DECLARATION

The research as presented in this report is my original work and has not been presented for any other university award.

Signed: ....Gathiti..

Name: Leah Nyambura Gathitu

S. NO: P56/p/7638/04

Date: ....16/8/2010..

This research has been submitted as part fulfilment of requirements of the Masters of Science in Information Systems of the University of Nairobi with my approval as the Supervisor

Signed: ....

Supervisor: Dr. Elijah Omwenga

Date: ....16/8/2010..............................

ABSTRACT

This research focused on the applicability of computer adaptive testing in the selection of candidates from an interview process. A computer adaptive test adapts to the ability of an individual student and presents questions that are suited to the candidate's ability. It does this by revising the ability estimate of the individual as the test progresses. With the revised ability estimate, the question that presents the most information about the student is selected for presentation. The test progresses until the change in ability estimation becomes negligible, meaning that the estimated ability is very near the true ability of the student. After the stopping criterion is met and the test completes, a report is immediately presented to the student indicating his performance as a percentage score.

The research also looked at previous studies conducted on the subject and also how interviews are conducted. The researcher identified that there is a gap and a need to improve the interview process. The gap is that written interviews do not accurately capture the ability of a student and therefore may result to good candidates being eliminated from an interview. It is important that a test be able to capture the ability of a student accurately so that the best candidates are selected. To demonstrate how such a system works, a design, implementation and evaluation of a model of an adaptive test was done, and then findings from the research were highlighted.

The research identifies the benefits and drawbacks of using the computer adaptive test approach and also the areas that it is best applied and used.

ACKNOWLEDGEMENTS

This work has not been easy and I would like to recognize and applaud the individuals who encouraged me and enabled me to achieve its completion.

First I give all my thanks to God who has given me life, strength, courage and resources to complete the course.

Secondly, I am very grateful to my supervisor, Dr. Elijah Omwenga for his insights and guidance. His constructive comments guided me and ensured that the research was well conducted.

I am also very grateful to other supervisors at the school of computing who encouraged me forward and told me not to give up.

I am indebted to a number of people with whom I have discussed many of the ideas presented in this work and they offered their comments and suggestions despite their busy schedules. In particular, I thank Samuel Wanjere and Steve Nyaga.

Special thanks go to my parents for their unfailing support and belief in me all through my life. They have been a pillar of strength to me. Finally a big thank you goes to my husband Pinto for pushing me forward when I thought the road was too tough. THANK YOU VERY MUCH.

Table of Contents:

List of Figures

1. CHAPTER 1 - INTRODUCTION

1.0 Introduction

The purpose of this research is to demonstrate the applicability of computer adaptive tests in examinations. The research will compare the test with comparable paper tests. In addition, this research will practically demonstrate the working of a prototype of a Computer Adaptive Test.

In the Kenyan curriculum, examinations have been, and are still administered through the use of paper tests. While this has been the norm and is perceived to be a reasonable mode of testing, the research will establish whether this testing system enables assessors ascertain the mental skills of their students.

The study will seek, through the use of an adaptive testing system, to find a way of enabling a student's ability to be determined progressively as the test is being undertaken. The result will then be presented in a report format that indicates the analysis of the results.

1.1 Problem Statement

Traditional fixed-length, pencil-and-paper fixed-item testing, remains the predominant testing strategy in educational and training settings. It involves the administration of a fixed set of questions to a student population. An examinee is expected to answer all questions within a fixed period of time. There is a predefined ordering of questions but an examinee does not need to answer in that order; he may skip questions and return to them later. As this type of testing has to reach out to examinees of all capabilities within a population, therefore, there may be relatively few questions, which are of the appropriate difficulty for any one examinee. Questions may be too difficult for the weak examinee, or too easy for a good examinee. As a consequence, large numbers of questions may be needed to obtain an acceptable degree of precision.

In addition, questions are arranged in order of difficulty. This may work well for a less proficient examinee as he will be able to answer the earlier questions which are easier before reaching the more difficult ones. For a good examinee, however, he would have to wade through the easier ones before reaching the more challenging questions. In both cases, there is a possibility of extraneous noise such as guesswork and careless slips. For the less proficient student, anxiety may set in when he attempts to tackle the more difficult questions and he may attempt to solve them through guessing. For a more proficient student, boredom may set in when he wades through the easy questions and this wastes time and may increase the possibility of noise mainly caused by careless errors or slips (Abdullah, 2003).

In many of the recruitment companies aptitude tests are used in interviews to determine the best candidates for an advertised position. Aptitude tests are designed to assess your logical reasoning or thinking performance. They consist of multiple choice questions and are administered under exam conditions. They are strictly timed and a typical test might allow 30 minutes for 30 or so questions (Psychometric-Success, 2009). These tests are largely paper based in Kenya. The tests are usually to be undertaken within a strict period of time which is usually too short. These tests favor those who have speed and can tackle most of the questions. From my observation and experience, a student may fail simply because they were not fast enough or because they wasted time answering questions that were not crucial in determining their ability. In such cases, very bright and able students have been locked out of opportunities because of a system of testing that is limited in certain aspects. To counter the weakness of the system failing in determining the best candidates from the interview, the research has focused on computer adaptive tests as a tool for testing in such situations.

Computer-adaptive testing (CAT) is a technologically advanced method of assessment in which the computer selects and presents test items to examinees according to the estimated level of the examinee's ability. The basic notion of an adaptive test is to mimic automatically what a wise examiner would normally do. Specifically, if an examiner asked a question that turned out to be too difficult for the examinee, the next question asked would be considerably easier. This approach stems from the realization that we learn little about an individual's ability if we persist in asking questions that are far too difficult or far too easy for that person. We learn the most about an examinee's ability when we accurately direct our questions at the current level of the examinee's ability (Dunkel, 1999).

In computer adaptive testing (CAT), the examinee's estimated ability level is used to predict the probability of getting an item correct. Knowing nothing about an examinee in the beginning, s/he is assumed to be of average ability. Thus, the CAT begins by administering an item of average difficulty. An examinee that gets the first item correct is then given a more difficult item; if that item is answered correctly, the computer administers an even more difficult item. Conversely, an examinee who gets the first item wrong is then administered an easier question. In short, the computer interactively adjusts the difficulty of the items administered based on the success or failure of the test taker.

