

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.04 – Программная инженерия
Направленность (профиль) образовательной программы – Программная инженерия

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

«_____» _____ 2025 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка программного обеспечения биллинговой системы для ООО «ИнфоКом»

Исполнитель

студент группы 1105-об _____ В.А. Крылов
(подпись, дата)

Руководитель

доцент, канд. техн. наук _____ А.В. Бушманов
(подпись, дата)

Консультант

по безопасности и

экологичности

доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль

инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Благовещенск 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____ А.В. Бушманов
подпись И.О. Фамилия

«_____» _____ 2025 г.

З А Д А Н И Е

К выпускной квалификационной работе студента Крылова Василия
Анатольевича

1. Тема выпускной квалификационной работы: Разработка программного обеспечения биллинговой системы для ООО «ИнфоКом»

(утверждена приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы (проекта): 10.06.2025

3. Исходные данные к выпускной квалификационной работе: предметная область, перечень литературных источников, отчёт по преддипломной практике.

4. Содержание выпускной квалификационной работы: (перечень подлежащих разработке вопросов): анализ деятельности предприятия, проектирование биллинг-системы, разработка программного обеспечения.

5. Перечень материалов приложения: (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.): техническое задание.

6. Консультанты по выпускной квалификационной работе (с указанием относящихся к ним разделов): консультант по безопасности и экологичности Булгаков А.Б., доцент, канд. техн. наук.

7. Дата выдачи задания: 14.02.2025 г.

Руководитель выпускной квалификационной работы: Бушманов А. В. доцент, канд. техн. наук

(фамилия, имя, отчество, должность, ученая степень, ученое звание)

Задание принял к исполнению (дата): 14.02.2025 г.

(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 84 с., 23 рисунка, 9 таблиц, 31 источник, 8 приложений.

БИЛЛИНГ-СИСТЕМА, МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ, ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, UML ДИАГРАММЫ

В работе рассматривается проектирование и разработка биллинг-системы для обработки заказов, выставления счетов и управления платежами за цифровые услуги и лицензии в ООО «ИнфоКом».

Цель работы – разработать программное обеспечение биллинг-систему для ООО «ИнфоКом».

Для достижения цели работы необходимо произвести: анализ деятельности организации ООО «ИнфоКом», составление технического задания на разрабатываемую биллинг-систему, рассмотрение API сервисов Bitrix24 и Robokassa, проектирование архитектуры программного обеспечения, разработку интерфейса биллинг-системы и реализацию программного обеспечения.

На этапе проектирования была выбрана микросервисная архитектура. Серверная часть разработана на языке Python с применением фреймворка FastAPI, а интерфейс реализован на JavaScript с использованием Vue.js и PrimeVue.

В результате выполнения работы была разработана биллинг-система, обеспечивающая обработку заказов, выставления счетов и управления платежами за цифровые услуги.

СОДЕРЖАНИЕ

Введение	9
1 Теоретические основы изучаемой проблемы	11
1.1 Общие сведения о компании ООО «ИнфоКом»	11
1.2 Организационная структура компании	12
1.3 Документооборот компании ООО «ИнфоКом»	14
1.4 Анализ используемого программного обеспечения компанией	16
1.5 Обзор и анализ аналогов программного обеспечения	18
1.6 Обоснование необходимости разработки биллинг-системы для ООО «ИнфоКом»	22
1.7 Изучение API Bitrix24 и Robokassa	24
1.7.1 Рассмотрение API Bitrix24	25
1.7.2 Рассмотрение API Robokassa	31
2 Проектирование программного обеспечения	35
2.1 Общие сведения о программном обеспечении	35
2.2 Функциональная модель	37
2.3 Архитектурный проект	40
2.3.1 Выбор архитектуры разрабатываемой биллинг-системы	40
2.3.2 Архитектура разрабатываемой биллинг-системы	43
2.4 Проектирование базы данных	45
2.4.1 Инфологическое проектирование	45
2.4.2 Логическое и физическое проектирование	46
2.5 Выбор средств реализации проекта	50
2.5.1 Выбор средств реализации для бэкенд биллинг-системы	50
2.5.2 Выбор средств реализации для модуля интеграции «CRest» с REST API Bitrix24	51
2.5.3 Выбор средств реализации фронтенд части	52

2.5.4 Технологии деплоя программного обеспечения	53
3 Разработка интерфейса программного обеспечения	54
4 Безопасность и экологичность	64
4.1 Безопасность	64
4.1.1 Требования к техническому оснащению рабочих мест	64
4.1.2 Нормативные требования к помещениям для размещения рабочих мест с персональными компьютерами	67
4.1.3 Нормы освещённости при работе с компьютером	68
4.1.4 Требования к микроклимату и воздуху на рабочих местах	69
4.1.5 Эргономические и технические требования к организации рабочих мест с персональными компьютерами	70
4.1.6 Нормирование работы за персональным компьютером	72
4.1.7 Требования к интерфейсу программного обеспечения с учётом требований удобства, доступности	73
4.2 Экологичность	75
4.3 Чрезвычайные ситуации	76
Заключение	79
Библиографические ссылки	81
Библиографический список	82
Приложение А Документооборот компании	85
Приложение Б Техническое задание	87
Приложение В Диаграмма вариантов использования	96
Приложение Г Функциональная модель программы	103
Приложение Д Диаграмма компонентов	108
Приложение Е Диаграммы последовательностей и состояний	119
Приложение Ж Сущности базы данных	123
Приложение И Логическое и физическое проектирование	124

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты:

ГОСТ 19.201-78 ЕСПД Техническое задание. Требования к содержанию и оформлению

ГОСТ Р 50571.5.54-2013 Электроустановки низковольтные. Выбор и монтаж электрооборудования. Заземляющие устройства, защитные проводники и защитные проводники уравнивания потенциалов

ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности

ГОСТ Р 52872-2019 Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности

Постановление Правительства РФ от 16.09.2020 N 1479 «Об утверждении правил противопожарного режима в Российской Федерации»

СанПиН 2.1.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

СП 52.13330.2016 Естественное и искусственное освещение

Федеральный закон «Об отходах производства и потребления» от 24.06.1998 N 89-ФЗ

Федеральный закон «Технический регламент о требованиях пожарной безопасности» от 22.07.2008 N 123-ФЗ

ISO/IEC 19505-2012 Information technology – Object Management Group Unified Modeling Language (OMG UML) Part 2: Superstructure

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

В настоящей бакалаврской работе применяют следующие термины с соответствующими определениями:

ЭДО – система обмена юридически значимыми документами в электронном виде между организациями и государственными структурами;

Access token – временный ключ доступа, выдаваемый после успешной авторизации, позволяющий приложению выполнять запросы от имени пользователя;

API – программный интерфейс, предоставляющий набор функций и протоколов для взаимодействия между программными компонентами;

CRM – система управления взаимоотношениями с клиентами, предназначенная для автоматизации процессов продаж, маркетинга, поддержки и обслуживания;

Batch – механизм пакетной обработки запросов в Bitrix24, позволяющий выполнять несколько вызовов API одновременно;

Bitrix24 – облачная и коробочная платформа для управления бизнесом, включающая инструменты CRM, задач и проектов, и автоматизации бизнес-процессов. Обладает API для интеграции со сторонними системами;

HTTP – протокол передачи гипертекста, используемый для обмена данными между клиентом и сервером в сети Интернет;

HTTPS – защищённая версия HTTP, использующая шифрование (TLS/SSL) для безопасной передачи данных;

SSL – криптографический протокол, обеспечивающий защищённую передачу данных в интернете;

TLS – криптографический протокол, обеспечивающий безопасность передачи данных в интернете, пришедший на смену SSL;

JSON – текстовый формат обмена данными;

JWT – стандарт для обмена данными в формате JSON, часто используемый для авторизации и аутентификации;

OAuth 2.0 – протокол авторизации, позволяющий сторонним приложениям получать ограниченный доступ к ресурсам пользователя без передачи его учетных данных. В Bitrix24 используется для интеграции приложений с REST API;

Refresh Token – ключ для продления access token без необходимости повторной авторизации;

REST API – архитектурный стиль взаимодействия компонентов распределённого приложения в сети;

REST API Bitrix24 – программный интерфейс для взаимодействия с платформой Bitrix24, основанный на принципах REST-архитектуры. Позволяет осуществлять интеграцию с CRM-системой, управлять сущностями, обрабатывать события и выполнять бизнес-операции через HTTP/HTTPS-запросы;

SaaS (Software as a Service) – модель предоставления программного обеспечения, при которой пользователи получают доступ к софту онлайн;

URL (Uniform Resource Locator) в информатике – это уникальный адрес ресурса в интернете;

HTTP Basic Auth – это метод предоставления агентом пользователя HTTP имени пользователя и пароля при выполнении запроса.

ВВЕДЕНИЕ

В условиях роста числа электронных услуг разработка биллинг-системы становится важной для управления финансовыми процессами. Такие системы обеспечивают управление платежами, учет транзакций и интеграцию с бизнес-инструментами, такими как CRM и платежные системы. Однако универсальные решения не соответствуют специфическим требованиям компаний, что подчеркивает актуальность разработки индивидуального программного продукта. Исходные данные для работы основаны на потребностях компании ООО «ИнфоКом», предоставляющей платные услуги, включая подписки и цифровые товары. Для автоматизации финансовых процессов требуется биллинг-система, интегрированная с пользовательским интерфейсом, через который клиент может выбирать цифровые услуги, например, подключение линий WhatsApp или групповых чатов, интегрированная с CRM Bitrix24 для управления заказами, а также с платежной системой Robokassa для обработки непосредственных транзакций. Существующие универсальные решения не обеспечивают необходимой гибкости. Разрабатываемая система позволит интегрировать функции выставления счетов и управления платежами в единый механизм, упрощая взаимодействие с клиентами за счет учета транзакций.

Необходимость разработки обусловлена ограничениями готовых решений, которые не поддерживают интеграцию с Bitrix24 и не адаптированы к специфике услуг компании. Это создает сложности в отслеживании заказов, управлении данными и обработке платежей в реальном времени. Предлагаемая биллинг-система устранил эти недостатки, обеспечив прозрачность операций.

Актуальность работы обусловлена ростом спроса на цифровые услуги и необходимостью оптимизации финансовых процессов в условиях высокой конкуренции.

В связи с этим разработка собственной биллинг-системы для ООО «ИнфоКом» обеспечит автоматизированный учет платежей и интеграцию с CRM Bitrix24. Новизна заключается в исследовании и реализации интеграции биллинг-системы с

Bitrix24 и Robokassa. Ещё одной причиной актуальности является возможность дальнейшего масштабирования системы. В будущем она может поддерживать новые платежные системы. Кроме того, автоматизация финансовых процессов снизит нагрузку на сотрудников, позволяя им сосредоточиться на стратегических задачах.

Цель работы разработка программного обеспечения биллинг-системы для ООО «ИнфоКом», обеспечивающей работу с CRM Bitrix24 и платежной системой Robokassa.

Предметная область работы охватывает биллинг-систему, включая управление платежами, учет транзакций CRM Bitrix24 и платежной системой Robokassa.

Объектом исследования выступает компания ООО «ИнфоКом».

Предметом исследования является процесс проектирования, реализации и интеграции биллинг-системы с внешним приложением компании, CRM Bitrix24 и платежной системой Robokassa.

Для осуществления цели необходимо решить следующие задачи:

- выполнить анализ предметной области компании ООО «ИнфоКом»;
- рассмотреть аналоги разрабатываемого программного обеспечения;
- рассмотреть API Bitrix24 и Robokassa с которыми будет взаимодействовать проектируемое программное обеспечение;
- спроектировать программное обеспечение ООО «ИнфоКом»;
- выбрать технический стек для реализации программного обеспечения;
- реализовать программное обеспечение биллинг-систему.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИЗУЧАЕМОЙ ПРОБЛЕМЫ

1.1 Общие сведения о компании ООО «ИнфоКом»

Исследуемым объектом является ООО «ИнфоКом», зарегистрированная 24 апреля 2007 года. Руководителем компании является Евгений Александрович Иванов. ООО «ИнфоКом» специализируется на создании и внедрении информационных систем, целью которых является оптимизация и автоматизация бизнес-процессов клиентов.

Основные бизнес-процессы данной организации включают в себя: анализ требований, проектирование и разработку программного обеспечения, а также его внедрение и сопровождение.

Организация предоставляет услуги:

- анализ и описание существующих бизнес-процессов, создание схем;
- построение системы управления задачами, которая включает в себя предоставление лёгкого и доступного инструмента контроля и учёта рабочего времени сотрудников;
- внедрение решений для автоматизации рутинных задач;
- разработка и внедрение корпоративных порталов Bitrix24;
- автоматизация внешней коммуникации с клиентами, включая автоматизированное ведение диалога с клиентами по средствам программного обеспечения;
- установка и настройка систем управления взаимоотношениями с клиентами CRM для отслеживания и управления взаимодействиями;
- предоставление решений для электронного документооборота;
- оптимизация процесса сбора информации о клиентах, объединение информации о клиентах из писем, сообщений, социальных сетей WhatsApp и Telegram прямо в CRM Bitrix24 для автоматизации продажи и консультации;
- решения для организаций и хранения данных о клиентах в удобной форме, систематизация информации.

1.2 Организационная структура компании

Исследуемая компания обладает организованной структурой, которая позволяет управлять всеми направлениями своей деятельности и обеспечивать предоставление услуг. Организационная структура компании представляет собой формальную систему, которая определяет, как управляются и координируются различные функциональные направления и подразделения в организации, а также определяет иерархические отношения между сотрудниками, структуру управления, потоки коммуникации, полномочия и обязанности сотрудников.

В ходе анализа организационной структуры компании ООО «ИнфоКом» была выявлена структура, представленная на рисунке 1.

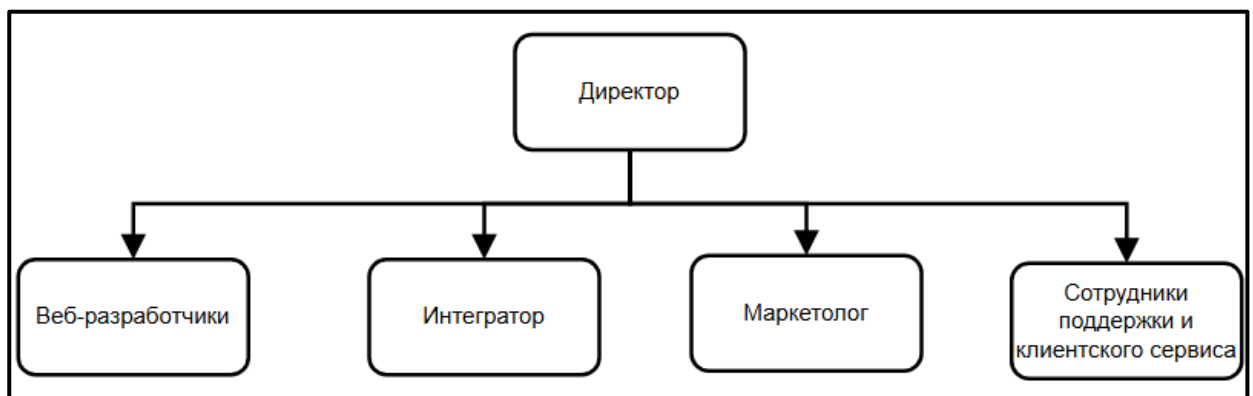


Рисунок 1 – Организационная структура компании ООО «ИнфоКом»

Рассмотрим анализ организационной структуры компании ООО «ИнфоКом». Во главе компании стоит директор, который выполняет основные функции управления и представления интересов компании. Все работники компании, включая веб-разработчиков, интеграторов, маркетологов и сотрудников поддержки и клиентского сервиса, находятся в подчинении директора.

Как и в большинстве компаний, во главе стоит директор, выполняющий следующие функции:

- осуществление руководства деятельностью организации;
- представляет предприятие во всех учреждениях и организациях, в судах, как на территории России, так и за ее пределами;

- заключение сделок и договоров в соответствии с целями организации;
- обеспечение сохранности и эффективного использования имущества;
- предоставление всех видов отчетности в установленные сроки в соответствующие органы статистики, финансовые и налоговые органы;
- осуществление бухгалтерского учета;
- утверждение организационной структуры организации;
- прием на работу и увольнение сотрудников, заключение, изменение и расторжение трудовых договоров в соответствии с трудовым законодательством и нормативными правовыми актами.

Обязанности веб-разработчиков включают:

- разработка серверной части приложения;
- проектирование, создание и управление базами данных;
- разработка и реализация архитектуры приложения, включая управление соединениями, обработку запросов и ответов, аутентификации пользователей;
- создание и интеграция API для обмена данными между серверной и клиентской частями приложения;
- создание веб-интерфейса.

Деятельность системного интегратора включает в себя:

- оценка потребностей бизнеса, анализ существующих систем и инфраструктуры, выявление потенциальных проблем и прогнозирование рисков;
- проектирование интегрированных решений ИТ-инфраструктуры с максимально подходящими компонентами и техническими спецификациями;
- разработка и внедрение решений, установку, настройку и объединение отдельных компонентов в единую систему;
- составление методических рекомендаций для пользователей ИТ-инфраструктуры;
- тестирование и устранение проблем;
- внедрение Bitrix24 в бизнес-процессы компаний.

Область ответственности маркетолога включает в себя:

- оценка рекламных каналов и предложения по оптимизации;
- подбор оптимальных рекламных каналов для продвижения услуг;
- исследование конкурентов, включающее анализ их каналов продвижения, сильных и слабых сторон, позиционирования;
- исследование состояния рынка, его объёма, ключевых игроков и составление прогнозов;
- разработка маркетинговой стратегии.

В задачи сотрудников поддержки и клиентского сервиса входят:

- обеспечение технической поддержки пользователей;
- консультацию клиентов по вопросам функциональности продуктов компании;
- обучение пользователей, демонстрация возможностей ПО.

1.3 Документооборот компании ООО «ИнфоКом»

Рассмотрим реализацию внешнего и внутреннего документооборота компании ООО «ИнфоКом». Внешний и внутренний документооборот позволяют обеспечивать согласование, выполнение и контроль задач.

Внешний документооборот представляет собой все входящие и исходящие документы компании, которыми она обменивается с организациями, клиентами и контролирующими органами. Схема внешнего документооборота организации ООО «ИнфоКом» приведена в приложении А.1.

В нём предприятие обменивается информацией с клиентами, ООО «Яндекс» для аренды вычислительных мощностей, управлением пенсионного фонда, фондом социального страхования, территориальным фондом обязательного медицинского страхования, отделением ПАО «СберБанк», межрайонной инспекцией ФНС, ООО «1С-Битрикс».

Компания взаимодействует с клиентами посредством заявок на автоматизацию и требований к проводимым работам. На основе этих заявок разрабатывается техническое задание, которое после формирования передаётся заказчику для согласования.

После того как заказчик согласовал техническое задание начинается процесс разработки, по окончании которого осуществляется приёмка проведённых работ. В следствии чего заказчиком в организацию предоставляется акт о приёме работ, а также заказчику передаётся счёт за предоставленные услуги.

Далее ООО «ИнфоКом» взаимодействует с ООО «Яндекс». предоставляя заявку на аренду вычислительных мощностей. Затем ООО «Яндекс» предоставляется договор об аренде вычислительных мощностей и политику конфиденциальности, а также счёт за предоставленные услуги.

ООО «ИнфоКом» также взаимодействует с структурами управления пенсионного фонда, фонд социального страхования, фонд обязательного медицинского страхования и инспекцией федеральной налоговой службы, которые предоставляют данной организации требования к отчётности, а организация ООО «ИнфоКом», непосредственно предоставляет им отчёты. Кроме того, компания взаимодействует с отделением ПАО «СберБанк», предоставляя ему платёжные поручения и получая от него банковские выписки. Также ООО «ИнфоКом» реализует технологическое партнёрство с ООО «1С-Битрикс», в рамках которого получает партнёрское предложение и направляет в ответ согласованное предложение о сотрудничестве.

Внутренний документооборот компании описывает движение документов между структурными подразделениями и сотрудниками. Внутренний документооборот ООО «ИнфоКом» приведён в приложении А.2.

Ключевая роль в этом случае генеральный директор, он выполняет взаимодействия с ООО «Яндекс» арендуя у него вычислительные мощности, получает требования к отчётности и создаёт отчёты для различных государственных учреждений, проводит анализ требований, согласовывает техническое задание, полученное от клиентов организации, а также рассматривает партнёрские предложения от ООО «1С-Битрикс». Он издаёт распоряжения и поручения, которые определяют порядок работы сотрудников и устанавливают цели, задачи для них.

Анализ требований начинается с детального изучения проблем и потребностей клиентов, в процессе которого директор и сотрудники организации используют такие

методы выявления требований как интервью и опрос заинтересованных лиц проекта, с целью формирования наиболее полной картины требований к проектируемой системе, определения ключевых цели и задач конечного продукта.

Веб-разработчики на основе согласованных технических заданий занимаются разработкой программного обеспечения, ведут журналы изменений и передают их интегратору.

После завершения разработки осуществляется внедрение программного обеспечения интегратором, которое охватывает процесс установки и конфигурации программного продукта, а также обучение персонала использованию программного обеспечения.

Интегратор также описывает изменения в программном обеспечении и предаёт их маркетологу и сотрудникам поддержки и клиентского сервиса. Интегратор получает от клиентов заявку на автоматизацию и требования к программному обеспечению, после чего рассматривает их и создаёт техническое задание, передавая его заказчику для согласования.

Маркетолог, получив описания обновлений программного обеспечения, проводит анализ этих нововведений и разрабатывает коммерческие предложения, которые подчеркивают преимущества и новые функции продукта для клиентов, после чего он направляет эти предложения клиентам.

Сотрудники поддержки и клиентского сервиса взаимодействует с клиентами организации, рассматривая и отвечая на их обращения.

1.4 Анализ используемого программного обеспечения компанией

В ООО «ИнфоКом» для организации рабочего процесса используются ИТ-сервисы, наиболее важными из которых являются система управления проектами, CRM-система, система управления версиями программного обеспечения, ERP-система и комплекс мер по обеспечению безопасности:

- Customer Relationship Management система (CRM-система), предназначена для управления взаимоотношениями организации с ее клиентами. Она предоставляет

инструменты для управления аспектами взаимодействия с клиентами, включая маркетинг, продажи, обслуживание клиентов и анализ данных. В компании в качестве CRM-системы используется Bitrix24.

- Система управления версиями программного обеспечения GitHub, GitLab, позволяет разработчикам, отслеживать изменения в исходном коде программного обеспечения, возвращаться к предыдущим версиям кода, а также координировать работу над проектом между несколькими разработчиками. В качестве системы управления версиями в организации используется GitHub, GitLab.

- Enterprise Resource Planning-система (ERP-система) 1C:ERP предназначена для интеграции и автоматизации бизнес-процессов в компании. Она объединяет данные в единую информационную систему, что позволяет улучшить управление ресурсами. Для организации электронного документооборота в ООО «ИнфоКом» применяется система ЭДО Контур.Диадок.

- В компании используется комплекс мер по обеспечению безопасности, включает антивирусное программное обеспечение, системы обнаружения и блокировки несанкционированного доступа.

Инфраструктура организации ООО «ИнфоКом» включает несколько ключевых элементов, обеспечивающих её эффективную работу:

- Серверное оборудование, организация арендует серверы, в ООО «Яндекс», которые базируются на аппаратуре от ведущих производителей, таких как HP, Dell и Supermicro. Они расположены в центре обработки данных ООО «Яндекс», который обеспечивает их непрерывную работу.

- Сетевое оборудование, организация использует сетевое оборудование от Cisco для создания надежной и эффективной сети, которая способствует быстрой и бесперебойной передаче данных.

- Рабочие станции, организация использует рабочие станции, оснащенные графическими ускорителями Nvidia, которые характеризуются мощностью и производительностью, так как построены на платформах Intel и AMD.

С учётом активного укрепления позиций ООО «ИнфоКом» рост объёмов транзакций и увеличение числа клиентов обуславливают необходимость унифицированного подхода к автоматизации процессов, связанных с выставлением счетов и проведением оплат. Стандартные решения, предлагаемые на рынке, не отвечают специфике внутренних бизнес-процессов компании, не могут обеспечить полноценную интеграцию с внешними приложениями и CRM-системами, такими как Bitrix24.

Разработка биллинг-системы позволит ООО «ИнфоКом» обеспечить работу бизнес-процессов, создать механизм управления заказами и платежами, а также повысить качество обслуживания клиентов. Предлагаемое решение интегрируется с внутренним приложением компании и CRM Bitrix24 через смарт-процесс, для обеспечения отслеживания заказов на всех стадиях оплаты, централизованный учёт транзакций.

Такое решение поддерживает подписочные модели, так и единовременные платежи, что важно для компании, предоставляющего спектр цифровых услуг. Это, в свою очередь, способствует улучшению контроля над процессами, сокращению затрат на ручное выставление счета и обеспечение гибкости при адаптации к требованиям рынка.

1.5 Обзор и анализ аналогов программного обеспечения

Перед разработкой программного обеспечения необходимо провести обзор существующих решений, чтобы оценить, насколько готовые продукты способны закрыть ключевые требования поддержке подписочных и разовых платежей, формирование «корзины» услуг, интеграцию с Bitrix24 и платёжным шлюзом Robokassa.

Начнем сравнение функциональных возможностей Stripe и PricePlan, это необходимо, чтобы понять, насколько каждая платформа удовлетворяет потребности бизнеса в сфере приёма платежей и биллинга.

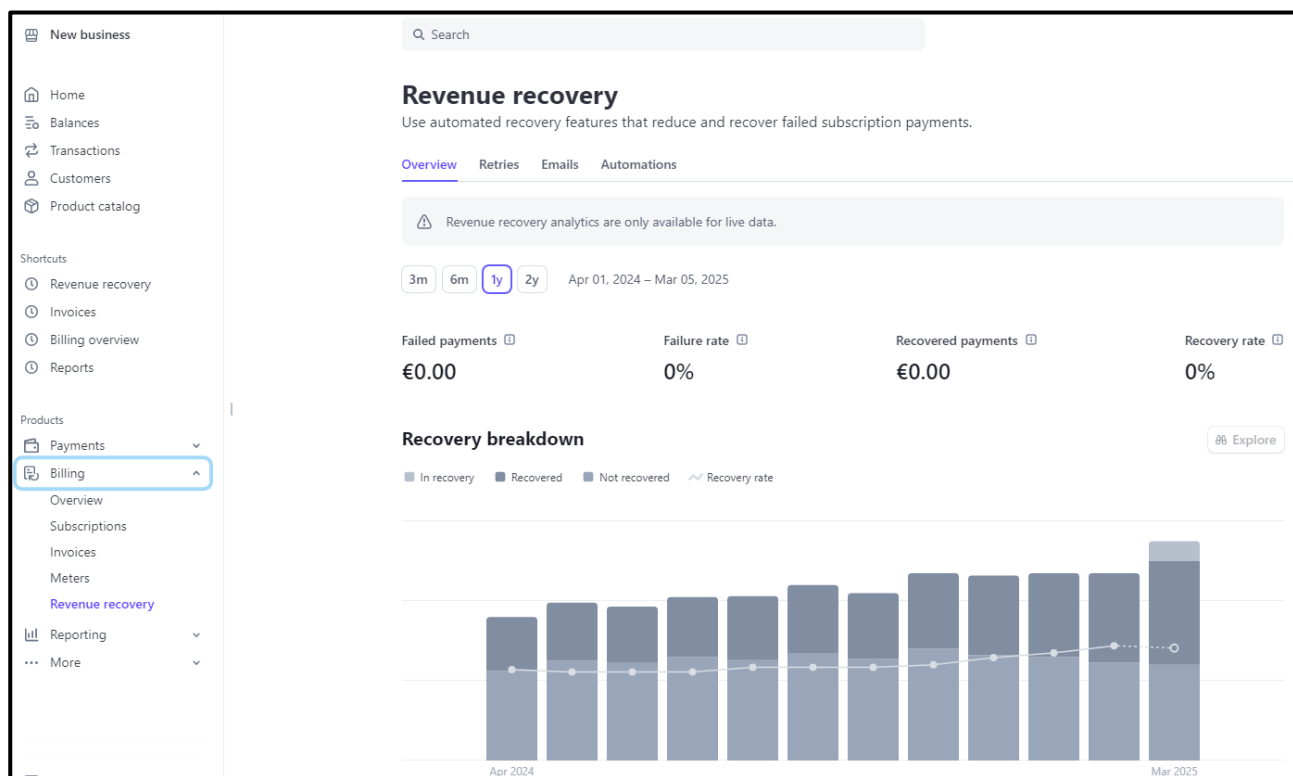


Рисунок 2 – Интерфейс Stripe

PricePlan Редактирование продукта

Список продуктов Редактирование продукта

Наименование **шаблоны**
Портал-24 Команда

Базовая цена 4990 руб. 1 месяц НДС НДС 18%

Предоплата 100 % Статус продукта **Активный**

Подпродукты/Услуги

Добавить опцию...

Место в облаке: 100 Гб **шаблоны**

Включено 0 Шт с возможностью расширить до 1 по цене 1990 (НДС) руб. за Шт (предоплата 1. за базовый период добавить скидку)

Место в облаке: 250 Гб **шаблоны**

Включено 0 Шт с возможностью расширить до 1 по цене 2490 (НДС) руб. за Шт (предоплата 1. за базовый период добавить скидку)

Место в облаке: 1 Тб **шаблоны**

Включено 0 Шт с возможностью расширить до 1 по цене 4490 (НДС) руб. за Шт (предоплата 1. за базовый период добавить скидку)

Отмена Сохранить изменения

Первый период 0 скидка 0 %

Периоды продления

3 месяц скидка 0 %

6 месяц скидка 5 %

12 месяц скидка 10 %

В каждом периоде В первом периоде По факту использования Переменные

Редактировать

Укажите число

1990

НДС 18%

Сохранить

Рисунок 3 – Интерфейс PricePlan

Функциональные возможности двух систем представлены в таблице 1.

