

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

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Автоматизация бизнес-процесса обработки заявок от клиентов

Обучающийся

К.В. Комаров

(Инициалы Фамилия)

(личная подпись)

Руководитель

Н.Н. Рогова

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

Тольятти 2023

Аннотация

Бакалаврская работа выполнена на тему «Автоматизация бизнес-процесса обработки заявок от клиентов».

Цель работы заключается в автоматизации бизнес-процесса компании БиАйЭй-Технолоджи по обработке заявок от клиентов.

Во введении выпускной квалификационной работы описывается: актуальность темы, объект и предмет, а также цель работы и требуемые задачи.

В первой главе бакалаврской работы рассмотрена характеристика компании БиАйЭй-Технолоджи, проведено моделирование бизнес-процесса обработки заявок от клиентов компании БиАйЭй-Технолоджи, определены требования к разрабатываемому программному продукту, оценены существующие программы, для решения рассматриваемых задач.

Во второй главе выпускной квалификационной работы проведено проектирование программного продукта, а именно созданы диаграммы UML, а также представлена логическая модели базы данных.

В третьей главе показана реализация контрольного примера, разработанного программного продукта, а также оценена экономическая эффективность внедрения информационной системы.

Бакалаврская работа состоит из 69 страниц и включает 33 рисунка, 11 таблиц, 20 источников, 2 приложения.

Оглавление

Введение.....	5
Глава 1 Анализ предметной области БиАйЭй-Технолоджиз.....	7
1.1 Технико-экономическая характеристика БиАйЭй-Технолоджиз....	7
1.2 Концептуальное моделирование бизнес-процесса обработки заявок от клиентов БиАйЭй-Технолоджиз.....	9
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	16
1.4 Постановка задачи на разработку проекта внедрения АИС обработки заявок от клиентов БиАйЭй-Технолоджиз.....	18
1.5 Разработка модели бизнес-процесса «как должно быть».....	19
Глава 2 Логическое проектирование информационной системы обработки заявок от клиентов БиАйЭй-Технолоджиз.....	25
2.1 Выбор технологии логического моделирования АИС.....	25
2.2 Логическое проектирование автоматизированной информационной системы	26
2.3 Проектирование базы данных автоматизированной информационной системы	28
Глава 3 Реализация проекта разработки информационной системы и оценка его эффективности	32
3.1 Архитектура и программное обеспечение информационной системы	32
3.2 Описание функциональности информационной системы обработки заявок от клиентов БиАйЭй-Технолоджиз.....	33
3.3 Оценка и обоснование экономической эффективности внедрения информационной системы	42
Заключение	48

Список используемой литературы и используемых источников.....	50
Приложение А Отчет по проектированию	52
Приложение Б Код программы.....	55

Введение

Тема выпускной квалификационной работы «Автоматизация бизнес-процесса обработки заявок от клиентов». В работе рассматривается проблема сбора, обработки и выполнения заявок от клиентов компании на техническую поддержку и сопровождение программных и аппаратных решений, разработкой которых занимается компания.

Данная тема является актуальной, поскольку в компании БиАйЭй-Технолоджиз часть функций по технической поддержке и сопровождения, разрабатываемых клиентских программных продуктов осуществляет удаленно. Поэтому актуально разработать автоматизированную систему обработки заявок от клиентов БиАйЭй-Технолоджиз, для повышения эффективности работы сотрудников.

В выпускной квалификационной работе объект исследования - процесс обработки заявок от клиентов.

Предмет исследования – автоматизация процесса обработки заявок от клиентов БиАйЭй-Технолоджиз отделом технической поддержки.

Цель выпускной квалификационной работы – автоматизация бизнес-процесса обработки заявок от клиентов.

Для достижения поставленной цели необходимо решить ряд задач:

- изучить бизнес-процесс по обработки заявок от клиентов БиАйЭй-Технолоджиз;
- провести проектирование базы данных для информационной системы;
- реализовать прототип системы;
- провести тестирование прототипа;
- рассчитать экономическую эффективность проекта.

Для реализации выпускной квалификационной работы применялись следующие методы исследования:

- анализ - рассмотрение предмета выпускной квалификационной работы с учетом особенностей деятельности компании БиАйЭй-Технолоджиз;
- синтез - объединение отдельных элементов (признаки, свойства) в одно целое;
- эксперимент. Заключается в тестировании и настройке разработанного программного продукта.

Структура бакалаврской работы состоит из введения, трех глав, в которых рассматриваются вопросы постановки задачи, проектирования и реализации системы обработки заявок от клиентов компании, а также заключения и списка литературы и используемых источников, и приложения с кодом программ.

Глава 1 Анализ предметной области БиАйЭй-Технолоджиз

1.1 Технико-экономическая характеристика БиАйЭй-Технолоджиз

«БиАйЭй-Технолоджиз» (BIA Technologies) – российский системный интегратор и разработчик программного обеспечения. Более 10 лет создает ИТ-продукты и оптимизирует процессы для повышения эффективности бизнеса заказчиков.

«Целью БиАйЭй-Технолоджиз является создание ИТ-решений и продуктов, удовлетворяющих потребности бизнеса и государства в цифровизации в сферах производства, транспорта и логистики, агропромышленности, металлургии, топливной энергетики, добычи нефти и газа» [1].

Структура управления компанией представлена на рисунке 1.

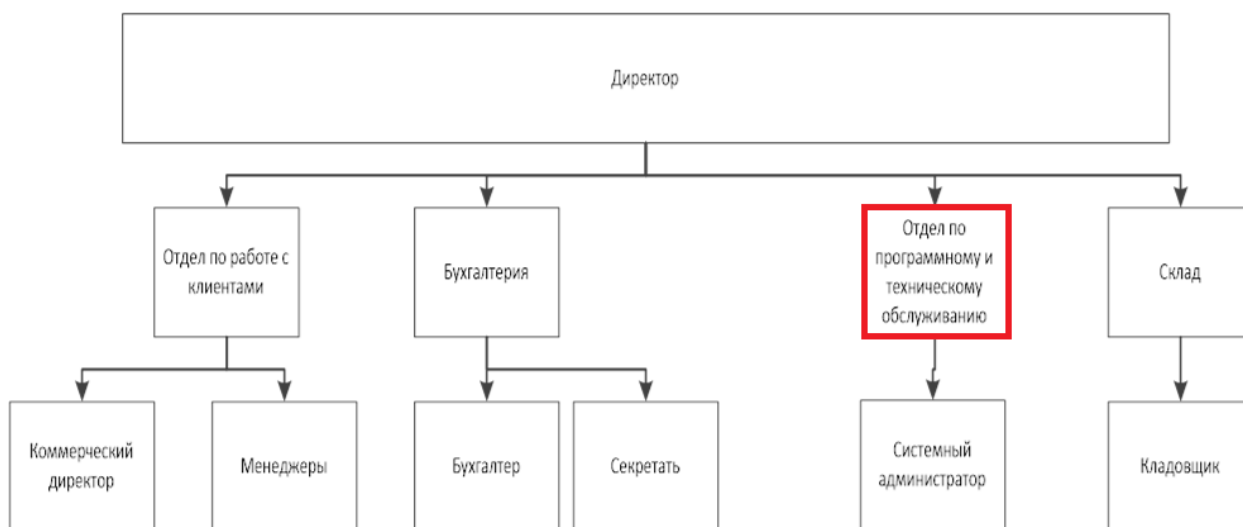


Рисунок 1 – Структура управления БиАйЭй-Технолоджиз

Коммерческий директор является руководителем отдела по работе с клиентами. В его обязанности входит:

- исполняет обязанности директора в его отсутствие;

- организация работы менеджеров, проверка профессиональных знаний и умений;
- проводит переговоры с контрагентами предприятия, заключает от имени предприятия договоры, обеспечивает выполнение договорных обязательств;
- руководство организации снабжения и сбыта продукции.

В обязанности менеджеров входит:

- работа с клиентами, поиск новых клиентов,
- проведение первичных переговоров с клиентами.

В обязанности бухгалтера входит:

- ведение бухгалтерского учета,
- прием, контроль и обработка первичной документации,
- начисление заработной платы.

В обязанности секретаря входит:

- прием и обработка документов, писем и заявлений,
- составление договоров, писем и заявление.

В обязанности кладовщика входит:

- прием на склад, взвешивание, хранение и выдача со склада продукции,
- проверка соответствия принимаемой продукции сопроводительными документами.

Функции и обязанности отдела по программному и техническому обеспечению. В данном отделе работает системный администратор, в задачи которого входят:

- контроль состояния информационной инфраструктуры организации,
- развитие программного обеспечения компании,
- обработка заявки клиентов компании по техническому и программному обслуживанию.

1.2 Концептуальное моделирование бизнес-процесса обработки заявок от клиентов БиАйЭй-Технолоджиз

Одной из функции БиАйЭй-Технолоджиз является консультирование клиентов по технической поддержке, разрабатываемых решений с помощью электронной почты, заявок на сайте или с помощью телефонных звонков.

При изучении бизнес-процесса необходимо построить ряд функциональных моделей, для определения потока информации, а также для определения границ внешней среды и поступающий оттуда информации.

На рисунке 2 показана контекстная диаграмму бизнес-процесса обработки заявок от клиентов БиАйЭй-Технолоджиз, согласно методологии IDEF0.

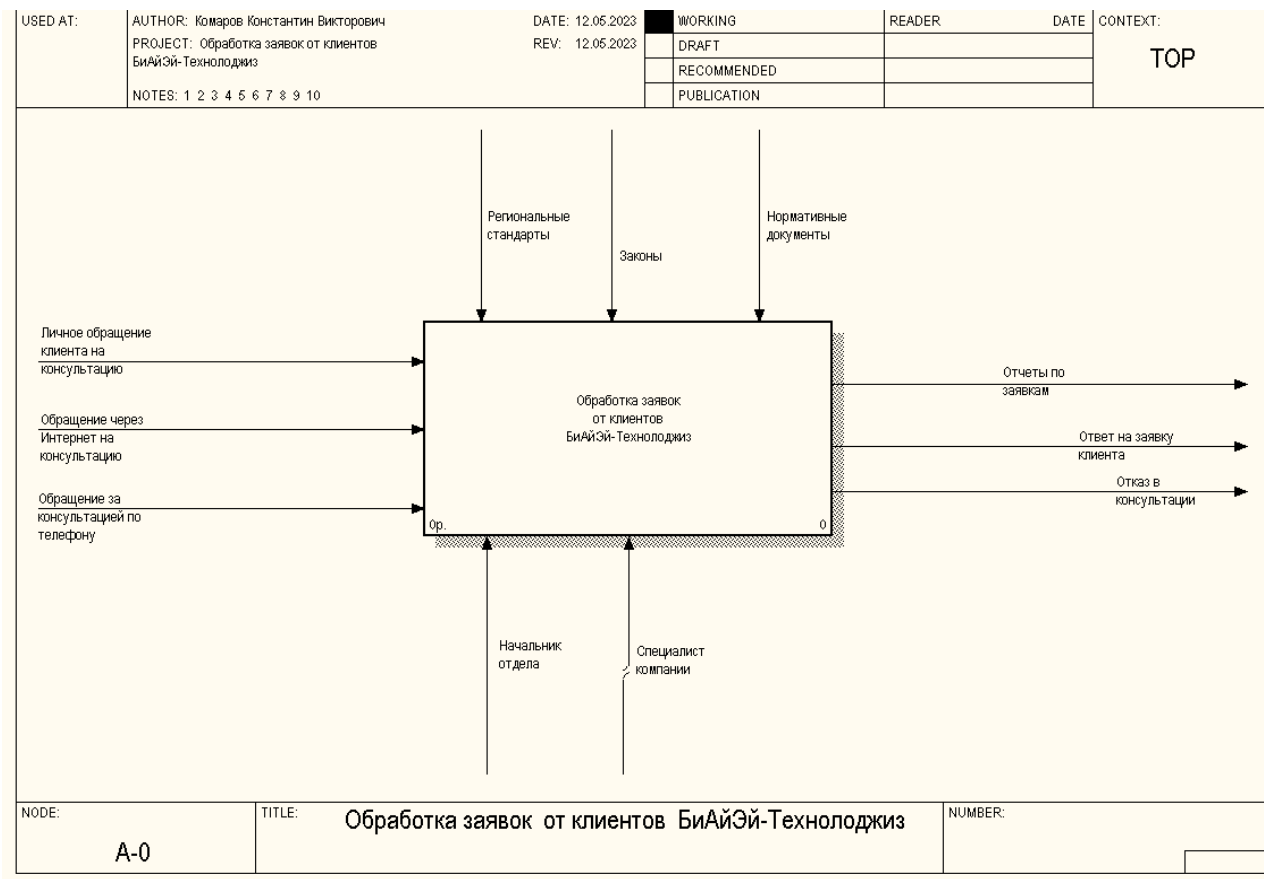


Рисунок 2- Контекстная диаграмма бизнес-процесса «Обработка заявок от клиентов»

При управлении бизнес-процесс обработки заявок от клиентов компании руководствуется следующими управляющими документами

- стандарты по работе с клиентами, принятые в регионе;
- Законы;
- документы, регулирующие консультационные услуги.

Декомпозиция бизнес-процесса, представлена на рисунке 3.

