June 29, 2015

QR Code Generation

Resources – Table 1

|  |  |  |
| --- | --- | --- |
|  | **Resource** | **Document** |
|  | Project Request Form | Advanced Analytics and Informatics Support Project Request Form.pdf |
|  | QR Code Project at GitHub | <https://github.com/semantix/QRCode> |
|  | MSC (Medical Safety Code) | <http://safety-code.org> |
|  | MSC Application (New Location) | <https://gitlab.com/medication-safety/msc-server> |
|  | MSC Application (Current Location) | <https://code.google.com/p/genomic-cds/source/browse/#svn%2Fmedicine%20safety%20code%20server%253Fstate%253Dclosed> |
|  | Discussion Images and this document | <https://github.com/semantix/QRCode/tree/master/docs> |
|  | Compression and Encoding Reports (Analysis) | <https://github.com/semantix/QRCode/tree/master/src/main/reports> |
|  | Generated Files with compressed and encoded files | <https://github.com/semantix/QRCode/tree/master/src/main/resources> |
|  | Zebra Crossing | <https://github.com/zxing/zxing> |

**Solution Development Plan:**

The goal of this project is to augment an existing Medical Safety Code web application to handle data from PGRN Seq2 (NSG) platform.

The plan is to follow the workflow discussed in Figure 1.

1. NSG data comes in a VCF File. (Step 1 in Figure)
2. This VCF data is processed using VCF Tools library to filter out data based on the context. (Validate and Filter) – Step 2,3.
3. Since the selected VCFIDs can be thousands, there is a need to shorten the service URL that gets formed and sent to QR Code generation service. The Selected Variants will be encoded to have relatively smaller IDs (preferably 2-3 character IDs as compared to 8-10 character VCF IDs read from VCF file). There is a Database (internal to MSC or external) that will translate the encoding of incoming VCF IDs to smaller length IDs. (Step 4)
4. Since these smaller IDs can still exceed the maximum number of characters that are allowed in an HTTP GET Request (to the QR Code generation service). There is a need to compress and possibly encode the resultant long string. This URL is supposed to sent to the QR Service and to get a QRCode String (which can be used to generate image) later. (Step 5, 6)

