ATHABASCA UNIVERSITY

AUTOMATIC TEST ITEM GENERATION FROM KNOWLEDGE STRUCTURE

BY

EBENEZER KOFI AGGREY

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER
OF SCIENCE IN INFORMATION SYSTEMS

FACULTY OF SCIENCE AND TECHNOLOGY
SCHOOL OF COMPUTING AND INFORMATION SYSTEMS

ATHABASCA UNIVERSITY
AUGUST, 2018

Approval of Thesis

The undersigned certify that they have read the thesis entitled

AUTOMATIC TEST ITEM GENERATION FROM KNOWLEDGE STRUCTURE

Submitted by

Ebenezer Aggrey

In partial fulfillment of the requirements for the degree of

Master of Science in Information Systems

The thesis examination committee certifies that the thesis

and the oral examination is approved

Supervisor:

Dr. Maiga Chang

Athabasca University

Committee Members:

Dr. Xiaokun Zhang

Athabasca University

Dr. Rita Kuo

New Mexico Institute of Mining and Technology

External Examiner:

Dr. Seng Yue Wong

University of Malaya

January 3, 2019

1 University Drive, Athabasca, AB, T9S 3A3 Canada
P: 780.675-6821 | Toll-free (CAN/U.S.) 1.800.788.9041 (ext 6821)
fgs@athabascau.ca | fgs.athabascau.ca | athabascau.ca

ACKNOWLEDGEMENTS

First of all, I want to thank Dr. Maiga Chang for his considerable patience, understanding and assistance throughout the development of this Thesis. He provided me with valuable suggestions and guidance to improve my thesis and writing conference papers as well as given me the opportunity to attend academic conferences to present the features of the Online Test System and Algorithms designed and implemented for this research.

Next, l would like to thank Dr. Rita Kuo and Dr. Xiaokun Zhang, thesis committee members, for the time spent reviewing the thesis.

Finally, would also like to thank my wife Emelia , my daughter (Maame) and my son (Ekow) for their support, encouragement and assistance to get me through this process as such l dedicate this thesis to them.

ABSTRACT

This research designs and implements an item generation engine that can automatically create higher order thinking multiple-choice items for online tests based on knowledge maps developed by teachers. Furthermore, this study leveraged questionnaire to collect data from teachers to analyze the agreement between system and participants classification of the cognitive items generated by the algorithms designed and implemented for this study. Results indicated that there are areas where the participants agreed with the systems classification of the cognitive items and in some areas they disagree. However, the system implemented for this research might go a long way to help teachers save the time they need to spend on preparing tests and assessing their students' understandings of the concepts they have learnt. Moreover, students will benefit from the online test system in terms of having opportunity to self-assess their knowledge at any time and getting rapid test results.

Table of Contents

List of Tables

List of Figures

Chapter I – INTRODUCTION

1.1 Motivation

Higher Cognitive Level Assessment (HCLA) is an integral part of learning in many domains such as education (Veeravagu, Muthusamy &, Marimuthu, 2010), pharmacy (Tiemeier, Stacy, Burke, 2011). For teachers HCLA provides a means to measure student's ability to apply concepts, analyze data or situation as well as designing, constructing and generating new ideas (Bloom & Krathwohl, 1956; Krathwohl, 2002). In the case of students, it is a way to get feedback on their learning and to improving on their performance.

Perhaps the most widely use HCLA tool is multiple-choice based test or examination because it is easy to score and makes it ideal choice in large classes (Kim, Patel, Uchizono, & Beck, 2012; Palmer, Devitt, 2007; Nicol, 2007). However, it takes considerable amount of time and effort for teachers to construct multiple choice test questions (Simon, 2011; Christensen, 2005).

In an effort to make construction of higher cognitive question less labour intensive and reliable, many researchers in literature have used different techniques (e.g. template-based- Gierl & Lai, 2013, Natural Language Processing-based- Gütl, Lankmayr, Weinhofer & Höfler, 2011, Ontology-based- Romero, Gutierrez & Caliusco, 2012, and Knowledge map-based - Hsu, Chang, Chang, Jehng, & Heh, 2002 to develop automatic item generation (AIG) systems. However, most of these systems lack the ability to generate higher multiple-choice questions to test students' critical thinking.

1.2 Goal and Contribution

The goal of this study is to design and implement test item generation engine for an Online Test System (OTS). The proposed system will be valuable tool for teachers, as it will allow them to assess students beyond memorizing and recalling facts. Moreover, it will allow students to exercise their application, analytical and creative skills. Furthermore, it will go a long way to cut down the time teachers spend on creating multiple-choice questions.

1.3 Thesis organization

Chapter II discusses relevant literature (Conceptual Models, Theoretical Frameworks, and Cognitive Classification Models) required to resolving the research issues. Chapter III describes the workflow for the algorithm as well as the major and minor algorithm designed and implemented for this study. Chapter IV explains the architecture of item generation engine as well as the system features for teachers and students. Chapter V describes the evaluation plan, data collection, analysis of the data collected and findings of this research. Finally, Chapter VI makes brief summary and discusses possible future works.

Chapter II – LITERATURE REVIEW

Problem Statement: Teachers in higher education spend considerable amount of time to prepare higher level multiple choice questions for students' assessment.

Purpose Statement: The purpose of this study is to design and implement automatic item generation engine capable of generating higher level items from knowledge map created by teachers and to determine if there will be agreement between the items classified as lower or higher cognitive item by the systems and that of the teachers.

Research Question: Will there be agreement between the items classified as lower or higher cognitive item by the system and teachers?

In order to answer the research question above and present clear ideas this section will present definitions of terms from existing relevant body of work for this study.

Effectiveness: According to International Organization for Standardization, (ISO, 1998) usability involves a users' ability to use a product to accomplish a task in a particular context with effectiveness, efficiency, and satisfaction. Effectiveness measures the percentage of goal users achieved for using a product, the error rates for performing a task (Preece, Rogers & Sharp, 2002). Efficiency measures the outcome of a user's interaction with a system such as time to complete a task, the error rate, and the amount of effort. Satisfaction deals with users comfort and attitude toward the use of a particular product (Bevan & Macleod, 1994).

Effective item: This research will design and implement an algorithm to transform proposed template model to higher order multiple-choice items. If the algorithm generates higher order multiple-choice items then the item is effective.

2.1 Knowledge Structure and Bloom Taxonomy

Knowledge structure (KS) depicts concepts and their relationship to each other (Day, Arthur, & Gettman, 2001). Ontology (Gruber, 1993; Oberle, Guarino & Staab, 2009) and Knowledge map KM (Hsu, Chang, Chang, Jehng, & Heh, 2002) are kinds of knowledge structure. Although, both are widely applied in Philosophy, Computer Science, Artificial Intelligence, Software Engineering, and related fields to organize knowledge (Taye, 2010) there are basic differences between them. One of the biggest differences between ontology and KM is that whereas, it is possible to define rules, restrictions, and axioms between concepts using semantic web tools at conceptual model design time (Chandrasekaran, Josephson & Benjamins,1999), KM may defines constraints between concepts at system implementation time. In the case of ontologies, in contrast to KM is closely guided to standardized technical implementations, specifically depend on the deferent formal languages utilized in ontology engineering (Noy & Hafner, 1997).

Many researchers in literature have developed ontology-based (Papasalouros, Kanaris, & Kotis, 2008; Tosic & Cubic 2009; Papasalouros, Kotis, & Nikitakos, 2010; Romero, Gutierrez, & Caliusco, 2012) and knowledge map-based (Chang, Kuo, Chen, Liu, & Heh, 2008; Chang & Kuo, 2009) automatic test item generation systems.

Bloom's Taxonomy (BT) (Bloom & Krathwohl, 1956; Krathwohl, 2002) provides a useful framework to characterize factual, conceptual, procedural, and meta-cognitive knowledge as well as cognitive processes to demonstrate learning. Furthermore, researchers (Chang, Kuo, Chen, Liu, & Heh, 2008) have applied it successfully to evaluate cognitive abilities of students and teachers have leveraged it to prepare learning objectives. Bloom's Taxonomy uses verbs (i.e. remember, understand, apply, analyze, evaluate, and create) [1] as order of cognitive level and classify analyze, evaluate, and create as the levels that can be used to measure higher cognitive. This research will apply Bloom's Taxonomy to classify the item to be generated as higher cognitive item. This research will evaluate the items that will be classified by the system and participants',

[1] http://www.uleth.ca/teachingcentre/blooms-taxonomy

hence this research will prepare questionnaire with forty items to be generated by the item generated engine to be designed and implemented.

2.1.1 How to Design a Knowledge Structure for Automatic Test Item Generation

A number of researchers (O'Donnell, Dansereau, & Hall, 2002; Hsu, Chang, Chang, Jehng & Heh, 2002; Nesbit & Adesope, 2006; Yang, Song, Lu, & Zhang, 2007; Qiu, 2015) have proposed approaches in literature to design knowledge structure (KS) (e.g. Knowledge Map). The summary of the existing conceptual model and the steps to design KS shown below.

The knowledge structure conceptual model and the steps this research defined knowledge structure as KS = (C, A, B, R) where

1. C = Set of concepts
2. A = Set of attribute of the concept
3. B = Set of behaviour of the concept
4. R = Relationship between concept with another concept

Steps to design Knowledge Structure

1. Define a concept in a particular domain
2. Define the attributes of the defined concept
3. Define the behaviour the defined concept can do
4. If the defined concepts has relationship with other concept then define the type (e.g. "Typeof," "Partof," etc.)
5. Repeat step 1-4 until the desired knowledge is formed

The knowledge structure (KS) as described above will be used as conceptual model in this research. The algorithms developed will leverage the KS to generate the test items.

For example in Computer Science data structures course, a Queue is a linear data structure. Items can be added (enqueue) and remove (dequeue) from the queue. Other behaviour of a queue is the ability to remove item from the front (peek). Using the definition in section 2.2.1

1. C = Linear Data Structures
2. A= Internal attributes of the Queue such as buffer (array) to hold the items added
3. B = Behaviors of the queue such as Enqueue, Dequeue, Peek, etc.
4. R = Relationship between Linear Data Structures and Queue. i.e. Queue is a type of Linear Data Structures

2.2 Concept of Template

A template can be defined as abstract representation of concept, which serves as a pattern for reference. The idea of template has deferent meaning in many contexts. For example in word processing template means creating a preformatted word document, say business letter with a placeholder for client information. The placeholder will then be replaced by specific client information when sending the letter.

Similarly, researchers in literature have applied template concept to develop template-based item generation systems. With this approach, researchers first define placeholder for stimulus, stem, distractor, and the key on template. Then they substitute the placeholders with actual parameters at runtime with the help of parsers to construct the multiple-choice questions (Gierl & Lai, 2013; Stanescu, Spahiu & Ion, 2008).

Template-based approach is cost effective in the sense that it can be uses to generate large amount test items by manipulating stimulus, stem, and options placeholder only. With large pool of test questions available at the question banks educators will be able to administer assessments with minimum effort (Gierl, Lai, & Turner, 2012). Moreover, common errors in developing multiple choice item such as omissions and additions of words, phrases, spelling, punctuations, capitalization, item structure, typeface, formatting can be avoided(Gierl, Lai, Hogan, & Matovinovic, 2015). This is because only the stimulus, stem, and options of the questions are being changed during question generation (Schmeiser & Welch, 2006).Furthermore, Template-based generation is relative straightforward to implement and has minimal development time (Deane & Sheehan, 2003).However, Deane and Sheehan (2003) argues that it lacks theory as it does not utilize any linguistic knowledge. They also pointed out that template-based approach tend to manipulate string hence it require storing and maintaining large amount of list to be able to replace the placeholders. In addition, it requires time and effort when multiple language generation is required as it will require separate template translation engine to be developed.

2.2.1 How to Design Template for Test Item Generation

Researchers have proposed and presented techniques on to design template to generate multiple choice test items in various domain such as (Medicine -Al-Rukban,2006), (Biology- Alves, Gierl, &

Lai,2010), and (Dentistry- Lai, Gierl, Byrne, Spielman, & Waldschmidt ,2011). Although, these templates are designed for different domains they share common features which includes Stimulus (Text, Images, graphs, etc.), Stem, Answer Options, Key which is the answer to the question and a set of rules to ensure that valid values are inserted into the template place holders. These rules are domain specific. Applying the components of the them template this research will defined a template as T = (S, $S_{M,}$ $A_{o,}$ K, R), where

1. S= Stimulus
2. $S_{M\,=}$ Stem of the test item
3. $A_{o\,=}$ Answer Options
4. $K \subset A_o$
5. R = Set of place holder rules

The template discussed above will be use to unsure that common errors in developing multiple choice item such as omissions and additions of words, phrases, spelling, item structure are avoid and test items are formatted correctly . Using the definition in section 2.3.1 a test item template for a typical computer science data structure can be describe as

Stimulus (S)

A computer programmer want to design and implement address book with entries sorted by persons last name in alphabetical order.

Stem of the test item (S_M)

Which of the following data structure is likely to be used?

Answer Options (A_o)

A. Unsorted List

B. Sorted List

C. Stack

D. Balanced Search Tree

Correct Answer (Key)

Answer Option D

The item in section 2.2.1 is an example of lower level item generated from the work of (Chang, Kuo, Chen, Liu, & Heh, 2008; Chang & Kuo, 2009). When a teacher wants the existing system to generate such an item, first, he or she will select lower cognitive type (ct) such as Remember and a concept node c_t (Balance Search Tree). Then the variables ct and c_t was pass to existing algorithm to generate the item. The stimulus and the stem are static. To generate the answer options for the item the algorithm uses the concept node selected as the key of the item, then retrieves its siblings to be the distractors.

Set of template place holder rules (R)

Examples of such rules are (1) Each test item must have four answer options (2) Each test item must have a stimulus (e.g. Text block, Graph, Algorithm, Code fragment, etc.) (3) Each test item must have a stem that pose a question.

2.3 Research Objectives and Issues

This section describes the tasks to be accomplished to achieve the two objectives.

Objective #1: Generation of higher order multiple choice questions for an online test based on the knowledge map created by the teacher.

This research intend to generate higher order multiple-choice for OTS hence for this objective will focus reviewing relevant body of work on techniques and methodologies to generate multiple-choice questions will be compared and contrast to find gaps to improve.

Objective #2: Evaluation of lower and higher item classification by the item generation engine and participants.

This research will evaluate the items classified by the systems and the participants to see if teachers agree with the system classification. Therefore, the focus of this objective is to find acceptable taxonomy to classify items to be generated by the item generation engine into lower and higher cognitive item from literature.

Each objective has one or more research issues needed to be solved by this research.

Objective #1: to generate higher order multiple-choice questions for an online test based on the knowledge map created by the teacher.

The Online Test System shall provide functionality for teachers to create, import, and manage their knowledge maps In order to transform these knowledge maps to higher order multiple question the following issues has to be resolved to accomplish the objectives.

Issue #1-1: How to classify multiple-choice test item as higher order Item?

Blooms Taxonomy (BT) as described in detail in section 2.4 provides a framework to classify cognitive complexity. Many researcher and education system have applied BT to prepare course process skills objectives and cognitive rigor matrix. Likewise, this study will apply BT to classify higher order thinking items.

Issue #1-2: How to generate different types of higher order thinking questions?

In learning a subject, area such as science test may have different types of higher cognitive level items such as the once listed in College Science Teachers Guide to Assessment[2] (NSTA, 2009) page 37, paragraph 5

- Premise-Consequence: Students are required to identify the correct outcome of a given circumstance.
- Analogy Questions: Students map out the relationship between two items into another context.
- Case Study Questions: In this case, students use stimulus such as a single well-written paragraph or graph to answer questions and several follow up questions.

[2] http://static.nsta.org/files/PB231Xweb.pdf

- Problem Solving/Solution Evaluation Questions: Students are supposed to evaluate problem and proposed solution based on given criteria.

At this stage the types of higher cognitive level item listed are guideline and will consider what is possible to generate in computer science domain, in particular Data Structures, and the paragraph might change accordingly.

To generate different types of test item, this study will design a conceptual model, which encapsulate template model and knowledge map (KM) as discussed in detail in section 2.3 and section 2.4 respectively. Furthermore, this research will implement an algorithm and parser to replace the placeholder in the template with actual values stored in the KM to generate test item.

Issue #1-3: How to prevent common errors associated with test item generation?

Some of the common errors relating to test item generation are grammar, formatting, punctuations, capitalization, item structure, typeface etc. This study will leverage template as describe in issue 1-2 to avoid these issues.

Objective #2: to evaluate lower and higher item classification by the item generation engine and participants.

One of the goals of the users of the proposed systems is to generate effective higher order thinking items from the knowledge map they create. In order, the compare the items classified by the systems and that of the participants will require answering the flowing questions.

Issue #2-1: How to obtain data to do classification analysis?

