Amazoom: Automated Warehouse

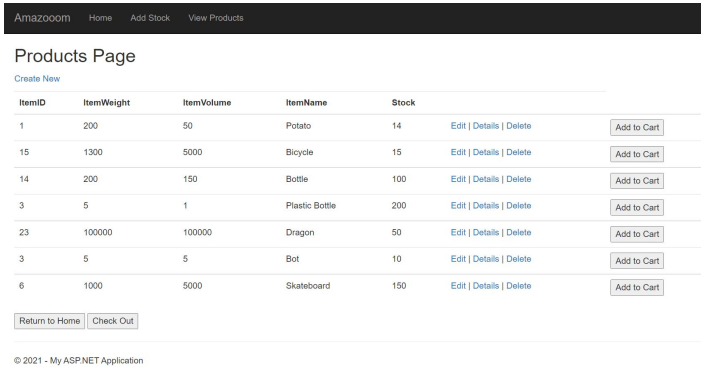
Metehan Sahin
Kevin Jiang
Lucy Hua
Nic Ricci

Tech Stack

Warehouse Computer

- **C# Console Application**
  - Allows initiation of multiple warehouses
  - Each warehouse initiation creates a new process that handles incoming orders, restocking, as well as robot + truck movement
- **Warehouse Interface**
  - Allows warehouse manager to easily manage warehouses at the clicks of buttons
  - Interface shows movement of robots, inventory and more

Tech Stack



Webserver Database and Interface:

- **ASP.NET (MVC)**
  - Interactive Web UI development
  - Real-time, customizable
- **SQL**
  - Portable database software
  - Fast query processing
- **Dapper**
  - Stores data in data tables
  - Executes queries and maps results from database to project file and vice versa
  - Convenient functions for INSERT, UPDATE, DELETE etc.

Overall Design

- **Warehouse**
  - **Warehouse Computer**
    - Handles communication and interaction between objects within the warehouse
    - Storage of information regarding robots (active and inactive), trucks (docked, waiting to be docked), items (list, location, and quantity), order queue (incoming, processing, ready for shipping), shelves (empty, non-empty, full and location) and racks
  - **Grid**
    - Shelves, racks and loading docks, loading dimensions from file
  - **Robot**
    - Pathfinding algorithm, collision avoidance, item loading/restocking
  - **Truck**
    - Docking algorithm
- **Webserver**
  - **Client Interface**
    - Product list, cart/orders, manually update inventory
  - **Database**
    - Warehouse inventory, order information

Demo

Client Order Querying

Updating Warehouse Item Inventory

Robot Pathfinding and Collision Avoidance

Warehouse Trucks

Warehouse Flexibility (Multiple units, Dimensions)

+ Other functionalities