# Звіт до лабораторної 3

Студента групи ТТП-32

Остренка Олександра

## 1. Умова лабораторної: Изображение выглядит как текст, Шрифт, снимок экрана, документ

## 2. Опис виконання:

Лабораторна була виконана на мові С# за допомогою фреймворку xUnit з використанням бібліотеки Moq.

* Суть програми, що тестується.  
   Програма являє собою web api проект (побудований за допомогою ASP.NET Core). В проекті є лише одна модель – User та контролер UserController, що оновлює БД відповідної сутності за допомогою класу Repository, що працює безпосередньо з контекстом APIContext. Контролер реалізує весь стандартний апарат – GetAll, GetById, Put, Post, Delete.
* З самого початку треба об’явити глобальні змінні – мок репозиторія та відповідний контроллер, методи якого будуть тестуватися, та створити екземпляр репозиторію.  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание
* Далі, за допомогою класу Faker з бібліотеки Bogus, створимо метод, що буде генерувати списки об’єктів User.  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание
* Метод, що тестує функцію GetAll:  
  Изображение выглядит как текст, снимок экрана, программное обеспечение

  Автоматически созданное описание
  + Спочатку створює фейковий список користувачів.
  + Далі створюємо мок об’єкт, що імітує поведінку справжнього IRepository. Налаштовуємо цей мок-репозиторій, щоб він, при викликані метода GetAllAsync() повертав фейковий список користувачів.
  + Далі створюється екземпляр контролера на основі ініціалізованого мок-репозиторія.
  + Далі викликається метод контролера GetUsers, що імітує витягування користувачів за допомогою того самого мок-репозиторія.
  + Після чого ідуть три Assert перевірки: перевірка типу результату, перевірка коду операції (200 – операція була успішною), перевірка ідентичності самих списків користувачів.
  + Заключна перевірка того, що метод GetAllAsync() був викликаний один раз.
* Метод, що тестує функцію GetAll на виключення.  
  Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

  Автоматически созданное описание
  + Спочатку створюємо мок-об’єкт, що має повернути виключення ви виклику метода GetAllAsync.
  + Створюємо та налаштовуємо відповідно до мок-об’єкту екземпляр контролера.
  + Тестуємо відповідний метод контролера.
  + Потім ідуть два Assert вирази: один тестує тип об’єкту, що повертає метод, інший – порівнює код операції (400 – операція виконалась невдало), що і треба було протестувати.
* За цим же принципом реалізовані тести до кожного з методів (один тестує «правильний» випадок, інший – виключення).

## 3. Тестування

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание