|  |
| --- |
| CSE 2341 : Data structures |
| Semester Project |
| Design Documentation |
|  |
| **Ethan Busbee**  **Courtney Redlinger**  **Paul Renten** |
| **November 4, 2012** |

|  |
| --- |
| The following document contains team information for this 3-person group for the Let’s Search semester project. It also contains the high-level design overview for this project. |

Table of Contents

[Team Information 2](#_Toc339824552)

[EPC UML 3](#_Toc339824553)

[EPC UML LEGEND 3](#_Toc339824554)

[UML Screen Shots 4](#_Toc339824555)

[Class Descriptions 7](#_Toc339824556)

[Preliminary “Stress Test” Mode Commands 9](#_Toc339824557)

[Maintenance Mode 9](#_Toc339824558)

[Interactive Mode 9](#_Toc339824559)

[Preliminary Time Line 10](#_Toc339824560)

[Miscellaneous Design Items 11](#_Toc339824561)

# Team Information

Ethan Busbee

27988211

[ebusbee@smu.edu](mailto:ebusbee@smu.edu)

Lab Section: Friday 2 – 3:50 pm

Courtney Redlinger

24072156

[credlinger@smu.edu](mailto:credlinger@smu.edu)

Lab Section: Friday 2 – 3:50 pm

Paul Renten

26984962

[prenten@smu.edu](mailto:prenten@smu.edu)

Lab Section: Friday 2 – 3:50 pm

# EPC UML

## EPC UML LEGEND

- - - - |> Composition Relationship

--------|> Arrow representing flow of operations

+ Public attribute/method

- private attribute/method

Blue Method Contsructor

Underlined Method Static Scope

Italic Heading Has abstract methods

: : delimits name of variable from type (Name : type)

\* Variable is a pointer

**Note:** Some variables, like vector and map have no element type specified. This is because it will be determined at run time or the UML software doesn't support the syntax.

So if you see map(int,(int,bool)) this should be read as map<int,<int,bool>>

If you see vector it should be interpreted as vector<element> etc… etc…

**Other Notes:**

The software I used to make the UML is Astah UML.

We have included the astah uml file if you choose to download Astah UML and view it from the software's perspective.