As part of this research item generation engine will be developed and integrated with the existing Online Test System. The system will be used to generate both higher and lower cognitive items. The items generated by the system will serve as questions for participants to classify into lower or higher cognitive item.

Issue #2-2: How to evaluate whether or not the generated items can test student's higher cognitive abilities?

First of all, the system to be implemented will be used to generate and automatically classify 40 items into their respective cognitive level (Remember, Understand, Apply, Analyze and Evaluate). Then, teachers teaching data structure, computer science and related fields will be requested to respond to a questionnaire. The questionnaire will have two sections. The first section will collect teachers' information like the subject they teach and their gender. The second section will contain the 40 items generated by the systems for which the teachers will be requested to classify each item into its cognitive level. After that, the responses from the questionnaire will be analyzed using statistical method (inter-rater) reliability analysis to see if there will be agreement between the items classified by the system and teachers.

Chapter III – ALGORITHMS AND WORKFLOW OF GENERATING HIGHER COGNITIVE ITEMS

This chapter will present detailed description of how the higher cognitive level items (HCLI) will be generated. First, the symbols to be used for the algorithm workflow and item generation algorithms will be defined and tabulated (Appendix A). Secondly, the workflow of algorithms will be presented with explanation. Finally, the algorithms to generate HCLI will be presented and explained line by line with dataset.

3.1 Workflow of the Item Generation Algorithms

Figure 1 shows the symbols used to represent each object used in the algorithm workflow. For example a teacher participating in the workflow is represented by human like figure on the first column cell of the legend and repeated action is represented as incomplete circle in the second cell of column #5 of the legend. Each of the major algorithm (Item generation) and supporting algorithm (Stimulus, Stem, Item Answer Options) and given their respective symbols as well as the algorithm to retrieve the items rules for the item generation algorithm.

Legend					
	Teacher		Cognitive Types		Concept Hierarchy
	Item Generation Algorithm		Item Rules data source		loop
	Item Rules Attributes data source		Stimulus Creation Algorithm		Stem Creation Algorithm
	Item Answer Option Creation		Item Created		

Figure 1.The legend of item generation algorithm workflow

Figure 2 shows the Item generation algorithm workflow. The workflow is items are connected with arrows with numbers on them. The numbers represent the sequence of operation. The operation starts from step 1 and ends at step 6. In the case where the operation repeats it is represented by a loop.

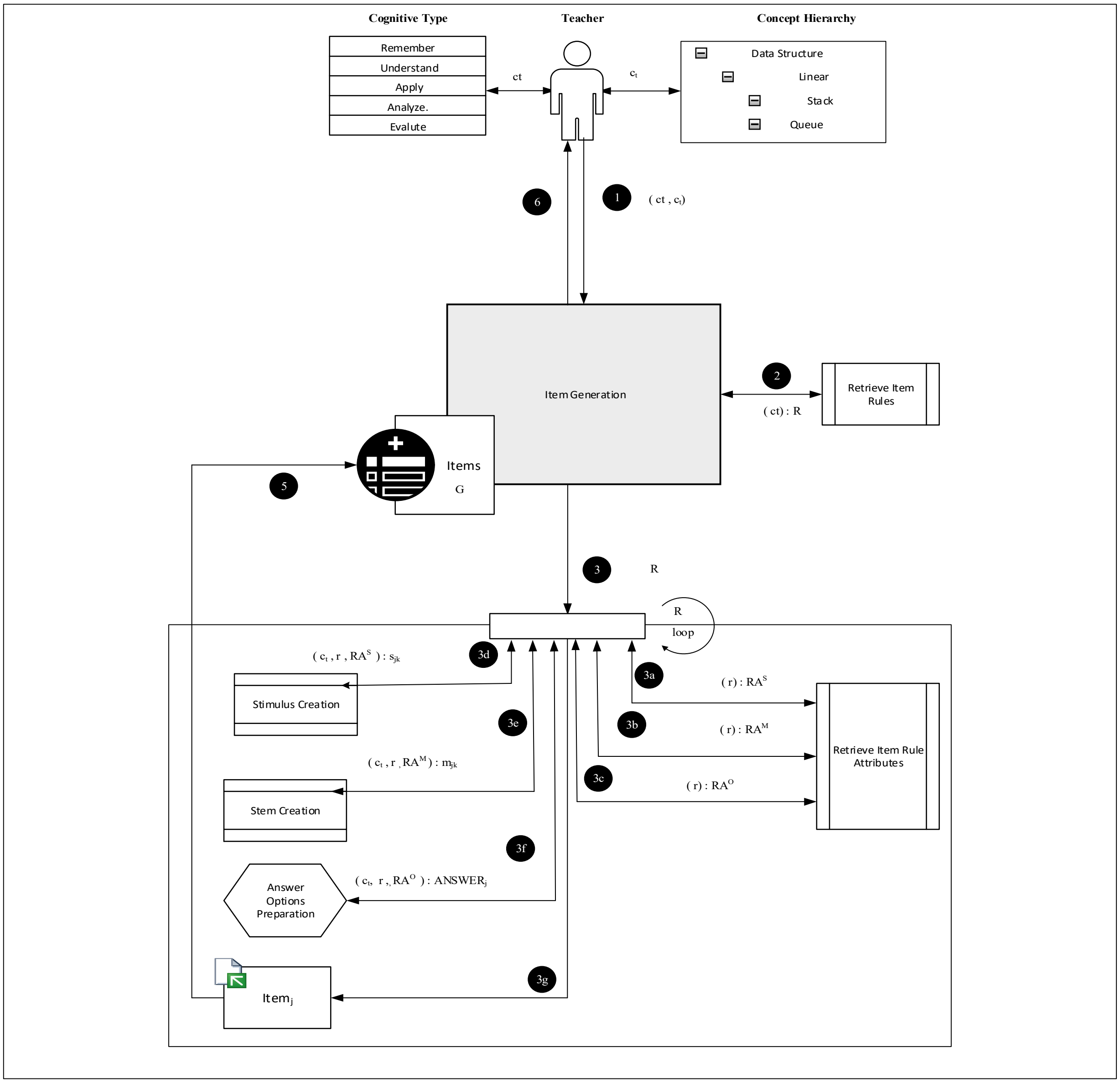


Figure 2.Workflow of the item generation algorithm.

The workflow has six steps described as following in details as shown with the numbers 1 to 6.

A teacher can ask the system to generate items by selecting cognitive type which can be either Remember, Understand, Apply, Analyze or Evaluate and denoted by ct and a concept node (c_t) from the concept hierarchy. The rest of the symbols use for the algorithm workflow and algorithms such as the set of answer options ANSWERj are tabulated in Appendix A. The selections is then are sent as inputs to the "Item Generation" function as Step 1 shows. This function is responsible for calling the supporting algorithms to generate the right cognitive items base on the cognitive type selected by the teacher. For example if the teacher selects Apply Cognitive type then Apply items will be generated. Likewise, if the teacher selects Analyze cognitive type is selected then analyze items will be generated.

1. The "Item Generation" Algorithm uses ct to retrieve item rules of cognitive type item rule (R) as Step 2 shows.
2. After the algorithm gets R, it enters into a loop enumerating through the rules and perform the following tasks sequentially:

(3a) Request for stimulus rule attributes (RA^S) passing the current rule (r) as parameter

(3b) Request stem rule attributes (RA^M) passing the current rule (r) as parameter

(3c) Request answer options rule attributes (RA^O) passing the current rule (r) as parameter

(3d) Ask "Stimulus Creation" supporting algorithm to create the item stimulus (s_{jk}) by passing cognitive type c_t, a rule r, and the retrieved stimulus rule attributes set RA^S.

(3e) Execute "Stem Creation Algorithm" to create the item stem (m_{jk}) passing c_t, current rule (r) and RA^M.

(3f) Call "Answer Options Preparation Algorithm" to prepare answer options and their key ($ANSWER_j$) passing c_t, current rule (r), RA^O.

(4g) Create an $item_j$ with the outputs of Steps 3d to 3f, {s_{jk}, m_{jk}, $ANSWER_j$} and add to the item generated set G as shown in Step 5.

3. In the end the "Apply Item Generation" algorithm passes back the generated item set G to the caller (as Step 5 shows) and finally to the teacher as Step 6 shows.

3.2 Item Generation Algorithms

This sections discusses the major and minor algorithms that work together to generate the items as well the rules and rule attribute the algorithms use to generate the items.

The Item Generation algorithm leverages rules to generate items. The rules and the associated rule attributes are stored in Table 1 and 2 respectively. Table 2 stores the templates for item stimulus, stem , and answers (column #3-#4) for a particular cognitive type (column #2) item and they are identified by unique id (column #1) . Each rule in Table 1 has associated attribute name which can be supporting type such as actor, task and concept attributes like description, operation, application etc. as shown in column #3 in Table 2.

Table 1. Rules for item generation

Rule Id	**Cognitive Type**	**Stimulus Template**	**Stem Template**	**Answer Options Template**
1	Apply	A {actor} was asked to implement {concept node name} data structure to be used in {task} that simulate {description}.	Select the suitable functions the {actor} has to implement	{{option letter}}. {{attribute names}}
2	Apply	A {actor} designed and implemented {concept node name} data structure for {task}. Upon uniting testing it was found that the data structure {description}.	Choose the possible data structure the {actor} implemented	{{option letter}}. {{attribute names}}
3	Apply	A {actor} was asked to implement {software task} leveraging {concept node name} data structure capable of {application}	Select the set of methods the {actor} has to implement to accomplish the task.	{{Option letter}}. {{attribute names}}
4	Apply	A {actor} was asked to implement {concept node name} data structure with the following specification: (a) stratify {performance measure} criteria (b) {application}	Select the appropriate functions the {actor} has to implement to satisfy the specification	{{option letter}}. {{attribute names}}

5	Analyze	A {actor} designed and implemented {concept node name} to be used in {task}
		The data structure implemented Array-Based List interface which supports M1 () operation.
		The {actor}} implemented the M1() as follows :
		for (int i =1 ; i <= c ; i++)
		{
		// some O(1) expressions
		}
			If c is constant then what is the time complexity of the {actor} expect for the algorithm?	{{option letter}}. {{attribute names

6	Analyze	An algorithm a {actor} implemented for one of the methods of {concept node name} spent 10 ms to process 4000 data items.	How much time will the {actor} expect to be spent to process {size} if computer processing T(n) of the algorithm as function of time complexity O(f(n) is defined as T(n)=cn^3 where c is constant.	{{option letter}}. {{attribute names}}
7	Evaluate	A {actor} is considering using a third party data structure(TPDS) to implement generic data structure S which should be able process about 10,000 data items. Additionally, it should implement {data structure name} interface and provide algorithm for the M1() operation. The computer processing time in relationship to time complexity for the M1() operation of the TPDS is as follows. A1 (n)=n^2 ,A2(n) = 5.3n^3, A3(n)=log n, A4(n)=200n^3	Which algorithms which do the {actor} expect to be better in terms of Big-Oh sense for the implementation of the {task}.	{{option letter}}. {{attribute names}}
8	Evaluate	Four {actor}s A, B, C D design and implemented M1() operation of data structure which implements {data structure name} interface. The computer processing as a function of time complexity for the M1 () operation algorithms were analysed and the result is shown below: TA(n) =n^2 , TB(n) = n^3, TC(n)=log n, TD(n) = n	Choose the algorithm which is better in terms of Big-Oh sense for the implementation of the {task}.	{{option letter}}. {{attribute names}}

Table 2. Item rule attributes

Rule Attribute Id	Rule Id	Attribute Name	Item Part	Attribute Source
1	1	actor	stimulus	supporting
2	1	task	stimulus	supporting
3	1	description	stimulus	concept schema
4	1	concept node name	stimulus	concept node
5	1	actor	stem	supporting
6	1	operation	answer Option	concept schema
7	2	actor	stimulus	supporting
8	2	concept node name	stimulus	concept node
9	2	task	stimulus	supporting
10	2	description	stimulus	concept schema
11	2	actor	stem	supporting
12	2	concept node name	answer Option	concept node
13	3	actor	stimulus	supporting
14	3	task	stimulus	supporting
15	3	concept node name	stimulus	concept node
16	3	application	stimulus	concept schema
17	3	actor	stem	supporting
18	3	operation	answer Option	concept schema

19	4	actor	stimulus	supporting
20	4	concept node name	stimulus	concept node
21	4	performance measure	stimulus	concept schema
22	4	application	stimulus	concept schema
23	4	actor	stem	supporting
24	4	operation	answer Option	concept schema
25	5	actor	stimulus	supporting
26	5	concept node name	stimulus	concept node
27	5	task	stimulus	supporting
27	5	actor	stem	supporting
29	5	supporting	Answer option	supporting
30	6	actor	stimulus	supporting
31	6	concept node name	stimulus	concept node
32	6	actor	stem	supporting
33	6	supporting	stem	supporting
34	6	description	answer option	concept schema
35	7	actor	stimulus	supporting
36	7	concept node name	stimulus	concept node
37	7	task	stem	supporting
38	7	description	answer option	concept schema
39	8	actor	stimulus	supporting
40	8	concept node name	stimulus	concept node
41	8	task	stem	supporting
42	8	description	answer option	concept schema

The algorithm shown in Figure 3 generates items. When the algorithm is called the items generated variable G is first initialized to empty set (at Line # 1); then it retrieves items rules R passing the cognitive level ct the teacher selected as parameter (at Line # 2). From (line # 3- #11) R is enumerated selecting rule attributes for item stimulus RA^S, stem RA^M and answer options RA^O respectively passing the current rule r.

ALGORITHM 1 : Item Generation

Input : c_t, a concept node selected ; ct, cognitive type

Output : $G = \{g_1, g_2 \ldots, g_n\}$, set of item generated

1 : G ← { Ø }
2 : R ← Retrieve Item Rules by ct
3 : for each item rule r in R
4 : RA^S ← Select Rule Attributes for item stimulus (r)
5 : RA^M ← Select Rule Attributes for item stem (r)
6 : RA^O ← Select Rule Attributes for item Answer Options (r)
7 : s_{jk} ← Stimulus Creation (c_t , r , RA^S)
8 : m_{jk} ← Stem Creation (c_t , r , RA^M)
9 : $ANSWER_j$ ← Answer Options Preparation (r, c_t ,RA^O)
10: G ← G ∪ {s_{jk} , m_{jk} , $ANSWER_j$ }
11: end for

Figure 3. Item Generation Algorithm

For example in the first iteration of item rules R (from Line# 4- Line#6) of Figure 4 if Rule Id=1 passed as parameters to select rule attribute for stem it will return RA^S from Table 2 which will contain records with Rule Attribute Id (1-4) with attribute names (actor, task, description, concept node name,). Similarly, RA^M will have records with Rule Attribute Id (5) with attribute name (actor) and RA^O items will be Rule Attribute Id =6 with attribute name =operation.

In Line # 7 "Stimulus Creation" algorithm is called passing concept node c_t selected, rule (r), and RA^S which then returns formatted item stimulus (s_{jk}); Then "Stem Creation" algorithm is called with rule (r), which returns formatted stem (m_{jk}) (at Line # 8). At Line # 9, "Answer Options Preparation" algorithm is asked to prepare the answer options accepting r, ct, and RA^O as parameters and returns set of answer options $ANSWER_j$. The stimulus, stem, and answer options created from Line # 7, # 8, # 9 respectively is used to create $item_j$ {s_{jk}, m_{jk}, $ANSWER_j$} and added to item generate set G. After all the iteration of the rules completed i.e. from Line # 3 to # 12 the items as shown in Table .4 will be generated.

When the algorithm in Figure 4 "Stimulus Creation" is called with concept node selected c_t, and element of stimulus rule ($r \in R$), item rule attribute set RA. First, the algorithm initialize stimulus attribute value pair variable AV^t to empty set (at Line #1): then loop through RA^S (from Line #2 - # 15) choosing the attribute name value name (ra_k) of rule attribute object (.i.e. ra_k). For "actor" (Line #4) it retrieves random actor from actor set (at Line #7) and added to AV^t with $name_k$ If the case is "task" (Line #5) it retrieves attribute value task (Line #6) and add it to AV^t with its attribute name (Line #10) . if the case is "other" (at Line # 11) the algorithm retrieves attribute value pair from the concept schema passing c_t , name (ra_k) , ra_k as parameters and add it to AV^t (at line #13). Lastly, (at line #16) it reads the item stimulus template TS from the item rule r and format the stimulus template with AV^t (at Line # 17) and returns item stimulus s_{jk}.

```
ALGORITHM 2: Stimulus Creation

Input : c_t , concept node selected
        r ∈ R , item rule.
        RA , set of item rule attributes for stimulus

Output : s_jk , item stimulus created
Local : AV^t = { av^t_1, av^t_2 , … , av^t_n }
        AV_j^S ⊂ AV^t ← { ∅ }
AV^t ← { ∅ }
for each rule attribute ra_k in RA
    name_k ← Get rule attribute name from rule attribute passing ra_k
    case name_k of
     actor:
      actor_jk ← Retrieve random actor from actors
       AV^t ← AV^t ∪ { name_k , actor_jk }
     task:
     task_jk ← Retrieve random task from tasks
       AV_j ← AV_j ∪ { name_k , task_jk }
    other:
       AV_j^S ← Retrieve attribute value pair passing c_t , ^n name_k , ra_k
       AV^t ← AV^t ∪ AV_j^S
     end case
  end for
  TS ← Read stimulus template from item rule r
  s_jk ← Format stimulus template passing TS and AV^t
```

Figure 4. Stimulus Creation Algorithm.

