

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

Кафедра «Прикладная математика и информатика»  
(наименование)

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

Разработка программного обеспечения  
(направленность (профиль)/специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка программного обеспечения для анализа и  
прогнозирования поведения потребителей»

Обучающийся

А. А. Голубенко  
(Инициалы Фамилия)

\_\_\_\_\_  
(личная подпись)

Руководитель

к.п.н., доцент, О. Ю. Копша  
(ученая степень, ученое звание, Инициалы Фамилия)

Тольятти 2025

## **Аннотация**

Тема бакалаврской работы: «Разработка программного обеспечения для анализа и прогнозирования поведения потребителей».

Ключевые слова: бизнес-процесс, анализ, автоматизация, клиентская база, поведение потребителей, сегментация.

Актуальность темы обусловлена растущим спросом на инструменты анализа и прогнозирования поведения потребителей, их способностью сокращать трудоёмкие задачи обработки данных, повышать эффективность бизнес-процессов и укреплять конкурентоспособность компании.

Целью работы является разработка программного обеспечения для оптимизации процессов анализа и прогнозирования поведения потребителей.

Объект исследования – процессы анализа и прогнозирования поведения потребителей компании. Предмет исследования – автоматизация процессов хранения, обработки, анализа данных о клиентах и прогнозирования их поведения.

Работа состоит из введения, трёх глав, заключения и списка используемой литературы.

Введение раскрывает актуальность темы, описывает цели и задачи работы, объект и предмет исследования. В первой главе выполнен анализ бизнес-процессов организации и их оптимизация с использованием разрабатываемого ПО. Вторая глава посвящена логическому проектированию АИС для анализа и прогнозирования поведения потребителей. В третьей главе выполнено физическое проектирование и тестирование АИС. В заключении приводятся результаты выполнения работы, практическая полезность и теоретическая значимость.

Работа включает 26 рисунков, 16 таблиц и 24 источника. Общий объем бакалаврской работы составляет 75 страницы.

## Оглавление

|                                                                                                                       |    |
|-----------------------------------------------------------------------------------------------------------------------|----|
| Введение.....                                                                                                         | 5  |
| Глава 1 Функциональное моделирование предметной области.....                                                          | 7  |
| 1.1 Характеристика деятельности объекта автоматизации.....                                                            | 7  |
| 1.2 Концептуальное моделирование предметной области .....                                                             | 11 |
| 1.2.1 Выбор технологии концептуального моделирования.....                                                             | 11 |
| 1.2.2 Разработка и анализ модели бизнес-процесса «AS IS» .....                                                        | 13 |
| 1.2.3 Обоснование необходимости автоматизированного варианта<br>решения и формирование требований .....               | 19 |
| 1.3 Анализ существующих разработок на предмет соответствия<br>сформулированным требованиям.....                       | 22 |
| 1.3.1 Определение критериев анализа .....                                                                             | 22 |
| 1.3.2 Сравнительная характеристика существующих разработок.....                                                       | 22 |
| 1.4 Постановка задачи на разработку проекта создания автоматизированной<br>информационной системы .....               | 25 |
| 1.5 Разработка модели бизнес-процесса «TO BE».....                                                                    | 26 |
| Глава 2 Логическое проектирование автоматизированной информационной<br>системы.....                                   | 32 |
| 2.1 Выбор технологии логического моделирования автоматизированной<br>информационной системы .....                     | 32 |
| 2.2 Разработка логической модели автоматизированной информационной<br>системы.....                                    | 33 |
| 2.3 Проектирование концептуальной и логической моделей базы данных<br>автоматизированной информационной системы ..... | 36 |
| 2.4 Требования к аппаратно-программному обеспечению<br>автоматизированной информационной системы.....                 | 38 |

|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| Глава 3 Физическое проектирование автоматизированной информационной системы.....                               | 41 |
| 3.1 Выбор архитектуры автоматизированной информационной системы ..                                             | 41 |
| 3.2 Выбор технологии разработки программного обеспечения автоматизированной информационной системы.....        | 44 |
| 3.3 Выбор СУБД автоматизированной информационной системы .....                                                 | 45 |
| 3.4 Разработка физической модели данных автоматизированной информационной системы .....                        | 47 |
| 3.5 Разработка программного обеспечения автоматизированной информационной системы .....                        | 51 |
| 3.6 Описание функциональности автоматизированной информационной системы.....                                   | 56 |
| 3.7 Тестирование программного проекта .....                                                                    | 61 |
| 3.8 Оценка и обоснование экономической эффективности разработки автоматизированной информационной системы..... | 62 |
| Заключение .....                                                                                               | 69 |
| Список используемой литературы и используемых источников.....                                                  | 72 |
| Приложение А SQL-запросы для базы данных и хранимые процедуры .....                                            | 76 |
| Приложение Б Тестовые сценарии.....                                                                            | 84 |
| Приложение В Матрица трассируемости требований .....                                                           | 88 |
| Приложение Г Фрагменты кода.....                                                                               | 89 |

## **Введение**

Тема бакалаврской работы «Разработка программного обеспечения для анализа и прогнозирования поведения потребителей» предложена кафедрой. Актуальность темы обусловлена растущей потребностью компаний в понимании предпочтений клиентов. В условиях современного рынка анализ и прогнозирование поведения потребителей становятся важными инструментами, позволяющими оптимизировать процессы продаж, улучшать обслуживание клиентов повышая их лояльность, тем самым увеличивая конкурентоспособность. Разработка специального программного обеспечения (ПО) для этих целей становится важным шагом для повышения эффективности бизнес-процессов, увеличивая точность прогнозов и скорость принятия решений. Это делает тему особенно актуальной для бизнеса и IT-сферы.

Объектом исследования являются процессы анализа и прогнозирования поведения потребителей IT-компании ООО «ВорлдИнтерТех РУС».

Предметом исследования выступает автоматизация процесса хранения, обработки, анализа данных о клиентах и использование этой информации для прогнозирования их поведения.

Целью бакалаврской работы является разработка программного решения для анализа и прогнозирования поведения потребителей.

Задачи, решаемые в ходе выполнения работы:

- изучить предметную область, существующие методы и программные решения для анализа поведения потребителей;
- проанализировать текущие бизнес-процессы в компании с целью выявления неэффективных мест предметной области и предложить усовершенствованную модель бизнес-процессов;
- спроектировать архитектуру АИС и определить технологии разработки;
- определить СУБД и спроектировать базу данных АИС;

- разработать программное обеспечение и выполнить тестирование;
- оценить экономическую эффективность разработки.

Для функционального моделирования бизнес-процессов была применена нотация IDEF0, основанная на методологии структурного анализа SADT, с использованием графической среды Ramus. Логическое моделирование АИС выполнялось с использованием объектно-ориентированного языка моделирования UML и CASE-средства StarUML.

Бакалаврская работа состоит из введения, трех глав и заключения.

В первой главе представлен анализ организационной структуры и бизнес-процессов в организации и предложена усовершенствованная модель бизнес-процессов с использованием разрабатываемого ПО.

Во второй главе выполнено логическое проектирование АИС.

В третьей главе осуществлено физическое проектирование АИС, выбрана архитектура, принята СУБД и спроектирована физическая модель данных, выполнена разработка ПО, рассмотрен функционал разработанной АИС и ее тестирование.

Практическая полезность разработки ПО для анализа и прогнозирования поведения потребителей заключается в том, что оно позволяет автоматизировать работу менеджера по работе с клиентами и менеджера по рекламе, помогает значительно сократить время на анализ данных и принятие решений, обеспечивает более точное выявление целевой аудитории и позволяет компании оперативно реагировать на изменения рыночной ситуации.

## **Глава 1    Функциональное моделирование предметной области**

### **1.1    Характеристика деятельности объекта автоматизации**

IT-компания ООО “ВорлдИнтерТех РУС”, специализируется на разработке облачных сервисов, цифровизации бизнеса и интернет маркетинге. Основной вид деятельности по ОКВЭД – разработка компьютерного программного обеспечения (62.01).

Причинами появления и развития IT-компаний является потребность современного рынка в инновационных цифровых решениях для оптимизации бизнеса, они помогают эффективно управлять информацией, обеспечивая ее безопасность и доступность. Компании нуждаются в развитии онлайн присутствия для привлечения новых клиентов и укрепления позиций на рынке, автоматизации бизнес-процессов. Совокупность этих факторов является основой для существования и успешной деятельности.

Основной целью компании “ВорлдИнтерТех РУС” является помощь различным компаниям в цифровой трансформации и улучшении их процессов с помощью инновационных технологий. Руководит компанией Генеральный директор, он играет ключевую роль, его полномочия:

- управление деятельностью и принятие стратегических решений по развитию компании и созданию новых возможностей для роста;
- руководство персоналом;
- утверждение бюджета и финансовых отчетов компании.

Исполнительный директор играет ключевую роль в управлении ежедневной деятельностью компании, его полномочия:

- оперативное управление;
- создание эффективной команды;
- управление бюджетом в рамках утвержденных стратегий.

Главный бухгалтер является ключевым финансовым специалистом в компании и в его полномочия входит:

- финансовое управление;
- учет, отчетность и аудит.

Менеджер проектов играет ключевую роль в успешной реализации проектов. В его полномочия входит:

- планирование проектов, определение целей и требований;
- распределение и контроль выполнения задач и соблюдения сроков;
- управление бюджетом проекта;
- взаимодействие с заказчиком, отчетность о прогрессе проекта;
- организация контроля качества разработки, проверка соответствия окончательного продукта требованиям заказчика.

Руководитель команды разработчиков – Тимлид играет важную роль в формировании рабочего процесса и осуществляет:

- руководство командой, распределение задач и контроль за их выполнением, техническое руководство;
- представление команды разработчиков на совещаниях.

Менеджер по рекламе играет важную роль в продвижении продуктов компании и привлечении клиентов. В его полномочия входит:

- разработка и реализация рекламных кампаний для привлечения клиентов и продвижения услуг, мониторинг эффективности;
- определение целевой аудитории и прогнозирование ее поведения;
- анализ внешней среды для определения успешных стратегий.

Менеджер по работе с клиентами играет важную роль в удовлетворении потребностей клиентов и улучшении имиджа компании. Его полномочия:

- предоставление консультаций и информации по продуктам компании;
- решение проблем и вопросов клиентов, управление клиентским портфелем;
- сбор обратной связи от клиентов о продуктах и сервисе компании;
- ведение переговоров и заключение сделок с клиентами;
- достижение установленных целей по объему продаж;

– сбор и анализ данных о поведении и предпочтениях клиентов.

Организационная структура компании представлена на рисунке 1.

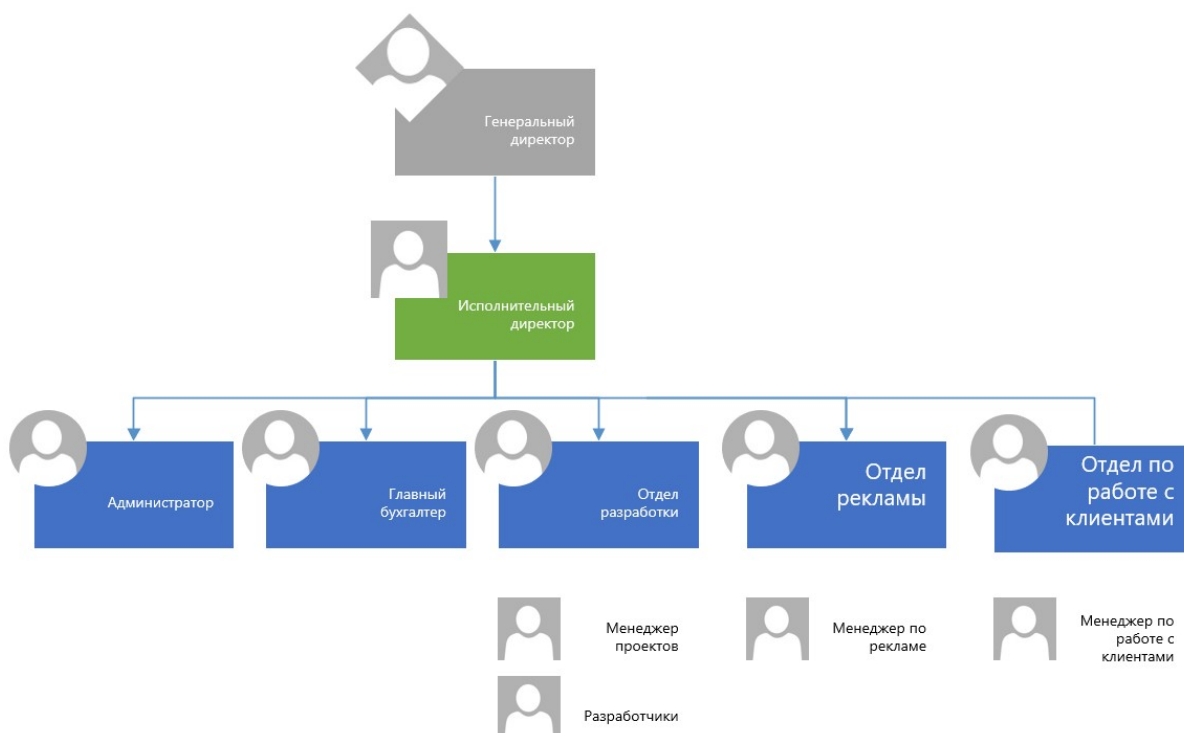


Рисунок 1 – Структурная схема ООО “ВорлдИнтерТех РУС”

В качестве предметной области выступает деятельность компании по анализу и прогнозированию поведения потребителей. Понимание поведения потребителей является важной составляющей для разработки успешной маркетинговой и рекламной стратегии, продаж и обслуживания клиентов. Как следует из Организационной структуры компании эти функции выполняются Отделом Рекламы – Менеджером по рекламе и Отделом по работе с клиентами – Менеджером по работе с клиентами.

Анализ и прогнозирование поведения потребителей включает:

Анализ клиентской базы. Классификация клиентов, основанная на их активности и значимости для компании. Определение ключевых и наиболее активных клиентов, принесших наибольший доход компании, средних клиентов, и клиентов, чьи покупки могут быть нестабильными. Это позволяет

выделить группы клиентов с разным уровнем важности для компании и разработать стратегии продаж и обслуживания для каждой группы клиентов.

Замер индекса лояльности – NPS (Net Promoter Score). Выявляется в ходе опросов на основе шкалы от 0 до 10 как клиенты относятся к компании и насколько они готовы ее рекомендовать. На основе NPS клиенты делятся на группы. NPS помогает выявлять слабые места в обслуживании или продукте и понимать, где необходимо внести улучшения. Высокий NPS свидетельствует о довольстве клиентов, что может привлечь новых потребителей через рекомендации [15]. На основе обратной связи от клиентов разрабатываются стратегии по улучшению обслуживания и удовлетворенности.

Определение целевых сегментов потребителей. Выделяются группы потребителей, которые являются наиболее привлекательными для компании, они имеют сходные потребности, характеристики, которые компания может удовлетворить с наибольшим успехом. Сегментация осуществляется по покупательскому поведению, уровню использования продукта, лояльности. Определение целевых сегментов позволяет разрабатывать продукты и услуги, соответствующие потребностям целевых сегментов, направлять рекламу и промоакции на определенные сегменты для увеличения эффективности [14].

Прогнозирование поведения потребителей. Выполняется анализ связанных покупок с целью выявления шаблонов, что позволяет узнать, какие продукты чаще всего покупаются вместе. На основе этих данных формируются коммерческие предложения для будущих клиентов. Прогнозирование объемов продаж выполняется на основе предыдущих данных о продажах в разные периоды и аналитических индикаторов прогнозирования.

В своей деятельности Менеджер по работе с клиентами сотрудничает с Менеджером по рекламе и Исполнительным директором:

- представляет анализ Клиентской базы и удовлетворенности клиентов в виде отчетов для определения и разработки стратегии компании;

- участвует в обсуждении результатов анализа для принятия мер по улучшению обслуживания и удовлетворенности клиентов.

Менеджер по рекламе сотрудничает с Исполнительным директором:

- представляет данные в виде отчетов о целевых сегментах, прогнозах поведения потребителей и результатах рекламных кампаний;
- принимает участие в обсуждении бюджета рекламных кампаний.

Приведенный выше анализ деятельности и организационной структуры позволяет понять как компания работает с клиентами и взаимодействует внутри своей структуры. На основе этих материалов переходим к разработке концептуальной модели, которая поможет исследовать существующие процессы в области анализа и прогнозирования поведения потребителей и предложить решения для повышения их эффективности.

## **1.2 Концептуальное моделирование предметной области**

### **1.2.1 Выбор технологии концептуального моделирования**

Концептуальное моделирование играет важную роль в разработке информационных систем. «Бизнес – это совокупность взаимосвязанных и взаимодействующих процессов. А раз так, то дело за малым: аккуратно описать каждый процесс и изучить взаимосвязи процессов.» [8, с. 6]. Для этих целей широко применяются нотации, такие как BPMN, DFD и IDEF0, позволяющие наглядно описать бизнес-процессы, потоки информации и взаимосвязи.

Нотация IDEF0 (Integration Definition for Function Modeling) основана на методологии структурного анализа SADT (Structured Analysis and Design Technique). Основу методологии IDEF0 составляет графический язык, позволяющий создать наглядную функциональную модель, отображающую структуру и функции системы, потоки информации и материальных объектов, связывающие эти функции. Позволяет декомпозировать процессы на подпроцессы и строить иерархические модели бизнес-процессов [2, 6].

BPMN (Business Process Model and Notation) представляет собой графическую нотацию для отображения бизнес-процессов при моделировании потоков работ [6]. BPMN-схемы наглядно демонстрируют последовательность действий и перемещение информационных потоков, необходимых для выполнения процесса, дают возможность отметить исполнителей и описать условия выполнения каждого шага в процессе. Каждый процесс может быть декомпозирован на более низкие уровни.

DFD (Data Flow Diagram) представляет собой графическую нотацию для моделирования потоков данных в информационной системе, для представления процессов обработки информации [6]. Используется для идентификации и анализа входных и выходных данных, процессов и хранилищ данных. Позволяет визуально показать процессы с точки зрения движения данных и преобразования данных. Модель DFD – иерархическая модель, каждый процесс может быть подвергнут декомпозиции.

Результаты сравнительного анализа представлены в таблице 1.

Таблица 1 – Сравнительные характеристики нотаций

| Критерии                                       | IDEF0          | BPMN       | DFD            |
|------------------------------------------------|----------------|------------|----------------|
| Подход к проектированию                        | функциональный | процессный | функциональный |
| Возможность представления: процесса, операции; | +              | +          | +              |
| входных и выходных ресурсов;                   | +              | –          | +              |
| состава процесса (декомпозиции);               | +              | +          | +              |
| средств выполнения процесса;                   | +              | +          | +              |
| потока ресурсов;                               | +              | -          | +              |
| асинхронных и синхронных процессов;            | –              | +          | -              |
| документооборота                               | +              | +          | +              |
| элементов организационной структуры            | +              | +          | –              |

Нотация IDEF0 наиболее подходит для моделирования бизнес-процессов:

- IDEF0 предоставляет структурированный метод анализа бизнес-процессов, позволяя четко определить функции, описать потоки ресурсов и взаимосвязи между ними;
- IDEF0 позволяет декомпозировать сложные процессы на более мелкие функциональные блоки;
- в 2001 г. IDEF0 был использован в нормативном документе РФ по функциональному моделированию в статусе «Рекомендации по стандартизации» [3].

Методологии IDEF послужили основой для создания CASE-средств (computer-aided software engineering), в том числе пакета Ramus – графической среды для проектирования и моделирования различных систем [22]. Ramus дает возможность наглядно представить любую деятельность или структуру в виде модели и поддерживает нотации IDEF0, DFD. Программа визуализирует бизнес-процессы и позволяет создавать многоуровневые иерархические модели. Обладает удобным интерфейсом, доступной методической и справочной документацией [6].

### **1.2.2 Разработка и анализ модели бизнес-процесса «AS IS»**

Нормативная документация дает определение «бизнес-процесс: частично упорядоченный, часто вложенный, набор видов деятельности предприятия, которые следует исполнять для получения желаемого результата, стремясь достичь заданной цели предприятия или части предприятия» [1, с. 2].

Для выявления неэффективных мест деятельности компании необходимо построить модель «AS IS» – модель бизнес-процесса на основе существующей организации работ. Модель даст представление о работе системы в целом, информационных и ресурсных потоках, понимании взаимосвязей [6].

Как было сказано выше деятельность компании “ВорлдИнтерТех РУС” заключается в помощи различным компаниям в цифровой трансформации и улучшении их процессов с помощью IT технологий – предоставление IT услуг.

Построение модели «AS IS» начинается с построения контекстной диаграммы верхнего уровня, на которой объект представлен единственным блоком с граничными стрелками A0. Единственный блок на диаграмме представляет весь объект [2]. В нашем случае это деятельность компании по предоставлению IT услуг (рисунок 2). Стрелки отображают связи объекта с окружающей средой. Входные стрелки, входящие с левой стороной блока, обозначают данные и материальные объекты, которые преобразуются в процессе деятельности, это потребность на рынке в IT услугах; финансовый поток от Клиентов и информация о внешней среде. Выходные стрелки, выходящие с правой стороной блока, обозначают результаты деятельности, это оказанные IT услуги; отчеты и аналитика; чДП; рекламные акции. Входящие стрелки с нижней стороны блока обозначают ресурсы, это оборудование, технологии, инструменты и персонал.

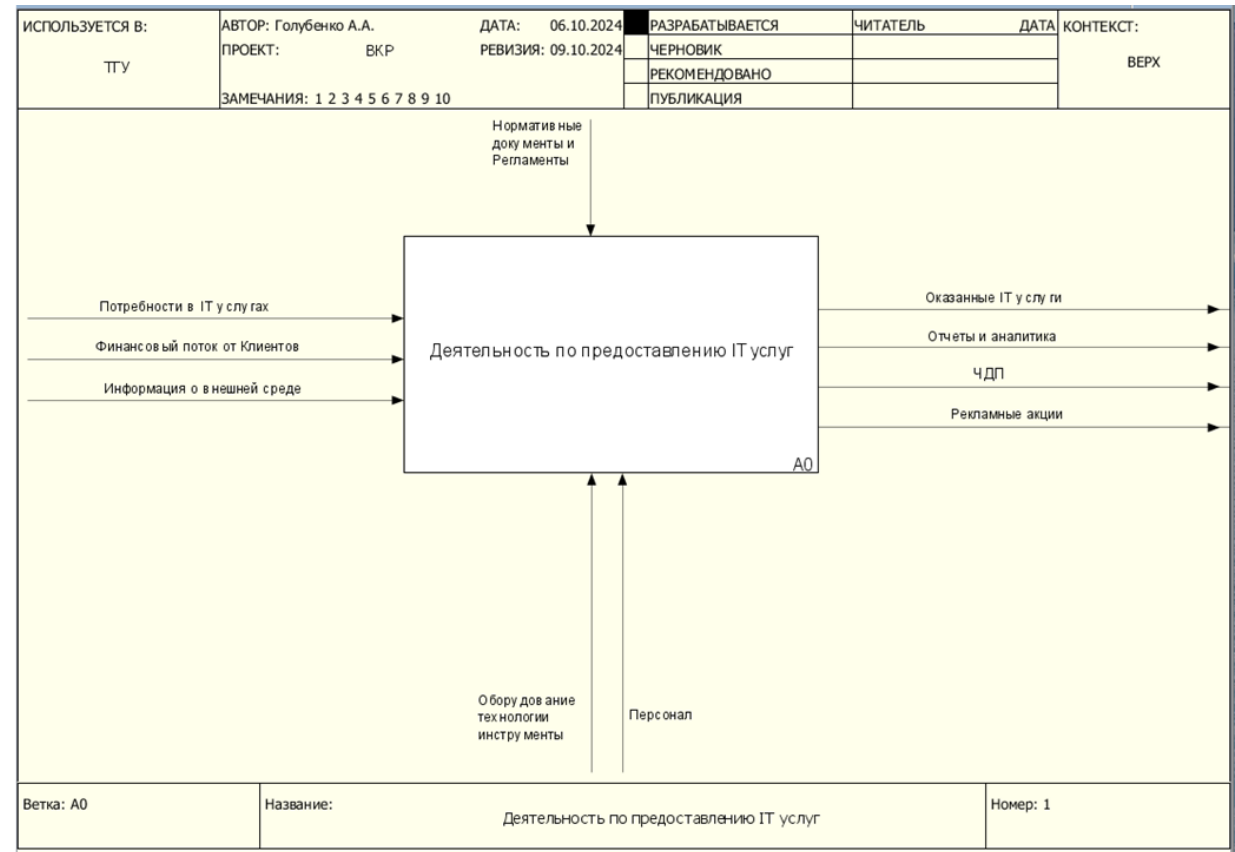


Рисунок 2 – Контекстная диаграмма A0 ООО “ВорлдИнтерТех РУС”

Выходные стрелки с правой стороны блока это данные или материальные объекты, произведенные в процессе деятельности – оказанные ИТ услуги, отчеты и аналитика, ЧДП (чистый денежный поток) и рекламные акции.

