Use Cases

for

Text Analysis API / News API

Version 1.0

Prepared by rkearns@bu.edu

5 October 2016

# Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

# Use Case Identification

## Use Case ID

1.0

## Use Case Name

Search for a specific news outlet.

## Use Case History

Initial Write-up

### Created By

Ryan Kearns

### Date Created

5 October 2016

### Last Updated By

Ryan Kearns

### Date Last Updated

5 October 2016

# Use Case Definition

## Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

## Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

## Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.

## Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5.

## Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form “X.Y.E.Z”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

## Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

## Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## Business Rules

List any business rules that influence this use case.

## Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

## Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

## Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case List

|  |  |  |
| --- | --- | --- |
| ID | Primary Actor | Use Case Title |
| 1 | User | Search for specific key word / sentiment |
| 2 | User | Search for specific news outlet |
| 3 | User | Display home page / news feed |
| 4 | User | Display / update user preferences |
| 5 | User | User Log In |
| 6 | User | Display search results |

Use Case Template

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1.0 | | |
| Use Case Name: | Search for specific key word / sentiment | | |
| Created By: | Ryan Kearns | Last Updated By: | Ryan Kearns |
| Date Created: | 5 October 2016 | Date Last Updated: | 5 October 2016 |

|  |  |
| --- | --- |
| Actors: | User |
| Description: | Search for specific key word / sentiment |
| Trigger: | A key word or sentiment is entered into the search function on the search page |
| Preconditions: | 1. The user must be on the search page 2. The user must be logged in to the application 3. Establish a connection to the text analysis API / news API |
| Postconditions: | 1. Store the results from our query into a database 2. Close the connections to both APIs 3. Assign a sentiment to the article based on Watson API result |
| Normal Flow: | 1. User logs in to the web application 2. User is redirected to the search function page 3. User enters a query for a specific key word or sentiment 4. Display JSON results on results page, formatted into HTML 5. Only display headline of the article and the news source that it is from in a list-like structure on the page |
| Alternative Flows: | 1. The user can use the alternative search function located in a navigation bar on the application |
| Exceptions: | 1.0.E.0 - No Results  1.0.E.1 - Too many results for one page  1.0.E.2 - If the user is searching for a sentiment, the sentiment must be a valid Watson sentiment  1.0.E.3 - Loss of connection |
| Includes: | 1.6 - Display search results |
| Priority: | High |
| Frequency of Use: | 10 uses per hour |
| Business Rules: | N / A |
| Special Requirements: | Display search results in < 2 seconds so that the user does not leave the site |
| Assumptions: | Watson API can handle entire news articles at once |
| Notes and Issues: | Determine maximum number of API calls for individual news sites as well as the Watson Text Analysis API – Ryan 10/6/16  Determine best way to format display results - Nelson 10/6/16  Determine min and max number of results to display – Jarvis 10/6/16 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial write-up | 10/5/16 | Initial write-up | 1.0 |
|  |  |  |  |

Short paragraph:

The primary use cases for our app are: searching for a keyword, searching for a specific news outlet, displaying the home page / news feed page, displaying the user preferences, logging in, and displaying search results. The case we are focusing on is the keyword search. The user will enter a key word. Next, we will query the news outlet APIs for relevant articles pertaining to the key word. The results will be parsed from JSON to HTML and displayed on the results page for the user to see. The user can click on a result and be taken to the article. If there are no results for the specific key word, an exception is raised and the user will be prompted for another keyword.

Git at https://github.com/ryan-kearns/cs411