The inputs to "Stem Creation" algorithm in Figure 5 are concept node selected c_t, item rule r, and item rule attributes RA. When the algorithm is called it first initializes stem attribute value pair AV^t to empty set (at Line #1). Then it goes on to loop through the rule attribute RA^M (Line # 2 to # 12) selecting option of the rule attribute object attribute name $name_k$. If the case is "actor" (at line # 4) it retrieves random actor form set of actor and adds it to AV^t (at line # 4). In case of "task" it retrieves random task from set of task and added to AV^t (at Line #10).Finally, it reads the stem template TS from the item rule r (at line #13) and format the stem template with TS and AV^t (at Line #14) which returns m_{jk}.

```
ALGORITHM 3 : Stem Creation

Input : c_t , concept node selected
        r ∈ R, item rule.
        RA , set of item rule attributes for stem

Output : m_jk , stem of an item

1: AV^t ← { ∅ }
2: for each rule attribute ra_k in RA
3     name_k ← Get rule attribute name from rule attribute passing ra_k
4:      if name_k = actor then
5:       actor_jk ← Retrieve random actor from actors
6:        AV^t ← AV_j ∪ { name_k , actor_jk }
7       end if
```

if $name_k$ = task then
  $task_{jk}$ ← Retrieve random task from tasks
    AV^t ← $AV_j \cup \{ name_k , task_{jk} \}$
  end if
end for
  TS ← Read stem template from item rule r
  m_{jk} ← Format stem template passing TS and AV^t

Figure 5. Stem Creation Algorithm

The inputs to the "Answer options Preparation algorithm" shown in Figure 6 are concept node selected (c_t), item rule (r), item rule attributes (RA). When the algorithm is asked to prepare answer options it first initialized set $ANSWER_j$, KEY_j to empty set (at Line # 1).After that the rule attributes RA is enumerated (from Line # 2 to # 19) checking the type of item attribute name $name_k$ (from Line # 4 to Line # 18). If the attribute name is "concept node name" (as Line #5 shows), then it will retrieve the siblings (S^t) passing concept node c_t as parameter (at Line #7); then it will loop through the S^t and create a set of distractors $DISRACTOR_{jk}$ (from Lines #8 and #9). After that the concept node (c_t) name is added to the set KEY_j, it becomes the key of the item. If the case of attribute name being "other", the algorithm will retrieve the attribute value pair AV^t of c_t (at Line #13). After the attribute value of av_{ct} is retrieved from AV^t and added to KEY_j, it becomes the key of the answer options (at Line #15). In Line #16 the algorithm creates the distractor $DISRACTOR_{jk}$ by removing av_{ct} followed by adding the KEY_j and $DISRACTOR_{jk}$ to create the answer options $ANSWER_{jk}$ (at Line #17). Thereafter, the algorithm retrieves the answer option template (TO) from the rule object (r) (at Line #20) and format template TO passing $ANSWER_{jk}$ which returns formatted set of answers ANSWERj with the labels (e.g. **A**. Pop **B**. Push, **C**. Peek).

ALGORITHM 4 : Answer Options Preparation

Input : c_t , a concept node selected
$r \in R$, item rule.
RA , set of apply item rule attributes for answer option

Output : $ANSWER_j$, set answer for item
Local : $DISTRACTOR_{jk} \subset DISTRACTOR_j \leftarrow \{ \emptyset \}$, $ANSWER_{jk} \subset ANSWER_j \leftarrow \{ \emptyset \}$

1 : $ANSWER_j \leftarrow \{ \emptyset \}$, $KEY_j \leftarrow \{ \emptyset \}$
2 : for each rule attribute ra_k in RA
3 : $name_k$ ← Get rule attribute name from rule attribute passing ra_k
4 : case $name_k$ of
5 : concept node name:
6 $KEY_j \leftarrow KEY_j \cup \{ name(c_t) \}$
7 : S^t ← Retrieve the siblings of concept node passing c_t
8 : for each sibling s^t in S^t
9 : $DISTRACTOR_{jk} \leftarrow DISTRACTOR_{jk} \cup \{ s^t \}$
10 : end for

$ANSWER_{jk} \leftarrow KEY_j \cup DISTRACTOR_{jk}$
other:
$AV^t \leftarrow$ Retrieve attribute value pair from concept schema passing c_t, ra_k, $name_k$
$av_{ct} \leftarrow$ Get the attribute value of the concept node selected from AV^t by c_t
$KEY_j \leftarrow KEY_j \cup \{ av_{ct} \}$
$DISRACTOR_{jk} \leftarrow AV^t \setminus \{ av_{ct} \}$
$ANSWER_{jk} \leftarrow KEY_j \cup DISTRACTOR_{jk}$
end case
end for
$TO \leftarrow$ Retrieve answer options template from rule object r
$ANSWER_j \leftarrow$ Format answer options with labels passing TO , $ANSWER_{jk}$

Figure 6. Answer Options Preparation Algorithm

Table 3 show partial concept schema for concept node Stack, Queue, Deque and Linked-List. It is used in conjunction with the rule attribute (Table 2) when the algorithms are creating the stimulus, stem and answers. Using Figure 7 for example, at line # 3 if the attribute name from column #3 of Table 2 is retrieved e.g. “operation”, the attribute is sent to the Table 3, to select all attribute with the same name and their value i.e. attribute value pair (AV^t) for selected concept node c_t. If c_t = Stack and attribute name ($name_k$ = operation) then records with #2 -#5 will be selected.

Table 3. Partial Concept Schema for Data Structure knowledge map

#	Concept Node	Relation Name	Concept Name	Concept Action	Attribute Name	Attribute Value
1	Data Structure	has			performance measure	correctness
2	Stack	is	linear data structure			
3	Stack	has			description	Last in first out mechanism (LIFO)
4	Stack	has			operation	push
5	Stack	has			operation	pop
6	Stack	has			operation	peek
7	Stack	has			application	reverse string
8	Stack	has			application	implement undo operation of editor
9	Queue	is	linear data structure			
10	Queue	has			description	first in first out mechanism (FIFO)
11	Queue	has			operation	enqueue
12	Queue	has			operation	dequeue
13	Queue	has			operation	peek
14	Queue	has			application	simulating people waiting to get passport in first come first serve
15	Deque	is	linear data structure			
16	Deque	has			description	insertion and removal of elements at both end points
17	Deque	has			operation	addFirst
18	Deque	has			operation	addLast
19	Deque	has			operation	removeFirst
20	Deque	has			operation	removeLast
21	Linked-List	is	linear data structure			
22	Linked-List	has			description	does not need to know size in advance
23	Linked-List	has			operation	add
24	Linked-List	has			operation	remove
25	Linked-List	has			operation	contains
26	Linked-List	has			application	remaking the amazing race
27	Linked-List	has			application	Storing values in a hash table to prevent collisions

The lower and higher cognitive items shown in Appendix C and Table 4 both relates to knowledge map. Whiles the algorithm implemented uses the knowledge map to generate both of the items, the lower cognitive items are generated using the concept node selected from the knowledge map as the key of the item and its siblings as the distractors of the item. On the other hand, when generating higher cognitive items the algorithm utilizes the concept node, the concept schema of the knowledge map, and domain specific supporting dataset to create the items stimulus, stem and the answers options.

Table 4. Example items generated for cognitive type "Apply"

Cognitive Type	Stimulus	Stem	Answer Options
Apply	A programmer was asked to implement Stack data structure to be used for software component that simulate Last In First Out mechanism (LIFO).	Select the suitable functions the programmer has to implement	A. Push ,pop, peek B. peek ,enqueue, dequeue C. addFirst ,addLast, removeFirst D. add ,remove, contains
Apply	A student designed and implemented data structure for software module. Upon uniting testing it was found that the data structure exhibit Last In First Out mechanism (LIFO).	Choose the possible data structure the student implemented.	A. Queue B. Stack C. Deque D. Linked-List
Apply	A software developer was asked to implement software component leveraging stack data structure capable of reversing list of items	Select the set of methods the software developer has to implement to accomplish the task.	A. Push ,pop, peek B. peek ,enqueue, dequeue C. addFirst ,addLast, removeFirst D. add ,remove, contain
Apply	A computer science teacher was asked to implement Stack data structure with the following specification: (a) stratify correctness criteria (b) model undo operation of word editor	Select the appropriate functions the computer science teacher has to implement to satisfy the specification	A. Push ,pop, peek B. peek ,enqueue, dequeue C. addFirst ,addLast, removeFirst D. add ,remove, contains

Chapter IV – HIGHER COGNITIVE LEVEL GENERATION ENGINE

This chapter will give detail description of the implementation of the Item Generation Engine. First, the Online Test System Architecture will be illustrated and explained. This will be followed by the System Features for teachers and student.

4.1 Item Generation Engine Architecture

The system architecture of the item generation engine is shown in Figure 7. It consist of three layers namely presentation layer, application layer, and data layer.

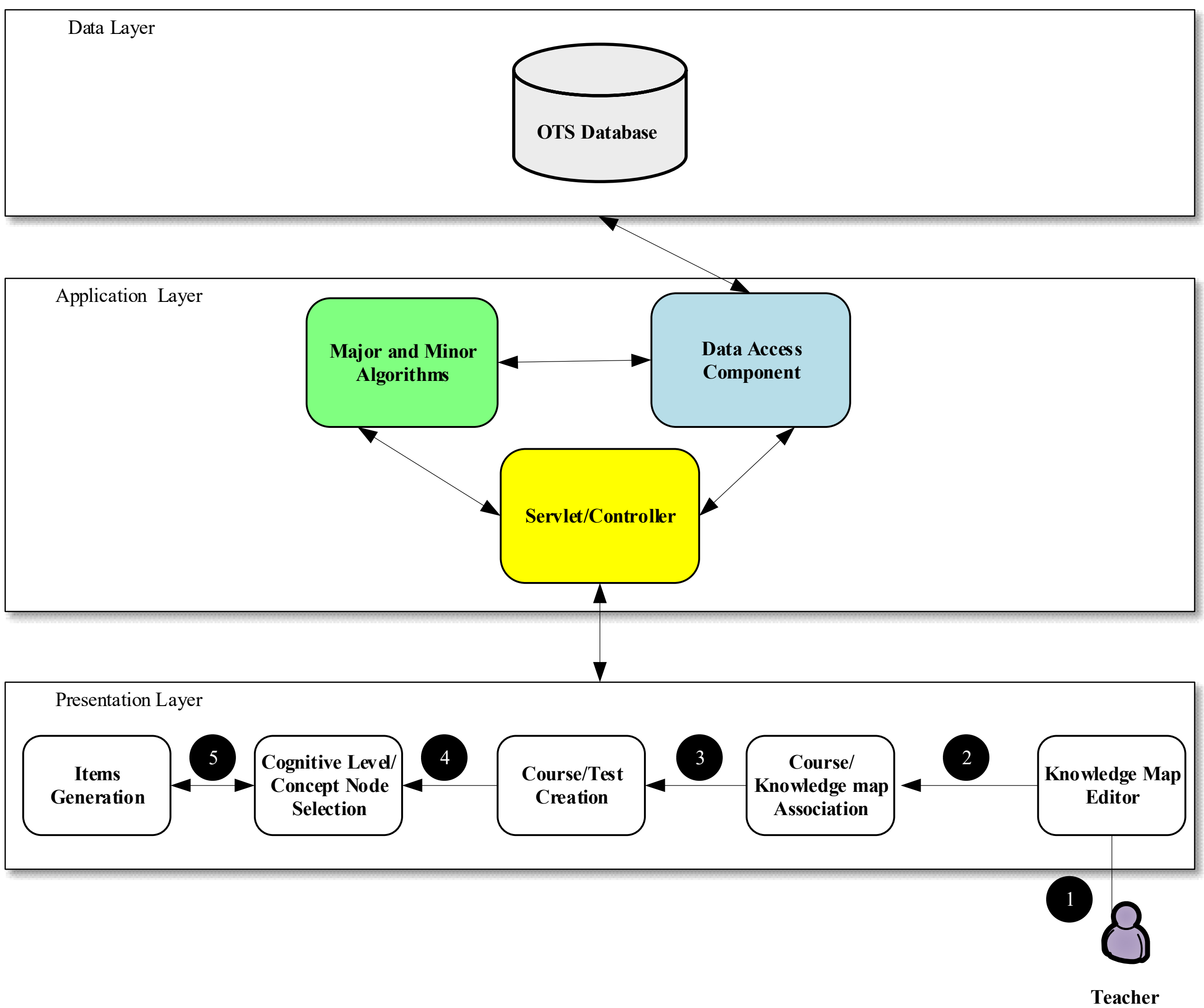


Figure 7. System Architecture of item Generation Engine

The presentation layer provides knowledge map editor for teachers to create, modify and delete knowledge maps and associated concept schemas. In addition, it gives teachers the ability to import knowledge maps from other teachers. Furthermore, there is user interface which allows teachers to select cognitive level and concept node for the systems to generate items for their chosen cognitive level. The application layer is where the servlet or controller is located to receive the teachers request to generate the items. Moreover, this layer has all the major and minor algorithms interacting with the data access components. The knowledge maps and the associate concept schemas together with item rules, item rule attributes, supporting dataset are stored in Online Test System (OTS) database hosted in the data layer.

As shown in figure 7, the item generation engine requires five steps to generate cognitive level items as shown in the presentation layer. First, teacher can use the knowledge map editor to create new knowledge maps (e.g. Data Structure) in their chosen domain such as computer science. Once the knowledge map is created, they can add, remove concept nodes to the concept hierarchy. Moreover they can add, edit and delete concept schemas for a concept node. The knowledge map editor has function that allows teachers to import knowledge maps from other teachers as well as shearing knowledge maps. Next, teachers can associated the knowledge map created or imported to associate with his or her course. After that, teachers can create a test for the course that has knowledge map association. Afterwards, if a teacher wants to the systems to generate items for him or her, he or she will select cognitive level and concept node from the concept hierarchy and submit for the systems to generate the items.

When a teacher submits the cognitive level and the concept node selected, it is sent as data to the servlet/controller (item generation servlet) as shown the application layer of Figure 7. The servlet interact with the major and minor algorithms which intend ask the data access component to select the appropriate item rules based on the cognitive level the teacher selected and any supporting data set to generate the items and stored in database hosted in the data layer. Once the major algorithm receives all the necessary dataset it generates the items and goes through the servlet again to return the items generated to the teacher.

4.2 System Features for Teachers

The Online Test System has a number of feature for teachers to request for an account, create and manage their courses, creating knowledge maps for the courses they have created, importing knowledge maps other teachers has created, creating accounts for students, creating and managing test for their course and reviewing students as shown on the menu on Figure 3 . The menu slides in and out so if the menu is close the teacher has to click on the menu icon on left hand corner of the page.

4.2.1 Requesting an Account as a Teacher

If a teacher wants to use Online Test System to generate items and manage their courses and tests, he or she has to browse to web site with URL http://onlinetest.is-very-good.org:443/OTS/index.jsp to request for an account. Once the teacher is on OTS web page as shown in Figure 7, he or she has to click on "Request Account" link located on the teacher panel on the right hand corner of the page. This action will redirect the teacher to the account request page as shown on Figure 2.

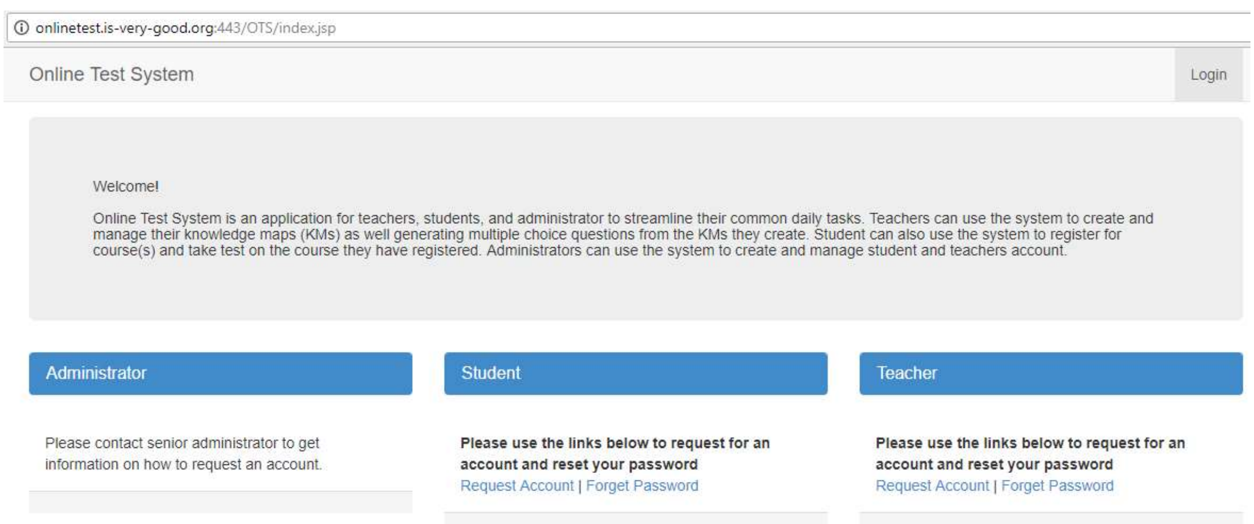


Figure 8. Online Test System web site

After the teacher has entered all the required fields namely first name, last name, login email, password, and retype their password as shown on Figure 8, he or she can then click on "Submit" button on the bottom right of the form . If the system creates teachers account successful it will redirect the teacher to a page to create and manage their courses.

