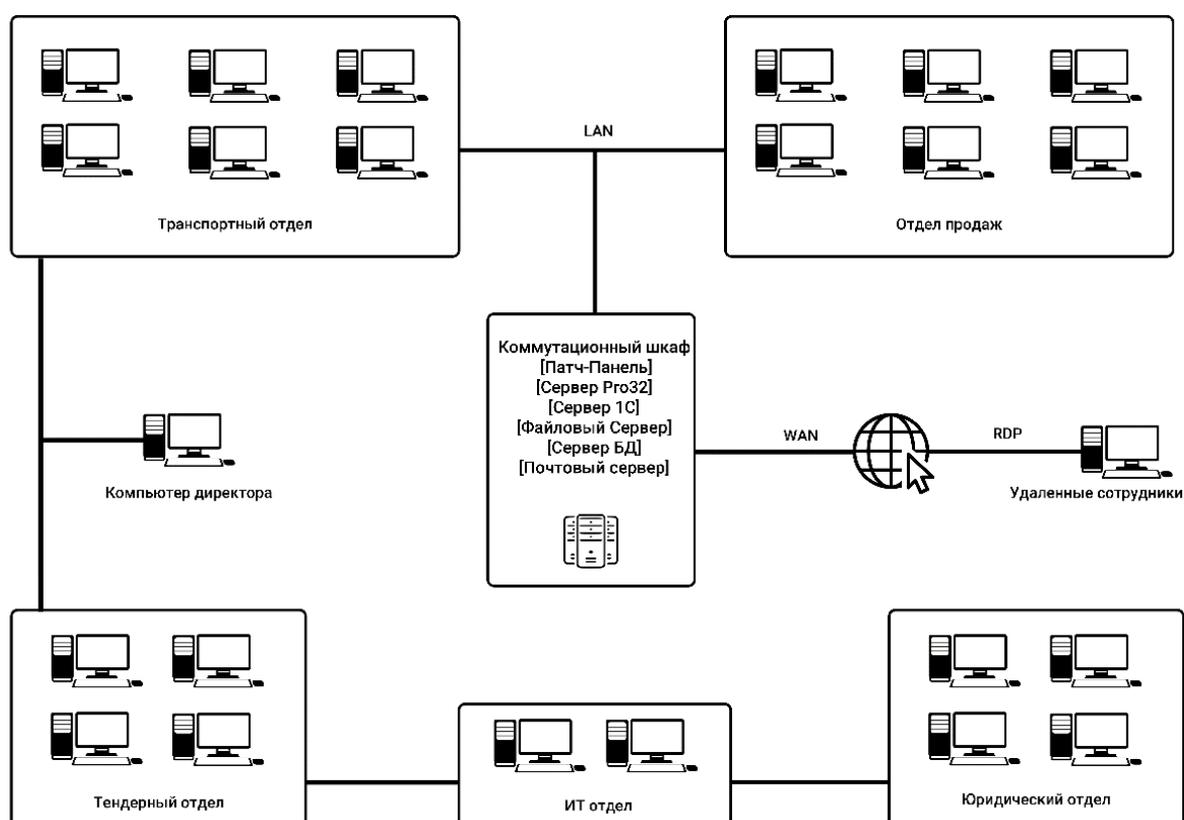


Рисунок 2 – ИТ-инфраструктура ООО ТД «Профиль-Снаб»

Эти характеристики не являются жестким требованием, а представляют собой усредненный комплекс рекомендаций для обеспечения сотрудников необходимой на сегодняшний день вычислительной мощностью, требуемой для решения комплексных рабочих задач. Также, компоненты РМ могут

подбираться или меняться в зависимости от задач, которые предстоит выполнять конкретному сотруднику. Такой подход обеспечивает оптимальное соответствие вычислительных мощностей с потребностями персонала. Часть сотрудников использует стационарные ПК, но большинство предпочитают ноутбуки, которые способны обеспечить им мобильность при работе вне пределов офиса.

1.3 Концептуальное моделирование предметной области

1.3.1 Разработка и анализ модели бизнес-процесса «как есть»

В настоящее время в работе организации не используется какое-либо программное решение, которое могло бы упростить и повысить эффективность работы транспортного отдела и его взаимодействие с остальными подразделениями. На рисунке 3 изображена BPMN-диаграмма бизнес-процесса «Как есть» [21].

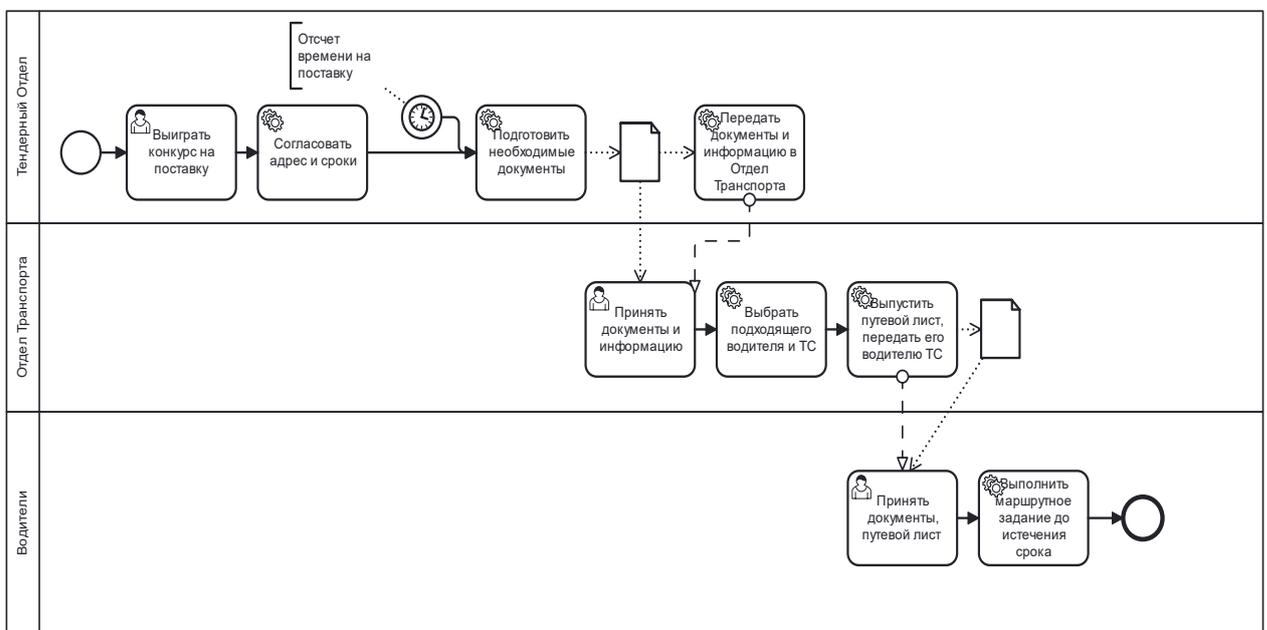


Рисунок 3 – BPMN-диаграмма бизнес-процесса «Как есть»

Тендерные специалисты передают информацию о выигранном тендере или поступившем заказе на поставку продукции в транспортный отдел устно, либо через средства электронной коммуникации. После получения информации, сотрудники транспортного отдела начинают планировать маршрут грузоперевозки. Для этого необходимо выбрать подходящее под задачу транспортное средство и прицеп или цистерну которые будут пристыкованы к выбранному транспорту. Далее необходимо осуществить выбор водителя для командировки. После этого составляется путевой лист, содержащий всю подготовленную информацию о рейсе, и передается водителю. Путевой лист заполняется вручную с использованием программного обеспечения офисного пакета MS Office.

1.3.2 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

Современные требования к эффективности бизнес-процессов диктуют необходимость внедрения автоматизированного решения для управления транспортными процессами в компании. Бизнес стремится к оптимизации операций и сокращению издержек, в то время как отсутствие информационной платформы накладывает ограничения на работу транспортных специалистов и взаимодействие между отделами компании [20].

Уведомление транспортного отдела о выигранных тендерах или поступивших заказах через устные сообщения или электронные каналы связи, является первым шагом в организации логистических процессов. Такой вид обмена сообщениями является достаточно оперативным, но в связи с человеческим фактором, существуют риски утери части важной информации при передаче. Дальнейшее планирование маршрутов, выбор транспортных средств и водителей, а также составление путевых листов вручную также сопряжены с рядом недостатков.

Этот процесс, требует значительных временных и трудовых затрат. Ручной ввод данных увеличивает вероятность совершения ошибок, а также создает необходимость в дополнительном контроле и корректировках. Отсутствие централизованной информационной платформы не позволяет проводить эффективный мониторинг и координацию работы транспортного отдела с другими подразделениями компании.

Разработка и внедрение АИС для управления транспортными процессами призваны стать важным элементом оптимизации бизнес-процессов организации. Информационная система позволит уменьшить затрачиваемое время на планирование маршрутов и снизить вероятность ошибок и задержек. Успешное внедрение АИС должно повысить эффективность работы транспортного отдела, а также компании в целом.

Требования к новой технологии включают в себя возможность автоматизации процессов заполнения путевых листов, интеграцию с другими информационными системами компании для обмена данными, а также гибкость и масштабируемость, необходимые для адаптации к изменяющимся потребностям бизнеса [3].

Внедрение системы управления транспортными процессами призвано стать важным шагом, направленным на повышение конкурентоспособности компании и обеспечение более качественного взаимодействия с потребителем.

1.4 Анализ существующих разработок на предмет соответствия сформулированным требованиям

1.4.1 Определение критериев анализа

Определение критериев анализа является ключевым этапом для грамотного формулирования оценки существующих программных решений и совершения выбора, отвечающего запросам компании. Ниже приведены основные критерии, которые будут использоваться в анализе.

Функциональность. Оценка по этому критерию позволит понять, насколько разработка покрывает все основные потребности компании в управлении транспортными процессами.

Интеграция. Важно оценить возможность интеграции разрабатываемой системы с уже внедренными и функционирующими в компании информационными системами.

Гибкость и настройка. Массовый продукт, как правило, разрабатывается исходя из усредненных требований множества потребителей. В связи с этим, необходимо оценить, насколько существующие решения отвечают требованиям описываемой организации с точки зрения настройки под специфические нужды.

Стоимость. Оценка затрат играет не менее важную роль при принятии решения. Необходимо учесть начальные расходы на приобретение и настройку ПО и дальнейшие траты на обновления, техническую поддержку и обучение персонала. Сравнение стоимости различных программных продуктов поможет выбрать наиболее экономически выгодное решение.

На основе критериев, описанных выше, можно провести глубокий анализ существующих решений на предмет их соответствия требованиям компании и выбрать наиболее подходящее решение [2].

1.4.2 Сравнительная характеристика существующих разработок

На российском рынке присутствует несколько готовых программных решений, направленных на управление транспортными процессами.

Одним из решений является программа «ПРО.Путевые листы», предназначенная для формирования и заполнения путевых листов на различные виды транспортных средств и грузов. Согласно описанию, данное решение ориентировано на широкий круг пользователей – от коммерческих организаций и индивидуальных предпринимателей до государственных учреждений.

Основной функционал программы сосредоточен на автоматизации процесса работы с путевыми листами. Пользователям предоставляется возможность добавления информации о водителях и транспортных средствах в базу данных. Впоследствии, внесенная информация может быть использована для заполнения готовых шаблонов путевых листов. Также, система обеспечивает возможность одновременной работы с базой данных нескольких пользователей в локальной сети, что может быть полезно транспортному отделу «Профиль-Снаб». Интерфейс ПО изображен на рисунке 4.