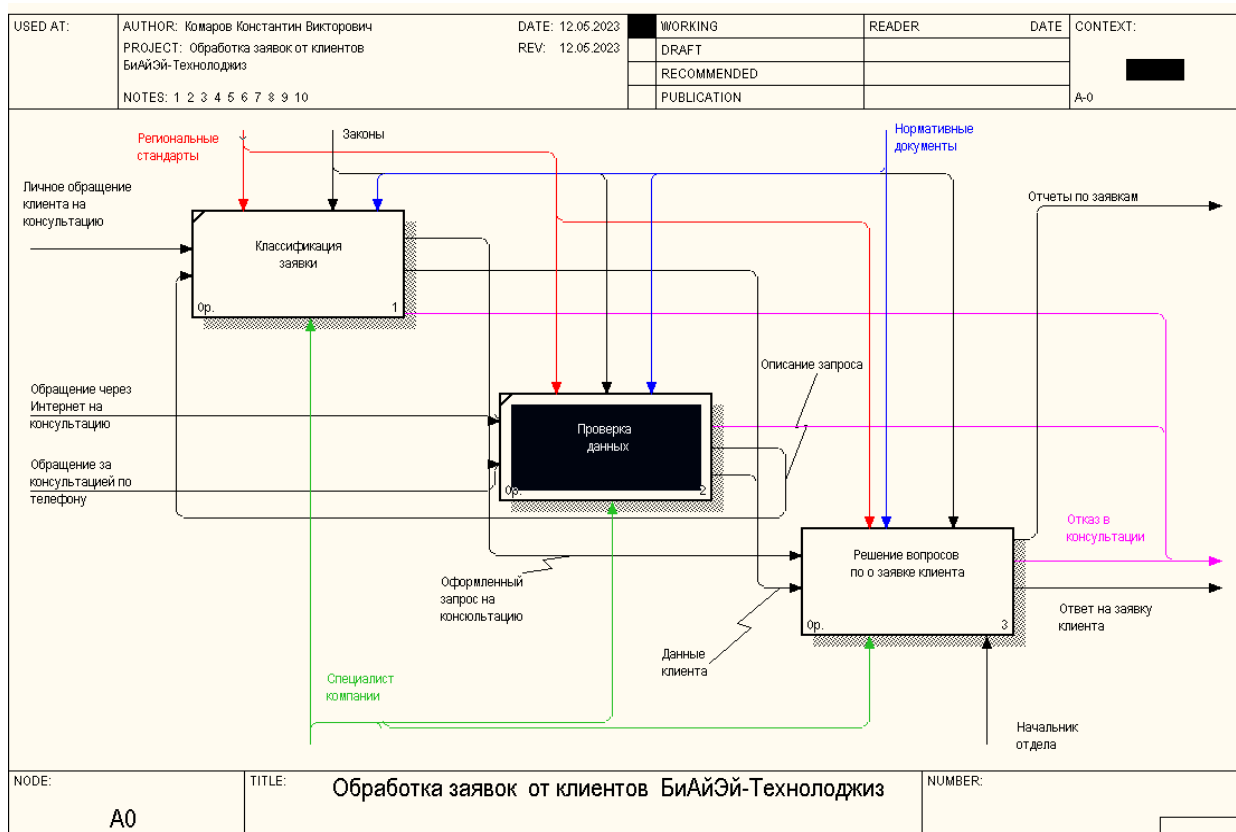


Рисунок 3 – Диаграмма декомпозиции контекстной диаграммы «Обработка заявок от клиентов»

Входными потоками бизнес-процесса обработка заявок от клиентов БиАйЭй-Технолоджиз отношений являются:

- личная заявка клиента-компании на консультацию,
- заявка клиента-компании через Интернет на консультацию,
- заявку клиентов-компании за консультацией по телефону.

Механизмами является:

- «Начальник отдела»;
- «Специалист компании», отвечающих за работу с клиентскими заявками.

Выходными потоками являются:

- «Отчеты по заявкам по обработке заявок от клиентов»;
- «Ответ на заявку клиента»;
- «Отказ в консультации клиенту (сюда входят разные причины)».

Рассмотрены подробно операции бизнес-процесса обработка заявок от клиентов БиАйЭй-Технолоджиз более подробно (Таблица 3)

Таблица 3 - Бизнес-процесс «Обработка заявок от клиентов БиАйЭй-Технолоджиз». «Как есть»

Название	Вход	Выход	Управление	Механизм
Классификация заявки от клиентов БиАйЭй-Технолоджиз	Личная заявка клиента БиАйЭй-Технолоджиз на консультацию, Описание запроса	Оформленный запрос на консультацию, Контактные данные и данные запроса, поступающих от клиента Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками
Проверка данных	Запрос на консультацию, поступивших от клиента через Интернет на консультацию, Запрос на консультацию по телефону	Описание запроса, Контактные данные и данные запроса, поступающих от клиента Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками
Решение вопросов по заявке клиентов	Контактные данные и данные запроса, поступающих от клиента, Оформленный запрос на консультацию	Ответ на заявку клиентов, Отчеты по заявкам Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками, Начальник отдела

В ходе выпускной работы большое внимание уделено процессу фиксации и классификации заявки клиентов и проверка данных клиента (с учетом проверки, приобреталась ли у данных клиентов пакет на сопровождение и техническое консультирование), обратившихся дистанционно для получения консультации.

Процесс «Решение вопросов по заявке клиента» (Рисунок 4) состоит из следующих блоков:

- прием заявки клиентов,
- прием заявки клиентов,
- обработка заявки клиентов,
- учет результатов обработки заявки клиентов.

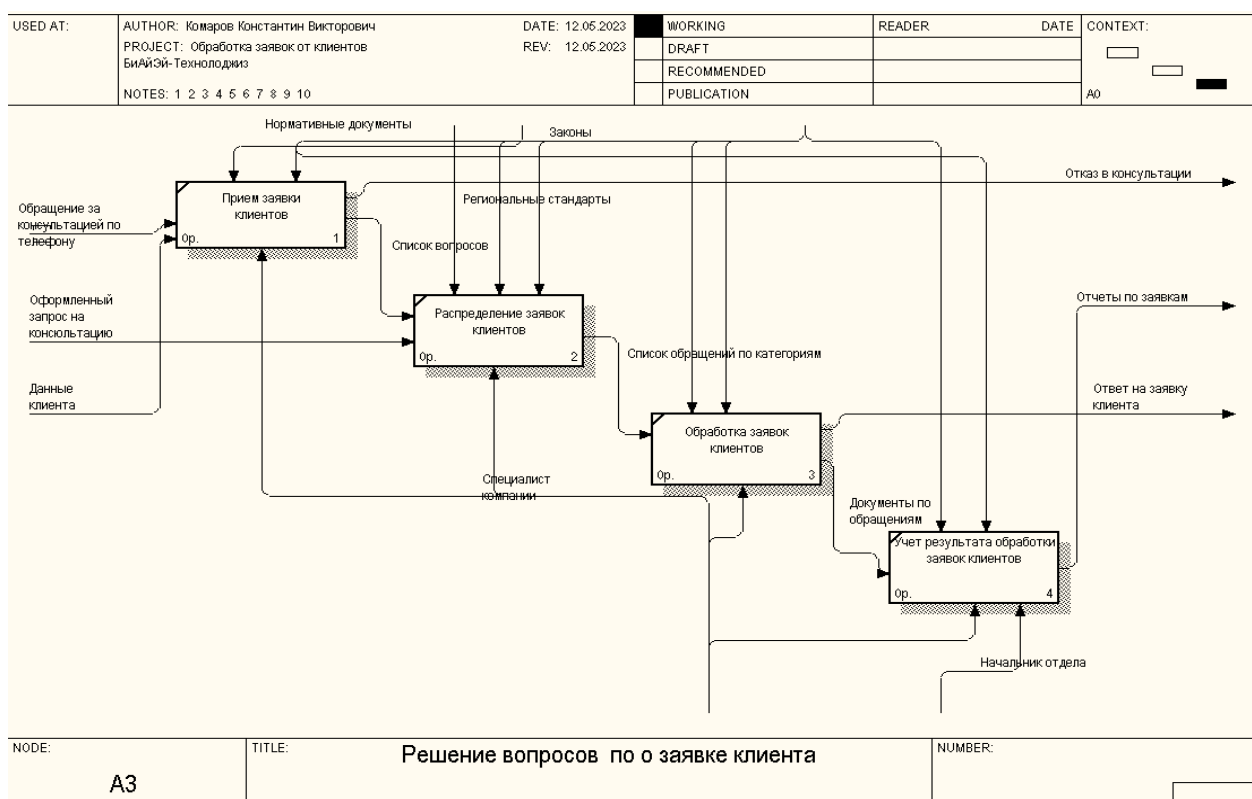


Рисунок 4 – Процесс «Решение вопросов по заявке клиента»

Операции по бизнес-процессу «Решение вопросов по заявке клиента» более подробно показаны в таблице 4

Таблица 4 - Бизнес-процесс «Решение вопросов по заявке клиента». «Как есть»

Название	Вход	Выход	Управление	Механизм
Прием заявки клиентов	Заявку клиентов за консультацией по телефону, Контактные данные и данные запроса, поступающих от клиента	Список вопросов, Отказ в консультации	Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками
Прием заявки клиентов	Список вопросов, Оформленный запрос на консультацию	Список заявки клиентов по категориям	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками
Обработка заявки клиентов	Список заявки клиентов по категориям	Документы по заявкам, Ответ на заявку клиента	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками
Учет результатов обработки заявки клиентов	Документы по заявкам	Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками, Начальник отдела

Отчет по процессу проектирования представлен «Как есть» в приложении А.

В процессе решения поставленной задачи задействованы специалисты компании. Результатными показателями бизнес-процесса является количество принятых заявок от клиентов и количество обработанных заявок от клиентов

БиАйЭй-Технолоджиз. Исполнителями бизнес-процесса являются специалист компании, отвечающих за работу с клиентскими заявками из разных отделов, т.к. заявки могут быть различного характера от клиентов.

Для того, чтобы дать обоснованное решение о необходимости использования средств вычислительной техники, необходимо проанализировать трудозатраты сотрудников.

В процессе участвуют несколько документов: заявка от клиентов БиАйЭй-Технолоджиз и отчет по проделанной работе. Все документы формируются специалистами компании (рисунок 5).



Рисунок 5 - Схема документооборота по обработке заявок от клиентов

«Трудозатраты на осуществление документооборота процесса складываются из затрат специалистов ИТ-отдела на прием и распределение заявок, обработку заявок и составление отчетов о проделанной работе» [2]. Характеристика документооборота по существующей схеме представлена в таблице 5.

Таблица 5 -Характеристика формирования документооборота по обработке заявок от клиентов БиАйЭй-Технолоджиз

Характеристика процесса по обработке заявок от клиентов БиАйЭй-Технолоджиз	Заявка от клиентов БиАйЭй-Технолоджиз	Отчет о проделанной работе по обработке заявок от клиентов БиАйЭй-Технолоджиз
Количество заявок, поступающих от клиентов компании в год, шт.	1	1
Количество символов в заявке, поступившей от клиентов компании, шт.	1000	3000
Частота возникновения в год	10 000	12
Трудозатраты на обработку заявок, поступающих от клиентов компании в год, чел-час	7300	3000

Трудозатраты на формирование документооборота по существующей схеме составляют 10300 человеко-часа в год. С помощью средств вычислительной техники формирование отчетности будет автоматизировано. Это позволит снизить трудозатраты компании.

Характеристика документооборота после применения средств вычислительной техники представлена в таблице 6.

Таблица 6 - Характеристика формирования документооборота по обработке заявок от клиентов БиАйЭй-Технолоджиз после автоматизации

Характеристика процесса по обработке заявок от клиентов БиАйЭй-Технолоджиз	Заявка от клиентов БиАйЭй-Технолоджиз	Отчет о проделанной работе по обработке заявок от клиентов БиАйЭй-Технолоджиз
Количество заявок, поступающих от клиентов компании в год, шт.	1	5
Количество символов в заявке, поступившей от клиентов компании, шт.	1000	3000
Частота возникновения в год	10 000	12
Трудозатраты на обработку заявок, поступающих от клиентов компании в год, чел-час	6800	1000

После автоматизации процесса по обработке заявок от клиентов БиАйЭй-Технолоджиз трудозатраты составят 7800 человеко-часов. Снижение трудозатрат по обработке заявок от клиентов БиАйЭй-Технолоджиз на

формирование документооборота является достаточным основанием для автоматизации бизнес-процесса.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Готовые решения, представленные на рынке это - программа «Аналитика: Service Desk» от компании «1С» предназначена для повышения эффективности работы как IT-отдела, так и всех сотрудников предприятия [20]. В функциональные возможности программного продукта входят:

- оформление и обработка заявок в техническую поддержку от сотрудников и клиентов организации;
- ведение учета оборудования организации и учет инцидентов с ним;
- формирование базы знаний;
- настройка маршрутов обработки заявок и инцидентов.

Программный код подсистемы полностью открыт и может быть изменен для учета всех особенностей работы организации.

Система «hd.rstm» является «help-desk» веб-системой, предназначенной для организации учёта выполнения заявок между структурными подразделениями и пользователями или клиентами организации [1]. Система написана на языке программирования PHP.

Система «GLPI» не является «help-desk»-системой в классическом понимании этого слова. Она представляет собой систему для работы с заявками и инцидентами, система для инвентаризации компьютерного оборудования и ПО [19]. Система написана на языке программирования PHP.

К функциональным возможностям системы относятся:

- учет оборудования информационной инфраструктуры организации;
- распределение задач, планирование и т.д;
- назначение оборудования по географическим регионам для пользователей и групп пользователей;

- управление данными;
- управление работой оборудования;
- управление заявками;
- использование нескольких каналов передачи заявки;
- управление документооборотом;
- резервирование оборудования;
- ведение «базы знаний»;
- формирование отчетности;
- наличие множества плагинов для добавления дополнительных функций.

Для выбранных программных продуктов разработаем критерии, по которым будет проводиться оценка соответствия требованиям. Критериями выбора программного продукта будут являться:

- простота и удобство,
- безопасность,
- удобство навигации,
- юзабилити,
- функциональность.

Выбранные системы оценены по выделенным критериям по шкале, где «1» - полное несоответствие критерию, а «5» - полное соответствие. Оценка программных продуктов приведена в таблице 7.

Таблица 7- Анализ систем, представленных на рынке

Критерий	Аналитика: Service Desk	hd.rustem	GLPI
Простота управления	4	4	4
Безопасность	4	5	3
Удобство навигации	3	3	4
Юзабилити	3	4	3
Функциональность	2	1	3
Итого	16	17	17

В результате сделан вывод, что программное обеспечение, представленное на рынке, не соответствует поставленной задаче.