Управляющие стрелки, входящие с верхней стороной блока, это нормативные документы, регламенты и процедуры, которые определяют условия, необходимые функции, чтобы выход блока был правильным.

Стрелки механизма, входящие с нижней стороной блока, это персонал, оборудование и инструменты, используемые для выполнения функции.

Выполним декомпозицию контекстной диаграммы А0 «Деятельность по предоставлению ИТ услуг». Представлена на рисунке 3.

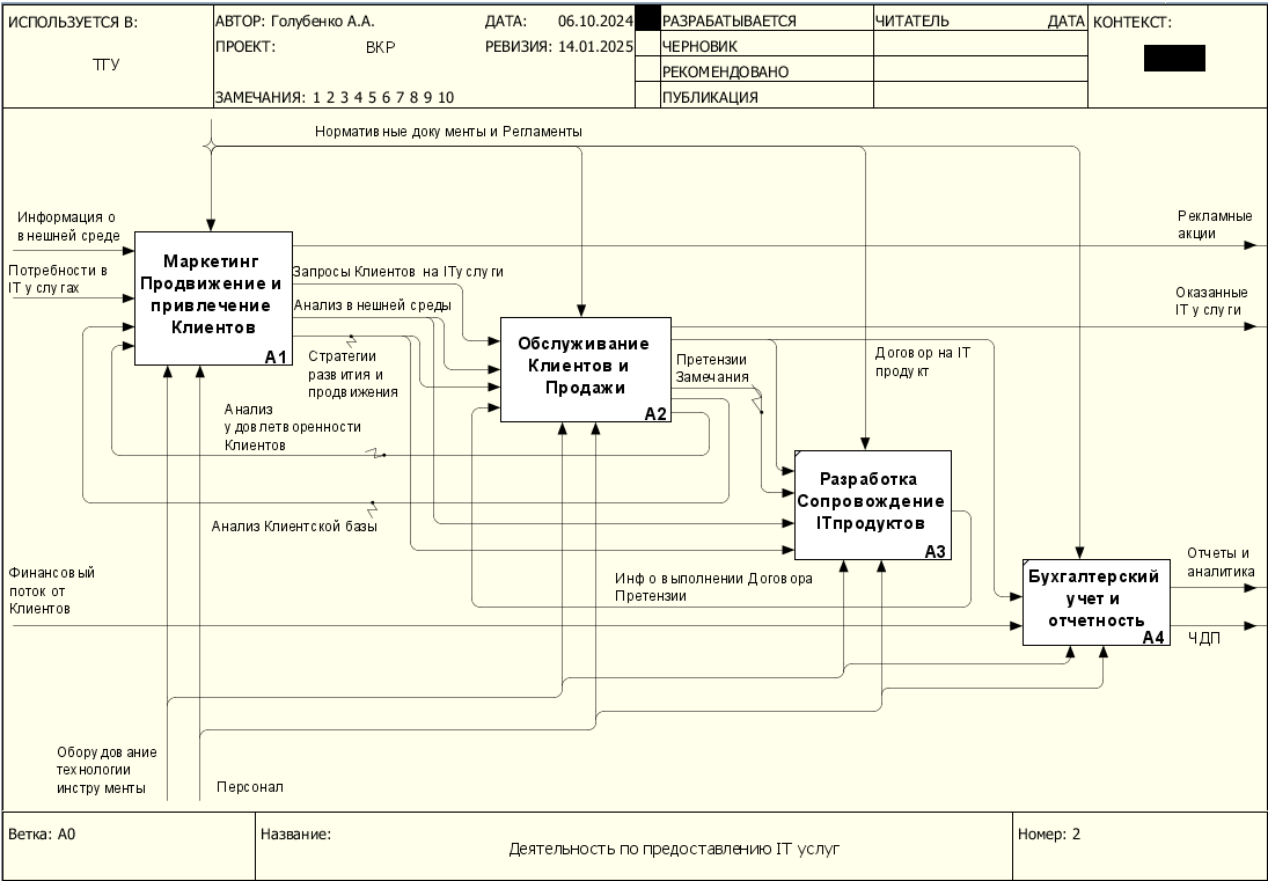


Рисунок 3 – Декомпозиция диаграммы А0

Деятельность по предоставлению ИТ услуг складывается из:

- маркетинга, продвижения и привлечения Клиентов А1;

- обслуживания Клиентов и продаж А2;
- разработки и сопровождения IT продуктов А3;
- бухгалтерского учета и отчетности А4.

Рассматриваемая предметная область – анализ и прогнозирование поведения потребителей выполняется на этапах «Маркетинг, продвижение и привлечение Клиентов» – блок А1, и «Обслуживание Клиентов и продажи» – блок А2, поэтому рассмотрим подробно эти процессы.

Выполним декомпозицию блока А1 «Маркетинг, продвижение и привлечение Клиентов», рисунок 4.

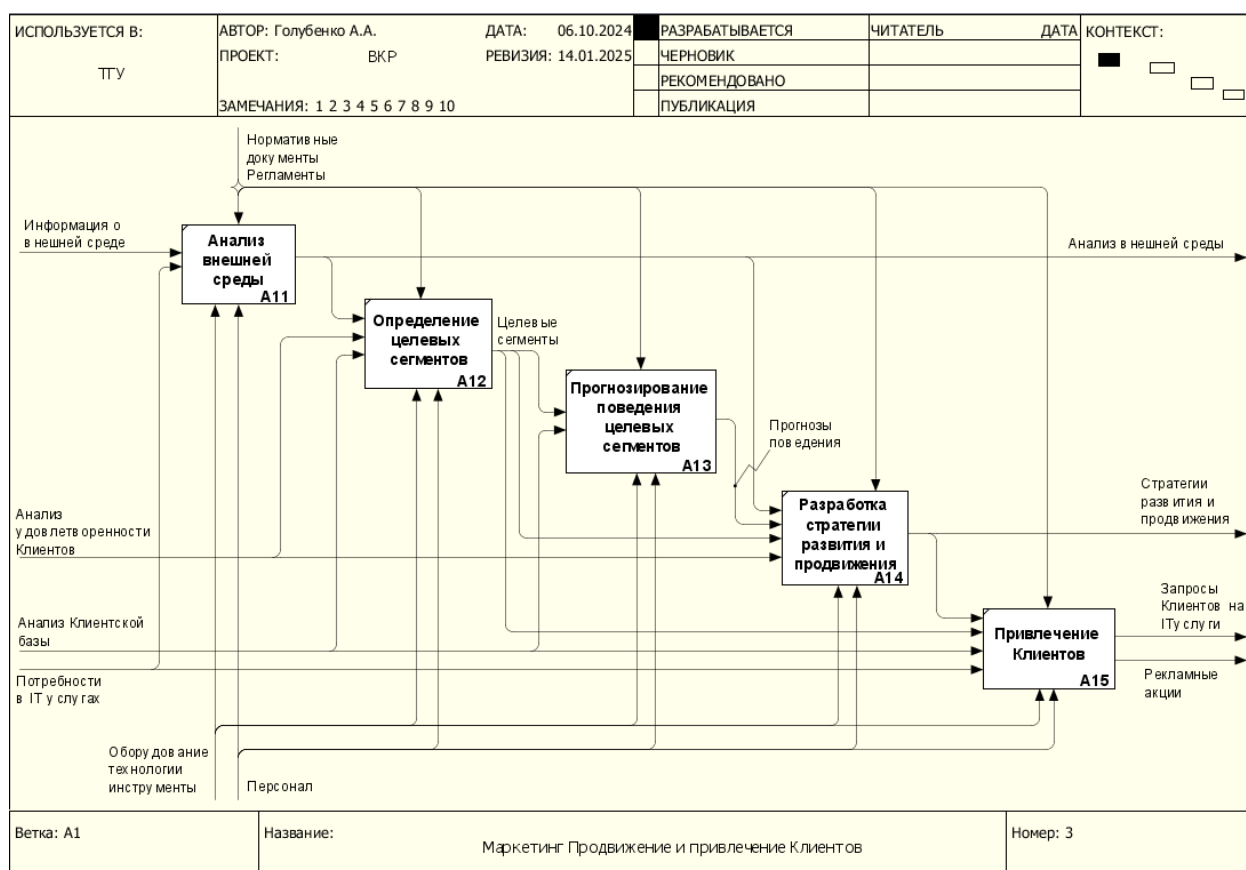


Рисунок 4 – Декомпозиция блока А1

Процесс можно разделить на составляющие подпроцессы:

Анализ внешней среды А11. Входит сбор и обработка информации о состоянии рынка IT, конкурентном окружении, предпочтениях потребителей (входной поток). Выполняется Менеджером по рекламе и используются при

определении Целевых сегментов потребителей, разработке стратегии развития компании, обслуживании Клиентов и Разработке IT продуктов.

Определение целевых сегментов потребителей A12. На основе Анализа внешней среды и Анализа клиентской базы (входной поток) выделяются группы наиболее привлекательных для компании потребителей. Выполняются работы Менеджером по рекламе без использования АИС, информация обрабатывается с помощью таблицы Microsoft Excel. Информация о целевых сегментах (выходной поток) используется при разработке Стратегии развития компании и в процессе Привлечения клиентов.

Прогнозирование поведения целевых сегментов потребителей A13. Выполняются работы на основе информации о целевых сегментах и Анализа клиентской базы Менеджером по рекламе без использования АИС на основе таблиц Microsoft Excel. Выходной поток – прогнозы поведения.

Разработка стратегии развития компании и продвижения A14. Выполняется руководством компании, Менеджером по рекламе и Менеджером по работе с клиентами на основе Анализа внешней среды, Анализа клиентской базы, Прогнозов поведения потребителей (входные потоки).

Привлечение клиентов A15. На основе Стратегии развития, определенных Целевых сегментов потребителей, Анализа клиентской базы и потребности на рынке в IT услугах разрабатываются рекламные акции – выходной поток. Выполняются работы Менеджером по рекламе.

Декомпозиция процесса «Обслуживания Клиентов и продажи» – блок A2 на составляющие подпроцессы представлена на рисунке 5.

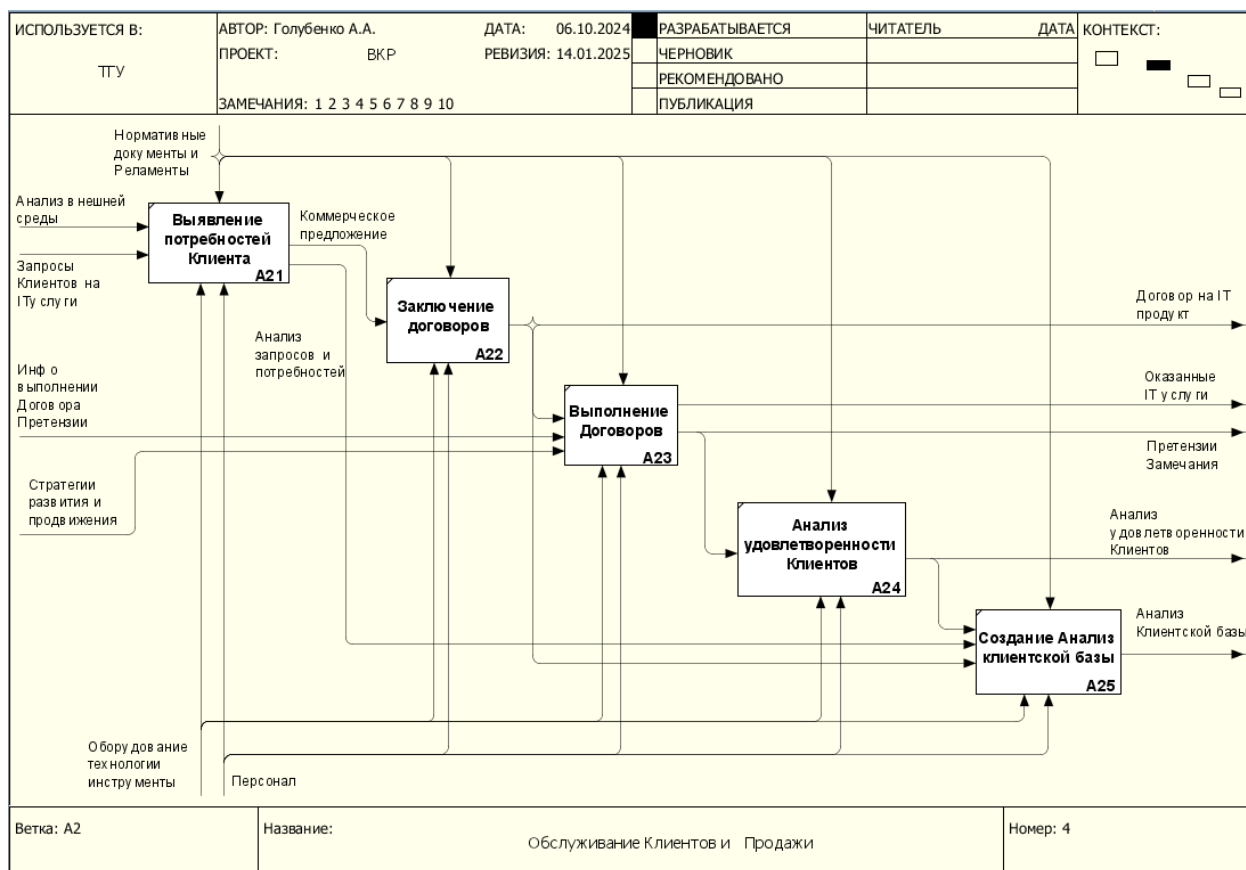


Рисунок 5 – Декомпозиция блока A2

**Выявление потребностей Клиента A21.** На этом этапе на основе Анализа внешней среды и Запросов Клиентов на ИТ услуги (входные потоки) анализируются запросы потребителей, уточняются параметры ИТ продуктов и услуг, определяются сроки и разрабатываются Коммерческие предложения (выходной поток). Выполняются работы Менеджером по работе с клиентами.

**Заключение договоров A22.** На основе Коммерческого предложения согласовываются и заключаются Договоры (выходной поток). Выполняются работы Менеджером по работе с клиентами.

**Выполнение договоров A23.** Входным потоком является заключенный Договор. Менеджер по работе с клиентами контролирует ход исполнения договорных условий, оформляет документацию об исполнении Договоров. Фиксирует, обрабатывает и направляет для устранения в отдел Разработки жалобы и Претензии (выходные потоки).

Анализ удовлетворенности Клиентов A24. Менеджер по работе с клиентами проводит Опросы клиентов, анализирует Претензии клиентов (входной поток) и на основании этих сведений формирует Анализ удовлетворенности клиентов (выходной поток). Работы выполняются без использования АИС при помощи таблиц Microsoft Excel.

Создание и Анализ клиентской базы A25. Менеджер по работе с клиентами на основе заключенных Договоров, Анализов запросов и удовлетворенности (входные потоки) формирует Клиентскую базу и выполняет Анализ Клиентской базы (выходные потоки). Работы выполняются без использования АИС при помощи таблиц Microsoft Excel.

### **1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований**

Изучение модели «AS IS» бизнес-процесса на основе существующей организации работ демонстрирует, что подпроцессы «Определение целевых сегментов потребителей» A12; «Прогнозирование поведения целевых сегментов потребителей» A13; «Анализ удовлетворенности Клиентов» A24; «Создание и Анализ клиентской базы» A25 недостаточно автоматизированы, работы выполняются без использования АИС при помощи таблиц Microsoft Excel. Внедрение специализированного ПО решает следующие задачи:

- увеличение эффективности, автоматизация трудоемких процессов увеличит скорость обработки и обеспечит более точный анализ;
- снижение влияния человеческого фактора и уменьшение ручного ввода данных, что снизит риск ошибок;
- снижение рисков, связанных с удовлетворенностью клиентов;
- оптимизация маркетинговых кампаний;
- экономия времени и ресурсов, автоматизация позволит сотрудникам выделять больше времени на стратегические задачи.

Требования к разрабатываемому ПО приведены в Таблице 2 и сформированы в соответствии с моделью FURPS+. Акроним FURPS обозначает следующие категории требований: Functionality (Функциональные требования), Usability (Удобство использования), Reliability (Надёжность), Performance (Производительность), Supportability (Поддерживаемость) [21].

Таблица 2 – Требования к программному обеспечению

| Требование                                                                                                                                                                                                             | Полезность  | Риск    | Стабильность |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------|--------------|
| Functionality – Функциональные требования                                                                                                                                                                              |             |         |              |
| 1 ПО должно обеспечивать возможность ввода, редактирования, хранения различных типов данных о Клиентах, включая наименование, реквизиты, историю (договоры, сумма, дата, состав покупок, результаты опросов).          | Критическая | Средний | Средняя      |
| 2 ПО должно автоматически обрабатывать данные о клиентах для формирования и поддержания актуального Профиля клиента.                                                                                                   | Критическая | Средний | Средняя      |
| 3 ПО должно предоставлять функционал для автоматической сегментации клиентской базы в соответствии с заданными критериями и выделять целевые сегменты на основе заданных критериев.                                    | Критическая | Средний | Средняя      |
| 4 ПО должно автоматически рассчитывать индекс NPS (Net Promoter Score) на основе опроса клиентов, сегментировать их на группы и предоставлять результаты для прогнозирования поведения потребителей.                   | Критическая | Средний | Средняя      |
| 5 ПО должно обеспечивать возможность представления данных о клиентах в сегментах клиентской базы и целевых сегментах в виде таблиц, отчетов или графиков                                                               | Важная      | Средний | Средняя      |
| 6 ПО должно обеспечивать доступ к сводной информации о клиентах для создания индивидуальных подходов к обслуживанию.                                                                                                   | Важная      | Средний | Средняя      |
| 7 ПО должно предоставлять инструменты для прогнозирования поведения потребителей:<br>– анализ связанных покупок с целью выявления ассоциаций между продуктами/услугами;<br>– аналитические индикаторы прогнозирования. | Критическая | Средний | Средняя      |
| 8 ПО должно позволять зарегистрированным пользователям входить в систему с использованием валидных учетных данных (логин и пароль).                                                                                    | Важная      | Средний | Средняя      |

Продолжение таблицы 2

| Usability – Удобство использования                                                                             |           |         |         |
|----------------------------------------------------------------------------------------------------------------|-----------|---------|---------|
| 9 ПО должно иметь адаптивный дизайн (ПК и мобильная версии).                                                   | Важная    | Средний | Средняя |
| 10 ПО должно иметь дружелюбный интуитивный интерфейс.                                                          | Важная    | Низкий  | Средняя |
| 11 ПО должно выполнять:<br>– проверка корректности введенных данных;<br>– подтверждение действий пользователя. | Важная    | Низкий  | Средняя |
| 12 ПО должно иметь справочную информацию.                                                                      | Важная    | Низкий  | Средняя |
| Reliability – Надёжность                                                                                       |           |         |         |
| 13 Время доступности системы 24/7.                                                                             | Важная    | Низкий  | Средняя |
| 14 Частота сбоев – не более чем 1 раз в месяц.                                                                 | Важная    | Средний | Средняя |
| 15 Возможность восстановления после сбоев                                                                      | Важная    | Средний | Высокая |
| Performance – Производительность                                                                               |           |         |         |
| 16 Допустимое количество одновременно работающих пользователей – 7 человек.                                    | Важная    | Средний | Средняя |
| 17 Время отклика системы не более 5 секунд                                                                     | Важная    | Средний | Средняя |
| Supportability – Поддерживаемость                                                                              |           |         |         |
| 18 Предусмотреть резервное копирования. Частота – 1 раз в неделю.                                              | Важная    | Средний | Средняя |
| 19 Возможность расширения                                                                                      | Важная    | Средний | Средняя |
| Design – ограничения проектирования                                                                            |           |         |         |
| 20 Ограничить доступ к критическим компонентам для обеспечения целостности данных и стабильности работы ПО.    | Важная    | Средний | Средняя |
| 21 Обеспечить низкую стоимость реализации и поддержки программного обеспечения                                 | Важная    | Средний | Средняя |
| Implementation – ограничения разработки                                                                        |           |         |         |
| 22 В качестве СУБД использовать MySQL.                                                                         | Критичная | Средний | Высокая |
| 23 Документация должна создаваться в MS Excel.                                                                 | Критичная | Средний | Высокая |
| 24 Система должна быть совместима с платформами Windows и Mac.                                                 | Критичная | Средний | Высокая |

FURPS+ является расширенной версией классификации требований. Символ «+» добавляет к модели FURPS: Implementation (ограничения разработки), Interface (ограничения на интерфейсы), Design (ограничения проектирования), Physical (физические ограничения) [16].

### **1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям**

#### **1.3.1 Определение критериев анализа**

На рынке ПО имеется большой выбор разработок для анализа и прогнозирования поведения потребителей. Для их оценки и принятия решения о возможности использования в компании определим критерии, по которым будет производиться сравнение.

Система должна соответствовать функциональным требованиям, описанным в предыдущей главе:

- формирование Клиентской базы и Профиля клиента;
- сегментация клиентов;
- определение индекса лояльности – NPS;
- прогнозирование продаж на основе имеющихся данных.

Дополнительно оценим:

- простоту использования и необходимость в обучении персонала;
- развертывание Облако (SaaS англ. Software-as-a-Service), Сервер, ПК;
- стоимость и наличие избыточного функционала;
- наличие и доступность технической поддержки.

#### **1.3.2 Сравнительная характеристика существующих разработок**

Для автоматизации функционала, необходимого для анализа и прогнозирования поведения потребителей существуют различные CRM-системы – системы управления взаимоотношениями с клиентами (Customer Relationship Management). CRM-системы решают такие задачи как работа с клиентской базой, обработка заказов на всех этапах воронки продаж, запуск и автоматизация CRM-маркетинга [10]. Для сравнительного анализа выберем наиболее широко представленные на российском рынке программные средства:

1С: CRM – это CRM-система, разработанная компанией ООО «1С» для автоматизации управления взаимоотношениями с клиентами, существует на российском рынке много лет и занимает значительный сегмент, обладает высокой надежностью. Обеспечивает интеграцию с другими программами 1С. Включена в Единый реестр российского программного обеспечения №1326 от 05.09.2016. Лучше всего подходит для крупного бизнеса и для работы на технологической платформе «1С: Предприятие 8.3.

Функциональные возможности включают хранение и управление информацией о клиентах, системы анкетирования, подготовка отчетов, анализ клиентской базы [19]. Из недостатков – сложная настройка, избыточный функционал, необходимость привлекать специалиста по 1С, высокая стоимость внедрения и поддержки, отсутствие мобильных приложений [20].

Битрикс24.CRM – это многофункциональный корпоративный портал для управления коммуникациями с клиентами, позволяет автоматизировать процессы сбора данных о клиентах, ведения и сегментации базы клиентов, аналитики и прогнозирования. Возможно расширение функционала за счет маркетплейса приложений, имеет как коробочную версию (установка на сервер), так и облачную [12]. Включена в Единый реестр российского программного обеспечения №25 от 20.02.2016 и входит в список крупнейших CRM. Ориентирована на средний и крупный бизнес.

Из недостатков – нецелесообразна для небольших компаний ввиду технической сложности и необходимость обучения персонала, нет возможности выбрать нужные опции и отказаться от лишних. Имеет высокую стоимость, при сбоях не всегда быстрый ответ от службы поддержки [20].

EnvyCRM – простая и логичная CRM-система, не требующая обучения персонала. Имеет интуитивный интерфейс и современный дизайн. EnvyCRM довольно гибкая, ее можно расширять дополнительными модулями под нужды конкретного бизнеса. Есть мобильные приложения под iOS и Android. Соответствует федеральным законам № 152-ФЗ о Персональных данных и № 242-ФЗ. Работает в браузере и на платформах Android и Apple.

Предоставляет такие опции как управление базой клиентов и историей взаимодействия, аналитика продаж, интеграции с большим количеством сервисов. Пользователи отмечают оперативную техподдержку.

Недостатки – перед запуском требуется значительный объем работ по настройке и доработке системы, отмечаются проблемы с выгрузкой информации в Excel и частые сбои, большинство функций платные [20].

