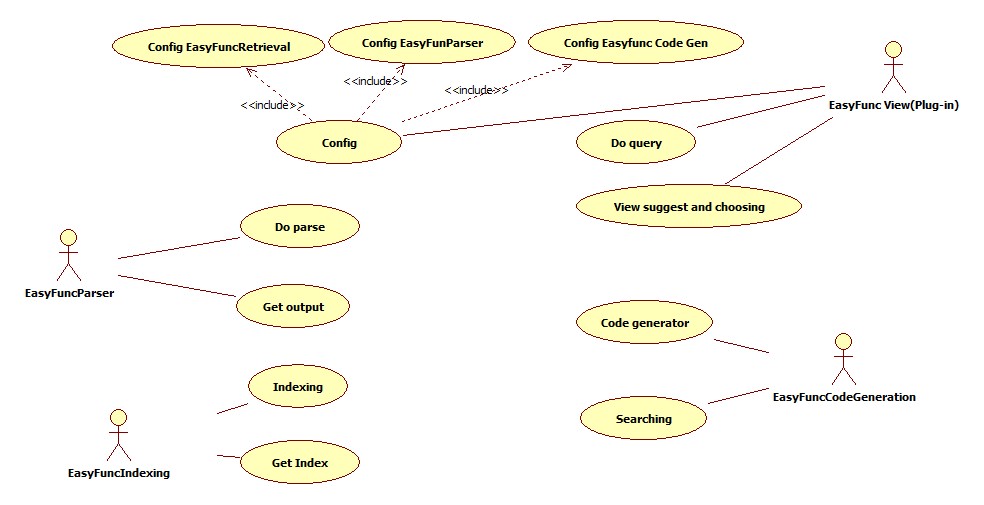
Báo cáo tuần 02/09 -09/09/2013

# Công việc đã làm được

* Mô hình hóa hệ thống
* Tìm hiểu về các khái niệm trong Lucene
* Đánh index cho tập dữ liệu Java Standard

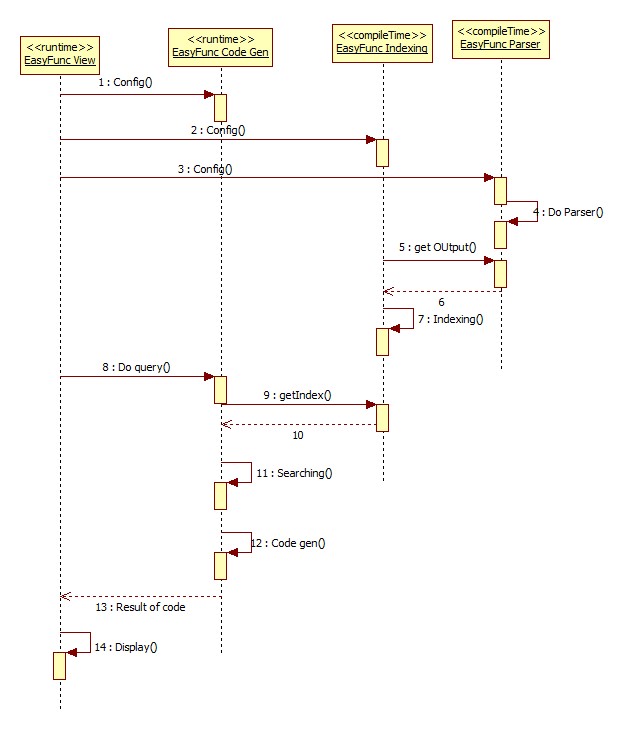
# Mô hình hóa hệ thống

## Use-case



|  |  |  |
| --- | --- | --- |
| Module | Use-case | Mô tả |
| View | Config | Cho phép người dùng cấu hình các module còn lại của hệ thống |
| Do query | Tạo môi trường để người dùng xây dựng câu truy vấn |
| View suggest and choosing | Tạo môi trường để người dùng có thể xem được kết quả của câu truy vấn và lựa chọn kết quả phù hợp với họ |
| Code generation | Searching | Tìm kiếm kết quả từ tập dữ liệu(Sử dụng Lucene) |
| Code generation | Từ kết quả của tìm kiếm sinh ra mã và gửi kết quả này tới module view |
| Indexing | Indexing | Đánh index cho tập dữ liệu(Sử dụng Lucene) |
| Get index | Hỗ trợ lấy thông tin về index( Được sử dụng trong module Searching) |
| Parser | Do parse | Phân tích tài liệu API |
| Get output | Hỗ trợ lấy thông tin về kết quả sau phân tích |

