

Практическая работа

Построение архитектуры информационной системы

1. Определение архитектуры разрабатываемого программного средства

Тип архитектуры: Клиент-серверная двухзвенная архитектура (2-tier)

Компоненты системы:

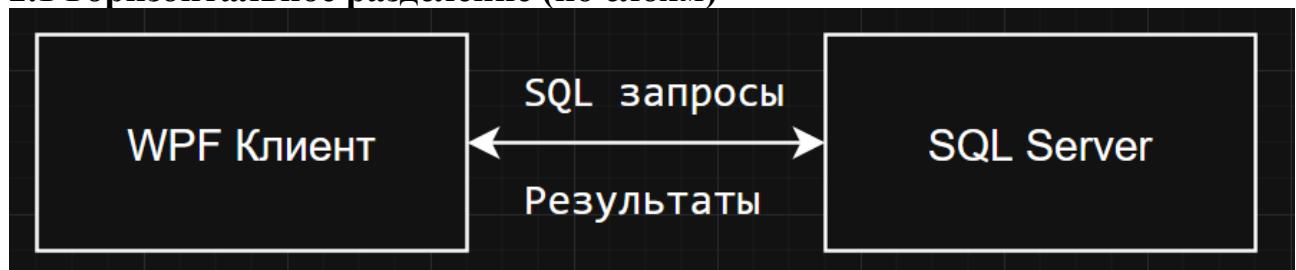
1. **Клиентская часть (Client Tier):** WPF-приложение на C#
2. **Серверная часть (Server Tier):** Microsoft SQL Server

Архитектурный стиль: Монолитное приложение с модульной структурой

Стандарты: ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств)

2. Взаимосвязи и взаимодействия частей системы

2.1 Горизонтальное разделение (по слоям)



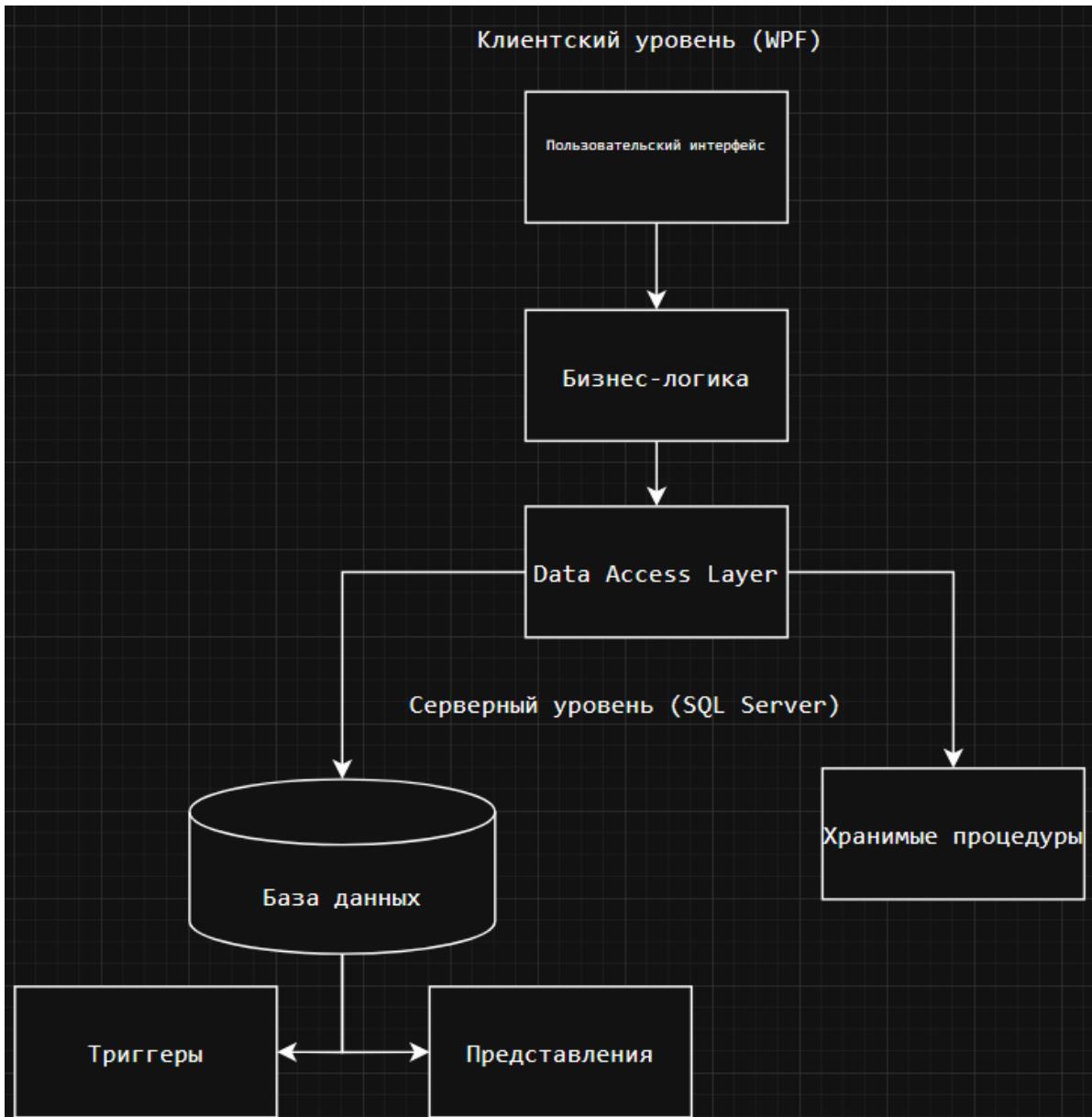
2.2 Вертикальное разделение (по модулям)

