**CS408 Test Plan**

**Team 2 The Info Crawler**

**Hanxun Huang, Sun Kai, Yijie Wu, Longding Zhang, Di Yao.**

1. **Test Plan**

The test plan is ordered by the requirement on the backlog not by severity.

**(1)F-1 As a user, I would like to provide a target URL for the program to use.**

**Functional Test Case:**

F-1-01

**A. Identification and classification:**

Test Case ID: F-1-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can take a URL as input.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a URL and time interval or increment times

**C. Expected Result:**

1. The program asks for Keyword or HTML element.

2. The program will display error message if the URL is invalid.

**Equivalence class Test Case:**

F-1-02

**A. Identification and classification:**

Test Case ID: F-1-02

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program should validate the URL before using

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a non-exist URL

**C. Expected Result:**

1. The program should return “MalformedURLException” or other error message.

2. The program should ask for a new URL

F-1-03

**A. Identification and classification:**

Test Case ID: F-1-03

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program should validate the URL before using

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a valid URL

(The URL should exist and contains a plain text file)

**C. Expected Result:**

1. The program asks for Keyword or HTML element.

F-1-04

**A. Identification and classification:**

Test Case ID: F-1-04

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can handle empty file.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a URL with a empty file (A plain text file)

**C. Expected Result:**

1. The program should return “Empty file” or other error message.

2. The program should ask for a new URL

**Boundary Value Test Case:**

F-1-05

**A. Identification and classification:**

Test Case ID: F-1-05

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can handle other file.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a URL with a other type of file (Not a plain text file)

E.g. https://www.cs.purdue.edu/homes/sbykau/cs448/slides/Ch2.pdf

**C. Expected Result:**

1. The program should return a error message.

2. The program should ask for a new URL

**(2) F-2 As a user, I would like to crawl the data by providing either keyword or HTML element.**

**Functional Test Case:**

F-2-01

**A. Identification and classification:**

Test Case ID: F-2-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can take a keyword or HTML element as input.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a valid URL and time interval or increment times

4. Input keywords or HTML elements

**C. Expected Result:**

1. The program ask user to choose method or algorithm.

2. The program will display invalid HTML element or other error message.

**Equivalence class Test Case:**

F-2-02

**A. Identification and classification:**

Test Case ID: F-2-02

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program should validate the HTML element before using

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a random exist HTML element

**C. Expected Result:**

1. The program ask user to choose method or algorithm.

F-2-03

**A. Identification and classification:**

Test Case ID: F-2-03

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program should validate the HTML element before using

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input a random non-exist HTML element

E.g. (<cs408>, has to have “<>” or it will count as keywords)

**C. Expected Result:**

1. The program should return error message.

2. The program should ask for a new keywords or element.

**Boundary Value Test Case:**

F-2-04

**A. Identification and classification:**

Test Case ID: F-2-04

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program should validate the HTML element before using

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input partial HTML element

E.g. “<a” or “</”

**C. Expect Result:**

1. The program ask user to choose method or algorithm.

2. The partial HTML element will be counted as keyword

F-2-05

**A. Identification and classification:**

Test Case ID: F-2-05

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can handle large keyword length.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose mode

3. Input 48 or longer string as keywords

**C. Expected Result:**

1. The program will display error message “Out of keyword boundary” or other similar error messages.

**(3) F-3 As a user, I would like to choose between the repeat mode and periodic mode. Functional Test Case:**

F-3-01

**A. Identification and classification:**

Test Case ID: F-3-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can distinguish repeat mode and periodic mode.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=X

4. Replace X with increment times 10

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expect Result:**

1. The program will print out result from first 10 pages of the boilerlink.

F-3-02

**A. Identification and classification:**

Test Case ID: F-3-02

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can distinguish repeat mode and periodic mode.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose periodic mode

Reply “-p” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input time interval 20

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will printout the result every 20 seconds.

**Equivalence class Test Case:**

Functionaltest case is also equivalence test case for this requirement.

No more Equivalence Class Test Case needed.

**Boundary Value Test Case:**

F-3-03

**A. Identification and classification:**

Test Case ID: F-3-03

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can distinguish repeat mode and periodic mode.

