**Unit 6 Assignment**

Patricia Dimaano

Purdue University Global

IN451-01

Prof. Matthew Garberding

10 July 2025

The Clothing Store Dressing Room Simulator determines how many dressing rooms would be ideal for a new clothing store using quantitative data to help with decision-making. If customers are waiting too long for a dressing room, it means more dressing rooms should be added. Similarly, if some dressing rooms are not being used most of the time, it means there should be fewer dressing rooms. This application will allow testing with different configurations without actually physically building a mock-up store. The application has three different parts:

* **Data:** The *IScenarioResultRepository* and its in-memory implementation *(InMemScenarioResultRepository)* is defined in this layer. *InMemScenarioResultRepository* is the one that handles the storage and retrieval of the simulation results. It also has the *ScenarioResult* class that encapsulates the key metrics for every simulation run, such as the number of customers, rooms, total scenario time, average items tried on by the customers, average dressing room usage time, and average dressing room waiting time.
* **Business:** This layer contains the simulation's logic. The *DressingRooms* class uses a *SemaphoreSlim* to control customer traffic on the dressing rooms, making sure that only one customer uses a room at a time. The *Customer* class simulates customers trying on a random number of items (up to 6), trying on each item for one to three minutes. *RoomTimer* and *WaitTimer* are used to track room usage and waiting times. Every simulation run is managed by the *Scenario* class, which also creates customers and rooms, runs their simultaneous activities, and gathers data for the *ScenarioResult* class.
* **Program:** This layer starts the simulation, runs three predefined scenarios, and logs real-time customer activities to the console. After each scenario is finished, it will display a detailed summary of its results. After all three scenarios are finished, it evaluates which of the three scenarios is the best configuration based on the lowest average waiting time (*AvgWaitTimeMin*).

**Simulator Output**

A screen shot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.