Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет инженерно-экономический

Кафедра экономической информатики

Дисциплина «Объектно – ориентированное проектирование и программирование»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  Старший преподаватель кафедры ЭИ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю. Ю. Петрович |
|  | \_\_\_.\_\_\_\_.2024 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИ СКА**

к курсовой работе

на тему:

**«ПРОГРАММНОЕ СРЕДСТВО С АВТОМАТИЗИРОВАННОЙ СИСТЕМОЙ РАСЧЁТА БАНКОВСКИХ УСЛУГ И ТРАНЗАКЦИОННЫХ ПЛАТЕЖЕЙ»**

БГУИР КР 1–40 05 01–02 ПЗ

|  |  |
| --- | --- |
|  | Выполнил студент группы 272303  БАДЕЙ Павел Юрьевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовая работа представлена на проверку \_\_\_.\_\_\_\_.2024  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2024

**РЕФЕРАТ**

БГУИР КР 1–40 05 01–02 ПЗ

**Бадей, П.Ю.** Программное средство с автоматизированной системой расчёта банковских услуг и транзакционных платежей/              П.Ю. Бадей. – Минск: БГУИР, 2024. – 39 с.

Пояснительная записка 39 с., 33 рис., 7 источников, 4 приложения

РазработкА автоматизированной системы функционирования магазина электроники, ЗАКАЗ ТОВАРА, СОРТИРОВКА ТОВАРОВ, МОДЕЛИ *UML*, *IDEF*0, СХЕМЫ АЛГОРИТМОВ, ПРОГРАММНОЕ СРЕДСТВО.

*Цель* *курсовой работы*: систематизация и углубление теоретических знаний, полученных за время обучения, а так же приобретение и закрепление навыков самостоятельной работы.

*Методология проведения работы*: в процессе решения поставленных задач использованы принципы системного подхода и аналитические методы, принципы объектно–ориентированного проектирования и программирования.

*Результаты работы*: выполнена постановка задачи и определены основные методы ее решения; в ходе объектного моделирования системы построен ряд *UML*–диаграмм; разработаны модели бизнес–процессов предметной области на основе нотаций IDEF0 и BPMN; описаны основные алгоритмы работы программного средства; разработано руководство пользователя; выполнено тестирование программного средства, показавшее его соответствие функциональным требованиям, поставленным в задании на разработку.

Программный продукт разработан на языке *C++* с применением MS Visual Studio 2022.

*Область применения результатов*: c помощью разработанного программного средства можно автоматизировать работу магазинов электроники, упростить и наладить поиск, сортировку и выбор товаров, основываясь на предпочтениях пользователя.

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ**

В современном банковском секторе автоматизация играет ключевую роль в обеспечении эффективности и надежности банковских услуг. Программное средство с автоматизированной системой расчёта банковских услуг и транзакционных платежей (АБС) представляет собой комплекс программного и технического обеспечения, разработанный для автоматизации банковской деятельности.

Автоматизированные банковские системы (АБС) имеют долгую и интересную историю развития. Выделяют несколько этапов становления данной системы, начиная с первых шагов в СССР и до современных решений:

1. Первый этапы (1960—1975 гг.): в СССР начали разрабатываться аналогичные системы в отраслевых научно-исследовательских институтах, таких как “Киевский операционный день” и “Тульский операционный день”. На рынке появились первые коммерческие разработчики АБС, такие как компания ПрограмБанк (позднее переименованная в Центавр), Инверсия и АСОФТ. Первые версии АБС работали на операционных системах MS-DOS и Netware, используя текстовый интерфейс.
2. Второй этап (1972—1989 гг.): в этот период развития банковской сферы появились новые технологии и стандарты. Программные продукты начали переходить на графический интерфейс и использовать СУБД, такие как Oracle Database, Sybase и Microsoft SQL Server.
3. Третий этап (с 1989 г. по настоящее время): в 1998 году в России был введен новый план счетов бухгалтерского учета для банков и кредитных организаций. Этот момент стал переломным, заставив банки перейти на промышленные АБС. Специализированные решения, такие как DiasoftBANK, RS-Bank и другие, стали широко распространены.

