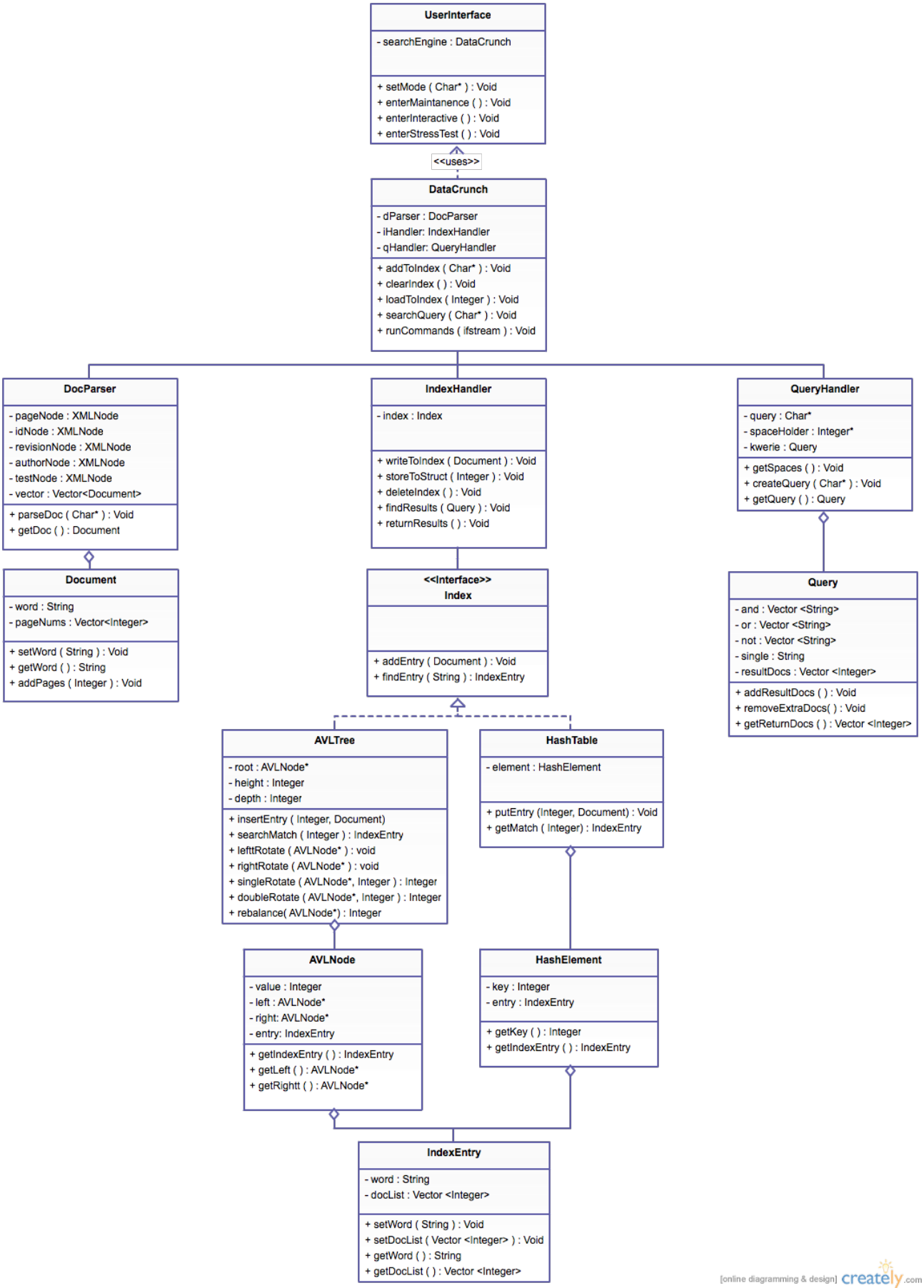
Team Zombie Coders

Members: Aurora Havens and Emely Villeda-Principe



Team Zombie Coders

Members: Aurora Havens and Emely Villeda-Principe

Class Roles and Responsibilities

- Class UserInterface **[Owner: Emely]**
  - Acts as the connector from main to DataCrunch in order to start the program
- Class DataCrunch **[Owner: Aurora]**
  - Accesses the 3 main processes that run the search engine (DocParser, IndexHandler, and QueryHandler)
  - Handles functions required for completing user requests
- Class DocParser **[Owner: Aurora (Including Document)]**
  - Takes in XML Documents to parse data to store relevant information
  - Responsible for retrieving specific data from the file and creating document objects with strings and pages numbers in side of them
- Class Document
  - Creates an object that will be stored out into the index so we can access the string and the documents that string is located in
- Class IndexHandler **[Owner: Emely (Including Index and its implementation classes)]**
  - Handles writing an index onto disk as well as storing the index data into an accessible data structure
  - Responsible for retrieving entries from the index and returning them to the user
- Class Index
  - Serves as the interface of the data structure that will hold the index data
  - Composed of index entries and the structure is dependent on the user
- Class IndexEntry
  - Serves as an entry containing the word and list of documents that include the term
  - For storing in a data structure that would allow for easy searching upon user request
- Class AVLTree
  - Serves as a potential data structure for storing index entries
  - Composed of nodes that are accessible to the user
- Class AVLNode
  - Node for the AVL Tree data structure
  - Holds IndexEntry object containing a string and a vector of document numbers
- Class HashTable
  - Serves as a potential data structure for storing index entries
  - Composed of HashElements that are accessible to the user
- Class HashElement
  - Element for the Hash Table data structure
  - Holds IndexEntry object containing a string and a vector of document numbers
- Class QueryHandler **[Owner: Aurora (Including Query)]**
  - Parses through query requests to determine what the IndexHandler must find
  - Responsible for interpreting the parts of a query (terms and Boolean operators)
- Class Query
  - Stores the parts of a query (terms associated with each Boolean operator)
  - Necessary for storing relevant documents once the IndexHandler finds matches

Team Zombie Coders

Members: Aurora Havens and Emely Villeda-Principe

Initial Commands

1. mmode // enter maintenance mode
2. add // add document(s) to index
3. <path-to-file> // supplied path
4. return // return to mode selection
5. im // enter interactive mode
6. load // load index to a data structure
7. avl // use an avl tree
8. search // request a query
9. <query-string> // supplied query
10. return
11. mm
12. clear // clear index file and data structure
13. add
14. <path-to-file>
15. return
16. im
17. load
18. hash // use a hash table
19. search
20. <query-string>
21. exit // exit program