1.4 Постановка задачи на разработку проекта внедрения АИС обработки заявок от клиентов БиАйЭй-Технолоджиз

Определим требования к информационной системе для автоматизации обработки заявок от клиентов БиАйЭй-Технолоджиз в методологии FURPS+ [3] и отобразим их в таблице 7.

Таблица 7 – Требования к информационной системе обработки заявок от клиентов БиАйЭй-Технолоджиз в методологии FURPS+

Вид требований	Содержание требований
F. Функциональные	<ul style="list-style-type: none"> – поддержка отправки электронных сообщений, – средства для печати документов, – ввод данных, – поиск данных
U. Удобство использования	<ul style="list-style-type: none"> – эстетика и логичность пользовательского интерфейс, – справочная информация
R. Надежность	<ul style="list-style-type: none"> – верификация данных, использование справочников
P. Производительность	<ul style="list-style-type: none"> – «потребление ресурсов, определяется требованиями к персональному компьютеру сотрудника» [3]
S. Поддерживаемость	<ul style="list-style-type: none"> – тиражирование в филиалах/подразделениях БиАйЭй-Технолоджиз, – соответствие международным стандартам
+. Ограничения	<ul style="list-style-type: none"> – «хранение необходимо реализовать с помощью реляционной БД, – документация должны создаваться в MS Word, – форматы данных входных и выходных документов, должны поддерживать стандарты бухгалтерской отчетности БиАйЭй-Технолоджиз» [3]

После определение требований к разрабатываемой системе построена модель бизнес-процесса «как должно быть».

1.5 Разработка модели бизнес-процесса «как должно быть»

Предлагается к механизмам каждого бизнес-процесса по обработке заявок от клиентов БиАйЭй-Технолоджиз «добавить автоматизированную информационную систему и тогда процесс после реинжиниринга будет выглядеть следующим образом» [5] (рисунок 6).

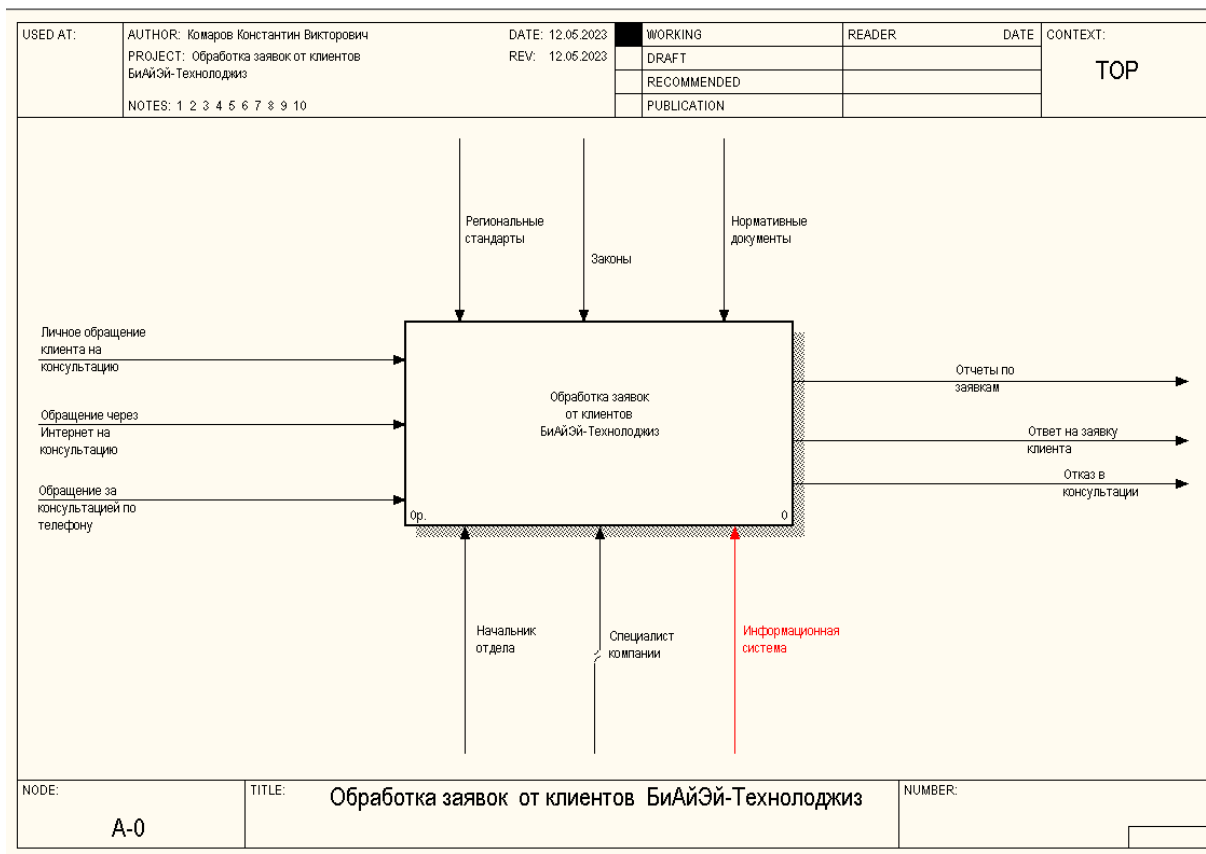


Рисунок 6 – Контекстная диаграмма «Обработка заявок от клиентов» «Как должно быть»

Бизнес-процесса «Обработки заявок от клиентов БиАйЭй-Технолоджиз» выглядит следующим образом после автоматизации (рисунок 7).

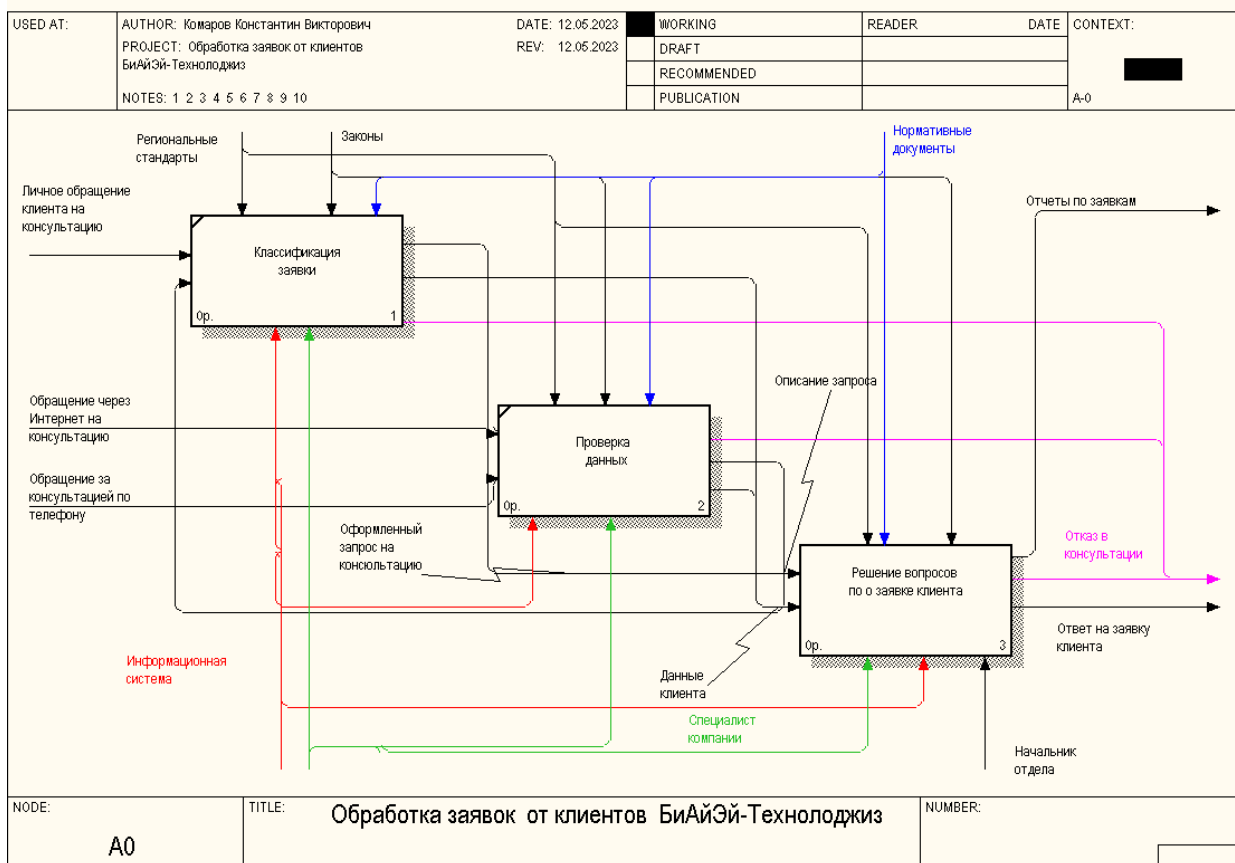


Рисунок 7 - Обработка заявок от клиентов БиАйЭй-Технолоджиз. «Как должно быть»

Бизнес-процесс обработка заявок от клиентов БиАйЭй-Технолоджиз более подробно после автоматизации выглядит как показано в таблице 8

Таблица 8 - Бизнес-процесс «Обработка заявок от клиентов БиАйЭй-Технолоджиз». «Как должно быть»

Название	Вход	Выход	Управление	Механизм
Классификация заявки	Личное Запрос на консультацию, поступивших от клиента клиента на консультацию, Описание запроса	Оформленный запрос на консультацию, Контактные данные и данные запроса, поступающих от клиента Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками Информационная система

Продолжение таблицы 8

Название	Вход	Выход	Управление	Механизм
Проверка данных от клиентов БиАйЭй-Технолоджиз	Запрос на консультацию, Интернет, Запрос на консультацию, по телефону	Описание запроса, Контактные данные и данные запроса, поступающих от клиента Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками Информационная система
Решение вопросов по заявке клиентов	Контактные данные и данные запроса, поступающих от клиента, Оформленный запрос на консультацию	Ответ на заявку клиентов , Отчеты по заявкам Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками, Начальник отдела Информационная система

После автоматизации бизнес-процесса «Решение вопросов по о заявке клиента» выглядеть следующим образом (рисунок 8).

Процесс «Решение вопросов по заявке клиента «Как должно быть»» (Рисунок 4) состоит из следующих блоков:

- прием заявки клиентов,
- прием заявки клиентов,
- обработка заявки клиентов,
- учет результатов обработки заявки клиентов.

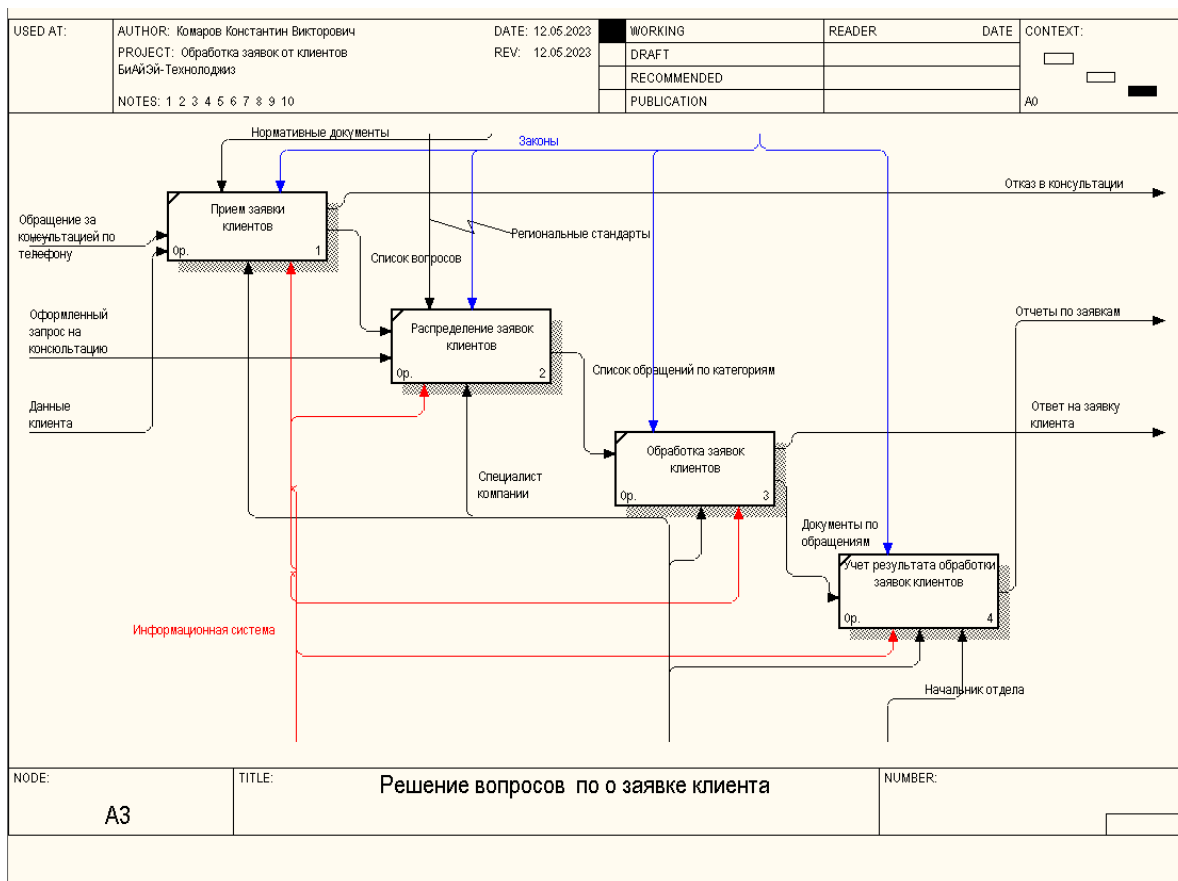


Рисунок 8 – Автоматизации процесса Решение вопросов по заявке клиента. «Как должно быть»

Рассмотрим операции бизнес-процесса «Решение вопросов по заявке клиента», «как должно быть» более подробно (Таблица 9)

Таблица 9 - Бизнес-процесс «Решение вопросов по заявке клиента «Как должно быть»

Название	Вход	Выход	Управление	Механизм
Прием заявки клиентов	Запрос на консультацию, поступивших от клиента по телефону, Контактные данные и данные запроса, поступающих от клиента	Список вопросов, Отказ в консультации	Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками Информационная система

Продолжение таблицы 9

Название	Вход	Выход	Управление	Механизм
Прием заявки клиентов	Список вопросов, Оформленный запрос на консультацию	Список заявки клиентов по категориям	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками Информационная система
Обработка заявки клиентов	Список заявки клиентов по категориям	Документы по заявкам, Ответ на заявку клиента	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками Информационная система
Учет результатов обработки заявки клиентов	Документы по заявкам	Отказ в консультации	Стандарты по работе с клиентами, принятые в регионе, Законы, Документы, регулирующие консультационные услуги	Специалист компании, отвечающих за работу с клиентскими заявками, Начальник отдела Информационная система

Введение данного механизма позволит упростить некоторые операции по обработке заявок от клиентов компании и автоматизировать распределение заявок от клиентов по специалистам.