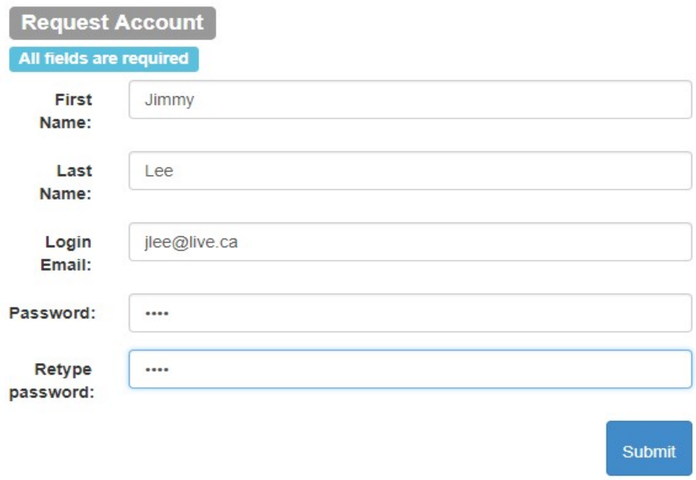


Figure 9. Teacher requesting an account.

4.2.2 Managing Courses

A teacher can see courses they have created by clicking on the Courses link on the menu. If the teacher has not yet created any course he or she can use new course creation form on the left hand side of the page labelled "Add New Course" to create new course as shown in Figure 9. First, the teacher has to enter the course name for example "Computer Science" then click on "Save" button on the left hand corner of the form.

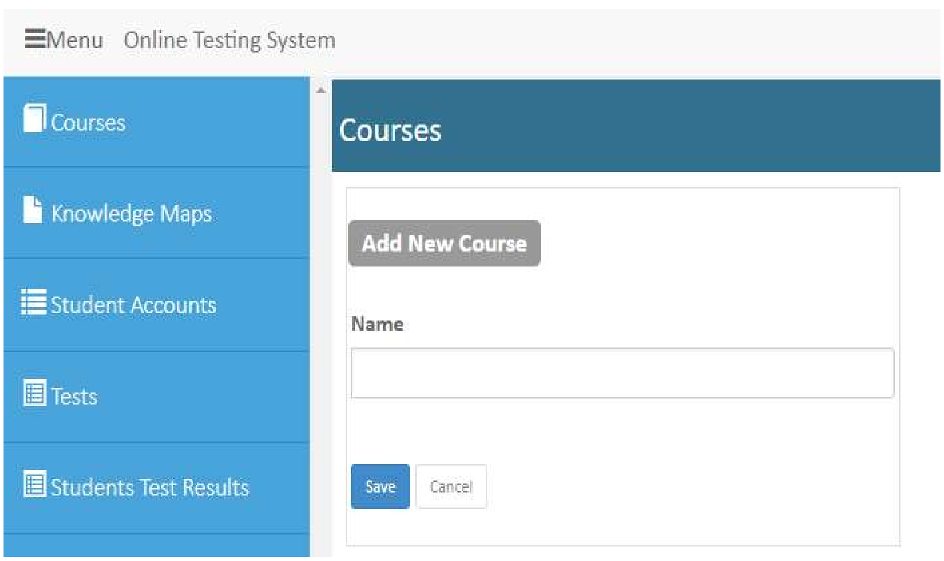


Figure 10. Menu Items available for teachers

Once the course is created successfully it will be added to course list on the right hand side of the form as shown on Figure 10.

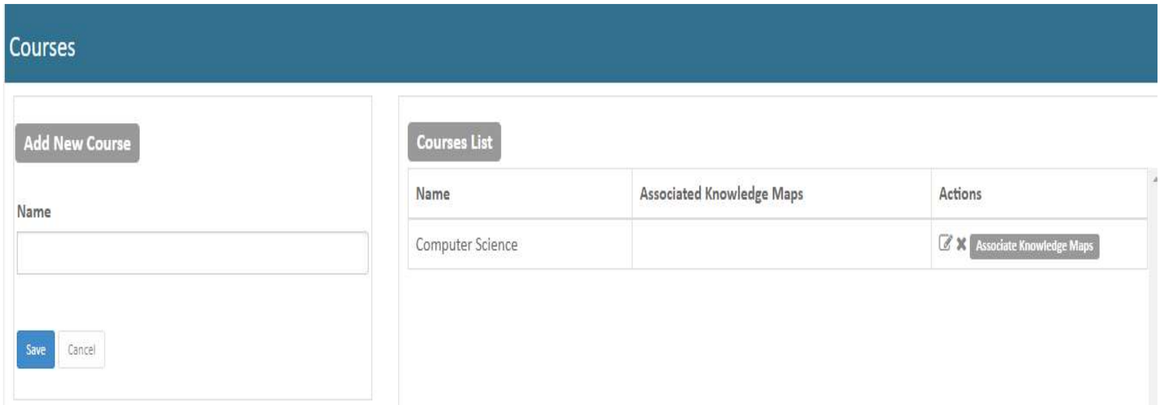


Figure 11. Teacher created computer science course

If the teacher wishes to edit the name of the course he or she has to click on the edit icon on last column of the course list for a particular course. The course name will be displayed on the text box on the right hand side of the page as shown on Figure 11. The teacher can change the name of course and then click on the "Save" button on the bottom left of the form. The teacher can also cancel editing the course name by clicking on the "Cancel" button on the bottom left of the form.

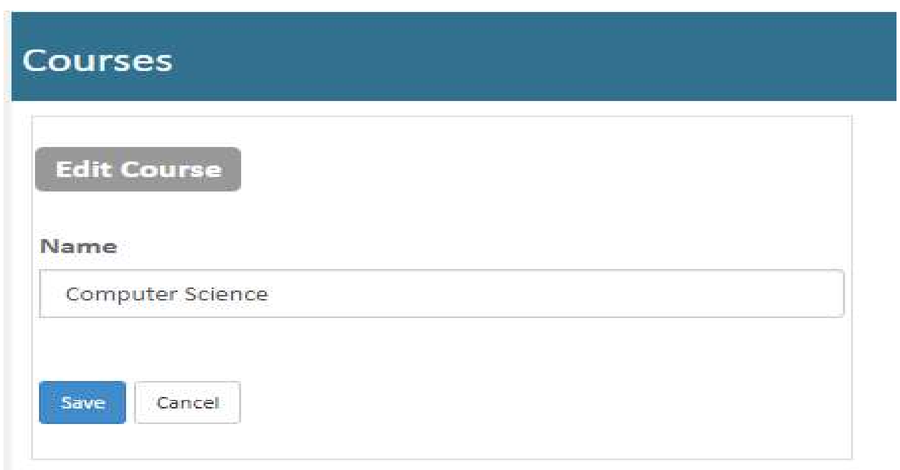


Figure 12.Teacher editing course name

4.2.3 Managing Knowledge Maps

If a teacher wishes to see the list of knowledge map other teachers are shearing by clicking on the "Knowledge Maps" option on the left hand side of the menu to display a list of knowledge maps with icons indicating sharing. The teacher can create new knowledge map by typing the name for example "Data Structure" and the description on the form labelled the "Add New Knowledge" and then click on Save button on the bottom left of the form as shown in Figure 12. Upon successful creation the knowledge map it will be added to "Knowledge Map List" on the right hand side of the page. If the teacher wants to cancel the creation of the knowledge map he or she can click on the "Cancel" button on the button left hand side of the form.

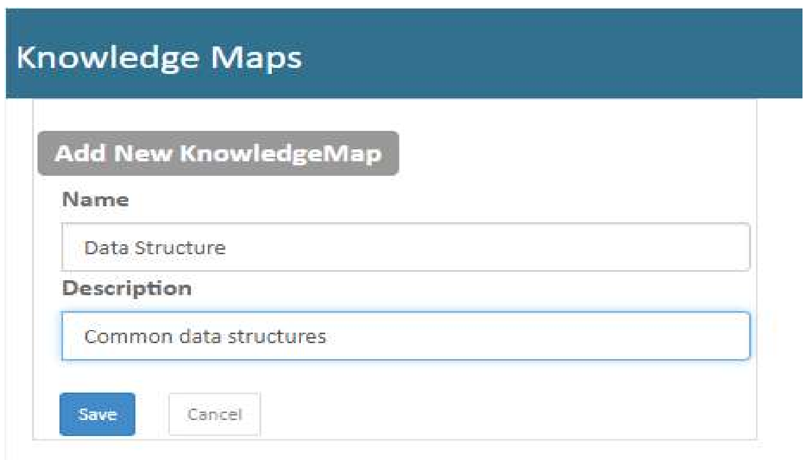


Figure 13. Teacher created new knowledge map "Data Structure"

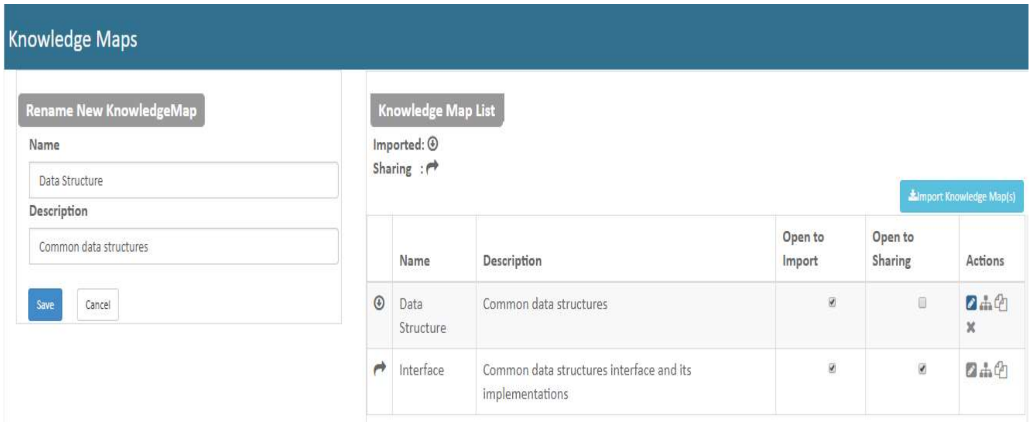


Figure 14. Teacher editing knowledge map "Data Structure"

If the teacher want to rename the knowledge map he or she has to click on the on the pencil icon on the row which will display the knowledge map name and description on the respective input box on form on the left hand side of the page. The teacher can then change the name and description and click on "Save" for the knowledge maps to be updated. On the other hand, if the teacher wishes to cancel remaining the knowledge he or she can click on "Cancel".

The teacher can also delete a knowledge map by clicking on the times (x) icon on the row he or she want to delete. Furthermore, the teacher can duplicate any knowledge map by clicking on the third icon from the left in any row of the knowledge map list.

Teachers can also obtain knowledge map from other teachers that has been marked as sharing. These knowledge maps will be displayed automatically on the teachers' knowledge map list. An example of knowledge map being shared is "Interface" knowledge map as shown in Figure 13.

4.2.4 Importing Knowledge Maps

Teachers have the choice to import knowledge maps that has been created by other teachers by clicking on the "Import Knowledge Map(s)" as shown in Figure 13. This will display a list of knowledge maps from which the teacher can check to import. In this case the teacher has checked on one item to be imported (on Figure 14). When the teacher clicks on the "Import Selected Knowledge Map(s) the knowledge map the teacher checked will be added teachers knowledge map list.

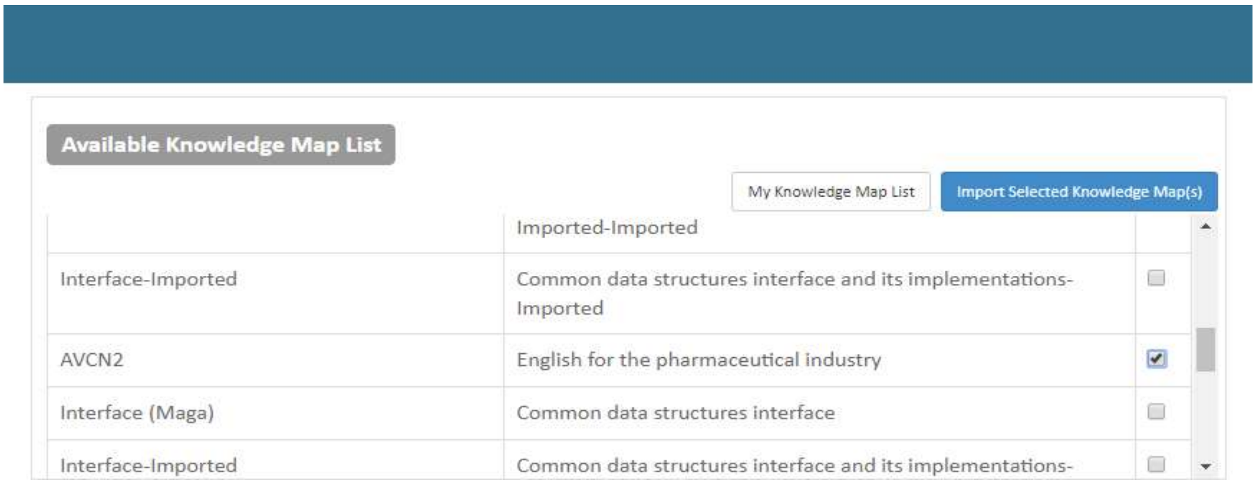


Figure 15. Teacher importing knowledge map from other teachers

4.2.4 Managing Concept Node and Concept Schema

Once a teacher has created knowledge map he or she can click on the tree icon on any row in the knowledge map list. This will show a view to create and manage the concept nodes and concept schemas of the knowledge map. If a teacher wants to add a concept node to the knowledge map "Data Structure" he or she has to select the node first, then type the name of concept node in the input box on top the concept hierarchy then click on "Add". As shown on Figure 15. The teacher has added Concept nodes Stack, Queue and Deque to Array-Based List.

The teacher can also remove concept node by selecting the concept node then click on the "Remove". When a teacher select a concept node for example "Stack" on the concept hierarchy on the left hand side of the view, the system will display the node information such as the node parent and the relation type with its parent on the right hand side of the view. The teacher can add concept schema to concept node such as Stack by clicking on the "Add New Concept Schema". This will show a concept schema form as shown on Figure 16. The teacher has to select "Relation name" from the drop down list on the left hand side of the form and enter the concept name if the relation name is "IS" then click on the "Save" button on the bottom left of the form to add the concept schema to the concept node. If the teacher want to add additional concept schema to the concept node he or she can select different or the same relation name and enter the associated attributes and click on "Save". The concept node will be added to concept schema list of the concept node as shown on Figure 15. Alternatively, if the teachers do not want to add concept schema to the concept node he or she can click on Cancel button at bottom left of the form.

