****

**Asia Pacific College**

School of Computing and information Technology

Medical Question Answering System

Amata, John Patrick

Febrer, Jayvee D.

Espiritu, Immanuel T.

Table of Contents

[I. Introduction 4](#_Toc468958847)

[**Abstract** 4](#_Toc468958848)

[**Background of the Problem** 4](#_Toc468958849)

[**Statement of the Problem** 4](#_Toc468958850)

[**Objectives** 4](#_Toc468958851)

[**Significance** 4](#_Toc468958852)

[**Scope and Limitation** 5](#_Toc468958853)

[**Scope of Initial Release** 5](#_Toc468958854)

[II. Review in Related Literature 5](#_Toc468958855)

[**Stanford CoreNLP** 5](#_Toc468958856)

[**Apache UMIA** 5](#_Toc468958857)

[**Apache Solr** 5](#_Toc468958858)

[**Apache Lucene** 5](#_Toc468958859)

[**Apache OpenNLP** 6](#_Toc468958860)

[**LingPipe** 6](#_Toc468958861)

[**Natural Language Toolkit** 6](#_Toc468958862)

[**Elasticsearch** 6](#_Toc468958863)

[**NlpTools** 6](#_Toc468958864)

[**Question Answering** 6](#_Toc468958865)

[**Automated Question-Answering Techniques and the Medical Domain** 6](#_Toc468958866)

[**New Trends in Automatic Question Answering** 7](#_Toc468958867)

[**Structure of the work** 7](#_Toc468958868)

[**Computational Linguistic** 8](#_Toc468958869)

[**Information Extraction** 8](#_Toc468958870)

[**Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System** 8](#_Toc468958871)

[**Question Processing** 9](#_Toc468958872)

[**Query Analyzer** 9](#_Toc468958873)

[**Question Classification** 10](#_Toc468958874)

[**Query Reformulation** 10](#_Toc468958875)

[**Document Retrieval** 10](#_Toc468958876)

[**Document Processing** 10](#_Toc468958877)

[**Answer Extraction** 11](#_Toc468958878)

[III. Technical Background 11](#_Toc468958879)

[**Natural Language Processing** 12](#_Toc468958880)

[**Question Answering System** 12](#_Toc468958881)

[**Information Retrieval** 12](#_Toc468958882)

[**Artificial Intelligence** 12](#_Toc468958883)

[IV. Design and Methodology 12](#_Toc468958884)

[**Components** 12](#_Toc468958885)

[**Corpus** 13](#_Toc468958886)

[**Query** 13](#_Toc468958887)

[**Evaluation** 13](#_Toc468958888)

[V. Results and Discussions 14](#_Toc468958889)

[**First Cycle of Development: Console Based QA** 14](#_Toc468958890)

[**Second Cycle of Development: Browser Based QA** 15](#_Toc468958891)

[**Third Cycle of Development: Apache UIMA Based QA** 16](#_Toc468958892)

[VI. Conclusion and Recommendations 17](#_Toc468958893)

[Appendices 18](#_Toc468958894)

[**Context Diagram** 18](#_Toc468958895)

[**Data Flow Diagram** 18](#_Toc468958896)

[**Activity Diagram** 19](#_Toc468958897)

[**Class Diagram** 19](#_Toc468958898)

[**Communication Diagram** 20](#_Toc468958899)

[**Component Diagram** 20](#_Toc468958900)

[**Deployment Diagram** 21](#_Toc468958901)

[**Interaction Overview Diagram** 22](#_Toc468958902)

[**Object Diagram** 23](#_Toc468958903)

[**Package Diagram** 24](#_Toc468958904)

[**Sequence Diagram** 24](#_Toc468958905)

[**State Machine Diagram** 25](#_Toc468958906)

[**Use Case Diagram** 25](#_Toc468958907)

[**Detailed Use Case Diagram** 26](#_Toc468958908)

[References 31](#_Toc468958909)

# Introduction

## **Abstract**

The Question Answering (QA) task aims to provide precise and quick answers to user questions from a collection of documents or a database. This is the foundation of our study: we aim to be familiar about studying different appropriate models of Question Answering systems that will not only provide users answers in bulk, but precisely deliver answers that is expected to solve a user's question.

## **Background of the Problem**

The medical profession requires a great deal of memorization and information retrieval of medical texts. Doctors and physicians often need to refer to medical documents and most get theirs from the internet. Due to their profession, time is of great essence. A study showed that it took an average of more than 30 minutes for a healthcare provider to search for answers from PubMed.

The problem lies on the cause of information overloading; a simple query immediately presents hundreds of articles. Question answering systems provide answers for questions in a natural language form. They accomplish this task by analyzing the text and providing a summarized answer from a database of articles and documents. An early feasibility study on Medical Question Answering systems showed that In QA as in IR, one cannot search for every kind of information on the internet, since they are not equally represented. Due to the promising value that Medical Q&A systems bring, our aim is to study the modern trends of information retrieval and question answering models and their application to the Medical field.

## **Statement of the Problem**

What are the impact of modern advancements in QA in making a Medical Question Answering system?

## **Objectives**

Our objective is to experiment and study applicable trends in Question Answering and its application to a Medical Question Answering system.

## **Significance**

Our study will provide a modern model(s) of a question answering system for researchers and developers to follow in creating their own Medical Question Answering system. The promising value that Medical Q&A systems bring in regards to the summarizing texts and answering in quick manner, addresses the problem of information overload that medicine practitioners encounter.

## **Scope and Limitation**

Our study will only examine and experiment with Factoid Question Answering systems. For the context of our corpus, we shall gather at least 50 questions about Diabetes. With these serving as our usage context, we shall apply information retrieval methods for the first release of our system.

## **Scope of Initial Release**

* Question Classification - The first stage, this determines the type of answer that shall be provided.
* Text Selection - This module typically uses several components that apply increasingly complex NLP techniques on a gradually reduced amount of text. The focus of this stage is to identify the documents or paragraphs in the document set that are likely to contain the answer, and a filter preselects small text fragments that contain strings of the same type as the expected answer.
* Answer Construction - This module expands on previous stage -- text selection -- by looking for further clues in the text to determine if the candidate answer can indeed answer the question.

# Review in Related Literature

## **Stanford CoreNLP**

It provides a set of natural language analysis tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, and mark up the structure of sentences in terms of phrases and word dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract open-class relations between mentions, etc.

## **Apache UMIA**

UIMA is a component software architecture for the development, discovery, composition, and deployment of multi-modal analytics for the analysis of unstructured information and its integration with search technologies developed by IBM. The source code for a reference implementation of this framework has been made available on Source Forge, and later the website of the Apache Software Foundation.

## **Apache Solr**

Solr is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites.