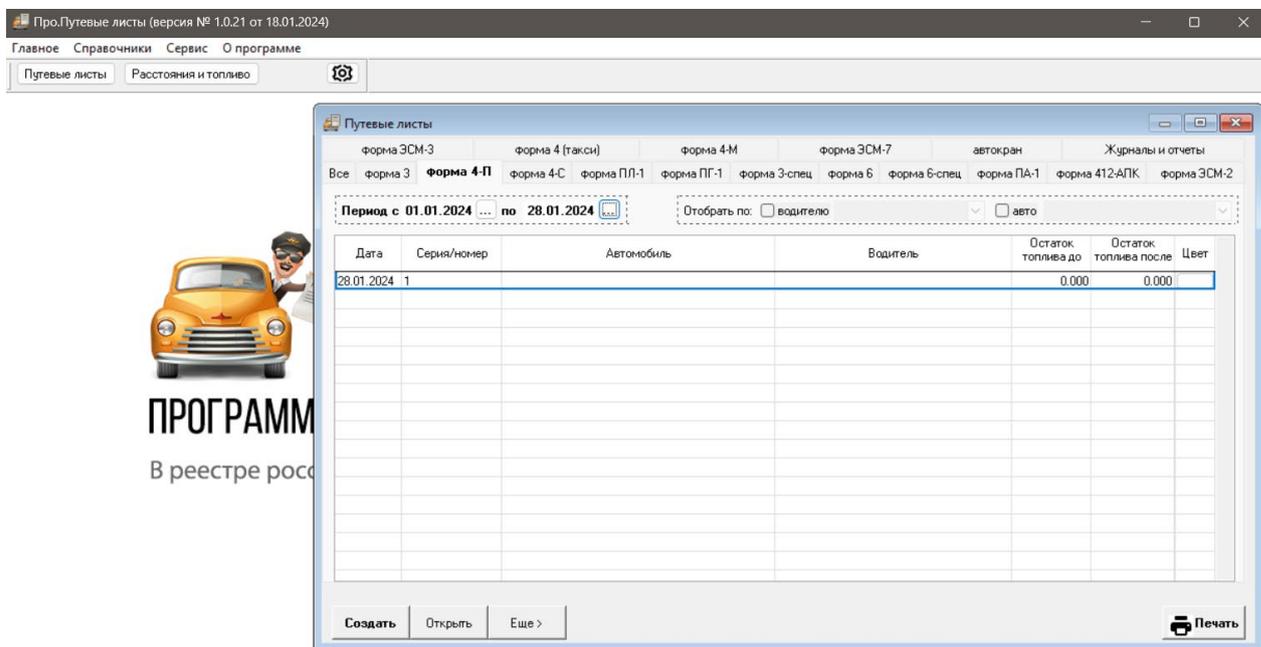


Рисунок 4 – Скриншот программы «ПРО.Путевые листы»

Существенным недостатком программы является ее работа в автономном режиме. Данное ПО не предусматривает интеграции с внешними системами. Это ограничивает ее функциональные возможности и не позволяет полностью удовлетворить специфические запросы компании в контексте сложных бизнес-процессов. Кроме того, ограниченное количество настроек

делает систему менее гибкой и малоприспособленной к оперативной адаптации под изменяющиеся потребности организации.

Стоимость данного автоматизированного решения составляет 5000 рублей в год за одно рабочее место.

Программа «Путевые листы для 1С» от компании «Победасофт» представляет собой еще одну АИС, направленную на автоматизацию процесса заполнения путевых листов. В сравнении с ранее описанным решением можно отметить схожесть в основной логике работы. Однако, функционал данной программы охватывает более широкий спектр возможностей. Скриншот ее интерфейса продемонстрирован на рисунке 5.

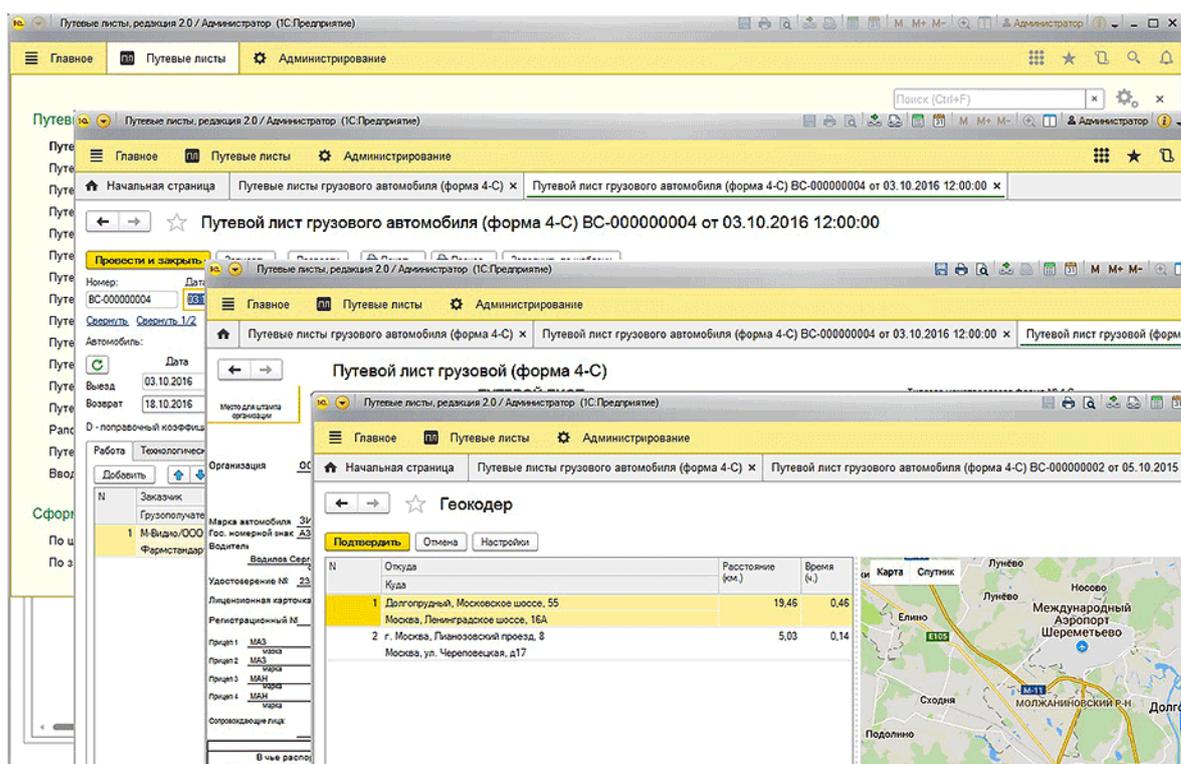


Рисунок 5 – Скриншот программы «Путевые листы для 1С» от компании «Победасофт»

Важным аспектом данного решения, согласно описанию на официальном сайте разработчика, является возможность автоматически вносить изменения в шаблоны форм в случае изменения законодательства. Эта

функция обеспечивает актуальность документации и ее соответствие требованиям законодательства. Интеграция с картографическим сервисом позволяет данному ПО осуществлять расчет топлива на поездку, что позволяет оптимизировать учет и планирование затрат на топливо.

Помимо прочего, данная АИС предлагает ряд дополнительного функционала, например, автоматическое заполнение пробега с использованием онлайн-сервиса или сохраненных данных, учет расхода топлива и заправок с различными опциями, контроль сроков замены расходных материалов транспорта и расчет зарплаты водителей с учетом различных параметров.

Важным преимуществом данного решения является то, что оно основано на платформе 1С: Предприятие. За счет этого, возможна бесшовная интеграция данного продукта в работу компании. Расчет заработных плат водителей легко переносится в зарплатные ведомости, затраты на ТО и ОСАГО транспорта также отражаются в расходах компании автоматически. Поддержка импорта справочника контрагентов организации также упрощает процесс интеграции и обмена данными.

Стоимость единоразового приобретения лицензии на программу составляет 18500 рублей, при этом, число пользователей в локальной сети не ограничено. Учитывая более широкий функционал и возможности настройки, этот продукт представляет собой более гибкое и масштабируемое решение по сравнению с программой «ПРО.Путевые листы».

1.5 Постановка задачи на разработку проекта создания ИС

Для автоматизации операций в транспортном отделе и организации коммуникации связи с другими отделами организации ООО ТД «Профиль-Снаб», требуется разработать автоматизированную информационную систему (АИС), которая будет соответствовать описанным ниже функциональным критериям [13].

Формирование и управление базой данных водителей компании, включая возможность добавления, удаления и редактирования каждой записи. Каждая запись должна содержать необходимую нормативную информацию о водителе.

Формирование и управление базой данных транспортных средств и прицепов (цистерн) компании, с аналогичной функциональностью по добавлению, удалению и редактированию записей.

Формирование и управление базой заказов продукции позволяющей наладить коммуникацию между отделами при помощи АИС.

Поддержка интеграции с внешними информационными системами либо возможность ручного добавления реквизитов контрагентов в базу данных.

Автоматическое создание и сохранение путевых листов на рабочем месте пользователя согласно типовой форме № 4-С межотраслевого стандарта на основе выбора данных из базы данных для заполнения соответствующих полей.

Возможность размещения в сетевой среде для обеспечения многопользовательского доступа.

Реализация авторизации пользователей с разными уровнями прав доступа к редактированию БД.

В качестве документации программного обеспечения необходима разработка руководств пользователя и системного администратора.

2 Логическое проектирование АИС

2.1 Разработка модели бизнес-процесса «Как будет»

Внедрение автоматизированной информационной системы (АИС) в работу тендерного и транспортного отделов организации призвано ощутимо ускорить процесс горизонтальной коммуникации и оптимизировать бизнес-процессы компании. На рисунке 6 изображена подробная BPMN-диаграмма бизнес-процесса «Как будет» [21].

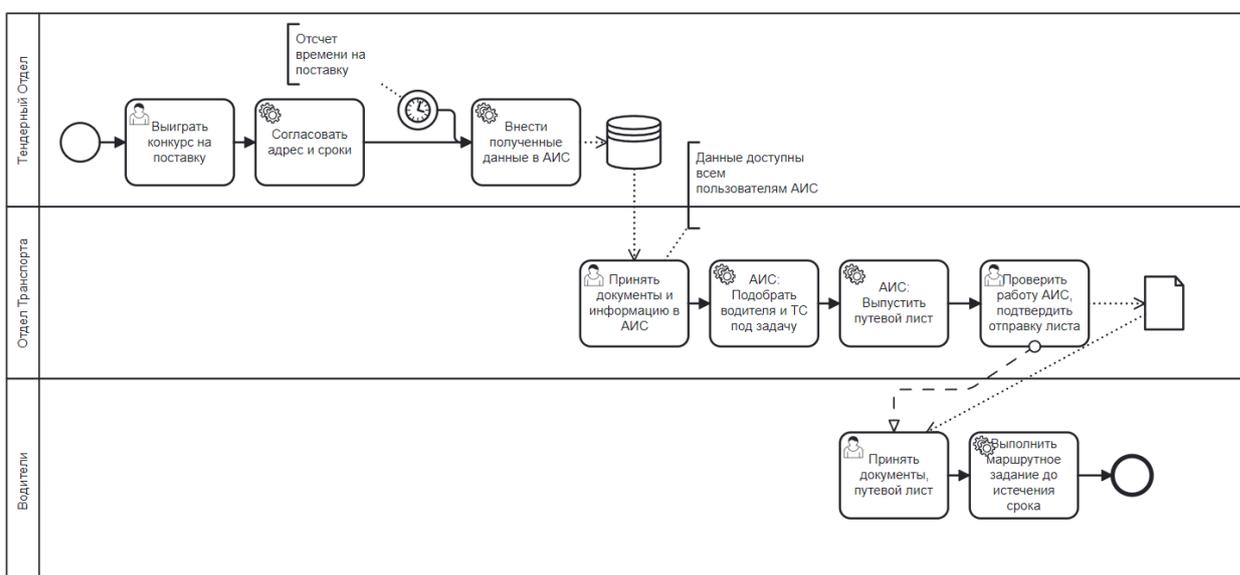


Рисунок 6 – BPMN-диаграмма бизнес-процесса «Как будет»

Вместо того, чтобы уведомлять сотрудников отдела транспорта о полученных заказах лично, тендерные специалисты смогут вносить записи в базу данных АИС. Это позволит отразить полученную информацию для всех участников процесса, у которых есть доступ к БД.

Транспортный отдел приступит к планированию рейса, как только информация о новом заказе появится в базе. На основе записей о доступных водителях и транспорте можно будет быстро осуществить рациональный выбор подходящего сочетания в зависимости от условий и требований.

После выбора водителя и ТС, программа сгенерирует путевой лист. Соответствующие данные будут извлекаться из базы данных и вноситься в документ. Автоматизация процесса заполнения листа сведет к минимуму риск совершения ошибок, связанных с человеческим фактором.

Результатом внедрения АИС в работу компании, должно стать повышение эффективности выполнения регулярных операционных задач, выполняемых сотрудниками транспортного отдела. Благодаря автоматизации операционной деятельности, они смогут уделять больше времени разработке новых маршрутов и оптимизации перевозок. Взаимодействие с сотрудниками тендерного отдела станет проще и прозрачнее. Вся необходимая информация будет передаваться в срок в полном объеме. Цифровизация совместной работы двух отделов повысит общую эффективность работы компании. Это положительно скажется на оптимизации всех бизнес-процессов в организации.

2.2 Выбор технологии логического моделирования АИС

Основой процесса разработки АИС является построение подробной логической модели. Основываясь на контекстной диаграмме «Как будет», такая модель представляет первичные сущности системы наглядно.

