**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc198068226)

[**1.1.** **Исходная постановка задачи** 6](#_Toc198068227)

[**1.2 Среда разработки** 7](#_Toc198068228)

[**1.3 Техническое задание** 10](#_Toc198068229)

[**1.4 Руководство по стилю** 18](#_Toc198068230)

[**1.5 UML-диаграмма** 19](#_Toc198068231)

[**1.7 Схема структуры** 24](#_Toc198068232)

[**2.1 Разработка логической модели данных** 25](#_Toc198068233)

[**2.2. Добавление таблиц к базе данных** 26](#_Toc198068234)

[**2.3. Просмотр связей на диаграмме базы данных** 26](#_Toc198068235)

[**2.4. Определение ограничений, налагаемые на данные** 27](#_Toc198068236)

[**2.5. Определение объема и типов данных** 29](#_Toc198068237)

[**2.6. Определение способа использования данных** 30](#_Toc198068238)

[**2.7. Разработка физической структуры данных** 33](#_Toc198068239)

[**2.7.1. Создание таблицы в БД** 34](#_Toc198068240)

[**2.7.2. Данные таблиц** 34](#_Toc198068241)

[**ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ** 44](#_Toc198068242)

[**3.1. Разработка пользовательского интерфейса** 44](#_Toc198068243)

[**3.2. Листинг программы** 56](#_Toc198068244)

[**3.4. Руководство пользователя** 63](#_Toc198068245)

[**ГЛАВА 4. РАЗРАБОТКА 1С** 75](#_Toc198068246)

[**4.1 Разработка подсистем** 75](#_Toc198068247)

[**4.2 Разработка справочников** 75](#_Toc198068248)

[**4.3 Разработка документов и регистров** 75](#_Toc198068249)

[**4.4 Разработка отчетов** 75](#_Toc198068250)

# **ВВЕДЕНИЕ**

База данных системы учета валютных операций представляет собой организованную систему хранения и управления информацией, касающейся заявок на конвертацию, поступлений и списаний валюты, курсов валют и операций клиентов. Эта база данных служит важным инструментом для автоматизации процессов, повышения эффективности учета валютных операций и улучшения обслуживания клиентов.

Предприятия, осуществляющие валютные операции, нуждаются в защите и оптимизации своих данных. Они занимаются конвертацией валюты, учетом поступлений и списаний, ведением клиентских счетов и предоставлением информации регуляторам. Для успешного ведения валютных операций необходима надежная система учета, которая обеспечит:

* Организацию и структурирование информации: Базы данных позволяют аккуратно организовывать информацию о валютных счетах, курсах, операциях, клиентах и сотрудниках. Это значительно облегчает поиск нужной информации и ускоряет процесс обработки операций.
* Обеспечение целостности данных: Важно, чтобы данные о валютных остатках, курсах и операциях были точными и актуальными. База данных помогает предотвратить дублирование записей и обеспечивает согласованность информации.
* Совместное использование данных: Множество сотрудников может одновременно работать с базой данных, что упрощает процессы проведения операций, формирования отчетности и взаимодействия с клиентами. Это повышает эффективность работы валютного отдела.
* Повышение эффективности бизнес-процессов: С автоматизацией рутинных задач, таких как формирование заявок на конвертацию, учет поступлений и списаний и расчет курсовых разниц, предприятие может сосредоточиться на предоставлении качественных услуг клиентам.

В рамках данной работы будет использоваться Microsoft SQL Server для создания и управления базой данных, а также Visual Studio для разработки приложения, которое будет взаимодействовать с этой базой. Такой подход поможет создать эффективную систему управления данными для учета валютных операций.

**Тестирование базы данных**

Тестирование базы данных является ключевым этапом разработки, целью которого является обеспечение ее соответствия требованиям и корректной работы. Тестирование включает в себя несколько аспектов:

* Функциональное тестирование: Проверяет правильность выполнения функций базы данных и их соответствие техническому заданию (например, правильность расчета курсовых разниц, корректность формирования проводок).
* Тестирование производительности: Анализирует, насколько быстро и эффективно база данных может обрабатывать запросы (например, скорость формирования отчетов по валютным операциям, время проведения конвертации).
* Тестирование нагрузки: Определяет, как база данных справляется с большим числом одновременных запросов (например, при массовой загрузке выписок из банков, при одновременной работе большого количества пользователей).
* Тестирование безопасности: Выявляет уязвимости базы данных и обеспечивает защиту информации от несанкционированного доступа (например, защита данных о валютных счетах клиентов, предотвращение несанкционированной конвертации валюты).

Тестирование позволяет выявить и устранить ошибки на ранних этапах, что значительно улучшает качество базы данных и снижает риск проблем в будущем.

Основные задачи, которые планируется решить с помощью новой системы:

* Автоматизация бизнес-процессов: Улучшение процессов формирования заявок на конвертацию, учета поступлений и списаний валюты, расчета курсовых разниц и формирования отчетности.
* Повышение прозрачности и контроля: Создание единой системы учета всех валютных операций, что позволит отслеживать каждый этап, включая движение средств по счетам, конвертацию валюты и взаимодействие с банками.
* Улучшение коммуникации: Облегчение обмена информацией между различными отделами предприятия, а также с банками и регуляторами.
* Снижение издержек: Автоматизация рутинных операций снизит затраты на бумажную документацию, ошибки ручной обработки данных и другие потери.
* Соответствие требованиям регуляторов: Упрощение подготовки отчетности, требуемой регулирующими органами в сфере валютного контроля.

Таким образом, разработка базы данных для системы учета валютных операций — это не только ответ на современные вызовы цифровизации, но и важный шаг к улучшению качества обслуживания клиентов, повышению эффективности работы и конкурентоспособности на рынке.

**ГЛАВА 1. ПРЕДМЕТНАЯ ОБЛАСТЬ.**

# **Исходная постановка задачи**

Разрабатывается приложение с целью более удобного предоставления отчета об прибыли и затратах автостанции, а также улучшить удобство пользования автостанции для граждан

Таблица №1

Система учета валютных операций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Поле | Тип | Размер | Описание |
| 1 | ID Operacii | Числовой | 10 | Номер операции |
| 2 | Symma | Текстовый | 25 | Определенная сумма для оплаты |
| 3 | Tip operacii | Текстовый | 25 | Тип операции |
| 4 | Data | Дата | 10 | Дата проведения операции |
| 5 | ID Svaznoi operacii | Числовой | 10 | Идентификатор связанной операции |
| 6 | Polzovatel | Текстовый | 25 | Имя пользователя выполнившего операцию |
| 7 | Opisanie | Текстовый | 25 | Дополнительное описание или комментарий к операции. |
| 8 | Kyrsovaya raznitsa | Числовой | 10 | Сумма курсовой разницы, возникшей в результате операции. |
| 9 | Osnovanie | Числовой | 10 | Реквизиты документа, на основании которого проведена операция |
| 10 | Status | Числовой | 10 | Статус операции |
| 11 | Polychen | Дата |  | Дата поступление валюты/средств на счет |
| 12 | FIO | Текстовый | 25 | ФИО Гражданина для заявки |

**1.2 Среда разработки**

**Visual Studio** — это одна из самых популярных интегрированных сред разработки приложение, разработанная Microsoft. Она предлагает широкий набор инструментов и функций, которые делают процесс разработки программного обеспечения более эффективным и удобным.

Плюсы данной программы: достаточно сильно улучшают и упрощают процесс разработки вот собственно и они:

* + Многофункциональность и поддержка языков
  + Мощные инструменты разработки
  + Интеграция с системами контроля версий
  + Гибкость и расширяемость
  + Простота в использовании

**SQL Server Management Studio** — это мощная интегрированная среда для работы с Microsoft SQL Server. Она предоставляет обширный набор инструментов для управления базами данных и позволяет выполнять множество задач, связанных с разработкой и администрированием.

Так же и с Sql server Management Studio, вот пункты которые я считаю наиболее привлекательными которые я заметил при выборе этого приложения для разработки:

* Удобный интерфейс
* Поддержка различных версий SQL Server
* Построение и визуализация схем баз данных
* Управление безопасностью и доступом
* Импорт и экспорт данных
* Большое сообщество и поддержка

УТВЕРЖДАЮ УТВЕРЖДАЮ

Директор ГБПОУ «ВВК» Директор филиала

\_. \_ . \_ \_\_\_\_\_\_\_/Арсеньев А.В./ \_. \_. \_ \_\_\_\_\_\_/Михайлов А.С./

СОГЛАСОВАНО

Зам. директора ГБПОУ «ВВК»

\_ . \_ . \_ \_\_\_\_\_\_\_\_\_\_\_\_/ Бордодымова Т.В./

Система автостанций

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

02068120.508100.027-02 2А 01.М-ЛУ

Листов 6

Представители разработчика

Начальник отдела ИУ

\_ . \_ . \_ \_\_\_\_\_\_\_/Яковлева Ж.С./

Руководитель разработки

Ведущий программист

\_ . \_ . \_ \_\_\_\_\_\_\_/Соловьев А.А./

2024

УТВЕРЖДЕНО Приложение № 1 02068120.508100.027-022А 01.М-ЛУ к Договору № 2 от 02.04.06

Система автостанций

Техническое задание

02068120.508100.027-02 2А 01.М

Листов 6

2024

# **1.3 Техническое задание**

**Введение**

Автоматизированная информационная система учета валютных операций (АИС) предназначена для автоматизации процессов учета, контроля и анализа валютных операций в организации. Система обеспечит хранение, обработку и предоставление информации о валютных операциях, что позволит оптимизировать работу бухгалтерии, финансового отдела и руководства.

**Основание для разработки**

Документы: План разработки АИС УВО

Организация: ООО «ГлобалИмпорт» Документ одобрен 15.01.2025

**Назначение разработки**

Функциональное назначение программы заключается в автоматизации учета, контроля и анализа валютных операций, предоставлении информации для принятия управленческих решений, а также формировании отчетности в соответствии с требованиями законодательства.

**Требования к программе или программному изделию**

**Требования к функциональным характеристикам**

Программа должна содержать в себе следующие функции:

* Регистрация пользователей (Администраторов и Операторов).
* Управление учетными записями пользователей (создание, редактирование, удаление).
* Назначение ролей и прав доступа.

**Управление справочниками:**

* Ведение справочника валют.
* Ведение справочника контрагентов.
* Ведение справочника счетов.
* Ведение справочника видов валютных операций.

**Учет валютных операций:**

* Ввод данных о валютных операциях (покупка, продажа, конвертация, переводы, поступления, выплаты и т.д.).
* Автоматическое формирование проводок.
* Контроль соответствия операций установленным лимитам и ограничениям.
* Учет курсовых разниц.

**Учет курсов валют:**

* Ручной ввод курсов валют.
* Автоматическая загрузка курсов валют из внешних источников (например, с сайта ЦБ).
* История изменения курсов валют.

**Формирование отчетности:**

* Формирование регламентированной отчетности по валютным операциям.
* Формирование аналитических отчетов по запросам пользователей.
* Экспорт отчетов в различные форматы (например, Excel, PDF).

**Поиск и фильтрация данных:**

* Поиск валютных операций по различным критериям (дата, контрагент, счет, сумма и т.д.).
* Фильтрация данных по заданным параметрам.

**Аудит:**

* Ведение журнала действий пользователей.
* Отслеживание изменений в базе данных.

**Интеграция:**

* Возможность интеграции с другими системами (например, с бухгалтерской системой).

**Поддержка:**

* Предоставление справочной информации и документации.
* Возможность связи с разработчиком в случае неисправности приложения.

**Главное меню:**

* Интуитивно понятное главное меню для доступа ко всем функциям системы.
* Кнопки для быстрого выбора нужной категории.

**Навигация:**

* Переключатели для просмотра нескольких записей (пагинация).
* Возможность добавления, удаления, редактирования и сохранения записей в базе данных через приложение.

**Требование к надежности**

* Приложение должно быть надежным и обеспечивать сохранность данных. Заказчик должен обеспечить:
* Бесперебойное питание и работу сервера.
* Использование лицензионного программного обеспечения.
* Регулярное создание резервных копий базы данных.
* Своевременную установку обновлений программного обеспечения.
* Обеспечение безопасности сети и серверов.
* Условие эксплуатации
* Использовать приложение только в законных целях, связанных с учетом валютных операций организации.
* Нести ответственность за правильность и достоверность вводимых данных.
* Соблюдать правила безопасности и не допускать несанкционированного доступа к системе.
* Принимать приложение "как есть" без гарантий (кроме предусмотренных договором поддержки).
* Соблюдать изменения в условиях использования, предоставляемых разработчиком.
* Принять возможность прекращения доступа при нарушении условий использования.

**Требования к составу и параметрам технических средств**

Сервер БД (SQL Server, PostgreSQL или аналог).

Процессор: ARM64 или x64; Рекомендуется использовать четырехъядерный процессор или лучше. Процессоры ARM 32 не поддерживаются.

Не менее 4 ГБ ОЗУ. На используемые ресурсы влияет множество факторов; для стандартных профессиональных решений рекомендуется 16 ГБ ОЗУ.

Место на жестком диске: от 850 МБ до 5 ГБ свободного места в зависимости от кол-ва заполненной информации. Для повышения производительности рекомендуется устанавливать ОС и БД на твердотельном накопителе (SSD).

Операционная система: Windows Server (2012 и выше), Linux (CentOS, Ubuntu и др.).

**Требования к информационной и программной совместимости**

Проектирование базы данных должно быть выполнено в соответствии с требованиями предметной области и обеспечивать целостность и консистентность данных.

Приложение 1 (Backend/API) разрабатывается на [указать язык программирования, например, C#, Java, Python] с использованием [указать фреймворк, например, .NET Core, Spring, Django].

Приложение 2 (Frontend/UI) разрабатывается на [указать язык программирования, например, JavaScript, TypeScript] с использованием [указать фреймворк, например, React, Angular, Vue.js].

База данных должна быть совместима с используемым сервером БД (SQL Server, PostgreSQL и т.д.).

Приложение должно поддерживать возможность интеграции с другими системами (например, с бухгалтерской системой 1С).

**Требования к маркировке и упаковке**

Маркировка содержит:

Название приложения: АИС Учет валютных операций.

Версию приложения.

Имя разработчика.

Дата реализации приложения.

Контактную информацию разработчика (электронную почту и номер телефона).

Приложение после разработки упаковывается в виде инсталляционного пакета для развертывания на сервере заказчика.

**Требование к транспортировке и хранению**

Инсталляционный пакет приложения должен храниться в защищенном месте для предотвращения несанкционированного доступа.