Выводы по главе 1

В первой главе выпускной квалификационной работы проанализированы бизнес-процессы рассматриваемой предметной области.

Проведен анализ существующих разработок. Рассмотрены процессы компании, такие как «Решение вопросов по заявке клиента», «Обработка заявок от клиентов».

Приведено описание бизнес-процесса «Решение вопросов по заявке клиента «Как должно быть» в методологии SADT.

Определены требования к разрабатываемой системе по автоматизации обработки заявок от клиентов в методологии FURPS+.

Глава 2 Логическое проектирование информационной системы обработки заявок от клиентов БиАйЭй-Технолоджиз

2.1 Выбор технологии логического моделирования АИС

На этапе логического моделирования можно использовать несколько подходов к построению моделей. Но самой распространенной является технология UML [17].

«UML — это объектно-ориентированный язык со следующими характеристиками:

Основными понятиями языка UML являются:

- модель - объект, отображающий наиболее значимые для конкретной задачи характеристики системы. Модели бывают разные – нематериальные и материальные, естественные и искусственные, математические и декоративные;
- подсистема — показывает поведения других элементов;
- система - совокупность управляемых взаимосвязанных подсистем, которых объединили с общей целью;
- диаграмма - может быть диаграммой: блок-схема, и схемы монтажа оборудования, и дерево файлов и каталогов на диске и т.д. Рисунок воспринимается легче, чем текст» [12].

«С помощью диаграмм можно представить информационную систему с различных точек зрения. Набор диаграмм будет составлять модель системы и наиболее полно будет ее описывать» [11].

«CASE-средства (от Computer Aided Software/System Engineering) - позволяют проектировать любые системы на компьютере. Результат использования CASE-средств - оптимизация систем, снижение расходов, повышение эффективности, снижение вероятности ошибок.

Visio - решение для построения диаграмм от Microsoft. По словам разработчиков, Visio помогает преобразовать технические и бизнес-

концепции в визуальную форму. Пакет из семейства Microsoft Office предназначен исключительно для рисования диаграмм» [6].

2.2 Логическое проектирование автоматизированной информационной системы

В ходе проектирования и описания процесса моделирования предметной области по обработке заявок от клиентов БиАйЭй-Технолоджиз создана диаграмма вариантов использования.

«Диаграммы вариантов использования позволяют наглядно представить ожидаемое поведение системы» [7].

Все варианты использования системы по обработке заявок от клиентов БиАйЭй-Технолоджиз, так или иначе, связаны с требованиями к функциональности разрабатываемой системы и могут сильно различаться по объему выполняемой работы (Рисунок 9).

Пользователями системы по обработке заявок от клиентов БиАйЭй-Технолоджиз будут все сотрудники организации, поэтому у всех сотрудников организации должны быть права доступа к реализуемой системе по обработке заявок от клиентов БиАйЭй-Технолоджиз.

Специалисты ИТ-отдела осуществляют прием заявок от клиентов БиАйЭй-Технолоджиз, поэтому они должны иметь права на создание и редактирование заявок, а также на формирование отчетов.

Все изменения в системе должны протоколироваться для возможности выявления лица, нарушившего Политику безопасности.

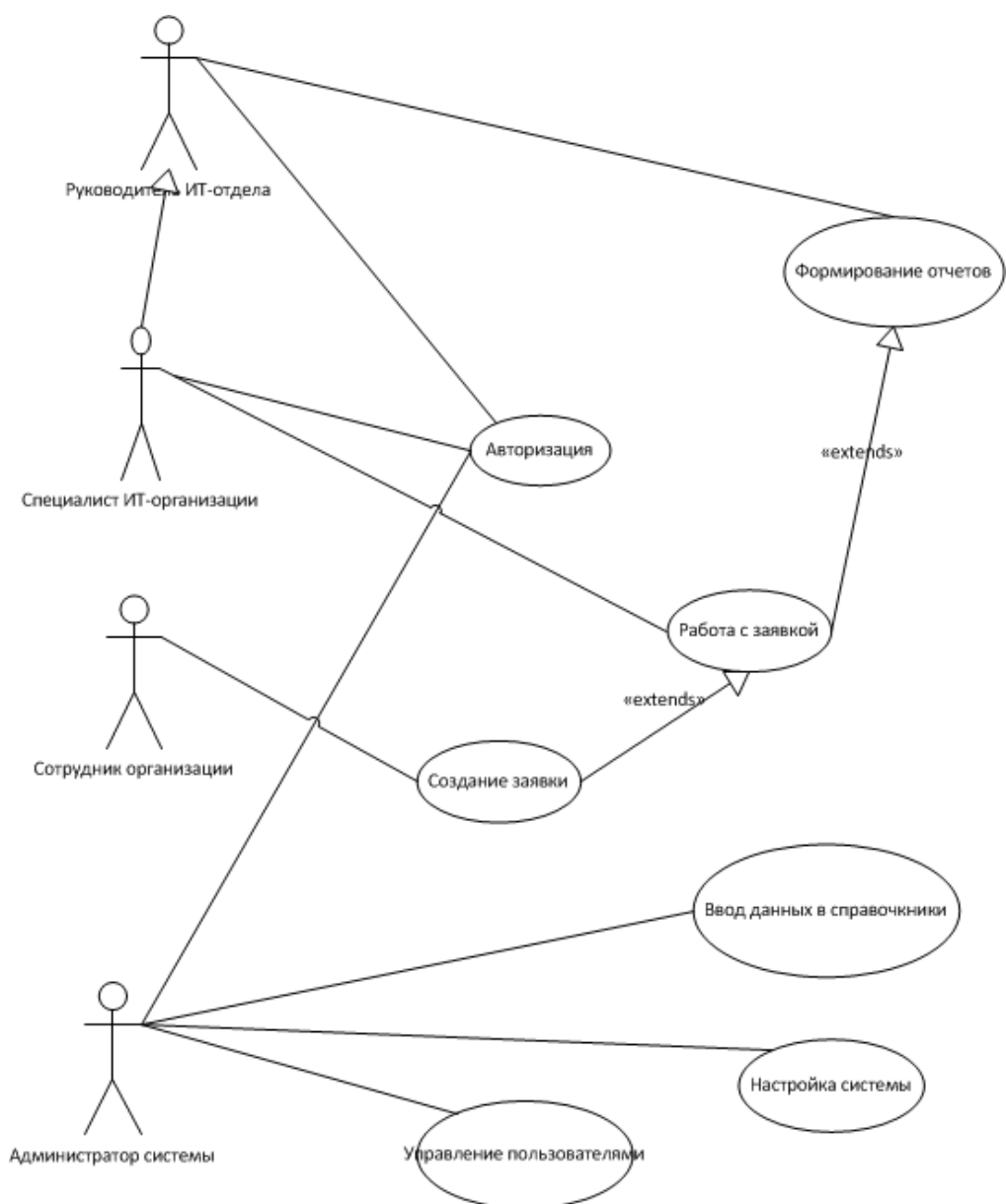


Рисунок 9 – Диаграмма вариантов использования информационной системы по обработке заявок от клиентов БиАйЭй-Технолоджиз

Специалист ИТ организация – это сотрудник ИТ –отдела, который обрабатывает заявки.

Сотрудник организации – это сотрудники других отделов, которые имеют право создавать заявки.

2.3 Проектирование базы данных автоматизированной информационной системы

При разработке информационных систем проект базы данных является тем фундаментом, на котором строится вся система в целом.

Информационная модель системы по обработке заявок от клиентов БиАйЭй-Технолоджиз представляет собой новый вариант организации информационной системы, автоматизирующей процесс по обработке заявок от клиентов БиАйЭй-Технолоджиз. Информационная модель системы по обработке заявок от клиентов БиАйЭй-Технолоджиз представлена на рисунке 10.

Входным документом проектируемой системы по обработке заявки клиентов компании БиАйЭй-Технолоджиз является заявка клиента компании за консультацией. Она будет содержать в себе следующий перечень данных:

- ФИО гражданина или данные компании;
- место консультации;
- раздел классификации проблемы;
- вид проблемы;
- описание проблемы.

Информационная система по обработке заявки клиентов компании БиАйЭй-Технолоджиз содержит формы заявки, формы отчетов и редактирования справочников. В результате работы системы сформируется отчет о проделанной работе.

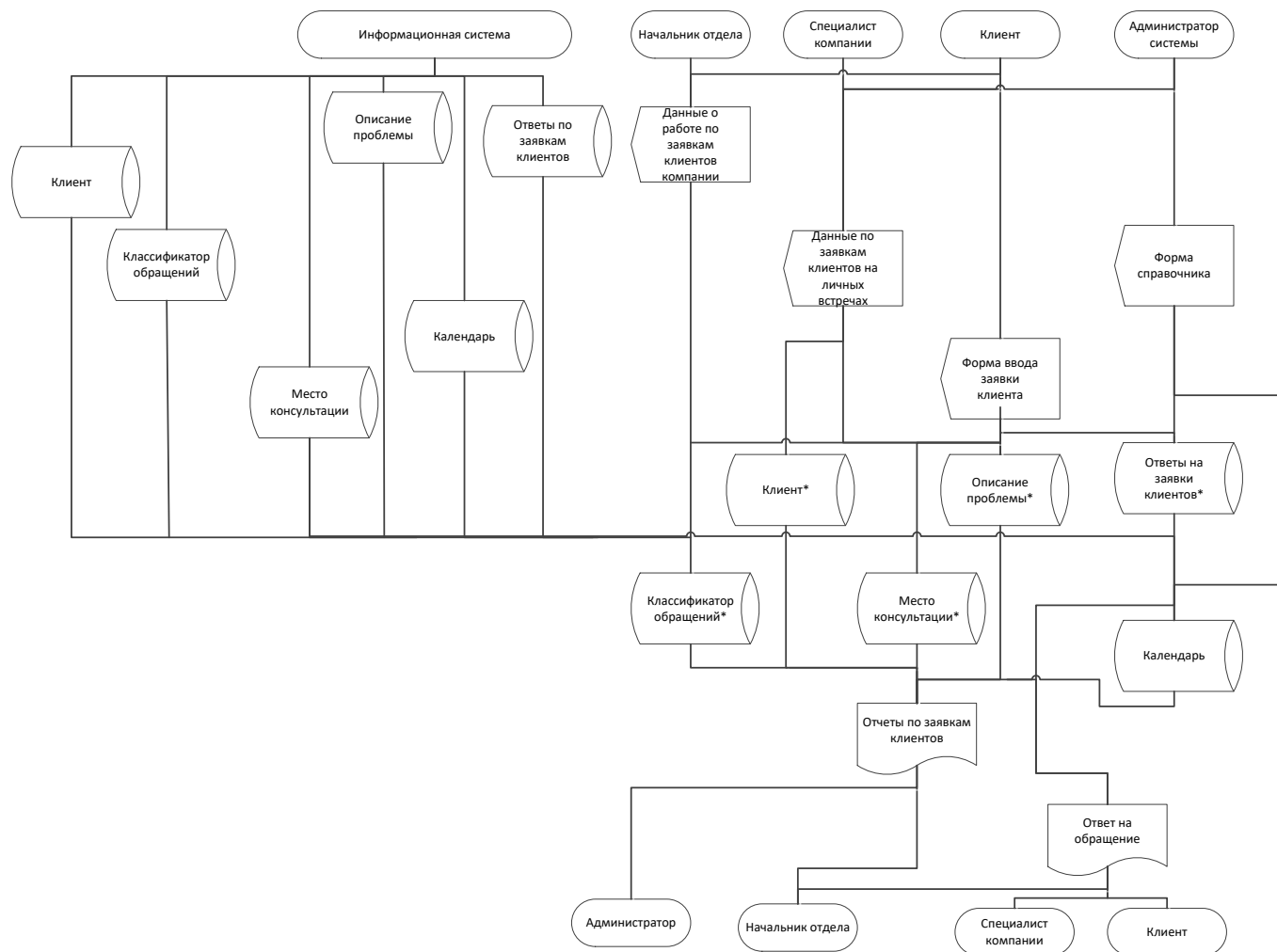


Рисунок 10 - Информационная модель

После окончания этапа логического проектирования информационной системы по обработке заявки клиентов компании БиАйЭй-Технолоджиз, была построена физическая модель базы данных, сгенерированная в СУБД SQLite, которая показана на рисунке 11. Связи на схеме показывают по какому полю идет связь с родительской таблицей.