Логическая модель помогает сосредоточиться на основных компонентах при разработке системы. Она не учитывает второстепенные функциональные возможности. Рациональный подход к анализу и исследованию необходим для декомпозиции комплексных целей на конкретные задачи, тем самым обеспечивая логичный и структурированный процесс принятия решений. Эффективное моделирование позволяет команде разработчиков сформировать целостную картину проекта и наметить дорожную карту его реализации. Заранее определяя основные сущности и их взаимодействие в системе, разработчик рационализирует реализацию проекта и минимизирует риск ошибок.

Одним из самых распространенных языков логического моделирования является UML – полезный унифицированный инструмент для системного анализа, проектирования и визуализации. UML предоставляет широкий спектр диаграмм, облегчая графическое представление различных аспектов системы. Благодаря своей простоте и гибкости он стал стандартом моделирования программного обеспечения, способствующим эффективному общению между членами команды разработчиков [1].

Для создания графических моделей разрабатываемой системы используется веб-версия сервиса draw.io. Она поддерживает все основные типы диаграмм, предоставляя пользователю возможность свободного моделирования. Ниже, на рисунке 7, представлена диаграмма прецедентов (Use Case) разрабатываемой системы.

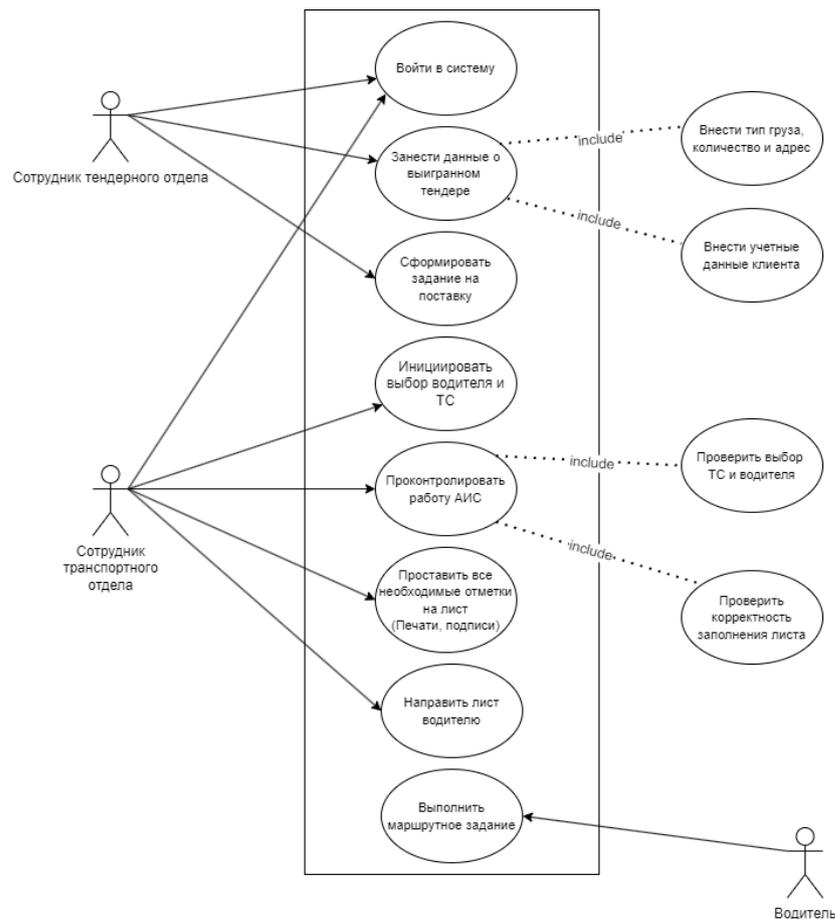


Рисунок 7 – Диаграмма прецедентов

Кроме того, при разработке системы используются диаграммы потоков данных (DFD). Они применяются для моделирования информационных систем с точки зрения хранения, обработки и передачи данных. Определение ключевых компонентов системы и их взаимосвязей является важным этапом процесса оптимизации и повышения общей функциональности системы. Таким образом, использование UML и DFD диаграмм позволяет эффективно моделировать сложные системы.

Логическая модель является неотъемлемым компонентом процесса разработки. Применяя логическое моделирование, можно успешно визуализировать основной функционал системы на ранних этапах разработки, избегая ненужных сложностей при ее разработке и развертывании [16].

2.3 Информационное обеспечение АИС

2.3.1 Используемые классификаторы и системы кодирования

Автоматизация бизнес-процессов неразрывно связана с использованием классификаторов и систем кодирования. Единая система номенклатуры товаров и услуг, созданная с их помощью, упрощает многие операции в компании. Стандартизация играет важную роль в согласовании продукции и услуг организации с запросами клиентов и требованиями поставщиков, обеспечивая совместимость с общепринятыми стандартами.

Использование систем кодирования контрагентов позволяет ускорить документооборот. Применение стандартизированных идентификаторов клиентов и поставщиков обеспечивает быстрый обмен данными и минимизирует риск допущения ошибок.

Классификаторы транспортных средств необходимы для оптимизации логистических операций. После их внедрения, появится возможность автоматизировать подбор транспорта и формирование рейсовой документации.

Классификаторы являются основой разработки автоматизированной системы. Они обеспечивают необходимую стандартизацию и оптимизацию обмена информацией, что, в свою очередь, повышает общую эффективность компании.

2.3.2 Характеристика нормативно-справочной и входной оперативной информации

Основой автоматизированной системы управления является информация, которую можно разделить на слои нормативно-справочной и оперативной. Нормативно-справочная информация включает в себя широкий диапазон данных: каталог продукции, справочник контрагентов, список сотрудников и реестр транспортных средств. Этот слой имеет значение для бесперебойного функционирования компании, так как содержит в себе справочники основной информации, регулярно используемой сотрудниками.

Оперативная информация, напротив, характеризуется динамичностью, непрерывным потоком обновлений, касающихся новых заказов и корректировок в логистике. Данный тип информации предполагает немедленную обработку и оперативное реагирование [6].

В качестве входящей информации выступают запросы на товары и услуги от клиентов, а также, информация о поставляемой продукции от поставщиков. Эта категория информации, так-же как и оперативная, требует немедленной обработки, например, передачу данных о заказе в другой отдел, или дополнение справочника продукции компании.

Эффективность компании напрямую зависит от грамотной обработки всех видов информации. Такой подход создает цикл, гарантирующий стабильность и обеспечивающий бесперебойное и плавное протекание всех бизнес-процессов организации [3].

2.3.3 Характеристика выходной информации

В качестве выходной информации выступает ряд документов и отчетов, являющихся результатом обработки входящей, оперативной и нормативно-справочной информации. Рисунок 8 визуализирует диаграмму потоков данных (DFD).

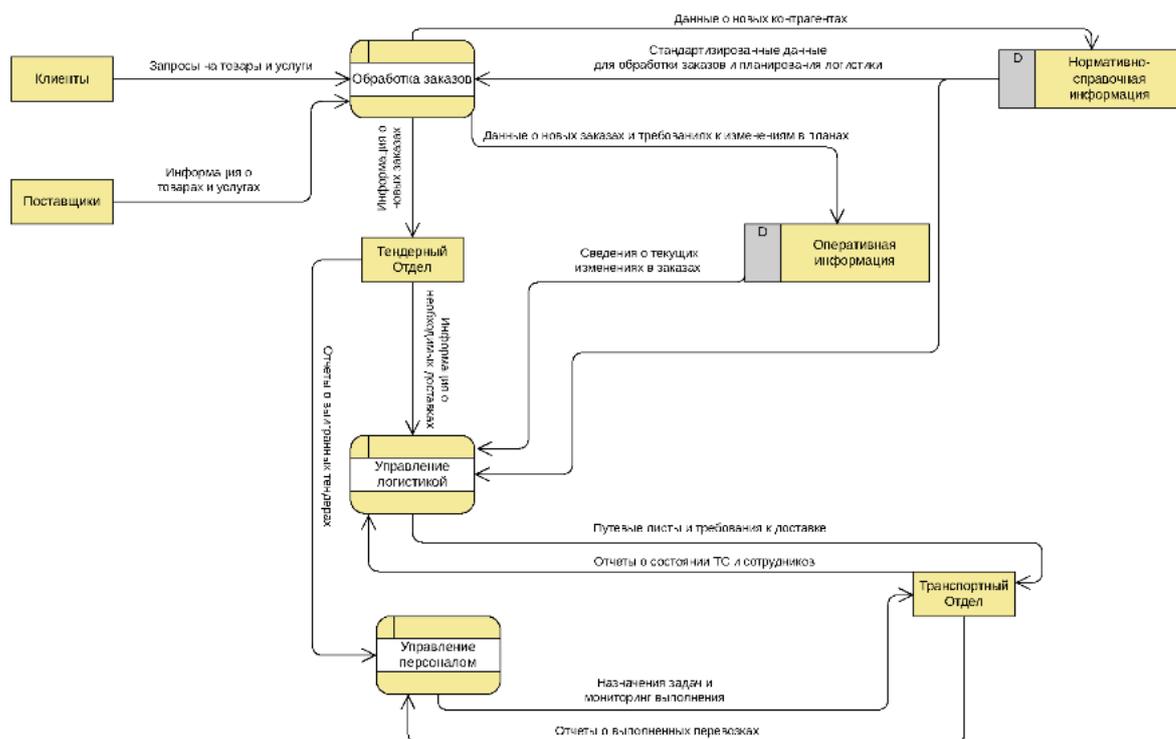


Рисунок 8 – Диаграмма потоков данных

Отчеты предлагают информацию о внутренних процессах работы отделов, эффективности их работы, а также о выполнении планов поставок. На их основе можно провести анализ деятельности каждого вовлеченного отдела и компании в целом. Также, при необходимости, используя информацию, полученную из отчетов, можно скорректировать деятельность отдельных сотрудников или подразделений.

Путевые листы, генерируемые системой при использовании нормативно-справочной информации, позволяют обеспечить выполнение требований законодательства в организации логистических операций. Вся необходимая информация заносится в них из заранее сформированных справочников, что позволяет минимизировать риск допущения ошибок при их формировании и обеспечить стабильную логистическую деятельность без задержек и штрафов.

В качестве еще одного примера выходной информации можно отметить формирование табличного представления данных из справочников организации, например, возможность по запросу ознакомиться со списком продукции компании.

Интеграция классификаторов и автоматизация обработки информации способствуют значительному повышению эффективности управления организацией и оперативности реагирования на изменения [22].

2.4 Проектирование базы данных АИС

2.4.1 Выбор технологии проектирования БД АИС

Перед началом разработки модели базы данных был проведен тщательный анализ с целью выбора наиболее подходящей технологии проектирования. В качестве языка концептуального и логического моделирования, как и ранее, был выбран UML. Для визуализации моделей использовано ПО draw.io [29].

В качестве вида базы данных была выбрана реляционная модель благодаря ее проверенной надежности, гибкости и масштабируемости. Реляционные базы данных обладают множеством сильных сторон. Одно из главных – целостность данных. Реляционные СУБД используют различные ограничения, такие как первичные ключи и внешние ключи, позволяющие обеспечить целостность данных. При постоянном обновлении информации о

продукции и заказах, поддержание целостности является важнейшим преимуществом [5].

Другим, не менее важным, качеством является масштабируемость реляционной модели. Реляционные СУБД могут обрабатывать большие объемы данных, а также, поддерживают одновременную работу нескольких пользователей, что может быть полезно в контексте разработки описываемой АИС. Возможность горизонтального масштабирования особенно ценна для эффективного управления растущими объемами данных [8].

Для реализации проекта была выбрана СУБД PostgreSQL. Она предлагает широкий спектр функциональных возможностей и обеспечивает поддержку реляционной модели данных. Этот выбор был обусловлен стремлением создать надежную, эффективную и гибко настраиваемую систему. Она должна адаптироваться к изменяющимся требованиям бизнеса и технологическому прогрессу, позволяя компании улучшить управление данными и оптимизировать бизнес-процессы. Еще одним преимуществом данной СУБД является возможность интеграции с различными сценариями использования. PostgreSQL отличается высокой степенью совместимости со стандартами SQL и обширным набором функциональных возможностей, что обеспечивает гибкость в проектировании базы данных и реализации бизнес-логики [17].

2.4.2 Разработка концептуальной модели данных АИС

В рамках разработки концептуальной модели данных, было произведено моделирование основных сущностей системы [1]. Это позволило учесть как существующие на сегодняшний день, так и возможные будущие потребности организации.

Визуализация концептуальной модели представлена на рисунке 9.

