**Use Case Document**

**Reddit Crawler**

**Michael Nollette**

**Tim Thor**

**04/18/2019**

**Table of Contents**

[Introduction 3](#_Toc381699876)

[Explanation of Contents 3](#_Toc381699877)

[Use Cases 4](#_Toc381699878)

[Add Email 4](#_Toc381699876)

[Set Subreddit 5](#_Toc381699877)

[Add Search Criteria 6](#_Toc381699878)

[Clear Search Criteria 7](#_Toc381699878)

[Listen 8-9](#_Toc381699878)

# 

# Introduction

This Use Case document provides information on how RedditCrawler, as well as all of its actors, are utilized to create a functional result. Each Use Case will provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome. This document’s purpose is to bring together this system’s business and design needs in order to ensure that the completed system achieves the goals established in the Concept Analysis and High Level Product Description.

# 

# Explanation of Contents

Name of Use Case: Provide a short name for the use case which lends itself to the objective of the system.

Description: Provides a description of both the reason for using the use case and the expected outcome of the use case.

Actors: List of actors, who are the people who will be initiating the system described in the use case.

Precondition: Describes any conditions that must be true or activities that must be completed prior to executing the use case.

Postcondition: Describes the state of the system at the conclusion of the use case.

Flow: Describes all actions of the user and the expected system responses for planned normal execution of the use case. Flow is sequential and provides adequate detail to understand all user actions and system responses.

Alternative Flows: Where applicable, detail any flows that vary or special extensions or conditions which are separate from the main flow but also necessary.

Exceptions: When use cases are executed, there may be various conditions which result in errors. This section should describe any errors that may result during use case execution and how the system will react or respond to those errors.

Requirements: This section describes any non-functional or special requirements for the system as the use case is executed. Since the application is working with a library that interacts with the Reddit API, application users are subject to Reddit’s terms and conditions when operating it.

