**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

# **Дисциплина:** Бэк-энд разработка

Отчет

Практическая работа

Выполнил:

Крохин Владимир

ИСУ 490271

Проверил:

Добряков Д. И.

Санкт-Петербург

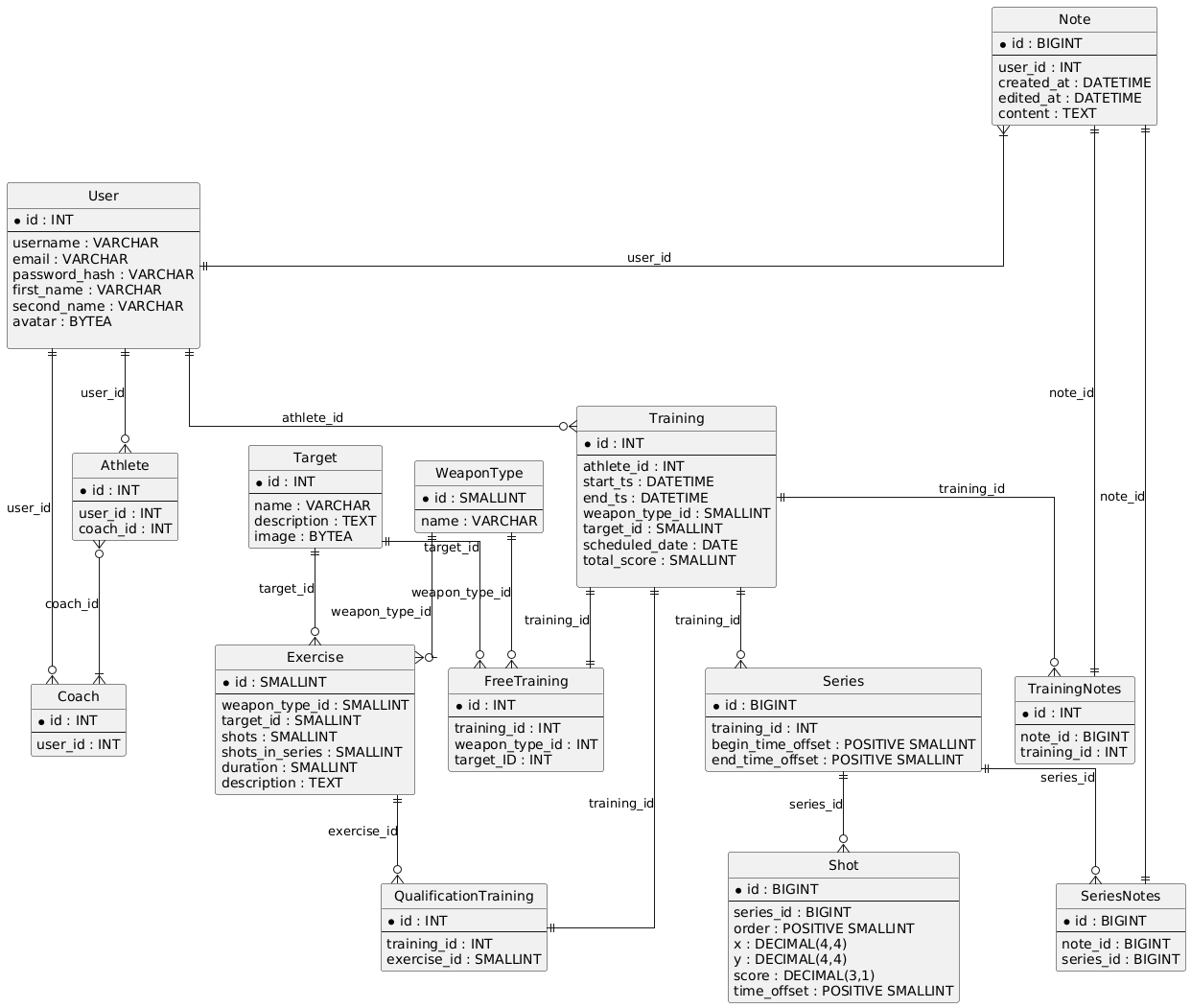
2025 г.

**Задача**

* Реализовать все модели данных, спроектированные в рамках ДЗ1
* Реализовать набор из CRUD-методов для работы с моделями данных средствами Express + TypeScript
* Реализовать API-эндпоинт для получения пользователя по id/email

**Ход работы**

Перед приступлением к работе, были переработаны сущности моделей данных, спроектированных в ДЗ1.



Был проинициализирован пакет npm, установлены необходимые зависимости (TypeORM, typescript, Express)

Далее, используя знания, полученные с лекции, были разработаны с использованием TypeORM сущности, прописаны их связи. Сущности расположены в src/entities/.

После этого были разработаны контроллеры согласно заданию (CRUD-методы). Контроллеры расположены в src/controllers/.

Далее к контроллерам были разработаны роутеры в src/routes/.

Сущности, контроллеры и роутеры были построены с опорой на материал лекции и документацию.

Конфигурация ORM происходит в src/ormconfig.ts

Конфигурация и запуск приложения расположен в src/index.ts

Приложение необходимо скомпилировать в JS-файлы при помощи:

node run build

Запустить приложение можно при помощи:

node run start

После этого с использованием curl были протестированы API. Составлена документация API в файле README.md

**Вывод**

Были получены базовые навыки по работе с TypeScript, TypeORM, Express. Была выстроена примитивнейшая тривиальная структура серверного веб-приложения.

Выявлены очевидные недостатки подхода с использованием только Entities, Controllers, Routers:

* модели данных - сущности TypeORM - не поддерживают ограничения, связанные с возможными состояниями.
* есть затруднения в выстраивании моделей-агрегатов в приложении, приближенных к понятным моделям из реальной жизни (в приложении 3 сущности, связанных с тренировками, 3 сущности, связанных с заметками)
* контроллеры являются “толстыми”, что усложняет их переиспользуемость и тестируемость
* тестируемость усложнена, т.к. много IO-bound задач.

Лично я предпочитаю использовать DDD-подход с выделением “ядра” приложения и надстройкой над ним инфраструктуры и интерфейсов.