In a relatively short period of time, the CAT is consistently administering items appropriate to the examinee's ability level. Doing so maximizes the information gained about the examinee's ability level. The CAT stops administering items when some criterion is met. The most common stopping criterion includes precision (e.g., when then standard error of the ability estimate falls below some value), time, and number of items administered (Weekley, 2004). Other stopping criterions include: number of items answered, when the test is complete, when all topics have been covered, among others.

1.2 The objectives of the study

i. Identify issues in designing and implementing a CAT application.

ii. Examine the existing system of testing in recruitment institutions noting its strengths and drawbacks. In addition, the research will examine computer adaptive testing, also noting its positives and negatives.

iii. The second objective of this study will be to design and implement a CAT application. The corresponding results of the test will be presented in a report format showing the question answered, the answer given, whether the answer is correct or not, the expected outcome and the final ability estimate as a percentage.

1.3 Hypothesis of the study

1. Paper based testing as used in recruitment has resulted to examiner's failure to realize examinee's true ability level, resulting to 'poor' and inaccurate examination results that do not necessarily reflect on the individual's capability.
2. Computer adaptive tests are a way of accurately determining the examinees level of ability and knowledge.

1.4 Significance of the study

The hope underpinning this research is that just as paper based testing is currently considered to be the most effective form of examining students, computer to human one-on-one testing to assess the state of knowledge of a student may be the most effective form of assessment for certain kinds of subjects/situations, and that the testing strategy is worth capturing, so as to arrive at the best system for determining the capabilities of students and employees in these subjects.

The students in today's education system are the future leaders and policy makers of our nation. They will comprise our workforce and help define our economy. The education and testing system must ensure that they possess the necessary skills that enable them to be active participants in our democracy and to realize their economic and personal goals.

1.5 General scope and Limitations of study

The research will be centered on the adaptivity of the system only. It will not be involved in the construction of the tests which are assumed to be done by an expert and is beyond the scope of this research.

The researcher appreciates that computer adaptive testing cannot entirely substitute paper based testing. This research will therefore find out the areas that Computer Adaptive Testing can be used most appropriately.

1.6 Aims of the research

The research aims to answer two questions:

i. Application of computer adaptive testing (CAT) in the interviewing process
ii. Can computer adaptive testing aid in improving the accuracy of selection of candidates from an interview?

1.7 Definition of key terms

a) ***Ability*** - the quality of being able to perform; a quality that permits or facilitates achievement or accomplishment possession of the qualities (especially mental qualities) required to do something or get something done (Dictionary.com, 2000)
b) ***Computer Adaptive Test*** - technologically advanced method of assessment in which the computer selects and presents test items to examinees according to the estimated level of the examinee's ability

c) ***Computer Based Test*** - This is a type of an examination administered by computer. These tests may range from conventional multiple-choice tests administered on a personal computer to virtual reality simulations.

d) ***Efficiency*** - Production of the desired effects on results with minimum waste of time, effort or skill. (Dictionary.com, 2000)

e) ***Equity*** – State, quality or ideal of being just, impartial and fair. (Dictionary.com, 2000)

f) ***Excellence*** – displaying a quality of superiority. According to Statefam.com, there are nine aspects of education excellence namely: Safety & discipline, Parent involvement , Assessment , Learning readiness , Standards , Accountability , Technology , Professional development , School autonomy. This research focuses on assessment.

g) ***Paper Based Testing*** – Testing whereby each a student is required to answer a fixed number of questions that are presented to him/her on a paper. Every student doing that particular test is presented with the exact set of questions.

h) ***Student modeling*** - Student modeling, as the model of a learner, represents the computer system's belief about the learner's knowledge. It is generally used in connection with applications computer-based instructional systems (Stauffer, 1996).

2. CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction

A look at the Kenyan examination system by the Kenya High Commission reveals that admission requirements to various Kenyan training institutions were previously based on the Kenya Certificate of Education and the Kenya Advanced Certificate of Education. The Ordinary Level Course had divisions I, II, III and IV with grades for each subject running from 1 to 9. But the advanced Level Course had its candidates graded in principals A, B, C, D, E and subsidiary passes. However, the Kenya Certificate of Secondary Education grading system has no provision for divisions and principals. In the 8.4.4 system, each subject is graded on the basis of a 12-point scale. 12 points is the highest and 1 point the lowest score. These points are accompanied by an expanded grading system of A, A-, B+ ... E.

This system takes into account the level of performance as a major criterion for determining the quality of candidates to be considered for further education. Institutions offering further education and training for the 8.4.4 secondary school leavers are required to use these grades and points to select suitable candidates for their training programs.

After the student is trained in university, s/he will apply for employment in various institutions. Many professional institutions have interview procedures that include undertaking of aptitude tests to grade and select the best candidates to progress to later stages of assessment.

Aptitude tests are formal, timed tests, taken either online or by filling in a printed answer sheet. In Kenya, they are usually administered on paper tests. They usually take the form of multiple choice questions. A candidate will be given full instructions before they start the test.

The tests most commonly used in graduate recruitment are:

- Verbal tests - such as verbal reasoning, analysis and word sort;
- Numerical tests - such as reasoning, analysis and sequential tests;
- Diagrammatic and spatial reasoning - testing your sense of logic and ability to deal with shapes;
- Specific tests - for example syntax for computer programming, data checking or mechanics.

This research will address the question of how a part of the Kenyan recruitment system, namely testing, can be improved in order to ensure that students are well evaluated and graded thereby resulting to proper selection of candidates for recruitment positions.

2.1.1 ICTs in the Kenyan education system

Omwenga (2003) notes that technologies are being employed as tools for aiding in the management and administration of the system or in some cases they are being introduced into the curriculum as subjects of study or as tools to enhance teaching. He further notes that the latter is only in the conceptual stages and concrete stages on the ground are yet to be implemented.

Computer adaptive testing is a new and untapped area in the Kenyan recruitment system. It is only applied in international exams such as Graduate Record Exam (GRE), GMAT and Toefl exams among others. The importance of this kind of testing is yet to be realized and appreciated among Kenyan testing systems.

This study is concerned with modeling a computer adaptive testing system in a pedagogically sound manner.

2.1.2 Potential of ICTs in our education system

ICTs are considered by many to have tremendous potential for enhancing the learning process (Hughes, 1990). It is felt that they can increase not only the effectiveness of the educational process but also its overall efficiency, whether in terms of classroom activities or administration. The possibilities they offer have the potential to transform the organization and structure of schooling and may promote the development of cognitive processes.