Передача инсталляционного пакета заказчику осуществляется по защищенным каналам связи (например, с использованием шифрования).

**Требования к программной документации**

Состав документации, который должен быть составлен:

Руководство оператора (для пользователей, вводящих данные).

Руководство пользователя (для пользователей, формирующих отчеты и анализирующих данные).

Руководство системного администратора (для администраторов, обслуживающих систему).

Техническое задание (настоящий документ).

Описание базы данных (схема БД, описание таблиц и полей).

Тестовая документация (план тестирования, сценарии тестирования, результаты тестирования).

**Технико-экономические показатели**

Экономическая эффективность внедрения АИС УВО будет оценена после завершения проекта на основании следующих показателей:

* Сокращение времени на обработку валютных операций.
* Снижение количества ошибок при вводе данных.
* Улучшение контроля за соблюдением валютного законодательства.
* Повышение эффективности принятия управленческих решений на основе данных о валютных операциях.
* Сокращение затрат на оплату труда.
* Стадии и этапы разработки
* Разработка технического задания.
* Разработка документации.
* Проектирование БД и приложения.
* Реализация (разработка программного кода).
* Тестирование.
* Внедрение.
* Сопровождение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии разработки документации должен быть выполнен этап разработки проектной документации.

На стадии проектирования БД и приложения должен быть выполнен этап создания схемы БД, разработки интерфейса приложения и проектирования архитектуры системы.

На стадии реализации должен быть выполнен этап написания программного кода приложения.

На стадии тестирования должен быть выполнен этап проверки работоспособности приложения и выявления ошибок.

На стадии внедрения должен быть выполнен этап установки приложения на сервере заказчика и обучения пользователей.

На стадии сопровождения должен быть выполнен этап исправления ошибок и добавления новых функций в приложение.

Порядок контроля и приемки

Испытание программы должно проводиться рядом методик, согласованных разработчиком с заказчиком (так же соблюдение требования к надежности и безопасности).

При прохождении испытательного срока, Заказчик совместно с разработчиком подписывает акт приемки приложения в эксплуатацию.

**Приложение**

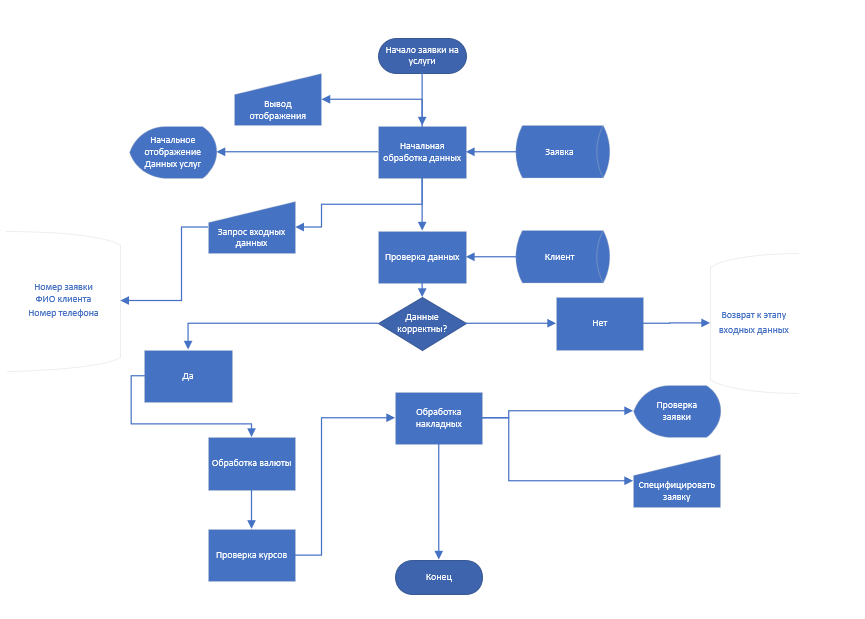


Рис. 1 «Блок-схема»

## **1.4 Руководство по стилю**

**Общие требования**

Не допускать орфографические и грамматические ошибки. Размер форм, элементов на форме должны быть одинаковы, они не должны отличаться.

**Использование логотипа**

Логотип используется в каждой форме справа в верхнем углу экрана, а также на главной странице

**Шрифт**

Шрифт будет использоваться Times New Roman, диапазон примерных размеров от 14 до 20.

**Основные цвета**

|  |
| --- |
| #18F0B7 |
|  |

**Дополнительные**

|  |
| --- |
| #79FF6B |
|  |

**Для текста**

|  |  |
| --- | --- |
| #FF0000 | #000000 |
|  |  |

## **1.5 UML-диаграмма**

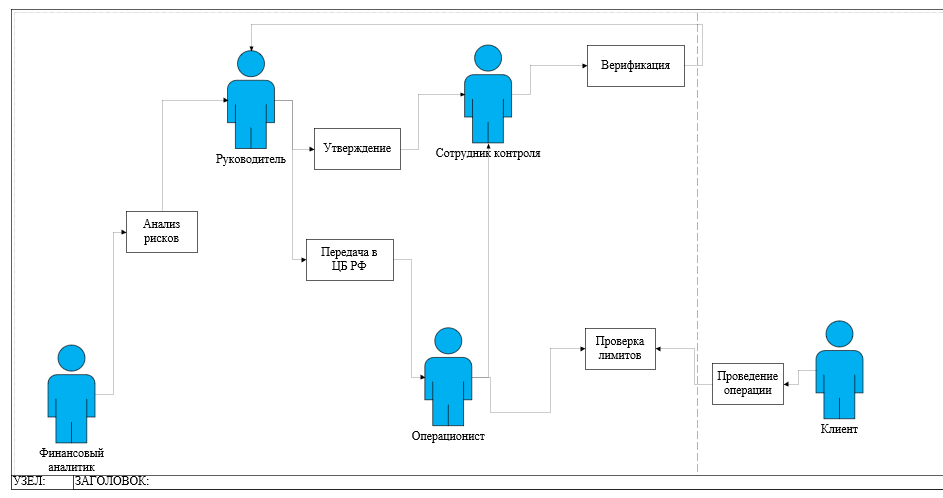
****

Рис. 2 «Диаграмма вариантов использования»

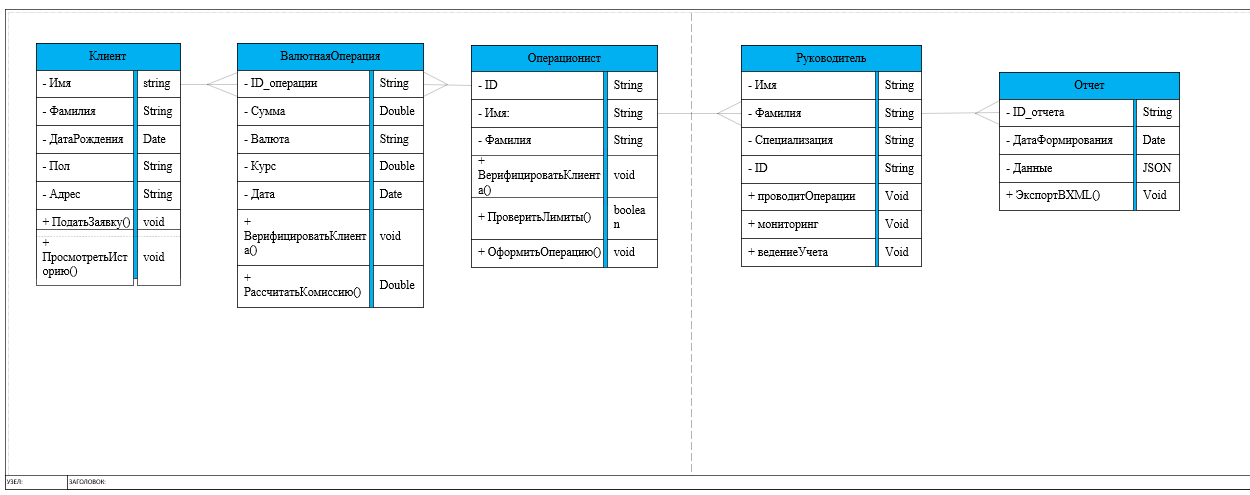


Рис. 3 «Диаграмма классов»

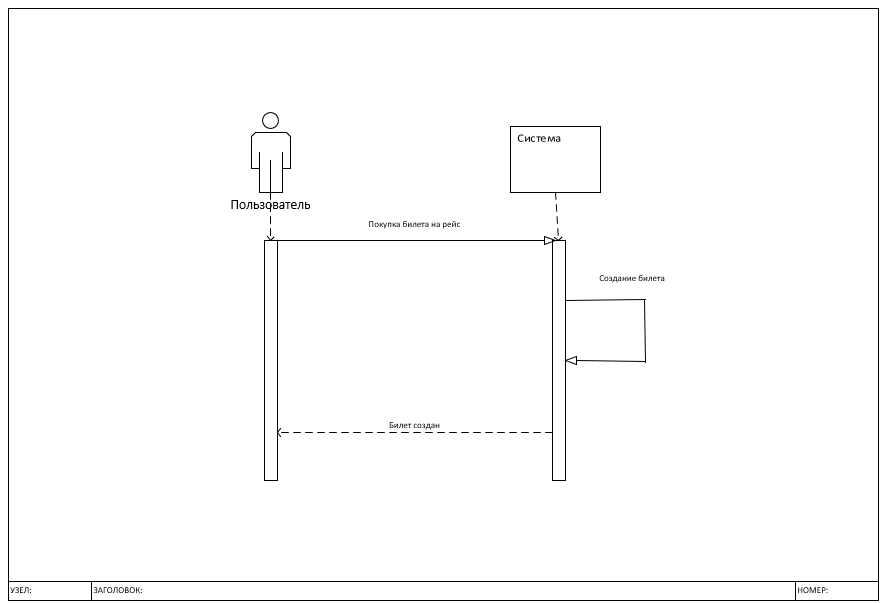


Рис. 4 «Диаграмма поведения»

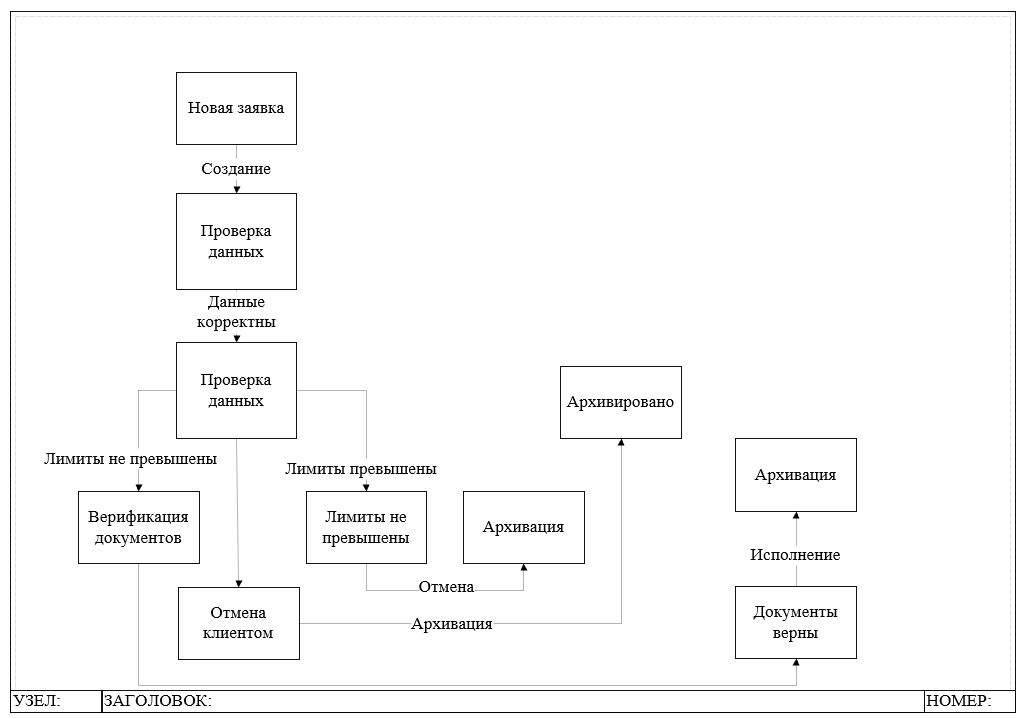


Рис. 5 «Диаграмма состояния»

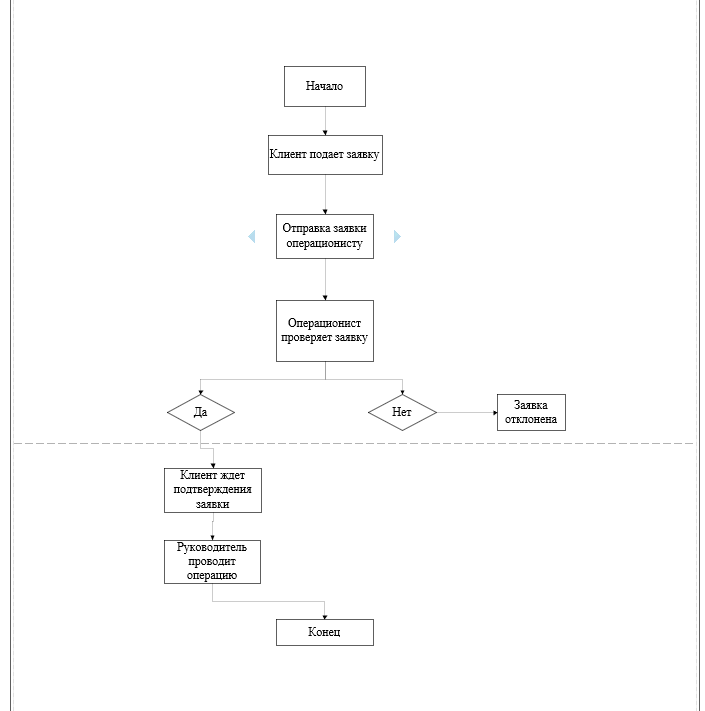


Рис. 6 «Диаграмма деятельности»