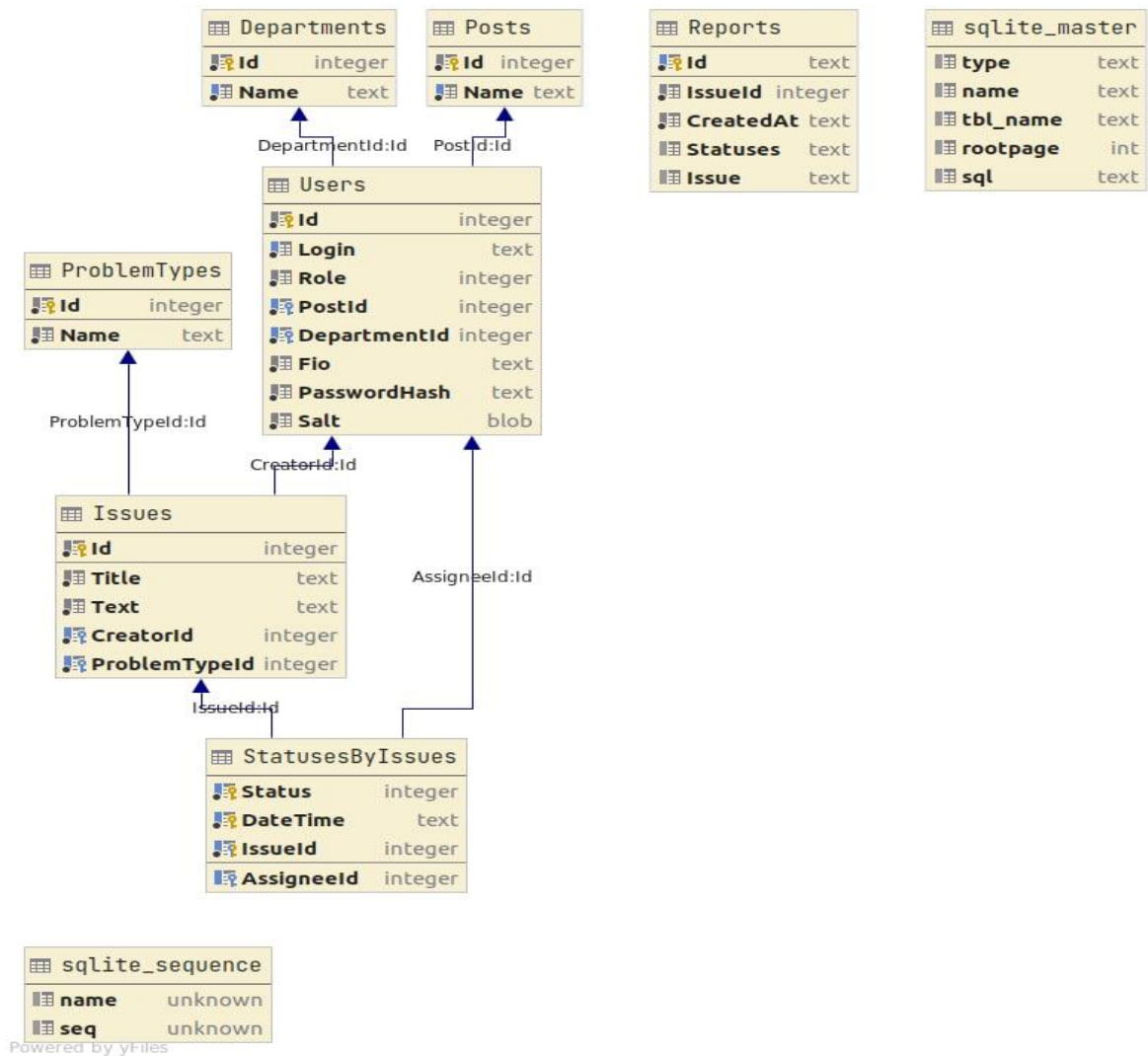


Рисунок 11 – Физическая модель базы данных

После окончания разработки базы данных разработан прототип информационной системы по обработке заявки.

Выводы по главе 2

Во второй главе были рассмотрены вопросы логического проектирования информационной системы по обработке заявки клиентов компании БиАйЭй-Технолоджиз.

Выбрана методология для проектирования информационной системы.

Приведена диаграмма вариантов использования, в которой отражены функции (7 функций) и основные действующие лица (4 актера).

Определена информационная модель, где отражены входные и выходные документы.

На физической схеме показаны сущности базы данных и атрибуты.

Физическая схема данных содержит 9 таблиц, которые отражают сущности решаемой проблемы, а также внутренние таблицы базы данных, которые необходимы для эффективной работы информационной системы.

Глава 3 Реализация проекта разработки информационной системы и оценка его эффективности

3.1 Архитектура и программное обеспечение информационной системы

В рамках рассматриваемой задачи, программный продукт реализован с помощью веб архитектуры, т.к. к такому виду программ, доступ осуществляется из любой точки.

«Для реализации приложения выбран стек технологий ASP.NET Core + Razor Pages + Entity Framework Core» [9]

«ASP.NET Core - современный, кроссплатформенный фреймворк для построения веб-приложения. Приложение построено на структуре

Основную логику при запросе к приложению выполняют контроллеры. Для каждой сущности используется собственный контроллер, по урлу определяется какой именно должен обработать запрос (данный функционал из коробки умеет делать ASP.NET Core, достаточно только указывать атрибут Route у контроллера). Контроллер получает данные в запросе (например, идентификатор заявки), получает необходимые данные из БД и возвращает сгенерированную HTML-страницу.

Для генерации HTML-кода на основе полученных данных используется шаблонизатор Razor. Это широкий по функционалу шаблонизатор для C#, который изначально поддерживается фреймворком ASP.NET Core. Он позволяет удобно описывать как должна выглядеть страница при помощи смеси HTML и C# кода» [9].

«Для создания интерфейса используется CSS фреймворк Bootstrap. Содержит набор компонент, из которых можно построить удобный интерфейс, адаптируемый для мобильных устройств»[13].

«Для работы с БД используется библиотека Entity Framework. Помимо подключения и отправки запросов, библиотека позволяет использовать ORM

- технологию для представления таблиц из базы данных в виде классов, а строк в виде объектов» [10].

Рассмотри взаимодействие созданных модулей. Для этого построим диаграмму развёртывания (рисунок 12).

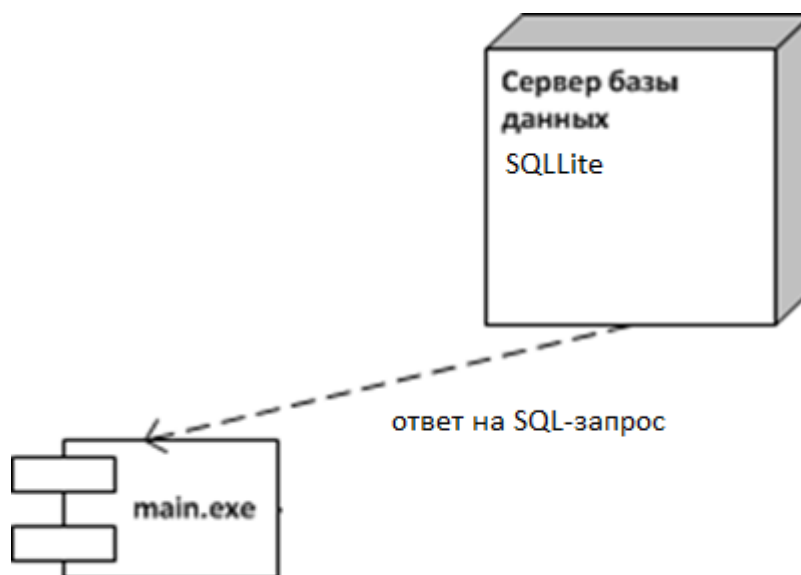


Рисунок 12 – Диаграмма развёртывания

«Диаграмма развёртывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развёртывания едина для системы в целом, поскольку должна всецело отражать особенности ее реализации» [8].

3.2 Описание функциональности информационной системы обработки заявок от клиентов БиАйЭй-Технолоджиз

Перечисленные функции, автоматизированы с помощью разрабатываемой системы их можно разделить на служебные и основные функции. Распределим перечисленные выше функции по этим категориям.

К служебным функциям системы по обработке заявок от клиентов компании относятся:

- функция контроля и разграничения доступа - авторизация;
- функция по настройке системы;
- функция для управление окнами;
- функция подсказки.

К основным функциям системы по обработке заявок от клиентов компании относятся:

- функция ввода данных;
- функция по доступу к данным;
- основная функция - создание заявки;
- функция для формирования отчетов.

В результате проделанной работы было создано дерево функций системы, представленное на рисунке 13.



Рисунок 13 - Дерево функций системы

В процессе разработки и тестирования контрольного примера разработаны тестовые данные, к которым относятся:

- данные справочников, которые представляют собой фамилии, имена, отчества, номера телефонов сотрудников организации, названия должностей и отделов, в которых сотрудники работают, перечень статусов заявок и видов проблем;
- данные о заявке: номер заявки (генерируется программой), дата создания заявки (задается программой), вид проблемы (выбирается пользователем из справочника), статус заявки (первоначально задается системой, в процессе обработки заявки задается сотрудником техподдержки), описание проблемы (текст) и решение проблемы (текст);
- данные о пользователях системы: фамилия, имя и отчество сотрудника организации (выбирается из справочника), логин (вводится пользователем), пароль (генерируется системой).

Обработка тестовых данных включает в себя ввод данных в формы системы и проверку корректности отображения введенных данных. Также нужно проверить данные, которые генерируются самой системой (даты, номера заявок, учетные данные пользователя). Результаты работы пользователя в системе должны также корректно отображаться на формах вывод результатной информации.

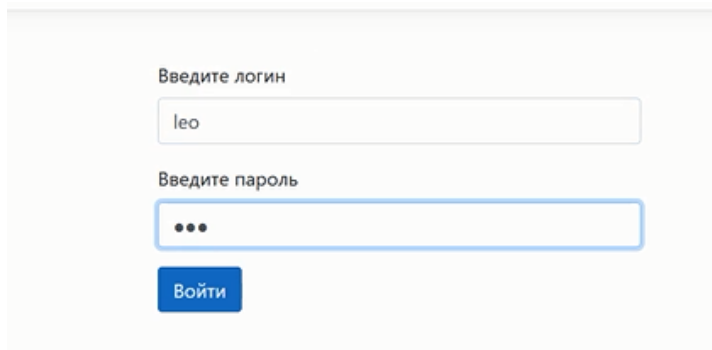
Далее описано реализация контрольного примера, которая проводится в несколько этапов:

- ввод тестовых данных в справочники;
- ввод тестовых данных в формы входной информации;
- результат формирования отчета.

Вход в систему возможен только авторизованным пользователям, поэтому при запуске системы открывается окно авторизации, представленное на рисунке 14. В программе предусмотрены три сценария работы, как и было

рассмотрено выше от лица: сотрудника организации, администратора системы и специалистов компании.

Далее приведена работа системы под пользователем «Администратор».



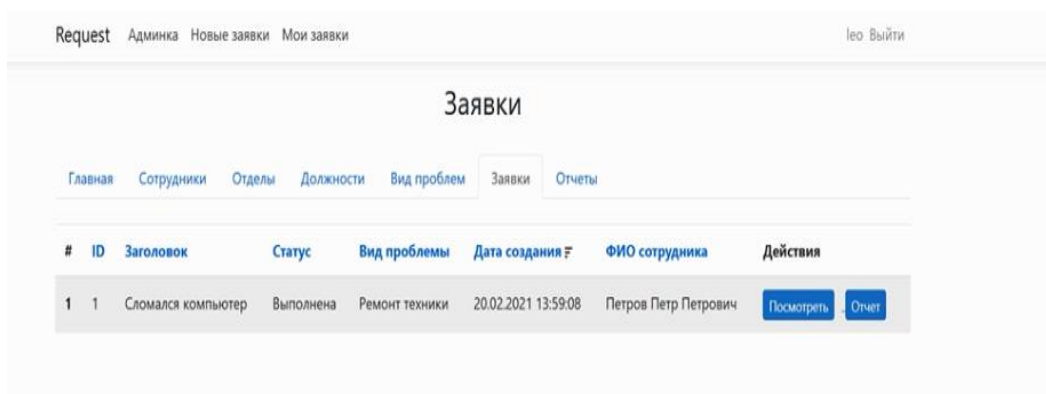
Введите логин

Введите пароль

[Войти](#)

Рисунок 14 – Работа приложения под администратором

Администратор может работать с созданием и поиском заявок. Главная форма приложения под работой «Администратор» представлена на рисунке 15.



Request Админка Новые заявки Мои заявки leo Выйти

Заявки

Главная Сотрудники Отделы Должности Вид проблем Заявки Отчеты

#	ID	Заголовок	Статус	Вид проблемы	Дата создания	ФИО сотрудника	Действия
1	1	Сломался компьютер	Выполнена	Ремонт техники	20.02.2021 13:59:08	Петров Петр Петрович	Посмотреть Отчет

Рисунок 15 - Главная форма приложения под пользователем «Администратор»

Администратор имеет возможность заполнять справочники. Далее показана работа с этой функцией. На рисунке 16 - 18 представлено заполнение справочника «Сотрудник».

Request Админка Новые заявки Мои заявки leo Выйти

Сотрудники

Главная Сотрудники Отделы Должности Вид проблем Заявки Отчеты

Добавить сотрудника

#	Логин	ФИО	Отдел	Должность	Роль	Действия
1	leo	Иванов Иван Иванович	Отдел технического контроля	Специалист	Администратор	Изменить Удалить
2	user	Петров Петр Петрович	Отдел технического контроля	Специалист	Сотрудник	Изменить Удалить
3	moder	Савельев Савелий Савельевич	Отдел технического контроля	Специалист	ИТ-сотрудник	Изменить Удалить

Рисунок 16 – Работа администратора со справочником Сотрудник в информационной системе по работе с заявками

Request Админка Новые заявки Мои заявки leo Выйти

Сотрудники

Главная Сотрудники Отделы Должности Вид проблем Заявки Отчеты

Изменить сотрудника

Логин для входа

ФИО сотрудника

Отдел

Должность

Роль

[Изменить](#)

Рисунок 17 – Изменение справочника «Сотрудник»

Request Админка Новые заявки Мои заявки leo Выйти

Сотрудники

Главная Сотрудники Отделы Должности Вид проблем Заявки Отчеты

Добавить нового сотрудника

Логин для входа

ФИО сотрудника

Пароль

Повторите пароль

Отдел

Должность

Роль

[Создать](#)

Рисунок 18 – Добавление нового сотрудника в справочник

Так же на рисунке 19 представлена функция администратора «Просмотр заявок клиентов».

