Technical University of Crete

Department of Electronic and Computer Engineering

REQUIREMENTS ANALYSIS FOR MEDICAL INFORMATION SYSTEMS DESIGN

by

RENA PERAKI

Chania, March 2008

Abstract

The World Wide Web (the Web) is an increasingly popular source for health information. Also, vast amounts of health information are available in medical data bases. Making this information available in a concise and uniform way is of crucial importance to individual users or organizations. Users of the medical domain can be either health care professionals (*experts*) or consumers (*novice users).* Consumers searching for medical information usually issue a natural language query and need to retrieve medical documents on a topic of interest that is easy to read and comprehend (e.g. medical documents that don't contain complex medical terminology). On the other hand, experts usually do more specialized searches (involving complex terminology) looking for state-of-the-art documents to fulfill their information needs. This thesis deals with issues related to the design and implementation of medical information systems to fulfill the need of both types of users. Medical information is acquired by web sources or by data repositories available (e.g. Medline). Acquired medical documents from authoritative (validated) sources are analyzed (though text analysis) and automatically indexed by author, type, content and association (e.g. link) information. System's functionality is described using **use cases tools** and the **Unified Modeling Language** (UML). UML case, package, activities and class diagrams are used to model users, data entities, system functionalities and their inter-relationships. Also part of the system's functionality has been implemented in a system prototype.

Περίληψη

Ο συνεχώς αυξανόμενος όγκος της ιατρικής πληροφορίας που είναι διαθέσιμος τόσο στο διαδίκτυο όσο και σε ιατρικές βάσεις δεδομένων, έχει αυξήσει σημαντικά το ενδιαφέρον για συστήματα που παρέχουν αυτή την πληροφορία. Οι χρήστες τέτοιων συστημάτων μπορεί να είναι είτε επαγγελματίες από το χώρο της παροχής υγειονομικής φροντίδας (γιατροί, φαρμακοποιοί, νοσηλευτές, φοιτητές τμημάτων ιατρικής ειδικότητας κλπ), είτε απλοί χρήστες που αναζητούν πληροφορία για κάποια προσωπική υπόθεση υγείας, για πρόληψη κλπ. Οι ανάγκες αυτών των δύο ομάδων χρηστών είναι διαφορετικές με κοινό στοιχείο τον εντοπισμό ιατρικών κειμένων από έγκυρες και αξιόπιστες πηγές. Οι απλοί χρήστες, σε αντίθεση με τους επαγγελματίες στο χώρο της υγείας, επιθυμούν να εντοπίζουν ιατρικά κείμενα σε κατανοητή γλώσσα, χωρίς πολύπλοκους ιατρικούς όρους. Σε αυτή την εργασία παρουσιάζεται ένα σύστημα παροχής ιατρικής πληροφορίας που προσπαθεί να καλύψει τις ανάγκες και των δύο διαφορετικών τύπων χρηστών. Τα εργαλεία που χρησιμοποιούνται για την καταγραφή των απαιτήσεων και τον σχεδιασμό του συστήματος είναι οι περιπτώσεις χρήσης (use cases) και τα διαγράμματα της γλώσσας μοντελοποίησης UML. Πιο συγκεκριμένα χρησιμοποιούνται τα διαγράμματα περιπτώσεων χρήσης (use case diagrams), πακέτου (packet diagrams), δραστηριοτήτων (activities diagrams) για τον σχεδιασμό των διαδικασιών που θα ακολουθηθούν στην υλοποίηση και τέλος τα διαγράμματα κλάσεων (class diagrams) για τον σχεδιασμό των δεδομένων του συστήματος. Ένα μέρος αυτής της λειτουργικότητας έχει υλοποιηθεί και παρουσιάζεται στην πρώτη έκδοση του συστήματος.

Αυτή η εργασία αφιερώνεται στη

μνήμη του αγαπημένου μου πατέρα Στέλιου …

Acknowledgements

I would like to sincerely thank my supervisor Professor Euripides Petrakis, for his assistance, support and the encouragement he showed to me. I thank also the other members of my thesis committee, Professors Stavros Christodoulakis and Michail Lagoudakis for their helpful suggestions. In addition, I am grateful to my colleague in the Intelligent Laboratory of Technical University of Crete Angelos Hliaoutakis for his collaboration and for the valuable comments he provided.

I offer special thanks to my parents Stelios and Gitsa, my husband Costis Halkiadakis and of course my children Stela, Dimitris and Stelios. I thank them for their love and encouragement in all aspects of my life.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

The amount of medical information available on the Internet is huge and it is growing every day. Recent surveys indicate that 80 percent of adult Internet users, say they have searched at least one health topic at some point [15]. Usually, the users are uncertain about their exact questions and unfamiliar with the medical domain and the medical vocabulary (called **consumers**). Of course, there are health care professional users **(**the **experts),** who use medical terminology for their search and have extra needs, such as searching the literature by following citations in articles.

To facilitate users to acquire medical information from the Web, many medical information systems have been developed, but medical information retrieval remains difficult for both consumers and experts. The problem that consumers usually face in using existing medical search engines is that the available health information is most of the times 'not friendly' to them, complex and full of medical terminology. Another problem is that when the consumer enters a medical topic, many medical web sites use similar, but not identical phrases for the same topic.

Another critical issue in designing these systems is not just to find information, but to locate the right piece of information, especially in the critical field of health, where the demand for quality in information is even more critical.

The proposed system is a medical information system that integrates the consumer's and the expert's needs in one application, providing at the same time reliable and trusted information to the users. It's functionality is described using **Unified Modeling Language** (UML). Today, UML is accepted by the Object Management Group (OMG) as the standard for modeling object oriented systems. UML has evolved from the originally adopted version 1.1 in November 1997 and widely used for modeling software systems. This thesis uses UML's latest major revision which is version 2.0.

1.1 THE SYSTEM

The system is a medical information system that, first of all, fulfills two basic requirements. First, it provides **only reliable and trustworthy sources of information**. Finding high quality medical content on the Internet can be difficult. The proposed system suggests two ways to get this kind of information. Either by using existing medical databases such as Medline, or by crawling the Web for sites accredited by the Health On the Net foundation (called HON accredited sites, *see Glossary*). Second, the system can be used effectively both by consumers and experts. To accomplish this, the system categorizes each document to consumer/expert categories, with a weight of belief that it belongs to each of these categories. The benefit of this categorization is that when the user submits a query the system responds by giving different set of answers depending on the user type, whether a consumer or expert (e.g. consumers view first the documents that have less medical terminology so they understand them easily).

The system's functionality is described using **Unified Modeling Language** (UML) which is the state-of-the-art modeling language. A brief description of this functionality follows:

- The system allows many different ways of indexing a corpus of medical documents, which in this thesis is called *medical document collection* (*see Glossary*). As mentioned earlier, this collection consists of documents that belong either to existing medical databases or to Web's HON accredited sites.
- Users must first log in to the system. They find information either by searching or by browsing. When searching, the user enters a free text query, the query is mapped to the type of terms that the user has selected before, which might be either lexicographic(single word terms) or semantic terms (AMTEx, MMTx, MeSH terms *see Glossary*) and the retrieval is performed. When browsing the (expert) user selects the type of document's categorization, he wants. The system can support three alternative ways for browsing. The user is allowed to switch between them and enhance his retrieval.
- Also, the system enhances the retrieval performance by combining various ranking schemes. The documents in the answer set are ordered by a similarity measure between each document and the user's query (the classical tf*idf

weighting scheme), the document's authority score (for articles only) and the document's weight of belief that it belongs to the consumer category (for consumer users) or to the expert category (for expert users). This ranking scheme causes the documents that are more likely to belong to the expert category to rank high in the answer set only if the user is expert. The opposite happens if the user is consumer. Retrieval performance is also enhanced is by letting the user activate the query expansion feature. The user can select the terms used for the expansion from a list of choices such as MeSH terms lower or higher in the MeSH hierarchy of a query term, synonyms of a query term etc. Optionally, the expert can filter the documents in the answer set by dropping the documents that do not belong to his favorite categories.

- The retrieval is also enhanced by allowing the users to browse the system's database via citation links. This is essential, especially for the expert users, as they can navigate the literature backward and forward in time.

1.2 CONTRIBUTIONS

It is essential for a system to have a well-designed architecture that is optimized early and incrementally. Current systems are difficult to expand when the requirements change or new requirements arise. In practice, most architectural descriptions are informal documents, usually centered on block diagrams, not supported by tools to help the programmers.

The goal of this thesis is to describe the system's requirements of a medical information system using Unified Modeling Language (UML), the current state-of-the-art language for system design. UML provides a family of diagrammatic notations by which a software system can be described at a high level of abstraction and enforces system **modularization**, by splitting the system's functionality into a collection of connected components. Each component is a replaceable part of the overall system that fulfils a clear function, evolves independently, can be reused, can be updated by alternative components. When all of these components have been implemented they are **integrated** together to form the complete system.

By using UML the **iterative and incremental development** of the system is highly characterized by:

- **transparency** : by decoupling the design from the implementation,

- **flexibility** : by offering alternative models and implementations, and
- **extensibility**: by allowing implementation of new models and algorithms without interfering with existing software e.g. new features are added by adding new code, rather than by changing the old one.

UML is supported by a variety of tools. In this work, UML diagrams were created using *EclipseUML 2007 Europa for Eclipse 3.3 by Omondo*. This tool is selected mainly because it provides **automatic code generator** for the UML class diagrams, thus saving a considerable amount of work for the system's development stage. Another benefit of this feature is that later when additional requirements arise (there is a cycle from requirements to design to code), UML diagrams are also modified, the automatic code generator is activated again, so there is always a correspondence between UML diagrams and the implemented system. As a result the system is easily maintained and enhanced.

Another benefit of the system is that, unlike other existing medical information systems, it can be used effectively to satisfy both consumer and expert users. The system allows the user to enter queries in extended length in plain English, when most other existing systems place limits on query length. This is a great convenience especially for the consumers, who due to lack of medical knowledge, have difficulties in choosing accurate medical phrases as a starting point for their search. Also, the system supports browsing the database via citation links. This feature already exists in other fields (e.g. CiteSeer in the computer science field etc), but not in the medical domain field.

1.3 THESIS STRUCTURE

The remainder of the thesis is organized as follows: First (chapter 2), some basic facts for both of the thesis fields (modeling and medical information systems), are introduced. The concept of modeling, the diagrams of UML 2.0, the four diagrams that are of special importance for this thesis, the use cases tool are explained in brief, followed by a description of tools (MMTx and AMTEx) that automatically map free text into medical terms and are used for indexing and retrieval by the system. Finally, existing medical information systems (e.g. digital libraries, medical search engines, portals etc) are explored and briefly presented.

After the basics (in chapter 3) the system's architecture is presented, followed by a detailed description of the system's functional requirements based on use cases and the four UML diagrams (package, use case, activity and class diagrams).
The last chapter (chapter 4) presents the part of the system implemented so far.
Finally a Glossary of the terms used in the requirements analysis phase is attached

Chapter 2

BACKGROUND & RELATED WORK

In this chapter, UML, which is the current state-of-the-art system modeling language, is presented. From UML only the diagrams used for the system's modeling are further described. These are the package, use case, activity and the class diagrams. Then the use cases tool, used for capturing the system's functional requirements, is presented.
The system maps both documents and queries to terms from MeSH and UMLS terminology, which are used for indexing and retrieval, so a brief description of these vocabularies follows. Then, the tools used for this mapping (U.S. National Library of Medicine MMTx and AMTEx) are also highlighted.
Finally, CiteSeer, a scientific literature digital library and several medical information systems (medical search engines, portals, medical meta search engines etc) that already exist in the Web are briefly presented.

2.1 UML (Unified Modeling Language)

Modeling, the design of a system before coding, is an essential part for every application development. System modeling is considered to be successful when the following requirements are met: the system's functionality is formally described and correct, the end-user needs are met, scalability and extensibility are supported, the system's design is visualized and checked before the coding starts, etc. **UML** is the current state-of-the-art language. It must be mentioned that UML is not just a list of diagrams, it is a **knowledge representation language**. This means that each diagram's element, e.g. box, line, rectangle, etc, is supported by certain syntax and semantics.
UML was selected for the system's modeling for several reasons. First of all it is an **industry standard**, controlled by the *Object Management Group* (OMG) an international, open membership, not-for-profit consortium of companies. Another benefit is that UML is built upon fundamental **Object Oriented concepts** (e.g. class and operation), so it helps better in describing, designing and developing particularly **object-oriented** software. Also, as mentioned earlier (*see Contributions*) UML is supported by a variety of tools that not only help the designer to 'draw' the diagrams,

they also offer advanced features, such as building a UML diagram from existing code in order to understand it (reverse-engineering) or generating program language code from a UML diagram (forward engineering). Also, it is easy to transfer a UML model from one tool into another by using the **XMI** (XML Metadata Interchange, which is another OMG standard). Other benefits in using UML is that it emphasizes **Use Case Analysis,** which is a valuable tool for capturing the process from a user's perspective (*see section 2.2*). UML also allows the designer to select the level of detail he wants in his description.

Finally, UML makes the system's design modular and easily extensible. Plugins can be added to alter the process at every stage of the system's architecture, e.g. in parsing and analyzing each individual document from the medical document collection, in categorizing each document in three different types of categories, in extracting different type of terms from both documents and queries, in writing the index data structures, in retrieval, in browsing, etc. As a result, the system is easily maintained and enhanced later even when the initial system's programmers or requirements have changed. This is essential as it is often more difficult to maintain a system than building it from the beginning.

UML 2.0 defines thirteen types of diagrams, divided into two categories:

- **Structure Diagrams** show the static structure, the 'things' that make up the system being modeled (e.g. the classes, the objects, the relationships among them, etc), irrespective of time. Structure diagrams are the following : Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram and Deployment Diagram
- **Behavior Diagrams** emphasize what must happen in the system being modeled, such as the interactions with the users, the various states of the system as it executes over time, the effects and the results of an operation or an event etc. Behavior diagrams are the following: Use Case, Activity Diagram and State Machine Diagram and the **Interaction Diagrams,** which include the Sequence Diagram, Communication Diagram, Timing Diagram, and the Interaction Overview Diagram.

Each of the diagrams mentioned above views the system from a different perspective, e.g. the **use case diagrams** show how users interact with the system, the **class diagrams** show the system's classes and their relationships, the **activity diagrams** show the details of how a use case works, the **package diagrams** group classes into packages when the classes of the application become too many, the **sequence diagrams** describe also a use case (like the activity diagrams), but it shows the objects and the messages that are passed between these objects etc.

Four diagrams are found to be appropriate for the system's modeling. A brief description for each one of them follows (the symbols and notation used in these diagrams are explained briefly in the corresponding sections):

1) **Package diagram** *(see section 3.1)***:** Package diagram is used only when the system is large enough, e.g. many use cases, classes etc. and the model must be divided into 'packages'. Package diagrams show these packages and the interactions between them at a high level. Each UML package organizes the model's use cases, classes, or both into groups.

2) **Use case diagram** *(see section 3.2)***:** The highest level is the use case diagram which captures the behavior of a system as it appears to an outside user. It describes all of the system's functionality in a graphical table of contents (what the system should do, not how it should be done). The use case diagram has four elements: (a) the **actors**, who are persons, an organization, or external systems (in this work the actors are only persons), etc. Each actor is associated with several use cases, each of which describes what the actor wants to do with the system (this is the actor's goal), for instance '*retrieve documents in respect to a query*' and the user interacts with the system in order to achieve this goal, (b) the **use cases** (represented with ovals with their names on it), that describe a sequence of actions that must be executed in order to achieve or abandon (e.g. the user can't register) the actor's goal, (c) the lines that represent several types of **relationships** (e.g. *isa, include, extend* etc) between an actor and a use case, between two use cases and between two actors and (d) the **system boundary** which is a rectangle that includes all the use cases which indicate all the system's functionality.

3) **Activity diagrams***(see section 3.4)***:** Activity diagrams are useful for analyzing a use case by describing what **actions** need to take place (the use case scenarios *see section 2.2*) from the system's point of view. Strictly, an activity is a sequence of actions. The main reason to use them is that they model the workflow behind the system being designed. For each use case they show a sequence of steps (**actions**) and the flow of control from one step to another. These steps can cover multiple use cases. There are many different ways to use them. Here activity diagrams concentrate mostly on 'what gets done' in a use case scenario, rather than on 'who does each

action' (which class is responsible for each action). Activities diagrams are recommended when the system has parallel activities (multiple activities that occur at the same time) and identifying these activities early in the requirement analysis phase is essential.

4) **Class diagrams** *(see section 3.5)***:** Class diagrams are widely used. They describe the structure of the system e.g. the types of objects, the various kinds of relationships that exist among them, the properties and operations of each class etc. From these diagrams, there are tools that automatically generate code in a programming language (such as Java) and this code helps the team work on the same set of files without interfering with each other's work.

2.2 USE CASES

A use case is a collection of possible scenarios between the system and the external actors. Each use case is characterized by the goal that the primary actor has. This goal might be delivered or fail.

They are a widely used, valuable tool for capturing the functional requirements of a system. These requirements can be expressed either in text or (better) in a more structured format to avoid inconsistencies and redundancy. Here, the one-*column table template* is used [5] :

UC-x	**Use case title** (usually it is an active verb phrase that names the goal of the primary actor e.g. '*Retrieve relevant documents*', where the phrase starts with the active verb 'R*etrieve*')

A brief description of the use case's behaviour is (**use case brief**) is presented.

Goal in Context
A longer statement, which describes the use case's goal in more detail.
Details
Primary Actors: Are those with the goal that the use case addresses **Preconditions**: what is already true, before the use case starts. Since it is known that this condition is true, it will not be checked again during the use case (e.g. for the use case '*Retrieve relevant documents*', the precondition is that 'the user has entered a non-empty natural language query'). **Level:** the goal level is one of the following (note that the user goal use cases must be

detected first because these are the important ones and justify the system's existence):

- *Summary:* A summary level goal use case provides a table of contents for the lower-level use cases (which are user and other summary goal use cases)
- *User*: A user level goal use case provides the goal the primary actor has when using the system. The lists of these use cases make most of the system's functionality.
- *Subfunction* : A subfunction level goal use case carries out the user goals mentioned above. They are usually needed because many other user goals use cases, use them.

Scope : each use case is labelled with one of the following :

- *System* : only the first use case that glues together all the others has this scope. It includes all the system's functionality.
- The appropriate subsystem's name (see the system's architecture shown in *Figure 3.3)*. That means that the main system is opened and the use case describes how a piece of it works e.g. the System Management subsystem, the Document Retrieval subsystem etc.

Triggering Action: It states what event gets the use case started. Here, the triggering action is the first step of the use case.

Flow of Events

Main Success Scenario (is the body of the use case):

The main success scenario describes the case where nothing goes wrong. It is a sequence of numbered steps (a well written use case has more that 3 and less than 9) where each step is a simple statement and clearly shows who is controlling the action (*who has the ball*). It must be mentioned that here only active verbs that move the process forward are used e.g. the verb *check* is not used because it leaves open what the result of the check (either passes or fails). Instead the verbs *ensure*, *verify* and other goal achieving verbs are used. A typical example is instead of writing : '*The system checks for the type of user and …..*'the following is written: '*The system verifies that the user is a consumer and ….*' or '*The system verifies that the user is an expert and ….*'

Also it must be mentioned that when a use case references another use case, the second one is written in blue and underlined (like a hyperlink).

Flow of Events
Extensions: The Extensions describe what can happen differently during the main success scenario. Extensions can lead to a success (the actor achieves the use case's goal) or to failures. Each extension consists of two parts: (a) the extension conditions which are the conditions under which the system takes a different behaviour. A colon (:) is put after the conditions. (b) the extension handling which is a sequence of steps to deal with the conditions. **Variations:** Variations to a step from the main success scenario exist because what is happening in a step is the same, but there could be several different ways to do this. For example, see *UC-10 Categorize the document to consumer/expert category*. Here there is a step (step 2) that categorizes the document as an expert one if it has at least one expert term. There is also a variation to this step that categorizes the document as an expert if it contains at least 2 expert terms. Notice that in both cases the categorization will be done, what is different is how this is done.

2.3 MeSH

MeSH[1] (Medical Subject Headings) is a taxonomic hierarchy of medical and biomedical terms suggested by the U.S. National Library of Medicine[2] (NLM). It contains almost 25,000 terms covering life sciences, medicine etc. MeSH vocabulary reflects the progress in biomedical sciences. Every year several hundred new terms are added to the MeSH vocabulary and some existing terms are modified. MeSH is used by numerous health organizations and medical libraries, including MEDLINE, to organize materials and index information.

The MeSH vocabulary is available in XML format [8]. MeSH headings (MH) are organized in a tree with 16 main braches. Each branch has many levels of sub-branches, and each heading has a position in the hierarchy which is called **tree**

[1] http://www.nlm.nih.gov/mesh
[2] http://www.nlm.nih.gov

number. Some terms appear in more than one branch of the tree, e.g. the term 'ear' appears in two locations in the hierarchy, the '*Body Regions – Head*' and the '*Sense Organs*', therefore it carries two different tree number, [A01.456.313] and [A09.246]. Most of the MeSH heading are accompanied by a list of synonyms or very similar terms, known as **entry terms**. They indicate that information related to one term will be found under a different term, e.g. '*nose bleed*' and '*Nosebleed*' are entry terms for the MeSH heading '*epistaxis*'. Also each MeSH heading is associated with the **scope note** which is a short piece of free text that defines the corresponding MH.

2.4 UMLS

UMLS[3] (Unified Medical Language System) includes many controlled vocabularies in the biomedical sciences, designed and maintained by the US National Library of Medicine. It consists of the following components:

- **Metathesaurus**, which is a very large multi-purpose and multi-lingual vocabulary that contains information about biomedical and health related concepts, their various names and the relationships among them. It is designed for use by system developers and built from the electronic versions of over 100 sources, e.g. thesauri, vocabularies etc called "*source vocabularies*" of the Metathesaurus. Some of them are *Dental Terminology*, the *Diseases database*, *Gene Ontology*, *International Classification of Primary Care*, *MeSH* etc
- **Semantic Network**, which consists of the **semantic types** that provide a consistent categorization of all concepts represented in the UMLS Metathesaurus, and the **semantic relations** that exist between semantic types. The primary link between the semantic types is the '*isa*' link. The 'isa' link establishes the hierarchy of types within the Network and is used for deciding on the most specific semantic type available for assignment to a Metathesaurus concept. There is also a set of non-hierarchical relationships, which are grouped into five major categories: \`*physically related to*,' \`*spatially related to*,' \`*temporally related to*,' \`*functionally related to*,' and \`*conceptually related to*.' [9]

[3] http://umlsks.nlm.nih.gov

- **SPECIALIST Lexicon** which is intended to be a general English lexicon that includes both commonly occurring English words and biomedical vocabulary. It is used for natural language processing.

2.5 The MMTx tool

MMTx[4] (MetaMap Transfer) is a tool from the U.S. National Library of Medicine, that maps biomedical text (from documents, queries) into concepts in the UMLS Metathesaurus. MMTx uses the Metathesaurus and the SPECIALIST lexicon during the term extraction process. This process works in the following steps [3]: First, the text is parsed and a simple linguistic filter isolates the noun phrases. Then variants are generated from the resulting phrases, using the SPECIALIST lexicon, e.g. the phrase '*obstructive sleep apnea*' has variants: *obstructive sleep apnea*, *sleep apnea*, *sleep*, *apnea*, *osa*, etc. Afterwards, the candidate concepts from the UMLS Metathesaurus are retrieved and evaluated against the phrases, e.g. for the variant *osa* the following Metathesaurus concepts are retrieved: *osa [osa antigen],osa [osa gene product], osa [osa protein] etc*. Finally, the best of the above candidate concepts are chosen.
MMTx has some limitations especially during the variant generation stage [3] where the expansion of the initial text phrase to all possible variants is quite exhaustive. As a result, this process also results in term <u>over-generation</u>, which diffuses the original term concept and unrelated terms are added to the final candidate list. Another limitations is that MMTx extracts general UMLS Metathesaurus terms rather than MeSH (but MEDLINE indexing is based on MeSH).