Результат анализа аналогов информационных систем для анализа и прогнозирования поведения потребителей представлен в таблице 3.

Таблица 3 – Сравнение существующих аналогов ИС

| Критерии                                                | 1С: CRM    | Битрикс24. CRM     | EnvyCRM |
|---------------------------------------------------------|------------|--------------------|---------|
| Формирование Клиентской базы и Профиля клиента          | 3          | 3                  | 3       |
| Сегментация клиентов                                    | 3          | 3                  | 2       |
| Определение индекса лояльности                          | 1          | 2                  | 0       |
| Прогнозирование продаж на основе имеющихся данных       | 1          | 1                  | 0       |
| Юзабилити                                               | 1          | 2                  | 3       |
| Доступность по стоимости                                | 0          | 0                  | 1       |
| Развертывание                                           | ПК, Сервер | ПК, Сервер, Облако | Облако  |
| Отсутствие или возможность убрать избыточный функционал | 0          | 0                  | 3       |
| Отсутствие необходимости в обучении персонала           | 0          | 0                  | 3       |
| Наличие и доступность технической поддержки             | 2          | 2                  | 3       |
| Надежность                                              | 3          | 3                  | 2       |

Исходя из анализа таблицы 3, делаем следующие выводы:

- все рассматриваемые аналоги имеют определенные недостатки и не соответствуют всем требуемым функциям;
- стоимость использования высока или значительно возрастает при добавлении необходимых опций;
- потребуется доработка системы;
- системы с большим функционалом сложны и требуют дополнительного обучения персонала.

Следовательно, оптимальным вариантом решения задачи будет разработка собственной информационной системы.

#### **1.4 Постановка задачи на разработку проекта создания автоматизированной информационной системы**

На основе исследований, проведенных в разделе 1.2, сформулируем цели и основные требования к будущей информационной системе.

Бизнес-цели проекта:

- создание эффективного ПО для оптимизации процессов анализа Клиентской базы, определения целевых сегментов потребителей и прогнозирования их поведения;
- разработка ПО позволит предоставлять персонализированные услуги на основе предпочтений клиентов, улучшая их опыт взаимодействия с компанией;
- оптимизировать маркетинговые стратегии, проводя целевые рекламные кампании;
- эффективное управление ресурсами с использованием ПО.

Требования к ПО:

- ПО должно обеспечивать возможность хранения данных о клиентах, их взаимодействии с продуктами и других метрик для анализа;
- ПО должно иметь необходимый набор алгоритмов для анализа данных, прогнозирования и выявления закономерностей поведения потребителей;
- ПО должно иметь интуитивный интерфейс и визуализацию результатов анализа для удобства пользователей;
- ПО должно обеспечить защиту и конфиденциальность данных клиентов, соответствие стандартам безопасности и законодательства.

Требования к архитектуре:

- централизованное хранение данных для обеспечения надежности и целостности данных;
- легкая масштабируемость, возможность добавлять новых клиентов и расширять функционал системы без значительных изменений;
- архитектура должна обеспечивать необходимую производительность и быстрый отклик системы на запросы;
- совместимость клиентских приложений с операционными системами и устройствами, а также совместимость серверных компонентов с базами данных и приложениями;
- архитектура должна быть удобной, обеспечивая интуитивный интерфейс и удобный доступ к функционалу.

### **1.5 Разработка модели бизнес-процесса «ТО ВЕ»**

На основе модели «ТО ВЕ» проектируется модель данных и затем информационная система. Построение на основе модели «ТО ВЕ» приводит к тому, что информационная система автоматизирует несовершенные бизнес-процессы [6]. Модель «ТО ВЕ» процесса «Деятельность по предоставлению IT услуг» разработана так же, как и модель «AS IS» с использованием CASE-средства Ramus, в нотации IDEF0.

Построение начинается с контекстной диаграммы верхнего уровня, представленной на рисунке 6.

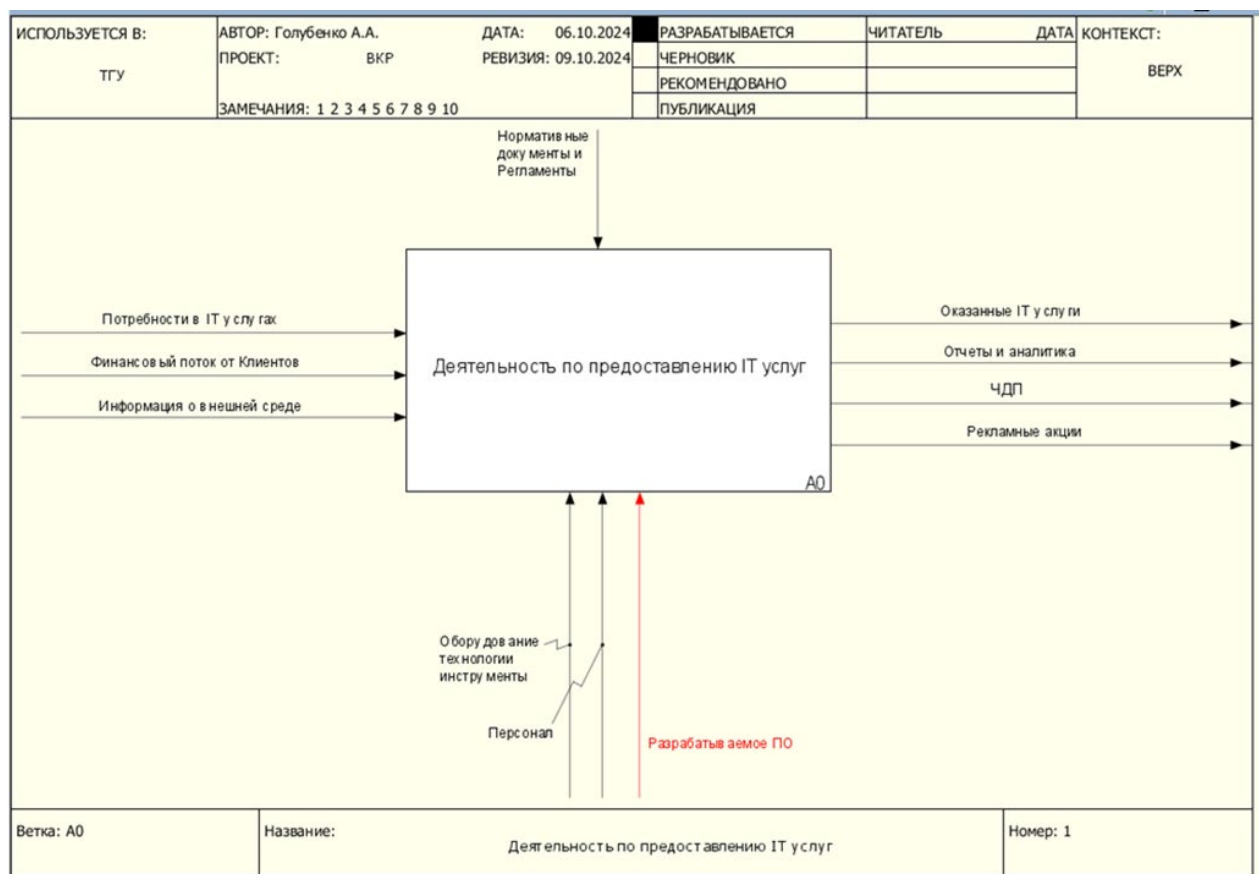


Рисунок 6 – Контекстная диаграмма модели «ТО BE»

Добавлена стрелка механизма, входящая с нижней стороной блока, это «Разрабатываемое ПО» для анализа и прогнозирования поведения потребителей.

Выполним декомпозицию контекстной диаграммы A0 «Деятельность по предоставлению IT услуг». Предметом автоматизации в соответствии с темой ВКР «Разработка программного обеспечения для анализа и прогнозирования поведения потребителей» являются процессы «Маркетинг, продвижение и привлечение Клиентов» A1 и «Обслуживания Клиентов и продажи» A2. В эти процессы добавлена стрелка механизма «Разрабатываемое ПО». Декомпозиция контекстной диаграммы A0 представлена на рисунке 7.

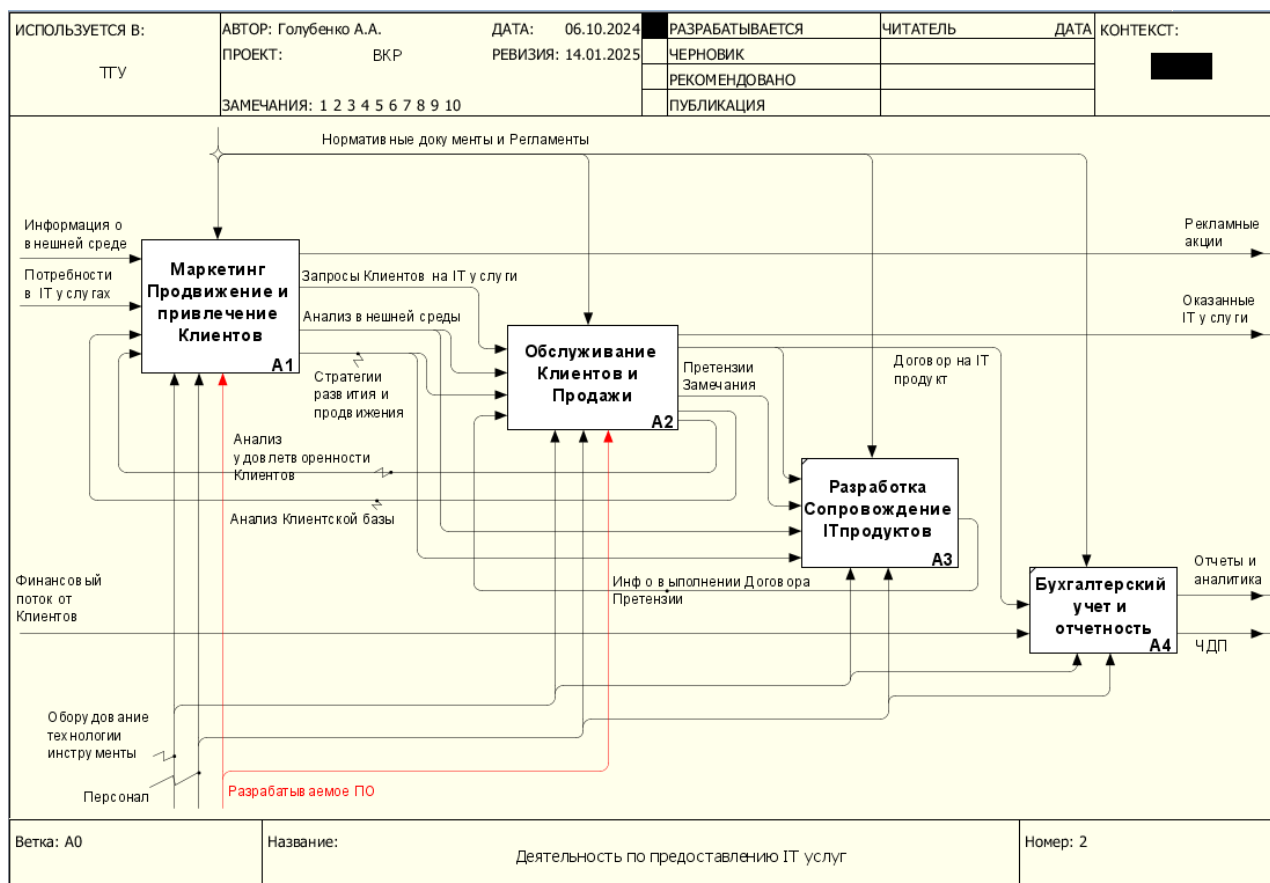


Рисунок 7 – Декомпозиция контекстной диаграммы модели «ТО ВЕ»

Декомпозируем функциональный блок «Маркетинг, продвижение и привлечение Клиентов» A1 модели «ТО ВЕ». Процессы «Определение целевых сегментов» и «Прогнозирование поведения целевых сегментов» модели «AS IS» (рис. 8) будут выполняться с помощью разрабатываемого ПО и объединены в единый процесс «Определение целевых сегментов, Анализ и Прогнозирование их поведения» A12. Разрабатываемое ПО позволит автоматически, на основании данных Анализа Клиентской базы и удовлетворенности Клиентов определять целевые сегменты Клиентов, анализировать и прогнозировать их поведение. Декомпозиция блока A1 «Маркетинг, продвижение и привлечение Клиентов» модели «ТО ВЕ» представлена на рисунке 8.

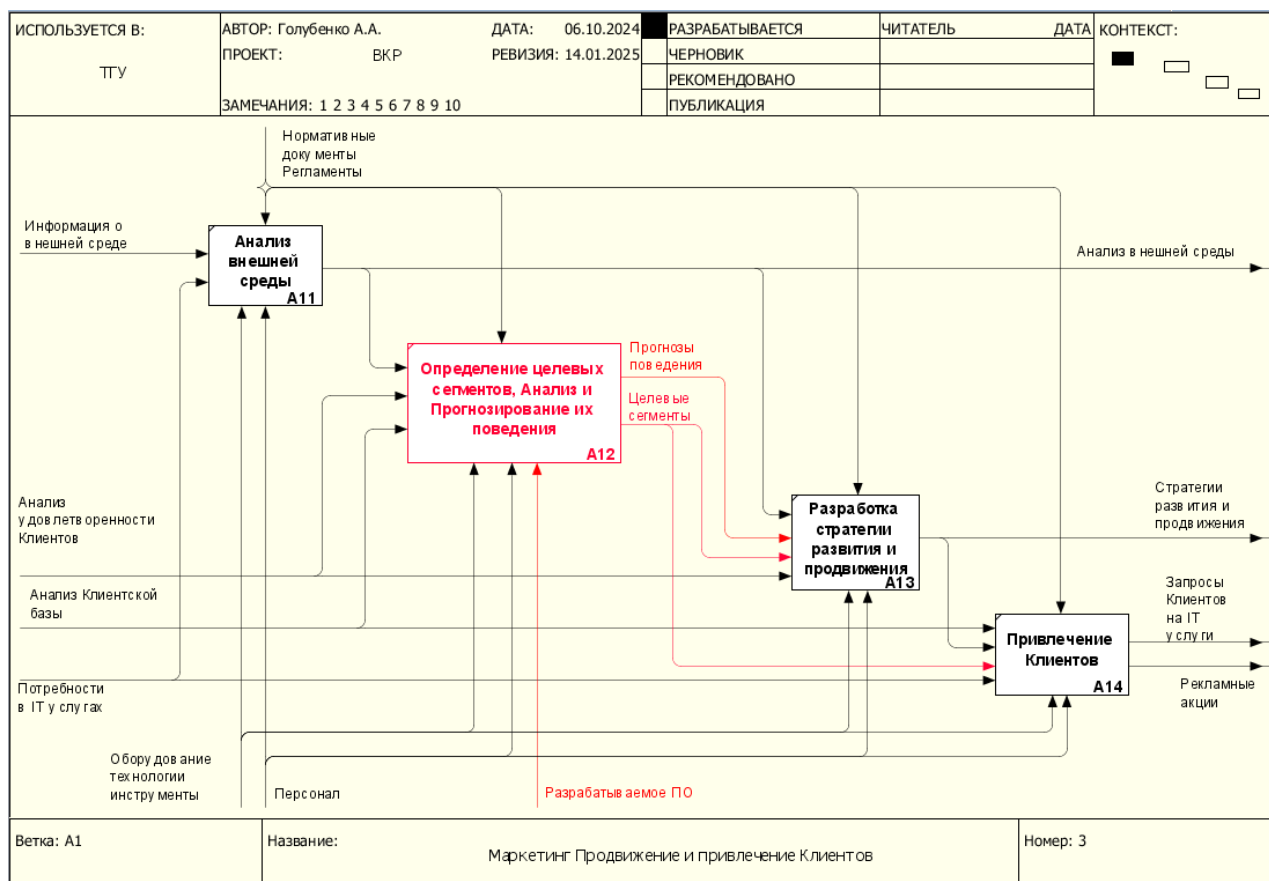


Рисунок 8 – Декомпозиция блока A1 модели «TO BE»

Переходим к декомпозиции функционального блока «Обслуживание Клиентов и продажи» A2. Результат декомпозиции представлен на рисунке 9. Подпроцесс A25 «Ведение и Анализ клиентской базы» также будет выполняться при помощи разрабатываемого ПО для анализа и прогнозирования поведения потребителей.

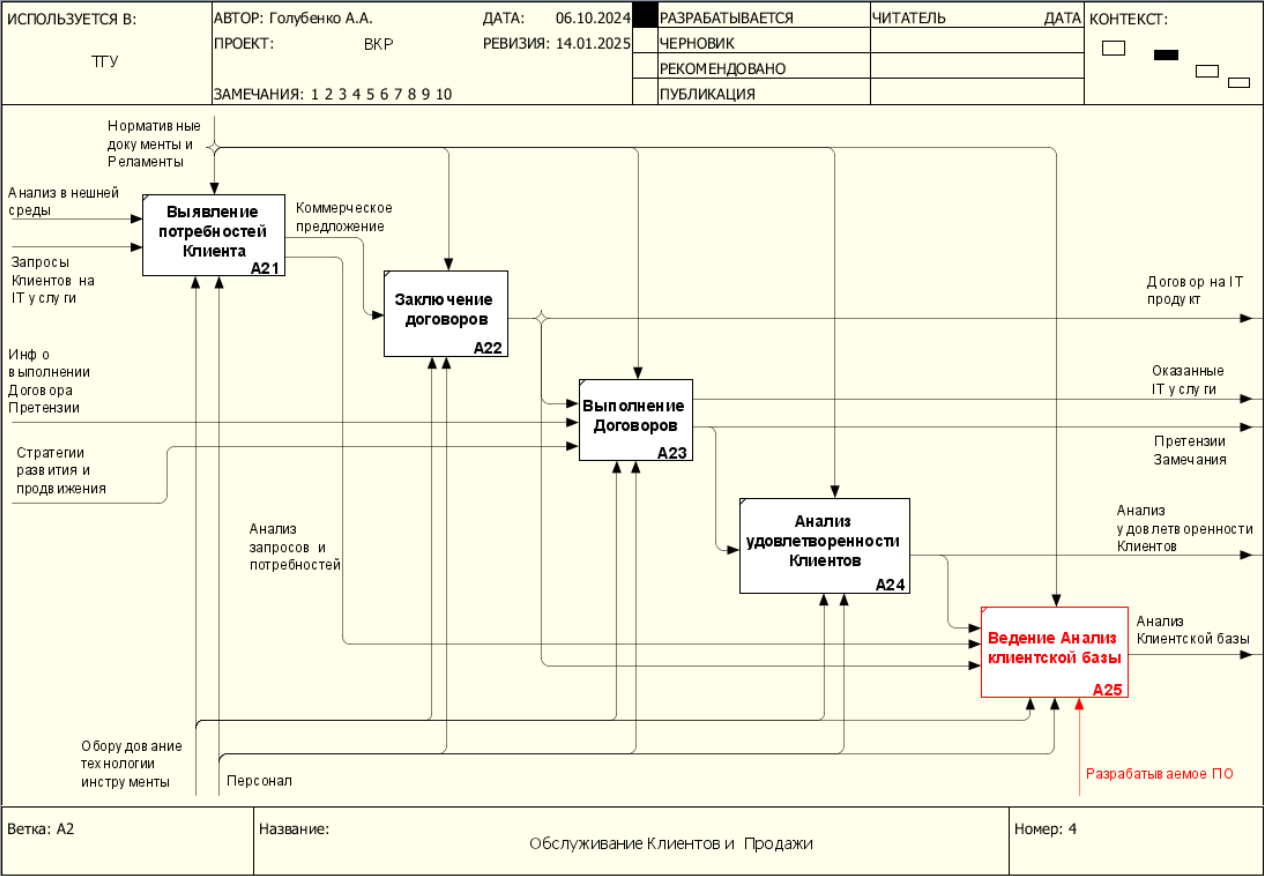


Рисунок 9 – Декомпозиция блока A2 модели «ТО ВЕ»

Разрабатываемое ПО в каждом из рассмотренных процессов будет использоваться для автоматизации трудоемких операций по анализу и прогнозированию поведения потребителей и позволит значительно увеличить скорость и точность обработки данных, улучшить качество обслуживания и повысить удовлетворенность клиентов.

Выводы по главе 1

В первой главе приведено описание объекта исследования – IT-компании ООО “ВорлдИнтерТех РУС” и дано представление о целях и задачах компании.

Представлен анализ организационной структуры и бизнес-процессов в организации «AS IS», описана сущность исследуемой задачи по анализу и прогнозированию поведения потребителей. Исследование существующих процессов ведения клиентской базы, определения целевых сегментов

потребителей и прогнозирования их поведения выявило недостаточную их автоматизацию, что может препятствовать эффективной работе и развитию компании.

Для принятия решения по автоматизации были проанализированы существующие разработки в области анализа и прогнозирования поведения потребителей. Анализ показал, что все существующие разработки имеют определенные недостатки – это высокая стоимость и сложность систем с широким функционалом, необходимость дополнительного обучения персонала или необходимость доработки недорогих и простых систем. Поэтому принято решение о разработке собственного специализированного программного обеспечения.

Выполнена постановка задачи на разработку ПО, которое позволит автоматически создавать и анализировать Клиентскую базу, определять целевые сегменты Клиентов и прогнозировать их поведение. Определены цели проекта, требования к ПО и к его архитектуре.

В итоге предложена усовершенствованная модель бизнес-процессов «ТО ВЕ» с использованием разрабатываемого ПО. Это решение позволит увеличить эффективность существующих бизнес-процессов, точность обработки данных, снизить риски, связанные с удовлетворенностью клиентов, сократит количество рутинных задач и позволит сотрудникам компании использовать свое время более продуктивно.

## **Глава 2    Логическое    проектирование    автоматизированной информационной системы**

### **2.1    Выбор    технологии    логического    моделирования автоматизированной информационной системы**

Логическое моделирование играет важную роль в процессе разработки программного обеспечения, обеспечивая понимание структуры и поведения системы и выполняет следующие задачи:

- позволяет определить функциональные возможности информационной системы, разбить ее на компоненты и определить способы их взаимодействия;
- включает построение моделей данных системы, определение структуры базы данных, сущностей, атрибутов и связей между ними;
- является промежуточным звеном между концептуальной моделью и физической моделью, определяя основные принципы для последующей разработки.

На данном этапе предпочтителен подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом, это объектно-ориентированное проектирование. Для реализации этого подхода применяется унифицированный язык моделирования UML (Unified Modeling Language) – стандартизированный набор диаграмм и нотаций, который обеспечивает описание требований и моделей системы [4].

В работе будет использоваться CASE-средство StarUML. StarUML™ – программная платформа моделирования, которая поддерживает UML (Унифицированный Язык Моделирования). Она основана на версии UML 1.4 и поддерживает нотацию UML версии 2.0 и одиннадцать различных типов диаграмм [23].

## 2.2 Разработка логической модели автоматизированной информационной системы

Для описания взаимодействия между пользователями и системой разработана Диаграмма вариантов использования Use Case Diagram, на ней изображены отношения между актерами и вариантами использования. Актером (actor) является объект, взаимодействующий с моделируемой системой извне [17].

В представленной модели (рисунок 10) выделено два главных актера:

- Менеджер по работе с клиентами – пользователь приложения, который вносит данные о Клиенте, занимается процессом формирования и анализа Клиентской базы;
- Менеджер по рекламе – пользователь приложения, который просматривает Клиентскую базу, ее анализ и занимается процессами выделения целевых сегментов и прогнозирования.