С развитием технологий, автоматизированные системы расчёта банковских услуг и транзакционных платежей становятся всё более мощными и гибкими. Выделяют несколько направлений, которые могут определить будущее АБС:

* Искусственный интеллект и машинное обучение;
* Блокчейн и криптовалюты;
* Интернет вещей (IoT);
* Кибербезопасность;

Автоматизированные системы расчёта банковских услуг и транзакционных платежей продолжают эволюционировать, обеспечивая банкам и клиентам более удобные и безопасные услуги. Надеюсь, что этот курсовой проект поможет вам лучше понять и оценить значимость АБС в современной банковской индустрии.

## 1 АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОГРАММНОГО СРЕДСТВА

Предметная область данного программного средства - это автоматизированная система расчёта банковских услуг и транзакционных платежей. Оно включает в себя обработку и анализ данных о банковских операциях, расчёт комиссий и процентов, а также управление транзакциями.

В рамках этой предметной области основными сущностями являются:

1. Банковский счёт. Это основная единица для проведения транзакций. Каждый счёт имеет уникальный номер и принадлежит определённому клиенту.
2. Транзакция. Это операция перевода денег с одного счёта на другой. Каждая транзакция имеет уникальный номер, дату и время проведения, а также сумму перевода.
3. Услуга. Это дополнительные услуги, предоставляемые банком, за которые взимается плата. Примеры услуг включают обслуживание счёта, предоставление кредитов и страхование.

Для моделирования предметной области можно использовать различные подходы, такие как ER-диаграммы (Entity-Relationship), UML-диаграммы (Unified Modeling Language) или BPMN-диаграммы (Business Process Model and Notation).

В контексте данного проекта, ER-диаграмма может быть наиболее подходящим инструментом для визуализации структуры данных и отношений между сущностями. Например, сущности «Банковский счёт», «Транзакция» и «Услуга» могут быть связаны следующим образом:

* банковский счёт может иметь множество транзакций;
* каждая транзакция связана с двумя банковскими счетами: счётом отправителя и счётом получателя;
* каждый банковский счёт может быть связан с множеством услуг, которые предоставляются клиенту.

Важно отметить, что модель должна быть достаточно гибкой, чтобы учесть возможные изменения в бизнес-процессах или требованиях к программному средству. Поэтому процесс моделирования предметной области должен включать постоянное взаимодействие с заинтересованными сторонами, такими как бизнес-аналитики, разработчики и конечные пользователи.

**1.1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

В современном мире банковские услуги и транзакционные платежи играют ключевую роль в финансовой сфере. С развитием информационных технологий возникла необходимость в создании программных средств с автоматизированными системами, способными обеспечивать эффективное управление банковскими операциями и платежными процессами. В данной главе производится обзор предметной области, рассматриваются основные аспекты банковских услуг и транзакционных платежей.

Банки предоставляют кредиты и дебетовые карты своим клиентам для осуществления покупок, снятия наличных средств, оплаты услуг и т. д. Кредитные продукты включают в себя кредитные линии, потребительские кредиты, ипотечные кредиты и другие виды кредитования.

Банки предоставляют клиентам возможность инвестировать свои средства в различные финансовые инструменты, такие как акции, облигации, фонды и другие. Инвестиционные услуги могут включать в себя управление портфелем, консультации по инвестициям и т. д.

Транзакционные платежи представляют собой передачу денежных средств между различными участниками финансовых операций. Они могут быть осуществлены как внутри одной страны, так и между различными странами.

Платежи между клиентами и банками – это платежи, осуществляемые между клиентами и их банками. Они могут включать в себя оплату счетов, переводы между счетами, покупку товаров и услуг с использованием платежных карт и т. д. Предметная область программного средства с автоматизированной системой расчета банковских услуг и транзакционных платежей охватывает широкий спектр финансовых операций, осуществляемых в банковской сфере. Для успешной разработки такого программного продукта необходимо глубокое понимание всех аспектов банковских услуг и платежных процессов, что и было представлено в данной главе.

