Task 8.3

Isaac Pittolo – 7695438
Hoang Linh Bui – 100017631
Adam Bubonya – 9986677

Product Review

Items Completed:

- Sales records increases when you create a sale.
- In Sales, you can show items by date and year.
- Adding inventory is simple – merely typing in a record to the command line.
- Inventory Increases and decreases depending on records being added. Assumed to be working but there was an error in implementation.

Satisfying the Definition of Done

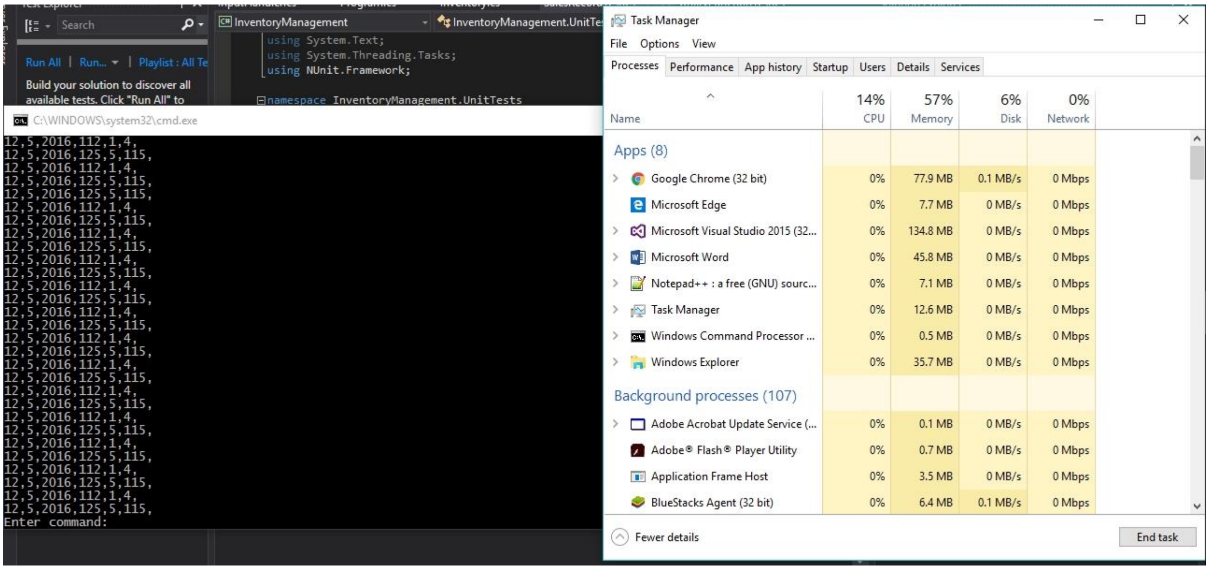
We have tested the demo with our tutor, who commented that our demo was working correctly and all sprint items were implemented.

The application loads almost instantaneously. This includes reading the whole database to RAM. We loaded 11,000 lines into the database and saw an insignificant rise in RAM and CPU usage, so low as to almost be unseeable. It took approximately 17 milliseconds.

Showing the database takes a bit of time, but this is the action of actually printing to screen rather than read/write. This is due to command prompt limitations and can be solved with a GUI.

Adding and showing additional records is also instantaneous. This is more than likely due to our means of storing data, as a .txt.

Overall, we are very satisfied with the speed of our application. In regards to functionality, all necessary functions have been added in this sprint. Improvements can be made but it all functionality and functions work really well.


Search
Run All | Run... | Playlist : All Te
InventoryManagement
InventoryManagement.UnitTe

Build your solution to discover all available tests. Click "Run All" to

```
using System.Text;
using System.Threading.Tasks;
using NUnit.Framework;

namespace InventoryManagement.UnitTests
```

C:\WINDOWS\system32\cmd.exe

```
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
12,5,2016,112,1,4,
12,5,2016,125,5,115,
Enter command:
```

Task Manager

File Options View
Processes Performance App history Startup Users Details Services
Fewer details
End task

Name	14% CPU	57% Memory	6% Disk	0% Network
Apps (8)				
Google Chrome (32 bit)	0%	77.9 MB	0.1 MB/s	0 Mbps
Microsoft Edge	0%	7.7 MB	0 MB/s	0 Mbps
Microsoft Visual Studio 2015 (32...	0%	134.8 MB	0 MB/s	0 Mbps
Microsoft Word	0%	45.8 MB	0 MB/s	0 Mbps
Notepad++ : a free (GNU) sourc...	0%	7.1 MB	0 MB/s	0 Mbps
Task Manager	0%	12.6 MB	0 MB/s	0 Mbps
Windows Command Processor ...	0%	0.5 MB	0 MB/s	0 Mbps
Windows Explorer	0%	35.7 MB	0 MB/s	0 Mbps
Background processes (107)				
Adobe Acrobat Update Service (...	0%	0.1 MB	0 MB/s	0 Mbps
Adobe® Flash® Player Utility	0%	0.7 MB	0 MB/s	0 Mbps
Application Frame Host	0%	3.5 MB	0 MB/s	0 Mbps
BlueStacks Agent (32 bit)	0%	6.4 MB	0.1 MB/s	0 Mbps

Items Not Completed:

While parts of the application could be improved as a whole, we have completed essentially everything we have set out to do this sprint.

Part of this was due to our sprint 1 carrying over to sprint 2, another factor would be our renewed work ethic after our failures in sprint 1 encouraged us to work as hard as possible.

Improvements

I think that next time we attempt something such as this it would be a good idea to implement at least a rough GUI to demonstrate what we need to. The command line application works really well and looks good, but it could still be further expanded upon with a GUI.

Another possible improvement could have been a bit better time management. In an ideal world, we could have contributed a lot more if we were not busy with other subjects and completed the project in considerably less time.