МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

*Систем сбора и обработки данных*

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название кафедры)

# Утверждаю

#### Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, инициалы, фамилия)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

*ССОД*

*Бакаев М.А.*

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

*Буханцева Максима Олеговича*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество студента – автора работы)

*Разработка сервиса телемаркетинга для финансовых организаций*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(тема работы)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Автоматики и вычислительной техники*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название факультета)

*09.03.02 Информационные системы и технологии*

Направление подготовки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Старший преподаватель*

### Руководитель

### от НГТУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, ученое звание)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

### Автор выпускной квалификационной работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(факультет, группа)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

*АВТФ, АТ-04*

*Буханцев Максим Олегович*

*Воронов Виталий Владимирович*

Новосибирск 2024

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

*Систем сбора и обработки данных*

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название кафедры)

##### УТВЕРЖДАЮ

# Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

*Бакаев М.А.*

*Введите здесь свой текст*

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

*Буханцеву Максиму Олеговичу*

студенту *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(фамилия, имя, отчество)

*09.03.02 Информационные системы и технологии*

Направление подготовки \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и наименование направления подготовки бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Автоматики и вычислительной техники*

(полное название факультета)

*Разработка сервиса телемаркетинга для финансовых организаций*

Тема \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название темы выпускной квалификационной работы бакалавра)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Введите здесь свой текст*

*текст*

*текст*

*текст*

*текст*

*текст*

Исходные данные (или цель работы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Введите здесь свой текст*

*текст*

*текст*

*текст*

*текст*

*текст*

*текст*

Структурные части работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Введите здесь свой текст*

*текст*

*текс*

*текст*

*текст*

*текст*

*текст*

*текст*

*текст*

*текст*

*текст*

*текст*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Задание согласовано и принято к исполнению.

### Руководитель

### от НГТУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ученая степень, ученое звание)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

### Студент

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(факультет, группа)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

*Буханцев Максим Олегович*

*Воронов Виталий Владимирович*

*АВТФ, АТ-04*

*Старший преподаватель*

Тема утверждена приказом по НГТУ № \_\_\_\_\_\_\_\_\_ от «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

ВКР сдана в ГЭК № \_\_\_\_\_\_\_, тема сверена с данными приказа

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись секретаря государственной экзаменационной комиссии по защите ВКР, дата)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество секретаря государственной

экзаменационной комиссии по защите ВКР)

**7**

**7**

***Примечание: поля данного блока заполняются секретарем ГЭК при приеме ВКР к защите.***

# **АННОТАЦИЯ**

Тут я напишу краткую выжимку о работе: кол-во страниц, рисунков, источников и приложений. Пишу цель, предметную область применения моего сервиса. Пишу что именно было разработано.

**Содержание**

[**АННОТАЦИЯ** 4](#_Toc163326306)

[**1 ВВЕДЕНИЕ** 6](#_Toc163326307)

[**2 ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 9](#_Toc163326308)

[2.1 Что такое телемаркетинг: цели, принципы 9](#_Toc163326309)

[2.2 Телемаркетинг в финансовой сфере. 10](#_Toc163326310)

[2.3 Специфика бизнес – процесса внутри рассматриваемой платежной системы «Золотая Корона» 11](#_Toc163326311)

[2.4 Обзор существующих решений 12](#_Toc163326312)

[2.4.1 1С:CRM 12](#_Toc163326313)

[2.4.2 MegaCRM 17](#_Toc163326314)

[**3 ГЛАВА 2. Проектирование системы** 19](#_Toc163326315)

[3.1 Выбор инструментальных средств 19](#_Toc163326316)

[3.1.1 Язык программирования Java 19](#_Toc163326317)

[3.1.2 Автоматизированная сборка проектов с помощью Apache Maven 20](#_Toc163326318)

[3.1.3 Фреймворк Spring 21](#_Toc163326319)

[3.1.4 Объектно-реляционный фреймворк Hibernate 23](#_Toc163326320)

[3.1.5 Система управления базами данных PostgreSQL 24](#_Toc163326321)

[3.1.6 Система управления миграциями баз данных Liquibase 27](#_Toc163326322)

[3.2 Взаимодействие клиент – серверной архитектуры 30](#_Toc163326323)

[3.3 Проектирование сервисной архитектуры разрабатываемой системы 32](#_Toc163326324)

[3.3.1 Предметно-ориентированное проектирование сервисов 32](#_Toc163326325)

[3.3.2 Интеграции со сторонними сервисами 32](#_Toc163326326)

[3.3.3 Проектирование основного контура Телемаркетинга 32](#_Toc163326327)

[3.3.4 Проектирование домена «Займы» 32](#_Toc163326328)

[3.4 Проектирование структуры хранения данных 32](#_Toc163326329)

[**4 Реализация системы** 33](#_Toc163326330)

[4.1 Создание структуры хранения данных 33](#_Toc163326331)

[4.2 Программирование сервисов системы 33](#_Toc163326332)

[**5** ГЛАВА 3 34](#_Toc163326333)

[**6** ЗАКЛЮЧЕНИЕ 35](#_Toc163326334)

[**7** СПИСОК ЛИТЕРАТУРЫ 36](#_Toc163326335)

[**8** ПРИЛОЖЕНИЯ 37](#_Toc163326336)

# **ВВЕДЕНИЕ**

1 – 2 страницы. Описываю проблему, как люди живут без неё, почему её нужно решить, актуальность, и т.д.

В современном мире, информационные технологии развиваются стремительно, их применение в финансовой сфере имеет первостепенное значение. Финансовым организациям необходимо продавать свои продукты и услуги в условиях растущей конкуренции, поэтому необходимо использовать содержательные и заманчивые предложения, которые могли бы убедить потребителей воспользоваться данными товарами и услугами.

Одним из основных способов продвижения является телемаркетинг. Телемаркетинг – это способ продвижения товаров и услуг, при котором осуществляется коммуникация с потенциальными клиентами через телефонные звонки с целью продвижения продукции, получение обратной связи и проведения опросов.

Система телемаркетинга в финансовой организации решает определенные проблемы:

Во-первых, это увеличение продаж. Благодаря прямому контакту с потенциальными клиентами, предоставлению информации о продуктах и услугах, а также предложению индивидуальных условий и акций, система телемаркетинга позволяет эффективно привлекать новых клиентов и увеличивать продажи финансовых продуктов и услуг.

Во-вторых, система телемаркетинга помогает улучшить качество обслуживания клиентов. Оперативное реагирование на вопросы и запросы клиентов, решение проблем и предоставление необходимой информации способствуют повышению удовлетворенности клиентов.

Кроме того, система телемаркетинга позволяет собирать и анализировать данные о клиентах, их потребностях и предпочтениях. Это помогает лучше понимать свою целевую аудиторию и разрабатывать более эффективные маркетинговые стратегии.

Также система телемаркетинга позволяет управлять клиентской базой, отслеживать историю взаимодействия с клиентами, проводить сегментацию и персонализацию маркетинговых кампаний.

Наконец, система телемаркетинга автоматизирует процесс обзвона клиентов, что позволяет сократить время и ресурсы, затрачиваемые на контакт с каждым клиентом.

В целом, разработка системы телемаркетинга в финансовой организации может быть обоснована необходимостью повышения эффективности продаж, улучшения обслуживания клиентов, сбора и анализа данных, управления клиентской базой и экономии времени и ресурсов. Поэтому разработка сервиса телемаркетинга является актуальной задачей.

В данной выпускной квалификационной работе будет рассматриваться разработка сервиса телемаркетинга на базе платежной системы «Золотая Корона» группы компаний «Центр Финансовых Технологий».

Сложность заключается в том, что готовые системы от сторонних разработчиков не подходят для интеграции с существующим процессом, поэтому будет трудно внедрить стороннее решение, не перестраивая существующие бизнес-процессы, что может иметь непредвиденные расходы на переобучение персонала.

Разработка собственного сервиса телемаркетинга с учетом привычных бизнес-процессов внутри компании (*перенести после слова «привычных»*)позволяет решить это. Не придется перестраивать процессы, но все этапы станут автоматизированными, что позволит легко внедрить систему и увеличить эффективность работы.

В результате выполнения данной работы будет разработан и реализован сервис телемаркетинга для платежной системы «Золотая Корона» группы компаний «Центр Финансовых Технологий».

Также выполнение проекта подразумевает выбор инструментов для разработки: языка программирования, фреймворков, базы данных. Будут изучены существующие инструменты, а по результатам их сравнения произведён окончательный выбор.

15 – 20 страниц. Глава будет посвящена более полному погружению в проблему. Нужно сделать плавный переход из введения. Взять проблему, описать варианты готовых решений/как еще можно решить эту проблему. В конце сделаю вывод: как буду решать проблему, почему есть мотивация её решать вообще.

Ставим задачу перед собой: «Будем делать …, вот так …, потому что …, опираясь на обзор существующей системы …».

