

# ПОДСИСТЕМА УЧЕТА УСПЕВАЕМОСТИ ОБУЧАЮЩИХСЯ «UNIVERSITY JOURNAL»

Плетнёв М.Э., Мальцев Д.В.

БФ УУНиТ, г. Бирск.

**Аннотация.** Статья посвящена разработке подсистемы «University Journal» для автоматизации учета успеваемости обучающихся. Описано проектирование системы на основе Clean Architecture с использованием ASP.NET Core и Flutter. Представлена структура базы данных в PostgreSQL и реализованная ролевая модель доступа (RBAC). Результатом работы является масштабируемое решение, оптимизирующее сбор и хранение данных об учебном процессе.

**Ключевые слова:** University Journal, учет успеваемости, Clean Architecture, ASP.NET Core, Flutter, PostgreSQL, RBAC, автоматизация образования.

## **Актуальность.**

Цифровизация образования требует отказа от бумажных журналов в пользу электронных систем, автоматизирующих учет успеваемости. Существующие решения часто не учитывают специфику конкретного вуза, что делает разработку собственной подсистемы «University Journal» актуальной задачей [1]. Цель работы — проектирование архитектуры подсистемы с разграничением доступа на основе анализа предметной области.

## **Функции подсистемы.**

Реализована многопользовательская модель (RBAC) с тремя ролями. *Администратор* управляет пользователями, группами и предметами, назначает преподавателей на дисциплины, экспортирует отчеты. *Преподаватель* ведет электронный журнал, выставляет оценки, фиксирует посещаемость. *Студент* просматривает свои оценки, расписание и учебные планы в личном кабинете.

## **Технологии.**

Проект построен на клиент-серверной архитектуре. Backend: C# (ASP.NET Core Web API, .NET 8) с использованием Clean Architecture, БД PostgreSQL, ORM Entity Framework Core, паттерн Repository. Frontend: кроссплатформенный Flutter (Dart) для iOS, Android и Web. Сетевое взаимодействие — библиотека Dio, JSON over HTTP. Дизайн — Material Design 3. Валидация данных двухуровневая: на клиенте (Flutter) и сервере (FluentValidation).

## **Схема базы данных.**

Логическая структура БД (рис. 1) обеспечивает целостность данных. Центральная таблица *Users* содержит учетные записи (Login, PasswordHash, Role). Таблицы *Students* и *Teachers* хранят персональные данные и связаны с *Users* и *Groups*. Связь «многие ко многим» между студентами и предметами реализована через промежуточную таблицу *StudentSubjects*. Оценки фиксируются в таблице *Grades*, посещаемость — в *Attendances*. Схема соответствует требованиям нормализации и поддерживает все описанные функции.

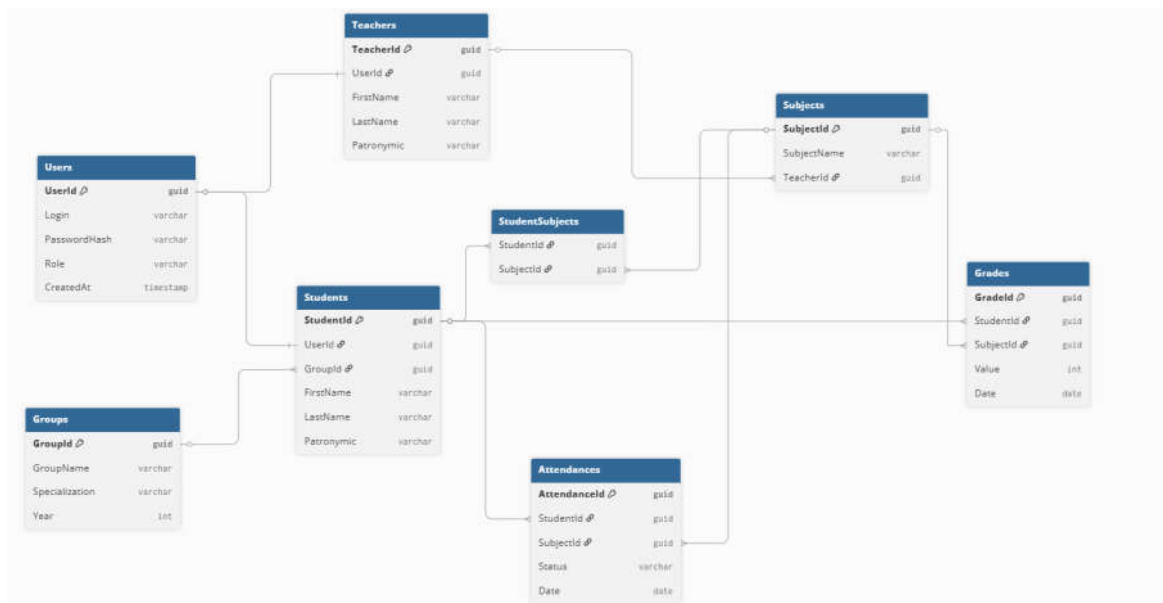


Рисунок 1 – Логическая схема базы данных

## Заключение.

Спроектирована подсистема учета успеваемости, определены функции пользователей. При разработке применены технологии C#/.NET/PostgreSQL + Flutter. Использование Clean Architecture обеспечивает гибкость и масштабируемость решения для внедрения в учебный процесс.

## Литература

1. Гвоздева, Т. В. Проектирование информационных систем. Методы и средства структурно-функционального проектирования. Практикум : учебное пособие для СПО / Т. В. Гвоздева, Б. А. Баллод. — 3-е изд., стер. — Санкт-Петербург : Лань, 2024. — 148 с.
2. Вафин, Р. Р. Программирование на C#.NET : учебное пособие / Р. Р. Вафин, А. Р. Бикмурзина. — Казань : КНИТУ-КАИ, 2020. — 108 с.
3. Садыков, А. М. Методы разработки веб-приложений : учебно-методическое пособие / А. М. Садыков. — Иваново : ИГЭУ, 2019. — 72 с.

4. Рудалев, В. Г. Введение В ASP.NET CORE MVC : учебно-методическое пособие / В. Г. Рудалев. — Воронеж : ВГУ, 2021 — Часть I — 2021. — 59 с.