**B. Instructions**

1. Run the program in Command line

2. Random select 3 character other than “p” or “r”

Reply “-x” in command line

3. Replace -x with the character.

**C. Expected Result:**

1. The program will print a error message and ask for a new input.

**(4) F-4 As a user, I would like to have all the information I need from the URL before search.**

**Functional Test Case:**

F-4-01

**A. Identification and classification:**

Test Case ID: F-4-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can grab all the data from the URL

**B. Instructions**

1. Run GetURLContentTest in command line.

Command: Java GetURLConeteTest -U

2. Replace -U with URL that needs to be tested

3. Random select a URL and open the source code in the Browser

(URL should contains a valid file explained in Case F-1-03)

**C. Expected Result:**

1. The Program will printout some data from the URL

**Equivalence class Test Case:**

F-4-02

**A. Identification and classification:**

Test Case ID: F-4-02

System: Info Crawler and Command Line

Severity:2

Explanation: Test the program can grab all the data from the URL

**B. Instructions**

1. Run GetURLContentTest in command line.

Command: Java GetURLConeteTest -U

2. Replace -U with URL that need to be tested

3. Replace X in the following url, random select a number from 1 to 105 to replace.

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=0&CurrentPage=X

**C. Expected Result:**

1. The Program will printout the data from the URL

2. The printout should be the same with the source code in Browser

**Boundary Value Test Case:**

No Boundary Value Test Case needed.

GetURLContentTest only test the GetURLContent method, we will put URL validation process in the main program. There is no need to check the input Validation here.

**(5) F-5 As a user, I would like to have the ability of using keywords to search and view the result.**

**Functional Test Case:**

F-5-01

**A. Identification and classification:**

Test Case ID: F-5-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can grab specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input random keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will print out matched result.

2. The program will print out “0 result found” if there are no match.

**Equivalence class Test Case:**

F-5-02

**A. Identification and classification:**

Test Case ID: F-5-02

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can grab specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page.

F-5-03

**A. Identification and classification:**

Test Case ID: F-5-03

System: Info Crawler and Command Line

Severity: 2

Explanation: Test the program can grab specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “<h3>” as keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will print “0 result found”

**Boundary Value Test Case:**

No Boundary Value Test Case needed.

All the invalid input arguments are tested individually.

**(6) As a user, I would like to choose a different method such as algorithm to do the crawling.**

**Functional Test Case:**

F-6-01

**A. Identification and classification:**

Test Case ID: F-6-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Able to use basic algorithm

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose “word by word method”

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page1.

2. The program also should print out Method name and the time used.

**Equivalence class Test Case:**

F-6-02

**A. Identification and classification:**

Test Case ID: F-6-02

System: Info Crawler and Command Line

Severity:2

Explanation: Able to choose different algorithm

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose other method from available method list

e.g. -r Regular Expression

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page1.

2. The program also should print out Method name and the time used.

**Boundary Value Test Case:**

F-6-03

**A. Identification and classification:**

Test Case ID: F-6-03

System: Info Crawler and Command Line

Severity: 3

Explanation: Able to handle non exist method

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose non exist method to input (Not on the available method list)

**C. Expected Result:**

1. The program ask for exist method to input

**(7) F-7 As a user, I would like to run different job at the same time.**

**Functional Test Case:**

F-7-01

**A. Identification and classification:**

Test Case ID: F-7-01

System: Info Crawler and Command Line

Severity: 2

Explanation: Able to run two jobs at the same time.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose “word by word method”

7. Choose not to receive Notification

8. Choose a new job.

9. Input the same with above only change the URL to

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=2

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page1.

2. The program also should print out Method name and the time used for page 1.

3. The program will print out all the organization name with few html source code from the boilerlink Page2.

4. The program also should print out Method name and the time used for page 2.

5. Two job runs at the same time, the time used should be similar with each other.

**Equivalence class Test Case:**

F-7-02

**A. Identification and classification:**

Test Case ID: F-7-02

System: Info Crawler and Command Line

Severity: 3

Explanation: Able to run several jobs at the same time.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose “word by word method”

7. Choose not to receive Notification

8. Choose 1-4 new jobs.

9. Input the same with above only change the X in the URL from 1-4 accordingly

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=X

**C. Expected Result:**

1. The print out result from each page and time used.

2. All the time used is similar with each other.

**Boundary Value Test Case:**

F-7-03

**A. Identification and classification:**

Test Case ID: F-7-03

System: Info Crawler and Command Line

Severity: 3

Explanation: Able to handle too more new jobs.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose “word by word method”

7. Choose not to receive Notification

8. Choose 5 new jobs.

9. Input the same with above only change the URL X from 1-5 accordingly

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=X

**C. Expected Result:**

1. The program will printout error message

“Only 4 jobs at the same time”

2. Ask if the user want to run first 4 jobs or start over.

Note: We may change how many jobs can be run at the same time, we will also update the test plan if we change.

