Document Updater - Notes during testing

- Without realising I tested for some of the error cases mentioned in the description. So far I got the error below **as expected**:

```
ERROR: updates folder not found in target directory
```

```
ERROR: document " Fruits " doesn't contain an appropriately formatted address
```

- Decided to test for the presence of both "allowlist" and "droplist" and got the error below **as expected**:

```
ERROR: both droplist and allowlist found in target directory
```

- I automatically deleted every scenario from my Decision Table that included both "allowlist" and "droplist"

- I initially wanted to create a new directory for every test case. I thought that it would lower the chances of a mistake because I wouldn't have to delete my "finals" folder for every test (what if I thought I'd deleted it but I hadn't? I might end up with false results). But in the end, due to lack of time I decided to have two test directories, one with an "allowlist" and one with a "droplist". I then had to improvise as there were cases where it made more sense to keep the same data and just change the name of the file from "allowlist" to "droplist" and vice versa.
- There were possibly many more scenarios to test but had to stop because of time.

Conclusion:
- This project has been quite challenging in the sense that I had to seriously consider which test cases were worth testing and which not. I had to change my approach a few times as I was getting more familiar with the application's behaviour. On top of that, I felt like I had to be extra cautious when manually entering the data, files etc, to make sure I don't produce false test results.
- Thoughts/Considerations:
• How would I create test automation for this project? I'd have to write some code that would create directories and files and I'd have to populate these with data generated using "faker". This is a project in itself.
• What could I have done differently? I feel like I spent too much time creating the decision table before I had the chance to fully understand the functionality of the application. That meant that I had to update my decision table a few times. I wonder if I should have run a couple of exploratory tests before making the table. The main reason being that by the time I started finding the more interesting scenarios and bugs, I was already fatigued from the decision process and the amount of different scenarios to test felt very overwhelming.
• Didn't have the chance to test edge cases e.g. 1000 files in "originals" and 2000 files in "updates". 'Faker' would offer such possibilities for testing.