List of use cases

[I. Run the crawler to crawl on MAS page 2](#_Toc300725731)

[1. Run the crawler. 2](#_Toc300725732)

[2. Set configurations. 4](#_Toc300725733)

[2.1 Set basic configurations. 4](#_Toc300725734)

[2.2 Set database connection. 5](#_Toc300725735)

[2.3 Set schedule. 6](#_Toc300725736)

[3. Load configurations. 6](#_Toc300725737)

[4. Collect data and save to database. 7](#_Toc300725738)

[4.1 Crawl the publication 7](#_Toc300725739)

[4.2 Crawl the author data 9](#_Toc300725740)

[4.3 Crawl the organization data 10](#_Toc300725741)

[4.4 Crawl the journal data 11](#_Toc300725742)

[5. Pause the crawling process. 12](#_Toc300725743)

[6. Handling the error. 13](#_Toc300725744)

[6.1 Internet connection error. 13](#_Toc300725745)

[6.2 Handling the database connection error 14](#_Toc300725746)

# Run the crawler to crawl on MAS page

## Run the crawler.

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.1 Run the crawler | |
| The user’s aim | To start the crawling process on MAS page. | |
| Description | A user can set configurations via GUI. Then the system loads configurations and starts collecting data. Any outputs are sent back to the user on the output console. | |
| Main actor | User | |
| Other actors | None | |
| Conditions for starting | User has access to the Internet.  The user has chosen the MAS tab. | |
| Desired result | The crawl starts to collect data on the MAS page.  Numbers of authors and publications saved to database are returned to GUI.  Crawling process information are shown on the output console. | |
| Other results | Result | Reason/Condition |
| Cannot connect to DB.  Message: “DB Connection failed”. | The existing DB Connection is incorrectly configured. |
| Request is timed out.  Message: “Timed out.” | The server doesn’t response to the request. |
| No of objects are reached.  Message: “Maximum number of objects are reached.”. | Maximum number of objects is reached. The system stops crawling. |
| Invalid number of maximum number of objects  Message: “Invalid maximum object number”. | The user enters a negative number on the maximum number of objects field. |
| Main flow | 1. The user selects the MAS tab. 2. The user can choose to set basic configurations or use default settings.    1. If YES – Go to *UC01.2.1 Set basic configurations.*    2. If NO – Go to 3. 3. The user clicks on the START button or the green PLAY icon on the control-bar, the system goes to *UC01.3 Load configurations.* 4. The system goes to *UC01.4 Collect data and saves to database.* | |
| **Alternative flow** | 1. The user selects the MAS tab. 2. The user can choose to set configurations or use default settings. 3. The user starts the crawler. 4. The system checks the configuration.    1. If the maximum number of objects is negative – Go to 5.    2. If the system cannot connect to DB – Go to 6. 5. The system outputs “Invalid maximum object number”. 6. The system outputs “DB Connection failed.”   OR   1. The user chooses to set DB Connection. 2. The user clicks the Configuration icon. 3. The user chooses the DB Connection tab. 4. The system continues with *UC01.2.2 Set database connection.*   OR   1. The user chooses to set schedule. 2. The user clicks the Configuration icon. 3. The user clicks on the Schedule tab. 4. The system continues with *UC01.2.3 Set schedule.* | |
| Comments | The UC01.2.3 Set schedule is not necessary at present.  Default settings of basic configurations are   |  |  | | --- | --- | | Object name | Default setting value | | Object | Publication | | Domain | Computer Science | | Sub domain | All | | Duration | All | | Maximum number | -1 (infinite number) |   Database connection is set default to use with MYSQL, including username “root”, password “localhost” and DB Name ”co-authorship”.(the settings can be different in different computers, so use your own settings)  Schedule is not considered at present. | |
| Questions |  | |
| Desired, non-approved changes |  | |
| Priority | 1 | |