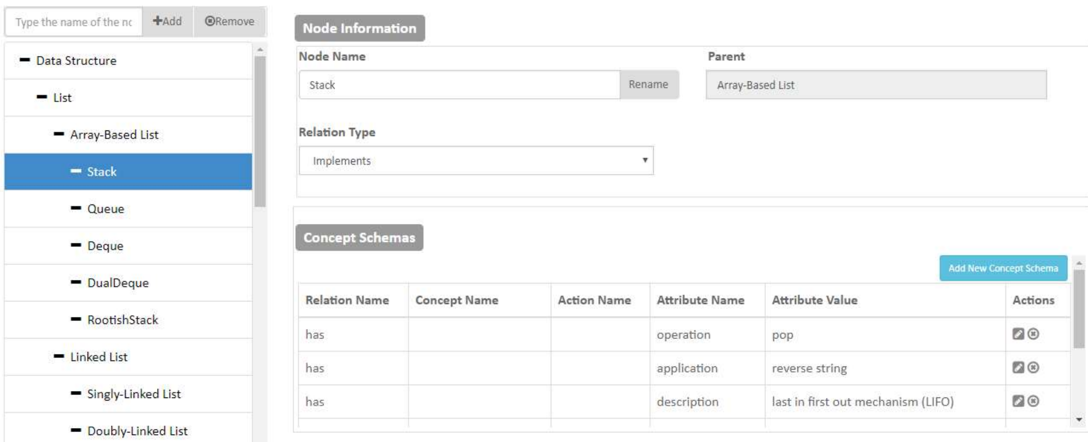


Figure 16. Teacher adding concepts (e.g. Stack) to knowledge map

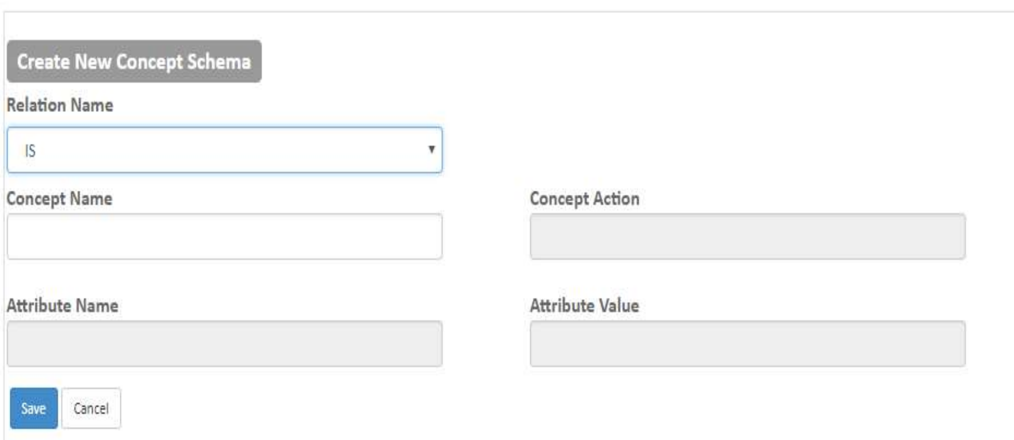


Figure 17. Teacher adding new concept schema

4.2.5 Associating Knowledge Maps to Course

Once the teacher has created knowledge map he or she can now go back to the list of courses by clicking the "Menu" icon on the left hand corner of the page and then click on courses option of the menu items. This will show all the list of the previously created courses. For each of the courses the teacher can associate any number of knowledge map. To do that the teacher can click on “Associate knowledge Maps” in the last column labelled “Actions” of the courses list for any of the course(s). For instance the teacher clicked on the “Associate Knowledge Maps” for computer science hence the course name (e.g. Computer Science) will be displayed on form at the left hand side of the page as shown in Figure 17. After that the teacher can select knowledge map previously created by himself or herself or the knowledge map being shared by other teachers from the dropdown list labelled “Select Knowledge Maps” on the left hand side of the form. For example the teacher has selected Data Structures knowledge map to be associated with Computer Science Course. Finally, the teacher can now click on the “Save” button on the left hand of the form for the system to associate the Data Structure knowledge map to the computer Science Course. If the association is done successfully the knowledge map the teacher selected will be shown in the second column (Associated Knowledge Maps) of the course list as shown in Figure 18.

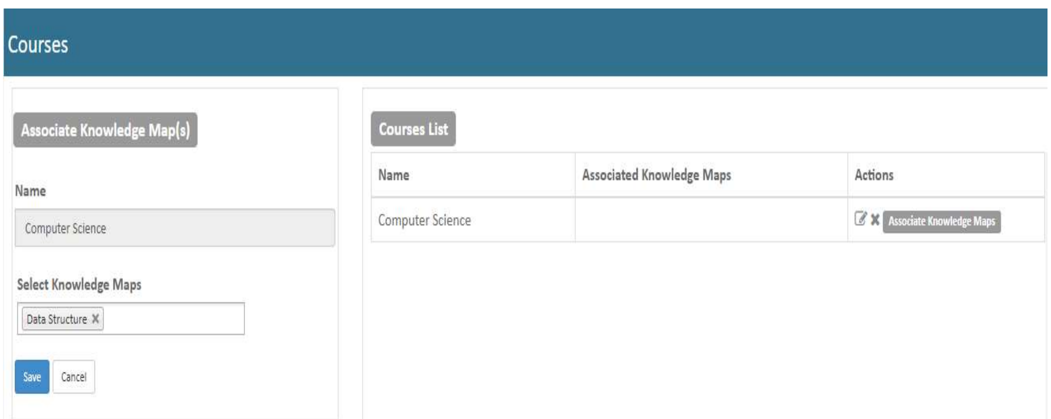


Figure 18. Teacher associating knowledge map(s) to his/her courses

Figure 19. Teacher associated Data Structure knowledge map to computer science course

4.2.6 Managing Course Test

The teacher can see list of test that has been created for his or her course by clicking on the “Tests” option on Menu as shown on Figure 20. If a teacher wishes to create a new test for his or her course, First he or she has to select the course from the dropdown list on top of the Test list on the right hand side of the form. For example the teacher has selected Computer Science, then he or she has to enter all the required fields for creating test (i.e. Name, Start Date, End Time, and Start Date) as shown in figure 19. After that the teacher has to click on Save for the system to create a test for the course.

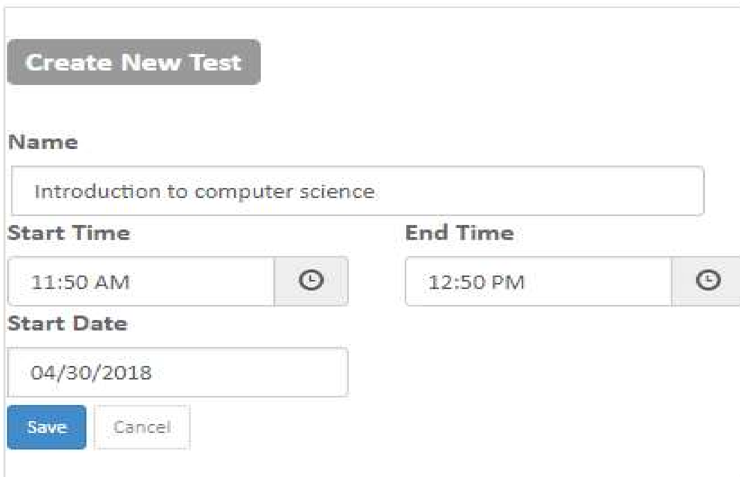


Figure 20. Teacher created one Test “Introduction to computer science” for Computer Science

If the course is created successfully it will be added to the test list of the course as shown on Figure 20. For each test for a course the teacher can “Edit” by clicking on the edit icon to display the test information on the form at the left hand side of the page. The teacher can then make the necessary changes to the test information and click on “Save” button for the system to make changes to the test information. The teacher can also delete the test by clicking the delete icon (x) if the test is not activated for student to take the test. For each test the teacher can activate it for the student to take the test by clicking on the “up arrow icon” in the last column of the test list. If the teachers do not want the student to take the test he or she can “De-Active” the test by clicking the down arrow icon on the “Actions” column.

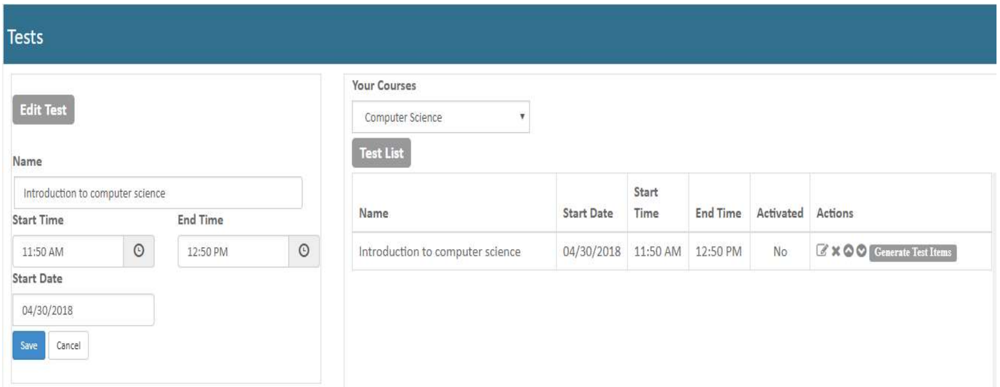


Figure 21. Teacher created Introduction "Introduction to computer science" test

Finally the teacher can ask the system to generate items for him or her by clicking on "Generate Test Items" for the test. This will show Item generation editor as shown on Figure 21. First, the teacher has to select cognitive type (e.g. Apply) from the dropdown list on the left hand side of the form and Concept node (e.g. Stack) from the concept hierarchy. Next the teacher can click on the "Generate" for the system to generate Apply items. Finally, the items generated will be added to the test question bank and also displayed on the right hand side of the form on the "Item generation" Tab.

4.2.7 Generating Test Items

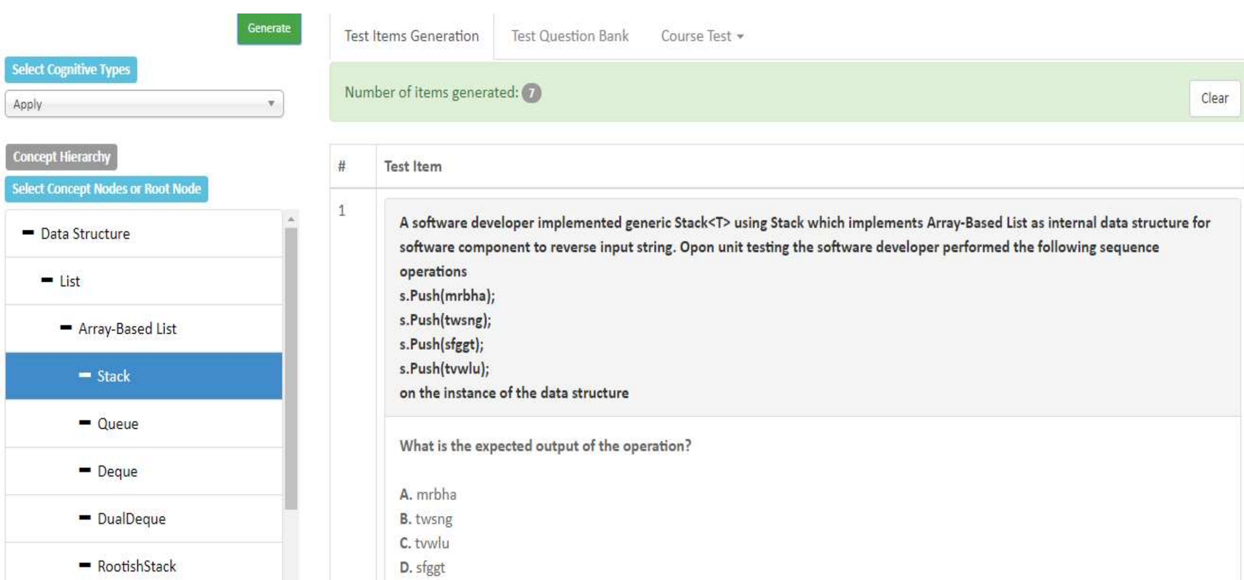


Figure 22.Teacher generated test items for concept node "Stack" and "Apply Cognitive Type"

4.2.8 Preparing Test Sheet

The teacher can click on the "Test Question Bank" to see all the items that has been generated for a particular test. Next, he or she can select any row by checking the check box and then click on the "Add To Test Sheet" for the system to add the items to the test sheet as shown on Figure 22. As shown in Figure 23 the system adds the items to the test sheet and at the same time it also create answer sheet for the test as shown in Figure 24.

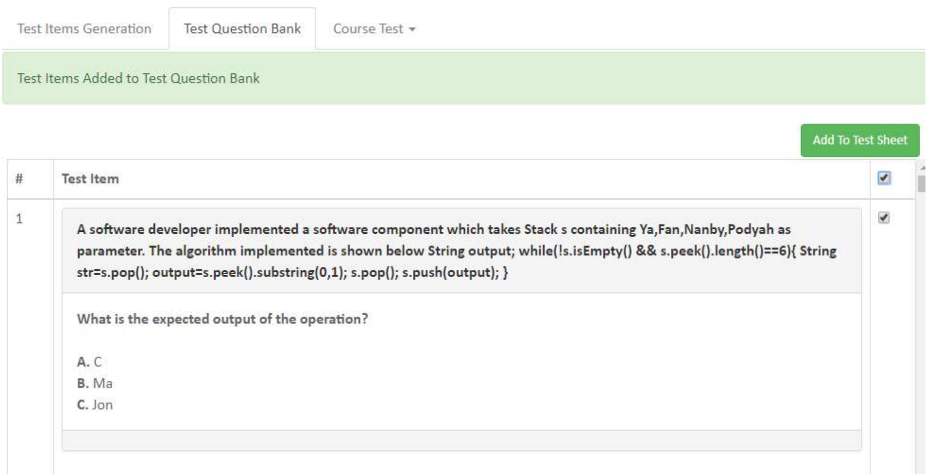


Figure 23. Teacher selected Items to be added to the Test Sheet

If the teacher wants to remove items from the test sheet he or she can select the item by checking the check box of the item and click on "Remove Selected Test Items" on the top right hand corner of the form (Figure 23). Once the items are removed the system automatically regenerates the answer sheet to reflect the item items on the test sheet.

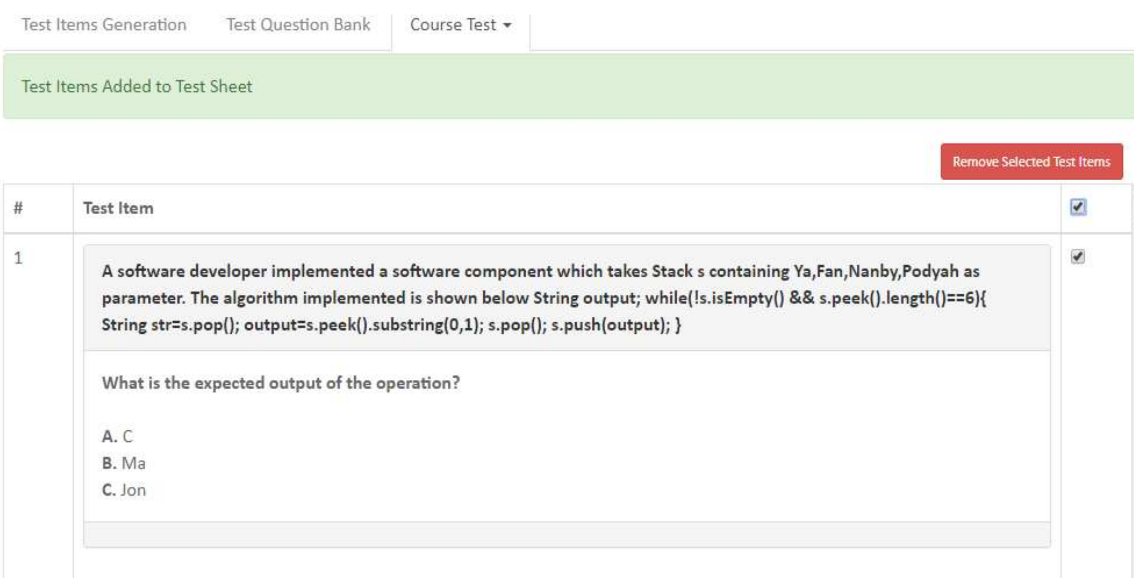


Figure 24. Teacher added items to course (Introduction to computer science) Test Sheet

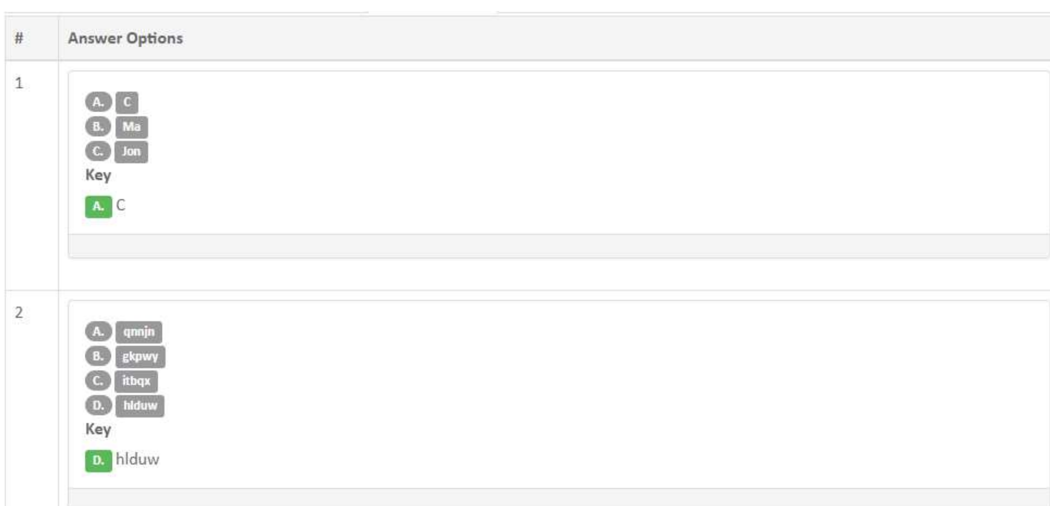


Figure 25. Answer Sheet for Introduction to computer science Test

4.3 System Features for Students

There are features in Online Test System which allow students to register and unregister for courses, view their pending tests and take a test for the courses they have registered. In addition there is a functionality for the students to view the result of the test they have already taken as shown in Figure 25.

