



Thywin – Design Document

Geschreven door: Bobby Bouwmann, Thomas Gerritsen, Thomas Kooi, Erwin Janssen, Imre Woudstra

Datum: 14-5-2014

Project begeleider: Alice de Groot

Opdrachtgever: Joost Kraaijeveld

Table of contents

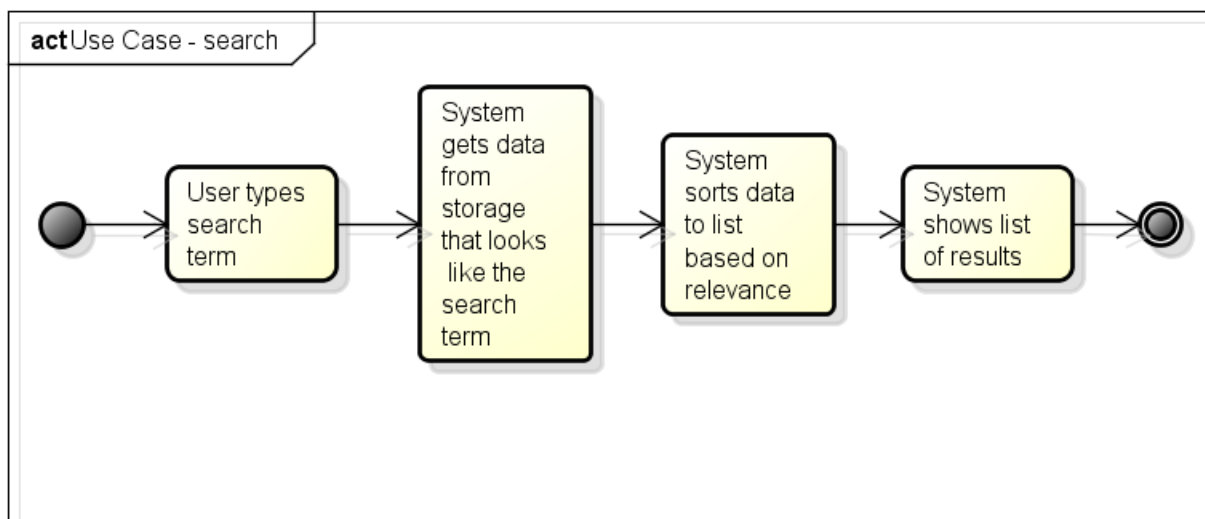
table of contents	2
Use Case 1: Search.....	3
Activity diagram.....	3
Sequence diagrams	3
Use Case 2: Crawler	4
Activity diagram.....	4
Sequence Diagram	5
Use Case 3: Parsen	6
Activity diagram.....	6
Sequence diagrams	7
Use Case 4: Master	8
Get URI.....	8
Put URI.....	8
Get Document	9
Put Document	9
State diagram	10

Use Case 1: Search

The user goes to <http://thywin.com>. The browser then shows a search page. The user types a search term in the search field. The server then shows all the results based on the search term.

Primary Actor: User	
Stakeholders: Site owners	
Preconditions: The site is online and the database is available.	
Post conditions: The user gets results based on their searchterm.	
Main success scenario:	
1. User goes to http://thywin.com .	2. Webserver sends page back.
3. User types search term in the search field.	4. Webserver shows results.
Extensions: (or Alternative flow)	
	[If no results] 4a. Webserver shows message "0 results".