2.1.3 Factors inhibiting ICTs from being used in the country

The main factors that impede adequate use of ICTs in educational institutions include the following:

- Costs – Price of hardware and software is still considerable for many educational institutions. The rate of evolution of these technologies implies constant upgrading of equipment which has a cost to it (Omwenga, 2003).
- Teacher education is considered to be the single most important factor in ensuring the successful use of ICTs in education (Walker, 1989; Duguet, 1989; Lally, 1989). Teacher education is not only vital for equipping educators with the necessary skills for using ICTs effectively in the classroom, but for helping teacher overcome their often resistance to these technologies (Omwenga, 2003)
- Conflict with curriculum – Ideally teachers should be involved in the development and evaluation of such testing systems so as to genuinely meet their needs, gain their trust and be accepted by them as valid testing material.

Herremans (1995) suggests that one of the reasons for using technology in education is the need to improve the quality of the teaching/learning process. Institutions mustn't be left behind while the rest of the globe improves its operations through the use of ICTs. The future of institutions depends on the capacity to adapt to the new information society and meet the needs of an ever more demanding professional market.

2.2 History of adaptive tests

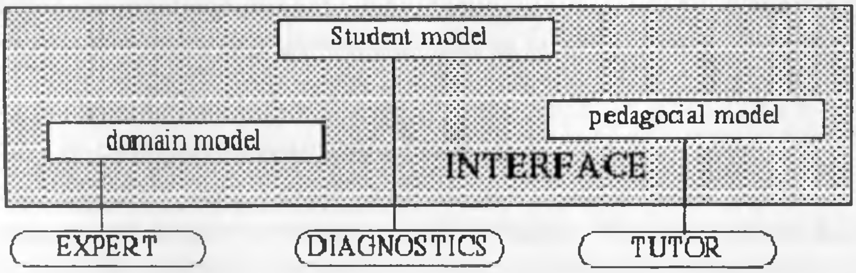
According to the Certification and Skills Assessment Group, Adaptive testing is not new. In fact it has probably been around for centuries as a better way to test. And even since the beginning of the 20th century, when large-scale testing began, adaptive tests were some of the first tests ever constructed. Alfred Binet created the first adaptive test at the end of the 18^{th} century, designing an IQ test that began with questions that matched the child's age and ended when the child could not answer a few questions in a row correctly. Binet's IQ test, which is still in use today in a more modern version, was not computerized, but used individual examiners to administer the tests. Computer adaptive testing stems from intelligent tutoring system as described below:

2.3 Intelligent Tutoring System

An intelligent tutoring system is one that is able to mimic a human tutor. A definition from Wikipedia.org (2006) explains it as follows: "An **Intelligent Tutoring System** (ITS) replaces a human tutor by a machine. In most cases, it's "intelligent" Computer Based Training. Research in ITS organizes the "problem" in
(1) Knowledge about a domain,
(2) Knowledge about the learner, and pedagogy (knowledge of teaching strategies).

The major components of a typical ITS are therefore an expert (or domain) model, student model and tutoring model. The expert model should be able to solve the problems the tutoring module submits to the students. The tutor module controls the interaction with the student, based on its teaching knowledge and comparisons between the student model and the domain knowledge. The student model reflects what the machine can infer about the student's cognitive state. Most ITS are implemented with expert system technology". This is illustrated in **figure 1**

Figure 1: Components of an Intelligent Tutoring model



2.3.1 Components of an Intelligent Tutoring System (ITS)

According to Abdulla (2003), there are three important functions of an Intelligent Tutoring System (ITS) (McCalla and Greer, 1994). First, an ITS models the knowledge of the learner in some computationally useful and inspectable way. Next, based on the student model of the learner, the ITS intervenes in the interaction between system and learner with the goal of facilitating learning. Finally, the ITS evaluates the success of its intervention and adjusts its model of the learner, and the loop repeats.

In order to carry out these functions in a one-on-one interaction with a learner, an ITS must have a model of instructional content that specifies *what* to teach, a model of a teaching strategy that specifies *how* to teach, and a model of a student that specifies *who* to teach (Ohlsson, 1987). Wenger (1987) describes an ITS as a *knowledge communication system* which comprises at least four functional interacting components: the domain knowledge model, the pedagogical module, the student model, and the interface or communication module.

• Domain Knowledge Module
This module contains the knowledge of *what* to teach. It represents an area of syllabus and usually requires knowledge engineering in its construction. Domain knowledge is usually represented as skills, concepts, procedures and problems of the subject domain under study.

• The Pedagogical Module
This component contains the knowledge of *how* to teach, that is a teaching or tutoring strategy. It orchestrates the whole tutoring process and deals with issues like which topic to present, when to present a new topic, when to present a problem, when to review, and when to offer remedial help.

• Student Model
This component contains the knowledge about *who* it is teaching. It keeps track of information that is specific to each individual student, such as his mastery or competence of the material being taught, and his misconceptions. In effect, it stores the computer tutor's beliefs about the student. This information is used by the pedagogical module to tailor its teaching to the individual needs of the student.

• Interface Module
This module provides a communication mechanism for handling the interactions between the computer tutor and the student, such as mixed-initiative dialogues.

2.4 What is computer adaptive testing?

Computerized adaptive testing or **CAT** (Wainer, 1990) is a recent arrival into the scene of student modeling. With its roots in psychometric measurement, CAT is characterized by the efficiency and accuracy at inferring a student's knowledge in a domain with the minimum number of problems. The student is presented with problems of appropriate difficulty. This has the advantage of reducing test anxiety, sustaining the motivation of students during testing, and more importantly, of reducing the overall testing time.

An interesting analogy between measurement within a tutoring system and psychometric measurement was made by Frederiksen and White (1990). The experience of taking a CAT can be loosely compared to participating in a high-jump event in track-and-field competition. The high-jumper, regardless of ability, quickly reaches a challenging level where there is about an equal chance of clearing the bar or knocking it down. The "score" for the high-jumper is the last height he or she was able to jump over. The high-jumper earns the score without having to jump over every possible lower height, and isn't required to try all the higher levels. Similarly, for the person taking a CAT, where test questions are ranked from easy to hard, the score would be based on the point where the person encounters questions that are too difficult (Certification and Skills Assessment Group, 2004).

