Regular Expressions

Write code using the language of your choice to implement regular expressions to screen for the following:

1. Social Security Number (can be with dashes, whitespace, or no spaces at all)
2. US Phone number - parentheses are optional, as well as the dash between the last two sections
3. E-mail address
4. Name on a class roster, assuming zero or more middle initials - Last name, First name, MI
5. Date in MM-DD-YYYY format - separators can be -'s, /'s
6. House address - Street number, street name, abbreviation for road, street, boulevard or avenue (full version of those items should also be accepted)
7. City followed by state followed by zip as it should appear on a letter
8. Military time, including seconds
9. US Currency down to the penny (ex: $123,456,789.23)
10. URL, optionally including http:// or https://. (Links to an external site.) upper and lower case should be accepted
11. A password that contains at least 10 characters and includes at least one upper case character, one lower case character, one digit, one punctuation mark, and does not have more than 3 consecutive lower case characters
12. All words containing an odd number of alphabetic characters, ending in "ion"

Capture output that shows clearly what you used to test each case. Make sure you show that valid responses were accepted and invalid responses were rejected. You might want to collaborate with other group members or even other groups when it comes to testing!

Make sure you document any cases you could not get to work.

Place all items (code, input test file and output captures) in a zip file with your name. I will run your programs myself, so if there are any compilation instructions I need, please include them :-)

Place each regular expression in its own function/method. Provide documentation so that I can easily follow your code. Utilize good naming conventions, whitespace, etc.

Note that you should try and use regular expression(s) to solve the problem first and foremost, but it is okay to include a little helper code where necessary or it makes sense.

BE SURE AND DOCUMENT ANYTHING YOU COULD NOT GET TO WORK, OR ANY SHORTCOMINGS FOR A GIVEN TASK.

Provide a menu system that gives letter options of lower case **a** through **l** for each of the 12 expressions. Make sure **a** corresponds with expression 1 (SSN), etc. Also provide as a last menu option a lower case **q**, for quit. Input expected to the program should be a letter representing which menu option, followed by a single newline, followed by a string that will be tested by the chosen regular expression. To clarify, here is what your menu should look like:

a. `Social Security Number`
b. `US Phone number`
c. `E-mail address`
d. `Name on a class roster, assuming one or more middle initials - Last name, First name, MI`
e. `Date in MM-DD-YYYY format`
f. `House address - Street number, street name, abbreviation for road, street, boulevard or avenue`

g. City followed by state followed by zip as it should appear on a letter
h. Military time, including seconds
i. US Currency down to the penny (ex: $123,456,789.23)
j. URL, including http:// (upper and lower case should be accepted)
k. A password that contains at least 10 characters and includes at least one upper case character, lower case character, digit, punctuation mark, and does not have more than 3 consecutive lower case characters
l. All words containing an odd number of alphabetic characters, ending in "ion"

q. quit

With a properly designed menu, and by reading input as specified above, a program can be tested by redirecting a specially crafted input file that contains a menu option followed by the string. Thus, a sample input file to test the above might look like this:

```
a
123-45-6789
a
123456789
c
tcapaul@mail.ewu.edu
c
thomas.capaul@mail.ewu.edu
q
```

So, the program could be tested interactively, or by redirecting a file at the console. Make sure your program deals with both. You do *NOT* have to error check menu choices for this assignment (assume the user will enter a through l or q, followed by the input string to test).