## Set configurations.

### Set basic configurations.

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.2.1 – Set basic configurations | |
| The user’s aim | To select object, domain, sub-domain, duration and max number of object to crawl. | |
| Description | The user selects object, domain, sub-domain, duration and max number of object to crawl via selection-boxes. | |
| Main actor | User | |
| Other actors |  | |
| Conditions for starting |  | |
| Desired result | Basic configurations are set correctly. | |
| Other results | Result | Reason/Condition |
|  |  |
| Main flow | 1. User selects type of object using the object drop-box. 2. User selects domain using the domain drop-box. 3. User selects sub-domain using the sub-domain drop-box. 4. User selects duration using the object drop-box. 5. User types the maximum number of object. | |
| **Alternative flow** |  | |
| **Comments** | List of type of object includes Author, Publication, Conference, Journal, Organization and Keyword.  List of domain includes Computer Science, …. (the rest on the MAS site)  List of sub-domain includes Data Mining, Databases,… (the rest on the MAS site)  List of durations includes Within 5 years, Within 10 years, All.  The user is required to type a non-negative number for the maximum number of saved objects. Any negative number is considered to infinite number of objects. | |
| Questions |  | |
| Desired, non-approved changes |  | |
| Priority | 3 | |

### Set database connection.

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.2.2 Set database connection | |
| The user’s aim | To set the JDBC suit with using DBMS | |
| Description | The user provides URL, username and password, database name and then clicks the Test Connection to test the connection to the using DBMS. | |
| Main actor | User | |
| Other actors |  | |
| Conditions for starting |  | |
| Desired result | The system can successfully connect to the using DBMS. | |
| Other results | Result | Reason/Condition |
| Cannot connect to DB.  Message: “DB Connection failed”. | The existing DB Connection is incorrectly configured. |
| Main flow | 1. User clicks the Configuration icon. 2. User chooses the DB Connection tab. 3. User types the URL in the URL textbox. 4. User types the username in the username textbox. 5. User types the password in the password textbox. 6. User types the DB name in the DB name textbox. 7. User clicks the Test Connection button. 8. The system outputs “Successful” and saves user input value to configuration file. | |
| **Alternative flow** | 1. User clicks the Configuration icon. 2. User chooses the DB Connection tab. 3. User types the URL in the URL textbox. 4. User types the username in the username textbox. 5. User types the password in the password textbox. 6. User types the DB name in the DB name textbox. 7. User clicks the Test Connection button. 8. The system fails to connect to DB with input values. 9. The system outputs “Unsuccessful”. | |
| **Comments** |  | |
| **Questions** |  | |
| **Desired, non-approved changes** |  | |
| **Priority** | 2 | |

### Set schedule.

## Load configurations.

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.3 Load configurations | |
| The user’s aim | To load configurations from configuration file. | |
| Description | The system loads parameters from config.xml to a class’s static attributes. | |
| Main actor | System | |
| Other actors |  | |
| Conditions for starting | User clicks the Start button or the green “PLAY” icon on the control-bar. | |
| Desired result | All configuration data is set up successfully. | |
| Other results | Result | Reason/Condition |
| Cannot find the config.xml  “File not found.” | The config.xml file is removed or placed in wrong path. |
|  | Config.xml is in wrong format  “Invalid file format.” | The config.xml file is not in xml format. |
| Main flow | 1. The system creates a new instance of the CONFIG class. 2. The system loads the path of the config.xml file. 3. The system reads all parameters to static variables of the CONFIG class. | |
| **Alternative flow** | 1. The system creates a new instance of the CONFIG class. 2. The system loads the path of the config.xml file. 3. The system returns “File not found.” If it cannot find the config.xml file.   OR   1. The system creates a new instance of the CONFIG class. 2. The system loads the path of the config.xml file. 3. The system reads all parameters to static variables of the CONFIG class. | |
| Comments | Configuration file can be in XML format or JSON format. If it is in JSON format, all the flow above is considered in JSON standards. | |
| Questions |  | |
| Desired, non-approved changes |  | |
| Priority | 2 | |