In view of that, Computer adaptive testing (CAT) aims to determine your level of competency by *adapting* each question to your previous response. When you answer a question correctly, a more difficult question will appear. If you answer incorrectly, an easier question comes up. Each question's level of difficulty is based on the answers to previous questions until a consistent level of competency is determined. Adaptive testing technology can establish this competency level more quickly than conventional examination techniques. (Thomson Prometric, 2004)

A traditional fixed-length computerized or paper-and-pencil exam presents the same number of questions to each test taker, without considering how well the person is doing on the exam. The score from this type of test usually depends on the number of questions answered correctly. The more a person knows then the more questions he or she should be able to answer correctly. Traditional exams have a long and successful history dating back to the second decade of the 20th century; however, it is clear that for any one person, the traditional test presents more questions than are necessary. For any single person there are questions on the test that are far too easy and those that are far too hard. Answering easy questions correctly doesn't tell us much about the person, because most people answer the easy ones correctly. Likewise, and for a similar reason, answering the difficult questions incorrectly tells us very little as well. It would be better if a test were able to discover the level, on a scale of easy to difficult, where the person begins to encounter personally challenging questions. A score could be derived for that level. A computerized adaptive test (CAT) does just that (Certification and Skills Assessment Group, 2004).

In computer adaptive testing (CAT), the examinee's estimated ability level is used to predict the probability of getting an item correct. By taking into account how each person answered previous questions, taking the same CAT, a low-ability examinee and a high-ability examinee will see quite different sets of questions: The low-ability examinee will mainly see relatively easy questions, and the high-ability examinee will see more difficult questions. Both individuals may answer the same percentage of questions correctly, but because the high-ability person can answer more difficult questions correctly, he or she will get a higher score (Certification and Skills Assessment Group, 2004).

In a relatively short period of time, the CAT is consistently administering items appropriate to the examinee's ability level. Doing so maximizes the information gained about the examinee's ability level. The CAT stops administering items when some criterion is met.

2.5 What types of questions appear on the examinations?

The types of questions that appear on computer-based examinations vary by exam. They include the following (prometric, 2005):

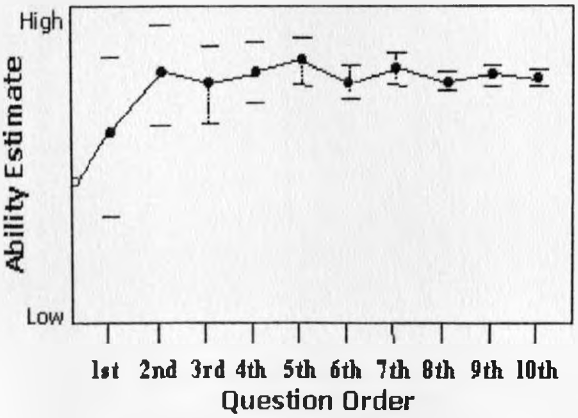
- Multiple choice
- True/False

- Fill-in-the-blank
- Short answer
- Essay
- Drag and drop

2.6 How a CAT works

A CAT works like a good oral exam. It first presents a question of moderate difficulty. After the answer is given, the question is scored immediately. If correct, the test statistically estimates the person's ability as higher than previously estimated. It then finds and presents a question that matches that higher ability. (If the first question is answered incorrectly, the opposite sequence occurs.) The test then presents the second question and waits for the answer. After the answer is given it scores the second question. If correct, it re-estimates the person's ability as higher still; if incorrect, it re-estimates the ability as lower. It then searches for a third question to match the new ability estimate. This process continues with the test gradually locating the person's competence level. The score that serves as an estimate of competence gets more accurate with each question given. The test ends when the accuracy of that estimate reaches a statistically acceptable level (or when a maximum number of items have been presented). Figure 2 shows the estimation of a person's competence after each of 10 test questions. Notice that how the ability is estimated lower after questions are answered incorrectly (Questions 3, 6, 8 and 10). The dotted vertical lines indicate the amount of error associated with the ability estimates (and correspondingly, the degree of confidence in the score). As more questions are presented and answered this error amount decreases (Certification and Skills Assessment Group, 2004).

Figure 2. A typical pattern for a CAT.