## **Apache Lucene**

Apache LuceneTM is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

## **Apache OpenNLP**

The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP also includes maximum entropy and perceptron based machine learning.

## **LingPipe**

LingPipe is tool kit for processing text using computational linguistics. LingPipe is used to do tasks like:

* Find the names of people, organizations or locations in news
* Automatically classify Twitter search results into categories
* Suggest correct spellings of queries

## **Natural Language Toolkit**

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

## **Elasticsearch**

Elasticsearch is a search server based on Lucene. It provides a distributed, multitenant-capable full-text search engine with a RESTful web interface and schema-free JSON documents. Elasticsearch is developed in Java and is released as open source under the terms of the Apache License.

## **NlpTools**

NlpTools is a library for natural language processing written in php.

## **Question Answering**

From a very general perspective QA can be defined as an automatic process capable of understanding questions formulated in a natural language such as English and responding exactly with the requested information.

It is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

## **Automated Question-Answering Techniques and the Medical Domain**

This paper discusses which of three major QA approaches, i.e. deep Natural Language Processing(NLP), Information Retrieval (IR) enhanced by shallow NLP, and Template-based QA, better fit medical applications, eliciting their context of pertinence. To our knowledge, this is the first formal comparison of the three QA approaches that focuses on the medical domain. A common feature of deep NLP systems is that they convert text input into formal representation of meaning such as logic (first order predicate calculus), semantic networks, conceptual dependency diagrams, or frame-based representations (Jurafsky and Martin, 2000, p. 502). In other words, deep NLP systems perform a semantic analysis of text in NL. Semantic analysis is the process of studying the meaning of a linguistic input and giving a formal representation of it. Jurafsky and Martin (2000, p. 548) provide a possible approach for semantic analysis (see figure 1 on the next page): the user input is first passed through a syntactic parser, whose output, represented with a parse tree, is then processed by asemantic analyzer which delivers a meaning representation.

## **New Trends in Automatic Question Answering**

The following paper focuses on Automatic Question Answering (Automatic QA), a sub- field of Information Retrieval. The first chapters focus on historical developments and the definition of the field. Thereafter, an overview of current research topics and key aspects will be given, as well as a classification of the most interesting approaches. The main part of this paper is an analysis of new trends regarding Automatic QA, primarily focusing on approaches in association with web and new media technologies. A discussion of available tools will follow up. Finally, a summary of the things learned during the research on the topic will conclude this work. In the beginning of Automatic Q&A the studies where in a shape of creating an intelligent computer system, which can interact with a human being. It evolved from simple interaction systems without a knowledge database relying on a domain specific field to complex systems, which are web-scaling and able to answer elaborate questions in an interactive and context based way.

## **Structure of the work**

The paper focus on new trends in Automatic QA. The Automatic QA Process can be divided into 4 major parts: **Question Analyzing**, **Preparing the Dataset**, **Text Processing**, and **Data Mapping**. The Information Retrieval part in the Automatic QA project takes care of finding documents, which contain useful information for the question answering. As it was common in the previous years that in Automatic Q&A only document retrieval was used, the information retrieval approach nowadays can also deliver text passages where valuable information can be extracted. The **Natural Language Processing** approach performs a semantic analysis of text. It therefore uses machine learning algorithms to learn rules for text analysis. It uses sets of theories and technologies. In the early years, the most common algorithms were decision trees, the latest approaches base on statistical model and probalistic decisions. (Gunawardena, Lokuhetti, Pathirana, Ragel & Deegalla, 2010. The first kind of Automatic QA system is so called ELIZA. The system relies on natural language processing and formed a question out of a statement by using simple pattern matching. It worked on a MAC time-sharing system at MIT. The program could run different scripts to simulate a human conversation partner.

## **Computational Linguistic**

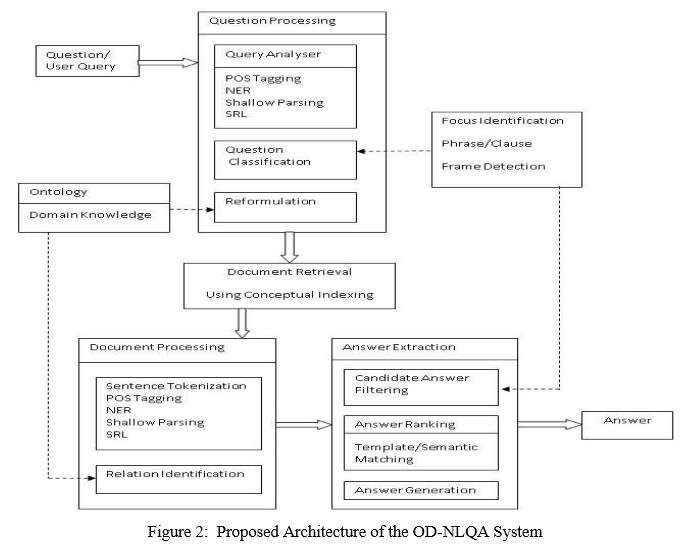
|  |  |
| --- | --- |
| Maximum Entropy Model | The basic idea of the maximum entropy model is, that the probability distribution which best represents the current state of knowledge is the one with the largest entropy. It is used to build up models of many different sources with limited information. |
| Decision Tree Learning | Decision tree learning is a method used in many domains of knowledge discovery, pattern recognition and data mining. Decision trees are hierarchical trees which try to predict an output variable for a given input. Each leave represents an attribute. Each path is a conjunction of the attributes which are on the path. |
| Artificial Neural Networks | Artificial neural networks are mathematical models based on the biological neuron structure of the brain. The neurons, also called nodes, together with the weighted connections are the basic components. The advantages of artificial neural networks are the possibility of enhancing the processing speed by parallelization, adaptation of knowledge, robustness and implementation in low power applications. |

## **Information Extraction**

|  |  |
| --- | --- |
| Entity extraction | It identifies and classifies all phrases in a free text which refer to objects of semantic classes like names, nouns, pronouns etc. In addition all object mentions are linked together which refer to the same entity. |
| Relation extraction | The relations between entities are identified. A relation is always represented by two entities and can be described in many languages. |
| Event extraction | It is also a common application in order to derive specific knowledge from a text. Event extraction identifies events of particular types and the corresponding arguments. A type of an event would be for example ”car crash” or ”natural disaster”. |

## **Architecture of an Ontology-Based Domain-Specific Natural Language Question Answering System**

The proposed architecture of an ontology-based domain-specific NLQA system is depicted in Figure 2. The model integrates key components such as Natural Language Processing techniques; Conceptual Indexing based Retrieval Mechanism, and Ontology Processing.



### **Question Processing**

In the question processing module, with the help of various components, the following actions are performed. · Analysis of the natural language question · Question classification · Reformulation of the user query.

