Testing Approach:

To approach testing for this system, I decided to split them into two parts: black box and white box testing.

For white box testing, I employed unit and integration testing. These two are necessary and valid approaches because I am the sole developer of this system as well as its tester, thus I have full knowledge of the implementation details and can change the internal structure of the code as needed.

On the other hand, examples of black box testing that I have used are functional testing, compatibility testing, and performance tests. Functional tests such as category partition tests were used to verify inputs and their corresponding outputs. Furthermore, I used compatibility testing in conjunction with performance tests to not only verify that the system ran efficiently, but that it ran efficiently on all operating systems.

Evaluation:

Due to the nature of this course and ILP, I only managed to start the testing phase when most of my system was finished. Thus, faults in the unit classes were not caught particularly early into development. If I had more resources (time, personnel, etc), the testing phase would have more accurately reflected a professional setting where unit classes are coded first, tested, then integrated with others.

Furthermore, the lack of resources, as well as the nature of this system only existing as a back end, meant that there were some limitations on the types of testing that I could use. One class of testing that I could not utilise was security testing, as I do not carry out transactions and/or store card details, while another, usability testing, was not possible as there was no front end to my system as well as not having any alpha/beta testers.

Although I have created tests that should cover most edge cases in general, there will be some that have not been found. This is due to not having more time to develop these tests, as well as not having enough data to test the performance in extreme cases. However, for something like the performance tests, it is unlikely for these situations to ever occur and thus the testing that I have done will be sufficient for now. Another limitation is that during compatibility testing, I did not have access to a Linux based system, so data for that operating system is incomplete.

On the whole however, my testing approach seems to cover most aspects of this system, by verifying the software at the unit, integration and system level, as well as ensuring that it runs within the required timeframe.