Activity diagram



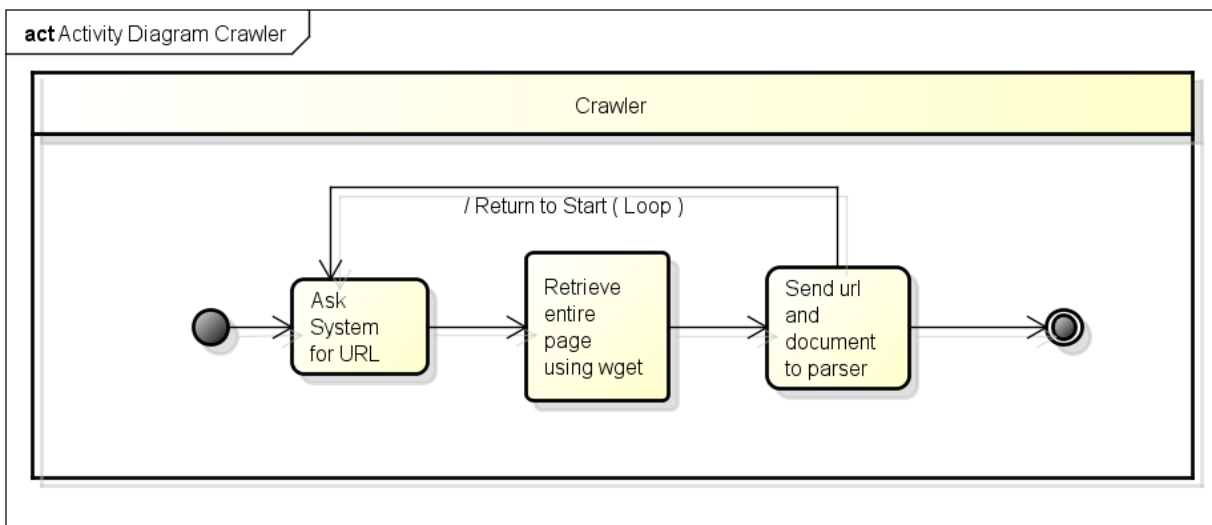
Sequence diagrams

Use Case 2: Crawler

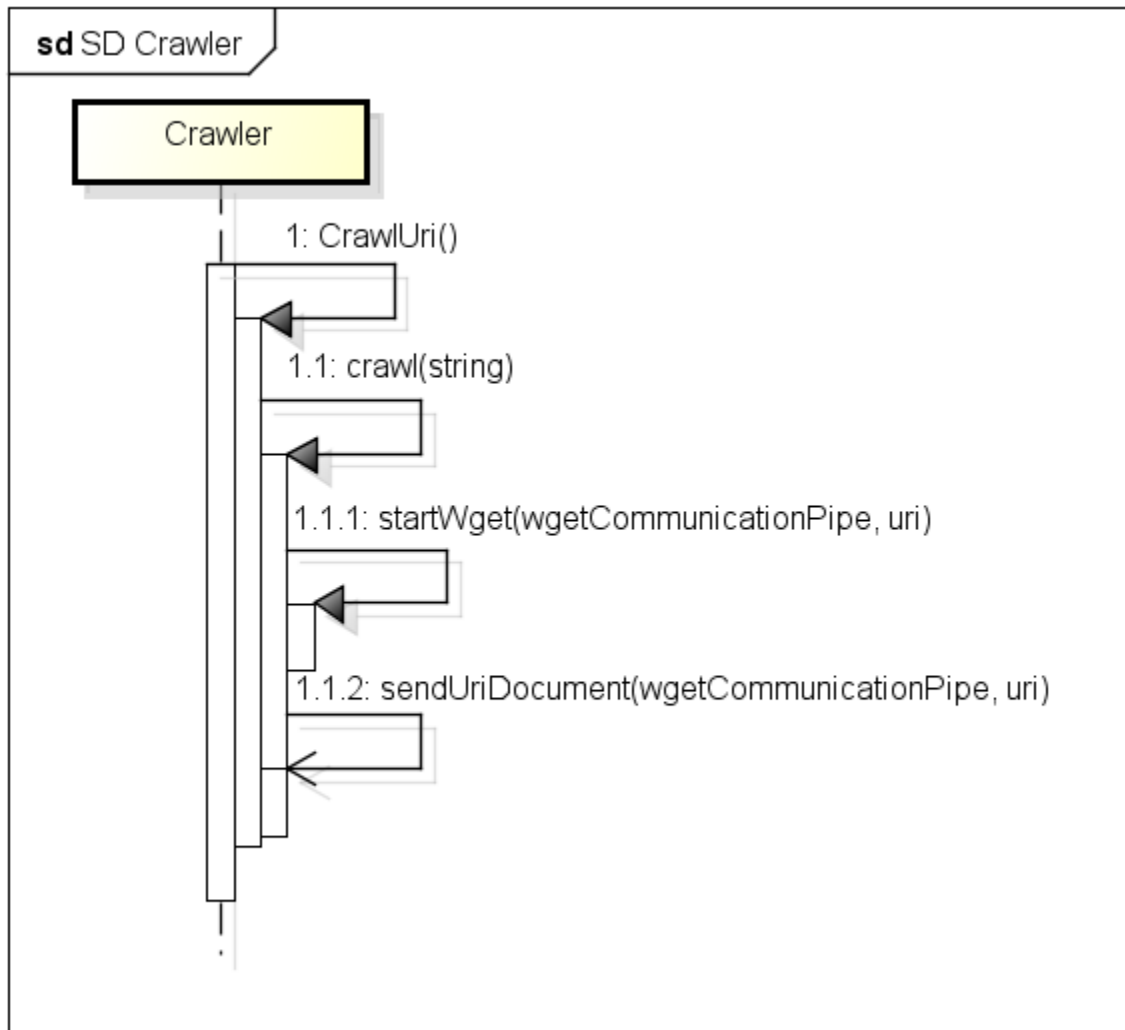
The crawler asks the master for an URI. The master then gets an URI from the URI queue and sends this to the crawler. The crawler then uses WGET to get the data from the URI. This data and the URI will be sent back to the master.