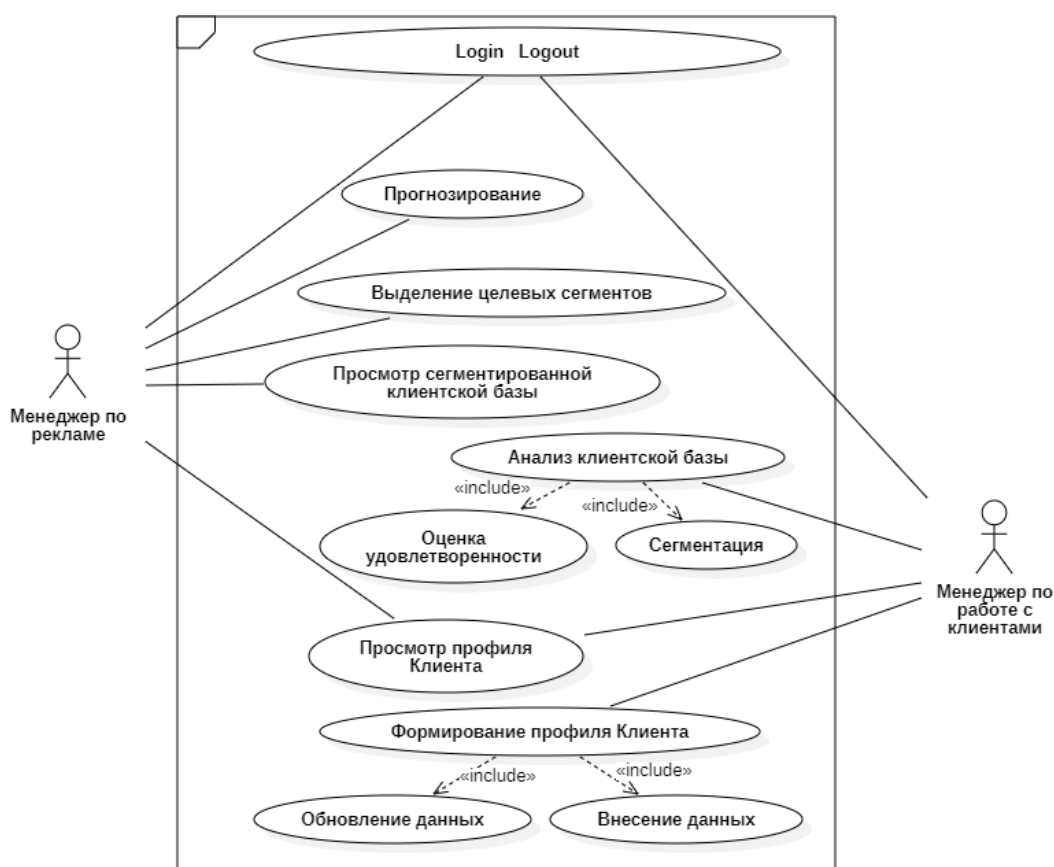


Рисунок 10 – Диаграмма вариантов использования

Вариант использования (прецедент, Use Case) – это функциональность системы, соответствующая отдельному сервису, предоставляемому моделируемой системой в ответ на запрос пользователя, и определяет способ использования этой системы [17].

В таблице 4 приведена характеристика прецедентов диаграммы.

Таблица 4 – Характеристика прецедентов

| Прецедент                                 | Характеристика                                                                                       |
|-------------------------------------------|------------------------------------------------------------------------------------------------------|
| Login / Logout                            | Регистрация пользователя в системе с последующим созданием его профиля                               |
| Формирование профиля Клиента              | Пользователь создает страницу с профилем (информацией) Клиента                                       |
| Внесение данных                           | Пользователь вносит необходимые данные о Клиенте                                                     |
| Обновление данных                         | Пользователь вносит изменения в информацию на странице Клиента                                       |
| Просмотр профиля Клиента                  | Пользователь просматривает страницу с профилем Клиента                                               |
| Анализ Клиентской базы                    | Пользователь выполняет анализ Клиентской базы по требуемым критериям                                 |
| Сегментация                               | Пользователь выполняет сегментацию Клиентской базы по требуемым критериям                            |
| Оценка удовлетворенности                  | Пользователь рассчитывает индекс лояльности Клиентов NSP                                             |
| Просмотр сегментированной клиентской базы | Пользователь просматривает сегментированную клиентскую базу                                          |
| Выделение целевых сегментов               | Пользователь выделяет группы наиболее привлекательными для компании Клиентов, по требуемым критериям |
| Прогнозирование                           | Пользователь производит расчет прогнозов продаж по требуемым критериям                               |

Для описания структуры и взаимодействия между компонентами системы разработана Диаграмма классов (class diagram), на которой изображены классы системы с атрибутами и методами, а также взаимосвязи между классами [18]. Диаграмма классов представлена на рисунке 11.

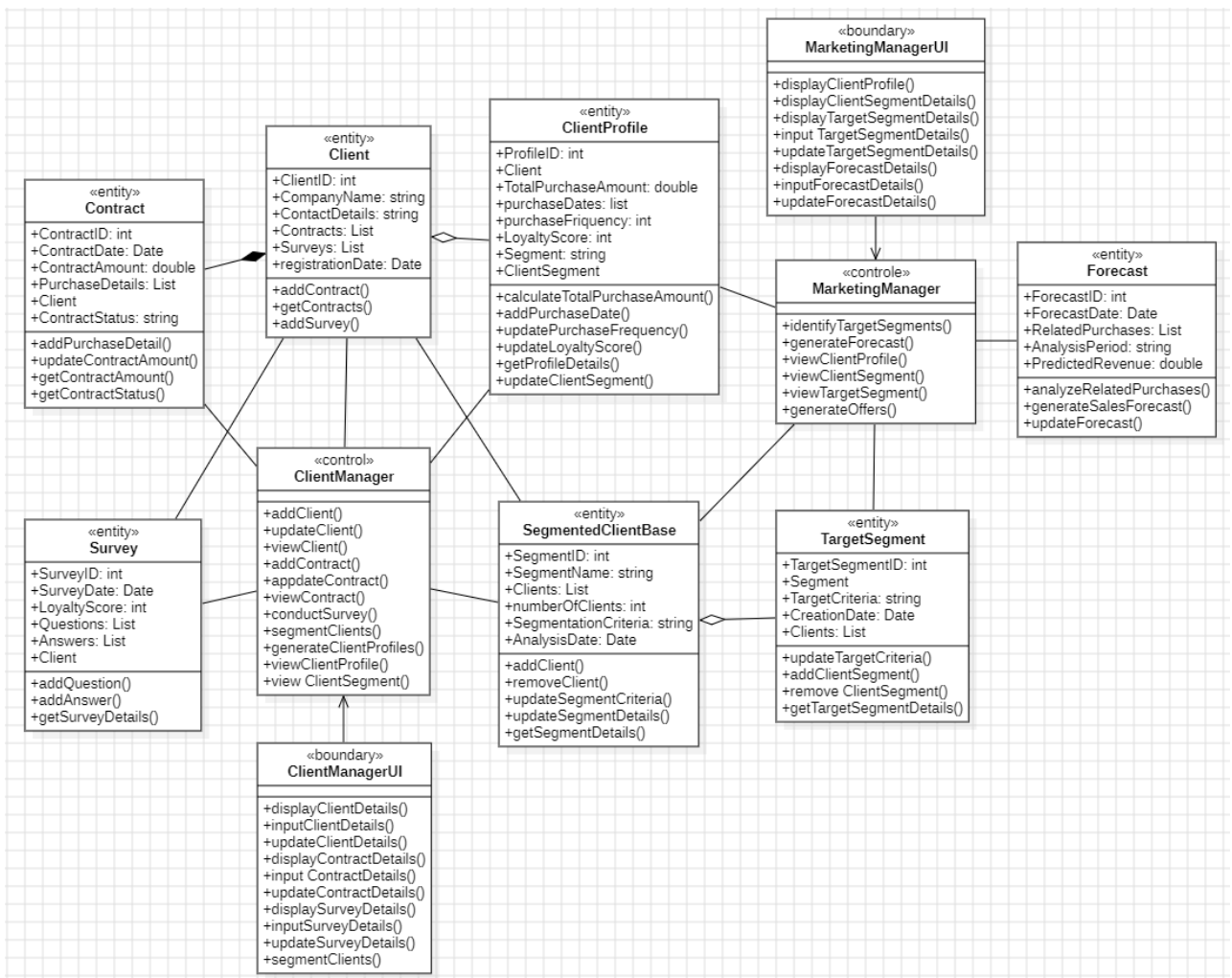


Рисунок 11 – Диаграмма классов

Диаграмма включает:

- граничные классы «boundary» ClientManagerUI и MarketingManagerUI взаимодействуют с внешним миром и работают как интерфейсы между пользователями и логикой приложения;
- классы управления «control» ClientManager и MarketingManager отвечают за реализацию логики приложения, обрабатывают пользовательские запросы, взаимодействуют с сущностями и выполняют необходимые операции;
- классы сущности «entity» Client, Contract, Survey, ClientProfile, SegmentedClientBase, TargetSegment, Forecast представляют данные и их связи в системе, хранят информацию, реализуют бизнес-логику, связанную с данными.

## 2.3 Проектирование концептуальной и логической моделей базы данных автоматизированной информационной системы

Для представления общей структуры данных и их взаимосвязей разработана Концептуальная модель– диаграмма «сущность-связь» ER модель (рисунок 12). Для разработки также использовано CASE-средство StarUML.

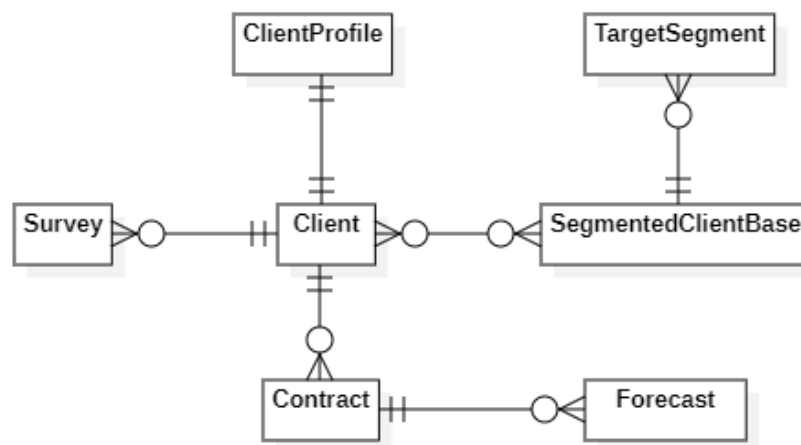


Рисунок 12 – Концептуальная модель данных

Основным элементом Модели данных является сущность («entity») – объект или понятие из реального мира, значимое для системы, которое подлежит хранению и обработке в базе данных [5]. Предметная область содержит следующие сущности:

Client – содержит информацию о клиентах.

ClientProfile – содержит обобщенную информацию о клиенте, связан с сущностью Client отношением «один-к-одному».

Contract – содержит информацию о договорах, заключенных с клиентом, связан с сущностью Client отношением «один-ко-многим», то есть один клиент может иметь несколько договоров.

Survey – содержит данные о проведенных опросах клиентов, включая оценки удовлетворенности, связан с сущностью Client отношением «один-ко-многим», то есть один клиент может участвовать в нескольких опросах.

SegmentedClientBase – содержит сегментированную клиентскую базу, связана с сущностью Client отношением «многие ко многим».

TargetSegment – содержит информацию о целевых сегментах клиентов, связан с сущностью SegmentedClientBase отношением «один ко многим».

Forecast – содержит информацию о выполненных прогнозах, связан с сущностью Contract отношением «один ко многим».

После завершения работы над концептуальной моделью переходим к созданию логической модели базы данных. Логическая модель используется для описания структуры данных и их характеристик без привязки к конкретной СУБД [5]. На этом этапе определяются атрибуты каждой сущности и устанавливаются ключи (первичные РК и внешние FK). Логическая модель данных представлена на рисунке 13.

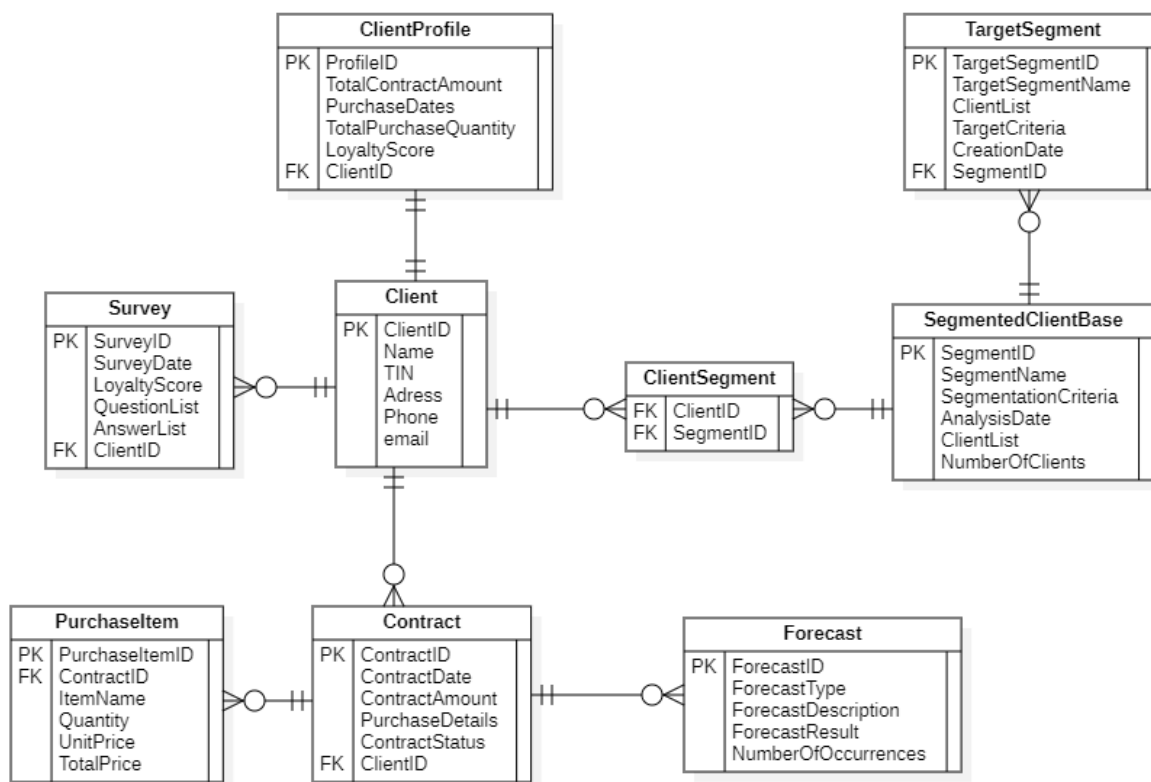


Рисунок 13 – Логическая модель данных

Для более эффективного анализа данных, выявления паттернов покупок и связанных покупок создана отдельная таблица PurchaseItem для хранения

информации о покупках с детальной структурой, где каждая покупка будет представлять отдельную запись.

Для обеспечения гибкости при работе с сегментами клиентской базы и управления связями между клиентами и сегментами создана дополнительная сущность ClientSegment как промежуточная таблица. Промежуточная таблица ClientSegment используется для реализации связи many-to-many между клиентами и сегментами. Связь many-to-many позволяет каждому клиенту принадлежать к нескольким сегментам и каждому сегменту содержать нескольких клиентов.

## **2.4 Требования к аппаратно-программному обеспечению автоматизированной информационной системы**

Для успешной реализации и обеспечения стабильной и эффективной работы информационной системы необходимо определить минимальные требования к аппаратно-программному обеспечению. Рассмотрим основные технические средства, необходимые для функционирования АИС, включая серверную часть, пользовательские ПК и мобильные устройства.

Сервер является ключевым компонентом АИС, обеспечивая хранение, обработку и управление данными. Его производительность, надежность и безопасность напрямую влияют на общую эффективность системы.

Минимальные требования к серверу:

- процессор – 2-ядерный с тактовой частотой 2.0 ГГц или выше;
- оперативная память не менее 8 ГБ;
- свободное место на жестком диске 256 ГБ SSD;
- операционная система Windows Server 2019.

Для обеспечения гарантий, что все пользователи смогут эффективно взаимодействовать с системой, необходимо определить минимальные требования к аппаратному и программному обеспечению их персональных компьютеров.

Минимальные требования к ПК пользователей:

- процессор 2-ядерный с тактовой частотой 2.0 ГГц или выше;
- оперативная память 8 ГБ;
- свободное место на жестком диске 256 ГБ;
- операционная система macOS Catalina;
- браузер – Google Chrome, Mozilla Firefox, Microsoft Edge (последние версии);
- сетевое подключение – стабильное интернет-соединение с пропускной способностью не менее 10 Мбит/с;
- графический процессор – интегрированная графика, достаточная для отображения пользовательского интерфейса;
- разрешение экрана – не менее 1366x768 пикселей.

Минимальные требования к мобильным устройствам пользователей:

- операционная система iOS 13;
- оперативная память 2 ГБ;
- свободное место на устройстве – 100 МБ для установки приложения и хранения временных данных;
- стабильное интернет – соединение с пропускной способностью не менее 10 Мбит/с;
- графический процессор – интегрированная графика, достаточная для отображения пользовательского интерфейса и выполнения базовых графических задач.

Соблюдение минимальных требований позволит избежать непредвиденных сбоев, а также обеспечит высокую производительность и надежность системы.

## Выводы по главе 2

Во второй главе выполнено логическое проектирование автоматизированной информационной системы для анализа и прогнозирования поведения потребителей с использованием унифицированного языка моделирования UML и CASE-средства StarUML, что обеспечило наглядность и структурированность логических моделей.

В рамках логического проектирования АИС были разработаны диаграмма вариантов использования и диаграмма классов, которые позволили определить основные функции системы и её структуру.

Разработка концептуальной модели данных в виде диаграммы «сущность-связь» и логической модели данных обеспечили ясное понимание структуры данных и взаимосвязей между ними.

Наличие четких и структурированных логических моделей обеспечивает возможность перехода к физическому проектированию АИС, что включает разработку физических схем баз данных, определение архитектуры системы и детальную проработку компонентов.

## **Глава 3 Физическое проектирование автоматизированной информационной системы**

### **3.1 Выбор архитектуры автоматизированной информационной системы**

Рассмотрим файл-серверную архитектуру. Это простая модель, которая используется для обмена файлами и данными между клиентами и сервером. Все файлы и данные хранятся на центральном файловом сервере, а клиенты подключаются к серверу через сеть для получения доступа к файлам.

Преимуществами файл-серверной архитектуры являются простота реализации и низкая стоимость. Для небольших объемов данных и ограниченного числа пользователей файл-серверная архитектура обеспечивает быстрый доступ к данным.

Недостатки – все операции выполняются через один сервер. При увеличении количества пользователей производительность значительно снижается, хранение данных на общем файловом сервере создает риски безопасности. Отсутствие централизованной обработки данных затрудняет выполнение аналитических задач и прогнозирования, поэтому файл-серверная архитектура является простым и доступным вариантом для систем с небольшими требованиями к производительности и безопасности.

Рассмотрим клиент-серверную архитектуру. Клиент-серверная архитектура представляет собой модель, где функции системы распределены между клиентами и серверами. Клиенты отвечают за интерфейс пользователя и отправку запросов, а серверы – центральное устройство или набор устройств, обрабатывают эти запросы и управляют данными.

Преимущества клиент-серверной архитектуры:

- простота – клиент-серверная архитектура легко и реализуется, что ускоряет процесс разработки и тестирования;
- централизованное хранение данных позволяет обеспечить их надежность, целостность и управление доступом к данным;

- позволяет легко добавлять новых клиентов и расширять функционал системы без значительных изменений;
- изменения в бизнес-логике и обновления могут быть сделаны на серверной стороне и распространены на клиентов;
- клиентские приложения могут быть реализованы на различных платформах (веб, настольные или мобильные приложения);
- сервер может выполнять сложные аналитические задачи и прогнозирование используя более мощные вычислительные ресурсы.

Недостатки клиент-серверной архитектуры:

- при большом увеличении количества пользователей нагрузка на сервер может возрасти, что приведет к снижению производительности;
- при сбое сервера вся система может перестать работать.

В контексте решаемой задачи «Разработка ПО для анализа и прогнозирования поведения потребителей» клиент-серверная архитектура имеет весомые преимущества перед файл-серверной:

- позволяет использовать централизованный сервер для выполнения аналитических задач;
- централизованное управление безопасностью и доступом к данным снижает риски несанкционированного доступа.

Клиент-серверная архитектура является популярной и гибкой моделью, подходит для широкого спектра приложений. Она обеспечивает масштабируемость, централизованное управление данными, что особенно важно, если безопасность и защита конфиденциальных данных являются приоритетами. На этих основаниях выбрана клиент-серверная архитектура.

Рассмотрим трехзвенную клиент-серверную архитектуру. Трехзвенная клиент-серверная архитектура разделяет функциональность на три отдельных уровня: клиентский уровень, уровень приложения и серверный уровень.

Клиентский уровень отвечает за интерфейс пользователя и взаимодействие с ним. Отправляет запросы к серверу приложений и отображает результаты пользователю. Компоненты на клиентском уровне.

Интерфейс пользователя (UI) – обеспечивает взаимодействие пользователя с системой.

Клиентская логика (Client Logic) – отвечает за обработку бизнес-логики на стороне клиента, включая валидацию данных, управление состоянием приложения и обработку пользовательских событий.

Клиентская библиотека для работы с API (API Client Library) – управляет взаимодействием с сервером приложений через API, отправляет запросы и обрабатывает ответы, включая аутентификацию.

Уровень бизнес-логики (Сервер приложений) содержит функциональность приложения, обрабатывает запросы от клиентского уровня, взаимодействует с базой данных на серверном уровне и обеспечивает функционал приложения. Компоненты на уровне бизнес-логики.

API – отвечает за обработку запросов от клиентского уровня и взаимодействие с бизнес-логикой, обеспечивает интерфейс для коммуникации между клиентом и сервером.

Бизнес-логика (Business Logic) – реализует основную бизнес-логику приложения.

Сервис обработки данных (Data Processing Service) – отвечает за получение, обработку и подготовку данных для анализа, может взаимодействовать как с API, так и с сервером базы данных для получения необходимых данных.

Уровень данных (Сервер базы данных) отвечает за хранение данных о клиентах и предоставление доступа к информации из базы данных. Выполняет обработку запросов на доступ к данным, поступающих от сервера приложений, и возвращает результаты. В рамках разрабатываемого ПО, серверный уровень будет содержать базу данных для хранения информации и

результаты анализа. Обеспечивает быстрый доступ к данным и их надежное хранение. Принята СУБД MySQL.

Представленная архитектура позволяет легко масштабировать систему, добавляя или изменяя компоненты каждого уровня без нарушений в работе всей системы. Разделение функций на уровни упрощает разработку, позволяет специализироваться на конкретных задачах и обеспечивает более простую поддержку и обновление кода, гарантирует высокую скорость работы ПО. Диаграмма развертывания, приведена на рисунке 14.

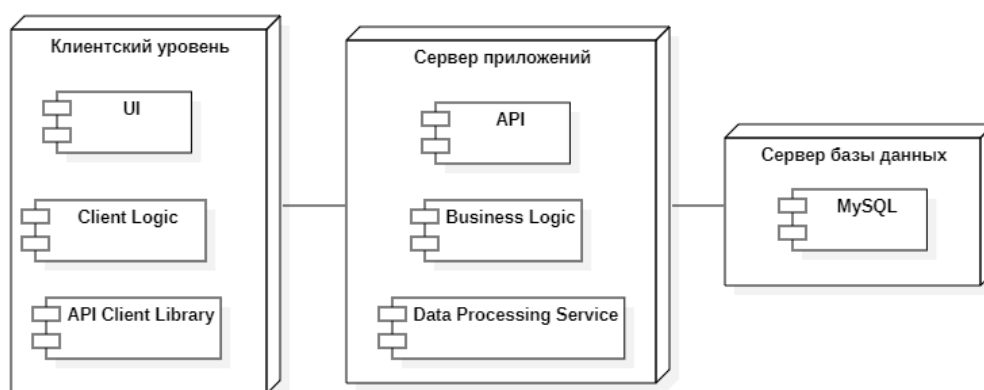


Рисунок 14 – Диаграмма развертывания

Диаграмма визуализирует распределение компонентов АИС и их взаимодействие.

### **3.2 Выбор технологии разработки программного обеспечения автоматизированной информационной системы**

Для разработки программного обеспечения на macOS выбираем интегрированную среду разработки (IDE) Xcode. Она поддерживает языки программирования Swift и Objective-C и предлагает широкий набор инструментов для разработки, отладки и тестирования приложений. Xcode является бесплатно распространяемым и хорошо интегрирован с экосистемой Apple, что упрощает создание приложений для iOS и macOS [11, 24].

Для создания пользовательского интерфейса используем современный фреймворк SwiftUI, который обеспечивает декларативный подход, удобство работы с компонентами интерфейса и автоматическое обновление представлений при изменении данных. SwiftUI позволяет создавать кроссплатформенные приложения, адаптируемые для различных устройств Apple. Для реализации логики приложения будет использован Swift – простотой, безопасный и высокопроизводительный язык программирования.

Выбор Xcode, SwiftUI и языка Swift в сочетании с четко определенной архитектурой обеспечивает создание современного и масштабируемого ПО для анализа и прогнозирования поведения потребителей.

