What human users if any does the system need to accommodate?

Two categories:

1. Nontechnical person at computer with a program running where a person can type address and hit validate, and it will be validated
2. A tech-savvy person who will provide batches of addresses of arbitrary size and wants it processed in a batch

Access to data: any restrictions from nontechnical users?

Category 1 users will just type

cat 2: whoever uses will just feed files to the system

Do you want system to crosscheck data against a preexisting database for error?

No.

How will you use the application?

Cat 1: same as above

Cat 2: Flash drive, put in slot navigate for file name and it will give a report.

Exact specifications on the collection of addresses?

CSV file most likely.

What output?

Preserve original give new file.

Should the system be able to interact with google maps?

No.

What should the output file be?

May need exploration, ex. if you can’t validate something or you’re not sure.

Validation?

Given an address give a few different alternatives if something unsure happens.

What should be done with an incorrect address?

Reject questionable result/given to operator to resolve.

Give options at end of batch run of what to do with questionable things.

Print report of rejected addresses.

Critical Properties of the system?

The ability to normalize any correct address is most critical

Next is to recognize incorrect addresses.

If an error is undetectable then you can’t deal with it.

What platforms?

Office setting, any and all operating systems.

Correct Spelling errors?

Do correct safely correctable/detectable errors.

Maximum Batch size:

50K

Motivation for product?

Statewide system housed at atsu maintains info on physicians med students, medical programs, and it has a lot of bad address data in it. Connect home district and get funds.

What level of documentation?

No clue about code or documentation.

How often to be used?

Does not need to be online, but won’t be used very frequently.

Time Frame for development:

Last week of class.

No automated email.

What to do with duplicate entries?

Identify them and report. pre/post normalization.

side note: maybe use neural net to find error source

For output file how should the address be listed

Give the option of how it should be output ordered at the end.

Access to data?

No direct access, will be given sample though.

If terminated mid-batch, restarting from beginning would be safe.

Be as safe as possible.

Proficiency?

GUI/command line is not required but both is nice.

Special consideration for users with disability? not at this time

What language?

English, May need to recognize additional fields from puerto rico.

How is the address string broken up?

All on one line, no consistency in seperation.

User doesn’t need to stop in the middle of processing batch.

Security?

Not necessary.

Formatting?

Will be figured out later.

Evaluation?

By grade in regards to the specifications we develop.

Possibility of fields being mixed up?

No, assume the field order is correct.

Do you want a functional prototype?

It would be helpful.

Open source requirement?

That’s the preference

Bulk or singles?

first bulk most, then singles.

output where?

It should be configurable.

Configurable output?

give the option of a duel CSV

Space requirement?

none.

Usps standards configurable?

would be good not necessary.

What to do with non-address characters?

discretion.

Postal or business standards?

Should cover both.

Should we expect non-ascii characters?

He’ll get back on that.

No PO box fields.

ZIP 5 and 4 will be separated.

Errors can be expected on every step.

Don’t need to know if address exists.

Assume an internet connection.

Figure out if an address COULD exist.

Worry the most about Missouri address.

Zip 4 is not necessarily included.

(Steph’s notes)