CompanyPayrollApp (WPF)

- Модуль авторизации (LoginWindow)
- Модуль управления сотрудниками (EmployeesPage)
- Модуль управления отделами (DepartmentsPage)
- Модуль управления должностями (PositionsPage)
- Модуль расчета зарплаты (CalculatePayrollWindow)
- Модуль выплат (PaymentsWindow)
- Модуль отчетности (ReportEmployeesPage, ReportDepartmentsPage)
- Модуль аудита (AuditLogWindow)

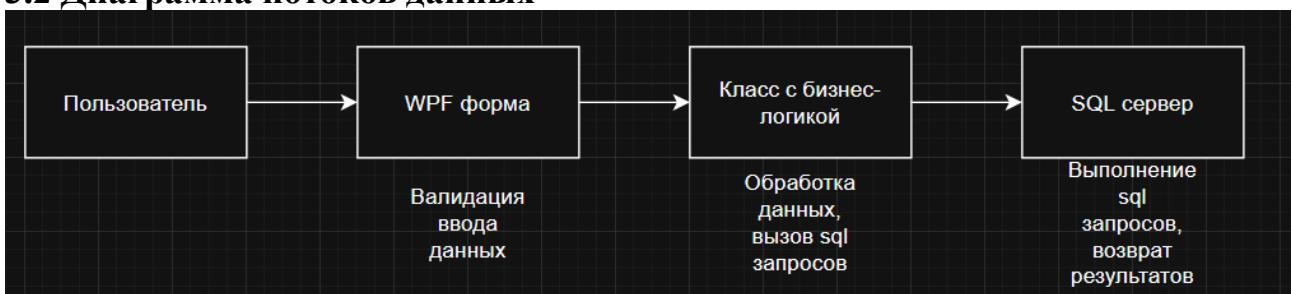
3. Графическое представление архитектуры

3.1 Диаграмма компонентов системы



Пояснение: WPF-приложение через слой доступа к данным (DAL) взаимодействует с SQL Server, используя хранимые процедуры, триггеры и представления.