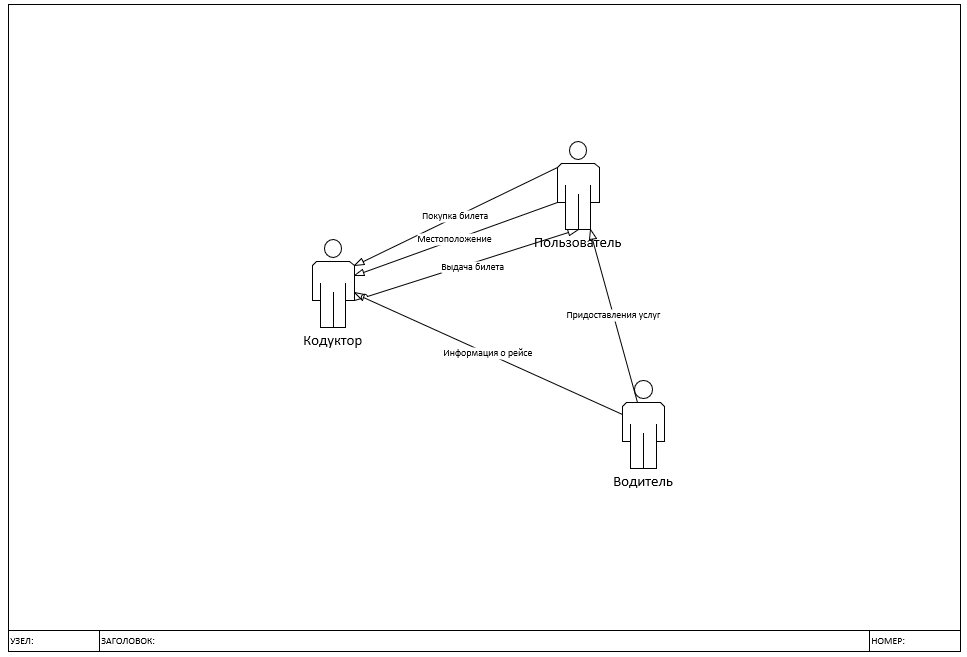


Рис. 7 «Диаграмма взаимодействия»

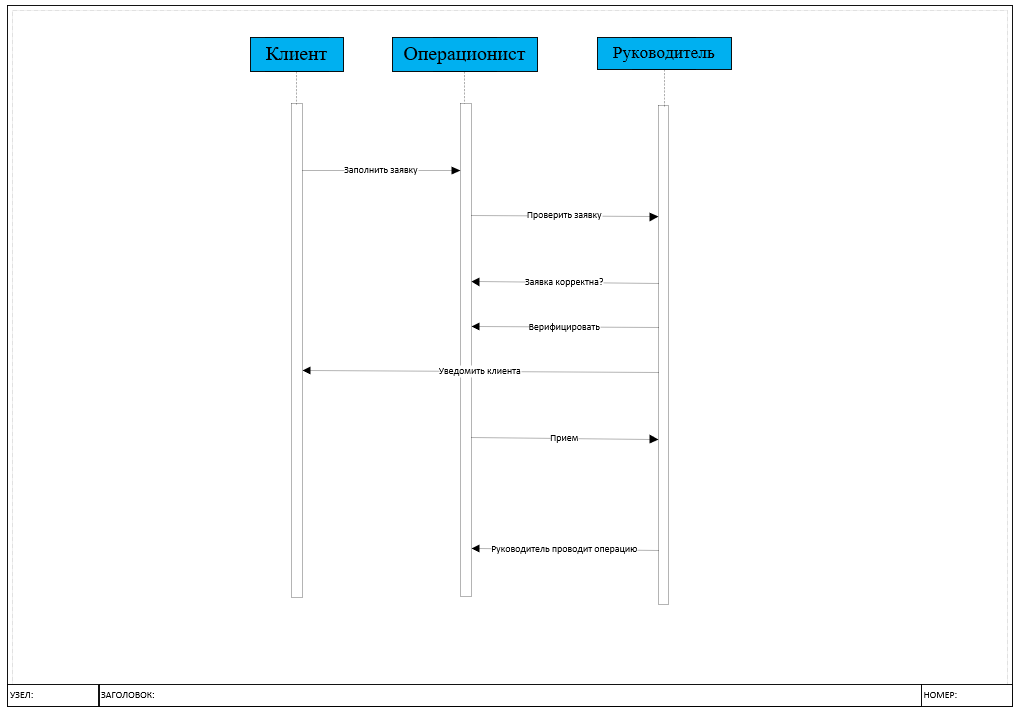


Рис. 8 «Диаграмма последовательности»

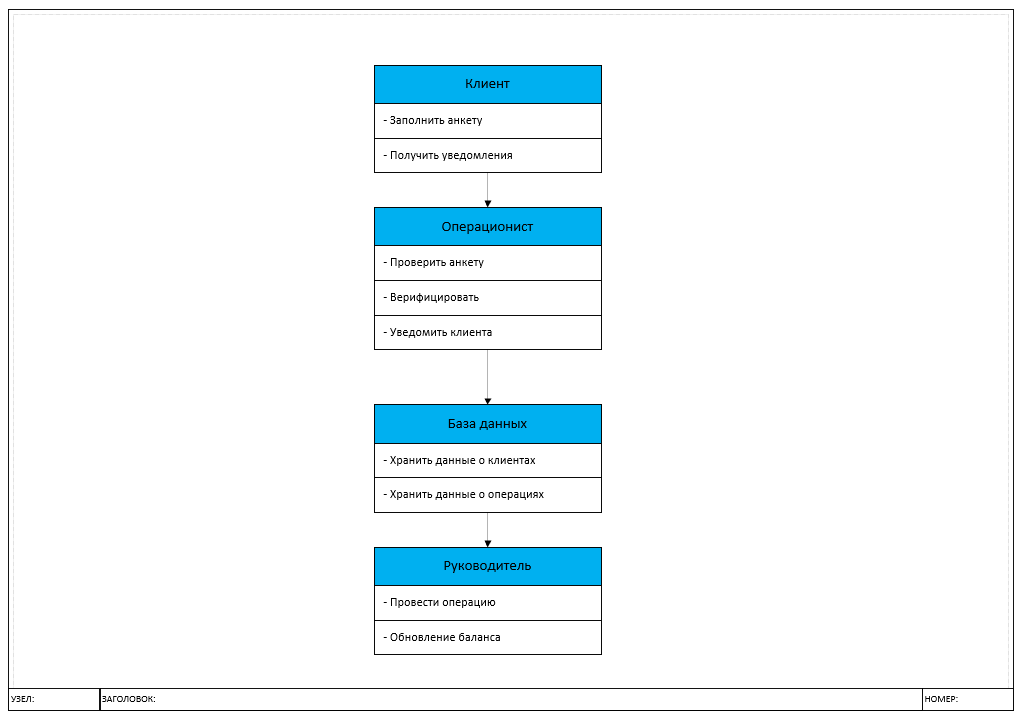


Рис. 9 «Диаграмма компонентов»

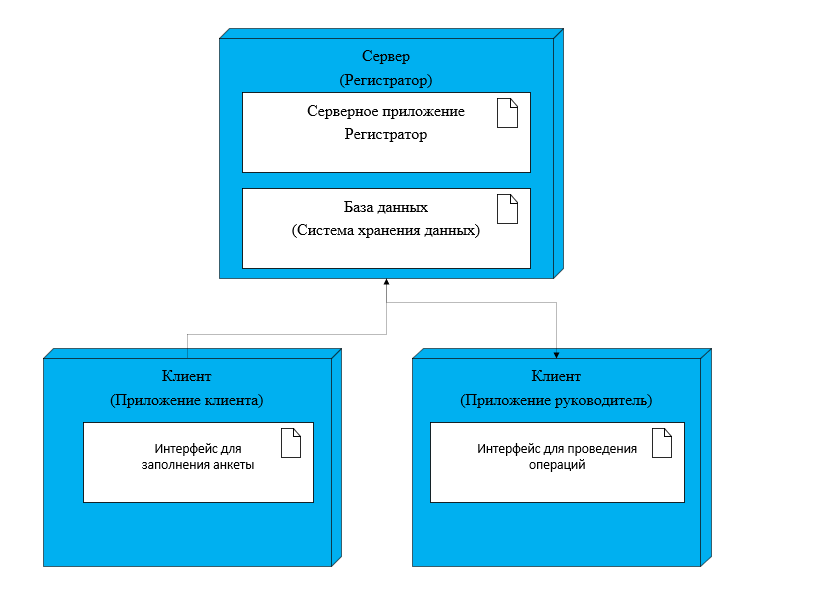


Рис. 10 «Диаграмма развертывания»

**1.6 Wireframe**

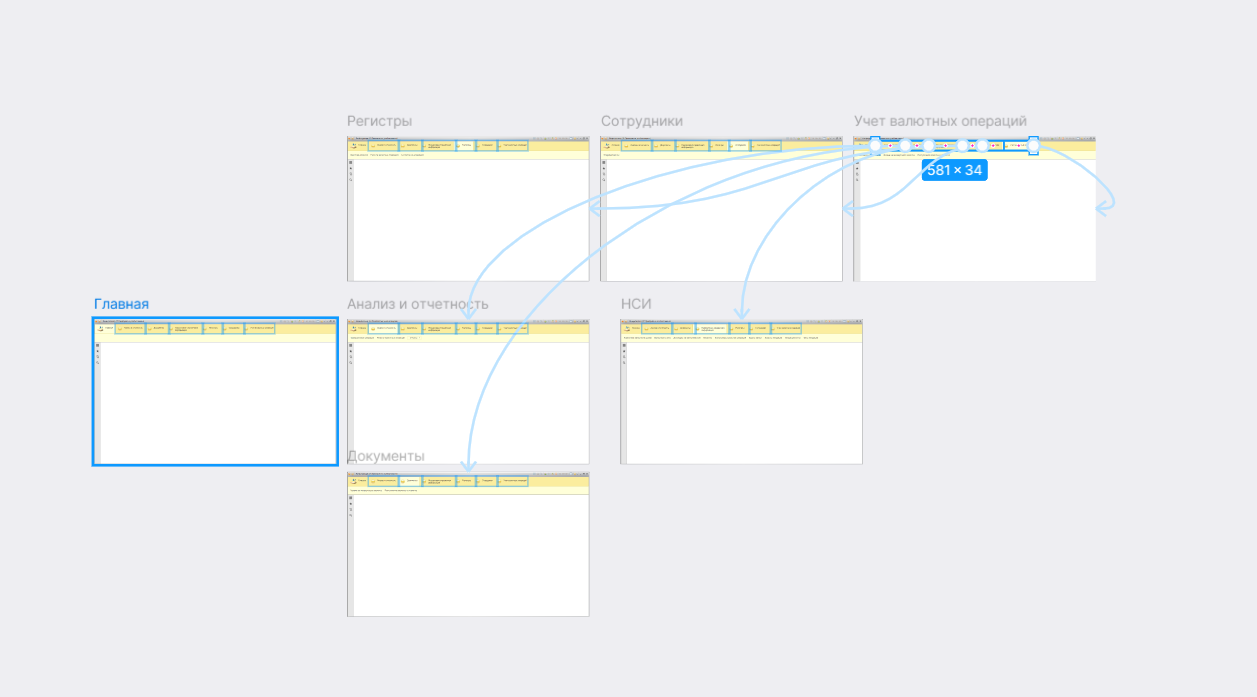
****

Рис. 11 «Warframe»

Cсылка на проект:

<https://www.figma.com/design/rpRRTtXaPbk0mLxxaM8j0w/User-Flows--%C2%A0Annotation-Kit--Community-?node-id=802-747&m=dev&t=B9WEzny1W7otzQwB-1>

# **1.7 Схема структуры**

****

Рис. 12 «Комплексная схема»**ГЛАВА 2. РАЗРАБОТКА БАЗЗ ДАННЫХ**

**2.1 Разработка логической модели данных**

Сущность «Валюты»: id, код, название.

Сущность «Группы пользователей»: id, название.

Сущность «Журнал действий»: id, пользователь\_id, действие, дата.

Сущность «Курс валюты»: id, валюта\_id, дата, курс.

Сущность «Отчеты»: id, пользователь\_id, дата, содержимое.

Сущность «Переводы»: id, счет\_отправителя\_id, счет\_получателя\_id, сумма, дата.

Сущность «Пользователи»: id, имя, email.

Сущность «Пользователи группы»: пользователь\_id, группа\_id.

Сущность «Счета»: id, пользователь\_id, валюта\_id, баланс.

Сущность «Транзакции»: id, счет\_id, сумма, дата, тип.

**Уникальный идентификатор сущностей.**

Сущность «Валюты»: id\_Валюты

Сущность «Группы пользователей»: id\_Пользователя

Сущность «Журнал действий»: id\_Записи

Сущность «Курс валюты»: id\_Курса

Сущность «Отчеты»: id\_Отчета

Сущность «Переводы»: id\_Перевода

Сущность «Пользователи»: id\_Пользователя

Сущность «Пользователи группы»: пользователь\_id, группа\_id.

Сущность «Счета»: id\_Счета

Сущность «Транзакции»: id\_Транзакции

## **2.2. Добавление таблиц к базе данных**

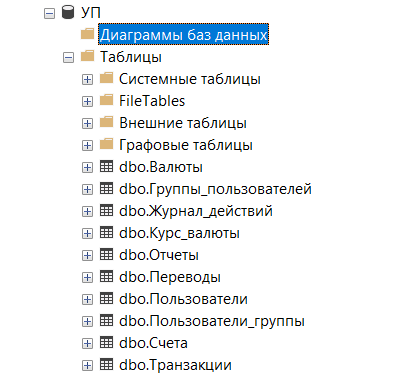


Рис. 13 «Список таблиц»

## **2.3. Просмотр связей на диаграмме базы данных**

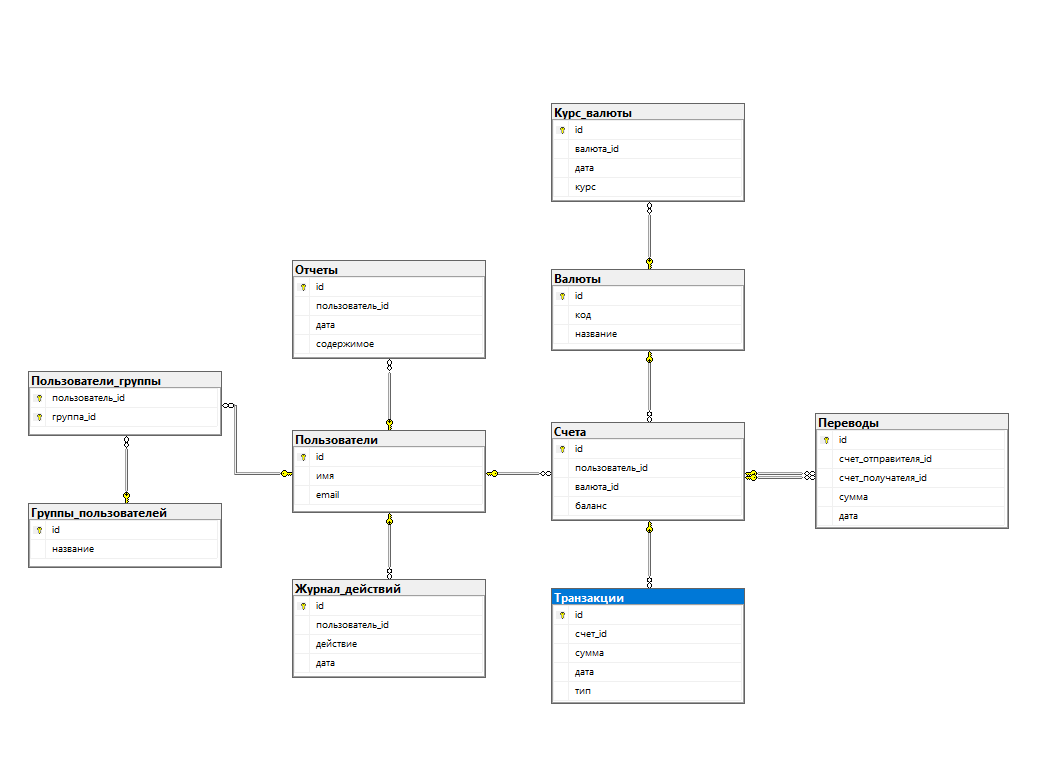


Рис. 14 «Диаграмма базы данных»

## **2.4. Определение ограничений, налагаемые на данные**

Таблица «Валюты»

Id\_Валюты: int, уникальное значение, не может быть null.