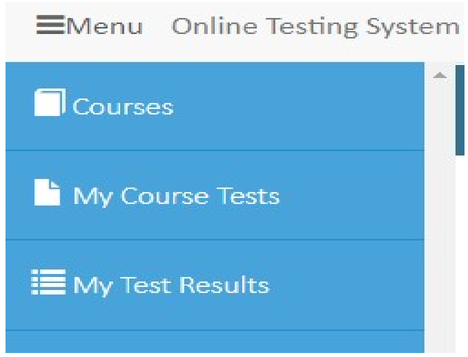
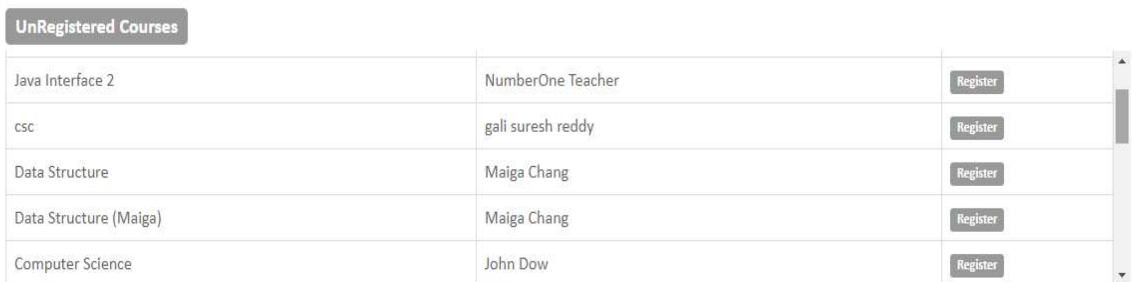


Figure 26. System features available for students

Students can see the list of all courses by clicking on the menu icon on the top left corner as show in Figure 26 and then click on the Courses menu item option.

4.3.1 Registering and Unregistering for a Course

This will display a list of all the courses teachers have created. For a student to register for a course he or she has look for the course on the courses list and click on the corresponding “Register” button as shown in Figure 27.


UnRegistered Courses

Java Interface 2	NumberOne Teacher	Register
csc	gali suresh reddy	Register
Data Structure	Maiga Chang	Register
Data Structure (Maiga)	Maiga Chang	Register
Computer Science	John Dow	Register

Figure 27. Available Course for students to register

Once the student course is registered it will be displayed on the Registered Course list as shown in Figure 27. For example the student registered for Computer Science Course. If the student decides not to take the course he or she can click on the Unregister button of the course. This action will remove the course from the students registered courses.


Registered Courses

Course Name	Professor	Actions
Computer Science	John Dow	UnRegister

Figure 28. Course(s) student have registered

4.3.2 Viewing Pending Course and Taking a Test

If the teacher for the course has not yet activate the test then the students will not see any tests on their “My Course Tests” list. Figure 28 show that the teacher for the Computer Science Course has activated the course hence all the students that registered for this course will see the test. Once a student decides to take the test he or she can click on the "Take Test" in the action column of the course.

Figure 29. Test pending for student to take

This action will display test sheet as shown in Figure 29. As the student chooses an answer for a test item it will be highlighted in green. Once a student is satisfy with his or her the answers he or she can click on the "Submit" but on the right hand side below the test information of the test sheet.

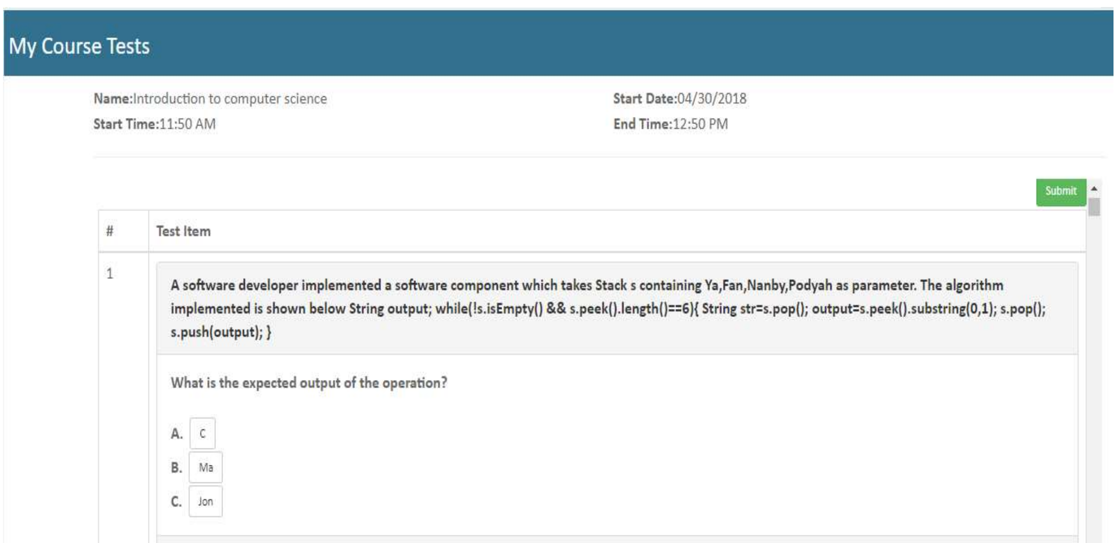


Figure 30. Student taking introduction to computer science test

4.3.3 Viewing Test Results for Courses

After successful submission of student test the system will remove the test the student took from his registered courses. Students can see the results of all the test they have taken by clicking on the menu on the left hand corner of the page and click on the "My Test Result" menu item option. This will display a list of all the test he or she has taken for a particular course as shown in Figure 31.

Figure 31. Student viewing their these results

Chapter V – EVALUATION AND DISCUSSION

The primary goal of this study was to answer the research question that relate to agreement between the items classified into lower and higher cognitive by the system (item generation engine designed and implemented for this research) and teachers. The methodology used to test the research question is presented in this chapter. The chapter is organized into three sections: (1) Research Question (2) Evaluation Plan; (3) Data Analysis and results for this study

5.1 Research Question(s)

The current research was designed to answer one research question: *Will there be agreement between the items classified as lower or higher cognitive item by the system and teachers?*

In order to answer the research questions an evaluation plan was designed which consist of experimental design which includes participant selection, data collection, data analysis and results.

First of all, teachers from universities and colleges were invited to use the Online Test System for which the item generation engine designed and implemented has been integrated. There were two female teachers and two male teachers. In total there were four participants of which two teaches Data Structures course, one teacher teaches Computer Engineering and the other teacher teaches Computer language.

The teachers were given a questionnaire which has two sections. Section one collects information about the course the teachers teaches and the gender. The second section consist of 40 items (Appendix C) generated by the algorithms designed and implemented for this study. Then the teachers were requested to classify the items in the section two of the questionnaire into appropriate cognitive level (Remember, Understand, Apply, Analyze, and Evaluate) based on Bloom Taxonomy.

Finally, statistical analysis was performed on the data collected from participants. Inter-rater reliability was determined using the scores generated by the participants. IBM SPSS Statistics version 25 was used to obtain these values. A measure (Kappa value) in the range of 0.61-0.80 or greater (Appendix B) with alpha of 0.00 was selected as the criteria to judge whether there is substantial agreement between the cognitive items classified by the system and participants.

5.2 Data Analysis and Results

This study intended to investigate whether the will be agreement on the items classified by the item generation engine developed and that of the participants. The research question under consideration is as follows:

R1: Will there be agreement between systems classified items and teachers?

Descriptive statistics was used to investigate the research questions. To investigate the research question two cases of Cohen's Kappa Analysis were performed. Case one involves comparing the items classified as High (Apply, Analyze, Evaluate) and Low (Remember, Understand) cognitive levels by the system and that of the participants. The results of the analysis are shown in Tables 5 to 11. In all the analysis cases the number items (N=40) and all the participants classified all the items as shown in Table 5. Table 5 show the summary of the system and participants Low/High classification of the items. As show in the table both the system and the participants classified all the items (N=40, 100%).

Table 5. System and Participant Low/High Classification Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
System Scale * Participant 4	40	100.0%	0	0.0%	40	100.0%
System Scale * Participant 5	40	100.0%	0	0.0%	40	100.0%
System Scale * Participant 9	40	100.0%	0	0.0%	40	100.0%
System Scale * Participant 14	40	100.0%	0	0.0%	40	100.0%

Table 6 depicts the Kappa value and the associate p value for each of the participant that categorized the items into high and low cognitive items. From the table it is seen that the participant who is teaching computer language shows almost perfect agreement (Kappa value=.827, p=.000) the system classification and

Computer Engineering teacher had substantial agreement (Kappa value=.0701, p=.000). On the other hand the Data Structure teacher recorded Fair Agreement in the Kappa range between (.451 and .419) and p value between .001 and .004 respectively with the system.

Table 6. Participant High/Low classification, Kappa Value and p value

Participants										
#	System		4		5		9		14	
Teaching Subject	None		Data Structure		Computer Engineering		Computer Language		Data Structure	
Kappa Value	1		.419		.701		.827		.451	
p Value	.000		.001		0.000		.000		.004	
Cognitive Level	High	Low	High	Low	High	Low	High	Low	High	Low
	26	14	9	5	5	9	3	11	5	9
Total	40		14		14		14		14	

In the second case the items are categorized into cognitive level and kappa analysis was run to the level of agreement between the system categorization and that of the participants.

Figure 7 reports system and participant classification for Remember cognitive level. While there was substantial agreement between the Computer Engineering teacher and the systems classification (k=.895, p= .000) Teacher #4 and 5 achieved slightly agreement (k=0.000, p=0.000). Teacher # 14 had “Fair Agreement” (k=.459, 0.01)

Table 7. System and Participants Remember classification

Participants						
#	System		4	5	9	14
Teaching Subject	None		Data Structure	Computer Engineering	Computer Language	Data Structure
Cognitive Level	**Remember**	6	0	0	5	2
	Not Remember	34	40	40	35	38
	Total	40	40	40	40	40
Kappa value	1		0.000	0.000	.895	.459
p value			-	-	.000	0.01

Figure 8 depicts system and participant classification for Remember cognitive level. Cohen's k was run to determine to determine if there was agreement between the system and participants classification of Understand cognitive level items. Teacher #5 and #9 had Substantial Agreement (k=.727, p= 0.00) whiles teacher #14 had Slight Agreement (k=.186, 0.43) and Teacher #14 had Poor Agreement (k <0, p=1.00)

Table 8. System and Participants Understand classification

Participants						
#	System		4	5	9	14
Teaching Subject	None		Data Structure	Computer Engineering	Computer Language	Data Structure
Cognitive Level	**Understand**	8	0	5	5	1
	Not Understand	32	40	35	35	39
	Total	40	40	40	40	40
Kappa value	1		.000	.727	.727	.186
p value	0.000		1.000	.000	0.00	0.43

The system and participants classification for Apply items and the associates kappa value and p values is shown in Figure 9. There was Substantial Agreement (k=.848, p=0.00) between the items classified by teacher # 4. Also there was Substantial Agreement (k=.746, p=0.00) between the items classified by teacher # 5. Teacher # 9 and #14 had Fair Agreement (k=.219, p=.027) and (k=.433, p=0.001) respectively

Table 9. System and Participants Apply classification

Participants						
#	System		4	5	9	14
Teaching Subject	None		Data Structure	Computer Engineering	Computer Language	Data Structure
Cognitive Level	**Apply**	19	16	14	4	8
	Not Apply	21	24	26	36	32
	Total	40	40	40	40	40
Kappa value	1		.848	.746	.219	.433
p value	.000		.000	.000	.027	0.001

Table 10 shows systems and participants items classification for Analyze cognitive level. Teacher #9 and #14 had Substantial Agreement with the items classified by the system with (k =.724, p=0.00) and (k=.875, p=.000) respectively. On the other hand, Teacher #4 recorded Fair Agreement (k=.219, p=.027) and teacher #5 had Poor Agreement (k=.000, p=0.000).

Table 10. System and Participants Analyze classification

Participants						
#	System		4	5	9	14
Teaching Subject	None		Data Structure	Computer Engineering	Computer Language	Data Structure
Cognitive Level	**Analyze**	5	1	0	3	4
	Not Analyze	35	39	40	37	36
	Total	40	40	40	40	40
Kappa value	1		.304	.000	.724	.875
p value	0.000		.007	.000	.000	.000

Table 11 presents the result of system and participant items classification for Evaluate cognitive level. Teacher #1 had perfect agreement (k=1, p=0.00) whereas Teacher #9 had Substantial Agreement (k=.655, p=0.00) with the system. On the other hand, teacher #5 and #14 had Poor Agreement (k=0.00, p=0.00) with the system.

Table 11. System and Participants Evaluate classification

			Participants			
#	System		4	5	9	14
Teaching Subject	None		Data Structure	Computer Engineering	Computer Language	Data Structure
Cognitive Level	**Evaluate**	2	2	0	1	0
	Not Evaluate	38	38	40	39	40
	Total	40	40	40	40	40
Kappa value	1		1	0.00	.655	.000
p value	0.00		.000	1.00	.000	.000

5.3 Discussion of Findings

This section discusses, analyze, and explain the findings of the each research questions. First the research questions will be restated and the findings will follow.

1. *Will there be agreement between the items classified by the system and that of all the participants*

The first findings resulting from research question one indicates that Data Structures teachers disagree with the items the system classified as High (Apply, Analyze, Evaluate) and Low (Remember, Understand) cognitive item. On the other hand, the teacher who teaches Computer Engineering and Computer Language agrees with the items classified by the system.

For the Remember cognitive items the findings was that one of the Data Structure teachers and the Computer Engineering Teacher did not agree with the Remember cognitive items classified by the system. While, the Data Structure slightly agreed with the systems classification the Computer Engineering teacher fully agreed with the items the system classified as Remember.

In the case of Understand cognitive items, the finding found was that both the Computer Engineering and the Computer Language teacher classified the items as Understand which matches the system classification whereas one of the Data Structures teachers disagree or slightly agreed and other Data Structure teacher disagree with the system classification.

Considering the Apply cognitive items the findings was that one Data Structure teacher and the Computer Engineering teacher fully agreed with the systems classification but the other Data Structure teacher disagree with the system classification of the Apply items.

For the Analyze items the Computer Engineering teacher completely disagree with the system classification, on the other hand, the Computer Language teacher and Data Structure teacher agrees with the system classification whereas the other Data Structure teacher disagree with the system classification.

The final findings for this research questions was in relation to the Evaluate cognitive items. While one of the data structure teacher and computer language teacher agrees with the system classification, the Computer Engineering teacher and the other Data Structure teacher disagree with the system classification.

In conclusion, there was not a case where all the participants agreed with the system classification of the cognitive items. But in most of the test cases the computer Engineering and the Computer Programming teacher tends to agree with the system classification of the cognitive items. One possible explanation might be that the teachers have not applied Blooms Taxonomy when preparing their items. In addition, it might be that the teachers are teaching different courses that might have different interpretation of the cognitive levels (Remember, Understand, Apply, Analyze, and Evaluate).

Chapter VI – CONCLUSION

In the preceding chapter the result of the data analysis was presented. This chapter is consist of summary of the study, limitations of this study, recommendations for further research, and conclusions.

6.1 Summary of Study

The purpose of this study was to evaluate if there was agreement between the cognitive items classified by the algorithms designed and developed and that of the participants.

The research included male and female participants. A total of four participants of which two of them are teachers teaching Data Structures and one teacher teach Computer Engineering and the other participant teaches Computer Language. The participants were asked to classify forty items into respective cognitive level (Remember, Understand, Apply, Analyze, and Evaluate). The data collected was analyzed to measure the agreement between the system classification of the forty items and that of the teachers. The study include the following research questions

1. Will there be agreement between the items classified by the system and that of all the participants?

Research Question 1 was answered using inter-rater analysis was performed comparing the kappa value and the p value of the system to that of the participants.

6.2 Limitations

The goal of this study was to investigate whether there will be agreement between the systems and the participants cognitive items classification. Data was collected to test the research question relating to this goal. The information was analyzed and although there were significant findings, the research has also some limitations.

One of the limitation of this study was that there weren't inadequate participants. This research considered teachers teaching Data Structures, Computer Engineering, and Computer Languages. Moreover, the forty questions the participants classified did not have equal number of items for (Remember, Understand, Apply, Analyze, Evaluate) cognitive levels. This was due to the fact that the some algorithms designed and implemented for this research did not output good items to be included in the items. For example there was only two evaluate questions that was included in the in the forty questions. Furthermore, due to the limited data size the initial research model and hypothesis stated for this research could not be tested hence removed from the thesis.

The original intent of this research was to perform quantitative statistical analysis to evaluate the usability of the system and the effectiveness of the items generated, however since the size of the data collected was too small to do such analysis , this research performed interpretative descriptive analysis instead. Furthermore, the current analysis was done using the small dataset collection which may affect the result of this research.

6.3 Future Work

A study of this nature could go a long way to help teachers to cut down the amount of time and effort they spend to prepare tests for assessing students. Students can also benefit from the system by doing self-assessment at anytime and anywhere.

Future research into this subject should include repeating the experiment with large group of participants with teachers from Science, Technology, Engineering and Mathematics and also deferent domain like chemistry, biology. This will help to acquire more data to do analysis to test all the initial hypothesis stated for this research. In addition, it will be beneficial if the participants have brief tutorials on Blooms Taxonomy before they classify the 40 items into their respective cognitive level. The idea is to bring the participants up to speed with cognitive levels and how it is applied to items

Another avenue of this research could be to improve the algorithms such as the “Evaluate Item Generation” algorithm to output items that can be added to the forty items to measure the agreement between the items classify by the system and the participants. For example with the Evaluate Generation algorithm when the attribute name of the concept schema is “Example” and the attribute value is “Number” the algorithm is able to generate the item key and distractors by manipulating the key. For instance it can have multipliers of the key to generate the distractors. However, when the attribute value is “String” the algorithm is not able to generate key and distractors that make sense in context of the item stimulus and stem.

One possible solution to integrate Natural Language Processing or Ontology which the algorithm can use to deduce the meaning of the string.

Another possible future work is integration of Big data analytics. Currently, the Online Test System (OTS) for this research has features for students. Students have the ability to take a test for a course they have registered. One possible feature work for the OTS is to apply Big data analysis on the students’ data that will be collected with the system. This analysis will allow teachers to measure, monitor and respond in real time student’s understanding of material, as well as personalized student learning experience.

REFERENCES

Al-Rukban, M. O. (2006). GUIDELINES FOR THE CONSTRUCTION OF MULTIPLE CHOICE QUESTIONS TESTS. Journal of Family & Community Medicine, 13(3), 125–133. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3410060/

Alves, C. B., Gierl, M. J., & Hollis, L. (2010). Using automated item generation to promote principled test design and development. Paper presented at the annual meeting of the American Educational Research Association, (AERA 2010), Denver, Colorado, USA, April 30 – May 4, 2010. Retrieved April 4, 2016, from http://www.crame.ualberta.ca/files/AERA%202010%20Denver%20Task%20Model%20AIG.pdf

Albion, P. (1999) Heuristic evaluation of educational multimedia: From theory to practice. In: ASCILITE 1999: 16th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education: Responding to Diversity, 5 - 8 Dec 1999, Brisbane, Australia.

Al-Adwan, A., Al-Adwan, A., & Smedley, J. (2013). Exploring students' acceptance of e-learning using Technology Acceptance, Model in Jordanian universities. International .Journal of Education and Development using Information and Communication Technology, (IJEDICT), 9(2), 4-18.

Bevan, N. and Macleod, M., (1994) Usability measurement in context, Behaviour and Information Technology, 13, 132-145.

Bloom, B.S. and Krathwohl, D. R. (1956). Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain. New York: Longmans, Green and Co.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In the Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, October 6-8 2005, 819-826.

Byrne, G. (2007) .A statistical primer: Understanding descriptive and inferential statistics. Evidence Based Library and Information Practice, 2(1), 32-47.

Bharati, P., & Chaudhury, A. (2004). An empirical investigation of decision-making satisfaction in web-based decision support systems, Decision Support Systems. 37 (2), 187–197.

Corbett, A. & Mostow, J. (2008). Automating comprehension questions: lessons from a reading tutor. In the Proceedings of the 1st workshop on the Question Generation Shared Task and Evaluation Challenge. Arlington, Virginia, USA, September 25-26, 2008, 318-320. Retrieve April 4, 2016, from https://www.cs.cmu.edu/~listen/pdfs/3-CorbettEtAl-QG08.pdf

Chang, M., Kuo, R., Chen, S., Liu, T., & Heh, J. (2008). Developing True/False Test Sheet Generating System with Diagnosing Basic Cognitive Ability. In the Proceedings of World Conference on Educational Media and Technology, Vienna, Austria, June 30-July 4, 2008, 5740-5748.

Chang, M., & Kuo, R. (2009). Elementary Level Botanical Item Generation. Learning Technology. Newsletter, 11(3), 7-9 7-9. Retrieved April 4, 2016, from http://www.ieeetclt.org/issues/july2009/lt_july2009.pdf

Chandrasekaran, B., Josephson, J. R., Benjamins, R. V (1999). What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, 14(1), 20–26.

Carmines, E. G., Zeller, R. A. (1979). Reliability and Validity Assessment. Beverly Hills, CA: Sage.

Cicek, M., & Ozcan, S. (2013). Examining the demographic features of Turkish social media users and their attitudes towards social media tools. Control, Decision and Information Technologies (CoDIT), 2013 International Conference on, Hammamet, Tunis, Tunisia, 6-8 May, 2013, 511-515. doi:10.1109/CoDIT.2013.6689597.

Chang, S. C., & Tung, F. C. (2008). An empirical investigation of students' behavioral intentions to use the online learning course websites. British Journal of Educational Technology, 39(1), 71-83.

Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: theory and measures, Software Process: Improvement and Practice 11 (2), 123–148.

Costabile, M. F, De Marsico, M., Lanzilotti, R., Plantamura, V. L., & Roselli, T. (2005). On the usability evaluation of e-learning applications. In Proceedings of the 38th Hawaii International Conference on System Science (hiccs), 1(1), 1-10.

Christensen, M .G (2005). Essay vs Multiple-Choice Exams. The Ups & Downs of Each Format. In the Proceedings CLEAR Annual Conference, Phoenix, Arizona, September 15, 2005; Retrieved: 2016-11-20 http://www.clearhq.org/resources/2005%20presentations/MCQ.pdf

Day, E. A., Arthur, W., Jr., & Gettman, D. (2001). Knowledge structures and the acquisition of a complex skill. Journal of Applied Psychology, 86, 1022–1033.

Deane, P., & Sheehan, K. (2003). Automatic item generation via frame semantics: Natural language generation of math word problems. Paper presented at the annual meeting of the National Council on Measurement in Education, Chicago.

Doll, W. J., & Torkzadeh, G. (1988). The measurement of end-user computing satisfaction. MIS quarterly, 12(2), 259-274.

Deng, X., Doll. J., Hendrickson, A. R., & Scazzero, J. A. (2005). A multi-group analysis of structural invariance: an illustration using the technology acceptance model. Information & Management 42(5), 745-759.

Gierl, M. J., & Lai, H. (2013). Instructional Topics in Educational Measurement (ITEMS) Module: Using Automated Processes to Generate Test Items. Educational Measurement: Issues and Practice, 32(3), 36-50.

Gütl, C., Lankmayr, K., Weinhofer, J., & Höfler, M. (2011). Enhanced Automatic Question Creator – EAQC: Concept, Development, and Evaluation of an Automatic Test Item Creation Tool to Foster Modern eEducation. The Electronic Journal of e-Learning, 9(1), 23-38.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human–Computer Studies, 43(5/6), 907–928.

Gierl, M. J., Lai, H., & Turner, S.R. (2012). Using automatic item generation to create multiple choice test items. Medical Education, 46(8), 757-765

Gierl, M. J., Lai, H., Hogan, J. B., & Matovinovic, D. (2015). A Method for Generating Test Items that are aligned to the Common Core State Standards. Journal of Applied Testing Technology, 16(1), 1-18.

Hsu ,C-K, Chang, J-C, Chang, M., Jehng, J-C & Heh, J-S.(2002).An Approach for Automatic Learning and Inference by Knowledge Map," AACE (Asia-Pacific Chapter) International Conference on Computers in Education, (ICCE 2002),Auckland, New Zealand, December 3-6, 2002.

Hollis Lai, Mark J. Gierl, B. Ellen Byrne, Andrew I. Spielman, and David M. Waldschmidt (2011). Three Modeling Applications to Promote Automatic Item Generation for Examinations in Dentistry. Journal of Dental Education March 1, 2016 vol. 80 no. 3 339-347.

International Organization for Standardization (ISO) (1998). Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability (ISO 9241-11).

Krathwohl, D. R. (2002): A Revision of Bloom's Taxonomy: An Overview. Theory Into Practice, 41(4), 212-218.

Kominski, C. (2012).Designing Multiple Choice Tests to Measure Higher Order Thinking. Test Item Writing. Paper 10. http://digitalcommons.hsc.unt.edu/test_items/10

Kim, M.-K., Patel, R. A., Uchizono, J. A., & Beck, L. (2012). Incorporation of Bloom's Taxonomy into Multiple-Choice Examination Questions for a Pharmacotherapeutics Course. American Journal of Pharmaceutical Education, 76(6), 114. http://doi.org/10.5688/ajpe766114.

Lu, C., Chang, M., Kinshuk, Huang, E., & Chen, C. (2011). Usability of Context-Aware Mobile Educational Game. Knowledge Management & E-Learning: An International Journal, 3(3), 448-477. Retrieved April 4, 2016, from http://kmel-journal.org/ojs/index.php/online-publication/article/viewArticle/129

Lu, C., Chang, M., & Kinshuk, Huang, E., & Chen, C. (2012a). Usability of Story in Context-Aware Mobile Educational Game. In the Proceedings of the Global Chinese Conference on Computers in Education 2012 (GCCCE 2012), Kenting, Taiwan, May 28-June 1, 2012, 337-340.

Lu, C., Chang, M., Kinshuk, Huang, E., & Chen, C. (2012b). Examining Computer Game Attitude Influence on Students' Voluntariness of Using Educational Mobile Role-Playing Game. In the Proceedings of International Conference on Digital Game-based Learning, (DGL 2012), Hangzhou, China, October 25-27, 2012, 161-165.

Likert, R. (1932). A technique for the measurement of attitudes. Archives of Psychology, 22(140), 5-55. Retrieved April 4, 2016, from http://www.voteview.com/pdf/Likert_1932.pdf

Mason, E. J., & Bramble, W. J. (1989). Understanding and Conducting Research: Applications in Education and the Behavioral Sciences. New York: McGraw-Hill.

Noy, F. N. & Hafner, C.D. (1997).The State of the Art in Ontology Design. AI Magazine, 18(3), 53–74.

Nunnally, J., & Bernstein, I. (1994). Psychometric theory (3rd ed.). New York: McGraw-Hill.

Nicol, D (2007). E-assessment by design: using multiple-choice tests to good effect. Journal of Further and Higher Education 31(1); 53–64

Nesbit, J.C., & Adesope, O. O (2006). Learning with Concept and Knowledge Maps: A Meta-Analysis Review of Educational Research. Fall 2006, 76(3), 413–448http://www.sfu.ca/~jcnesbit/articles/NesbitAdesope2006.pdf

National Science Teachers Association (NSTA, 2009). College Science Teachers Guide to Assessment. Arlington, Virginia, USA. NSTA Press. Retrieved 2016-11-21; http://static.nsta.org/files/PB231Xweb.pdf

Oberle, D., Guarino, N., & Staab, S. (2009). What is ontology? Handbook on Ontologies. Springer, 2nd edition, 2009.

O'Donnell, A.M., Dansereau, D.F., & Hall, R. H (2002).Knowledge Maps as Scaffolds for Cognitive Processing. Educational Psychology Review, 14(1), 71–86; Retrieved November 21, 2016 from http://www.sfu.ca/~jcnesbit/EDUC220/ThinkPaper/OdonnellDansereau2002.pdf

Papasalouros, A., Kotis, K. & Kanaris, K. (2008) .Automatic generation of multiple-choice questions from domain ontologies. IADIS e-Learning conference, Amsterdam, The Netherlands, July 22-25, 2008, 427-434.

Papasalouros, A., Kotis, K., & Nikitakos, N. (2010). Towards an Intelligent Tutoring System for Environmental Decision Makers. Environmental Engineering and Management Journal, 9(2), 197-204.

Preece, J., Rogers, Y., & Sharp, S. (2002). Interaction design: Beyond human-computer interaction. New York: John Wiley.

Parsian, N., & Dunning, T. (2009). Developing and Validating a Questionnaire to Measure Spirituality: A Psychometric Process. Global Journal of Health Science. 1(1), 2-11

Palmer E. J, Devitt P. G (2007). Assessment of Higher order cognitive skills in undergraduate education: modified essay or multiple-choice questions? BMC Medical Education 2007; 7: 49-55. http://bmcmededuc.biomedcentral.com/articles/10.1186/1472-6920-7-49.

Romero, L., Caliusco, M. L. and Gutiérrez, M. (2012). EAOnto: a heavyweight ontology for supporting e-assessment Generation. In the Proceedings of XVIII Congreso Argentino de Ciencias de la Computación, (CACIC 2012), Buenos Aires, Argentina, October 8-12, 2012, 657-666.

Qiu, L. (2015). A Graph-based Algorithm to Build Knowledge Map for Minority Languages. International Journal of Signal Processing, Image Processing and Pattern Recognition. 8(2); 57-64.

Stanescu, L., Spahiu, C. S., & Ion, A. (2008). Question generation for learning evaluation. In the Proceedings of the International Multi conference on Computer Science and Information Technology (IMCSIT 2008), Wisla, Poland, October 20-22, 2008, 509-513.

Schmeiser, C. B., & Welch, C. J. (2006). Test development. In R. L. Brennan (Ed.), Educational measurement. 4th Ed.) Westport, CT: Praeger Publishers.

Simon (2011). Wrong is a relative concept: part marks for multiple-choice questions. Wrong is a relative concept: part marks for multiple-choice questions. In the Proceeding ACE '11 Proceedings of the Thirteenth Australasian Computing Education Conference - Perth, Australia, January 17 - 20, 2011, 47-54.

Tosic, M., & Cubic, M. (2009). SeMCQ-Protégé plugins for automatic ontology-driven multiple-choice questions test generation. In the Proceedings of the 11th International Protégé Conference, Amsterdam, The Netherlands, June 23-16, 2009, Demo & Poster Session: Retrieved April 4, 2016 from http://protege.stanford.edu/conference/2009/poster-session.html.

Taye, M. M (2010).Understanding Semantic Web and Ontologies: Theory and Applications. Journal of Computing. 2(6), 182-192.

Tiemeier A.M., Stacy Z. A, Burke J. M (2011). Using Multiple Choice Questions Written at Various Bloom's Taxonomy Levels to Evaluate Student Performance across a Therapeutics Sequence. Inov Pharm. 2011; 2(2): Article 41. http://pubs.lib.umn.edu/innovations/vol2/ iss2/5

Veeravagu, J., Muthusamy, C, Marimuthu, R. (2010).Using Bloom's Taxonomy to Gauge Students' Reading Comprehension Performance. CANADIAN SOCIAL SCIENCE; 6(3), 2010, 205-212. http://www.cscanada.net/index.php/css/article/viewFile/1069/1088

Wolfe, J. H. (1975). An aid to independent study through automatic question generation (AUTOQUEST). San Diego, CA: Navy Personnel Research and Development Center. Retrieved April 4, 2016, from http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA017059

Yang, Q., Song, B., Lu, W.F., & Zhang, Y. (2007). Integrating knowledge maps in design process configurations for concurrently engineered product development. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. 221(3); 431-445

Zoumpatianos, K., Papasalouros, A., & Kotis, K. (2011). Automated transformation of SWRL rules into multiple-choice questions. In the Proceedings of the 24th Florida Artificial Intelligence Research Society Conference, (FLAIRS 2011), Palm Beach, Florida, USA, May 18-20, 2011, 570-575. Retrieved April 4, 2016, from http://aaai.org/ocs/index.php/FLAIRS/FLAIRS11/paper/view/2631/3108

APPENDIX A– Certification of Ethics Approval

Athabasca University
RESEARCH CENTRE

CERTIFICATION OF ETHICAL APPROVAL

The Athabasca University Research Ethics Board (AUREB) has reviewed and approved the research project noted below. The AUREB is constituted and operates in accordance with the current version of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS) and Athabasca University Policy and Procedures.

Ethics File No.: **22300**

Principal Investigator:
Mr. Ebenezer Aggrey, Graduate Student
Faculty of Science & Technology\Master of Science in Information Systems

Supervisor:
Dr. Maiga Chang (Supervisor)

Project Title:
Automatic Test Item Generation from Knowledge Structure

Effective Date: September 16, 2016 **Expiry Date**: September 15, 2017

Restrictions:

Any modification or amendment to the approved research must be submitted to the AUREB for approval.

Ethical approval is valid *for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.

A Project Completion (Final) Report must be submitted when the research is complete *(i.e. all participant contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable))* or the research is terminated.

Approved by: **Date:** September 16, 2016

Ali Akber-Dewan, Chair
School of Computing & Information Systems, Departmental Ethics Review Committee

Athabasca University Research Ethics Board
University Research Services, Research Centre
1 University Drive, Athabasca AB Canada T9S 3A3
E-mail rebsec@athabascau.ca
Telephone: 780.675.6718

APPENDIX B– Certification of Ethics Approval

Athabasca University
RESEARCH CENTRE

CERTIFICATION OF ETHICAL APPROVAL - RENEWAL

The Athabasca University Research Ethics Board (AUREB) has reviewed and approved the research project noted below. The AUREB is constituted and operates in accordance with the current version of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS) and Athabasca University Policy and Procedures.

Ethics File No.: **22300**

Principal Investigator:
Mr. Ebenezer Aggrey, Graduate Student
Faculty of Science & Technology\School of Computing & Information Systems

Supervisor:
Dr. Maiga Chang (Supervisor), Associate Professor
Faculty of Science & Technology

Project Title:
Automatic Test Item Generation from Knowledge Structure

Effective Date: September 8, 2017 **Expiry Date**: September 07, 2018

Restrictions:

Any modification or amendment to the approved research must be submitted to the AUREB for approval.

Ethical approval is valid *for a period of one year*. An annual request for renewal must be submitted and approved by the above expiry date if a project is ongoing beyond one year.

A Project Completion (Final) Report must be submitted when the research is complete *(i.e. all participant contact and data collection is concluded, no follow-up with participants is anticipated and findings have been made available/provided to participants (if applicable))* or the research is terminated.

Approved by: **Date:** September 8, 2017

Joy Fraser, Chair
Athabasca University Research Ethics Board

Athabasca University Research Ethics Board
University Research Services, Research Centre
1 University Drive, Athabasca AB Canada T9S 3A3
E-mail rebsec@athabascau.ca
Telephone: 780.675.6718

APPENDIX C – Symbol Table for Item Generation Workflow and Algorithm

Symbol	Definition
ACTOR	Role set for item stimulus and stem creation; ACTOR ={$actor_{programmer}$, $actor_{student}$, $actor_{software\_developer}$, ...}
$actor_{programmer}$	A programmer.
$actor_{software\_developer}$	A software developer.
$actor_{student}$	A student.
$ANSWER_j$	set of distractors and keys for $item_j$; $ANSWER_j = KEY_j \cup DISTRACTOR_j$
A^t	Set of ancestor of a given concept node c_t;. $A^t = \{a^t{}_1, a^t{}_2, \ldots, a^t{}_n\}$
AV^t	Set of rule attribute value pairs, e.g., {<operation, push>, <description, Last in First Out>} if c_t is "Stack".
c_t	Concept node selected by teacher.
CT	Set of cognitive types; CT = $\{ct_1, ct_2, \ldots, ct_n\}$.
ct	Cognitive type "Apply"; ct $\subset$ CT.
$c^t{}_p$	A parent node of a given concept node c_t.
$DISTRACTOR_j$	Set of answer distractors for $item^A{}_j$; $DISTRACTOR_j = \{distractor_{j1}, distractor_{j2} \ldots distractor_{jn}\}$.
G	Set of items generated; $G = \{g_{j1}, g_{j2}, \ldots, g_{jn}\}$.
H^t	Set of children of a given concept node c_t; $H^t = \{h^t{}_1, h^t{}_2, \ldots, h^t{}_n\}$
$item_j$	Apply item generated to be added to set $G^A = \{item_1, item_2, \ldots, item_n\}$
KEY_j	set of answer keys for $item_j$; KEY = $\{key_{j1}, key_{j2}, \ldots, key_{jn}\}$;
m_{jk}	Stem created for item with a given a given rule r
$name(c_j)$	A concept node's name
$name(ra_j)$	Rule attribute name, e.g. description, application, operation, etc.
RA	Set of item rule attribute. RA = $\{ra_1, ra_2, \ldots ra_n\}$
ra_k	An element of RA. $ra_k \in$ RA
RA^M	Rule attribute set for item stem for a given r. $RA^M \subset RA$
RA^O	Rule attribute set for item answer options for a given r. $RA^O \subset RA$
RA^S	Rule attribute set for item stimulus for a given r. $RA^S \subset RA$
R	Rule set of items. R = $\{r_1, r_2, \ldots, r_n\}$
r	Rule of item. $r \in R$
s_{jk}	Stimulus created for item with a given a given rule r.
S^t	Set of sibling of a given concept node c_t; $S^t = \{s^t{}_1, s^t{}_2, \ldots, s^t{}_n\}$
TASK	Task set for item stimulus and stem creation. TASK ={$task_{software}$, $task_{component}$, $task_{module}$}
$task_{component}$,	A component task.
$task_{module}$	A module task.
$task_{software}$	A software task.
TM	Stem template set for item; TM = $\{tm_1, tm_2, \ldots, tm_n\}$.
TO	Answer option template set for item; TO = $\{to_1, to_2, \ldots to_n\}$.
TS	Stimulus template set for item; TS = $\{ts_1, ts_2, \ldots, ts_n\}$.

APPENDIX D– Cohen's Kappa Scale Interpretation/ Bloom's Cognitive Levels

Table A1: Cohen's Kappa Scale Interpretation

Kappa	Interpretation
< 0	Substantial Agreement Poor Agreement
0.0 - 0.20	Slight Agreement
0.21 - 0.40	Fair Agreement
0.61 - 0.80	Substantial Agreement
0.81 – 1.00	Almost Perfect Agreement

Table A2: Blooms Cognitive Level

Blooms Cognitive Level	Classification	Symbol
Remember	Low	R
Understand	Low	U
Apply	High	A
Analyze	High	N
Evaluate	High	E

APPENDIX E– Cognitive Level Items Generated

#	Item	Cognitive Level
1	A programmer was asked to design a data structure which implements Linked List interface and can be used to check for matching braces in compiler syntax. Choose the best data structure the programmer has to implement for the software task. A. Stack B. Tree C. Queue D. Dictionary	Apply
2	A USet is an interface which can provide specification about the types of arguments of each operation supported. A. True B. False	Remember
3	A programmer was asked to implement generic data structure G which implements Binary Search Tree interface. The specification for the data structure is as follows: 1. Implement algorithm for M1() 2. The algorithm should run efficiently The algorithm implemented is shown below: for (int i=1; i <=n i *=c) {//some 0(1) expressions} Assuming c constant and the expressions within the loop are 0(1) then what is the time complexity of the algorithm? A. O(log n) B. O(1) C. O(n) D. O(n^2)	Analyze
4	A student wants to design a data structure which can be used to model transport networks in a big city. Choose the best data structure the student has to implement for the software task. A. Queue B. Tree C. Dictionary D. Graph	Apply
5	A programmer wants to implement ArrayDeque for software module. In order for the data structure to conform to correctness specification the programmer has to implement all the operations of Array-Based List. A. True B. False	Remember
6	A student wants to design a data structure which implements HashTable interface and can be used to lookup of students' records very fast. Choose the best data structure the student has to implement for the software task. A. Tree B. Stack C. Dictionary D. Queue	Apply
7	A software developer implemented a software component which takes Stack s containing Ya, Fan, Nanby, Podyah as parameter. The algorithm implemented is shown below String output; while(!s.isEmpty() && s.peek().length()==6){ String str=s.pop(); output=s.peek().substring(0,1); s.pop(); s.push(output); } What is the expected output of the operation? A. Ya B. Fan C. N	Apply
8	Four programmers A, B, C D design and implemented M1() operation of data structure which implements HashTable interface. The computer processing as a function of time complexity for the M1 () operation algorithms were analysed and the result is shown below: TA(n) =n^2 , TB(n) = n^3, TC(n)=log n, TD(n) = n Choose the algorithm which is better in terms of Big-Oh sense for the implementation of the software component . A. TB(n)= n^3 B. TD(n)= n C. TA(n)=n^2	Evaluate

9	A SSet is an interface which can describes the implementation of each operations it supports. A. True B. False	Remember
10	10. A Graph is an interface which can define set of operations supported by data structure. A. True B. False	Remember
11	A programmer implemented a software component which takes Stack s containing Yank,Fan,Nanb,Pod as parameter. The algorithm implemented is shown below String output; while (s.peek ().length () %2! =0){ String str=s.peek(); output=str; } What is the expected output of the operation? A. Yank B. Nanb C. Pod D. Fan	Apply
12	A Graph is an interface which can defines the algorithms of each operations it supports. A. True B. False	Remember
13	A Graph is an interface which can define set of operations supported by data structure. A. True B. False	Remember
14	A student implemented generic Map utilizing ChainedHashTable which implements HashTable for software component to allow lookup of values with key. During unit testing of the component the following operations m.put(0,rxlqu); m.put(1,gbehq); m.put(2,iahal); m.put(3,vlwph); s.remove(1) were executed in sequence. What will be the expected output? A. vlwph B. gbehq C. rxlqu D. iahal	Remember
15	A student wants to implement ArrayQueue for software component. In order for the data structure to conform to correctness specification the student has to implement all the operations of DLList (Doubly-Linked List). A. True B. False	Understand
16	In order to satisfy correctness specification of DLList (Doubly-Linked List), a student should implement all the operations defined in Linked List interface. A. True B. False	Understand
17	A programmer implemented a software component which takes Stack s containing La,Sip,Bloby,Bulbyn as parameter. The algorithm implemented is shown below String output; while(!s.isEmpty() && s.peek().length()==6){ String str=s.pop(); output=s.peek().substring(0,1); s.pop(); s.push(output); } What is the expected output of the operation? A. So B. Sop C. S	Apply
18	A software developer implemented generic Deque utilizing RootishArrayStack which implements Array-Based List for software module to model a line in a store. The first customer in line is the first one served.Given input data gwcgk,rjerp,zwfcb,fpcrn to be added to the queue in the same sequence, when operation pollLast() is executed What will be the expected output? A. fpcrn B. rjerp C. gwcgk D. zwfcb	Apply
19	A software developer implemented a software component which takes Stack s containing Jan,Fran,Nan,Stan as parameter. The algorithm implemented is shown below String output; while (s.peek().length()%2==0){ String str=s.peek(); output=str; } What is the expected output of the operation? A. Cat B. Pong C. Van D. Ping	Apply

20	In order to satisfy correctness specification of SkipListSet, a programmer should implement all the operations defined in SSet interface. A. True B. False	Understand
21	A student designed and implemented ArrayDeque to be used in software component. The data structure implemented Array-Based List interface which supports M1 () operation. The student implemented the M1() as follows : //Here c is a constant for (int i =1 ; i <= c ; i++){ // some O(1) expressions } If c is constant then what is the time complexity of the algorithm? A. O(log n) B. O(n^2) C. O(n) D. O(1)	Analyze
22	A programmer wants to design a data structure which implements SSet interface and can be use to model folders and files on a hard drive. Choose the best data structure the programmer has to implement for the software task. A. Dictionary B. Queue C. Graph D. Tree	Apply
23	A student wants to do performance analysis after designing and implementing SLList (Singly-Linked List) data structure. Which of the following performance analysis should the student perform I: Correctness II: Time Complexity III: Space Complexity A. I B. II C. I,II,III D. III	Understand
24	A programmer implemented generic Deque utilizing Linked List which implements List for software module to model jobs to a network printer. Given input data 1791,273,1004,468 to be added to the queue in the same sequence, when operation peekLast() is executed What will be the expected output? A. 468 B. 273 C. 1791 D. 1004	Apply
25	A programmer is considering using a third party data structure(TPDS) to implement generic data structure S which should be able process about 10,000 data items. Additionally, it should implement SSet interface and provide algorithm for the M1() operation. The computer processing time in relationship to time complexity for the M1() operation of the TPDS is as follows. A1 (n)=n^2 ,A2(n) = 5.3n^3, A3(n)=log n, A4(n)=200n^3 Choose the algorithm which is better in terms of Big-Oh sense for the implementation of the software component . A. A2(n)= 9n^3 B. A1(n)=n^2 C. A3(n)=logn	Evaluate
26	An algorithm a student implemented for one of the methods of Binary Search Treepent 10 ms to process 4000 data items. How much time will be spent to process 100000 if computer processing T(n) of the algorithm as function of time complexity O(f(n) is defined as T(n)=cn^3 where c is constant A. 312500 ms B. 468750 ms C. 156250 ms D. 625000 ms	Analyze
27	A programmer designed and implemented SEList (Space-Efficient List) which implements Linked List interface. Identify the most appropriate performance analysis the programmer has to perform to ensure that the algorithm of each method implemented works efficiently A. Space Complexity B. Time Complexity C. Auxiliary Space Complexity D. Correctness	Understand
28	In order to satisfy correctness specification of Linked List, a software developer should implement all the operations defined in List interface. A. True B. False	Understand

29	A student designed and implemented generic data structure G for software component leveraging BinaryHeap.The data structure implemented Heap interface which supports M1() operation. The student implemented the M1() as follows : //Here c is a constant for (int i =1 ; i <= n ; i+=c){ // some O(1) expressions } Assuming c is constant then what is the time complexity of the algorithm? A. O(log n) B. O(n^2) C. O(1) D. O(n)	Analyze
30	A programmer implemented generic Map utilizing LinearHashTable which implements HashTable for software component to allow lookup of values with key.During unit testing of the component the following opertions m.put(0,peltt); m.put(1,rlvai); m.put(2,yfhlr); m.put(3,mwewd); s.remove(0) were executed in sequence. What will be the expected output? A. yfhlr B. mwewd C. peltt D. rlvai	Apply
31	A software developer designed and implemented Linked List to be used in software component. The data structure implemented List interface which supports M1() operation. The student implemented the M1() as follows : //Here c is a constant for (int i =1 ; i <= c ; i++){ // some O(1) expressions } If c is constant then what is the time complexity of the algorithm? A. O(log n) B. O(n) C. O(1) D. O(n^2)	Analyze
32	A software developer is considering implementing a BinaryHeap data structure which implements Heap interface for a software module. Select the best performance analysis the software developer has to perform to ensure that the algorithm of each method implemented has reasonably memory usage A. Correctness B. Auxiliary Space Complexity C. Space Complexity D. Time Complexity	Understand
33	A programmer was given a SLList (Singly-Linked List) data structure to implement software module. After profiling one of the methods of the module it took 1 ms to process 100 data contain in the data structure How much time will be spent to process 5000. Assume that the processing time of the method algorithm is T(n)= Cn^2 where n is the number of data items and C is constant A. 10000 ms B. 2500 ms C. 7500 ms D. 5000 ms	Apply
34	A student implemented a software component which takes Queue s containing zp,matt,cret,creek as parameter. The algorithm implemented is shown below String output; while(!queue.isEmpty() && queue.peek().length()==2){ String str=queue.remove(); output=queue.peek(); } What is the expected output of the operation? A. cret B. matt C. creek D. zp	Apply
35	A software developer implemented generic Deque utilizing Array-Based List which implements List for software module to model a line in a store. The first customer in line is the first one served.Given input data mfjbb,ktzry,jcsjv,vhdjy to be added to the queue in the same sequence, when operation remove() is executed What will be the expected output? A. vhdjy B. ktzry C. jcsjv D. mfjbb	Apply
36	A programmer wants to do performance analysis after designing and implementing ScapegoatTree data structure. Which of the following performance	Understand

	analysis should the programmer perform I: Correctness II: Time Complexity III: Space Complexity A. I,II,III B. I C. III D. II	
37	A programmer implemented generic Deque utilizing ArrayQueue which implements Array-Based List for software module to model people arrive and get in line for a teller at a bank.Given input data 737,1064,203,72 to be added to the queue in the same sequence, when operation pollFirst() is executed What will be the expected output? A. 1064 B. 737 C. 203 D. 72	Apply
38	A software developer implemented a software component which takes Stack s containing So, Sop, Soupy, Punkyh as parameter. The algorithm implemented is shown below String output; while(!s.isEmpty() && s.peek().length()==6){ String str=s.pop(); output=s.peek().substring(0,1); s.pop(); s.push(output); } What is the expected output of the operation? A. S B. Sop C. So	Apply
39	A programmer was given a SLList (Singly-Linked List) data structure to implement software module.After profiling one of the methods of the module it took 1 ms to process 100 data contain in the data structure How much time will be spent to process 5000. Assume that the processing time of the method algorithm is T(n)= Cn^2 where n is the number of data items and C is constant A. 10000 ms B. 2500 ms C. 7500 ms D. 5000 ms	Apply
40	A student implemented a software component which takes Stack s containing Jan,Fran,Nan,Stan as parameter. The algorithm implemented is shown below String output; while (s.peek().length()%2==0){ String str=s.peek(); output=str; } What is the expected output of the operation? A. Stan B. Jan C. Fran D. Nan	Apply