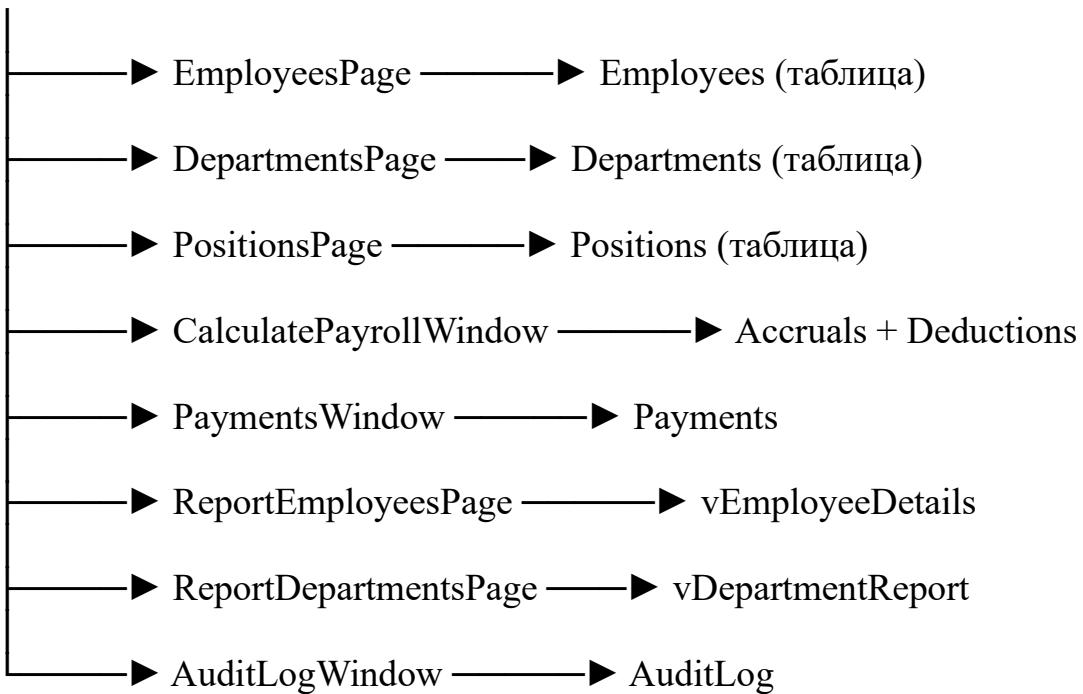
3.2 Диаграмма потоков данных



Пояснение: Данные проходят цепочку обработки: ввод пользователя → валидация в UI → обработка бизнес-логикой → выполнение SQL → возврат результатов.