### **Query Analyzer**

The natural-language question given by the user is analyzed using various natural language processing techniques.

Syntactic Analysis – The question is analyzed syntactically using NLP techniques. Part-ofspeech tagging and named entity recognition (NER) are performed. Tools such as Pythonnltk, OpenNLP, Stanford CoreNLP can be used for this purpose. In the proposed system, we used Stanford CoreNLP tool-kit. The CoreNLP processes the document and creates an XML file as output. Shallow parsing is performed to identify the phrasal chunks. The phrasal chunks can be identified using the Regular-expression chunker and the Conll-2000 trained chunker.   
Semantic Analysis - Semantic role labeling is an important step in this module, which enables to find the dependencies or restriction that, can be imposed, after getting the user query. This greatly eliminates the chances of irrelevant set of answers. Semantic roles are identified using the verbnet frames.

### **Question Classification**

|  |  |  |
| --- | --- | --- |
| **Question Word** | **Function** | **Question Focus** |
| Who | Asking what or which person or people(subject) | PERSON |
| Whom | Asking what or which person or people(object) | PERSON |
| Whose | Asking about ownership | PERSON |
| Where | Asking in or at what place or position | LOCATION |

### **Query Reformulation**

The user queries may be reformulated by adding domain knowledge and ontological information.

### **Document Retrieval**

This module selects a set of relevant documents from a domain specific repository. Conceptual indexing is used for the retrieval process since the key word based indexing ignores the semantic content of the document collection. Both the documents and queries can be mapped into concepts and these concepts are used as a conceptual indexing space for identifying and extracting documents.

### **Document Processing**

The retrieved documents are processed for extracting candidate answer set. This module is responsible for selecting the response based on the relevant fragments of the documents.

|  |  |
| --- | --- |
| Syntactic Analysis | The documents analyzed syntactically using the NLP techniques such as part-of-speech tagging and named-entity recognition. |
| Semantic Analysis | Shallow parsing can be performed for finding the semantic phrases or clauses. The semantic roles are identified and mapped to semantic frames. The sentences whose semantic frames map exactly to the semantic frames of the question are also extracted. |
| Relation Identification | The base ontology is populated with the domain knowledge incrementally as we go through different set of documents. By this method a valid knowledge on any specialized discipline can be incorporated to the system. The relations among different concepts are identified using the domain knowledge and the ontological information obtained. |

### **Answer Extraction**

The filtering of candidate answer set and answer generation is performed. The user is supplied with a set of short and specific answers ranked according to their relevance. The different stages are, Filtering, Answer Ranking and Answer Generation.

# Technical Background

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human. Alan Turing." A machine capable of understanding a text like we humans do? I know, it seems hard to believe, and for years it seemed like magic, until I discovered Natural Language Processing (NLP), a field that deals with this kind of problems.Yes, by combining the power of artificial intelligence, computational linguistics and computer science, NLP allows a machine to understand natural language, a task that was so far the exclusive privilege of humans. NLP is everywhere even if we don’t know it. Language transaltor used in different social media sites or websites are just one example of NLP at work.

Another example which we are focusing on is Question Answering. The Question Answering (QA) task aims to provide precise and quick answers to user questions from a collection of documents or a database. This kind of IR system is sorely needed with the dramatic growth of digital information. One domain that is mostly in need of this QA systems is medical domain. Why? simply because every act of knowledge and application in medicine is now worked by computers. One example is an online web application called "webmd" this application serves as a portal for questioning and it gives results and information to the user.

There are many applications that is involving medical fields in Natural language Processing. IBM watson is one of the most famous artificial intelligence that uses NLP and IR (information retreival) for its main functionality of answering questions. In engaging in this field of study, it is recommended to be familiar with the unimaginable number of exsisting application or tools to be use (tenserflow, apache open nlp, ibm watson, etc..)

## **Natural Language Processing**

A field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction.

## **Question Answering System**

Question Answer (Q AND A) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

## **Information Retrieval**

Information retrieval (IR) is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on full-text or other content-based indexing.

## **Artificial Intelligence**

Artificial intelligence (AI) is intelligence exhibited by machines. In computer science, an ideal "intelligent" machine is a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal.

# Design and Methodology

## **Components**

The general components of a typical (mostly factoid based) Question Answering system that we will also include are the following:

1. **Question Processing**

Under the Question Processing lies two main processes: Question Classification and Query Generation.

* **Question Classification** - Question and answer types are categories in taxonomies (Moldovan et al. 1999) and defines a two-level hierarchy of questions. The simplest method to classify questions is the use of a set of rules that map patterns of questions into question types. The patterns are expressed by means of regular expressions on the surface form. The identification of the answer type is usually performed by analyzing the interrogative terms of the question (Handbook of Natural Language Processing). Question Classification is typically done by using a set of patterns, a rule based approach. Another method of question classification is through a statistical based approach using machine learning.
* **Query Generation** - Once the question has been classified, the next process will be to select which of the words or entities will be used to retrieve the correct passage for the answer. This is accomplished by typically using keyword selection or again using a set of patterns.

1. **Document or Passage Retrieval**

Document or Passage Retrieval tasks are generally done through Information Retrieval systems rather than Natural Language Processing techniques as according to Gaizauskas (2004), using IR-Systems significantly reduces the time to search an entire document for the answer. When IR-Systems are used, it is done typically with ranking methods.

When using NLP techniques for Document or Passage Retrieval, passage retrieval is done through indexes. An example would be Litkowski (2000) who indexes with word pairs.

1. **Answer Extraction**

This component will complete the process of passage retrieval by corroborating the correct answer from the passages by either ranking them through the similarity in the text, ranking them based on the frequency of the words, ranking them according to rules already set in place, or ranking them through patterns.

## **Corpus**

For our corpus, we have chosen to pick a minimum of 50 questions in the field of diabetes. These questions will be chosen by gathering the top common results in search engines, websites, and for some variety, books. Currently, the corpus contains questions mainly from a book. An example from these set of questions is “What is diabetic coma?” Respective answer pairings for the questions are also chosen for comparison and data training.

## **Query**

Following a previous study, (Towards a Medical Question-Answering System: a Feasibility Study; Jacquemart, Zweigenbaum) we shall elicit keywords by collecting named entities. In general taxonomies, these are names of people, disease, dates, etc. For our system, these will mainly be diseases, anatomical parts, signs, treatments, etc. For the purpose of reading a larger, more expansive input, we will implement common NLP tasks such as stemming and lemmatization for the input. We will also look into studying “stop word” elimination, for a more optimize querying.

## **Evaluation**

Since our study is to review the relevant advancements of NLP trends and technology, we shall conclude our study by analysing, reviewing, and comparing the results of a simple, traditional factoid based Question Answering system and a more complex, modern Question Answering system that uses neural networks.

# Results and Discussions

We have approached the process of development following the traditional software development life cycle since we ought to develop first a working question answering system. Originally, we have looked for current platforms and frameworks to use for our development since apparently, the development of a question answering system from scratch is quite an overwhelming task. For the platforms that we have analysed, four stood out:

1. OpenEphyra
2. QANUS
3. Quepy
4. Apache UIMA

Nevertheless, we chose to develop a question answering system from scratch as the four platforms and frameworks above proved to have their flaws. We have discovered that OpenEphyra’s documentations are archaic. On QANUS, we failed to deploy the system due to it not supporting the new windows 10, attempts were made to deploy the system on a Linux OS yet the lack of documentation for the technical side had us blindly rummaging for each of its components. Quepy has shown promised but we have soon learned that it restricts the addition of several components that we need. Apache UIMA, the framework that IBM Watson itself has used, has its own merits but the development, research, and deployment of our system using UIMA would take longer than the 4 months left for us to develop a working prototype.

The platform analysis phase took us 2 weeks until we chose to develop our question answering system using Java and several libraries for Natural Language Processing.

## **First Cycle of Development: Console Based QA**



For the first cycle, it took us two months (September-October) to develop a working prototype. The system has successfully parsed and classified 68% of the 200 test questions; the 32% that failed the parsing component were relational questions. The following formula was used to score the accuracy:



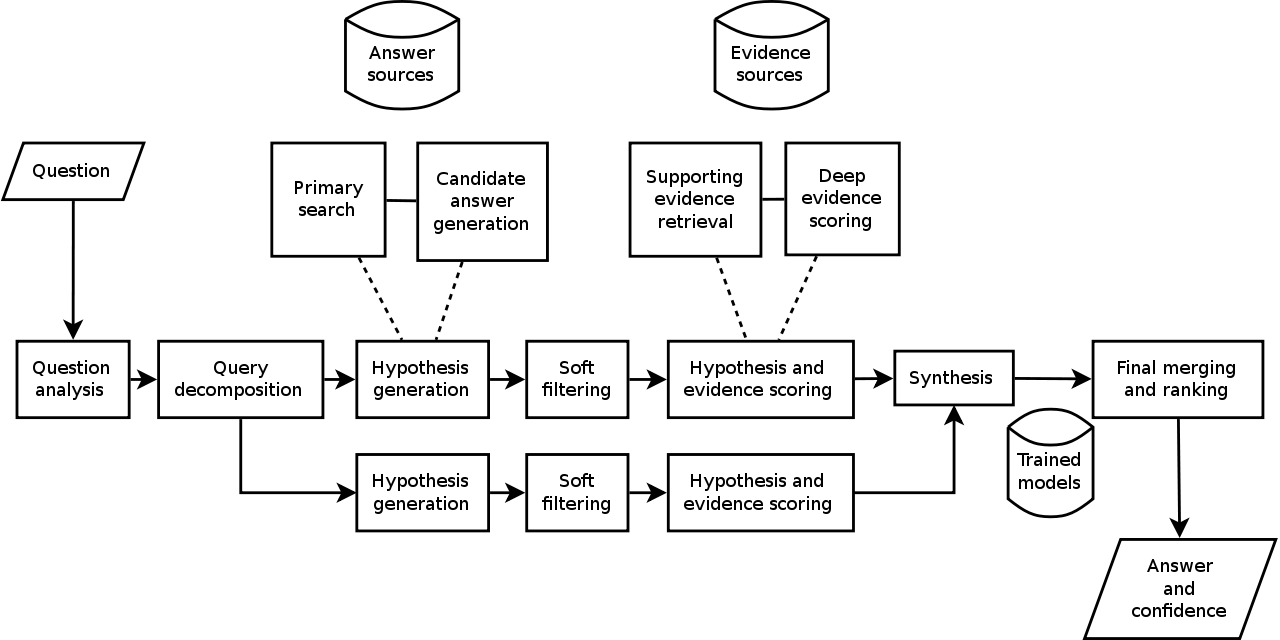
Though the success of its question processing component was adequate, the main pitfall of our developed system was its inadequacy to extract the answers. Text retrieval proved to be a daunting task, especially since we have to develop mathematical formulas and retrain the test questions along with the test answers.

## **Second Cycle of Development: Browser Based QA**



The solution that we have found for our text retrieval problem posed from the previous development cycle was to develop the system on top of a search engine; which we soon learned that most question answering systems in fact, are built with search engines. The development started on November and the prototype was then deployed by November 29, a week before the final defense. For the search engine, we have developed it from scratch using Elasticsearch to handle the database. A PHP page was coded for the user interface, and for the question processing, NlpTools was used. Though the system successfully parses the questions and retrieves a correct answer, we have found that the scalability and modifiable entities using this architecture is too restricting. For example, we cannot extend the system by using machine learning methods or implement dynamic auto summarization of texts by adding further components.

## **Third Cycle of Development: Apache UIMA Based QA**



Since the development of the QA system conforming to the framework set by IBM Watson was too good to ignore, we have chosen to also develop the application along the background from October to present. For the question processing component, we are currently using Apache OpenNLP to train and classify the training questions and models using the Bayes theorem:



*Where A and B are*[*events*](https://en.wikipedia.org/wiki/Event_(probability_theory))*and P(B) ≠ 0.*

*P(A) and P(B) are the*[*probabilities*](https://en.wikipedia.org/wiki/Marginal_probability)*of observing A and B without regard to each other.*

*P(A | B), a*[*conditional probability*](https://en.wikipedia.org/wiki/Conditional_probability)*, is the probability of observing event A given that B is true.*

*P(B | A) is the probability of observing event B given that A is true.*

For the text retrieval component, we will be using Apache Lucene and building on top of that, is a component using Apache Solr. Passage rankings will then be in hard coded Java.

# Conclusion and Recommendations

Though we have succeeded in building a question answering system, the project is still a work in progress since we have yet to evaluate methods of traditional question answering systems using information retrieval and a more modern one, such as that of neural network model.

From a research perspective, we have succeeded in developing grammar parsing models using semantics for the first phase of development. We have also succeeded in creating working architectures for developing question answering systems for a console (which can easily be added a graphical user interface) and for browsers.

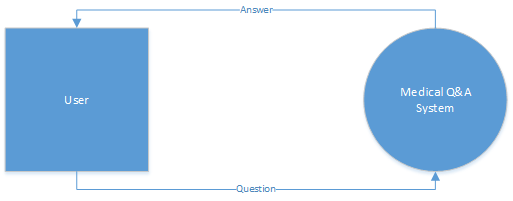
From these accomplishments, for the next term, we can then choose to branch our study to continue developing a question answering system using Apache UIMA; adding components such as automated text summarization of documents and stronger models for grammar parsing.

