Milestone 3

Design Decisions:

Supported Constructs

We support the "foreach" operation over the "patient" constructor- we can do filtering for population (population is id:5)and can generate code for the computation block (foreach Patient p{...}). Right now, adding another filter in the filter list apart from population and Patient will result in an error. Currently, we only allow the printing of a patient's fields (print p), as determined by the user determined filter. We can also print specific fields of the patient filter (print id, postalcode of p).

We have decided to generate JavaScript. For each .onc file, we generate a .js file, generate SQL queries and execute using the excellent npm MySQL package. Note that we are not using the toy database - instead, we are executing queries directly against the MySQL server running on **hig.cs.mcgill.ca**. To achieve this, we open an SSH tunnel, map the port 3306 (which is what MySQL on the remote server is listening on) to a local port, and then execute queries against that port. The instructions for establishing the SSH tunnel are given in the README. Each SQL query is provided directly as the argument to a JS function, and within the function body we perform various computations.

Additionally, we also provided an auxiliary config file, *database.conf*, to allow a user to map filters and fields allowed by the OncoTime language (which users also define in their config file) to column names in their particular database. For example, the field "id" within patient can be mapped to the key "PatientSerNum" within the database. The reasoning behind this auxiliary file is the same behind *config.conf*: We want to make our OncoTime compiler, which we have taken to calling The Dragon OncoTime Compiler (The DOC for short), as adaptable as possible.