GEBZE TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CSE 343 – SOFTWARE ENGINEERING
ONLINE SEARCH ENGINE
Version 1.00

GROUP – 1

FURKAN MUSTAFA AKSOY

EMRULLAH GENÇOĞLU

DENİZ BABAT

MUHAMMED CANER BAKAR

BEKİRCAN AĞAOĞLU

EMRAH KORKMAZ

RIDVAN PORTAKAL

INSTRUCTOR
ASSISTANT PROFESSOR URAZ CENGİZ TÜRKER

14 NOVEMBER 2016
GEBZE / KOCAELİ

~~*MODULES*~~

DATABASE:

Rıdvan PORTAKAL
Emrullah GENÇOĞLU
Deniz BABAT

SOFTWARE:

Caner BAKAR
Bekircan AĞAOĞLU
Emrah KORKMAZ

INTERFACE:

Furkan Mustafa AKSOY
Bekircan AĞAOĞLU
Emrah KORKMAZ

Table Of Content

Design and Implementation

The stage in the software engineering process at which an executable software system is developed. Design and implementation activities are invariably interleaved.

Design

Software design is a is a creative activity in which you identify software components and their relationships, based on a customer's requirements.

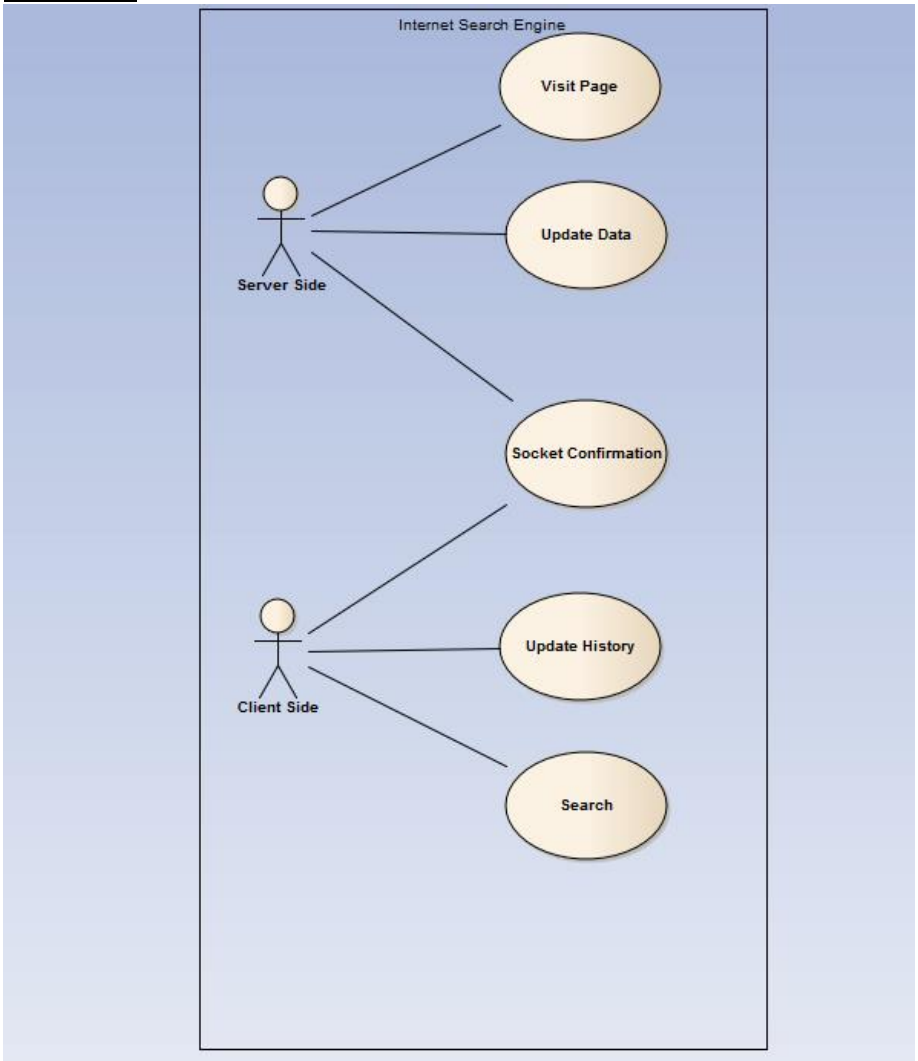
Architectural Design

There is two main module in system, server and client.