Код: nvarchar(3), код валюты.

Название: nvarchar(100), название валюты.

Таблица «Группы пользователей»

ID\_Пользователя: int, уникальное значение, не может быть null.

Название: nvarchar(100).

Таблица «Журнал действий»

Id\_Действия: int, уникальное значение, не может быть null.

Пользователь\_id: int, уникальное значение, не может быть null.

Действие: nvarchar(255).

дата: date.

Таблица «Курс Валюты»

ID\_Валюты: int, уникальное значение, не может быть null.

Валюта\_id: int, уникальное значение, не может быть null.

Дата: date.

Курс: decimal(15, 4).

Таблица «Отчеты»

ID\_Отчета: int, уникальное значение, не может быть null.

Пользователь\_id: int, уникальное значение, не может быть null.

Дата: date.

Содержимое: nvarchar(MAX).

Таблица «Переводы»

ID\_Перевода: int, уникальное значение, не может быть null.

Счет\_отправителя\_id: int,не может быть null.

Счет\_получателя\_id: int, не может быть null.

Сумма: decimal(15, 2).

Дата: date.с

Таблица «Пользователи»

ID\_Пользователя: int, уникальное значение, не может быть null.

Имя: nvarchar(100)

email: nvarchar(100)

Таблица «Пользователи групп»

Пользователь\_id: int, уникальное значение, не может быть null.

Группа\_id: int, уникальное значение, не может быть null.

Таблица «Счета»

ID\_Счета: int, уникальное значение, не может быть null.

Пользователь\_id: int, не может быть null.

Валюта\_id: int, не может быть null.

Баланс: decimal(15, 2).

Таблица «Транзакции»

ID\_Транзакции: int, уникальное значение, не может быть null.

Счет\_id: int, не может быть null.

Сумма: decimal(15, 2)

Дата: date.

Тип: nvarchar(10).

**2.5. Определение объема и типов данных**

Для разработки ПО «Ветеринарной аптеки» следует выделить след категории данных. Это позволить организовать и структурировать информацию о товарах, услугах и цене для нуждающихся в этом питомцах. Категории данных включают:

* Валюты
* Группы пользователей
* Журнал действий
* Курс валюты
* Отчеты
* Переводы
* Пользователи
* Пользователи группы
* Счета
* Транзакции

Таблица №2

Описание таблиц

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Валюты | Сущность «Валюты»: Номер Валюты. |
| Группы пользователей | Сущность «Группы пользователей»: Номер Пользователя. |
| Журнал действий | Сущность «Журнал действий»: Номер Действия. |
| Курс валюты | Сущность «Курс валюты»: Номер Курса. |
| Отчеты | Сущность «Отчеты»: Номер Отчета. |
| Переводы | Сущность «Переводы»: Номер Перевода. |
| Пользователи | Сущность «Страховка»: Номер Пользователя. |
| Пользователи группы | Сущность «Автобус»: Номер Группы Пользователя. |
| Счета | Сущность «Льготного проездного»: Номер Счета. |
| Транзакции | Сущность «Клиенты»: Номер Транзакции. |

Таблица №3

Объем данных таблиц

|  |  |
| --- | --- |
| **Название таблицы** | **Объем данных** |
| Валюты | 20 |
| Группы пользователей | 20 |
| Журнал действий | 20 |
| Курс валюты | 20 |
| Отчеты | 20 |
| Переводы | 20 |
| Пользователи | 20 |
| Пользователи группы | 20 |
| Счета | 20 |
| Транзакции | 20 |

## **2.6. Определение способа использования данных**

Для эффективной работы автоматизированной системы учета валютных операций (АИС), необходимо четко определить роли пользователей и их задачи. Предлагается разделить пользователей на две основные категории: Администратор и Пользователь (Оператор).

**Администратор**

Администратор обладает полным доступом ко всем функциям системы. В его обязанности входит управление пользователями, настройка системы и обеспечение ее бесперебойной работы, а также контроль за соблюдением правил учета валютных операций.

**Основные задачи:**

Управление пользователями:

* Создание, редактирование и удаление учетных записей пользователей (операторов).
* Назначение ролей и прав доступа (например, права на ввод транзакций, формирование отчетов, просмотр истории операций и т.д.).
* Блокировка/разблокировка учетных записей.

**Настройка системы:**

Настройка справочников валют, курсов валют и других параметров, необходимых для корректного учета операций.

* Определение лимитов валютных операций для различных категорий пользователей и счетов.
* Настройка параметров безопасности системы (например, требования к сложности паролей, политики аудита).
* Настройка интеграции с другими системами (например, банковскими системами, системами отчетности).

**Мониторинг и отчетность:**

* Анализ данных о валютных операциях, выявление подозрительных транзакций и нарушений.
* Формирование регламентированных отчетов для контролирующих органов (например, отчеты о валютном контроле).
* Подготовка аналитических отчетов по запросам руководства.
* Мониторинг производительности системы и выявление потенциальных проблем.

**Обслуживание данных:**

* Создание резервных копий базы данных для предотвращения потери информации.
* Восстановление данных из резервных копий в случае сбоев.
* Обеспечение безопасности данных (защита от несанкционированного доступа, модификации и удаления).
* Аудит действий пользователей в системе.
* Оптимизация базы данных для повышения производительности.
* Архивация старых данных.

**Пользователь (Оператор)**

Пользователь (Оператор) работает с системой для выполнения ежедневных задач, связанных с учетом валютных операций. Его права доступа ограничены в соответствии с его ролью и обязанностями.

**Основные задачи:**

Ввод данных:

* Ввод информации о валютных операциях (покупка, продажа, конвертация, перевод и т.д.).
* Ввод данных о контрагентах и счетах.
* Ввод данных о курсах валют (при отсутствии автоматической загрузки).
* Просмотр информации:
* Просмотр истории валютных операций по счетам и контрагентам.
* Просмотр текущих курсов валют.
* Просмотр остатков на валютных счетах.
* Просмотр отчетов (в соответствии с правами доступа).
* Формирование документов:
* Формирование первичных документов по валютным операциям (например, ордеров, выписок).

**Контроль:**

* Проверка правильности введенных данных.
* Контроль соблюдения лимитов валютных операций.

## **2.7. Разработка физической структуры данных**

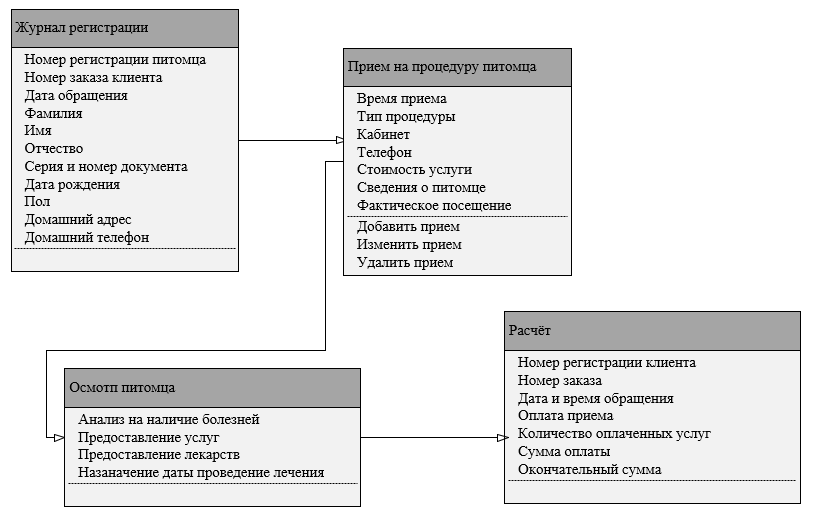


Рис. 15 «Физическая структура данных»

## **2.7.1. Создание таблицы в БД**

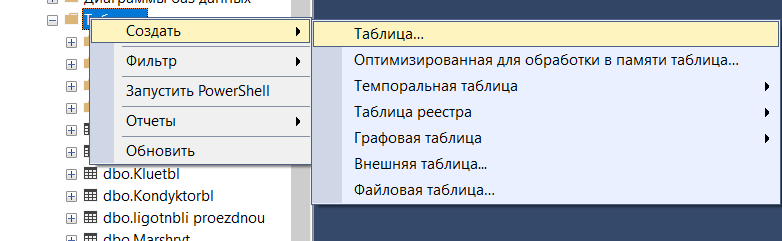


Рис. 16 «Создание таблицы»

## **2.7.2. Данные таблиц**

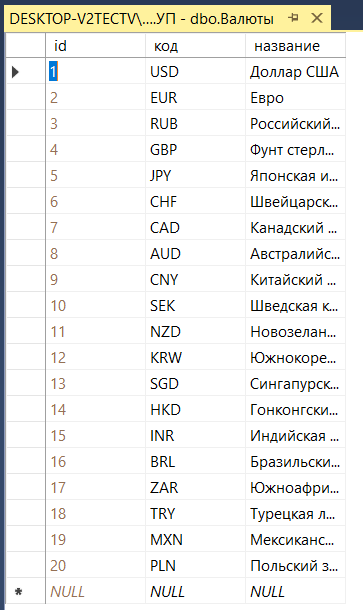


Рис. 17 Таблица «Валюты»

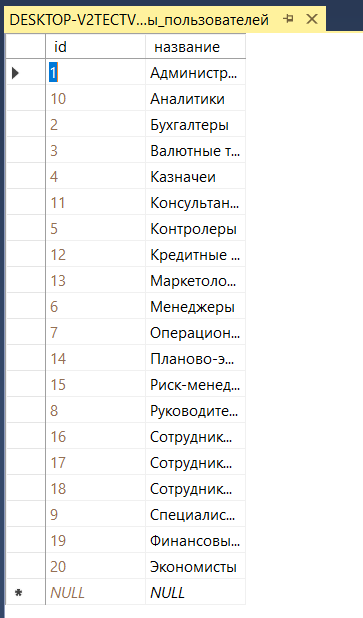


Рис. 18 Таблица «Группы пользователей»

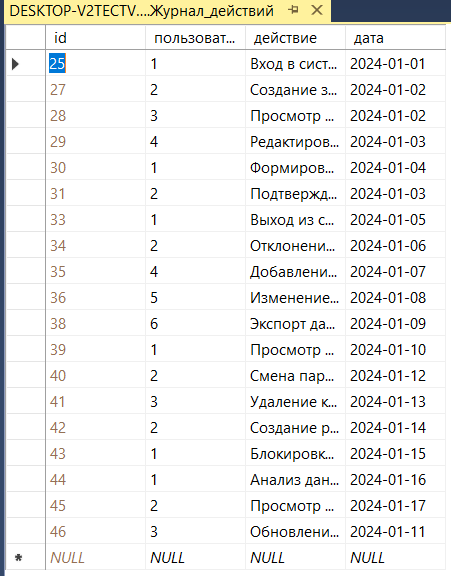


Рис. 19 Таблица «Журнал действий»

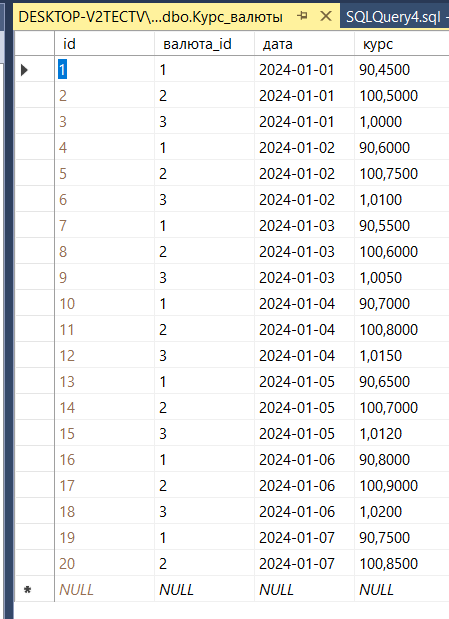


Рис. 20 Таблица «Курс валюты»

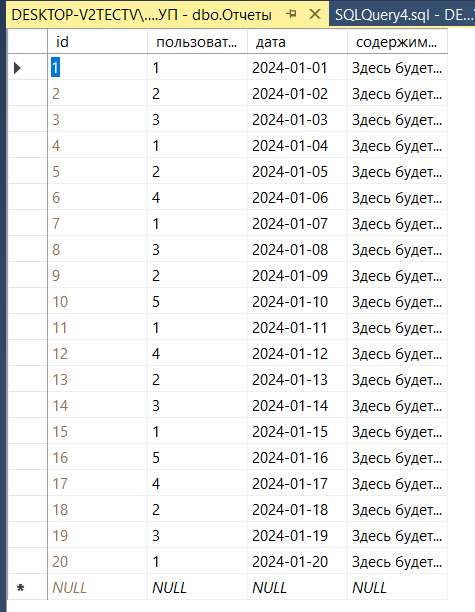


Рис. 21 Таблица «Отчеты»

