Les Petits Plats
Functionality Investigation Sheet

Feature: Search for recipes containing keyword
Problem: Needs to be efficient and fast to be run on client side

Option 1:
Dictionary Approach:
Preprocess data on load (or server side) and store all possible three letter combinations from keywords contained in the recipes into a dictionary. Directly read the related entry after search input.

Benefits:
Lightning fast (4 microseconds) for actual search
Very scalable for big datasets

Disadvantages:
Overhead for small datasets (no noticeable difference in performance)
Code is less flexible

Option 2:
Recursive Crawler:
Takes any kind of data(array) and compares entries to the search input on runtime, returning references to matches. Recursively calls itself if confronted with a nested object.

Benefits:*
Very flexible in its use case
Very robust and efficient code

Disadvantages:
Runs through whole dataset on each search input
Not scalable for big datasetsFunctionality Investigation Sheet

Conclusion:

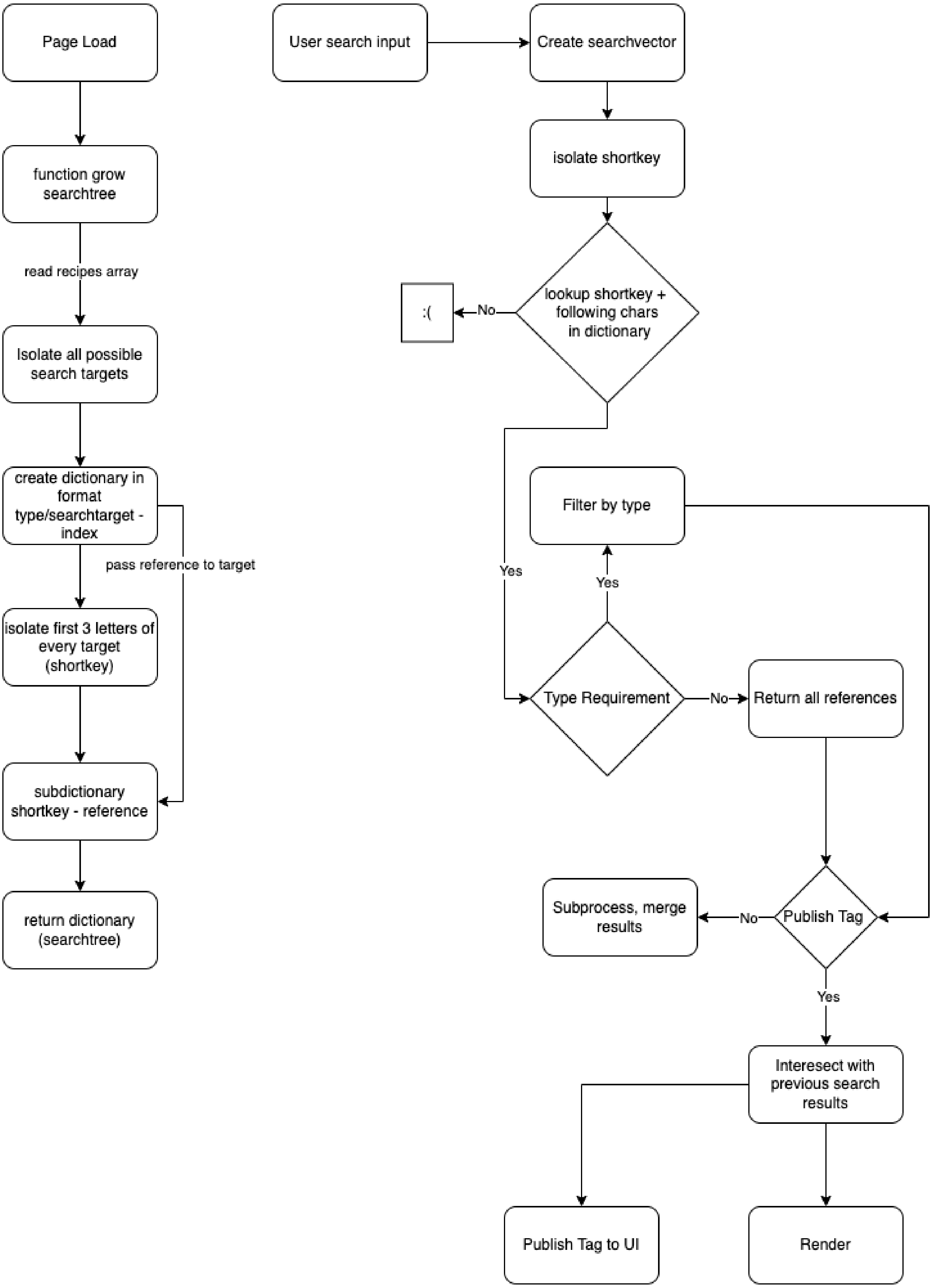
Algorithm 2 for this use case and dataset size, Algorithm 1 for substantially bigger datasets

Link to Github Repo:

