Steps for this challenge:

1. Spend some time reviewing the description and scenarios above in order to gain some understanding about what the program is expected to do. Afterwards, try hand-crafting a scenario or two of your own and run the program to *explore* it, particularly if you're not sure of some of the behaviour, taking caution - maybe there are bugs...
2. Next, spend at least 30 minutes using appropriate techniques from these first two phases to better understand your task and the program, to plan scenarios the program may encounter and hence plan what tests you will run.
3. Create a script (or scripts) that makes use of Python Faker in order to create data you will use for your testing, set to use the "en\_UK" locale. You should be using Faker-generated surnames as filenames (also in droplist/allowlist content) and Faker-generated addresses as the content of the documents in originals and updates.
4. Run your testing session however you think is appropriate, for no more than 2 hours. Feel free to automate as much of the running of tests and checking as you wish. Don't feel limited to *only* use test data generated before you began running tests - make new data as you go, as you feel is necessary i.e. in order to produce the minimal steps required for reproduction.
5. After your testing, make sure you create feedback that would be appropriate for the intended audience i.e. bug reports, summary of testing.

* Example 1: A document with a particular name exists in originals, and a droplist file exists which lists that document name. That version of the document must not ultimately be included in finals.
* Example 2: A document with a particular name exists in originals, and an allowlist file exists which lists that document name. That version of the document would have ultimately been included in finals, but a document of the same name also exists in updates, and it's that version of the document that must ultimately be included in finals.
* Example 3: A document with a particular name does not exist in originals. A droplist file exists which lists that document name, but has no effect. A document of the same name exists in updates, and it's that version of the document that must ultimately be included in finals.
* Example 4: A document with a particular name exists in originals, and a droplist file exists which doesn't list that document name. As there is no document of the same name in updates, the version of the document from originals must ultimately be included in finals.

| **What’s tested** | **Expected result** | **Actual result** | **Notes** |
| --- | --- | --- | --- |
| Coach document in originals and also in the droplist | Coach document must not appear in finals | As expected |  |
| Coach document in originals and also in the allowlist, but there’s an updated version of coach in updates | Only Updated version of Coach shows up in finals | As expected | What happens when you get two people with the same last name and different addresses |
| Coach document not in originals but is in the droplist and the updates. | Coach from updates appears in the finals directory. | As expected |  |
| Coach document in originals directory but not in the updates or droplist. | Coach from originals appears in finals directory | As expected |  |
| Coach document in originals and also in the updates, but not in allowlist | Coach from updates appears in finals directory | As expected |  |
| Coach in originals but not in allowlist or in updates | Coach does not appear on the finals directory | As expected |  |
| An empty droplist | No effect | As expected |  |
| An empty allowlist | No effect | Brings up an index out of range error | An empty space is enough to bypass this error. This is a bug |
| Two different documents with the same last name. One in originals and the other in updates | The version in updates appears in finals | As expected | This means that two documents with the same name cannot exist in the system, no matter how different their content. |
| Two different documents with the same last name. Both in originals with the last name on the droplist | Neither show up on the finals directory | The most recently added document appears on the finals | For some reason, the document on the droplist is completely ignored. I entered two identical names into the droplist and The result remained the same. |
| Two same named documents listed on the allowlist | No effect | As expected |  |
| Remove lines from the addresses in documents in updates | Should return an error | No error returned | This is a bug |