A blue sign with white text

Description automatically generated with low confidence

# DARWIN EU® Coordination Centre

# EMA/2021/08/TDA

Software Requirement Specification for the CodelistGenerator R package

|  |  |  |
| --- | --- | --- |
| Lead contributor |  | Edward Burn |
| **Lead contributor email** |  | e.burn@darwin-eu.org |
|  |  |  |
| **Other contributors** |  | p.rijnbeek@darwin-eu.org  ….... |
| **Date** |  | 07/09/2022 |
| **Study Type** |  | N/A |
| **Dissemination level** |  | Confidential |

# Table of contents

[Table of contents 2](#_Toc113182099)

[Document History 3](#_Toc113182100)

[Glossary 4](#_Toc113182101)

[Executive Summary 5](#_Toc113182102)

[1 Scope 5](#_Toc113182103)

[2.1 Purpose 5](#_Toc113182104)

[2.2 User Characteristics 6](#_Toc113182105)

[2.3 Constraints 6](#_Toc113182106)

[2.4 Package Perspective 7](#_Toc113182107)

[2. Requirements 7](#_Toc113182108)

[2.1 Functional Requirements 7](#_Toc113182109)

[2.1.1 Identify a candidate code list based on a specified search strategy 7](#_Toc113182110)

[2.1.1 Show the mappings from non-standard concepts to standard concepts 7](#_Toc113182111)

[3.2 Non-functional Requirements 8](#_Toc113182112)

[3.2.1 Exectution time 8](#_Toc113182113)

[3.2.2 Memory usage 8](#_Toc113182114)

[4. Approval 8](#_Toc113182115)

[5. References 8](#_Toc113182116)

# Document History

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| V0.1 | 05/09/2022 | First Draft for internal and external consultation |
| VX.X | DD/MM/YYYY | Comments |
| VX.X | DD/MM/YYYY | Draft |
| VX.X | DD/MM/YYYY | Final Version |

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SDLC | Software Development Life Cycle. A methodology for designing, creating, and maintaining software. |
| SRS | Software Requirement Specification. A document describing all the requirements that will drive the development of the R Package. |
|  |  |
|  |  |
|  |  |
|  |  |

# Executive Summary

This document provides the Software Requirements Specification (SRS) for the CodelistGenerator R Package. The purpose of this package is to provide functionality to generate a list of candidate codes to identify a particular clinical event of interest by querying the standardized vocabularies. These candidate codes can then be assessed by the clinical experts for inclusion as part of a cohort definition.

# 1 Scope

## 1.1 Purpose

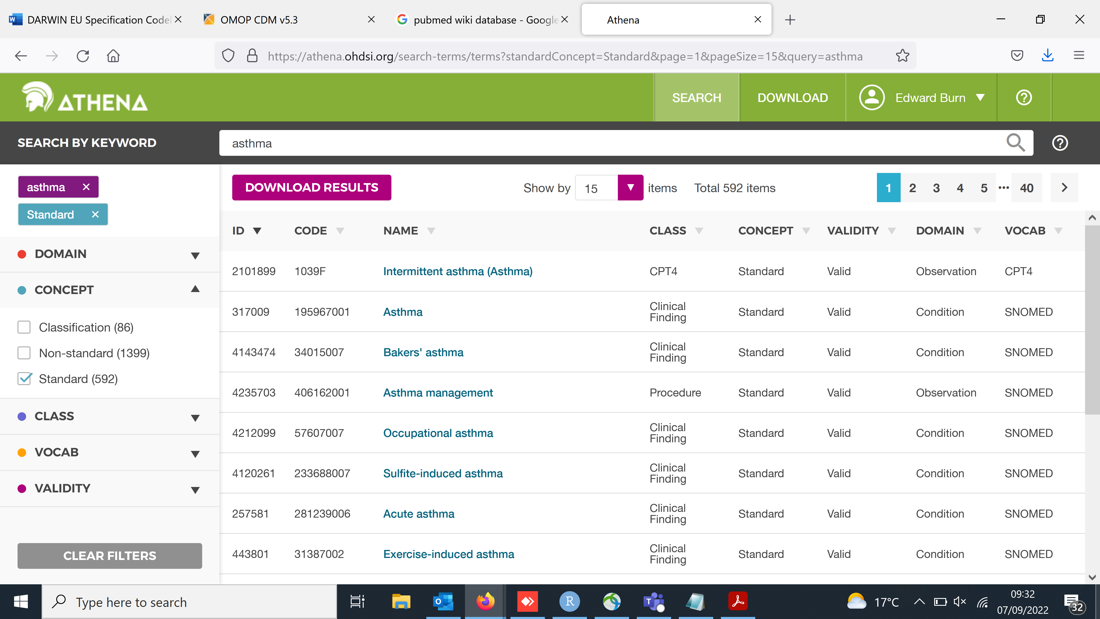
Data sources used in DARWIN EU® are standardised to the OMOP Common Data Model (OMOP-CDM). This aims to improve the syntactic interoperability by standardizing the structure and semantic interoperability by mapping to the Standard Concepts in the Standardized Vocabulary.