Таблица 1 – Функциональные возможности Stripe и PricePlan

Функциональность	Stripe	PricePlan
Поддерживаемые способы оплаты и валюты	Поддерживает более 100 способов оплаты по всему миру. Доступен в более 40 странах; поддерживает приём платежей в более чем 135 валютах.	Не является прямым оператором платежей; интегрируется с внешними платёжными сервисами для приёма платежей картами и через электронные кошельки.
Управление подписками	Stripe оперирует объектом Subscription для регулярных платежей. Подписка привязывает клиента к тарифу и позволяет автоматически взимать плату на регулярной основе. Создание подписки выполняется через API.	В PricePlan подписка создаётся через REST API метод, где в теле запроса указываются идентификатор тарифного плана, период тарификации и ID клиента. При создании подписки требование, задать поле клиента.
Формирование корзины	Stripe поддерживает выставление счетов с несколькими позициями. Для разовых счетов можно создавать несколько объектов позиций счета и добавлять их в один счет-фактуру.	PricePlan позволяет объединять несколько услуг в одном счёте. Если у клиента подключено несколько подписок или разовых услуг, система может сформировать единый счет.
Разовые платежи	Для одноразовых платежей Stripe предоставляет Payments API. Разовый платеж оформляется созданием PaymentIntent с указанием суммы, валюты и платежного метода.	В PricePlan платежи клиентов осуществляются через систему счетов. При поступлении разового платежа, оплата через интегрированную платежную систему, сумма автоматически зачисляется на баланс клиента, откуда может списаться за услуги.
Webhooks (события)	Stripe поддерживает вебхуки для обработки событий.	В PricePlan встроен механизм вебхуков для интеграции со сторонними системами.
Выставление счетов и документы	Поддерживает выставление счетов клиентам: платформа может автоматически генерировать и отправлять счета на разовые или регулярные платежи. Счёт может быть оплачен онлайн, доступны PDF-квитанции и уведомления о предстоящем или просроченном платеже.	Биллинг формирует счета для юридических лиц: на основе начислений генерируется счёт и комплект закрывающих документов. Реализована двусторонняя интеграция с 1С: выставленные счета передаются в 1С, а сведения о поступивших оплатах загружаются в систему.
Метрики и отчетность	В Stripe Dashboard доступна аналитика по подпискам и платежам: можно отслеживать рекуррентную выручку (MRR), динамику активных подписок.	В PricePlan доступны отчёты по метрикам SaaS-бизнеса: автоматически считаются показатели MRR, оттока, ARPU, LTV.
Транзакционные email и SMS-уведомления	Встроена автоматическая рассылка квитанций и уведомлений через email. SMS по умолчанию не поддерживаются.	Email и SMS встроены. Платформа шлёт документы и события через Amazon SES и SMS-шлюз.

Рассматриваемые в таблице 2 решения не покрывают требования. Stripe лишён готовой связки с Robokassa и предлагает сторонний коннектор к Bitrix24. У PricePlan ситуация зеркальная, интеграция с Robokassa реализована из коробки, что удобно для приёма платежей в рублях, но готового модуля Bitrix24 нет. Помимо основного функционала, важно оценить, насколько платформа интегрируется с внешними сервисами, платёжным агрегатором Robokassa и CRM-системой Bitrix24. Stripe уже содержит собственный эквайринг, потому не требует Robokassa, что касается Bitrix24, в маркетплейсе CRM доступно приложение Stripe Integration, оно разработано и поддерживается командой Bitrix24, а не самим Stripe. PricePlan может использовать Robokassa как систему для приёма платежей, однако готовой интеграции с Bitrix24 у него нет. Интеграционные возможности представлены в таблице 2.

Таблица 2 – Интеграция с Robokassa и Bitrix24

Интеграция	Stripe	PricePlan
Robokassa	Нет. Stripe сама принимает карточные и альтернативные платежи, хранит данные карт, выполняет 3-D Secure, рассчитывает комиссию и зачисляет средства.	Да. PricePlan действует как биллинг ядро, а Robokassa платёжный агрегатор.
Bitrix24	Нет.	Нет.

В таблице 3, приведены цифры и условия, позволяющие оценить, в какой ситуации каждая система оказывается выгоднее. Важно уточнить, платформа Stripe официально не обслуживает юридические лица, зарегистрированные в России. Взаимодействие возможно через зарубежные структуры.

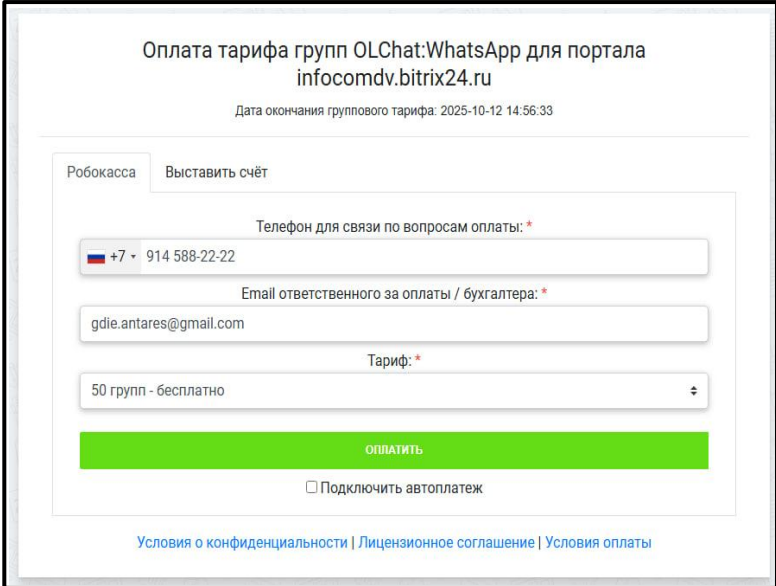
Таблица 3 – Сравнение тарифных планов

Параметр	Stripe	PricePlan
Тарифные планы и подписки	Модель оплаты, один с тариф с оплатой по факту для всех клиентов, не вносится абонентская плата за использование платформы. Комиссия удерживается, когда происходит платёж. Starter 0,4 % с рекуррентного списания, покрывает базовые подписки. Plus 0,5 %, включает функции Starter, а также: PDF-инвойсы, напоминания оплаты, метрики, создания смет.	«Revenue Share» с ежемесячной платой 9500 руб. + 1,5 % от оборота, доступ к API, работа с юридическими и физическими лицами; «Per Account», стоимость за количество счетов; «On-premises», лицензия на 5 000 л.счетов физических лиц и 500 л.счетов юридических лиц, стоимость коробочной версии обсуждается индивидуально.

Параметр	Stripe	PricePlan
Комиссия за эквайринг	Комиссия за обработку платежей: ~1,5 % + 0,25 €. Более высокая ставка для иностранных карт, до 3,25 % + 0,25 €, если валюта платежа отличается от валюты выплаты мерчанту.	Комиссия с оборота: 1,5 % берётся со всех платежей, проведённых через систему в Revenue Share. PricePlan не является эквайером, а интегрируется с внешними платежными системами. Помимо 1,5 % PricePlan, придётся платить тарифы платежного шлюза. Фиксированные платежи: 9 500 руб. в месяц Revenue Share абонентская плата за использование платформы. Per Account фиксированная ставка, зависящая от числа клиентов. Коробочная лицензия: оплачивается разово.
Плата за биллинг/подписки	Комиссия 0,7 % от каждой рекуррентной транзакции. Альтернативно, можно выбрать фиксированную плату: подписка на Stripe Billing с предоплаченным объёмом от 500 евро в месяц за оборот до 80 тысяч евро, с комиссией 0,67 %. Доступны планы с увеличенным объёмом и скидками до 18 %. Для больших оборотов выгоднее фиксированный план, а для небольших % от транзакций.	PricePlan взимает плату за использование функционала управления подписками в рамках тарифа. «Revenue Share» 1,5 % от оборота, процент включает весь функционал биллинга. «Per Account» процентной комиссии нет, вместо неё оплачивается сумма, рассчитываемая от количества клиентов. «On-premises» единовр. лицензионная плата, % отсутствует.
Стоимость использования API	API Stripe предоставляется бесплатно.	API PricePlan включён в тарифы.
Бесплатные лимиты и пробный период	Бесплатный тариф отсутствует.	Бесплатного плана нет. Доступна демоверсия системы.
Техподдержка / SLA (Service Level Agreement) «Соглашение об уровне сервиса».	Growth поддержка с назначенным менеджером, SLA ответа 6 часов. Premium проактивная поддержка, консультации по оптимизации, SLA ответа 4 часа. Enterprise максимально персонализированная поддержка: личный менеджер, встроенный в команду клиента, участие в планировании, выделенный Slack-канал, SLA 4 часа на обычные запросы.	Корп 8 × 5 поддержка по будням, консультации по телефону, приоритетные ответы 8 часов, решение проблем до 16 ч. Корп 8 × 7 – поддержка каждый день 8:00, 19:00 МСК + экстренная помощь вне рабочего времени. SLA ответа 1 час, решение критического сбоя до 8 часов.

1.6 Обоснование необходимости разработки биллинг-системы для ООО «ИнфоКом»

В рамках анализа деятельности компании ООО «ИнфоКом» было выявлено, что существующая система, используемая для управления оплатой услуг приложения компании OLChat, имеет ряд ограничений, которые препятствуют дальнейшему масштабированию бизнеса компании. OLChat, предоставляющий интеграцию чатов и групп WhatsApp и Telegram с платформой Bitrix24, является ключевым продуктом компании на текущий момент. Однако текущая система ориентирована исключительно на это приложение. На данный момент биллинг для OLChat предполагает оплату за каждую «линию» отдельно, что создает неудобства для клиентов при подключении нескольких услуг. Клиентам приходится оплачивать каждую линию или группу по отдельности, что усложняет процесс оплаты. Сам процесс оплаты приведен на рисунке 4. Хотя текущая система поддерживает подписочную модель оплаты, например, тариф «50 групп бесплатно» с определенным периодом действия, она ограничена в гибкости, не позволяет объединять несколько услуг в единую оплату через функцию корзины. Это снижает удобство для клиентов, особенно в сегменте SaaS (Software as a Service), где пользователи ожидают интуитивного управления своими подписками и возможности добавлять или изменять услуги в рамках одного платежа.



Оплата тарифа групп OLChat:WhatsApp для портала
infocomdv.bitrix24.ru

Дата окончания группового тарифа: 2025-10-12 14:56:33

Робокасса Выставить счёт

Телефон для связи по вопросам оплаты: *

+7 914 588-22-22

Email ответственного за оплаты / бухгалтера: *

gdie.antares@gmail.com

Тариф: *

50 групп - бесплатно

ОПЛАТИТЬ

☐ Подключить автоплатеж

[Условия о конфиденциальности](#) | [Лицензионное соглашение](#) | [Условия оплаты](#)

Рисунок 4 – Форма оплаты тарифа OLChat:WhatsApp в текущей биллинг-системе

Разработка биллинг-системы стала необходимостью для ООО «ИнфоКом» по нескольким причинам. Компания планирует расширение портфеля своих продуктов, включая новые сервисы с платными функциями. Новая система должна обеспечивать единый подход к управлению оплатой. Это позволит компании масштабировать бизнес, не создавая отдельные биллинг-системы для каждого нового продукта, что снижает затраты на разработку и сопровождение.

Важной особенностью биллинг-системы является внедрение общей функции «корзины», которая позволяет клиентам добавлять различные услуги, такие как линии и группы для портала, и оплачивать их одним платежом. Это значительно упрощает процесс оплаты для клиентов, делая его более интуитивным и удобным, что, в свою очередь, повышает удовлетворенность пользователей и способствует росту продаж. Клиент сможет подключить несколько линий и групп для OLChat или другого приложения, добавить их в корзину и оплатить одним платежом, вместо того чтобы совершать несколько транзакций.

Разработка собственной биллинг-системы позволяет ООО «ИнфоКом» снизить зависимость от сторонних решений, которые могут не соответствовать требованиям компании. Сторонние биллинг-системы часто требуют дополнительных затрат на лицензии. Собственная система, напротив, может быть полностью адаптирована под нужды компании, включая интеграцию с существующими системами, такими как Bitrix24. Современные клиенты ожидают от поставщиков ИТ-решений не только качественного продукта, но и удобного пользовательского опыта, включая процесс оплаты. Такая биллинг-система позволяет компании предложить клиентам более удобный сервис, что может стать преимуществом в сравнении с другими игроками рынка.

1.7 Изучение API Bitrix24 и Robokassa

Перед проектированием программного обеспечения необходимо изучить API сервисов, с которыми взаимодействует система. Рассмотрим аспекты взаимодействия с API Bitrix24 и Robokassa, включая механизмы аутентификации, принципы взаимодействия, их функциональные возможности, а также форматы передаваемых данных.

1.7.1 Рассмотрение API Bitrix24

Начнем с API Bitrix24 [1], предоставляет программный интерфейс для взаимодействия с платформой Bitrix24, основанный на принципах REST-архитектуры. Позволяет осуществлять интеграцию с CRM-системой, управлять сущностями, обрабатывать события и выполнять HTTP/HTTPS-запросы. Этот интерфейс построен на обмене данными в формате JSON. С помощью REST API можно создавать, обновлять, удалять и получать данные о сущностях системы, от контактов и сделок до задач и пользовательских полей, а также настраивать права доступа.

API Bitrix24 поддерживает несколько способов интеграции, включая входящие и исходящие вебхуки. Вебхуки позволяют получать информацию о событиях в реальном времени, например, о создании сделки или изменении статуса задачи. Локальные вебхуки представляют упрощенный механизм работы с REST API, предназначенный для конкретного портала, и делятся на входящие и исходящие. Входящие вебхуки дают возможность пользователям отправлять данные в систему, при этом генерируется ключ авторизации для вызова REST-методов. Данный механизм годится для «внутренних» интеграций. Этот ключ не имеет срока действия, в отличие от токенов OAuth 2.0.

После создания такого вебхука в результате сформируется URL: `https://doc-test-b24.bitrix24.ru/rest/1/173glortu42lvpju/crm.lead.add.json?title=Новый_лид&name=Иван`.

Рассмотрим, что каждый параметр в URL означает:

- «doc-test-b24.bitrix24.ru»: адрес портала, /rest указывает на работу REST с вебхуками;
- «/1»: идентификатор пользователя, создавшего вебхук;
- «/173glortu42lvpju»: секретный код, который необходимо хранить в тайне, так как он обеспечивает доступ к portalу;
- «/crm.lead.add»: метод добавления лида в CRM Bitrix24;
- «?title=Новый_лид&name=Иван»: данные для создания записи;

– «.json»: параметр, автоматически добавляемый конструктором для указания формата ответа.

Исходящий вебхук, напротив, инициирует отправку данных из Bitrix24 на внешний обработчик при наступлении событий, что позволяет автоматизировать реакции на изменения без постоянного опроса API. Для его настройки указывается URL обработчика и выбирается событие, а Bitrix24 предоставляет токен для проверки подлинности запроса, достаточно сгенерировать секретный ключ и указать обработчик.

Для доступа к API требуется авторизация, через протокол OAuth 2.0, который предполагает использование `access_token` с ограниченным сроком действия, что требует периодического обновления токенов через `refresh_token`. Альтернативно можно использовать ключ вебхука, который менее гибок для сложных сценариев. У API Bitrix24 есть и ограничения: лимит в 2 запроса в секунду для облачных версий или необходимость учета нагрузки на сервер в коробочных решениях, что может привести к временной блокировке при превышении допустимых значений.

API Bitrix24 регулирует нагрузку через лимиты на интенсивность запросов, этот механизм позволяет приложениям кратковременно выполнять запросы с высокой интенсивностью, но ограничивает постоянную нагрузку. Каждый запрос увеличивает условный счетчик на стороне Bitrix24, если его значение превышает пороговое 50, последующие запросы блокируются с ошибкой `QUERY_LIMIT_EXCEEDED` (статус 503). Однако специфика некоторых интегрированных приложений в Bitrix24, может требовать кратковременных всплесков активности, что допустимо, но не на постоянной основе. Учет интенсивности ведется отдельно для каждого портала, но зависит от IP-адреса источника запросов, что важно учитывать при работе нескольких приложений с одного сервера, так как их лимит будет общим. Помимо этого, существуют ограничения на ресурсоемкость: время выполнения запросов к каждому методу суммируется, и при превышении 480 секунд за 10 минут метод блокируется для всех приложений портала до сброса лимита, о чем сообщает параметр `operating` в ответе API. Эти ограничения 2 запроса в секунду дают 172 800 запросов в сутки, усложняют работу с большими объемами данных, особенно если не использовать оптимизации

вроде пакетных запросов (batch) или списочных методов, которые позволяют получать до 50 записей за один вызов. Для взаимодействия с API Bitrix24 в рамках проектирования биллинг-системы была разработана собственная реализация модуля интеграции CRest.

Функциональные возможности CRest являются:

- взаимодействие с Bitrix24 через CRest – выполнение одиночных и пакетных (batch) запросов к API Bitrix24;
- управление авторизацией – OAuth 2.0, обновления токенов, при выполнении запросов;
- асинхронная обработка запросов – поддержка асинхронных вызовов для снижения задержек;
- унифицированная структура вызовов API – возможность работы как с webhook-интеграциями, так и с локальными приложениями;
- логирование – запись всех запросов и ответов для отладки;
- поддержка интеграции – обработчики событий.

CRest выступает в роли библиотеки, которая предоставляет интерфейс для взаимодействия с API Bitrix24. CRest проверяет авторизацию по OAuth 2.0 или локальным вебхукам, маршрутизирует запросы, поддерживает асинхронную обработку запросов и ведет логирование. Модуль позволяет выполнять одиночные, и пакетные запросы, обеспечивая работу при большом количестве обращений, и поддерживает работу совместимую с вебхуками, локальными приложениями и обработчиками событий. Модуль разворачивается на сервере, где настраиваются параметры подключения URL вебхука или данные OAuth 2.0, принимая запросы от клиентских приложений, обрабатывая их и передавая в Bitrix24.

Архитектура модуля CRest базируется на серверно-клиентском взаимодействии. CRest обеспечивает точку взаимодействия, Bitrix24 выступает в роли сервиса управления данными, а клиентская часть подключается к модулю, чтобы взаимодей-

ствовать с Bitrix24. Все обмены происходят через REST API, что обеспечивает гибкость, масштабируемость и удобство развития системы. Архитектура модуля CRest представлена рисунке 4.

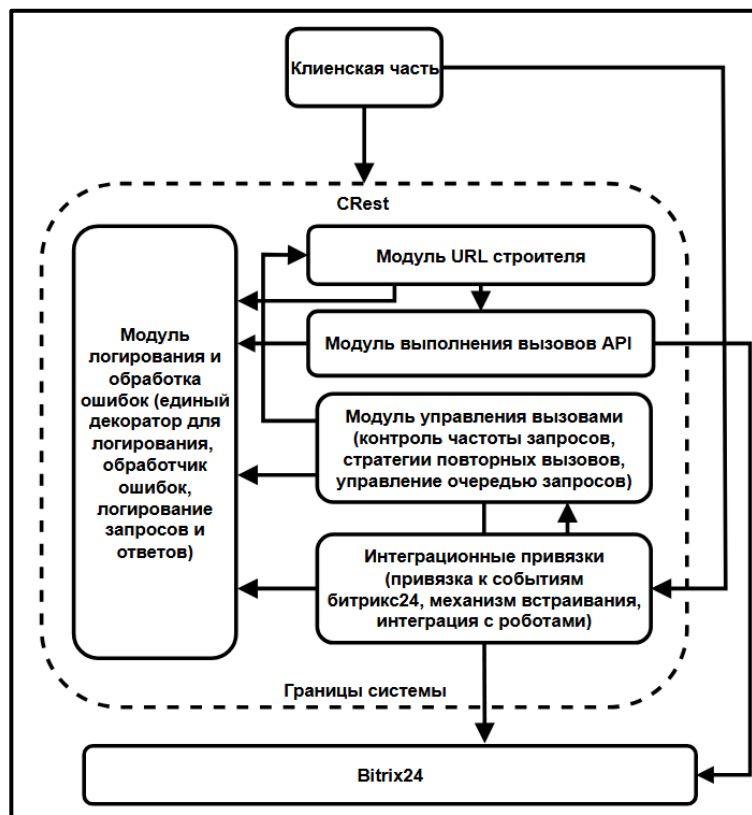


Рисунок 5 – Архитектура CRest

«Модуль URL-строителя» отвечает за формирование адресов для REST-вызовов Bitrix24. При инициализации модуль также определяет необходимость обновления токенов аутентификации. Один из ключевых процессов в работе обновление токена аутентификации. Когда текущий токен устарел или недействителен, формируется новый запрос на сервер авторизации. Этот запрос содержит необходимые параметры клиента, такие как `client_id`, `client_secret` и `refresh_token`. После успешного обновления аутентификации формируется URL-адрес, который используется для выполнения API-запросов. Генерация URL происходит на основе метода вызова и переданных параметров. Заключительный этап функционала заключается в отправке POST-запроса с параметрами аутентификации на сервер Bitrix24. При ответе от

Bitrix24 CRest получает новый токен, срок его действия и дополнительные параметры, необходимые для запросов.

«Модуль выполнения вызовов API» реализует асинхронный клиент, который обновляет токены авторизации, обрабатывает редиректы и повторяет запросы при возникновении ошибок. Основная задача модуля состоит в том, чтобы инкапсулировать логику отправки запросов через REST API, гарантируя формирование URL-адресов, передачу параметров и обработку возвращаемых данных. При этом модуль взаимодействует с другими компонентами системы, такими как «UrlBuilder» и «CallDirector» для управления частотой запросов и стратегиями повторных вызовов. Этот же модуль поддерживает пакетную (batch) обработку запросов, что позволяет оптимизировать нагрузку и сократить общее время выполнения операций. При выполнении одиночного запроса принимается в качестве входных параметров сборщик URL, название метода API, параметры запроса и дополнительное тело, если оно требуется. На основе этих данных формируется запрос, который отправляется к Bitrix24, а затем результат, полученный от сервера, возвращается обратно в приложение.

Рассмотрим модуль управления вызовами, обеспечивает контроль частоты запросов и управление ограничениями, накладываемыми Bitrix24, а также координирует взаимодействие между остальными модулями. Он следит за корректной отправкой данных, обрабатывает сбои и реализует стратегии повторных вызовов при ошибках. Он позволяет снизить нагрузку, уменьшить задержки, отправляя несколько запросов одновременно. Функция модуля формирование пакета запросов. Для этого входные данные группируются в список, где каждый элемент представляет собой API-запрос с указанием метода, параметров и необходимых данных. Затем этот список структурируется в единый JSON-объект, соответствующий формату пакетного вызова API Bitrix24. Такой подход позволяет отправлять до 50 запросов в одном HTTP-запросе. При отправке пакетного запроса модуль использует асинхронный механизм, чтобы избежать блокировки выполнения программы. HTTP-запрос передается на сервер Bitrix24, где он обрабатывается, а затем возвращается ответ со всеми результатами.

Рассмотрим «Стратегия управления запросами» которая реализуется также в модуле управления вызовами. Управляет выполнением API-запросов, обеспечивая контроль за ограничениями и обработку ошибок. Он работает с данными, связанными с количеством вызовов, таймингами запросов и ошибками, возникающими при превышении лимитов. Основная цель модуля выполнение запросов без нарушения API-ограничений, используя повторные попытки при сбоях и контроль интервалов между вызовами. Ключевая функция модуля контроль за API-ограничениями. Каждому приложению соответствует установленный лимит запросов, после превышения которого сервер может вернуть ошибку о превышении лимита количества запросов. Отслеживает количество вызовов в единицу времени и при необходимости приостанавливает выполнение до восстановления лимита, предотвращая блокировки. Данные, с которыми работает модуль, включают методы API, параметры вызовов, тела запросов и время сброса лимита. Еще одна особенность, объединение вызовов в batch-запросы. Вместо множества отдельных API-вызовов модуль собирает их в пакеты, сокращая нагрузку на сервер и ускоряя выполнение. Если в batch-запросе встречаются ошибки, модуль повторяет только неудачные вызовы, снижая объем повторных обращений и повышая эффективность использования API. Функция, отвечающая за обработку пакетных API-запросов, предназначена для управления вызовов к серверу. При работе, когда выполнение необходимо остановить при первой же ошибке, обработка запроса выполняется блоками. В случае отсутствия ошибок, полученные данные, сохраняются, а работа продолжается с нового блока запросов.

«Модуль интеграционные привязки» выполняет настройку взаимодействия между FastAPI-приложением и Bitrix24. Его работа начинается с конфигурации интеграционного режима, который определяет способ подключения к API Bitrix24. В зависимости от выбранного режима модуль устанавливает механизм формирования URL-адресов и управления авторизацией. Модуль предоставляет возможность регистрации привязок мест встраивания (PlacementBind), которые позволяют добавлять элементы в интерфейс Bitrix24. Для этого формируются запросы с параметрами расположения элементов и их функционала, после чего выполняется пакетная отправка

данных в Bitrix24. Еще одним аспектом является регистрация пользовательских роботов (RobotBind) в Bitrix24, модуль определяет параметры сценария, событие и передаваемые данные, после регистрирует робота.

«Модуль логирования обработки ошибок», предназначенный для фиксации исключений при работе с API Bitrix24. Он включает декоратор, который применяется к методам взаимодействия с API, а также классы исключений, помогающие классифицировать и обрабатывать ошибки.

1.7.2 Рассмотрение API Robokassa

Robokassa предоставляет возможность принимать платежи от клиентов, поддерживая различные способы оплаты, такие как банковские карты, электронные кошельки. Для интеграции биллинг-системы с Robokassa [2] необходимо пройти два этапа, настройку личного кабинета и подключение к API для обработки платежей. Настройка начинается с входа в личный кабинет Robokassa. Для этого необходимо войти в раздел «Управление», перейти на вкладку «Мои магазины» и нажать кнопку «Добавить новый», процесс продемонстрирован на рисунке 6.

При добавлении нового магазина открывается форма, в которой нужно указать регистрационные данные «Наименование магазина» заполняется названием, а в «Идентификатор магазина» вводится уникальный идентификатор, который затем потребуется указать в «Логин продавца». Указывается URL, куда будут поступать запросы. После корректного заполнения всех данных в личном кабинете появляется вкладка «Технические настройки», на которую нужно перейти, как показано на рисунке 7. Здесь из строки «Идентификатор магазина» копируется значение, которое вставляется в поле «Логин». Настройка «Алгоритм расчета хеша» остается без изменений, а для полей «Пароль #1» и «Пароль #2» выполняется генерация значений.

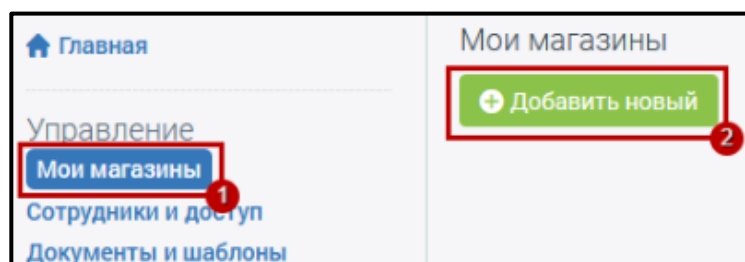


Рисунок 6 – Создание магазина для приема платежей

Следующий этап настройка URL-адресов для взаимодействия с платежной системой. В системе копируются строки «ResultURL», «SuccessURL» и «FailURL», они вставляются в поля в личном кабинете Robokassa. Затем в настройках Robokassa для параметров «Метод отсылки данных по Result URL», «Метод отсылки данных по Success URL» и «Метод отсылки данных по Fail URL» устанавливается метод POST. После завершения этих действий настройки сохраняются. На данном этапе Robokassa предоставляет тестовый режим, который позволяет отладить интеграцию без проведения реальных платежей. Для активации режима в запросах используется параметр «IsTest» с значением «1».

Рисунок 7 – Технические настройки

После настройки можно переходить к взаимодействию с API. Одним из способов взаимодействия с Robokassa является интерфейс оплаты через HTTP-запросы

GET или POST, который перенаправляет на страницу оплаты. Процесс начинается с формирования запроса с параметрами:

- «MerchantLogin», идентификатор магазина;
- «OutSum», сумма платежа;
- «InvId», номер счета;
- «SignatureValue», контрольная сумма;
- «Shp_», дополнительные параметры.

Затем происходит перенаправление на URL:
<https://auth.robokassa.ru/Merchant/Index.aspx?MerchantLogin=...&OutSum=...&InvId=...&Description=...&SignatureValue=<хэш>>.

Взаимодействие включает отправку запросов на оплату, получение уведомлений об оплате на «ResultURL» и обработку переадресаций на «SuccessURL» или «FailURL». Эндпоинт «/result» «ResultURL» обрабатывает уведомления от Robokassa, проверяя подпись с использованием «Пароль#2» и обновляя статус заказа, если подпись верна.

После успешной оплаты Robokassa отправляет уведомление на «ResultURL». Это POST-запрос с параметрами «OutSum», «InvId», «SignatureValue». Проверяется подпись, используя «Пароль#2», если она верна, обновляется статус заказа и возвращается ответ OK<InvId>. Параметры «ResultURL» указаны в таблице 4.

Таблица 4 – Параметры уведомления «ResultURL»

Параметр	Описание
OutSum	Сумма платежа.
InvId	Номер счета.
EMail	Email покупателя (если указан).
SignatureValue	Контрольная сумма: MD5(OutSum:InvId:Пароль#2[:Shp_ параметры]).
PaymentMethod	Способ оплаты (например, BankCard).
IncCurrLabel	Валюта платежа.
Shp_*	Дополнительные параметры, переданные магазином.

Уведомление об оплате, отправляемое на «ResultURL», содержит данные о сумме платежа, номер счета, комиссия Robokassa, контрольная сумма

«SignatureValue». Скрипт в биллинг-системе проверяет подпись, сравнив с собственной расчетной суммой. Ответ системы при совпадении подписей: OK<InvId>.

После завершения оплаты происходит перенаправление на «SuccessURL», проверяя подпись с «Пароль#1». Параметры «SuccessURL» приведены в таблице 5.

Таблица 5 – «SuccessURL» успешная оплата

Параметр	Описание
OutSum	Сумма платежа.
InvId	Номер счета.
SignatureValue	Подпись: MD5(OutSum:InvId:Пароль#1[:Shp_ параметры]).
Culture	Язык страницы оплаты.
Shp_*	Дополнительные параметры.

В случае отказа от оплаты или возникновения ошибки пользователь перенаправляется на адрес, указанный в поле «FailURL». При этом в запросе присутствуют параметры OutSum, InvId, Culture, а также все дополнительные параметры Shp_*, их описание приведено в таблице 6.

Таблица 6 – «FailURL» отказ от оплаты

Параметр	Описание
OutSum	Сумма платежа.
InvId	Номер счета.
Culture	Язык страницы оплаты.
Shp_*	Дополнительные параметры.

Интеграция с API Robokassa включает в себя как предварительную настройку в личном кабинете, так и программную реализацию механизмов обработки платежей и уведомлений. Важным аспектом является корректная работа с контрольными суммами (подписями), а также обеспечение безопасности при передаче данных между системой и Robokassa. Реализация всех этапов позволяет обеспечить работу биллинговой системы с возможностью автоматической фиксации оплаты и информирования пользователей о статусе транзакций.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Общие сведения о программном обеспечении

В рамках работы осуществляется проектирование биллинг-системы, которая представляет собой систему для обработки платежей выполняющую автоматизацию процесса управления заказами, оплаты услуг и упрощения процессов выставления счетов.

Основной задачей биллинг-системы является управление созданием, обработкой и контролем заказов, оплаты и выставление счетов.