- Client sends query to server and receive results from server.
- Server receives query from server, search query in tree and send results to client.

Interface Design

Use Cases



Component Design

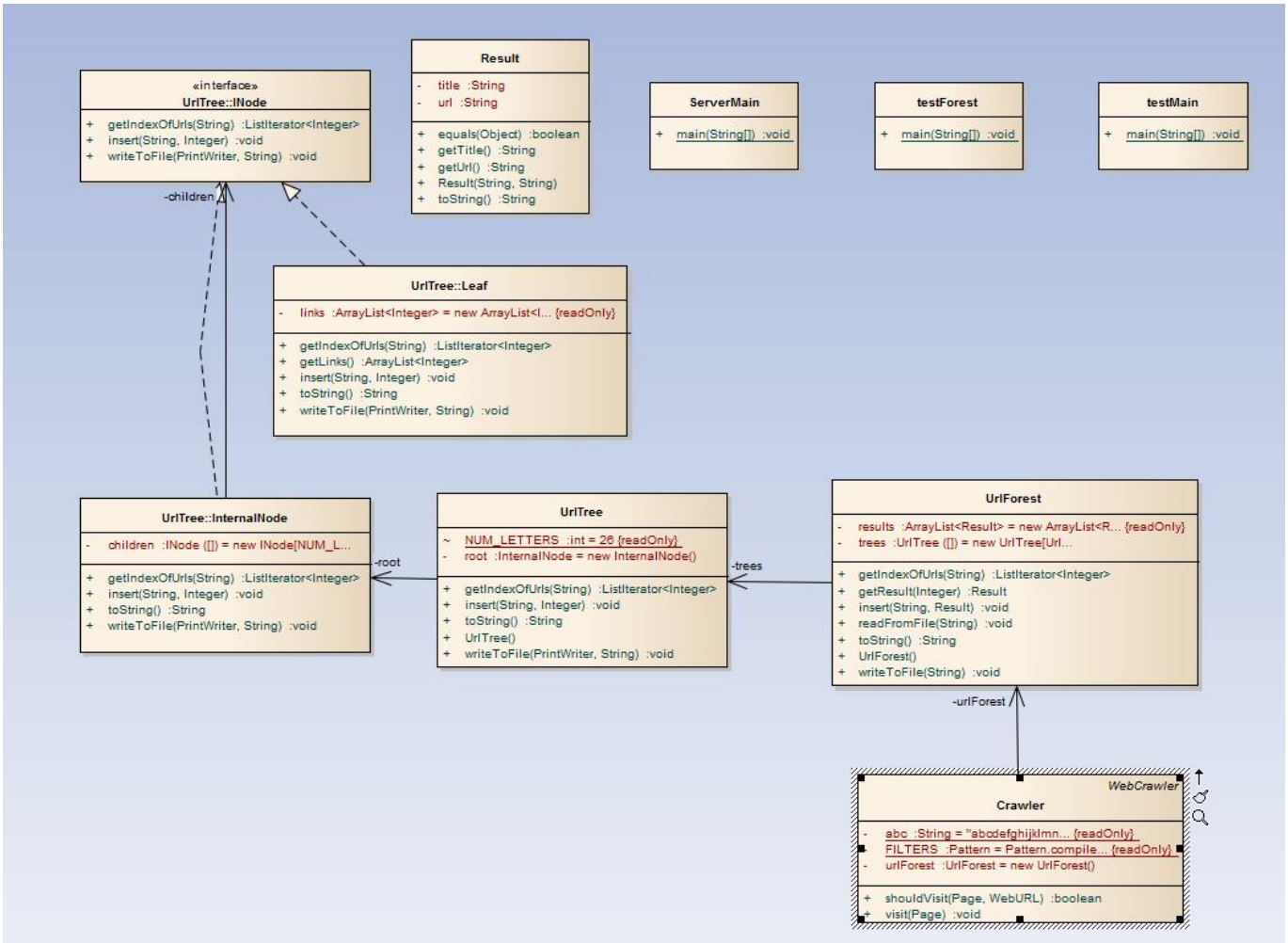
Server has two mode.