## UML Screen Shots

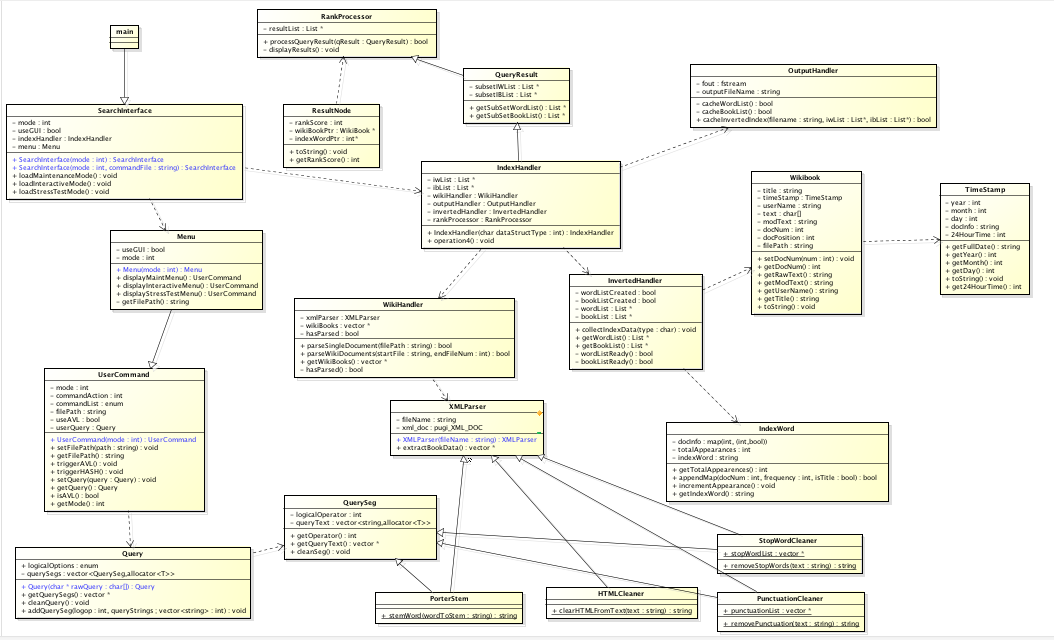


Figure 1: EPC UML Whole View

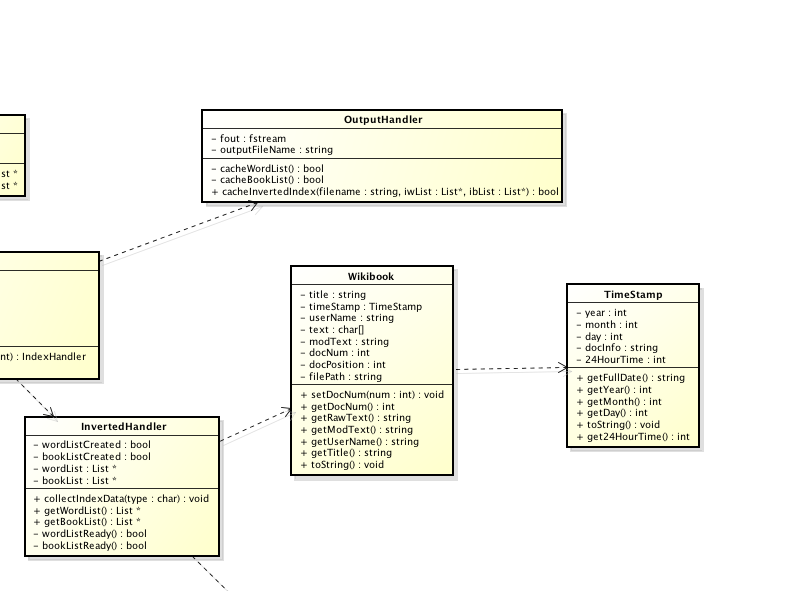


Figure 2: EPC UML Upper Right View

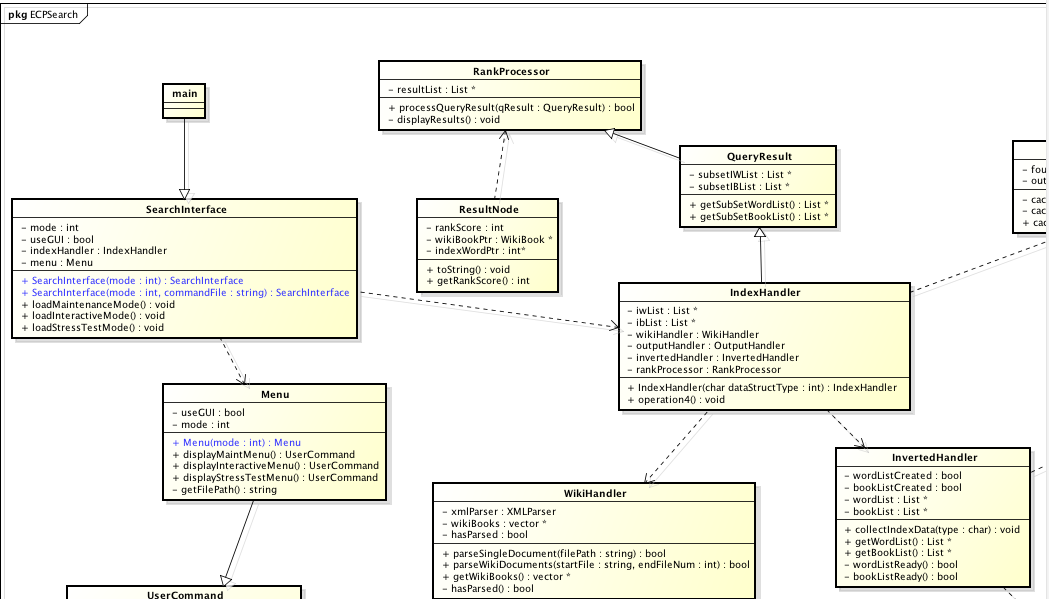


Figure 3: EPC UML Upper Left View

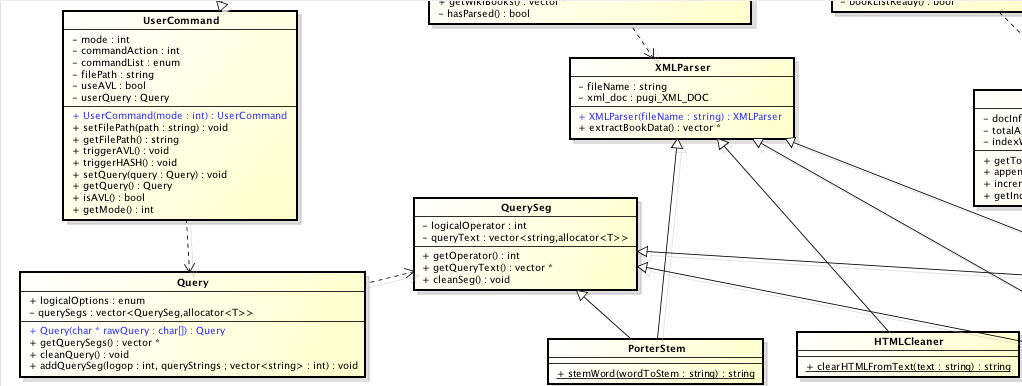


Figure 4: EPC UML Lower Left View

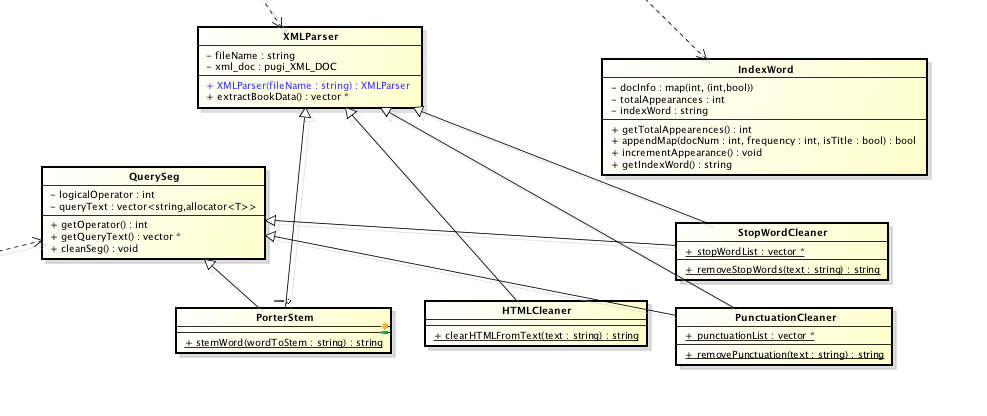


Figure 5: EPC UML Lower Right View

# Class Descriptions

SearchInterface: P

SearchInterface is the UI wrapper and overall handler for the entire program. It contains a Menu and an IndexHandler.

Menu: E

Menu contains functionality for inputting information from the user, processing that input for meaning, and then acting on it.

UserCommand: C

UserCommand reprsents a request from the user, including a search Query (if applicable).

Query: P

Query is a search query from the user to the system. It is primarily built from a vector of QuerySegs.

QuerySeg: E

QuerySeg is one segment of a query, ie, "AND cat dog" or "NOT catherine".

IndexHandler: C

IndexHandler is the root handler for the actual data indexed by the search engine, contains the index itself's handler (in whatever form that may be), and also processes Query objects, returning results.

QueryResult: P

QueryResult is the list of Words and Wikibooks that form the bulk, unsorted results of a user query.

RankProcessor: E

Processes a QueryResult into a sorted format and displays said QueryResult.

ResultNode: C

ResultNode contains a ranking, the word being searched for (reference), and the page that has been found via search (reference). It is the base unit of a QueryResult.

WikiHandler: P

WikiHandler contains the XMLParser and the "raw data" of the input from Wikibooks. It parses through the Wikibooks, passing data to XMLParser as needed.

XMLParser: E

XMLParser parses XML, extracting the appropriate data and pushing them back to WikiHandler.

InvertedHandler: C

InvertedHandler is the handler for the inverted index. It maintains Lists of IndexWord and Wikibook objects.

Wikibook: P

Wikibook represents one page of the corpus. It contains various data used to generate the inverted index and query results.