Зарисовка как было до системы телемаркетинга:

Была уже выстроенная цепочка действий начиная от того, как бизнес собирает как-то данные и т.д. После этого появляется какая-то группа клиентов, которым мы хотим позвонить. Т.е. бизнес составляет Excel-файлы с группами пользователей, далее эти файлики вручную выгружают админы в БД, а в конце супероператоры выгружают опять вручную списки в MightyCall (сервис-звонилка). В итоге, операторы совершают звонки. В этой цепочке очень много ручного труда, который хотелось автоматизировать, кроме этого, на каждом этапе находятся разные люди из разных команд -> сложнее контролировать процесс (что/где/как сломалось)

Беглый анализ:

Как видно из описания, процесс достаточно уникальный, поэтому я не уверен, что есть какое-то коробочное решение, которое можно было бы внедрить сюда, не ломая существующий процесс. Т.е., наверное, есть какие-то системы для звонков, но они вряд ли дадут себя настроить так как уже построен процесс в золотой короне. -> у нас кроме затрат на внедрения появляются затраты на перестройку самого процесса совершения звонков, переобучение персонала и т.д.

В любом случае, нужно поискать какие-то системы со схожим функционалом, чтобы точно дать оценку.

# **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## Что такое телемаркетинг: цели, принципы

Стремительное развитие информационных технологий в финансовой сфере повысило уровень сбора информации о клиентах. В настоящее время для общения с потребителем все чаще используется телемаркетинг.

Телемаркетинг (или telemarketing) представляет собой использование телефона и современных средств связи для ведения коммуникаций и продаж без посредников. Этот метод хоть и один из старшейших, но при этом остается самым эффективным для увеличения объема продаж, изучения целевой аудитории, привлечения новых клиентов и повышения узнаваемости бренда на рынке.

Предпосылки возникновения телемаркетинга появились во второй половине XIX века с изобретением телефона. Первое зарегистрированное использование телефона для продвижения товаров относится к 1909 году. В то время телемаркетинг в основном использовался для продажи подписки на газеты и журналы. Только в 1960-х годах телемаркетинг превратился в полноценную возможность для бизнеса.

В 1965 году AT&T создала первую систему телемаркетинга как способ охвата клиентов. Со временем достижения в области коммуникационных технологий и деловой практики позволили телемаркетингу стать мощным инструментом в мире продаж и маркетинга. Сегодня это один из самых популярных методов работы с клиентами, позволяющий предприятиям устанавливать контакты с большой клиентской базой экономически эффективным способом.

Кроме того, широкое использование баз данных и автоматизированной рассылки звонков привело к улучшению обслуживания клиентов. Это улучшило *(увеличило)* охват клиентов и дало компаниям преимущество в привлечении и удержании клиентов. Телемаркетинг продолжает оставаться важной частью успешной маркетинговой стратегии для предприятий любого размера.

Телемаркетинг делится на два вида:

• Входящий - прием звонков от клиентов;

• Исходящий - обзвон пользователей операторами.

В данной выпускной квалификационной работе будет рассматриваться исходящий вид телемаркетинга.

## Телемаркетинг в финансовой сфере.

Когда дело доходит до финансовых услуг — будь то страхование, управление капиталом или кредитование, — ничто не может заменить индивидуальный подход.

Телемаркетинг предоставляет возможность прямого контакта с клиентом, что приносит (*предоставляет*) компании целый ряд преимуществ в ее продвижении.

Одним из основных плюсов является мгновенная обратная связь и общение с клиентом в режиме реального времени. Никакой другой инструмент маркетинга не обеспечивает такую быструю возможность понимания настроений клиента и оказания влияния на него в пользу бизнеса.

При использовании этой тактики компания может взаимодействовать с потенциальными клиентами и создавать списки новых потенциальных контактов.

Главный плюс использования телемаркетинга для продвижения финансовых продуктов заключается в том, что он позволяет вам мгновенно оценить, насколько интересен клиенту ваш сервис или продукт. Более того, он позволяет:

* **Удерживать клиентов**: телемаркетинг может быть эффективным инструментом для связи с текущими клиентами и информирования их о новых услугах или продуктах, которые могут их заинтересовать, тем самым повышая уровень удержания.
* **Повышать спрос:** финансовые консультанты могут использовать телемаркетинг для привлечения "холодных" потенциальных клиентов или отслеживания "теплых" потенциальных клиентов.
* **Проводить маркетинговые исследования:** можно проводить быстрые опросы или интервью, чтобы понять потребности рынка или собрать отзывы о новых продуктах или услугах.

## Специфика бизнес – процесса (*дефис ставится без пробелов*) внутри рассматриваемой платежной системы «Золотая Корона»

В рамках данной работы разработка система телемаркетинга осуществляется на базе платежной системы «Золотая Корона». Поэтому необходимо учитывать специфику и потребности этого бизнеса. Предлагаю рассмотреть то, как сейчас вручную работает данный процесс.

В самом начале платежная система «Золотая Корона» получает заявки на займы от партеров. В качестве сервисов, из которых идет поток заявок, может быть мобильное приложение «KoronaPay» или другие сервис – партнеры.

Далее бизнес – аналитики на основе этих заявок создают фокус – группы клиентов, которые обычно называют кампаниями, собирают данные из различных сервисов, чтобы определить, каким клиентам и на каких условиях «Золотая Корона» могла бы выдать займы.

Далее идет отсев потенциальных клиентов с помощью полученных данных и уже имеющихся черных списков. Под данными в этом случае воспринимается (*понимается / подразумевается*): история коммуникаций, информация о клиенте, которая была получена во время оформления заявки на займы, и т.д.

Вся собранная информация хранятся в виде Excel – документов, которые передаются «из рук в руки». Далее системные администраторы сгружают их в базы данных для дальнейшей обработки.

В самом конце, супер – операторы снова вручную выгружают списки на обзвон (*для обзвона*) в сервис телефонии – Mighty Call. И только теперь обычные операторы могут совершать звонки по выбранным фокус – группам.

Результаты каждой коммуникации точно так же формируются оператором вручную в Excel – документах, которые отправляются в конце каждой смены и сгружаются системными администраторами в базу данных.

Как видно, в этой цепочке очень много человеческого труда, который можно было бы автоматизировать. Кроме этого, на каждом этапе находятся люди из разных команд, что усложняет процесс контроля и коммуникации между ними. С ростом нагрузки становится сложнее локализовывать проблемы и разбирать инциденты. Поэтому разработка сервиса, или использование готового решения значительно упростило бы сопровождение этого бизнес – процесса.

## Обзор существующих решений

Перед тем как разрабатывать своё решение, необходимо проанализировать уже существующие решения, чтобы понять, можно ли внедрить их с учетом процесса, описанного в п. 1.3.

Сейчас на рынке есть несколько основных решений для построения системы телемаркетинга. Предлагаю рассмотреть их.

### 1С:CRM

<https://1crm.ru/help/telemarketing/>

<https://habr.com/ru/companies/trinion/articles/296406/>

1С CRM Телемаркетинг — это специализированная программа, разработанная для автоматизации процессов телемаркетинга и управления взаимоотношениями с клиентами. Она предназначена для использования в компаниях, которые активно занимаются продажами по телефону. Основные функции программы включают:

Управление базой данных клиентов: программа позволяет хранить информацию о клиентах, включая контактные данные, историю покупок, предпочтения и т.д.

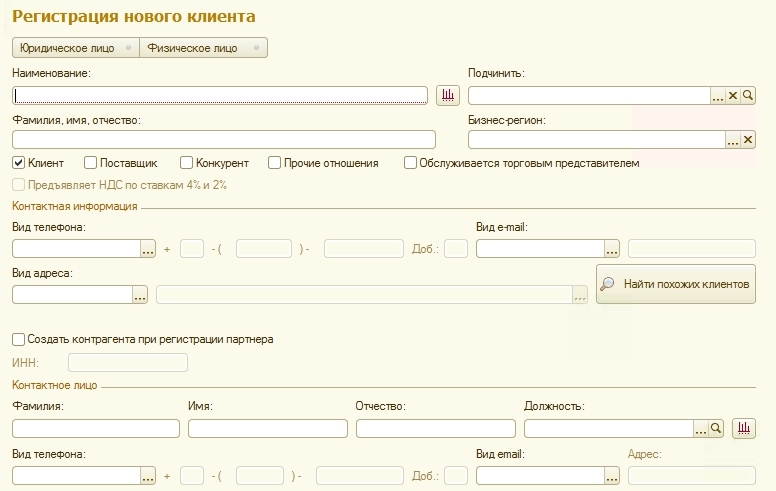


Рисунок Х – Форма регистрации нового клиента

Планирование звонков: с помощью 1С CRM Телемаркетинг можно создавать расписание звонков, назначать ответственных за них сотрудников и отслеживать выполнение задач.