The vocabulary tables in the OMOP-CDM include the concept, concept hierarchy, concept relationship, concept synonym, and vocabulary table. More information on each of these tables can be found at <https://ohdsi.github.io/CommonDataModel/cdm53.html#Vocabulary_Tables>

Clinical phenotype definitions will be needed for all studies performed in DARWIN EU®. These definitions express the logic used to identify patients that experience the event of interest, the date of occurrence, and the date at which it ended. A clinical phenotype can consist of one or multiple concept sets that identify components of the logical algorithm. For example, a phenotype of Asthma can be identified by a logical combination of two components: a condition occurrence of Asthma or the use of Asthma medication. For each of these two components concept lists need to define to identify patients correctly. For example, the Asthma component can have an associated concept set containing concepts such as “asthma unspecified”, “bronchial asthma”, “severe persistent asthma”, and “mild intermittent asthma”. The inclusion or exclusion of concepts in the concept set is driven by the clinical definition of the event of interest. Such consideration can be informed by review of the codes by domain experts supported by additional data analyses (e.g., to describe how frequently a given code is used in data source, the characteristics of patients with such a code, and so on).

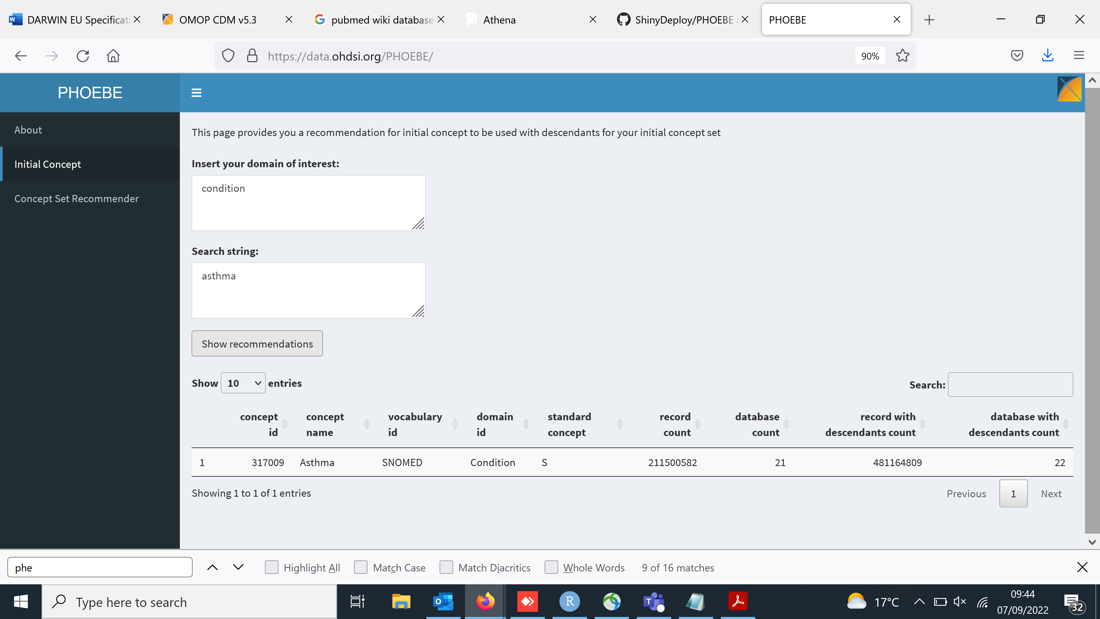
A candidate code list should be created following a transparent, reproducible, and systematic approach, i.e. a structured process. This candidate code list should contain all the potential codes of interest for the domain expert since the final selected concept set is a subset of this list.