- At one, crawls the internet and put results to the tree until reach the limit. After that, it stops crawling and write tree to a text file.
- At other one, restores tree from text file and wait for connections.

Database Design

Data kept in a tree which is writable/readable from text file. Implemented tree that is 3 levels were created Internet Source Engine Project to Database.

Server UML



Interface UML

Implementation

Implementation is the process of realizing the design as a program. Translate software structure into an executable program. The activities of design and implementation are closely related and may be inter-leaved.

Server

Class Crawler

public boolean shouldVisit(Page referringPage, WebURL url)

Checks if url should be crawled or not.

public void visit(Page page)

Crawls the parameter page.

Class Result

Has 2 String data field, url and title.

public Result(String title, String url)

Constructor

public String getUrl()

Returns url as String

public String getTitle()

Returns url's title as String

Class ServerMain

- Creates a socket and accept connections from client.
- Receives <query> from client.
- Send results to client.

Data Structure

There is UrlForest class which contains Result arraylist and UrlTree classes. Url's kept in this arraylist, nodes have index of that url, so there is no duplication.

Class Result

Keeps two String data, url and title.

public String getUrl()
    Returns url

public String getTitle()
    Returns title

Class UrlTree

public void insert(String query, Integer index)
    Calls root node's insert method with query(1:), index.

public ListIterator<Integer> getIndexOfUrls(String query)
    Returns root node's getIndexOfUrls(query[1:]) method.

public void writeToFile(PrintWriter printWriter, String prefix)
    Calls root node's writeToFile method.

Interface INode

Interface for tree nodes.

public void insert(String query, Integer index)
public ListIterator<Integer> getIndexOfUrls(String query)
public void writeToFile(PrintWriter printWriter, String prefix)

Class InternalNode implements INode

This nodes just keeps references to lower nodes.

public void insert(String query, Integer index)
    Calls (query[0] - 'a'). child's insert method with query(1:), index.

public ListIterator<Integer> getIndexOfUrls(String query)
    Returns (query[0] - 'a'). child's getIndexOfUrls(query[1:]) method.

public void writeToFile(PrintWriter printWriter, String prefix)
    Calls i. child's writeToFile method after add prefix 'a' + i
    (ex. prefix + a for 0. child, prefix + b for 1. child, ...)

Class Leaf implements INode

This nodes just keeps Integer arraylist.

final private ArrayList<Integer> links = new ArrayList<Integer>()
public void insert(String query, Integer index)
    Inserts index to arraylist.

public ListIterator<Integer> getIndexOfUrls(String query)
    Returns ListIterator of arraylist.

public ArrayList<Integer> getLinks()
    Returns arraylist.

public void writeToFile(PrintWriter printWriter, String prefix)
    Prints prefix and indexes to printWriter stream.

Class UrlForest

public void insert(String query, Result result)
    Adds result to arraylist.
    Calls (query[0] - 'a'). tree's insert method with query[1:], index

public ListIterator<Integer> getIndexOfUrls(String query)
    Returns <query> node's ListIterator.

public Result getResult(Integer index)
    Returns result at index.

public void writeToFile(String filename)
    Write forest to text file named <filename>.

public void readFromFile(String filename)
    Restore forest from text file named <filename>.