#	ID	Заголовок	Статус	Вид проблемы	Дата создания	ФИО сотрудника	Действия
1	4	Принтер не печатает	Выполнена	Устройства	23.02.2021 19:15:02	Петров Петр Петрович	Посмотреть Отмет
2	3	Тормозит компьютер	Выполняется	ПК	23.02.2021 19:11:13	Барбара Артём Олегович	Посмотреть Отмет

Рисунок 19 – Просмотр заявок клиентов под учетной записью «Администратор»

Администратору доступна функция просмотра отчетов по заявкам (рисунок 20 -22)

Request Админка Заявки Мои заявки admin Выйти

Отчет по заявке №4 от 23.02.2021 20:31:51

Автор заявки: [Петров Петр Петрович](#)

Заявка "Принтер не печатает", проблема "Устройства"
Нажимаю "Печать" и не печатает.

История статусов:

- 23.02.2021 19:15:02
Статус: Новая
- 23.02.2021 19:17:43
Статус: Новая —> Выполняется
Исполнитель: [Иванов Иван Иванович](#)
- 23.02.2021 20:30:25
Статус: Выполняется —> Выполнена
Исполнитель: [Иванов Иван Иванович](#)

Рисунок 20 – Просмотр отчетов по заявкам

Request Заявки Мои заявки iii Выйти

#4 **Выполняется** Исполнитель: [Иванов Иван Иванович](#) Создал(а) [Петров Петр Петрович](#) 23.02.2021 19:15:02

Заявка №4 - Принтер не печатает

Проблема: Устройства

Нажимаю "Печать" и не печатает.

[Выполнена](#)

Рисунок 21 – Заявка в работе, как ее видит Администратор в информационной системе по работе с заявками

Request Админка Заявки Мои заявки admin Выйти

Отчеты

Главная Сотрудники Отделы Должности Вид проблем Заявки **Отчеты**

#	Заявка	Дата	Действия
1	[3] Тормозит компьютер	23.02.2021 20:32:11	Посмотреть Удалить
2	[4] Принтер не печатает	23.02.2021 20:31:51	Посмотреть Удалить

Рисунок 22 – Отчеты по заявкам

Рассмотрим еще один сценарий работы программы от лица пользователя. Вначале ему также необходимо зайти под своим логином и паролем (рисунок 23-24)

Введите логин

Введите пароль

[Войти](#)

Рисунок 23 – Вход под учетной записью «Пользователь» в информационной системе по работе с заявками

Пользователь может создать новую заявку с описанием проблемы или редактировать ранее созданную

Создать новую заявку

Заголовок

Вид проблемы

Не выбрано

Описание

[Создать](#)

Рисунок 24 – Создание заявки клиента под учетной записью «Пользователь»

Пользователь может просмотреть созданные свои заявки клиента (рисунок 25)

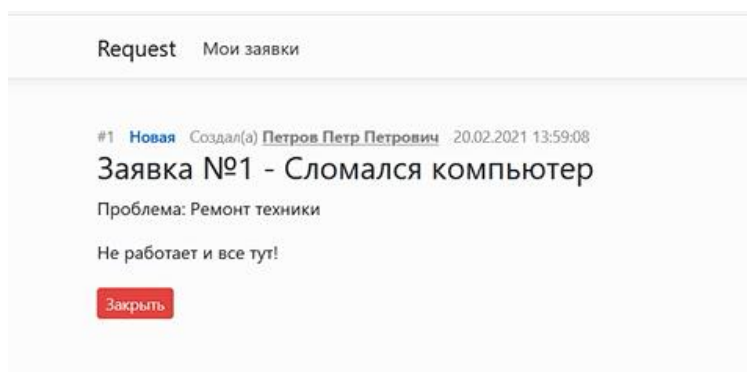


Рисунок 25 – Просмотр заявок пользователей под учетной записью «Пользователь»

Может удалить созданные свои заявки (рисунок 26)

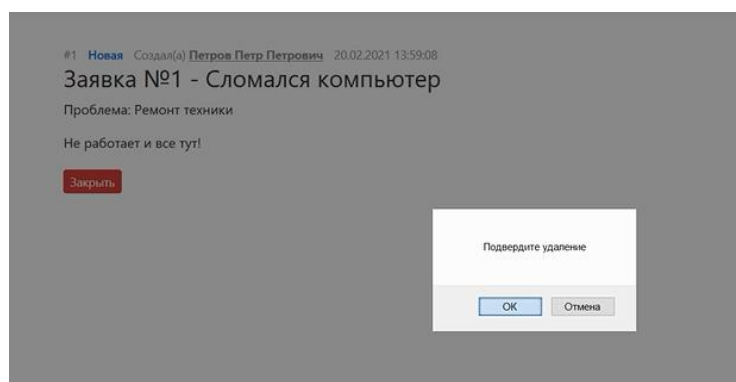


Рисунок 26 – Удаление клиентских заявок в информационной системе по работе с заявками

И рассмотрим последний сценарий работы от лица сотрудника ИТ-отдела. Сотрудника ИТ-отдела также должны авторизоваться в системе (рисунок 27).

Введите логин

Введите пароль

[Войти](#)

Рисунок 27 – Вход в систему под учетной записью специалиста ИТ-отдела

Специалист выбирает из списка заявки, которые требуют решения (рисунок 28)

Request [Новые заявки](#) [Мои заявки](#) moder [Выйти](#)

Список новых заявок

#	Заголовок	Вид проблемы	Дата создания	ФИО сотрудника	Действия
1	Сломался компьютер	Ремонт техники	20.02.2021 13:59:08	Петров Петр Петрович	Посмотреть

Рисунок 28 – Список открытых заявок

Далее специалиста ИТ-отдела может ее взять в работу (рисунок 29)

Request [Новые заявки](#) [Мои заявки](#) moder [Выйти](#)

#1 **Новая** Создал(а) [Петров Петр Петрович](#) 20.02.2021 13:59:08

Заявка №1 - Сломался компьютер

Проблема: Ремонт техники

Не работает и все тут!

[Взять в работу](#)

Рисунок 29 – Взять в работу заявку

После выполнения специалиста ИТ-отдела меняет статус заявки (рисунок 30).

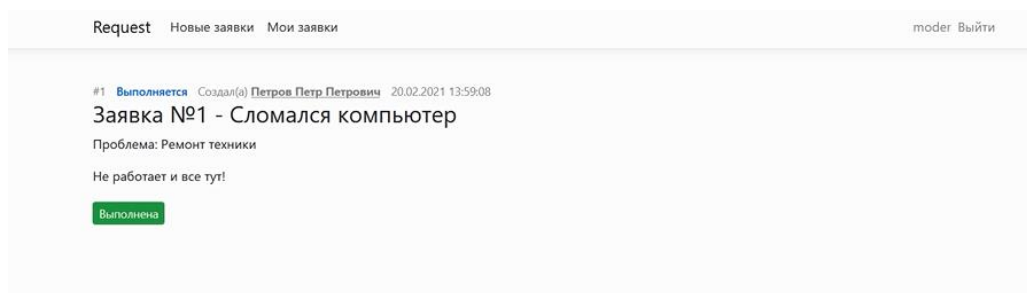


Рисунок 30 – Выполнить заявку

После выполнения заявки список заявок обновляется и выпавшие заявки из него удаляются (рисунок 31).

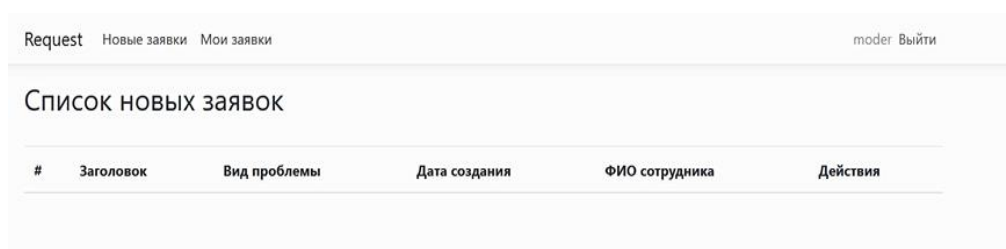


Рисунок 31 – Обновления списка заявок видимость интерфейса под учетной записью специалиста ИТ-отдела

После окончания описания прототипа системы, перейдем к оценке экономической эффективности.

3.3 Оценка и обоснование экономической эффективности внедрения информационной системы

При появлении на предприятии необходимости автоматизации каких-либо бизнес-процессов, проектирования и внедрения какой-либо системы, возникает потребность в расчете экономической эффективности планируемого проекта. Любое обновление имеющегося или внедрение нового программного продукта требует немалых финансовых вложений. По этой причине перед началом реализации проекта необходимо рассчитать затраты

на его проектирование, программирование отладку и внедрение. А также проанализировать экономическую эффективность проекта.

Суть обоснования экономической эффективности кроется в сравнении существующего технологического процесса и внедряемого. Другими словами, базового и проектируемого варианта решения задачи.

Обоснование экономической эффективности позволяет увидеть, насколько вложения в разработку новых решений или доработку уже имеющихся являются полезными.

Экономическая эффективность строится из двух основных составляющих, косвенного и прямого эффекта. Прямой экономический эффект, еще называемый сравнительным анализом, оценивается денежным выражением и, как правило, характеризуется понижением затрат на обработку информации, уменьшением трудовых затрат и увеличением производительности» [11].

Далее будут рассмотрены показатели, относящиеся к трудовым:

«Абсолютное снижение трудовых затрат (ΔT) в часах за год:

$$\Delta T = T_0 - T_1 \quad (1)$$

где:

T_0 – трудовые затраты на обработку информации по базовому (до автоматизации) варианту в часах за год;

T_1 – трудовые затраты на обработку информации по разрабатываемому варианту в часах за год» [3].

«Коэффициент относительного снижения трудовых затрат, иными словами, повышение производительности труда (K_T) в %:

$$K_T = \left(\frac{\Delta T}{T_0} \right) \times 100\% \quad (2)$$

Индекс снижения трудовых затрат (Y_T):

$$Y_T = \frac{T_0}{T_1} \quad (3)$$

Коэффициенты K_T и Y_T характеризуют повышение производительности труда за счет внедрения более экономичного варианта проектных решений» [4].

К стоимостным показателям относят следующие:

«Абсолютное снижение стоимостных затрат (ΔC) в рублях за год:

$$\Delta C = C_0 - C_1 \quad (4)$$

где:

C_0 – стоимостные затраты на обработку информации по базовому (до автоматизации) варианту в рублях за год,

C_1 – стоимостные затраты на обработку информации по проектируемому варианту в рублях за год» [5].

«Коэффициент относительного снижения стоимостных затрат (K_C) в %:

$$K_C = \left(\frac{\Delta C}{C_0} \right) \times 100\% \quad (5)$$

Индекс снижения стоимостных затрат (Y_C):

$$Y_C = \frac{C_0}{C_1} \quad (6)$$

Для проведения анализа эффективности проекта используются обобщающие и частные показатели.

К основным обобщающим показателям экономической эффективности относятся:

- годовой экономический эффект в рублях;
- расчетный коэффициент эффективности капитальных вложений;
- срок окупаемости системы» [6].

«Годовой экономический эффект от внедрения проекта (\mathcal{E}) (в рублях в год) определяется, как разность между годовой экономией и нормативной прибылью и рассчитывается по представленной формуле:

$$\mathcal{E} = \Delta C - K_{\Pi} \times E_{\Pi} \quad (7)$$

где:

K_{Π} – единовременные затраты или капитальные вложения, тыс. руб.,

E_n нормативный коэффициент эффективности капитальных вложений.

К капитальным затратам (K_n) относятся затраты на проектирование, программирование комплекса задач, а также затраты на отладку и внедрение программ» [7].

Осуществим расчет показателей экономической эффективности проекта автоматизации процесса по обработке заявок от клиентов БиАйЭй-Технолоджи. Характеристика затрат по базовому варианту по обработке заявок от клиентов БиАйЭй-Технолоджи на обработку по базовому плану составили 3141000 рублей.

«После внедрения информационной системы, автоматизирующей процесс обработки заявок от клиентов компании, в процесс обработки заявок будут внесены изменения. Например, появится возможность назначения специалистов на проведение заявок и формирования отчетности» [11].

Характеристика затрат по проектному варианту по обработке заявок от клиентов БиАйЭй-Технолоджи на учет приема и анализа заявок составили 2379000 рублей.

Рассчитаем прямой эффект от разработки системы по обработке заявок от клиентов БиАйЭй-Технолоджи, который включает в себя расчет трудовых и стоимостных показателей.

«Абсолютное снижение трудовых затрат в год в часах составило:

$$\Delta T = 10300 - 7800 = 2500 \text{ часов.}$$

Рассчитаем коэффициент относительного снижения трудовых затрат:

$$K_T = 2500 / 10300 * 100\% = 24\%.$$

Рассчитаем индекс снижения трудовых затрат или повышение производительности труда:

$$Y_T = 10300 / 7800 = 1,32 \text{» [7]}$$

«Осуществим расчет стоимостных показателей. Абсолютное снижение стоимостных затрат в год составит:

$$\Delta C = 3141000 - 2379000 = 762000 \text{ рублей» [14].}$$

«Коэффициент относительного снижения стоимостных затрат составит:

$$K_C = 762000 / 3141000 * 100\% = 25\% \gg [15].$$

Индекс снижения стоимостных затрат составит:

$$Y_C = 3141000 / 2379000 = 1,33 \gg [8].$$

Динамика трудовых затрат представлена на рисунке 32.



Рисунок 32- График снижения трудовых затрат по обработке заявок от клиентов БиАйЭй-Технолоджи

Динамика стоимостных затрат представлена на рисунке 33



Рисунок 33 - График снижения стоимостных затрат по обработке заявок от клиентов БиАйЭй-Технолоджи

«Произведем расчет срока окупаемости проекта. Для этого необходимо оценить затраты на разработку и внедрение информационной системы.