Currently, there are a number of tools available from the OHDSI community that can help in identifying candidate code lists based on the OMOP-CDM. For example, ATLAS (<https://github.com/OHDSI/Atlas>) and ATHENA (<https://athena.ohdsi.org>) provide the ability to run a key word search and will return a list of possible matches. See Figure 1 below for an example search of asthma in ATHENA, restricting to standard concepts. Although extremely useful, these tools do not allow for search strategies of keywords under varying logic (for example whether to use the concept synonym table or not when performing search strategies, whether to search for multiple key words using OR logic, and so on).

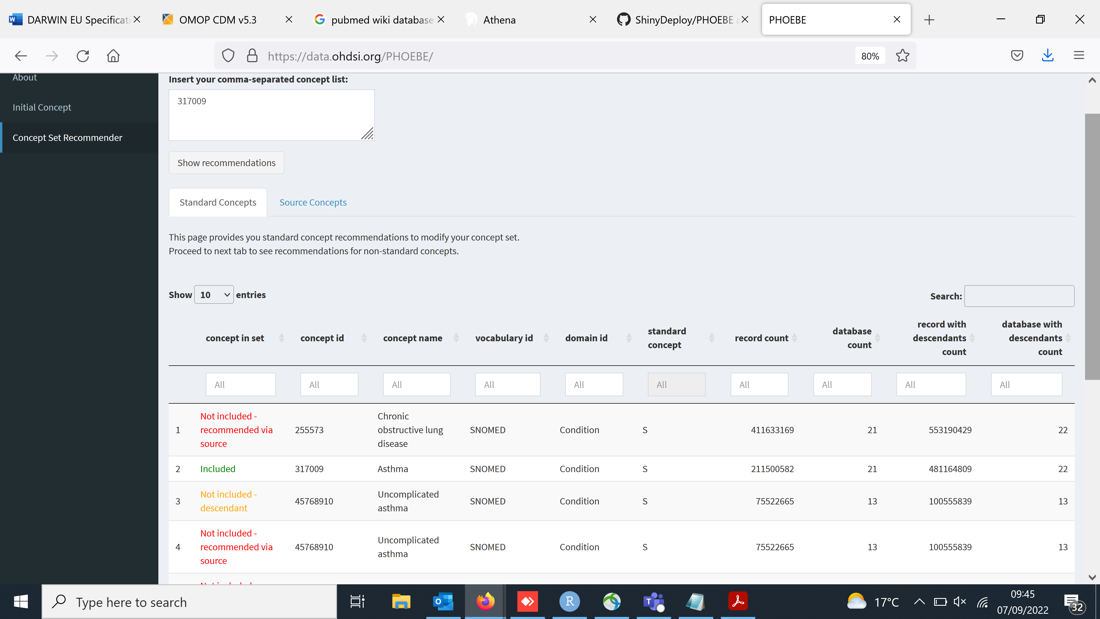


**Figure 1. Search for “asthma” in ATHENA**

In addition to the above tools, the OHDSI community has also developed the PHenotype Observed Entity Baseline Endorsements (PHOEBE) tool. This is an RShiny app available at <https://data.ohdsi.org/PHOEBE/> (with its source code at <https://github.com/OHDSI/ShinyDeploy/tree/master/PHOEBE>). PHOEBE allows the user to first find an initial concept set based on a key word search, see Figure 2, and then using such a concept (or multiple concepts), it will provide a recommended list of concepts that could be used. The search is based on text search and the results from the Concept Prevalence OHDSI study (<https://github.com/ohdsi-studies/ConceptPrevalence>) where counts of the standard and source concepts were recorded for a subset of databases standardised to the OMOP CDM (but at the current time are not publicly available). This tool has proved valuable for previous research, allowing for a more systematic search than could be done with ATLAS or ATHENA alone. A limitation of the tool is only available as an RShiny app, not an R package, and so users cannot use it interactively with a database that has the version of the OMOP CDM vocabularies that will be used in their study ). Additionally, as the tool has not been updated for the last 2 years, this means that the tool appears to be using outdated Standardized Vocabularies. Furthermore, for DARWIN EU® we need to have information about databases included in our own network and not all these data sources are included in the Concept Prevalence Study. Finally, performance issues have been identified when searches have been performed for multiple concepts.



**Figure 2. Search for “asthma” initial concept in PHOEBE**



**Figure 3. Search for “asthma” concept set in PHOEBE**

The CodelistGenerator R package is intended to supplement the above existing tools to define clinical phenotypes. In particular, it aims to facilitate the systematic generation of candidate code lists based on vocabularies currently being used in a database. The use of the tool with the use of it recorded, for example using RMarkdown, will allow for the provenance of code lists to be better understood and help ensure reproducibility for a central input to research studies – the definitions of clinical events of interest.

