22 novembre 2015

Louis Bompart et Pierre Juhel

ESILV A4-ISN

User client for Ottawa Construction Permits Database

Software Requirements Specifications

# Purpose and goals

The purpose of this software is to provide user-friendly access to a permit construct database. It must, at least:

* Provide an import module.
* Provide an easy and straightforward way to do requests.
  + With specific requests (AON answers)
  + With textual requests (score-based answers)
* Answer the requests of the user with reliable and solid answer.
* Display the relevant data in a comprehensive view.
* Display statistics of the database.

It may:

* Give the reliability of the request and warn the user if it’s too low.
* Provide a user-friendly way to Insert/delete/update Data
* Provide a way to research data with coordinate (e.g.: Permits delivered in a 20km area).
* Display the geographical data on a map
* Display graphics and charts.

# System Architecture

The software solution is developed on Java 8 SE Update 60.

Is used with this software (needed for the software execution):

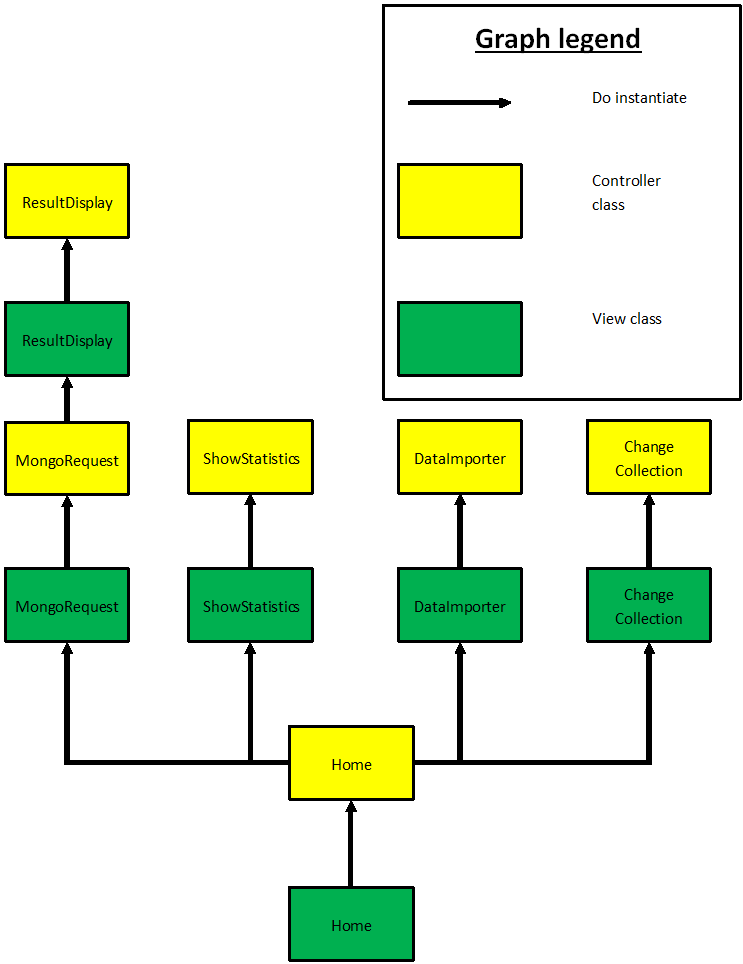
* MongoDB 3.0.7 (Windows 64-bit 2008 R version during development)
* ElasticSearch 2.0.0

The API used are:

* Junit: for testing (version 4.12, see pom.xml)
* MongoDB driver (version 3.1.1, see pom.xml)
* ElasticSearch Java Integration (version 2.0.0, see pom.xml)
* GMapsFX : for Google Maps support (version 1.1.1, see pom.xml)

# Data Model

# Pseudo-Class Diagram



# Queries.

## Custom queries, depending of user choice on MongoRequest

## Map-Reduce request, for ShowStatistics:

function() {

emit(this.permits.MUNICIPALITY, 1);

}  
function(key,values) {

return Array.sum(values)

}

# Implementation issues

Due to an awful JSON file composition we encountered severe issues with Elasticsearch implementation. Moreover, River are deprecated so it’s was really too hard to implement a junction between MongoDB & Elasticsearch. However, in an attempt, we worked on Logstash and correct and debug an open-source plugin for it (see : <https://github.com/phutchins/logstash-input-mongodb/pull/23> ). At least, we gave up on the junction and import data separately.

The MongoDB Java API is really user-friendly and we do not encounter many difficulties with it.

The Elasticsearch Java API is however more difficult to use.

# COnclusions

Thanks to this project we managed to successfully complete a real software solution, using NoSQL databases for the back-end and JavaFX for the front-end. In short, we’ve done more than summarize our semester of NoSQL in one single project, we used our new knowledge acquired in Software Development Process and Java EE & frameworks courses to do one and somehow “big” project