CAB302 Assignment 2

Inventory Management Application

Student Name: Luke Reynolds, Johnathan Gonzalez

Student Number: n18023481, n9821112

Date Submitted:

# Summary

This report documents the functionality, theoretical predictions, and experimental results of a simple binary searching algorithm. The algorithm was implements using a Java Integrated Development Environment which was used to record the number of basic operations performed by the algorithm and its execution time. Once this was achieved the program was then used to create a graph displaying the data in an easy to view format.

Contents

[Summary 1](#_Toc511563205)

[1. Description of the Algorithm 3](#_Toc511563206)

[2. Theoretical Analysis of the Algorithm 3](#_Toc511563207)

[2.1 Algorithms Basic Operation 3](#_Toc511563208)

[2.2 Best Case Efficiency 3](#_Toc511563209)

[2.3 Worst Case Efficiency 3](#_Toc511563210)

[2.4 Average Case Efficiency 4](#_Toc511563211)

[3. Methodology, Tools, & Techniques 4](#_Toc511563212)

[3.1 Methodology 4](#_Toc511563213)

[3.2 Tools 4](#_Toc511563214)

[3.3 Techniques 4](#_Toc511563215)

[4. Experimental Results 4](#_Toc511563216)

[4.1 Functional Testing 4](#_Toc511563217)

[4.2 Best and Worst Number of Basic Operations 5](#_Toc511563218)

[4.3 Average Number of Basic Operations 5](#_Toc511563219)

[4.4 Average Execution Time 5](#_Toc511563220)

[References 5](#_Toc511563221)

[Figure 1. Algorithm to be analysed. 6](#_Toc511563222)

[Figure 2. Graph displaying number of basic operations relative to dataset size. 7](#_Toc511563223)

[Figure 3. Graph displaying execution times relative to dataset size. (No warm up) 8](#_Toc511563224)

[Figure 4. Graph displaying execution times relative to dataset size. (With warm up) 9](#_Toc511563225)

[Appendix A: The Algorithm (Code) 10](#_Toc511563226)

[Appendix B: Functionality Testing (Code) 11](#_Toc511563227)

[Appendix C: Counting Basic Operations (Code) 12](#_Toc511563228)

[Appendix D: Averaging Basic Operations (Code) 13](#_Toc511563229)

[Appendix E: Measuring Execution Times (Code) 14](#_Toc511563230)

# Technical Description

A technical description of your program architecture, drawing reference to object-oriented design concepts such as polymorphism and abstraction. You may want to use a diagram to illustrate your type hierarchies and interaction between classes

This program was designed through a test-driven development approach. In general, tests were designed with a high-level scope and then refined to meet specific goals. The UML diagrams shown throughout the report are used to illustrate an architectural representation of the relationships among the packages, classes and other objects implemented throughout the program.