From the software we have created on our second cycle, we can also choose to pursue a path towards intelligent systems by extending it as an expert system; one that diagnoses medical problems using symptoms written in natural language format.

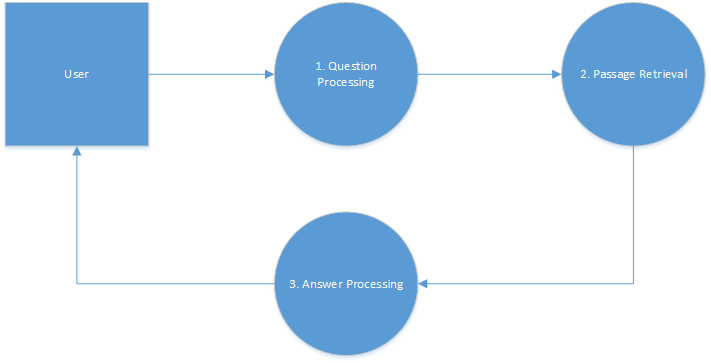
Lastly, we can also choose to pursue a research focused endeavor by removing our current models of information retrieval, as we replace it with memory networks sponsored by the Facebook Artificial Intelligence Research laboratory.

# Appendices

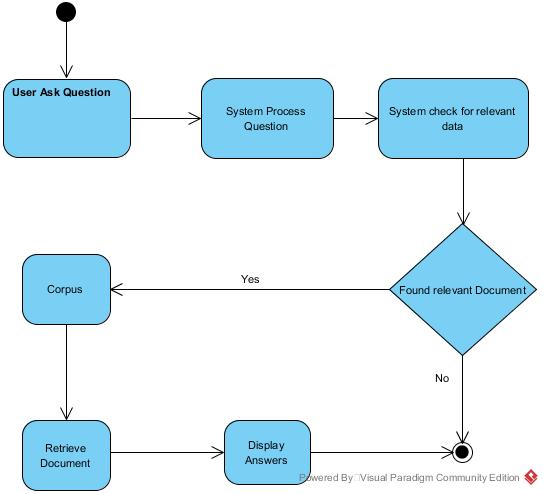
### **Context Diagram**



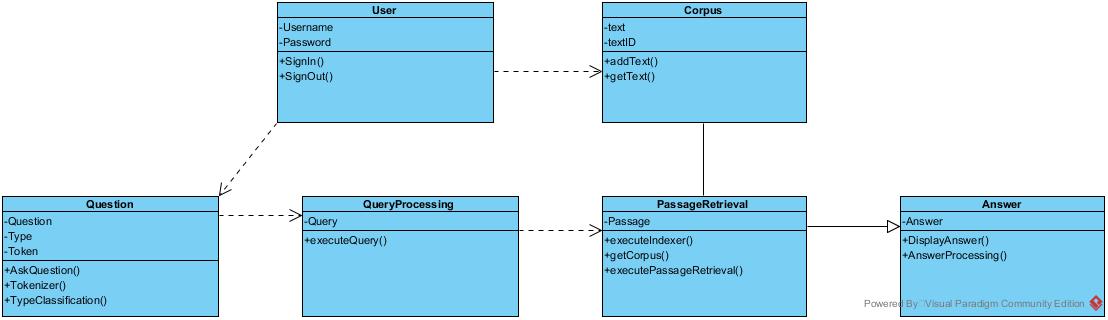
### **Data Flow Diagram**



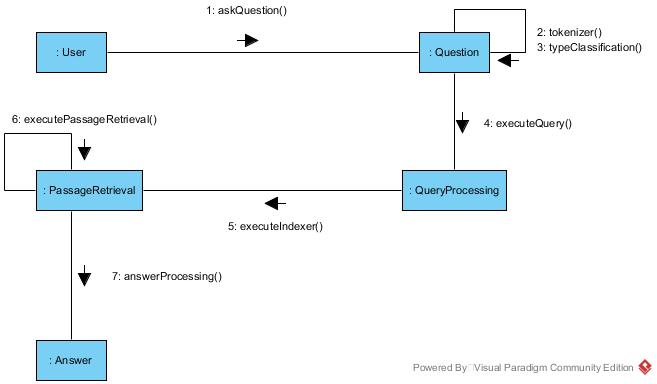
### **Activity Diagram**



### **Class Diagram**



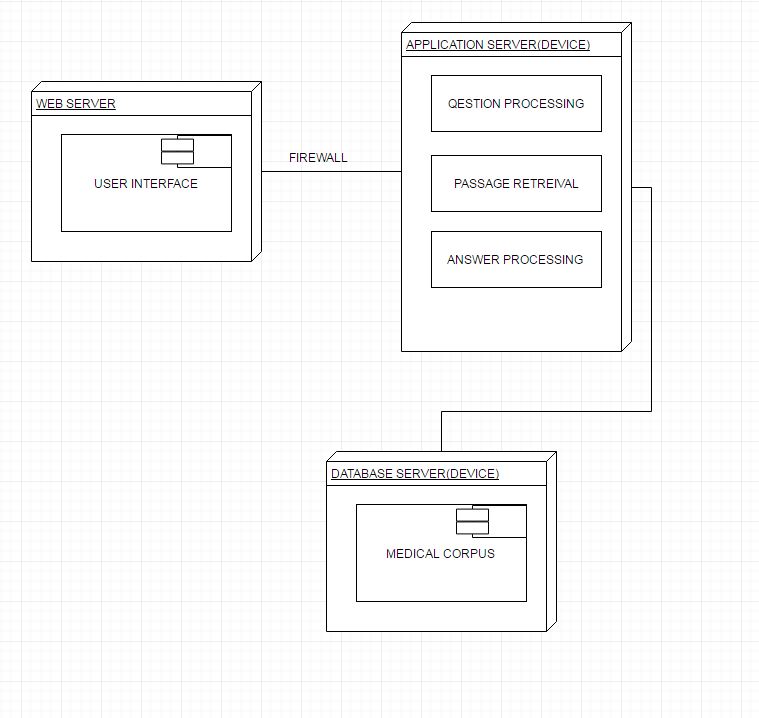
### **Communication Diagram**



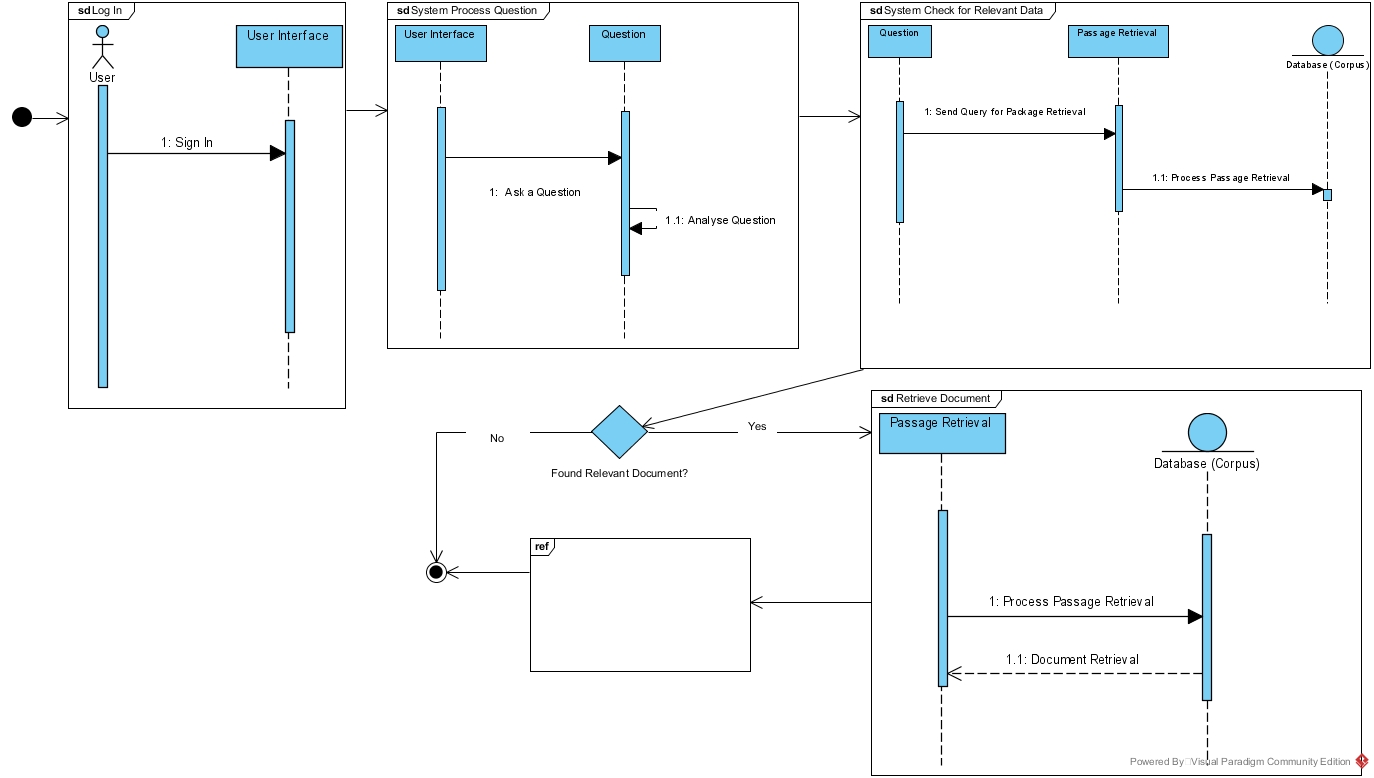
### **Component Diagram**

VZdgbi5.png

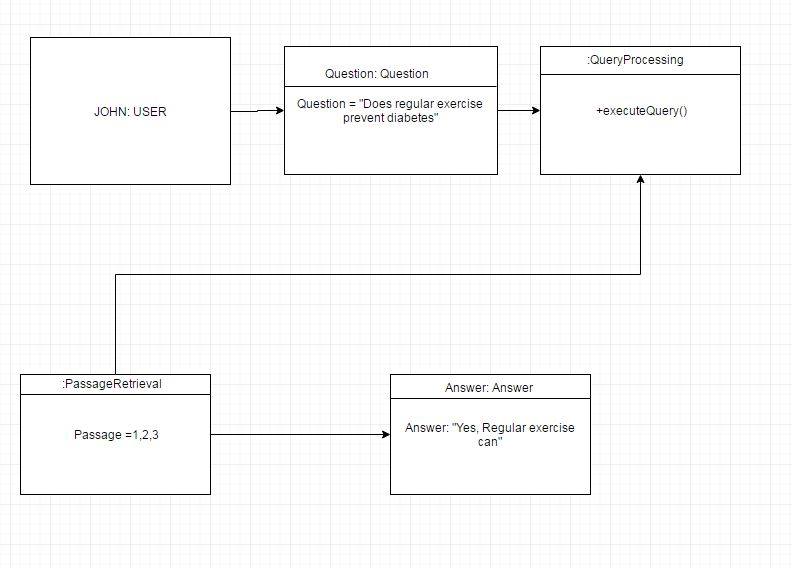