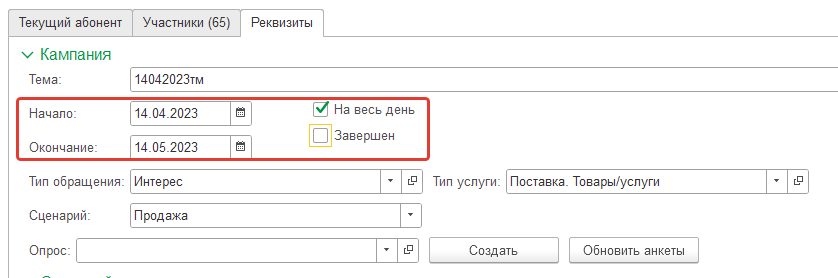


Рисунок Х – Форма планирования звонков

Сама система 1С:CRM не предусматривает связи с телефонией — для этого существует продукт под названием 1С-Рарус: СофтФон. 1С-Рарус: СофтФон обеспечивает интеграцию с офисной АТС (телефонной станцией) и необходим для того, чтобы вы могли фиксировать исходящие и входящие звонки в системе 1С:CRM. Звонить с использованием СофтФона можно, но сама по себе его настройка — дело не тривиальное. 1С-Рарус: СофтФон — это встроенный продукт. СофтФон нельзя дорабатывать, его код закрыт и написан не на языке 1С.

СофтФон позволяет получать информацию о звонящем, делать исходящие звонки — то есть со своей минимальной задачей он справляется. Но все-таки то, что СофтФон — это отдельный продукт, ощущается в процессе работы. При поступлении звонка он не сразу попадает в 1С:CRM, а только при команде пользователя “Передать в 1С”. Конечно, с одной стороны, это хорошо: сторонние звонки не попадут в систему, но есть опасность того, что и не все звонки клиентов попадут в базу.

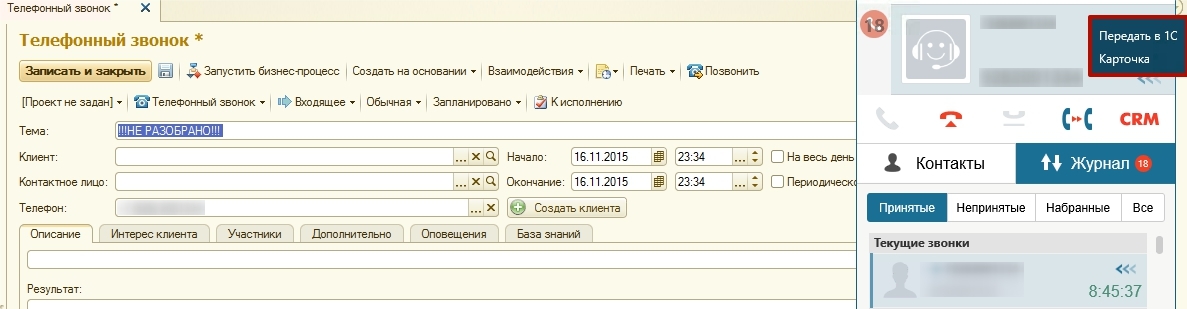


Рисунок Х – Карточка телефонного звонка. Совместная работа «1С Телемаркетинг» и «1С-Рарус: СофтФон»

Карточка телефонного звонка предоставляет широкие возможности, например, такие, как учет трудозатрат, участников события, первичного спроса клиента. Пользоваться этими возможностями или ограничиться полем «Описание», где в свободной форме фиксируется результат звонка, — решать вам.

Телефонные обзвоны: в системе 1С:CRM предусмотрена также возможность регистрации телефонных обзвонов. Эта функция называется «Телемаркетинг». «Телемаркетинг» предоставляет возможность создавать список обзвона вручную, либо по сегментам, либо на основании результатов какого-либо отчета. В документе телемаркетинга можно написать сценарий телефонного звонка в свободной форме. Телемаркетинг дает простейшую аналитику: количество обработанных звонков, количество неуспешных звонков.

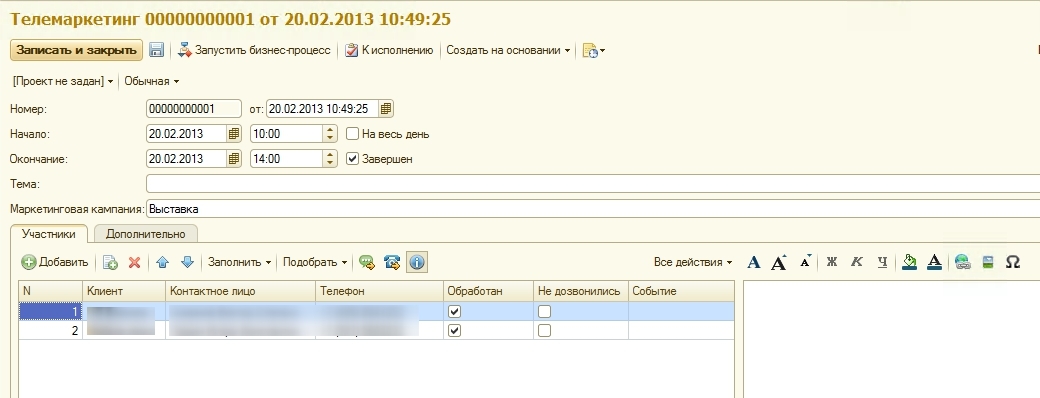


Рисунок Х – Форма создания телефонных обзвонов

Однако в самом документе «Телемаркетинг» отсутствует возможность фиксации результата звонка. Для этого придется создать событие «Телефонный звонок». Также отсутствует возможность создания, например, сделки или задачи на основании звонка, зафиксированного в телемаркетинге.

Таким образом, телемаркетинг представляет из себя, скорее, инструмент контроля сотрудников, осуществляющих звонки. Здесь фиксируется факт того, позвонили клиенту или нет, нежели результат звонка.

В результате, можно выделить плюсы и минусы.

Плюсы:

* Управление отношениями с клиентами и Управление торговлей находятся в ???
* Огромное количество специалистов 1с на рынке, поэтому поддержка не должна вызвать проблем
* Большой функционал, который позволяет автоматизировать процессы телемаркетинга и управления взаимоотношениями с клиентами.

Минусы:

* Сложность доработки
* Громоздкий интерфейс системы и запутанный функционал
* Необходимость обучения сотрудников работе с программой, что может занять время и ресурсы.
* Не всегда точная интеграция с другими системами, что может привести к ошибкам в работе.

Резюмируя, хочется сказать, что функционал на данный момент оставляет желать лучшего. В системе нет стройной, легкой логики работы с клиентом. Прежде чем начать вести взаимодействия с клиентом, пользователь программы должен хорошо продумать, чем он будет пользоваться и как он будет это делать.

### MegaCRM

<https://megacrm.ru/telemarketing>

<https://help.megagroup.ru/kak-rabotat-so-zvonkami-v-megacrm#step--1>

https://wiki.domru.biz/megacrm

MegaCRM — это программа, которая помогает компаниям управлять своими контактами и продажами. Она позволяет создавать и управлять списками контактов, отслеживать историю взаимодействия с клиентами, а также автоматизировать процесс продаж.

Одной из главных особенностей MegaCRM является возможность интеграции с другими программами и сервисами, такими как телефония, электронная почта и социальные сети. Это позволяет пользователям получать доступ к информации о клиентах из разных источников в одном месте.