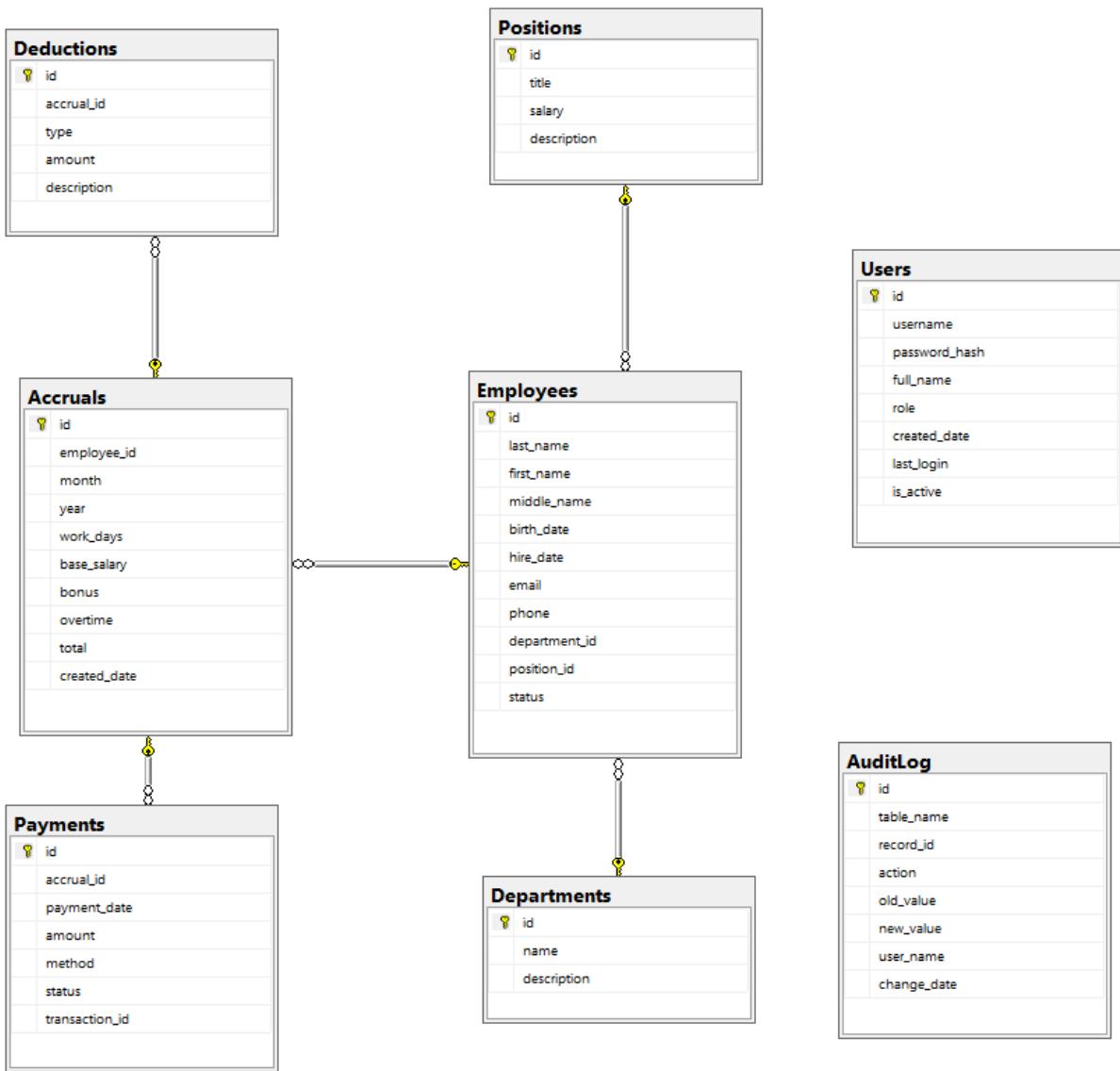
3.3 Диаграмма взаимодействия модулей

MainWindow



Пояснение: Главное окно (MainWindow) обеспечивает навигацию между модулями, каждый из которых работает с соответствующей таблицей или представлением в БД.