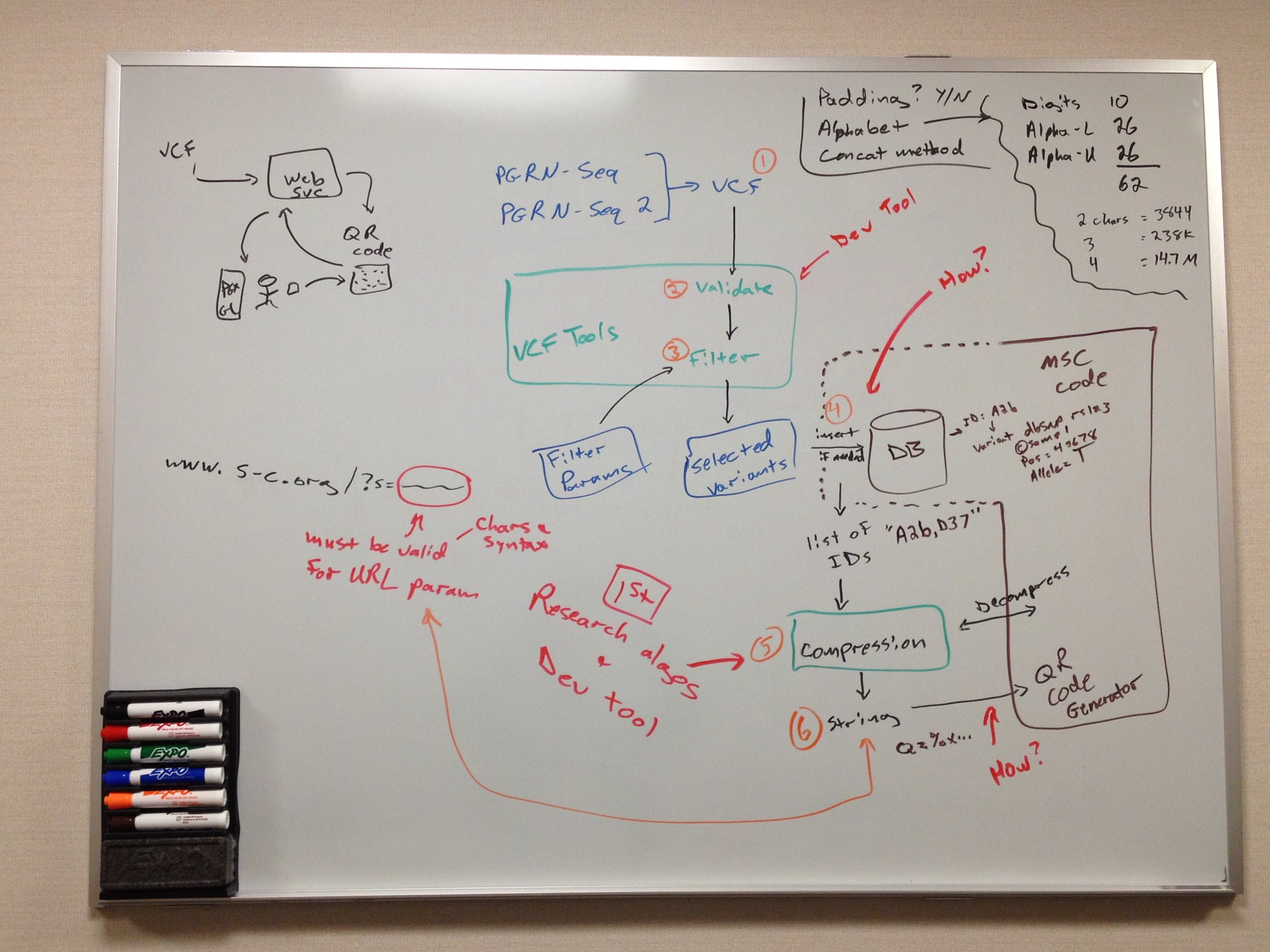


Figure : Workflow

Compression & Encoding Tests

The Github Project QR Code (Resource 2 in the table 1), is a java program to test out compression and encoding to see how many encoded (smaller tokens) VCF IDs (generated by step 4 in Figure 1) can fit in an acceptable URL.

Following Compression algorithms tested:

* GZIP
* Zip
* LZ4

Encoding with Base64 was applied and then URL encoding to make sure all characters are acceptable for a valid URL string.

Please See Figure 2 for Compression and encoding implementation details.

Parameters to the test program (QRCode Project) are:

1. Character Sets – Numbers, Alphanumeric, Alphanumeric with URL acceptable characters, and character set of Alphanumeric with all special characters.
2. Length of each ID – variables of 2,3,4 and 5 characters.
3. How many IDs to try – 1000, 2000, 5000, 10000, 50000.

Please see resources 7,8 in table 1 to review and compare results of QR Code Project.