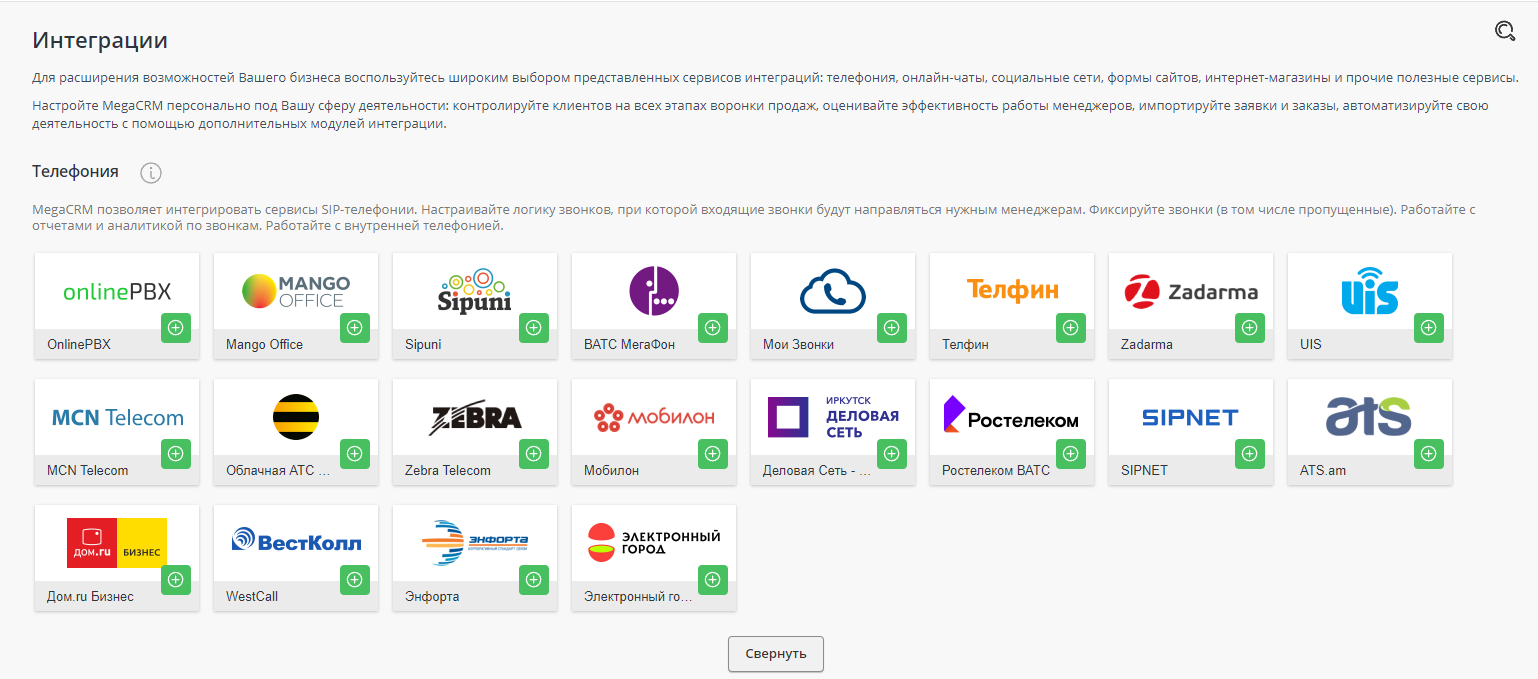
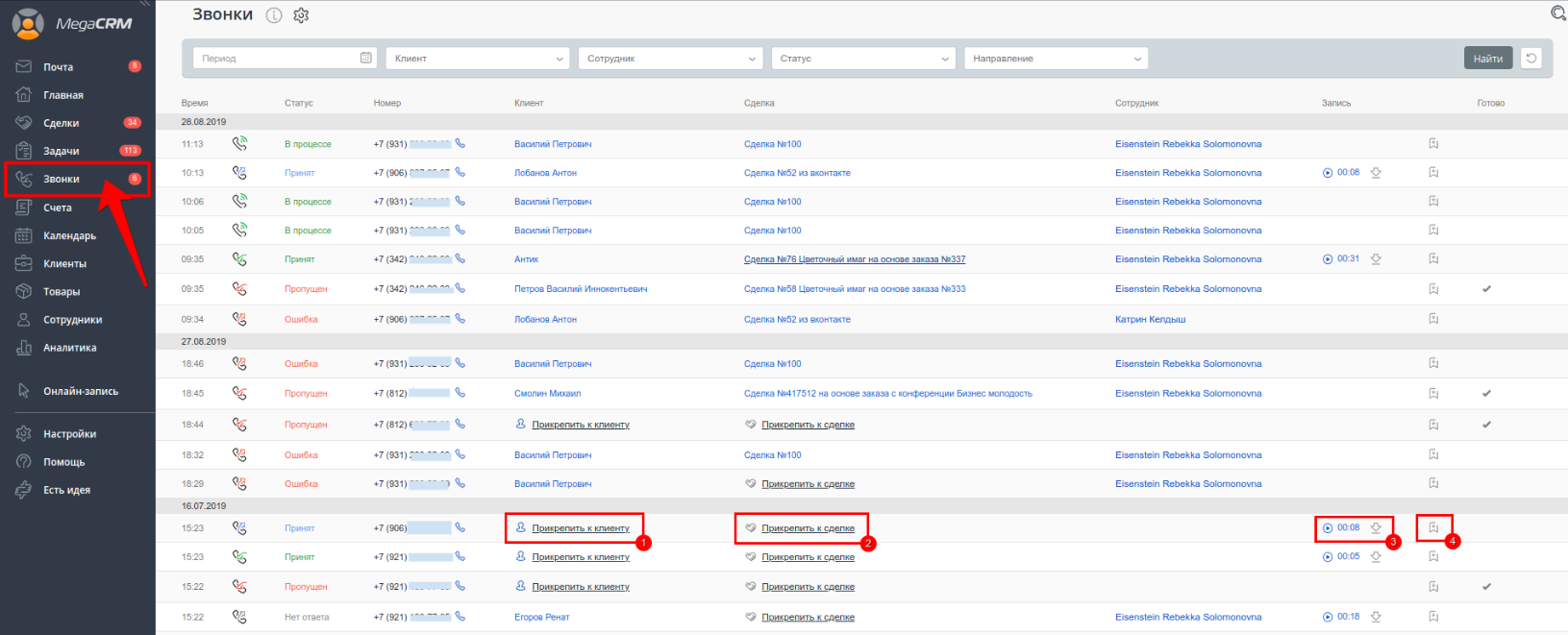


Рисунок Х – Интеграция MegaCRM с телефоний

Программа также имеет функцию автоматического обзвона клиентов, что значительно упрощает процесс продаж. Пользователи могут создавать сценарии звонков и настраивать правила для автоматического обзвона.

Рисунок Х – Раздел обзвона клиентов

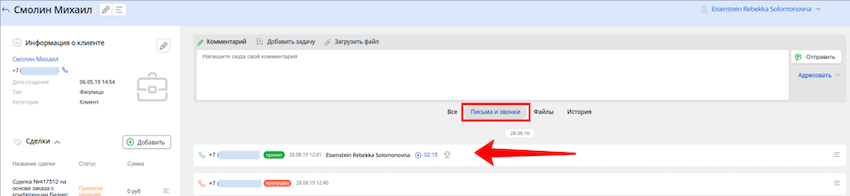
Карточка клиента представляет собой место, где хранится вся информация о клиенте и история коммуникаций с ним. Карточка клиента является ключевым инструментом для эффективного управления отношениями с клиентами и обеспечивает персонализированный подход в работе с каждым клиентом.

Рисунок Х – Карточка клиента

Основные элементы, которые есть в карточке клиента:

* Контактная информация: основные данные о клиенте, такие как имя, контактный номер телефона, электронная почта и адрес.
* История взаимодействия: записи о предыдущих контактах и взаимодействии с клиентом, такие как звонки, встречи, электронные письма, чаты и т.д.
* Сделки и заказы: информация о текущих и прошлых сделках, заказах, счетах и договорах с клиентом.
* Интересы и предпочтения: информация о предпочтениях клиента, его интересах, покупках, предыдущих обращениях и любой другой информации, которая может помочь улучшить работу с клиентом.
* Аналитика и отчетность: статистика по взаимодействию с клиентом, конверсии, динамике продаж и другие ключевые показатели для анализа эффективности работы с клиентом.

Карточка клиента помогает операторам лучше понять потребности и предпочтения клиента, улучшить взаимодействие и обслуживание, а также эффективно вести процессы продаж. Благодаря централизованному хранению информации о клиенте, компания может значительно повысить эффективность и эффективность работы с клиентской базой.

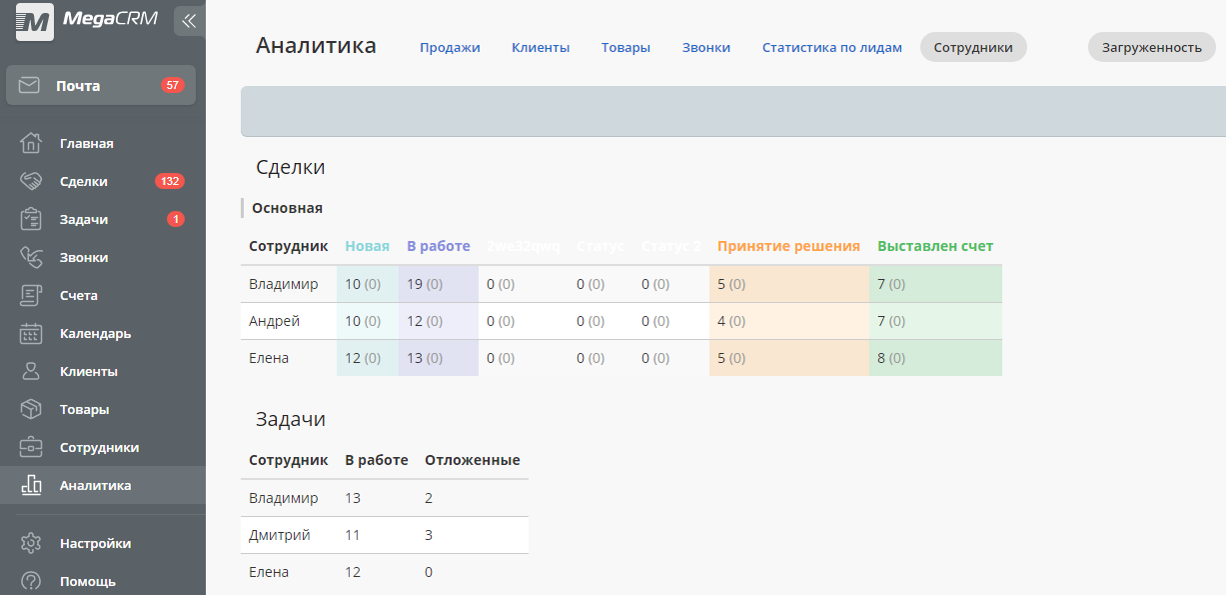
Кроме того, MegaCRM предоставляет возможность анализировать данные о продажах и клиентах. Пользователи могут создавать отчеты по различным параметрам, таким как количество звонков, время разговора и конверсия.