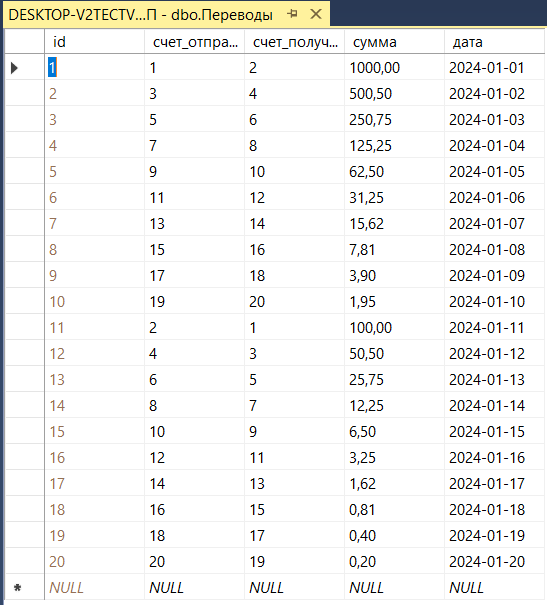


Рис. 22 Таблица «Переводы»

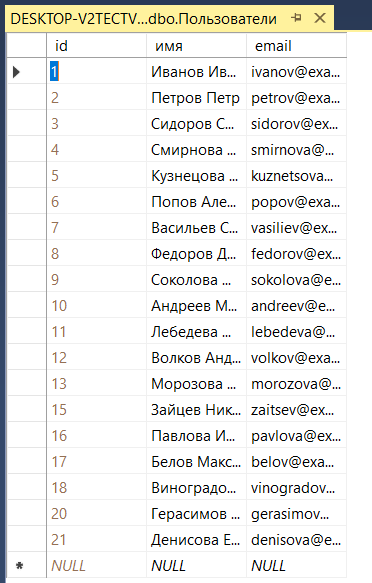


Рис. 23 Таблица «Пользователи»

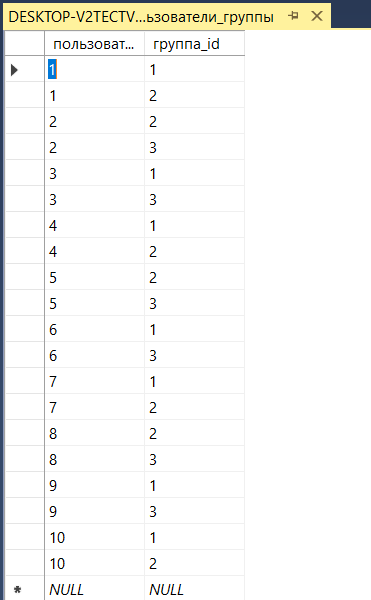


Рис. 24 Таблица «Пользователи группы»

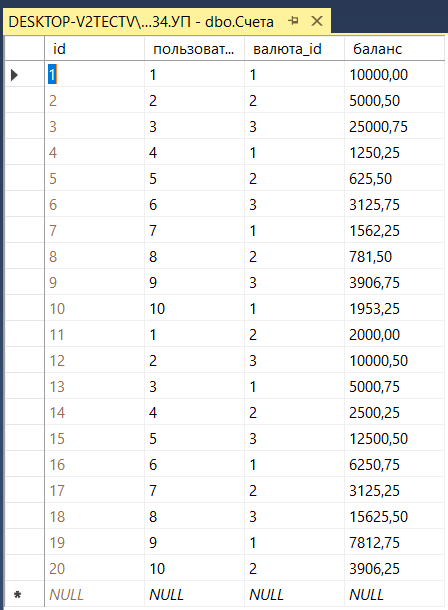


Рис. 25 Таблица «Счета»

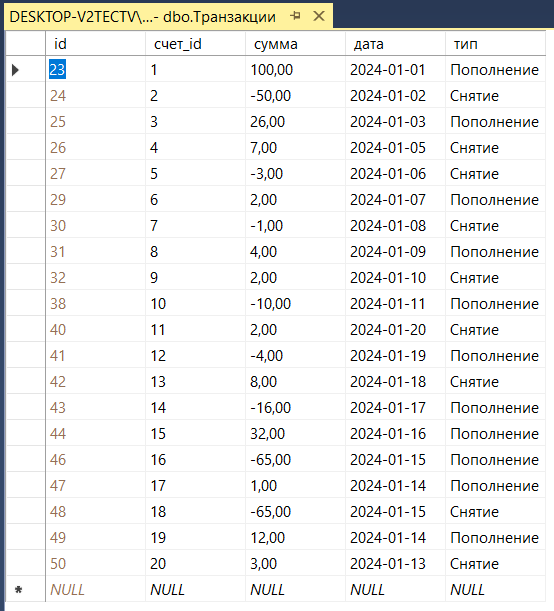


Рис. 26 Таблица «Транзакции»

# **ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ**

## **3.1. Разработка пользовательского интерфейса**

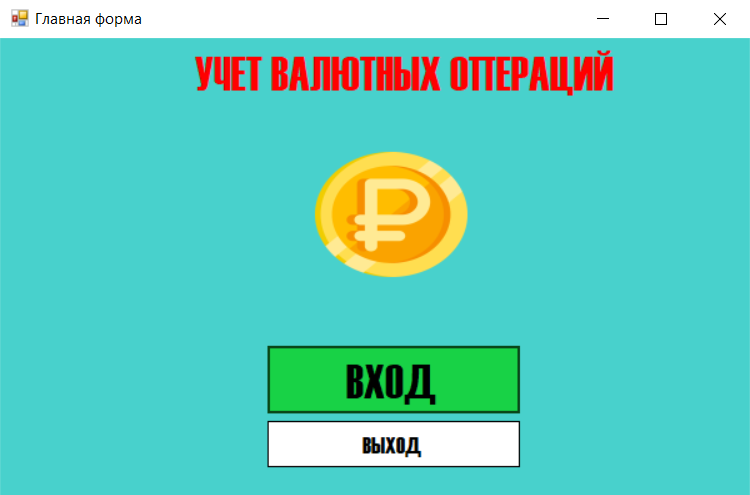


Рис. 27 «Титульная форма»

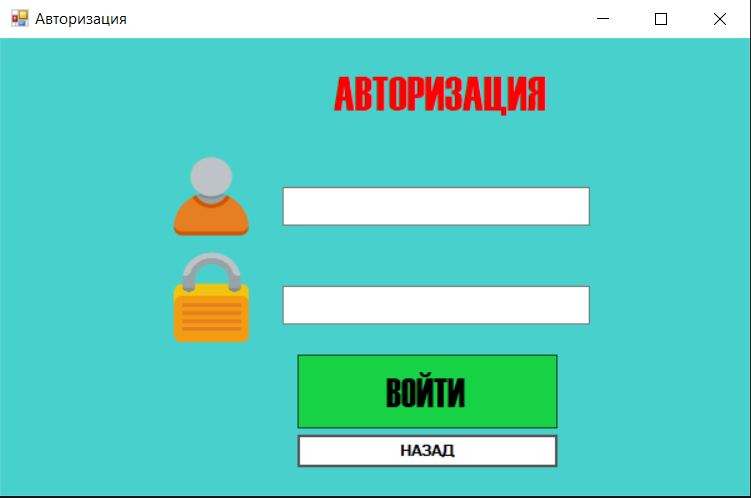


Рис. 28 «Авторизация»

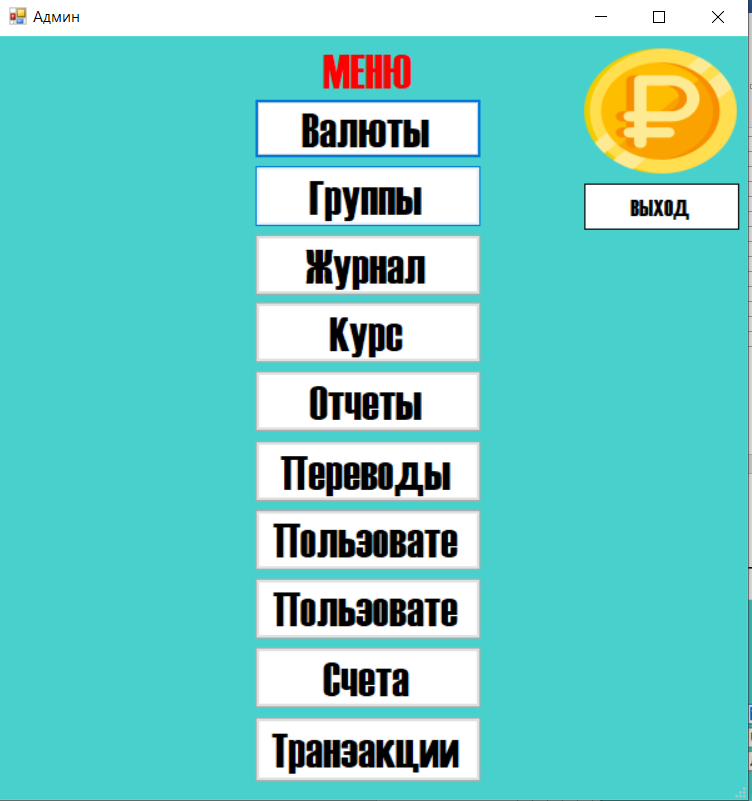


Рис. 29 «Меню»

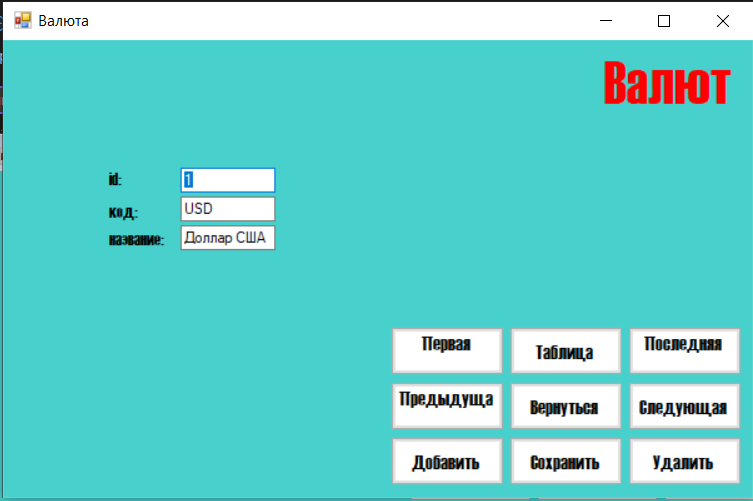


Рис. 30 Форма «Валюта»

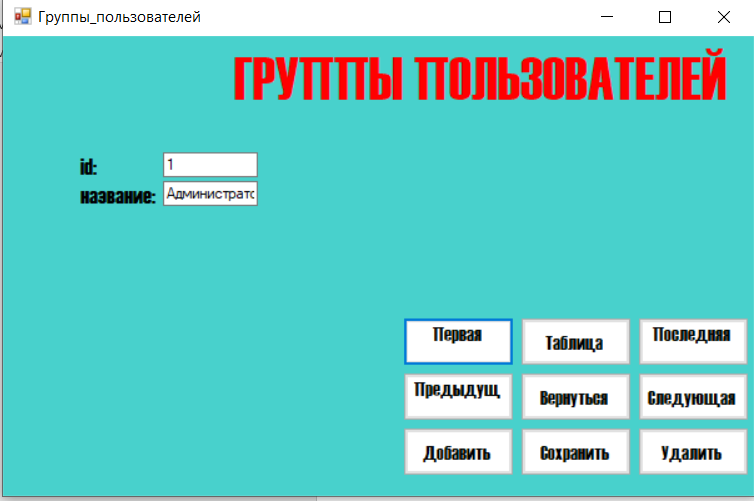


Рис. 31 Форма «Группа Пользователей»

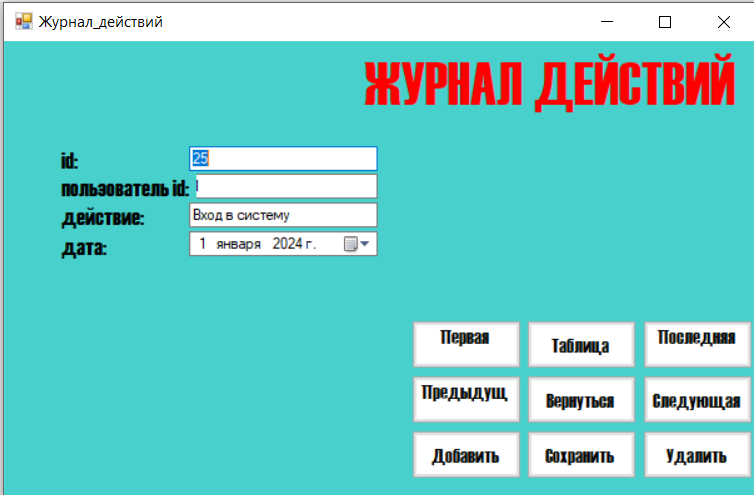


Рис. 32 Форма «Журнал Действий»

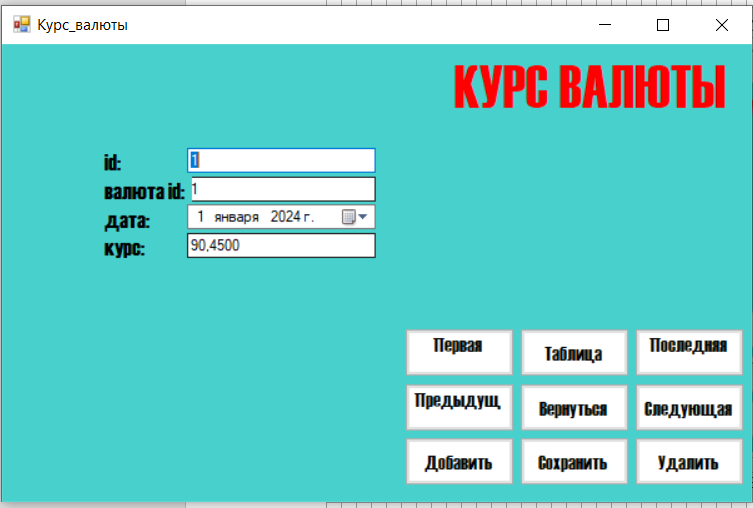


Рис. 33 Форма «Курс Валюты»