F-7-04

**A. Identification and classification:**

Test Case ID: F-7-04

System: Info Crawler and Command Line

Severity: 3

Explanation: Able to handle incorrect number

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose “word by word method”

7. Choose not to receive Notification

8. Choose -1 new jobs.

**C. Expected Result:**

1. The program will printout error message

“Please enter 0 - 4”

2. The program will ask the user to enter a new input.

**(8) F-8 As a user, I would like to receive notification when the job is done.**

**Functional Test Case:**

F-8-01

**A. Identification and classification:**

Test Case ID: F-8-01

System: Info Crawler and Command Line

Severity:3

Explanation: Test the program can send notifications

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose to receive Notification

8. Provide a valid email that can be accessed.

9. Name the current job.

10. Choose no more new jobs.

11. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page.

2. The user should receive a email says the jobs is done and the name of the jobs.

**Equivalence class Test Case:**

F-8-02

**A. Identification and classification:**

Test Case ID: F-8-02

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can send notifications and name of the job

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose to receive Notification

8. Provide a valid email that can be accessed.

9. Name the current job with random 1-10 characters.

10. Choose no more new jobs.

11. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from the boilerlink Page.

2. The user should receive a email says the jobs is done. The email also contains the name of the job.

Note: We wont check the name of the program other the length.

**Boundary Value Test Case:**

F-8-03

**A. Identification and classification:**

Test Case ID: F-8-03

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can handle invalid email.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose to receive Notification

8. Provide a 5 invalid email.

E.g. 12345.com without “@”

     12345@.com “@” connect with “.” without domain name.

**C. Expected Result:**

1. Program will ask the user to input correct email account.

F-8-04

**A. Identification and classification:**

Test Case ID: F-8-04

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can send notifications and handle NULL input

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose to receive Notification

8. Provide a valid email that can be accessed.

9. No input for the job name

10. Choose no more new jobs.

11.. Start

**C. Expected Result:**

1. The program will print out all the organization name with few html source code from he boilerlink Page.

2. The user should receive a email says the jobs is done. The email will show there is no name for the job.

F-8-05

**A. Identification and classification:**

Test Case ID: F-8-05

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can send notifications and longer string input

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose to receive Notification

8. Provide a valid email that can be accessed.

9. Input more than 10 char as the name of the job

**C. Expected Result:**

1. The program will print out error message “0-10 characters only”

**(9) F-9 As a user, I would like to access this tool using command line.**

**Functional Test Case:**

F-9-01

**A. Identification and classification:**

Test Case ID: F-9-01

System: Info Crawler and Command Line

Severity: 1

Explanation: Test the program be used in command line

**B. Instructions**

1. Open command line

2. Input command: Javac InfoCrawler.java

3. Input command: Java InfoCrawler

**C. Expected Result:**

1. The program asks to choose a mode.

**Equivalence class Test Case:**

No Equivalence class Test Case needed.

**Boundary Value Test Case:**

No Boundary Value Test Case needed.