Рисунок Х – Раздел «Аналитика», вкладка «Сотрудники»

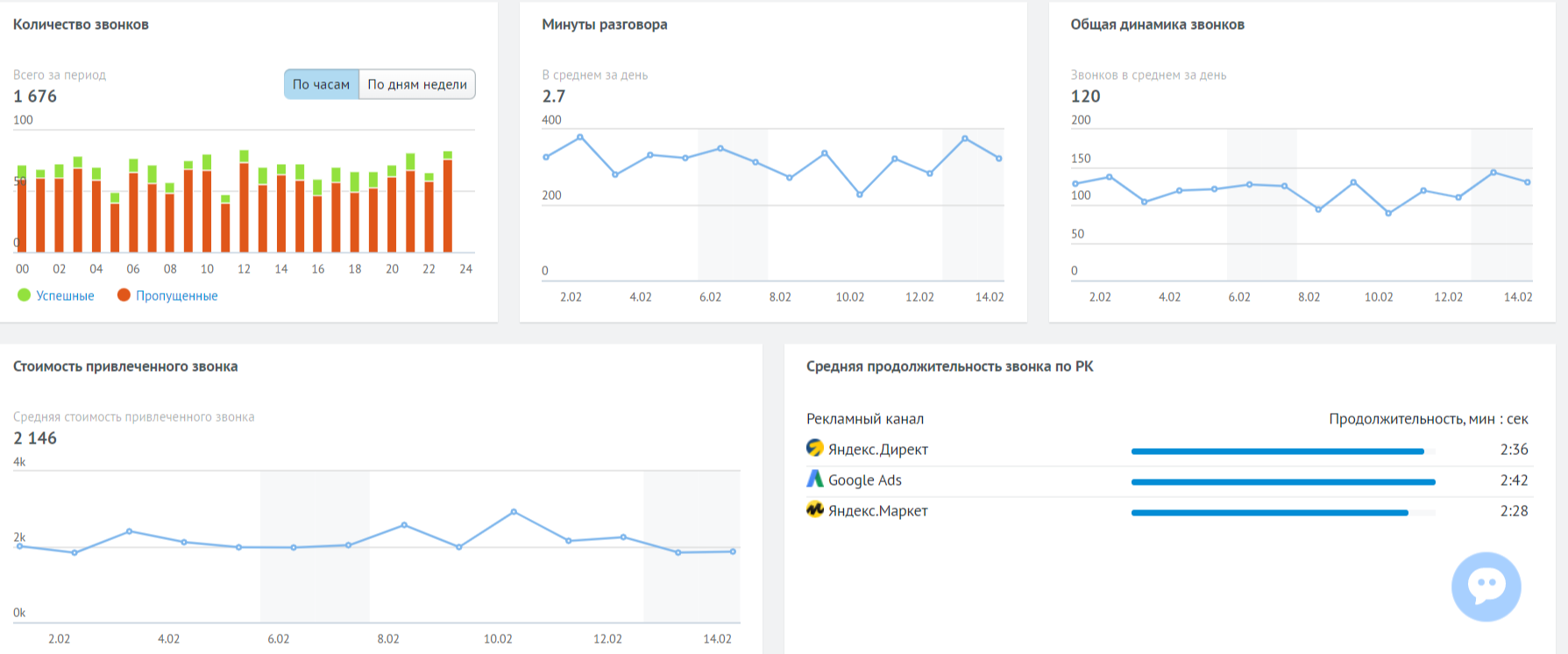


Рисунок Х – Раздел «Аналитика», вкладка «Звонки»

В целом, MegaCRM Телемаркетинг – это мощный инструмент для управления продажами и контактами. Он позволяет автоматизировать процесс продаж и упростить работу с клиентами.

Однако, при использовании платформы MegaCRM, пользователи могут столкнуться с несколькими потенциальными недостатками. В частности, высокая стоимость лицензий и дополнительных функций может быть проблемой для малых компаний с ограниченным бюджетом. Кроме того, сложность настройки и необходимость обучения сотрудников при внедрении новой CRM-системы также могут вызвать определенные трудности. Ограничения в конфигурировании, безопасности, зависимость от стабильного интернет-соединения и возможные проблемы с интеграцией с другими инструментами также могут создать препятствия для эффективного использования MegaCRM в повседневной работе компании.

Плюсы:

1. Интеграция с другими программами и сервисами.
2. Возможность создания отчетов о продажах и аналитики данных.
3. Удобный интерфейс.
4. Возможность управления списками контактов и отслеживания истории взаимодействия с клиентами.

Минусы:

1. Отсутствие автоматического обзвона клиентов
2. Ограниченный функционал по сравнению с другими CRM-системами.
3. Недостаточная гибкость настройки под нужды конкретной компании.

В результате проведенного анализа существующих систем я выявил проблему: среди имеющихся решений – нет системы, которая бы могла бы полностью автоматизировать бизнес – процесс, описанный в п. 2.3., т.к. он уникален в рамках платежной системы «Золотая Корона».

## Формирование требования к разрабатываемой системе

# **ГЛАВА 2. Проектирование системы**

## Выбор инструментальных средств

### Язык программирования Java

Java – популярный язык программирования, используемый уже более 20 лет. Миллионы приложений написаны на Java и используются и по сей день. Он может выступать как платформа сам по себе. Java – быстрый, безопасный и надежный язык, подходящий для широкого спектра приложений, от мобильных до корпоративных, от обработки больших данных до серверных технологий.

Java выделяется среди других языков программирования по нескольким причинам:

* Синтаксис, похожий на С, что делает его понятным для многих программистов, а также облегчает изучение языка новичками.
* Возможности объектно-ориентированного программирования, позволяющие создавать гибкие системы, способные легко взаимодействовать с новыми компонентами без изменений в коде.
* Статическая типизация, которая помогает избежать ошибок на ранних этапах разработки путем явного определения типов переменных.
* Java ориентирована на Java Virtual Machine (JVM), что позволяет запускать Java-код на различных устройствах и операционных системах. Это также обеспечивает обратную совместимость, позволяя разработчикам использовать старый код с новыми версиями JVM без изменений.

### Автоматизированная сборка проектов с помощью Apache Maven

Apache Maven – это инструмент для автоматизации сборки проектов, который широко используется Java-разработчиками, но также имеет плагины для интеграции с другими языками, такими как C/C++, Ruby, Scala, PHP и другие.

Для простых проектов на Java, как, например, "Hello, world!", можно воспользоваться командной строкой для сборки. Однако при разработке более сложного ПО, требующего сторонних библиотек и ресурсов, процесс сборки становится сложнее. Здесь на помощь приходит Maven, созданный для упрощения этой задачи.

Одной из ключевых особенностей Maven является декларативное описание проекта. Это означает, что разработчику не нужно детально настраивать каждый аспект сборки, так как все необходимые параметры уже установлены по умолчанию. Если нужно отклониться от стандартных настроек, изменения требуется (*внести*) в нужные параметры конфигурации.