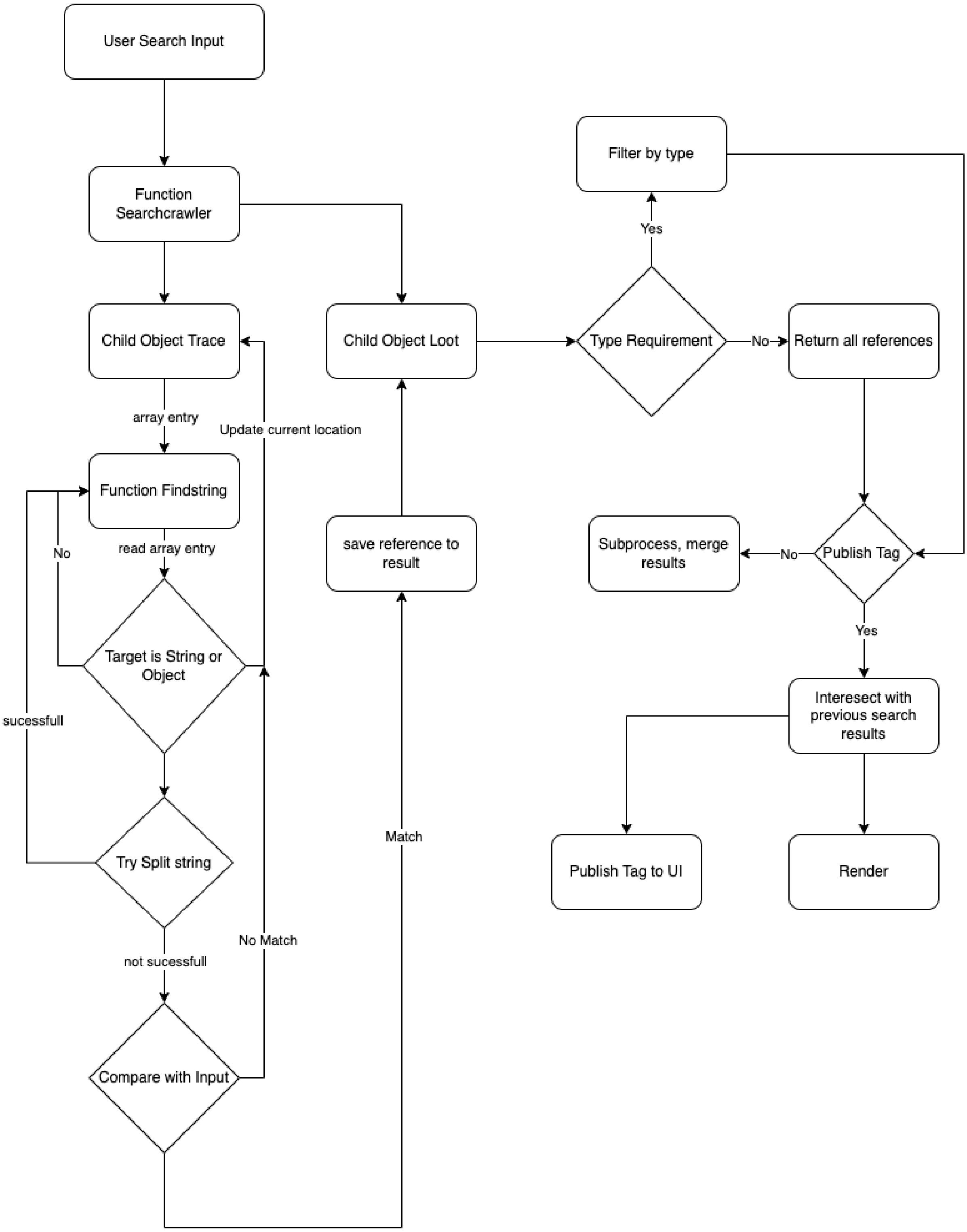
https://github.com/TomOeggl/p7.Searchtool

Annexes:

Option 1


Page Load
function grow searchtree
read recipes array
Isolate all possible search targets
create dictionary in format type/searchtarget - index
pass reference to target
isolate first 3 letters of every target (shortkey)
subdictionary shortkey - reference
return dictionary (searchtree)
User search input
Create searchvector
isolate shortkey
lookup shortkey + following chars in dictionary
No
:(
Yes
Filter by type
Yes
Type Requirement
No
Return all references
Subprocess, merge results
No
Publish Tag
Yes
Interesect with previous search results
Publish Tag to UI
Render

Option 2


User Search Input
Function Searchcrawler
Child Object Trace
array entry
Function Findstring
read array entry
No
Target is String or Object
sucessfull
Try Split string
not sucessfull
Compare with Input
No Match
Update current location
Child Object Loot
save reference to result
Match
Type Requirement
Yes
Filter by type
No
Return all references
Publish Tag
No
Subprocess, merge results
Yes
Interesect with previous search results
Publish Tag to UI
Render