## Collect data and save to database.

### Crawl the publication

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.4.1 Publication crawling | |
| The user’s aim | To crawl publication | |
| Description | This use case describes the process of getting all data about 1 and all publication in MAS in group of sub domain. System use information about sub domain and configuration to create, then use this URL to request data from Publication site. Then system parses results and save data to DB. | |
| Main actor | System, User | |
| Other actors |  | |
| Conditions for starting | Configuration is OK.  User starts crawling. | |
| Desired result | All data about 1 and all publication in MAS in group of sub domain is stored in DB. | |
| Other results | Result | Reason/Condition |
| System pauses | System has internet connection error, or db access error, or user chooses pausing crawling. |
| System stops. | User does not want to wait for file access and stops crawling. |
| Main flow | 1. Get sub domain. 2. Generate publication list request URL. 3. Request publication list URL. 4. Enqueue publication list URL. 5. Dequeue publication URL. 6. If Publication exists then update sub domain info. 7. Else    1. Request publication.    2. If Publication ID does not exists then save new    3. Parse publication data.       1. Save all “visible” data       2. Call UC01.4.2.       3. Request citation list       4. Enqueue citation list       5. Dequeue citation list       6. If ref paper does not exists then save new citation paper       7. update ref info       8. Request citation context list       9. Enqueue citation context list       10. Dequeue citation context list       11. If citation context paper does not exists then save new citation context paper       12. update citation context info | |
| **Alternative flow** | 1. System has internet connection error or DB access error. 2. Call UC01.6 to handle errors. | |
| Comments |  | |
| Questions |  | |
| Desired, non-approved changes |  | |
| Priority | 2 | |

### Crawl the author data

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.4.2 Crawl the author data | |
| The user’s aim | To get information about author object on MAS site. | |
| Description | The user provides an URL to an author page on MAS site. The system uses this link to load content on that site. Then, it extracts author information and saves to the database. | |
| Main actor | System | |
| Other actors |  | |
| Conditions for starting | User has access to the Internet.  This use case is called from the UC01.4.1use case. | |
| Desired result | Author information is saved to database. | |
| Other results | Result | Reason/Condition |
| Cannot connect to DB.  Message: “DB Connection failed”. | The existing DB Connection is incorrectly configured. |
|  | Cannot access to the link.  Message: “Internet connection failed.” | 1. Invalid link. 2. Request is timed out. |
| Main flow | 1. System reads the URL. 2. System loads content on the website based on the provided URL. 3. System extracts needed information related to authors. 4. System locates link to the organisation where the author is working in. Then, system continues with use case UC01.4.3 Crawl the org data. 5. System saves these information to the database. | |
| **Alternative flow** | 1. System reads the URL. 2. System checks the URL. 3. If the URL does not lead to an author site, the system outputs “Link is not correct.” | |
| **Comments** | Author information includes:   1. Author ID -> Auto generated. 2. authorName. 3. Affiliation. (Where?) 4. Image-> Save the link to the image. 5. Email address. 6. Website -> Homepage 7. idOrg -> get idOrg from UC01.4.3 8. H-index. 9. G-index. | |
| **Questions** |  | |
| **Desired, non-approved changes** |  | |
| **Priority** | 3 | |

### Crawl the organization data

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.4.3 Crawl the organisation data | |
| The user’s aim | To get information about organisation object on MAS site. | |
| Description | The user provides an URL to an org page on MAS site. The system uses this link to load content on that site. Then, it extracts org information and saves to the database. | |
| Main actor | System | |
| Other actors |  | |
| Conditions for starting | User has access to the Internet.  This use case is called from the UC01.4.1use case. | |
| Desired result | Org information is saved to database. | |
| Other results | Result | Reason/Condition |
| Cannot connect to DB.  Message: “DB Connection failed”. | The existing DB Connection is incorrectly configured. |
|  | Cannot access to the link.  Message: “Internet connection failed.” | 1. Invalid link. 2. Request is timed out. |
| Main flow | 1. System reads the URL. 2. System loads content on the website based on the provided URL. 3. System extracts needed information related to org object. 4. System saves these information to the database. | |
| **Alternative flow** | 1. System reads the URL. 2. System checks the URL. 3. If the URL does not lead to an org site, the system outputs “Link is not correct.” | |
| **Comments** | Org information includes:   1. IdOrg -> auto generated. 2. Name. 3. Website 4. Continent 5. idOrg1 6. h-index. | |
| **Questions** |  | |
| **Desired, non-approved changes** |  | |
| **Priority** | 3 | |