Функциями программного обеспечения являются:

- создание и управление заказами;
- выставление счетов на оплату с учётом выбранного способа оплаты;
- обработка платежей с использованием платёжной системы Robokassa;
- хранение истории заказов;
- интеграция с CRM Bitrix24 через смарт-процесс для отображения стадий заказов и синхронизации статусов;
- уведомление пользователей о статусах заказа и оплате;
- подключения к внешнему приложению компании для интеграции.

Требования к проектируемому программному обеспечению были оформлены в виде технического задания в соответствии с ГОСТ 19.201. Полученное техническое задание приведено в приложении Б.

В соответствии с требованиями к составу выполняемых функций проектируемого программного обеспечения была построена UML диаграмма вариантов использования, представленная на рисунке В.1, которая иллюстрирует взаимодействие между ролями и системой, а также ключевые действия системы.

На данной диаграмме видно, организация создает заказ в биллинг- системе. Процесс проведения оплаты инициирует процесс оплаты для обработанного заказа. Оплата передается получателю платежа.

Далее, для созданного заказа получатель платежа инициирует процесс оплаты через вариант использования «Продление оплаты». Этот процесс включает взаимодействие с Robokassa для обработки транзакции.

При этом важно отметить, заказ может находиться в одном из 7 состояний, отражающих этапы его обработки:

- «ожидание выбора способа оплаты», заказ создан, но ещё не выбран метод оплаты;
- «ожидание вебхука оплаты», способ оплаты выбран, ожидается подтверждение транзакции от Robokassa;
- «оплачено», средства зачислены, заказ считается завершённым, и система инициирует активацию услуги;
- «ошибка оплаты», возникла ошибка при обработке платежа;
- «ошибка активации», ошибка при активации данных;
- «ожидание ручного подтверждения оплаты», при оплате по реквизитам, система ожидает проверки поступления средств;
- «ожидание ручного выставления счета», ручное создание счета.

Для понимания логики обработки заказов в биллинг-системе, рассмотрим диаграмму состояний на рисунке 4, которая отражает состояния заказа в процессе его жизненного цикла.

После оплаты обновляется статус заказа на «Оплачено». Получатель также инициирует генерацию счета для заказа.

Счет, связанный с заказом, создается в системе. После подтверждения выполняется продление оплаченной услуги. В системе каждый участник выполняет определённую роль, соотносящуюся с определёнными условиями взаимодействия.

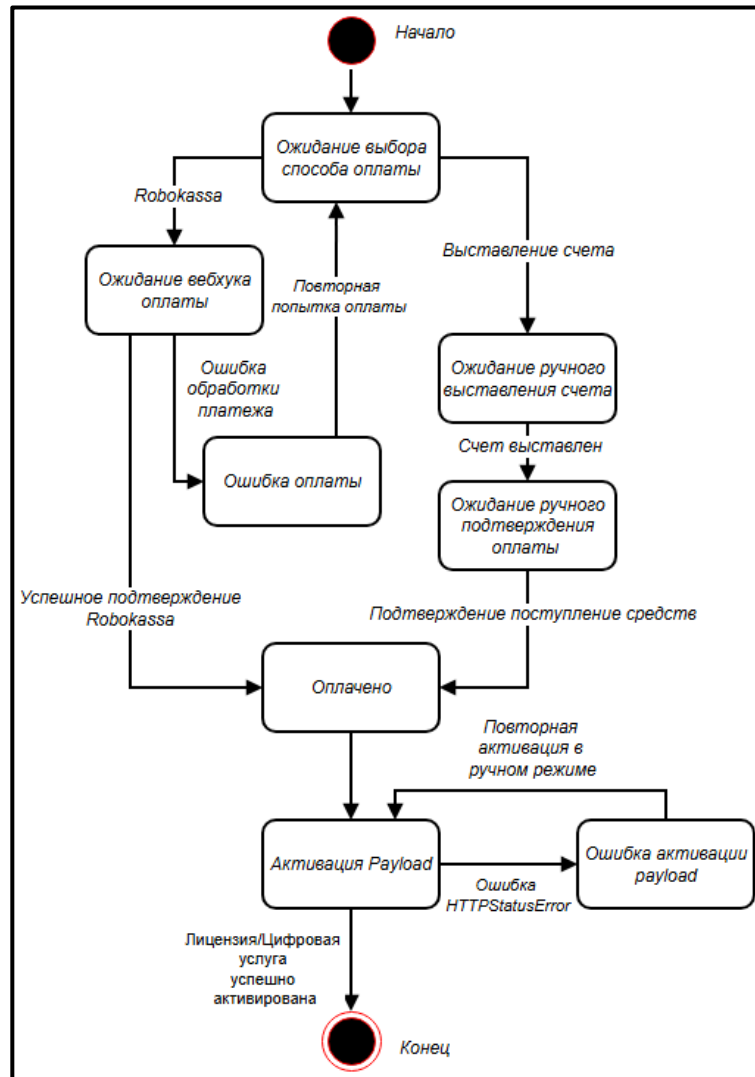


Рисунок 8 – UML-диаграмма состояний заказа

2.2 Функциональная модель

Для описания взаимодействия программного обеспечения используем функциональную модель в нотации IDEF0, приведённую на рисунке Г.1.

С левой стороны представлены входные данные, поступающие в систему. Это в первую очередь данные и конфигурационные параметры, необходимые для работы с внешними API-интерфейсами. Для Robokassa это логин `MERCHANT_LOGIN` и два пароля `MERCHANT_PASSWORD_1`, `MERCHANT_PASSWORD_2`, а для Bitrix24, параметры приложения `IS_CIRCULATION_APP`, веб-хук `C_REST_WEB_HOOK_URL`, идентификатор клиента `C_REST_CLIENT_ID` и секретный ключ `C_REST_CLIENT_SECRET`, необходимые для синхронизации данных с

CRM-системой. Помимо этого, системой обрабатывается информация о плательщике, а также запросы на оплату, инициируемые пользователем, которые через интерфейс системы.

Справа расположены выходные данные, включающие получение оплаты и подтверждение оплаты пользователям, которые информируют клиента об успешной транзакции, а также генерацию счёта для безналичного расчёта, который сохраняется в системе. Система формирует и отправляет вызовы к API Robokassa и Bitrix24, что позволяет наладить двустороннюю интеграцию с внешними сервисами, обеспечивающими платёжные и коммуникационные функции. В нижней части диаграммы расположены механизмы, то есть ресурсы, участвующие в выполнении функций. Пользователь, инициирующий действия, связанные с оплатой, а с другой, администратор, ответственный за настройку «Биллинг-системы». В верхней части указаны управляющие воздействия, которые направляют выполнение функций системы. Это правила оплаты, определяющие способ оплаты через счёт для безналичного расчёта, или Robokassa для онлайн-платежей, что позволяет оплачивать услуги непосредственно через платёжную систему Robokassa, и тарифы, применяются внешним приложением (OLChat) на этапе формирования корзины, которая передается к биллингу. Перейдем к декомпозиции первого уровня, приведённой на рисунке Г.2.

На данной диаграмме представлена декомпозиция функциональной модели «Биллинг-системы» на первом уровне, в которой выделено три ключевых подсистемы: «Бэкенд Биллинг-системы», «Внешнее приложение (OLChat)» и «Модуль интеграции «CRest» с REST API Bitrix24». Далее по тексту под «внешним приложением OLChat» будет пониматься клиентская часть, через которую пользователи оформляют заказы. Каждый из этих компонентов выполняет свою роль в обработке информации о заказах, управлении оплатами и взаимодействии с внешними сервисами. Центральным элементом выступает «Бэкенд Биллинг-системы», который отвечает за обработку заказов, генерацию счетов, прием и подтверждение оплаты, а также про-

дление цифровой услуги. Он принимает запрос на обработку от внешнего приложения (OLChat), который включает сформированный заказ с корзиной товаров или услуг, стоимостью, выбранным способом оплаты и данными для активации услуги.

«Бэкенд Биллинг-системы» использует параметры интеграции с Robokassa для формирования вызовов к API Robokassa, если выбран онлайн-платёж, или генерирует счёт для безналичного расчёта. После подтверждения оплаты система инициирует активацию услуги и передаёт в интегрированное приложение OLChat набор необходимых данных payload. В этот набор включаются параметры подписки, выбранный тариф и период, идентификатор портала, а также дополнительные настройки, которые требуются для корректного запуска или продления услуги. OLChat, получив эти данные по API, автоматически активирует соответствующую функцию для пользователя. Также происходит передача о изменении статуса заказа через «CRest» в Bitrix24. Администратор, настраивает параметры для работы с Robokassa. «Внешнее приложение (OLChat)», через него пользователь инициирует запрос продление услуг. Оно принимает данные плательщика и содержащий сформированный заказ, и передаёт их в виде запроса на обработку в «Бэкенд Биллинг-системы». После обработки заказа внешнее приложение получает от «Бэкенд Биллинг-системы» объект для активации услуги. «CRest» предназначен для интеграции «Бэкенд Биллинг-системы» с CRM Bitrix24 через смарт-процесс, который обеспечивает отслеживание жизненного цикла заказов, синхронизацию данных о заказе, а также контроль статусов заказов после проведения транзакций. «CRest» принимает от «Бэкенд Биллинг-системы» данные о заказе и использует их для взаимодействия с Bitrix24 через API.

Таким образом, «Бэкенд Биллинг-системы» реализует обработку платежей, «Внешнее приложение (OLChat)» обеспечивает взаимодействие и маршрутизацию запросов в «Бэкенд Биллинг-системы», а «CRest» занимается интеграцией с внешней CRM-системой.

2.3 Архитектурный проект

2.3.1 Выбор архитектуры разрабатываемой биллинг-системы

Оценим архитектуру разрабатываемой биллинг-системы от которой зависит структура системы, её производительность и способность к интеграции с внешними сервисами, такими как Robokassa и Vitrix24. Наиболее распространёнными подходами к созданию подобных систем являются монолитная архитектура, отличающаяся простотой реализации, и микросервисная архитектура, обеспечивающая гибкость и масштабируемость. [3] Начнём рассмотрение с традиционной монолитной архитектуры, чтобы определить её применимость для разрабатываемого программного обеспечения, выявим преимущества и недостатки. Монолитная архитектура на рисунке 9, представляет подход, когда приложение состоит из пользовательского интерфейса, базы данных и бизнес-логики.

В монолитной архитектуре изменения, обновления или исправления ошибок требуют переразвертывания всего приложения. [4] Рассмотрим вызов функции из одного модуля в другой происходит напрямую, без необходимости использования сетевых протоколов или API, что обеспечивает высокую скорость выполнения операций. Для биллинг-системы это может означать, обновление логики интеграции с Vitrix24 потребует остановки и перезапуска всего приложения, включая интерфейс и модули взаимодействия с подключаемыми приложениями.

Достоинства такой архитектуры простота разработки и отладки, все компоненты системы находятся в едином кодовом базисе. Это упрощает тестирование взаимодействия модулей, например, обработки платежей, интеграции с Vitrix24 или генерации счетов, поскольку вызовы между ними происходят напрямую, без сетевых задержек, а производительность достигается за счет отсутствия необходимости в межсервисном взаимодействии.

Недостатки монолитной архитектуры сложность ограниченная масштабируемость: при росте нагрузки на отдельные модули, потребуется масштабировать всю систему целиком, включая менее нагруженные компоненты, такие как интерфейс

корзины или модуль отчетности. Это приводит к неэффективному использованию ресурсов. Зависимость компонентов создает риски сбоев, например, ошибка в интеграции с Bitrix24, из-за превышения лимитов API, может остановить работу всего приложения, включая выставление счетов, эту проблему можно решить, если взаимодействие с сторонним API вынесено в отдельный микросервис.

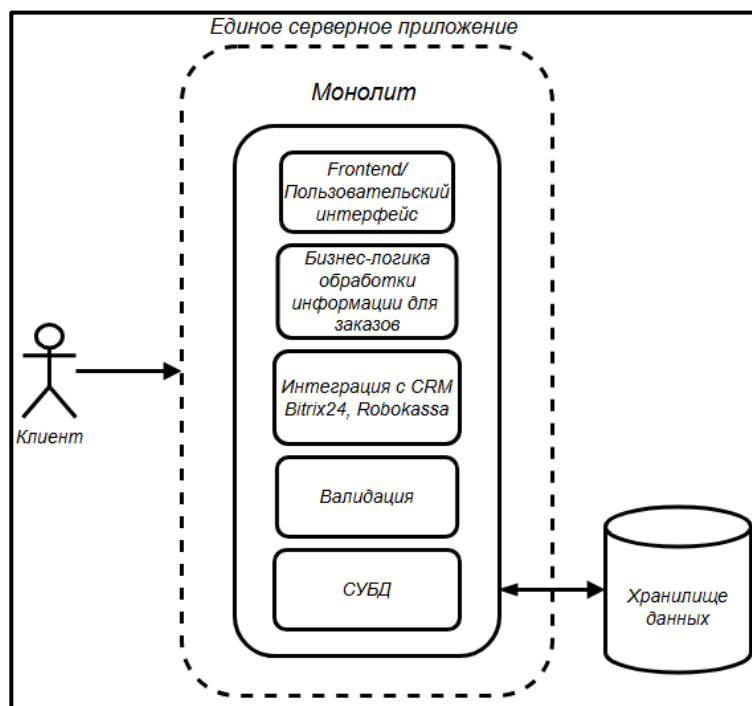


Рисунок 9 – Монолитная архитектура

Обновление системы требует полной пересборки и перезапуска, что усложняет внедрение новых функций. Кроме того, монолитная архитектура менее гибка в плане поддержки интеграции с внешними сервисами, если бизнес-процессы требуют частых модификаций. Эти недостатки делают монолитную архитектуру менее пригодной для биллинг-системы, требующей интеграции с внешними сервисами.

Учитывая ограничения монолитной архитектуры рассмотрим альтернативный подход микросервисной архитектуры изображенной на рисунке 10. Этот подход позволяет разбить систему на независимые сервисы, каждый из которых отвечает за конкретную бизнес-функцию, а также взаимодействуя через API. [5] Для биллинг-си-

системы это означает, что управление заказами, взаимодействие с внешним приложением (OLChat) и интеграция с Bitrix24 через CRest, могут быть реализованы как отдельные сервисы.

Монолитные системы объединяют все сервисы приложения в один процесс, а потом дублируют эту систему на нескольких серверах. Микросервисная архитектура для каждого сервиса имеет отдельную упаковку, а на разных серверах они хранятся также изолированно друг от друга, дублируясь, если есть необходимость. Это позволяет отдельно работать с каждым функциональным элементом. Такая независимость позволяет написать сервисы на Python, другой на JavaScript для фронта.

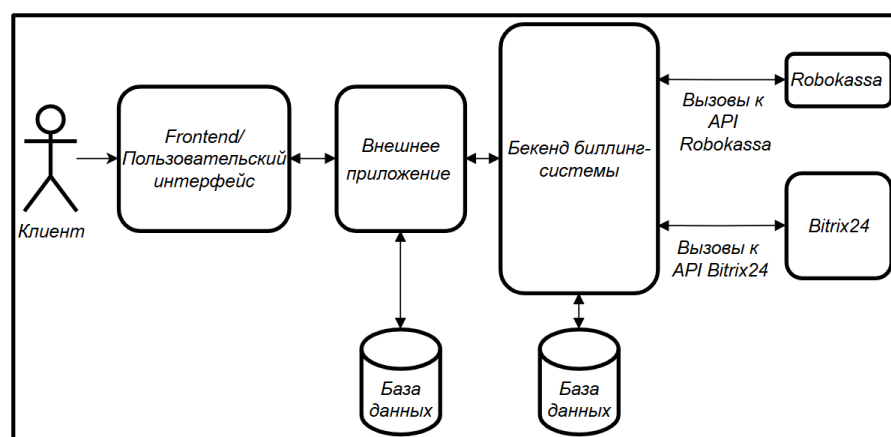


Рисунок 10 – Микросервисная архитектура

Рассмотрим преимущества микросервисной архитектуры:

- отказоустойчивость, при падении одного из сервисов, все остальные остаются в строю;
- гибкость, можно попробовать внедрить новую технологию малыми усилиями. Меняя локально один из сервисов, мы не рискуем всей системой и время, требующееся для изменений, меньше;
- простота, каждый отдельный сервис представляет собой цельную систему, поэтому не нужно разбираться в огромном количестве деталей, не касающихся данной конкретной функции, тем проще разобраться, что и как работает;
- масштабируемость, сервисы можно дополнить и расширить, когда появится такая необходимость. Можно обновить модуль интеграции с Robokassa, не затрагивая

интеграцию с API Bitrix24.

Микросервисная архитектура предполагает распределённое управление данными. В отличие от монолита, где все данные обычно хранятся в одной базе данных, в микросервисах каждый сервис может иметь собственную базу данных, что повышает изолированность системы. Для биллинг-системы используется PostgreSQL для хранения данных о заказах, в то время как CRest, обеспечивающий связь с CRM Bitrix24, использует отдельную базу данных для хранения параметров авторизации и учётных данных, связанных с внешним API. К недостаткам микросервисной архитектуры можно отнести сложность разработки, т.к. разделение на сервисы требует управления взаимодействием, задержки в коммуникации между сервисами.

После анализа подходов предпочтение было отдано микросервисной модели. Микросервисная архитектура позволяет разбить систему на модули. Например, если потребуется подключить новую платёжную систему, достаточно будет обновить модуль интеграции с платёжными системами. Аналогично, при необходимости изменить логику передачи данных в Bitrix24, например, добавить новые поля в смарт-процесс, изменения коснутся только интерфейса взаимодействия через CRest. Одним из достоинств микросервисов является возможность масштабировать отдельные компоненты системы в зависимости от нагрузки. В случае биллинг-системы нагрузка на создание заказов через бекенд может расти значительно быстрее, чем на интеграцию с Bitrix24 или интерфейс внешнего приложения (OLChat).

2.3.2 Архитектура разрабатываемой биллинг-системы

В соответствии с принципами микросервисной архитектуры, разработанная биллинг-система разделена на несколько модулей, которые представлены на диаграмме компонентов программы в приложении Д.1.

На схеме выделены три компонента: «Бэкенд биллинг-системы», «Внешнее приложение (OLChat)» и «CRest», каждый отвечает за свою задачу и взаимодействуют через интерфейсы.

«Бэкенд биллинг-системы», отвечает за обработку заказов, управление платежами и генерацию счетов. Его задача заключается в обеспечении жизненного цикла

заказа, от получения запроса на оплату до подтверждения транзакции и активации услуги. принимает запросы от внешнего приложения по HTTP/HTTPS, содержащие данные о сформированном заказе и взаимодействует с другими модулями для выполнения этих операций, как показано на рисунке Е.1. Для обработки платежей «Бэкенд биллинг-системы» взаимодействует с Robokassa через API, формируя вызовы для инициации оплаты или получения уведомлений о статусе транзакции. После оплаты «Бэкенд биллинг-системы» обновляет статус заказа, передает данные в Bitrix24 через CRest и возвращает внешнему приложению подтверждение оплаты с данными активации услуги.

«Внешнее приложение», выполняет роль интерфейса взаимодействия с пользователем и маршрутизатора запросов в «Бэкенд биллинг-системы». Оно предоставляет интерфейс взаимодействия с фронтом, через который пользователи оформляют заказы выбирая услуги, для внешнего приложения (OLChat), например, подключение линий WhatsApp или групп Telegram, добавляют их в корзину и инициируют оплату. «Внешнее приложение (OLChat)» взаимодействует с «Бэкенд биллинг-системы» по HTTP/HTTPS, передавая данные о заказе в формате JSON.

«CRest» отвечает за интеграцию биллинг-системы с CRM Bitrix24, обеспечивая синхронизацию данных и управление жизненным циклом заказов. CRest принимает данные о заказах и их статусах от «Бэкенд биллинг-системы» и передает их в Bitrix24 через API. CRest построена на принципе серверно-клиентского взаимодействия: серверная часть модуля реализует REST API для обработки входящих запросов, а клиентская часть взаимодействует с Bitrix24 через его API, выступая в роли посредника. CRest управляет маршрутизацией данных и авторизацией через OAuth 2.0. На рисунке Е.2 представлена UML-диаграмма коммуникаций, описывающая процесс взаимодействия приложения, интегрированного с CRest. Сначала клиент, желающий получить доступ к API Bitrix24, предоставляет адрес портала и инициирует процесс авторизации. Приложение с модулем CRest направляет на авторизацию в Bitrix24, передавая в запросе идентификатор приложения. После авторизации происходит перенаправление обратно в приложение с временным кодом авторизации.

Для иллюстрации жизненного цикла `access_token` начиная с его инициализации, использования, проверки валидности и вплоть до обновления или истечения срока действия целесообразно обратиться к диаграмме состояний, которая приведена на рисунке Е.3. Токен выдаётся после авторизации по `client_id`, `client_secret`, `authorization_code`, действует ограниченное время и по истечении обновляется через `refresh_token`. Пока срок действия `access_token` не вышел, приложение через CRest подписывает им все REST-вызовы к API Bitrix24. При обнаружении истечения срока `expires_in` действия токена запрашивается новая пара токенов, отправляя запрос на <https://oauth.bitrix.info/oauth/token>, в котором вместо кода авторизации используется `refresh_token`. Обмен токенами выполняется по HTTPS. Получив новую пару токенов, CRest сохраняет полученные аутентификационные параметры в базе данных. При работе с конкретным приложением, CRest получает одну пару `access_token` и `refresh_token`, закреплённая именно за приложением. Сколько угодно пользователей могут пользоваться интерфейсом самого приложения, но все их запросы к REST-API уходят «от имени» приложения и подписываются той же самой единственной парой токенов. Отдельные токены на каждого пользователя не требуются, что упрощает управление доступом и минимизирует риск утечек аутентификационных данных.

2.4 Проектирование базы данных

2.4.1 Инфологическое проектирование

Проектирование базы данных включает три этапа: инфологическое, логическое и физическое проектирование.

Инфологическое проектирование является первым этапом, направленным на концептуальное описание данных, необходимых для функционирования системы, без привязки к конкретным техническим решениям. Задача инфологического проектирования выделить сущностей, атрибутов и связей между ними. [6]

Из архитектурного проекта можно выделить сущности:

- «Организация» описывает юридическое лицо;
- «Получатель платежа» представляет организацию принимающую платежи;
- «Приложение» позволяет связать заказы с конкретным приложением;

- «Заказ» является центральной сущностью биллинг-системы, отражающей заказ;

- «Токен» отражает выданный приложению ключ для вызовов API биллинга.

Следующим этапом инфологического проектирования является определение атрибутивного состава выделенных сущностей, а также ключевых атрибутов.

Полный перечень атрибутов сущности «Организация», включающий как идентификаторы, так и контактные данные, а также реквизиты, представлен в таблице Ж.1 приложения Ж.

Атрибуты сущности «Получатель платежа», продемонстрированной в таблице Ж.2 приложения Ж, подобраны таким образом, чтобы обеспечить формирования платёжных документов: для хранения сведений используется поле в формате JSON.

Для сущности «Приложение» структура атрибутов продемонстрирована в таблице Ж.3. Она отражает требования интеграции здесь задаются как идентификатор, ключи для работы с API биллинга.

В структуре сущности «Заказ», изображенной в таблице Ж.4, приложения Ж, важным является использование формата JSON для хранения данных корзины и параметров активации услуги.

Сущность «Токен», проиллюстрированной в таблице Ж.5, предназначена для хранения ключей, используемых при вызовах API биллинга.

Следующим этапом инфологического проектирования является определение связей между сущностями. Между представленными сущностями были определены следующие связи: «Организация – Заказ», «Заказ – Получатель платежа», «Заказ – Приложение», «Приложение – Токен». Описание данных связей представлено в таблице Ж.6. Полученная в результате инфологического проектирования ER-диаграмма приведена на рисунке Ж.1.

2.4.2 Логическое и физическое проектирование

Логическое проектирование представляет собой следующий этап после концептуального моделирования. На этом этапе происходит преобразование концептуальной модели в реляционную, а также выполняется нормализация отношений, чтобы

обеспечить целостность данных, перевести сущности и связи между ними, в структуру таблиц, атрибутов и ключей.

Процесс отображения концептуальной модели на реляционную начинается с преобразования сущностей в отношения, а связей между сущностями в связи между отношениями. [7] Для связей «один к одному» с обязательным классом принадлежности обеих сущностей создаётся единое отношение, объединяющее атрибуты обеих сущностей, причём первичным ключом может стать любой из первичных ключей этих сущностей. Если классы принадлежности различаются, то формируются два отношения: одно для сущности с обязательным классом принадлежности, которое считается родительским, и другое для сущности с не обязательным классом, которое становится дочерним, а первичный ключ родительского отношения добавляется в дочернее для установления связи. Когда обе сущности имеют не обязательный класс принадлежности, возникает необходимость в трёх отношениях: по одному для каждой сущности и дополнительное отношение для связи, содержащее первичные ключи обеих сущностей.

Для связей «один ко многим» подход зависит от класса принадлежности. Если он обязателен, то односвязная сущность становится родительской, а многосвязная дочерней, создаются два отношения, и первичный ключ родительской сущности включается в дочернее. Если же класс принадлежности многосвязной сущности не обязателен, то формируются три отношения: два для сущностей и одно для связи, включающее первичные ключи обеих. В случае «многие ко многим» структура неизменна вне зависимости от класса принадлежности.

Классы принадлежности определяется характером связи между сущностями. Если экземпляр одной сущности обязательно связан с экземпляром другой, то класс принадлежности считается обязательным. Если же связь не является обязательной и экземпляры могут оставаться несвязанными, класс принадлежности не обязательный.

На этапе инфологического проектирования биллинг-системы были выделены сущности: «Организация», «Получатель платежа», «Приложение» и «Заказ», а также

связи между ними: «Организация – Заказ», «Получатель платежа – Заказ» и «Приложение – Заказ». Необходимо преобразовать сущности, связи в реляционную модель, определив классы принадлежности и добавив соответствующие ключи.

На следующем этапе были проанализированы связи между сущностями. Результаты рассмотрения продемонстрированы в таблице И.1, а также на рисунках И.1 – И.8. Для каждой пары сущностей указаны кратность связи, класс принадлежности для каждой сущности, а также способ отображения этой связи в структуре реляционной базы данных. После преобразования связей, включающими «Организацию», «Получателя платежа», «Приложение», «Заказ» и «Токен», были получены реляционные отношения, которые отражают структуру данных с учётом внешних ключей, «Идентификатор плательщика», «Идентификатор получателя» и «Идентификатор приложения» в таблице «Заказ», как показано в приложении И на рисунке И.9.

Следующим этапом логического проектирования является нормализация отношений, целью которой является устранение избыточности данных и обеспечение их целостности. Для нормализации отношений реляционной базы данных необходимо привести все её отношения к 3НФ, проходя 1НФ и 2НФ.

Для приведения отношений к первой нормальной форме требуется, чтобы все атрибуты были атомарными, то есть содержали единственное значение и не представляли собой списки или множества.

Отношения «Организация», «Получатель платежа», «Приложение», «Токен» и «Заказ» удовлетворяют требованиям 1НФ, поскольку их атрибуты атомарны и содержат единственное значение. JSON-атрибуты также считаются атомарными, так как их внутренняя структура не анализируется базой данных.

Для приведения отношений к 2НФ они должны находиться в 1НФ, а каждый не ключевой атрибут должен зависеть от первичного ключа. Отношения «Организация», «Получатель платежа», «Приложение», «Заказ» и «Токен» имеют простые первичные ключи «Идентификатор организации», «Идентификатор получателя», «Идентификатор приложения» и «Идентификатор заказа», автоматически приводит их во вторую

нормальную форму, в них отсутствуют составные ключи, которые могли бы вызвать частичные зависимости.

Третья нормальная форма требует, чтобы отношения находились в 2НФ и не содержали транзитивных зависимостей, то есть ситуаций, когда не ключевой атрибут зависит от другого не ключевого атрибута, который, в свою очередь, зависит от первичного ключа. Во всех указанных отношениях «Организация», «Получатель платежа», «Приложение», «Заказ» и «Токен» транзитивные зависимости отсутствуют.

Отношение «Организация». Атрибуты «Название», «Юридический адрес» или «Электронная почта», зависят от первичного ключа «Идентификатор организации». Атрибут «Код страны» зависит от первичного ключа, его значение (RU, BY, KZ) определяется для каждой организации независимо. Транзитивная зависимость могла бы возникнуть между атрибутами «Идентификатор в Bitrix24», «Идентификатор портала», атрибуты являются независимыми идентификаторами, задаваемыми Bitrix24.

Отношении «Приложение», атрибуты «URL для продления услуги», «Инструкции», зависят от «Идентификатор приложения» транзитивных зависимостей нет.

В отношении «Токен» атрибуты, «Идентификатор приложения» внешний ключ, но от него не зависит ни «Ключ доступа», ни «Дата выпуска» эти поля выдаются при генерации токена и остаются неизменными.

В отношении «Заказ» атрибуты «Код валюты», «Способ оплаты» и «Статус», являются перечислениями, их значения зависят от «Идентификатор заказа». «Код валюты» не зависит от других не ключевых атрибутов, валюта заказа определяется при его создании. Внешние ключи «Идентификатор приложения», «Идентификатор плательщика», «Идентификатор получателя» не создают транзитивных зависимостей.

Зависимости атрибутов отношений представлены на рисунке И.10. В отношениях отсутствуют транзитивные зависимости, они находятся в 3НФ. Итоговая IDEF1X-диаграмма после логического проектирования приведена на рисунке И.11.

Заключительный этап проектирования создание физической модели. [8] На этом этапе формируются структуры данных для каждого отношения, с указанием их типов, ограничений, форматов и индексации, соответствующих выбранной СУБД.

Физические структуры данных для отношений, «Организация», «Получатель платежа», «Приложение», «Заказ» и «Токен», представлены в таблицах И.2 – И.6. Каждая таблица содержит перечень атрибутов, их типы данных, условия, формат и сведения об индексации, что отражает требования к созданию соответствующих таблиц в базе данных. В результате выполнения физического проектирования базы данных была получена физическая модель базы данных, приведённая на рисунке И.12.

2.5 Выбор средств реализации проекта

2.5.1 Выбор средств реализации для Бэкенд биллинг-системы

Выбор технологий для реализации бэкенда биллинг-системы оказывает влияние на эффективность разработки, производительность и сопровождения системы.

В качестве языка программирования был выбран Python, так как его лаконичный и читаемый синтаксис позволяет ускорить разработку и упростить поддержку бизнес-логики. Python предоставляет необходимые инструменты для быстрой интеграции с внешними системами, включая платёжные сервисы, и поддерживает асинхронную модель обработки запросов. После выбора Python необходимо было определиться с серверным фреймворком. Рассматривались варианты: Django и FastAPI. В качестве серверного фреймворка был выбран FastAPI, поскольку он демонстрирует более высокую производительность за счёт встроенной поддержки асинхронности на основе библиотек Starlette и asyncio. В отличие от Django, FastAPI более лёгок и не требует использования всего набора компонентов MVC-архитектуры, таких как встроенная админ-панель или шаблонизатор. Вместо этого FastAPI позволяет эффективно реализовать REST API и автоматизировать генерацию документации с помощью стандарта OpenAPI. Встроенная автоматическая валидация входных данных через библиотеку Pydantic упрощает обработку запросов и повышает надёжность данных. Дополнительным преимуществом FastAPI является поддержка механизмов авторизации, с использованием токенов JWT.