### **3.3 Выбор СУБД автоматизированной информационной системы**

Система управления базами данных отвечает за управление хранением данных, преобразование и представление данных, управление безопасностью. Правильный выбор СУБД влияет на производительность, масштабируемость, надежность и гибкость системы [5]. Рассмотрим наиболее популярные СУБД – MySQL, PostgreSQL и Microsoft SQL Server.

MySQL – это быстрая, надежная, открыто распространяемая СУБД, функционирует как клиент-сервер, имеет высокую производительность при работе с большими объемами данных и высокими нагрузками на чтение и запись [13]. Обладает высокой скоростью выполнения запросов и простотой настройки. Используется в веб-приложениях, средних или крупных проектах.

PostgreSQL – продвинутая СУБД с открытым исходным кодом, расширенной функциональностью. Поддерживает стандарты SQL и ACID. Идеально подходит для сложных и критически важных приложений. Обладает высокой надежностью и высокой производительностью, поддерживает сложные запросы и транзакционные нагрузки. Предоставляет расширенные возможности для работы с данными и поддержку множества типов данных.

Microsoft SQL Server – корпоративная система управления базами данных, разработанная компанией Microsoft, также поддерживает сложные

запросы и транзакционные нагрузки. Известна своей высокой производительностью, интеграцией с другими продуктами Microsoft и мощными инструментами для анализа данных.

В таблице 5 приведено сравнение рассмотренных выше СУБД.

Таблица 5 – Сравнение характеристик СУБД

| Характеристики                    | MySQL                                     | PostgreSQL                           | Microsoft SQL Server                           |
|-----------------------------------|-------------------------------------------|--------------------------------------|------------------------------------------------|
| Производительность                | +                                         | +                                    | +                                              |
| Надежность и отказоустойчивость   | +                                         | +                                    | +                                              |
| Кроссплатформенность              | Linux, Windows, macOS                     | Linux, Windows, macOS                | Linux, Windows                                 |
| Лицензирование и стоимость        | Бесплатные версии, открытая лицензия, GPL | Бесплатные версии, открытая лицензия | коммерческая лицензия                          |
| Установка и настройка             | ++, существуют готовые сборки             | + сложнее, доп. ресурсы              | – сложная                                      |
| Наличие документации и руководств | ++                                        | ++                                   | + сложна для новичков                          |
| Сообщество и поддержка            | ++                                        | ++                                   | + ориентировано на корпоративных пользователей |
| Простота использования            | ++                                        | +                                    | –                                              |

В качестве СУБД АИС выбрана MySQL. MySQL обеспечивает высокую производительность, легка в настройке и использовании, обладает механизмами резервирования данных и восстановления после сбоев, обеспечивает высокую отказоустойчивость. Бесплатная версия с открытым исходным кодом дает широкие возможности для разработки и использования без дополнительных затрат. Обширное сообщество пользователей, большое количество учебных материалов позволяют быстро найти решения. Это делает MySQL оптимальным выбором, обеспечивая надежную платформу для хранения и управления данными.

### 3.4 Разработка физической модели данных автоматизированной информационной системы

После разработки логической модели данных и выбора системы управления базами данных MySQL, переходим к разработке физической модели данных. Разработка физической модели представляет собой конкретизацию логической модели, при этом учитываются технические особенности выбранной СУБД. На данном этапе необходимо определить структуры таблиц, типы данных, ключи, связи между таблицами, хранимые процедуры. MySQL предлагает ряд специфических типов данных и механизмов, таких как автоинкрементные поля для первичных ключей, поддержка внешних ключей для обеспечения целостности данных. Физическая модель данных приведена на рисунке 15.

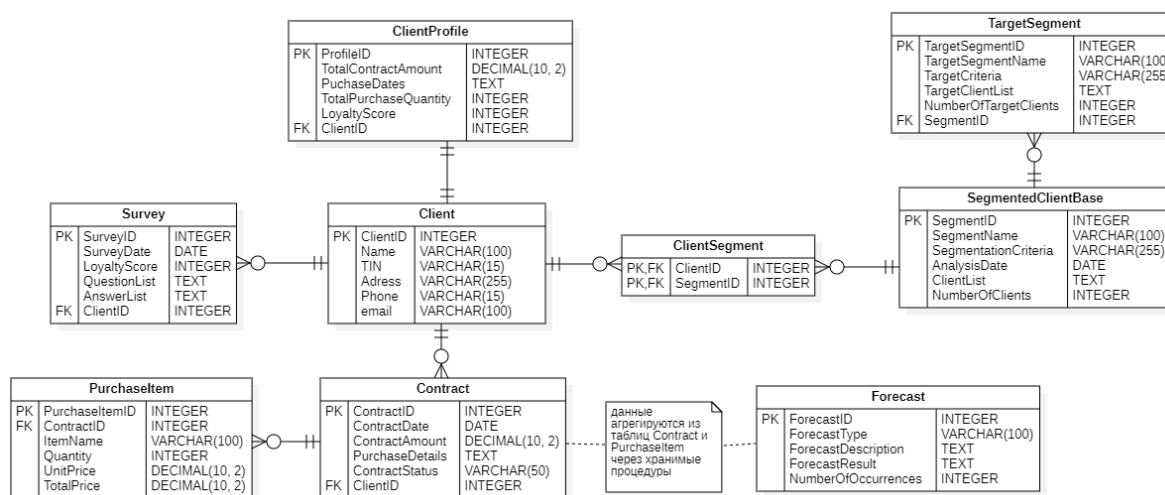


Рисунок 15 – Физическая модель данных

Модель данных содержит следующие таблицы-сущности:

- Client – содержит информацию о клиентах;
- Contract – содержит информацию о договорах;
- PurchaseItem – содержит информацию о каждой позиции покупки;
- Survey – содержит данные о проведенных опросах клиентов;
- SegmentedClientBase – информацию о сегментах клиентской базы;
- ClientSegment – промежуточная таблица;

- ClientProfile – профиль Клиента – обобщенная информация о клиенте;
- TargetSegment – содержит информацию о целевых сегментах;
- Forecast – содержит информацию о прогнозах.

Структуры сущностей представлены в таблицах 6 – 14.

Таблица 6 – Структура таблицы Client

| Наименование столбца | Тип данных   | Ключи                       | Описание                                        |
|----------------------|--------------|-----------------------------|-------------------------------------------------|
| ClientID             | int          | PRIMARY KEY, AUTO_INCREMENT | Уникальный идентификатор клиента                |
| Name                 | varchar(100) | -                           | Имя клиента                                     |
| TIN                  | varchar(15)  | -                           | Идентификационный номер налогоплательщика (ИНН) |
| Address              | varchar(255) | -                           | Адрес клиента                                   |
| Phone                | varchar(15)  | -                           | Телефонный номер клиента                        |
| Email                | varchar(100) | -                           | Электронная почта клиента                       |

Таблица 7 – Структура таблицы Contract

| Наименование столбца | Тип данных     | Ключи                                    | Описание                          |
|----------------------|----------------|------------------------------------------|-----------------------------------|
| ContractID           | int            | PRIMARY KEY, AUTO_INCREMENT              | Уникальный идентификатор договора |
| ContractDate         | Date           | -                                        | Дата заключения договора          |
| ContractAmount       | decimal(10, 2) | -                                        | Сумма договора                    |
| PurchaseDetails      | Text           | -                                        | Детали покупки                    |
| ContractStatus       | varchar(50)    | -                                        | Статус договора                   |
| ClientID             | int            | FOREIGN KEY REFERENCES Client(client_id) | Идентификатор клиента             |

Таблица 8 – Структура таблицы PurchaseItem

| Наименование столбца | Тип данных     | Ключи                                        | Описание                                                                      |
|----------------------|----------------|----------------------------------------------|-------------------------------------------------------------------------------|
| PurchaseItemID       | int            | PRIMARY KEY, AUTO_INCREMENT                  | Уникальный идентификатор позиции покупки                                      |
| ContractID           | int            | FOREIGN KEY REFERENCES Contract(contract_id) | Идентификатор договора                                                        |
| ItemName             | varchar(100)   | -                                            | Наименование товара услуги                                                    |
| Quantity             | int            | -                                            | Количество                                                                    |
| UnitPrice            | decimal(10, 2) | -                                            | Цена за единицу                                                               |
| TotalPrice           | decimal(10, 2) | -                                            | Общая стоимость позиции<br>GENERATED ALWAYS AS (quantity * unit_price) STORED |

Таблица 9 – Структура таблицы Survey

| Наименование столбца | Тип данных | Ключи                                    | Описание                        |
|----------------------|------------|------------------------------------------|---------------------------------|
| SurveyID             | int        | PRIMARY KEY, AUTO_INCREMENT              | Уникальный идентификатор опроса |
| SurveyDate           | Date       | -                                        | Дата проведения опроса          |
| LoyaltyScore         | int        | -                                        | Оценка удовлетворенности        |
| QuestionList         | text       | -                                        | Список вопросов                 |
| AnswerList           | text       | -                                        | Список ответов                  |
| ClientID             | int        | FOREIGN KEY REFERENCES Client(client_id) | Идентификатор клиента           |

Таблица 10 – Структура таблицы SegmentedClientBase

| Наименование столбца | Тип данных   | Ключи                       | Описание                          |
|----------------------|--------------|-----------------------------|-----------------------------------|
| SegmentID            | int          | PRIMARY KEY, AUTO_INCREMENT | Уникальный идентификатор сегмента |
| SegmentName          | varchar(100) | -                           | Название сегмента                 |
| SegmentationCriteria | varchar(255) | -                           | Критерии сегментации              |
| AnalysisDate         | Date         | -                           | Дата анализа                      |
| ClientList           | text         | -                           | Список клиентов в сегменте        |
| NumberOfClients      | int          | -                           | Кол-во клиентов в сегменте        |

Таблица 11 – Структура промежуточной таблицы ClientSegment

| Наименование столбца | Тип данных | Ключи                                                                 | Описание                          |
|----------------------|------------|-----------------------------------------------------------------------|-----------------------------------|
| SegmentID            | int        | FOREIGN KEY REFERENCES SegmentedClientBase(segment_id)<br>PRIMARY KEY | Уникальный идентификатор сегмента |
| ClientID             | int        | FOREIGN KEY REFERENCES Client(client_id)<br>PRIMARY KEY               | Идентификатор клиента             |

Стандартным подходом для реализации связи many-to-many в реляционных базах данных MySQL между двумя таблицами является использование промежуточной таблицы. Для этих целей создана промежуточная таблица ClientSegment, которая будет содержать внешние ключи SegmentID и ClientID, ссылающиеся на Client и SegmentedClientBase. Каждый клиент может принадлежать к нескольким сегментам, а каждый сегмент может содержать несколько клиентов. Использован составной первичный ключ, состоящий из двух столбцов: segment\_id и client\_id. Это

обеспечивает уникальность каждой записи в таблице ClientSegment, то есть, один и тот же клиент не может быть дважды включен в один и тот же сегмент.

Таблица 12 – Структура таблицы ClientProfile

| Наименование столбца  | Тип данных     | Ключи                                    | Описание                         |
|-----------------------|----------------|------------------------------------------|----------------------------------|
| ProfileID             | int            | PRIMARY KEY, AUTO_INCREMENT              | Уникальный идентификатор профиля |
| PurchaseDates         | text           | -                                        | Даты покупок                     |
| TotalContractAmount   | decimal(10, 2) | -                                        | Общая сумма покупок              |
| TotalPurchaseQuantity | int            | -                                        | Частота покупок                  |
| LoyaltyScore          | int            | -                                        | Оценка лояльности                |
| ClientID              | int            | FOREIGN KEY REFERENCES Client(client_id) | Идентификатор клиента            |

Таблица 13 – Структура таблицы TargetSegment

| Наименование столбца  | Тип данных   | Ключи                                                   | Описание                                   |
|-----------------------|--------------|---------------------------------------------------------|--------------------------------------------|
| TargetSegmentID       | int          | PRIMARY KEY, AUTO_INCREMENT                             | Уникальный идентификатор целевого сегмента |
| TargetSegmentName     | varchar(100) | -                                                       | Название сегмента                          |
| TargetCriteria        | varchar(255) | -                                                       | Критерии сегментации                       |
| TargetClientList      | text         | -                                                       | Список клиентов сегмента                   |
| NumberOfTargetClients | int          | -                                                       | Количество клиентов в сегменте             |
| SegmentID             | int          | FOREIGN KEY REFERENCES SegmentedClientBase (segment_id) | Уникальный идентификатор сегмента          |

Таблица 14 – Структура таблицы Forecast

| Наименование столбца  | Тип данных   | Ключи                       | Описание                                            |
|-----------------------|--------------|-----------------------------|-----------------------------------------------------|
| ForecastID            | int          | PRIMARY KEY, AUTO_INCREMENT | Уникальный идентификатор прогноза                   |
| ForecastType          | varchar(100) | -                           | Указывает тип прогноза                              |
| ForecastDescription   | text         | -                           | Содержит описание прогноза                          |
| ForecastResult        | text         | -                           | Содержит результат прогноза                         |
| Number Of Occurrences | int          | -                           | Количество случаев, когда товары приобретены вместе |

Таблица Forecast будет наполняться данными с использованием хранимых процедур, которые агрегируют данные из таблиц Contract и PurchaseItem. Связь осуществляется логически через хранимые процедуры, поэтому внешние ключи не требуются. Данные из таблиц Contract и PurchaseItem собираются и обрабатываются хранимыми процедурами, которые затем вставляют результаты в таблицу Forecast.

SQL-запросы и хранимые процедуры для создания таблиц базы данных представлены в Приложении А.

### **3.5 Разработка программного обеспечения автоматизированной информационной системы**

На рисунке 16 представлено дерево функций разрабатываемой АИС. Дерево позволяет визуализировать иерархию основных и служебных функций, каждый блок включает логически и функционально объединенные операции, формируя единую структуру [7].

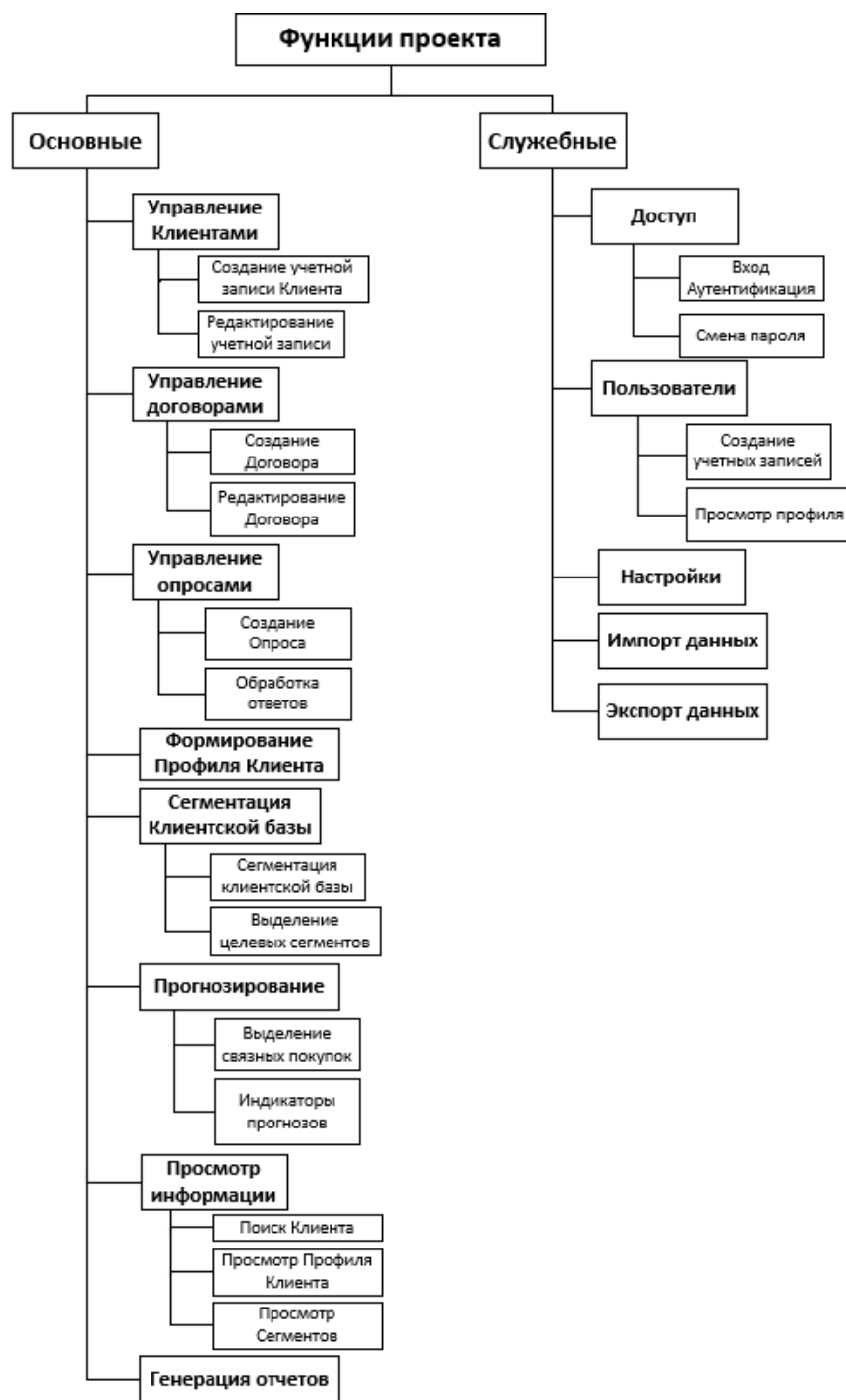


Рисунок 16 – Дерево функций

На основе Древа функций разработана структура сценария диалога типа Меню. Сценарий включает в себя основные разделы: «Справочники», «Управление», «Анализ» и «Выход». Каждый раздел содержит подменю, позволяющие пользователям легко ориентироваться и получать доступ к необходимой информации. Сценарий диалога представлен на рисунке 17.

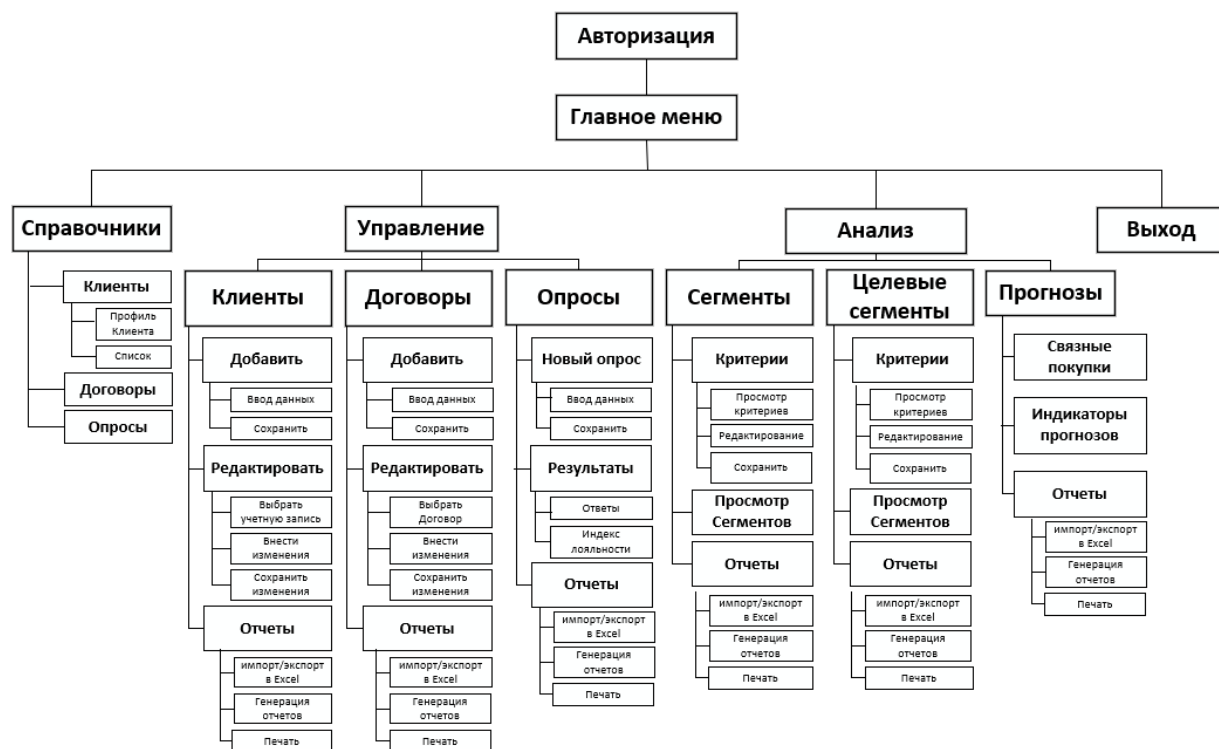


Рисунок 17 – Сценарий диалога

На основе полученных результатов, построено дерево программных модулей – структурная схема, отражающая взаимосвязи и подчиненность различных модулей программного обеспечения. Представленные программные модули можно разделить на выполняющие служебные функции, управляющие модули и функциональные модули обработки данных [7]. Дерево программных модулей представлено на рисунке 18.

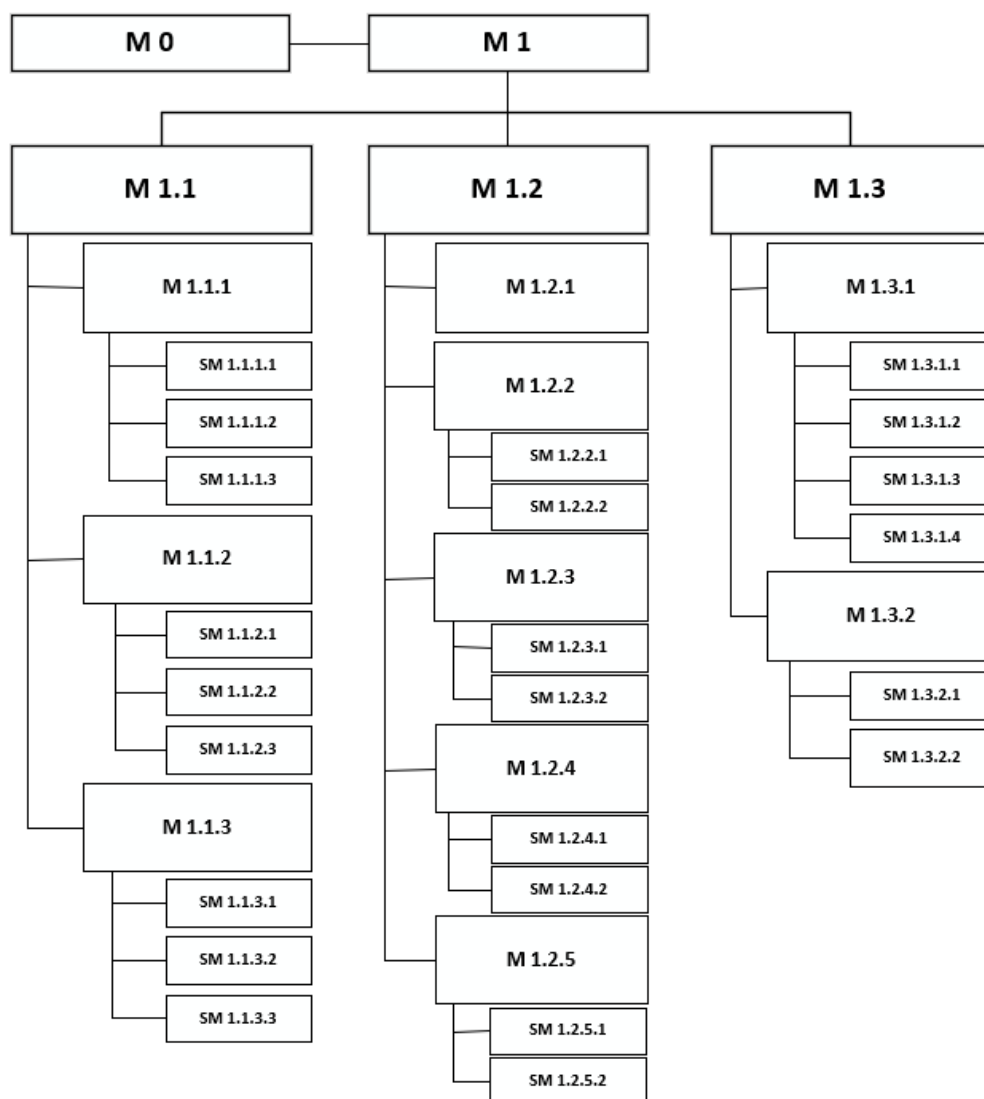


Рисунок 18 – Дерево программных модулей

Описание программных модулей представлено в таблице 15.