Далее рассчитаем единовременные капитальные затраты на разработку и внедрение системы (K_{II}). Затраты на разработку и внедрение системы рассчитываются по формуле:

$$K_{II} = C_{\text{проект}} + C_{\text{прогр}} + C_{\text{внедр+отладка}} + C_{\text{доп}}$$

$C_{\text{проект}} = 150\,070$ рублей.

$C_{\text{прогр}} = 341\,360$ рублей.

$C_{\text{внедр}} = 119\,840$ рублей.

$C_{\text{доп}} = 88\,530$ рублей.

$K_{II} = 699\,800$ рублей» [16].

«Рассчитаем срок окупаемости проекта:

$$T_{\text{ок}} = 699\,800 / 762\,000 = 0,9 \text{ лет}» [18]$$

В результате получилось, что трудовые затраты снизятся в год на 2500 часов, что составит 24 %. Стоимостные затраты составят 762 000 рублей или 25 %. Стоимостные затраты по проектному варианту ниже, чем по базовому в 1,32 раз, а трудовые затраты по проектному варианту ниже базовых в 1,33 раз. Срок окупаемости проекта составил 0,9 лет.

Выводы по главе 3

В третьей главе представлен контрольный пример тестирования информационной системы и оценена экономическая эффективность проекта. Срок окупаемости проекта составит 0,9 лет, а также описан прототип системы по обработке заявок от клиентов БиАйЭй-Технолоджи

Заключение

Целью настоящей работы было автоматизировать бизнес-процесс обработки заявок от клиентов.

В первой главе выпускной квалификационной работы «Анализ предметной области БиАйЭй-Технолоджиз» была изучена деятельность компании, а именно описана структура управления, описана деятельность каждого отдела компании.

Далее подробно рассмотрены бизнес-процессы по обработке заявок от клиентов БиАйЭй-Технолоджи, выделены слабые места и рассчитана эффективность автоматизации процесса, а именно после автоматизации процесса по обработке заявок от клиентов БиАйЭй-Технолоджиз трудозатраты составят 7800 человеко-часов. Снижение трудозатрат по обработке заявок от клиентов БиАйЭй-Технолоджиз на формирование документооборота является достаточным основанием для автоматизации бизнес-процесса.

Во второй главе выпускной квалификационной работы «Логическое проектирование информационной системы обработки заявок от клиентов БиАйЭй-Технолоджиз» проведено проектирование архитектуры решения. Выбрана методология для проектирования информационной системы. Приведена диаграмма вариантов использования, в которой отражены функции (7 функций) и основные действующие лица (4 актера). Определена информационная модель, где отражены входные и выходные документы.

На физической схеме показаны сущности базы данных и атрибуты. Физическая схема данных содержит 9 таблиц, которые отражают сущности решаемой проблемы, а также внутренние таблицы базы данных, которые необходимы для эффективной работы информационной системы.

В третьей главе выпускной квалификационной работы «Реализация проекта разработки информационной системы и оценка его эффективности» показана реализация прототипа системы, а также проведено тестирование

прототипа по трем сценариям работы, которая показала работоспособность всех модулей и сценариев работы.

Рассчитана экономическая часть проекта. На основании проведенной работы были получены следующие практические результаты: разработан прототип информационной системы по обработке заявок, внедрение которого позволяет снизить трудовые затраты сотрудников на 2500 часов в год.

Таким образом, задачи решены в полном объеме, цель разработка информационной системы обработки заявок от клиентов БиАйЭй-Технолоджи – достигнута.

Теоретическая значимость выпускной квалификационной работы будет в систематизации полученных теоретических знаний

Практическая значимость работы состоит в разработке системы.

Результаты работы могут быть успешно внедрены в любую компанию, где протекают похожие бизнес-процессы по обработке заявок от клиентов.

Список используемой литературы и используемых источников

1. Bruce Silver Bpmn Method and Style: A Levels-Based Methodology for Bpm Process Modeling and Improvement Using Bpmn 2.0 - Cody-Cassidy Press – 2017.
2. Jakob Freund, Bernd Rucker Real-Life BPMN: Using BPMN 2.0 to Analyze, Improve, and Automate Processes in Your Company – CreateSpace – 2016
3. Joseph M. Hellerstein, Michael Stonebraker, James Hamilton Architecture of a Database System – Hanover, USA 2020.
4. Korotkevitch Dmitri Pro SQL Server Internals - M. Apress – 2019.
5. Балдин, К.В. Информационные системы в экономике: учебник для студентов высших учебных заведений. Том 1 / К.В. Балдин. - М.: ИЦ Академия, 2018. 125 с.
6. Балдин, К.В. Информационные системы в экономике: учебник для студентов высших учебных заведений. Том 2 / К.В. Балдин. - М.: ИЦ Академия, 2018.
7. Граничин О., Кияев В. Информационные технологии в управлении предприятием / Граничин О., Кияев В – М.Интуит – 2018
8. Дэвид А.Марка, Клемент МакГоуэн Методология структурного анализа и проектирования SADT - McGraw-Hill Companies – 2019.
9. Матяш С. А. Корпоративные информационные системы/С.А.Матяш – М.-Берлин:Директ-Медиа, 2018.-471 с.
10. Мкртычев С.В., Гущина О.М., Очеповский А.В. Прикладная информатика. Бакалаврская работа [Электронный ресурс] : электрон. учеб-метод. пособие. Тольятти. ТГУ: Изд-во ТГУ, 2019. URL: <https://dspace.tltsu.ru/handle/123456789/8868> (дата заявки: 18.05.2023).
11. Никитаева А. Ю. Корпоративные информационные системы: учебное пособие / Никитаева А.Ю. - Таганрог: Южный федеральный университет, 2019. - 149 с.
12. Похилько, А.Ф. CASE-технология моделирования процессов с

использованием средств ВРWin и ERWin: учебное пособие / А.Ф. Похилько, И.В. Горбачев. - Ульяновск: УлГТУ, 2016. - 120 с.

13. Проектирование современных баз данных: учебно-методическое пособие / Дадян Э.Г. - М.:НИЦ ИНФРА-М, 2017. - 120 с.

14. Свод знаний по управлению бизнес-процессами. BPM СВОК 3.0: учебное пособие / Под ред. Белайчук А.А. - М.:Альпина Пабл., 2016. - 480 с

15. Симдянов И. В., Программирование. Ступени успешной карьеры. / Симдянов И. В., Кузнецов М. В. // - БХВ-Петербург, 2016. – 320 с.

16. Учитесь видеть бизнес-процессы: Практика построения карт потоков создания ценности : учебное пособие / Ротер М., Шук Д., Муравьева Г., - 4-е изд. - М.:Альп. Бизнес Букс, 2019.

17. ВрWin [Электронный ресурс] URL: <http://habrahabr.ru/>.(дата заявки: 12.05.2023)

18. Visio 2010: руководство для начинающих [Электронный ресурс]. URL: support.office.com (дата заявки: 12.05.2023)

19. Сайт компании БиАйЭй-Технолоджиз [Электронный ресурс]. URL: <https://bia-tech.ru/> (дата заявки: 12.05.2023)

20. Сайт бухгалтерской и налоговой отчетности о компании БиАйЭй-Технолоджиз <https://sbis.ru/contragents/7810385714/781001001> [Электронный ресурс]. URL: <https://bia-tech.ru/> (дата заявки: 12.05.2023)

Приложение А

Отчет по проектированию

Report for Diagram: A0, Обработка заявок от клиентов БиАйЭй-Технолоджиз

Activity Name: Классификация заявки
Object Type: Activity
Activity Status: WORK
Activity Number: A1

Activity Name: Проверка данных
Object Type: Activity
Activity Status: WORK
Activity Number: A2

Activity Name: Решение вопросов по о заявке клиента
Object Type: Activity
Activity Status: WORK
Activity Number: A3

Link Name: Законы
Link Status: WORK

Link Name: Документы, регулирующие консультационные услуги
Link Status: WORK
Link Name: Стандарты по работе с клиентами, принятые в регионе
Link Status: WORK
Link Name: Обращение за консультацией по телефону
Link Status: WORK
Link Name: Обращение через Интернет на консультацию
Link Status: WORK

Link Name: Описание запроса
Link Status: WORK

Link Name: Оформленный запрос на консультацию
Link Status: WORK

Link Name: Отказ в консультации
Link Status: WORK

Link Name: Личное обращение клиента на консультацию
Link Status: WORK

Link Name: Начальник отдела
Link Status: WORK

Link Name: Специалист компании, отвечающих за работу с клиентскими заявками
Link Status: WORK

Продолжение приложение А

Link Name: Отчеты по заявкам

Link Status: WORK

Link Name: Ответ на заявку клиента

Object Type: Activity

Activity Status: WORK

Activity Number: A31

Activity Name: Распределение заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A32

Activity Name: Обработка заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A33

Activity Name: Учет результата обработки заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A34

Link Name: Законы

Link Status: WORK

Activity Name: Учет результата обработки заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A34

Link Name: Законы

Link Status: WORK

Link Name: Документы, регулирующие консультационные услуги

Link Status: WORK

Link Name: Стандарты по работе с клиентами, принятые в регионе

Link Status: WORK

Link Name: Обращение за консультацией по телефону

Link Status: WORK

Link Name: Оформленный запрос на консультацию

Link Status: WORK

Link Name: Документы по обращениям

Продолжение приложение А

Link Status: WORK

Link Name: Список обращений по категориям

Link Status: WORK

Link Name: Список вопросов

Link Status: WORK

Link Name: Отказ в консультации

Link Status: WORK

Link Name: Начальник отдела

Link Status: WORK

Link Name: Специалист компании, отвечающих за работу с клиентскими заявками

Link Status: WORK

Link Name: Отчеты по заявкам

Link Status: WORK

Link Name: Ответ на заявку клиента

Link Status: WORK

Link Name: Контактные данные и данные запроса, поступающих от клиента

Link Status: WORK

Link Name: Информационная система

Link Status: WORK

Activity Name: Прием заявки клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A31

Activity Name: Распределение заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A32

Activity Name: Обработка заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A33

Activity Name: Учет результата обработки заявок клиентов

Object Type: Activity

Activity Status: WORK

Activity Number: A34

Приложение Б
Код программы

```
class Task
{
    private $conn;
    private $table_name = "tasks";

    public $id;
    public $name;
    public $project_id;
    public $priority;
    public $creator_id;
    public $executor;
    public $description;
    public $status;
    public $date_created;
    private $where;

    public function __construct($db)
    {
        $this->conn = $db;
    }

    public function read()
    {
        $query = "SELECT t.*,p.name as pname,p.priority,p.status as pstatus,
p.description,p.id as pid FROM"
            . $this->table_name .
            " t inner join project_list p on p.id = t.project_id" . $where . "order by p.name asc
            u inner join users p on p.executor = u.id OR p.creator_id=u.id" . $where . """;

        // подготовка запроса
        $stmt = $this->conn->prepare($query);

        // выполняем запрос
        $stmt->execute();

        return $stmt;
    }

    function create()
    {
        // запрос для вставки (создания) записей
        $query = "INSERT INTO
            " . $this->table_name . "
            SET
                name=:name,          project_id=:project_id,          description=:description,
                creator_id=:creator_id,executor=:executor,priority=:priority, status=:status";
    }
}
```

Продолжение приложение Б

```
// подготовка запроса
$stmt = $this->conn->prepare($query);

// очистка
$this->executor = htmlspecialchars(strip_tags($this->executor));
$this->priority = htmlspecialchars(strip_tags($this->priority));
$this->status = htmlspecialchars(strip_tags($this->status));

// привязка значений
$stmt->bindParam(":description", $this->description);
$stmt->bindParam(":creator_id", $this->creator_id);
$stmt->bindParam(":executor", $this->executor);
$stmt->bindParam(":priority", $this->priority);
$stmt->bindParam(":status", $this->status);

// выполняем запрос
if ($stmt->execute()) {
    return true;
}

return false;
}

function readOne()
{
    $query = "SELECT t.*,p.name as pname,p.priority,p.status as pstatus,
p.description,p.id as pid FROM"
        . $this->table_name .
        " t inner join project_list p on p.id = t.project_id" . $where . "order by p.name asc
u inner join users p on p.executor = u.id OR p.creator_id=u.id" . $where . " LIMIT
1";

    $this->description = $row['description'];
    $this->creator_id = $row['creator_id'];
    $this->executor = $row['executor'];
    $this->priority = $row['priority'];
    $this->status = $row['status'];
}

function update()
{
    " . $this->table_name . "
    SET
        name=:name,          project_id=:project_id,          description=:description,
creator_id=:creator_id,executor=:executor,priority=:priority, status=:status
    WHERE
        id = :id";
}
```


Продолжение приложение Б

```
// подготовка запроса
$stmt = $this->conn->prepare($query);

// очистка
$this->executor = htmlspecialchars(strip_tags($this->executor));
$this->priority = htmlspecialchars(strip_tags($this->priority));
$this->status = htmlspecialchars(strip_tags($this->status));
$this->id = htmlspecialchars(strip_tags($this->id));

$stmt->bindParam(':creator_id', $this->creator_id);
$stmt->bindParam(':executor', $this->executor);
$stmt->bindParam(':priority', $this->priority);
$stmt->bindParam(':status', $this->status);
$stmt->bindParam('id', $this->id);

// выполняем запрос
if ($stmt->execute()) {
    return true;
}

return false;
}

}

<v-card-text :id="'task-list__' + col.value" class="task-list" :data-
status="col.value">
    <template v-for="task in getTaskByStatus(col.value)">
        <div :key="task.id" :data-id="task.id" class="task-list__item">
            <v-list-item :value="task">
                <v-list-item-avatar>
                    <v-icon v-if="!task.owner">mdi-account-circle</v-icon>
                    <figure class="c-avatar">
                        <slot name="top"></slot>
                        
                        <div v-else class="c-avatar__initial" :style="charStyles">
                            {{ username.charAt(0).toUpperCase() }}
                        </div>
                    <span v-
if="showUnread" :class="alertClasses" :style="alertIndicatorStyles">
                        <span v-show="unread > 0">{{ unread }}</span>
                    </span>
                    <v-icon v-if="icon" size="18" class="c-avatar__icon">{{ icon }}</v-
icon>
                    <span v-else-
if="status" :class="statusClasses" :style="indicatorStyles"> </span>
                </figure>
            </v-list-item-avatar>
            <v-list-item-content>
                <v-list-item-title>#{{ task.id }} {{ task.name }}</v-list-item-title>
            </v-list-item-content>
        </div>
    </template>
</v-card-text>
```