### Crawl the journal data

|  |  |  |
| --- | --- | --- |
| Use Case ID and name | UC01.4.3 Crawl the journal data | |
| The user’s aim | To get information of journal on MAS site. | |
| Description | The user provides an URL to an journal page on MAS site. The system uses this link to load content on that site. Then, it extracts journal information and saves to the database. | |
| Main actor | System | |
| Other actors |  | |
| Conditions for starting | User has access to the Internet.  This use case is called from the UC01.4.1use case. | |
| Desired result | Org information is saved to database. | |
| Other results | Result | Reason/Condition |
| Cannot connect to DB.  Message: “DB Connection failed”. | The existing DB Connection is incorrectly configured. |
|  | Cannot access to the link.  Message: “Internet connection failed.” | 1. Invalid link. 2. Request is timed out. |
| Main flow | 1. System reads the URL. 2. System loads content on the website based on the provided URL. 3. System extracts needed information related to journal object. 4. System saves these information to the database. | |
| **Alternative flow** | 1. System reads the URL. 2. System checks the URL. 3. If the URL does not lead to an org site, the system outputs “Link is not correct.” | |
| **Comments** | Journal information includes:   1. IdJournal -> auto generated. 2. Name. 3. Website 4. yearStart 5. yearEnd 6. organization | |
| **Questions** |  | |
| **Desired, non-approved changes** |  | |
| **Priority** | 3 | |

## Pause the crawling process.

|  |  |
| --- | --- |
| Use Case ID and name | UC01.5 Pause the crawling process |
| The user’s aim | To pause the crawling process |
| Description | When system is crawling, it can pause. Its current state can be resumed later. |
| Main actor | System, User |
| Other actors |  |
| Conditions for starting | System is crawling data. |
| Desired result | System pause, all data is preserved. |
| Other results | System stops. |
| Main flow | 1. The system or user calls pause function. 2. The system saves all current state of objects and variables in system to file.    1. No error happens.    2. Saves. 3. The system closes internet connection. 4. The system closes DB connection. 5. The system sets status as paused. |
| **Alternative flow** | 1. The system or user calls pause function. 2. The system saves all current state of objects and variables in system to file.    1. File access error happens    2. The system shows “File access error” message, waits to save again.    3. Saving again is ok. 3. The system closes internet connection. 4. The system closes DB connection. 5. The system sets status as paused.   OR   1. The system or user calls pause function. 2. The system saves all current state of objects and variables in system to file.    1. File access error happens    2. The system shows “File access error” message, waits to save again.    3. User chooses stopping. 3. The system closes internet connection. 4. The system closes DB connection. 5. The system stops. |
| Comments | Pause function have to save all current state, variables ‘s data to file, not DB. |
| Questions |  |
| Desired, non-approved changes |  |
| Priority | n |

## Handling the error.

### Internet connection error.

|  |  |
| --- | --- |
| Use Case ID and name | UC01.6.1 Handling internet connection error |
| The user’s aim | To handle internet connection error |
| Description | When system is running, internet connection may have error. This use case handle this problem, preserve all data. |
| Main actor | System |
| Other actors | User |
| Conditions for starting | System is crawling data.  Internet connection error occurs. |
| Desired result | System pause, all data is preserved. System resume when internet connection is available again. |
| Other results | System stops. |
| Main flow | 1. The system calls pause function in pause use case. 2. Loop:    1. The system checks for internet connection availability 3. The system loads saved state. 4. The system resumes. |
| **Alternative flow** | 1. The system calls pause function in pause use case. 2. Loop:    1. The system checks for internet connection availability    2. User chooses stopping crawling. 3. The system stops. |
| Comments | Pause function needs to be written well to save all current state data. |
| Questions |  |
| Desired, non-approved changes |  |
| Priority | n |

### Handling the database connection error

|  |  |
| --- | --- |
| Use Case ID and name | UC01.6.2 Handling DB access error |
| The user’s aim | To handle DB access error |
| Description | When system is running, DB access may have error. This Usecase handle this problem, preserve all data. |
| Main actor | System |
| Other actors | User |
| Conditions for starting | System is crawling data.  DB access error occurs. |
| Desired result | System pause, all data is preserved. System resume when DB access is available again. |
| Other results | System stops. |
| Main flow | 1. The system calls pause function in pause use case. 2. Loop:    1. The system checks for DB access availability 3. The system loads saved state. 4. The system resumes. |
| **Alternative flow** | 1. The system calls pause function in pause usecase. 2. Loop:    1. The system checks for DB access availability    2. User chooses stopping crawling. 3. The system stops. |
| Comments | Pause function has to save current state data to file, not DB. |
| Questions |  |
| Desired, non-approved changes |  |
| Priority | n |