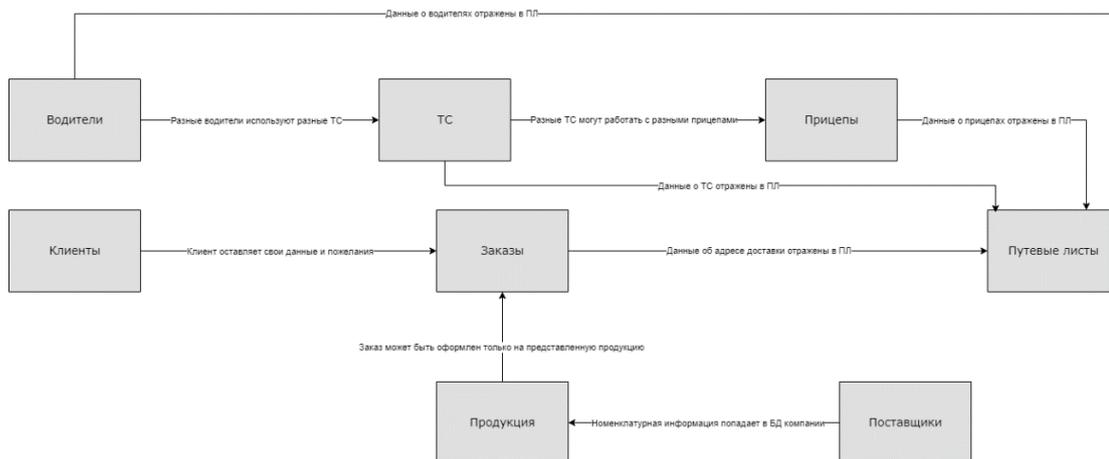


Рисунок 9 – Концептуальная модель БД

Ниже описаны ключевые сущности разрабатываемой БД и их атрибуты.

Сущность «Водители» включает сведения о всех водителях организации, охватывая такие атрибуты, как учетный номер водителя, ФИО, номер водительского удостоверения, дата выдачи ВУ, контактные данные и СНИЛС.

Сущность «Транспортные средства» описывает все ТС, которыми располагает организация. Атрибутами являются их идентификатор, марка, государственный номер и год выпуска.

Сущность «Прицепы» содержит информацию о прицепах и цистернах, которые могут работать в паре с транспортным средством. Содержащиеся атрибуты: идентификатор, марка и государственный номер.

Сущность «Заказы» содержит детализированные данные о заказах на поставку продукции, включая информацию об идентификаторе заказа, информацию о клиенте, дате доставки и адресе.

Сущность «Путевые листы» содержит учетный номер путевого листа и ссылается на сущности «Водители», «Транспортные средства» и «Заказы». Путевые листы генерируются автоматически и наполняются информацией, хранящейся в других таблицах БД

Сущности «Поставщики» и «Клиенты» представляют собой элементы модели, включающие данные о поставщиках и клиентах компании, такие как идентификаторы, названия, учетные данные, адреса и контактная информация.

Сущность «Продукция» описывает характеристики продукции, такие как идентификатор, название, агрегатное состояние и класс опасности.

2.4.3 Разработка логической модели данных АИС

На основе разработанной концептуальной модели была создана логическая модель данных. Для каждой из рассмотренных сущностей были определены ключевые атрибуты, а также установлены связи между таблицами базы данных:

- таблица «Водители» (Drivers) включает атрибуты идентификатора водителя (id), ФИО (full_name), номер водительского удостоверения (license_number), дату выдачи ВУ (vu_issue_date) контактные данные (contact_details) и СНИЛС (snils);
- таблица «Транспортные средства» (Vehicles) содержит идентификатор транспортного средства (id), марку (brand), государственный номер (license_plate) и год выпуска (year_of_manufacture);
- таблица «Прицепы» содержит идентификатор прицепа (trailer_id), его название (trailer) и регистрационный номер (trailer_plate);
- таблица «Заказы» (Orders) описывает каждый заказ, включая идентификатор заказа (id), информацию о клиенте (client_info), дату доставки (delivery_date) и адрес (address);
- таблицы «Поставщики» (Suppliers) и «Клиенты» (Clients) содержат сведения о взаимодействующих сторонах, уточняя их идентификаторы (id), названия организаций (name), учетные данные

- (account_info), адреса (address) и контактную информацию (contact_info);
- таблица «Продукция» (Products) детализирует характеристики продукции, такие как идентификатор (id), название (name), агрегатное состояние (state) и класс опасности (hazard_class);
 - формируемый файл путевого листа (Waybill) является сущностью, наполняемой данными о каждом маршрутном задании, ссылаясь на записи из других таблиц при использовании запросов к БД и включает в себя ФИО водителя (full_name), номер водительского удостоверения (license_number), дату выдачи ВУ (vu_issue_date), адрес доставки (address), марку ТС (brand), его государственный номер (license_plate), а также название (trailer) и регистрационный номер (trailer_plate) прицепа.

При логическом проектировании базы данных были реализованы ограничения на уровне таблиц, включающие первичные ключи, обеспечивающие точность записей и надежность их хранения. Использование первичных ключей позволяет однозначно идентифицировать каждую запись в таблице [6]. Визуализация ERD диаграммы логической модели БД представлена на рисунке 10.

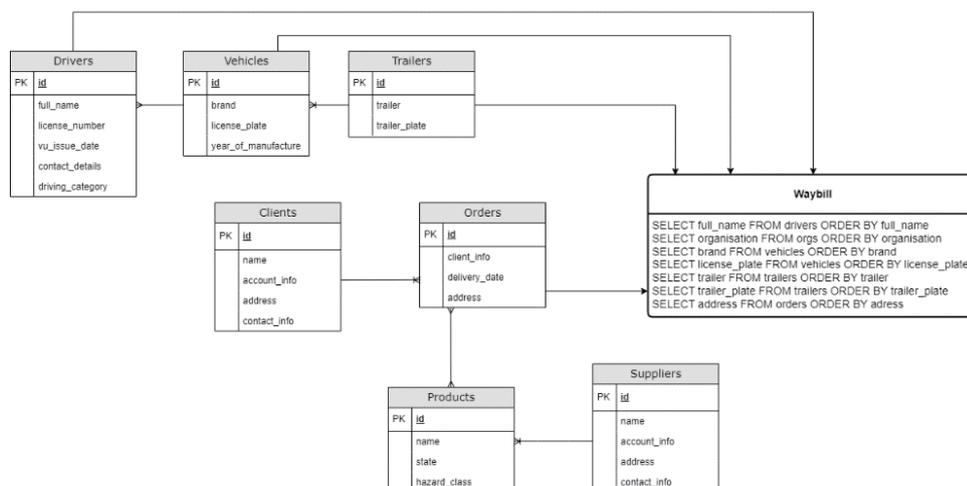


Рисунок 10 – Логическая модель БД

Использование диаграмм зависимостей сущностей (ERD) для визуализации логической модели позволяет наглядно представить структуру разрабатываемой базы данных, а также сложные взаимосвязи между ее ключевыми компонентами.

Методический подход к проектированию БД значительно улучшает понимание процессов управления данными. Он способствует созданию эффективной и масштабируемой информационной системы. Проектирование архитектуры базы данных, поддерживающей строгие требования к целостности данных и реляционному отображению, закладывает устойчивую платформу для создания информационной системы, обладающей широким функционалом, оставаясь, при этом, устойчивой в эксплуатации [16].

2.5 Требования к аппаратно-программному обеспечению АИС

Для успешного внедрения и эффективной эксплуатации АИС в работе организации, ИТ-инфраструктура компании должна соответствовать предполагаемым системным требованиям. Они разработаны с учетом необходимости обеспечения высокой производительности и надежности системы.

Аппаратные требования:

а) сервер:

- 1) процессор, не менее 4 ядер, 1.8 ГГц или выше;
- 2) оперативная память, не менее 8 ГБ;
- 3) свободное место на жестком диске, не менее 1 ГБ.

б) рабочие станции пользователей:

- 1) процессор, не менее 2 ядер, 1.5 ГГц или выше;
- 2) оперативная память, не менее 4 ГБ;
- 3) свободное место на жестком диске, не менее 500 МБ.

Программные требования:

а) операционная система:

1) сервер и рабочие станции на базе Windows, соответствующие вышеуказанным версиям.

б) система управления базами данных (СУБД):

1) PostgreSQL 16.2, совместимая с операционной системой Windows;

в) интерпретатор Python.

1) Python 3.12.2, последняя стабильная версия, совместимая с операционной системой Windows.

г) программное обеспечение и библиотеки:

1) набор библиотек Python для работы с PostgreSQL, а также другие библиотеки, требуемые для реализации функционала разрабатываемой системы.

Требования к сетевой инфраструктуре:

- для обеспечения стабильной многопользовательской работы АИС с базой данных, размещенной на сервере, необходима локальная сеть с минимальной скоростью соединения 100 Мбит/с;
- VPN соединение для пользователей системы, находящихся вне пределов локальной сети, обеспечивающее доступ клиентского приложения АИС к серверной части.

Выполнение этих требований обеспечит возможность комфортного внедрения и эксплуатации программного продукта в организации. На сегодняшний день, данные требования можно назвать низкими и выполнимыми практически в любой ИТ-инфраструктуре без необходимости обновления аппаратной части.

Реализация проекта разработки АИС для ООО ТД «Профиль-Снаб» характеризуется комплексным характером, направленным на оптимизацию и автоматизацию основных бизнес-процессов организации. Особое внимание было уделено рабочим процессам транспортного и тендерного отделов.

Внедрение программного решение обусловлено необходимостью повысить эффективность работы сотрудников за счет автоматизации регулярных операций, таких как учет новых заказов и результатов тендеров, а также формирование путевых листов. В результате цифровизации этих процессов планируется достичь повышения уровня доступности и централизации информационных потоков.

Выбор технологических решений для создания разработки АИС основан на углубленном изучении потребностей компании. Решение использовать реляционную модель данных, реализованную на СУБД PostgreSQL, в качестве основного инструмента разработки базы данных системы, продиктовано потребностью в высокой надежности и масштабируемости проекта.

Использование унифицированного языка моделирования UML наряду с DFD диаграммами помогло эффективно спроектировать четкую и структурированную модель системы.

Сформулированные требования к аппаратному и программному обеспечению обеспечат комфортное и стабильное функционирование системы после ее внедрения. Требования выполнены таким образом, что дополнительных затрат на модернизацию ИТ-инфраструктуры компании не потребуется.

3 Физическое проектирование АИС

3.1 Выбор архитектуры АИС

Выбор архитектуры автоматизированной информационной системы – важный этап ее разработки. Для его совершения требуется всесторонняя оценка операционных потребностей организации и совместимости технологических стеков. Архитектура системы должна способствовать ускорению выполнения стандартных операций, обеспечивая при этом централизацию и доступность данных.

Наиболее подходящей моделью оказалась клиент-серверная архитектура, обеспечивающая сбалансированное распределение вычислений без потери возможности централизованного управления данными. Система разделена на два основных компонента: клиентское приложение, содержащее интерфейс пользователя и отображение всей необходимой информации, и серверную составляющую, отвечающую за обработку и хранение данных. Разделение системы на компоненты позволяет независимо дорабатывать и масштабировать каждый из них по отдельности, в зависимости от возникающих потребностей компании. Визуализация архитектуры системы отражена в виде диаграммы пакетов (Рисунок 11).

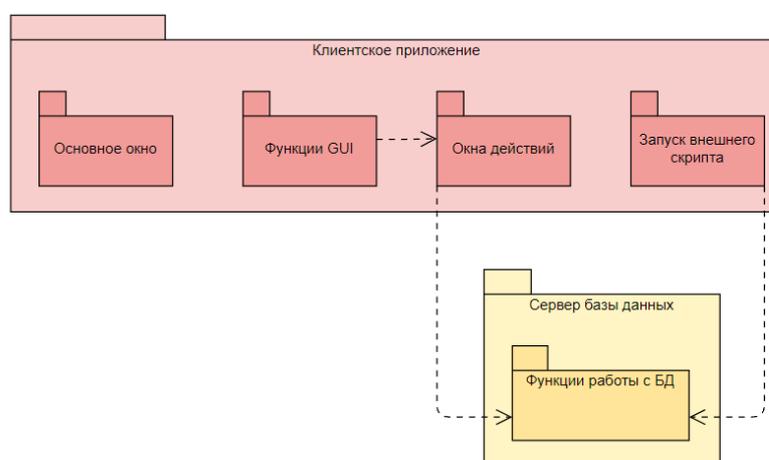


Рисунок 11. Диаграмма пакетов

Адаптивность клиент-серверной архитектуры обеспечивает эффективное развитие системы [2]. По мере роста компании, серверная часть может быть расширена для удовлетворения растущего числа запросов, а новые клиентские приложения, использующие доступ к той же БД, могут быть разработаны и внедрены в работу компании без существенных изменений на стороне сервера.