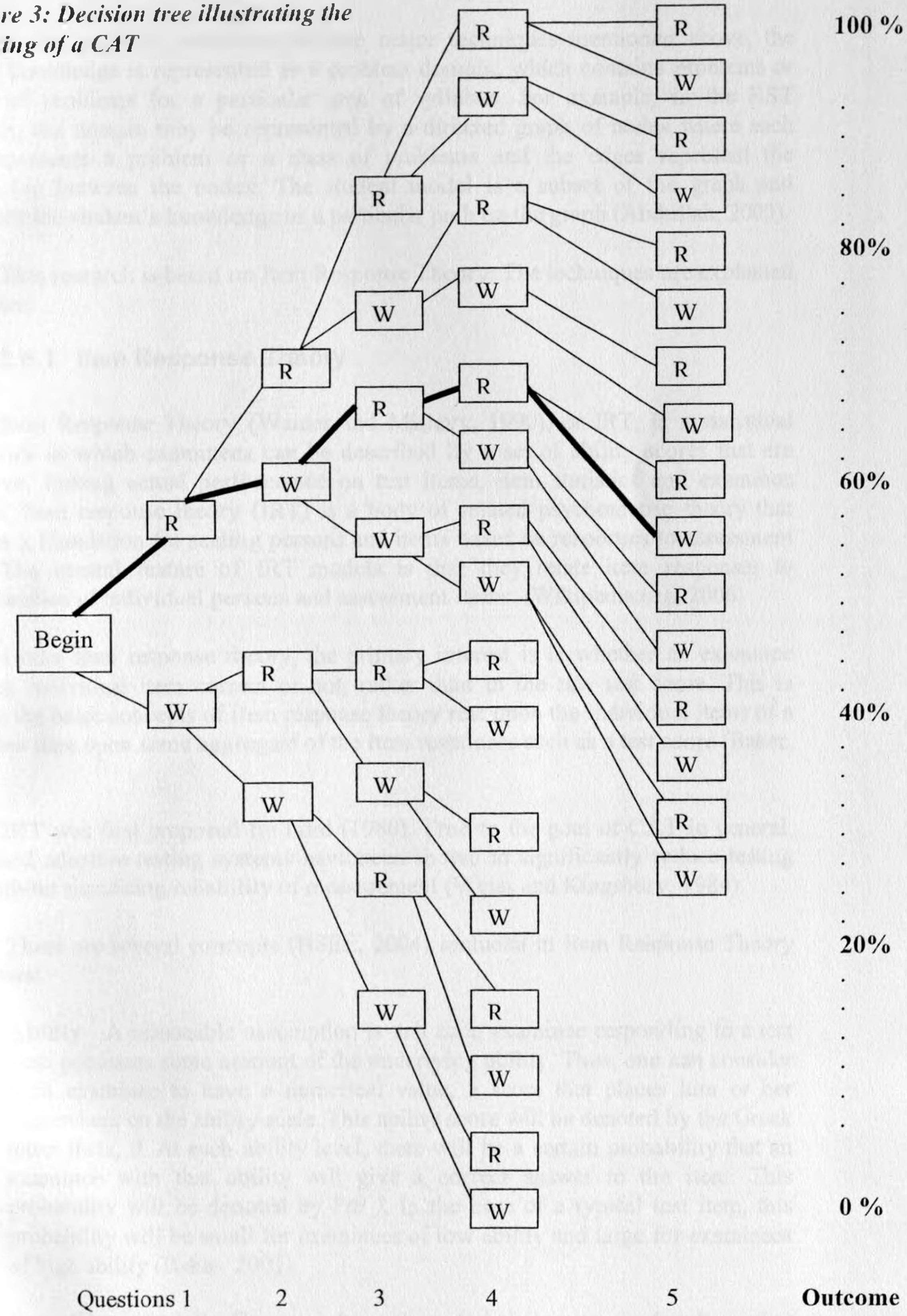


The CAT usually ends when the amount of measurement error around the ability estimate reaches an acceptable level. Low levels of measurement error are required for high-stakes certification tests and indicate that the test would likely produce a similar score if re-administered right away. Because it is unclear exactly when the test will end, a CAT usually presents a variable number of questions, and minimum and maximum numbers of questions are typically set.

In a CAT it is possible that a person with less competence is able to answer the same number of questions correctly as a more able person. Comparing the questions answered correctly for both persons would reveal that the higher-ability person was able to answer more difficult questions correctly. Therefore he or she should receive a higher score. And that is exactly what happens. The score is not based on the number of questions answered correctly, but instead it is derived from the level of difficulty of the questions answered correctly.

A decision tree (Figure 3) is used below (Schumacker, 2004) to illustrate how the CAT works:

Figure 3: Decision tree illustrating the working of a CAT



There are two major techniques that explain the logic of computerized adaptive testing. These are:

1. Item Response Theory or IRT (Wainer and Mislevy, 1990) and

2. Knowledge Space Theory or KST (Falmagne *et al.*, 1990).

In the approach based on the two major techniques mentioned above, the domain knowledge is represented as a problem domain, which contains problems or classes of problems for a particular area of syllabus. For example, in the KST approach, the domain may be represented by a directed graph of nodes where each node represents a problem or a class of problems and the edges represent the relationship between the nodes. The student model is a subset of the graph and represents the student's knowledge as a particular path on the graph (Abdullah, 2003).

This research is based on Item Response Theory. The techniques are explained as follows:

2.6.1 Item Response Theory

Item Response Theory (Wainer and Mislevy, 1990), or IRT, is a statistical framework in which examinees can be described by a set of ability scores that are predictive, linking actual performance on test items, item statistics and examinee abilities. Item response theory (IRT) is a body of related psychometric theory that provides a foundation for scaling persons and items based on responses to assessment items. The central feature of IRT models is that they relate item responses to characteristics of individual persons and assessment items. (Wikipedia.org, 2006)

Under item response theory, the primary interest is in whether an examinee got each individual item correct or not, rather than in the raw test score. This is because the basic concepts of item response theory rest upon the individual items of a test rather than upon some aggregate of the item responses such as a test score (Baker, 2001).

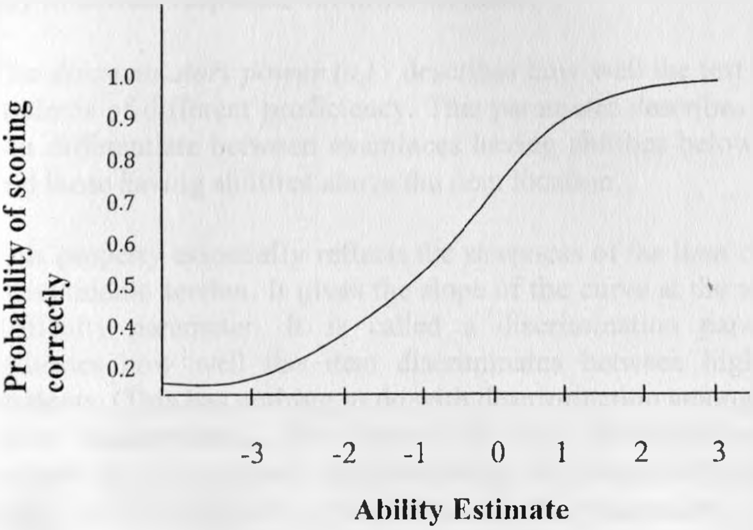
IRT was first proposed by Lord (1980). True to the goal of CAT in general, IRT-based adaptive testing systems have been shown to significantly reduce testing time without sacrificing reliability of measurement (Weiss and Kingsbury, 1984).

There are several concepts (HSEE, 2004) included in Item Response Theory as follows:

a) **Ability** - A reasonable assumption is that each examinee responding to a test item possesses some amount of the underlying ability. Thus, one can consider each examinee to have a numerical value, a score that places him or her somewhere on the ability scale. This ability score will be denoted by the Greek letter theta, θ. At each ability level, there will be a certain probability that an examinee with that ability will give a correct answer to the item. This probability will be denoted by $P(\theta)$. In the case of a typical test item, this probability will be small for examinees of low ability and large for examinees of high ability (Baker, 2001).

b) **Item Characteristic Curves** - Item characteristic curves are functions that predict the probability of passing a given item for all students at a given level of ability. If one plotted $P(\theta)$ as a function of ability, the result would be a smooth S-shaped curve. This is as shown in Figure 4. The probability of correct response is near zero at the lowest levels of ability. It increases until at the highest levels of ability, the probability of correct response approaches 1. This S-shaped curve describes the relationship between the probability of correct response to an item and the ability scale. In item response theory, it is known as the item characteristic curve. Each item in a test will have its own item characteristic curve.