## 1.2 User Characteristics

The CodelistGenerator R package will be used by researchers involved in defining clinical phenotypes. In collaboration with clinical domain experts, who will help define an appropriate search strategy, the data analyst will execute the search against a specific version of the OMOP CDM vocabularies. The data analyst can then provide the candidate code list to the clinical expert for further screening, as well as using them as the starting point for further data driven development (for example, using the CohortDiagnostics package: <https://ohdsi.github.io/CohortDiagnostics/>).

As well as users interacting directly with the CodelistGenerator R package, users could also potentially interact with it via a web application (for example created using RShiny). The design of such a web application is out of scope of this document, but its creation would allow users without familiarity with R to still make use of the functionality provided by the CodelistGenerator R package.

**User stories**

“I am a data analyst and often tasked with helping to create clinical phenotypes. There are very helpful tools for later in the process, like the CohortDiagnostics package, but it is difficult to follow a systematic process at the start. ATHENA is a great way of quickly finding codes that might relate to a clinical idea, but it is not designed to use in a systematic way”

“…….”

“….”

“Codelist generator is a useful tool as it allows for a reproducible and transparent way to show how clinical phenotypes for my studies can be generated which is not possible with Athena.”

“….”

## 1.3 Constraints

1. The R package assumes that data is available in the OMOP CDM version >=5.3
2. The R package should work for R version 4.2.1 or newer.
3. The R package will not query patient-level data, only the Standardized Vocabularies.
4. The R package should at a minimum work with the following database management systems: Postgres, SQL Server, Snowflake

## 1.4 Package Perspective

CodelistGenerator has to be an independent R package used to help define Phenotypes as input to the analytic pipelines used in DARWIN EU® studies. The analytic pipelines themselves will not have CodelistGenerator as a dependency.

# ****2. Requirements****

## 2.1 Functional Requirements

### 2.1.1 Identify a candidate code list based on a specified search strategy

The CodelistGenerator R package will provide the means for identifying candidate codes for a concept set. That is, the codes that are under consideration for inclusion. Based on a search strategy specified by the user, a query against the OMOP CDM vocabulary tables will produce a list of concepts that can then be reviewed. The search functionality has the following requirements:

* Keywords – A set of search terms to use. Each of these may be multiple words, in which case they should not be required to be seen in the same order. For example, “osteoarthritis of the knee” should be considered as a candidate code in a search of “knee osteoarthritis”. These search terms will be combined using OR logic, that is a search of [“knee osteoarthritis”; “hip osteoarthritis”] would still return “osteoarthritis of the knee”.
* Excluded – A set of terms to be excluded.
* Exact or fuzzy matching – Whether the user wants to allow for some difference in spelling when running a search (for example, so that a search for “leukemia” also picks up “leukaemia”
* Which domains to search – Whether to look in a subset of OMOP CDM domains. For example, a search of candidate codes for “asthma” with only codes in the condition domain.
* Whether to use the OMOP CDM synonym table – The synonym table could be used to supplement searches
* Whether to search via non-standard vocabularies – Mappings from non-standard to standard could also help identify the standard concepts that represent a particular clinical idea
* Whether to return just standard concepts – Depending on the study a user may want to only include standard codes, or may wish to also consider including non-standard codes.
* Whether to include descendants – the user may wish to include codes lower in the hierarchy of an identified concept
* Whether to include an ancestor – the user may wish to also include codes one level higher in the hierarchy of an identified concept

The output of the search should results in a table in the following format:

### 2.1.2 Export a candidate code list

Functionality is needed to export the candidate code list in the following ways:

1. Save to a comma-separated file and a selected location
2. Store in a table in database for further processing

### 2.1.3 Show the mappings from non-standard concepts to standard concepts

As well as identifying standard codes, the domain expert may wish to also see what source concepts map to these. For example, a user defining a code list for CPRD may wish to check the mappings to standard condition codes from the non-standard READ vocabulary (which is used for CPRD).

## 3.2 Non-functional Requirements

### 3.2.1 Execution time

A simple search should be expected to return results in less than a minute, with more complex searches (e.g. of multiple domains, multiple search terms, using synonyms, and so on) possible in less than 5 minutes.

### 3.2.2 Memory usage

No more than 2 Gb of available RAM and no more that 5 Gb of free memory of local memory should be required.

# 4. Approval

The SRS is approved by the following Product Owner:

Signature

Name:

Date

# 5. References