В качестве серверного решения была выбрана СУБД PostgreSQL. Благодаря ее стабильности, невысоким системным требованиям и бесплатной модели распространения, она стала одним из наиболее эффективных решений для организации SQL-сервера. Совместимость PostgreSQL с различными типами данных, широкие возможности контроля многопользовательской работы и комплексные настройки безопасности делают ее хорошим выбором в контексте управления разнообразными типами данных и сложными запросами.

Клиентское приложение использует монолитную архитектуру графического интерфейса пользователя (GUI. Такой подход эффективен для небольших приложений, где преимущества монолитной структуры, такие как простота развертывания и отладки, перевешивают сложности доработки и внедрения нового функционала, которых можно избежать при использовании более распределенных архитектур. Графический интерфейс разработан простым и интуитивно понятным, что позволит сотрудникам эффективно выполнять рабочие задачи, например, учет результатов тендеров и формирование маршрутных листов, без необходимости длительного обучения.

3.2 Выбор технологии разработки программного обеспечения АИС

В процессе выбора технологии разработки АИС был проведен комплексный сравнительный анализ. Его целью ставилось выявление наиболее подходящих для данного проекта языка программирования и

интегрированной среды разработки (IDE). В ходе анализа были рассмотрены как платные, так и свободно распространяемые IDE по нескольким ключевым критериям, включая стоимость, поддержку сообщества, гибкость и простоту интеграции с системами баз данных.

В таблице 1 представлен краткий сравнительный анализ различных языков программирования и IDE.

Таблица 1 – Сравнительный анализ языков программирования и IDE

Критерий	Язык программирования и IDE			
	Python с IDE от JetBrains	Java с Eclipse	C# с Visual Studio	JavaScript с Visual Studio Code
Стоимость	Бесплатная и платная версии	Бесплатно	Бесплатная и платная версии	Бесплатно
Поддержка сообщества	Обширная	Обширная	Обширная	Обширная
Гибкость	Высокая	Высокая	Средняя	Высокая
Удобство интеграции с БД	Высокое	Высокое	Высокое	Среднее
Сложность освоения	Низкая	Высокая	Средняя	Средняя

Согласно результатам проведенного анализа, язык программирования Python в сочетании с IDE PyCharm от JetBrains представляется оптимальным выбором для разработки описываемой АИС по нескольким причинам. Первая заключается в том, что синтаксис Python заметно проще и читабельнее, чем у Java и C# [10]. Это значительно снижает затрачиваемое время на обучение начинающих разработчиков и повышает скорость выполнения задачи. Во-вторых, обширный набор Python-библиотек обеспечивает вариативность при создании приложений. Кроме того, возможность интеграции приложения, написанного на Python с PostgreSQL упрощает процесс разработки и поддержки функций приложения, связанных с базами данных [27].

IDE PyCharm от команды JetBrains раскрывает сильные стороны Python, предоставляя качественную среду, специально разработанную для работы с

Python. Полезные функции, такие, как интеллектуальное завершение кода, его анализ и поддержка загрузки плагинов значительно повышают производительность и качество кода. Бесплатной версии PyCharm достаточно для разработки практически любого проекта, а платная версия предлагает разнообразные дополнительные возможности, способные еще больше повысить эффективность работы [15].

Java, C# и JavaScript также являются мощными языками программирования, поддерживаемыми обширным сообществом разработчиков, но каждый из них имеет некоторые ограничения, которые делают их менее подходящими для данного проекта. Например, Java и C# гораздо более сложны в освоении и, как правило, более нагружены, чем Python. JavaScript, несмотря на высокую гибкость и широкое применение в веб-разработке, не обладает необходимым уровнем поддержки интеграции баз данных, как Python.

3.3 Разработка физической модели данных АИС

В процессе разработки системы было важно уделить особое внимание проектированию физической модели данных. Она позволит реализовать логическую модель, разработанную ранее, в выбранной СУБД. Визуализация физической модели представлена в виде диаграммы зависимостей сущностей (Рисунок 12) [11].

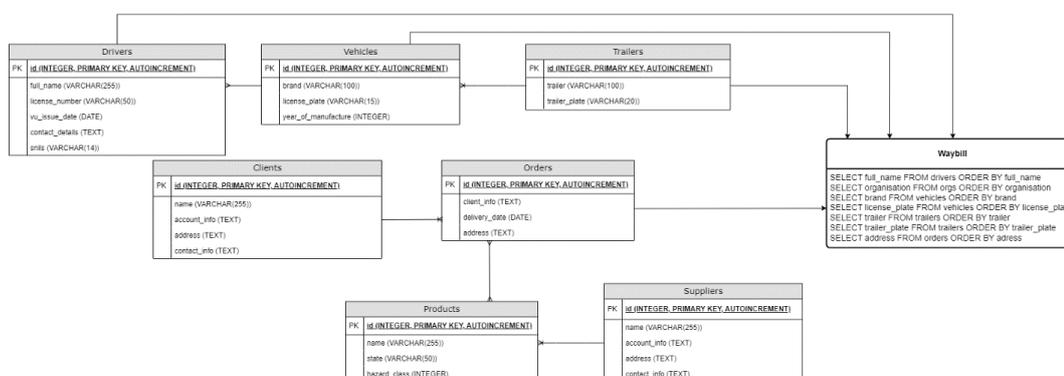


Рисунок 12 – ER-диаграмма физической модели данных АИС

Задача, решаемая при помощи использования физической модели данных, заключается в определении наименований таблиц и настройке их структуры для оптимального хранения и обработки информации.

Таблица «Drivers» содержит ФИО (name) водителя в формате VARCHAR длиной до 255 символов. Также в ней содержатся номер водительского удостоверения (license_number) и дата его выдачи (vu_issue_date), записанные в форматах VARCHAR и DATE соответственно. По этим полям возможно выполнять операции сравнения и сортировки. Контактные данные (contact_details) и номер СНИЛС (snils), хранятся в текстовом формате.

Таблица «Vehicles» содержит информацию о транспортных средствах. Марка (brand) и номер транспортного средства (license_plate) записаны в строковом формате VARCHAR для удобства поиска и фильтрации, а год выпуска (year_of_manufacture) хранится в формате INTEGER, что позволит внедрить сортировку транспорта по его возрасту.

Для хранения данных о прицепах используется таблица «Trailers», в которой запись о каждом прицепе содержит название (trailer) и регистрационный номер (trailer_plate) в формате VARCHAR.

Таблица «Orders» заполняется информацией о поступивших заказах. Столбцы таблицы включают информацию о клиенте (client_info), дату доставки (delivery_date) и адрес (address) в форматах DATE и TEXT, соответственно.

В таблицах «Suppliers» и «Clients» содержатся данные о контрагентах, в частности, названия организаций (name), учетные данные (account_info), адреса (address), а также, контактная информация (contact_info), хранящиеся в форматах VARCHAR и TEXT.

Таблица «Products» включает данные о продуктах. Столбцы «Название» (name), «Агрегатное состояние» (state) и «Класс опасности» (hazard_class), предоставляют подробную информацию о поставляемой продукции, ее характеристиках, условиях хранения и транспортировки.

Связи между таблицами реализованы через использование внешних ключей (id). Использование этого инструмента позволяет поддерживать целостность данных, а также отображать именно ту информацию, которая была запрошена. Индексация ключевых полей, в частности, номеров водительских удостоверений, ускоряет поиск в большом массиве данных, что является важным фактором быстродействия системы. Поддержка транзакционности в PostgreSQL гарантирует надежное управление данными и высокий уровень согласованности операций, обеспечивающие защиту структуры БД.

3.4 Разработка программного обеспечения АИС

3.4.1 Схема взаимосвязи модулей приложения АИС

В процессе создания АИС, важное значение имеет грамотно построенная схема связей между ее компонентами. Корректную работу системы обеспечивают различные модули, каждый из которых отвечает за свою область функциональности. Основная задача состояла в обеспечении максимальной эффективности взаимодействия между ними, при использовании языка Python совместно с библиотекой tkinter, отвечающей за формирование пользовательского интерфейса [15].

Одним из ключевых является модуль управления конфигурацией базы данных. Он обеспечивает загрузку и сохранение параметров подключения к БД, хранящихся в локальном файле формата JSON. Его использование позволяет динамично адаптировать параметры подключения к базе данных без необходимости повторной компиляции приложения. Функции `load_db_config` и `save_db_config` обеспечивают гибкость и адаптивность системы, позволяя изменять настройки подключения в случае изменения сетевой инфраструктуры или пользователя системы.

Еще один важный элемент системы – модуль интерфейса пользователя. В описываемой системе он разработан при использовании библиотеки tkinter.

Этот модуль отвечает за формирование и управление графическим интерфейсом, предоставляя интуитивно понятные инструменты взаимодействия с АИС. Функции для инициализации основного окна приложения, создания и настройки контекстных меню, форм ввода данных и кнопок облегчают процесс управления информацией, защищая систему от ошибок со стороны пользователя [23].

Центральным элементом системы является модуль управления данными, связывающий пользовательский интерфейс с сервером базы данных. Использование библиотеки `psycopg2` в основе этого модуля позволяет реализовать функции подключения к базе данных и CRUD-операции (создание, чтение, обновление и удаление данных) [24].

Реализация функционала генерации путевых листов выполняется при использовании разработанного внешнего скрипта `waylist2.py`. Он формирует документы в формате Excel, и заполняет их информацией из базы данных, тем самым автоматизируя процесс подготовки необходимой логистической документации. Использование библиотеки `openpyxl` обеспечивает эффективное взаимодействие с содержимым файлов формата `.xlsx` [24].

Взаимодействие между модулями обеспечивает высокий уровень интеграции и согласованности операций. Модуль управления конфигурацией БД непосредственно взаимодействует с модулем управления данными, предоставляя актуальные данные для подключения. Интерфейсный модуль, в свою очередь, постоянно использует методы модуля управления данными для реализации функционала CRUD-операций.

Множественные связи внутри структуры АИС повышают эффективность работы клиентского приложения и обеспечивает удобство эксплуатации системы, максимально автоматизируя процессы управления логистикой. Описываемая система представляет собой комплексное решение, призванное повысить эффективность управления ключевыми бизнес-процессами организации «Профиль-Снаб».

3.4.2 Описание модулей приложения АИС

Каждый модуль клиентского приложения играет важную роль в обеспечении бесперебойного и эффективного функционирования системы. Их подробное описание позволит лучше понять функционал и логику работы каждого отдельного элемента и всего приложения целиком.

Модуль управления конфигурацией БД отвечает за работу с параметрами подключения к базе данных. Данные конфигурации хранятся в локальном JSON-файле, что обеспечивает удобство их редактирования и обновления.

Основной функцией в этом модуле является `load_db_config`. Она проверяет наличие файла конфигурации по определенному в коде программы пути и загружает его содержимое. При условии обнаружения необходимого файла, функция считывает его и возвращает данные в необходимом формате. На основе этих данных осуществляется работа всего функционала связанного с взаимодействием приложения и серверной части.

В случае отсутствия файла конфигурации, например, при первом запуске программы, происходит запуск функции `ask_db_config`. Эта функция инициализирует окно для ввода конфигурационных данных. Также, вызов этой функции можно осуществить посредством нажатия кнопки «Сменить БД» в верхней части главного окна приложения, что может быть полезно при необходимости сменить пользователя системы.

Функция `save_db_config` взаимодействует с описанной ранее функцией `ask_db_config` и используется для записи введенных впервые или измененных параметров конфигурации в файл `json` [23]. На рисунке 13 представлен скриншот формы ввода параметров конфигурационных данных.