Помимо основных аспектов банковских услуг и транзакционных платежей, важно также уделить внимание технологическим и регуляторным аспектам данной области. Технологические инновации играют существенную роль в развитии банковской отрасли. Современные информационные системы, включая базы данных и прикладные программы, обеспечивают не только эффективное управление банковскими операциями, но и гарантируют высокий уровень безопасности при проведении платежных транзакций. Также стоит отметить внедрение технологий блокчейн и цифровых валют, которые предоставляют новые возможности для оптимизации транзакционных процессов и укрепления доверия между участниками финансовых операций.

Кроме того, регуляторные нормативы играют ключевую роль в функционировании банковской системы. Законодательство и нормативы регулируют широкий спектр банковских операций, включая кредитование, платежные системы, а также обеспечивают стабильность финансовой системы в целом. Регулирование банковской сферы включает в себя нормативные акты центральных банков, регуляторных органов и международных организаций, таких как Банк международных расчетов (БМР) и Международный валютный фонд (МВФ).

**1.2 РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ**

IDEF0 (сокращение от "Integrated Definition for Function Modeling 0") – это методология, которая используется для моделирования и анализа функций в системах.

Методология IDEF0 использует графическое представление в виде функциональных блоков и стрелок для обозначения потока данных, управления и данных. Все функциональные блоки связаны взаимосвязями, отображающими логические отношения между функциями.

IDEF0 может использоваться в различных областях, таких как разработка программного обеспечения, производство, бизнес–процессы и т. д. Она помогает улучшить понимание функций системы, идентифицировать проблемы и улучшить процессы.

Ниже представлен нулевой уровень модели «Автоматизировать систему расчета банковских услуг и транзакционных платежей» (рисунок 1.1). Здесь на вход поступают документы клиента и сам клиент, а на выходе получаем оказанную услугу или отказ в услуге.