2.6 The AMTEx method

AMTEx (Automatic MeSH Term Extraction method) is an automatic term extraction method, specifically designed for the automatic indexing of documents in large medical collections such as MEDLINE. AMTEx first uses the **C/NC-value method** for term extraction (C/NC value is domain-independent and combines statistical and linguistic information for the extraction of multi-word and nested terms, that are very common in the biomedical domain). Finally it validates the extracted terms against MeSH, rather than the full UMLS Metathesaurus (that contains over a hundred source vocabularies, including MeSH *see section 2.4*) mapping of MMTx [15].

[4] http://mmtx.nlm.nih.gov

2.7 RELATED SYSTEMS

The amount of health data accessible on the Web is increasing and Internet has become a major source of health information. Many medical information systems such as search engines, portals, meta search engines, digital libraries etc are available. Here is a brief description for some of these existing systems:

Experts are interested (and this feature is not supported by the existing systems) in browsing the medical literature via citation links, which means that it's important for them to navigate backward (the list of cited articles) and forward in time (the list of subsequent articles that cite the current article).There are systems e.g. **CiteSeer**[5] that support this feature but not in the medical domain. CiteSeer allows navigation in the computer science literature. It downloads articles (e.g. Postscript files) that are made available on the Web, converts them to text and performs *Autonomous Citation Indexing* (ACI). During this process the article's citations are parsed, citations to the same article but in different formats are identified, the context of citations within the body of the articles are also identified etc. Finally, CiteSeer performs full-text indexing of the articles and citations as well [13]. It can also generate statistics on citation frequency and allows the estimation of the importance of articles. CiteSeer's disadvantage is that it has grown enough pushing the limits of the current system's capabilities; it is hard to administer and modify. To overcome these drawbacks **CiteSeer**$^{\chi}$ is introduced [14].

Medscape[6] is used only by experts and provides clinical information relevant to their chosen speciality. Also, experts are interested in searching bibliographic databases e.g. **MEDLINE** through **PubMed**. MEDLINE is a database of over 15 million medical and scientific articles (most of them in English) indexed from the sixties to date, created and maintained by the U.S. National Library of Medicine (NLM). PubMed[7] is developed and maintained by the National Center for Biotechnology Information (NCBI) at NLM. It provides access to MEDLINE and to articles in selected life sciences journals not included in MEDLINE through an easy to use free Internet site. MEDLINE indexers describe the content of biomedical articles by assigning (typically 10 to 12 per article) the most specific MeSH terms(*see section 2.3*). PubMed uses them for retrieval and the search strategy is enhanced (e.g.

[5] http://citeseer.ist.psu.edu
[6] http://www.medscape.com
[7] http://www.ncbi.nlm.nih.gov/PubMed/

the query '*bad breath*' is mapped automatically to the MeSH term '*halitosis*'). Within PubMed's Consumer Health link leads to **MedlinePlus**[8] which is intended to be used mostly by consumers and also provides information that is authoritative and up to date.

Other medical information systems that are used mainly by consumers are the following. **MedicineNet.com**[9]**,** provides in an easy to understand language, authoritative medical information only for consumers. These are articles are written by a network of health professionals and that's the system's drawback since it's content is not always up to date. Health and medical information can also be provided by portals with too confusing user interface, such as **WebMD**[10] **,MEDNETS**[11] **Healthline**[12] etc, by metasearch engines such as **OmniMedicalSearch.com**[13] that also offers a special link for the experts.

Also there exist medical search engines used by both experts and consumers. Some of them are listed below: **MedHunt**[14], developed and maintained by the Health On the Net Foundation (HON - a not-for-profit organization that aims to provide access to reliable sources of online medical information). MedHunt retrieves medical information either from HON's accredited sites or from medical pages crawled from the Web. **WRAPIN**[15] (Worldwide online Reliable Advice to Patients and Individuals) developed from the EU-WRAPIN project, that also uses medical trustworthy sources such as the NLM's PubMed, HON's MedHunt, U.S. Food and Drug Administration (FDA) etc. WRAPIN also supports different types of query from a few keywords to entire web pages (specified by their URL), it maps both query and documents to MeSH terms(the *HONMeSHMapper* module is used) that are used for indexing and retrieval.

The proposed system selects some interesting features of the above systems(such as it uses documents only from authoritative sources to create the system's database, allows the users to navigate though literature via citation links etc) and adds more functionality (such as it improves ranking by the article's authority score, the

[8] http://medlineplus.gov
[9] http://www.medicinenet.com
[10] http://www.webmd.com
[11] http://www.mednets.com
[12] http://www.healthline.com
[13] http://www.omnimedicalsearch.com
[14] http://www.hon.ch/MedHunt
[15] http://www.wrapin.org/

document’s consumer/expert score etc).in order to be efficient and effective for both consumers and experts. See *section 1.1*. and *section 1.2* for more details about the system’s features.

Chapter 3

REQUIREMENTS ANALYSIS

During the analysis phase the system’s functionality is defined. The techniques used to capture this functionality are described in sections 2.1 and 2.2. During this phase is also important to discover classes, associations between classes, class operations, etc. The results of the analysis (UML diagrams and use cases) serve as input to the next phase.

First the package diagram (see *section 3.1*) is presented which shows the system’s packages and their relationships. Then the use case diagram (*see section 3.2*) follows and highlights the relationships between the three different type of users (administrator, expert, consumer) and the system’s use cases. For simplicity, only summary and user goal use cases are presented. Then the seventeen use cases *(see section 3.3)* are presented. For each use case an activity diagrams *(see section 3.4)* is created, in order to show graphically, from the system’s point of view, each use case’s behavior in more detail. Finally, the UML’s backbone, the class diagrams *(see section 3.5)* are presented. There are four class diagrams, one for each package described in the package diagram.

3.1 PACKAGE DIAGRAM

Initially there was no need to create a package diagram. But as the system’s functionality increased and the system became larger, more classes were introduced leading to a very complicated class diagram. Then it was decided to split classes (and use cases as well) into packages and the package diagram was created. The system’s package diagram is shown in *Figure 3.1*. The packages are displayed with a tabbed folder with the package’s name written on it. Also it shows the dependencies between the packages. There is a dependency from one package to another if any class in the first package has a dependency to any class in the other package.

Each package leads to a UML class diagram (*see section 3.5*), and each class is a member of a single package. In Java terms UML packages correspond to java packages (every class in the package must have a unique name within its owning package). It must be mentioned that the package diagram, was very helpful to control the large-scale structure of the proposed system.

Here follows a brief description for each package:
The User_Interface_Package contains the classes that interact with the user for query entry, results presentation, user login, user registration etc. The Document_Retrieval_Package contains the classes that are mainly responsible for processing the user's query, fetch the retrieved documents (it interacts with the System_Management_Package) and display the results to the user (it interacts with the User_Interface_Package). The User_Management_Package contains the classes that manage the users. Finally, the System_Management_Package contains the entity classes (these are classes that include the data that must be stored in the system's database).

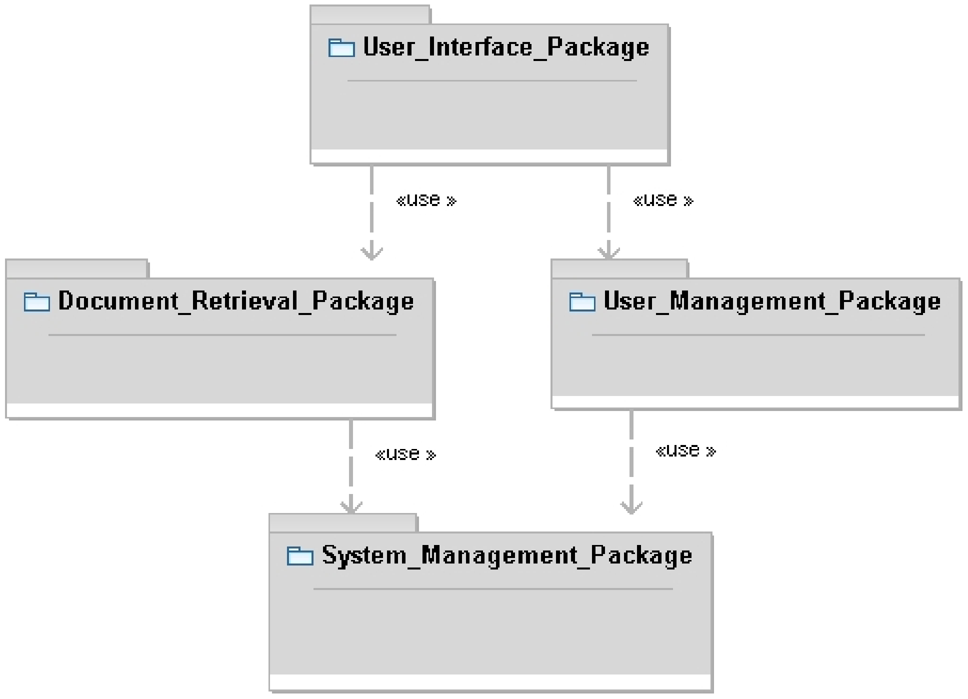


Figure 3.1: Package diagram

3.2 USE CASE DIAGRAM

The use case diagram is like a graphical table of contents that presents the system's behavior in respect to the users. The system's use case diagram (*see Figure 3.2*) presents:

- the system's actors. Here the actors (expert, consumer and administrator) are only humans. See *UC-1, Use the application* for a brief description for each actor's profile (e.g. job, typical background, skills etc).
- a list of use cases (represented as ovals). In order to keep use case diagram as clear as possible, low level system interactions (that is use cases lower than <u>user goal</u> level) are not presented.
- relationships (represented as arrows). Two relationship types are presented. Those that show the **inheritance** relationship between the actors (the expert user is a consumer plus some extra functionality that the system offers to him e.g. browsing the system's database by medical topics). The other is the **<<include>>** relationship, a directed relationship between two use cases. It denotes the invocation of a use case by another one, e.g. the behavior of the use case '*Present results to user (UC-15)*' is inserted into the behavior of the '*Answer the query (UC-12)*' use case.

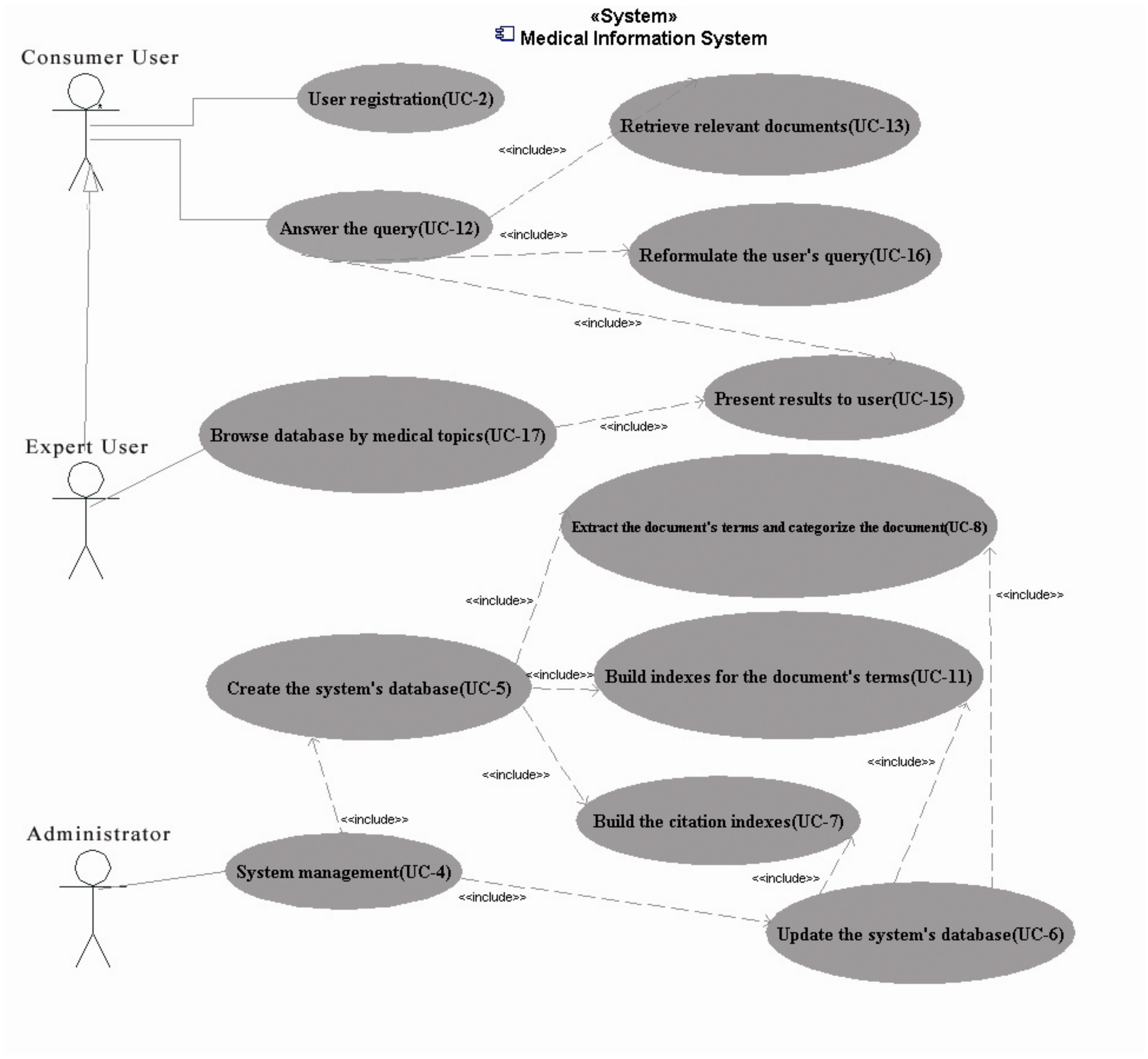

«System»
Medical Information System
Consumer User
Expert User
Administrator
User registration(UC-2)
Answer the query(UC-12)
Retrieve relevant documents(UC-13)
Reformulate the user's query(UC-16)
Present results to user(UC-15)
Browse database by medical topics(UC-17)
Extract the document's terms and categorize the document(UC-8)
Create the system's database(UC-5)
Build indexes for the document's terms(UC-11)
Build the citation indexes(UC-7)
System management(UC-4)
Update the system's database(UC-6)
<<include>>

Figure 3.2 : Use case diagram

3.3 USE CASES

The system's architecture is shown in *Figure 3.3* below. The 17 use cases are split up into the following modules:

- System Management,
- Document Retrieval,
- User Management, and
- User Interface Module

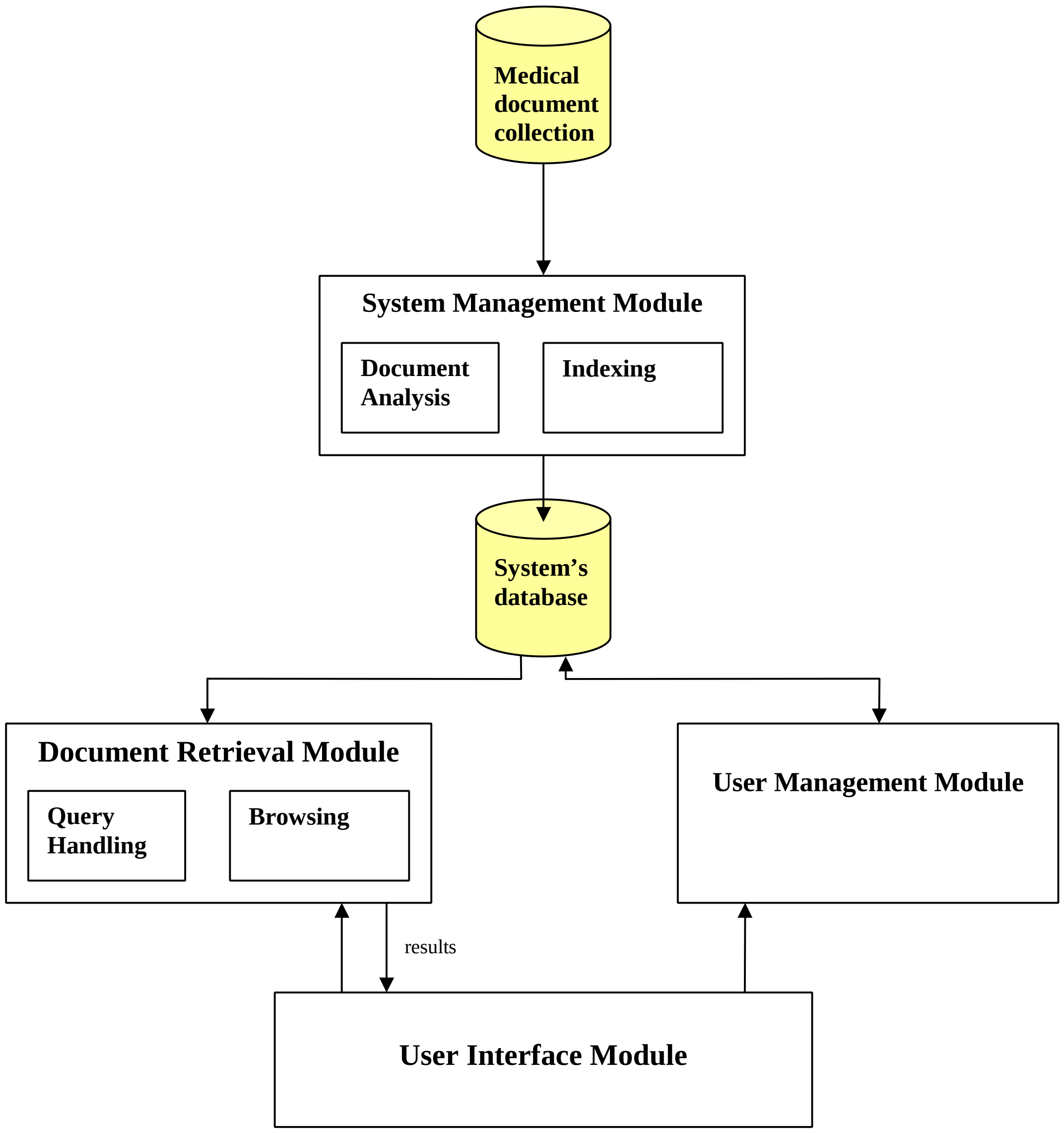


Figure 3.3: The system's architecture

The use cases have a **hierarchical structure** *(see Figure 3.4)* and each use case belongs to the module(s) mentioned in *Figure 3.3*. To be more specific the use cases written in blue belong to the *User Interface Module*, in orange to the *User Management*, in green to the *Retrieval* and in red to the *System Management Module*. Each use case belongs to exactly one module except from *UC-9, Create the representation vectors for the document or the user's query*, which belongs to two modules, the *System Management Module* and the *Retrieval Module* (it is both red and green).

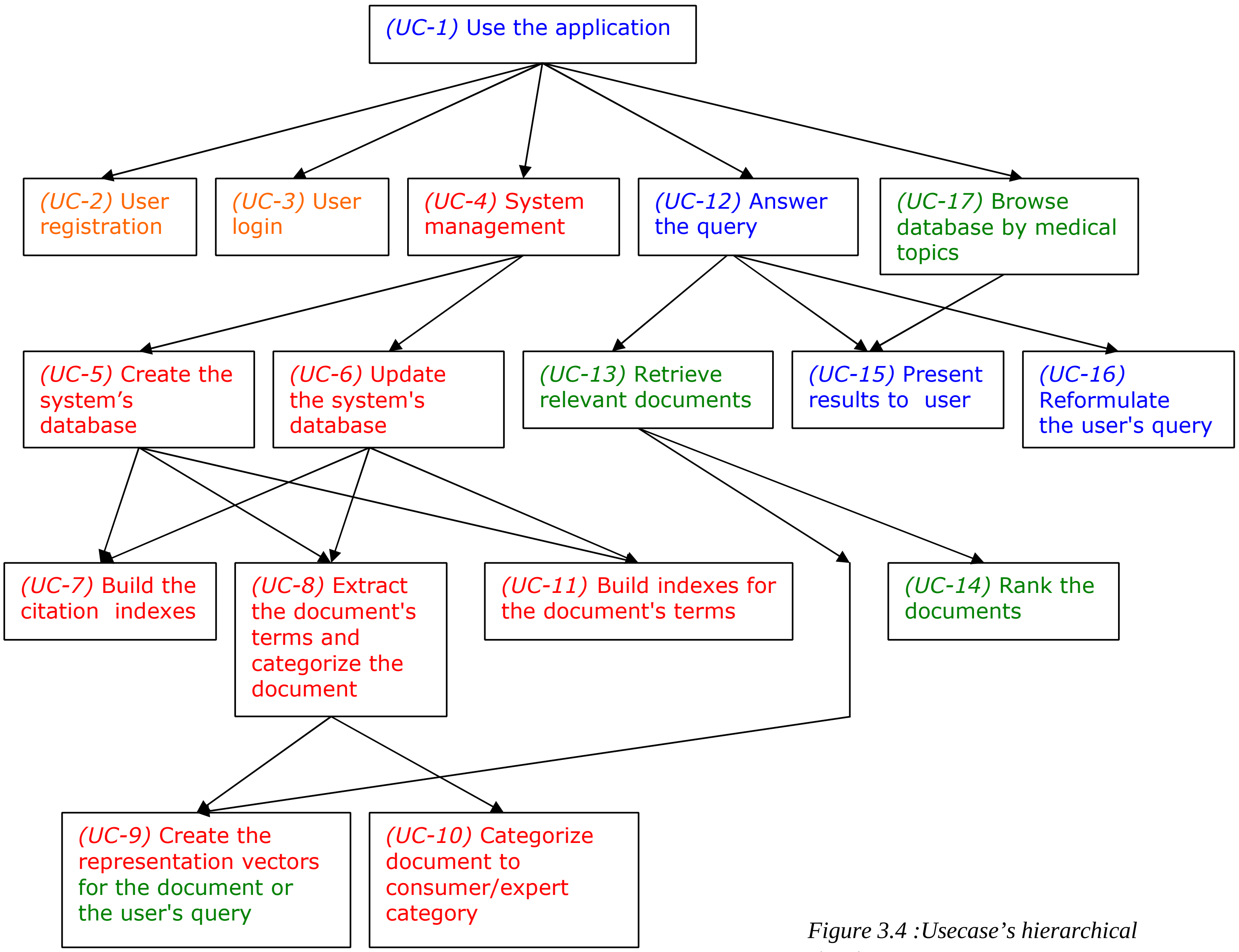

(UC-1) Use the application
(UC-2) User registration
(UC-3) User login
(UC-4) System management
(UC-12) Answer the query
(UC-17) Browse database by medical topics
(UC-5) Create the system's database
(UC-6) Update the system's database
(UC-13) Retrieve relevant documents
(UC-15) Present results to user
(UC-16) Reformulate the user's query
(UC-7) Build the citation indexes
(UC-8) Extract the document's terms and categorize the document
(UC-11) Build indexes for the document's terms
(UC-14) Rank the documents
(UC-9) Create the representation vectors for the document or the user's query
(UC-10) Categorize document to consumer/expert category

Figure 3.4 :Usecase's hierarchical structure
(A) System Management Module
(B) Retrieval Module
(C) User Management Module
(D) User Interface Module

A brief description for each module and a list of the corresponding use cases is followed.

(A) System Management Module

This module parses each document from the medical document collection *(see Glossary)* and adds them to the system's database (*see Glossary*).

It includes the following use cases:

- **System management** *(UC-4)*, where the system's database is created/updated
- **Create the system's database** *(UC-5)*, where the system parses each document from the medical document collection, analyzes and indexes its content. Also, it categorizes the document and assigns an authority score for the articles only.
- **Update the system's database** *(UC-6)*, where the system processes the updated or the new documents from the medical document collection and updates the system's database accordingly.

Before adding the document to the system's database, it must be processed by the following subsystems (*see Figure 3.3*):

(I) The **Document Analysis subsystem** which parses the document, extracts it's (semantic and lexicographic) terms and categorizes it to Semantic Network MeSH categories *(see Glossary)* and to consumer/expert categories. The **Document Analysis subsystem** includes the following use cases:

- **Extract the document's terms and categorize the document** *(UC-8)*, where the system extracts the document's terms and creates three representation vectors. Finally, the system categorizes the document to: (a) consumer/expert categories, and (b) Semantic Network MeSH categories.
- **Create the representation vectors for the document or the user's query** *(UC-9)*, where the system extracts terms from the document and creates three representation vectors.
- **Categorize the document to consumer/expert category** *(UC-10)*, where the system categorizes the document as consumer or expert with a weight of belief that it belongs to each category

(II) **The Indexing subsystem** that builds the citation indexes and the document's terms indexes (extracted from the **Document Analysis subsystem** described above). The Indexing subsystem includes the following use cases:

- **Build the citation indexes** *(UC-7)*, where the system processes the bibliography part (only for research documents) and builds the citation indexes.
- **Build indexes for the document's terms** *(UC-11)*, where the system indexes the document by using the Inverted Files method.

(B) Retrieval Module

This module retrieves documents from the system’s database either in respect to the user’s query or by browsing the medical topics. It’s obvious that this module consists of the following subsystems (*see Figure 3.3*):

(I) The **Query Handling subsystem** which parses the user’s query, extracts it’s (semantic and lexicographic terms), retrieves and ranks a list of relevant documents to the query and finally it suggests a list of terms for the query expansion process, if it is necessary. The Query Handling subsystem includes the following use cases:

- **Retrieve relevant documents** *(UC-13)*, where the system compares the user's query with each document from the database and retrieves a list of relevant documents in response to the user’s query.
- **Create the representation vectors for the document or the user's query** *(UC-9)*, where the system extracts terms from the user’s query and creates three representation vectors. Notice that this use case also belongs to the *Document Analysis subsystem* from *the System Management Module* described above.
- **Rank the documents** *(UC-14)*, where the system uses one of the: (a) lexicographic terms or (b) MMTx terms or (c) AMTEx terms representation vectors for both the query and the documents and produces a ranked list of retrieved documents.

(II) The **Browsing subsystem**. Only one use case belongs to this subsystem:

- **Browse database by medical topics** *(UC-17)*, where the system allows the Expert user to view the documents through browsing the system’s database by the medical topics.

(C) User Management Module

This module manages the user’s database. It includes the following use cases:

- **User registration** *(UC-2)*, where the user registers to the system.

- **User login** *(UC-3),* where the system validates the user (administrator, consumer or expert). In case where he hasn’t registered, the system forces him to register.

(D) User Interface Module

This module provides the system’s functionality to its users. It includes the following use cases:

- **Use the application** *(UC-1),* where the user uses the system's functionality according to his type (Administrator, expert, consumer, *see UC-1 about each user’s profile*))
- **Answer the query** *(UC-12),* where the user enters a query, retrieves relevant documents and reformulates the query, if the results are inefficient.
- **Present results to user** *(UC-15)*
- **Reformulate the user's query** *(UC-16)*, where the user's query is expanded with new terms.

UC-1 Use the application

This is the higher level use case and glues together the lower level ones. It is little more than a table of contents that names the **actors** and their highest level goals. In our system all actors are humans. It is important to refer not just to the actor’s names but also to each actor's **profile** (brief description that includes information such as their job, typical background, skills etc). This kind of information is useful later in the system's behavior and the user's interface design so the software will suit the needs of the end users.

Our system supports three different types of actors:

1. the Administrator, who is a computer science expert. He creates the database, manages the application's parameters and he is not expected to be familiar with the medical terminology.
2. the Consumer, who is an ordinary user (patient, relative, the public generally). He is usually unfamiliar with the medical terminology, having a low degree of health literacy. He is often unclear about the problem that he is facing. As a result, it is difficult for him to choose a few accurate medical phrases as a starting point of his search. He is also not expected to operate a GUI with ease and may use the system occasionally.
3. the Expert who is a health care professional (medical doctor, nurse, physiotherapist, nutritionist, medical researcher etc), therefore familiar with the medical terminology. He might have difficulties in operating a GUI, he might want to customize the User Interface and he is expected to use the system frequently.

The actors mentioned above, use the system in order to accomplish some goals. The highest level goals for each actor are presented in the use case's Main Success Scenario steps shown below:

Goal in Context
The user uses the system's functionality according to his type (Administrator, Expert, Consumer)
Details
Primary Actors: Administrator, Consumer, Expert **Supporting Actors:** **Preconditions**: none **Level:** Summary **Scope :** System **Triggering Action :** The user opens the application
Flow of Events
Main Success Scenario: 1. The system provides the User with the following list of options and the user logs in : 1.1. Register to the system (UC-2) 1.2. Login to the system (UC-3) 2. The system verifies that the user is the Administrator and provides him the option to manage the system (UC-4) 3. The system verifies that the user is an Expert or a Consumer and provides him

Flow of Events

with the following options:

3.1. The user enters a query, the system retrieves relevant documents in response to his query and the user reformulates the query if necessary (UC-12)

3.2. The user customizes his profile from the following list of (advanced) options:

3.2.1. update the number of the top documents from the retrieved list. MeSH terms will be extracted from these documents and will be used for the query reformulation process

3.2.2. update the similarity threshold (it defines terms lower or higher in the MeSH hierarchy used for the query reformulation process)

3.3. The user updates his personal information, username and password

4. The system verifies that the user is an expert one and allows him to browse the documents through the Semantic Network MeSH categories (UC-17)
5. The system repeats step 2-4 until the user indicates that he wants to exit the application
6. The system terminates the session

Extensions:

1.a The user's registration or the user's login process fails:

1. The system continues to step 1 until the user succeeds to login or to register

3.3.a The user updates his username:

1. The system verifies that the username has not been used by another user

3.3.b The user is an expert one:

1. The system allows him to update his favorite medical topics

Variations:

4. The expert can switch to browse to another type of document's categorization such as: (a) categories that come from the document clustering process or (b) categories that come from the citation graph Link Clustering process.

UC-2 User registration

Only consumer and expert users can register to the system. They provide necessary information for the registration such as personal information, username, password etc. Additionally, the experts can select their favorite medical topics from the Semantic Network MeSH categories (*see Glossary*). Alternatively, the they can select their favorite medical topics from categories that are created through (a) the document clustering process or (b) the citation graph Link Clustering process.

Goal in Context

The user provides information to the system and the system registers him as a valid user

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: The user was not a valid user
Level: User **Scope :** User Management Module
Triggering Action : The user registers to the application

Flow of Events

Main Success Scenario:

1. The user enters personal information such as his name, last name, email, job, phone number(s), addresses(s) and the type of user he is(consumer or expert)
2. The user enters a username and a password
3. The system verifies that the username has not been used by another user
4. The system stores the user's personal information, username, password and the registration date (which is the current date)
5. The system verifies that the user is an expert and allows him to select favorite medical topics from the Semantic Network MeSH categories

Extensions:

3.a The username has been used by another user:
   1. The user is prompted to enter a new username
   2. The system continues to step 3

Variations:

UC-3 User login

The system validates the user (administrator, consumer or expert). In case where the user hasn't registered, the system forces him to register (*see UC-2 User registration*).

Goal in Context

The user enters the application

Details

Primary Actors: Administrator, Consumer, Expert
Supporting Actors:
Preconditions: The user is the system's Administrator or he is a registered user (Expert or Consumer)
Level: Subfunction **Scope :** User Management Module
Triggering Action : The system asks the user to provide his username and password

Flow of Events

Main Success Scenario:

1. The user enters his username and password
2. The system validates the username
3. The system validates the password

Extensions:

2.a Invalid username:
  1. The user registers in the application (UC-2)

3.a Invalid password:
  1. The system notifies user and requests the password again
  2. The user re-enters the password
  3. The system continues to step 3

3.b Invalid password entered too many times:
  1. The system notifies the user
  2. The system terminates the session

Variations:

UC-4 System management

This is a high level use case, a table of contents, that names the system Administrator's goals which are :

(1) He defines the medical document collection (*see Glossary*). This collection is created either:
   a. by using an existing medical database such as Medline (*see Glossary*) or
   b. by crawling the Web's HON accredited sites (*see Glossary*). The crawling process starts with a small set of HON accredited sites (the 'seed') and then explores other pages by following the page's links. These sites are identified easily by the a **HON-icon** in the home page or in the first level pages (pages that are linked from the home page). This HON-icon is associated with a link to the site's **HON certificate**, which includes details about the HON's accreditation process such as the date of initial review, the HONcode PIN number, the date of last subsequent review etc.

(2) Once the medical document collection is defined, the administrator's main goal is satisfied, which is to create the system's database (*see Glossary*). The documents from the medical document collection are parsed one by one, analyzed and indexed (*see UC-5 Create the system's database*).

(3) Also, one of the system administrator's goals is to customize the system's parameters (*see step 1.3 for details*) such as how often the re-crawling process (if the medical document collection is created by crawling) is triggered, e.g. once a month, every two months etc. When the re-crawling process finishes the system's database is updated (*see UC-6 Update the system's database*).

Goal in Context
The system manages the system's database and customizes the system's parameters
Details
Primary Actors: Administrator **Supporting Actors:** **Preconditions**: The system's Administrator is logged on to the system. **Level:** Summary **Scope :** System Management Module **Triggering Action :** The system presents a list of options to the Administrator
Flow of Events
Main Success Scenario: 1. The Administrator views the following list of options and selects one of them: 1.1. Create the system's database: 1.1.1. Create the medical document collection either: 1.1.1.1. by using existing medical databases, or 1.1.1.2. by crawling the Web's HON-accredited documents 1.1.2. The system parses each document from the medical document collection one by one, and creates the system's database (UC-5) 1.2. Update the system's database: 1.2.1. Update the medical document collection either: 1.2.1.1. by adding/updating articles from latest versions of existing medical databases, or

Flow of Events

- 1.2.1.2. by re-crawling the Web's HON-accredited documents
- 1.2.2. The system updates the system's database (UC-6)
- 1.3. Customize the system's parameters from the following list:
  - 1.3.1. the crawling parameters such as the crawling strategy, the seed URL's, how often the re-crawling process starts etc
  - 1.3.2. parameters that are used for the query expansion process such as the number of the top-ranked retrieved documents, the similarity threshold etc
  - 1.3.3. define the type of the medical categories that expert users use for browsing the medical documents (default are the Semantic Network MeSH categories)
- 1.4. The Administrator selects from the following list of options:
  - 1.4.1. he defines a period of time, the system lists information about the users that have been (or have not been) using the system during this period
  - 1.4.2. he notifies users with new articles, medical topics etc

Extensions:

Variations:

UC-5 Create the system's database

This use case is mainly responsible for managing the **document indexing** and **categorization process**.

Prior to indexing, the system iterates through the documents of the medical document collection and for each individual document (such as *html, plain text documents, Microsoft Word, Excel, PowerPoint, Adobe Acrobat files etc*):

(1) It parses and extracts the document's parts (all parts are optional). These are the document's
   - title,
   - text,
   - bibliography part (for articles only),
   - metadata, such as information about the authors(name, lastname etc) and their affiliation(university, departure name etc), the article's source e.g. journal, book, conference etc, the date of the article's publication or the date of the page's creation and finally the Medline terms (only for the Medline articles)

(2) It builds the author and the Medline terms indexes (if they exist)

(3) It processes the bibliography part (only for research documents) and builds **the citation indexes** (*see UC-7 Build the citation indexes*). From now on the terms article, paper and research document will be used interchangeably.

(4) It **extracts the document's terms**(semantic or lexicographic) and creates the corresponding document representation vectors(*see UC-9 Create the representation vectors for the document ot the user's query*).Then, based on the document's MeSH terms representation vector, the system **categorizes** the document to one of the consumer/expert categories and to Semantic Network MeSH categories (*see UC-8 Extract the document's terms and categorize the document*).

(5) It creates (optionally) another representation vector for the document which consists of user-defined MeSH terms. At this point, the Administrator uses the system on behalf of a medical expert.

Once the representation vectors mentioned above are created, they are passed to the Indexer, which creates four main data structures (*see UC-11 Build indexes for the document's terms*).

When all the documents from the medical document collection are indexed and stored to the system's database(or the system's resources are finished), the system computes:

- each **term's frequency in the whole collection** and stores that value in the corresponding term's Vocabulary and
- an **authority score** based on the citation graph link analysis process.

The system uses these values later on the retrieval process (*see UC-13 Retrieve relevant documents*).

Finally, **further document categorization** can be performed through the **clustering** process based either on:

- the citation graph link analysis or
- document clustering using the document vector representations (e.g. by kMeans, xMeans etc).

Goal in Context

The system parses each document from the medical document collection, analyzes, indexes its content, categorizes it and assigns an authority score for the articles only.

Details

Primary Actors: Administrator
Supporting Actors:
Preconditions: The medical document collection has been created
Level: Summary **Scope :** System Management Module
Triggering Action : The system gets a document from the medical document collection

Flow of Events

Main Success Scenario:

1. The system converts the document into text and extracts it's structural parts
2. The system builds the index for the authors field
3. The system ensures that the document is a research one by testing for the existence of a bibliography part and builds the citation indexes *(*UC-7*)*
4. The system extracts the document's terms, creates the corresponding representation vectors and categorizes it to : (a) consumer - expert categories and (b) Semantic Network MeSH categories (UC-8*)*
5. The administrator assigns user-defined MeSH terms to the document
6. The system builds the indexes for the document's terms (lexicographic, MeSH, UMLS) *(*UC-11*)*
7. The system gets the next document from the medical document collection and repeats steps 1-6 until all of them are stored in the system's database
8. The system augments the vocabularies of the database by the addition of global statistics about each term that occurs in the document collection
9. The system performs link analysis to the citation graph, computes and stores an authority score for each research document
10. The system performs clustering based on the citation graph and achieves further categorization for the research documents
11. The system performs document clustering and extracts further document categorization

Extensions:

5.a The administrator doesn't assign user-defined MeSH terms to the document:

1. The system continues to step 6

7.a The system resources are finished:

1. The system continues to step 8

Variations:

1.a. For the ps or pdf files the conversion into text can be accomplished using the PreScript from the New Zealand Digital Library project

1.b. The document's parts might be extracted using various models, such as regular expressions, classifiers etc

UC-6 Update the system's database

This use case is responsible for updating the system's database when new documents have entered the medical document collection or the existing ones have been updated. They are parsed, the document's terms are extracted and (new) categories are assigned.
Finally, the system updates or adds the corresponding indexes and the document to the system's database.
It is important to notice that for the updated articles the system keeps the same authority score and the same idf's (term frequency in the whole collection).

Goal in Context

The system processes the updated or the new documents from the medical document collection and updates the system's database accordingly

Details

Primary Actors: Administrator
Supporting Actors:
Preconditions: The re-crawling process has finished or new documents have entered the medical document collection
Level: Summary **Scope :** System Management Module
Triggering Action : This use case is triggered by time or by Administrator

Flow of Events

Main Success Scenario:

1. The system gets a document (from the medical document collection) that has been updated or added
2. The system converts the document into text and extracts its structural parts
3. The system extracts the document's terms, creates the corresponding representation vectors and categorizes it to : (a) consumer - expert categories and (b) Semantic Network MeSH categories *(*UC-8*)*
4. The administrator updates (or adds) document's user-defined MeSH terms
5. The system assigns further categorization to the document
6. The system ensures that the document already exists in the system's database and performs the following:
   6.1. The system creates or deletes the corresponding posting files for each of the document's representation vector that has been changed and saves the updated document to the document's file keeping the existing document ID
   6.2. The system creates or deletes the corresponding posting files for the document's authors, Medline terms and user defined MeSH terms that have been changed
7. The system ensures that the document is new and performs the following:
   7.1. The system builds the author indexes
   7.2. The system ensures that the document is a research one by testing for the existence of a bibliography part and builds the citation indexes *(*UC-7*)*
   7.3. The system builds the indexes for the document's terms (lexicographic, MeSH, UMLS) *(*UC-11*)*
8. The system gets the next document from the medical document collection that has been updated or added and repeats steps 1-6 until all of them are updated or added to the system's database

Flow of Events
Extensions: 4.a The Administrator doesn't assign new MeSH terms: 1. The system continues to step 5 **Variations:**

UC-7 Build the citation indexes

This use case is triggered only for articles appeared in journals, workshop proceedings, technical reports etc. First, the system normalizes the article's bibliography part by:

- converting the letters to uppercase and the hyphens to spaces,
- removing the citation tags at the beginning of the citation
- removing some words from a known pre-defined list such as '*in press*', '*accepted for publication*' etc and
- expanding abbreviations e.g. the word *conf.* to *conference*, *proc*. to *proceedings* etc.
- normalizing the author names so that the misspellings and the name variants are identified, e.g. the system :
  - checks for first, last and middle names and
  - converts the first and middle names to single letter initials

The normalization process is important as it helps the system to identify citations to the same article written in different formats.
Afterwards the system processes the article's bibliography part and extracts (in a tagged format e.g. XML) information about each of the article's references such as title, information about the authors and information about the article's source e.g. journal, book, conference etc, the date of publication and the context of each citation from the body of the article.
Then the system creates the:

- forward links (*see Glossary*) to articles and
- backward links (*see Glossary*) from articles

that exist in the system's database. There are links associated with articles that do not exist in the system's database yet. These links are added to a citation list (*see Glossary*).
Finally the system builds the citation index which catalogues the citations that an article makes, linking the articles with the cited works.

Goal in Context
The system processes the bibliography part of published documents and builds the citation indexes.
Details
Primary Actors: Administrator **Supporting Actors:** **Preconditions**: The document is an article (it has a bibliography part) **Level:** User **Scope :** Indexing subsystem **Triggering Action :** The system performs normalization to the document's bibliography part.
Flow of Events
Main Success Scenario: 1. The system extracts information about the article's references 2. The system verifies that the cited article exists in the system's database 3. The system creates a **forward link** to this article

Flow of Events
4. The system repeats steps 2-3 until all the forward links of the article have been created 5. The system verifies that there are citations to this article 6. The system creates the corresponding <u>backward links</u> 7. The system builds the citation indexes based on the article's forward and backward links 8. The system creates the citation graph **Extensions:** 2.a The cited article does not exist in the system's database: 1. Add information about the forward link to the citation list **Variations:**

UC-8 Extract the document's terms and categorize the document

This use case extracts the document's terms and creates the document's representation vectors (*see UC-9 Create the representation vectors for the document*). All the other modules of the system mostly depend on the representation vectors produced here. Then the system, based on the document's AMTEx terms, categorizes it to two different types of categories: (1) consumer/expert categories with a weight of belief that the document belongs to each category (*see UC-10 Categorize the document to consumer/expert category*) and (2) the Semantic Network MeSH categories (*see Glossary*).

Goal in Context

The system extracts the document's terms, creates three representation vectors and categorizes it to: (a) consumer/expert categories, and (b) Semantic Network MeSH categories

Details

Primary Actors: Administrator

Preconditions: The document has been converted to text (in case of articles this text includes the title, abstract and the article's body)

Level: User **Scope :** Document Analysis subsystem

Triggering Action : The system processes the document

Flow of Events

Main Success Scenario:

1. The system extracts from the document the following terms: (a) the document's lexicographic terms, which are mainly single word terms, (b) AMTEx terms and (c) MMTx terms and creates the corresponding representation vectors *(*UC-9*)*
2. The system performs statistical terms analysis on the AMTEx terms representation vector of the document and categorizes it (with a weight of belief) to one of the consumer/expert categories (UC-10*)*
3. The system processes the AMTEx terms representation vector of the document, extracts the Semantic Network MeSH categories and categorizes the document to these categories

Extensions:

Variations:

UC-9 Create the representation vectors for the document or the user's query

This use case is responsible for creating the following representation vectors for the document or the user's query:

(1) The d_{lex_terms}, that contains the lexicographic terms, which are mainly single word terms. Prior creating this vector, the system performs the standard process in the information retrieval domain: it removes "useless" to the system words e.g. stopwords, it reduces terms to their roots (stemming), it expands acronyms, it merges multiple term entries for the medical text and finally it compiles the within term frequency information.

(2) The d_{MMTx_terms}, that contains MMTx *(see Glossary)* terms. These are UMLS Metathesaurus terms*(see Glossary)*.

(3) The d_{AMTEx_terms}, that contains the AMTEx *(see Glossary)* terms. These are MeSH terms.

Each of these vectors includes:

- the actual string (textual form of the terms found) and
- the term's frequency in the document or in the user's query.

Goal in Context
The system extracts terms from the document or the user's query and creates three representation vectors
Details
Primary Actors: Administrator, Consumer, Expert
Supporting Actors:
Preconditions: The document's text has been extracted or the user has entered a query
Level: Subfunction **Scope :** Document Analysis subsystem & Query Handling subsystem
Triggering Action : The system processes the user's query or a new document from the medical document collection
Flow of Events
Main Success Scenario: 1. The system performs the standard process in the information retrieval domain and creates the representation vector that consists of lexicographic terms 2. The system uses the AMTEx method and produces the representation vector of AMTEx terms 3. The system uses the MMTx method and produces the representation vector of MMTx terms **Extensions:** **Variations:**

UC-10 Categorize the document to consumer/expert category

This use case describes the process which categorizes a medical document as expert or consumer and assigns a weight of belief that the document belongs to each of these categories. During this process the AMTEx terms *(see Glossary)*, which are actually MeSH terms, are divided to one of the following categories:

1. **consumer terms** that belong both to Wordnet and MeSH (Wordnet$\cap$MeSH),
2. **expert terms** that belong to MeSH but not to Wordnet (MeSH-Wordnet).

A document is considered to be an expert one, if it contains at least one expert term. Otherwise it is considered to be a consumer.

Furthermore, the document is assigned a value (which is the ratio $\frac{\#consumer\_terms}{\#total\_terms}$ for the consumer documents and the ratio $\frac{\#\exp ert\_terms}{\#total\_terms}$ for the expert documents).

These values are (its obvious that they are summed to one) are used later to improve the system's retrieval performance (*see UC-13 Retrieve relevant documents*).

Goal in Context

The system categorizes the document as consumer or expert with a weight of belief that it belongs to each category

Details

Primary Actors: Administrator
Supporting Actors:
Preconditions: The AMTEx terms representation vector of the document has been created
Level: Subfunction **Scope :** Document Analysis subsystem
Triggering Action : The system performs statistical term analysis on the AMTEx terms representation vector of the document

Flow of Events

Main Success Scenario:

1. The system computes the number of consumer and the number of expert terms
2. The system ensures that the document has at least one expert term and categorizes it as an expert one.
3. The system computes the weight of belief that the document belongs to the consumer category
4. The system computes the weight of belief that the document belongs to the expert category

Extensions:

2.a The document hasn't any expert terms:
   1. The system categorizes it as a consumer document

Variations:

2. The system could categorize the document as an expert one, if it contains at least 2, 3 etc expert terms

UC-11 Build indexes for the document's terms

This use case is responsible for managing the indexing process based on the document's terms, which are:

- the lexicographic terms,
- the AMTEx terms,
- the MMTx terms
- the user defined MeSH terms and
- the Medline terms (only for Medline documents).

During this process the system creates/updates four main data structures:

1. the **Documents File** which stores each document with a document ID,
2. a **Vocabulary** for each type of terms (*lexicographic terms*, *MeSH terms*, *UMLS terms*) ,
3. the **Inverted Index** which stores the postings list for each of the terms mentioned above (it also contains statistics on term frequency in each document) and
4. the **Direct Index** which associates a posting list for each document. This list consists of:
   - the document's AMTEx terms and
   - the AMTEx term's frequency in the document. The Direct Index can be used to facilitate the query expansion process (*see UC-16 Reformulate the user's query*) and the documents clustering process based on the AMTEx terms (*see step 10 UC-5 Create the system's database*).

Goal in Context
The system indexes the document by using Inverted Files.
Details
Primary Actors: Administrator **Supporting Actors:** **Preconditions**: The document's representation vectors have been created **Level:** User **Scope :** Indexing subsystem **Triggering Action :** The system stores the document to the Document's File
Flow of Events
Main Success Scenario: 1. The system repeats steps 2-4 for each of the following terms : (a) lexicographic terms, (b) AMTEx terms (c) MMTx terms (d) user-defined MeSH terms and (e) Medline terms 2. The system adds all the distinct values of the representation vector to the corresponding Vocabulary 3. The system creates the posting lists for the document's terms 4. The system verifies that the AMTEx terms representation vector is processed and adds the document and the document's AMTEx terms to the Direct Index **Extensions:** 1.a The document hasn't any user-defined MeSH or Medline terms : 1. The system doesn't repeat the steps 2-4 2. The system Exits 2.a There are terms that already exist in the Vocabulary:

Flow of Events
1. The system continues to step 3 **Variations:**

UC-12 Answer the query

The use case starts when the user submits a free text query. Consumers usually enter long queries that describe their situation in great detail (e.g. medical history, family medical history, where and how they feel uncomfortable, what happened the last several days etc) in plain English - much like the way they talk to a doctor. The user can also specify **additional operations** (and, or, not) to achieve more complex queries such as:

- a particular query term should or should not appear in the retrieved documents,
- two or more query terms should appear in the retrieved documents, and
- either one or another query term should appear in the retrieved documents.

Also, the user can switch between different types of representation for both the document and the query *(UC-14, Rank the documents, lists all the combinations that our system supports).*

Prior handling the query the **spelling corrector** is called which corrects spelling errors made while entering specialized, infrequently used medical terms in search queries. The spelling corrector is valuable especially for the consumers who may approach the medical domain in an approximate way. Then the system retrieves relevant documents in response to the user's query (*see UC-13 Retrieve relevant documents*). Finally, the user views the results (*see UC-15 Present results to user*). Usually he obtains a large list of documents or he had restricted so much the query that the results are not sufficient. At this point he has to **reformulate** his query. The system provides a variety of reformulating techniques in order to improve the retrieval performance *(see UC-16 Reformulate the user's query).*

Goal in Context

The user enters a query, retrieves relevant documents in response to his query and reformulates the query if the results are inefficient.

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: The user is logged in the system
Level: User **Scope :** User Interface Module
Triggering Action : The user submits a free text query

Flow of Events

Main Success Scenario:

1. The user specifies additional operations (and, or, not) to achieve more complex queries
2. The user selects the type of terms he desires for the retrieval process
3. The system ensures that the query hasn't any spelling errors
4. The system retrieves relevant documents in response to the query *(*UC-13*)*
5. The system presents the results to the user *(*UC-15*)*
6. The user reformulates his query *(*UC-16*)*
7. The System repeats steps 4-6 until the user indicates that he has finished

Flow of Events
Extensions: 1.a The user doesn't specify additional operators or values to metadata fields: 1. The system continues to step 2 3.a The query has spelling errors: 1. The system suggests the possible corrections 2. The user selects a suggestion or ignores them 7.a The user doesn’t select to reformulate his query: 1. The system Exits **Variations:** 3. Various correction algorithms could be used, such as the Levenstein edit distance algorithm, in order to create the list with the most similar candidates

UC-13 Retrieve relevant documents

This use case is responsible for the retrieval process. During this process:

(a) The system creates three representation vectors for the user's query *(see UC-9 Create the representation vectors for the document or the user's query)*

(b) The system (based on the Vector Space Model), retrieves a list of relevant documents. A similarity score is assigned to each of the documents retrieved by estimating the relevance between the document's content and the user's query *(see UC-14 Rank the documents).* The documents in the answer set are ordered by this similarity score. All documents up to a prespecified degree of similarity (defined by a query threshold) are retrieved. Alternatively the top k (e.g. 20) documents are retrieved.

(c) Finally, the system allows the retrieved list of documents mentioned above, to be altered to take into account additional types of evidence *(see steps 3-5)* such as the user's type (consumer or expert) and the user's favorite categories(only for expert users). The filtering process is valuable in cases where he wants to restrict the domain of the documents being retrieved.

Goal in Context

The system compares the user's query with each document from the database and retrieves a list of relevant documents.

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: The user has logged in the system
Level: User **Scope :** Query Handling subsystem
Triggering Action : The user enters a non empty free text query

Flow of Events

Main Success Scenario:

1. The system extracts (a) the lexicographic terms, (b) the AMTEx terms and (c) the MMTx terms from the query and creates the corresponding query's representation vectors *(*UC-9*)*
2. The system computes a similarity score, determines the documents that match the specific query and ranks the documents by this score*(*UC-14*)*
3. The system verifies that the user is a consumer and computes a score for each of the documents retrieved by multiplying the document's similarity score with the document's weight of belief that it belongs to the consumer's category
4. The system verifies that the user is an expert and computes a score for each of the documents retrieved by multiplying the document's similarity score with the document's weight of belief that it belongs to the expert's category
5. The system verifies that the user is an expert, and filters the re-ranked list of documents by dropping the documents that do not belong to his favorite categories

Extensions:

5.a The expert hasn't any favorite categories or has disabled this option :
1. The system Exits

Variations:

2. The system could keep only the documents up to a prespecified degree of similarity or the top k documents (where k is a user defined parameter)

UC-14 Rank the documents

This use case is responsible for computing the **similarity score** between each document and the user's query, using the **Vector Space Model** (VSM). According to this model the similarity score is computed as the sum of the inner products of the document's and the query's representation vectors. It's up to the user to select *(see UC-12 Answer the query*) from the following list which type of terms will be used for the representation of both the documents and the query:

document	**query**
AMTEx terms(default)	AMTEx terms(default)
MMTx terms	MMTx terms
lexicographic terms	lexicographic terms
Medline terms*(for Medline documents only)*	AMTEx terms
Userdefined MeSH terms	AMTEx terms

Also, the system retrieves documents based on values in the author's metadata field

Goal in Context

The system produces a ranked list of retrieved documents

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: The query's terms (lexicographic, MMTx and AMTEx terms) have been extracted and the corresponding representation vectors have been created
Level: Subfunction **Scope :** Query Handling subsystem
Triggering Action : The system presents the user with a list of the different types of terms that can be used for the retrieval process

Flow of Events

Main Success Scenario:
1. The system fetches the inverted lists for the query's terms
2. The system intersects them for conjunctive query or merges them for disjunctive query
3. The system assigns a similarity score to each of the documents retrieved
4. The system ranks the retrieved documents according to this similarity score

Extensions:

Variations:

UC-15 Present results to user

This use case is responsible for presenting the retrieved list of documents to the user. The user has the option to change the ranking scheme and get the retrieved list sorted by the document's (publication or creation) **date** or by the document's **authority score** *(see Glossary and UC-5 Create the system’s database step 9)*. Finally, the system allows the user for navigation through the literature in time by following the links to the articles that:

- the current document references and
- cite to the current document providing at the same time the context of each citation i.e. the sentence(s) in which a reference is made to this article.

Goal in Context

The system presents the results to the user

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: A list of relevant documents to the user's query has been retrieved.
Level: Subfunction **Scope :** User Interface Module
Triggering Action : The system presents the total number of the relevant documents retrieved in response to the user's query

Flow of Events

Main Success Scenario:

1. The system presents information for each document from the retrieved list such as the document's title, the author name(s) etc
2. The system ensures that the document is an article, presents the document's metadata i.e. publication date, lists out information about articles that are being referenced by the current article (references), articles that cite to the current article (citations) and the context of each citation
3. The system ensures that the document is an unstructured medical text and presents the date the document was created or last updated and a link to the original URL
4. The user selects from the following list of options and changes the ranking scheme :
   4.1. Order the results by descending date (publication date for articles, creation for unstructured medical texts)
   4.2. Order the results by the document's authority score
5. The User selects a document and views extensive information
6. The system repeats steps 4-5 until the user indicates that he has finished

Extensions:

4.a The user changes the ranking scheme:
   1. The system re-ranks the retrieved list of documents according to the user's selected criteria
   2. The system continues to step 1

5.a The document is an article:
   1. The system indicates the article’s references and citations that exist in the system's database
   2. The user navigates through these articles by following the links (backward

Flow of Events
or forward) **Variations:** 1. The system can present as answers to the query the top r documents, where r is a parameter set by the user 5. The user might select to print, save or e-mail the document

UC-16 Reformulate the user's query

Often the retrieved documents are relevant to the query, but useless to the searcher. In order to improve the search results, our system interacts with the user offering him suggestions to add precision to his query. This is very valuable especially in the medical field where consumers don't have the knowledge or vocabulary to specify what really they want. Often, they need to retrieve relevant documents which may not contain any occurrences of the original user's query terms. This use case describes the query expansion process in order to help the users refine their queries. The query is enriched with related terms such as :

- MeSH terms which are automatically extracted from the top-ranked documents using the AMTEx method *(see Glossary)*. Alternatively, the user can suggest the documents that he considers to be relevant and the MeSH terms can be extracted from these documents. The presence of the Direct Index (*see UC-11 Build indexes for the document's terms step 4*), makes this expansion efficient.
- MeSH terms in the neighbourhood of a query term (i.e. terms lower or higher in the MeSH hierarchy) using a **similarity threshold** between 0 and 1. High values of this threshold are desirable for very specific queries (in this case few terms from the MeSH hierarchy are used for query expansion) while lower values are desirable for too general queries where many documents can be retrieved. Notice though that very low values for the threshold might de-focus the query.
- synonyms (**entry terms**) from the MeSH hierarchy, e.g. 'pain' and 'ache' are synonyms

Goal in Context

The user's query is expanded with new terms

Details

Primary Actors: Consumer, Expert
Supporting Actors:
Preconditions: The user has entered a query and retrieved relevant documents in response to this query
Level: User **Scope :** User Interface Module
Triggering Action : The system provides the user with a list of options

Flow of Events

Main Success Scenario:

1. The user has the following list of options to select the source of the terms that will be used for the query expansion:
    1.1. AMTEx terms
        1.1.1. The system extracts the MeSH terms from the top r retrieved documents using the AMTEx method
        1.1.2. The system performs frequency analysis and a frequency score is associated to each term
        1.1.3. The system ranks these terms according to frequency
        1.1.4. The system presents the terms in order of decreasing frequency and a brief explanation for each MeSH term e.g. display the 'Scope Note' field from MeSH to inform the users, especially the consumers.
    1.2. MeSH terms from the MeSH hierarchy using a similarity threshold

Flow of Events
1.3. synonyms (entry terms) from the MeSH hierarchy 2. The user selects the list of terms that interest him 3. The system expands the query with these new terms 4. The system re-weights the expanded query **Extensions:** **Variations:** 1.1.1a The user could select which of the retrieved documents are relevant and the system extracts the MeSH terms from these documents 1.1.1b The user could select the number r of the top-ranked retrieved documents (the system Administrator defines a default value) that will be used to extract the MeSH terms for the query expansion 1.2. The user can update the value of the similarity threshold.

UC-17 Browse database by medical topics

This use case allows the expert to browse the system's database documents by medical topics. Our system can support three different types of categorization for the documents: the Semantic Network MeSH categories ,categories produced during the clustering process (based on the Citation Graph) and categories produced during the document clustering process.
The user can switch between these different types of categories and retrieve different results.

Goal in Context

The system allows the expert to view the documents by browsing the medical topics

Details

Primary Actors: Expert
Supporting Actors:
Preconditions: The expert is logged on
Level: User **Scope :** Browsing subsystem
Triggering Action : The expert selects the type of categories he wants to use for browsing

Flow of Events

Main Success Scenario:

1. The system presents the medical topics
2. The user selects the topics that interest him
3. The system retrieves all the documents that belong to this topic (and to the topic's children etc)
4. The system computes a score for each of the documents retrieved by multiplying the document's relevance score with the document's weight of belief that it belongs to the expert's category
5. The system presents the results to the user *(*UC-15*)*
6. The system repeats steps 1-5 until the user indicates that he has finished with the browsing process

Extensions:

Variations:

3.4 ACTIVITY DIAGRAMS

The activity diagrams are used to model a use case scenario in great detail. Each activity diagram starts with a *black circle* and stops at (at least one) *concentric white/black circles*. The actions are *eclipses.* When an action is decomposed into sub-actions, *boxes with rounded ends* are used. From each action comes out an *arrow*, called transition, connecting it to the next one. The arrow can also lead to a *diamond*, which indicates branching into two or more mutually exclusive transitions. Expressions inside brackets [ ], e.g. [username exists] or [username doesn't exist], label the transitions coming out of a branch. Also a transition might fork into two or more parallel activities e.g. the activities 'Categorize the document to consumer/expert category' and 'Categorize the document to Semantic Network MeSH categories' (*see Activity Diagram#8*) can be performed in parallel. When both activities have finished the flow continues. Each fork must have a subsequent join. Forks and joins both appear in the activity diagram as *solid bars*.

The seventeen activities diagrams (one for each of the use cases described above) follow:

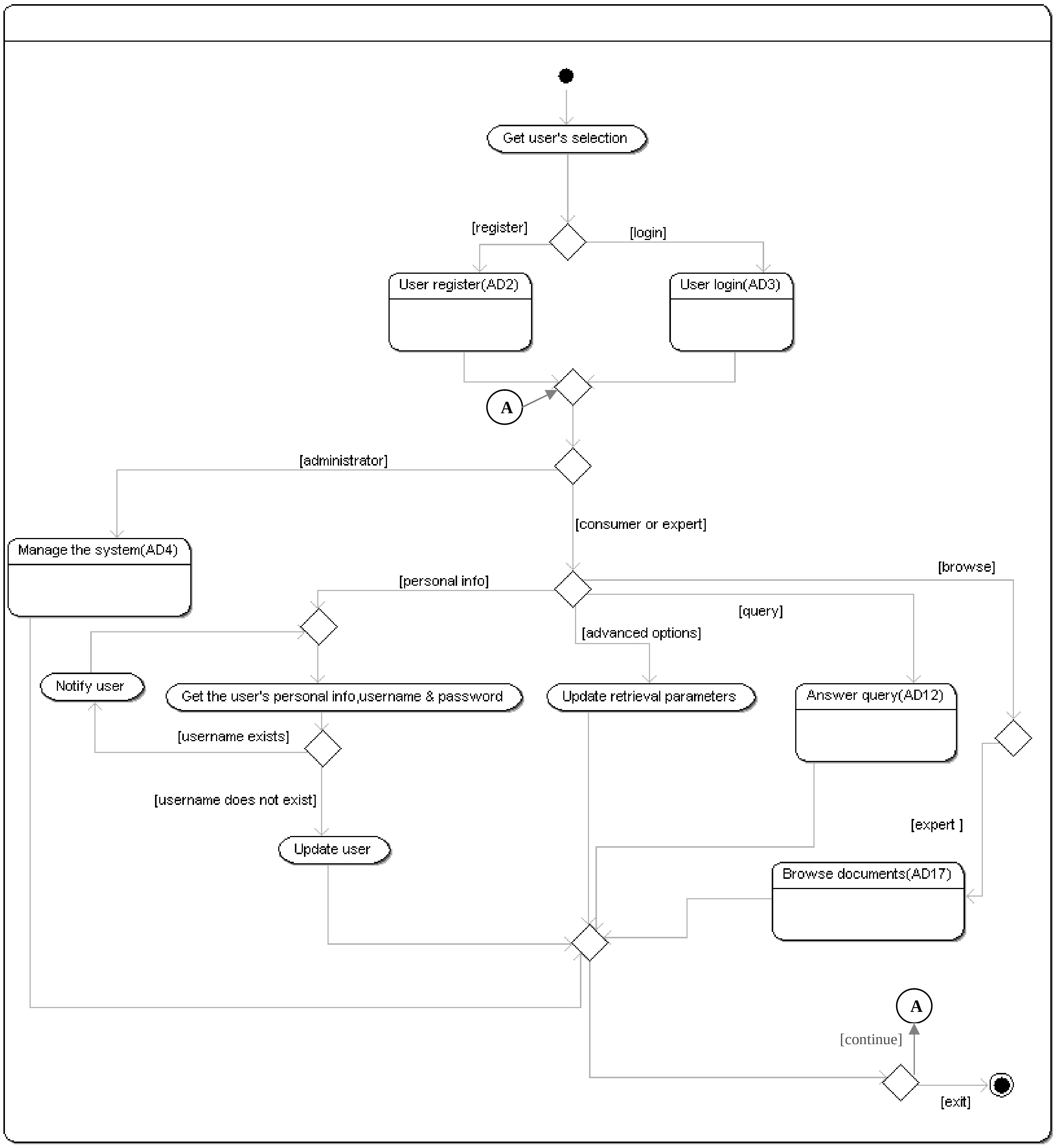


Figure 3.5: Activity diagram#1 – Use the application

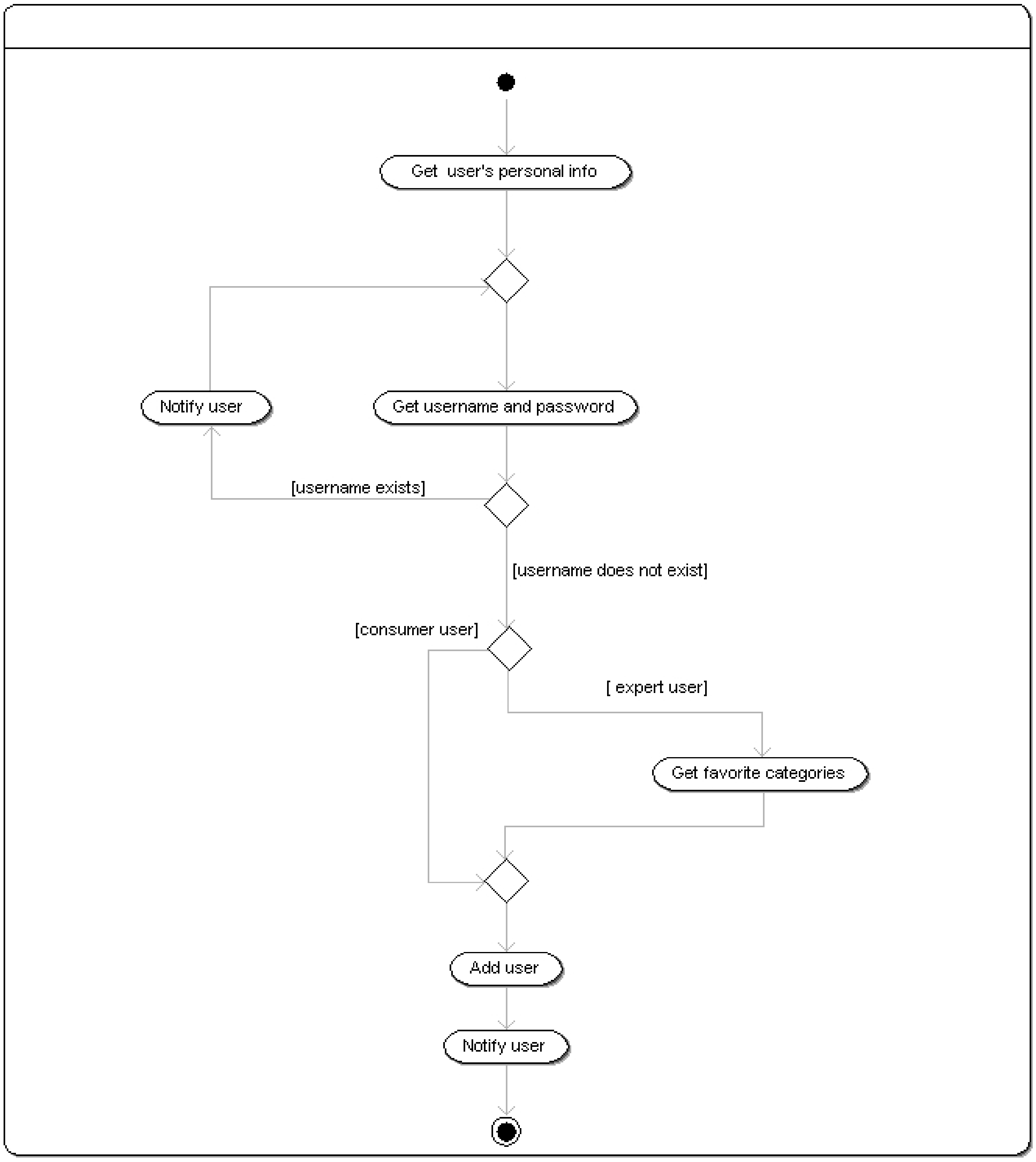


Figure 3.6: Activity diagram#2 – User registration

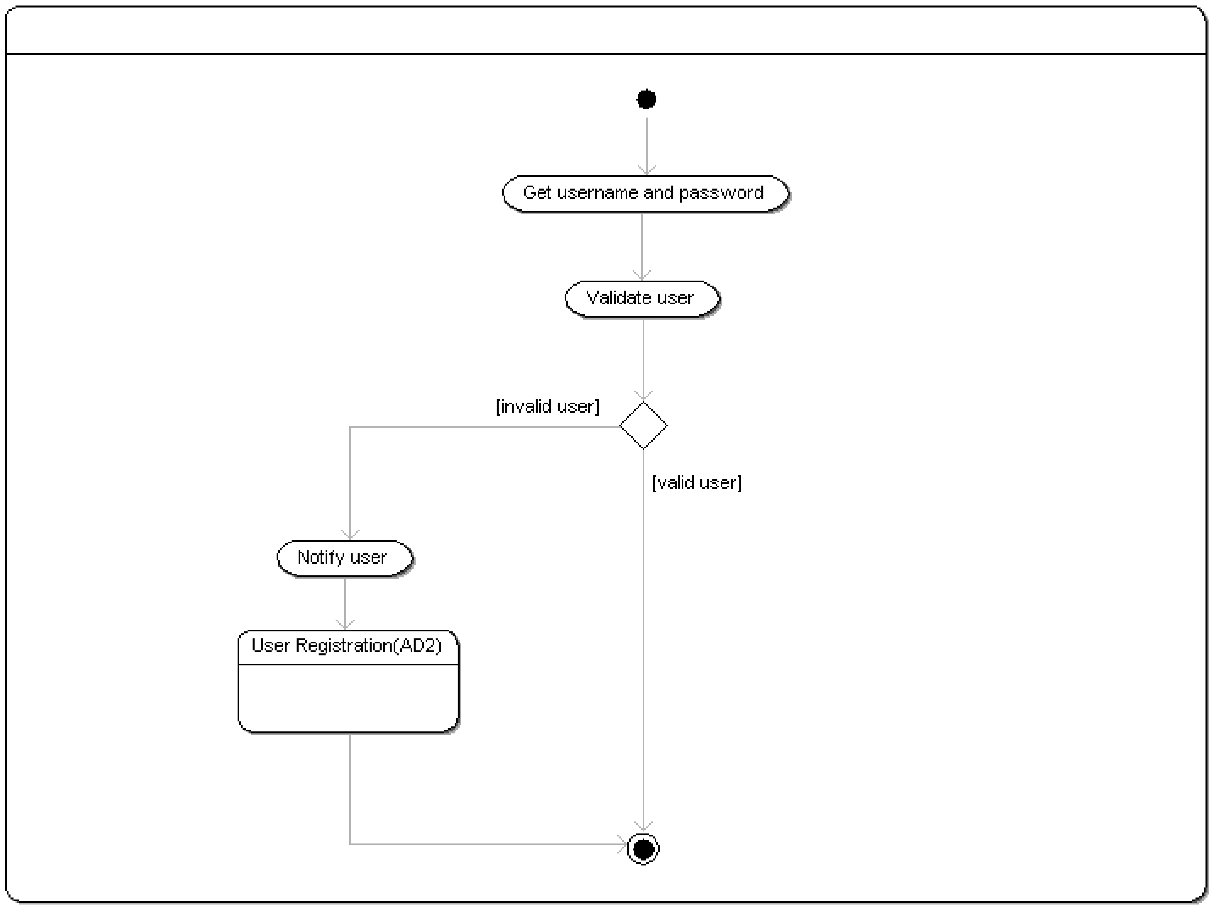


Figure 3.7: Activity diagram#3 – User login

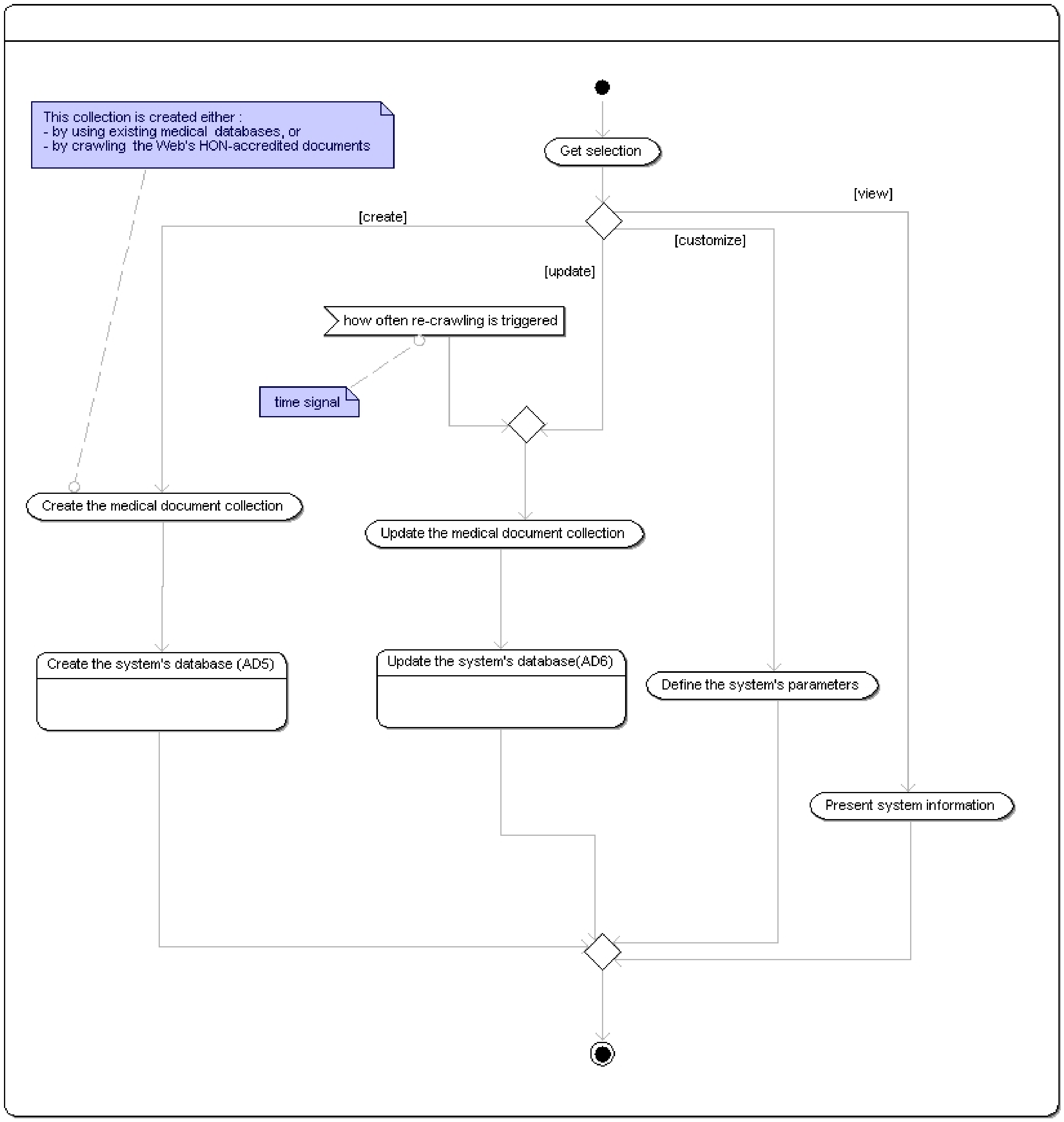

This collection is created either :
- by using existing medical databases, or
- by crawling the Web's HON-accredited documents
Get selection
[view]
[create]
[customize]
[update]
how often re-crawling is triggered
time signal
Create the medical document collection
Update the medical document collection
Create the system's database (AD5)
Update the system's database(AD6)
Define the system's parameters
Present system information

Figure 3.8: Activity diagram#4 – System management

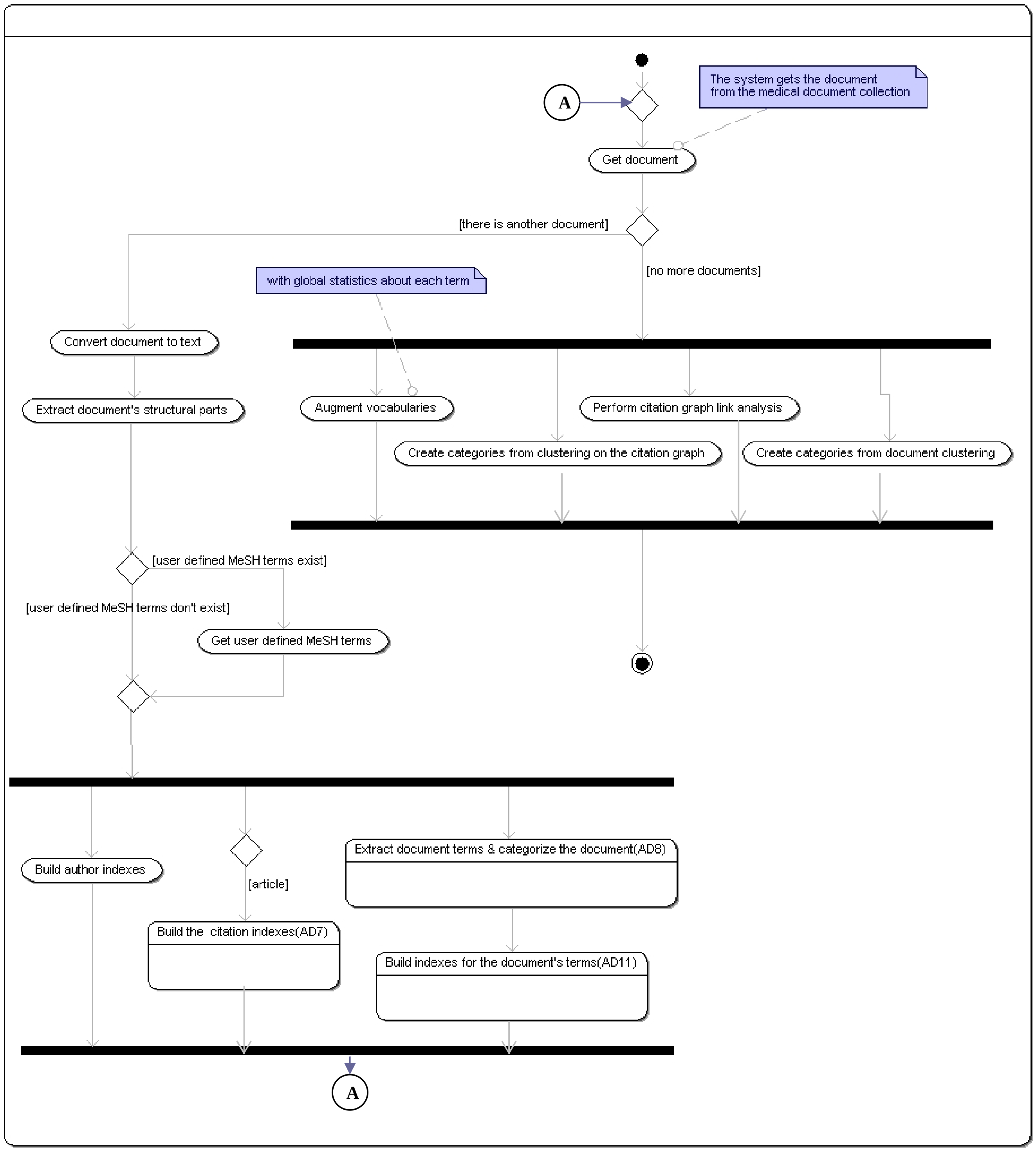

A
The system gets the document from the medical document collection
Get document
[there is another document]
[no more documents]
with global statistics about each term
Convert document to text
Extract document's structural parts
Augment vocabularies
Perform citation graph link analysis
Create categories from clustering on the citation graph
Create categories from document clustering
[user defined MeSH terms exist]
[user defined MeSH terms don't exist]
Get user defined MeSH terms
Build author indexes
[article]
Extract document terms & categorize the document(AD8)
Build the citation indexes(AD7)
Build indexes for the document's terms(AD11)
A

Figure 3.9: Activity diagram#5 – Create the System's database

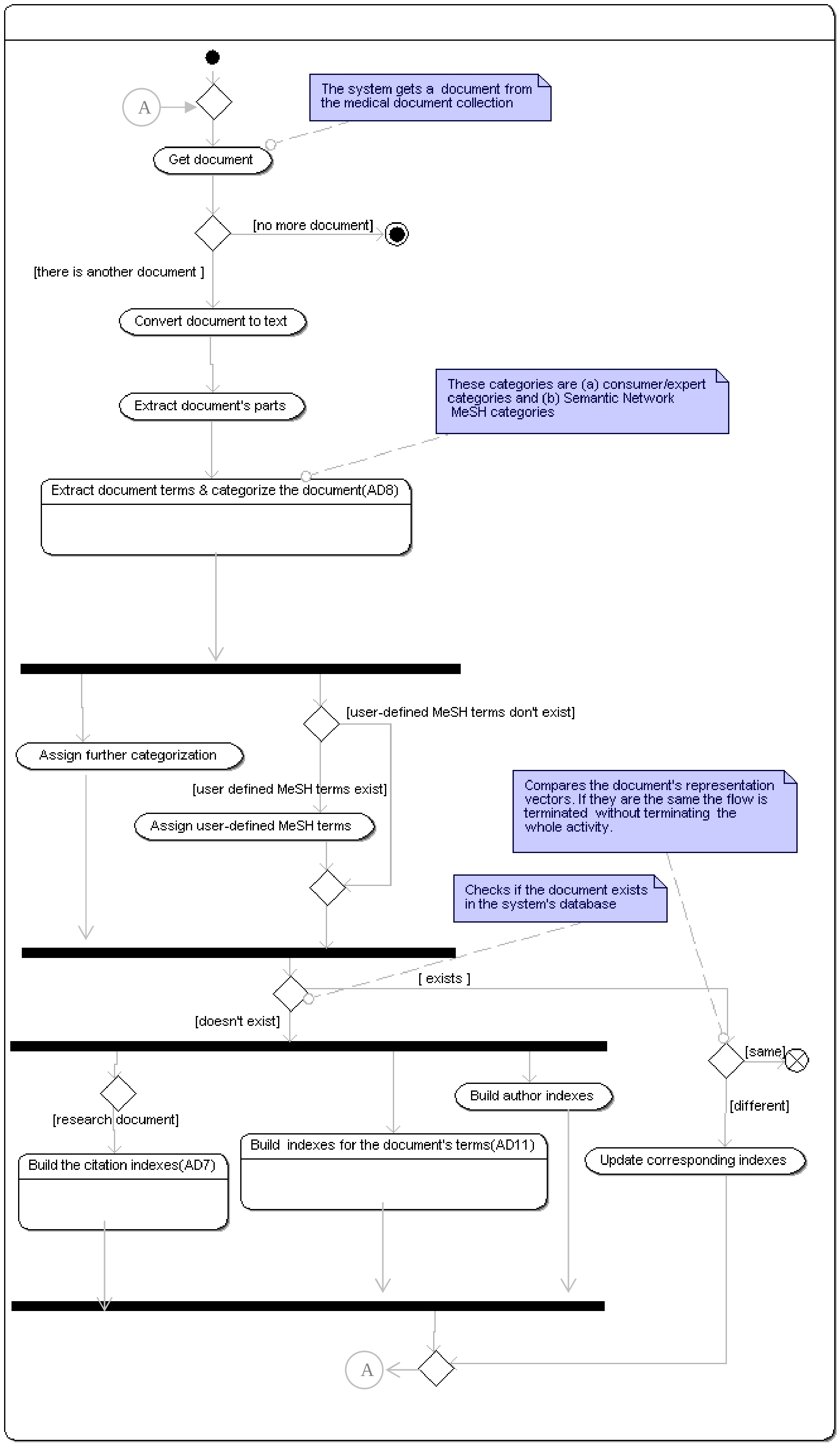

A
The system gets a document from the medical document collection
Get document
[no more document]
[there is another document]
Convert document to text
Extract document's parts
These categories are (a) consumer/expert categories and (b) Semantic Network MeSH categories
Extract document terms & categorize the document(AD8)
[user-defined MeSH terms don't exist]
Assign further categorization
[user defined MeSH terms exist]
Assign user-defined MeSH terms
Compares the document's representation vectors. If they are the same the flow is terminated without terminating the whole activity.
Checks if the document exists in the system's database
[exists]
[doesn't exist]
[same]
[different]
[research document]
Build author indexes
Build the citation indexes(AD7)
Build indexes for the document's terms(AD11)
Update corresponding indexes
A

Figure 3.10: Activity diagram#6 – Update the System's database

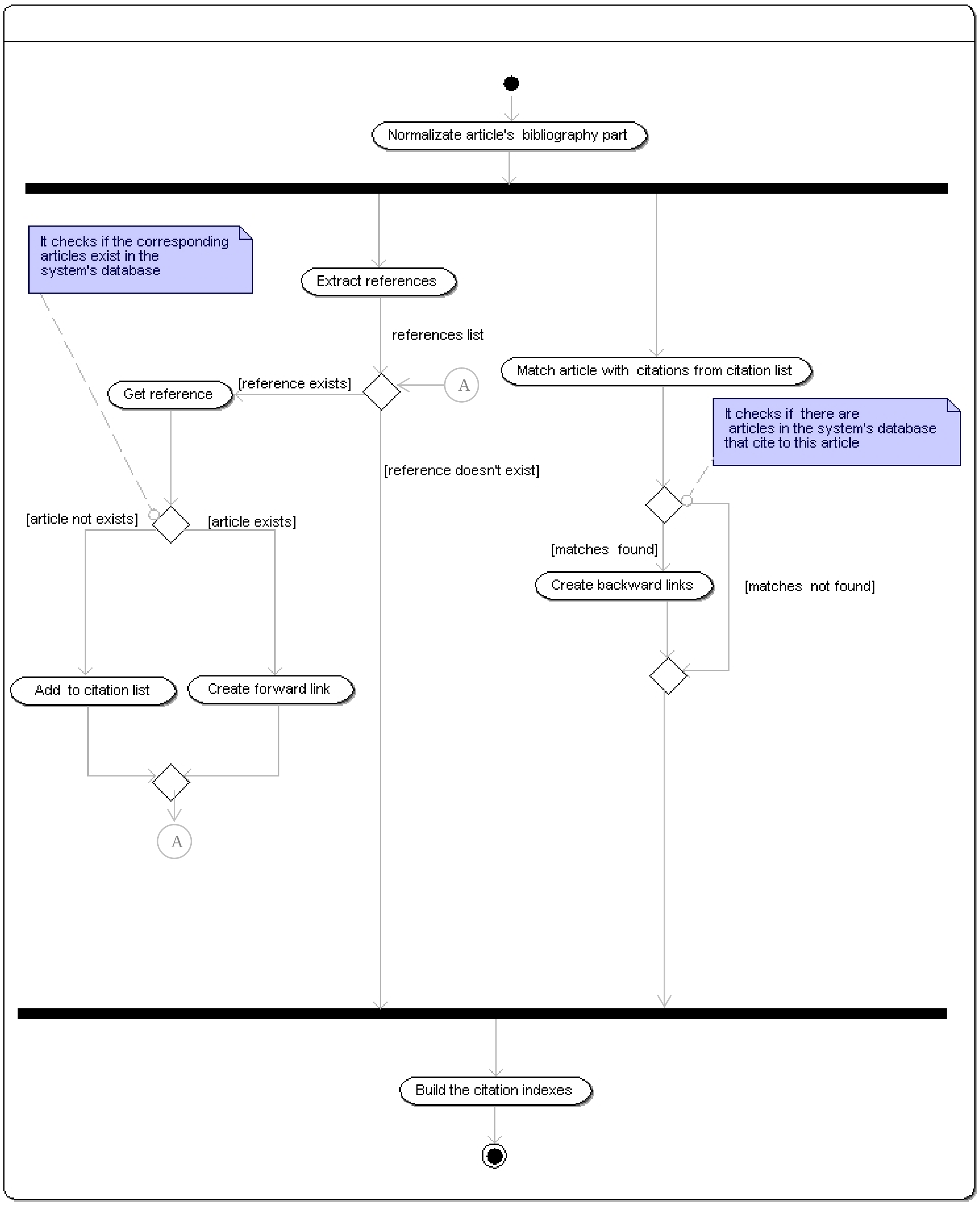

Normalizate article's bibliography part
Extract references
references list
It checks if the corresponding articles exist in the system's database
Get reference
[reference exists]
A
[reference doesn't exist]
[article not exists]
[article exists]
Add to citation list
Create forward link
A
Match article with citations from citation list
It checks if there are articles in the system's database that cite to this article
[matches found]
Create backward links
[matches not found]
Build the citation indexes

Figure 3.11: Activity diagram#7 – Build citation indexes

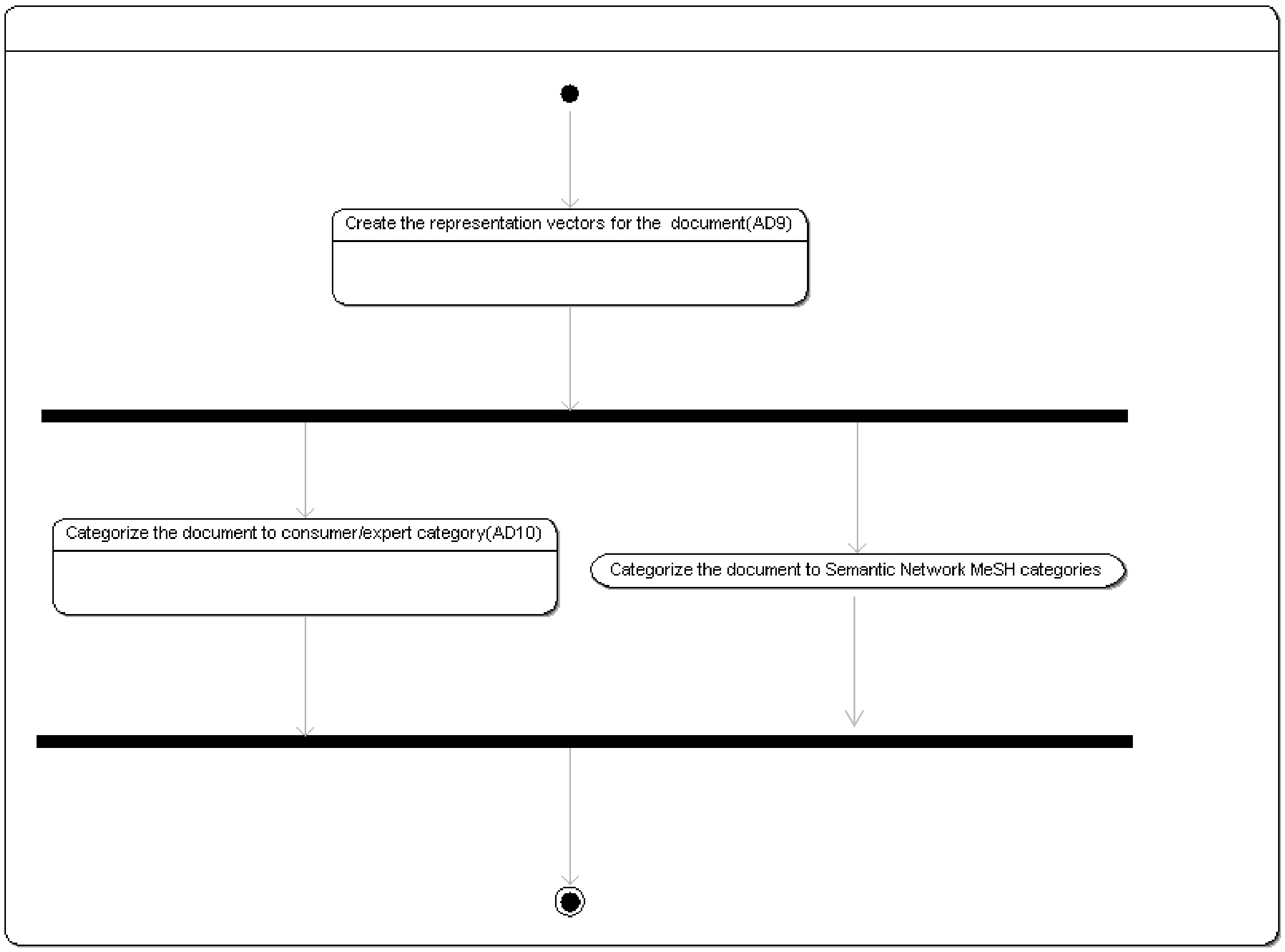

Create the representation vectors for the document(AD9)
Categorize the document to consumer/expert category(AD10)
Categorize the document to Semantic Network MeSH categories

Figure 3.12: Activity diagram#8 – Extract the document's terms and categorize the document

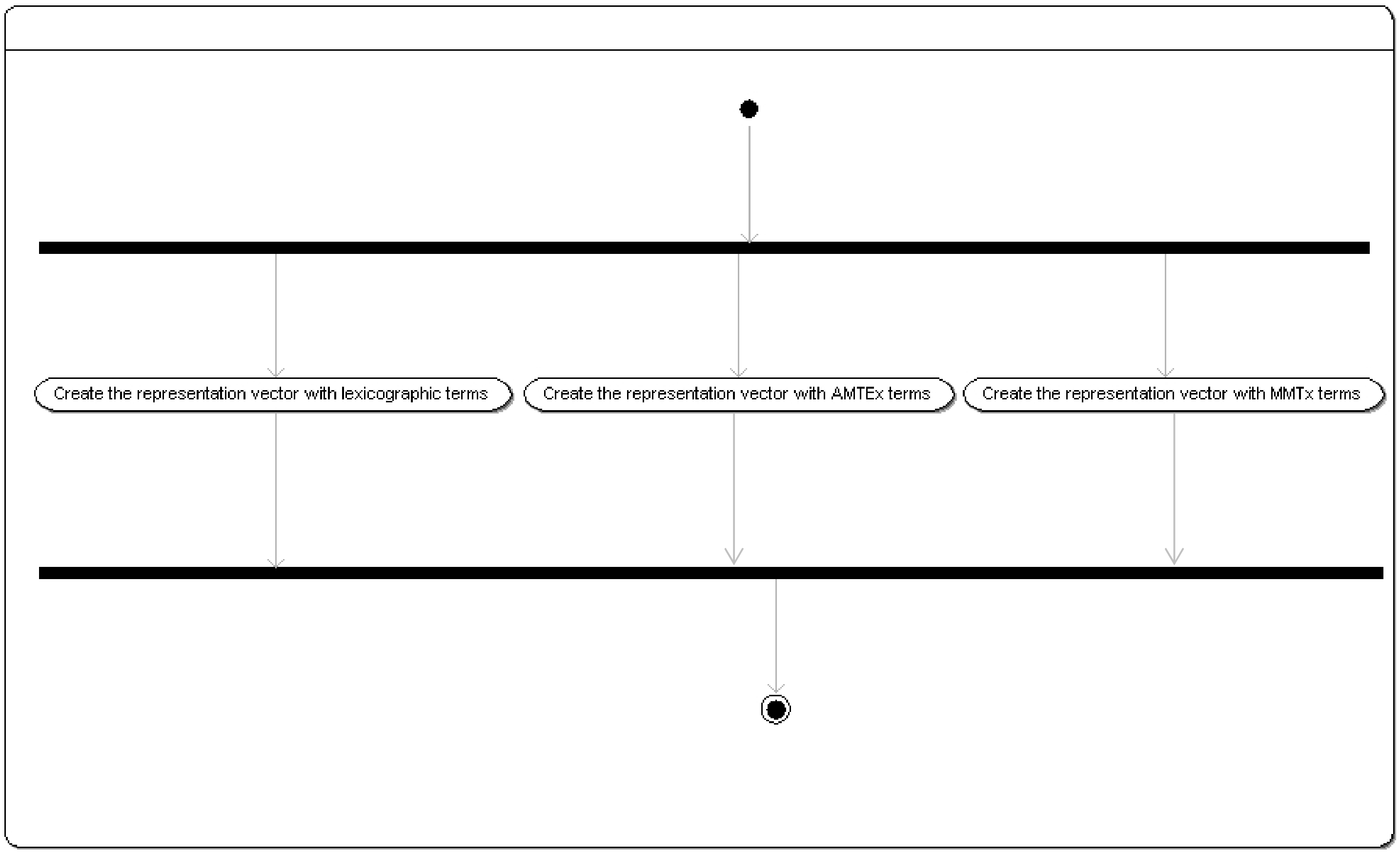

Create the representation vector with lexicographic terms
Create the representation vector with AMTEx terms
Create the representation vector with MMTx terms

Figure 3.13: Activity diagram#9 – Create the representation vectors for the document or the user's query

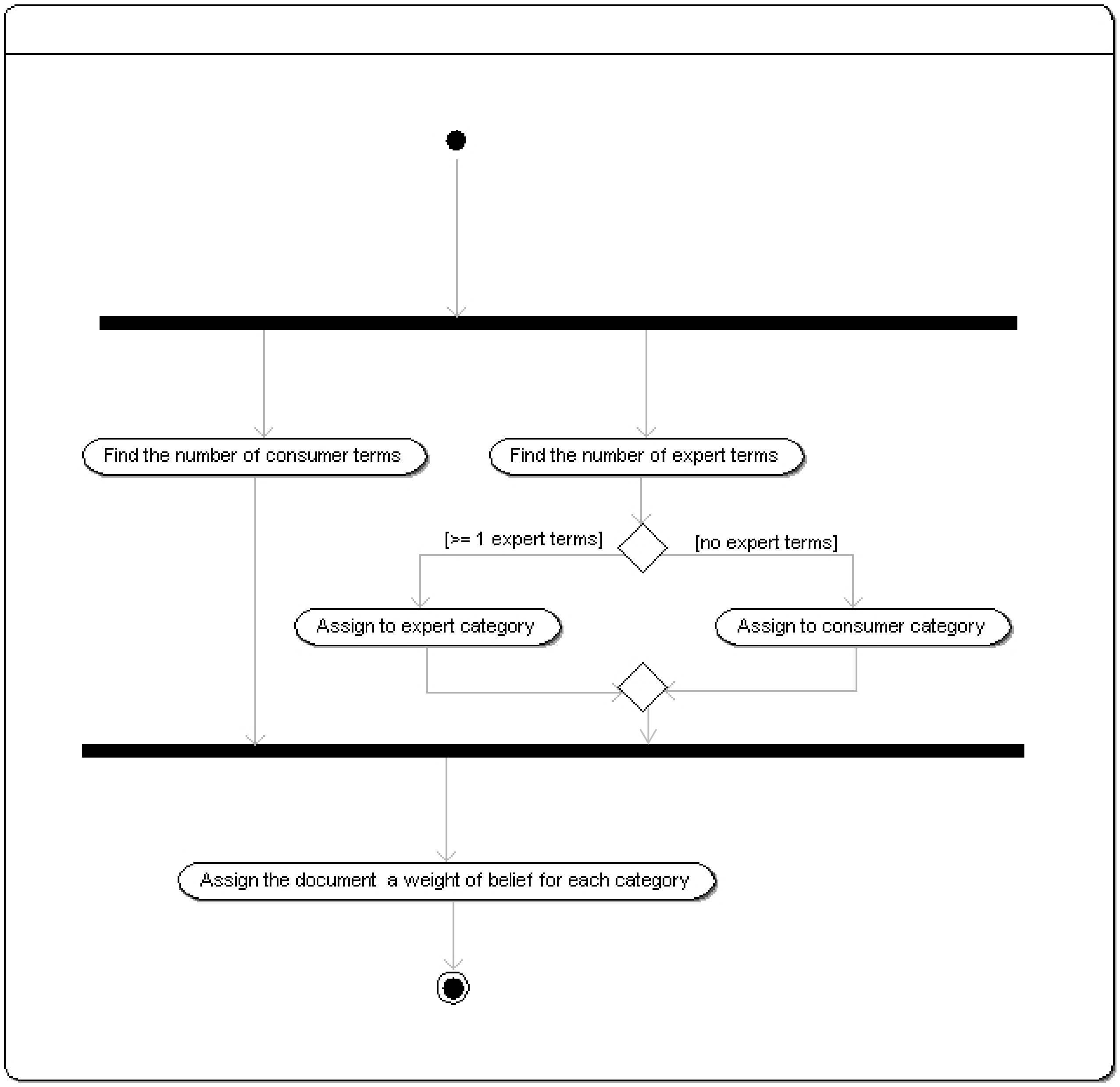

Find the number of consumer terms
Find the number of expert terms
[>= 1 expert terms]
[no expert terms]
Assign to expert category
Assign to consumer category
Assign the document a weight of belief for each category

Figure 3.14: Activity diagram#10 – Categorize the document to consumer/expert category

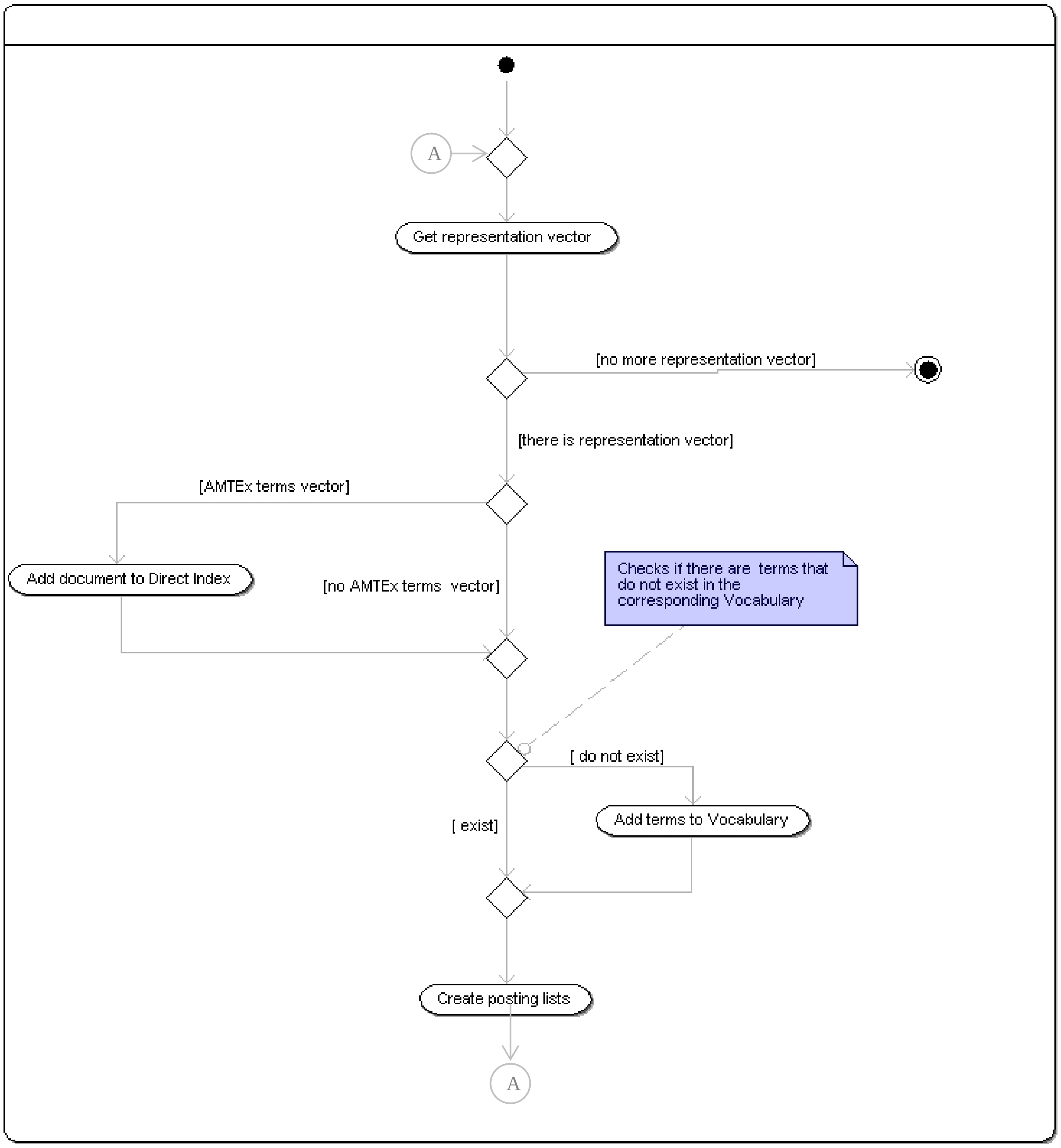

A
Get representation vector
[no more representation vector]
[there is representation vector]
[AMTEx terms vector]
Add document to Direct Index
[no AMTEx terms vector]
Checks if there are terms that do not exist in the corresponding Vocabulary
[do not exist]
[exist]
Add terms to Vocabulary
Create posting lists
A

Figure 3.15: Activity diagram#11 – Build indexes for the document's terms

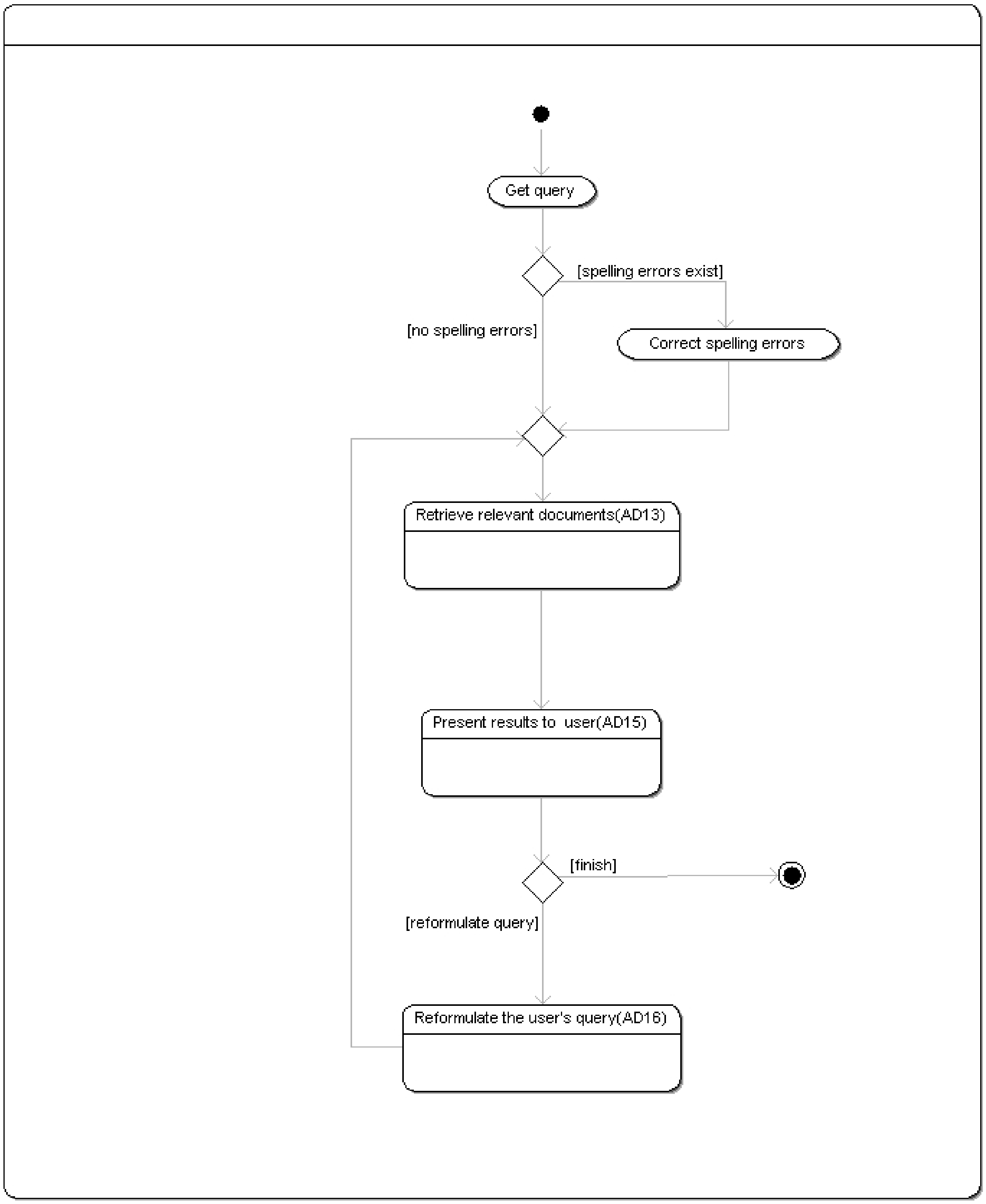


Figure 3.16: Activity diagram#12 – Answer the query

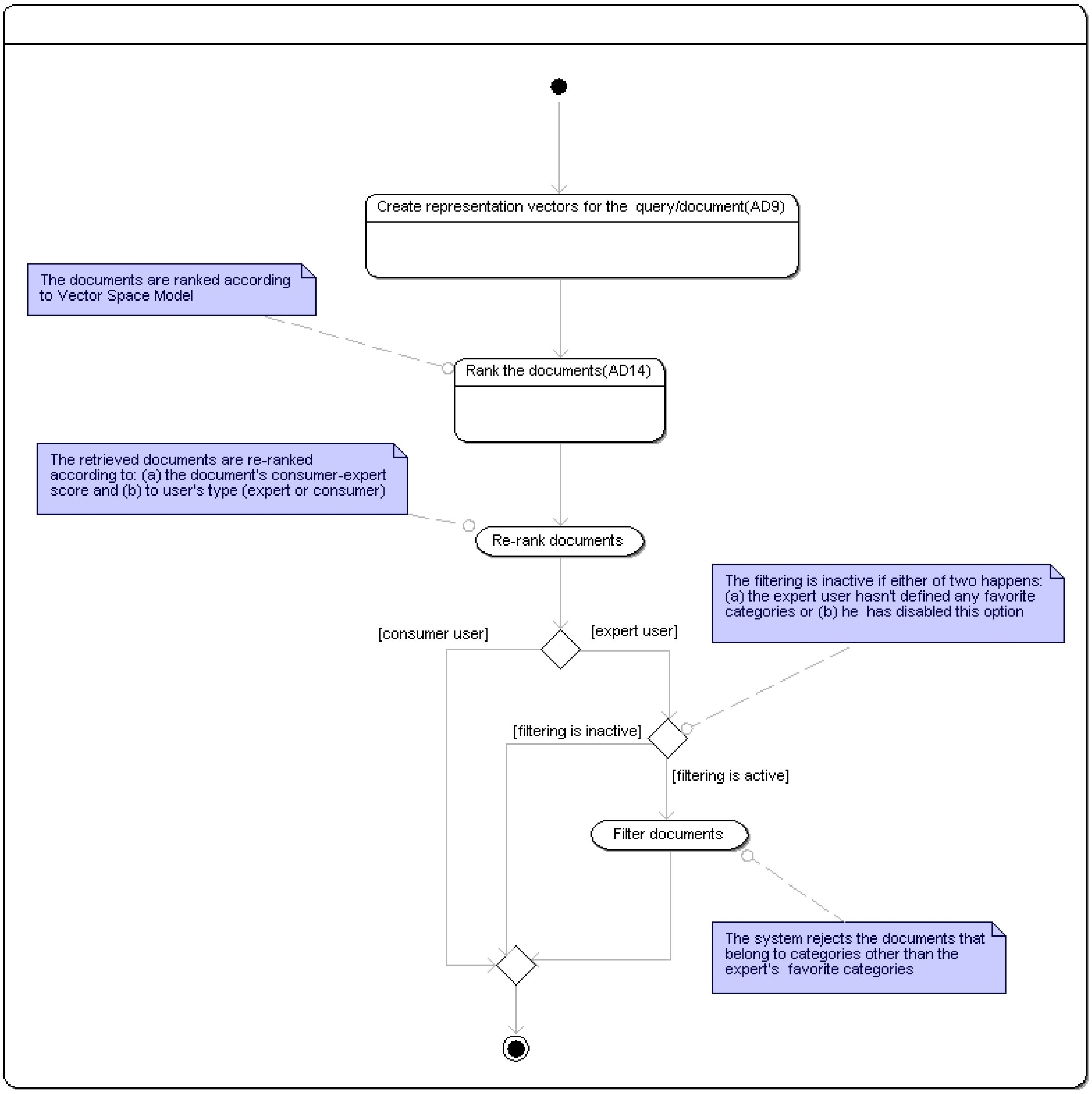

Create representation vectors for the query/document(AD9)
The documents are ranked according to Vector Space Model
Rank the documents(AD14)
The retrieved documents are re-ranked according to: (a) the document's consumer-expert score and (b) to user's type (expert or consumer)
Re-rank documents
The filtering is inactive if either of two happens: (a) the expert user hasn't defined any favorite categories or (b) he has disabled this option
[consumer user]
[expert user]
[filtering is inactive]
[filtering is active]
Filter documents
The system rejects the documents that belong to categories other than the expert's favorite categories

Figure 3.17: Activity diagram#13 – Retrieve relevant documents

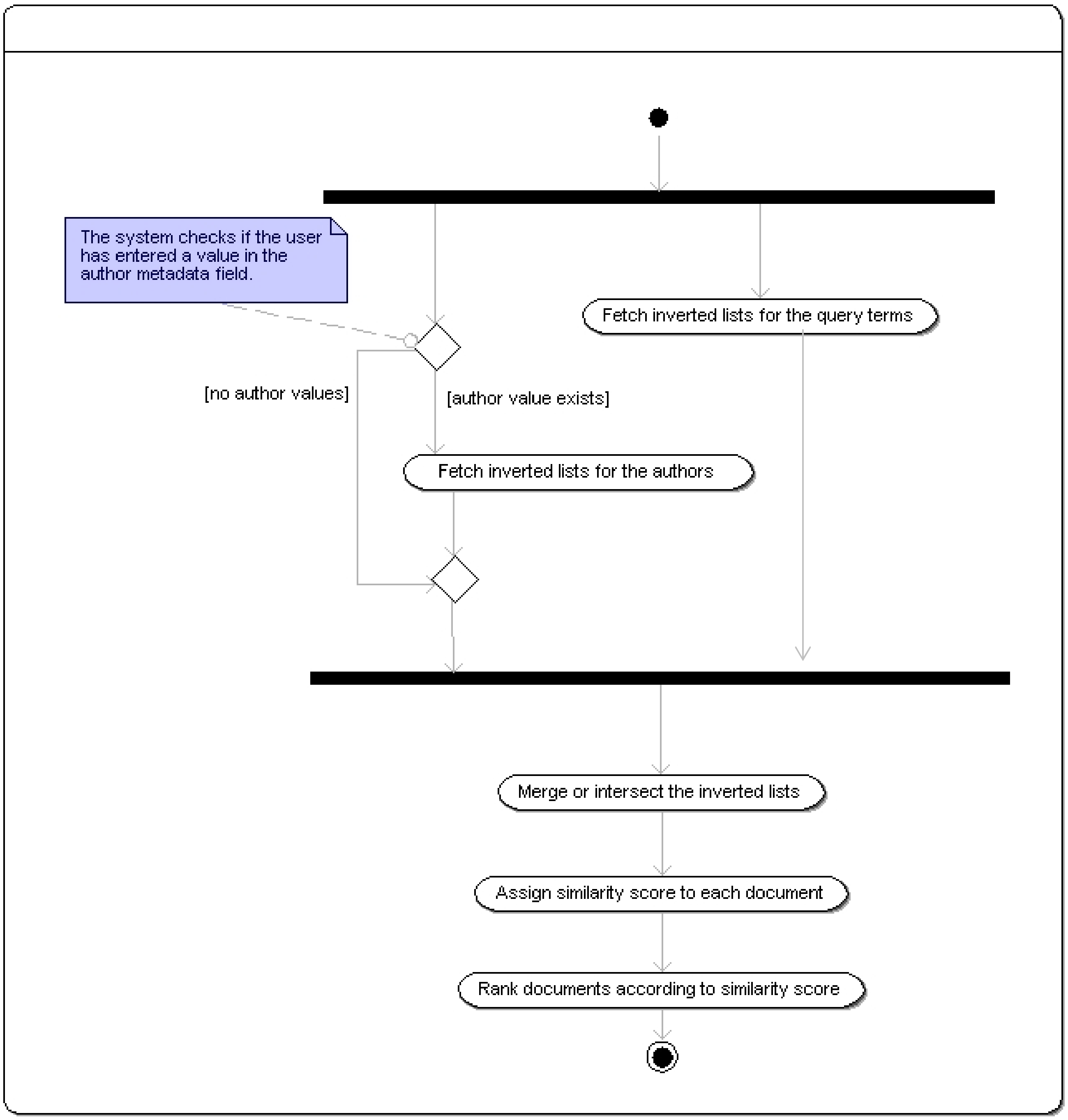

The system checks if the user has entered a value in the author metadata field.
Fetch inverted lists for the query terms
[no author values]
[author value exists]
Fetch inverted lists for the authors
Merge or intersect the inverted lists
Assign similarity score to each document
Rank documents according to similarity score

Figure 3.18: Activity diagram#14 – Rank the documents

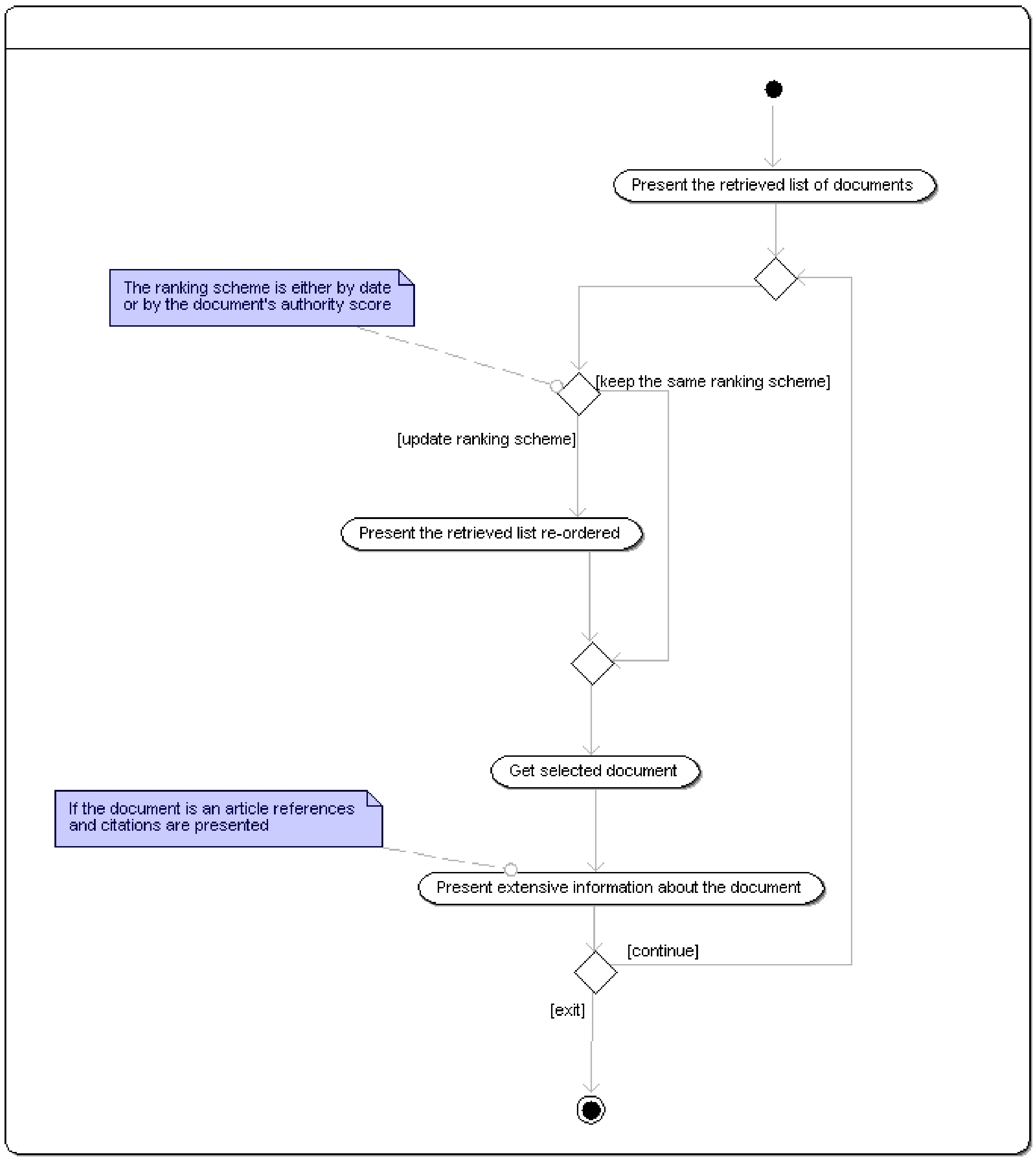

Present the retrieved list of documents
The ranking scheme is either by date or by the document's authority score
[keep the same ranking scheme]
[update ranking scheme]
Present the retrieved list re-ordered
Get selected document
If the document is an article references and citations are presented
Present extensive information about the document
[continue]
[exit]

Figure 3.19: Activity diagram#15 – Present results to user

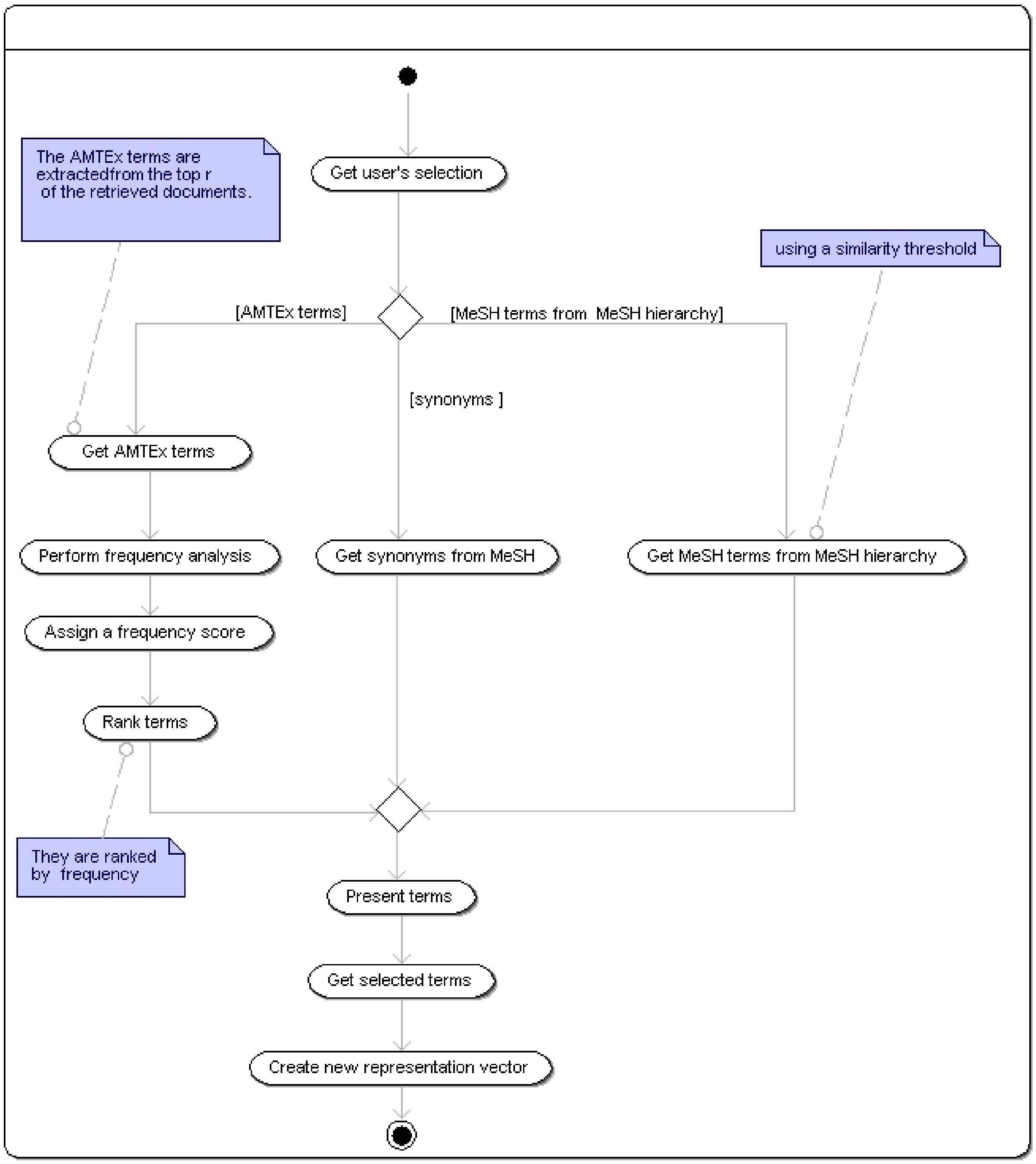

The AMTEx terms are extractedfrom the top r of the retrieved documents.
Get user's selection
using a similarity threshold
[AMTEx terms]
[MeSH terms from MeSH hierarchy]
[synonyms]
Get AMTEx terms
Perform frequency analysis
Get synonyms from MeSH
Get MeSH terms from MeSH hierarchy
Assign a frequency score
Rank terms
They are ranked by frequency
Present terms
Get selected terms
Create new representation vector

Figure 3.20: Activity diagram#16 – Reformulate the user's query

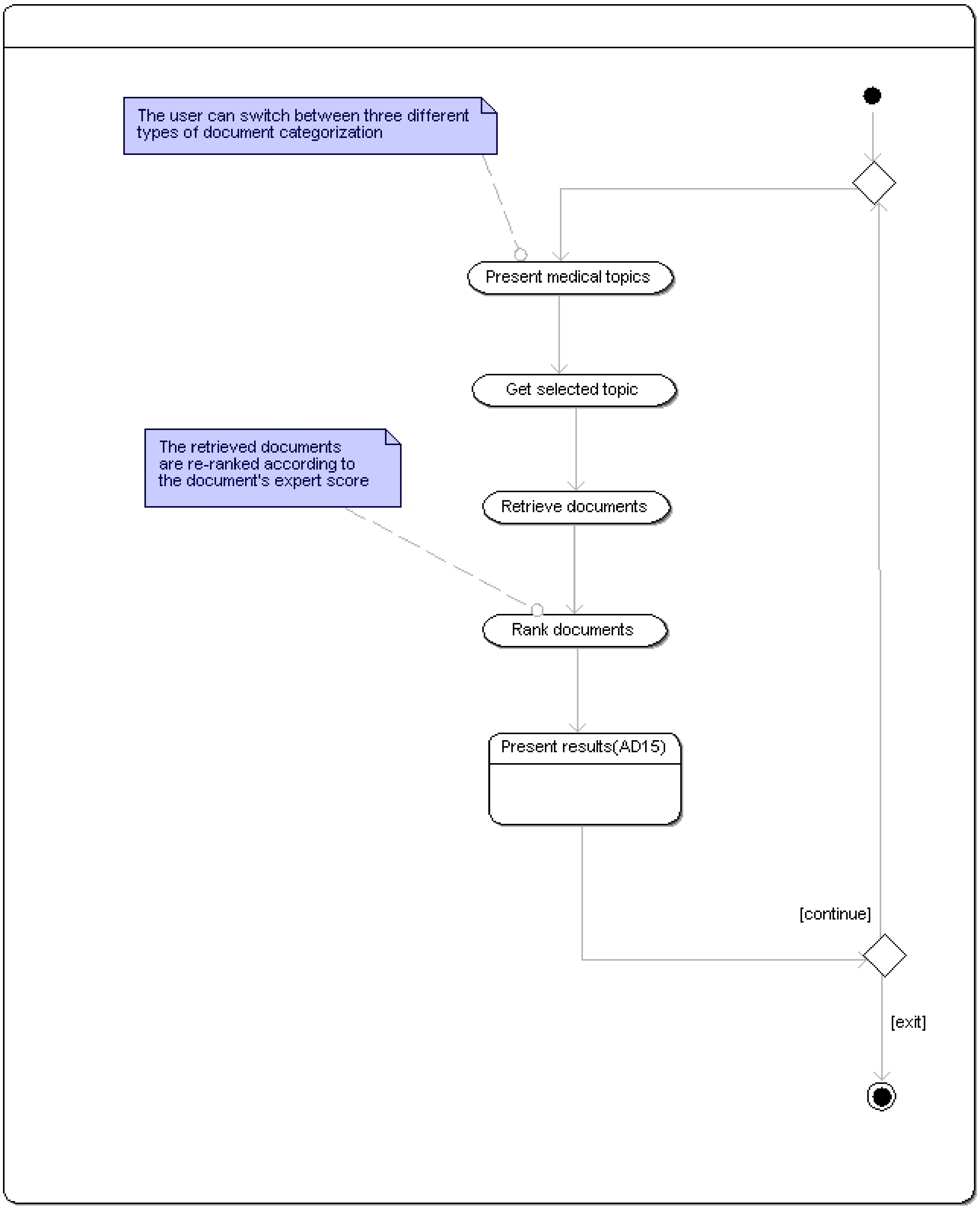

The user can switch between three different types of document categorization
Present medical topics
Get selected topic
The retrieved documents are re-ranked according to the document's expert score
Retrieve documents
Rank documents
Present results(AD15)
[continue]
[exit]

Figure 3.21: Activity diagram#17 – Browse database by medical topics

3.5 CLASS DIAGRAMS

Class diagrams describe the classes inside the system and the relationships between them. As the proposed system is large, the system's classes are organized into packages (*see Figure 3.1- package diagram*).

The elements of a class diagram that this work uses are:

- A class is drawn as a *rectangle* with three compartments. The first describes the class's name, For naming the class two rules have been followed [6]: (a) the vocabulary of the medical domain is used and (b) all names are singular nouns starting with a capital letter. The second compartment displays the class's attributes, Here usually only the single valued attributes are described. The multi-valued or the complex attributes form new classes and an association link between the classes is shown e.g. the classes Date, Address, Phone (*see Figure 3.22 - System Management Package-class diagram*). Finally, the class's operations are displayed in the third compartment and represent the behaviour of the class. When a class is imported by another package, then only the class's name and namespace is displayed (e.g. see *Figure 3.25 - User Interface Package-class diagram*, where the class *User_Management_Package::loginManager* is imported).
- The system's class diagrams have three kinds of relationships:
  - association which is the simplest type of relationship between the instances of two classes, shown as a *uni or bi directional line* connecting the two classes,
  - aggregation is between the whole and its parts and is a stronger relationship than association. It is shown as a line with a white diamond next to the class representing the whole e.g. the class SemanticData(*see Figure 3.22 - System Management Package-class diagram*) consists of a collection of Medline_terms, AMTEx_terms etc
  - composition which is a stronger relationship than aggregation. Here the whole has responsibility for the existence of its parts, e.g. if parsedQuery (*see Figure 3.23 – Document Retrieval Package-class diagram*) is deleted then MMTx_terms, AMTEx_terms etc are also deleted.

The relationships mentioned above have multiplicity numbers on their edges (they define how many objects participate in this relationship).

Also the system's class diagrams have inheritance relationships (one class, the child, inherits the functionality of another class, super class, and then adds new functionality of its own). To model inheritance on a class diagram, a solid line is drawn from the child class with a triangle pointing to the super class (*see Figure 3.22 - System Management Package-class diagram*) where an Expert is a registeredUser with extra functionality.

3.5.1 CLASS DIAGRAM#1 – SYSTEM MANAGEMENT PACKAGE

Here follows a brief description of the most important classes of the System's Management Package (*see Figure 3.22*):

Class Name	Description
MedicalDocumentCollection	This class models a set of medical documents which is either a **Medical_Database** or a **CrawledCollection**.
Document	This class encapsulates the most fundamental concept to indexing which is a medical document. It is essentially an iterator over the documents in the **MedicalDocumentCollection**
SemanticData	This class encapsulates the central aspects that can be found in the document's contents as well as other significant information discussed in the document. These are a collection of: **Userdefined_MeSH_term**s, **Medline_term**s,**AMTEx_term**s and **MMTx_term**s.
Descriptive_Data	This class encapsulates the document's metadata such as the **AuthorInfo**, and the document's publication **Date**.
LexicographicTerm	This class models the document's lexicographic terms.
Article	This class encapsulates another fundamental concept for indexing which is a medical article. Each article has a collection of **CitationMetadata**, belongs to a Categories (**Category**) and is published to a source (**SourceInfo**) by a publisher (**PublisherInfo**).
User	This class models the different types of users. Each user is either the **Administrator** or a **registeredUser** (**Consumer** or **Expert**).

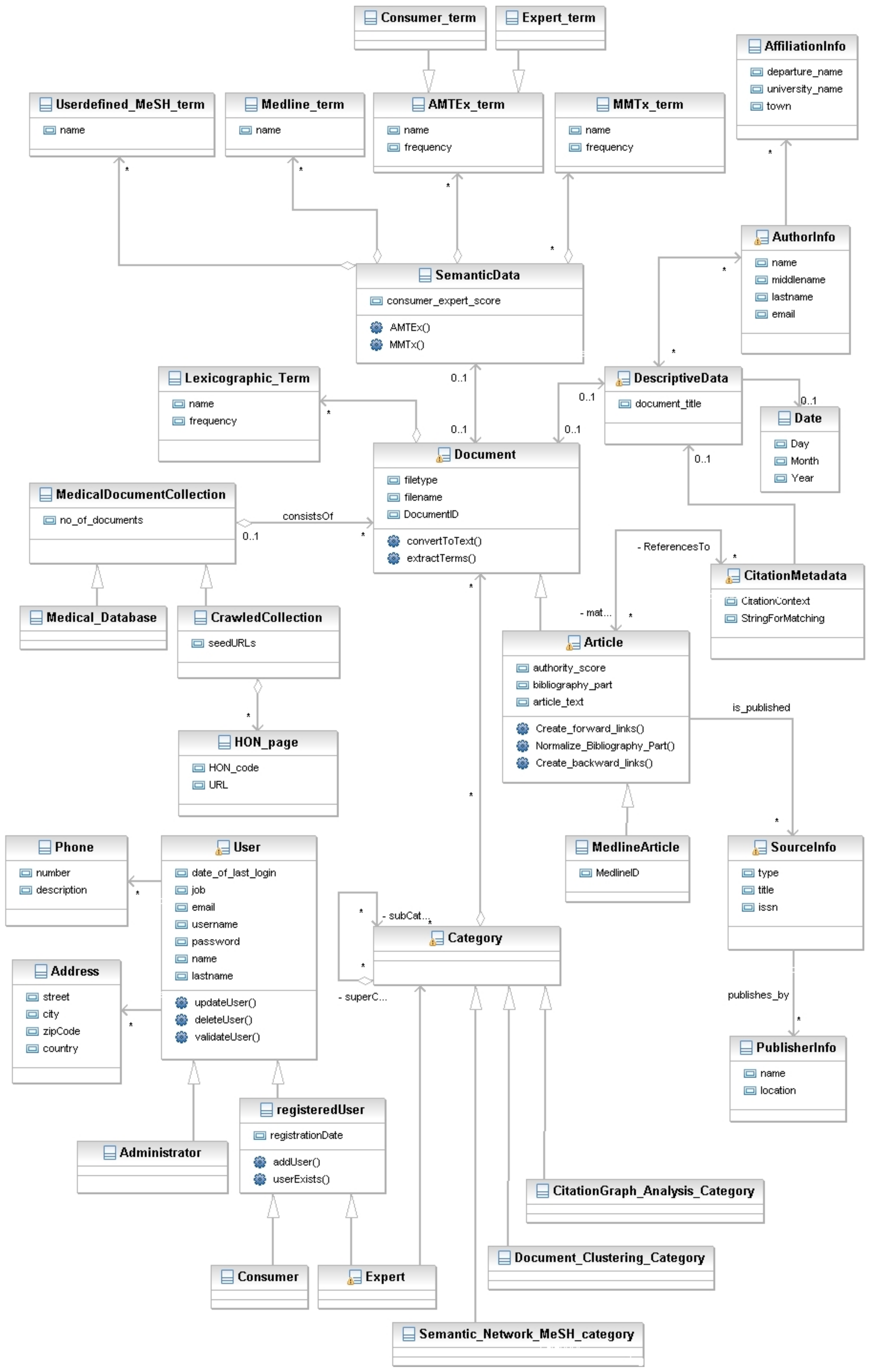


Figure 3.22: System Management Package – Class diagram

3.5.2 CLASS DIAGRAM#2 – DOCUMENT RETRIEVAL PACKAGE

Here follows a brief description of the most important classes of the Document Retrieval Package (*see Figure 3.23*):

Class Name	Description
parsedQuery	This class encapsulates the query terms that will be used for matching the query with the stored documents. These terms are a collection of: **Lexicographic_term**s, **MMTx_term**s and **AMTEx_term**s
expandedQuery	This class encapsulates the query terms after expansion with new **AMTEx_term**s, **MeSH_hierarchy_term**s or **synonym_term**s
resultSet	This class models a result set of documents in respect to the user's query to a medical **Category** (selected by the user). The result set is a collection of documents (**DocumentRetrieved)**.

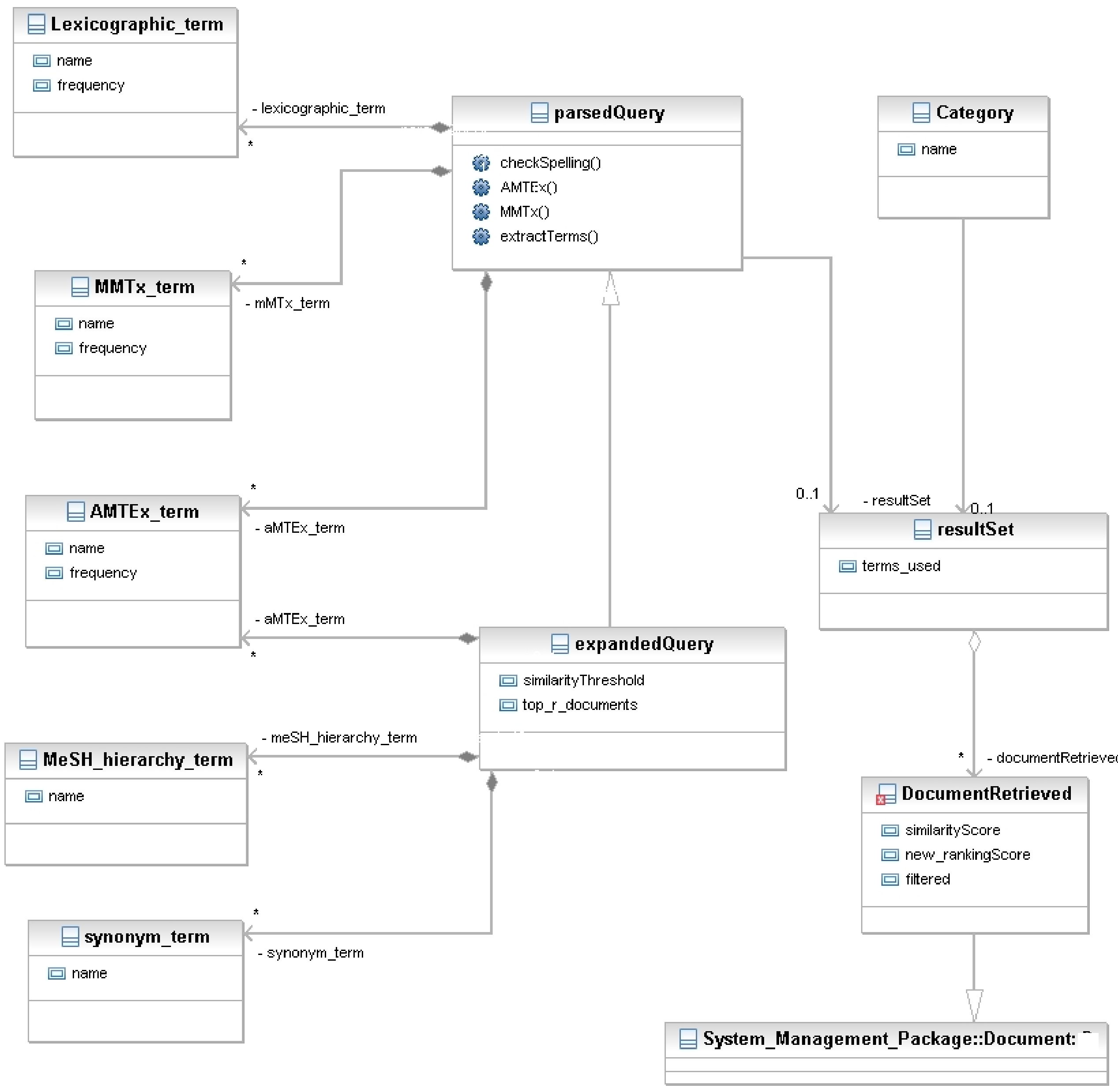


Figure 3.23 : Document Retrieval Package – Class diagram

3.5.3 CLASS DIAGRAM#3 – USER MANAGEMENT PACKAGE

Here follows a brief description of the most important classes of the User Management Package (*see Figure 3.24*).

Class Name	Description
loginManager	This class is responsible for user's login
registrationManager	This class is responsible for user's registration

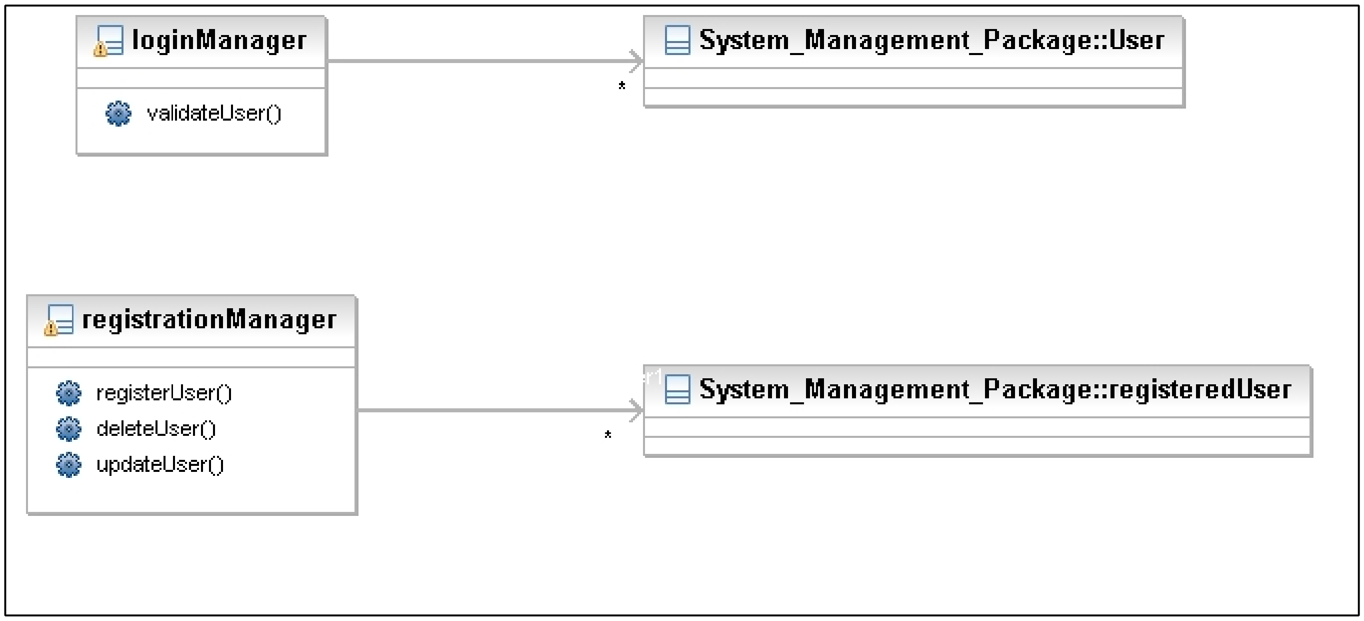


Figure 3.24: User Management Package – Class diagram

3.5.4 CLASS DIAGRAM#4 – USER INTERFACE PACKAGE

Here follows a brief description of the most important classes of the User Interface Package (*see Figure 3.25*):

Class Name	**Description**
registrationForm	It encapsulates the concept of the user.
query	It models the user’s natural language query.
results	It models the retrieved documents in response to the user’s query or in response to the browsing **Category**. This set is a collection of elements (the **resultElement**s) which are either medical **Web_page**s or **Article**s.

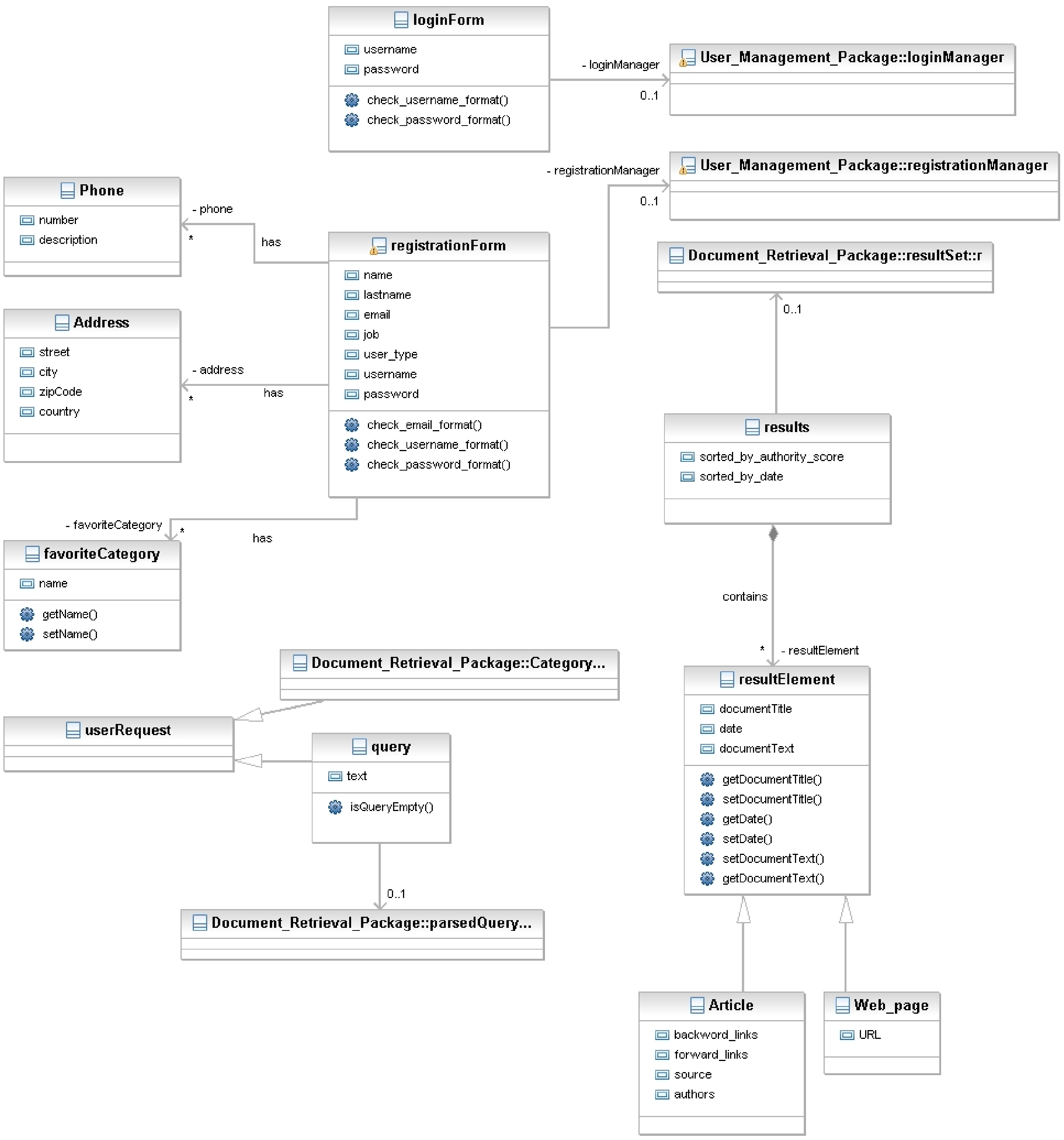


Figure 3.25 : User Interface Package – Class diagram

Chapter 4

IMPLEMENTATION

To demonstrate and objectively asses the quality of the design, a demonstrator medical information system is implemented. This implementation doesn't support the full range of functionalities presented in the requirements analysis phase (section 3):

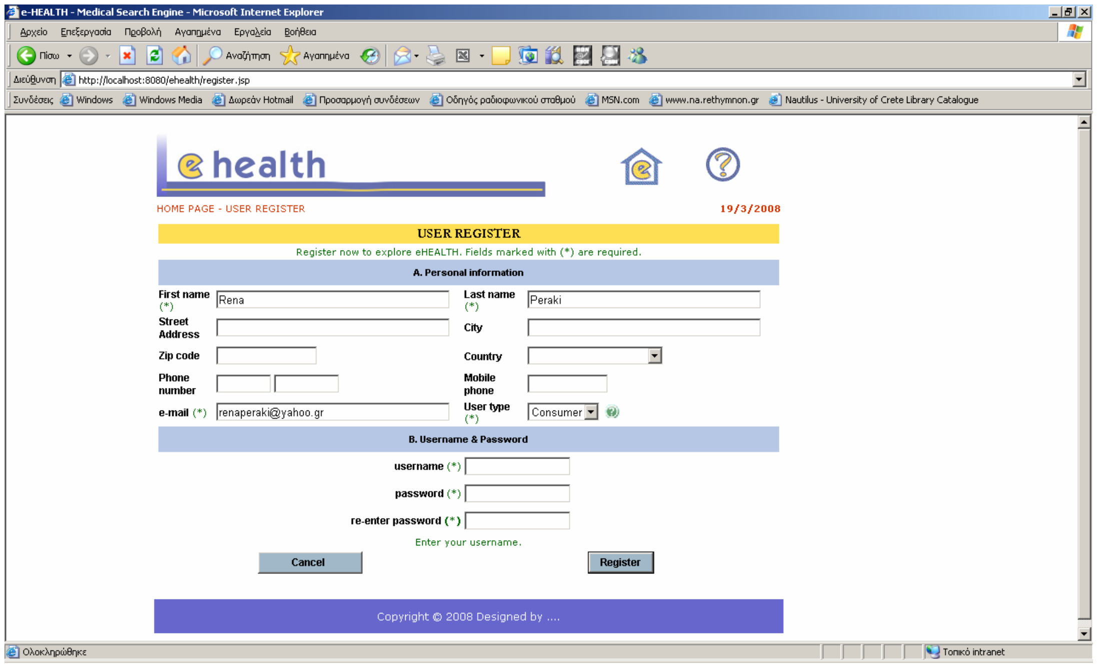
- The example medical document collection (*see Glossary*) consists of around 30,000 references from the **OHSUMED** collection. OHSUMED is a MEDLINE (*see Glossary*) subset, consisting of 348,566 references, covering all references from 270 medical journals over the period 1987-1991. The available fields are title, abstract, MESH indexing terms (in this thesis called MEDLINE terms), author, source and publication type. Most references are accompanied by abstracts, while others are not.
- Lexicographic, AMTEx terms (MeSH terms extracted from the title and abstract) and MeSH terms assigned to documents by domain experts (already available with the documents) are used for indexing.
- Each document if further categorized to one or more Semantic Network categories (*see Glossary*). The additional types of medical categories described in the requirements analysis phase (categories created during the document clustering process or during the citation graph Link Clustering process) are not supported yet.

The following use cases (or part of them) have been implemented:

- (UC-2) *User registration*
- (UC-5) *User login*
- (UC-8) *Extract the document's terms and categorize the document*
- (UC-9) *Create the representation vectors for the document or the user's query*. Only the representation vectors for the lexicographic and AMTEx terms are created.
- (UC-10) *Categorize document to consumer/expert category*
- (UC-11) *Build indexes for the document's terms*. The first version doesn't perform indexing based on user defined MeSH terms and on MMTx terms.
- (UC-13) *Retrieve relevant documents*. This version maps the query to AMTEx terms only. Also in this version the expert's favorite categories are ignored e.g. they are not used to improve ranking in the answer set.
- (UC-14) *Rank the documents*

- *(UC-15) Present results to user*. Most of this use case's functionality is not supported. Extensive information documents are not available (OHSUMED documents have only abstracts). The retrieved answers to user queries contain information such as the document's title, abstract (if it exists), document's categories (from the Semantic Network MeSH hierarchy) and the PubMED identifier.
- *(UC-17) Browse database by medical topics.* This system supports only topics from the Semantic Network MeSH hierarchy.

For the text indexing and retrieval process Apache **Lucene** library is used. Lucene is a free source search engine library written entirely in Java, supported by the Apache Software Foundation. The following figures illustrate systems interface for user registration, query expression and results presentation.


e-HEALTH - Medical Search Engine - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα
Διεύθυνση http://localhost:8080/ehealth/register.jsp
Συνδέσεις Windows Windows Media Δωρεάν Hotmail Προσαρμογή συνδέσεων Οδηγός ραδιοφωνικού σταθμού MSN.com www.na.rethymnon.gr Nautilus - University of Crete Library Catalogue
e health
Ολοκληρώθηκε
Τοπικό intranet

HOME PAGE - USER REGISTER 19/3/2008

USER REGISTER

Register now to explore eHEALTH. Fields marked with (*) are required.

A. Personal information

First name (*)	Rena	Last name (*)	Peraki
Street Address		City	
Zip code		Country	
Phone number		Mobile phone	
e-mail (*)	renaperaki@yahoo.gr	User type (*)	Consumer

B. Username & Password

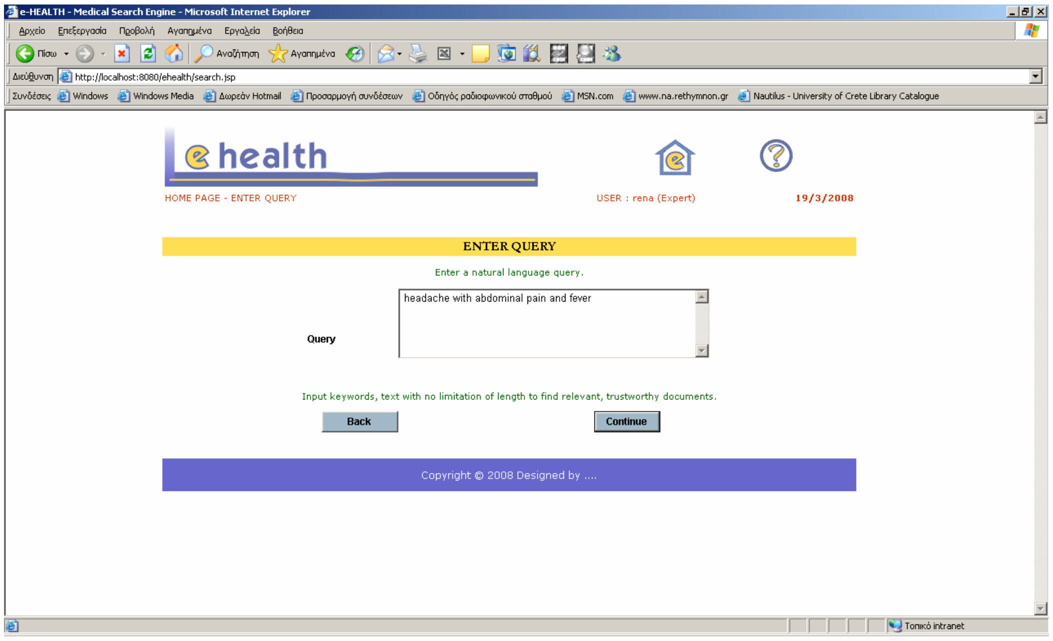
username (*)	
password (*)	
re-enter password (*)	

Enter your username.

Cancel Register

Copyright © 2008 Designed by ....

Figure 4.1: User registers into the system


e-HEALTH - Medical Search Engine - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα
Διεύθυνση http://localhost:8080/ehealth/search.jsp
Συνδέσεις Windows Windows Media Δωρεάν Hotmail Προσαρμογή συνδέσεων Οδηγός ραδιοφωνικού σταθμού MSN.com www.na.rethymnon.gr Nautilus - University of Crete Library Catalogue
Τοπικό intranet

e health

HOME PAGE - ENTER QUERY   USER : rena (Expert)   19/3/2008

ENTER QUERY

Enter a natural language query.

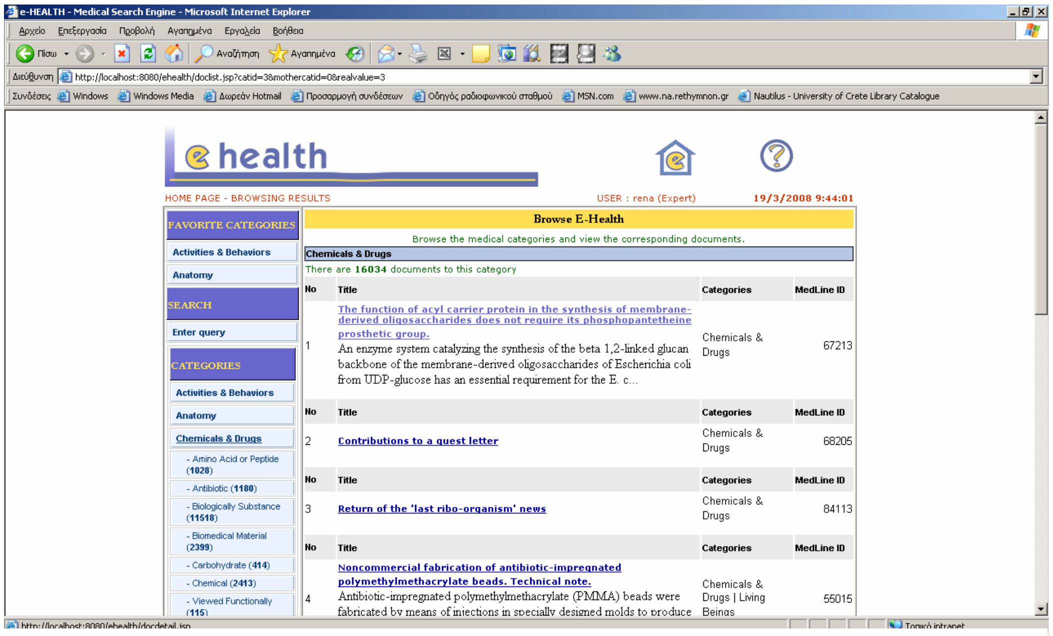
Query: headache with abdominal pain and fever

Input keywords, text with no limitation of length to find relevant, trustworthy documents.

Back   Continue

Copyright © 2008 Designed by ....

Figure 4.2: User submits a free text query


e-HEALTH - Medical Search Engine - Microsoft Internet Explorer
Αρχείο Επεξεργασία Προβολή Αγαπημένα Εργαλεία Βοήθεια
Πίσω Αναζήτηση Αγαπημένα
Διεύθυνση http://localhost:8080/ehealth/doclist.jsp?catid=3&mothercatid=0&realvalue=3
Συνδέσεις Windows Windows Media Δωρεάν Hotmail Προσαρμογή συνδέσεων Οδηγός ραδιοφωνικού σταθμού MSN.com www.na.rethymnon.gr Nautilus - University of Crete Library Catalogue
http://localhost:8080/ehealth/docdetail.jsp
Τοπικό intranet

e health

HOME PAGE - BROWSING RESULTS   USER : rena (Expert)   19/3/2008 9:44:01

FAVORITE CATEGORIES
- Activities & Behaviors
- Anatomy

SEARCH
- Enter query

CATEGORIES
- Activities & Behaviors
- Anatomy
- Chemicals & Drugs
  - Amino Acid or Peptide (**1028**)
  - Antibiotic (**1180**)
  - Biologically Substance (**11518**)
  - Biomedical Material (**2399**)
  - Carbohydrate (**414**)
  - Chemical (**2413**)
  - Viewed Functionally (**115**)

Browse E-Health

Browse the medical categories and view the corresponding documents.

Chemicals & Drugs

There are **16034** documents to this category

No	Title	Categories	MedLine ID
1	The function of acyl carrier protein in the synthesis of membrane-derived oligosaccharides does not require its phosphopantetheine prosthetic group. An enzyme system catalyzing the synthesis of the beta 1,2-linked glucan backbone of the membrane-derived oligosaccharides of Escherichia coli from UDP-glucose has an essential requirement for the E. c...	Chemicals & Drugs	67213
2	Contributions to a guest letter	Chemicals & Drugs	68205
3	Return of the 'last ribo-organism' news	Chemicals & Drugs	84113
4	Noncommercial fabrication of antibiotic-impregnated polymethylmethacrylate beads. Technical note. Antibiotic-impregnated polymethylmethacrylate (PMMA) beads were fabricated by means of injections in specially designed molds to produce	Chemicals & Drugs | Living Beings	55015

Figure 4.3: Results presentation

Glossary

Term	Meaning
AMTEx	AMTEx is a method that automatically extracts MeSH terms from medical text.
authority score	This score is computed during the Citation Graph Link Analysis process. It characterizes the article's popularity and is used for the ranking process.
backward links	These are directed links from a list of articles to one article and imply that the first articles cite to the second one.
citation graph	Is a directed graph where each node represents an article and a directed link from one node to another node implies that the article associated with the first node cites the article associated with the second node.
citation list	This list consists of the forward and backward links of the system's database articles that reference to or are cited by articles that do not exist in the system's database yet.
forward links	These are directed links from one article to a list of other articles and imply that the first article has references to all the other articles.
HON accredited sites	These are sites that are accredited from the Health On the Net Foundation (HON). HON is a non Governmental Organization whose mission is to *'guide Internet users by highlighting reliable, understandable, relevant and trustworthy sources of online health and medical information'*. Each HON accredited site is re-evaluated every year.
medical document collection	It consists of documents that belong either to existing medical databases (such as Medline) or to Web's HON accredited sites.
MEDLINE	It is a large medical collection. Each document contains semi-structured information in XML, has a unique PubMed id, includes citation information such as author(s), journal, year etc., has manually assigned MeSH terms, called Medline terms etc
MeSH	MeSH (*Medical Subject Headings*) is a taxonomic hierarchy of medical and biomedical terms suggested by the U.S. National Library of Medicine.
MMTx	MMTx (*MetaMap Transfer*) is a tool from the U.S. National Library of Medicine, that maps biomedical text into concepts in

Term	Meaning
	the UMLS Metathesaurus.
Semantic Network MeSH categories	It is part of the National Library of Medicine's (NLM) Semantic Network(SN). It consists of : (1) the 135 Semantic Types of the SN which provide a consistent categorization of the MeSH terms only (SN categorizes all terms represented in the UMLS Metathesaurus) and (2) only the 'isa' link from the Semantic Relations that exist between the Semantic Types, establishing a hierarchy of types within the SN.
system's database	It consists of the documents from the medical document collection which are analyzed, indexed and stored.
UMLS Metathesaurus	The UMLS (*Unified Medical Language System*) Metathesaurus includes over 100 incorporated controlled vocabularies (such as SNOMED CT, MeSH, ICD-9-CM etc).

References

1. Rumbaugh J, Jacobson I, Booch G. *The Unified Modelling Language Reference Manual*, Addison Wesley Longman Inc, 1999
2. Fowler Martin, *UML Distilled Third Edition A Brief Guide to the standard Object Modelling Language*, Addison-Wesley, 2003
3. Hliaoutakis A, Zervanou K, Petrakis E, *Medical Document Indexing and Retrieval: AMTEx vs. NLM MMTx* , Proceedings of the 12th International Symposium on Health Information Management Research 2007
4. Gaudinat A, Ruch P, Joubert M, Uziel P, Strauss A, Thonnet M, Baud R, Spahni S, Weber P, Bonal J, Boyer C, Fieschi M, Geissbuhler A. *Health search engine with e-document analysis for reliable search results.* International Journal of Medical Informatics 2006 Jan;75(1):73-85
5. Cockburn Alistair, *Writing Effective Use Cases*, Addison Wesley, 2001
6. Christodoulakis S., Notes from the course 'Ανάπτυξη εφαρμογών πληροφοριακών συστημάτων στο Διαδίκτυο' Chapter 7, 2007
7. Object Management Group - Unified Modeling Language [2008 January]; www.uml.org
8. MeSH: The Medical Subject Headings. [2008 January]; http://www.nlm.nih.gov/mesh
9. Unified Medical Language System (UMLS). [2008 February]; http://www.nlm.nih.gov/research/umls/
10. MetaMap Transfer (MMTx). [2007 March]; http://mmtx.nlm.nih.gov
11. Online Indexing Training Module [2007 June]; http://www.nlm.nih.gov/bsd/indexing
12. Hliaoutakis A, Zervanou K, Petrakis E, Milios E, *Automatic Document Indexing in Large Medical Collections*, ACM International Workshop on Health Information and Knowledge Management (HIKM 2006), November 11, 2006, Arlington, VA, US
13. C. Lee Giles , Kurt D. Bollacker , Steve Lawrence, *CiteSeer: an automatic citation indexing system*, Proceedings of the third ACM conference on Digital libraries, p.89-98, June 23-26, 1998, Pittsburgh, Pennsylvania, United States
14. Huajing Li, Isaac G. Councill, Levent Bolelli, Ding Zhou, Yang Song, Wang-Chien Lee, Anand Sivasubramaniam, C. Lee Giles, 'CiteSeer^{X} – *A Scalable Autonomous Scientific Digital Library*, ACM International Conference Proceeding Series; Vol. 152, 2006, Hong Kong
15. U.S. Department of Health and Human Services, Office of Disease Prevention and Health Promotion,'Expanding the reach and impact of consumer e-health tools', June 2006
16. OHSUMED Test Collection ir.ohsu.edu/ohsumed/ohsumed.html