*Greyden Scott & Alex Holm*

*[Company Name] | N9918205*

Assignment 2: Inventory Management Application

CAB302 Software Development

Table of Contents

[Technical Description 2](#_Toc514661207)

[Item 2](#_Toc514661208)

[Stock 2](#_Toc514661209)

[Truck (Abstract) 2](#_Toc514661210)

[RefrigeratedTruck 2](#_Toc514661211)

[OrdinaryTruck 2](#_Toc514661212)

[Manifest 2](#_Toc514661213)

[Store 2](#_Toc514661214)

[UML Diagram 3](#_Toc514661215)

[GUI Test Report 4](#_Toc514661216)

# Technical Description

## Item

The item class instantiates an object type of item which consists of several key properties:

Name, Cost, Price, Reorder Point, Reorder Amount, Storage Temp and Quantity.

These properties are loaded through an item properties files. The polymorphic type Item(s) are instantiated based on the file supplied. The item object is stored in a Stock type object

## Stock

The stock class represents a collection of items, this class is utilised for representing different types of inventory based on their properties like; store inventory, cargo to be loaded onto trucks, cold items and non-refrigerated items, inventory to be reordered.

These differing properties categorise the stock, which assists when generating manifests, as some stock requires refrigeration and needs a particular truck type instantiated which is why the stock class can sort through the collection of items by temperature

## Truck (Abstract)

Truck is an abstract class for the two truck types, RefrigeratedTruck and OrdinaryTruck. It has the follow abstract functions: getCapacity, getCost, getCargo. In addition, the Truck class also has a function to return the inventory on the truck as a list and the inventory on the truck as a Stock type.

## RefrigeratedTruck

The RefrigeratedTruck class extends the Abstract class Truck, It overrides getCost, getCapacity and getCargo with its own specific code related to the Rerigerated Truck type. In addition, the Refrigerated truck has 2 addition functions, getTemperature which returns the temperature the truck operates at and setTemperature, which sets the trucks temperature based on the coldest item in its inventory.

## OrdinaryTruck

The OrdinaryTruck class extends the Abstract class Truck, it overrides getCost, getCapacity and getCargo with its own specific code related to the Ordinary Truck type.

## Manifest

The manifest class creates a collection of trucks on instantiation, all with cargo ready to be exported to a manifest file. The manifest class has a function to return the collection of trucks as a List.

## Store

The store class uses a singleton pattern, ensuring that the only once instance exists. It does this by check if it already exists, if so it returns the existing Store, otherwise it creates a new one.