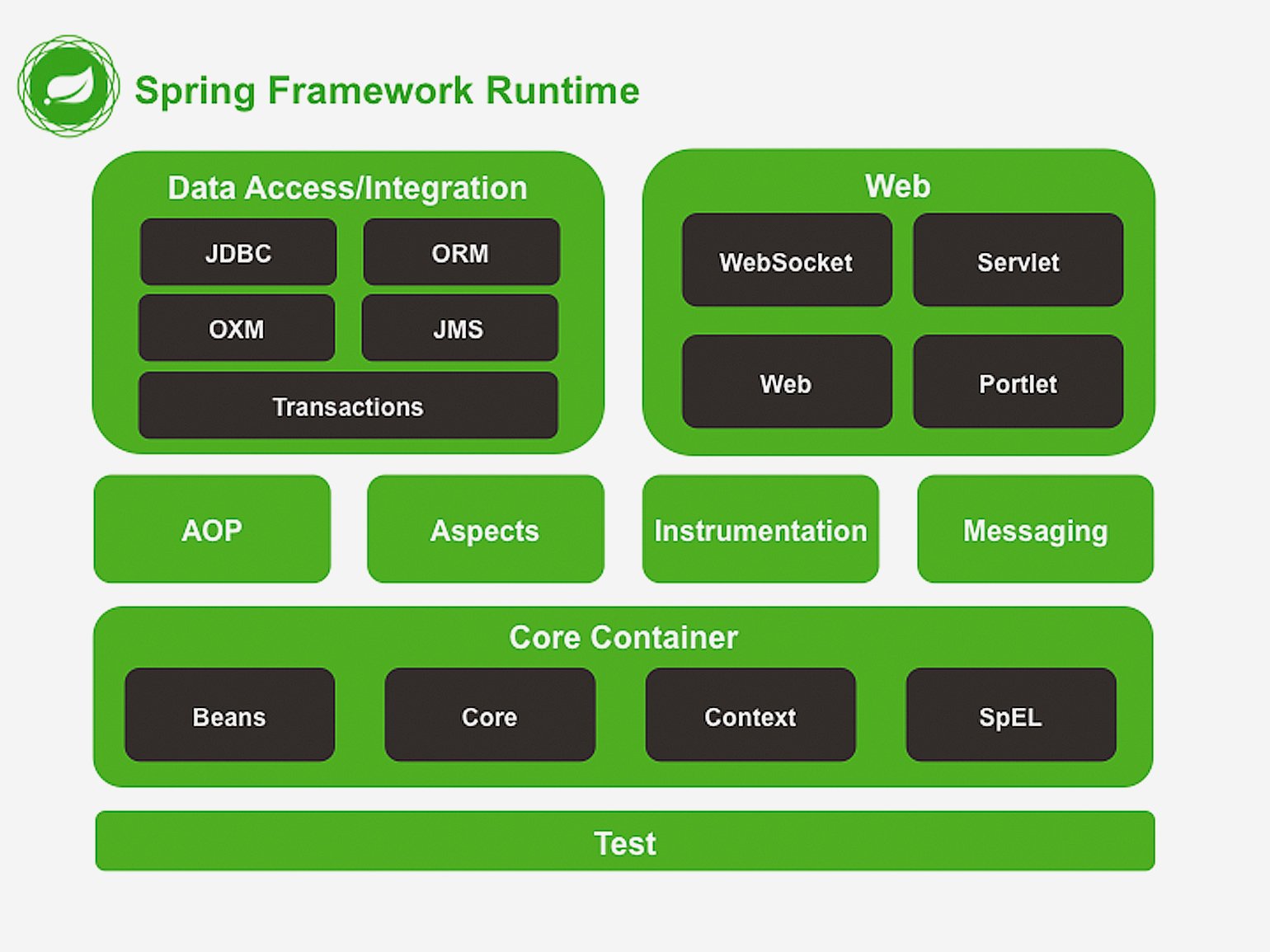
Еще одно преимущество Maven – гибкое управление зависимостями. Он может автоматически загружать сторонние библиотеки в локальный репозиторий, выбирать нужные версии пакетов и обрабатывать транзитивные зависимости.

Также стоит отметить, что Maven независим от операционной системы. В то время как при работе из командной строки параметры могут зависеть от платформы, Maven позволяет не обращать внимания на этот аспект.

В случае необходимости, систему сборки можно настраивать под индивидуальные потребности, используя готовые плагины или создавая собственные, если подходящего решения не нашлось.

### Фреймворк Spring

Spring — это набор взаимосвязанных мини-фреймворков, созданных для работы над разными частями приложения. Подключать их можно по отдельности, в зависимости от задач.



Особенности и преимущества Spring

* Простота, гибкость, универсальность. Фреймворк разработан для поддержки всех типов развертывания, от небольших приложений до крупных корпоративных решений. А для начинающих это означает, что кривая обучения будет крута, но потенциал для применения — огромен.
* Возможность внедрения зависимостей (DI). DI — это шаблон проектирования, который позволяет реализовывать слабую связь, что упрощает тестирование и поддержку кода. Его можно сравнить с личным помощником, который приносит вам то, что вам нужно, именно (*тогда*), когда вам это нужно. Это делает процесс написания кода более гладким и эффективным, что особенно полезно для новичков.
* Контейнер инверсии управления (IoC) Spring предоставляет мощный способ управления компонентами приложения. Это похоже на главный контроллер, который обрабатывает создание и связывание объектов, что помогает избежать большого объёма ручной работы.
* Поддержка аспектно-ориентированного программирования (АОР). Это парадигма программирования, которая стремится увеличить модульность, позволяя разделять сквозные аспекты разработки. Она, как волшебная палочка, которая может добавлять дополнительную функциональность там, где вам это нужно, не затрагивая существующий код. Это позволяет изменять подход к организации кода, что делает его более понятным для начинающих и упрощает навигацию по структуре сложных проектов.
* Интеграция с другими Java-фреймворками и технологиями. Spring хорошо совместим с другими популярными фреймворками, такими как Hibernate, Struts и JUnit, JPA и даже облачными платформами. Вы можете объединить преимущества различных моделей для создания надёжного и эффективного приложения. Возможность работать в гармонии с другими инструментами может значительно улучшить опыт обучения для новичков, предоставив им более широкое понимание Java-экосистемы.
* Spring Boot, проект в составе экосистемы Spring, предоставляет более быстрый и эффективный способ настройки и запуска приложений Spring. Он устраняет много шаблонного кода конфигурации, позволяя вам сконцентрироваться на написании бизнес-логики.

### Объектно-реляционный фреймворк Hibernate

Hibernate Framework — это фреймворк для языка Java, предназначенный для работы с базами данных. Он реализует объектно-реляционную модель — технологию, которая «соединяет» программные сущности и соответствующие записи в базе.

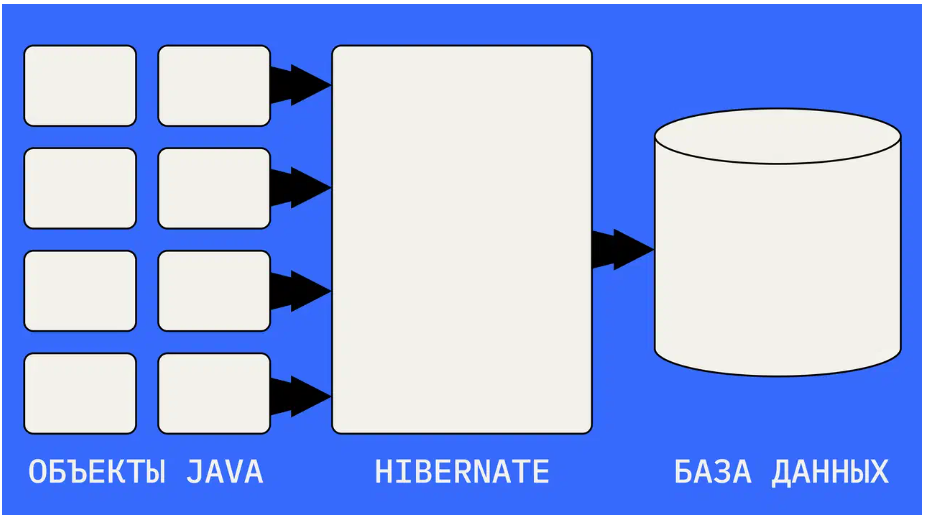
Hibernate Framework – это фреймворк, созданный для работы с базами данных. Он реализует объектно-реляционную модель – технологию, которая связывает программные сущности и соответствующие записи в базе данных.

*(повтор абзаца)*

Объектно-реляционная модель, или ORM (Object-Relational Mapping), позволяет создать программную «виртуальную» базу данных из объектов, описанных с применением принципов ООП.

Если описывать упрощённо, ORM предлагает присвоить столбцу имя переменной объекта кода, а в строку записать значение. Причём типы данных должны определяться автоматически. Всё это требует настройки, и без специальных инструментов приходилось бы проводить все операции вручную. К счастью, готовые инструменты для автоматизации этого процесса существуют.

Hibernate — это инструмент, который как раз работает по ORM. Он работает на особом наборе правил — спецификации JPA 2.1. Она описывает, как именно можно сохранять данные из Java-кода в базы, однако это только теоретические правила. Hibernate превращает теорию в практику и позволяет загружать объекты из кода Java в базы данных напрямую.



Для работы с БД необязательно использовать возможности Hibernate — можно писать весь код стандартными средствами Java. Однако фреймворк значительно облегчает работу:

* ускоряет написание кода;
* создаёт удобную модель отображения уже существующей базы внутри кода;
* позволяет задать стандарт переноса информации из кода в БД.

Hibernate совместим с различными СУБД, такими как MySQL, PostgreSQL, Oracle, Microsoft SQL Server и другими. Благодаря общим принципам работы с базами, он подходит для большинства реляционных баз данных.

Внутри фреймворка используется Hibernate Query Language (HQL), который напоминает SQL, но является (*более простым*) проще и компактным, что делает его удобным для работы с базами данных из Java

Hibernate поддерживает "ленивую загрузку", которая позволяет эффективно работать с базой данных, загружая данные только по необходимости, что повышает производительность приложения

### Система управления базами данных PostgreSQL

PostgreSQL – это объектно-реляционная система управления базами данных (СУБД), наиболее развитая СУБД с открытым исходным кодом. Является альтернативой коммерческим СУБД.