Figure 4: Item Characteristic Curve



In IRT models, this probability depends on the relative difference between the item's difficulty and the examinee's ability. Initially, neither item difficulty nor examinee ability are known, so ability is placed on an arbitrary scale and item difficulties are "scaled" (estimated) relative to these abilities.

The Item characteristic forms the basic building block of Item Response Theory (Baker, 2001).

c) **Test Characteristic Curves (TCC)**. Test characteristic curves are similar to item characteristic curves except that what is predicted is the total number of items in a whole test that an examinee at a given ability level will answer correctly rather than the probability of answering an individual item correctly. The expected number correct is simply the sum of the probabilities of answering each of the individual items correctly, so TCCs are computed by summing the ICCs. For example, for a three-item test, a given examinee's probability of answering the first item correctly was .25, the probability for the second item was .50 and the probability for the third item was again .25. This examinee's expected or average score would be 1.0 (.25 + .5 + .25). Note that the estimated scores are averages and do not have to be whole numbers.

d) **Conditional Standard Errors.** In the preceding example, an examinee of a given ability had an expected score of 1.0 on a three-item test. This does not mean that the examinee will get exactly one item correct every time. Some of the time (about 28%) he or she will miss all of the items and some of the time (about 3%) he or she will answer all of the items correctly. The "standard error of measurement" provides an indication of how often an examinee's actual score on a particular test form will differ significantly from their expected score (across all parallel forms).

This research will use a three parameter model for the Item Characteristic Curve. This model provides a mathematical equation for the relation of the probability of correct response to ability. Under the three parameter model, each item is associated with three parameters that capture important differences in the relationship of ability to the probability of correct responses for different items.

- The *discriminatory power* (a_i) - describes how well the test item discriminates students of different proficiency. This parameter describes how well an item can differentiate between examinees having abilities below the item location and those having abilities above the item location.

  This property essentially reflects the steepness of the item characteristic curve in its middle section. It gives the slope of the curve at the ability given by the difficulty parameter. It is called a discrimination parameter because it indicates how well the item discriminates between high and low ability students. (This has nothing to do with discrimination among students based on other characteristics.) For items with high discrimination parameters, the probability of a correct response drops off (increases) sharply for students below (above) the ability level defined by the b-parameter. (HSEE, 2004)

  The value of the discrimination parameter ranges from -2.80 to +2.80

- The *difficulty level* (b_i) - describes how difficult an item is. Under item response theory, the difficulty of an item describes where the item functions along the ability scale. For example, an easy item functions among the low-ability examinees and a hard item functions among the high-ability examinees; thus, difficulty is a location index.

  The value of the difficulty parameter ranges from -3.00 to +3.00

- The *guessing factor* (c_i) - the probability that a student can answer the item correctly by guessing. It is also is the probability of answering correctly for students with no ability at all.

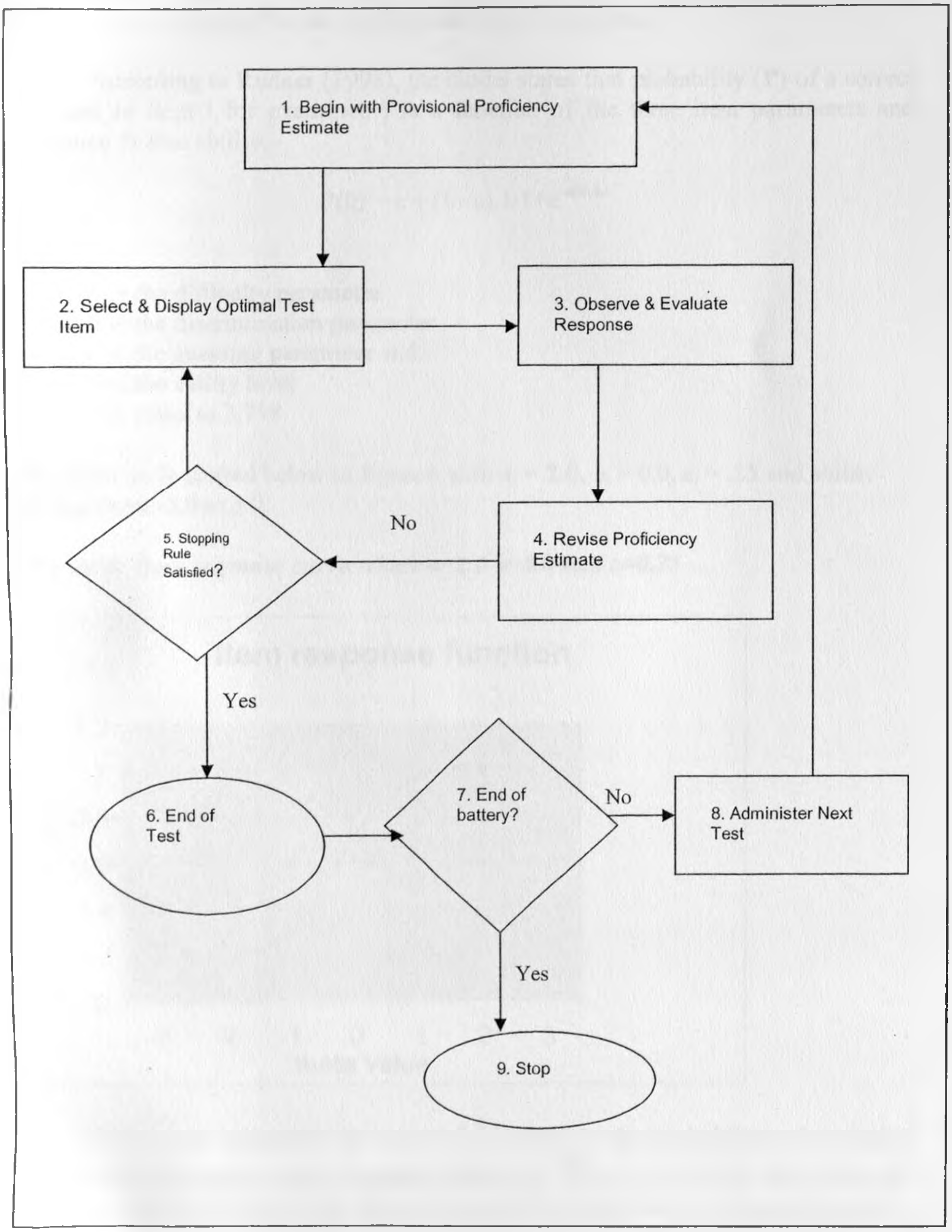
  The value of the guessing parameter ranges from 0 to 0.35

