This assignment was written in java with an associated SQLite backend database to store users names and phone number. The database was written with the UNIQUE identifier to ensure that there are no duplicate names, or phone numbers, to ensure a single delete statement doesn’t remove more than a single record at a time. Additionally, regular expressions were used to ensure the user input patterns matched what they’re intended to be. Names, need to look like names, with a max of one hyphenation, and one apostrophe. Phone numbers need to match any country’s pattern to be considered valid. This program has one external dependent library that needs to be included, that is included in the submission. Included in that library are the required drivers to connect to the associated SQLite database telephone.db. In order to compile this program, javac can be used to compile and include the new library, and the command ‘**java -cp .\bin inputvalidation.assignment2\_cxr4596**’ followed by the requested arguments will execute the program on windows.

Findbugs was ran against the submitted .java file to find any potential security violation, and it appears there remains one false positive. The first regular expression listed in the validateNumber function seems to be a potential for a denial of service attack. While Findbugs did record it, it doesn’t seem to be a problem as there are many other comparisons that will continue to compare as expected. Considering the validateName function has no Findbugs errors, it is safe to assume that the reported bug is a false positive.

The regular expressions are broken up into multiple expressions for the sake of individual simplicity. In the phone numbers, there are four individual regular expressions. Rex\_US matches with any typical US phone number, either (XXX)XXX-XXXX or with the +1 at the beginning. Rex\_English will match the format +XX (XX) XXX-XXXX. Rex\_groups will search for any words that match five numbers a period, then five numbers, as well as any number that is five numbers long. Finally, Rex\_US\_country will search for all US phone numbers that have the country code associated with it, or XXX X XXX XXX XXX. These patterns are all compared to the phone number that was read, if any of them pass then the phone number will be stored in the database.

To match a name, the strategy is a little different. This makes it possible to simplify the expressions down to two required. Before a regular expression is checked, the name input will be checked for word count, to confirm a maximum of three words were entered. Next, rex\_name will check for any two to three name words, and double check that there is at most one apostrophe and at most one hyphenation in either name. Lastly, rex\_single will check for the same rules as rex\_name, but only one word in length.