Преимуществами СУБД PostgreSQL являются:

* ***Свободный доступ*** – любой специалист может бесплатно скачать, установить и начать работу с базами данных (БД);
* ***Поддержка различных платформ*** – PostgreSQL подходит для работы на всех популярных операционных системах (ОС): Windows, Linux, macOS. Пользователь получается готовую систему «из коробки» - чтобы установить и использовать программу, не нужны дополнительные инструменты;
* ***Поддержка множества типов данных*** – PostgreSQL поддерживает много разных типов и структур данных, в том числе сетевые адреса, данные в текстовом формате JSON, геометрические данные для координат геопозиций, XML и д.р. Все эти форматы можно хранить и обрабатывать в СУБД. Также, можно создавать собственные типы данных, их называют пользовательскими. Пользовательские типы данных нужны, чтобы упростить работу с БД или установить ограничения;
* ***Работа с большими размерами данных*** *–* размер базы в PostgreSQL не ограничен и зависит от того, сколько свободной памяти есть в месте хранения: на сервере, локальном компьютере или облаке. Максимальный размер таблицы – 32 терабайта. Этого более чем достаточно для хранения данных компании типа Amazon. Одна строка в БД не может превышать 1,6 терабайта, а максимальный размер одной ячейки – 1 гигабайта. В такую ячейку можно записать даже видео;
* ***Соответствие требованиям ACID***

*Атомарность* (от англ. *Atomicity*) – возможность совершать транзакции. Транзакции — это операции или действия, которые происходят полностью или не происходят вообще,

*Согласованность* (от англ. *Consistency*) - В отличие от атомарности, согласованность означает, что транзакция будет выполнена только в том случае, если она не нарушает согласованность данных в базе.

*Изолированность* (от англ. *Isolation*) - означает, что параллельные действия не влияют друг на друга.

*Устойчивость* (от англ. *Durability*) - означает, что уже выполненная транзакция не отменится из-за технических неполадок, например, если отключат свет в серверной.

Это четыре требования для надёжной работы систем, которые обрабатывают данные в режиме реального времени. Если все требования выполняются, данные не будут теряться из-за технических ошибок или сбоев при работе оборудования.

PostgreSQL соответствует всем четырём требованиям ACID и обеспечивает сохранность данных при выполнении транзакций и других работ.

* ***Поддержка сложных запросов*** - В PostgreSQL можно писать собственные функции — пользовательские блоки кода, которые выполняют те или иные действия. Эта возможность есть практически в любых СУБД, но PostgreSQL поддерживает больше языков, чем аналоги. Кроме стандартного SQL, в PostgreSQL можно писать на C и C++, Java, Python, PHP, Lua и Ruby. Он поддерживает V8 — один из движков JavaScript, поэтому JS тоже можно использовать совместно с PgSQL. Реализована поддержка Delphi, Lisp и прочих редких языков. При необходимости можно расширить систему под другие ЯП.

Исходя из всех вышеперечисленных преимуществ, было решено в качестве системы управления базами данных выбрать PostgreSQL.

### Система управления миграциями баз данных Liquibase

Система хранит данные в базе данных. Она может публиковаться на несколько контуров: контур разработки, тестовый контур и «боевой» контур.

В связи с этим, возникает проблема синхронизации схемы БД между контурами и командой разработчиков. Надо как-то передать изменения, которые внесли всем остальным, и при этом не получить конфликты.

Эти проблемы решает система управления миграциями Liquibase. Это своего рода система контроля версий вашей базы данных. При этом она не зависит от используемой базы данных. Liquibase поддерживает множество БД, включая DB2, Apache Derby, MySQL, PostgreSQL, Oracle, Microsoft® SQL Server и прочие.

Существует другие системы управления миграциями: Doctrine 2 migrations, Rails AR migrations, DBDeploy и т.д. Но некоторые из них платформо-зависимые, некоторые не обладают таким широким функционалом, как Liquibase.

Серьезный недостаток этих систем — невозможность применения некоторых изменений без потери данных, например, переименование столбца произойдет как две операции: drop + add, что приведет к потере данных.

***Работа с LiquiBase***

Liquibase — кросс платформенное Java приложение, вы можете скачать JAR файл и запускать его на Windows, Mac или Linux.

Изменения для БД записываются в формате понятном Liquibase, а уже он в свою очередь выполняет запросы к базе данных. Таким образом реализуется независимость от конкретной БД.

***ChangeLog***

Изменения структуры базы данных записываются в файлы, которые называются changeLog. Эти файлы могут быть описаны в разных форматах: XML, YAML, JSON или SQL.

ChangeLog файлы могут быть произвольно включены друг в друга для лучшего управления. Подробнее об этом будет ниже в примерах.

***ChangeSet***

ChangeSet – это аналог коммита в системах контроля версий, таких как Git.

Каждый changeSet имеет составной идентификатор id, author и filename, который должен быть уникальным. ChangeSet может содержать одно или несколько изменений базы данных. Хорошей практикой считается одна команда для одного ChangeSet.

При первом запуске Liquibase создает две технические таблицы:

databasechangelog – Содержит список изменений схемы БД. Туда записываются уже выполненные changeSet.

databasechangelock – Используется для блокировки на время работы, чтобы гарантировать одновременную работу только одного экземпляра Liquibase.

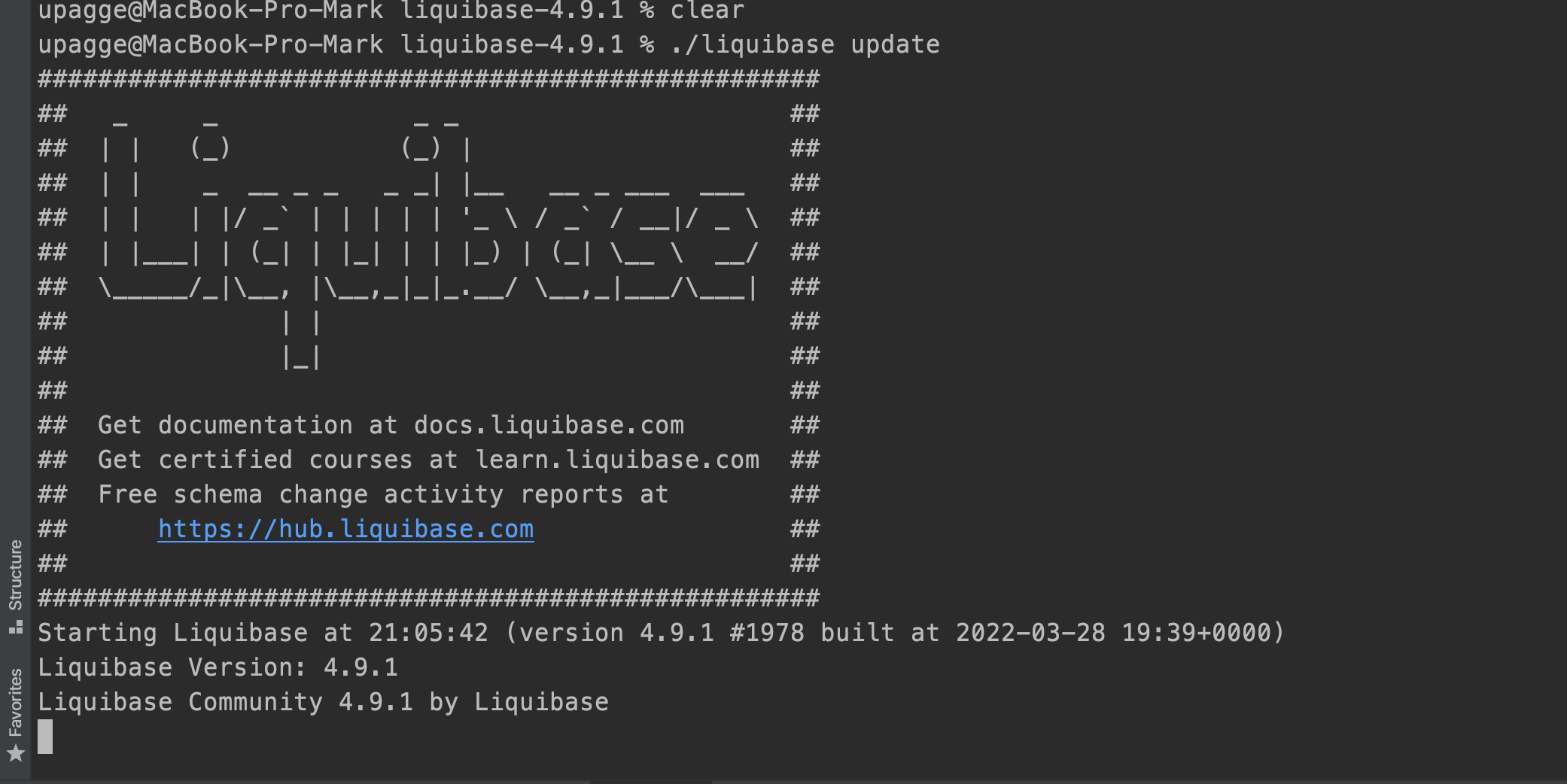
***Блокировки***

Если несколько экземпляров Liquibase будут выполняться одновременно с одной и той же базой данных, вы получите конфликты. Это может произойти, если несколько разработчиков используют один и тот же экземпляр базы данных, или если в кластере несколько инстансовы, которые автоматически запускают Liquibase при запуске.