**Answers**

1. Nontechnical (typing stuff into text boxes). Tech guy: have their own file of addresses
2. Cross-referencing data against other data bases for error checking: no
3. Nontech use case: sit down at computer, type in address into different buttons, hit run, get response
   1. Tech: flash drive, navigate for file name, select file, and get something back (report, result)
4. Maybe dealing with CSV file
5. Preserve original file and produce new output file (for sure batch)
   1. Should new file contain only corrected addresses or all addresses?
      1. Need to explore more in this area (developer to provide some ideas)
6. Should system interact with Google maps? No
7. During batch file, will tech person be at the computer the full time it’s taking place?
   1. Can errors be deferred to the operator and have operator choose option?
      1. Beck liked this (option a, option b, reject all)
8. Incorrect address, system “fixed” the address, but made it wrong
   1. Rather have questionable result rejected rather than making a guess and have it be wrong. (*Do no harm*). (Rather false negative, rather than false positive)
9. At the end, have questionable ones to go to interface and show output of what worked and what didn’t work
10. Critical properties: normalize any correct address (#1)
    1. #2 recognize incorrect addresses (so don’t accidently get accepted)
11. If we can’t detect error, we can’t react to it (like wrong street number)
12. System: no website, no mobile, just office setting at desktop
    1. No strong feeling about OS yet
13. Need to fix misspelling (if we can do it safely) (don’t “correct” something that’s already correct)
    1. Ideal: detect and correct
    2. Next best: detect but not correct all
14. How many addresses in a batch can we expect: 50,000
15. Motivation for product being developed:
    1. State wide system at ATSU and maintains info on physciaisn, med students, med programs
    2. Clean up terrible system
    3. Funded by state appropriations, and to get more state $ to connect to their legislature
16. What level of documentation for code?
    1. Not a clue
17. No kind of authentication needed for users
    1. Network access: operation is all local
    2. Post office has huge zip code list: (maybe once a month), download list of all zipcode files
    3. Doesn’t have to be online, but times where accessing external info might be good
18. Time frame for development of application? Last week of class
19. Automated email sent to user? No
20. Duplicate entries? Nice to know when there’s a duplication (put into output)
    1. Or after normalization to know if duplicate
21. If zip or city name is wrong?
    1. Might have both options with new zip and new city
    2. (For operator to chose)
    3. Might spell check (distance between this string and that string) and fix “closest one”
    4. Might be a pattern in errors and figure out errors by that?
22. Database: not direct access (FERPA locked down)
    1. Data: address, city, state, zip, zip (no names)
    2. Will be given sample addresses
23. Output file: doesn’t know what he wants yet
    1. Some way of identifying a line (preserve original order)
    2. If specific ordering make sense, have option to enable op to change the order
24. If system fails in the middle (like computer), restarting whole batch would be safest
    1. Important: don’t lose data
25. UI: GUI not required, but definite advantages to both ways (command/GUI)
    1. Like to have both
26. Users with disabilities: no considerations yet
27. Speed isn’t an issue, can run slow
28. Spanish st abbreviations possible
    1. , no extra fields that Puerto Rico has
    2. If for Puerto Rico, some strings might be stuffed into 1 field
29. Address string is 1 line, no consistency if things separated by , space, ;
30. Security measures:
    1. No, no authentication, no cryption, doesn’t need real-time access
    2. As long as it stand alone system, no system
31. CSD input file, output file in own separate boxes like input
    1. We’ll agree on common output
32. Single access at once
33. Grades: how well we adhere to specifications given
34. Zip-5, zip-4    (no fields not in wrong order), blindly assume that first field is address
    1. Safety checks
35. Does everything presented to the user be sketchy or things we can somewhat fix?
    1. Don’t know
36. Interface: good to have suggestion (go with suggestion or force your thought)
37. Depending on how sophisticated this gets, have confidence level?
    1. AI: might not be realistic
38. Working protype?
    1. If we thought it would be beneficial
39. Open source requirement or library we create?
    1. Preference open source, but not mandatory
40. Initially done in bulk and then later 1 at a time
41. Where should output be saved?
    1. Needs to be configurable (command or set-up option)
    2. Flash drive, file,
42. Code can be posted on our personal repo
43. Once we get normalized addresses, will there be an option to put them back in the database so we don’t have to renormalize them again?
    1. Can if you want
    2. We don’t need to worry about this, but another person’s problem (batch processes)
    3. 1 address at a time: type in 1 address, normalize it, paste into database
44. No space restriction
45. Normalized standard: USPS 2016: config file (yes)
46. Don’t store in RAM. All output goes to files before prog ends
47. Won’t need to add a batch to another batch
48. Won’t have to store 1 offs
    1. Input to output, unless we had canonical correct addresses (don’t store input)
49. Do business and postal ones
    1. Do we need to know if an address if business or postal one before we can calculate it? (Dr. Beck asked)
50. Just CSV files
51. Expecting non-alphabet things?
    1. Dr. Beck getting back to us (non-ASCII)
52. We can check if state and zip match (nail this down pretty well)
    1. City, state, zip (static thing and get once every 3 months)
    2. Can’t check if anything else exists
53. Computer has internet connection that we’re given
54. If things are impossible, don’t even go through the user
55. Post office boxes, air force academies (not necessary)
56. Can download zip, state match ups in file
57. Incomplete addresses can exist
58. Legislative addresses only matter in MO addresses
    1. Half of addresses are in MO
59. Zip-4 not necessarily included (but would be good to generate this)