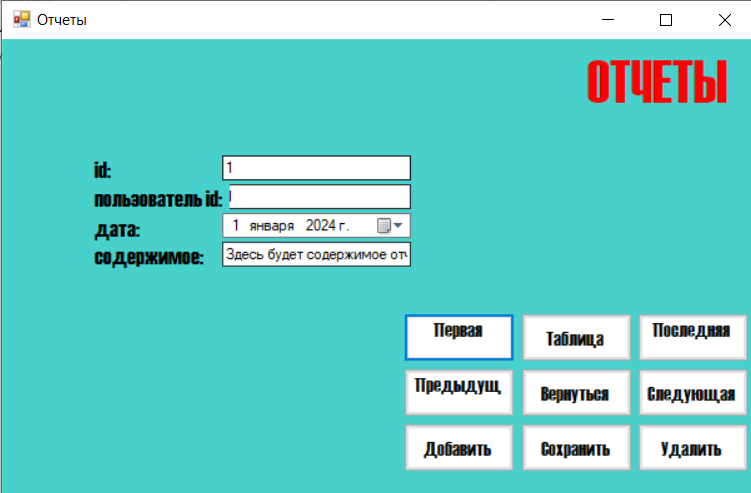


Рис. 34 Форма «Отчеты»

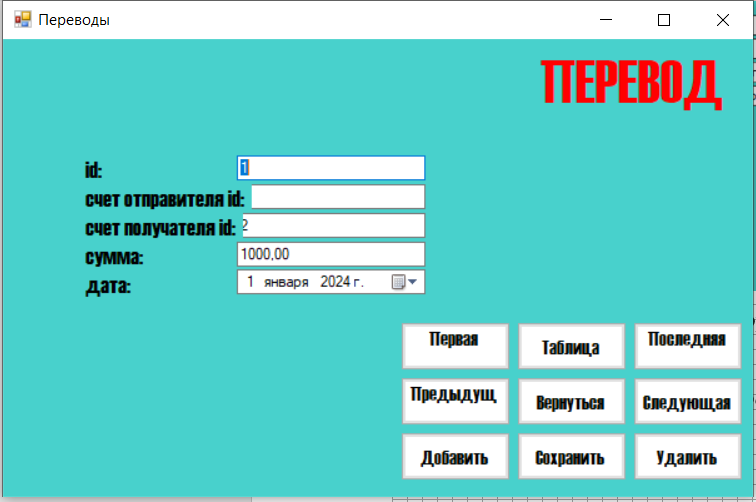


Рис. 35 Форма «Переводы»

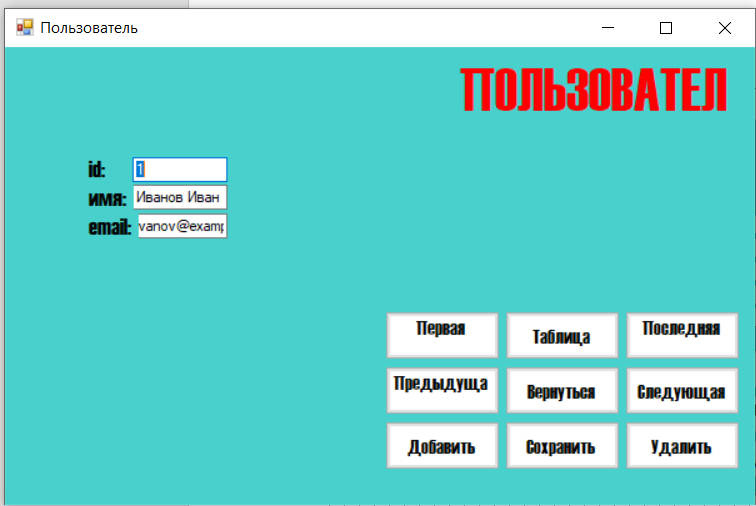


Рис. 36 Форма «Пользователь»

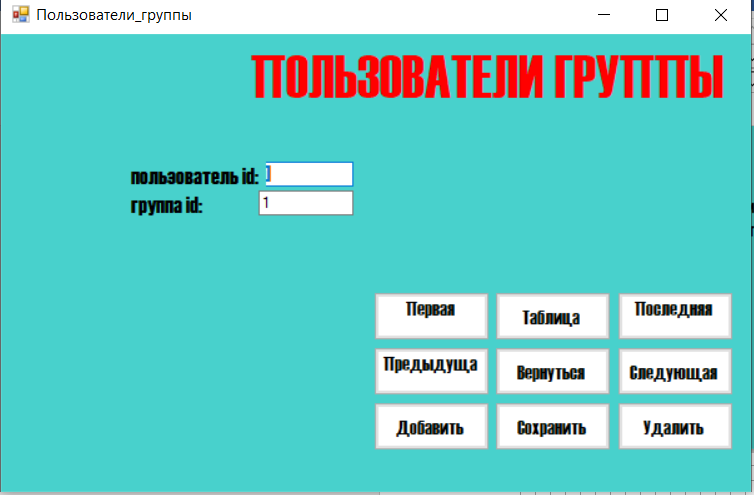


Рис. 37 Форма «Пользователи группы»

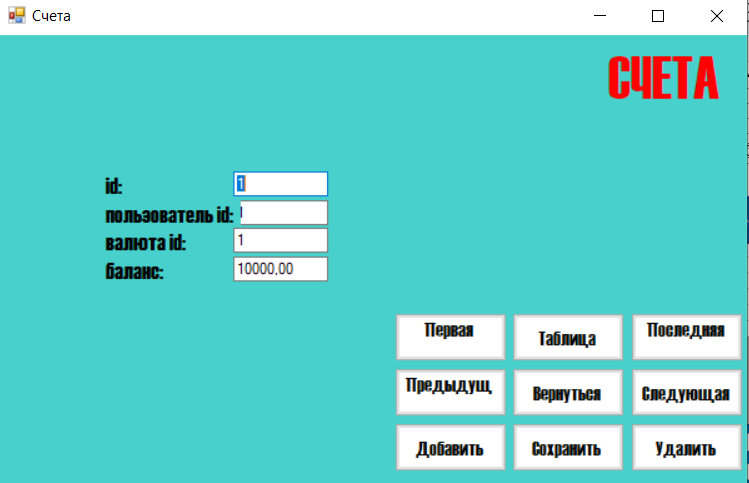


Рис. 38 Форма «Счета»

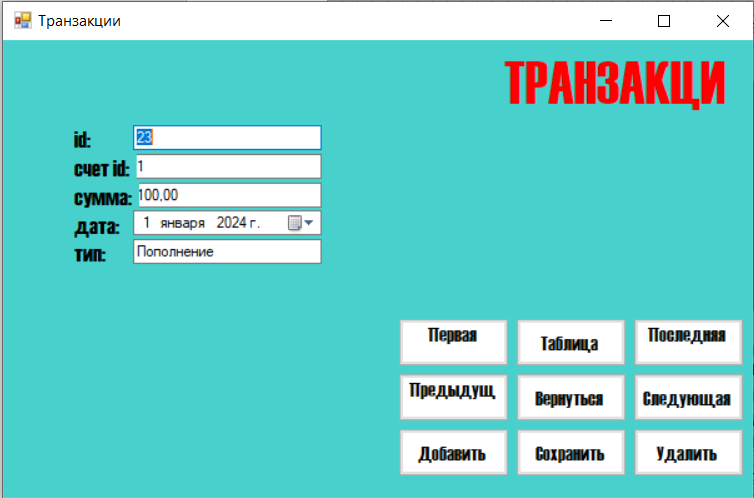


Рис. 39 Форма «Транзакции»

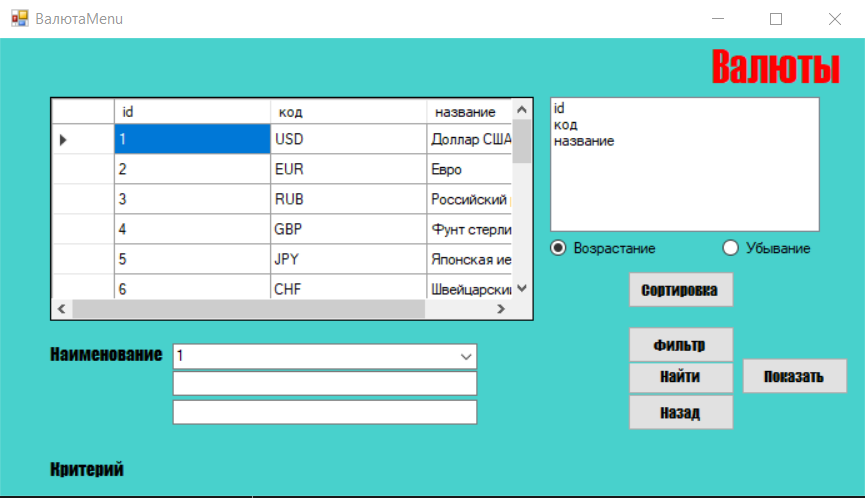


Рис. 40 Форма «Валюты Меню»

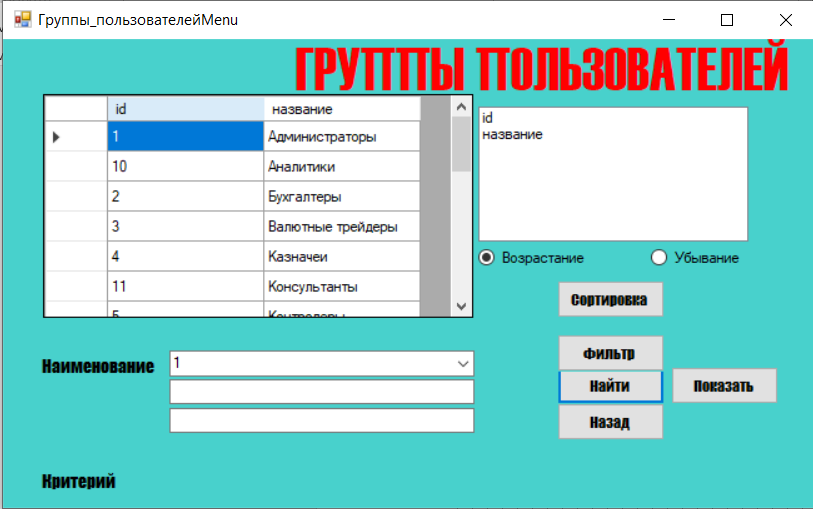


Рис. 41 Форма «Группы пользователей Меню»

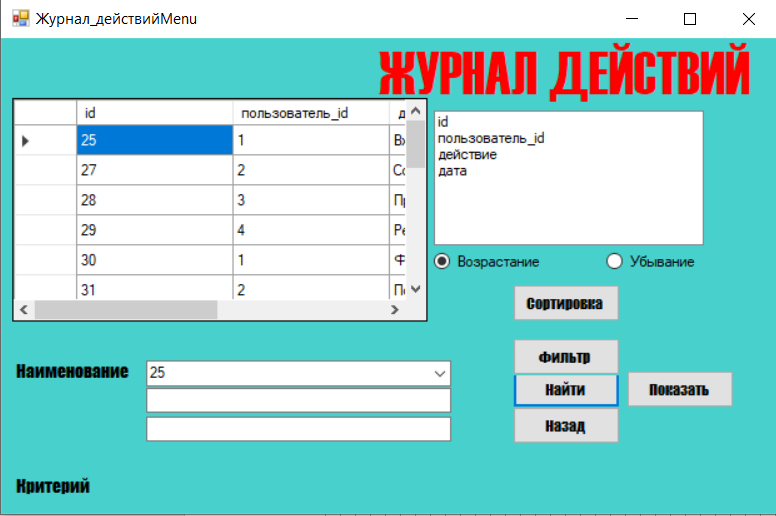


Рис. 42 Форма «Журнал действий Меню»

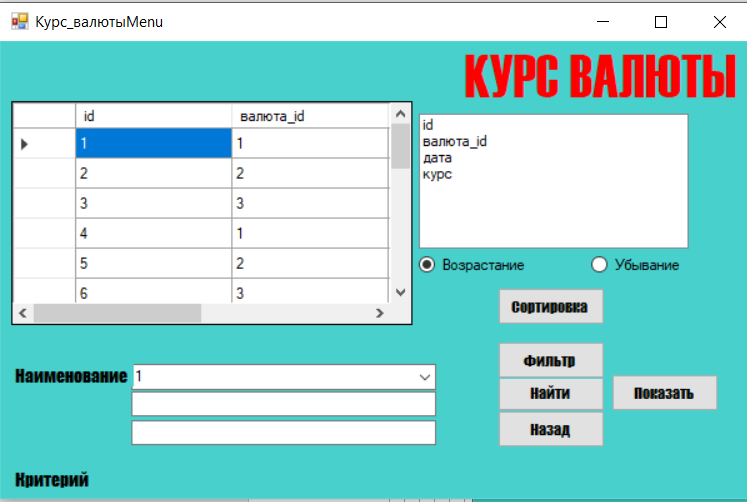


Рис. 43 Форма «Курс валюты Меню»

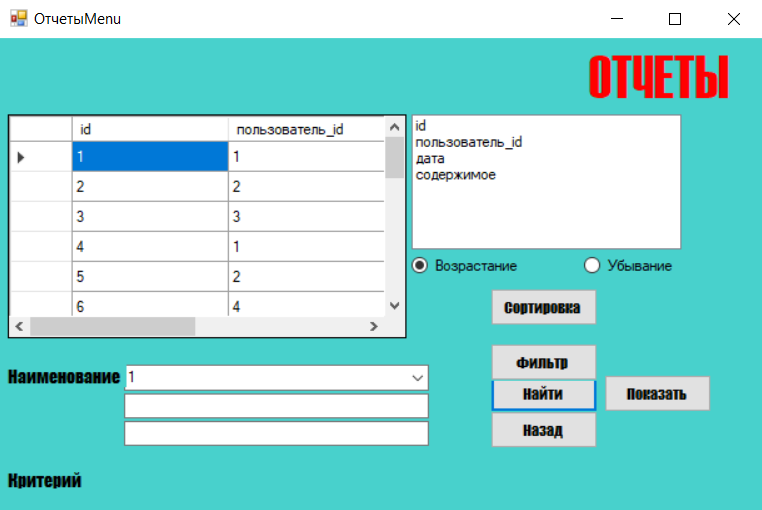


Рис. 44 Форма «Отчеты Меню»

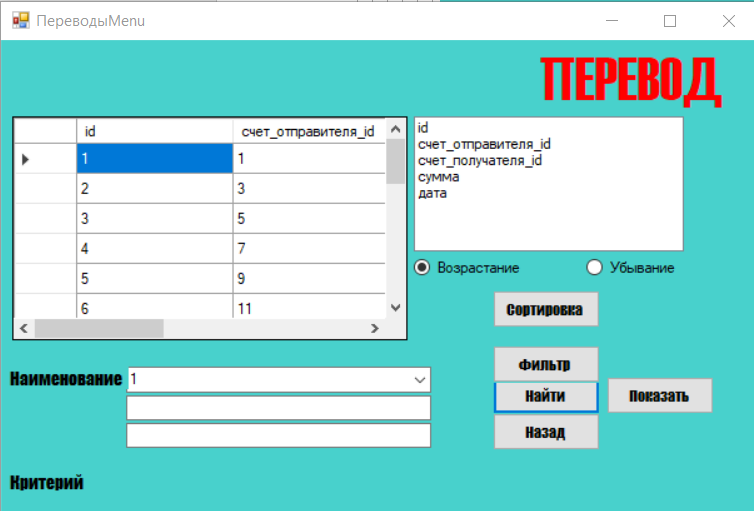


Рис. 45 Форма «Переводы Меню»

