Khmelnytskyi National University

Department of Computer Engineering and Information Systems

**Report**

Laboratory work №7

Discipline: “Object-oriented programming”

Topic: “MULTITHREADING AND ASYNCHRONOUS PROGRAMMING”

Completed: 1st year student, group CEs-24-1 Maksim Lapko

Name, Surname

Checked: Viacheslav Boiko

Name, Surname

Khmelnytskyi, 2024

Purpose: To learn the concepts and techniques of multithreading and asynchronous programming and understand how to manage concurrent tasks within applications effectively; figure out how to implement and control multiple threads, manage asynchronous operations, and explore ways to optimize application performance and responsiveness, especially in environments where tasks can run independently or in the background without blocking main processes.

**Task 1**

Explore your application to find places where the multithreading could be suitable. If there are no places in the application, come up with a new or extended functionality that may require the usage of multithreading. Implement the multithreading, use a lock mechanism where needed, and a ThreadPool for effective thread management. It could be a thread-safe singleton, simple web-server, CPUintensive task that requires multithreading to improve performance.

Multithreading using to update list of users and userStatistic.

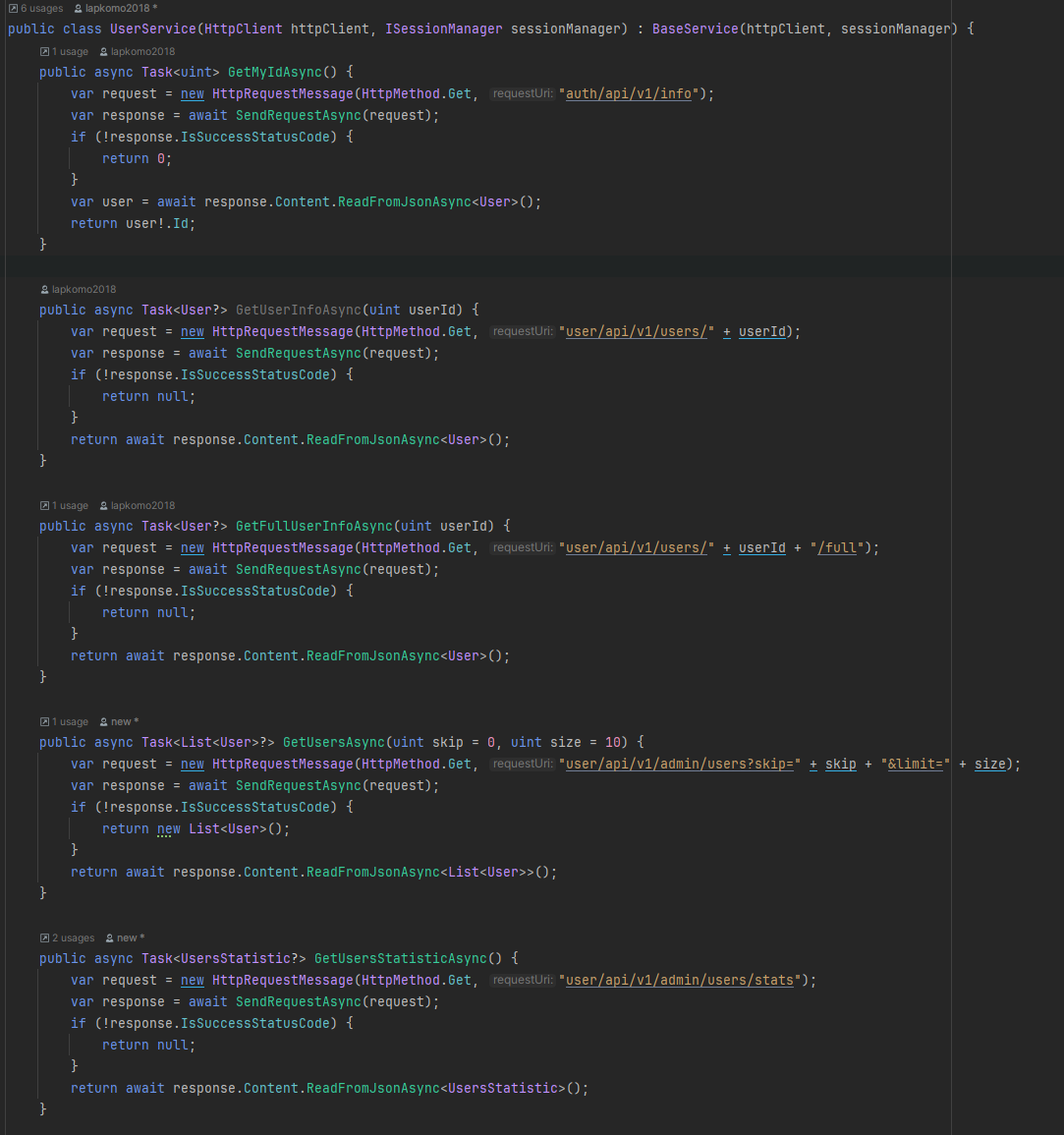


**Task 2**

Explore your application to find places where asynchronous programming

could be suitable. If there are no places in the application, come up with a new or extended functionality that may require the usage of asynchronous programming (except async methods that EF core provides). Implement the asynchronous execution. It could be sending some requests to the external APIs to retrieve some information or the UI management to avoid the main UI thread blocking while loading some data to not freeze a user interface but show the loading spinner or other element. Show the cancellation token usage, and use ConfigureAwait(false) when needed.

In almost all service functions, asynchronous programming is used. UserService`s code screenshot for example:



**Conclusions**

In completing this laboratory work, I gained practical experience and learned the concepts and techniques of multithreading and asynchronous programming and understand how to manage concurrent tasks within applications effectively.