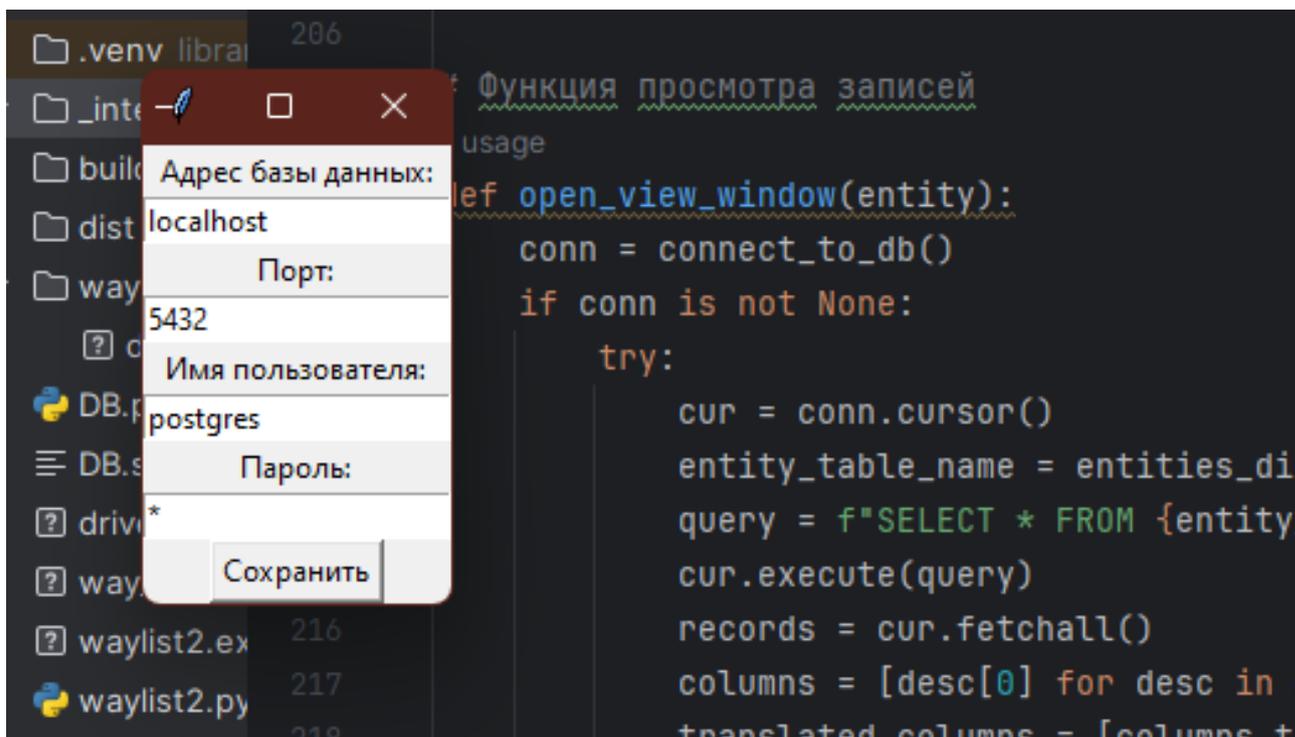


Рисунок 13 – Скриншот окна ввода конфигурации БД

Фундаментальным компонентом информационной системы является модуль интерфейса пользователя, напрямую взаимодействующий с конечными пользователями. В его основу легло использование tkinter – популярной библиотеки для формирования интерфейсов в Python [27]. Его основная задача состоит в обеспечении удобной и интуитивно понятной пользовательской среды для выполнения стандартных операций, таких как ввод данных, их редактирование и просмотр.

Функционал модуля состоит в настройке и отображении графического интерфейса пользователя. С его помощью происходит запуск главного окна приложения, форм взаимодействия с базой данных, а также, контекстных меню для управления формами ввода.

Функция `init_app_window` формирует и отображает главное окно приложения, а также определяет параметры и функционал кнопок. Она отвечает за структуру главного интерфейса. Скриншот главного окна клиентского приложения изображен на рисунке 14.

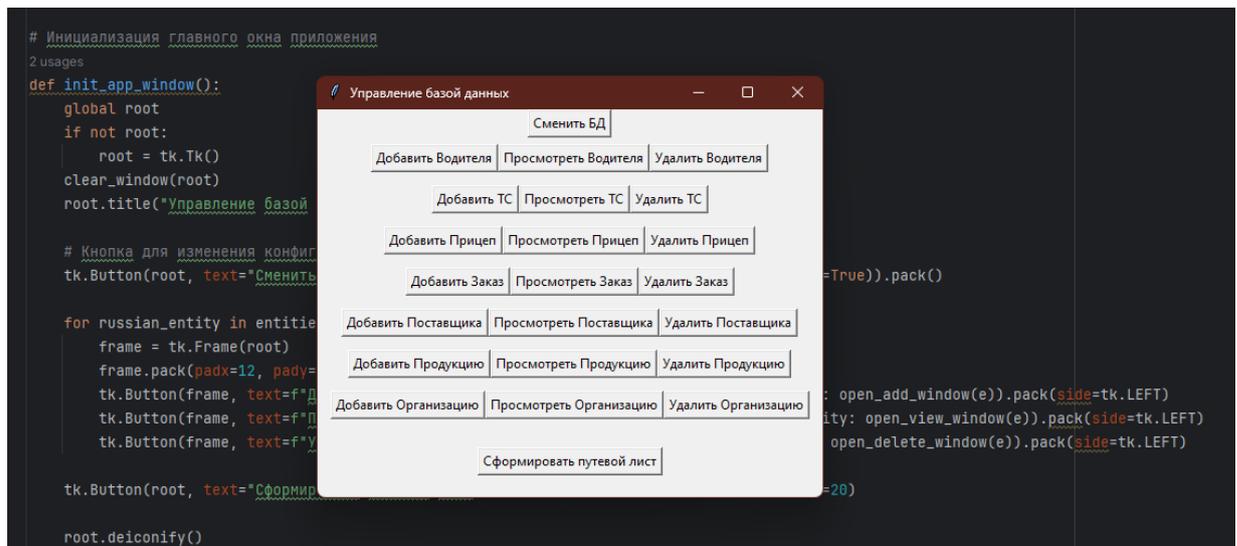


Рисунок 14 – Скриншот главного окна приложения

Другая функция модуля интерфейса пользователя, `apply_context_menu`, позволяет вызывать контекстное меню по нажатию правой кнопки мыши в любом текстовом поле клиентского приложения. Это позволяет использовать стандартные операции копирования, вставки и вырезания текста, значительно упрощая работу с текстовыми данными и создавая более дружелюбный к пользователю интерфейс. Пример реализации контекстного меню изображен на рисунке 15.

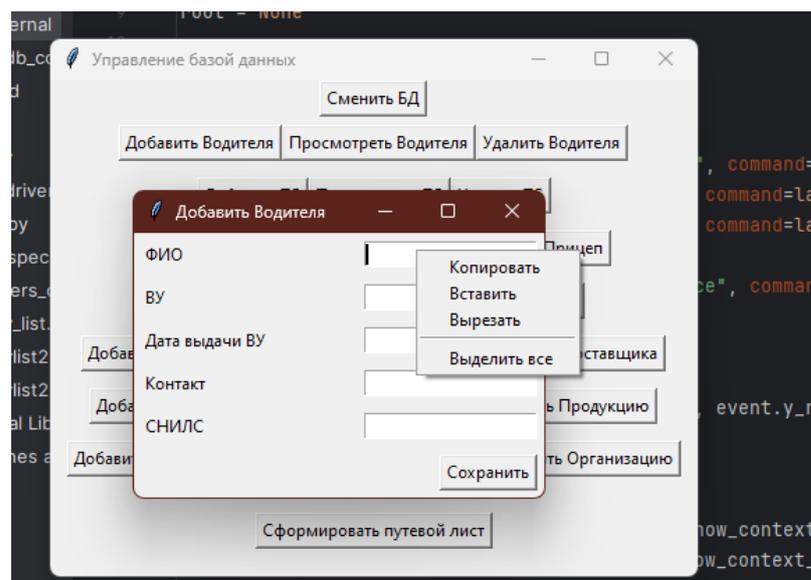


Рисунок 15 – Скриншот контекстного меню

Модуль управления данными представляет собой связующее звено между пользовательским интерфейсом и базой данных. Он выполняет все операции по обработке данных. Взаимодействие с БД осуществляется при использовании библиотеки `psycopg2`, позволяющей осуществлять SQL-запросы в среде выполнения Python [25]. Все транзакции данных проходят через этот модуль, что делает его работу критически важной для обеспечения стабильности системы.

Для установления соединения с базой данных используется функция `connect_to_db`. Параметры, загруженные модулем управления конфигурацией базы данных, используются для установления соединения с БД. При условии успешного соединения, функция возвращает объект подключения, который впоследствии используется остальными функциями модуля для выполнения SQL-запросов. Настройка подключения к базе данных предполагает спецификацию хоста, порта, имени пользователя и его пароля

Функция `add_record` применяется для добавления новых записей в БД. Скриншот ее работы представлен на рисунке 16.

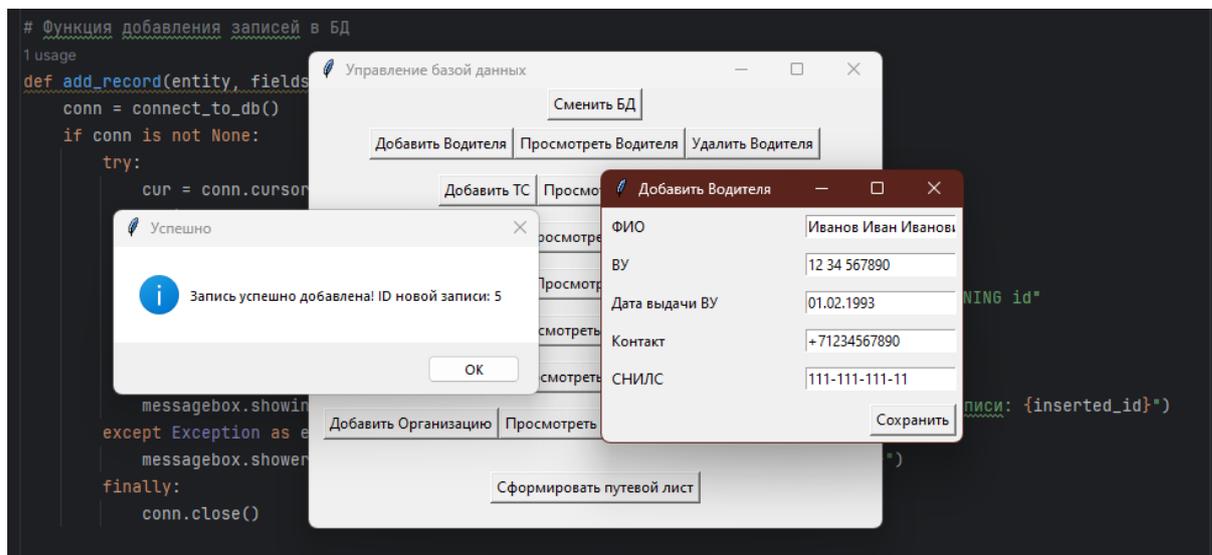


Рисунок 16 – Скриншот уведомления об добавлении записи в БД

Она принимает данные из текстовых полей приложения в качестве входной информации, которую впоследствии помещает в соответствующую таблицу. После инициализация подключения к базе данных, она формирует SQL-запрос на вставку и выполняет его с помощью курсора. После успешного внесения записи, в качестве ответа, возвращается идентификатор новой записи, что позволяет пользователю наглядно убедиться в успешности проделанных действий.

Функция `open_view_window` выполняет еще одну важную операцию – чтение и отображение на экране записей из базы данных. С ее помощью создается новое окно с таблицей, благодаря виджету `Ttreeview`, который является частью библиотеки `tkinter`. Сформированное окно наполняется всеми записи из выбранной таблицы базы данных, обеспечивая визуальную привлекательность и эффективное взаимодействие пользователя с системой. Наглядный пример реализации представлен на рисунке 17.

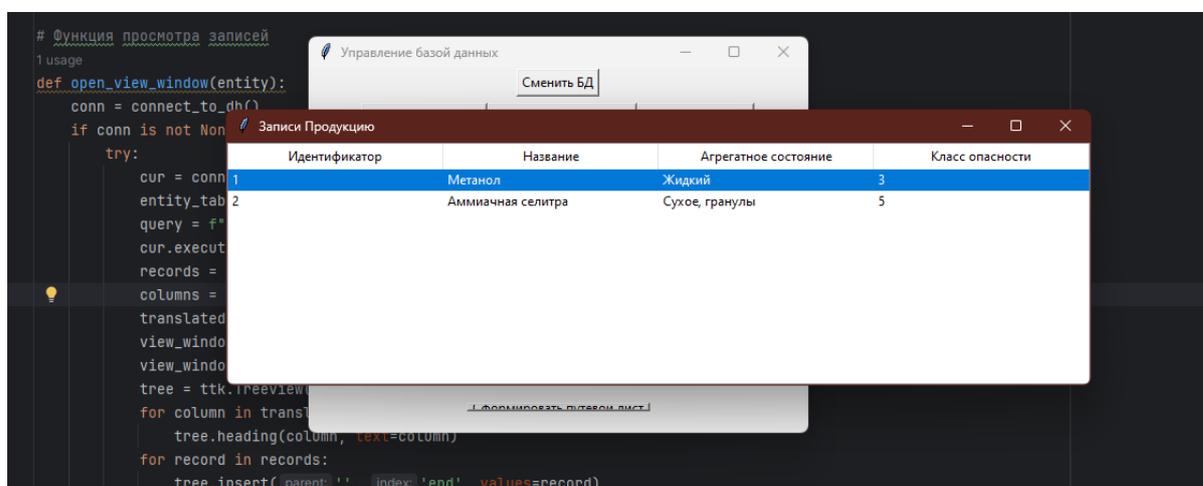


Рисунок 17 – Скриншот таблицы просмотра записей из БД

Для того, чтобы выполнить такую задачу, функция использует курсор базы данных для выполнения SQL-запроса `SELECT`. Полученные данные передаются в виджет `Ttreeview`, который, в свою очередь, форматирует и выводит на экран информацию в виде структурированной таблицы. Подобный

формат отображения позволяет пользователю визуализировать массив данных, а также упрощает процесс их анализа.

Скрипт `waylist2.py` представляет собой внешний subprocess, функциональность которого состоит в генерации путевых листов в формате `.xlsx` на основе данных, извлекаемых из БД. Он использует библиотеку `openpyxl` для работы с файлами Excel и автоматизирует формирование документов, необходимых для проведения организацией логистических операций [24].

Для запуска скрипта, в коде приложения используется функция `run_waylist_script`. Основываясь на Python-библиотеке `subprocess`, она запускает выполнение скрипта без компиляции в исполняемый файл. Запуск осуществляется через вызов интерпретатора Python с заранее определенным путем к файлу `waylist2.py` в качестве аргумента [26].

Функция `get_driver_data_from_db` производит извлечение записей из базы данных и внесение их в текстовые поля пользовательского интерфейса приложения.

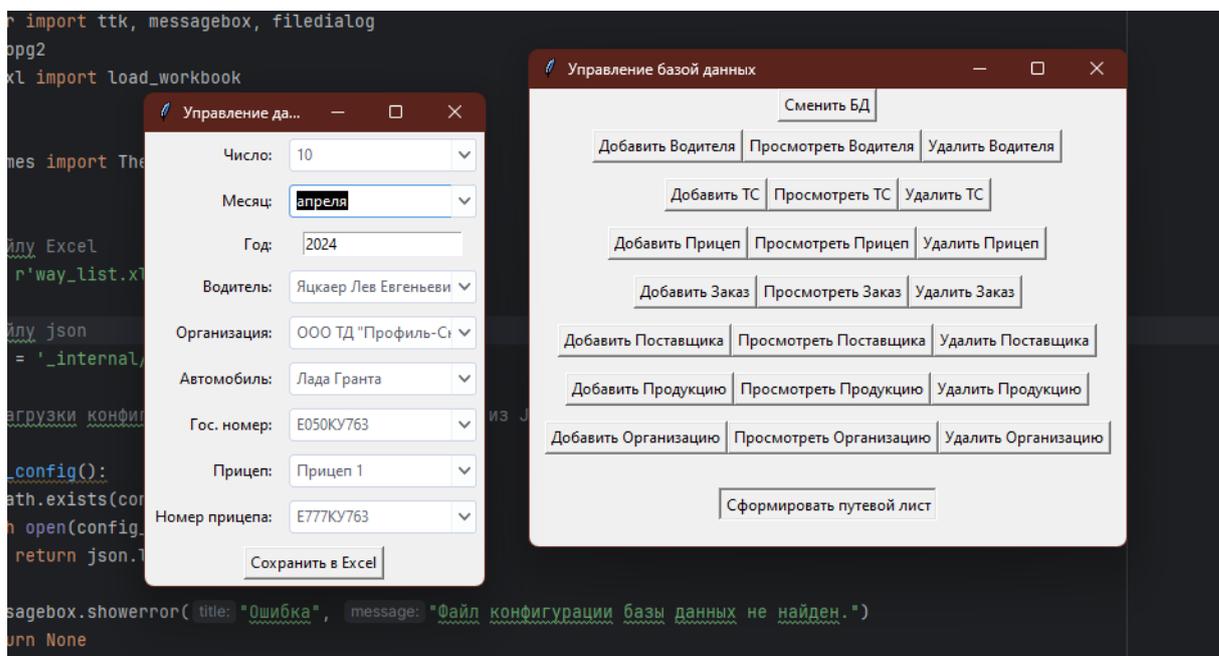


Рисунок 18 – Скриншот окна subprocess формирования путевого листа

Функцией, отвечающей за сбор данных из интерфейса пользователя, их обработку и сохранение заполненного документа в формате Excel является `save_to_excel`. Скриншот работы представлен на рисунке 19.

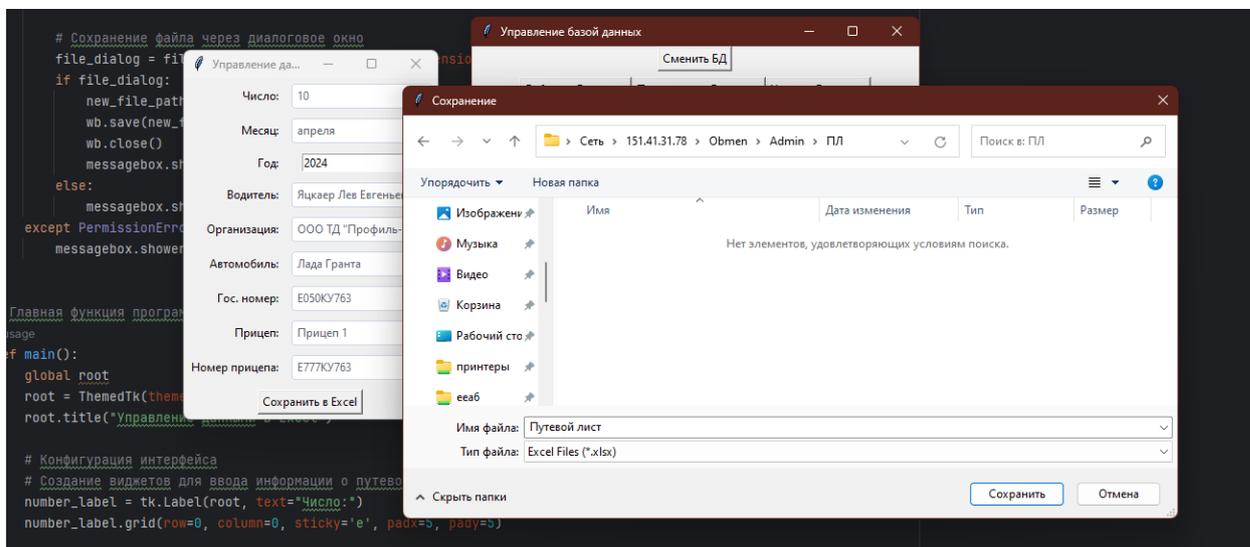


Рисунок 19 – Скриншот диалогового окна сохранения сформированного путевого листа

Механизм работы функции состоит в извлечении параметров путевого листа из пользовательского интерфейса программы, интеграции их в соответствующие ячейки документа с расширением `.xlsx` и последующем сохранении в указанную пользователем директорию. При сохранении, на экран выводится диалоговое окно проводника Windows с возможностью ввести имя сформированного документа и выбором места сохранения в привычном пользователю интерфейсе.

3.5 Описание функциональности АИС

Разработанная АИС предоставляет пользователям обширный набор инструментов для управления данными. Пользовательский опыт ориентирован на обеспечение удобства и повышение эффективности выполнения регулярных задач при использовании централизованной системы.

При первом запуске программы на клиенте происходит процедура настройки базы данных, включающая в себя ввод параметров подключения – адреса, порта, логина и пароля пользователя. Параметры, заданные при первичной настройке, записываются в конфигурационный файл, необходимый для последующего подключения и базой. После сохранения настроек, система автоматически инициализирует главное окно.

Главное окно предоставляет пользователям имеют доступ к функциональным кнопкам, позволяющим добавлять, просматривать и удалять записи из базы данных. Каждая из операций обеспечена соответствующими формами ввода-вывода, позволяющими пользователю системы легко добавлять и корректировать информацию. В качестве примера можно привести процесс добавления нового водителя: пользователь нажимает кнопку «Добавить водителя», заполняет открывшуюся форму, включающую поля для ввода ФИО, номера водительского удостоверения, даты его выдачи, контактных данных и номера СНИЛС, после чего, нажимает на кнопку «Сохранить», система выполняет процедуру записи в базу данных и отвечает подтверждающим уведомлением.

Функционал просмотра записей реализован в виде формируемых в интерфейсе приложения таблиц, которые отображают все записи из выбранной таблицы базы данных. Пользователи могут быстро ознакомиться и проанализировать данные, представленные в комфортном формате. Удаление записей также упрощено и выглядит схожим образом с их просмотром. Процедура удаления может быть выполнена без каких-либо трудозатрат благодаря табличному представлению формы удаления с единственной кнопкой «Удалить».

Особое внимание в системе уделено процессу формирования путевых листов, являющихся одними из важнейших документов в логистической деятельности компании. Функциональность формирования обеспечивается при помощи разработанного внешнего скрипта, автоматизирующего сбор записей из различных таблиц БД, их форматирования и записи в

соответствующие ячейки файла Excel с последующим сохранением в любой, удобной пользователю, директории.

Пользовательский путь в системе характеризуется высокой степенью автоматизации стандартизированных операций, позволяя увеличить скорость взаимодействия с данными. Система предлагает гибкие инструменты для обработки информации, облегчая горизонтальную коммуникацию между отделами и перераспределяя рабочее время специалистов, что позволяет назвать ее комплексным функциональным решением для внедрения в работу организации.

3.6 Тестирование программного проекта как неотъемлемая часть разработки программного обеспечения

Тестирование является важнейшим инструментом для оценки качества разрабатываемой информационной системы. Оно направлено на выявление допущенных ошибок в логике и коде программы на ранней стадии ее разработки. Проведение множественных тестов позволяет обеспечить быстрое действие, стабильность и соответствие системы заявленным функциональным требованиям. Также, тестирование снижает риски возникновения сбоев системы после ее непосредственного развертывания [9].

3.6.1 Выбор методов тестирования программного продукта

В ходе разработки описываемой автоматизированной информационной системы рассматривался выбор различных методик тестирования, направленных на оценку качественных характеристик программного продукта и его надежности. Итоговый выбор основывается на специфике разработанной системы, а также на необходимости обеспечить ее бесперебойное функционирование в производственной среде.

Модульное тестирование – одна из основных методик, предполагает проверку каждого отдельного модуля программного обеспечения. Этот

подход изолирует обнаруженные ошибки и дефекты на уровне отдельных компонентов системы, что упрощает процесс их поиска и исправления.

Интеграционное тестирование, наоборот, оценивает корректность взаимодействия между различными модулями системы. С его помощью выявляются ошибки, возникающие при совместной работе нескольких компонентов.

Функциональное тестирование ставит целью проверку соответствия системы обозначенным функциональным требованиям. В ходе тестирования проверяется корректность выполнения всех заявленных функций системы, а также ее быстродействие.

Тестирование пользовательского интерфейса – еще один немаловажный аспект тестирования. Оно позволяет оценить удобство и логичность интерфейса, а также его соответствие ожиданиям пользователей.

Применение комплекса различных методов тестирования позволяет выявлять и устранять возникающие проблемы на ранней стадии разработки, позволяют обеспечить высокую степень надежности и стабильности системы.

3.6.2 Описание программного кода тестирования АИС

Для автоматизации процесса функционального тестирования системы был разработан набор тестовых сценариев. Реализация осуществлена на языке программирования Python с использованием библиотеки unittest [30], обеспечивая комплексную нативную проверку функциональных возможностей системы. В проверку включена оценка возможности выполнять операции записи и удаления, отображения данных и запуска сторонних модулей. Тест позволил оценить стабильность и производительность системы, а также выявить допущенные ошибки и недостатки ее функционала. Благодаря полученным в ходе теста результатам, разработка системы стала более структурированной, процесс ее разработки ускорился, а также была обеспечена более высокая степень ее качества [4].

