This document will distinguish the two key points of the design of this system. Those points are the designing of a forecast system from a conceptual perspective, and the implementation of inheritance from a programming perspective.

The main focus of designing this software was creating the methods for generating forecasts from previous data. This seemed as an opportunity to try and apply some of my knowledge of mathematics, and to try and apply neural networks. Hence after doing some research on neural networks it was decided to create a neural network made up of 4 layers, with the first three having 12 nodes, corresponding to the amount of input, and the last layer having 3 nodes, corresponding to output. Due to my lack of experience the equation for calculating the value of each node was simplified to a weighted sum, without any activation function or biases. That made it easier to derive the equations for backpropagation, however, the neural network may not perform well at all without those two key features. Regardless, after deriving the necessary equations the design of the neural network was completed.

For the statistical method I used my personal knowledge of mathematics. Presuming that same history in price result in same outcome in price, a way to reduce each price history to an ambiguous and comparable dataset was needed. Hence calculating the percentage change in price was used, as it does not change with scale, allowing sales of different magnitudes to be compared. However, the different speed in price changes would still be preserved after the conversation, hence if one price pattern is just another price pattern scaled along the x axis, they would register as different. Hence price patterns are regarded similar even if they are scaled along the y axis, and this was presumed to be sufficient.

There is only one case of inheritance in the design. In the early design stages it was decided that the Forecast class and the Node class would be superclasses. NeuralNetworkForecast and StatisticalForecast would inherit from the Forecast class, and InputNode, HiddenNode and OutputNode would inherit from the Node class. However, within the Forecast subclasses there were no differences in methods and attributes – the only difference between them would be whether or not it would be chosen for the neural network to learn from. This would be substantially easier to implement using a Boolean attribute which can be checked, rather than trying to distinguish between classes. The node subclasses had a different type of problem – the output and hidden nodes had the same methods and attributes, and the input nodes lacked the method to calculate its value and the attributes of previous nodes and weights leading to it. Hence it was decided that the hidden layer and the output layer would be one subclass, ComplexNode, and the input nodes would be of class Node. It would have been easier to just have one class for all of them, but there were no other opportunities to implement inheritance, and this piece of work needs to show software engineering. Thus the project has one instance of inheritance.

The conceptual and programming parts have been covered in this document. However, a more detail derivation of the backpropagation equations and the algorithm for the statistical method can be found in the file vs00264\_maths\_appendix.pdf. In the bibliography of that file could be found the material that was used to study how to perform backpropagation.