Таблица 15 – Описание программных модулей

| Идентификатор модуля | Функции модуля                                                                              |
|----------------------|---------------------------------------------------------------------------------------------|
| M 0                  | Авторизация. Модуль для проверки и управления доступом пользователей. Служебный модуль.     |
| M 1                  | Менеджер главного меню. Центральный модуль, запускает остальные модули. Управляющий модуль. |
| M 1.1                | Модуль, формирующий подменю «Управление данными». Управляющий модуль.                       |
| M 1.1.1              | Модуль управления (ввода/редактирования) данными о Клиентах. Функциональный модуль.         |

Продолжение таблицы 15

| Идентификатор модуля | Функции модуля                                                                                                                                                                            |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SM 1.1.1.1           | Подмодуль ввода. Предоставляет форму для ввода данных о клиентах.                                                                                                                         |
| SM 1.1.1.2           | Подмодуль редактирования. Позволяет изменять существующую информацию о клиентах.                                                                                                          |
| SM 1.1.1.3           | Подмодуль проверки и хранения. Обеспечивает проверку введенных данных на корректность и отвечает за взаимодействия с БД для сохранения введенных или отредактированных данных о клиентах. |
| M 1.1.2              | Модуль управления (ввода/редактирования) данными о Договорах. Функциональный модуль.                                                                                                      |
| SM 1.1.2.1           | Подмодуль ввода. Форма для ввода данных о Договорах.                                                                                                                                      |
| SM 1.1.2.2           | Подмодуль редактирования. Позволяет изменять существующую информацию в Договорах.                                                                                                         |
| SM 1.1.2.3           | Подмодуль проверки и хранения. Обеспечивает проверку введенных данных на корректность и отвечает за взаимодействия с БД для сохранения введенных или отредактированных данных Договоров.  |
| M 1.1.3              | Модуль управления (ввода/редактирования) данными о Опросах. Функциональный модуль.                                                                                                        |
| SM 1.1.3.1           | Подмодуль ввода. Предоставляет форму для ввода данных Опросов.                                                                                                                            |
| SM 1.1.3.2           | Подмодуль редактирования. Позволяет изменять существующую информацию Опросов.                                                                                                             |
| SM 1.1.3.3           | Подмодуль проверки и хранения. Обеспечивает проверку введенных данных на корректность и отвечает за взаимодействия с БД для сохранения введенных или отредактированных данных Опросов.    |
| M 1.2                | Модуль, формирующий подменю «Анализ». Управляющий модуль.                                                                                                                                 |
| M 1.2.1              | Модуль формирования Профиля Клиента.                                                                                                                                                      |
| M 1.2.2              | Модуль сегментации.                                                                                                                                                                       |
| SM 1.2.2.1           | Подмодуль вызывает хранимые процедуры MySQL для сегментации данных Клиентской базы.                                                                                                       |
| SM 1.2.2.2           | Подмодуль обрабатывает результаты сегментации.                                                                                                                                            |
| M 1.2.3              | Модуль выделения целевых сегментов.                                                                                                                                                       |
| SM 1.2.3.1           | Подмодуль вызывает хранимые процедуры MySQL для выделения целевых сегментов.                                                                                                              |
| SM 1.2.3.2           | Подмодуль обрабатывает результаты выделения целевых сегментов.                                                                                                                            |
| M 1.2.4              | Модуль прогнозирования связанных покупок.                                                                                                                                                 |
| SM 1.2.4.1           | Подмодуль вызывает хранимые процедуры MySQL для прогнозирования связанных покупок.                                                                                                        |
| SM 1.2.4.2           | Подмодуль обрабатывает результаты прогнозирования связанных покупок.                                                                                                                      |
| M 1.2.5              | Модуль расчета аналитических индикаторов прогнозирования.                                                                                                                                 |
| SM 1.2.5.1           | Подмодуль, формирующий аналитические индикаторы прогнозирования.                                                                                                                          |
| SM 1.2.5.2           | Подмодуль обрабатывает аналитические индикаторы прогнозирования.                                                                                                                          |
| M 1.3                | Модуль, формирующий подменю «Выдача информации». Управляющий модуль.                                                                                                                      |
| M 1.3.1              | Модуль просмотра данных. Позволяет пользователям просматривать различные типы данных.                                                                                                     |
| SM 1.3.1.1           | Подмодуль просмотра информации о клиентах.                                                                                                                                                |

## Продолжение таблицы 15

|            |                                                                                                                                  |
|------------|----------------------------------------------------------------------------------------------------------------------------------|
| SM 1.3.1.2 | Подмодуль просмотра информации по договорам. Отображает детали договоров.                                                        |
| SM 1.3.1.3 | Подмодуль просмотра опросов.                                                                                                     |
| SM 1.3.1.4 | Подмодуль просмотра сегментов пой базы.                                                                                          |
| M 1.3.2    | Модуль генерации отчетов.                                                                                                        |
| SM 1.3.2.1 | Подмодуль печати документов.                                                                                                     |
| SM 1.3.2.2 | Подмодуль экспорта/импорта в MS Excel. Экспортирует отчеты в формат Excel или импортирует данные из Excel для генерации отчетов. |

В таблице даны описания выполняемых функций для каждого модуля.

### 3.6 Описание функциональности автоматизированной информационной системы

На основе дерева функций и структуры сценария диалога, описанных в предыдущей главе разработаны интерфейсы пользователя АИС для анализа и прогнозирования поведения потребителей. Пользователями данной системы являются Менеджер по работе с клиентами и Менеджер по рекламе.

После запуска системы на экране появляется форма авторизации, представленная на рисунке 19.

**Авторизация**

Логин

Пароль

☐ Запомнить пароль

Войти

[Восстановить пароль](#)

Рисунок 19 – Форма авторизации

После успешного ввода Логина и Пароля, пользователь попадает на главную страницу АИС (рисунок 20), на которой отображены главное меню и

рабочая область. Главное меню системы состоит из трех основных пунктов: «Справочники», «Управление» и «Анализ». Каждый пункт меню позволяет пользователю переключаться между разными разделами приложения. Кнопка выхода позволяет пользователю выйти из приложения и вернуться на экран авторизации. Фрагменты кода для реализации модуля «Авторизация» и «Менеджер главного меню» представлены в приложении Г.

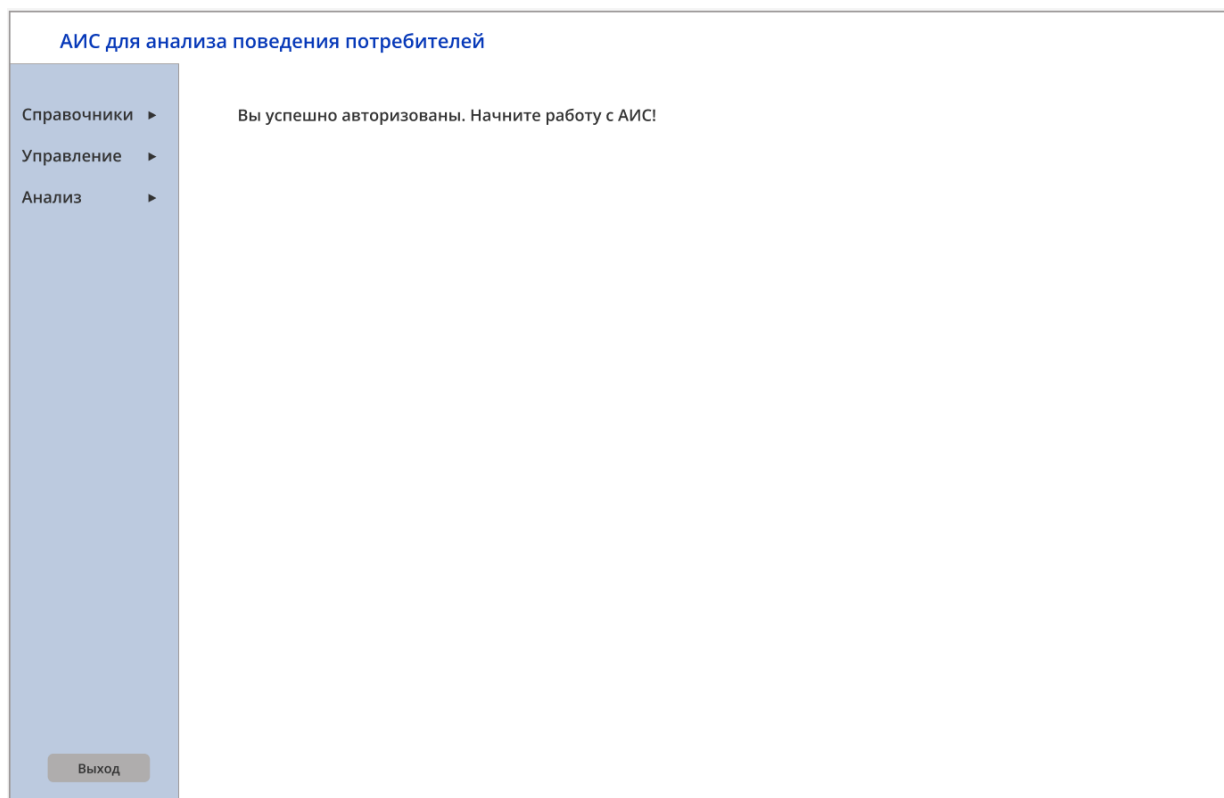


Рисунок 20 – Главная форма

Раздел «Справочники» используется для быстрого просмотра данных без необходимости их добавления или редактирования. Доступны подпункты «Клиенты», «Договоры», «Опросы» и «Сегменты».

При активации раздела «Управление» раскрываются подразделы «Клиенты», «Договоры» и «Опросы», которые позволяют добавить, отредактировать данные или сформировать отчет по каждому подразделу. Форма добавления нового Клиента представлена на рисунке 21. Форма добавления нового Договора представлена на рисунке 22.

**АИС для анализа поведения потребителей**

Справочники ▶  
Управление ▶  
Клиенты  
Добавить  
Редактировать  
Отчеты  
Договоры  
Опросы  
Анализ ▶

Наименование компании

Основная информация

ИНН

Адрес

Юридический адрес

Телефон

Email

Руководитель: ФИО

Должность:

Телефон

Email

Контактное лицо: ФИО

Должность:

Телефон

Email

Примечания:

Выход
Сохранить

Рисунок 21 – Форма добавления нового Клиента

**АИС для анализа поведения потребителей**

Справочники ▶  
Управление ▶  
Клиенты  
Договоры  
Добавить  
Редактировать  
Отчеты  
Опросы  
Анализ ▶

Введите наименование Клиента для заполнения данных о компании

Информация о компании Клиента

ИНН

Адрес

Юридический адрес

Телефон

Email

Руководитель:

Номер Договора:

Дата начала:

Дата окончания:

Позиции Договора:

Контактное лицо:

Телефон

Email

Комментарий:

Выход
Сохранить

Рисунок 22 – Форма добавления нового Договора

При активации раздела «Анализ» раскрываются подразделы «Сегменты», «Целевые сегменты» и «Прогнозы».

Подраздел «Критерии» предназначен для выбора параметров сегментации клиентов на основе объема контрактов, частоты заказов и индекса лояльности. Подраздел «Просмотр сегментов» позволяет просматривать результаты произведенной сегментации клиентской базы на основе заданных критериев. Пользователь выбирает необходимый сегмент для просмотра и в рабочей области отображаются название сегмента, критерий и таблица с информацией о клиентах, которая включает наименование клиента, его данные, информацию по контрактам и индекс лояльности. Таблицу можно сортировать по дате последнего контракта, количеству контрактов индексу лояльности. Пользователи могут переключаться между сегментами и видеть изменения в списке клиентов. Интерфейс страницы подраздела «Просмотр сегментов» представлен на рисунке 23.

АИС для анализа поведения потребителей

Справочники

Управление

Анализ

Сегменты

Критерии

Просмотр сегментов

Целевые сегменты

Прогнозы

Выход

Выберите сегмент клиентской базы

Сегменты клиентов по объему заказов:

Сегменты клиентов по частоте договоров:

Сегменты клиентов по индексу лояльности NPS:

Крупные заказы

Средние заказы

Малые заказы

Постоянное сотрудничество

Повторный заказ

Разовый заказ

Промоутеры

Нейтралы

Критики

Применить

Крупные заказы: ContractAmount >= 500000

Упорядочить по

| <input type="checkbox"/> | Наименование        | ИНН         | Email            | Общая сумма заказов | Количество контрактов | Дата последнего контракта | Индекс лояльности | Перейти в Профиль клиента |
|--------------------------|---------------------|-------------|------------------|---------------------|-----------------------|---------------------------|-------------------|---------------------------|
| <input type="checkbox"/> | ООО "Альфа плюс"    | 15278936061 | alfamc@yandex.ru | 62660               | 3                     | 25.07.23                  | 7                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 630120              | 2                     | 11.02.24                  | 9                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 612300              | 4                     | 25.07.23                  | 7                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 530000              | 2                     | 05.07.23                  | 6                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 587000              | 2                     | 15.12.22                  | 9                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 530000              | 2                     | 25.12.23                  | 5                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 530010              | 2                     | 25.07.23                  | 8                 | Перейти                   |
| <input type="checkbox"/> | ООО "Алгоритм плюс" | 15278936061 | algh@yandex.ru   | 507200              | 2                     | 05.05.21                  | 7                 | Перейти                   |
|                          | Итого по сегменту   |             |                  | 5097200             | 27                    |                           |                   |                           |

Сохранить в Excel

Сформировать отчет

Рисунок 23 – Подраздел «Просмотр сегментов»

При активации раздела «Прогнозы» раскрываются подразделы «Связные покупки» и «Аналитические индикаторы прогнозирования».

Подраздел «Связные покупки» отображает информацию о продуктах и услугах, приобретаемых вместе на основе заключенных контрактов. Пользователю необходимо выбрать период для анализа. Результаты отображаются в таблице, содержащей информацию об основном и сопутствующем продуктах, количестве контрактов, в которых оба продукта были приобретены. Информацию можно отобразить графически. Интерфейс страницы подраздела «Связные покупки» представлен на рисунке 24.

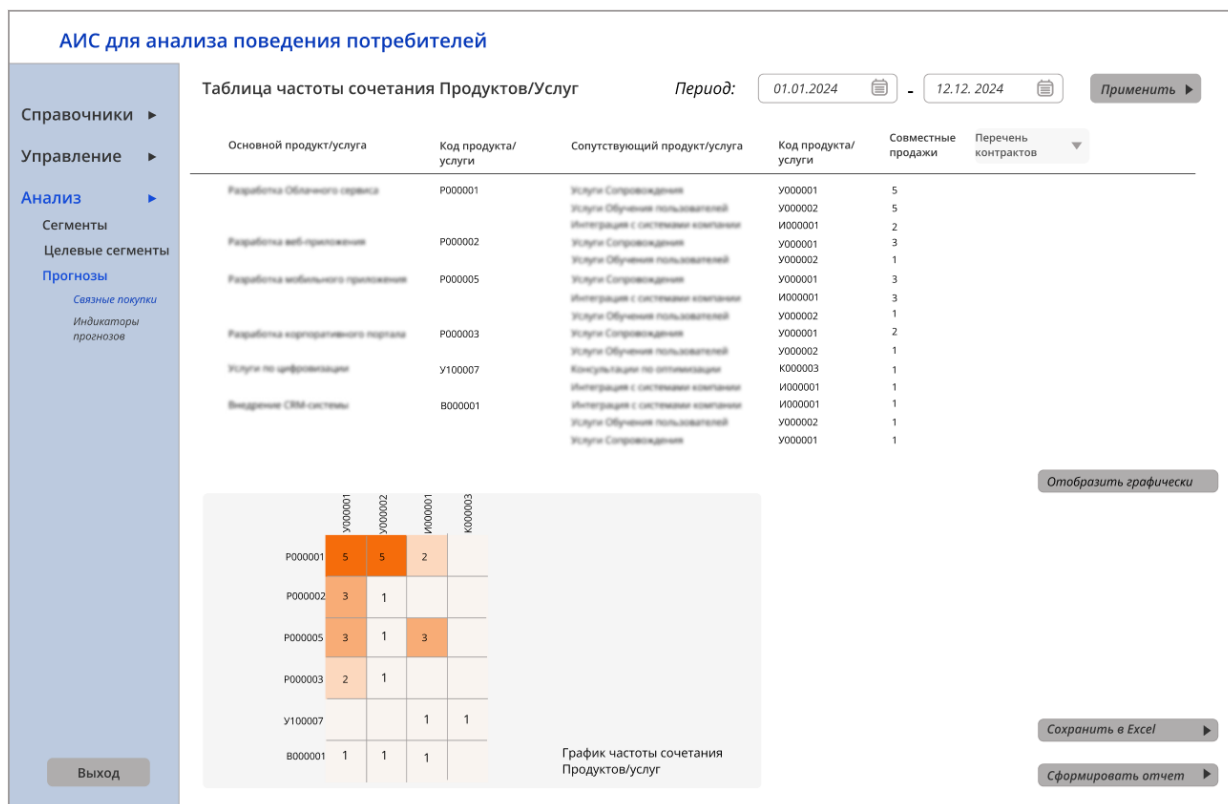


Рисунок 24 – Подраздела «Связные покупки»

Интерфейс страницы подраздела «Индикаторы прогнозов» представлен на рисунке 25.

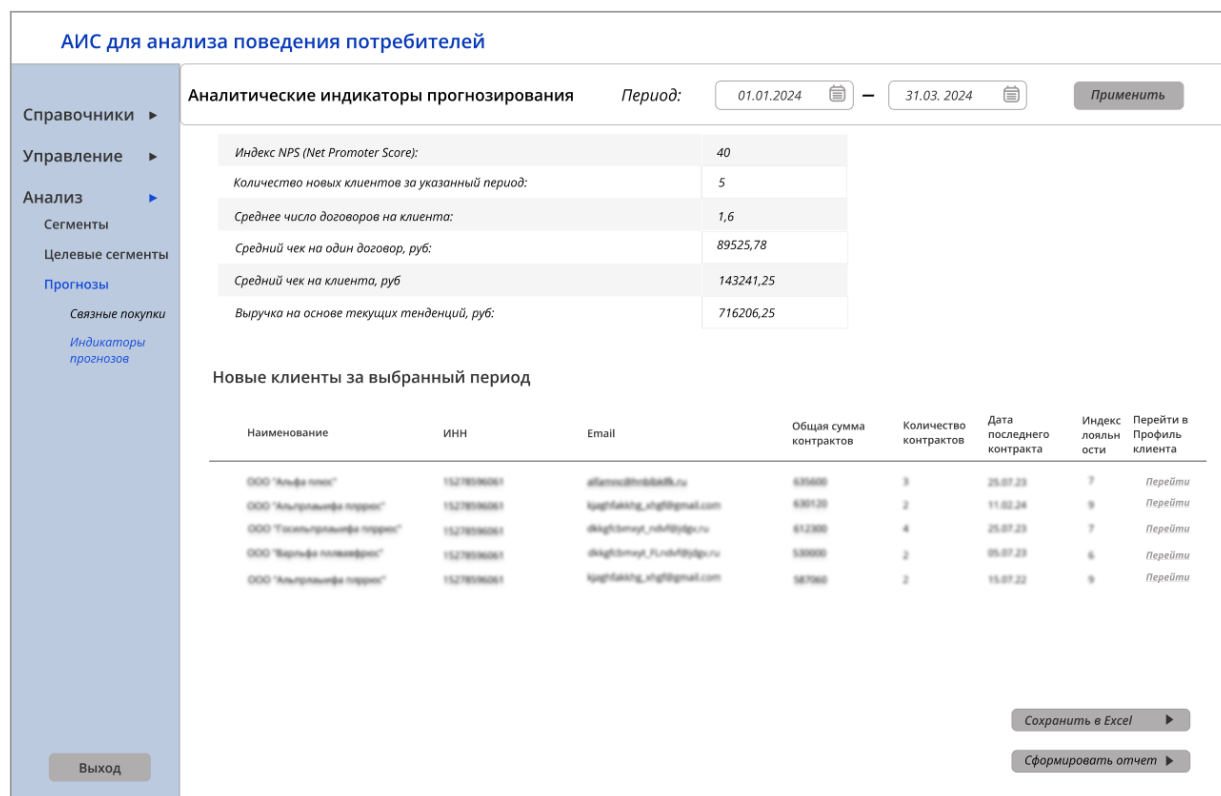


Рисунок 25 – Подраздел «Индикаторы прогнозов»

Подраздел «Индикаторы прогнозов» позволяет пользователю выбрать период для анализа и получить обзор ключевых показателей для прогнозирования будущих тенденций.

### 3.7 Тестирование программного проекта

SWEBOK определяет тестирование программного обеспечения как процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Для проверки соответствия ПО функциональным требованиям выполняется Функциональное тестирование. Цель функционального тестирования – убедиться, что система выполняет все заявленные функции и каждая функция работает корректно при заданных входных данных и генерирует ожидаемые выходные данные [9].

Для разработанного ПО функциональное тестирование будет проводиться с использованием методов черного ящика. Метод черного ящика позволяет сосредоточиться на внешнем поведении программы, проверяя, соответствуют ли ее функции заявленным требованиям без учета того, как они реализованы внутри. В рамках тестирования по методу черного ящика основной информацией для создания тестовых сценариев выступают требования. Тестовые сценарии для проверки функциональности разработанного ПО приведены в Приложении Б.

Для проверки покрытия функциональных требований проекта тестовыми сценариями заполнена Матрица трассируемости требований (Traceability matrix), представлена в Приложении В. Приведенная матрица продемонстрировала, что разработанные тестовые сценарии обеспечили полное покрытие всех функциональных требований проекта. Это гарантирует, что программное обеспечение соответствует всем необходимым функциональным критериям и готово к дальнейшим этапам внедрения.

### **3.8 Оценка и обоснование экономической эффективности разработки автоматизированной информационной системы**

Для обоснования экономической эффективности разработки используем методику расчета прямой эффективности от внедрения информационной системы по сравнению с базовым вариантом обработки информации. Для этого выполним сопоставление показателей существующего и внедряемого технологических процессов (базового и проектного вариантов).

Экономическая эффективность разработки складывается из:

- косвенного эффекта, выражаемого в увеличении прибыли и привлечении большего числа клиентов;
- прямой эффект, проявляющийся в снижении трудовых, стоимостных показателей за счет сокращения времени обработки и получения

данных, сокращения трудоемкости работы и стоимостных затрат обработки документов и повышении точности информации [7].

Для расчета прямого эффекта определим трудовые показатели:

Абсолютное снижение трудовых затрат  $\Delta T$ , час

$$\Delta T = T_{\delta} - T_n, \quad (1)$$

где  $T_{\delta}$  – базовое время выполнения операций до автоматизации, час;

$T_n$  – проектное время выполнения после автоматизации, час.

Коэффициент относительного снижения трудовых затрат  $K_T$ , %:

$$K_T = (\Delta T / T_{\delta}) \cdot 100\%, \quad (2)$$

Индекс снижения трудовых затрат  $Y_T$ :

$$Y_T = T_{\delta} / T_n. \quad (3)$$

И стоимостные показатели:

Абсолютное снижение стоимостных затрат  $\Delta C$ , руб:

$$\Delta C = C_{\delta} - C_n \quad (4)$$

где  $C_{\delta}$  – базовые затраты выполнения операций до автоматизации, руб;

$C_n$  – проектные затраты выполнения после автоматизации, руб.

Коэффициент относительного снижения затрат  $K_C$ , %:

$$K_C = (\Delta C / C_{\delta}) \cdot 100\%; \quad (5)$$

Индекс снижения затрат  $Y_C$ :

$$Y_C = C_{\delta} / C_n \quad (6)$$

Показатели эффективности от внедрения разработанного приложения рассчитаны и представлены в таблице 16.