Для хранения данных в Бэкенд биллинг-системы требуется СУБД. В рамках проекта была выбрана PostgreSQL, она поддерживает сложные запросы, включая ре-

курсивные функции, что делает ее подходящей для аналитики транзакций и отчетности в биллинг-системе. PostgreSQL также предлагает расширенные возможности, такие как поддержка JSON/JSONB для хранения полуструктурированных данных, что полезно для управления дополнительной информацией о заказах или клиентах. Кроме того, PostgreSQL обеспечивает строгую транзакционную согласованность, что важно для финансовых систем, где недопустимы ошибки в данных. Для работы с базой данных была выбрана библиотека SQLAlchemy, обеспечивающая удобный ORM-слой с поддержкой асинхронных запросов через драйвер asyncpg. SQLAlchemy упрощает манипуляции с реляционными данными, включая операции CRUD и выполнение сложных запросов с использованием типов данных PostgreSQL, таких как JSON. SQLAlchemy позволяет легко работать с такими данными, выполняя запросы, такие как фильтрация заказов по содержимому корзины или дополнительной информации. Для управления миграциями схемы базы данных применяется Alembic.

Для реализации асинхронного взаимодействия с внешним приложением (OLChat), платёжной системой Robokassa, был выбран HTTP-клиент httpx. Особенностью решения является поддержка асинхронного выполнения HTTP-запросов, что интегрируется в асинхронную модель работы FastAPI. При активации услуги система формирует JWT-токен и отправляет его асинхронным POST-запросом на URL внешнего приложения через AsyncClient.

2.5.2 Выбор средств реализации для модуля интеграции «CRest» с REST API Bitrix24

При выборе технологий для реализации модуля интеграции «CRest», взаимодействующего с REST API платформы Bitrix24, основными требованиями были асинхронность обработки, работа под нагрузкой, поддержка стандарта обмена данными JSON и соответствие особенностям API Bitrix24.

Язык Python был выбран исходя из возможностей быстрой интеграции с внешними API, наличия подходящих библиотек и удобного синтаксиса. Для организации асинхронных HTTP-запросов используется библиотека httpx, которая позволяет фор-

мировать POST-запросы к API Bitrix24, обрабатывать ответы в формате JSON, контролировать время ожидания и управлять исключениями, связанными с превышением лимитов API или истечением токенов. Дополнительная логика повторных попыток и обработки ошибок реализована на уровне самого модуля интеграции.

Серверная часть модуля интеграции реализована на FastAPI. Этот фреймворк используется для обработки запросов от внешних приложений, регистрации событий и вебхуков от Bitrix24. JSON выбран как стандартный формат обмена данными, соответствующий требованиям API Bitrix24. Он обеспечивает простоту парсинга и поддержку сложных структур, хотя при больших объемах данных производительность может снижаться, что решается оптимизированными методами обработки. Для взаимодействия с Bitrix24 применяются HTTP-запросы, формируемые модулем CRest с учётом требований к аутентификации через OAuth 2.0 или вебхуков. Подключение к API организовано с учётом ограничений платформы Bitrix24 лимитов по количеству запросов, необходимости пакетной отправки (batch) и обновления токенов.

Для управления очередями запросов и кэширования промежуточных результатов используется Redis. за счет быстрого действия, поддержки различных структур данных и удобного механизма публикации-подписки, позволяющего управлять событиями в рамках интеграции. Параметры авторизации и интеграционные настройки сохраняются в СУБД PostgreSQL, взаимодействие с которой организовано с использованием SQLAlchemy.

2.5.3 Выбор средств реализации фронтенд части

Клиентскую часть системы решено строить на стеке JavaScript с Vue. Этот фреймворк задействует декларативную компонентную модель и встроенную реактивность, поэтому позволяет быстро собирать сложные интерфейсы и при этом остаётся понятным. В качестве сборщика используется Vite, он запускает проект практически мгновенно, держит перезагрузку без заметных задержек и «понимает» SFC-файлы Vue без дополнительной конфигурации. Для маршрутизации подключён Vue Router, а состояние форм и пользовательского ввода хранится в реактивных источниках Vue.

Визуальный слой построен на связке PrimeVue и Tailwind CSS. Библиотека компонентов PrimeVue даёт формы, таблицы и уведомления, а Tailwind отвечает за утилитарную стилизацию и позволяет настроить адаптивность. Пакет `@primevue/themes` обеспечивает палитру, а `primeicons` добавляет иконографику. Автоматическое подключение компонентов настроено через `unplugin-vue-components` вместе с `@primevue/auto-import-resolver`. Сборка и локальный сервер работают на Vite 5.

Кодовая база поддерживается инструментами качества. ESLint с плагином `eslint-plugin-vue` отслеживает нарушения стиля и потенциальные ошибки ещё на этапе сохранения файла. Конфигурация Prettier подключена через `@vue/eslint-config-prettier`, чтобы правила форматирования не конфликтовали с линтером. Vite на запуске плагины `vite-plugin-eslint`, предупреждая о проблемах прямо в браузерной консоли. Sass выбран как препроцессор для редких случаев, когда требуется вычислительная мощь CSS-переменных или вложенные правила.

2.5.4 Технологии деплоя программного обеспечения

Развёртывание бекенда биллинг-системы требует инструментов, которые обеспечивают изоляцию компонентов, стабильность работы и возможность масштабирования при росте нагрузки. В данном проекте для этих целей была выбрана контейнерная платформа Docker, которая позволяет упаковать сервис, включающий FastAPI-приложение для обработки заказов, платежей и модуля интеграции «CRest» с Bitrix24, в изолированный контейнер. В сочетании с Docker используется Traefik, как обратный прокси для маршрутизации запросов и обеспечения безопасного HTTPS-соединения. Конфигурация развёртывания реализована через `docker-compose` и `Dockerfile`.

В файле `docker-compose.yml` выполняется настройка для работы в контейнере использует файл окружения для передачи переменных. Сервис подключается к внешней сети `proxynet`. В процессе сборки контейнера будут также установлены системные зависимости `libreoffice-writer`, `libreoffice-java-common`, `fonts-liberation`, `dos2unix`, `locales` и `default-jre-headless`, необходимы для генерации счетов в формате PDF, которые создаются с использованием LibreOffice.

3 РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Прежде чем приступить к проектированию собственной формы оплаты, полезно изучить, как эту задачу решают крупные сервисы. Изучение подобных примеров позволяет выявить устойчивые и эффективные интерфейсные практики, способствующие упрощению процесса оплаты и снижению количества незавершённых заказов.

Одним из ориентиров для проектирования послужил интерфейс платёжной формы системы Stripe, как показано на рисунке 11, отличающийся высоким уровнем визуальной организации и логической последовательности. Интерфейс разделён на два блока: информационный и платёжный. В левой части отображаются ключевые параметры заказа: стоимость, описание товара и дополнительные условия.

Правая часть содержит форму ввода платёжных данных с возможностью выбора способов оплаты: банковская карта, платёжные сервисы и электронные кошельки. Все элементы расположены таким образом, чтобы минимизировать количество действий и визуальных отвлечений, обеспечивая понятный и предсказуемый пользовательский сценарий. Важной особенностью является использование узнаваемых логотипов платёжных систем, а также минимальное количество полей, что сокращает время заполнения и повышает доверие к процессу. Расширенные способы оплаты, такие как Apple Pay и Google Pay, свёрнуты до компактного представления, не перегружающего интерфейс. Подтверждающие надписи и элементы обратной связи, например, маскированный ввод и визуальные подсказки, помогают избежать ошибок. Кнопка оплаты оформлена в виде визуального акцента с чётким действием, размещённого в нижней части формы, что соответствует типичным ожиданиям пользователей.

С учётом выявленных положительных практик, таких как двухколоночное распределение информации, использование типографической иерархии, ограничение количества визуальных акцентов была разработана собственная форма оформления

подписки в составе интерфейса биллинг-системы, отражающий указанные принципы, представлен на рисунке 12.

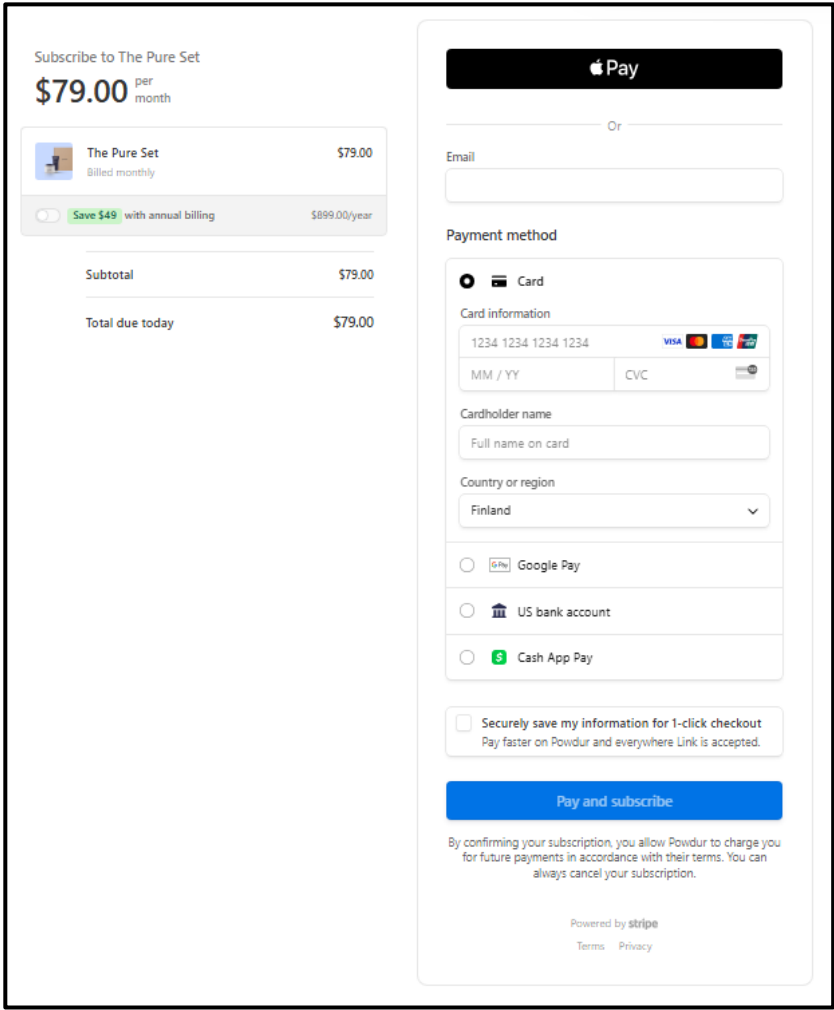


Рисунок 11 – Форма Stripe

На этом экране пользователь открывает страницу «Оформление подписки». В левой части, чуть ниже крупного заголовка, сразу виден выпадающий список для выбора компании. Как только компания выбрана, под комбобоксом раскрывается аккуратная визитка-карточка: крупным шрифтом повторяется название, правее ОГРН, а под ним двумя колонками располагаются юридический адрес и пара строк с ИНН / КПП. Ниже, в отдельном подблоке «Контакты», вынесены номер телефона и почта. Справа от заголовка карточки расположен небольшой значок «три точки» через него пользователь в любой момент может отредактировать информацию о компании, добавить новую или удалить компанию, если за ней не числятся активные подписки.

Оформление подписки

Компания

НАО "ИНФОКОМ" ▼

Название	ОГРН
НАО "ИНФОКОМ"	1087746418033
Юридический адрес	ИНН / КПП
г Москва, Рязанский пр-кт, д 8А стр 1, помещ III, ком 18	7717615385 772101001

Контакты

Номер телефона	Электронная почта
+7 (923) 423-44-35	corp_infocom@gmail.com

Портал

Выберите портал ▼

В корзине пока ничего нет
Здесь будут отображаться
выбранные пакеты
подписок

Перейти к оплате

Рисунок 12 – Оформление подписки

Если пользователю нужно добавить новую организацию, на экране появляется модальное окно-карточка, как показано на рисунке 13 фокусирующее внимание на вводе реквизитов новой компании. Вверху крупный заголовок «Компания», сразу под ним раскрывающийся список «Страна» с красной звёздочкой сейчас там уже выбран пункт «Россия». Ниже активное текстовое поле «Название компании» тоже отмечено обязательным индикатором; как только пользователь начинает вводить название, происходит автопоиск, и оставшиеся реквизиты заполняются автоматически. Под этим полем на сероватой подложке представлен предпросмотр карточки: название, юридический адрес, ОГРН, ИНН и КПП выстроены в две колонки.

Блок «Контакты» отделён жирным подзаголовком, первым идёт обязательное поле телефонного номера, оно ожидает формат +7 и подсвечивает маску во время ввода, дальше поле для корпоративной почты, тоже обязательное, подсказывает валидный домен и не даст сохранить запись, если пропущен знак «@». В правом нижнем углу окна две кнопки: слева текстовое «Отмена» возвращает к предыдущему экрану без сохранения, справа яркая синяя кнопка «Добавить» активна, только когда все обязательные поля прошли валидацию.

Компания

Страна *

Россия

Название компании *

КБ "РЕНЕССАНС КРЕДИТ" (ООО)

Название	ОГРН
КБ "РЕНЕССАНС КРЕДИТ" (ООО)	1027739586291
Юридический адрес	ИНН / КПП
г Москва, ул Кожевническая, д 14	7744000126 772501001

Контакты

Номер телефона *

+7 (800) 353-53-53

Электронная почта *

bankren@gmail.com

Отмена **Добавить**

Рисунок 13 – Добавление информации о организации

Следующий блок поле «Портал», продемонстрированный на рисунке 14. Оно визуально отделено серой плашкой, продолжающей стиль карточек. Пока портал не выбран, отображается «Выберите портал». Биллинг открыт именно для приложения OLChat, поэтому доступны только те порталы, где оно установлено.

Вся правая колонка целиком занята блоком корзины. Когда пользователь только заходит на страницу, корзина демонстрирует пустое состояние: серый значок тележки, строчка «В корзине пока ничего нет» и поясняющий подпункт о том, что здесь будут отображаться выбранные пакеты подписок. Кнопка «Перейти к оплате» окрашена в бледно-серый цвет и недоступна.

Как только из нижнего каталога добавляется хотя бы один пакет, внутри карточки плитки выбранных услуг с их краткими названиями, количеством и ценой, внизу динамически пересчитывается итог, а большая кнопка «Перейти к оплате» активируется и подсвечивается.

Оформление подписки

Компания

НАО "ИНФОКОМ"

▼

Название

НАО "ИНФОКОМ"

Юридический адрес

г Москва, Рязанский пр-кт, д 8А стр 1, помещ III, ком 18

ОГРН

1087746418033

ИНН / КПП

7717615385
772101001

Контакты

Номер телефона

+7 (923) 423-44-35

Электронная почта

corp_infocom@gmail.com

В корзине пока ничего нет

Здесь будут отображаться
выбранные пакеты
подписок

Перейти к оплате

Портал

test_0.domain.ru

▼

test_0.domain.ru

test_1.domain.ru

test_2.domain.ru

test_3.domain.ru

test_4.domain.ru

test_5.domain.ru

Рисунок 14 – Выбор параметров

Ниже блока с порталом раскрывается секция «Тип» первый селект-фильтр, от которого зависит состав позиций, доступных для продления, как показано на рисунке 15. По умолчанию показан вариант «Линии WhatsApp» это виртуальные номера, к которым приложение OLChat подключает диалоги из мессенджера. Как только пользователь меняет значение на «Группы WhatsApp», в таблице ниже подтягиваются названия и ID групп; логика та же, но набор колонок и иконки будут другими. Под селектом отображается таблица с чек-боксами в первой колонке. Каждая строка показывает отображаемое имя линии, служебный комментарий мелким серым шрифтом, подключённый номер и текущую дату окончания подписки. Если у линии подписка истекла, дата подсвечена красным; если активна – обычный серый. Пользователь может отметить сразу несколько строк, чтобы продлить их одинаковым тарифом, либо оставить все снятыми и продлить позже в корзину попадут только отмеченные.

Сразу под таблицей идут стилизованные тарифные плитки. Карточки подсвечены плавным градиентом, чтобы подчеркнуть тарификацию «корзины», и ведут себя

как радио-группа: активная карточка выделяется тонким контуром и ярким шильдиком-точкой в правом верхнем углу. Стандартный набор периодов 30 / 90 / 180 / 360 дней.

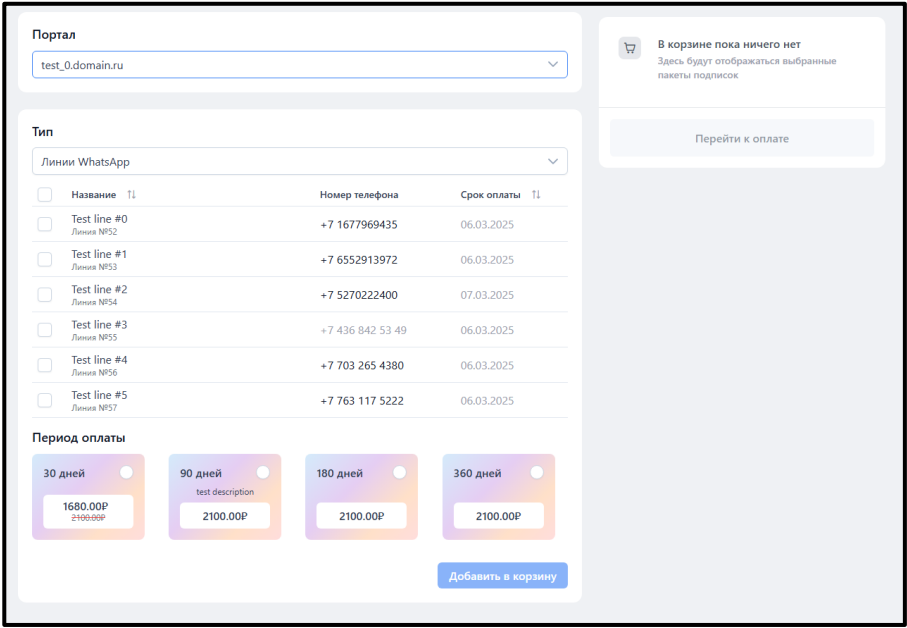


Рисунок 15 – Выбор состава позиций заказа

Как только отмечена хотя бы одна строка и выбран период, кнопка «Добавить в корзину» становится активной. По нажатию она анимировано перемещает выбранные позиции: внутри панели появляется список пакетов с краткой расшифровкой, итоговой суммой и крестиком для удаления.

После того как пользователь нажал «Добавить в корзину», правый сайдбар развернулся из «пустой» в наполненный режим: в шапке отобразился выбранный тип, а ниже добавленные позиции с их именами и ценой, как показано на рисунке 15.

Кнопка «Перейти к оплате» теперь залита ярким зелёным, что сигнализирует о готовности шага: она активируется, как только в корзине есть хотя бы одна позиция. Нажатие отправляет пользователя на следующую страницу, где бэкенд биллинга формирует счёт. Поле портала в основном блоке заблокировано: как только корзина содержит позиции, смена портала недоступна, чтобы не смешивать товары из разных инсталляций Bitrix24, для изменения нужно очистить корзину нажав на крестик.

Оформление подписки

Компания

Название: НАО "ИНФОКОМ"

Юридический адрес: г Москва, Рязанский пр-кт, д 8А стр 1, помещ III, ком 18

ОГРН: 1087746418033

ИНН / КПП: 7717615385 / 772101001

Контакты

Номер телефона: +7 (923) 423-44-35

Электронная почта: corp_infocom@gmail.com

Портал

test_0.domain.ru

Группы WhatsApp

100 2 100.00Р

Итого 2 100.00Р

[Перейти к оплате](#)

Рисунок 16 – Выбор позиций для корзины

После нажатия на кнопку «Перейти к оплате» происходит переход на страницу сформированного счёта с указанными позициями, как показано на рисунке 17. Правый информационный блок содержит реквизиты договорных сторон: сверху «Лицензиар» и блок «Лицензиат» с данными заказчика. Ниже маленький подзаголовок «Контакты для связи по оплате» выводит e-mail и телефон, чтобы клиенту было удобно уточнить детали. Иконка пера и ссылка «Изменить способ оплаты» дают возможность выбрать платёжный метод. После нажатия «Оплатить» пользователь попадает на стандартную защищённую форму Robokassa, приведенную на рисунке 18.

Оформление подписки

Счет № 173
14.05.2025

Ожидание оплаты

Наименование	Колич.	Цена, Р	Сумма, Р
Лицензия на ПО для ЭВМ 'OlChat' тариф WhatsApp Группы 100		2 100	2 100
Итого			2 100 Р

Лицензиар
ООО ИНФОКОМ
ИНН 7707083893
675000, Амурская область, г. Благовещенск, ул. Зейская 240

Лицензиат
НАО "ИНФОКОМ"
ИНН/КПП 7717615385/772101001
г Москва, Рязанский пр-кт, д 8А стр 1, помещ III, ком 18

Контакты для связи по оплате
Email: corp_infocom@gmail.com
Телефон: +7 (923) 423-44-35

Оплата картой [Изменить способ оплаты](#)

[Оплатить](#)

Рисунок 17 – Страница сформированного счета, заказа

Следующей строкой идёт пояснение «Оплата услуг ООО INFOCOM», чтобы покупатель понимал назначение платежа и мог сопоставить его со счётом на предыдущем шаге. Ниже начинается блок ввода данных карты. Чуть ниже располагается крупная тёмно-синяя кнопка «Оплатить», активная после корректного ввода всех реквизитов. Когда пользователь выбирает безналичный расчёт (банковский перевод) на рисунке 19, страница счёта перестраивается, слева по-прежнему крупный заголовок «Счёт №» с датой, но под номером появляется светло-жёлтый статус «Ожидание ручного подтверждения оплаты» он сразу объясняет, что оплата поступит не автоматически, а после ручной проверки. После формирования счёта пользователю доступно действие «Поделиться», продемонстрированное на рисунке 19. При нажатии открывается окно, сверху короткая ссылка на страницу счёта, её можно скопировать одним кликом, ниже QR-код с этой же ссылкой для сканирования с телефона.

The screenshot displays the Robokassa payment interface. At the top, the Robokassa logo and the text "Безопасная оплата" are visible. Below this, the merchant name "ИНФОКОМ" is shown, followed by the amount "2100 Р" and a small upward arrow. A button labeled "Отправим чек на почту" is present, with the email address "corp_infocom@gmail.com" entered below it. The payment description "Оплата услуг ООО INFOCOM" is also visible. The "Картой" (By Card) section is active, showing a table for card details: "Номер карты" (Card Number) with the value "*1111", "Срок" (Expiry) with "12/25", and "Код" (Code) with "...". To the right of the table is the "VISA" logo and a close button. Below the table, the cardholder's name "Владелец карты VASILY KRYLOV" is displayed. At the bottom of the form, there are two large buttons: a dark blue one labeled "Оплатить" and a purple one labeled "Оплатить" with the "сбп" (Russian payment system) logo.

Рисунок 18 – Форма оплаты Robokassa

Кнопка «Скачать счёт» формирует PDF, проиллюстрированный на рисунке 19 и отдаёт файл, показанный на рисунке 21, содержит блок платёжных реквизитов, заголовок «Счёт-оферта №...» с временной меткой, сведения о лицензиаре и лицензиате, таблицу позиций, итог к оплате прописью, а также текст с условиями лицензии. Клиент оплачивает счёт в своём банке, ООО «ИнфоКом» отмечает поступление. Счёт

меняет статус на «Оплачено» и активируется через событие в Bitrix24. Каждый заказ, созданный в биллинг-системе, независимо от способа оплаты, при создании отображается на портале Bitrix24 в виде отдельного элемента смарт-процесса специального настраиваемого бизнес-процесса, позволяющего отслеживать статус и этапы обработки заказа непосредственно в CRM Bitrix24, как показано на рисунке 20. Визуальное представление отражает заказы и их перемещения между стадиями по мере изменения его статуса. Далее, в зависимости от действий пользователя и состояния обработки, заказы последовательно переходят в стадии «Ожидание оплаты», «Счёт выставлен», «Счёт оплачен» и «Лицензия активирована».

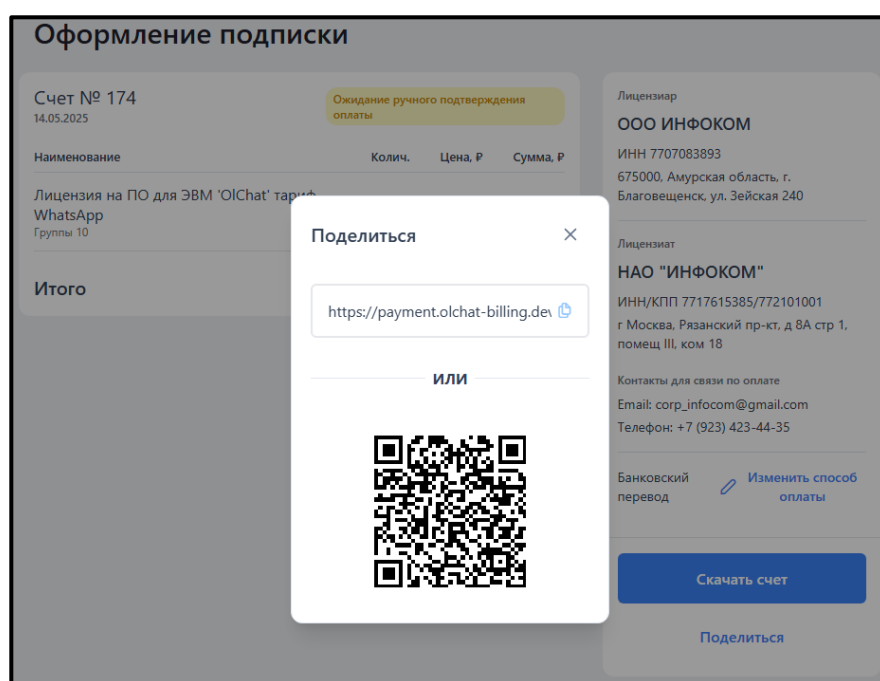


Рисунок 19 – Поделиться ссылкой

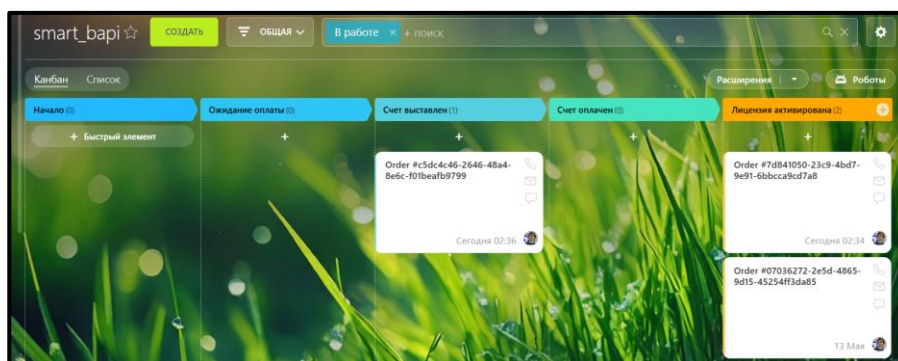


Рисунок 20 – Смарт-процесс

Внимание! Оплата данного счета означает согласие с условиями оферты.

ДАЛЬНЕВОСТОЧНЫЙ БАНК ПАО СБЕРБАНК				БИК	040813608
Банк получателя				Сч. №	30101810400000000225
ИНН	7707083893	КПП	772101001	Сч. №	None
ООО ИНФОКОМ					
Получатель					
Назначение платежа					
Оплата по счету-оферте № 174 от 2025-05-14 17:36:55.141684 за неисключительные права на использование ПО для ЭВМ, без НДС					

Счёт-оферта № 174 от 2025-05-14 17:36:55.141684

Лицензиар: ООО ИНФОКОМ, ИНН 7707083893, , тел.: 88005555777

Лицензиат: НАО "ИНФОКОМ", ИНН 7717615385, 772101001 КПП 772101001, , тел.: +7 (923) 423-44-35

Адрес портала Битрикс24: test_0.domain.ru

№	Наименование программы для ЭВМ	Кол-во	Вознаграждение за использование одного экземпляра, руб	Вознаграждение (общая сумма), руб
1	Лицензия на ПО для ЭВМ 'OlChat' тариф WhatsApp		2100	2100

Итого к оплате (без НДС): 2100.0 руб.

Всего к оплате: Две тысячи сто рублей 00 копеек

НДС не облагается в соответствии с подпунктом 26 пункта 2 статьи 149 Налогового кодекса

Индивидуальный предприниматель

Иванова Ольга Сергеевна

Настоящий Счет-оферта (далее - «Счет») является письменным предложением (офертой) Лицензиара заключить Лицензионный договор, которое направляется Лицензиату в соответствии со ст 432-444 Гражданского Кодекса РФ (далее - «ГК РФ»).

Лицензионный договор (неисключительная лицензия) заключается путем принятия (акцепта) оферты Лицензиатом в установленном порядке (п.3 ст.438 ГК РФ), что считается соблюдением письменной формы договора (п. 3 ст.434 ГК РФ).

Под указанными в оферте следующими терминами понимаются их нижеуказанные значения:

Условия оферты:

Правообладатель - лицо, обладающее исключительным правом на ПО или лица, правомерно обладающие на соответствующей территории всем тем необходимым объемом прав на использование ПО, включая право на предоставление разрешений на использование ПО (сублицензирование).

Лицензионное соглашение - соглашение между Правообладателем и Лицензиатом, которое предусматривает полномочия и ограничения использования Лицензиатом ПО, и условия которого безоговорочно принимаются Лицензиатом во время инсталляции ПО.

Условия оферты:

1 Предметом Лицензионного договора является предоставление Лицензиаром прав на использование указанного в Счете программного обеспечения для ЭВМ (далее - ПО).

2 Вознаграждением Лицензиара по Лицензионному договору является сумма, указанная в Счете.

3 Существенным условием заключения лицензионного договора является полная единовременная оплата Лицензиатом настоящего Счета, которая будет считаться единственно возможным надлежащим аспектом данной оферты (п.3 ст 433 ГК РФ). Оплата Счета без указания номера Счета в платежном поручении или третьим лицом не допускается. В случае оплаты Счета без указания номера Счета, Лицензиар имеет право зачесть его по своему усмотрению в счет оплаты по любому Счету Лицензиата

4 Лицензиар гарантирует, что получил от Правообладателя и правомерно обладает всем необходимым объемом прав, который предоставляется Лицензиату по лицензионному договору.

5 Счет действителен в течение 15 (пятнадцати) календарных дней с указанной в нем даты выставления (срок для акцепта оферты).