# 

# Use Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Add Email | | | |
| **Created By:** | Michael Nollette | | **Last Updated By:** | N/A |
| **Date Created:** | 04/18/2019 | | **Last Revision Date:** | N/A |
| **Description:** | | User submits new email address to receive notifications from application. | | |
| **Actors:** | | User | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | 1. If nonexistent before running, rcEmail.txt now exists. 2. rcEmail.txt now contains user email address for future read operations. 3. User is returned to main menu. | | |
| **Flow:** | | 1. User selects “email change” option from main menu. 2. User inputs email address when prompted. 3. System queries entry, confirms valid email address format. 4. System adds user email address to rcEmail.txt 5. System returns user to main menu. | | |
| **Alternative Flows:** | | In step 4 of the normal flow, if rcEmail.txt does not exist:   1. System creates rcEmail.txt in default directory. 2. Return to main flow at step 4. | | |
| **Exceptions:** | | In step 1 of the normal flow, if user inputs an invalid menu option:   1. Halt operation. 2. Inform user that the input is invalid. 3. Resume use case at step 1 of normal flow.   In step 3 of the normal flow, if user inputs an invalid email format:   1. Halt operation. 2. Inform user that the format is invalid. 3. Resume use case at step 2 of normal flow. | | |
| **Requirements:** | | No nonfunctional requirements are present for use case ‘Add Email’ | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Set Subreddit | | | |
| **Created By:** | Michael Nollette | | **Last Updated By:** | N/A |
| **Date Created:** | 04/18/2019 | | **Last Revision Date:** | N/A |
| **Description:** | | User submits subreddit details for monitoring during normal application operations. | | |
| **Actors:** | | User | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | 1. If nonexistent before running, rcSubreddit.txt now exists. 2. rcSubreddit.txt now contains subreddit name in string format for future read and monitoring operations. 3. User is returned to main menu. | | |
| **Flow:** | | 1. User selects “set subreddit” option from main menu. 2. User inputs Subreddit when prompted. 3. System queries entry, confirms string format with no blank spaces. 4. System pings subreddit URL to confirm existence. 5. System adds subreddit string to rcSubreddit.txt 6. System returns user to main menu. | | |
| **Alternative Flows:** | | In step 5 of the normal flow, if rcSubreddit.txt does not exist:   1. System creates rcSubreddit.txt in default directory. 2. Return to main flow at step 5. | | |
| **Exceptions:** | | In step 1 of the normal flow, if user inputs an invalid menu option:   1. Halt operation. 2. Inform user that the input is invalid. 3. Resume use case at step 1 of normal flow.   In step 3 of the normal flow, if user inputs an invalid subreddit format:   1. Halt operation. 2. Inform user that the format is invalid. 3. Resume use case at step 3 of normal flow.   In step 4 of the normal flow, if URL fails to return a successful ping:   1. Check internet connection by pinging google.com. 2. If no connection, inform user and return to step 1. 3. If connection is present, inform user the subreddit does not exist, and return to step 2. | | |
| **Requirements:** | | No nonfunctional requirements are present for use case ‘Set Subreddit’ | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Add Search Criteria | | | |
| **Created By:** | Michael Nollette | | **Last Updated By:** | N/A |
| **Date Created:** | 04/18/2019 | | **Last Revision Date:** | N/A |
| **Description:** | | User appends a new search term for monitoring. | | |
| **Actors:** | | User | | |
| **Preconditions:** | | None | | |
| **Postconditions:** | | 1. If nonexistent before running, rcSearchCriteria.txt now exists. 2. rcSearchCriteria.txt now contains new search term for normal monitoring operations. 3. User is returned to main menu. | | |
| **Flow:** | | 1. User selects “change search criteria” option from main menu. 2. User selects “add new search term” option from search menu. 3. User inputs search term. 4. System queries entry. 5. System adds search criteria string to rcSearchCriteria.txt 6. System returns user to main menu. | | |
| **Alternative Flows:** | | In step 5 of the normal flow, if rcSearchCriteria.txt does not exist:   1. System creates rcSearchCriteria.txt in default directory. 2. Return to main flow at step 5. | | |
| **Exceptions:** | | In step 1 of the normal flow, if user inputs an invalid menu option:   1. Halt operation. 2. Inform user that the input is invalid. 3. Resume use case at step 1 of normal flow. | | |
| **Requirements:** | | No nonfunctional requirements are present for use case ‘Add Search Criteria’ | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Clear Search Criteria | | | |
| **Created By:** | Michael Nollette | | **Last Updated By:** | N/A |
| **Date Created:** | 04/18/2019 | | **Last Revision Date:** | N/A |
| **Description:** | | User clears existing search criteria from stored document. | | |
| **Actors:** | | User | | |
| **Preconditions:** | | * rcSearchCriteria.txt exists, and contents are not null. | | |
| **Postconditions:** | | 1. If nonexistent before running, rcSearchCriteria.txt now exists. 2. rcSearchCriteria.txt contents are null. 3. User is returned to main menu. | | |
| **Flow:** | | 1. User selects “change search criteria” option from main menu. 2. User selects “clear search criteria” option from search menu. 3. User confirms menu selection. 4. System confirms existence of rcSearchCriteria.txt and its contents. 5. System clears all lines from rcSearchCriteria.txt 6. System returns user to main menu. | | |
| **Alternative Flows:** | | In step 4 of the normal flow, if rcSearchCriteria.txt does not exist:   1. System creates rcSearchCriteria.txt in default directory. 2. Inform user file did has been created & is blank. 3. Return to main menu. | | |
| **Exceptions:** | | In step 1 of the normal flow, if user inputs an invalid menu option:   1. Halt operation. 2. Inform user that the input is invalid. 3. Resume use case at step 1 of normal flow.   In step 4 of the normal flow, if rcSearchCriteria.txt contents are null:   1. Halt operation. 2. Inform user that the file is empty. 3. Return user to main menu. | | |
| **Requirements:** | | * rcSearchCriteria.txt exists and contents are not null. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name of Use Case:** | Listen | | | |
| **Created By:** | Michael Nollette | | **Last Updated By:** | N/A |
| **Date Created:** | 04/18/2019 | | **Last Revision Date:** | N/A |
| **Description:** | | Application default setting; runs application in standby mode, pinging the designated subreddit every three minutes for matches on the search criteria. If found, notify user. | | |
| **Actors:** | | User | | |
| **Preconditions:** | | * rcSearchCriteria.txt, rcSubreddit.txt, and rcEmail.txt exist and contents are not null. * rcEmail.txt contains a valid email address. * rcSearchCriteria.txt contains at least one search term. * rcSubreddit.txt contains a string-based subreddit to ping. | | |
| **Postconditions:** | | 1. If rcMatchRecords.txt did not exist, it is created. 2. rcMatchRecords.txt contents now include the URLs of all posts that matched the search criteria during runtime. 3. User is notified via email of any post matches. 4. Application is now running in listen mode. | | |
| **Flow:** | | 1. User selects “Run application in Listen mode” from main menu. 2. System checks for existence and contents of rcSearchCriteria.txt, rcEmail.txt, rcSubreddit.txt, and rcMatchRecords.txt. 3. System performs final validation checks:    1. Email format    2. Subreddit ping    3. Search contents null check 4. System stands by in listen mode. 5. System receives match in subreddit ping. 6. System checks match against existing records in rcMatchRecords.txt. 7. System posts email to user with URL and details of post. 8. System updates rcMatchRecords.txt with exact URL of match. 9. System returns to listen mode. | | |
| **Alternative Flows:** | | In step 2 of the normal flow, if rcMatchRecords.txt does not exist:   1. Create rcMatchRecords.txt in the default directory 2. Return to step 2 of the normal flow.   In step 6 of the normal flow, if a result matches an entry in rcMatchRecords.txt:   1. Break from match evaluation without notifying user. 2. If more matches exist, proceed to cross-check them. 3. When there are no more matches, return to step 4 of the normal flow. | | |
| **Exceptions:** | | In step 1 of the normal flow, if user inputs an invalid menu option:   1. Halt operation. 2. Inform user that the input is invalid. 3. Resume use case at step 1 of normal flow.   In step 2 of the normal flow, if rcEmail.txt, rcSubreddit.txt, or rcSearchCriteria.txt do not exist:   1. Halt operation. 2. Inform user of what files need to be created. 3. Return user to main menu.   In step 3 of the normal flow, if rcEmail.txt, rcSubreddit.txt, or rcSearchCriteria.txt have null contents:   1. Halt operation. 2. Inform user of what files need to be given contents. 3. Return user to main menu. | | |
| **Requirements:** | | * rcSearchCriteria.txt, rcEmail.txt, and rcSubreddit.txt all exist and have valid contents. | | |