Таблица 16 – Показатели эффективности

| Показатели                                                                                                                                                                                                                                                                                    | Затраты           |                   | Абсолютное изменение затрат        | Коэффициент изменения затрат         | Индекс изменения затрат |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-------------------|------------------------------------|--------------------------------------|-------------------------|
|                                                                                                                                                                                                                                                                                               | Базовый вариант   | Проектный вариант |                                    |                                      |                         |
| Трудовые                                                                                                                                                                                                                                                                                      | $T_0$ , час/мес.  | $T_n$ , час/мес.  | $\Delta T = T_0 - T_n$ , час/мес.  | $K_T = (\Delta T / T_0) \cdot 100\%$ | $Y_T = T_0 / T_n$       |
| Ввод данных                                                                                                                                                                                                                                                                                   | 20                | 20                | 0                                  | 0                                    | 1                       |
| Сегментация клиентской базы                                                                                                                                                                                                                                                                   | 8                 | 1                 | 7                                  | 87.5%                                | 8                       |
| Выделение целевых сегментов                                                                                                                                                                                                                                                                   | 8                 | 1                 | 7                                  | 87.5%                                | 8                       |
| Расчет индекса NPS                                                                                                                                                                                                                                                                            | 2                 | 0.5               | 1.5                                | 75%                                  | 4                       |
| Поиск данных                                                                                                                                                                                                                                                                                  | 10                | 2                 | 8                                  | 80%                                  | 5                       |
| Формирование отчетов                                                                                                                                                                                                                                                                          | 16                | 4                 | 12                                 | 75%                                  | 4                       |
| Расчет индикаторов                                                                                                                                                                                                                                                                            | 4                 | 0,5               | 3.5                                | 87.5%                                | 8                       |
| Анализ связанных покупок                                                                                                                                                                                                                                                                      | 4                 | 1                 | 3                                  | 75%                                  | 4                       |
| Итого                                                                                                                                                                                                                                                                                         | 72                | 30                | 42                                 | 58.3%                                | 2.4                     |
| Стоимостные                                                                                                                                                                                                                                                                                   | $C_0$ , руб./мес. | $C_n$ , руб./мес. | $\Delta C = C_0 - C_n$ , руб./мес. | $K_C = (\Delta C / C_0) \cdot 100\%$ | $Y_C = C_0 / C_n$       |
| Фонд оплаты труда                                                                                                                                                                                                                                                                             | 38 025            | 15 844            | 22 181                             | 58.3%                                | 2.4                     |
| Стоимость обслуживания                                                                                                                                                                                                                                                                        | 0                 | 5 000             | -5 000                             | —                                    | 0                       |
| Инфраструктурные расходы                                                                                                                                                                                                                                                                      | 10 000            | 4000              | 6000                               | 60%                                  | 2.5                     |
| Итого                                                                                                                                                                                                                                                                                         | 48 025            | 24 844            | 23 181                             | 48,3%                                | 1.93                    |
| Примечания<br>1 Для расчёта фонда оплаты труда зарплату менеджеров принимаем по 65 000 рублей в месяц, с налогами и отчислениями 169000 руб. в месяц.<br>2 В расчете используются суммы пропорционально времени, затрачиваемому на операции анализа и прогнозирования поведения потребителей. |                   |                   |                                    |                                      |                         |

Как показывают расчеты, после автоматизации операции анализа поведения потребителей выполняются в 2.4 раза быстрее, трудозатраты уменьшились на 58%. Автоматизация позволила сократить расходы на выполнение операций почти на 50%. Это хороший результат, который свидетельствует об эффективности внедрения ПО и оптимизации процессов.

Срок окупаемости затрат на внедрение проекта  $T_{OK}$ , мес.:

$$T_{OK} = K_{\Pi} / \Delta C, \quad (7)$$

где  $K_{\Pi}$  – капитальные затраты на реализацию проекта, руб.

Капитальные затраты складываются из материальных затрат, затрат на оплату труда, амортизационных отчислений и прочих расходов.

Материальные затраты включают стоимость приобретения необходимых компьютеров, оборудования и программного обеспечения. Все необходимое специализированное оборудование, используемое для разработки, имеется в компании и дополнительные затраты не требуются.

Затраты на электроэнергию  $Z_{эл}$ , руб.:

$$Z_{эл} = P \cdot C_{эл} \cdot T_u, \quad (8)$$

где  $P$  – потребляемая мощность оборудования, кВт/ч;

$C_{эл}$  – стоимость 1 кВт/ч, руб.;

$T_u$  – время использования оборудования, ч.

Для работы используется компьютер потребляемой мощностью 360 Вт. Время его работы составляет 8 часов в день, продолжительность 60 дней. Стоимость 1 кВт электроэнергии составляет – 8.54 руб./кВт.

$$Z_{эл} = 0,36 \cdot 8.54 \cdot 60 \cdot 8 = 1\,475,71 \text{ руб.}$$

Рассчитаем затраты на оплату труда. В разработке ПО задействованы руководитель команды разработчиков и разработчик организации без привлечения сторонних специалистов. Для данного расчета примем, что руководитель команды разработчиков имеет ставку 100 000 рублей и затратит на работу по проекту 10 рабочих дней, разработчик имеет ставку 70 000 рублей и затратит на работу по проекту 44 рабочих дня. Заработная плата  $ЗП$  составит:

$$ЗП_{рук} = (100\,000 / 2) 10 = 45\,454,55 \text{ руб.},$$

$$ЗП_{раз} = (70\,000 / 22) 44 = 140\,000,00 \text{ руб.}$$

Фонд оплаты  $\Phi_{зн}$  труда составит:

$$\Phi_{зн} = 45\,454,55 + 140\,000,00 = 185\,454,55 \text{ руб.}$$

Перечисления 30% от дохода сотрудников на страховые взносы  $З_{стр}$ :

$$З_{стр} = 185\,454,55 \cdot 0,30 = 55\,636,36 \text{ руб.}$$

Амортизационные отчисления не учитываются так как оборудование, подлежащее амортизации не используется. Величина прочих затрат  $З_{пр}$  рассчитывается от суммы общих прямых затрат в установленном размере 10%.

Капитальные затраты проекта составят:

$$K_{\Pi} = З_{эл} + \Phi_{зн} + З_{стр} + З_{пр} \quad (9)$$

$$1\,475,71 + 185\,454,55 + 55\,636,36 + 24\,256,66 = 266\,823,28 \text{ руб.}$$

Структура затрат представлена на рисунке 26.



Рисунок 26 – Структура затрат

Срок окупаемости затрат на внедрение проекта составит:

$$T_{OK} = 266\,823,28 / 23\,181 = 11,5 \text{ месяцев.}$$

Расчеты показывают, что внедрение ПО показало свою экономическую целесообразность. Проект не только окупится за счёт ежемесячной экономии в 23181 руб. за один год, но и создаст долгосрочные конкурентные преимущества за счет повышения производительности и качества работы с клиентами.

Автоматизация минимизирует ручную обработку данных, повышая точность отчетности, а менеджеры смогут переключиться на стратегические задачи.

### Выводы по главе 3

В ходе выполнения главы «Физическое проектирование автоматизированной информационной системы» была принята трехзвенная клиент-серверная архитектура, обеспечивающая гибкость, масштабируемость системы и централизованное управление данными. В качестве СУБД определена MySQL, позволяющая эффективно реализовать необходимые функции и обеспечить надежное хранение данных. На основании выбора СУБД спроектирована физическая модель данных.

Выявление состава и иерархии функций разрабатываемой системы, построение дерева функций позволили создать четкую структуру сценария диалога и взаимодействия программных компонентов, а также послужило основой для разработки интерфейсов, обеспечивающих интуитивно понятное взаимодействие пользователей с системой.

На заключительном этапе для проверки соответствия программного обеспечения выполнено функциональное тестирование с использованием методов черного ящика. Разработанные тестовые сценарии обеспечили полное

покрытие всех функциональных требований проекта и подтвердили общую готовность системы к использованию.

Внедрение ПО показало свою экономическую целесообразность. После автоматизации операции смогут выполняться в 2.4 раза быстрее и трудозатраты уменьшатся на 58%, а расходы на выполнение операций сократятся почти на 50%.

Проект окупится за счёт ежемесячной экономии в 23181 руб. в течение одного года. Это хороший результат, который свидетельствует об эффективности внедрения ПО и оптимизации процессов.

## Заключение

Результатом представленной бакалаврской работы является разработанное ПО для анализа и прогнозирования поведения потребителей.

В ходе работы поставленные цели были достигнуты и выполнены все обозначенные задачи:

- выполнен анализ предметной области и существующие программные решения для анализа поведения потребителей;
- проанализированы текущие бизнес-процессы в компании и выявлена недостаточная автоматизация процессов анализа и сегментации клиентской базы, выделения целевых сегментов клиентов, прогнозирования поведения потребителей;
- разработаны требования к проектируемой системе и предложена усовершенствованная модель бизнес-процессов с использованием разрабатываемого ПО;
- выполнено логическое проектирование системы, описаны взаимодействие пользователей с системой через диаграмму вариантов использования, структура системы через диаграмму классов, разработана логическая модель данных;
- спроектирована трехзвенная клиент-серверная архитектура, для разработки принята интегрированная среда Xcode, язык программирования Swift, в качестве СУБД определена MySQL, посредством хранимых процедур реализованы функции сегментации и выделения целевых сегментов;
- для детализации иерархии задач построено дерево функций, реализована структура сценария диалога, сформировано дерево программных модулей и разработано программное обеспечение;
- для проверки корректности работы АИС и выполнения функционального тестирования с использованием методов «чёрного ящика» разработаны тестовые сценарии, матрица трассируемости

требований показала, что тесты покрывают все функциональные требования проекта и система является работоспособной.

Разработанная информационная система позволяет автоматизировать следующие функции:

- внесение и хранение различных типов данных о клиентах и формирования актуального Профиля клиента;
- сегментация клиентской базы и выделение целевых сегментов клиентов на основе заданных критериев;
- автоматический расчет индекса NPS на основе опросов клиентов;
- выполнение анализа связанных покупок с целью выявления ассоциаций между продуктами и услугами, расчет аналитических индикаторов для прогнозирования.

Актуальность темы анализа и прогнозирования поведения потребителей обусловлена быстрым ростом объёмов данных и необходимостью автоматизации их обработки для эффективного использования.

Автоматизация вышеперечисленных функций позволит компании быстро и точно анализировать данные о клиентах сократив трудоёмкие задачи. Внедрение такой системы является ключевым фактором повышения эффективности бизнес-процессов и конкурентоспособности компании.

Практическая полезность разработки заключается в простоте её использования и возможности применения в аналогичных компаниях с малыми доработками.

Также, система существенно снижает трудоёмкость операций менеджеров компании по работе с клиентами и рекламе.

Теоретическая значимость работы заключается в систематизации и углублении знаний по автоматизации и моделированию бизнес-процессов организации.

В ходе работы были изучены ключевые понятия и методы, связанные с анализом поведения потребителей и применением программных решений для повышения эффективности бизнес-процессов.

В работе использованы учебная литература, нормативные документы, данные преддипломной практики и интернет ресурсы. Для сбора данных применялись опросы, наблюдения и анализ документов, что обеспечило достоверность и полноту информации.

В ходе работы мной был получен ценный опыт в проектировании и разработке информационных систем, который станет основой в моей дальнейшей профессиональной деятельности.

## Список используемой литературы и используемых источников

### *Нормативно-правовые акты*

1. ГОСТ Р ИСО 15704-2022. Моделирование и архитектура предприятия. Требования к стандартным архитектурам и методологиям предприятия: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 октября 2022 г. № 1045-ст: взамен ГОСТ Р ИСО 15704-2008. – Москва: РСТ, 2022. – 67 с. – Текст: непосредственный.

2. РД IDEF0 – 2000. Методология функционального моделирования IDEF0: руководящий документ: издание официальное: введен впервые. / разработан Научно-исследовательским Центром CALS – технологий Прикладная Логистика. – Москва: ИПК Издательство стандартов, 2000. – 62 с. Текст: непосредственный.

3. Р 50.1.028-2001. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования: рекомендации по стандартизации: издание официальное: приняты и введены в действие Постановлением Госстандарта России от 2 июля 2001 г. № 256-ст: введены впервые: дата введения 2002-07-02. / разработан Научно-исследовательским Центром CALS – технологий Прикладная Логистика. – Москва: ИПК Издательство стандартов, 2003. – 62 с. Текст: непосредственный.

### *Научная и методическая литература*

4. Бабич А. В. Введение в UML: учебное пособие / А. В. Бабич. – 4е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. – 198 с. – ISBN 978-5-4497-1637-8. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/120473.html> (дата обращения: 24.04.2025). – Режим доступа: для авторизованных пользователей.

5. Мамедли Р.Э. Системы управления базами данных: учебное пособие / Р.Э. Мамедли. – Нижневартовск: Изд-во Нижневартовского государственного университета, 2021. – 214 с. – ISBN 978-5-00047-585-0. Текст: непосредственный.

6. Миндалев И.В. Моделирование бизнес-процессов с помощью IDEF0, DFD, BPMN за 7 дней: учеб. пособие / И. В. Миндалёв; Краснояр. гос. аграр. ун-т. – Красноярск: ФГБОУ ВО «Красноярский государственный аграрный университет», 2018. – 123 с. – Текст: непосредственный.

7. Мкртычев С.В. Прикладная информатика. Бакалаврская работа: электронное учебно-методическое пособие / Гущина О.М., Очеповский А.В. – Тольятти: Изд-во ТГУ, 2019. – Электронный ресурс. – URL: <https://dspace.tltsu.ru/handle/123456789/8868> (дата обращения: 21.04.2025). – ISBN 978-5-8259-1386-5.

8. Ротер М. Учитесь видеть бизнес-процессы: построение карт потоков создания ценности / Шук Джон; перевод Г. Муравьева; под ред. С. Турко, М. Бурдиной. – 6-е изд. – Москва: Альпина Паблишер, 2018. – 136 с. – ISBN 978-5-9614-7089-5. – Текст: непосредственный.

9. Ткачук, Е. О. Методы отладки и тестирования программных продуктов: учебное пособие / составитель Е. О. Ткачук. – Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2018. – 102 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/89519.html> (дата обращения: 23.04.2025). – Режим доступа: для авторизованных пользователей.

10. Черкашин П. А. Стратегия управления взаимоотношениями с клиентами (CRM): учебное пособие / П. А. Черкашин. – 4-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. – 420 с. – ISBN 978-5-4497-3341-2. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/142306.html> (дата обращения: 23.04.2025). – Режим доступа: для авторизованных пользователей.

### *Электронные ресурсы*

11. Афанасьев И., Сергиенкова Я. Xcode: среда разработки от Apple / И. Афанасьев, Я. Сергиенкова. – Яндекс.Практикум [сайт]. – 15.02.2023. – URL: <https://practicum.yandex.ru/blog/sreda-razrabotki-xcode/> (дата обращения: 23.04.2025). – Текст: электронный.
12. Битрикс24. CRM №1 для продаж [сайт] / Битрикс24. – Москва. – URL: <https://www.bitrix24.ru/features/crm/> (дата обращения: 21.04.2025). – Текст: электронный.
13. Введение в СУБД MySQL / Интуит. Национальный открытый университет [сайт]. – Режим доступа: <https://intuit.ru/studies/courses/111/111/lecture/28508> (дата обращения: 23.04.2025). – Текст: электронный.
14. Елина А. RFM-анализ: метод сегментации клиентов для оптимизации рассылок [сайт] / А. Елина. – 18 января 2023. – URL: <https://mindbox.ru/journal/education/chto-takoe-rfm-analiz/> (дата обращения: 23.04.2025). – Текст: электронный.
15. Индекс потребительской лояльности NPS // Википедия: свободная энциклопедия. – URL: [https://ru.wikipedia.org/wiki/Индекс\\_потребительской\\_лояльности\\_NPS](https://ru.wikipedia.org/wiki/Индекс_потребительской_лояльности_NPS) (дата обращения: 23.04.2025). – Текст: электронный.
16. Лекция 2: Понятие требования. Классификации требований [сайт] / Интуит. Национальный открытый университет. – URL: <https://intuit.ru/studies/courses/2188/174/lecture/4714> (дата обращения: 23.04.2025). – Текст: электронный.
17. Лекция 3: Элементы графической нотации диаграммы вариантов использования [сайт] / Интуит. Национальный открытый университет. – URL: [https://intuit.ru/studies/professional\\_skill\\_improvements/1612/courses/32/lecture/1004](https://intuit.ru/studies/professional_skill_improvements/1612/courses/32/lecture/1004) (дата обращения: 23.04.2025). – Текст: электронный.
18. Лекция 5: Элементы графической нотации диаграммы классов [сайт] / Интуит. Национальный открытый университет. – URL:

[https://intuit.ru/studies/professional\\_skill\\_improvements/1612/courses/32/lecture/1008](https://intuit.ru/studies/professional_skill_improvements/1612/courses/32/lecture/1008)  
(дата обращения: 23.04.2025). – Текст: электронный.

19. 1С CRM: описание и обзор возможностей [сайт] / учредитель ООО «Первый Бит». – Москва. – URL: <https://www.1cbit.ru/blog/1s-crm-opisanie-i-obzor-vozmozhnostey/> (дата обращения: 21.04.2025). – Текст: электронный.

20. CRMindeX. Рейтинг CRM-систем 2025 [сайт]. – URL: <https://crmindeX.ru/> (дата обращения: 21.04.2025). – Текст: электронный.

21. FURPS [сайт] / Википедия. Свободная энциклопедия. – URL: <https://ru.wikipedia.org/wiki/FURPS> (дата обращения: 23.04.2025). – Текст: электронный.

22. Ramus Educational 1.1. Free version of Ramus which allows diagrams to be created in IDEF0 and DFD formats [сайт]. – URL: <https://ramus-educational.software.informer.com/> (дата обращения: 21.04.2025). – Текст: электронный.

#### *Источники на иностранном языке*

23. StarUML User Guide / StarUML Team. – 2025. – URL: <https://docs.staruml.io/user-guide/readme> (дата обращения: 22.04.2025). – Текст: электронный.

24. Xcode User Guide / Apple Inc. – 2025. – URL: <https://developer.apple.com/documentation/xcode> (дата обращения: 23.04.2025). – Текст: электронный.

## Приложение А

### SQL-запросы для базы данных и хранимые процедуры

SQL-запрос для создания таблицы Client:

```
CREATE TABLE Client (  
  client_id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100),  
  TIN VARCHAR(15),  
  address VARCHAR(255),  
  phone VARCHAR(15),  
  email VARCHAR(100)  
);
```

SQL-запрос для создания таблицы Contract:

```
CREATE TABLE Contract (  
  contract_id INT PRIMARY KEY AUTO_INCREMENT,  
  contract_date DATE,  
  contract_amount DECIMAL(10, 2),  
  purchase_details TEXT,  
  contract_status VARCHAR(50),  
  client_id INT,  
  FOREIGN KEY (client_id) REFERENCES Client(client_id)  
);
```

SQL-запрос для создания таблицы PurchaseItem:

```
CREATE TABLE PurchaseItem (  
  purchase_item_id INT PRIMARY KEY AUTO_INCREMENT,  
  contract_id INT,  
  item_name VARCHAR(100),  
  quantity INT,  
  unit_price DECIMAL(10, 2),  
  total_price DECIMAL(10, 2) GENERATED ALWAYS AS (quantity * unit_price) STORED,  
  FOREIGN KEY (contract_id) REFERENCES Contract(contract_id)  
);
```

SQL-запрос для создания таблицы Survey:

```
CREATE TABLE Survey (  
  survey_id INT PRIMARY KEY AUTO_INCREMENT,  
  survey_date DATE,  
  loyalty_score INT,  
  question_list TEXT,  
  answer_list TEXT,  
  client_id INT,
```

## Продолжение Приложения А

```
FOREIGN KEY (client_id) REFERENCES Client(client_id)
);
```

SQL-запрос для создания таблицы SegmentedClientBase:

```
CREATE TABLE SegmentedClientBase (
  segment_id INT PRIMARY KEY AUTO_INCREMENT,
  segment_name VARCHAR(100),
  segmentation_criteria VARCHAR(255),
  analysis_date DATE,
  client_list TEXT,
  number_of_clients INT
);
```

Значения для атрибута SegmentationCriteria в таблице SegmentedClientBase:

1. Крупные заказы: ContractAmount >= 500000
2. Средние заказы: ContractAmount BETWEEN 200000 AND 500000
3. Малые заказы: ContractAmount < 200000
4. Разовый заказ: OrderFrequency = 1
5. Повторный заказ: OrderFrequency BETWEEN 2 AND 3
6. Постоянное сотрудничество: OrderFrequency > 3
7. Промоутеры: LoyaltyScore >= 9
8. Нейтралы: LoyaltyScore BETWEEN 7 AND 8
9. Критики: LoyaltyScore < 7

Процедура для создания записей в таблице SegmentedClientBase

```
CREATE PROCEDURE PopulateSegmentedClientBase()
BEGIN
  -- Очистка существующих записей в таблице SegmentedClientBase
  DELETE FROM SegmentedClientBase;

  -- Вставка данных для сегмента «Крупные заказы»
  INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
  SELECT
    'Крупные заказы',
    'ContractAmount >= 500000',
    CURDATE(),
    GROUP_CONCAT(client_id),
    COUNT(client_id)
```

## Продолжение Приложения А

```
FROM
  (SELECT client_id, SUM(contract_amount) AS ContractAmount
   FROM Contract
   GROUP BY client_id
   HAVING ContractAmount >= 500000) AS subquery;

-- Вставка данных для сегмента «Средние заказы»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
  'Средние заказы',
  'ContractAmount BETWEEN 200000 AND 500000',
  CURDATE(),
  GROUP_CONCAT(client_id),
  COUNT(client_id)
FROM
  (SELECT client_id, SUM(contract_amount) AS ContractAmount
   FROM Contract
   GROUP BY client_id
   HAVING ContractAmount BETWEEN 200000 AND 500000) AS subquery;

-- Вставка данных для сегмента «Малые заказы»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
  'Малые заказы',
  'ContractAmount < 200000',
  CURDATE(),
  GROUP_CONCAT(client_id),
  COUNT(client_id)
FROM
  (SELECT client_id, SUM(contract_amount) AS ContractAmount
   FROM Contract
   GROUP BY client_id
   HAVING ContractAmount < 200000) AS subquery;

-- Вставка данных для сегмента «Разовый заказ»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
  'Разовый заказ',
  'OrderFrequency = 1',
  CURDATE(),
```

## Продолжение Приложения А

```
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, COUNT(contract_id) AS OrderFrequency
FROM Contract
GROUP BY client_id
HAVING OrderFrequency = 1) AS subquery;

-- Вставка данных для сегмента «Повторный заказ»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
'Повторный заказ',
'OrderFrequency BETWEEN 2 AND 3',
CURDATE(),
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, COUNT(contract_id) AS OrderFrequency
FROM Contract
GROUP BY client_id
HAVING OrderFrequency BETWEEN 2 AND 3) AS subquery;

-- Вставка данных для сегмента «Постоянное сотрудничество»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
'Постоянное сотрудничество',
'OrderFrequency > 3',
CURDATE(),
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, COUNT(contract_id) AS OrderFrequency
FROM Contract
GROUP BY client_id
HAVING OrderFrequency > 3) AS subquery;

-- Вставка данных для сегмента «Промоутеры»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
'Промоутеры',
```

## Продолжение Приложения А

```
'LoyaltyScore >= 9',
CURDATE(),
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, AVG(loyalty_score) AS LoyaltyScore
FROM Survey
GROUP BY client_id
HAVING LoyaltyScore >= 9) AS subquery;

-- Вставка данных для сегмента «Нейтралы»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
'Нейтралы',
'LoyaltyScore BETWEEN 7 AND 8',
CURDATE(),
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, AVG(loyalty_score) AS LoyaltyScore
FROM Survey
GROUP BY client_id
HAVING LoyaltyScore BETWEEN 7 AND 8) AS subquery;

-- Вставка данных для сегмента «Критики»
INSERT INTO SegmentedClientBase (segment_name, segmentation_criteria,
analysis_date, client_list, number_of_clients)
SELECT
'Критики',
'LoyaltyScore < 7',
CURDATE(),
GROUP_CONCAT(client_id),
COUNT(client_id)
FROM
(SELECT client_id, AVG(loyalty_score) AS LoyaltyScore
FROM Survey
GROUP BY client_id
HAVING LoyaltyScore < 7) AS subquery;
END;
```

SQL-запрос для создания таблицы ClientSegment:

## Продолжение Приложения А

```
CREATE TABLE ClientSegment (  
    segment_id INT,  
    client_id INT,  
    PRIMARY KEY (segment_id, client_id),  
    FOREIGN KEY (segment_id) REFERENCES SegmentedClientBase(segment_id),  
    FOREIGN KEY (client_id) REFERENCES Client(client_id)  
);
```

SQL-запрос для создания таблицы ClientProfile:

```
CREATE TABLE ClientProfile (  
    profile_id INT PRIMARY KEY AUTO_INCREMENT,  
    client_id INT,  
    purchase_dates TEXT,  
    total_contract_amount DECIMAL(10, 2),  
    total_purchase_quantity INT,  
    loyalty_score INT,  
    FOREIGN KEY (client_id) REFERENCES Client(client_id)  
);
```

Значения для атрибута TargetCriteria в таблице TargetSegment:

Крупные заказы и Постоянное сотрудничество:

OrderAmount >= 500000 AND OrderFrequency > 3

Средние заказы и Постоянное сотрудничество:

OrderAmount BETWEEN 200000 AND 500000 AND OrderFrequency > 3

Процедура для формирования таблицы TargetSegment:

```
CREATE PROCEDURE PopulateTargetSegment()  
BEGIN  
    -- Очистка существующих записей в таблице TargetSegment  
    DELETE FROM TargetSegment;  
  
    -- Вставка данных для целевого сегмента «Крупные заказы и Постоянное  
    сотрудничество»  
    INSERT INTO TargetSegment (target_segment_name, target_criteria,  
    target_client_list, number_of_target_clients, segment_id)  
    SELECT  
        'Крупные заказы и Постоянное сотрудничество',  
        'ContractAmount >= 500000 AND OrderFrequency > 3',  
        GROUP_CONCAT(DISTINCT c.client_id),  
        COUNT(DISTINCT c.client_id),
```

## Продолжение Приложения А

```
        scb.segment_id
FROM
    SegmentedClientBase scb
JOIN
    ClientSegment cs ON scb.segment_id = cs.segment_id
JOIN
    Client c ON cs.client_id = c.client_id
WHERE
    scb.segmentation_criteria = 'ContractAmount >= 500000 AND OrderFrequency > 3';

-- Вставка данных для целевого сегмента «Средние заказы и Постоянное
сотрудничество»
INSERT INTO TargetSegment (target_segment_name, target_criteria,
target_client_list, number_of_target_clients, segment_id)
SELECT
    'Средние заказы и Постоянное сотрудничество',
    'ContractAmount BETWEEN 200000 AND 500000 AND OrderFrequency > 3',
    GROUP_CONCAT(DISTINCT c.client_id),
    COUNT(DISTINCT c.client_id),
    scb.segment_id
FROM
    SegmentedClientBase scb
JOIN
    ClientSegment cs ON scb.segment_id = cs.segment_id
JOIN
    Client c ON cs.client_id = c.client_id
WHERE
    scb.segmentation_criteria = 'ContractAmount BETWEEN 200000 AND 500000
AND OrderFrequency > 3';
END;
```

Процедура для заполнения таблицы Forecast для связанных покупок:

```
CREATE PROCEDURE PopulateForecastRelatedPurchases()
BEGIN
    -- Очистка существующих записей с типом прогноза «Связные покупки»
    DELETE FROM Forecast WHERE forecast_type = 'Связные покупки';

    -- Вставка данных о связанных покупках
    INSERT INTO Forecast (forecast_type, forecast_description, forecast_result,
number_of_occurrences)
    SELECT
        'Связные покупки',
```

## Продолжение Приложения А

```
'Товары, которые часто приобретаются вместе в рамках одного договора',
GROUP_CONCAT(DISTINCT pi.ItemName ORDER BY pi.ItemName
SEPARATOR ', '),
COUNT(*) AS number_of_occurrences
FROM
PurchaseItem pi
JOIN
(SELECT ContractID
FROM PurchaseItem
GROUP BY ContractID
HAVING COUNT(DISTINCT ItemName) > 1) AS contracts_with_multiple_items
ON pi.ContractID = contracts_with_multiple_items.ContractID
GROUP BY
pi.ItemName
HAVING
COUNT(*) > 1;
END;
```

## Приложение Б

### Тестовые сценарии

Таблица Б.1 – Тестовые сценарии

| ID   | Название                  | Описание сценария                                                       | Тест-кейсы                                          | Шаги тестирования                                                                                                                                          | Ожидаемый результат                                                                                     |
|------|---------------------------|-------------------------------------------------------------------------|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| TS01 | Вход в систему            | Пользователь вводит логин и пароль для входа в систему                  | Вход с валидными учетными данными                   | 1 Ввести валидные логин и пароль<br>2 Нажать кнопку «Войти»                                                                                                | Успешный вход в систему с направлением на главную страницу                                              |
|      |                           |                                                                         | Вход с не валидным логином                          | 1 Ввести не валидные логин и валидный пароль<br>2 Нажать кнопку «Войти»                                                                                    | Появляется сообщение об ошибке «Неверный логин или пароль».                                             |
|      |                           |                                                                         | Вход с не валидным паролем                          | 1 Ввести валидный логин не валидный пароль.<br>2 Нажать кнопку «Войти»                                                                                     | Появляется сообщение об ошибке «Неверный логин или пароль».                                             |
| TS02 | Добавление нового клиента | Пользователь вводит информацию о новом клиенте и сохраняет ее в системе | Успешное добавление нового клиента                  | 1 Открыть форму добавления нового Клиента.<br>2 Ввести валидные данные о клиенте.<br>3 Нажать кнопку «Сохранить».                                          | Клиент успешно добавлен в базу данных, система отображает подтверждение.                                |
|      |                           |                                                                         | Добавление клиента с пустыми полями                 | 1 Открыть форму добавления нового клиента.<br>2 Оставить обязательные поля пустыми.<br>3 Нажать кнопку «Сохранить».                                        | Система отображает сообщение об ошибке, указывая на обязательные поля для заполнения                    |
|      |                           |                                                                         | Добавление клиента с уже существующим наименованием | 1 Открыть форму добавления нового клиента.<br>2 Ввести наименование или ИНН клиента, которое уже существует в базе данных.<br>3 Нажать кнопку «Сохранить». | Система отображает сообщение об ошибке, указывая на то, что клиент с таким наименованием уже существует |
|      |                           |                                                                         | Добавление клиента с не валидными данными           | 1 Открыть форму добавления нового клиента.<br>2 Ввести неверный формат ИНН, телефона.<br>3 Нажать кнопку «Сохранить».                                      | Система отображает сообщение об ошибке, указывая на неверный формат данных.                             |

## Продолжение Приложения Б

Продолжение таблицы Б.1

| ID   | Название                                                  | Описание сценария                                                                             | Тест-кейсы                                                               | Шаги тестирования                                                                                                                                                                                     | Ожидаемый результат                                                                                                         |
|------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| TS03 | Добавление нового договора                                | Пользователь создает и сохраняет информацию о договоре, связанном с клиентом.                 | Успешное добавление нового договора                                      | 1 Открыть форму добавления нового договора<br>2 Выбрать клиента вводом наименования, ИНН<br>3 Ввести валидные данные о договоре<br>4 Нажать кнопку «Сохранить».                                       | Договор успешно добавлен и связан с соответствующим клиентом                                                                |
|      |                                                           |                                                                                               | Добавление договора с пустыми полями                                     | 1 Открыть форму добавления нового договора.<br>2 Выбрать клиента вводом наименования, ИНН<br>3 Оставить обязательные поля пустыми.<br>4 Нажать кнопку «Сохранить».                                    | Система отображает сообщение об ошибке, указывая на обязательные поля для заполнения                                        |
|      |                                                           |                                                                                               | Добавление договора с неверным форматом даты                             | 1 Открыть форму добавления нового договора<br>2 Выбрать клиента введя наименования, ИНН<br>3 Ввести не валидные данные в форму<br>4 Нажать кнопку «Сохранить».                                        | Система отображает сообщение об ошибке «Введите корректную дату «                                                           |
| TS04 | Добавление нового опроса                                  | Пользователь создает и сохраняет результаты опроса, проведенного среди клиентов               | Успешное добавление нового опроса                                        | 1 Открыть форму добавления нового опроса<br>2 Выбрать группу клиентов, для опроса.<br>3 Ввести информацию об опросе<br>4 Нажать кнопку «Сохранить».                                                   | Результаты опроса успешно добавлены в базу данных                                                                           |
|      |                                                           |                                                                                               | Проверка обязательных полей при добавлении опроса                        | 1 Открыть форму добавления нового опроса<br>2 Оставить пустыми обязательные поля<br>Нажать кнопку «Сохранить».                                                                                        | Система отображает сообщения об ошибках для каждого пустого обязательного поля                                              |
| TS05 | Проверка отображения данных на странице «Профиль клиента» | Открытие страницы «Профиль клиента» и проверка наличия и правильности отображаемой информации | Отображение данных о клиенте                                             | 1 Открыть страницу «Профиль клиента» для существующего клиента.<br>2. Проверить наличие даты покупок, суммы покупок, частоты покупок, оценки лояльности.                                              | Все указанные данные отображаются в системе корректно и в нужном формате                                                    |
|      |                                                           |                                                                                               | Автоматическое обновление Профиля клиента при добавлении нового договора | Открыть форму добавления нового договора для выбранного клиента.<br>3 Ввести данные нового договора<br>4. Нажать кнопку «Сохранить».<br>5 Перейти на страницу «Профиль клиента» и проверить изменения | Даты добавленной покупки отображается корректно; Общая сумма покупок обновляется. Частота покупок увеличивается на единицу. |

## Продолжение Приложения Б

Продолжение таблицы Б.1

| ID   | Название                               | Описание сценария                                                                   | Тест-кейсы                                 | Шаги тестирования                                                                                                                                                   | Ожидаемый результат                           |
|------|----------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| TS06 | Сегментация по сумме контракта         | Проверка корректности распределения компаний по сегментам на основе суммы контракта | Крупные заказы (критерий $\geq 500\ 000$ ) | Добавить клиента и Договор ContractAmount = 600 000.<br>2. Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Крупные заказы»<br>3 Запустить сегментацию.  | Компания включена в сегмент «Крупные заказы»  |
|      |                                        |                                                                                     | Средние заказы (200 000–500 000)           | 1 Добавить клиента и Договор ContractAmount = 300 000.<br>2 Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Крупные заказы»<br>3 Запустить сегментацию. | Компания включена в сегмент «Средние заказы». |
| TS07 | Сегментация по частоте заказов         | Проверка распределения компаний по категориям частоты заказов                       | Разовый заказ (=1)                         | 1 Добавить нового клиента и один Договор<br>2 Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Разовый заказ»<br>3 Запустить сегментацию.                | Компания включена в сегмент «Разовый заказ»   |
|      |                                        |                                                                                     | Повторный заказ (2–3)                      | 1. Добавить нового клиента и два Договора<br>2 Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Повторный заказ»<br>3 Запустить сегментацию.             | Компания включена в сегмент «Повторный заказ» |
| TS08 | Сегментации по лояльности LoyaltyScore | Проверка распределения компаний по уровню лояльности                                | Критики (<7)                               | 1 Добавить Опрос клиента с LoyaltyScore = 6.<br>2 Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Критики»<br>3 Запустить сегментацию.                  | Компания включена в сегмент «Критики»         |
|      |                                        |                                                                                     | Промоутеры ( $\geq 9$ )                    | 1. Добавить Опрос клиента с LoyaltyScore = 9.<br>2. Перейти на страницу «Просмотр сегментов» и выбрать сегмент «Промоутеры»<br>3. Запустить сегментацию.            | Компания включена в сегмент «Промоутеры»      |
| TS09 | Расчет NPS                             | Проверка корректности расчета NPS на основе данных опроса                           | Расчет NPS с преобладанием промоутеров     | 1 Создать 10 клиентов: 8 промоутеров, 1 нейтрал, 1 критик.<br>2 Запустить расчет NPS.                                                                               | NPS = 70                                      |
|      |                                        |                                                                                     | Расчет NPS с преобладанием критиков        | 1 Создать 10 клиентов: 1 промоутер, 2 нейтрала, 7 критиков.<br>2 Запустить расчет NPS.                                                                              | NPS = -60                                     |

## Продолжение Приложения Б

Продолжение таблицы Б.1

| ID   | Название                                           | Описание сценария                                                       | Тест-кейсы                                                         | Шаги тестирования                                                                                                                                                                                                                                        | Ожидаемый результат                                              |
|------|----------------------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| TS10 | Представление данных в таблицах                    | Проверка отображения данных в табличном формате                         | Таблица с клиентами в сегментах                                    | 1 Перейти на страницу «Просмотр сегментов» и выбрать сегмент<br>2 Запустить сегментацию.                                                                                                                                                                 | Отобрана Таблица, содержащая список клиентов с ключевыми данными |
| TS11 | Представление данных в отчетах                     | Проверка формирования отчетов о сегментах                               | Отчет по сегменту                                                  | . Создать отчет для конкретного сегмента.<br>2 Проверить содержание отчета.                                                                                                                                                                              | Отчет содержит сводную информацию о сегменте                     |
| TS12 | Расчет совместных продаж                           | Проверка корректности подсчета парных покупок на типовых данных         | Заказ из 3 продуктов: A+B+C<br>Повторяющиеся заказы: A+B, A+B, A+C | 1 Открыть форму добавления нового договора<br>2 Ввести тестовые данные об продуктах/услугах<br>3 Запустить анализ за нужный период<br>4 Открыть итоговую таблицу                                                                                         | A и B - 3 совместные продажи<br>A и C – 1 совместная продажа     |
|      |                                                    |                                                                         | Заказ только продукта X:<br>5 заказов продукта Y                   | . Открыть форму добавления нового договора<br>2 Ввести тест данные об продуктах/услугах<br>3 Запустить анализ за нужный период<br>4 Открыть итоговую таблицу                                                                                             | Продукты X и Y отсутствуют в итоговой таблице                    |
| TS13 | Проверка аналитических индикаторов прогнозирования | Проверка корректности отображения и расчета индикаторов прогнозирования | Базовый расчет                                                     | 1 Открыть форму добавления нового клиента.<br>Создать тестовых клиентов.<br>2 Добавить тестовые договоры созданным клиентам 2. Зафиксировать тестовые суммы.<br>3 Перейти на страницу «Индикаторы прогнозов» Задать тест. период и проверить показатели. | Система отображает расчетные значения                            |
|      |                                                    |                                                                         | Обновление данных                                                  | 1 Открыть форму добавления нового Договора, добавить тест договор существующему Клиенту<br>2 Перейти на страницу «Индикаторы прогнозов» Задать тестовый период и проверить обновление значений.                                                          | Система обновила значения корректно                              |

## Приложение В

### Матрица трассируемости требований

Таблица В.1 - Матрица трассируемости требований

| Функциональные требования                                                                                                                                                                                                 | Тестовые сценарии |      |      |      |      |      |      |      |      |      |      |      |      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|
|                                                                                                                                                                                                                           | TS01              | TS02 | TS03 | TS04 | TS05 | TS06 | TS07 | TS08 | TS09 | TS10 | TS11 | TS12 | TS13 |
| 1. ПО должно обеспечивать ввод, редактирование, хранения различных данных о Клиентах, включая наименование и реквизиты, историю (договоры, состав покупок, опросы).                                                       |                   | +    | +    | +    |      |      |      |      |      |      |      |      |      |
| 2. ПО должно автоматически обрабатывать данные о клиентах для формирования и поддержания актуального Профиля клиента.                                                                                                     |                   |      |      |      | +    |      |      |      |      |      |      |      |      |
| 3. ПО должно предоставлять функционал для автоматической сегментации клиентской базы в соответствии с заданными критериями и выделять целевые сегменты на основе заданных критериев.                                      |                   |      |      |      |      | +    | +    | +    |      |      |      |      |      |
| 4. ПО должно автоматически рассчитывать индекс NPS на основе опроса клиентов, сегментировать их на группы и предоставлять результаты для прогнозирования поведения.                                                       |                   |      |      |      |      |      |      | +    | +    |      |      |      |      |
| 5. ПО должно обеспечивать возможность представления данных о клиентах в сегментах клиентской базы и целевых сегментах в виде таблиц, отчетов или графиков                                                                 |                   |      |      |      |      |      |      |      |      | +    | +    |      |      |
| 6. ПО должно обеспечивать доступ к сводной информации о клиентах, для создания индивидуальных подходов к обслуживанию.                                                                                                    |                   |      |      |      |      | +    | +    | +    |      | +    | +    |      |      |
| 7. ПО должно предоставлять инструменты для прогнозирования поведения потребителей:<br>- анализ связанных покупок с целью выявления ассоциаций между продуктами и услугами;<br>- аналитические индикаторы прогнозирования. |                   |      |      |      |      |      |      |      | +    |      |      | +    | +    |
| 8. ПО должно позволять зарегистрированным пользователям входить в систему с использованием валидных учетных данных.                                                                                                       | +                 |      |      |      |      |      |      |      |      |      |      |      |      |

## Приложение Г

### Фрагменты кода

#### Фрагмент кода модуля авторизации

```
swift
import SwiftUI

struct AuthorizationView: View {
    @State private var username: String = «»
    @State private var password: String = «»
    @State private var rememberMe: Bool = false

    var body: some View {
        VStack {
            Text(«Авторизация»)
                .font(.title)

            TextField(«Логин», text: $username)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .frame(width: 300, height: 50)
                .background(Color(red: 217/255, green: 217/255, blue: 217/255))
                .cornerRadius(10)

            SecureField(«Пароль», text: $password)
                .textFieldStyle(RoundedBorderTextFieldStyle())
                .frame(width: 300, height: 50)
                .background(Color(red: 217/255, green: 217/255, blue: 217/255))
                .cornerRadius(10)

            HStack {
                Button(action: {
                    self.rememberMe.toggle()
                }) {
                    Image(systemName: self.rememberMe ? «checkmark.square» : «square»)
                }
                Text(«Запомнить пароль»)
            }

            Button(action: {
                self.loginButtonTapped()
            }) {
                Text(«Войти»)
            }
            .frame(width: 300, height: 50)
        }
    }
}
```

## Продолжение Приложения Г

```
.background(Color(red: 140/255, green: 168/255, blue: 210/255))
.foregroundColor(.white)
.cornerRadius(10)

Link(destination: URL(string: «https://example.com/restore-password»!)) {
    Text(«Восстановить пароль»)
        .font(.system(size: 16, weight: .bold, design: .italic))
        .foregroundColor(Color(red: 36/255, green: 107/255, blue: 212/255))
}
}
.padding()
}

private func loginButtonTapped() {
    guard !username.isEmpty, !password.isEmpty else {
        print(«Введите логин и пароль.»)
        return
    }

    // URL сервера для авторизации
    let url = URL(string: «https://example.com/login»!)

    // Создание запроса
    var request = URLRequest(url: url, cachePolicy: .useProtocolCachePolicy)
    request.httpMethod = «POST»
    request.httpBody = «username=\(username)&password=\(password)».data(using: .utf8)

    // Отправка запроса на сервер
    URLSession.shared.dataTask(with: request) { data, response, error in
        if let error = error {
            print(«Ошибка запроса: \(error)»)
            return
        }

        // Обработка ответа от сервера
        if let data = data {
            do {
                let json = try JSONSerialization.jsonObject(with: data, options: [])
                print(«Ответ от сервера: \(json)»)

                // Если авторизация успешна, переход на главный экран
                if let dict = json as? [String: Any], dict[«success»] as? Bool == true {
                    DispatchQueue.main.async {
```

## Продолжение Приложения Г

```
let mainMenuView = MainMenuView()
let window = NSWindow(contentRect: NSRect(x: 0, y: 0, width: 800, height:
600),
                        styleMask: [.titled, .closable],
                        backing: .buffered,
                        defer: false)
window.contentView = NSHostingView(rootView: mainMenuView)
window.makeKeyAndOrderFront(nil)
}
} else {
    print(«Неправильный логин или пароль.»)
}
} catch {
    print(«Ошибка парсинга JSON: \(error)»)
}
}
}.resume()
}
```

### Фрагмент кода реализации главного меню

```
swift
struct MainMenuView: View {
    @State private var selectedMenu: String = «»

    var body: some View {
        VStack {
            Text(«АИС для анализа поведения потребителей»)
                .font(.title)

            Text(«Вы успешно авторизованы. Начните работу с АИС!»)
                .padding()

            HStack {
                VStack(alignment: .leading) {
                    Button(action: {
                        // Переключить на меню «Справочники»
                        self.selectedMenu = «Справочники»
                    }) {
                        Text(«Справочники»)
                            .font(.system(size: 20, weight: .semibold))
                    }
                }
            }
        }
    }
}
```

## Продолжение Приложения Г

```
Button(action: {
  // Переключить на меню «Управление»
  self.selectedMenu = «Управление»
}) {
  Text(«Управление»)
    .font(.system(size: 20, weight: .semibold))
}

Button(action: {
  // Переключить на меню «Анализ»
  self.selectedMenu = «Анализ»
}) {
  Text(«Анализ»)
    .font(.system(size: 20, weight: .semibold))
}

Button(action: {
  // Переход на экран авторизации
}) {
  Text(«Выход»)
    .frame(width: 125, height: 35)
    .background(Color(red: 175/255, green: 173/255, blue: 173/255))
}

}
.background(Color(red: 188/255, green: 205/255, blue: 223/255))

if selectedMenu == «Справочники» {
  VStack {
    Button(action: {
      // Открыть таблицу с Клиентами
    }) {
      Text(«Клиенты»)
        .font(.system(size: 18))
    }
    Button(action: {
      // Открыть таблицу с Договорами
    }) {
      Text(«Договоры»)
        .font(.system(size: 18))
    }
    Button(action: {
      // Открыть таблицу с Опросами
    }) {
```

## Продолжение Приложения Г

```
        Text(«Опросы»)
        .font(.system(size: 18))
    }
    Button(action: {
        // Открыть таблицу с Сегментами
    }) {
        Text(«Сегменты»)
        .font(.system(size: 18))
    }
}
} else if selectedMenu == «Управление» {
    VStack {
        Button(action: {
            // Открыть подменю для Клиентов
        }) {
            Text(«Клиенты»)
            .font(.system(size: 18))
        }
        Button(action: {
            // Открыть подменю для Договоров
        }) {
            Text(«Договоры»)
            .font(.system(size: 18))
        }
        Button(action: {
            // Открыть подменю для Опросов
        }) {
            Text(«Опросы»)
            .font(.system(size: 18))
        }
    }
    .onAppear {
        // Дополнительные действия при открытии подменю «Управление»
    }
    .onDisappear {
        // Дополнительные действия при закрытии подменю «Управление»
    }

    VStack {
        Button(action: {
            // Добавить данные
        }) {
            Text(«Добавить»)
```

## Продолжение Приложения Г

```
.font(.system(size: 16))
}
Button(action: {
    // Редактировать данные
}) {
    Text(«Редактировать»)
    .font(.system(size: 16))
}
Button(action: {
    // Открыть отчеты
}) {
    Text(«Отчеты»)
    .font(.system(size: 16))
}
}
} else if selectedMenu == «Анализ» {
    VStack {
        Button(action: {
            // Открыть подменю для Сегментов
        }) {
            Text(«Сегменты»)
            .font(.system(size: 18))
        }
        Button(action: {
            // Открыть подменю для Целевых сегментов
        }) {
            Text(«Целевые сегменты»)
            .font(.system(size: 18))
        }
    }
}
}
}
}
}
```

### Фрагмент кода для вызова API сегментации клиентской базы

```
swift
// Общая функция для вызова API и обработки данных
func fetchSegmentationData(completion: @escaping ([SegmentedData]) -> Void) {
    guard let url = URL(string: «http://localhost:3000/segmentation») else { return }
}
```

## Продолжение Приложения Г

```
var request = URLRequest(url: url, cachePolicy: .useProtocolCachePolicy)
request.httpMethod = «POST»

URLSession.shared.dataTask(with: request) { data, response, error in
    if let error = error {
        print(«Ошибка запроса: \(error)»)
        return
    }

    if let data = data {
        do {
            let json = try JSONDecoder().decode([SegmentedData].self, from: data)
            completion(json)
        } catch {
            print(«Ошибка парсинга JSON: \(error)»)
        }
    }
}.resume()
}

swift
// Представление для отображения сегментированных данных
struct SegmentationView: View {
    @State private var segmentedData: [SegmentedData] = []

    var body: some View {
        VStack {
            Button(action: {
                // Вызов API для сегментирования
                fetchSegmentationData { [weak self] data in
                    DispatchQueue.main.async {
                        self?.segmentedData = data
                    }
                }
            }) {
                Text(«Выполнить сегментирование»)
            }

            // Отображение сегментированных данных
            List(segmentedData, id: \.id) { item in
                VStack(alignment: .leading) {
                    Text(«Сегмент: \(item.segmentName)»)
                    .font(.headline)
                }
            }
        }
    }
}
```

## Продолжение Приложения Г

```
Text(«Критерий: \(\item.segmentationCriteria)\»)  
    .font(.subheadline)  
Text(«Дата анализа: \(\item.analysisDate)\»)  
    .font(.subheadline)  
Text(«Клиенты: \(\item.clientList.joined(separator: «, «))\»)  
    .font(.subheadline)  
Text(«Количество клиентов: \(\item.numberOfClients)\»)  
    .font(.subheadline)  
}  
}  
}
```