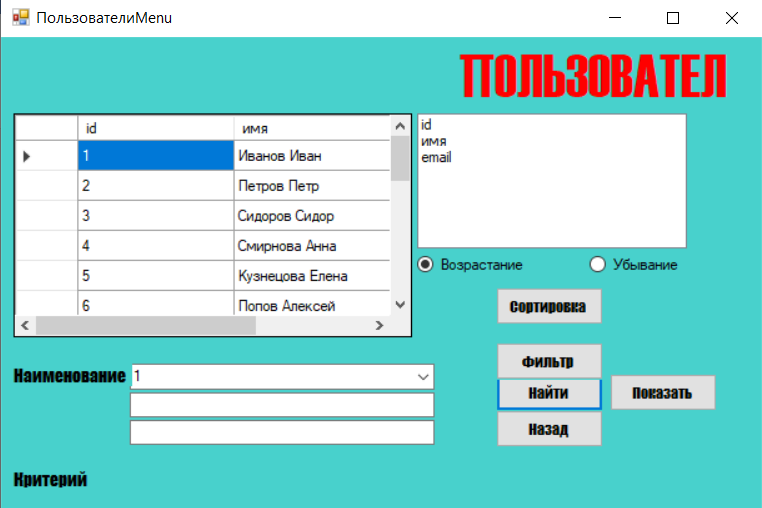


Рис. 46 Форма «Пользователи Меню»

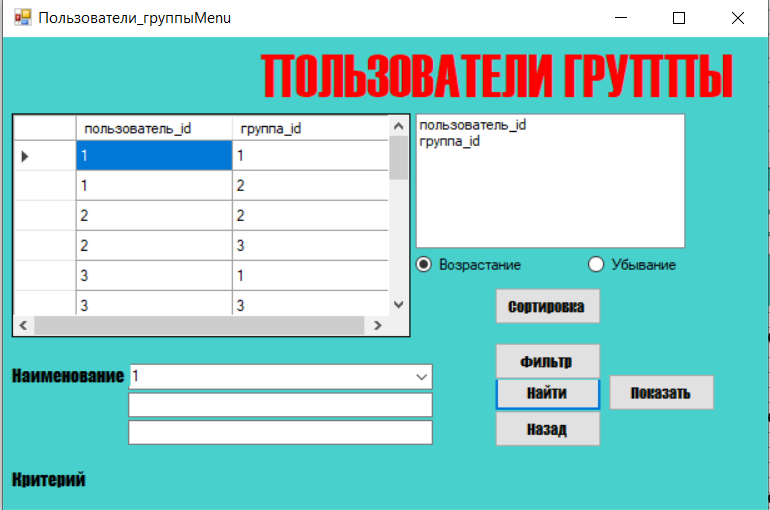


Рис. 47 Форма «Пользователи группы Меню»

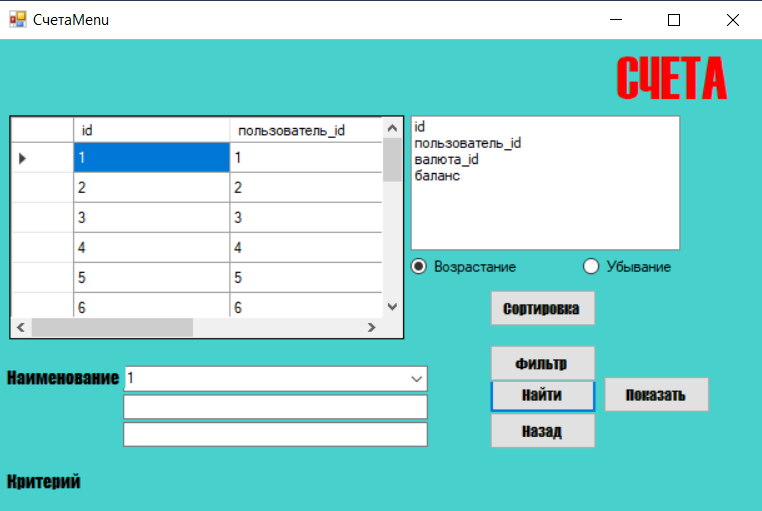


Рис. 48 Форма «Счета»

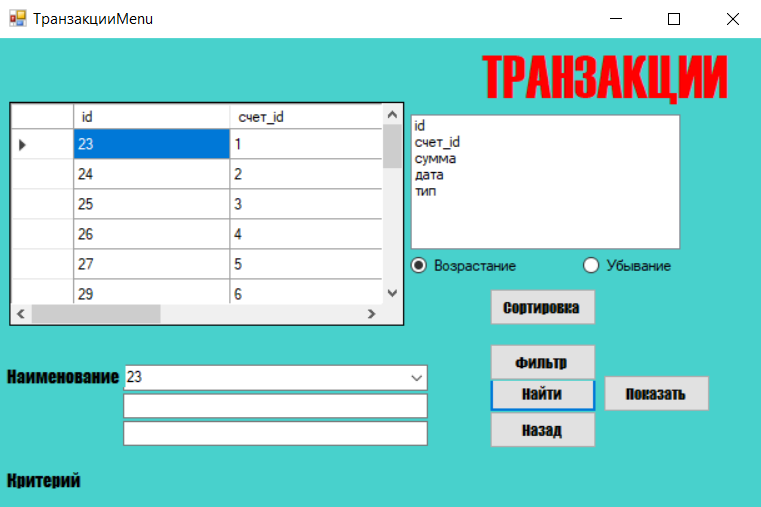


Рис. 49 Форма «Транзакции»

## **3.2. Листинг программы**

Титульная форма

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_2

{

public partial class Главная1 : Form

{

public Главная1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void Главная1\_Load(object sender, EventArgs e)

{

}

private void button1Vxod\_Click(object sender, EventArgs e)

{

Form Главная = new Авторизация();

Главная.Show();

this.Hide();

}

}

}

Форма Меню

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_2

{

public partial class Админ : Form

{

public Админ()

{

InitializeComponent();

}

private void Valut\_Click(object sender, EventArgs e)

{

Form Админ = new Валюта();

Админ.Show();

this.Hide();

}

private void Gryppa\_Click(object sender, EventArgs e)

{

Form Админ = new Группы\_пользователей();

Админ.Show();

this.Hide();

}

private void Djournal\_Click(object sender, EventArgs e)

{

Form Админ = new Журнал\_действий();

Админ.Show();

this.Hide();

}

private void Kyrc\_Сlick(object sender, EventArgs e)

{

Form Админ = new Курс\_валюты();

Админ.Show();

this.Hide();

}

private void Otchet\_Click(object sender, EventArgs e)

{

Form Админ = new Отчеты();

Админ.Show();

this.Hide();

}

private void Perevod\_Click(object sender, EventArgs e)

{

Form Админ = new Переводы();

Админ.Show();

this.Hide();

}

private void Polzovatel\_Click(object sender, EventArgs e)

{

Form Админ = new Пользователи();

Админ.Show();

this.Hide();

}

private void PolzovatelGrypp\_Click(object sender, EventArgs e)

{

Form Админ = new Пользователи\_группы();

Админ.Show();

this.Hide();

}

private void Scheta\_Click(object sender, EventArgs e)

{

Form Админ = new Счета();

Админ.Show();

this.Hide();

}

private void Tranzact\_Click(object sender, EventArgs e)

{

Form Админ = new Транзакции();

Админ.Show();

this.Hide();

}

}

}

Форма Валюта Меню

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace УП\_2

{

public partial class ВалютаMenu : Form

{

private SqlConnection connection;

private string connectionString = "Data Source=DESKTOP-V2TECTV\\TIK1234;Initial Catalog=УП;Integrated Security=True;";

private bool isAscending;

public ВалютаMenu()

{

InitializeComponent();

SetupUI();

LoadData();

}

private void LoadData()

{

}

private void SetupUI()

{

listBox1.Items.Add("id");

listBox1.Items.Add("код");

listBox1.Items.Add("название");

radioButton1.Text = "Возрастание";

radioButton2.Text = "Убывание";

sortButton.Text = "Сортировка";

sortButton.Click += sortButton\_Click;

}

private void ВалютаMenu\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "кУРСОВАЯDataSet.БИЛЕТЫ". При необходимости она может быть перемещена или удалена.

this.валютыTableAdapter.Fill(this.уПDataSet.Валюты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "кУРСОВАЯDataSet.БИЛЕТЫ". При необходимости она может быть перемещена или удалена.

this.валютыTableAdapter .Fill(this.уПDataSet.Валюты);

// Заполнение таблицы через TableAdapter

this.валютыTableAdapter.Fill(this.уПDataSet.Валюты);

// Устанавливаем источник данных для ComboBox

comboBoxBilety.DataSource = валютыBindingSource;

comboBoxBilety.DisplayMember = "id"; // Выберите какой столбец отображать

comboBoxBilety.ValueMember = "название"; // Это значение будет возвращать, если нужно

this.валютыTableAdapter.Fill(this.уПDataSet.Валюты);

валютыBindingSource.DataSource = уПDataSet.Валюты;

валютыDataGridView.DataSource = валютыBindingSource;

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

SortAndDisplayData(false);

}

private void SortAndDisplayData(bool v)

{

string selectedField = listBox1.SelectedItem?.ToString();

if (string.IsNullOrEmpty(selectedField))

{

MessageBox.Show("Пожалуйста, выберите поле для сортировки.");

return;

}

using (connection = new SqlConnection(connectionString))

{

string direction = isAscending ? "ASC" : "DESC";

string query = $"SELECT \* FROM Валюты ORDER BY {selectedField} {direction}";

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

валютыDataGridView.DataSource = dataTable;

}

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

SortAndDisplayData(true);

}

private void sortButton\_Click(object sender, EventArgs e)

{

string sortBy = "код"; // Замените на нужное поле

string sortOrder;

// Установите порядок сортировки в зависимости от выбранной кнопки

if (radioButton1.Checked)

{

sortOrder = "ASC";

}

else if (radioButton2.Checked)

{

sortOrder = "DESC";

}

else

{

MessageBox.Show("Пожалуйста, выберите порядок сортировки.");

return;

}

string query = $"SELECT \* FROM Валюты ORDER BY {sortBy} {sortOrder}";

LoadData(query);

}

private void LoadData(string query)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

try

{

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable table = new DataTable();

adapter.Fill(table);

валютыDataGridView.DataSource = table; // dataGridView1 - это ваш DataGridView

}

catch (Exception ex)

{

MessageBox.Show("Ошибка: " + ex.Message);

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

// Получение критерия из TextBox

string filtr = textBox1.Text;

// Применение фильтрации

if (!string.IsNullOrEmpty(filtr))

{

// Попытка конвертации строки в число

if (int.TryParse(filtr, out int visitorId))

{

валютыBindingSource.Filter = $"id = {visitorId}"; // Точное совпадение

}

else

{

MessageBox.Show("Введите корректный числовой идентификатор посетителя.");

валютыBindingSource.RemoveFilter();

}

}

else

{

валютыBindingSource.RemoveFilter();

}

}

private void Nayti\_Click(object sender, EventArgs e)

{

// Получение критерия из TextBox

string criterion = textBoxCriterion.Text;

// Поиск в DataSet по критерию в нужном столбце

var results = from bilety in уПDataSet.Валюты

where bilety.id.ToString() == criterion // Измените на нужный столбец и критерий

select bilety;

// Вывод результатов

if (results.Any())

{

// Выбор первого найденного результата. Вы можете отобразить все результаты, если нужно.

var biletyInfo = results.First();

labelResult.Text = $"id: {biletyInfo.id}, код: {biletyInfo.код},название: {biletyInfo.название}";

}

else

{

labelResult.Text = "Результаты не найдены.";

}

}

private void button11\_Click(object sender, EventArgs e)

{

Form ВалютаМеню = new Валюта();

ВалютаМеню.Show();

this.Hide();

}

private void Pokaz\_Click(object sender, EventArgs e)

{

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

string query = $"SELECT \* FROM Валюты"; // Замените YourTableName на имя вашей таблицы

using (SqlDataAdapter adapter = new SqlDataAdapter(query, connection))

{

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

валютыDataGridView.DataSource = dataTable; // Замените dataGridView1 на имя вашего DataGridView

}

}

}

catch (SqlException ex)

{

MessageBox.Show($"Ошибка при подключении к базе данных: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch (Exception ex)

{

MessageBox.Show($"Произошла неизвестная ошибка: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void comboBoxBilety\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void textBoxCriterion\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void labelResult\_Click(object sender, EventArgs e)

{

}

}

}

## **3.4. Руководство пользователя**

**1. Введение**

Приложение «АИС Учет валютных операций» (АИС УВО) разработано для автоматизации и упрощения процессов учета, контроля и анализа валютных операций в организации. Оно включает функции для ведения справочников, ввода и учета операций, формирования отчетности и анализа данных. Это приложение предназначено для бухгалтеров, финансистов и руководства, которым необходимо эффективно управлять валютными операциями.

**1.2. Краткое описание возможностей**

Основные возможности приложения включают:

Ведение справочников: Управление справочниками валют, контрагентов, счетов и видов операций.

Учет валютных операций: Ввод, учет и обработка данных о валютных операциях (покупка, продажа, конвертация, переводы и т.д.).

Учет курсов валют: Управление курсами валют (ручной ввод или автоматическая загрузка).

Формирование отчетности: Формирование регламентированной и аналитической отчетности по валютным операциям.

Анализ данных: Анализ данных о валютных операциях для принятия управленческих решений.

**1.3. Уровень подготовки пользователя**

Приложение «АИС Учет валютных операций» разработано с учетом потребностей пользователей с разным уровнем подготовки. Пользователи должны иметь базовые навыки работы с компьютером и понимание принципов бухгалтерского учета. Знание валютного законодательства приветствуется.

**1.4. Перечень эксплуатационной документации**

Для эффективного использования приложения рекомендуется ознакомиться со следующими документами:

Руководство по установке и настройке приложения: Информация о процессе установки приложения на сервер или компьютер пользователя, а также о первоначальной настройке системы.