6 В случае если Счет не будет оплачен указанный период Лицензиар имеет право приостановить доступ Лицензиара к до момента оплаты. При этом неисключительные права на использование ПО не пролонгируются.

7 Лицензиат обязуется не нарушать авторские права Правообладателя

8 Права по настоящему лицензионному договору предоставляются с момента его заключения Лицензиар составляет и направляет Лицензиату соответствующий односторонний Акт с указанием срока, на который предоставляются права на использование ПО. В случае,

если письменные мотивированные возражения со стороны календарных дней с момента акцепта данной оферты. Акт считается утвержденным Лицензиатом.

9 Принимая - оферту, Лицензиат подтверждает, что надлежащим образом ознакомлен и согласен с положениями, требованиями и ограничениями пользовательских Лицензионных соглашений, технической и иной документацией по всему соответствующему по, указанному в настоящем Счете на весь срок действия исключительных прав на него (ст 1231 ГК РС) и на условиях изложенных в пользовательском Лицензионном

10 Обязанности Лицензиара по настоящему Счету считаются выполненными надлежащим образом с момента предоставления Лицензиату Акта с указанием срока, на который предоставляются права на использование ПО

11 Любые споры, которые могут возникнуть между Лицензиаром и Лицензиатом в связи с настоящей офертой или лицензионным договором, подлежат рассмотрению по месту нахождения Лицензиара

12 Возврат прав после получения невозможен в связи с тем, что покупатель приобретает право на использование которое относится к объектам авторских прав и скрывается, как литературное произведение (ГК РФ, -4, ст 1259). В свою очередь, непериодическое издания в соответствии с Законом о защите прав потребителей Российской Федерации (гл. 2 ст, 25) относятся к перечню непродовольственных товаров, не подлежащих возврату или обмену (в ред Постановлений Правительства РФ от 20.10 1993 №1222, 06.02.2002

Рисунок 21 – Сформированный счёт

В результате анализа пользовательских сценариев, применяемых сервисом Stripe, удалось выстроить интерфейс оформления заказа, понятный для пользователя. Каждый этап взаимодействия от выбора компании и заполнения реквизитов до формирования счета и подтверждения оплаты был создан с учетом пользовательских потребностей.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

Проектирование и внедрение биллинг-системы для управления подписками и оплатами в ООО «ИнфоКом» предполагает не только техническую реализацию, но и обязательный учёт требований охраны труда, экологической безопасности и мер реагирования на чрезвычайные ситуации.

Деятельность, связанная с разработкой программного обеспечения, осуществляется в офисных условиях и включает продолжительное взаимодействие сотрудников с персональными компьютерами и иными техническими средствами. В связи с этим особую актуальность приобретает организация безопасного рабочего места, соблюдение санитарно-гигиенических норм, рациональное использование ресурсов и соблюдение требований законодательства в области обращения с отходами.

4.1 Безопасность

В процессе эксплуатации биллинг-системы важную роль играет организация безопасных и комфортных условий труда для сотрудников, сопровождающих её ежедневное функционирование. Такие специалисты работают за компьютером они контролируют прохождение платежей, обрабатывают заказы в CRM Bitrix24, а также подтверждают поступление оплат от клиентов в случае ручного выставления счета за услуги, передвигая заказы по стадиям в рамках смарт-процесса Bitrix24.

4.1.1 Требования к техническому оснащению рабочих мест

Длительное пребывание за экраном, высокая концентрация внимания и однотипные действия могут вызывать зрительное и мышечное напряжение, а также способствовать развитию хронической усталости.

К ключевым факторам, негативно влияющим на здоровье сотрудников, относятся:

- зрительное напряжение при длительной работе с мониторами;
- недостаточная эргономика рабочего места и офисной мебели;
- параметры микроклимата в помещении (температура, влажность, вентиляция);

- электромагнитные и шумовые воздействия от техники;
- психоэмоциональная нагрузка из-за высокой ответственности и информационной нагрузки.

Минимизация воздействия перечисленных факторов требует соблюдения санитарно-гигиенических требований к оборудованию рабочих мест, грамотной эргономики и правильного подбора офисной техники. Важной составляющей также является разработка удобного, интуитивного интерфейса биллинг-системы, снижающего нагрузку на пользователя и исключая визуальное перенапряжение.

Одним из ключевых аспектов обеспечения безопасных условий труда является соответствие персональных компьютеров установленным нормам, регламентирующим уровни шума, вибрации и электромагнитного излучения на рабочих местах. Эти показатели должны оцениваться комплексно, с учетом совместного воздействия всех находящихся поблизости устройств. [9]

В таблице 7 представлены нормативы звукового давления и максимально допустимые уровни звука для рабочих мест в соответствии с СанПиН 2.1.3685-21.

Таблица 7 – Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Максимальные уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
79 дБ	63 дБ	52 дБ	45 дБ	39 дБ	35 дБ	32 дБ	30 дБ	28 дБ	55

Также необходимо соблюдать предельно допустимые уровни электромагнитных полей на рабочих местах. В зависимости от диапазона частот, допустимые значения напряженности электрического поля составляют от 3 до 25 В/м, а плотность потока энергии – до 10 мкВт/см². Такие ограничения особенно важны при размещении оборудования вблизи рабочих мест и позволяют минимизировать потенциальное воздействие электромагнитного излучения на организм человека.

Технические особенности и эргономические параметры персональных компьютеров, используемых в организации, должны обеспечивать комфортный и безопасный процесс работы:

- уровень шума от устройств не должен превышать 55 дБА;
- корпуса и клавиатуры должны иметь матовое покрытие в спокойной цветовой гамме с коэффициентом отражения света в диапазоне (0,4 – 0,6) для предотвращения бликов;
- экраны мониторов должны обеспечивать возможность регулировки контрастности и яркости в соответствии с индивидуальными потребностями пользователей.

Внимание при выборе и эксплуатации мониторов уделяется соблюдению ГОСТ Р 50948-2001 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности», который устанавливает обязательные параметры визуального комфорта и безопасности. В соответствии с ним к экранам мониторов предъявляются следующие требования:

- минимальная яркость белого изображения на экране не менее 35 кд/м²;
- неравномерность яркости экрана не должна превышать 20 %;
- контрастность (для монохромного изображения) не менее 3:1;
- отсутствие заметной нестабильности изображения (как по яркости, так и по положению отдельных элементов).

Важным фактором, способствующим снижению общего уровня шума, вибрации и электромагнитного излучения на рабочих местах, является архитектура технической инфраструктуры организации. ООО «ИнфоКом» реализует подход, основанный на использовании облачных технологий, вычислительные мощности арендуются у провайдера ООО «Яндекс», а основные серверные ресурсы размещаются в удалённом центре обработки данных. Такой подход исключает необходимость размещения в офисе шумного, энергоёмкого и габаритного серверного оборудования. В помещениях остаются только пользовательские персональные компьютеры и минимальный

набор периферийных устройств, благодаря чему создаются более благоприятные условия труда, соответствующие санитарным нормам и требованиям охраны труда.

4.1.2 Нормативные требования к помещениям для размещения рабочих мест с персональными компьютерами

Помимо требований к персональным компьютерам, важную роль в обеспечении безопасности и комфортных условий труда играет организация рабочих помещений, в которых размещены ПК. Правильно оборудованное офисное пространство позволяет минимизировать негативное влияние физических факторов среды, таких как освещение, электромагнитное излучение и шум, что способствует сохранению работоспособности и здоровья сотрудников.

Согласно действующим санитарным нормам, при использовании персональных электронно-вычислительных машин с плоскими экранами минимальная площадь на одного работника должна составлять не менее 4,5 м². Рекомендуемая площадь 10 м² на человека, что обеспечивает более комфортные условия труда и минимизирует визуальную и акустическую перегрузку.

Внутренняя отделка помещений, предназначенных для работы, должна выполняться с использованием диффузно отражающих материалов, коэффициенты отражения, которых должны соответствовать следующим значениям: потолок – (0,7 – 0,8), стены – (0,5 – 0,6), пол – (0,3 – 0,5). Соблюдение этих требований необходимо для предотвращения бликов и создания благоприятных условий освещённости.

Помещение, где производится работа с персональными компьютерами, должно соответствовать требованиям электробезопасности. Согласно ГОСТ Р 50571.5.54-2013, каждое рабочее место оборудуется системой защитного заземления все персональные компьютеры и периферийные устройства подключаются только через розетки с заземляющим контактом, объединенным с контуром заземления здания. Электропроводка выполняется трехпроводной и проложена в соответствии с правилами (скрыто в стенах или в кабельных каналах). Для защиты персонала и оборудования линии электроснабжения рабочих мест оснащаются устройствами защитного отключения (УЗО) с током утечки 30 мА, обеспечивающими мгновенное отключение при

неисправностях. Кроме того, на каждую группу розеток устанавливаются автоматические выключатели, предотвращающие перегрузки и короткие замыкания.

ООО «ИнфоКом» арендует офис в бизнес центре, как показано на рисунке 22.

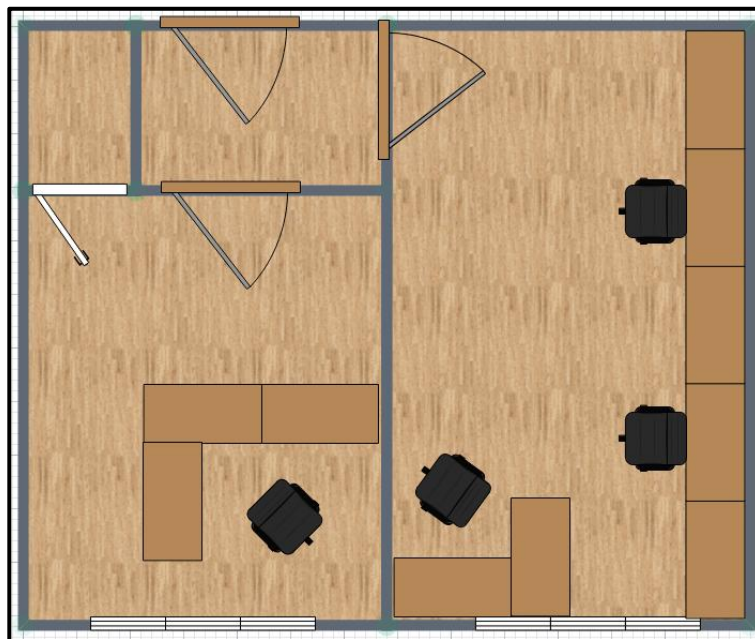


Рисунок 22 – Планировка офиса ООО «ИнфоКом»

ООО «ИнфоКом» рассчитан на четыре рабочих места и имеет общую площадь 30 м², что с избытком покрывает нормативные требования к размещению персонала. Минимально необходимая площадь при наличии плоских экранов составляет 18 м², таким образом фактические условия обеспечивают дополнительный резерв пространства для организации комфортных условий труда.

Отделка помещения выполнена в светлых тонах: потолок и стены окрашены в белый цвет, что способствует равномерному отражению света и предотвращает появление бликов на экранах мониторов. Распределительные электрощиты здания размещены за пределами офисного помещения, на лестничной клетке, что исключает влияние электромагнитных помех и соответствует установленным требованиям.

4.1.3 Нормы освещённости при работе с компьютером

Эффективное освещение является важным фактором обеспечения безопасных и комфортных условий труда при работе с персональными компьютерами. Согласно

требованиям СП 52.13330.2016, рабочие места должны быть оснащены как естественным, так и искусственным освещением.

Для естественного освещения предпочтительна ориентация окон на север или северо-восток, такой подход снижает уровень солнечной нагрузки и вероятность появления бликов. Важно предусмотреть возможность регулирования светового потока с помощью штор, жалюзи или козырьков. Мониторы следует располагать боком к окнам, чтобы свет падал слева, не попадая напрямую на экран.

Искусственное освещение должно быть равномерным и дополнять дневной свет. Рекомендуемая освещённость рабочего стола от 300 до 500 лк, экрана не более 300 лк. Яркость в поле зрения не должна превышать 200 кд/м², а допустимый уровень бликов на экране 40 кд/м². Светильники размещают сбоку или над рабочей зоной, избегая прямого попадания света в глаза. Коэффициент пульсации до 5 %, коэффициент запаса освещённости 1,4.

Основу искусственного освещения составляют люминесцентные лампы, допускается использование металлогалогенных и ламп накаливания для локального света. Светильники должны быть оборудованы отражателями и защитными углами не менее 40°. Поддержание светового комфорта требует регулярной чистки светильников и окон не реже двух раз в год, а также своевременной замены ламп.

В офисных помещениях ООО «ИнфоКом» реализовано комбинированное освещение. Окна выходят на запад, установлены жалюзи, что позволяет регулировать поток света в течение дня. Уровень освещённости соответствует действующим санитарным нормам.

4.1.4 Требования к микроклимату и воздуху на рабочих местах

В помещениях, где работа с персональными компьютерами является основной необходимо обеспечивать параметры микроклимата, в соответствии с категориями работ Ia и Ib, установленные СанПиН 2.1.3685-21 в пункте «Допустимые величины параметров микроклимата на рабочих местах в помещениях». В остальных случаях допустимо поддержание микроклимата на уровнях, предусмотренных указанным

стандартом. Оптимальные значения температуры воздуха, температуры поверхностей, относительной влажности и скорости движения воздуха зависят от времени года и категории работ и приведены в таблице 8.

Таблица 8 – Допустимые величины параметров микроклимата на рабочих местах в помещениях из СанПиН 2.1.3685-21

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
					Ниже оптимальных величин, не более	Выше оптимальных величин, не более
Холодный	Ia (до 139)	22 – 24	21 – 25	40 – 60	0,1	0,1
	Iб (140 – 174)	21 – 23	20 – 24	40 – 60	0,1	0,2
Тёплый	Ia (до 139)	23 – 25	22 – 26	40 – 60	0,1	0,2
	Iб (140 – 174)	22 – 24	21 – 25	40 – 60	0,1	0,3

Кроме параметров микроклимата, важным показателем является содержание аэроионов. Их уровни в помещениях должны соответствовать требованиям СанПиН 2.1.3685-21. Также необходимо контролировать содержание вредных химических веществ в воздухе помещений. Концентрации загрязняющих веществ не должны превышать предельно допустимых значений, установленных для атмосферного воздуха в соответствии с СанПиН 2.1.3685-21.

В офисных помещениях ООО «ИнфоКом» соблюдение этих норм обеспечивается за счёт централизованных инженерных систем бизнес-центра, в котором арендуется офис. Администрация бизнес-центра проводит проверку микроклимата, уровня ионизации и химического состава воздуха, что гарантирует соответствие условий труда санитарно-гигиеническим требованиям.

4.1.5 Эргономические и технические требования к организации рабочих мест с персональными компьютерами

Организация рабочего места должна учитывать эргономические, санитарные и технические нормы, направленные на сохранение здоровья сотрудников при длитель-

ной работе за персональными компьютерами. На рисунке 23 приведена схема правильного расположения пользователя за компьютером, отражающая ключевые параметры и рекомендации по оборудованию.

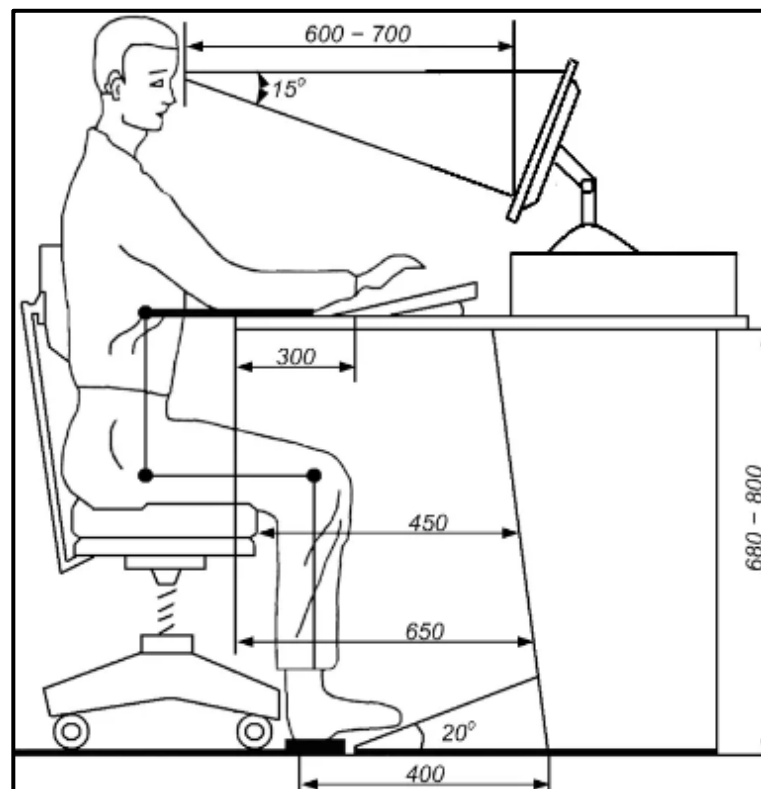


Рисунок 23 – Рекомендуемая эргономика при работе с персональным компьютером

Монитор персонального компьютера должен быть размещён на расстоянии от глаз пользователя в пределах (600 – 700) мм. Центр экрана должен находиться немного ниже уровня глаз, с углом наклона взгляда около 15°, что снижает зрительное напряжение. Рабочая поверхность стола должна иметь регулируемую по высоте столешницу в диапазоне (680 – 800) мм, либо фиксированную высоту 725 мм, обеспечивающую комфортную позу при сидении.

Пространство для ног под столом должно быть не менее 600 мм по высоте, 500 мм по ширине и 450 мм по глубине у колен. При вытянутых ногах глубина увеличивается до 650 мм. Дополнительно рекомендуется использование подставки для ног с глубиной не менее 400 мм, шириной не менее 300 мм и углом наклона до 20°, как показано на рисунке. Это способствует снижению статической нагрузки на ноги.

Рабочее кресло должно иметь, нескользящее и воздухопроницаемое покрытие, быть поворотным и обладать возможностью регулировки по высоте и углу наклона сиденья и спинки. Высота сиденья регулируется в пределах от 400 до 550 мм, глубина посадки около 450 мм, что позволяет обеспечить физиологически правильную позу. Спинка должна поддерживать позвоночник по всей длине, с высотой около 300 мм и радиусом кривизны около 400 мм. Угол наклона спинки допускается в пределах $\pm 30^\circ$. Расположение клавиатуры должно быть на расстоянии 300 мм от края стола, чтобы обеспечить правильное положение предплечий.

4.1.6 Нормирование работы за персональным компьютером

Работа сотрудника компании ООО «ИнфоКом» включает значительное время за персональным компьютером. В частности, ежедневно выполняются задачи, связанные с разработкой программного обеспечения, взаимодействием с CRM-системой Bitrix24, оформлением технической документации, обработкой заявок, ведением переписки с клиентами и командой. Деятельность сопровождается высокой концентрацией внимания, продолжительной фиксацией взгляда на экране, однообразными движениями рук и минимальной подвижностью, что увеличивает нагрузку на зрительный аппарат, опорно-двигательный аппарат и нервную систему.

Для минимизации негативного воздействия таких условий труда применяется система нормирования, основанная на методиках оценки тяжести и напряжённости труда. В зависимости от характера выполняемых операций работа с персональным компьютером подразделяется на три группы:

- группа А – считывание информации с экрана;
- группа Б – ввод данных;
- группа В – творческая работа в диалоговом режиме.

В условиях ООО «ИнфоКом» деятельность сотрудников относится преимущественно к группе В, так как включает интеллектуальный труд, связанный с анализом, кодированием и принятием решений в диалоговом режиме. Согласно нормативам, для группы В продолжительность непрерывной работы не должна превышать 6 часов

в течение смены. При этом суммарное время регламентированных перерывов зависит от продолжительности смены и уровня нагрузки, как показано в таблице 9.

Таблица 9 – Суммарное время регламентированных перерывов в зависимости от вида и объёма трудовой деятельности

Категория работы с ПК	Уровень нагрузки за рабочую смену при видах работ с ПК			Суммарное время регламентированных перерывов, мин	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20.000	до 15.000	до 2	50	80
II	до 40.000	до 30.000	до 4	70	110
III	до 60.000	до 40.000	до 6	90	140

Рекомендуется чередовать работу с другими видами деятельности. При невозможности смены задач необходимо делать 15-минутные перерывы каждые (45 – 60) минут. В ночные смены продолжительность перерывов увеличивается на 30 %.

4.1.7 Требования к интерфейсу программного обеспечения с учётом требований удобства, доступности

Графический интерфейс разрабатываемой биллинг-системы должен обеспечивать интуитивное и удобное взаимодействие пользователя с программным обеспечением, минимизируя количество действий, необходимых для оформления заказа и оплаты услуг. Основное внимание уделяется логике построения интерфейса, визуальной и функциональной доступности, соответствию требованиям и принципам, изложенным в ГОСТ Р 52872-2019.

Интерфейс пользователя реализуется по принципу пошаговой навигации: от выбора компании и портала до формирования корзины, генерации счёта и выполнения оплаты. Все элементы взаимодействия сгруппированы в логические блоки, что обеспечивает визуальную и функциональную связность данных. При выборе услуг пользователь последовательно указывает организацию, заполняет юридические реквизиты, выбирает портал, а затем отмечает необходимые подключения и срок дей-

ствия услуги. Выбранные параметры автоматически отображаются в корзине с итоговой стоимостью. Состояние оформления заказа отслеживается на всех этапах, что снижает вероятность ошибок и повышает прозрачность процесса.

Особое внимание уделяется визуальной структуре интерфейса. Используется лаконичный, современный дизайн, обеспечивающий чёткое разграничение смысловых блоков, применение читаемых шрифтов и элементов с высоким уровнем контрастности. Интерфейс адаптивен и корректно отображается на различных устройствах: от стационарных компьютеров до мобильных телефонов. Макеты формы, таблиц, выпадающих списков и кнопок масштабируются и выравниваются автоматически, обеспечивая единообразие восприятия информации.

Система предоставляет пользователю обратную связь на каждом этапе взаимодействия. Например, при выборе тарифов система отображает стоимость и срок действия, при переходе к оплате – автоматически формируется счёт, который включает информацию о наименовании лицензии, юридических реквизитах сторон и методе оплаты. Для повышения удобства предоставляется возможность поделиться счётом по ссылке или в виде QR-кода. Также реализована функция загрузки счёта в формате PDF. После оформления счёта пользователь может выбрать способ оплаты: по карте, через платёжную систему Robokassa или банковским переводом.

Интерфейс биллинг-системы разработан с учётом принципов цифровой доступности. В соответствии с ГОСТ Р 52872-2019 реализована поддержка масштабирования текстов, контраст цветовых схем, возможность навигации с клавиатуры.

Безопасность пользовательского взаимодействия также является важным требованием. Критические действия, такие как подтверждение оплаты или отправка счёта, требуют явного подтверждения пользователя. Все передаваемые данные шифруются, а чувствительная информация, включая платёжные данные, не сохраняется на клиентской стороне. Благодаря архитектурному разделению ролей и контуров доступа, система гарантирует надёжную защиту данных как клиентов, так и операторов.

Таким образом, графический интерфейс биллинг-системы спроектирован с целью обеспечить максимальную простоту, доступность и безопасность использования,

учитывая, как потребности конечных пользователей, так и технические ограничения, обусловленные интеграцией с внешними системами Bitrix24 и Robokassa. Такое сократит количество ошибок при оформлении заказов.

4.2 Экологичность

Эксплуатация современных программных решений, включая мультитчат и разрабатываемую биллинг-систему для ООО «ИнфоКом», требует внимания к вопросам устойчивого развития и минимизации негативного воздействия на окружающую среду. Несмотря на нематериальную природу программного обеспечения, его жизненный цикл связан с потреблением вычислительных и офисных ресурсов, а также с образованием отходов оборудования, расходных материалов и упаковки.

Согласно статье 14 федерального закона № 89-ФЗ «Об отходах производства и потребления», организации обязаны классифицировать отходы по степени опасности от I до V класса и обеспечивать их надлежащее обращение. ООО «ИнфоКом» соблюдает эти нормы: изношенное компьютерное и периферийное оборудование передаётся специализированным организациям, имеющим лицензии на работу с отходами I–IV классов. Офисная мебель при списании оценивается на предмет повторного использования, а непригодная для эксплуатации передаётся как отходы V класса. Устаревшие источники освещения (ртутные лампы) поэтапно заменяются на светодиодные, снижая риски загрязнения окружающей среды.

Важной составляющей экологического подхода стало внедрение собственной биллинг-системы, позволяющей отказаться от необходимости печатать и формировать счета и иные документы, связанные с оплатой. Формирование и отправка счётов осуществляется в цифровом виде через пользовательский интерфейс системы. Клиенты получают доступ к информации об оплате, счёту, QR-коду и ссылке на оплату через веб-интерфейс, что полностью исключает необходимость курьерской доставки документов, их распечатки или ручного согласования на бумажных носителях.

Кроме того, архитектура биллинг-системы спроектирована с применением микросервисного подхода, что позволяет гибко масштабировать вычислительные ресурсы в зависимости от нагрузки. Это решение снижает общее энергопотребление

ИТ-инфраструктуры, поскольку каждый компонент системы работает автономно и запускается только при необходимости. Такой подход снижает углеродный след, характерный для длительной и избыточной работы серверов.

Таким образом, экологическая устойчивость разрабатываемой биллинг-системы достигается за счёт рационального проектирования её архитектуры, снижения потребления вычислительных ресурсов и включения в существующую цифровую экосистему компании. Использование автоматизированных процессов расчётов и цифрового оформления заказов способствует сокращению бумажного документооборота, дополняя уже внедрённые в ООО «ИнфоКом» системы электронного документооборота и CRM Bitrix24. В совокупности с экологически ответственным подходом к утилизации техники, офисного оборудования и сортировке отходов, такая цифровизация формирует модель эксплуатации программных решений.

4.3 Чрезвычайные ситуации

При эксплуатации программных решений, включая разработанную биллинг-систему, необходимо учитывать риски, связанные с возможными чрезвычайными ситуациями. Наиболее вероятной угрозой в офисной среде, где осуществляется разработка, тестирование и сопровождение информационных систем, остаётся пожар. Его причиной могут стать короткое замыкание, перегрузка электросети, перегрев оборудования или человеческий фактор.

ООО «ИнфоКом» располагается в арендованных помещениях бизнес-центра, где предусмотрены базовые меры противопожарной защиты. Помещения оснащены системой автоматического оповещения, пожарной сигнализацией и средствами первичного пожаротушения. Эвакуационные выходы чётко обозначены, обеспечен свободный доступ к ним, а схемы эвакуации размещены на видных местах в соответствии с установленными нормами. Ответственность за техническое обслуживание противопожарной инфраструктуры возложена на администрацию здания и регламентируется условиями договора аренды.

Как ИТ-организация, ООО «ИнфоКом» критически зависит от бесперебойной работы серверов, сетевого и компьютерного оборудования. Потенциальные сбои

электропитания могут повлечь за собой остановку функционирования информационных систем, включая разработанную биллинг-систему. В связи с этим особое внимание уделяется обеспечению аварийной устойчивости: используются источники бесперебойного питания (ИБП), а также организовано резервное копирование данных.

Комплекс предпринимаемых мер по обеспечению пожарной безопасности соответствует положениям Федерального закона № 123-ФЗ «Технический регламент о требованиях пожарной безопасности», а также постановлению Правительства РФ от 16.09.2020 № 1479 «Об утверждении Правил противопожарного режима в Российской Федерации».

Мероприятия по обеспечению пожарной безопасности включают:

- оснащение арендуемых помещений автоматической системой пожарной сигнализации, датчиками дыма и температуры, а также системой оповещения;
- размещение планов эвакуации на видных местах;
- использование в отделке помещений материалов с пониженной горючестью;
- наличие и обслуживание средств первичного пожаротушения;
- инструкции о мерах пожарной безопасности и назначение ответственных лиц;
- организация тренировок по эвакуации персонала;
- проверка исправности технических средств противопожарной защиты.

В дополнение к техническим угрозам физического характера, в деятельности компании присутствуют и информационные риски, возникающие в процессе функционирования автоматизированных бизнес-процессов, интегрированных в разработанную биллинг-систему. Один из таких рисков связан с расхождениями между статусом оплаты и фактической активацией лицензии. Возможна ситуация, при которой клиент успешно завершает платёж через систему Robokassa, однако активация лицензии не происходит из-за сбоя на стороне внешнего сервиса или недоступности API.

Для обеспечения обработки заказов в рамках разработанной биллинг-системы реализована интеграция с CRM Bitrix24. Ключевым элементом данной интеграции явля-

ется смарт-процесс в Bitrix24, предназначенный для отслеживания и управления статусами заказов. Этот смарт-процесс не создаётся вручную, он автоматически развивается на портале компании в момент установки биллинг-системы. Таким образом, при внедрении решения в инфраструктуру заказчика, структура управления заказами через CRM создаётся по шаблону, предусмотренному в логике биллинга, без необходимости дополнительных действий со стороны администратора Bitrix24. При оформлении нового заказа на подписку или активацию лицензии биллинг автоматически формирует элемент смарт-процесса в Bitrix24 с передачей всех необходимых параметров: номера заказа, сумма, статуса оплаты и т.д.

Далее заказ отслеживается в CRM Bitrix24 по определённым стадиям от выставления счёта до активации лицензии. Если клиент успешно производит оплату через платёжный шлюз Robokassa, но по какой-либо причине, сбой API, ошибка активации, недоступность внешнего сервиса, лицензия не активируется автоматически, элемент заказа остаётся на стадии «Счёт оплачен». Это служит индикатором технической неисправности.

В такой ситуации в процесс включается сотрудник компании ООО «ИнфоКом». Он вручную инициирует проверку, устраняет причину сбоя перемещая элемент заказа на следующую стадию «Лицензия активирована». Это действие в Bitrix24 сопровождается генерацией события об изменении стадии элемента. Биллинг-система отслеживает такие события через подписку на вебхуки и, при получении уведомления, автоматически обновляет внутренние данные по заказу.

Таким образом, помимо классических мер противопожарной безопасности, в ООО «ИнфоКом» реализован подход к управлению ЧС и на уровне цифровых процессов. Это охватывает как физическую безопасность персонала и инфраструктуры, так и устойчивость автоматизированных сценариев работы. Комплексная подготовленность к нестандартным ситуациям позволяет поддерживать стабильную работу сервисов, минимизировать потери и обеспечить высокий уровень клиентского сервиса даже при возникновении технических сбоев.

ЗАКЛЮЧЕНИЕ

В рамках выполнения выпускной квалификационной работы было осуществлено проектирование и разработка биллинг-системы для ООО «ИнфоКом», нацеленной на управление заказами, выставление счетов и обработку платежей за цифровые услуги и подписки. Система спроектирована с учетом специфики деятельности компании, обеспечивая интеграцию с CRM Bitrix24 через API Bitrix24 и платежной системой Robokassa.

На подготовительном этапе был проведён анализ внешних API Bitrix24 и Robokassa, что позволило выявить оптимальные методы интеграции, а также определить возможные ограничения, связанные с аутентификацией, частотой запросов и обработкой ошибок. Эти данные легли в основу проектных решений и архитектуры программного обеспечения.