Продолжение приложение Б

```

        <v-list-item-subtitle>{{ task.description }}</v-list-item-subtitle>
    </v-list-item-content>
    <v-list-item-action>
        <v-icon small @click.stop="handleEditItem(task)">mdi-pencil</v-icon>
    </v-list-item-action>
    </v-list-item>
</div>
</template>
</v-card-text>
</v-card>
</v-col>
</template>
</v-row>
</v-container>
<v-dialog v-model="showDialog" scrollable width="840">
    <v-toolbar flat>
        <v-spacer></v-spacer>
        <template v-for="task in getTaskByStatus(col.value)">
            <div :key="task.id" :data-id="task.id" class="task-list__item">
                <v-list-item :value="task">
                    <v-list-item-avatar>
                        <v-icon v-if="!task.owner">mdi-account-circle</v-icon>
                        <figure class="c-avatar">
                            <slot name="top"></slot>
                            
                            <div v-else class="c-avatar__initial" :style="charStyles">
                                {{ username.charAt(0).toUpperCase() }}
                            </div>
                            <span v-
if="showUnread" :class="alertClasses" :style="alertIndicatorStyles">
                                <span v-show="unread > 0">{{ unread }}</span>
                            </span>
                        </v-list-item-avatar>
                    </v-list-item>
                </template>
            </div>
        </v-for>
    </v-toolbar>
    <v-card>
        <v-card-text>
            <task-form :task="task" :col="col" :update="update" :delete="delete" />
        </v-card-text>
    </v-card>
</v-dialog>
</template>
</script>
import { mapGetters } from 'vuex'
import CAvatar from '@/components/avatar/CAvatar'
import TaskForm from '@/components/form/TaskForm'
import Sortable from 'sortablejs'
import TooltipMixin from '@/mixins/Tooltip'
import fullNameMixin from '@/mixins/executorName'
export default {
    name: 'KanBoard',
    components: {
        CAvatar,
        TaskForm,
    },
    mixins: [TooltipMixin, fullNameMixin],
    data() {

```

```
return {
  dialog: false,
  dialogDelete: false,
  editedIndex: -1,
  preview: false,
  editedItem: {
    task: "",
    description: "",
    executor: 0,
    status: 0,
    priority: 0,
  },
  defaultItem: {
    task: "",
    description: "",
    executor: 0,
    status: 0,
    priority: 0,
  },
  src: {
    type: [String, null],
  },
  size: {
    type: [Number, String],
    default: 48,
  },
  status: {
    type: String,
  },
  icon: {
    type: String,
  },
  username: {
    type: String,
    default: "",
  },
  color: {
    type: [String, Number],
    default: '#2196f3',
  },
  unread: Number,
  project_id: 1,
  showDialog: false,
  loading: false,
  items: [],
  selectedItem: null,
  actions: [
    {
```

```

    text: 'Edit Item',
    icon: 'mdi-pencil',
    click: this.handleEditItem,
  },
  {
    text: 'Delete Item',
    icon: 'mdi-close',
    click: this.handleDeleteItem,
  },
],
}
},
computed: {
  ...mapGetters(['getTaskStatus', 'getProjectList']),
  showUnread() {
    return this.unread > 0 && !this.$route.path.includes(this.publicId)
  },

  imgStyles() {
    return {
      height: convertToUnit(this.size),
      minWidth: convertToUnit(this.size),
      width: convertToUnit(this.size),
    }
  },

  charStyles() {
    return {
      height: convertToUnit(this.size),
      minWidth: convertToUnit(this.size),
      width: convertToUnit(this.size),
      'background-color': this.color,
    }
  },

  indicatorStyles() {
    const size = this.size / 3
    return {
      height: convertToUnit(size),
      minWidth: convertToUnit(size),
      width: convertToUnit(size),
    }
  },

  alertIndicatorStyles() {
    const size = this.size / 2
    return {
      height: convertToUnit(size),
      minWidth: convertToUnit(size),

```

```

    }
  },

  alertClasses() {
    return ['c-avatar__alert', 'alert']
  },
  statusClasses() {
    return ['c-avatar__status', this.status]
  },
  formTitle() {
    return this.editedIndex === -1 ? 'Добавить задачу' : 'Изменить задачу'
  },
},
created() {
  this.$store.dispatch('fetchProject')
  this.fetchTask()
  this.$nextTick(() => {
    this.getTaskStatus.forEach((item) => {
      new Sortable(document.querySelector(`#task-list__${item.value}`), {
        group: 'shared',
        animation: 150,
        onEnd: (e) => {
          const taskId = e.item.dataset.id
          const status = e.to.dataset.status
          this.$store.dispatch('updateTaskStatus', { id: taskId, status: status })
        },
      })
    })
  })
})

if (window._AVATAR === undefined) {
  window._AVATAR = this
}
},

watch: {
  dialog(val) {
    val || this.close()
  },
  dialogDelete(val) {
    val || this.closeDelete()
  },
},
},
methods: {
  fetchTask() {
    this.items = []
    this.loading = true
    const query = {

```

Продолжение приложение Б

```
    project_id: this.project_id,
  }
  this.$store.dispatch('fetchTask', query).then((resp) => {
    this.items = resp.data
    this.loading = false
  })
},
computeAvatar(username) {
  const avatar = this.getUsername === username ? this.getAvatar : ""
  return avatar
},
handleProjectChange() {
  this.fetchTask()
},
getTaskByStatus(status) {
  return this.items.filter((item) => item.status === status)
},
handleCreateTask() {
  this.selectedItem = null
  this.showDialog = true
},
handleEditItem(item) {
  this.selectedItem = item
  this.showDialog = true
},
handleDeleteItem({ id }) {
  if (window.confirm('Are you sure to delete this')) {
    this.$store.dispatch('deleteTask', id).then(() => {
      this.items = this.items.filter((item) => item.id !== id)
    })
  }
},
handleFormSuccess() {
  this.fetchTask()
  this.showDialog = false
},
},
team() {
  let team = []
  for (let member of this.members) {
    team.push({ name: this.personFullName(member), id: member.id })
  }
  return team
},
taskStatuses() {
  let taskStatuses = []
  for (let i = 0; i < this.status.length; i++) {
    taskStatuses.push({ text: this.status[i], id: i })
  }
}
```

```

    }
    return taskStatuses
  },
  priorities() {
    let priorities = []
    for (let i = 0; i < this.priority.length; i++) {
      priorities.push({ text: this.priority[i].text, id: i })
    }
    return priorities
  },
  previewItem(item) {
    this.editedIndex = this.taskList.indexOf(item)
    this.editedItem = Object.assign({}, item)
    this.preview = true
    this.dialog = true
  },
  editItem(item) {
    this.editedIndex = this.taskList.indexOf(item)
    this.editedItem = Object.assign({}, item)
    this.dialog = true
  },
  deleteItem(item) {
    this.editedIndex = this.taskList.indexOf(item)
    this.editedItem = Object.assign({}, item)
    this.dialogDelete = true
  },
  deleteItemConfirm() {
    this.taskList.splice(this.editedIndex, 1)
    this.closeDelete()
  },
  close() {
    this.dialog = false
    this.preview = false
    this.$nextTick(() => {
      this.editedItem = Object.assign({}, this.defaultItem)
      this.editedIndex = -1
    })
  },
  closeDelete() {
    this.dialogDelete = false
    this.$nextTick(() => {
      this.editedItem = Object.assign({}, this.defaultItem)
      this.editedIndex = -1
    })
  }
}

```

```

},

if (window._AVATAR === undefined) {
  window._AVATAR = this
}
},

watch: {
  dialog(val) {
    val || this.close()
  },
  dialogDelete(val) {
    val || this.closeDelete()
  },
},
methods: {
  fetchTask() {
    this.items = []
    this.loading = true
    const query = {
      project_id: this.project_id,
    }
    this.$store.dispatch('fetchTask', query).then((resp) => {
      this.items = resp.data
      this.loading = false
    })
  },
  computeAvatar(username) {
    const avatar = this.getUsername === username ? this.getAvatar : ""
    return avatar
  },
  handleProjectChange() {
    this.fetchTask()
  },
  getTaskByStatus(status) {
    return this.items.filter((item) => item.status == status)
  },
  handleCreateTask() {
    this.selectedItem = null
    this.showDialog = true
  },
  handleEditItem(item) {
    this.selectedItem = item
    this.showDialog = true
  },
  handleDeleteItem({ id }) {
    if (window.confirm('Are you sure to delete this')) {
      this.$store.dispatch('deleteTask', id).then(() => {

```



```

        this.items = this.items.filter((item) => item.id !== id)
    })
  }
},
handleFormSuccess() {
  this.fetchTask()
  this.showDialog = false
},
},
team() {
  let team = []
  for (let member of this.members) {
    team.push({ name: this.personFullName(member), id: member.id })
  }
  return team
},
taskStatuses() {
  let taskStatuses = []
  for (let i = 0; i < this.status.length; i++) {
    taskStatuses.push({ text: this.status[i], id: i })
  }
  return taskStatuses
},
priorities() {
  let priorities = []
  for (let i = 0; i < this.priority.length; i++) {
    priorities.push({ text: this.priority[i].text, id: i })
  }
  return priorities
},
previewItem(item) {
  this.editedIndex = this.taskList.indexOf(item)
  this.editedItem = Object.assign({}, item)
  this.preview = true
  this.dialog = true
},
editItem(item) {
  this.editedIndex = this.taskList.indexOf(item)
  this.editedItem = Object.assign({}, item)
  this.dialog = true
},
deleteItem(item) {
  this.editedIndex = this.taskList.indexOf(item)
  this.editedItem = Object.assign({}, item)
  this.dialogDelete = true
},

```

```

deleteItemConfirm() {
  this.taskList.splice(this.editedIndex, 1)
  this.closeDelete()
},

close() {
  this.dialog = false
  this.preview = false
  this.$nextTick(() => {
    this.editedItem = Object.assign({}, this.defaultItem)
    this.editedIndex = -1
  })
},

closeDelete() {
  this.dialogDelete = false
  this.$nextTick(() => {
    this.editedItem = Object.assign({}, this.defaultItem)
    this.editedIndex = -1
  })
},

save() {
  if (this.editedIndex > -1) {
    Object.assign(this.taskList[this.editedIndex], this.editedItem)
  } else {
    this.taskList.push(this.editedItem)
  }
  this.close()
},
}
</script>

<style lang="sass" scoped>
.task-list
  min-height: 500px
  padding: 0
  &__item
    border-bottom: 1px solid #eee
.sortable-chosen,
.sortable-ghost
  border: 1px dotted var(--v-primary-base)
  background: #eee

.c-avatar
  position: relative
  display: inline-flex
  &__image

```

```
width: 100px  
height: 100px  
object-fit: cover  
border-radius: 100% !important
```

```
&__initial  
width: 100px  
height: 100px  
object-fit: cover  
display: inline-flex  
text-align: center  
justify-content: center  
align-items: center  
font-size: 20px  
border-radius: 100%
```

```
&__icon  
position: absolute !important  
bottom: 2%  
right: 2%  
border-radius: 100%
```

```
&__status  
border: 2px solid var(--v-secondary-base)  
position: absolute  
bottom: 0  
right: 0  
border-radius: 100%
```

```
&__status.online  
background: #4caf50
```

```
&__alert  
position: absolute  
top: 2%  
left: 2%  
border-radius: 100%  
height: 17px !important  
min-width: 17px !important
```

```
&__alert.alert  
background: #f44336 !important  
color: #fff  
font-size: 12px
```

```
&__status.idle  
background: #f5a532
```

```
&__status.busy
```

```
background: #f44336

&__status.offline
  background: #757f8d
</style>

&__status.online
  background: #4caf50

&__alert
  position: absolute
  top: 2%
  left: 2%
  border-radius: 100%
  height: 17px !important
  min-width: 17px !important

&__alert.alert
  background: #f44336 !important
  color: #fff
  font-size: 12px

&__status.idle
  background: #f5a532
&__status.busy
  background: #f44336

if (this.editedIndex > -1) {
  Object.assign(this.taskList[this.editedIndex], this.editedItem)
} else {
  this.taskList.push(this.editedItem)
}
this.close()
},
}
</script>

<style lang="sass" scoped>
.task-list
  min-height: 500px
  padding: 0
&__item
  border-bottom: 1px solid #eee
.sortable-chosen,
.sortable-ghost
  border: 1px dotted var(--v-primary-base)
  background: #eee
```

```
.c-avatar
  position: relative
  display: inline-flex
&__image
  width: 100px
  height: 100px
  object-fit: cover
  border-radius: 100% !important

&__initial
  width: 100px
  height: 100px
  object-fit: cover
  display: inline-flex
  text-align: center
  justify-content: center
  align-items: center
  font-size: 20px
  border-radius: 100%

&__icon
  position: absolute !important
  bottom: 2%
  right: 2%
  border-radius: 100%

&__status
  border: 2px solid var(--v-secondary-base)
  position: absolute
  .c-avatar
    position: relative
    display: inline-flex
&__image
  width: 100px
  height: 100px
  object-fit: cover
  border-radius: 100% !important

&__initial
  width: 100px
  height: 100px
  object-fit: cover
  display: inline-flex
  text-align: center
  justify-content: center
  align-items: center
  font-size: 20px
  border-radius: 100%
```