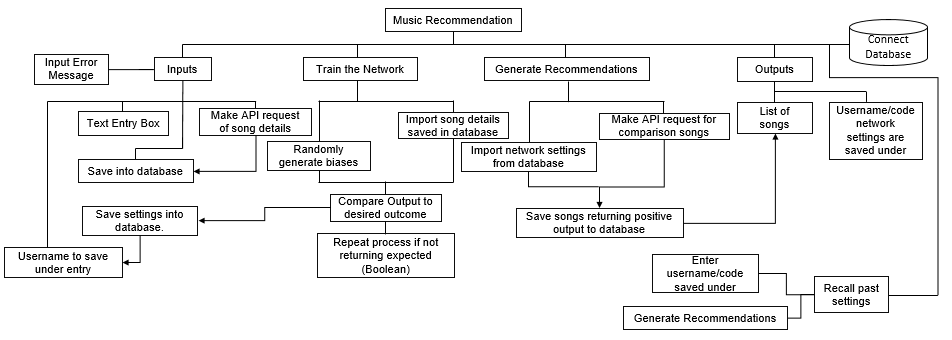
Design of the solution

Decomposing the problem

*“3.2.1 Decompose the problem*

1. *Break down the problem into smaller parts suitable for computational solutions justifying any decisions made”*

In order to break down how my program will run, I have created a hierarchical diagram for the processing behind the recommendation. This breaks down processes that each element of the program has to do to be able to its basic levels. This does not detail much of the interface and user interaction but it allows me to see each of the aspects I will need to develop and how data and processes interact between different branches of the program.

The main parts of my program are the training and application of the neural network, as such the above diagram tends to focus on those and how the interlink with each other, and the inputs and outputs involved. Everything that takes place will be saved into a database connected when the program loads.

The main idea behind this project is that the features of songs can be enumerated to a base level and that by doing this it would be possible to determine what about a song is important to a given person. This unlocks the potential to automate a recommendation process to be done without human interaction after the user enters a set of songs and clicks a ‘go’ button. This automation is where using a neural network presents its advantages, by having a pre-determined set of inputs and (knowing that the user already likes these songs) outputs you can set up the program to configure itself randomly until it reaches the expected output. This process is effectively a crude mimicking of a learning brain and allows the network to potentially simulate a human’s preference in music. By taking a new set of input songs (and data sets) without a known output, by passing them through the same set of conditions that are configured in a way that was determined correct in the ‘training’ stage of the process, it should be able to return a decision that states whether it believes the user will like a song. These outputs should be displayed in an easy to understand list, the value returned by the network should be converted into a Boolean result giving a yes or no decision.

The perfect simulation of a human’s though process would be mimicking the full complexity of connections, but considering that would require over 100 billion virtual ‘neuron like’ interlinked connections it would be impractical and take a huge amount of processing time and power. Therefore, I should take an abstraction of the idea taking it down to a few layers of inputs, biases and outputs, this should allow it to output reasonably accurate values while not taking a huge amount of time to work through its processes. Data such as the users name, and other personal information does not need to be taken into account when making recommendations as the system is basing it entirely off of the chosen songs and the numerical data associated with them.