XMLParser

- XMLFileParser
//parses XML for author, date, pageID, BodyTest,Title
- IndexHandler*
- XMLFileParser
- XMLDumpFile:char*
- xml_document<>
-xml_node<>*
- id: int
- pageMin: vector<int>
- pageMax: vector<int>

//Constructor destructor
XMLParser()
~XMLParser()

//public interface
+ loadStopTable(): void
+ loadPageRange():void
+ storeOffXMLData(int):void
+ setXMLDumpFile(string&): void
+ findUserWords():void
+storeOffNewData(string&):void
+ clearIndex():void
+buildIndexFromMemory(int):void
+storeOffIndex(char*)
+storeOffPageRange()

//functions utilitze by query machine:
+ getAuthor(): string
+ getText(): string
+ getDate(): string
+ getTitle(): string
+ navigateToPage(int): bool
+ createIndex(int): void
- binarySearch(size_t, size_t, int)

StressMode

- maint: MaintMode
- interact: InteractiveMode
- myParser: XMLParser
- structureChoice: int
- stressTestOutput: char*
- start: time_point
- end: time_point

//contstructor destructor
+ StressMode()
+ ~StressMode()
+ docFileParser(string&)

+ docFileParser(string& fileName);
+ stressUI(): void
+ startTime(): void
+ stopTime(): void
+ printTime(): void

MaintenanceMode

- myXMLParser: XMLParser

+ MaintMode()
+~MaintMode()
- addNewFile(string& fileName): void
- clearIndex(): void
- getUserCommand(): void

Has a...

Has a

InteractiveMode

IMHandler: IndexHandler*
myXMLParser: XMLParser*
finalPages: vector<int>
finalTitles: vector<string>
finalTF: vector<double>
totalPages: vector<int>
totalTitles: vector<string>
totalTF: vector<double>

//other filters
currAuthor:; string
currMaxDate: string
currMinDate:string

//Constructor Destructor
+ InteractiveMode()
+ ~InteractiveMode()

//public interface
+ setInputFileForIndex(char*): void
+ processQuery(string): bool
+ interactiveUI(): void
+ createIndex(int): void
+ setTotalDocs(int): void
+ clearAllCurrAndFinalMembers(): void
+ displayText(int choice): void
+ clearIndex(): void
+ reloadPageRange(): void
+ buildIndexFromMemory(): void

//private helper functions
- orPages(vector<string>&, vector<int>*&, vector<double>*&): void
- andPages(vector<string>&, vector<int>*&, vector<double>*&): void
- notPages(vector<string>&, vector<int>*&, vector<double>*&):void
- ltFunction(vector<string>&): void
- usernameFunction(vector<string>&):void
- gtFunction(vector<string>&): void

//for checking lists for page numbers
- containsPageStacked(vector<int>* myList,int page): int
- totalContainsPageStacked(vector<int>* passedPages,int page): int
- totalContainsPage(vector<int>* passedPages,int page): int

//helper functions for interactive UI
- displayPagesToUser(); void
- pickTop15(); void
- insertionSortStacked(vector<int>* passedVec, int start, int end): void
- insertion_sort(vector<int>& passedPages, vector<double>& passedTFs, int start,int end); void