При проектировании архитектуры был применён микросервисный подход, позволивший разделить функциональные блоки на независимые модули, такие как сервис заказов, модуль интеграции с внешними API и система выставления счетов. Каждый компонент системы разработан с учётом его роли и взаимодействия в общей структуре. Модель взаимодействия представлена в виде UML-диаграмм.

В качестве серверной платформы был выбран язык программирования Python и веб-фреймворк FastAPI, благодаря которому обеспечивается асинхронная обработка запросов и высокая производительность. Для хранения данных использована реляционная СУБД PostgreSQL в сочетании с ORM-библиотекой SQLAlchemy и системой миграций Alembic. Библиотека `httpx` применялась для асинхронного взаимодействия с внешними API, а Redis для управления очередями запросов к Bitrix24.

Для обеспечения развертывания системы было принято решение использовать контейнерную платформу Docker для упаковки сервиса в изолированный контейнер с необходимыми зависимостями и обратный прокси-сервер Traefik, обеспечивающий маршрутизацию запросов.

В рамках проектирования модуля интеграции с REST API Bitrix24 был разработан модуль CRest, обеспечивающий взаимодействие с CRM-платформой. Этот модуль реализует механизмы формирования и отправки API-запросов, а также автоматизирует процессы аутентификации, обновления токенов и регистрации интеграционных привязок, таких как события, места встраивания и роботы. Анализ требований Bitrix24 к API-запросам позволил оптимизировать упаковку параметров и обработку ответов. Модуль интеграционных привязок, спроектированный для CRest, автоматизирует регистрацию событий, мест встраивания и роботов в Bitrix24, расширяя функциональность.

В ходе проектирования информационной части системы были определены основные сущности, а также их ключевые атрибуты и взаимосвязи. Информационная модель системы была сформирована на основе инфологического, логического и физического моделирования. Были выделены основные сущности, их атрибуты и связи, а также построена реляционная структура данных с применением нотации IDEF1X, что обеспечило структурную целостность и соответствие требованиям нормализации.

В результате выполнения дипломной работы была спроектирована и реализована биллинг-система, полностью адаптированная под задачи компании ООО «Инфо-Ком». Разработанное программное решение отличается модульной архитектурой, технологической готовностью к дальнейшему развитию и масштабированию в рамках цифровой инфраструктуры.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 Bitrix24 REST API [Электронный ресурс]. – Режим доступа URL: <https://apidocs.bitrix24.ru/> (дата обращения: 09.04.2025).

2 Docs Robokassa [Электронный ресурс]. – Режим доступа URL: <https://docs.robokassa.ru/> (дата обращения: 09.04.2025).

3 Обзор архитектур современных биллинговых систем и перспективы их развития [Электронный ресурс]. – Режим доступа URL: <https://cyberleninka.ru/article/n/obzor-arhitektur-sovremennyh-billingovyh-sistem-i-perspektivy-ih-razvitiya> (дата обращения: 09.04.2025).

4 Микросервисная и монолитная архитектуры [Электронный ресурс]. – Режим доступа URL: <https://www.ibm.com/think/topics/monolithic-vs-microservices> (дата обращения: 09.04.2025).

5 Как разбить монолит на микросервисы [Электронный ресурс]. – Режим доступа URL: <https://martinfowler.com/articles/break-monolith-into-microservices.html> (дата обращения: 09.04.2025).

6 Стружкин Н. П. Базы данных: проектирование: учебник для вузов / Н. П. Стружкин В. В. Годин. – Москва: Юрайт, 2023. 477 с.

7 Кара-Ушанов В. Ю. SQL - язык реляционных баз данных: учебное пособие / В. Ю. Кара-Ушанов. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. 156 с.

8 Нестеров С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. – Москва: Изд-во Юрайт, 2023. – 259

9 Резчиков Е. А. Безопасность жизнедеятельности: учебник для вузов. Изд-во Юрайт, 2025. С. 16–17.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Белов, С. В. Безопасность жизнедеятельности и защита окружающей среды (техносферная безопасность): учебник для среднего профессионального образования / С. В. Белов. – 6-е изд., перераб. и доп. – Москва: Изд-во Юрайт, 2025. – 638 с.
- 2 ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению. – Взамен ГОСТ 19.201-73; введ. 1980-01-01. – Государственный комитет СССР по стандартам; М.: Изд-во Стандартиформ, 2010. – 16 с.
- 3 Использование диаграммы вариантов использования UML при проектировании программного обеспечения [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/566218/>. – 09.04.2025.
- 4 Как разбить монолит на микросервисы [Электронный ресурс]. – Режим доступа: <https://martinfowler.com/articles/break-monolith-into-microservices.html>. – 16.04.2025.
- 5 Кара-Ушанов, В. Ю. SQL - язык реляционных баз данных: учебное пособие / В. Ю. Кара-Ушанов. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2016. – 156 с.
- 6 Микросервисная и монолитная архитектуры [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/think/topics/monolithic-vs-microservices>. – 09.04.2025.
- 7 Нестеров, С. А. Базы данных: учебник и практикум для вузов / С. А. Нестеров. – Москва: Изд-во Юрайт, 2023. – 259 с.
- 8 Обзор архитектур современных биллинговых систем и перспективы их развития [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/obzor-arhitektur-sovremennyh-billingovyh-sistem-i-perspektivy-ih-razvitiya>.
- 9 Резчиков, Е. А. Безопасность жизнедеятельности: учебник для вузов / Е. А. Резчиков, А. В. Рязанцева. – 4-е изд., перераб. и доп. – Москва: Изд-во Юрайт, 2025. – 638 с.

10 Стандарт организации: Оформление выпускных, квалификационных и курсовых работ (проектов). – Благовещенск: Амурский государственный университет, 2018. – 75 с.

11 Стружкин, Н. П. Базы данных: проектирование: учебник для вузов / Н. П. Стружкин, В. В. Годин. – Москва: Юрайт, 2023. – 477 с.

12 Куомо, С. Vue.js 3 для начинающих: освоение основ Vue.js 3 и его экосистемы для создания современных веб-приложений / С. Куомо. – Бирмингем: Packt Publishing, 2024. – 302 с.

13 Alembic's documentation [Электронный ресурс]. – Режим доступа: <https://alembic.sqlalchemy.org/en/latest/>. – 12.04.2025.

14 API Docs Bitrix Framework [Электронный ресурс]. – Режим доступа: https://dev.1c-bitrix.ru/api_help/. – 12.04.2025.

15 Bitrix24 [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru/>. – 12.04.2025.

16 Bitrix24 REST API [Электронный ресурс]. – Режим доступа: <https://apidocs.bitrix24.ru/>. – 12.04.2025.

17 Docs docker [Электронный ресурс]. – Режим доступа: <https://docs.docker.com/>. – 12.04.2025.

18 Docs Robokassa [Электронный ресурс]. – Режим доступа: <https://docs.robokassa.ru/>. – 12.04.2025.

19 FastAPI framework documentation [Электронный ресурс]. – Режим доступа: <https://fastapi.tiangolo.com/>. – 12.04.2025.

20 HTTPX documentation [Электронный ресурс]. – Режим доступа: <https://www.python-httpx.org/>. – 12.04.2025.

21 Introduction to microservices [Электронный ресурс]. – Режим доступа: <https://cloud.google.com/architecture/microservices-architecture-introduction>. – 09.04.2025.

22 PostgreSQL 16.3 documentation [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/docs/16/index.html>. – 12.04.2025.

23 PricePlan [Электронный ресурс]. – Режим доступа: <https://priceplan.ru/>. – 12.04.2025.

24 Pydantic documentation [Электронный ресурс]. – Режим доступа: <https://docs.pydantic.dev/>. – 12.04.2025.

25 Pytest documentation [Электронный ресурс]. – Режим доступа: <https://docs.pytest.org/>. – 12.04.2025.

26 Python 3.12.4 documentation [Электронный ресурс]. – Режим доступа: <https://docs.python.org/>. – 12.04.2025.

27 Redis [Электронный ресурс]. – Режим доступа: <https://redis.io/>. – 09.04.2025.

28 RESTful API Design Guide by Google [Электронный ресурс]. – Режим доступа: <https://cloud.google.com/apis/>. – 09.04.2025.

29 SQLAlchemy documentation [Электронный ресурс]. – Режим доступа: <https://www.sqlalchemy.org/>. – 12.04.2025.

30 Stripe [Электронный ресурс]. – Режим доступа: <https://docs.stripe.com/api/>. – 12.04.2025.

31 Traefik [Электронный ресурс]. – Режим доступа: <https://traefik.io/traefik/>. – 12.04.2025.

ПРИЛОЖЕНИЕ А Документооборот компании

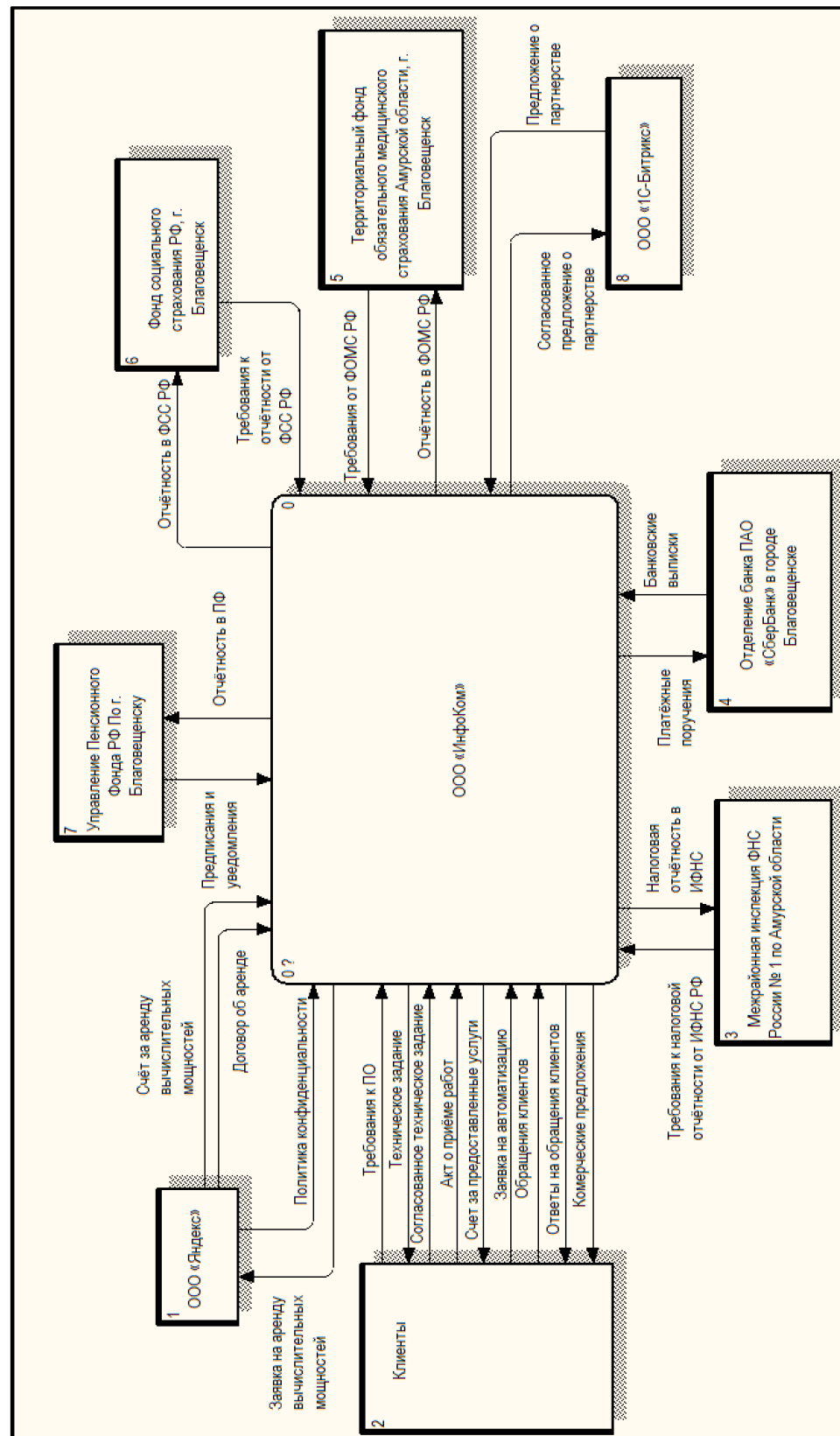


Рисунок А.1 – Диаграмма потока данных внешнего документооборота компании

Продолжение ПРИЛОЖЕНИЯ А

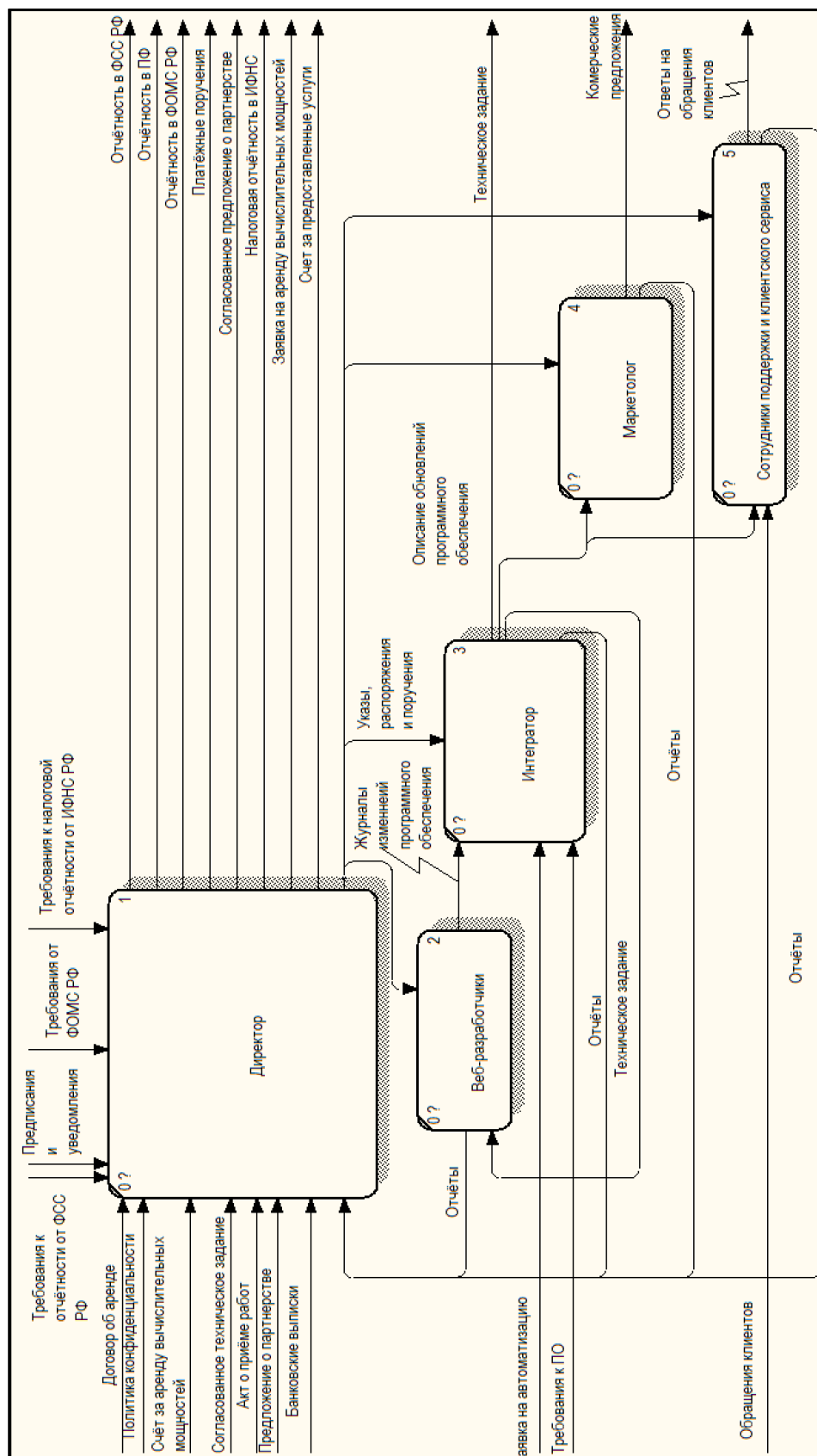


Рисунок А.2 – Диаграмма потока данных внутреннего документооборота компании

ПРИЛОЖЕНИЕ Б

Техническое задание

1 Введение

1.1 Наименованием программы

Наименование программы «Биллинг-система».

1.2 Краткая характеристика области применения программы

Программное обеспечение предназначено для автоматизации финансовых процессов в компании, предоставляющих цифровые услуги и товары, и интегрировано с приложениями компаний. Оно обеспечивает управление заказами, автоматизацию выставления счетов и контроль платежей в режиме реального времени. Сам биллинг интегрирован с CRM Bitrix24 через смарт-процесс, где отображаются стадии заказов, что позволяет отслеживать жизненный цикл заказов, синхронизировать данные о платежах и реквизитах, а также централизованно контролировать обновление статусов заказов после проведения транзакций.

Данная система будет использоваться ООО «ИНФОКОМ» для оптимизации внутренних бизнес-процессов, сокращения ручного выставления счета, повышения прозрачности финансовых операций, обеспечивая оперативное реагирование на изменения в состоянии платежей и улучшая контроль над процессом обработки заказов.

2 Основание для разработки

2.1 Основание для проведения разработки

Данный проект носит учебный и практический характер.

2.2 Наименование и условное обозначение темы разработки

Наименование темы разработки – «Биллинг-система».

Условное обозначение разработки – «BAPL».

3 Назначение разработки

3.1 Функциональное назначение программы

Продолжение ПРИЛОЖЕНИЯ Б

Функциональное назначение разрабатываемого программного обеспечения заключается в создании системы, обеспечивающей управление заказами, выставление счетов и обработку платежей за цифровые услуги и товары. Программа позволяет реализовать полный цикл обработки заказов, от их формирования заказа до подтверждения оплаты и активации услуги.

3.2 Эксплуатационное назначение программы

Эксплуатационное назначение разрабатываемой программы заключается в её использовании компанией ООО «ИНФОКОМ» для автоматизации процессов, связанных с управлением заказами, выставлением счетов и приёмом платежей за цифровые услуги. Программное обеспечение будет интегрироваться в существующие веб-приложения компании, расширяя их функциональность и обеспечивая пользователям удобный механизм оплаты и контроля заказов.

4 Требования к программе

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения следующих функций:

- создание и управление заказами;
- автоматическое выставление счетов на оплату с учётом выбранного способа оплаты;
- приём и обработка платежей с использованием платёжной системы Robokassa;
- хранение истории заказов;
- интеграция с CRM Bitrix24 через смарт-процесс для отображения стадий заказов и синхронизации статусов;
- уведомление пользователей о статусах заказа и оплате;
- подключения к приложению компании для интеграции механизма оплаты.

Продолжение ПРИЛОЖЕНИЯ Б

4.1.2 Требования к организации входных данных

Пользователь взаимодействует с системой через интерфейс оформления подписки, где ввод и выбор данных должны происходить по следующим требованиям:

- пользователь выбирает компанию из выпадающего списка. После выбора автоматически подставляются реквизиты организации ОГРН, ИНН, КПП, адрес, контакты;
- заполнение контактной информации: отображаются номер телефона и электронная почта, которые редактируемые;
- указание портала: выбор домена Bitrix24, например, check-check.bitrix24.ru через выпадающий список.
- выбор типа услуги (подписки), из выпадающего списка выбирается конкретный сервис, например, «Линии WhatsApp». После выбора система отображает доступные линии с указанием: название, привязанного номера телефона, даты окончания текущей оплаты;
- выбор периода оплаты, пользователю предоставляется возможность выбрать желаемый период. Каждый вариант имеет визуально выделенную кнопку и отображает стоимость, скидки и описания.
- Добавление услуги в корзину: после выбора всех параметров, пользователь нажимает кнопку «Добавить в корзину», что инициирует сбор введенных данных и передачу их в систему для расчёта итоговой суммы и формирования заказа.

Все входные данные валидируются в режиме реального времени: проверка заполнения обязательных полей, корректности формата (телефон, email), доступности выбранных линий.

4.1.3 Требования к организации выходных данных

Программа обеспечивает формирование и отображение выходных данных в процессе оформления и обработки заказа, в следующем виде:

Продолжение ПРИЛОЖЕНИЯ Б

- номер счёта и статус, система отображает уникальный номер сформированного счёта и текущий статус оплаты;
- дата формирования, отображается дата создания счёта;
- состав заказа, в табличной форме выводится информация по каждой позиции, наименование лицензии/услуги, например, «Лицензия на ПО для ЭВМ 'OlChat' тариф WhatsApp», привязанный аккаунт, цена за единицу;
- итоговая сумма, общая сумма заказа выводится крупно внизу таблицы;
- реквизиты сторон, данные лицензиара (ООО «ИнфоКом»): ИНН/КПП, адрес; данные лицензиата (покупателя): название, ИНН, адрес, контактные данные (email, телефон);
- платёжный метод, позволяющий выбрать способ оплаты через платёжную систему «Robokassa» или банковский перевод;
- возможность скачать счёт в формате PDF для последующей оплаты вне системы по реквизитам;
- кнопка для перехода к оплате, доступна кнопка «Оплатить», после нажатия которой пользователь переходит к платёжной системе.

Выходные данные представлены в структурированной, читаемой и наглядной форме, обеспечивающей удобство для пользователя при проверке заказа перед оплатой и последующем учёте. Таким образом, программа должна обеспечить полный цикл оформления заказа с возможностью дальнейшей оплаты через интегрированные платёжные механизмы либо вручную по реквизитам счёта.

4.1.4. Требования к временным характеристикам

Программа должна обеспечивать выполнение ключевых операций в следующие временные рамки при нормальной нагрузке и стабильной работе технических средств: время создания заказа не более 500 миллисекунд, время выставления счета: не более 1 секунды, время обработки платежа: не более 2 секунд, время обновления статуса заказа в CRM: не более 1 секунды.

Продолжение ПРИЛОЖЕНИЯ Б

4.2 Требования к надёжности

4.2.1. Требования к обеспечению надежного функционирования программы

Для надёжного функционирования системы должны обеспечиваться следующие организационно-технические мероприятия:

- организация бесперебойного питания технических средств;
- организация бесперебойного доступа к сети «Интернет»;
- организация бесперебойного доступа к Robokassa;
- организация бесперебойного доступа к Bitrix24;
- наличие доменного имени;
- наличие статического IP-адреса.

4.2.2. Время восстановления после отказа

Восстановление системы после отказа, вызванного отсутствием электропитания, не должно превышать времени загрузки и запуска служб операционной системы. Восстановление системы после отказа, вызванного отсутствием доступа к сети «Интернет», не должно превышать времени восстановления доступа к сети. Восстановление системы после отказа, вызванного отсутствием доступа к Robokassa или Bitrix24, не должно превышать времени восстановления доступа к нему.

4.3. Условия эксплуатации

4.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации определяются, требованиями к климатическим условиям эксплуатации используемых технических средств.

4.3.2 Требования к видам обслуживания

Требования к видам обслуживания включают в себя требованиями к видам обслуживания используемых технических средств.

4.3.3 Требования к численности и квалификации персонала

Продолжение ПРИЛОЖЕНИЯ Б

Для развёртывания и поддержания работоспособности биллинг-системы достаточно одного системного администратора.

В задачи системного администратора входят:

- поддержание работоспособности технических средств;
- развёртывание системы и необходимого программного обеспечения;
- владение навыками работы с Bitrix24.

Для корректного обслуживания клиентов организации требуется один специалист по поддержке, который изучил работу биллинг-системы, а также навыками работы с Bitrix24.

Требования к квалификации клиентов организации ограничиваются умением пользоваться веб-интерфейсом для оформления заказов и оплаты услуг, а также базовыми навыками работы с интернет-браузерами.

4.4 Требования к составу и параметрам технических средств

Состав технических средств ограничивается сервером и клиентскими компьютерами.

К серверному компьютеру предъявляются следующие минимальные требования:

- процессор x86 с двумя ядрами и тактовой частотой, не менее 3.5 ГГц или 2 vCPU;
- оперативная память объёмом, не менее 8 Гб;
- жёсткий диск объёмом не менее 40 Гб;
- выход в интернет, со скоростью от 1000Мбит/с.

Требования к клиентским устройствам определяются используемым web-браузером на основе Chromium начиная с версии 44.0.2403.157.

4.5 Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам и методам решения

Продолжение ПРИЛОЖЕНИЯ Б

Пользовательский интерфейс должен быть реализован на основе web-технологий. Взаимодействие клиентских устройств с серверной частью должно осуществляться посредством REST API в формате JSON.

4.5.2 Требования к исходным кодам и языкам программирования

Для написания серверной части системы используется язык программирования Python, а для клиентской части JavaScript.

4.5.3 Требования к программным средствам, используемым программой

Программа должна быть разработана с использованием программных средств, обеспечивающих поддержку контейнеризации в среде Docker.

Для хранения структурированных данных программы необходимо использовать реляционную систему управления базами данных (СУБД) PostgreSQL. Этот выбор обусловлен высокой надежностью, поддержкой транзакций и эффективной работой с большими объемами информации.

В случае необходимости реализации кэша необходимо использовать Redis.

Для обеспечения стабильной работы программы допускается использование обратного прокси-сервера. Он может выполнять функции балансировки нагрузки, маршрутизации запросов и защиты от внешних угроз. Рекомендуется применять используемые решения Nginx или Traefik, благодаря их совместимости с Docker и поддержке автоматической настройки.

4.6 Требование к маркировке и упаковке

Требования к упаковке отсутствуют так как система передаётся по сети «Интернет», через приватный репозиторий GitLab.

4.7 Требования к транспортированию и хранению

Отсутствуют, т.к. распространение по сети «Интернет», через приватный репозиторий GitLab.

4.8 Специальные требования

4.8.1 Требования к эргономике и технической эстетике

Продолжение ПРИЛОЖЕНИЯ Б

Эргономические и эстетические требования включают удобство взаимодействия пользователей с интерфейсом, простоту восприятия информации и лёгкость навигации. Интерфейс должен быть интуитивно понятен и удобен для всех категорий пользователей, независимо от их уровня подготовки. Кнопки, функции и меню должны располагаться логично и быть легко доступными. Поля ввода должны сопровождаться четкими подписями, а ключевые действия выделяться визуально.

Текст должен быть читаемым контрастным, с шрифтом без засечек, а данные: структурированными в таблицах или карточках для быстрого анализа. Процесс взаимодействия должен быть последовательным, с минимальным количеством шагов, завершаемых на одной странице. Должна быть гармоничной и спокойной преобладание белого фона с черным текстом и акцентными цветами, обеспечивая комфортное взаимодействие на протяжении длительного времени. Шрифты, значки и кнопки должны быть единообразными. Элементы управления должны быть достаточно крупными для удобного нажатия, с достаточным расстоянием между ними для исключения случайных нажатий. Интерфейс должен быть визуально привлекательным, сочетая минимализм.

5 Требования к программной документации

Требования к документации устанавливаются следующими документами:

- ГОСТ 19.101-77 Виды программных документов;
- ГОСТ 19.102-77 Стадии разработки;
- ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению;
- ГОСТ 19.401-78 Текст программы;
- ГОСТ 19.402 -78 Описание программы;
- ГОСТ 19.404-79 Пояснительная записка;
- ГОСТ 19.502-78 Описание применения;
- ГОСТ 19.503-79 Руководство системному программисту;

Продолжение ПРИЛОЖЕНИЯ Б

- ГОСТ 19.504-79 Руководство программиста;
- ГОСТ 19.505-79 Руководство оператору;
- ГОСТ 19.701-90 Схемы алгоритмов, программ данных и систем.

6 Стадии и этапы разработки

Таблица Б.1 – Стадии и этапы разработки

Стадия разработки	Этапы разработки	Сроки	Содержание работ
Техническое задание	Определение требований к системе.	27.03.2025 – 28.03.2025	Проведение собеседования с заказчиком с целью выявления требований.
	Разработка технического задания.	31.03.2025 – 05.04.2025	Составление технического задания.
	Согласование и утверждение технического задания.	06.04.2025	Техническое задание должно быть изучено заказчиком, после чего должны быть устранены все недочёты и подписано техническое задание.
Эскизный проект	Разработка эскизного проекта.	07.04.2025 – 20.04.2025	Декомпозиция задачи, определение форматов данных и интерфейсов.
Технический проект	Разработка технического проекта	21.04.2025 – 27.04.2025	Детальное описание методов и алгоритмов.
Рабочий проект	Разработка рабочего проекта.	28.04.2025 – 18.05.2025	Реализация программы.
	Разработка программной документации.	19.05.2025 – 25.05.2025	Разработка программных документов в соответствии с требованиями.
Внедрение	Тестирование программного средства.	26.05.2025 – 01.06.2025	Тестирование программного средства.

7 Порядок контроля и приёмки

Приемочные испытания биллинг-системы проводятся представителями исполнителя и заказчика в соответствии с «Программой и методикой испытаний», разработанной исполнителем и утвержденной заказчиком. Процесс приемно-сдаточных испытаний включает проверку соответствия системы требованиям технического задания. Результаты документируются в протоколе испытаний, на основании которого подписывается акт приемки-сдачи программы в эксплуатацию.