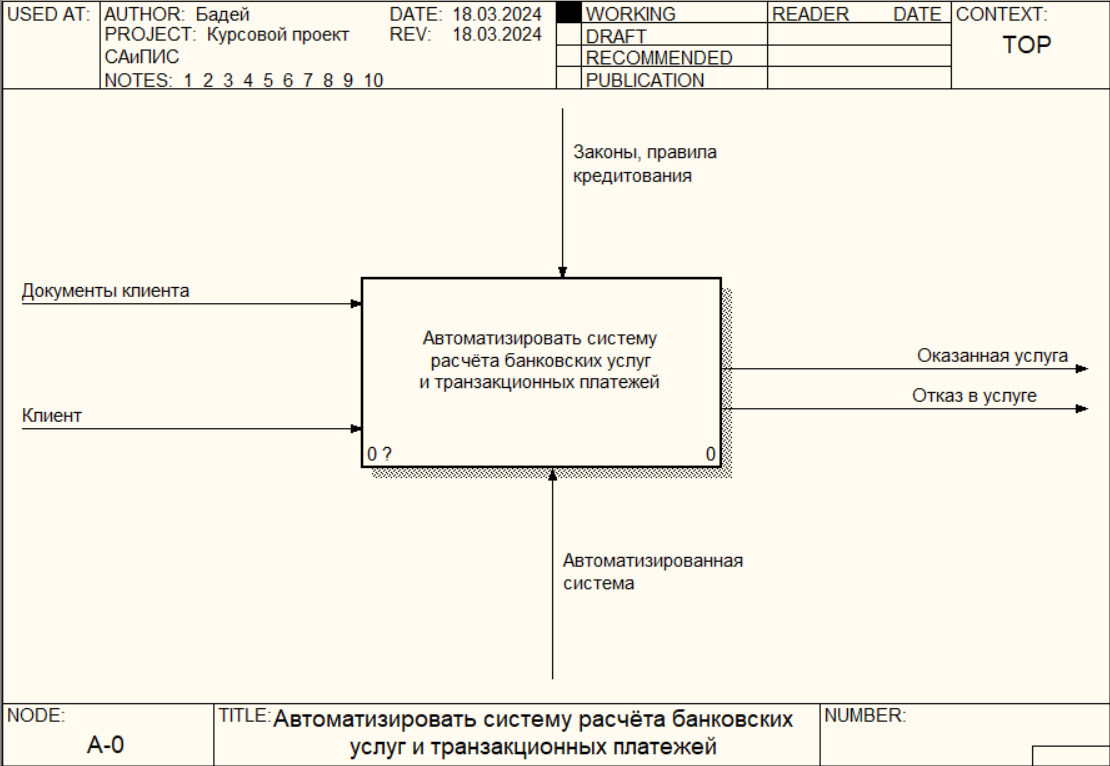


Рисунок 1.1 – Главный блок

В данном случае (рисунок 1.2) на вход поступают все те же документы клиента и сам клиент в блок «Собрать информацию о клиенте». На выходе получаем удовлетворяющую информацию или отказ в оказании услуги по причине ненадежности клиента, которая переходит в следующий блок «Выбрать предоставляемую услугу». Далее получаем выбранную услугу, которая входит в блок «Оказать услугу» и на выходе из этого блока получаем оказанную услугу.

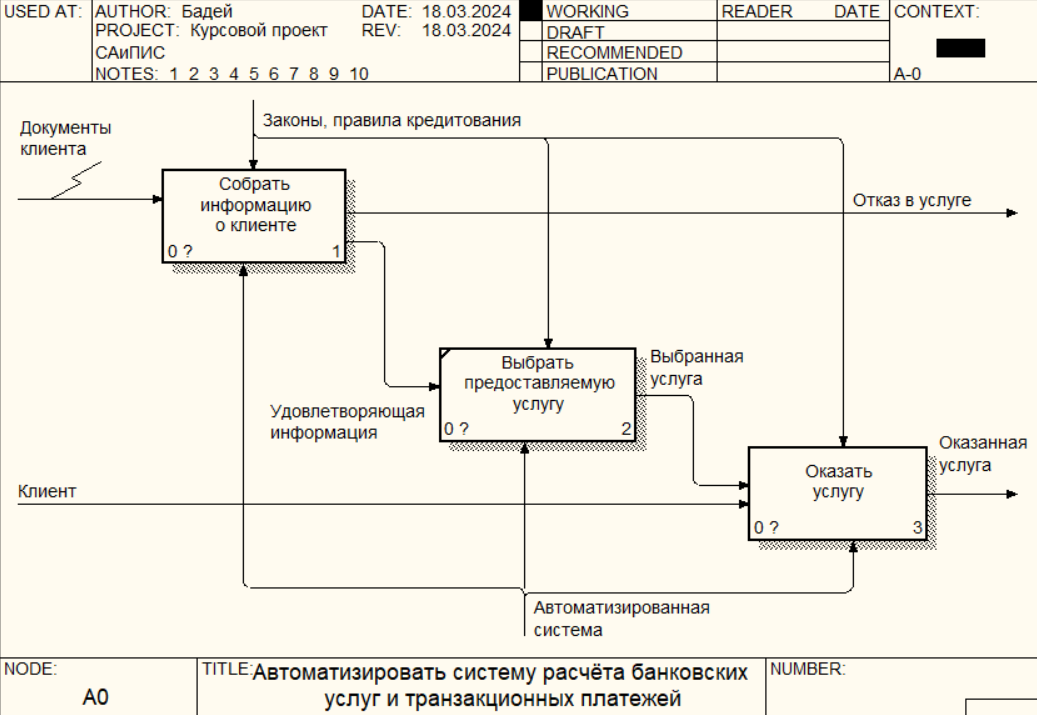


Рисунок 1.2 – Декомпозиция главного блока моделирования

В данном случае на вход поступают все те же документы клиента и сам клиент в блок «Собрать информацию о клиенте». На выходе получаем удовлетворяющую информацию или отказ в оказании услуги по причине ненадежности клиента, которая переходит в следующий блок «Выбрать предоставляемую услугу». Далее получаем выбранную услугу, которая входит в блок «Оказать услугу» и на выходе из этого блока получаем оказанную услугу.

Далее (рисунок 1.3) на вход поступают документы клиента в блок «Обработать документы». На выходе получаем обработанные документы, которые входят в следующий блок – «Проверить надежность клиента в соответствии с оказанием определенных услуг». На выходе получаем надежность клиента, который входит в блок «Занести клиента в соответствующий список». В итоге на выходе из блока получаем отказ в услуге или удовлетворяющую информацию.