### **Deployment Diagram**



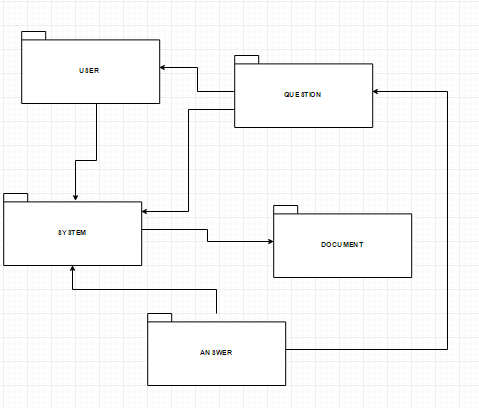
### **Interaction Overview Diagram**



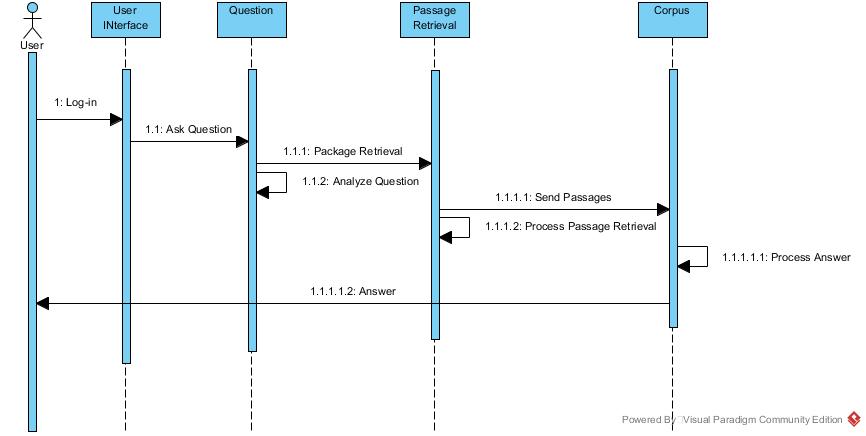
### **Object Diagram**



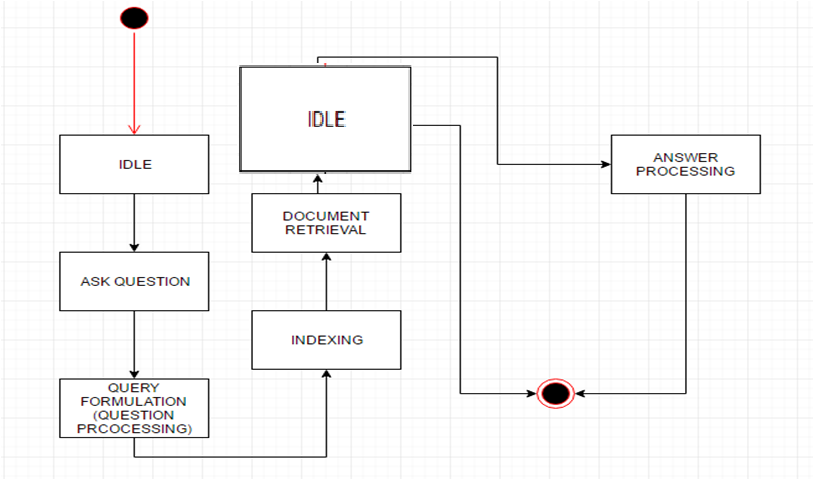
### **Package Diagram**



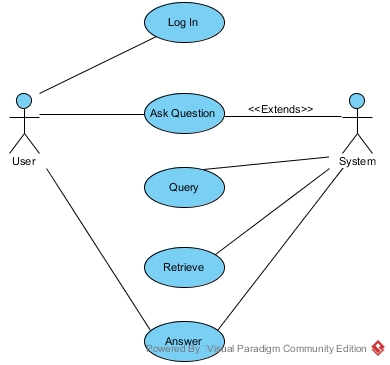
### **Sequence Diagram**



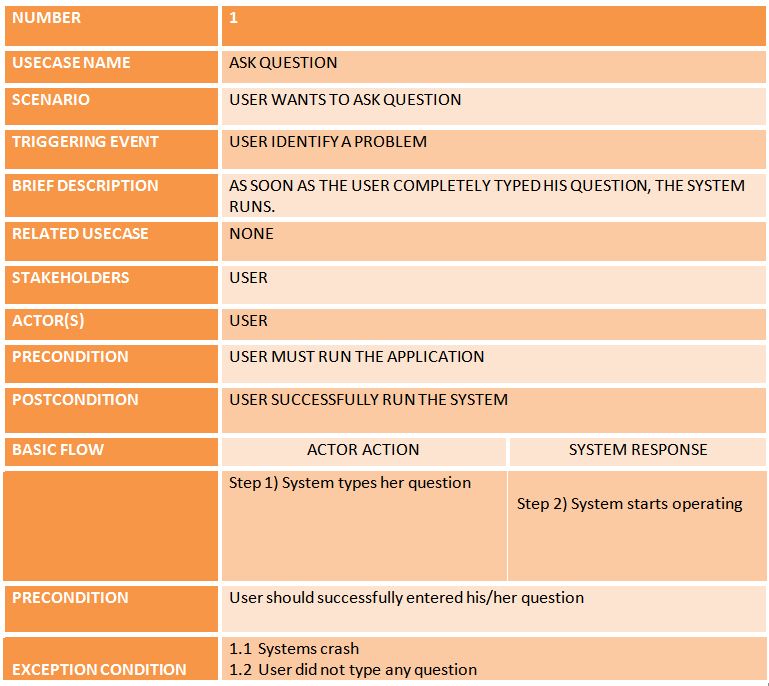
### **State Machine Diagram**

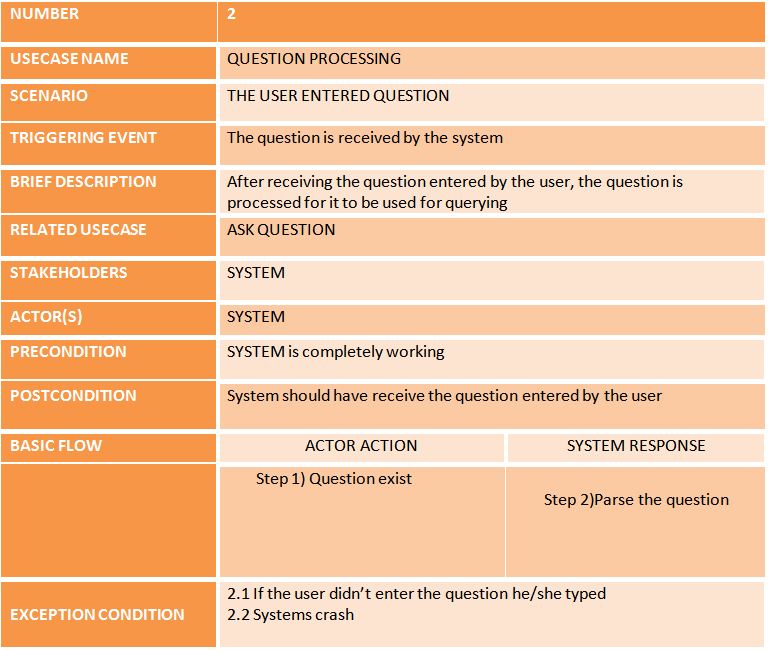


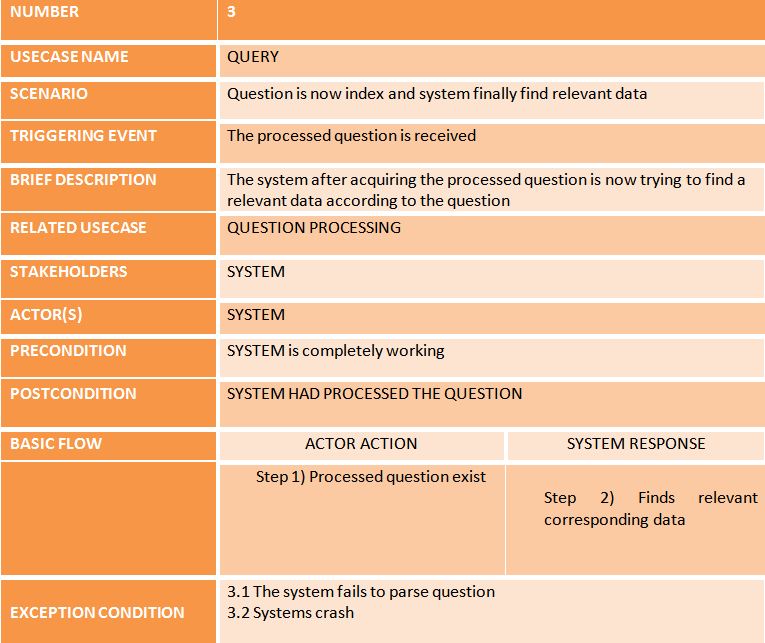
### **Use Case Diagram**

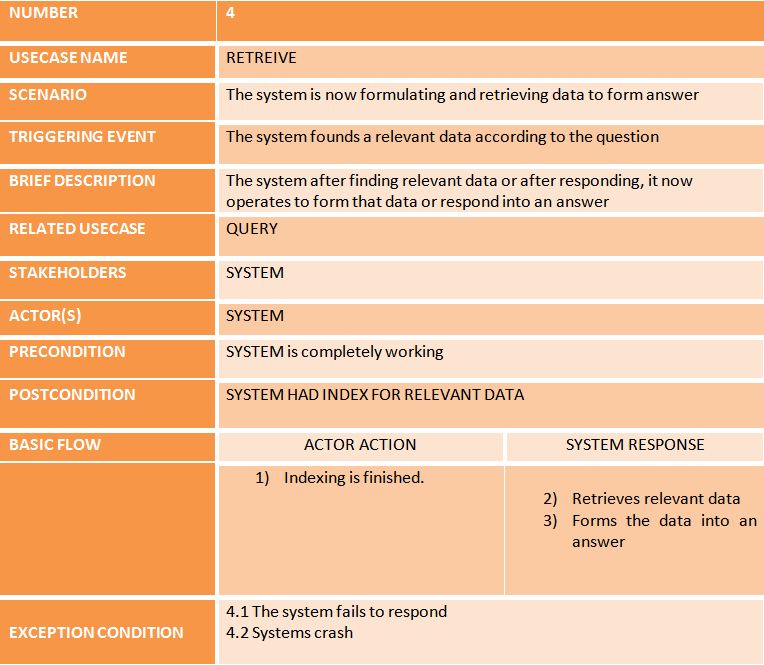


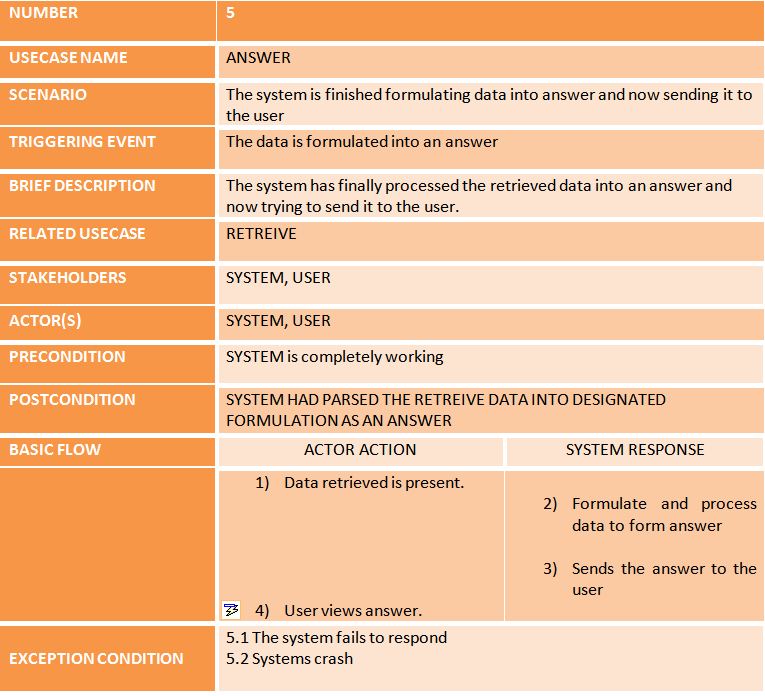
### **Detailed Use Case Diagram**











# References

* Towards a Medical Question-Answering System: a Feasibility Study; Pierre Jacquemart, Pierre Zweigenbaum, from <http://cluster.cis.drexel.edu:8080/sofia/resources/QA.Data/PDF/M_2003_Jacquemart_and_Zweigenbaum_Towards_a_Medical_Quesiton-Answering_System--A_Feasibility_Study-1144453120/M_2003_Jacquemart_and_Zweigenbaum_Towards_a_Medical_Quesiton-Answering_System--A_Feasibility_Study.pdf>