Primary actor: Crawler	
Stakeholders & interest: Scheduler, Parser	
Preconditions: Connection to the master, connection to the internet	
Main success Scenario	
1. Crawler asks master for a URI to crawl.	2. The master gets an URI from the URI queue.
	3. The master sends an URI to the crawler.
4. Crawler gets the data from the URI using WGET.	
5. Crawler sends the URI and the crawled data to the master.	
Extensions: (or Alternative flow)	

Activity diagram



Sequence Diagram

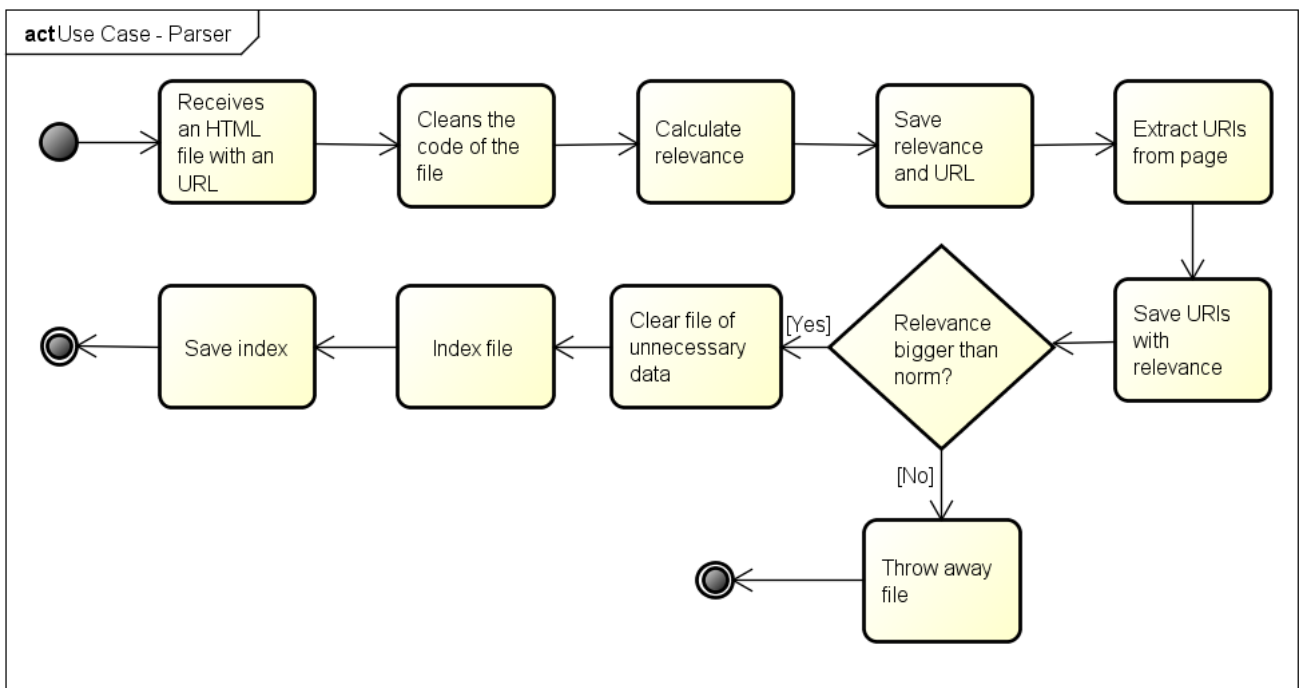


Use Case 3: Parsen

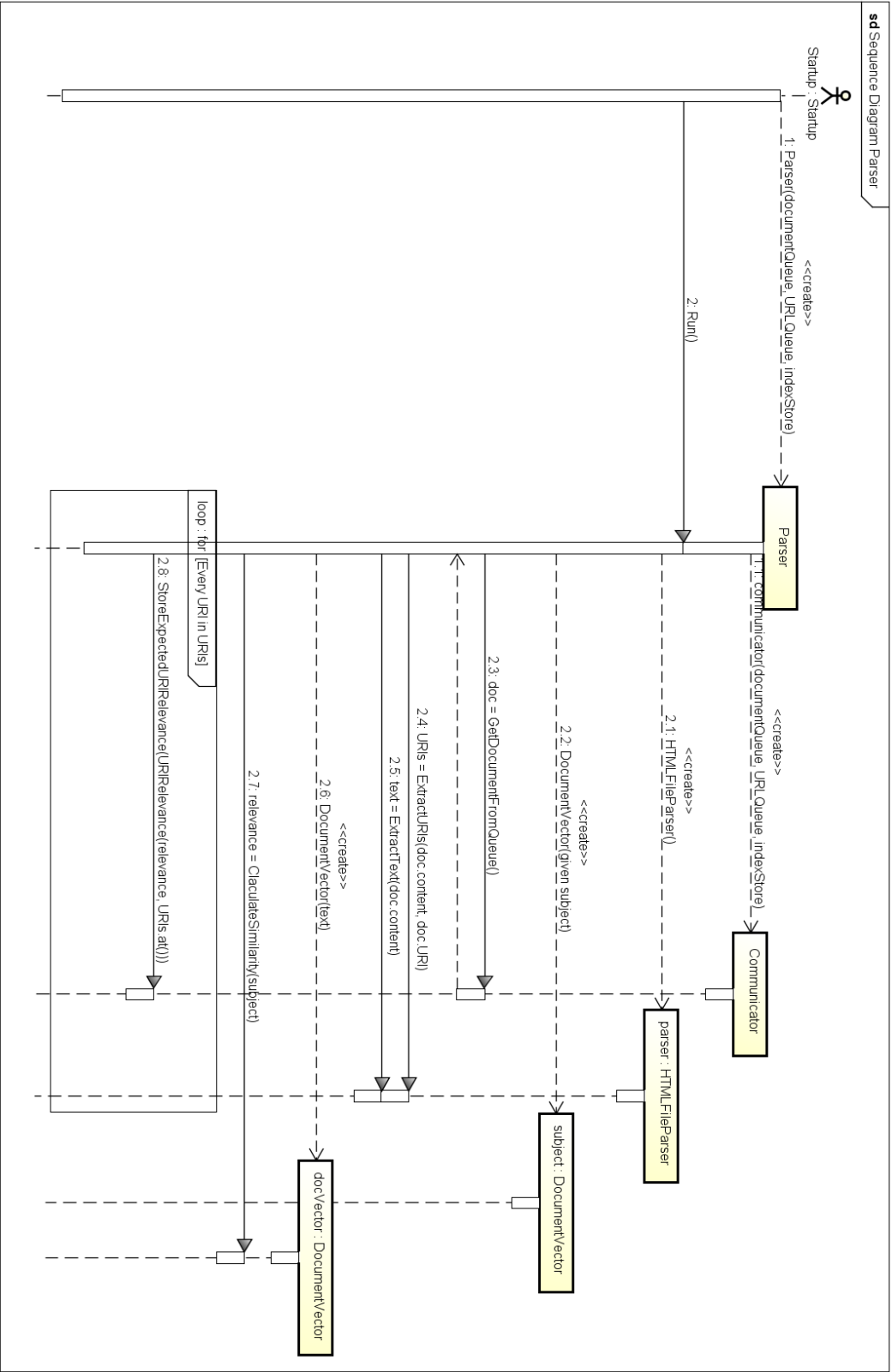
The parser requests a document and the related URI from the master. The parser then extracts all the URI's from the document. The parser then removes the html tags. The parser is now ready to determine the relevance of the document. After this is done the parser sends the found URI's to the master.