## Lược đồ tuần tự



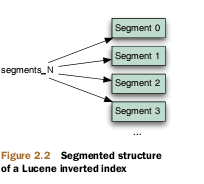
# Module Indexing:

## Các khái niệm trong Apache Lucene

## Các thuật ngữ

### Indexing

* IndexWriter: Nơi xử lý index tài liệu
* Directory: Nơi lưu trữ tài liệu sau khi được index
* Document: Tài liệu dùng để đánh index
* Field: Đơn vị của tài liệu
* Token: Một từ hay cụm từ được analyzer trước khi được index
* Inverted index:

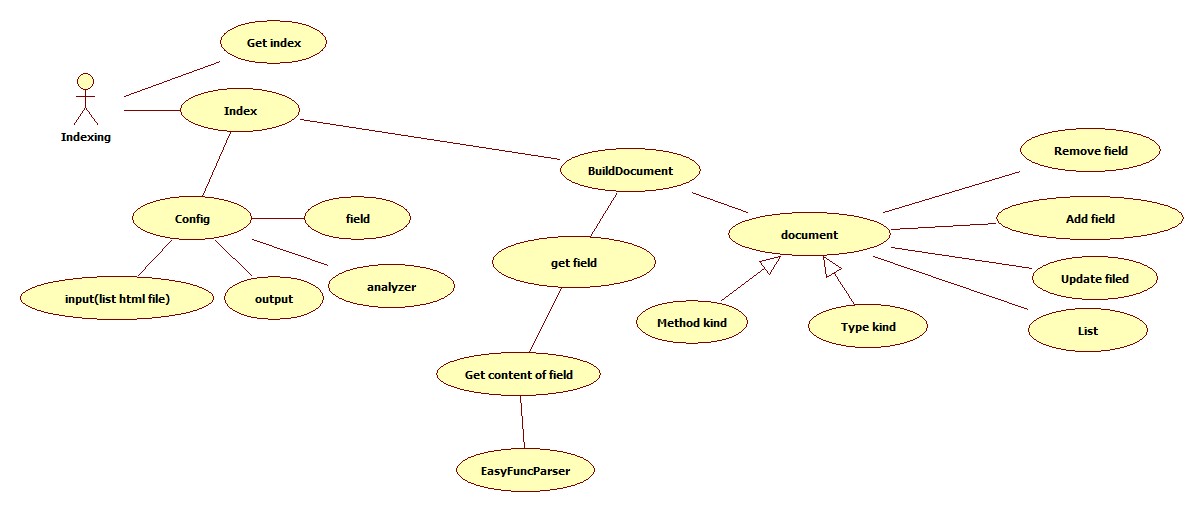


Với cách cấu trúc index thông thường, chúng ta sẽ quản lý các term dựa vào tài liệu. Ví dụ như ta sẽ nói tài liệu A có nội dụng gồm 3 term: “football”, “sport”, “player”.Như vậy, khi có yêu cầu tìm kiếm về term “football” chúng ta phải quét toàn bộ trang tài liệu để tìm vị trí của term “football”. Điều này sẽ khiến cho việc tìm kiếm của chúng ta không thực sự tốt.

Inverted index là cách mà chúng ta tạo ra một bảng vị trí của các term –tương tự như phần mục lục của một cuốn sách. Theo đó, chúng ta sẽ lập ra một bảng các term đi kèm với vị trí xuât hiện của nó trên toàn bộ tập tài liệu. Việc này có một số lợi ích như sau:

* Do độ dài của 1 term ngắn hơn độ dài của một tài liệu nên việc tìm kiếm trên toàn bộ term sẽ nhanh hơn tìm kiếm trên toàn bộ tài liệu
* Mỗi term luôn có một định danh duy nhất cho nên chúng ta có được kết quả tìm kiếm ngay trong lần hit đâu tiên( Best case: 1, Worst case: n). Với tài liệu chúng ta luôn phải tìm kiếm toàn bộ
* Field Option
  + Index: Nội dung của field sẽ được analyzer trước khi được index
  + Stored: Nội dung của field sẽ giữ nguyên(Không bị analyzer) và lưu xuống bộ nhớ
* TermVector:Dạng trung gian giữa index và stored. Nội dung của filed sẽ được analyzer và lưu xuống bộ nhớ

## Hiện thực



# Công việc tuần tới

-Thực hiện phần searching đơn giản( chưa bao gồm các thuật toán)

-Tích hợp plug-in vào hệ thống

-Chạy thử và demo