



Computer Engineering Department

A.P. Shah Institute of Technology

G.B.Road,Kasarvadavali, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Java/SBL Mini-Project

Project Title - Theia (God Search)

Arun Mathai Shaju Kollannur (20102049)

Ananya Manoj Ketkar (20102044)

Dhruv Dharmesh Mehta (20102074)

Course Code:
CSL304

Problem Definition

Theia :-

1. It eliminates all the user's needless efforts of visiting multiple search engines one by one and looking for the exact same term in all of those.
2. It provides people the convenience to acquire multiple search results from multiple search engines, but at the same time.
3. Because it scraps the web via a Searx Instance, all the searches are privacy respected and no meta - data of the user is exploited.

Hardware and Software Requirements

Hardware Requirements

1. CPU -> 1.0 GHZ or above
2. RAM -> 300 MB or more of Free Usable Memory
3. DISK -> 2 MB of Free Usable Storage
4. Network Card -> Minimum support 400 Kbps for smooth experience

Software Requirements

1. Any Operating System Running JDK 11 and above.
(Tested on OpenJDK-11, OpenJDK 13, OracleJDK 16 & OpenJDK 17)

Functions/ Libraries/ Packages Used

Classes of Swing Used

Class	Description
<code>AbstractAction</code>	This class provides default implementations for the JFC <code>Action</code> interface.
<code>AbstractButton</code>	Defines common behaviors for buttons and menu items.
<code>AbstractCellEditor</code>	A base class for <code>CellEditors</code> , providing default implementations for the methods in the <code>CellEditor</code> interface except <code>getCellEditorValue()</code> .
<code>AbstractListModel<E></code>	The abstract definition for the data model that provides a <code>List</code> with its contents.
<code>AbstractSpinnerModel</code>	This class provides the <code>ChangeListener</code> part of the <code>SpinnerModel</code> interface that should be suitable for most concrete <code>SpinnerModel</code> implementations.
<code>ActionMap</code>	<code>ActionMap</code> provides mappings from <code>Objects</code> (called <i>keys</i> or <i>Action names</i>) to <code>Actions</code> .
<code>BorderFactory</code>	Factory class for vending standard <code>Border</code> objects.
<code>Box</code>	A lightweight container that uses a <code>BoxLayout</code> object as its layout manager.
<code>Box.Filler</code>	An implementation of a lightweight component that participates in layout but has no view.
<code>BoxLayout</code>	A layout manager that allows multiple components to be laid out either vertically or horizontally.