To ensure that it best fit the current student population to be tested on, an item pool must undergo content-balancing. Content-balancing is used to ensure no content area is over-tested or under-tested. The test specifications for a mathematics computation test, for example, may call for certain percentages of items to be drawn from addition, subtraction, multiplication and division (Rudner, 1998).

2.6.1.1 The Problem Progression Strategy

The problem progression or adaptive testing algorithm in IRT (Thissen and Mislevy, 1990) is given in Figure 5.

Figure 5: A Flowchart describing an Adaptive Test


1. Begin with Provisional Proficiency Estimate
2. Select & Display Optimal Test Item
3. Observe & Evaluate Response
4. Revise Proficiency Estimate
No
5. Stopping Rule Satisfied?
Yes
6. End of Test
7. End of battery?
No
8. Administer Next Test
Yes
9. Stop

At the start of the test, the algorithm has an initial provisional proficiency estimate of the student and this is denoted by θ. This specifies an initial item which is selected from the item pool. The selected item is aimed at providing the 'most information' about the student. Once the student provides an answer for the selected item, a new proficiency estimate, θ', is calculated together with its confidence level. It is based on whether the student's answer is correct or incorrect, the old θ and the item parameters. If the confidence level of θ' reaches a designated level, or when some predetermined number of items has been administered, the test terminates. Otherwise another item is selected for the student, and the test continues.

According to Rudner (1998), the model states that probability (**P**) of a correct response to item i for examinee j is a function of the three item parameters and examinee j's true ability.

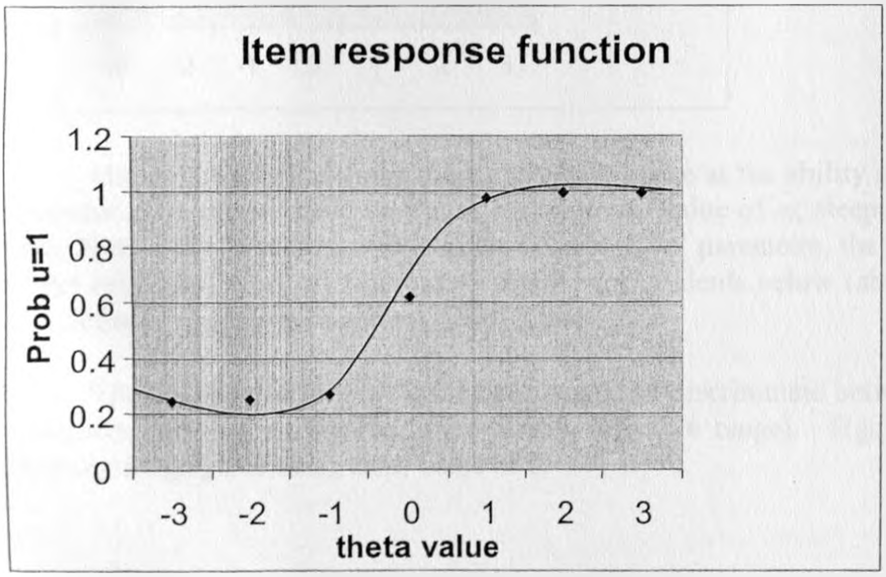
$$P(\theta) = c + (1 - c)\ 1/1 + e^{-a(\theta - b)}$$

Where:

b is the difficulty parameter
a is the discrimination parameter
c is the guessing parameter and
θ is the ability level
e is equal to 2.718

This function is plotted below in figure 6 with $a_i = 2.0$, $b_i = 0.0$, $c_i = .25$ and ability varying from -3.0 to 3.0.

Figure 6: Item response curve where a=2.0 b=0.0 and c=0.25



The lower asymptote is at c_i = 0.25. This is the probability of a correct response for examinees with very little ability (e.g. θ_j = -2.0 or -2.6). The curve has an upper asymptote at 1.0; high ability examinees are very likely to respond correctly.

The b_i parameter defines the location of the curve's inflection point along the theta scale. Lower values of b_i (easy questions) will shift the curve to the left; higher (harder questions) to the right. The b_i does not affect the shape of the curve. For example, when b_i is -2 and 2, the curve will be as in Figure 7 and Figure 8 respectively.

Figure 7 Item response curve where 'difficulty parameter' bi = -2

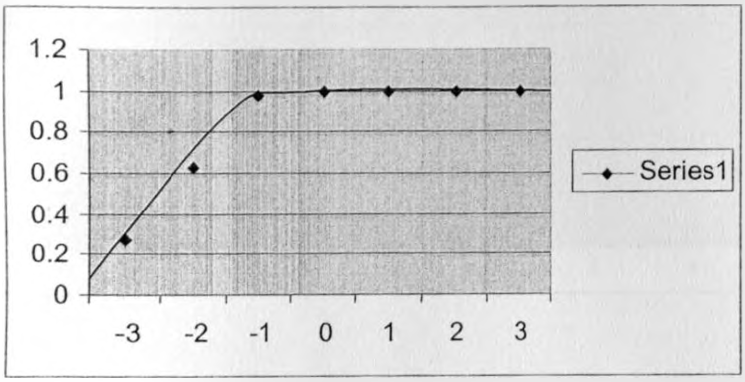
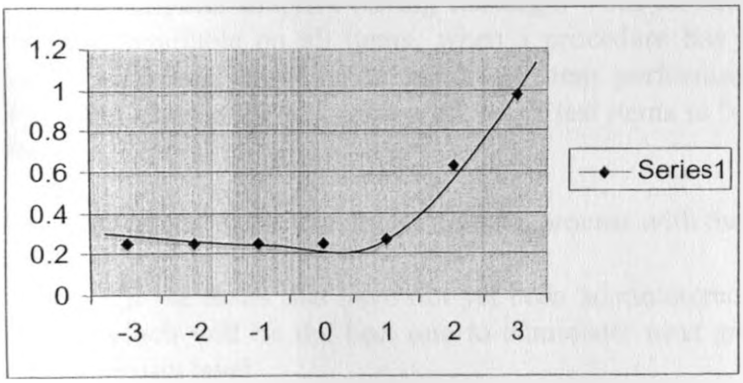