Руководство оператора: Подробное описание функций приложения для операторов, занимающихся вводом и обработкой данных о валютных операциях.

Руководство пользователя: Описание функций приложения для пользователей, занимающихся формированием отчетности и анализом данных.

Руководство системного администратора: Информация для администраторов системы, отвечающих за обслуживание, настройку и безопасность приложения.

Пожалуйста, ознакомьтесь с вышеуказанными документами для более глубокого понимания функционала приложения и повышения эффективности его использования.

**2. Назначение и условия применения**

**2.1. Виды деятельности и функции**

Приложение «АИС Учет валютных операций» предназначено для автоматизации следующих видов деятельности и функций:

Ведение справочников: Управление справочниками валют, контрагентов, счетов и видов операций.

Учет валютных операций: Ввод, учет и обработка данных о валютных операциях (покупка, продажа, конвертация, переводы и т.д.).

Учет курсов валют: Управление курсами валют (ручной ввод или автоматическая загрузка).

Формирование отчетности: Формирование регламентированной и аналитической отчетности по валютным операциям.

Анализ данных: Анализ данных о валютных операциях для принятия управленческих решений.

Контроль: Контроль соответствия операций требованиям валютного законодательства.

Аудит: Ведение журнала действий пользователей и отслеживание изменений в базе данных.

**2.2. Условия применения**

Для обеспечения эффективной работы приложения «АИС Учет валютных операций» необходимо соблюдение следующих условий:

Оборудование: Приложение совместимо с большинством современных настольных компьютеров и серверов, работающих на операционных системах Windows или Linux.

Технические средства: Рекомендуется наличие устройства с минимальными системными требованиями: процессор не ниже 2 ГГц, 4 ГБ оперативной памяти, достаточный объем дискового пространства для хранения данных, поддержка сетевого соединения (для работы в сетевом режиме).

Операционная среда: Установка приложения требует наличия соответствующих операционных систем (Windows Server, Linux) и СУБД (SQL Server, PostgreSQL).

Общесистемные программные средства: Установка и использование приложения предполагает наличие актуальных версий драйверов устройств и библиотек, необходимых для работы СУБД.

Входная информация и носители данных: Для работы с приложением пользователю необходимо иметь доступ к данным о валютных операциях, контрагентах, счетах и курсах валют. Данные могут быть введены вручную или загружены из внешних источников.

База данных: Приложение использует реляционную базу данных для хранения информации. Требуется наличие установленной и настроенной СУБД (SQL Server, PostgreSQL).

Подготовка специалистов: Пользователи приложения должны иметь базовые навыки работы с компьютером, знание бухгалтерского учета и валютного законодательства.

Соблюдение этих условий позволит максимально эффективно использовать приложение «АИС Учет валютных операций» для автоматизации учета и контроля валютных операций.

**3. Подготовка к работе**

**3.1. Состав и содержание дистрибутивного носителя данных**

Дистрибутивный носитель данных для приложения «АИС Учет валютных операций» включает в себя:

Установочный файл: Файл приложения (например, EXE или MSI для Windows, DEB или RPM для Linux), предназначенный для установки на сервере или компьютере пользователя.

Скрипты для создания базы данных: Скрипты SQL для создания базы данных и таблиц в используемой СУБД (SQL Server, PostgreSQL).

Документация: Руководство пользователя, руководство оператора, руководство системного администратора, описание базы данных.

Примеры данных: Примеры данных о валютных операциях, контрагентах, счетах и курсах валют, которые помогут новым пользователям быстрее освоить приложение.

Обновления: Файлы для обновления приложения (при наличии новых версий). Обновления могут содержать исправления ошибок, новые функции и улучшения производительности.

**3.2. Порядок загрузки данных и программ**

Чтобы загрузить и установить приложение «АИС Учет валютных операций», выполните следующие шаги:

Получите дистрибутивный пакет: Загрузите дистрибутивный пакет с официального веб-сайта разработчика или получите его от вашего системного администратора.

Установите СУБД: Установите и настройте выбранную СУБД (SQL Server, PostgreSQL) в соответствии с инструкциями, прилагаемыми к СУБД.

Создайте базу данных: Используйте скрипты, входящие в дистрибутивный пакет, для создания базы данных и таблиц в установленной СУБД.

Установите приложение: Запустите установочный файл приложения и следуйте инструкциям установщика для завершения установки.

Настройте приложение: Настройте приложение, указав параметры подключения к базе данных, настройки безопасности и другие параметры, необходимые для работы системы.

После установки и настройки рекомендуется перезапустить сервер или компьютер для применения всех изменений.

**3.3. Порядок проверки работоспособности**

Чтобы убедиться, что приложение «АИС Учет валютных операций» успешно установлено и работает корректно, выполните следующие действия:

Запустите приложение: Запустите приложение, используя ярлык на рабочем столе или из меню «Пуск».

Проверьте подключение к базе данных: Убедитесь, что приложение успешно подключается к базе данных и отображает данные из таблиц.

* Протестируйте основные функции:
* Добавьте запись в справочник валют.
* Введите данные о валютной операции.
* Сформируйте отчет по валютным операциям.

Проверьте журнал действий пользователей: Убедитесь, что приложение ведет журнал действий пользователей и записывает все важные события.

После выполнения всех этих шагов вы сможете уверенно использовать приложение «АИС Учет валютных операций» для автоматизации учета и контроля валютных операций.

**4. Описание операций**

**4.1. Создание профиля пользователя**

Наименование: Создание профиля пользователя.

Условия выполнения: Доступ к приложению с правами администратора.

Подготовительные действия:

* Вход в приложение под учетной записью администратора.
* Определение необходимых прав доступа для нового пользователя.
* Основные действия:
* Перейти в раздел «Управление пользователями» в главном меню приложения.
* Нажать кнопку «Создать пользователя».
* Заполнить поля с информацией о пользователе (логин, пароль, ФИО, должность и т.д.).
* Назначить пользователю необходимые права доступа.
* Нажать кнопку «Сохранить».
* Заключительные действия:

Проверить созданный профиль пользователя на корректность информации.

Сообщить пользователю его логин и пароль для входа в систему.

**4.2. Ввод данных о валютной операции**

Наименование: Ввод данных о валютной операции.

Условия выполнения: Доступ к приложению с правами оператора.

Подготовительные действия:

* Вход в приложение под учетной записью оператора.
* Подготовка необходимых документов (например, банковская выписка).
* Основные действия:
* Перейти в раздел «Валютные операции» в главном меню приложения.
* Нажать кнопку «Создать операцию».
* Выбрать тип операции (покупка, продажа, конвертация и т.д.).
* Заполнить поля с информацией о валютной операции (дата, сумма, валюта, курс, контрагент, счет и т.д.).
* При необходимости прикрепить сканы документов.
* Нажать кнопку «Сохранить».
* Заключительные действия:
* Проверить введенные данные на корректность.
* Провести операцию (если необходимо).

**4.3. Формирование отчета по валютным операциям**

Наименование: Формирование отчета по валютным операциям.

Условия выполнения: Доступ к приложению с соответствующими правами доступа.

Подготовительные действия:

* Вход в приложение под учетной записью с правами на формирование отчетности.
* Определение параметров отчета (период, валюта, контрагент и т.д.).
* Основные действия:
* Перейти в раздел «Отчеты» в главном меню приложения.
* Выбрать тип отчета (например, «Оборотно-сальдовая ведомость по валютным счетам»).
* Задать параметры отчета (период, валюта, контрагент и т.д.).
* Нажать кнопку «Сформировать отчет».

Заключительные действия:

* Просмотреть сформированный отчет на экране.
* При необходимости экспортировать отчет в файл (например, Excel, PDF) или распечатать его.

**4.4. Обновление курсов валют**

Наименование: Обновление курсов валют.

Условия выполнения: Доступ к приложению с соответствующими правами доступа.

Подготовительные действия:

* Вход в приложение под учетной записью с правами на обновление курсов валют.
* Проверка доступности источника данных о курсах валют (например, сайт ЦБ).

**Основные действия:**

* Перейти в раздел «Курсы валют» в главном меню приложения.
* Нажать кнопку «Обновить курсы».
* Выбрать источник данных о курсах валют (если есть несколько вариантов).
* Дождаться завершения загрузки курсов валют.
* Заключительные действия:
* Проверить обновленные курсы валют на корректность.
* Сохранить изменения.

Эти операции составляют основу функциональности приложения «АИС Учет валютных операций», предоставляя пользователям удобные инструменты для автоматизации учета и контроля валютных операций.

**5.** **Аварийные ситуации**

Таблица №4

Авариные ситуации

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс ошибки** | **Ошибка** | **Описание ошибки** | **Требуемые действия пользователя** |
| Отказ сервера | Сервер не найден. | Невозможно отобразить страницу. | Обратиться к технической поддержке (HelpDesk) или администратору системы. |
| Ошибка регистрации | Требуется ввести действительное имя пользователя. | При регистрации на портале не введено имя пользователя. | Ввести имя пользователя. |
| Требуется ввести пароль для регистрации. | При регистрации не введен пароль. | Ввести пароль. |
| Сбой аутентификации. | Неверно введено имя пользователя или пароль. | Повторить попытку. Если ошибка сохраняется, обратиться к администратору. |
| Несанкционированное изменение | Обнаружено несанкционированное вмешательство в данные. | Изменения в профиле, выполненные незнакомыми пользователями. | Сменить пароль от учетной записи. Обратиться в техническую поддержку. |

**6. Рекомендации по освоению**

**6.1. Общие рекомендации**

Перед началом работы в приложении «АИС Учет валютных операций» рекомендуется ознакомиться с интерфейсом, изучив основные разделы, такие как:

Главное меню: Навигация по основным функциям приложения.

Справочники: Раздел для ведения справочников валют, контрагентов, счетов и видов операций.

Валютные операции: Раздел для ввода и учета данных о валютных операциях.

Курсы валют: Раздел для управления курсами валют.

Отчеты: Раздел для формирования регламентированной и аналитической отчетности.

Администрирование: Раздел для управления пользователями, настройками системы и ведения журнала действий.

Это поможет быстрее ориентироваться в функционале приложения и находить нужную информацию.

**Создание учетной записи**

Рекомендуется создать учетную запись сразу после установки приложения (если это предусмотрено архитектурой системы). Это позволит получить доступ к персонализированным функциям и настройкам.

Для создания учетной записи выполните следующие шаги (если это необходимо):

* Запустите приложение.
* Перейдите в раздел «Администрирование» или «Настройки».
* Нажмите кнопку «Создать пользователя» или «Регистрация».
* Заполните необходимые поля (логин, пароль, ФИО, должность и т.д.).
* Сохраните данные.

**6.2. Контрольный пример**

Для практического освоения приложения можно использовать следующий контрольный пример:

Цель: Ввести данные о валютной операции (покупке валюты) и сформировать отчет об остатках на валютных счетах.

Шаги выполнения:

* Ввод данных о валютной операции:
* Перейдите в раздел «Валютные операции».
* Нажмите кнопку «Создать операцию».
* Выберите тип операции «Покупка валюты».
* Заполните поля с информацией об операции (дата, сумма, валюта, курс, контрагент, счет и т.д.).
* Сохраните данные.
* Формирование отчета об остатках на валютных счетах:
* Перейдите в раздел «Отчеты».
* Выберите тип отчета «Оборотно-сальдовая ведомость по валютным счетам».
* Задайте период для отчета.
* Сформируйте отчет.
* Проверка результатов:
* Убедитесь, что остатки на валютных счетах в отчете соответствуют данным о валютных операциях.

**6.3. Правила запуска и выполнения**

Запуск приложения:

* Убедитесь, что ваш компьютер подключен к сети (если приложение работает в сетевом режиме).
* Запустите приложение, используя ярлык на рабочем столе или из меню «Пуск».
* Введите свои учетные данные для входа в систему.
* Выполнение операций:
* Следуйте интуитивно понятному интерфейсу.
* Используйте доступные подсказки и инструкции в приложении в случае возникновения затруднений.
* Перед выполнением важных операций рекомендуется создавать резервные копии базы данных.
* При возникновении вопросов или сложностей, обращайтесь к руководству пользователя или к технической поддержке.

# **ГЛАВА 4. РАЗРАБОТКА 1С**

## **4.1 Разработка подсистем**

****

Рис. 50 Окно входа в систему

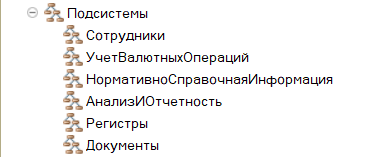
****

Рис. 51 Список подсистем

## **4.2 Разработка справочников**

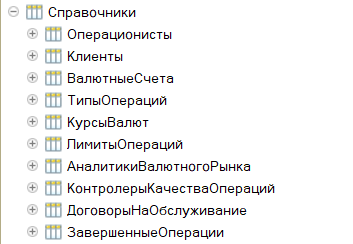


Рис. 52 Создание справочников

## **4.3 Разработка документов и регистров**

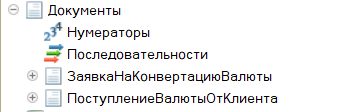


Рис. 53 Создание документации

## **4.4 Разработка отчетов**

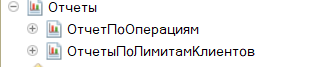


Рис. 54 Создание отчета

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Бессмертный, И. А. Системы искусственного интеллекта : учеб. пособие для СПО / И. А. Бессмертный. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2018. — 130 с.
2. Гниденко, И. Г. Технология разработки программного обеспечения : учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с.
3. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 501 с.
4. Жмудь, В. А. Моделирование замкнутых систем автоматического управления : учеб. пособие для академического бакалавриата / В. А. Жмудь. — 2-е изд., испр. и доп. — М. : Издательство Юрайт, 2019. — 128 с.
5. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2019. — 155 с.
6. Иванов, В. М. Интеллектуальные системы : учеб. пособие для СПО / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2019. — 93 с.
7. Иванов, В. М. Интеллектуальные системы : учеб. пособие для вузов / В. М. Иванов ; под науч. ред. А. Н. Сесекина. — М. : Издательство Юрайт, 2017. — 91 с.
8. Кубенский, А. А. Функциональное программирование : учебник и практикум для академического бакалавриата / А. А. Кубенский. — М. : Издательство Юрайт, 2019. — 348 с.
9. Кудрина, Е. В. Основы алгоритмизации и программирования на языке c# : учеб. пособие для СПО / Е. В. Кудрина, М. В. Огнева. — М. : Издательство Юрайт, 2019. — 322 с.