Interfaces of Swing Used

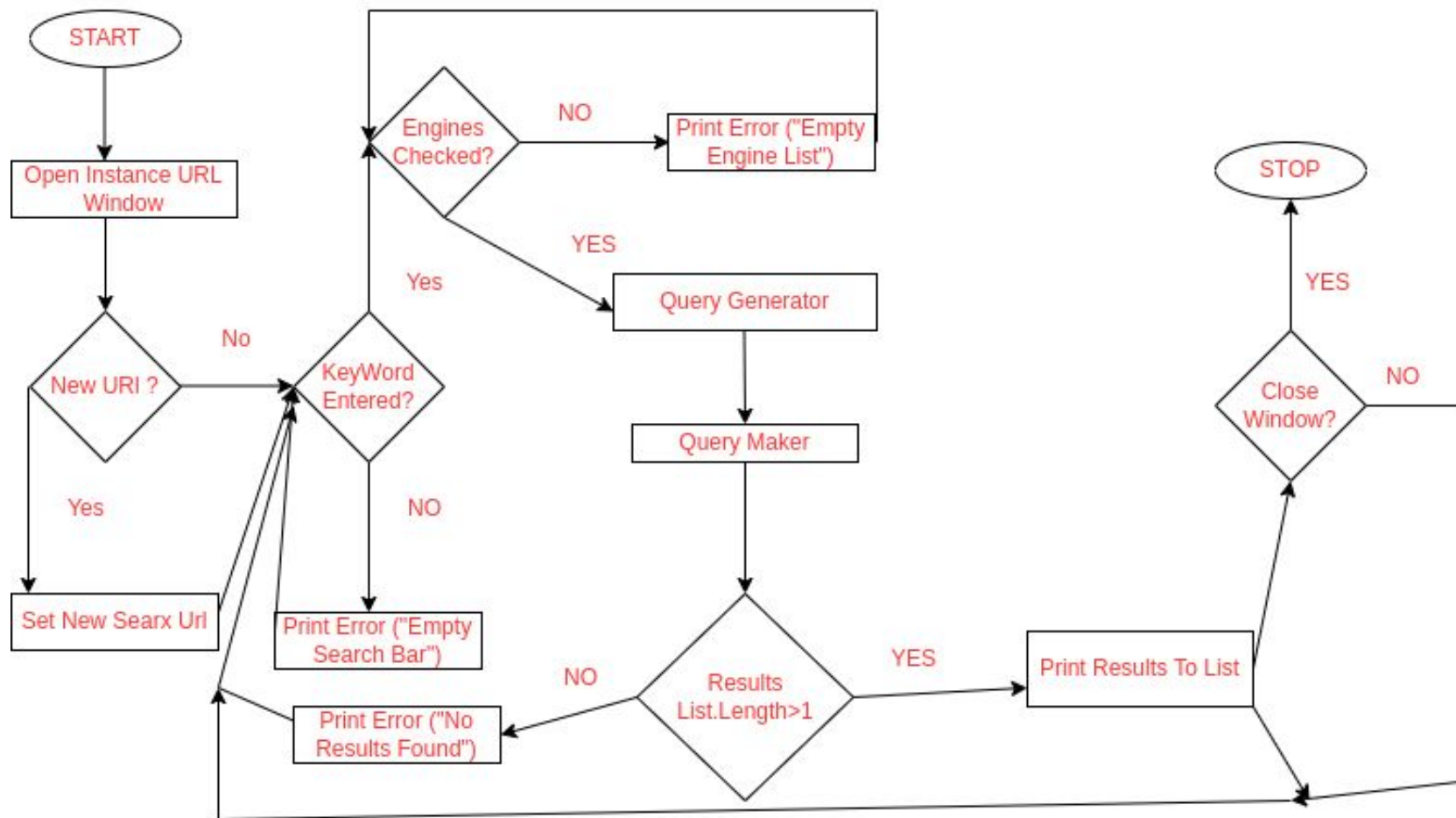
Action	The Action interface provides a useful extension to the ActionListener interface in cases where the same functionality may be accessed by several controls.
BoundedRangeModel	Defines the data model used by components like Sliders and ProgressBars.
ButtonModel	State model for buttons.
CellEditor	This interface defines the methods any general editor should be able to implement.
ComboBoxEditor	The editor component used for JComboBox components.
ComboBoxModel<E>	A data model for a combo box.
DesktopManager	DesktopManager objects are owned by a JDesktopPane object.
Icon	A small fixed size picture, typically used to decorate components.
JComboBox.KeySelectionManager	The interface that defines a KeySelectionManager.
ListCellRenderer<E>	Identifies components that can be used as "rubber stamps" to paint the cells in a JList.
ListModel<E>	This interface defines the methods components like JList use to get the value of each cell in a list and the length of the list.
ListSelectionModel	This interface represents the current state of the selection for any of the components that display a list of values with stable indices.
MenuItem	Any component that can be placed into a menu should implement this interface.
MutableComboBoxModel<E>	A mutable version of ComboBoxModel.
Painter<T>	A painting delegate.
Renderer	Defines the requirements for an object responsible for "rendering" (displaying) a value.
RootPaneContainer	This interface is implemented by components that have a single JRootPane child: JDialog, JFrame, JWindow, JApplet, JInternalFrame.
Scrollable	An interface that provides information to a scrolling container like JScrollPane.
ScrollPaneConstants	Constants used with the JScrollPane component.
SingleSelectionModel	A model that supports at most one indexed selection.
SpinnerModel	A model for a potentially unbounded sequence of object values.

Interfaces/Classes Of Jsoup External Library Used

Interface	Description
Connection	The Connection interface is a convenient HTTP client and session object to fetch content from the web, and parse them into Documents.
Class	Description
Jsoup	The core public access point to the jsoup functionality.