Один из сценариев тестирования, направлен на проверку функционала добавления и удаления записей из таблицы «Водители». При его выполнении формируются данные для добавления тестовой записи, затем выполняется проверка ее добавления в базу данных, после чего происходят удаление добавленной записи и проверка того, что она была успешно удалена. Оценка корректности работы системы при выполнении операций добавления и удаления данных обеспечивает понимание того, что система работает в соответствии с ожидаемым поведением. Благодаря этому тесту появляется возможность выявления потенциальных ошибок и недостатков в функционале, которая позволяет своевременно принять меры по их устранению. Скриншот блока кода тестирования представлен на рисунке 20.

```
8
9 ▶ def test_add_and_delete_driver(self):
10     # Данные для добавления водителя
11     test_driver_data = {
12         "full_name": "Тестовый Водитель",
13         "license_number": "1234567890",
14         "vu_issue_date": "2023-01-01",
15         "contact_details": "test@example.com",
16         "snils": "9876543210"
17     }
18
19     # Добавляем водителя
20     DB.add_record(entity: "Водителя", test_driver_data)
21
22     # Проверяем, что запись добавлена (через просмотр всех водителей)
23     conn = DB.connect_to_db()
24     cur = conn.cursor()
25     cur.execute("SELECT * FROM drivers WHERE full_name = %s", (test_driver_data["full_name"],))
26     added_driver = cur.fetchone()
27     self.assertIsNotNone(added_driver)
28
29     # Удаляем водителя по ID
30     driver_id = added_driver[0]
31     DB.delete_record(entity: "Водителя", driver_id)
32
33     # Проверяем, что запись удалена
34     cur.execute("SELECT * FROM drivers WHERE id = %s", (driver_id,))
35     deleted_driver = cur.fetchone()
36     self.assertIsNone(deleted_driver)
37
38     conn.close()
39
```

Рисунок 20 – Python-код тестирования операций записи в БД

Второй сценарий тестирования направлен на проверку корректности работы функционала просмотра записей. Он реализован благодаря применению метода `patch` из библиотеки `unittest.mock`. Суть тестирования состоит в проверке вызова функции `open_view_window` с правильным параметром. Скриншот блока кода изображен на рисунке 21.

```
39
40     @patch('DB.open_view_window')
41     def test_view_products(self, mock_open_view_window):
42         # Вызываем функцию просмотра продукции
43         DB.open_view_window("Продукцию")
44
45         # Проверяем, что функция была вызвана с правильным параметром
46         mock_open_view_window.assert_called_once_with("Продукцию")
47
```

Рисунок 21 – Python-код тестирования вызова функции просмотра записей

Еще один сценарий тестирования обеспечивает проверку корректного запуска subprocess генератора путевых листов (Рисунок 22).

```
47
48     def test_run_waylist_script(self):
49         # Проверяем, существует ли файл waylist2.exe
50         self.assertTrue(os.path.exists(r'waylist2.exe'), msg="Файл waylist2.exe не найден")
51
52         # Запускаем скрипт (здесь нужно проверить, что процесс запущен)
53         DB.run_waylist_script()
54
```

Рисунок 22 – Python-код тестирования запуска генератора путевых листов

Его работа состоит в проверке наличия файла `waylist2.exe`, используемого для формирования путевых листов, а также запуске этого внешнего скрипта.

3.6.3 Результаты тестирования АИС

В рамках тестирования АИС были проведены четыре теста, целью которых была проверка основных функциональных возможностей системы.

Результаты проведенных тестов представлены в таблице 2.

Таблица 2 – Результаты функционального тестирования

Название теста	Описание	Ожидание	Результат	Затраченное время
Добавление водителя	Проверка функции добавления водителя в базу данных	Запись добавлена успешно.	Запись добавлена успешно.	3.0 секунды
Удаление водителя	Проверка функции удаления водителя из базы данных.	Запись удалена успешно.	Запись удалена успешно.	3.5 секунды
Просмотр продукции	Проверка вызова функции для открытия окна просмотра продукции с правильным параметром.	Функция вызвана с параметром "Продукция".	Функция вызвана с параметром "Продукция".	2.0 секунды
Запуск скрипта путевого листа	Проверка существования файла waylist2.exe и запуска скрипта.	Файл waylist2.exe существует и скрипт запускается.	Файл waylist2.exe существует и скрипт запускается.	2.3 секунды

Целью первого теста была проверка функции добавления водителя в базу данных. Ожидаемым результатом являлось успешное добавление записи и получение соответствующего сообщения. Тест прошел в соответствии с

ожиданиями. Запись была добавлена успешно. Затраченное время на выполнение теста, составило 3 секунды.

Второй тест является частью первого сценария тестирования. Его цель – проверка функции удаления водителя из базы данных. Результат теста подтвердил ожидаемый результат, запись была удалена успешно. Длительность выполнения теста составила 3,5 секунды.

Цель третьего теста состояла в проверке вызова функции для открытия окна табличного просмотра записей с параметром «Продукция». Ожидался успешный вызов обозначенной функции. В результате тестирования, ожидания были подтверждены. Вызов указанной функции занял 2 секунды.

Цель последнего теста – проверка существования файла waylist2.exe и запуск скрипта генерации путевых листов. Ожидалось подтверждение существования файла в директории проекта и успешный запуск подпроцесса генератора. После выполнения теста, был получен ожидаемый результат. Файл waylist2.exe был найден в указанной директории и успешно запущен. Время, затраченное на выполнение теста, составило 2,3 секунды.

Результаты тестирования свидетельствуют о корректной работе системы и выполнении заявленных функциональных возможностей в соответствии с ожиданиями. Тесты подтверждают тезис о надежности и быстродействии АИС.

Заключение

Разработка и внедрение автоматизированной информационной системы управления промышленными поставками химических продуктов в работу организации ООО ТД «Профиль-Снаб» позволили оптимизировать часть бизнес-процессов компании.

После проведенного исследования предметной области была выявлена необходимость автоматизации ряда операций, выполняемых сотрудниками компании. Существовавшая система взаимодействия, основанная на личных контактах, не обеспечивала необходимый уровень прозрачности и оперативности обмена информацией. Отсутствие цифровой платформы для учета заказов и формирования логистической документации приводило к значительным временным затратам.

Разработка АИС позволила решить комплекс выявленных проблем. Внедрение централизованной системы хранения и обработки информации позволило автоматизировать часть задач, связанных с логистикой. Сотрудникам стало доступно отслеживание информации о новых заказах, а также автоматизированное формирование путевых листов.

В результате внедрения системы, удалось сократить время, затрачиваемое сотрудниками на обработку заказов и формирование документации.

Переход от ручного ввода данных к выбору необходимого варианта из записей, заранее занесенных в базу данных, позволил снизить риск совершения ошибок, связанных с невнимательностью персонала.

Сформированная платформа для управления заказами и логистикой помогла обеспечить прозрачность всех этапов процесса доставки продукции, от формирования заказа, до составления маршрутного задания. Это существенно упростило коммуникацию между отделами и повысило эффективность контроля качества.

Своевременная обработка заказов и оптимизация логистических процессов, в свою очередь, обеспечили повышение качества обслуживания клиентов и существенно укрепили их лояльность.

Проведенное автоматизированное функциональное тестирование продемонстрировало стабильность системы, ее быстродействие, а также, соответствие всем заявленным функциональным требованиям. Автоматизация тестирования позволила исключить влияние внешних факторов на ход исследования. Также, автоматизированное тестирование обладает фактором воспроизводимости, что может быть применимо при дальнейших модернизации и расширении системы.

Дальнейшее развитие системы предполагает реализацию следующих направлений:

- интеграция с другими информационными системами компании;
- разработка мобильного приложения;
- внедрение инструментов аналитики и отчетности;
- масштабирование системы.

Внедрение автоматизированной информационной системы в работу ООО ТД «Профиль-Снаб», а также дальнейшее развитие и реализация описанных выше направлений, являются важными шагами на пути к цифровизации компании. Применение современных технологий управления информацией позволит ей не только удерживать, но и существенно укрепить свои позиции на рынке.

Список используемой литературы

1. Арлоу, Арлоу UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Арлоу Арлоу, Айла Нейштадт. – пер. с англ. – Санкт-Петербург : Символ-Плюс, 2007. – 624 с. – ISBN 978-5-93286-094-6.
2. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем. / А. М. Вендров. – Москва : Финансы и статистика, 2020. – 203 с.
3. Гниденко И.Г. Информационные технологии в бизнесе: Учебное пособие / И.Г. Гниденко, С.А. Соколовская. – Москва : Вектор, 2005. – 160 с.
4. Дастин, Э. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация / Э. Дастин, П. Джон, Д. Рэшка. – М. : Лори, 2003. – 592 с.
5. Дейт К. Дж. Введение в системы базы данных / К. Дж. Дейт. – Москва : Эксмо, 2020. – 370 с.
6. Когаловский М.Р. Энциклопедия технологий баз данных / М.Р. Когаловский. – Москва : Финансы и статистика, 2005. – 800 с.
7. Компаниец, В. Проектирование и юзабилити-исследование пользовательских интерфейсов / В. Компаниец, А. Лызь. – Ростов-на-Дону : Южный Федеральный Университет, 2020. – 109 с.
8. Кузнецов С.Д, Основы баз данных / С.Д, Кузнецов. – Москва : Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2011. – 488 с.
9. Куликов С.С Тестирование программного обеспечения. Базовый курс / С.С Куликов. – Минск : Четыре четверти, 2020. – 310 с.
10. Лутц М. Изучаем Python, том 1, 5-е издание / М. Лутц. – СПб : Диалектика, 2019. – 832 с. – ISBN 978-5-907144-52-1

11. Максимчук Р.А. Проектирование баз данных с помощью UML / Р.А. Максимчук, Э. Д. Нейбург. – Москва : Издательский дом «Вильямс», 2002. – 0 с.
12. Олифер В. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. / В. Олифер, Н. Олифер. – СПб : ООО Издательство «Питер», 2016. – 992 с.
13. Остроух А. В., Суркова Н. Е. Проектирование информационных систем / Суркова Н. Е. Остроух А. В.,. – Санкт-петербург : Издательство "Лань", 2021. – 164 с. – ISBN 978-5-8114-8377-8
14. Плаксин М. А. Тестирование и отладка программ для профессионалов будущих и настоящих / М. А. Плаксин. – Москва : БИНОМ. Лаборатория знаний, 2013. – 167 с.
15. Постолиит, А. В. Python, Django и PyCharm для начинающих / А. В. Постолиит. – Москва : БХВ, 2021. – 465 с. – ISBN 978-5-9775-6779-4.
16. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка, 2-е изд. / Дж. Рамбо, М. Блаха. – СПб : Питер, 2021. – 544 с. – ISBN 978-5-4461-9428-5
17. Рогов Е. PostgreSQL 15 изнутри / Е. Рогов. – Москва : ДМК Пресс, 2023. – 662 с.
18. Саммерфилд , М. Программирование на Python 3. Подробное руководство / , М. Саммерфилд. – Москва : Символ-Плюс, 2009. – 11 с.
19. Советов, Б.Я. Информационные технологии / Б.Я. Советов, В.В. Цехановский. – Москва : Высшая школа, 2019. – 410 с.
20. Уткин В.Б. Информационные системы и технологии в экономике / В.Б. Уткин. – Москва : ЮНИТИ, 2003. – 334 с.
21. Федоров И. Г. Моделирование бизнес-процессов в нотации BPMN 2.0 [Текст] : научно-практическое издание / И. Г. Федоров. – Москва : МЭСИ, 2013. – 264 с. – ISBN 978-5-7764-0772-7

22. Data Flow Diagram (DFD) Tutorial. Learning Guides : сайт. – URL: <https://creately.com/blog/diagrams/data-flow-diagram-dfd/> (дата обращения: 12.03.2024)
23. json — JSON encoder and decoder : сайт. – URL: <https://docs.python.org/3/library/json.html> (дата обращения: 04.04.2024)
24. openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files : сайт. – URL: <https://openpyxl.readthedocs.io/en/stable/> (дата обращения: 02.04.2024)
25. Psycorg 2.9.9 documentation : сайт. – URL: <https://www.psycorg.org/docs/> (дата обращения: 30.03.2024)
26. Subprocess management : сайт. – URL: <https://docs.python.org/3/library/subprocess.html> (дата обращения: 04.04.2024)
27. The Hitchhiker’s Guide to Python : сайт. – URL: <https://docs.python-guide.org/> (дата обращения: 15.03.2024)
28. Tkinter Documentation : сайт. – URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 27.03.2024)
29. UML Diagrams Tutorial. Learning Guides : сайт. – URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language> (дата обращения: 10.03.2024)
30. unittest — Unit testing framework : сайт. – URL: <https://docs.python.org/3/library/unittest.html> (дата обращения: 25.04.2024)