

GEBZE TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CSE 343 – SOFTWARE ENGINEERING ONLINE SEARCH ENGINE Version 1.00

GROUP – 1

FURKAN MUSTAFA AKSOY

EMRULLAH GENÇOĞLU

DENIZ 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	5
Design	5
Architectural Design	5
Interface Design	5
Component Design	6
Database Design	6
Server UML	6
Interface UML	7
IMPLEMENTATION	8
Server	
Class Crawler	8
Class Result	
Class ServerMain	8
Data Structure	8
Class Result	
Class UrlTree	9
Interface INode	9
Class InternalNode implements INode	
Class Leaf implements INode	9
Class UrlForest	10

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.

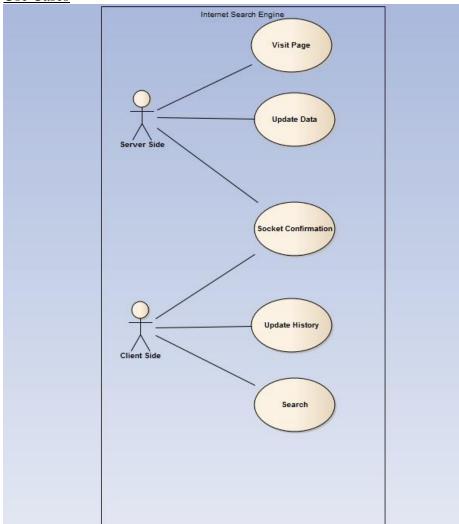
Architectural Design

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

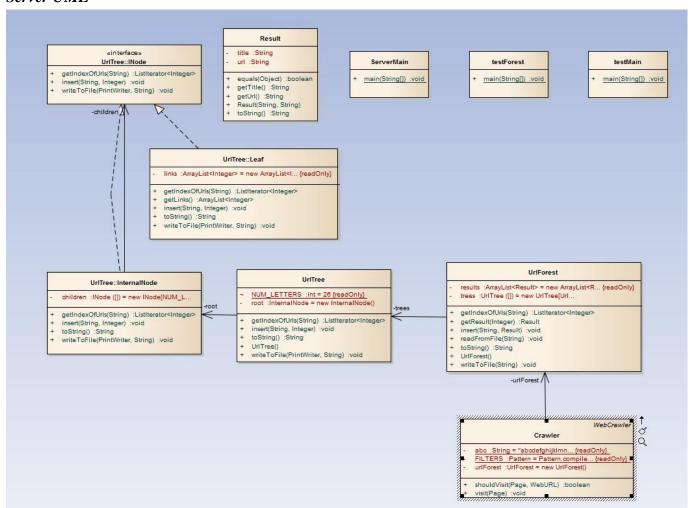
Server has two mode.

- 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

	javax.swing.JFrame
	Runnable NewJFrame
	input :BufferedReader
	jButton1 :javax.swing.JButton jButton2 :javax.swing.JButton
	jButton3 :javax.swing.JButton
	jLabel1 :javax.swing.JLabel
	jLabel2 :javax.swing.JLabel
	jLabel3 :javax.swing.JLabel
	jLabel4 :javax.swing.JLabel
	jLabel5 :javax.swing.JLabel
	jLabel6 :javax.swing.JLabel
	jMenu1 :javax.swing.JMenu jMenu2 :javax.swing.JMenu
	jMenuBar1 :javax.swing.JMenuBar
	jMenuItem2 :javax.swing.JMenuItem
	jTextField1 :javax.swing.JTextField
	jTextField2 :javax.swing.JTextField
	label1 :java.awt.Label
	label2 :java.awt.Label
	label3 :java.awt.Label
	label4 :java.awt.Label label5 :iava.awt.Label
	maximumIndex :int = 0
	out :PrintWriter
	ownResult :ArrayList <result> = new ArrayList<r< td=""></r<></result>
٠	pageIndex :int = 1
٠	result :String
۲	t :Thread
+	checkFirstPage():void
	checkinterPages(int) :void
	checkLastPage() :void
٠	getData() :void
H	getDialog(JFrame, String) :void
٠	getSite() :String
	hideElement() :void initComponents() :void
	jButton1ActionPerformed(java.awt.event.ActionEvent) :void
	jButton1KeyPressed(java.awt.event.KeyEvent) :void
	¡Button1MousePressed(java.awt.event.MouseEvent) :void
	jButton2ActionPerformed(java.awt.event.ActionEvent) :void
	jButton3ActionPerformed(java.awt.event.ActionEvent) :void
	jLabel1MousePressed(java.awt.event.MouseEvent) :void
	jLabel2MousePressed(java.awt.event.MouseEvent) :void
	jLabel3MousePressed(java.awt.event.MouseEvent) :void jLabel4MousePressed(java.awt.event.MouseEvent) :void
	jLabel5MousePressed(java.awt.event.MouseEvent) :void
	jMenu2MousePressed(java.awt.event.MouseEvent) :void
	jMenuItem2MousePressed(java.awt.event.MouseEvent) :void
	jTextField1ActionPerformed(java.awt.event.ActionEvent) :void
	jTextField1KeyPressed(java.awt.event.KeyEvent) :void
	jTextField2ActionPerformed(java.awt.event.ActionEvent) :void
	label1MousePressed(java.awt.event.MouseEvent) :void label2MousePressed(java.awt.event.MouseEvent) :void
	label3MousePressed(java.awt.event.MouseEvent) :void
	label4MousePressed(java.awt.event.MouseEvent) :void
	label5MousePressed(java.awt.event.MouseEvent) :void
+	main(String[]) :void
٠	NewJFrame()
٠	openWebpage(URI) :void
+	openWebpage(URL) :void
٠	run() :void
+	sendRequest(String) :void setSite(String) :void
	Setone(String) . You

Г	javax.swing.JFrame
	AboutFrame
-	jButton1 :javax.swing.JButton
-	jLabel1 :javax.swing.JLabel
-	jLabel11 :javax.swing.JLabel
-	jLabel12 :javax.swing.JLabel
-	jLabel13 :javax.swing.JLabel
-	jLabel14 :javax.swing.JLabel
-	jLabel15 :javax.swing.JLabel
-	jLabel16 :javax.swing.JLabel
-	jLabel2 :javax.swing.JLabel
-	jLabel3 :javax.swing.JLabel
-	jLabel4 :javax.swing.JLabel
-	jLabel5 :javax.swing.JLabel
-	jLabel8 :javax.swing.JLabel
-	jLabel7 :javax.swing.JLabel
-	jLabel8 :javax.swing.JLabel
_	AboutFrame()
	initComponents() :void
	jButton1MousePressed(java.awt.event.MouseEvent) :void
+	main(String[]) :void

	Region
	Browser
~ ~	browser :WebView = new WebView() {readOnly} webEngine :WebEngine = browser.getEngine() {readOnly}
+ #	Browser() layoutChildren() :void

Result - title :String - url :String + equals(Object) :boolean + getTitle() :String + getUrl() :String + Result(String, String) + toString() :String

WebFrame + searched :String = "" initAndShowGUI(String) :void initComponents() :void initFX(JFXPanel, String) :void main(String[]) :void WebFrame(String)

javax.swing.JFrame

javax.swing.JFrame WebViewClass

- initComponents():void + main(String[]) :void + setters() :void + WebViewClass()

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.

<u>final private ArrayList<Integer> links = new ArrayList<Integer>()</u>

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>.