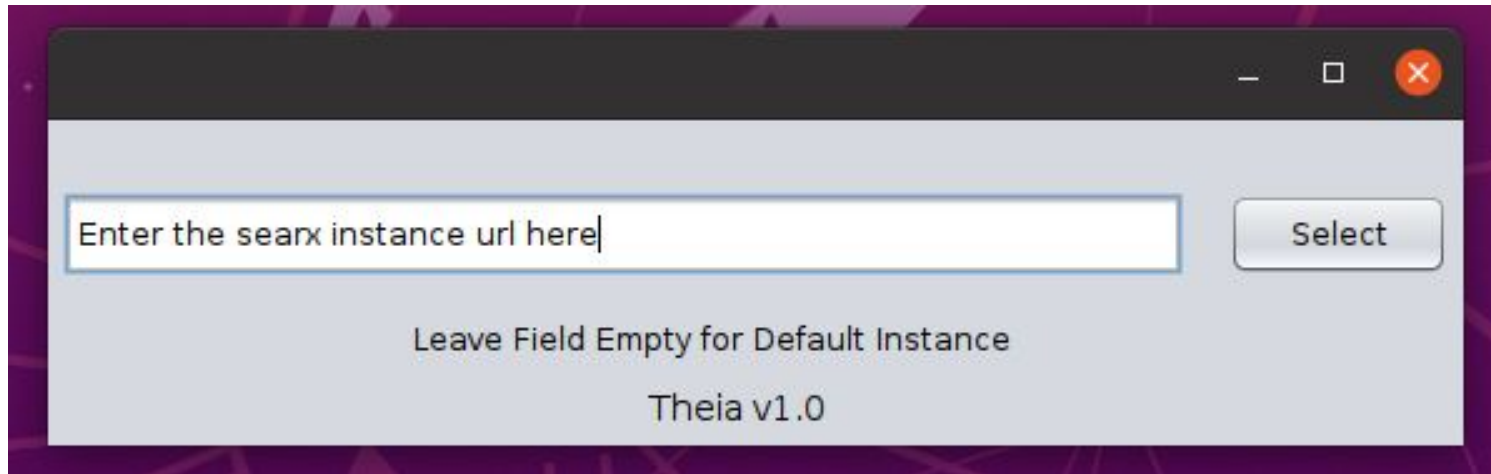
```
Document doc = Jsoup.connect("https://en.wikipedia.org/").get();
log(doc.title());
Elements newsHeadlines = doc.select("#mp-itn b a");
for (Element headline : newsHeadlines) {
    log("%s\n\t%s",
        headline.attr("title"), headline.absUrl("href"));
}
```

Flow Chart



Project Snapshots

Home Window



A screenshot of a web application window titled "Home Window". The window has a dark gray header bar with standard window controls (minimize, maximize, close) on the right. The main content area is light gray. It features a text input field with the placeholder text "Enter the searx instance url here". To the right of the input field is a button labeled "Select". Below the input field, centered, is the text "Leave Field Empty for Default Instance". At the bottom center of the window is the text "Theia v1.0".