TimeStamp: E

TimeStamp stores a time/date stamp. It is used as a component of Wikibook.

IndexWord: C

IndexWord represents one word in the index, plus references to those documents it appears in. IndexWord objects are used to perform searches.

OutputHandler: P

OutputHandler is the "close" of the entire program. It handles writing the inverted file index to a file.

Static Classes:

StopWordCleaner: E

StopWordCleaner removes stopwords from a given text.

PunctuationCleaner: C

PunctuationCleaner removes punctuation from a given text.

HTMLCleaner: P

HTMLCleaner removes tags from a given text.

PorterStem: E

PorterStem stems a word to a base format.

# Preliminary “Stress Test” Mode Commands

## Maintenance Mode

AII (Append Inverted Index)

-accepts path to file as parameter

-appends info from the given file to the current saved index

-example command call:

AII wikibooks/WikiDumpPart1

CII (Clear Inverted Index)

-accepts no parameters

-clears the entire saved index

-example command call:

CII

## Interactive Mode

SII (Set Inverted Index)

-accepts a character as a parameter (either ‘a’ or ‘h’)

-loads index into an AVL tree or hash map data structure based on input

-**MUST BE CALLED BEFORE ANY QUERY COMMAND MAY BE USED**

-example command call:

SII a

SSQ (Submit Search Query)

-accepts a line of text as a parameter

-line of text consists of a properly formatted search query (using the same logical operators as the true “Interactive” Mode

-searches through the index based on the query

-example command call:

SSQ AND cat dog NOT hamster USERNAME rainbowmech

# Preliminary Time Line

Sunday, November 04, 2012: Finished high-level design

Friday, November 9, 2012 – Sunday, November 11, 2012: Concept-proofs, Finished headers, Finished utility classes, Finished query processing, Data structures built for index, Populate index

Thursday, November 15, 2012: Finished all coding, Create documentation for project

# Miscellaneous Design Items

The following are pictures of whiteboards we filled with notes for designing.