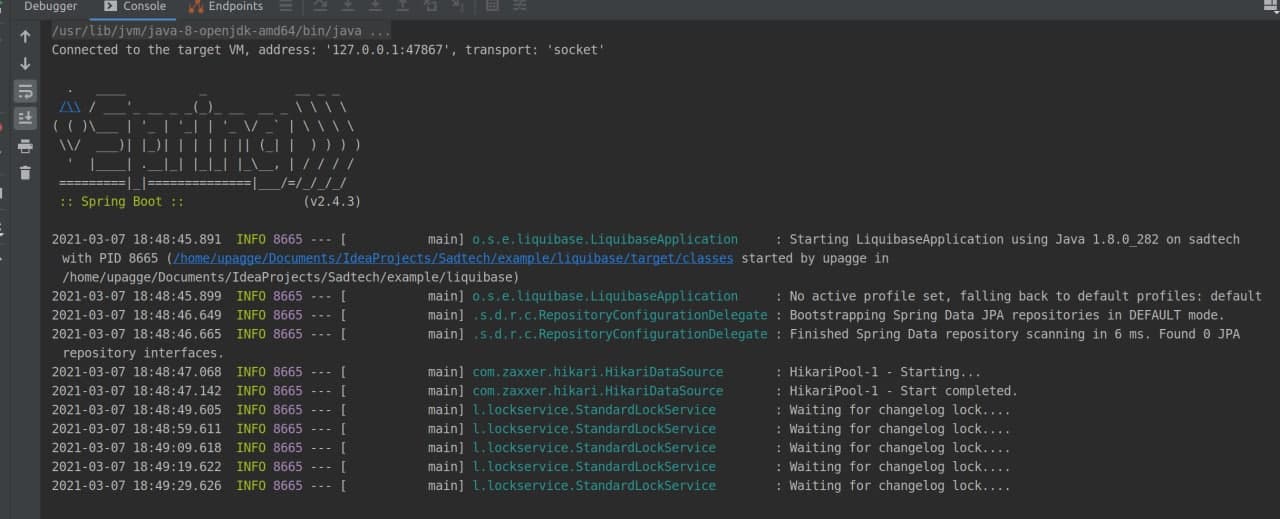
Для защиты от таких ситуаций Liquibase создает таблицу databasechangelock, в которой есть boolean поле locked. При запуске Liquibase проверяет его состояние, и если оно true, то ожидает смены на false.

Экстренно остановив выполнение программы в самом начале, может сложиться ситуация при котором Liquibase успеет поставить флаг, но не поменяет его на false.

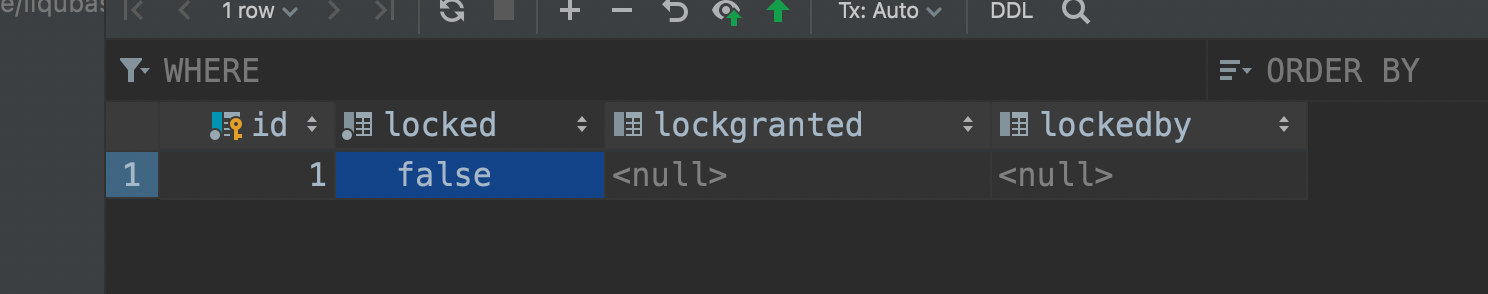
Например, при запуске в виде исполняемого файла все просто завсинет.



В логах Spring приложения это будет более наглядно:



Чтобы исправить эту проблему, в таблице databasechangelock измените поле locked на false.

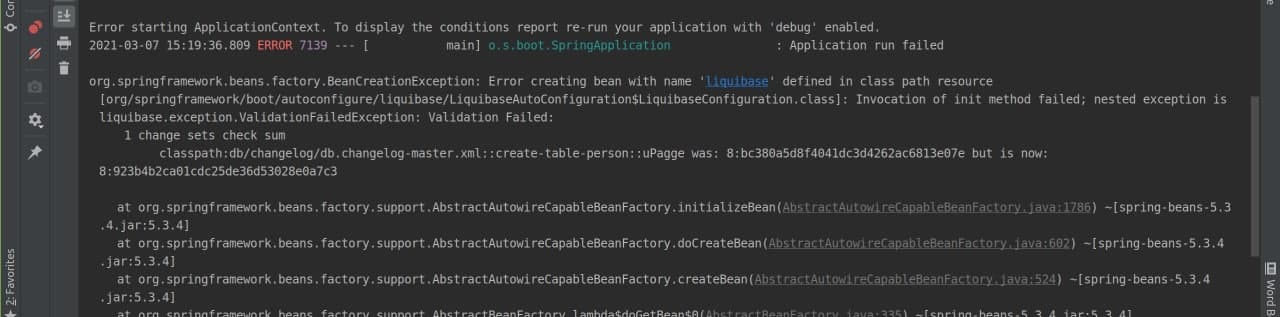


***Контрольная сумма***

Далее Liquibase читает главный changeLog, проверяя, какие изменения уже были приняты, а какие надо выполнить.

После выполнения changeSet в таблицу databasechangelog со всем прочим записывается MD5 хэш changeSet. Хэш высчитывается на основе нормализованного содержимого XML.

При следующем запуске Liquibase будет сверять вновь рассчитанные хэш суммы, со значениями в его таблице. Если вы изменили уже выполненный changeSet, то хэш сумма будет отличаться, и приложение упадет с ошибкой при старте.



## Взаимодействие клиент – серверной архитектуры

REST API — это архитектурный подход, который устанавливает ограничения для API: как они должны быть устроены и какие функции поддерживать. Это позволяет стандартизировать работу программных интерфейсов, сделать их более удобными и производительными.

Слово REST — акроним от Representational State Transfer, что переводится на русский как «передача состояния представления», «передача репрезентативного состояния» или «передача „самоописываемого“ состояния».

В отличие от, например, SOAP API, REST API — не протокол, а простой список рекомендаций, которым можно следовать или не следовать. Поэтому у него нет собственных методов. С другой стороны, его автор Рой Филдинг создал ещё и протокол HTTP, так что они очень хорошо сочетаются, и REST обычно используют в связке с HTTP. Хотя новичкам нужно помнить: REST — это не только HTTP, а HTTP — не только REST.

## Проектирование сервисной архитектуры разрабатываемой системы

### Предметно-ориентированное проектирование сервисов

### Интеграции со сторонними сервисами

### Проектирование основного контура Телемаркетинга

#### Автоматизированное рабочее места оператора

#### Сервис обработки поступающих заявок

### Проектирование домена «Займы»

## Проектирование структуры хранения данных

# **Реализация системы**

## Создание структуры хранения данных

## Программирование сервисов системы

# ГЛАВА 3

+- 15 страниц.

Описываю то что получилось. Т.к. разрабатываю программу, то в конце нужно написать некое руководство пользователя: картинки экранов, некоторые пользовательские сценарии.

«Делаем …, видим …»

# ЗАКЛЮЧЕНИЕ

Возврат к введению, передо мной стояла задача …, я её выполнил успешно …

Чего добились, куда двигаться дальше…

# СПИСОК ЛИТЕРАТУРЫ

https://www.unisender.com/ru/glossary/chto-takoe-telemarketing/#anchor-6

<https://1crm.ru/help/telemarketing/>

https://megacrm.ru/telemarketing

Лучше дописывать по ходу работы, т.к. так проще будет аннотировать. 15-20 источников. Если нет ссылки на источник, то он тут не нужен.

«Доп. инф. по теме можно прочитать …»

# ПРИЛОЖЕНИЯ

Буду использовать, если не хватает объема. В сумме работа должна быть +- 70 страниц

Обычно сюда выносятся кусочки кода, если они больше страницы. Можно вынести теоретические информацию, но лучше внешний ресурс.