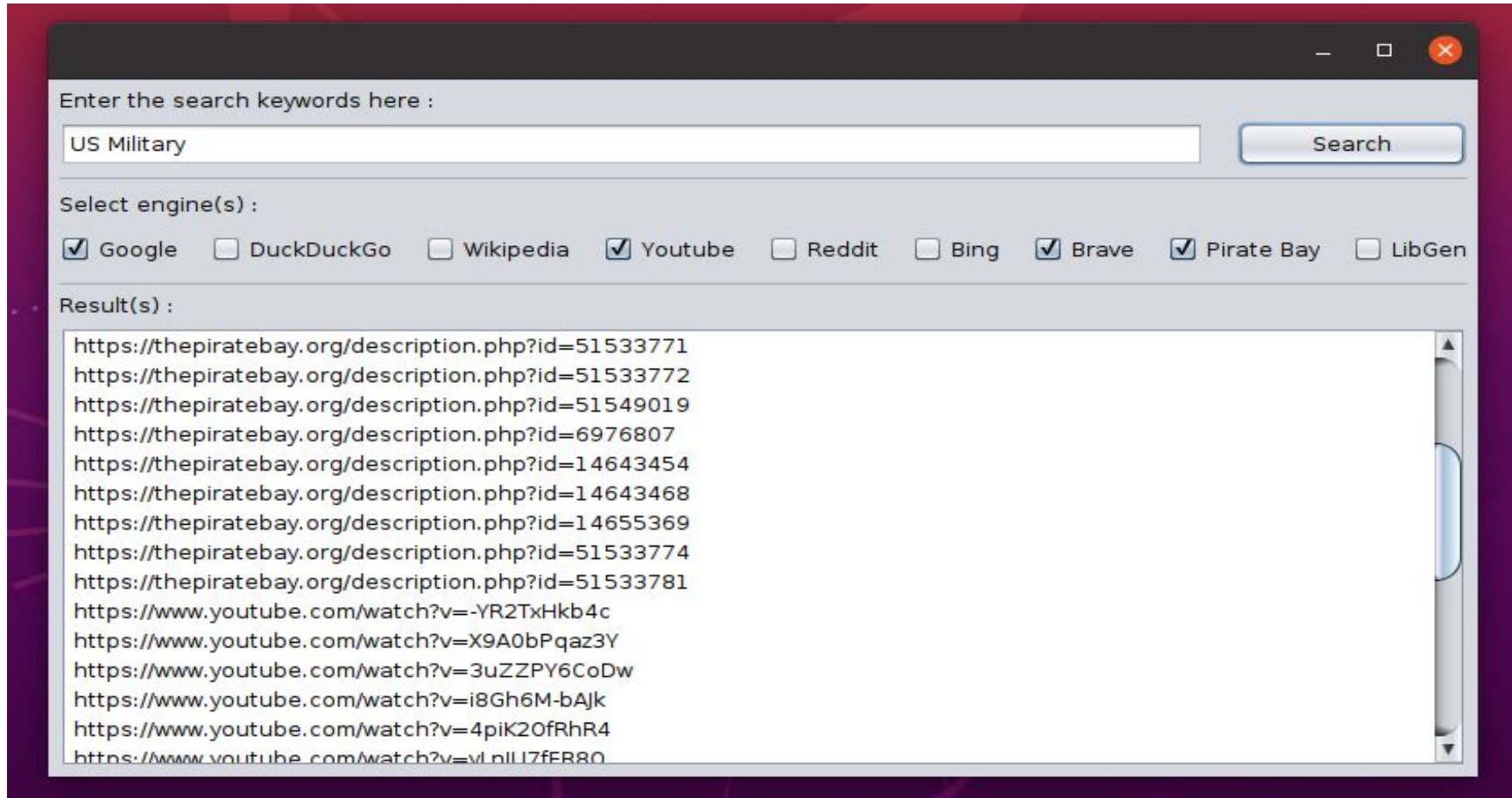
Enter the searx instance url here

Select

Leave Field Empty for Default Instance

Theia v1.0

Main Window



DEVELOPMENT ENVIRONMENT

Theia - Apache NetBeans IDE 12.5

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Projects Files Services

- Theia
 - Source Packages
 - theia
 - InstanceUrlCapture.java
 - QueryGenerator.java
 - QueryMaker.java
 - SearchWindow.java
 - Libraries

InstanceUrlCapture.java - Navigator

Members

- InstanceUrlCapture :: JFrame
 - InstanceUrlCapture()
 - initComponents()
 - instanceSelectButtonActionPerformed(ActionEvent ev)
 - instanceUrlFieldActionPerformed(ActionEvent ev)
 - main(String[] args)
 - defaultInstanceText : JLabel
 - instanceSelectButton : JButton
 - instanceUrlField : JTextField
 - ilabel1 : JLabel

Source

```
107
108
109     try {
110         Document document = Jsoup.connect(instanceUrl + "/search?q=apple").get();
111         Elements text = document.select("div[class=external-link]");
112         String[] links = new String[text.size()];
113         for (int i = 0; i < text.size(); i++) {
114             links[i] = text.get(i).text();
115         }
116         if (links.length == 0 || links == null) {
117             defaultInstanceText.setText("Wrong or Invalid Instance Url");
118         } else {
119             System.out.println("The searx instance being used is " + instanceUrl);
120             this.setVisible(false);
121             this.dispose();
122             new SearchWindow(instanceUrl).setVisible(true);
123         }
124     } catch (Exception e) {
```

Output - Theia (run)

```
run:
The searx instance being used is https://search.disroot.org
/search?q=US%20Military&engines=google,youtube,piratebay,brave
```

Theia (run) running... 1:1 INS Unix (LF)

References

APIs/Libraries Documentation

1. <https://jsoup.org/apidocs/>
2. <https://devdocs.io/openjdk/>

OpenJDK/OracleJDK Download

1. <https://openjdk.java.net/install/>
2. <https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html>

Swing Implementations

<http://web.mit.edu/6.005/www/sp14/psets/ps4/java-6-tutorial/components.html>

Thank You