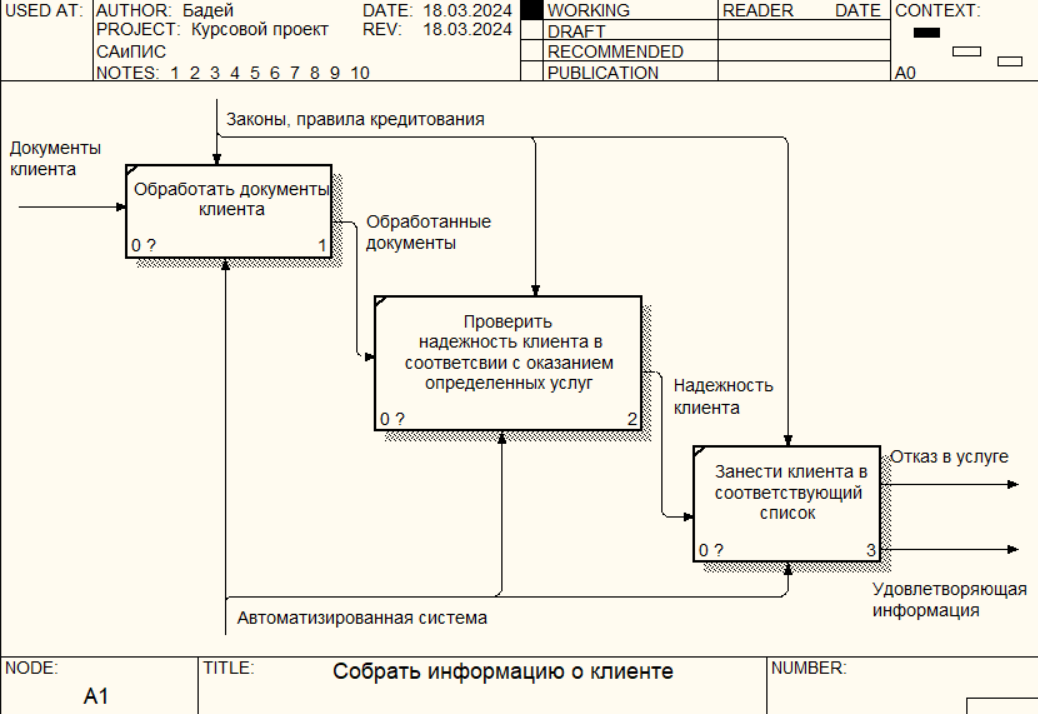


Рисунок 1.3 – Декомпозиция блока «Собрать информацию о клиенте»

Здесь (рисунок 1.4) на вход поступают клиент в блок «Повторно убедиться в надежности клиента» и выбранная услуга в блок «Обратиться в головной офис банка для оказания услуги».