**(10) F-10 As a user, I would like to access this tool using GUI.**

**Functional Test Case:**

F-10-01

**A. Identification and classification:**

Test Case ID: F-10-01

System: Info Crawler and Command Line

Severity: 1

Explanation: Test the program be used in GUI

**B. Instructions**

1. Open command line

2. Input command: Javac InfoCrawlerGUI.java

3. Input command: Java InfoCrawlerGUI

**C. Expected Result:**

1. The program will display GUI of the program

**Equivalence class Test Case:**

F-10-02

**A. Identification and classification:**

Test Case ID: F-10-02

System: Info Crawler, Command Line and GUI

Severity: 1

Explanation: Test the program be used in GUI

**B. Instructions**

1. Open command line

2. Input command: Javac InfoCrawlerGUI.java

3. Input command: Java InfoCrawlerGUI

4. Input a set of valid value in each textfield.

E.g. Use a set of Command line input

**C. Expected Result:**

1. The program response and give output same with the command line.

2. Each button should response accordingly.

**Boundary Value Test Case:**

F-10-03

**A. Identification and classification:**

Test Case ID: F-10-03

System: Info Crawler, Command Line and GUI

Severity:1

Explanation: Test the program be used in GUI

**B. Instructions**

1. Open command line

2. Input command: Javac InfoCrawlerGUI.java

3. Input command: Java InfoCrawlerGUI

4. Input a set of invalid value in each textfield.

E.g. Use a set of Command line invalid input

**C. Expected Result:**

1. The program response and give which input is invalid.

2. Ask the user to try again.

Note: The GUI is using same source code with command line, no need to check the functionality individually again.

**(11) F-11 As a user, I would like to remove some unwanted information.**

**Functional Test Case:**

F-11-01

**A. Identification and classification:**

Test Case ID: F-11-01

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can remove specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to use the filter

11. Input “View Recommended Organizations” as remove keywords

**C. Expected Result:**

1. The program will print a new output without “View Recommended Organizations”

**Equivalence class Test Case:**

F-11-02

**A. Identification and classification:**

Test Case ID: F-11-02

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can remove specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to use the filter

11a. Input the following keyword as remove keywords each time and start over

“Manage Your Interests”

“View Recommended Organizations”

“&quot;”(Means “ and ” in the source code)

11b. In put the above keyword each time and choose to use the filter again instead of start over

**C. Expected Result:**

1. The program will print out the result without the remove keyword accordingly.

2. The program will print out the fellowing result in case of step 11b.

A Cause for Paws (ACP)

21st Century Scholars (Scholar Corps for 21st Century Scholars)

3D Printing Club (3DPC)

A Cultural Connection promoting Leadership opportunities and Academic achievement for International and Multicultural students (ACCLAIM)

A Global Friendship Campus Club Purdue University (AGF Campus Club)

AAE Graduate Women’s Gathering (AAE GWG)

Acacia

Academy of Student Pharmacists (APhA-ASP)

Accounting Association (PAA)

ACM SIGGRAPH

**Boundary Value Test Case:**

F-11-03

**A. Identification and classification:**

Test Case ID: F-11-03

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can remove specific information.

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to use the filter

11. Input 3 non-exist as remove keywords (e.g. <h5> or <a>)

**C. Expected Result:**

1. Print out “0 result found”

**(12) F-12 As a user, I would like to save the result into a file.**

**Functional Test Case:**

F-12-01

**A. Identification and classification:**

Test Case ID: F-12-01

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can save the result into file

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to save into a file

11. Input a valid file name

**C. Expected Result:**

1. The program will save the result into the file in the same folder with the program.

**Equivalence class Test Case:**

F-12-02

**A. Identification and classification:**

Test Case ID: F-12-02

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can save the result into file in different folder

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to save into a file

11. Input a valid file name with a valid file path.

**C. Expected Result:**

1. The program will save the result into the file in the path accordingly.

**Boundary Value Test Case:**

F-12-03

**A. Identification and classification:**

Test Case ID: F-12-03

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can save the result into file and handle invalid path

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to save into a file

11. Input 3 different valid file name with 3 different invalid file path.

e.g. /homes/desktop/h3/file.txt where h3 folder does not exist.

**C. Expected Result:**

1. The program will print out error message.

F-12-04

**A. Identification and classification:**

Test Case ID: F-12-04

System: Info Crawler and Command Line

Severity: 3

Explanation: Test the program can save the result into file and handle invalid file name

**B. Instructions**

1. Run the program in Command line

2. Proceed to choose repeat mode

Reply “-r” in command line

3. Input the following URL

https://boilerlink.purdue.edu/Organizations?SearchType=None&SelectedCategoryId=1

4. Input 1 as increment times.

5. Input “target="\_self">”  and  “</a>” as the keywords

6. Choose a method

7. Choose not to receive Notification

8. Choose no more new jobs.

9. Start

10. Choose to save into a file

11. Input 3 different invalid file name with 3 different valid file path.

E.g. /homes/desktop/h3/file.txt where file.txt already exist.

**C. Expected Result:**

1. The program will ask if the user want to overwrite the file or give another name to the file