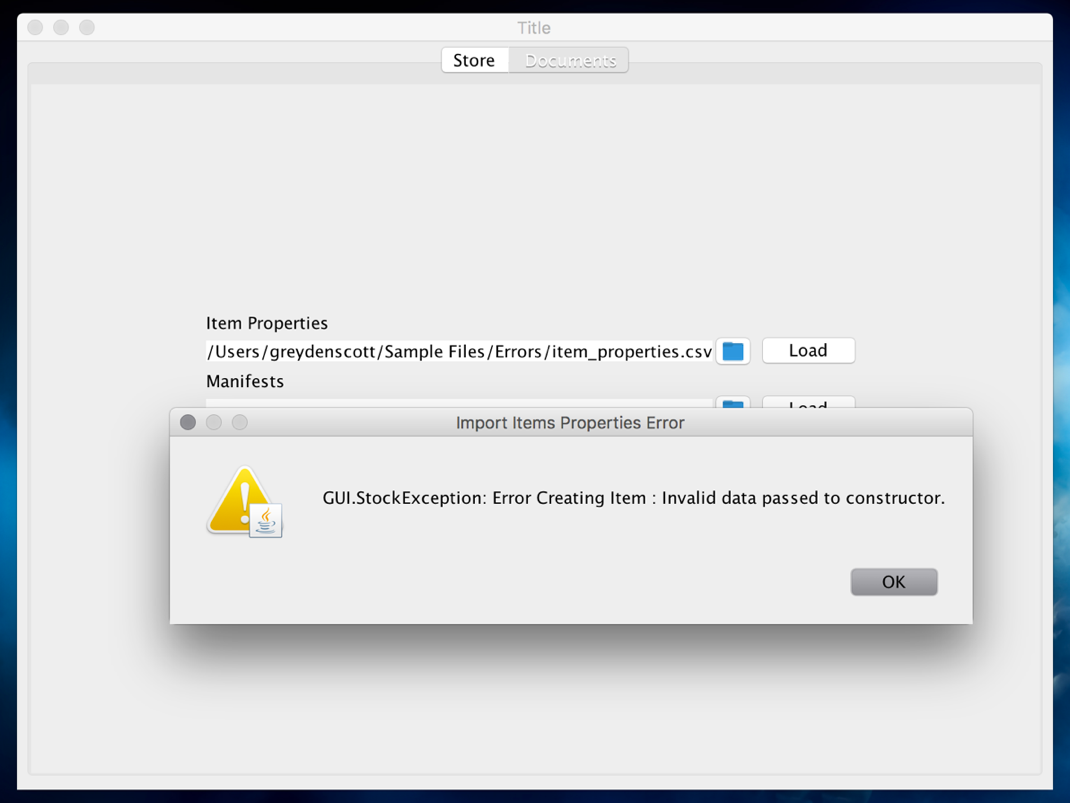
## UML Diagram



# GUI Test Report

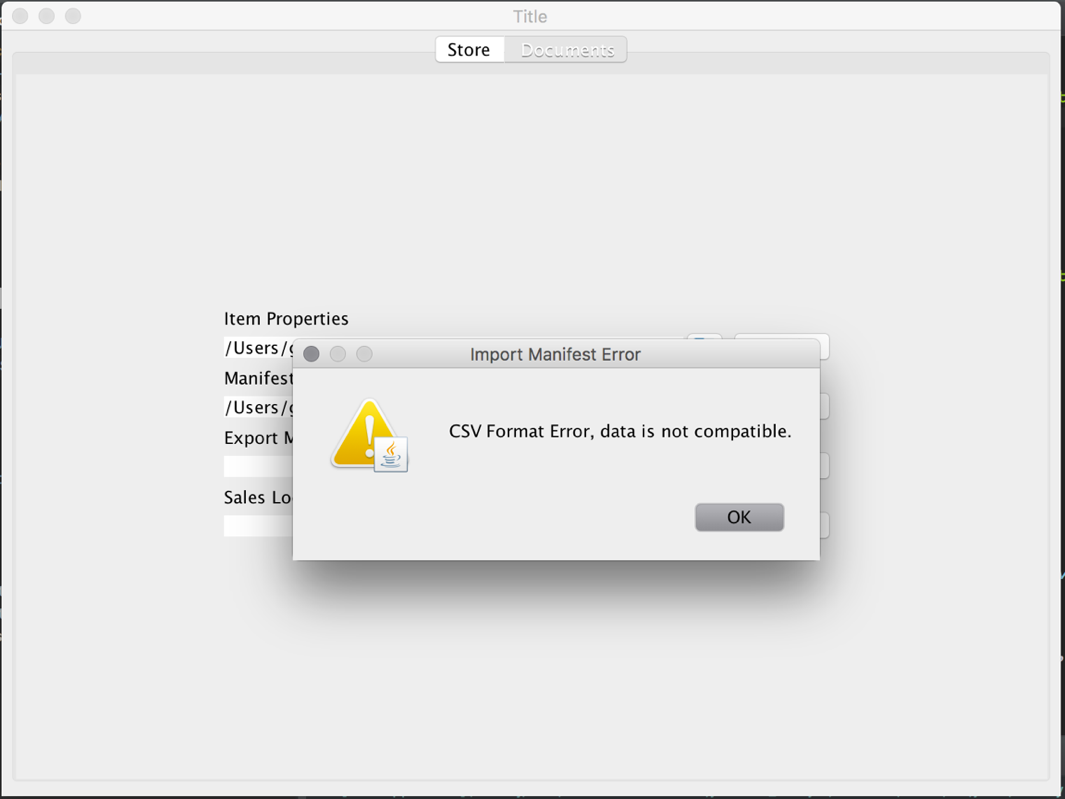
## Item Properties Import Error

Item Properties need to be loaded into the application, this is done using a CSV File Reader. If the file path does not go to a file, the data provided is invalid or values are missing, this error will appear.



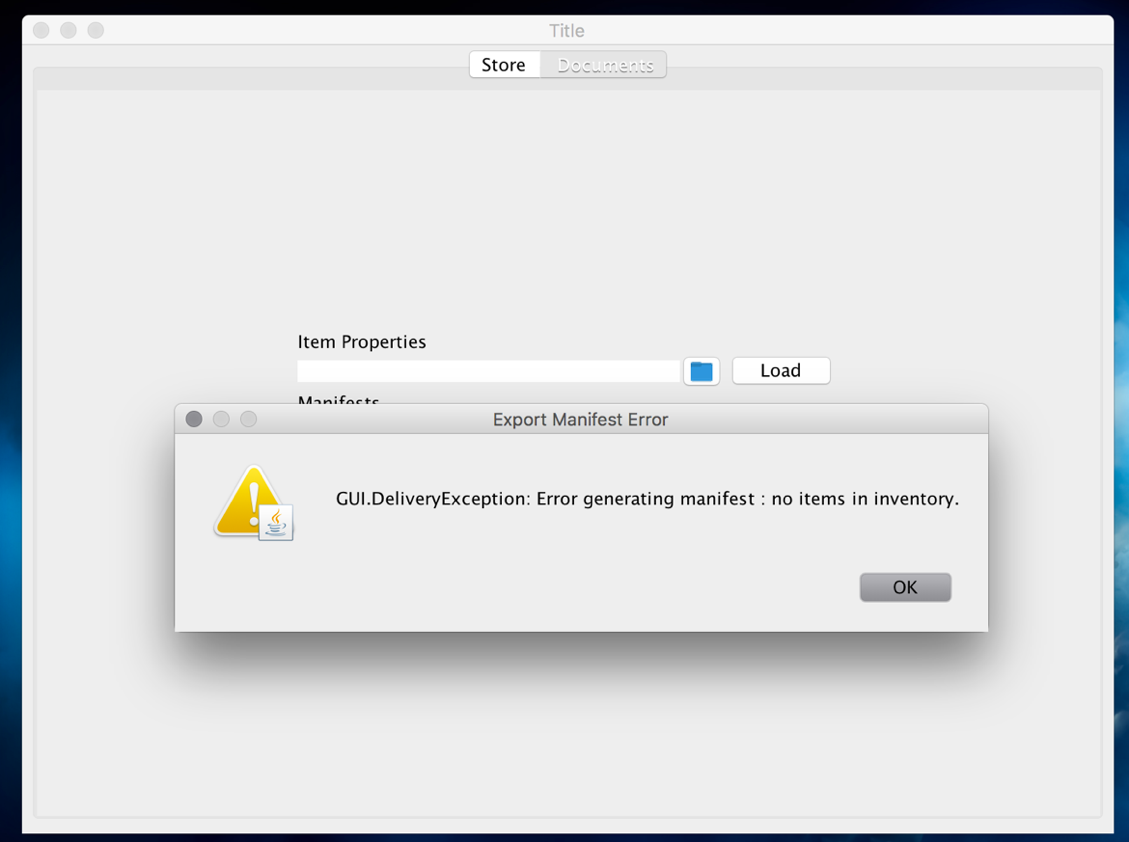
## Import Manifest Error

The manifest is used as a list of items to order. If the file path does not go to a file, the data provided is invalid or values are missing, this error will appear



## Export Manifest Error

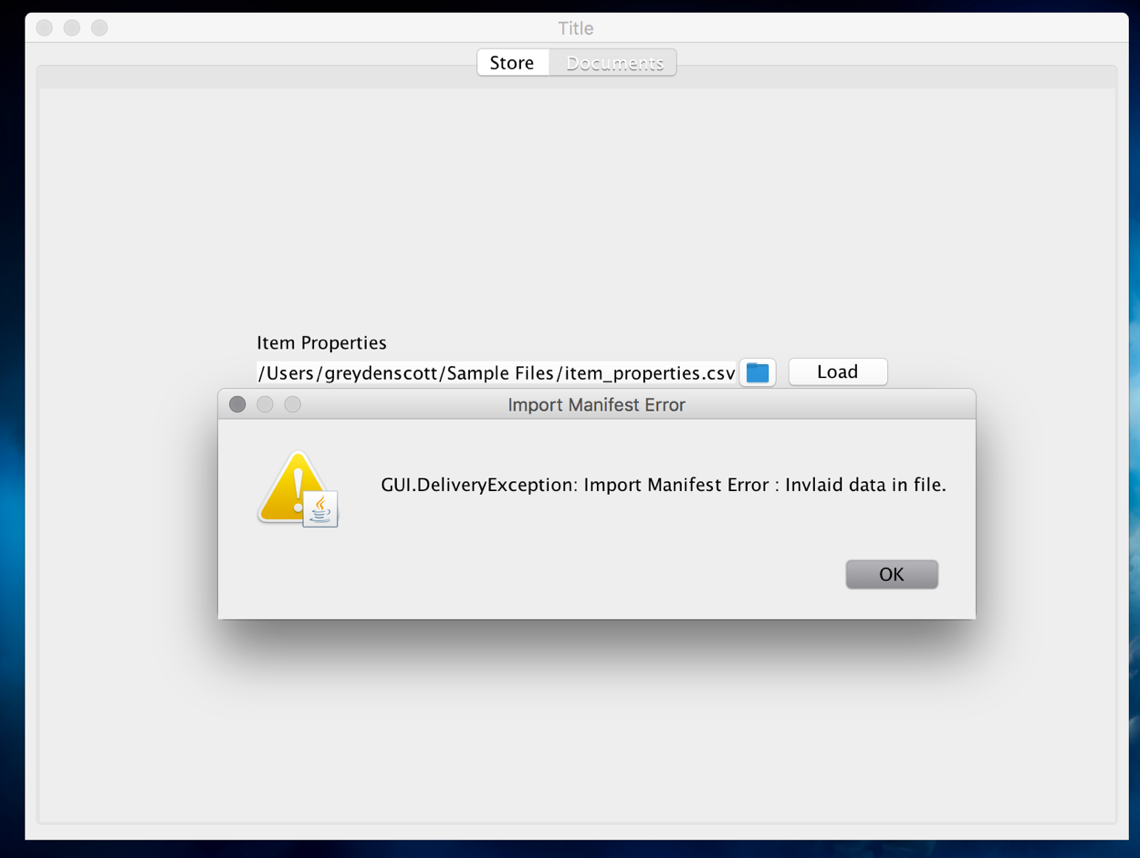
This error message appears when the user attempts to generate a manifest while the inventory is empty.



## Import Sales Log Error

The sales log is used to adjust stock and compares stock amount to its reorder amount.

If the file path does not go to a file, the data provided is invalid or values are missing, this error will appear



## No File Selected Error

If the user does not select a file through the FileChooser or the TextArea is blank, these errors will appear depending on which button is selected.

## 