Диаграмма вариантов использования

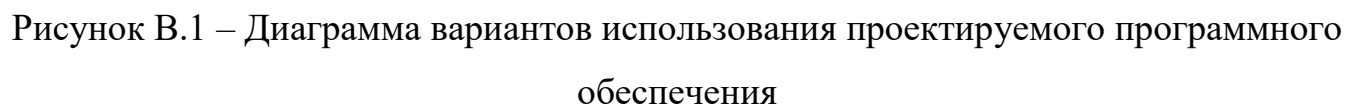


Рисунок В.1 – Диаграмма вариантов использования проектируемого программного обеспечения

ПРИЛОЖЕНИЕ Г

Функциональная модель программы

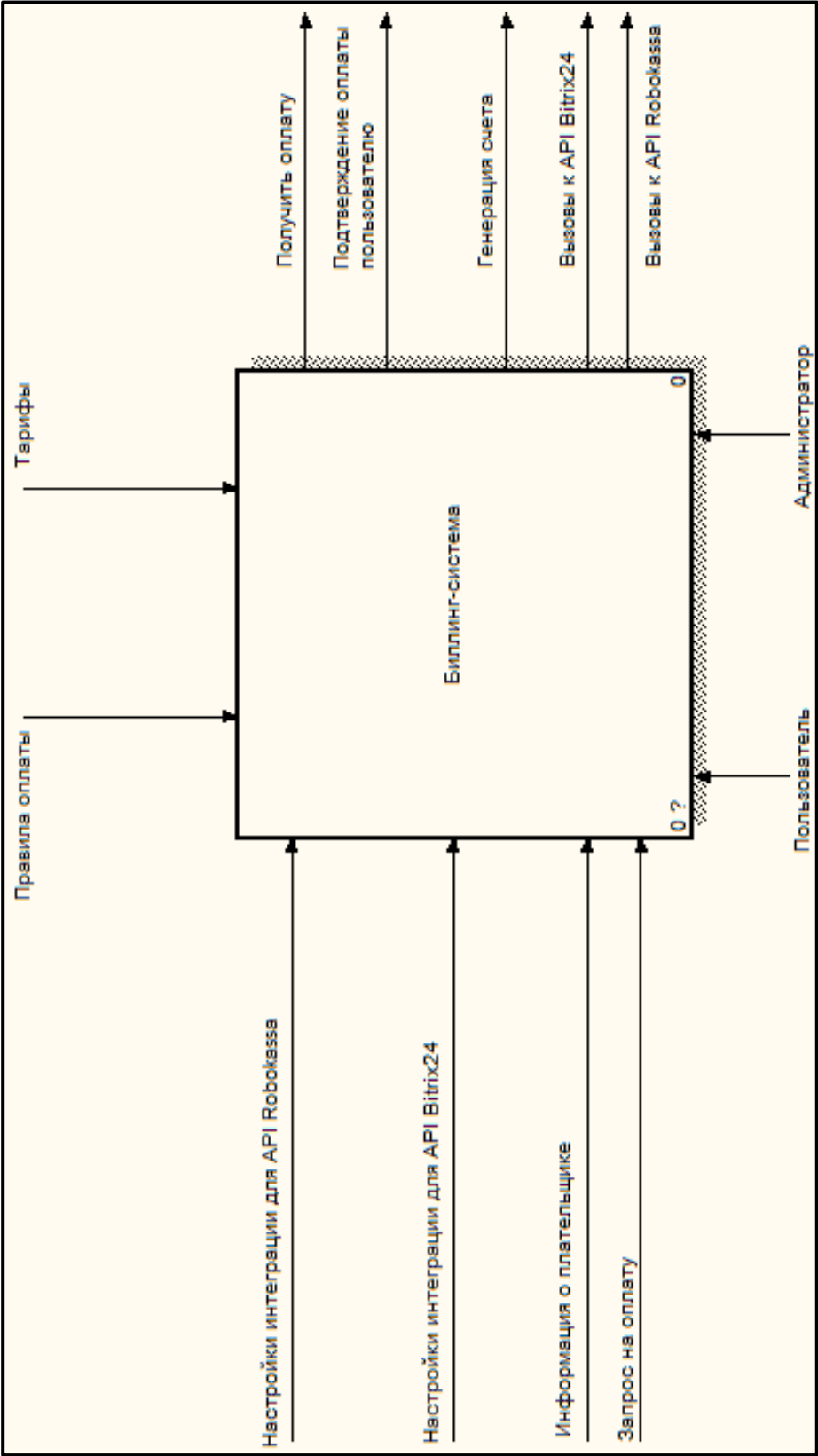


Рисунок Г.1 – Функциональная модель программы в нотации IDEF0

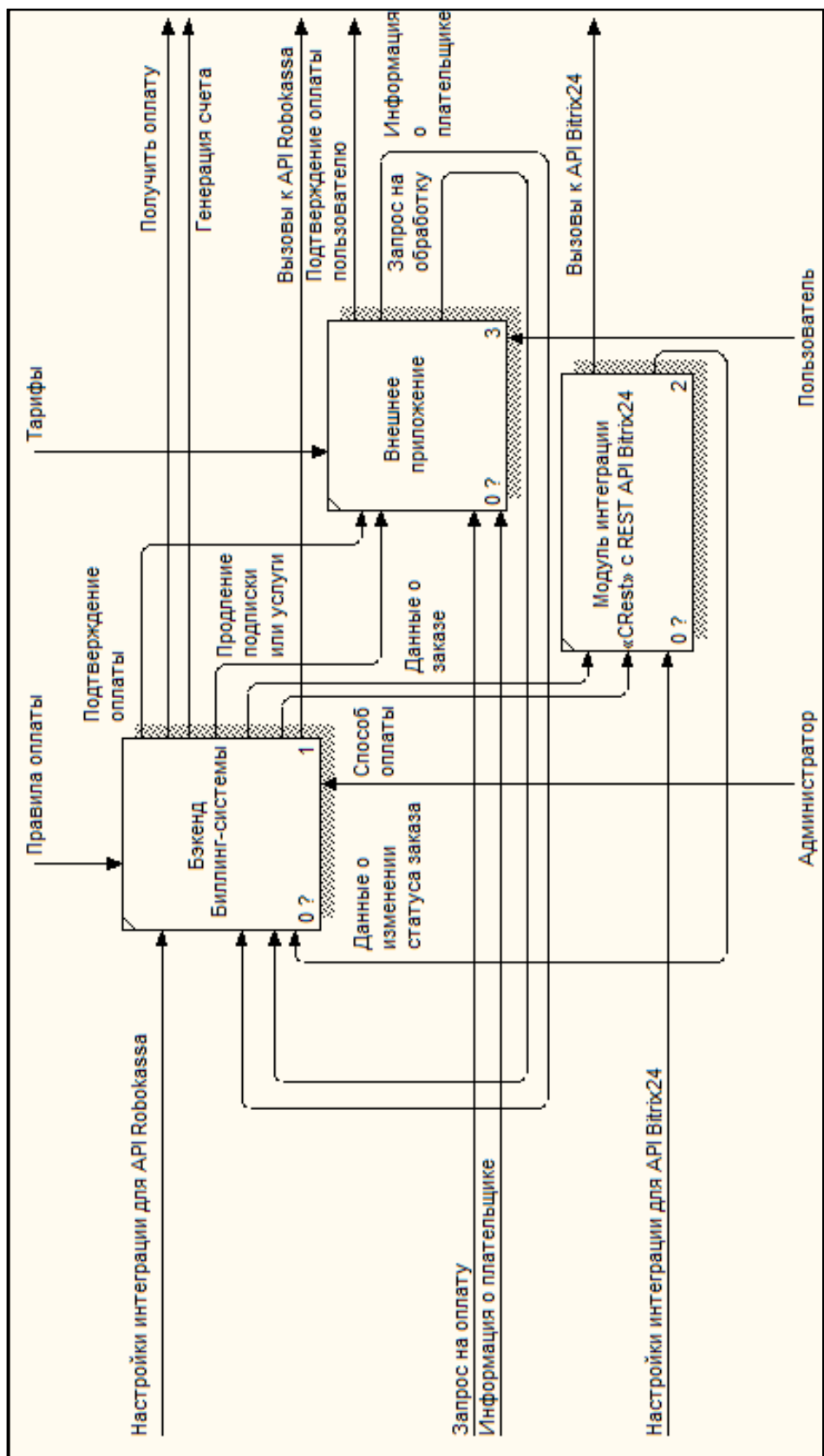


Рисунок Г.2 – Декомпозиция первого уровня функциональной модели программы

ПРИЛОЖЕНИЕ Д Диаграмма компонентов

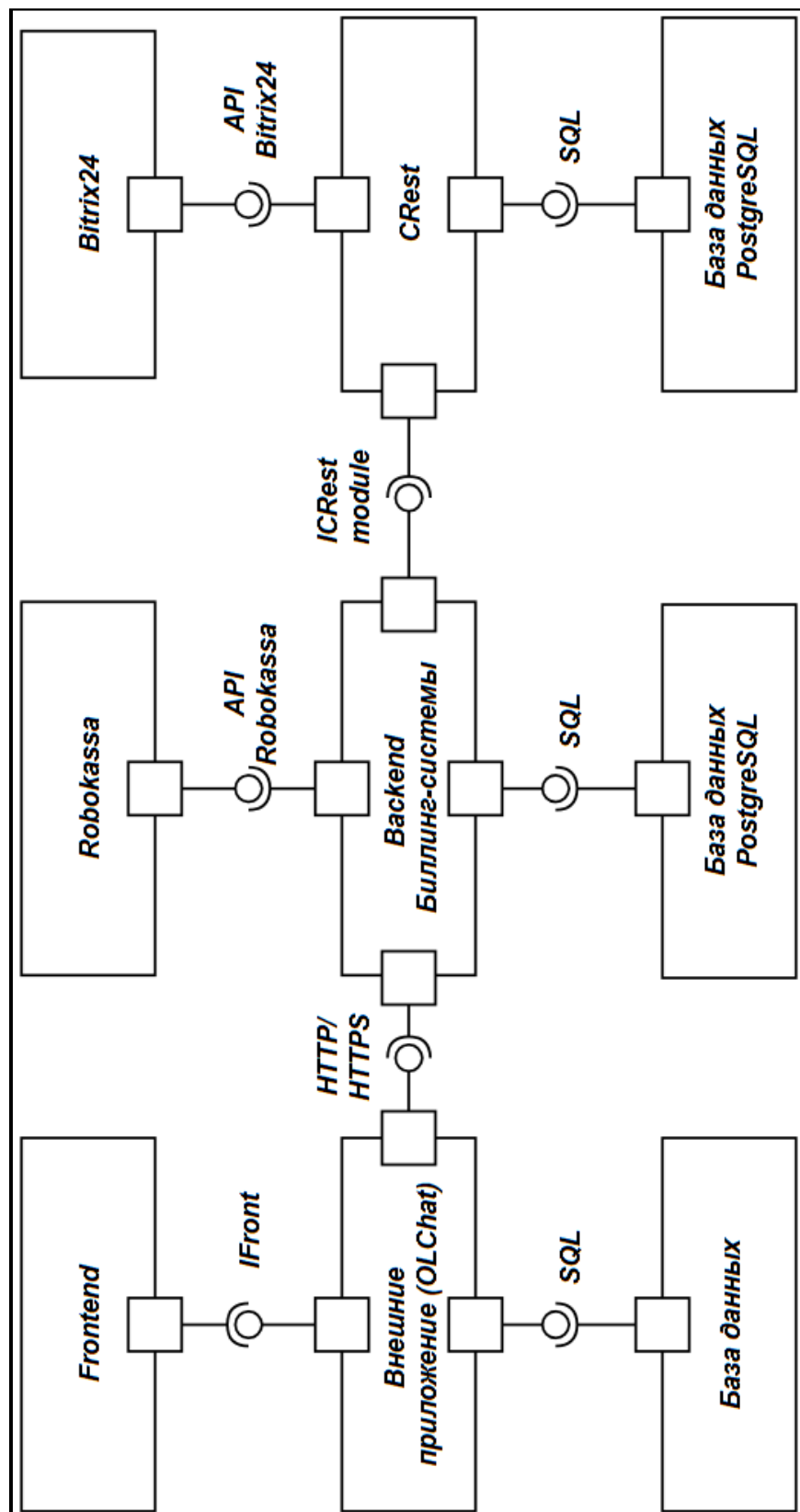


Рисунок Д.1 – Диаграмма компонентов программы

ПРИЛОЖЕНИЕ Е

Диаграммы последовательностей и состояний

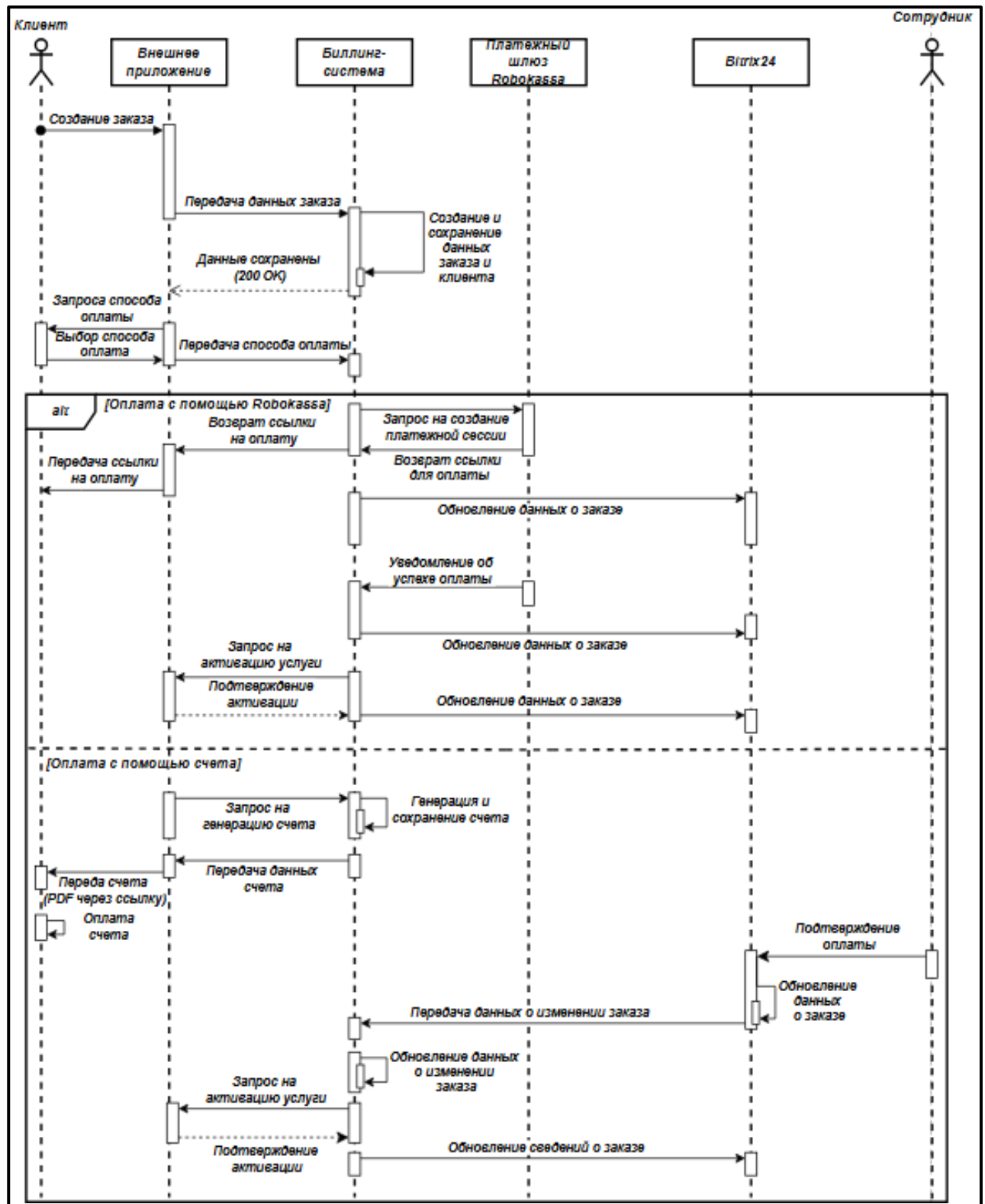


Рисунок Е.1 – UML-диаграмма последовательностей, показывающая взаимодействие биллинг-системы

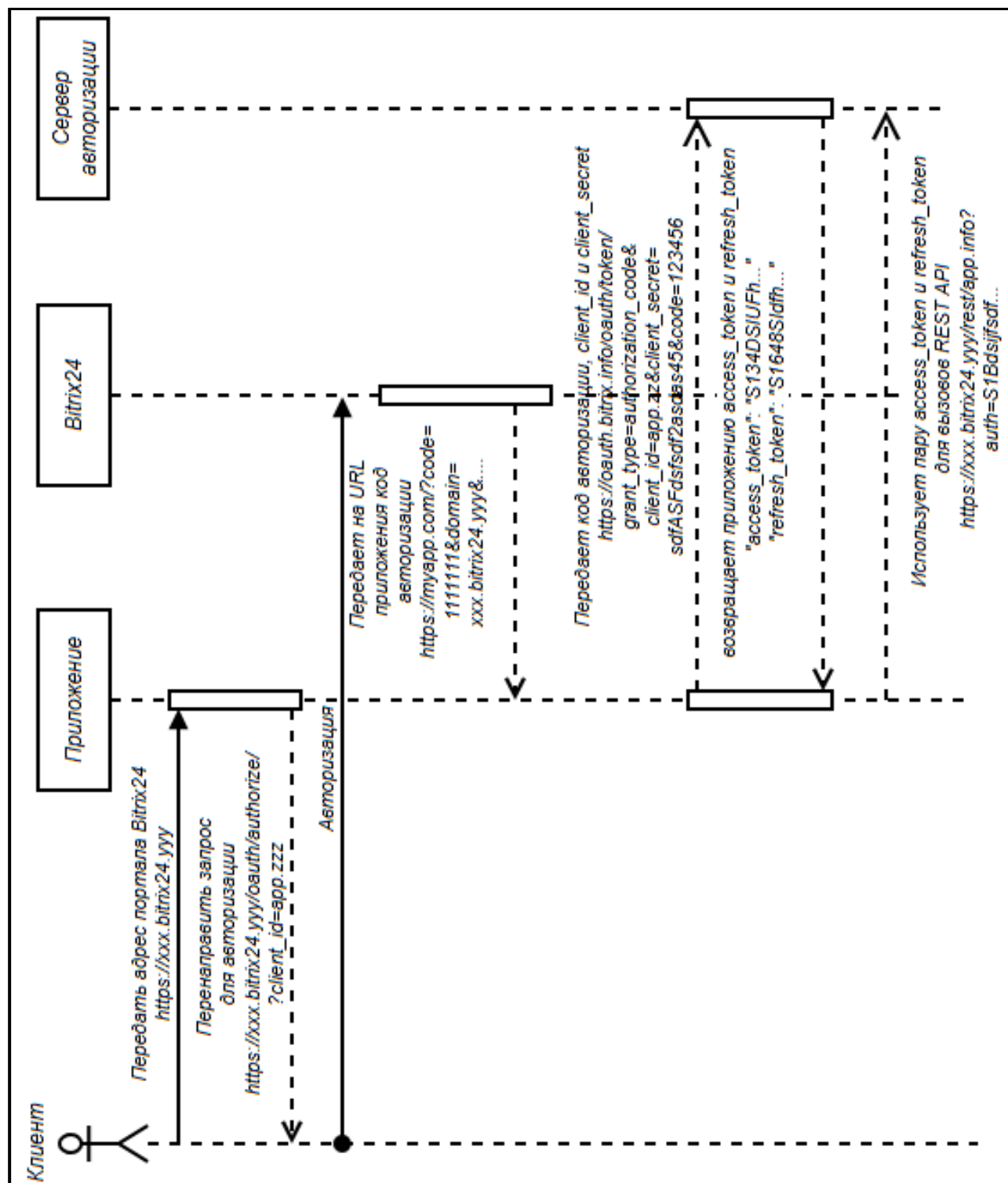


Рисунок Е.2 – UML-диаграмма последовательностей, описывающая взаимодействие приложения, интегрированного с CRest, с Bitrix24

Продолжение ПРИЛОЖЕНИЯ Е

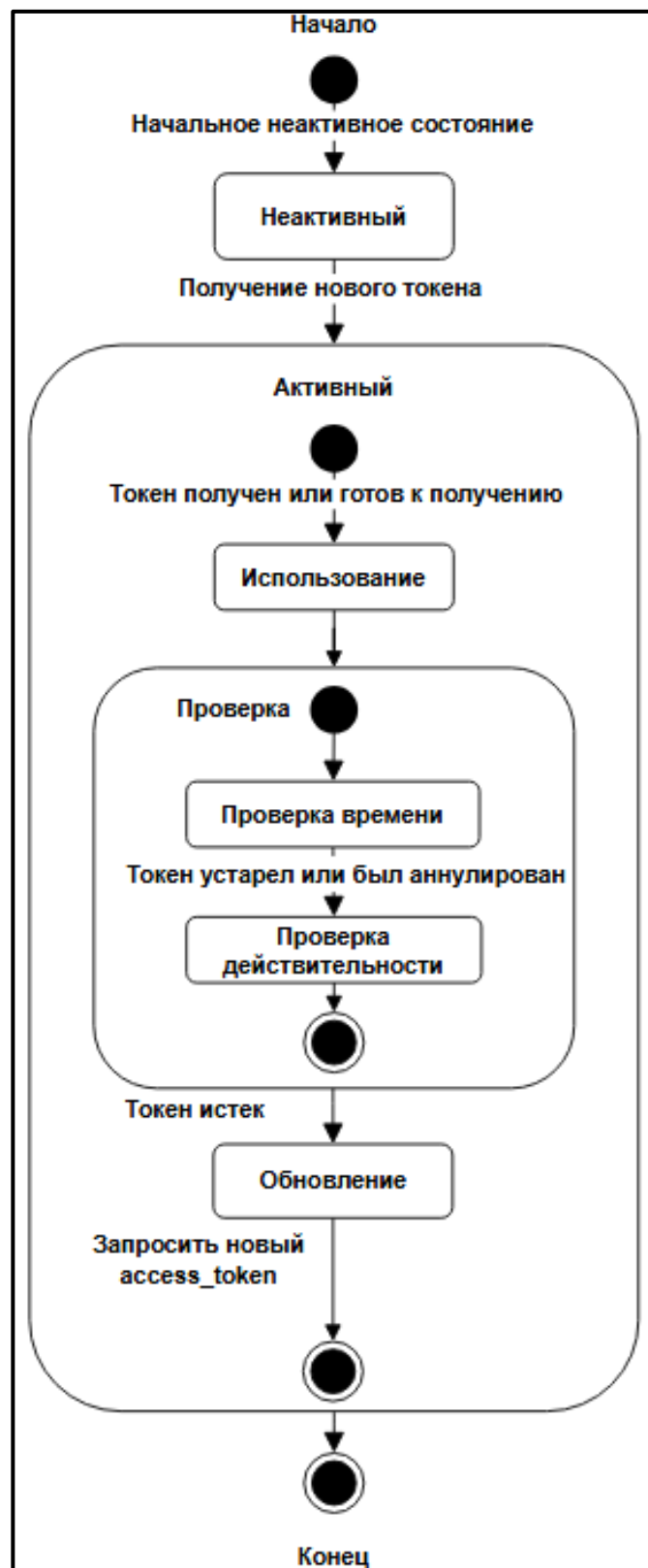


Рисунок Е.3 – UML-диаграмма состояний, описывающая жизненный цикл access_token в процессе авторизации и взаимодействия с API Bitrix24

ПРИЛОЖЕНИЕ Ж Сущности базы данных

Таблица Ж.1 – Атрибуты сущности «Организация»

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код организации	Идентификатор организации	UUID	-	123e4567-e89b-12d3-a456-426614174000
Название	Название организации	Текстовый	-	ООО «ИНФОКОМ»
Юридический адрес	Юридический адрес организации	Текстовый	-	г Москва, Рязанский пр-кт, д 8А стр 1, помещ III, ком 18
ОГРН	ОГРН (для организаций в РФ)	Числовой	-	1087746418033
ОГРНИП	ОГРНИП (для ИП в РФ)	Числовой	-	-
ИНН	ИНН (для организаций в РФ)	Числовой	-	7717615385
КПП	КПП (для организаций в РФ)	Числовой	-	772101001
Электронная почта	Контактная электронная почта	Текстовый	-	corp_infocom@gmail.com
Номер телефона	Контактный номер телефона	Текстовый	-	79234234435
Реквизиты	Реквизиты (для организаций не из РФ)	Текстовый	-	-
Код страны	Код страны	Перечисление	-	RU
Время последнего использования	Время последнего использования	Дата-время	-	2025-04-06T00:49:27.919Z
Идентификатор ЭДО	Идентификатор ЭДО	Текстовый	-	2BM-9617641160-961701000-...
Расчетный счет	Расчетный счет компании	Текстовый	-	40702810500000000033
Флаг скрытия	Флаг скрытия организации	Логический	-	False
Идентификатор в Bitrix24	Идентификатор Bitrix24	Числовой	-	1001

Продолжение ПРИЛОЖЕНИЯ Ж

Таблица Ж.2 – Атрибуты сущности «Получатель платежа»

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код получателя	Идентификатор получателя	UUID	-	123e4567-e89b-12d3-a456-426614174000
Название	Наименование получателя	Текстовый	-	ООО «ИНФОКОМ»
Данные для генерации счета	Данные для генерации счета	JSON	-	{"bank": "Сбербанк", "account": "42810100001"}

Таблица Ж.3 – Атрибуты сущности «Приложение»

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код приложения	Идентификатор приложения	UUID	-	123e4567-e89b-12d3-a456-426614174000
Название	Название приложения	Текстовый	-	OLChat
URL для продления услуги	URL для продления услуги	Текстовый	-	https://bapitest.dev.icdv.ru/cart/activate
Входящий секретный ключ	Входящий ключ	Текстовый	-	secret123
Исходящий секретный ключ	Исходящий ключ	Текстовый	-	secret456
Инструкции	Инструкции	JSON	-	{"payee_id": {"type": "const", "value": "3424ecfb-050e-4e4d-8906-61a44afd6473"}}

Таблица Ж.4 – Атрибуты сущности «Заказ»

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код заказа	Идентификатор заказа	UUID	-	123e4567-e89b-12d3-a456-426614174000
Код заказа в Bitrix24	Идентификатор заказа в Bitrix24	Числовой	-	5
Код портала	Идентификатор портала	Текстовый	-	53e4410f83cc457fbaa89054213a4781
Код приложения	Идентификатор приложения	UUID	-	3424ecfb-050e-4e4d-8906-61a44afd5173
Код плательщика	Идентификатор плательщика	UUID	-	123e4567-e89b-12d3-a456-426614174000

Продолжение ПРИЛОЖЕНИЯ Ж

Продолжение таблицы Ж.4

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код получателя	Идентификатор получателя платежа	UUID	-	123e4567-e89b-12d3-a456-426614174000
Корзина	Корзина (данные о услугах)	JSON	-	{"title": "OlChat", "type": "group", "period": 360, "group_count": 100, "price": 2100, "portal_id": "...", ...}
Объект для активации услуги	Объект для активации услуги	JSON	-	{"domain": "test_0.domain.ru", "portal_id": "...", "group": {"period": {"days": 360, "price": 2100}, ...}
Стоимость	Стоимость заказа	Числовой	-	2100.00
Код валюты	Код валюты	Перечисление	-	RUB
Дата-время создания	Дата-время создания заказа	Дата-время	-	2025-04-06T00:49:27.919Z
Дата-время оплаты	Дата-время оплаты заказа	Дата-время	-	2025-04-06T01:00:00.000Z
Способ оплаты	Способ оплаты	Перечисление	-	ROBOKASSA
Данные оплаты	Данные оплаты	JSON	-	https://auth.robokassa.ru/
Флаг скрытия	Флаг скрытия заказа	Логический	-	False
Статус	Статус заказа	Перечисление	-	Оплачено
URL на документ	URL на счет	Текстовый	-	https://.../orders/invoice/...
Дополнительная информация	Дополнительная данные	JSON	-	{"note": "Срочный заказ"}
Токен короткой ссылки	Токен короткой ссылки	Текстовый	-	xeVCT57O
Номер счета	Номер счета	Числовой	-	1

Таблица Ж.5 – Атрибуты сущности «Токен»

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код токена	Уникальный идентификатор	UUID	—	7b56c769-c8e4-4f80-8c07-8dbe8415bf9c

Продолжение ПРИЛОЖЕНИЯ Ж

Продолжение таблицы Ж.5

Наименование атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код приложения	Внешний ключ «Приложение»	UUID	—	123e4567-e89b-12d3-a456-426614174000
Входящий ключ токена	Строковое значение, которое передаёт приложение в заголовке/параметре запроса	Текстовый	—	tf1e3d4e0a5842c9
Дата-время выдачи	Дата генерации токена	Дата-время	—	2025-04-06T09:15:42Z
Флаг отзыва	Признак аннулированного токена	Логический	True / False	False

Продолжение ПРИЛОЖЕНИЯ Ж

Таблица Ж.6 – Таблица связей сущностей

№	Сущность 1	Сущность 2	Наименование связи	Кратность	Описание
1	Организация	Заказ	«Осуществляет / Осуществляется»	1 : N	Одна организация может осуществлять несколько заказов. Это отражает возможность того, что одна и та же организация совершает множество покупок или платежей. Один заказ осуществляется одной организацией. Каждый заказ привязан к конкретному плательщику, что обеспечивается внешним ключом payer_id в сущности «Заказ», ссылающимся на «Организацию».
2	Получатель	Заказ	«Получает / Направлен»	1 : N	Один заказ направлен одному получателю платежа. Каждый заказ имеет конкретного получателя, указанного через внешний ключ payee_id в сущности «Заказ». Один получатель платежа может получать платежи по нескольким заказам. Это означает, получатель может быть указан в разных заказах.
3	Приложение	Заказ	«Имеет заказы / Относится»	1 : N	Один заказ относится к одному приложению. Каждый заказ связан с конкретным приложением, через внешний ключ application_id. Одно приложение может быть связано с несколькими заказами.
4	Приложение	Токен	«Выдается / Выдан»	1 : N	Для интеграции внешних сервисов приложение генерирует несколько токенов API; Приложение содержит несколько токенов доступа. Каждый токен принадлежит единственному приложению.

