**Task Forge**

**Development of a Distributed Client-Server Simulation Execution Framework**

**Candidat: Bogdan-Sergiu BOLOȘ**

**Coordonator: ș.l.dr.ing. Alexandru IOVANOVICI**

Sesiunea: Iunie 2024

# **REZUMAT**

În cadrul acestei lucrări va fi prezentat un sistem client-server, intitulat Task Forge, ce are în vedere execuția distribuită a unor simulări, conceput pentru a facilita gestionarea și execuția eficientă a acestora. Sistemul are două componente principale: partea de Client și partea de Server.

Partea de server este alcătuită dintr-o aplicație Flask care gestionează trimiterea sarcinilor, actualizarea de status, încărcarea și descărcarea fișierelor. Sarcinile de simulare sunt procesate folosind fire de lucru care execută sarcinile și utilizează funcții utilitare.

Partea de client oferă o interfață grafică utilizatorului care permite trimiterea sarcinilor de simulare și monitorizarea statusului acestora, la fel descărcarea rezultatelor.

Datele sarcinilor sunt gestionate într-o bază de date MongoDB, asigurând scalabilitatea în ceea ce privește gestionarea sarcinilor. Acest sistem permite execuția asincronă a sarcinilor, distribuită pe mai multe servere, sporind eficiența și fiabilitatea sarcinilor complexe de simulare.

**ABSTRACT**

This paper is going to be focused on the client-server system, titled Task Forge, which focuses on the distributed execution of simulation tasks. It is designed to facilitate efficient task management and their execution. The system is made of two main components: the Client side and the Server side.

The server side represents a Flask application which manages task submission, status updates, as well as file uploads and downloads. These tasks are processed using worker threads which execute the simulations using utility functions.

The client side provides a graphical user interface which allows the user to send simulation tasks as well as monitor them and download the results once the task is ready.

Task data is managed in a MongoDB database, ensuring scalability in task handling. This framework enables the asynchronous execution of the tasks, which is distributed across multiple servers, enhancing the efficiency and reliability of complex simulation tasks.

# **INTRODUCTION**

* 1. **Context**

What did the users do before the app?

* 1. **Motivation**

What is the proposed solution?

* 1. **Problem Statement**

What will be found in the paper?

**! Analiza cerintelor/specificatii**

**elemente teoretice:**

**cerinte functionale(mandatory, product feature): data upload, data sending, stocarea task-urilor intr-un db (la selectarea buton start se va permite etc.)**

**metodologie: AGILE-> know it when I’ll see it.**

**Pregatire Epicul simserver cu tichete, feature din subordine(ca un fel de la client)**

**Lucreaza la stil: linii mai groase etc.**

**Stories: descriu ce trebuie sa faca sistemul, document formal de requirements (daca nu este, se va transforma)**

**cerinte nefunctionale: view pt statusul taskurilor, GUI pt user**