**Testing Strategy for the development of the Wood Stocks Inventory Management System**
**Prepared by: Darren Gansberg**

The application was developed using the Microsoft Visual C# 2008 Express Edition IDE. It should be noted that this edition of Visual Studio does not contain the testing frameworks contained in other versions of Visual Studio. Consequently, in developing a unit test strategy toi carry out tests during development alternative unit testing frameworks were reviewed. Ultimately a decision was made to use the NUnit testing framework (http://www.nunit.org/) in order to develop unit tests whilst developing the application.

The unit tests written for the application are contained in the WoodStocksIMS.UnitTests project contained within the Solution for the application.

After becoming aware of the test driven development method for developing applications it was decided that attempting to utilise TDD practice in developing the application would be pursued. This meant that unit testing became the key focus during development . This heavily influenced the way in which the development process was approached. The development process would begin by identifying a unit test to test functionality that was required of the program in order for use case functionality to be delivered. Once a unit test was identified, the code for the test was developed. In total 23 unit tests were ultimately developed whilst writing this application. After the test was developed code would be written with a view of passing the unit test.

Developing the application using the above unit testing process meant that regression testing was a continuous feature of the development process. As the program was incrementally built, with code being added to pass newly developed tests, each time the unit tests were run regression testing took place. Where newly added coded, or code changes, resulted in prior written tests failing, the code was reviewed/analysed to determine the cause of the problem. Code was redesigned and rewritten until the established unit tests would pass. This process was iterative and continued for the duration of the project.

Unit testing was focused upon testing components that were regarded as critical to functioning of the program. These components included the components responsible for importing and exporting toy data, and responsible for achieving sorting functionality.

Developing the application in the above manner meant that running the debugger to debug the program was less frequent than it otherwise may have been. However, when it became difficult to understand why unit tests were failing, the functioning of the application would be tested by using the unit test code temporarily as a small test program and running the debugger. Breakpoints were set at lines of code suspected of being the cause of problems and stepping through the execution of the program until the problem was identified and resolved.