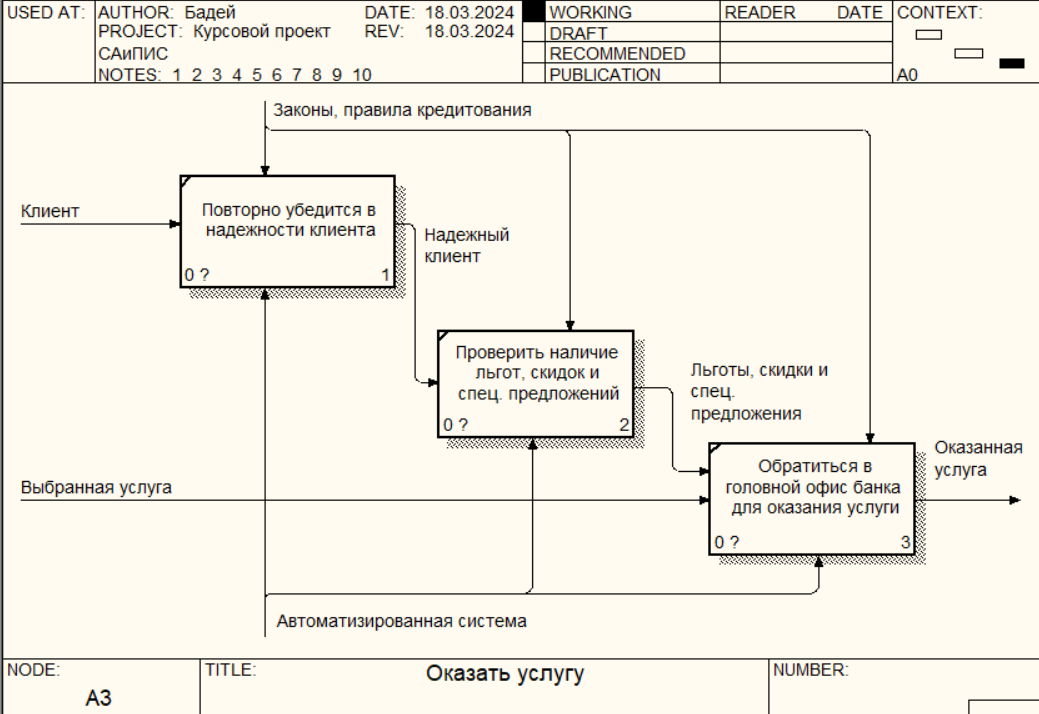


Рисунок 1.4 – Декомпозиция блока «Оказать услугу»

На выходе получаем надежного клиента, который входит в блок «Проверить наличие льгот, скидок и спец. предложений». На выходе из этого блока получаем льготы, скидки и спец. предложения и входим в блок «Обратиться в головной офис банка для оказания услуги». Далее на выходе получаем оказанную услугу.

**1.3 АНАЛИЗ ТРЕБОВАНИЙ К РАЗРАБАТЫВАЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ. СПЕЦИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ**

Программное средство должно быть выполнено как функционирующее клиент-серверное веб-приложение с самостоятельно установленным веб-сервером в архитектуре Model-View-Controller или Model-View-Presenter с использованием технологии Ajax и CommonJS /UMD/ ESmodules модульных концепций. Веб-страницы, объединенные общей темой и дизайном, должны быть созданы с использованием языков HTML, DHTML, JavaScript, XML и содержать гиперссылки для быстрого перехода на другие страницы.

Ко всему веб-приложению для форматирования текста/объектов должна применяться одна каскадная таблица стилей. Данные (логин, пароль), а также ресурсные данные (видео-, звуковые файлы, изображения и т.д.) и иная информация должны храниться в документоориентированной или реляционной базе данных. Конкретные версии фреймворков и технологий, применяемых для реализации программного средства, должны быть актуальными на январь 2023 года.

Серверная часть проекта реализуется на сервере Node.js с использованием npm и веб-фреймворка express. Бизнес-логика приложения должна быть реализована на основе языка JavaScript. Бизнес-логика реализует основные алгоритмы обработки данных. Клиентский интерфейс реализуется с использованием технологий HTML5, CSS3, DHTML, Ajax, актуальных JavaScript библиотек и других методов разработки веб-интерфейса (React/Angular - по согласованию с руководителем). СУБД для создания базы данных (реляционной/нереляционной) выбирается студентом самостоятельно. Схема базы данных должна обеспечивать ссылочную целостность данных.

Пользовательский интерфейс должен обеспечивать доступ ко всем функциям. Система должна реализовать не менее 10 бизнес-операций, не считая простых операций с базой данных (чтение, запись, изменение данных).

Необходимо предусмотреть механизм авторизации всех пользователей. Количество ролей – не менее 2-3 в зависимости от специфики предметной области (администратор, сотрудник компании, гость и т.п.).

**1.4 РАЗРАБОТКА ИНФОРМАЦИОННОЙ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ**

В данной главе будет представлена информационная модель предметной области разрабатываемого программного средства с автоматизированной системой расчёта банковских услуг и транзакционных платежей. Эта модель позволяет структурировать и описать основные сущности, их атрибуты и взаимосвязи в рамках системы (рисунок 1.5).

