**Name: Thankson Ernest**

**ID: 21452020**

In trying to create our mp3 player, we made a search for libraries that could be used in playing mp3 files. We used the first library we came upon: the javazoom library. We then went on to create the graphical user interface for the mp3 player. We were able to circumvent hardcoding the interface by using eclipse’s JavaBuilder graphical user interface. We successfully created a callback function that allowed a user to choose an mp3 file that allows a user to choose an mp3 file to play.

At this point, we could not find a way of making our Player object pause when it begun to play an mp3 file. We found out, however, that we could use threads: we would make the mp3 file play in a thread we had created and then call the wait() and notify() methods of the thread to pause and resume playing the mp3 file respectively. However, we found in testing these methods that the wait method always generated an exception when called. Even after an extensive search, we found ourselves left with no choice but to use the deprecated suspend() and resume() methods of our thread object. To stop a playing mp3 file, we ended up using the deprecated stop method of our thread object because all other methods we tried were generating errors.

To improve our program we would use a completely different library that would allow us to pause and stop playing mp3 files without having to resort to threading. To extend the player, we would make it able to play other audio file formats in addition to mp3 files. One thing I learnt is to figure out completely how to implement a solution before implementing it instead of figuring out how to implement particular features as you go.