* International Journal of Web & Semantic Technology (IJWesT) Vol.4, No.4, October 2013 - Architecture of an Ontology-Based DomainSpecific Natural Language Question Answering System, Athira P. M., Sreeja M. and P. C. Reghuraj Retrieved July 8, 2016 from <http://airccse.org/journal/ijwest/papers/4413ijwest03.pdf>
* Information Search and Retrieval Graz University of Technology WS 2012/2013 - New Trends in Automatic Question Answering, Christian Gailer, Stefan Kohl, Stephan Oberauer Retrieved July 8, 2016 from <http://www.iicm.tugraz.at/0x811bc82b_0x0011c036>
* Harabagiu S and Moldovan D. Tutorial on open-domain textual question answering. In: Proc 19th COLING, Taipei, Taiwan. 2002.
* Moldovan D, Harabagiu S, and Surdeanu M. Performance issue and error analysis in an open-domain question answering system. In: Proc 38 ACL, Philadelphia, PA. ACL, 2002.
* AUTOMATED QUESTION-ANSWERING TECHNIQUES AND THE MEDICAL DOMAIN BY Andrea Andrenucci [https://people.dsv.su.se/~andrea/QAapproachesHealthInf.pdf](https://people.dsv.su.se/%7Eandrea/QAapproachesHealthInf.pdf)
* The Structure and Performance of anOpen-Domain QUestion Answering System - Moldovan et al from <http://www.aclweb.org/anthology/P00-1071>
* Information Retrieval for Question Answering a SIGIR 2004 Workshop - Robert Gaizauskas, Mark Hepple and Mark Greenwood from [http://www.dcs.shef.ac.uk/~mark/nlp/pubs/gaizauskas\_sigirforum\_2004d.pdf](http://www.dcs.shef.ac.uk/%7Emark/nlp/pubs/gaizauskas_sigirforum_2004d.pdf)
* Unsupervised Sense Disambiguation Using Bilingual Probabilistic Models - Indrajit Bhattacharya, Lise Getoor, Yoshua Bengio from [https://users.soe.ucsc.edu/~getoor/Papers/bhattacharya-acl04.pdf](https://users.soe.ucsc.edu/%7Egetoor/Papers/bhattacharya-acl04.pdf)
* <https://en.wikipedia.org/wiki/UIMA>
* <https://lucene.apache.org/solr/>
* <http://lucene.apache.org/core/>
* <http://stanfordnlp.github.io/CoreNLP/>
* <http://alias-i.com/lingpipe/>
* <https://opennlp.apache.org/>