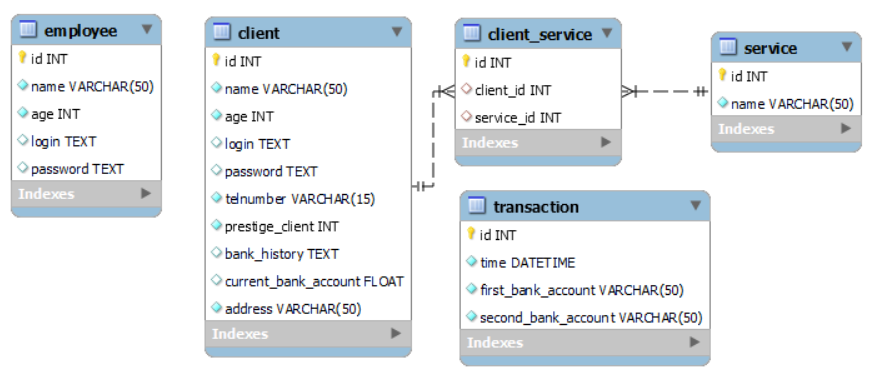


Рисунок 1.5 – Информационная модель предметной области

Таблица Client содержит информацию о клиентах банка, такую как их идентификаторы, имена, возраст, контактные данные и другие атрибуты. Каждая запись в этой таблице представляет отдельного клиента.

Таблица Service хранит список доступных банковских услуг, таких как кредиты, депозиты, переводы и т.д. Каждая запись представляет отдельную услугу, с ее уникальным идентификатором и названием.

Таблица Employee содержит информацию о сотрудниках банка, включая их личные данные и учетные данные для доступа к системе.

Таблица Transaction отражает информацию о проведенных транзакциях между банковскими счетами, включая идентификаторы счетов отправителя и получателя, время выполнения транзакции и ее текущий статус.

Таблица Client\_service представляет связь между клиентами и услугами, которыми они пользуются. Она содержит записи о том, какой клиент использует какую услугу. Каждая запись в этой таблице содержит идентификатор клиента (client\_id) и идентификатор услуги (service\_id).

Таким образом, связь между этими таблицами представляет собой многие-ко-многим. Один клиент может пользоваться несколькими услугами, и каждая услуга может быть использована несколькими клиентами. В таблице «Client\_service» каждая запись определяет отношение между конкретным клиентом и конкретной услугой.

**1.5 UML-МОДЕЛИ ПРЕДСТАВЛЕНИЯ ПРОГРАММНОГО СРЕДСТВА И ИХ ОПИСАНИЕ**

Диаграмма вариантов использования (Use Case) используется для описания поведения системы с точки зрения ее пользователей или внешних акторов. Она помогает идентифицировать и описать различные сценарии использования системы, ее возможности и функциональность.

Главная цель диаграммы вариантов использования заключается в понимании бизнес – требований к системе со стороны конечных пользователей.

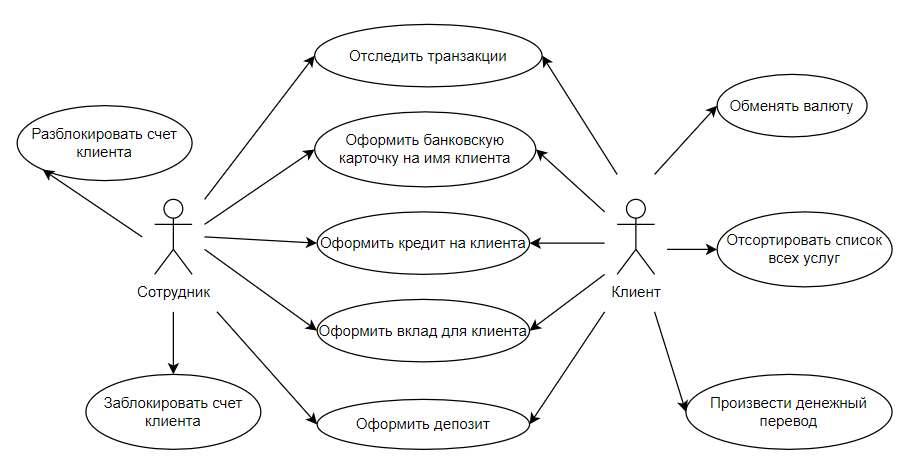


Рисунок 1.6 – Диаграмма вариантов использования

На рисунке 1.6 представлены 10 основных бизнес-функций программы, которые она должна выполнять. Так же на рисунке присутствуют два актера (роли): сотрудник и пользователь.

## ПРИЛОЖЕНИЕ А (обязательное) IDEF0-модель процессов предметной области