Продолжение ПРИЛОЖЕНИЯ Ж

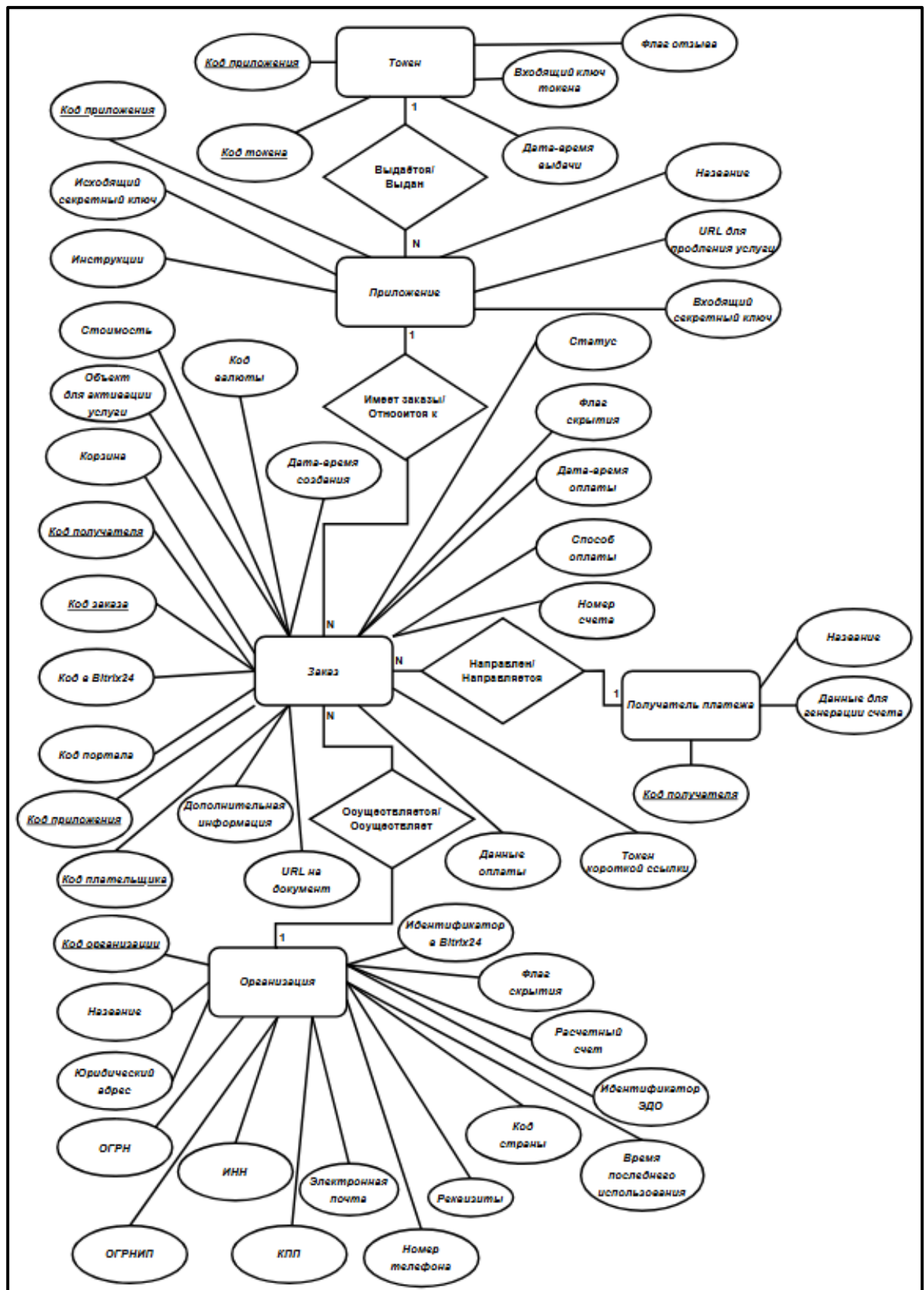


Рисунок Ж.1 – ER-диаграмма базы данных в нотации Чена

ПРИЛОЖЕНИЕ И

Логическое и физическое проектирование

Таблица И.1 – Классы принадлежности и схемы их отображения на отношения

№	Связь	Кратность	Класс принадлежности 1	Класс принадлежности 2	Результат преобразования
1	Организация – Заказ	1:N	Необязательный	Обязательный	Первичный ключ сущности «Организация» добавляется как внешний ключ («Идентификатор плательщика») в сущность «Заказ». Формируются отношения: «Организация» и «Заказ».
2	Получатель платежа – Заказ	1:N	Необязательный	Обязательный	Первичный ключ сущности «Получатель платежа» добавляется как внешний ключ («Идентификатор получателя») в сущность «Заказ». Формируются отношения: «Получатель платежа» и «Заказ».
3	Приложение – Заказ	1:N	Необязательный	Обязательный	Первичный ключ сущности «Приложение» добавляется как внешний ключ («Идентификатор приложения») в сущность «Заказ». Формируются отношения: «Приложение» и «Заказ».
4	Приложение – Токен	1:N	Необязательный	Обязательный	Первичный ключ сущности «Приложение» добавляется как внешний ключ («Идентификатор приложения») в сущность «Токен». Формируются отношения: «Приложение» и «Токен».

Продолжение ПРИЛОЖЕНИЯ И

Таблица И.2 – Физическая структура данных отношения «Организация»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Идентификатор организации	UUID	He NULL	UUID	PRIMARY KEY
Название	String	He NULL	VARCHAR	–
Юридический адрес	String	He NULL	VARCHAR	–
ОГРН	BigInteger	Может быть NULL	BIGINT	–
ОГРНИП	BigInteger	Может быть NULL	BIGINT	–
ИНН	BigInteger	Может быть NULL	BIGINT	–
КПП	BigInteger	Может быть NULL	BIGINT	–
Электронная почта	String	He NULL	VARCHAR	–
Номер телефона	String	He NULL	VARCHAR	–
Реквизиты	String	Может быть NULL	VARCHAR	–
Код страны	Enum	Может быть NULL	ENUM	–
Время последнего использования	DateTime	He NULL	TIMESTAMP	–
Идентификатор ЭДО	String	Может быть NULL	VARCHAR	–
Расчетный счет	String	Может быть NULL	VARCHAR(20)	–
Флаг скрытия	Boolean	He NULL	BOOLEAN	–
Идентификатор в Bitrix24	Integer	Может быть NULL	INTEGER	UNIQUE
Идентификатор email в Bitrix24	Integer	Может быть NULL	INTEGER	–

Таблица И.3 – Физическая структура данных отношения «Получатель платежа»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Идентификатор получателя	UUID	He NULL	UUID	PRIMARY KEY
Название	String	He NULL	VARCHAR	–
Данные для генерации счета	JSON	He NULL	JSON	–

Продолжение ПРИЛОЖЕНИЯ И

Таблица И.4 – Физическая структура данных отношения «Приложение»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Идентификатор приложения	UUID	He NULL	UUID	PRIMARY KEY
Название	String	He NULL	VARCHAR	–
URL для продления услуги	String	He NULL	VARCHAR	–
Входящий секретный ключ	String	He NULL	VARCHAR	–
Исходящий секретный ключ	String	He NULL	VARCHAR	–
Данные для генерации счета	JSON	He NULL	JSON	–
Инструкции	JSON	He NULL	JSON	–

Таблица И.5 – Физическая структура данных отношения «Заказ»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Идентификатор заказа	UUID	He NULL	UUID	PRIMARY KEY
Идентификатор заказа в Bitrix24	Integer	Может быть NULL	INTEGER	UNIQUE
Идентификатор портала Bitrix24	String	He NULL	VARCHAR(32)	–
Идентификатор приложения	UUID	He NULL	UUID	FOREIGN KEY
Идентификаторплательщика	UUID	He NULL	UUID	FOREIGN KEY
Идентификатор получателя	UUID	Может быть NULL	UUID	FOREIGN KEY
Корзина	JSON	He NULL	JSON	–
Объект для активации услуги	JSON	He NULL	JSON	–
Стоимость	Float	He NULL	FLOAT	–
Код валюты	Enum	He NULL	ENUM	–
Дата-время создания	DateTime	He NULL	TIMESTAMP	INDEX
Дата-время оплаты	DateTime	Может быть NULL	TIMESTAMP	–
Способ оплаты	Enum	Может быть NULL	ENUM	–
Данные оплаты	JSON	Может быть NULL	JSON	–
Флаг скрытия	Boolean	He NULL	BOOLEAN	–
Статус	Enum	He NULL	ENUM	–
URL на документ	String	Может быть NULL	VARCHAR	–
Дополнительная информация	JSON	Может быть NULL	JSON	–

Продолжение ПРИЛОЖЕНИЯ И

Продолжение таблицы И.5

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Токен короткой ссылки	String	He NULL	VARCHAR(12)	UNIQUE
Номер счета	Integer	He NULL	INTEGER	UNIQUE, IDENTITY

Таблица И.6 – Физическая структура данных отношения «Токен»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Идентификатор токена	UUID	He NULL	UUID	PRIMARY KEY
Идентификатор приложения	UUID	He NULL	UUID	FOREIGN KEY
Ключ токена	String	He NULL	VARCHAR	UNIQUE
Дата-время выдачи	DateTime	He NULL	TIMESTAMP	-
Отозван	Boolean	He NULL	BOOLEAN	-

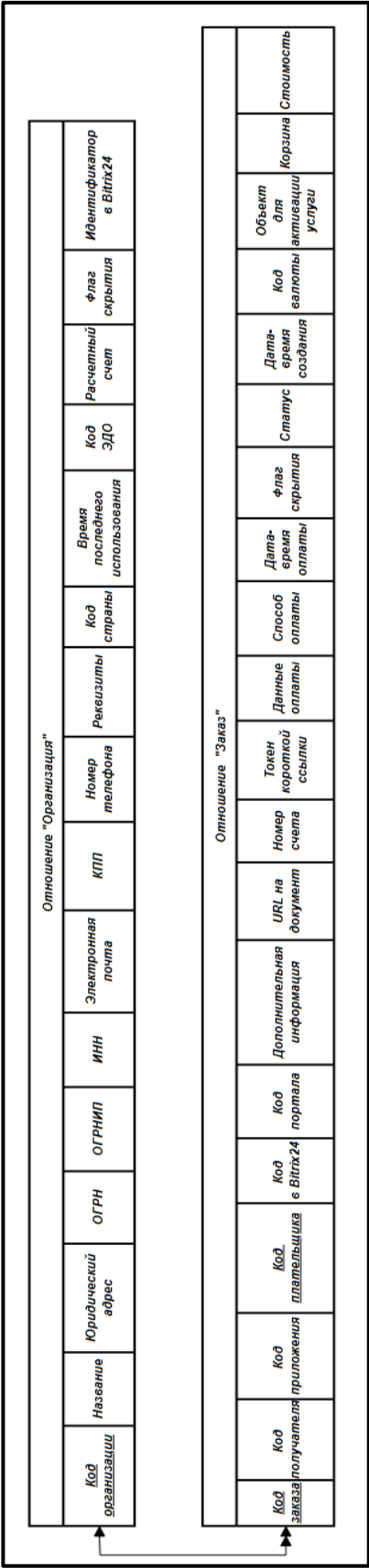


Рисунок И.1 – Связь «Организация – Заказ»

Продолжение ПРИЛОЖЕНИЯ И

Отношение "Организация"												
Код организации	Название	Юридический адрес	ОГРН	ОГРНИП	ИНН	Электронная почта	КПП	Номер телефона	Реквизиты	Код страны	Время последнего использования	Код ЭДО
											Расчетный счет	Флаг скрытия
												Идентификатор в Bitrix24
Отношение "Заказ"												
Код заказа	Код получателя	Код приложения	Код пользователя	Код портала в Bitrix24	Код портала в Bitrix24	Дополнительная информация	URL на документ	Номер счета	Токен короткой ссылки	Данные оплаты	Способ оплаты	Дата-время оплаты
										Флаг скрытия	Статус	Дата-время создания
											Объект для активации услуги	Код валюты
												Корзина
												Стоимость

Рисунок И.2 – Результат анализа связи «Организация – Заказ»

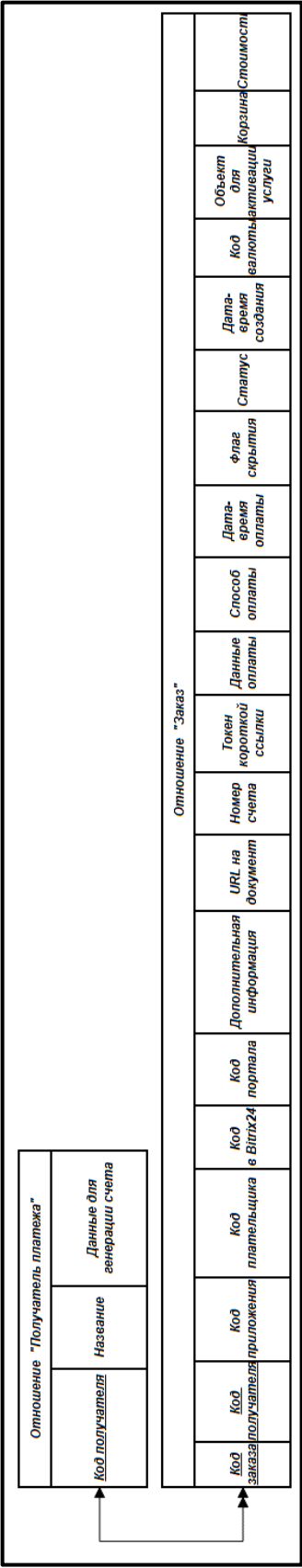


Рисунок И.3 – Связь «Получатель платежа – Заказ»

Продолжение ПРИЛОЖЕНИЯ И

Отношение "Получатель платежа"				Отношение "Заказ"															
Код получателя		Название	Данные для генерации счета																
Код заказа	Код получателя	Код приложения	Код платежителя	Код в Bitrix24	Код портала	Дополнительная информация	URL на документ	Номер счета	Токен короткой ссылки	Данные оплаты	Способ оплаты	Дата-время оплаты	Флаг скрытия	Статус	Дата-время создания	Код валюты активации	Объект для активации услуги	Корзина	Стоимость

Рисунок И.4 – Результат анализа связи «Получатель платежа – Заказ»

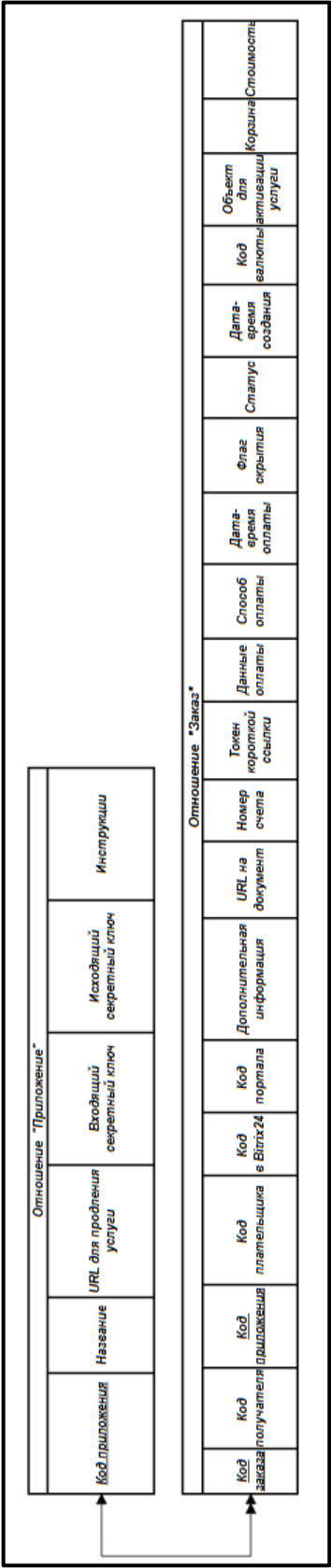


Рисунок И.5 – Связь «Приложение – Заказ»

Продолжение ПРИЛОЖЕНИЯ И

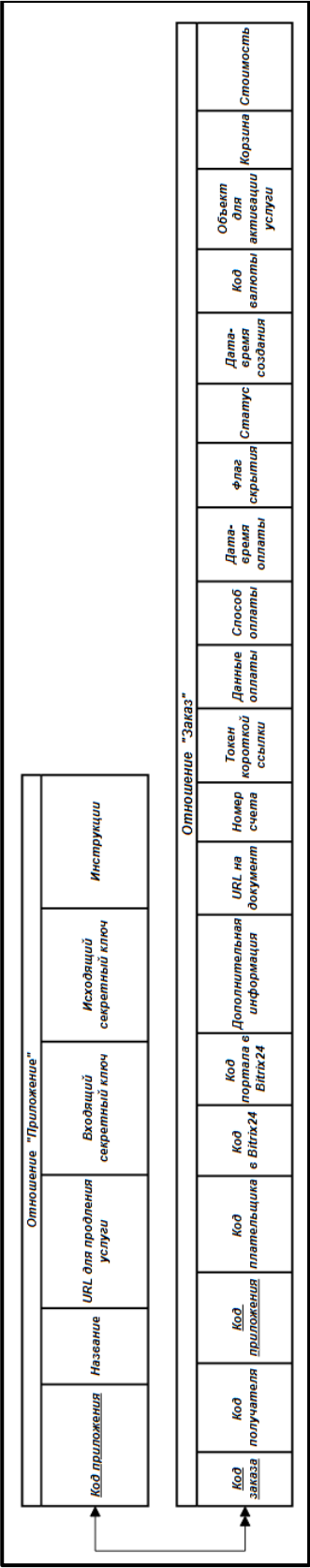


Рисунок И.6 – Результат анализа связи «Приложение – Заказ»

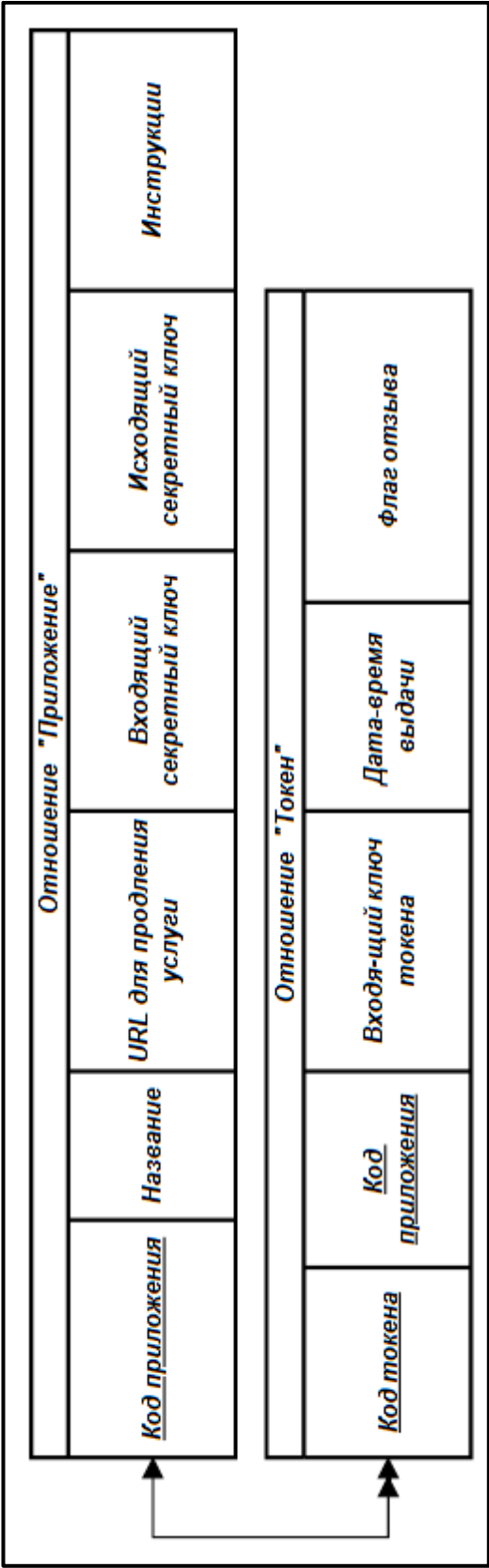


Рисунок И.7 – Связь «Приложение – Токен»

Продолжение ПРИЛОЖЕНИЯ И

Отношение "Приложение"				
<u>Код приложения</u>	Название	URL для продления услуги	Входящий секретный ключ	Исходящий секретный ключ
Инструкции				
Отношение "Токен"				
<u>Код токена</u>	<u>Код приложения</u>	Входящий ключ токена	Дата-время выдачи	Флаг отзыва

Рисунок И.8 – Результат анализа связи «Приложение – Заказ»

Отношение "Организация"																				
Код организации	Название	Юридический адрес	ОГРН	ОГРНИП	ИНН	Электронная почта	КПП	Номер телефона	Реквизиты	Код страны	Время последнего использования	Код ЭДО	Расчетный счет	Флаг скрытия	Идентификатор в Bitrix24					
Отношение "Получатель платежа"																				
Код получателя		Данные для генерации счета																		
Название																				
Отношение "Приложение"																				
Код приложения	Название	URL для продления услуги	Входящий секретный ключ	Исходящий секретный ключ	Инструкции															
Отношение "Заказ"																				
Код заказа	Код получателя	Код приложения	Код платящего	Код в Bitrix24	Код портала в Bitrix24	Дополнительная информация	URL на документ	Номер счета	Токен короткой ссылки	Данные оплаты	Способ оплаты	Дата-время оплаты	Флаг скрытия	Статус	Дата-время создания	Код валюты	Объект для активации услуги	Корзина	Стоимость	
Отношение "Токен"																				
Код токена	Код приложения	Входящий ключ токена	Дата-время выдачи	Флаг отзыва																

Рисунок И.9 – Набор отношений, полученных при преобразовании сущностей

Продолжение ПРИЛОЖЕНИЯ И

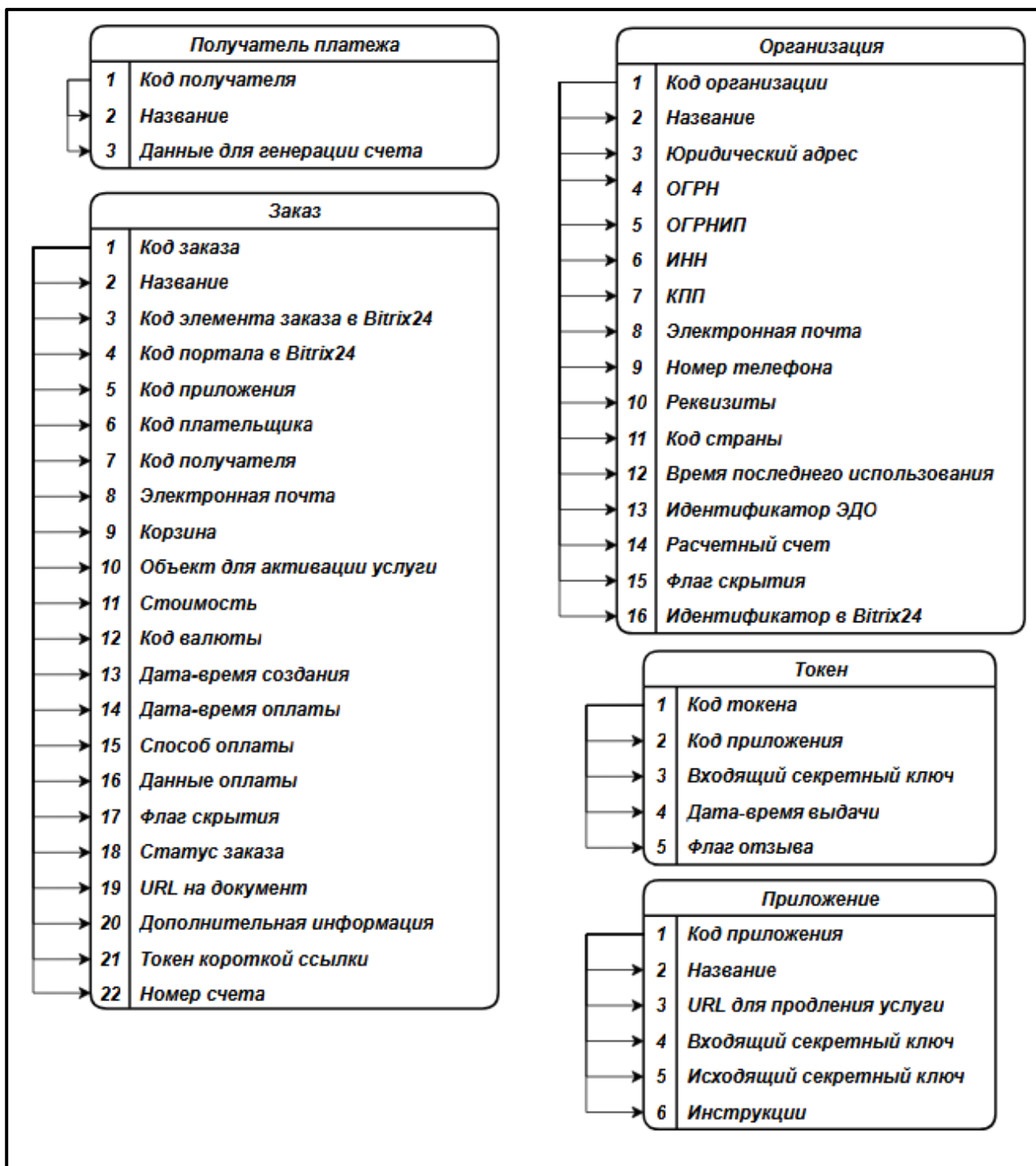


Рисунок И.10 – Зависимости атрибутов отношений базы данных

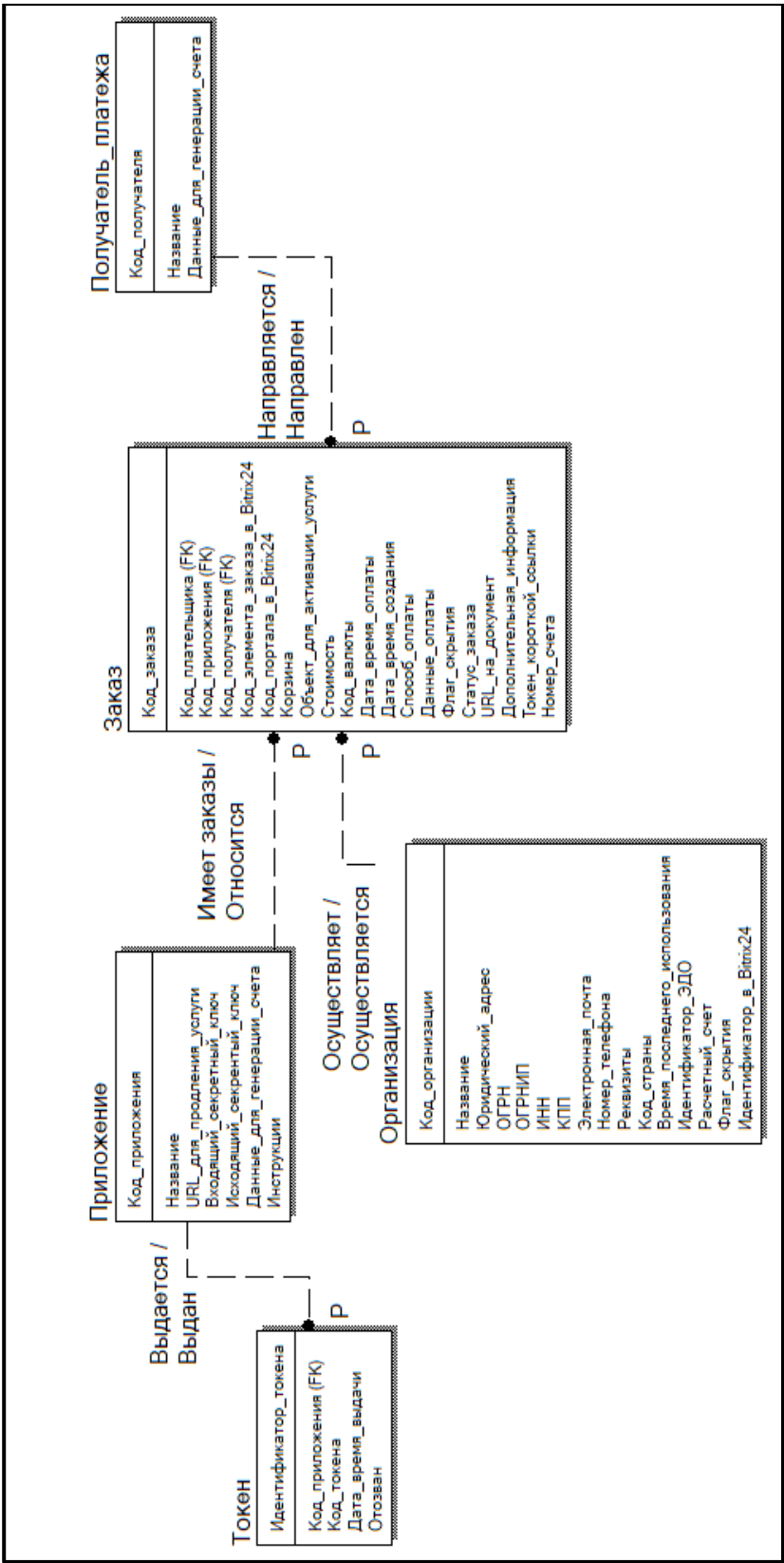


Рисунок И.11 – Диаграмма базы данных логического проектирования в нотации IDEF1X

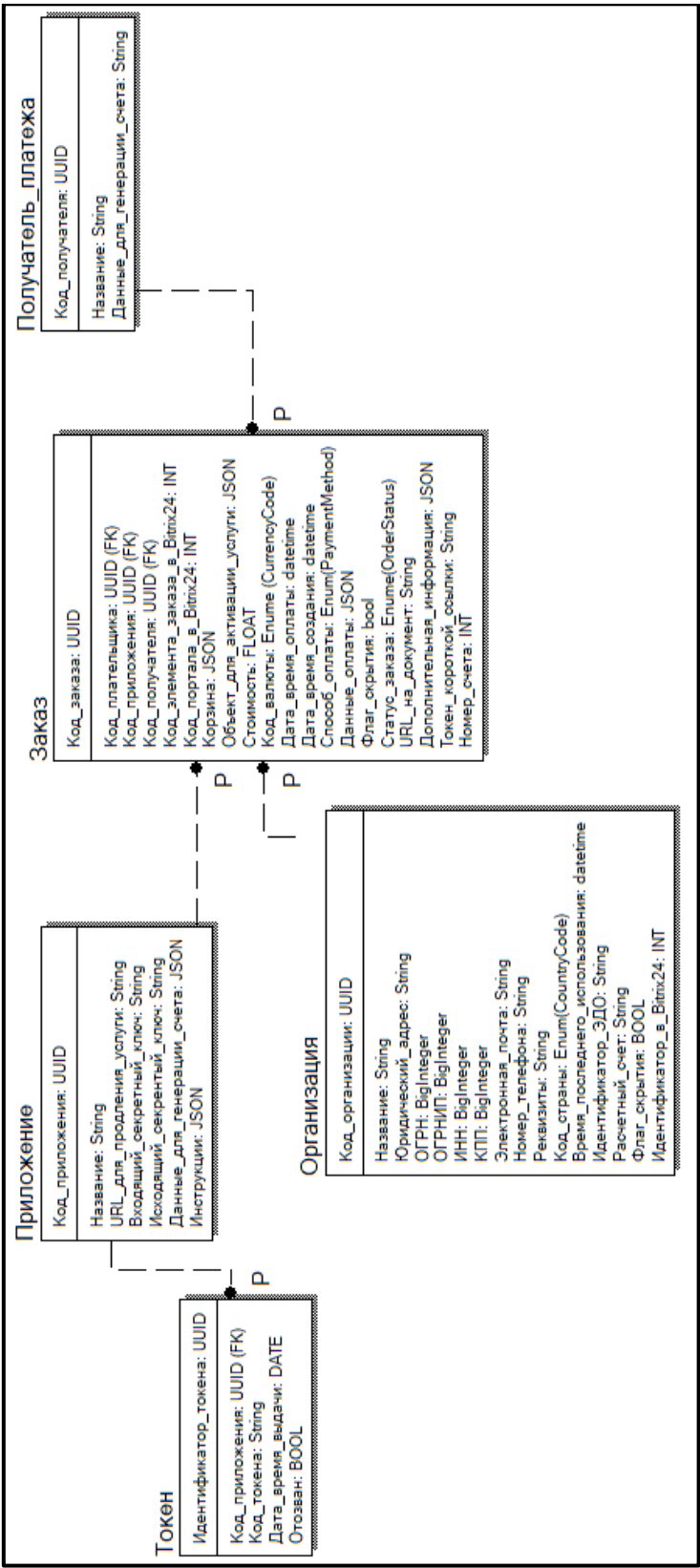


Рисунок И.12 – Диаграмма базы данных физического проектирования в нотации IDEF1X