Primary Actor: Parser	
Stakeholders: Master, Crawler, Database	
Preconditions: Connection to the database and the crawler, file is HTML format	
Post conditions: File is indexed and the index, URIs and relevance is stored	
Main success scenario:	
1. The parser requests a document from the master.	2. The master send the URI and document to the parser.
3. The parser extracts the URI's from the document.	
4. The parser removes the html tags.	
5. The parser determines the relevance of the document.	
6. The parser sends the URI's with the relevance to the master.	
Extensions: (or Alternative flow)	

Activity diagram



Sequence diagrams



Use Case 4: Master

Get URI

Primary Actor: Crawler	
Stakeholders: Crawler	
Preconditions: System is in waiting state.	
Post conditions: De Crawler has an URI. The URI queue has an element less.	
Main success scenario:	
1. Crawler asks for an URI.	2. System gets the first element of the URI queue.
	3. System sends the URI to the crawler.
	4. System removes the URI from the URI queue.
Extensions (Alternative flow)	
	[No element in queue] 2. System gets an URI of the default Queue (start point).

Put URI

Primary Actor: Parser	
Stakeholders: Parser	
Preconditions: System is in waiting state.	
Post conditions: The URI queue has one more element.	
Main success scenario:	
1. Parser sends an URI to the system.	2. System checks if the URI exists in the queue.
	3. [URI is Unique] System put URI in the queue.
Extensions (Alternative flow)	
	[URI already exists in URI queue] 3a. System does nothing with the URI.

Get Document

Primary Actor: Parser	
Stakeholders: Parser	
Preconditions: System is in waiting state.	
Post conditions: The parser has a document from the document queue. The document queue has one element less.	
Main success scenario:	
1. Parser asks for a document	2. System gets the first document from the document queue.
	3. System sends the document to the parser.
	4. System removes the document from the document queue.
Extensions (Alternative flow)	
	[No document in document queue] 2a. System waits until the document queue has a new document.

Put Document

Primary Actor: Crawler	
Stakeholders: Crawler	
Preconditions: System is in waiting state.	
Post conditions: The document queue has one more element.	
Main success scenario:	
1. Crawler send a document to the system.	2. System checks if the document exists in the document queue.
	3. [document is unique] Systeem plaatst Document in de queue.
Extensions: (or Alternative flow)	
	[document already exists in document queue] 3a. System does nothing with the document.

State diagram