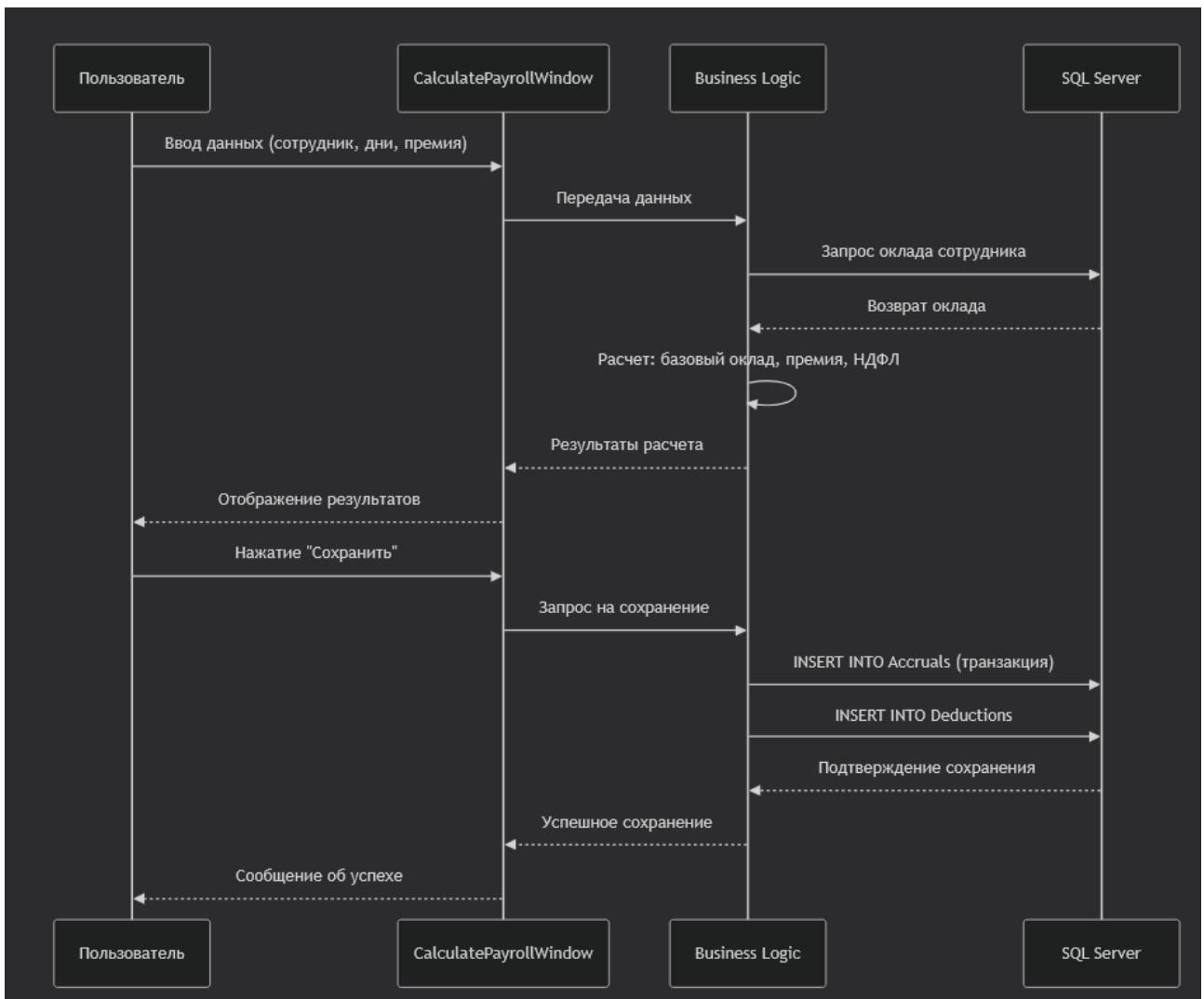
3.4 Структура базы данных (схема)



Пояснение: Схема БД показывает связи между таблицами:

- Employees связаны с Departments и Positions
- Accruals связаны с Employees
- Deductions и Payments связаны с Accruals
- AuditLog независима, фиксирует изменения
- Users независима, хранит данные входа

3.5 Диаграмма последовательности (расчет зарплаты)



Пояснение: Последовательность операций при расчете и сохранении зарплаты с транзакционной целостностью.

4. Технические характеристики архитектуры

Клиентская часть:

- **Технология:** WPF (Windows Presentation Foundation)
- **Язык:** C#
- **Паттерны:** MVVM (частично), Code-behind
- **Библиотеки:** .NET Framework, [ADO.NET](#)

Серверная часть:

- **СУБД:** Microsoft SQL Server
- **Механизмы:** Хранимые процедуры, триггеры, представления
- **Соединение:** SQL Authentication, строки подключения
- **Транзакции:** Поддержка ACID

Взаимодействие:

- **Протокол:** TDS (Tabular Data Stream)
- **Соединение:** Пул соединений
- **Безопасность:** Авторизация, хэширование паролей

5. Соответствие ГОСТ Р ИСО/МЭК 12207

Реализованные процессы жизненного цикла:

- Процесс согласования (Agreement Process):** Определены требования на основе анализа предметной области
- Процесс разработки (Development Process):** Реализованы все модули системы
- Процесс эксплуатации (Operation Process):** Система готова к использованию
- Процесс сопровождения (Maintenance Process):** Архитектура позволяет легко вносить изменения

Документация в соответствии со стандартом:

- Описание архитектуры (настоящий документ)
- Техническое задание (описание требований)
- Руководство пользователя (инструкции по работе с модулями)