**EEX5362 Performance Modelling 2024/25**

Deliverable 01

• Name: B. Galappaththi

• Registration No: 722510047

• SNumber: s22101386

• System: "An Among Us Match"

• Due Date: 31/Oct/2025

# 1. High-Level Problem & System Details

**System Identified**

A single, complete match of the multiplayer game Among Us.

**System Description**

A match of Among Us is a complex, session-based system. It is a time-sensitive race between two teams with opposing goals:

* The Crewmates (Team): A large group whose goal is to "process" a list of tasks (measured by Task Completed). Their victory is tied to their collective task throughput.
* The Impostors (Team): A small group whose goal is to "process" kills (measured by Imposter Kills), sabotage the system, and prevent the Crewmates from completing their tasks.

**The Core Performance Problem**

The system is a "performance race" balanced on resource management and time. We will analyze the "in-game system performance" the flow, bottlenecks, and throughput during the game.

The system's performance is defined by the balance between two key metrics, the Task Completion Rate of the Crewmates and the Kill Rate of the Impostors. The Game Length (in seconds) is our primary measure of the system's "total processing time" or "latency."

# 2. The Data Set

* Source: "Among Us Complete Dataset (29 players)" via Kaggle.com
* GitHub URL: <https://github.com/buddhikagalappaththi/EEX5362-data-analysis.git>
* Data Description: The data set is a collection of 29 individual .csv files (e.g., User1.csv, User2.csv, ... User29.csv), each representing the complete match history for a single player.
* Data Preparation & Cleaning: Before analysis, a critical data preparation step is required.

1. Combine: The 29 files will be combined into one single data file.
2. Clean & Transform: The Game Length and Time to complete all tasks columns are in a "00m 00s" text format. These will be converted into a numerical "Total Seconds" format for mathematical analysis.
3. Clean & Transform: The Imposter Kills and Task Completed columns contain a mix of numbers and hyphens ("-"). These will be cleaned so that a "0" is used instead of a hyphen for non-applicable rows.

* Key Data Columns for Analysis
* Game Completed Date
* Team (Crewmate or Impostor)
* Outcome (Victory or Defeat)
* Task Completed (Numerical)
* Imposter Kills (Numerical)
* Game Length (Will be converted to seconds)
* Sabotages Fixed (Numerical)
* Time to complete all tasks (Will be converted to seconds)

# 3. Performance Objectives

This analysis will focus on identifying the performance bottlenecks and efficiency of the "in-game" system.

1. Analyze System Throughput

* On the analysis primary objective is to measure the opposing throughputs of the system.
* Crewmate Throughput, I need to calculate the average Task Completed per minute (Total Task Completed / Game Length in minutes).
* Impostor Throughput, I need to calculate the average Imposter Kills per minute.
* This will show the "processing rate" of both teams.

1. Identify Performance Bottlenecks

I will treat "Sabotage" as a process that creates a bottleneck. Objective is to model the impact of this bottleneck on system latency. I will analyze the correlation between the number of Sabotages Fixed and the total Game Length to determine if sabotage is an effective "stalling" tactic that significantly increases the match's processing time.

1. Model System Balance (Resource Allocation)

The system's performance depends on a fair allocation of "resources" (i.e., the game rules) between the two teams. Objective is to measure this balance. I will analyze the overall win-rate (Outcome) for each Team (Crewmate vs. Impostor) to determine if the system is balanced or if one side has a significant performance advantage.