Project Proposal

I work as the database administrator for a museum and one of my job duties include entering in information for all the minerals into a database.

We have a lot of minerals that are not yet in our database, because they were recently donated, or because they are part of our backlog. Our data usually comes in excel spreadsheets like this from another collector.

B	C	D	E	F	G	H	I
Collec	Specimen	Locality (as transcribed)	Note	Country of Origi	Case	Shelf	Image1
Chinese	Schorl xls. (Tourmaline)	Babu, Guangxi Province, China		China	W	7	
Chinese	Siderite ball on Arsenopyrite xls.	Nandan, Guangxi Autonomous Region, China	ID - Raman	China	W	7	
Chinese	Silver (wire)	Fenhuang Co., Hunan Province, China		China	W	7	
Chinese	Zhonghuacerite-Ce	Bayan Obo, near Baotou, inner Mongolia, China		China	W	7	
Misc.	Calcite twin xls.	Poona, India		India	Z-piano	1	
Misc.	Epidote xls. On Magnetite xls., Quartz xls.	Dashkesan, Azerbaijan		Azerbaijan	Z-piano	1	
Misc.	Malachite (botryoidal) xled.	Kasonpo Mine, Katanga, Congo		Democratic Republic	Z-piano	1	
Misc.	Malachite botryoidal	Mubombe Mine, Katanga, Congo DR		Democratic Republic	Z-piano	1	
Misc.	Malachite Stalactites	Star of the Congo Mine, Lubumbashi, Katanga, Congo		Democratic Republic	Z-piano	1	
Misc.	Malachite Stalactites	Star of the Congo Mine, Lubumbashi, Katanga, Congo		Democratic Republic	Z-piano	1	
Misc.	Pargasite xls. (huge)	Kolonna, Uva Province, Sri Lanka		Sri Lanka	Z-piano	1	

We also have photos of our minerals that also need to be processed and placed into the database.

The way our current frontend database is set up, users have to enter information in one at a time to create an entry.

I would like to create a python project that will allow this process to be streamlined for two of the most time-consuming processes: localities and photos.

Localities

We currently have to add localities to our objects one at a time. The steps go like this: type in the locality that a mineral came from, do a search in our list of localities, and then pick the correct one out of many, or add if the locality doesn't currently exist.

The reason why we have to do these one at a time are twofold. First, localities are idiosyncratic and do not always match exactly what we have in our database. Some people might put in their mineral from Boston, Mass under the locality Boston, Sussex County, Massachusetts, USA. Someone else might put it in under Boston, Mass., United States.

Second, over the past century, many names and spellings of localities have changed. For example, we have a lot of 'Rhodesia' minerals from old collectors that need to be changed to 'Zimbabwe'. The USSR is another example. Because of this, we cannot do an exact 1:1 lookup.

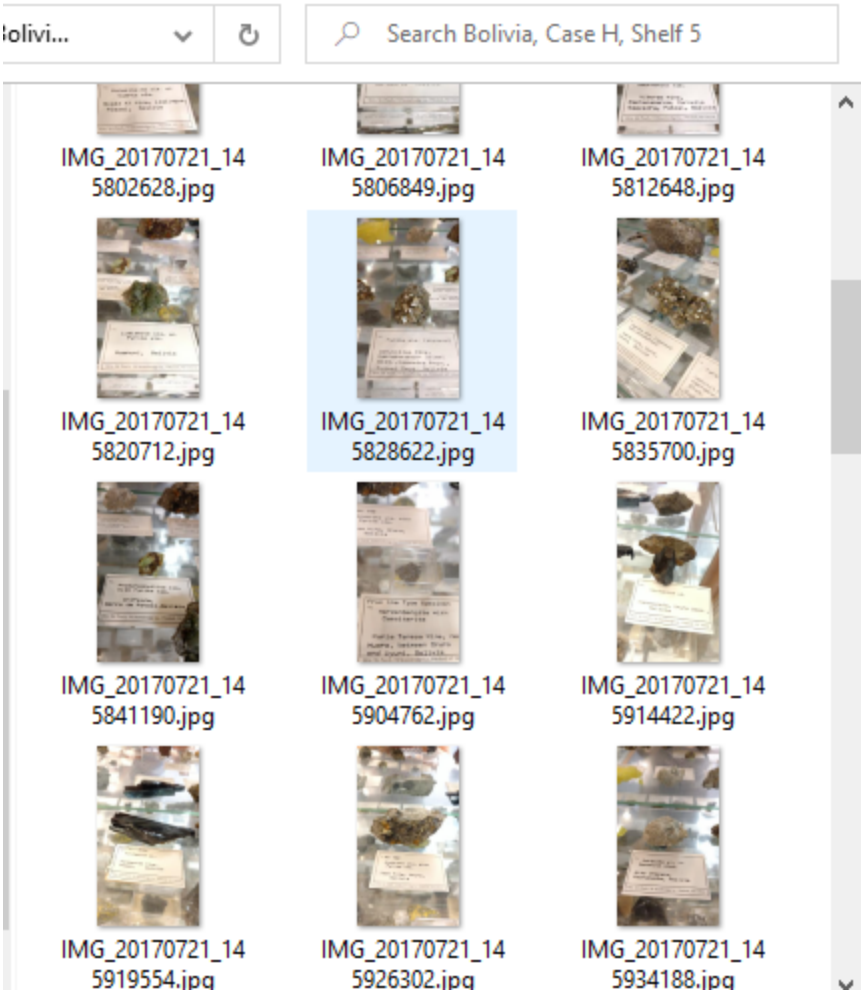
I would like to take the string of localities and do a fuzzy look up with what is currently in our database.

If it doesn't exist in our database, or the fuzzy look up returns some value that says its unlikely it matches with anything in our database, I would like it to go to mindat.org and return with the closest locality similar to the locality string.

The final result should be a csv with the id of a locality in our database that I can just copy paste into our sql database and update the results.

Photos

Our pictures come named by default settings. Usually img20938403_209340280349.jpg.



I would like to rename these files so that they are renamed to the object number of the mineral. Then I would also like to take the metadata print it out to a csv along with some other bits and bobs.

The final result should be a csv document that I can use to import into our database with all the columns filled out.

Libraries

Fuzzywuzzy
This is a fuzzy look up library I can use to match the strings to localities we have in our database even if they are not a 1:1 match.

Pypi
This library gets metadata from photos.

Websites

www.mindat.org
this is the world standard for minerals and their localities that I will be using to make sure our localities are correct.