# 1.1 Program Architecture

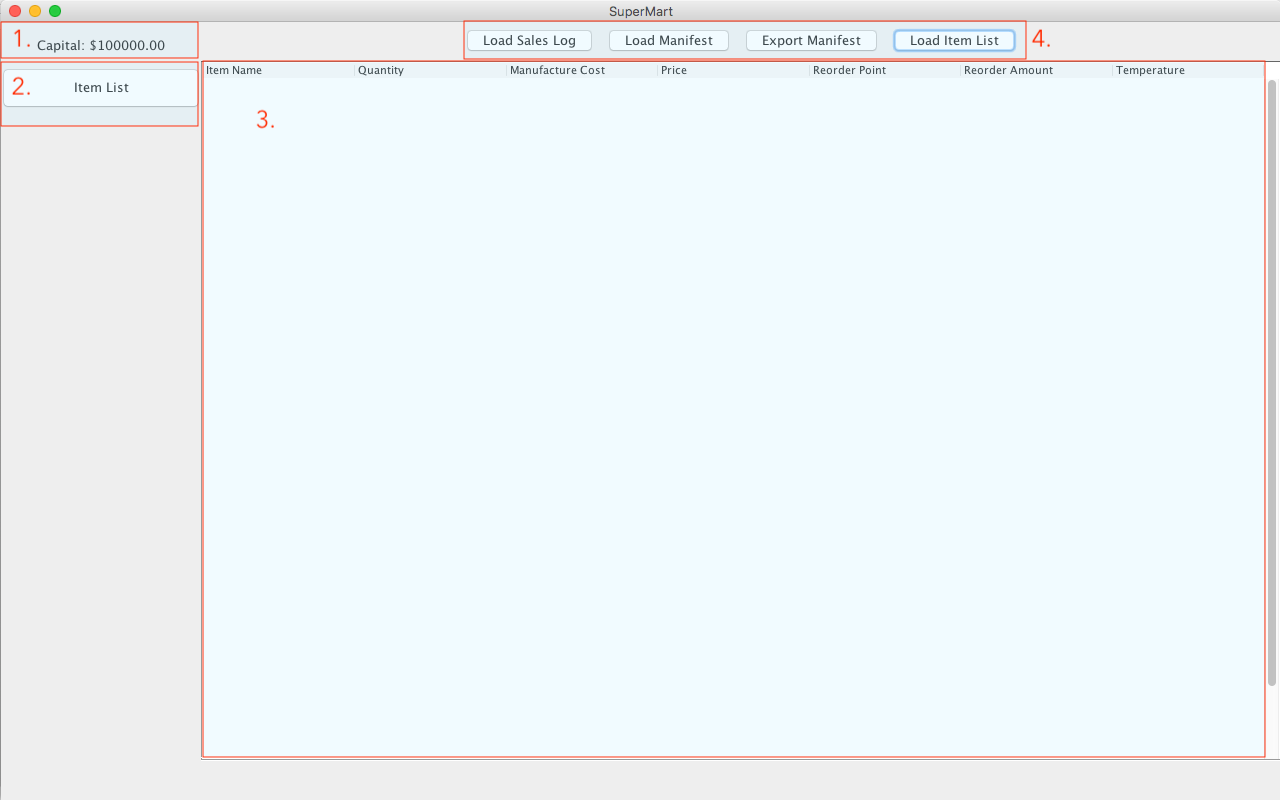
# 1.2 Type Hierarchies

# 1.3 Class Interaction

The worst-case efficiency scenario for this algorithm is when the array does not contain the search value. This causes the algorithm to execute the maximum number of basic operations for the array size until the left-most value is higher than the right-most value meaning that the search value is not in the array causing the algorithm to exit. This can be expressed as .

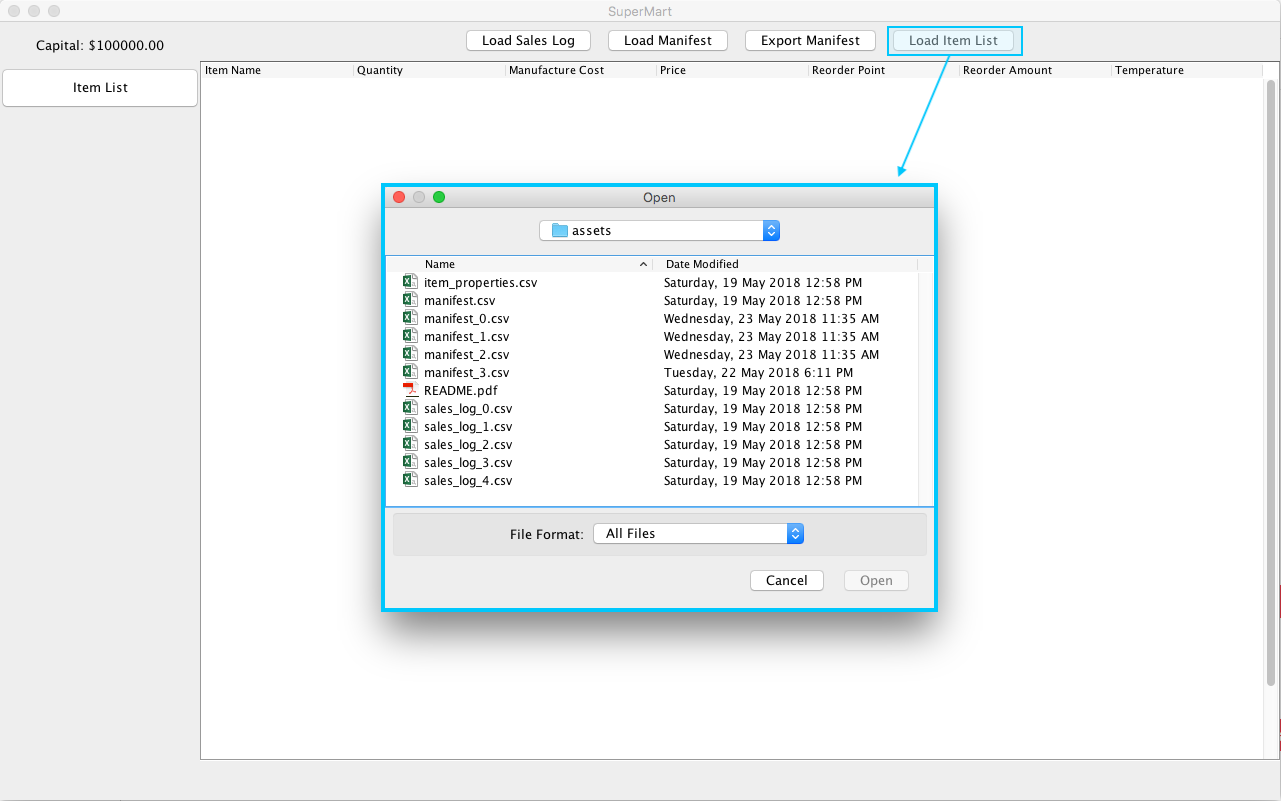
# 3. Graphics User Interface (GUI) Test Report

# 3.1 Home Page

The main screen of the user interface is laid into four main sections.

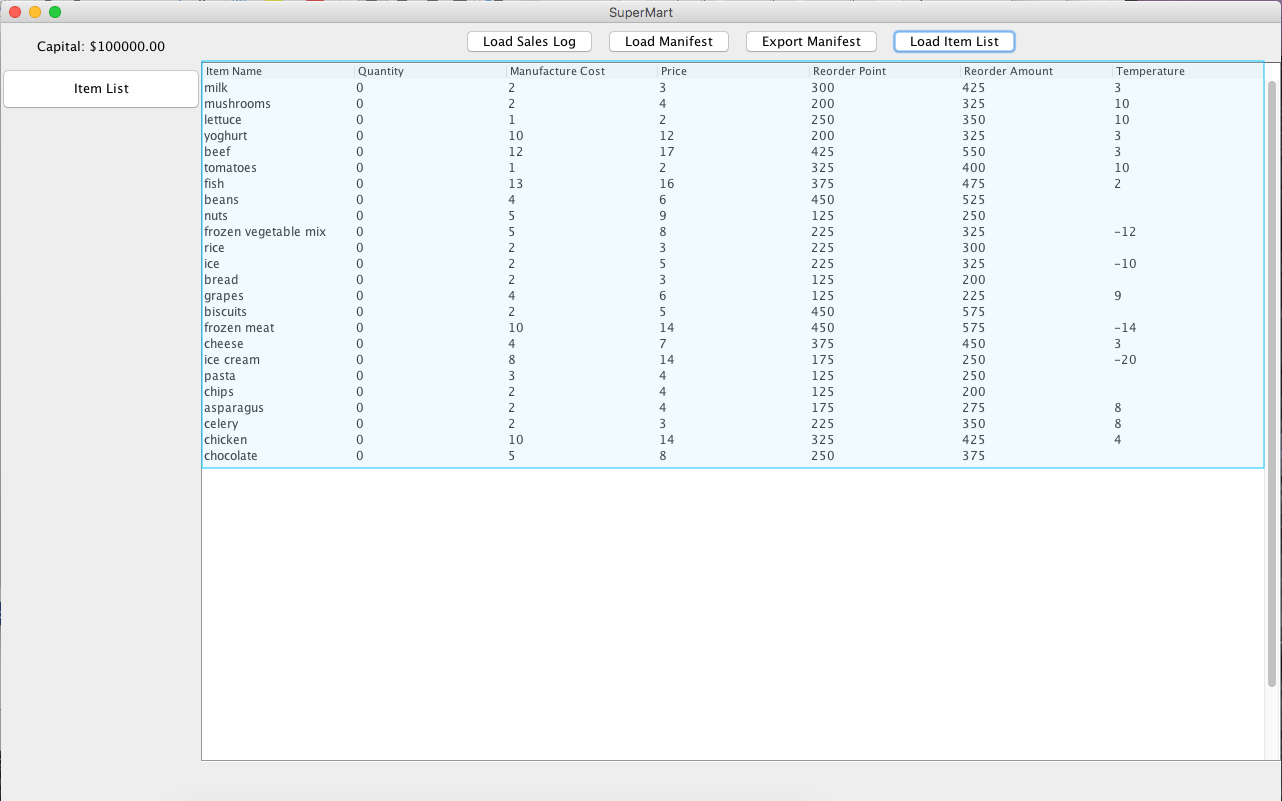
1. Current capital display
2. View items list button
3. Item list table
4. Load files buttons

# Item List



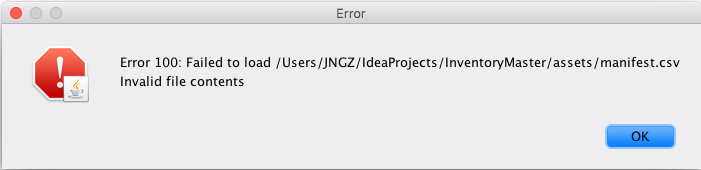
The load buttons section holds four buttons used to load the necessary files into the program.

1. Load sales log
2. Load Manifest
3. Export Manifest
4. Load Item List

In this example, we illustrate how to load the items list into the program. In the first frame, after clicking the **Load Item List** button a window prompt appears that allows you to select the appropriate file for loading.

After opening the item\_properties.csv file item list table is populated with the item list in a tabular format.

If you fail to load an appropriate item list file an exception will be thrown and the user will be shown this error message modal.



# 3.3 Generate & Load Cargo Manifest

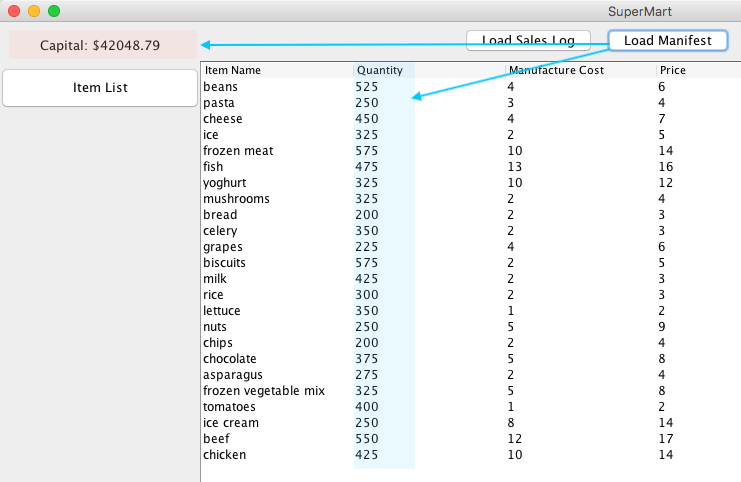
Generating a new manifest

To generate a new manifest the user would first click on the **Export Manifest** button. A manifest is then generated in the assets folder which can be loaded into the system.

Loading a new manifest

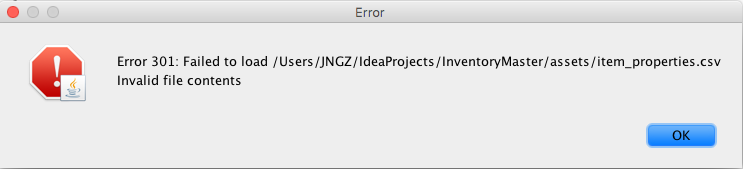
To load the new manifest the user should then click on the **Load Manifest** button. This will launch the file browser. In this example, the generated manifest is *manifest\_0.csv* which can then be opened and loaded.

Change in available capital

After the manifest has been loaded into the system the available **Capital** will be reduced to reflect the purchase and transportation of the goods. You will also notice that the quantity column has been updated to reflect the new stock levels.

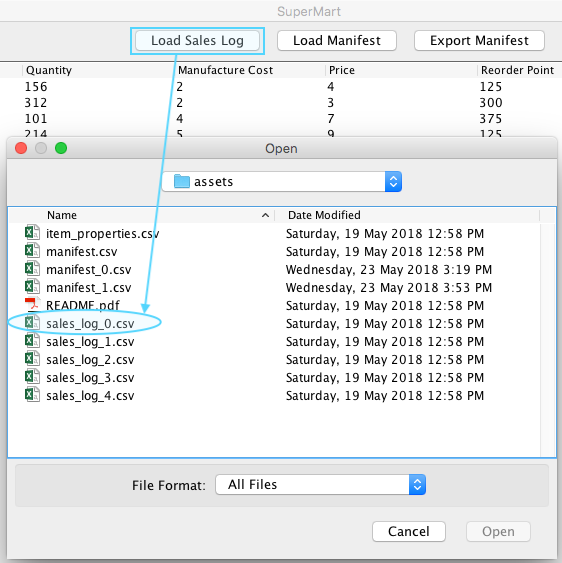
Manifest exception handling

Once again if the user fails to load the appropriate file type the system will throw an exception and display an error modal.



3.4 Load Sales Log

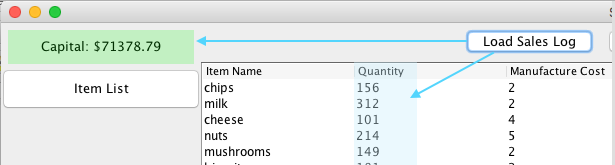
Loading a new sales log

To load a sales log, the user would click on the **Load Sales Log** button. This action would present the file browser, where the user would select the appropriate sales log file. Unlike the export manifest function, sales logs are not generated by the system.

As sales logs are loaded into the system the available capital value is increased to reflect the sales made. You will also notice that the quantity of items available in the item list will decrease with respect to the sales made in sales log.

In this example, the available capital is increased to $71,378.79.

Sales log exception handling

As with previous examples, when a user tries to load a file that is not a sales log an exception is thrown and an error modal is displayed. This exception is passed up through the hierarchy of classes and displayed on the GUI.

# ../../../Desktop/Screen%20Shot%202018-05-23%20at%206.00.49%2

# References

* <https://docs.oracle.com/javase/tutorial/uiswing/components/table.html#simple>
* <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>
* <https://stackoverflow.com/questions/20473325/gridlayout-java-center-alignment>