# 

# AUTOMATED QUESTION TAGGING USING

# NLP MACHINE LEARNING TECHNIQUES

## A PROJECT REPORT

***Submitted by***

**RATHEESH B [REGISTER NO:211417104223]**

**RAJA RAM A [REGISTER NO:211417104211]**

**SANJAY KUMAR V [REGISTER NO:211417104239]**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report **“Automatic Question Tagging Using NLP Machine Learning Techniques”**is the bonafide work of ”**RAJA RAM A [211417104211],SANJAY KUMAR V [211417104239],RATHEESH B[211417104223]”**who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.S.MURUGAVALLI,M.E.,Ph.D. Mr.M.SHANMUGANATHAN M.Tech**

**HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE,

NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ABSTRACT**

In recent years, computerized adaptive testing (CAT) has gained popularity as an important means to evaluate students’ ability. Assigning tags to test questions is crucial in CAT. Manual tagging is widely used for constructing question banks; however, this approach is time-consuming and might lead to consistency issues. Automatic question tagging, an alternative, has not been studied extensively. In this paper, we propose a position-based attention model and keywords-based model to automatically tag questions with knowledge units. With regard to multiple-choice questions, the proposed models employ mechanisms to capture useful information from keywords to enhance tagging performance. Unlike traditional machine learning-based tagging methods, our models utilize deep neural networks to represent questions using contextual information. The experimental results show that our proposed models outperform some traditional classification and topic methods by a large margin on an English question bank dataset.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **No** | **TITLE** | **PAGE** |
|  | LIST OF FIGURES | Vi |
|  | LIST OF ABBREVATIONS | Vii |
| **1** | **INTRODUCTION** |  |
| 1.1 | About the Project |  |
| **2** | **SYSTEM ANALYSIS** |  |
| 2.1 | Existing system |  |
| 2.2 | Proposed system |  |
| **3** | **REQUIREMENTS SPECIFICATION** |  |
| 3.1 | Introduction |  |
| 3.2 | Hardware and Software specification |  |
| 3.3 | Technologies Used |  |

|  |  |  |
| --- | --- | --- |
| **4** | **Project Purpose and Scope** |  |
| 4.1 | Purpose |  |
| 4.2 | Project Scope |  |
| 4.3 | Product Perspective |  |
| 4.4 | System Features |  |
| 4.5 | Design and Implementation Constraints |  |
| 4.6 | Other Nonfunctional Requirements |  |
| **5** | **SYSTEM DESIGN** |  |
| 5.1 | Architecture Diagram |  |
| **6** | **SYSTEM DESIGN – DETAILED** |  |
| 6.1 | Modules |  |
| 6.2 | Module explanation |  |
| **7** | **CODING AND TESTING** |  |
| 7.1 | Coding |  |
| 7.2 | Coding standards |  |
| 7.3 | Test procedure |  |
| 7.4 | Test data and output |  |
|  | **SOURCE CODE** |  |
|  | **SNAP SHOTS** |  |
|  | **REFERENCES** |  |

**LIST OF FIGURES**

5.1 System Design

5.2 Sequence Diagram

5.3 Use Case Diagram

5.4 Activity Diagram

5.5 Collaboration Diagram:

5.6 Data Flow Diagram

5.7 Class Diagram

**LIST OF ABBREVATIONS**

JDK Java Development Toolkit

DEX Dalvik Executables

TCP Transmission Control Protocol

IP Internet Protocol

**HTTP** Hyper Text Transfer Protocol

**ADT** Android Development Tool

**CHAPTER 1**

### INTRODUCTION

**Aim:**

The main aim of this project is to give knowledge regarding Health and diseases of human by receiving user query.

**Synopsis:**

Internet has demonstrated itself as a magnificent design over time. Today, billions of people have contact to internet and they promote by making proper use of that technology. This technology can be used in abundant different ways to aid our comfort and ease likewise Tagging is a machine learning technique which provide tags to the information that user can easily identify the related information user searching for but there is flaw in that method most technology uses manual tagging and semi-manual tagging which is time consuming and people must be expert in that domain has the ability to identify the question and tag them it is not possible in real time and high in cost. We propose an automatic tagging method using NLP which automatically tag question where user can get what information he is searching for which overcome earlier methods.

**CHAPTER 2**

**SYSTEM ANALYSIS**

**2.1 EXISTING SYSTEM**

In existing system user search different information in internet to get the knowledge about the content in this environment there is possibility that user can’t get the desired information or it’s hard to find the required information or can’t get the clear knowledge about it and doesn’t give any alternative solution to the user these methods are done with manual tagging and semi automatic tagging First manually tagging questions with knowledge units needs that the taggers be specialist in that subject a question bank usually contains a huge number of questions, which are updated constantly this makes manual tagging costly in terms of the time taken and related cost and Semi-automatic tagging analyzes content and returns tags that need to be more processed by users, making human help mandatory. There is high chance that questions can be mismatched.

**Problem Definition:**

* + - Manual tagging is time consuming and needs well manual attention.
    - Semi automatic tagging the content has to be again processed.
    - High possibility cant receive requested user query questions
    - High cost and time consuming

### 2.2PROPOSED SYSTEM

We propose an automatic tagging of questions by using NLP (Natural language processing) one of the machine learning techniques. Machine Learning (ML) is applications of artificial intelligence (AI) that provides systems the ability to automatically learn.NLP analyzes understand the human language in a smart and useful way. The previous systems are build using manual and semi-auto tag which is not suitable in making daily updates of data and is not possible to generate requested information and mines the unwanted answers (information) to the user that are confusing and tags are mismatched wrong it makes difficult to observe the information. In this work we done an automatic question tagging that overcome all existing problems. The Medical datasets are collected from the Medinet library. the data’s are stored in the CSV file formats for the later use. CSV file contains the questions and answers from the medical health domain and then the automatic tagging takes place of the question and Answer then the user search the query in AWS(Amazon web service) and the query is forwarded to the Medinet library to extract the user queried related answer .Medinet library contains 700 medical domain files which is stored in tc2011 api if the requested file not found in Medinet it redirect to PDF box it will provide pdf files(contains information about domain or diseases) related to the question The pdf files are extracted by using lucene indexing which provide fast retrieval of files If the answer is not related to query or not clear. the question will forwarded to the Expert(doctor) who can clear the doubt

and replay for the query.Finally providing the quiz for the student who want know their knowledge in the medical field and providing them domain score and overall score and feedback for the student.

**Advantages**

* + - Automatic tagging does not need help from human
    - Provide standard and consistent results
    - Low cost
    - Easily retrieves the answer tagging makes easier to understand the information
    - Provides Knowledge for the students

**CHAPTER 3**

**REQUIREMENT SPECIFICATIONS**

**3.1 INTRODUCTION**

WITH the proliferation of the Internet, computerized testing has become a popular method of assessing students with the aim of measuring their ability, adjusting their learning approach, and recommending materials to help them improve. Adaptively adjusting the approach and providing recommendations according to students’ individual level are crucial to improving the learning efficiency, [1], [2], [3]; this paradigm is called computerized adaptive testing (CAT) [4]. This approach requires a well-structured question bank, which is a collection of question items stored in a database. However, many question banks lack adequate question management, leading to difficulties in finding the knowledge that students are weak in. A knowledge map is an effective knowledge management tool [5] using which we can establish relationships among knowledge units. Tagging is a simple and efficient method to organize resources [6]. As knowledge units are used as tags, we can utilize tagging technology to associate questions with them. Thus, questions will be well-organized for realizing various CAT functions. Specifically, we can generate personal knowledge maps (user profile) [7] to identify students’ weaknesses and recommend suitable learning materials. There are three types of tagging methods: (1) manual tagging, (2) semiautomat

ic tagging, and (3) automatic tagging. Manual tagging is the most commonly used method for organizing questions in the industry. However, manual tagging suffers from some limitations. First, manually tagging questions with knowledge units requires that the taggers be experts in that subject. Second, a question bank usually contains a large number of questions, which are updated constantly; this makes manual tagging expensive in terms of the time taken and associated cost. Moreover, different taggers may analyze questions from different perspectives, which results in subjective differences, and makes it difficult to ensure consistency and accuracy. Semi-automatic tagging analyzes content and returns tags that need to be further processed by users, making human help mandatory. Automatic tagging processes content without human intervention, resulting in more standard and consistent results at lower costs. Therefore, in this paper, we focus on automatic question tagging. To the best of our knowledge, there are few studies targeting this topic. This paper presents a position-based attention model and keywords-based model to identify tags for questions. Both these models are based on the idea that answers aid in deducing the knowledge units more effectively, as in the case of multiple-choice questions. Traditional tagging tasks focus on ”one instance with one label.” However, a question may examine more than one knowledge unit at the same time, and the number might not be known beforehand. Therefore, we view automatic question tagging as a multi-label classification [8] task in this paper. Many traditional machine learning tagging methods use models derived from bag-of-words (BOW), such as term frequencyinverse document frequency (TF-IDF), to parse texts into calculable vector spaces [9]. These models ignore contextual information about words. In contrast, the proposed models are based on recurrent neural networks (RNNs) that use contextual information to boost tagging performance. Moreover, unlike some learning object (LO) tagging methods that require that synonyms of knowledge units or concepts appear in the content, the proposed methods are not subject to this restriction. Experimental results demonstrate the effectiveness of our tagging methods.

**3.2 HARDWARE AND SOFTWARE SPECIFICATION**

### 3.2.1 HARDWARE REQUIREMENTS

|  |  |  |
| --- | --- | --- |
|  Hard Disk | : | 80GB and Above |
|  RAM | : | 4GB and Above |
|  Processor | : | P IV and Above |

### 3.2.2 SOFTWARE REQUIREMENTS

* Windows 7 or above
* JDK 1.7 and JDK 1.6
* Tomcat 6 and Tomcat 7
* Struts, Servlets

#### 3.3 TECHNOLOGIES USED

* MACHINE LEARNING
* JAVA

#### 3.3.1.2 WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main,** which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out,** which is java’s output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user’s program.

Java consists of two things :

* Programming language
* Platform

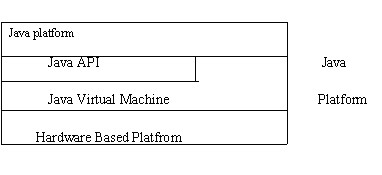
#### 3.3.1.3 THE JAVA PROGRAMMING LANGUAGE

Java is a high-level programming language that is all of the following:

* Simple
* Object-oriented
* Distributed
* Interpreted
* Robust
* Secure
* Architecture-neutral
* Portable
* High-performance
* Multithreaded
* Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

dependencies.

Fig.3.3(the java platform)

As a platform-independent environment, Java can be a bit slower than native code. However, smart compliers, weel-tuned interpreters, and just-in-time byte compilers can bring

Java’s performance close to that of native code without threatening portability.

### 3.3.2 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XMLformatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

**Environment**

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in hightraffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

**Product features**

Tomcat 3.x (initial release)

* implements the Servlet 2.2 and JSP 1.1 specifications
* servlet reloading
* basic HTTP functionality Tomcat 4.x
* implements the Servlet 2.3 and JSP 1.2 specifications
* servlet container redesigned as Catalina
* JSP engine redesigned as Jasper
* Coyote connector
* Java Management Extensions (JMX), JSP and Struts-based administration
* Tomcat 5.x
* implements the Servlet 2.4 and JSP 2.0 specifications
* reduced garbage collection, improved performance and scalability
* native Windows and Unix wrappers for platform integration
* faster JSP paring

**CHAPTER 4**

**4.1 Project Purpose and Scope**

**4.1 Purpose**

The main aim of this project is to give knowledge regarding Health and diseases of human by receiving user query.

~~.~~

**4.2 Project Scope**

We propose an automatic tagging of questions by using NLP (Natural language processing) one of the machine learning techniques. Machine Learning (ML) is applications of artificial intelligence (AI) that provides systems the ability to automatically learn.NLP analyzes understand the human language in a smart and useful way. The previous systems are build using manual and semi-auto tag which is not suitable in making daily updates of data and is not possible to generate requested information and mines the unwanted answers (information) to the user that are confusing and tags are mismatched wrong it makes difficult to observe the information. In this work we done an automatic question tagging that overcome all existing problems. The Medical datasets are collected from the Medinet library. the data’s are stored in the CSV file formats for the later use. CSV file contains the questions and answers from the medical health domain and then the automatic tagging takes place of the question and Answer then the user search the query in AWS(Amazon web service) and the query is forwarded to the Medinet library to extract the user queried related answer .Medinet library contains 700 medical domain files which is stored in tc2011 api if the requested file not found in Medinet it redirect to PDF box it will provide pdf files(contains information about domain or diseases) related to the question The pdf files are extracted by using lucene indexing which provide fast retrieval of files If the answer is not related to query or not clear. the question will forwarded to the Expert(doctor) who can clear the doubts and replay for the query.Finally providing the quiz for the student who want know their knowledge in the medical field and providing them domain score and overall score and feedback for the student.

**4.3 Product Perspective**

Internet has demonstrated itself as a magnificent design over time. Today, billions of people have contact to internet and they promote by making proper use of that technology. This technology can be used in abundant different ways to aid our comfort and ease likewise Tagging is a machine learning technique which provide tags to the information that user can easily identify the related information user searching for but there is flaw in that method most technology uses manual tagging and semi-manual tagging which is time consuming and people must be expert in that domain has the ability to identify the question and tag them it is not possible in real time and high in cost. We propose an automatic tagging method using NLP which automatically tag question where user can get what information he is searching for which overcome earlier methods.

**4.4 SystemFeatures**

We propose models for automatic question tagging, an important approach to question management that has received little attention. The mechanism proposed to capture important information from questions can be applied to other tasks such as question text and short text mining.

**4.5 Design and Implementation Constraints**

**4.5.1 Constraints in Analysis**

* Constraints as Informal Text
* Constraints as Operational Restrictions
* Constraints Integrated in Existing Model Concepts
* Constraints as a Separate Concept
* Constraints Implied by the Model Structure

**4.5.2 Constraints in Design**

* Determination of the Involved Classes
* Determination of the Involved Objects
* Determination of the Involved Actions
* Determination of the Require Clauses
* Global actions and Constraint Realization

**4.5.3 Constraints in Implementation**

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

**4.6 Other Nonfunctional Requirements**

**4.6.1 Performance Requirements**

The application at this side controls and communicates with the following three main general components.

* embedded browser in charge of the navigation and accessing to the web service;
* Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

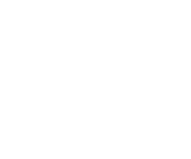
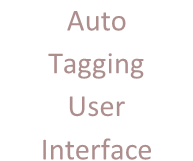
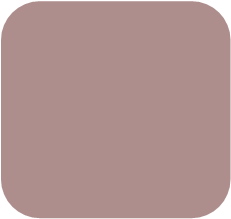
Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

**4.6.2 Safety Requirements**

* 1. The software may be safety-critical. If so, there are issues associated with its integrity level
  2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
  3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
  4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
  5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
  6. Systems with different requirements for safety levels must be separated.
  7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

## CHAPTER 5

**5.1 Architecture Diagram:**

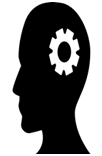
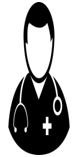


Auto

Tagging

User

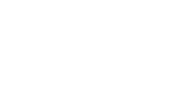
Interface



Clean



Analyze CSV



Analyze

Resource



NLP



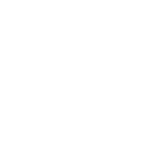
Admin



Student



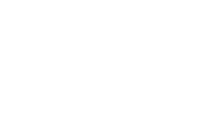
User



User

Search

Category



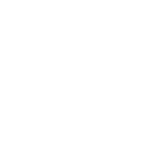
If Answer

not found in

PDF



Expert

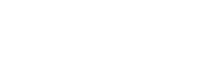


If

Answer

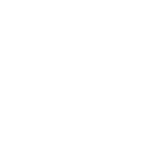
not

There



Ask question

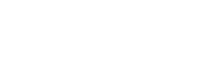
to Expert



Expert

forward

Answer

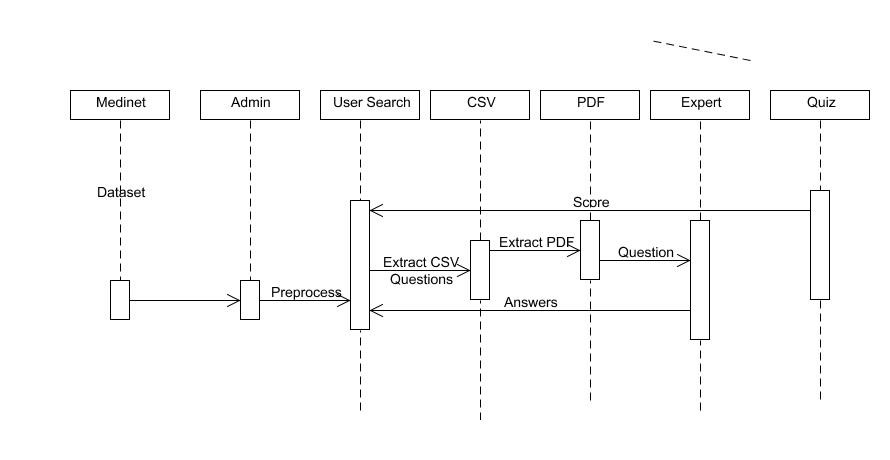


Feed Back

## Fig: 5.1

**5.2 Sequence Diagram:**

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



**5.3 Use Case Diagram:**

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

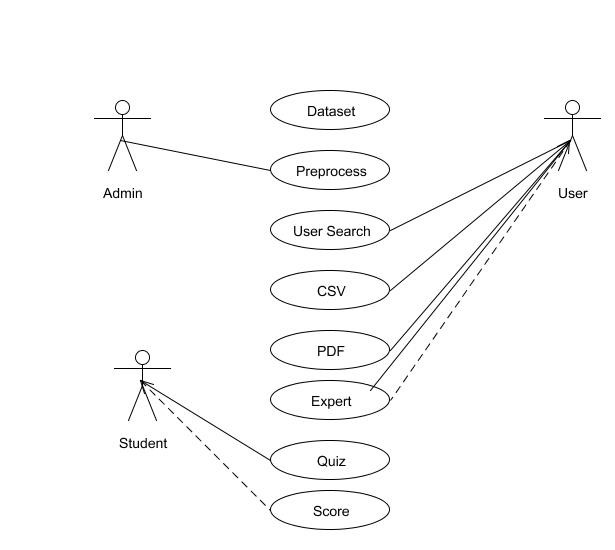
### 5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

**Use case:** A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:** An actor is a person, organization or external system that plays a role in one or more interaction with the system.

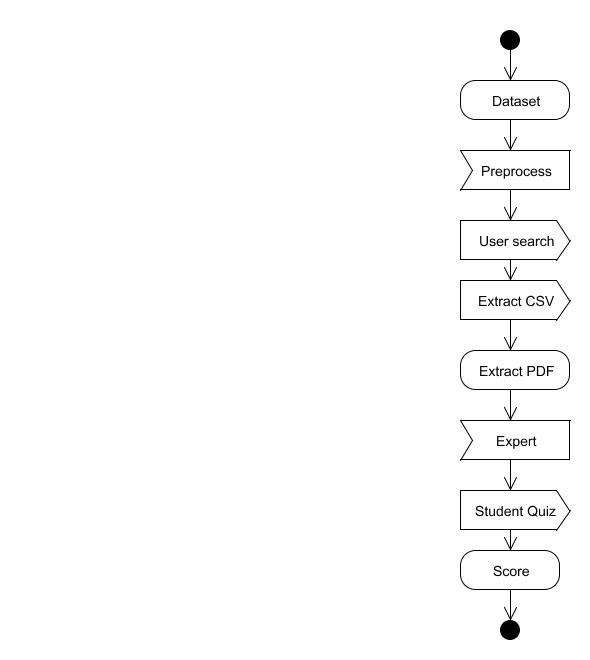


**5.4 Activity Diagram:**

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

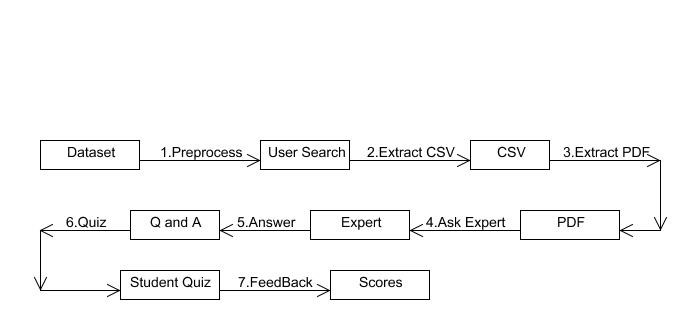
* Rounded rectangles represent activities.
* Diamonds represent decisions.
* Bars represent the start or end of concurrent activities.
* A black circle represents the start of the workflow.
* An encircled circle represents the end of the workflow.



* 1. **Collaboration Diagram:**

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist.

The collaboration diagram illustrates messages being sent between classes and objects.



* 1. **DATA FLOW DIAGRAM:**

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

**Level 0:**



Datasets



Preprocess



Analyze CSV



Analyze Resource



NLP

**Level 1:**



User Register



Search Question



Fetch Answer from CSV

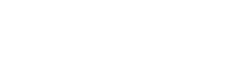


Q and A

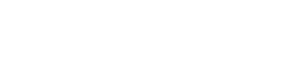


PDF Files

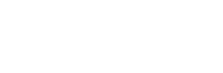
**Level 2:**



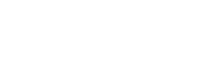
User Question



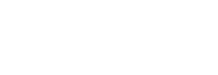
Forward To Expert



Expert

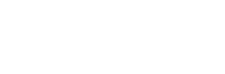


Expert Answer

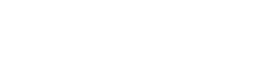


User

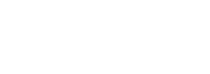
**Level 3:**



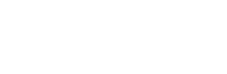
Student Quiz



Domain select



Q and A



Scores

**5.7 Class Diagram**

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

**CHAPTER 6**

#### SYSTEM DESIGN

First module Admin has to register first and then provide admin details (Name and password) valid admin enters the page. Admin is responsible for the whole operation admin work includes cleaning, adding the CSV files, analyzing resources, NLP (Natural language processing) and cleaning NLP. The user has to register their details first after that user login using (Name and Password) if user is valid enters the page then user chooses the categories of domain present in the page user clicks the categories it show list available domains then user select any one domain based on that related information or question and answers to that domain will be displayed. If the requested domain information or question answer is not available it provides PDFs related to that domain the provided answers of the question are not satisfied when user feels. User can ask question to the Expert (doctor) who can clearly answer the user query. Expert (doctor) has to register first like (Name, Expert id, category of doctor) after that login using valid credentials. Here expert id is unique id for the expert and category is to identify the expert domain. Expert enters the page and he gets the notification from the user who asked the question after that Expert analyze the question and provides the related answer to the question the answer is forwarded to the user profile. User can view the answer any time, student registers first after the login page is forwarded to the quiz page the student is asked to write quiz in medical domain. Questions from medical domain is provided student has to answer it finally student gets the overall score of the test, domain score, and gets the feedback for the test which the student can improve knowledge in that domain.

6.1 MODULES

1.Admin Preprocessing

1. Auto tagging Questions
2. Expert’s Answering Process
3. Student Assessment

* 1. **MODULE EXPLANATION:**

**Admin Preprocessing:**

First module Admin has to register first and then provide admin details (Name and password) valid admin enters the page. Admin is responsible for the whole operation admin work includes cleaning, adding the CSV files, analyzing resources, NLP (Natural language processing) and cleaning NLP.

**Auto tagging Questions:**

In this module the user has to register their details first after that user login using (Name and Password) if user is valid enters the page then user chooses the categories of domain present in the page user clicks the categories it show list available domains then user select any one domain based on that related information or question and answers to that domain will be displayed. If the requested domain information or question answer is not available it provides PDFs related to that domain the provided answers of the question are not satisfied when user feels. User can ask question to the Expert (doctor) who can clearly answer the user query.

**Expert’s Answering Process:**

In this module Expert (doctor) has to register first like (Name, Expert id, category of doctor) after that login using valid credentials. Here expert id is unique id for the expert and category is to identify the expert domain. Expert enters the page and he gets the notification from the user who asked the question after that Expert analyze the question and provides the related answer to the question the answer is forwarded to the user profile. User can view the answer any time,

**Student Assessment:**

In this module student registers first after the login page is forwarded to the quiz page the student is asked to write quiz in medical domain. Questions from medical domain is provided student has to answer it finally student gets the overall score of the test, domain score, and gets the feedback for the test which the student can improve knowledge in that domain.

**Algorithm:**

* + - Natural Language Processing
    - Clustering and Classification

#### CODING AND TESTING

##### 7.1 CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

##### 7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

#### 7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

##### Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

##### Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each

subsequent letters of the new words in uppercase and the rest of letters in lowercase.

#### 7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

* Proper default values for the variables.
* Proper validation of values in the field.
* Proper documentation of flag values.

#### 7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

#### 7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

* X – User has performed illegal operation.
* ! – Information to the user.

**7.3 TEST PROCEDURE**

#### SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

* Static analysis is used to investigate the structural properties of the Source code.
* Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

**7.4 TEST DATA AND OUTPUT**

#### 7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

#### 7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

* Performance Test
* Stress Test
* Structure Test

#### 7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

#### 7.4.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

#### 7.4.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

* Exercise all logical decisions on their true or false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures to assure their validity.
* Checking attributes for their correctness.
* Handling end of file condition, I/O errors, buffer problems and textual errors in output information

#### 7.4.6 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications.

The product development can be staged, and modules integrated in as they complete unit testing.

Testing is completed when the last module is integrated and tested.

**7.5 TESTING TECHNIQUES / TESTING STRATERGIES**

#### 7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the

documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

##### 7.5.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

* Flow graph notation
* Cyclometric complexity
* Deriving test cases
* Graph matrices Control

#### .5.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

* Graph based testing methods
* Equivalence partitioning
* Boundary value analysis
* Comparison testing

**7.5.2 SOFTWARE TESTING STRATEGIES:**

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

* Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
* Different testing techniques are appropriate at different points in time.
* The developer of the software and an independent test group conducts testing.
* Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

**.5.2.1 INTEGRATION TESTING:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

**7.5.2.2 PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

**.5.2.3 SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

##### 7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions

exists.

* The function or performance characteristics confirm to specifications and are accepted.
* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

#### .5.2.5 USER ACCEPTANCE TESTING

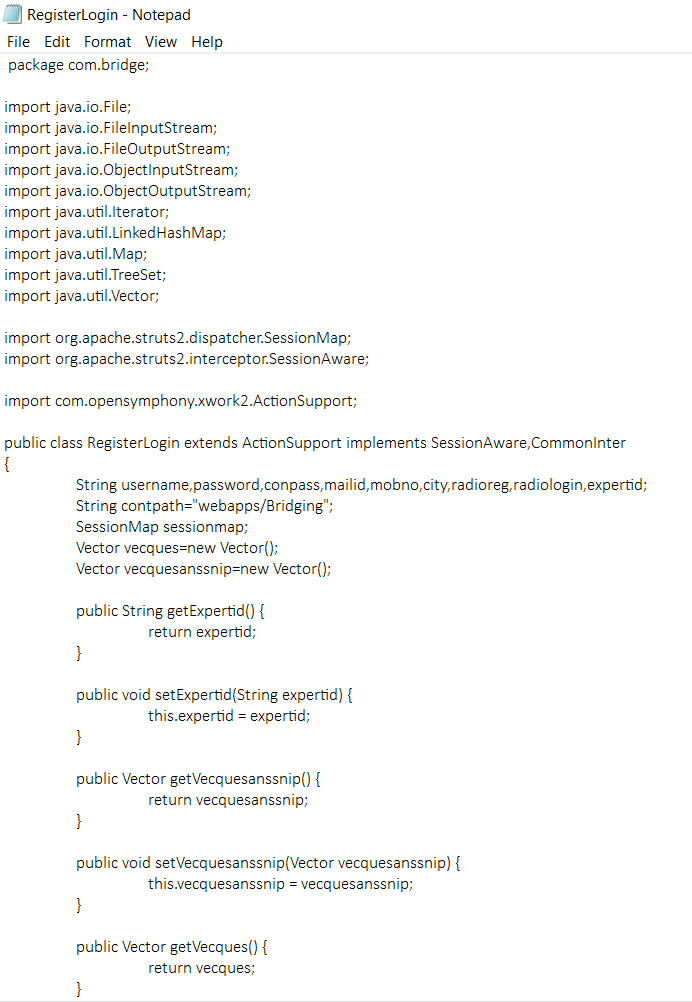
User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

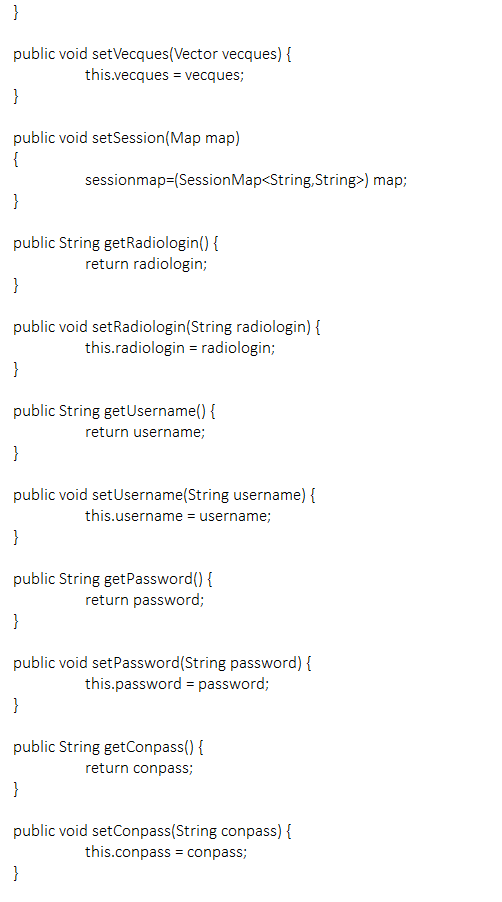
* Input screen design.
* Output screen design.

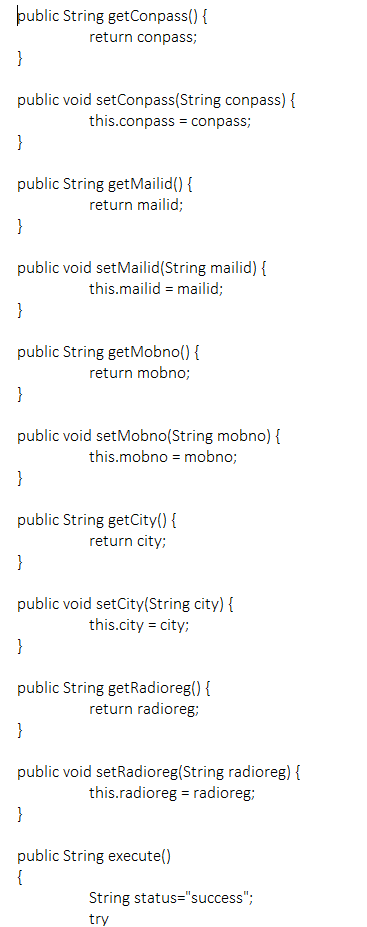
**Conclusion:**

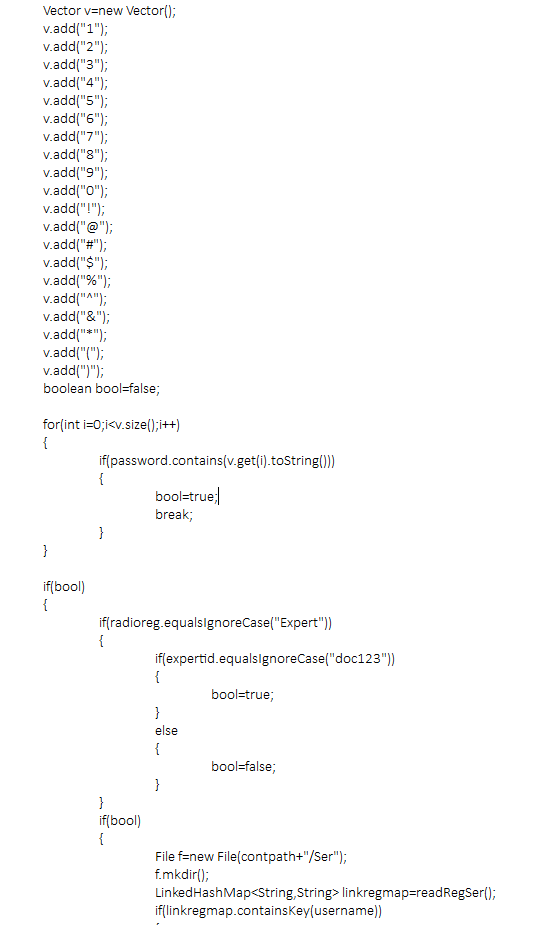
Thus the main aim is to give knowledge regarding Health and diseases of human by receiving user query was implemented in this project.

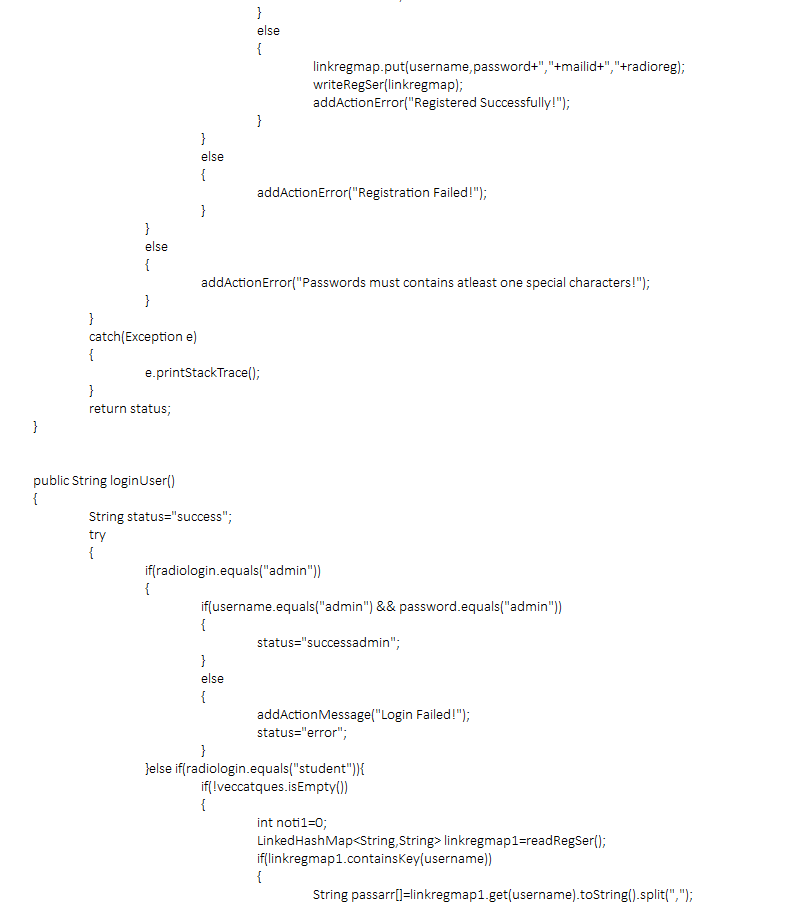
**Source Code :**



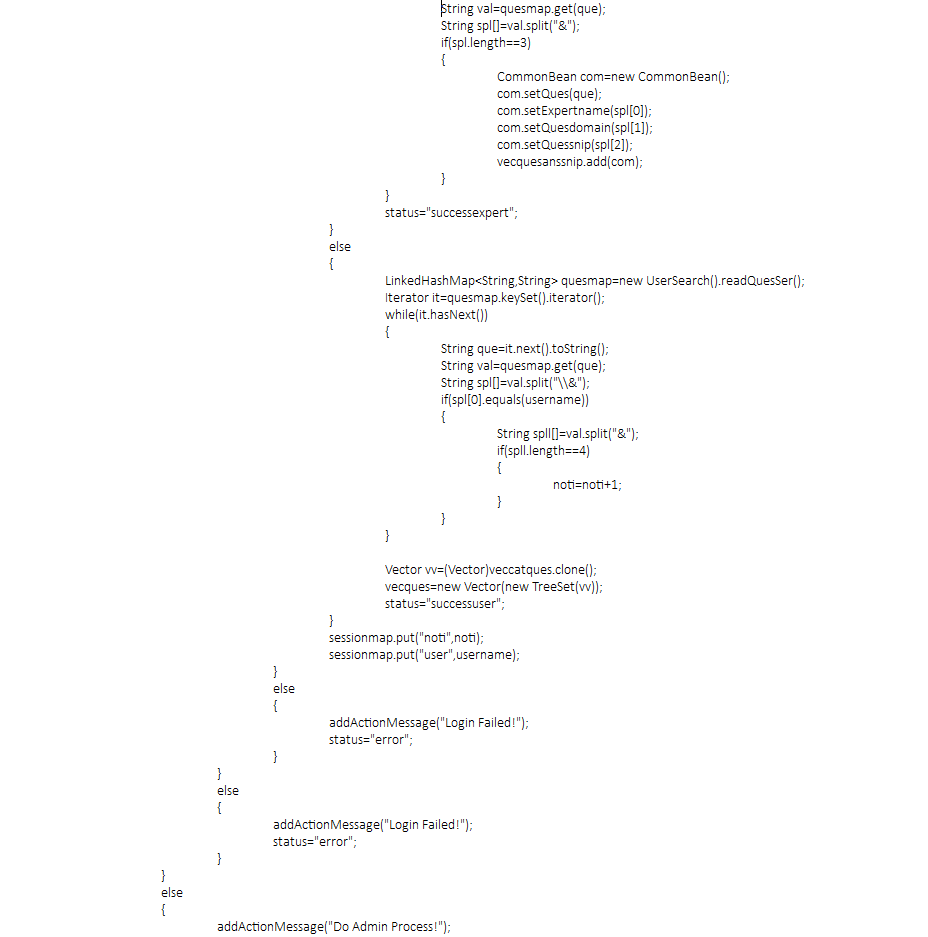


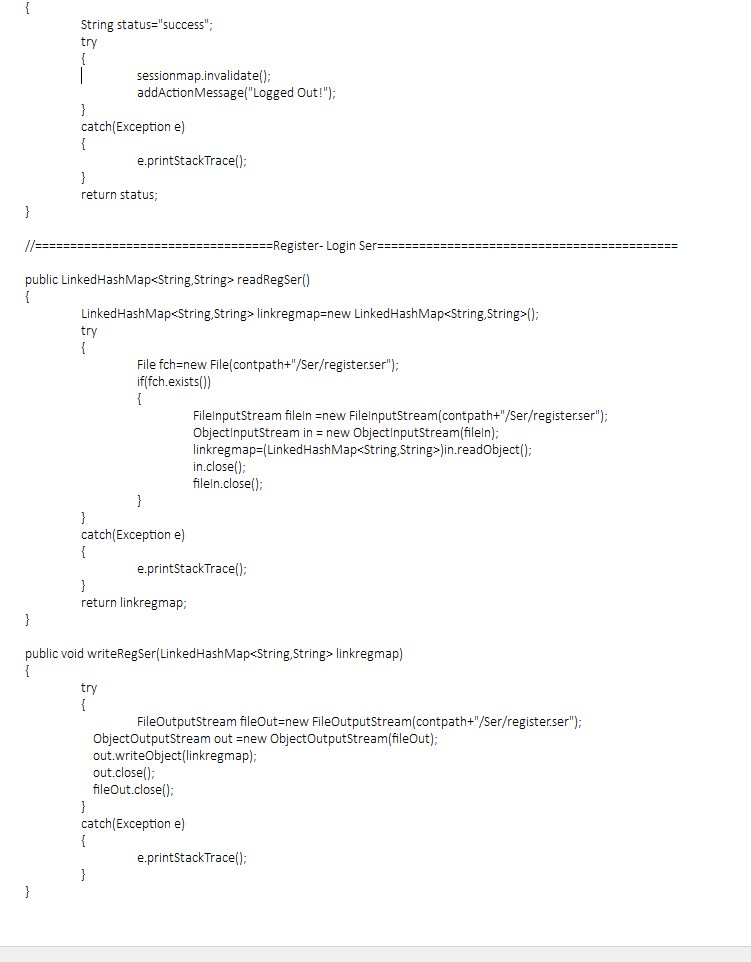




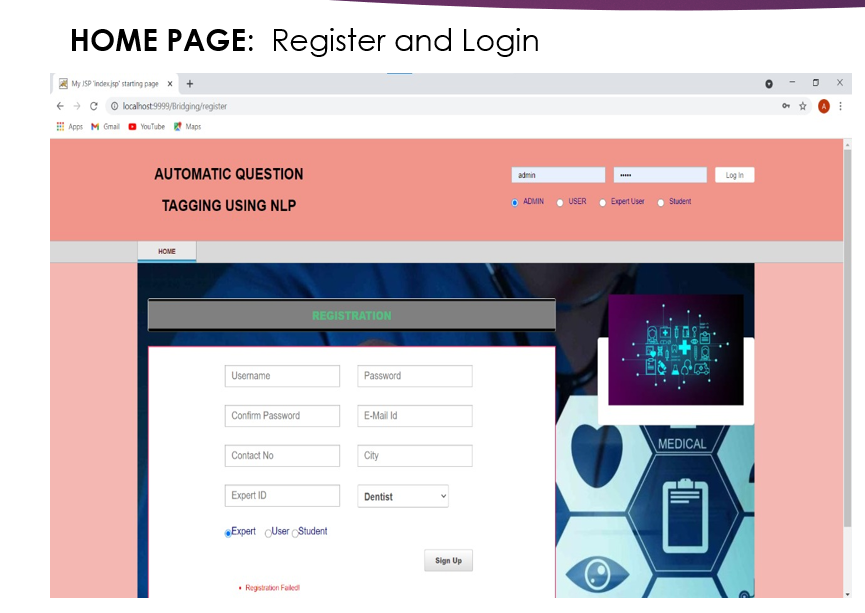
****

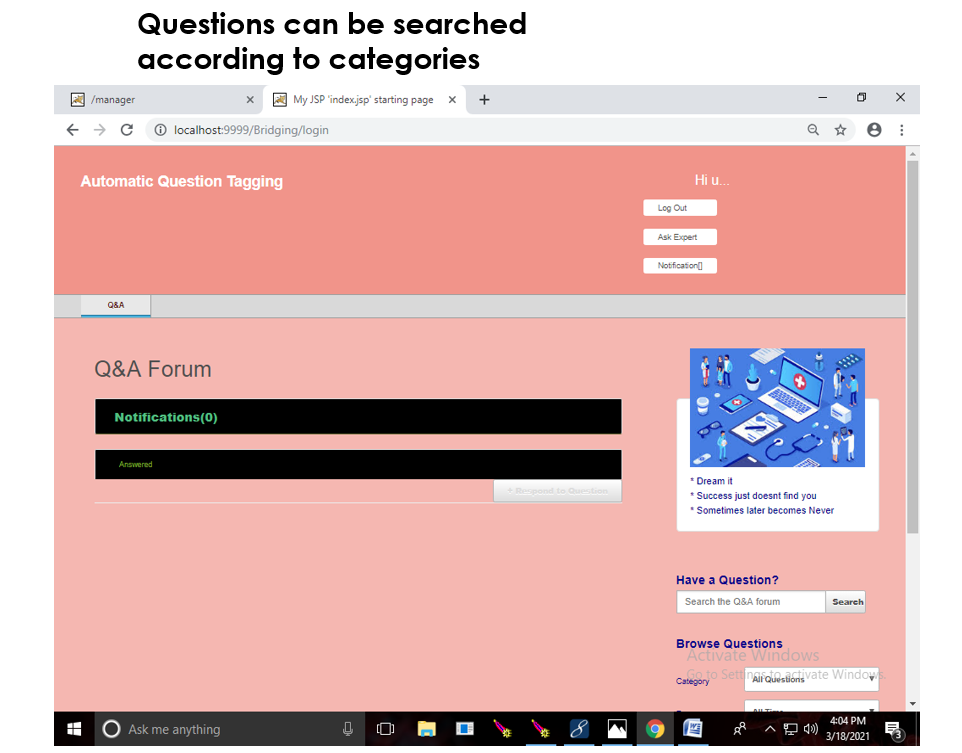
****

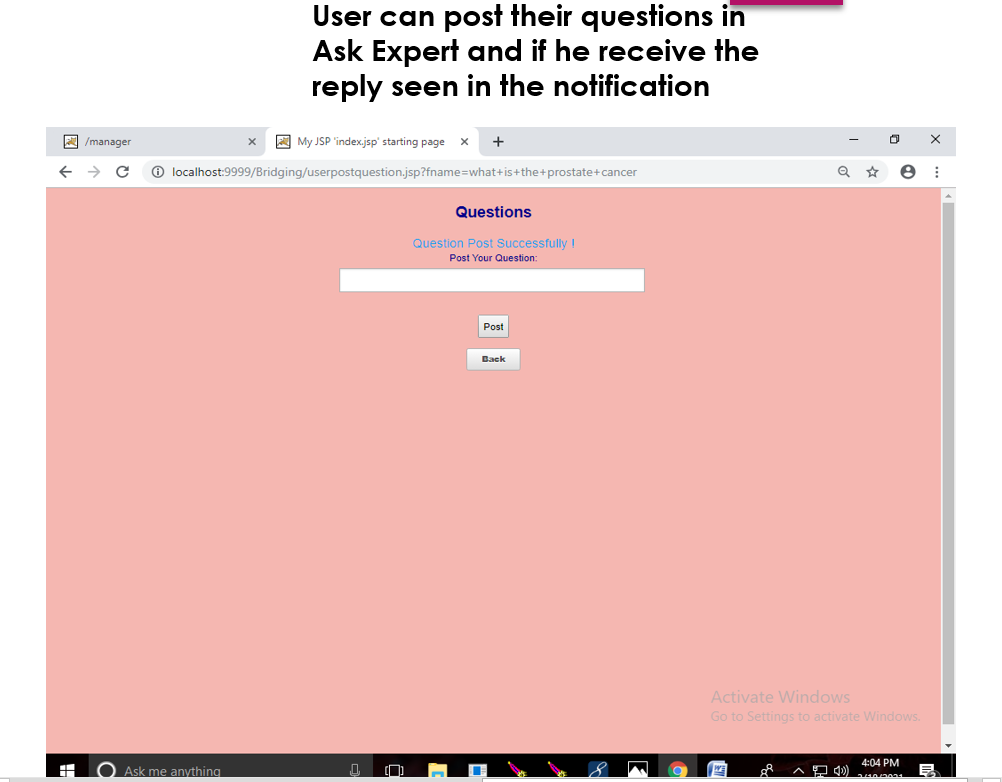
****

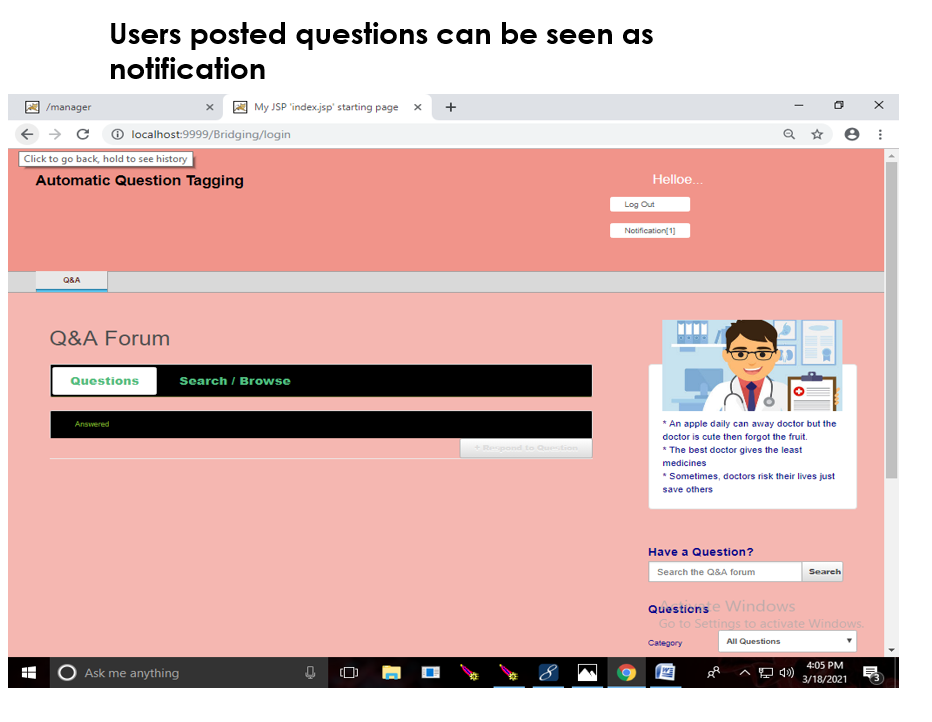
****

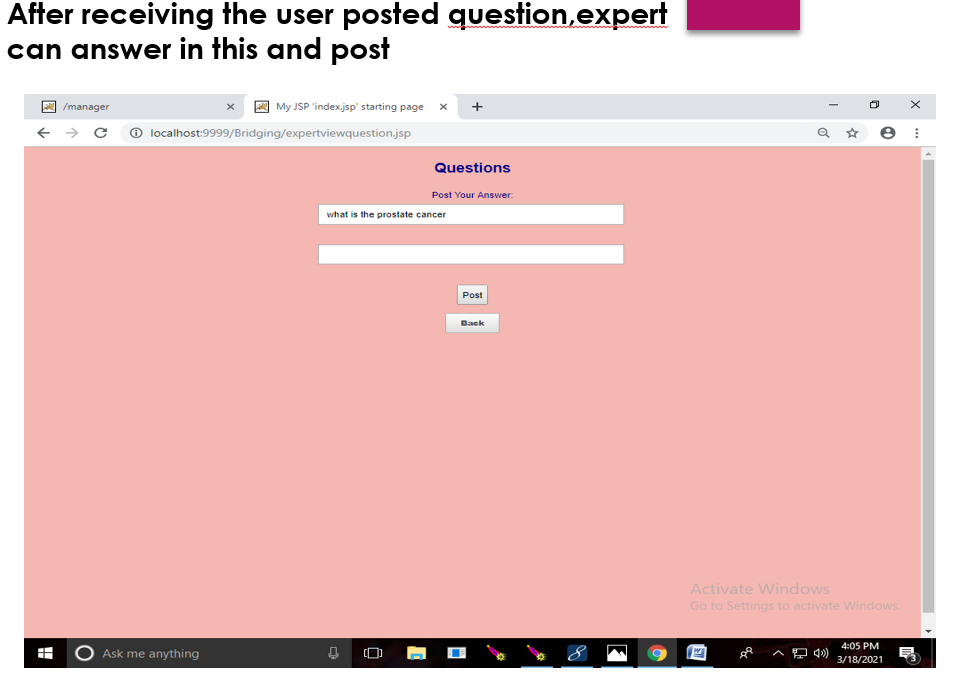
**Output Screenshot**

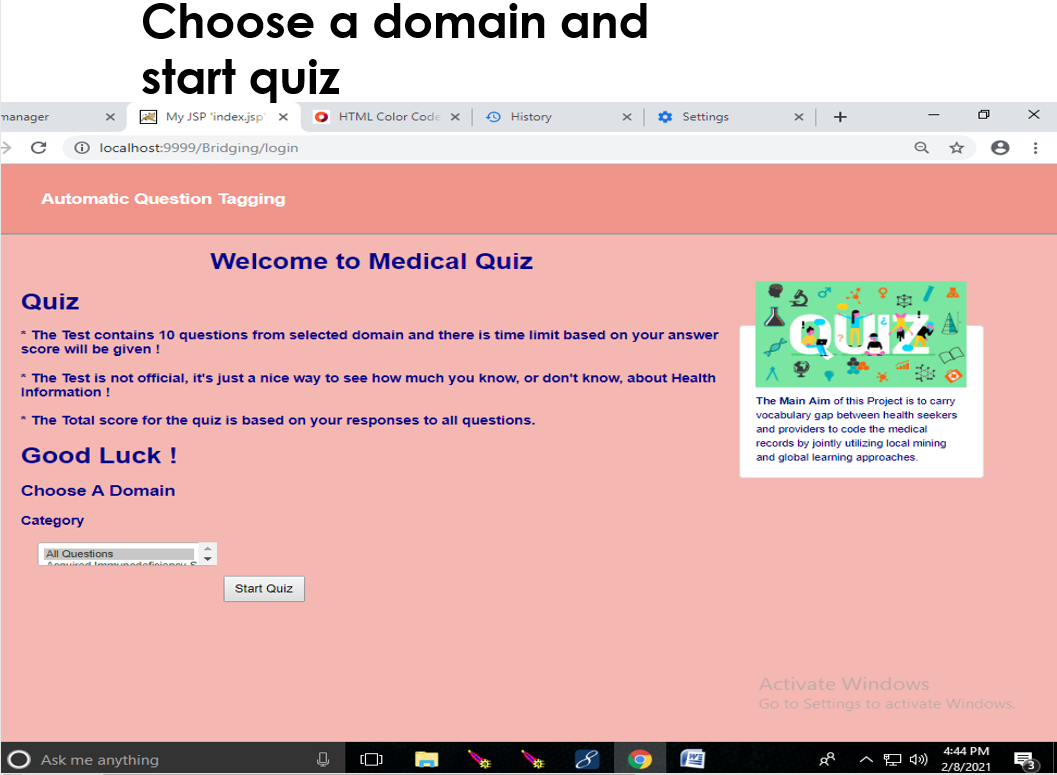
****

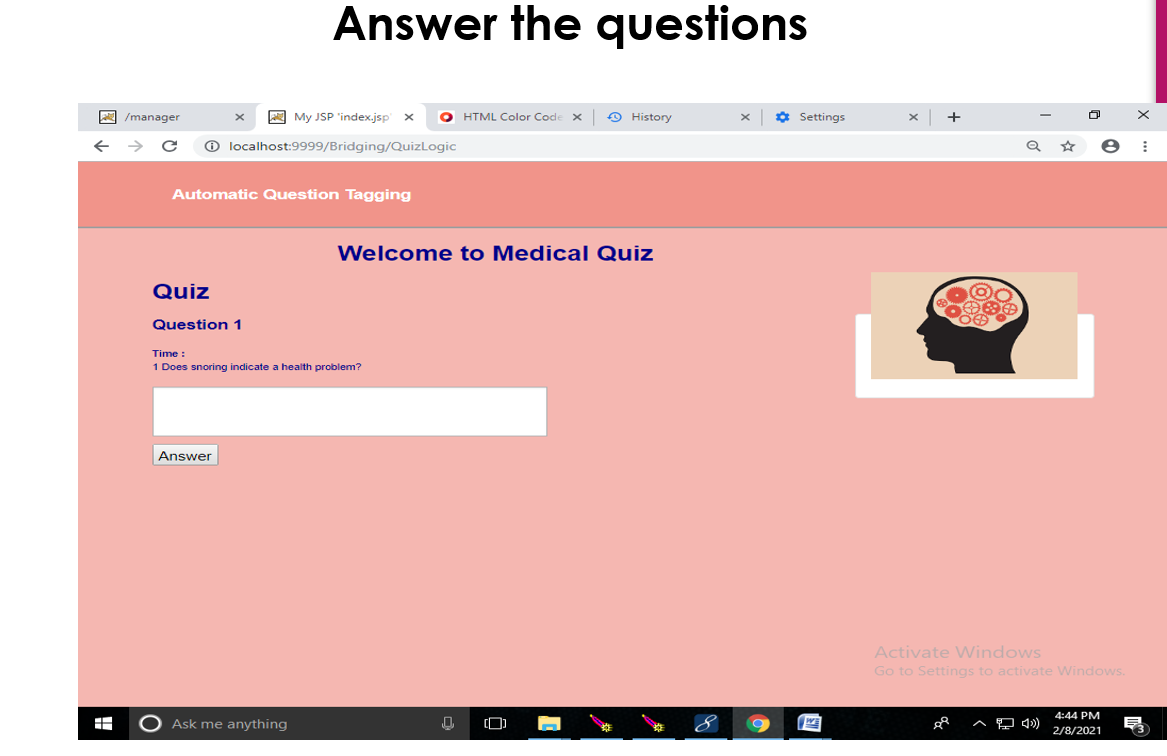


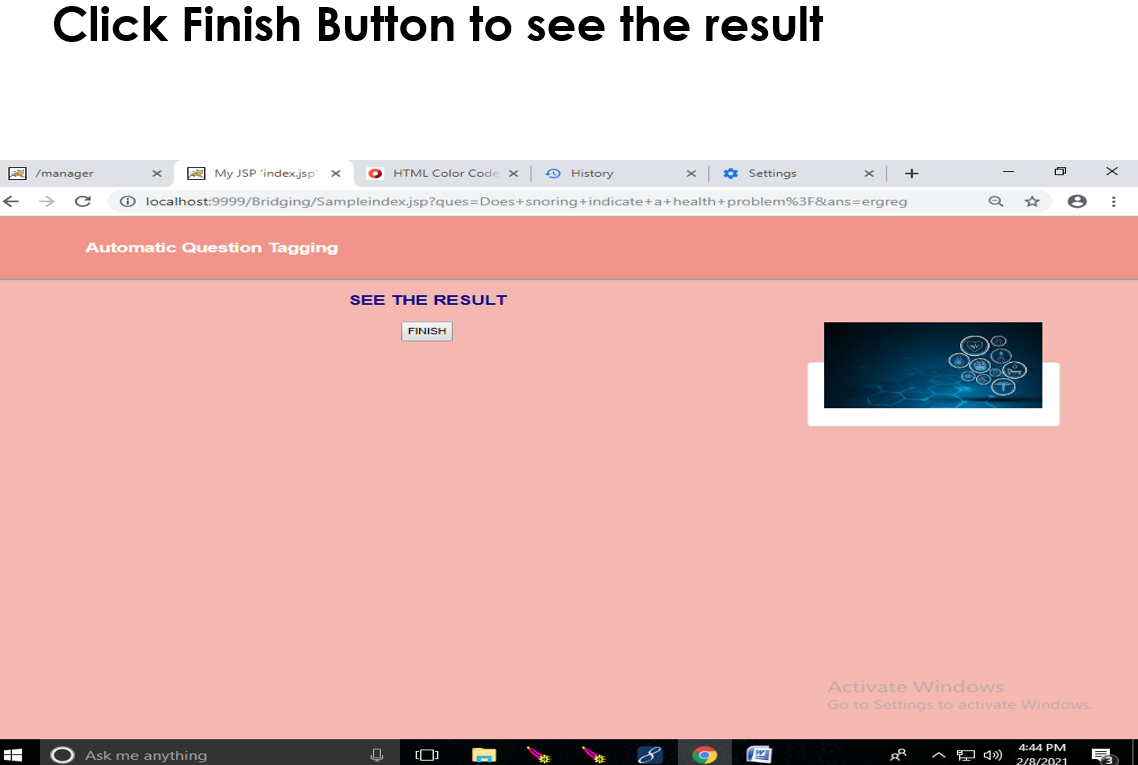
****

****



****





##### REFERENCES

1. A. Al-Hmouz, J. Shen, R. Al-Hmouz, and J. Yan, “Modeling and simulation of an adaptive neuro-fuzzy inference system (anfis) for mobile learning,” IEEE Trans. Learn. Technol., vol. 5, no. 3, pp. 226– 237, 2012.

1. P. Brusilovsky, “Knowledgetree: A distributed architecture for adaptive e-learning,” in Proc.

13th Int. World Wide Web Conf. Alternate Track Papers & Posters. ACM, 2004, pp. 104–113.

1. M. Mendicino, L. Razzaq, and N. T. Heffernan, “A comparison of traditional homework to computer-supported homework,” J. Res. on Technol. in Edu., vol. 41, no. 3, pp. 331–359, 2009.

1. J. D. Forbey and Y. S. Ben-Porath, “Computerized adaptive personality testing: a review and illustration with the mmpi-2 computerized adaptive version.” Psychological Assessment, vol. 19, no. 1, p. 14, 2007.

1. M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, “A review of relational machine learning for knowledge graphs,” Proc. IEEE, vol. 104, no. 1, pp. 11–33, 2016.

1. H. L. Kim, A. Passant, J. G. Breslin, S. Scerri, and S. Decker, “Review and alignment of tag ontologies for semantically-linked data in collaborative tagging spaces,” in Proc. IEEE Int. Conf.

Semantic Comput. IEEE, 2008, pp. 315–322.

1. M. Brut, F. Sedes, T. Jucan, R. Grigoras, and V. Charvillat, “An ontology-based modeling approach for developing a competencies-oriented collective intelligence,” Learn. to Live in the Knowl. Soc., pp. 219–222, 2008.

1. M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” IEEE Trans.

Knowl. Data Eng., vol. 26, no. 8, pp. 1819– 1837, 2014.

1. G. Salton, A. Wong, and C.-S. Yang, “A vector space model for automatic indexing,” Commun. the ACM, vol. 18, no. 11, pp. 613– 620, 1975.

1. A. K. Saha, R. K. Saha, and K. A. Schneider, “A discriminative model approach for suggesting tags automatically for stack overflow questions,” in Proc. 10th Work. Conf. Mining Software Repositories. IEEE, 2013, pp. 73–76.