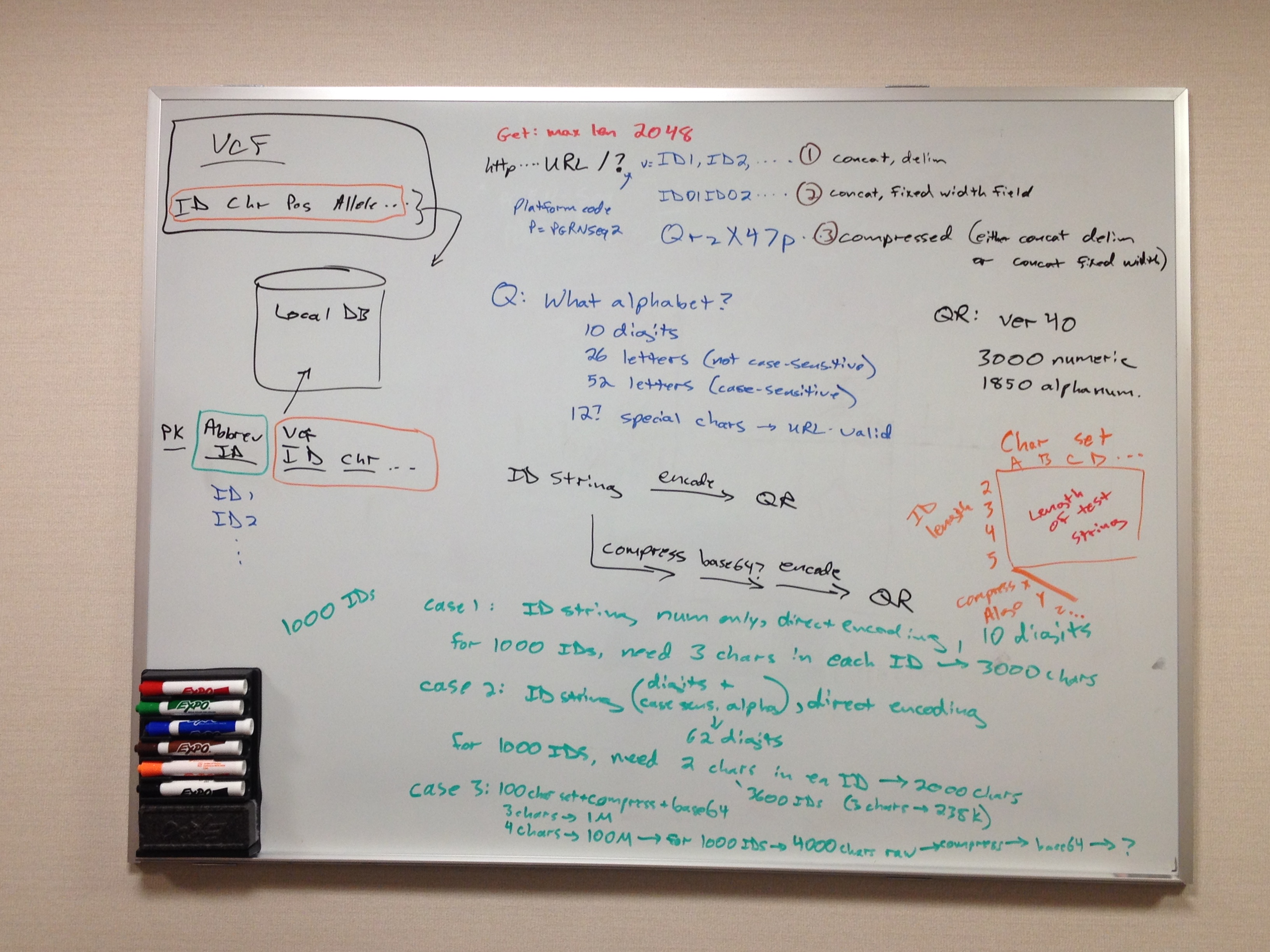


Figure : Compression & Encoding of Query String

Conclusion and Final planning (June 2015)

After meeting with MSC Application development group members and investigator, we received more information about the code base, and the actual tool that they use for QR Code Generation (Please see Resources 3,4,5 in table 1).

After analyzing the code and further discussion – Following conclusions were made:

1. MSC Application accepts QR Code Generation request as a HTTP GET, which has a limit of 2 KB length of HTTP Query string. One option is to send larger set of VCF IDs if we use HTTP POST instead of HTTP GET. But the MSC Application has not implemented a POST request processing. One option is to provide such implementation, which will definitely allow us to post lot more data to get a QR Code, generated. This option may reduce the need of harshest compression and/or encoding of the VCF IDs. Further experiments and test codes are required to see the right balance.
2. MSC Application server uses a popular QR Code Library Zebra Crossing (See Resource 9 in table 1). This library allows multiple ways to send a request and get an image generated. MSC Application server uses the URL request way to post the data and get an image generated. But there are other ways where a big file of VCD ID can be used to generate a QR Code/Image.
3. The approach discussed in step 2 above might have some limitation as we try to generate a QR Code Image 40 with H Level Error Correction, which limits how many VCF IDs could be encoded even if there is no restrictions on making data available to the QR Code library.

Following Java Classes in MSC Server and Zebra Crossing are the first few classes to get started with experimenting and start implementing a solution:

Zebra Crossing:

* <https://github.com/zxing/zxing/blob/master/javase/src/main/java/com/google/zxing/client/j2se/CommandLineRunner.java>
* <https://github.com/zxing/zxing/blob/master/core/src/main/java/com/google/zxing/qrcode/QRCodeWriter.java>

MSC Server

* <https://gitlab.com/medication-safety/msc-server/blob/master/src/servlets/MSCImageGenerator.java>
* <https://gitlab.com/medication-safety/msc-server/blob/master/src/servlets/SafetyCodeGenerator.java>

The following is the trace from a discussion with MSC Server development team and PI (Early June 2015):

[12:35:10 PM] robertrfreimuth: I have a link to the google code project, but thought it was being moved to git.  I couldn't find a new URL in my notes, though.  Could you please confirm where the latest codebase is located?

[12:35:35 PM] Matthias Samwald: sure, we moved to <https://gitlab.com/groups/medication-safety>

[12:37:09 PM] Matthias Samwald: we are currently doing most work at <https://gitlab.com/medication-safety/msc-server>

[12:37:29 PM] robertrfreimuth: Is that under the same project, or a different one?

[12:37:35 PM] Matthias Samwald: but it is work in progress (re-implementing), and still lacks major functionality.

[12:37:45 PM] Matthias Samwald: same project

[12:37:58 PM] robertrfreimuth: We want to look more closely at the QR code generator.  I assume you used an existing library for that?

[12:38:17 PM] Matthias Samwald: the latest fully functional version (v2) can be most easily downloaded from the old SVN repository i would say.

[12:38:30 PM] Matthias Samwald: ah i see

[12:39:09 PM] Matthias Samwald: that can be most easily be found here: <https://code.google.com/p/genomic-cds/source/browse/#svn%2Fmedicine%20safety%20code%20server%2Ftrunk%2FGenomic-CDS%2Fsrc%2Fservlets> (old repository)

[12:40:23 PM] Matthias Samwald: we use the <https://github.com/zxing> library for QR code generation

[12:40:36 PM] robertrfreimuth: Perfect - thanks for the direct reference