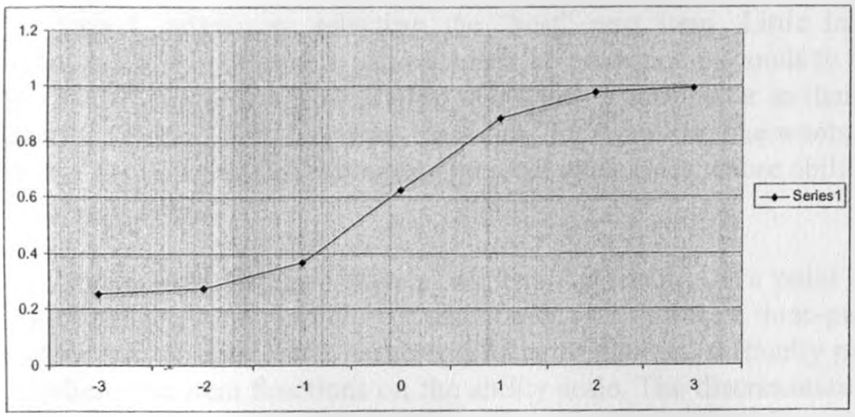
Figure 8 Item response curve where 'difficulty parameter' bi = 2



The a_i parameter defines the slope of the curve at the ability given by the 'b' parameter. The curve would be flatter with a lower value of a_i; steeper with a higher value. Note that when the curve is steep, i.e. a high 'a' parameter, the probability of a correct response drops off (increases) sharply for students below (above) the ability level defined by the b-parameter. (HSEE, 2004)

Thus a_i denotes how well the item is able to discriminate between examinees of slightly different ability (within a narrow effective range). Figure 9 shows the effect on the graph of using an a_i value of 1.

Figure 9 – Effect of reducing the discrimination parameter ai



2.6.1.2 The Logic of Computer Adaptive Testing with IRT

Computer adaptive testing can begin when an item bank exists with IRT item statistics available on all items, when a procedure has been selected for obtaining ability estimates based upon candidate item performance, and when there is an algorithm chosen for sequencing the set of test items to be administered to candidates (Rudner, 1998)

The CAT algorithm is usually an iterative process with the following steps:

1. All the items that have not yet been administered are evaluated to determine which will be the best one to administer next given the currently estimated ability level
2. The "best" next item is administered and the examinee responds
3. A new ability estimate is computed based on the responses to all of the administered items.
4. Steps 1 through 3 are repeated until a stopping criterion is met.

Several different methods can be used to compute the statistics needed in each of these three steps. Lord (1980) has shown how this can be accomplished using Item Response Theory.

Treating item parameters as given, the ability estimate is the value of theta that best fits the model. When the examinee is given a sufficient number of items, the initial estimate of ability should not have a major effect on the final estimate of ability. The tailoring process will quickly result in the administration of reasonably targeted items. The stopping criterion could be as follows:

- Time,
- Number of items administered,
- Change in ability estimate,
- Content coverage,
- A precision indicator such as the standard error,

- Or a combination of factors.

Step 1 references selecting the "best" next item. Little information about an examinee's ability level is gained when the examinee responds to an item that is much too easy or much too hard. Rather one wants to administer an item whose difficulty is closely targeted to the examinee's ability. Furthermore, one wants to give an item that does a good job of discriminating between examinees whose ability levels are close to the target level.

Under item response theory, an item's difficulty is a point on the ability scale where the probability of correct response is $(1 + c)/2$ for a three-parameter model. The proper way to interpret a numerical value of the item difficulty parameter is in terms of where the item functions on the ability scale. The discrimination parameter can be used to add meaning to this interpretation. The slope of the item characteristic curve is at a maximum at an ability level corresponding to the item difficulty. Thus, the item is doing its best in distinguishing between examinees in the neighborhood of this ability level. Because of this, one can speak of the item functioning at this ability level (Baker 2001).

The standard error associated with the ability estimate is calculated by first determining the amount of information the set of items administered to the candidate provides at the candidate's ability level--this is easily obtained by summing the values of the item information functions at the candidate's ability level to obtain the test information. Second, the test information is inserted in the formula below to obtain the standard error:

$$SE(\dot{\theta}) = \frac{1}{\sqrt{I_{rr}}}$$

Thus, the standard error for individuals can be obtained as a by product of computing an estimate of an examinees ability.

2.6.1.3 Reliability and standard error

In classical measurement, the standard error of measurement is a key concept and is used in describing the level of precision of true score estimates. With a test reliability of 0.90, the standard error of measurement for the test is about 0.33 of the standard deviation of examinee test scores. In item response theory-based measurement, and when ability scores are scaled to a mean of zero and a standard deviation of one (which is common), this level of reliability corresponds to a standard error of about 0.33 and test information of about 10. Thus, it is common in practice, to design CATs so that the standard errors are about 0.33 or smaller (or correspondingly, test information exceeds 10--recall that if test information is 10, the corresponding standard error is 0.33) (Abdulla, 2003).

2.6.2 Knowledge Space Theory

Another strand of development in adaptive testing is based the Knowledge Space Theory, KST for short, (Doignon and Falmagne, 1985), (Falmagne *et al.*, 1990).

2.6.2.1 Describing the Domain

As with IRT-based systems, the domain is defined by a collection of test items. Here, a test item can represent not only a problem but also a class of problems, and the relationship between the test items are explicitly stated through prerequisite relationships. Unlike IRT based systems which are one-dimensional in that only one student trait (such as mastery of a topic) can be measured at one time, adaptive testing systems based on the KST can measure more than one trait and can represent a set of skills or problems mastered by the student. This set is known as a *knowledge state*. The structure of the domain takes the form of a knowledge space which represents the area of the syllabus to be tested; the following example as illustrated by Abdulla (2003) will explain the notion of a knowledge space.

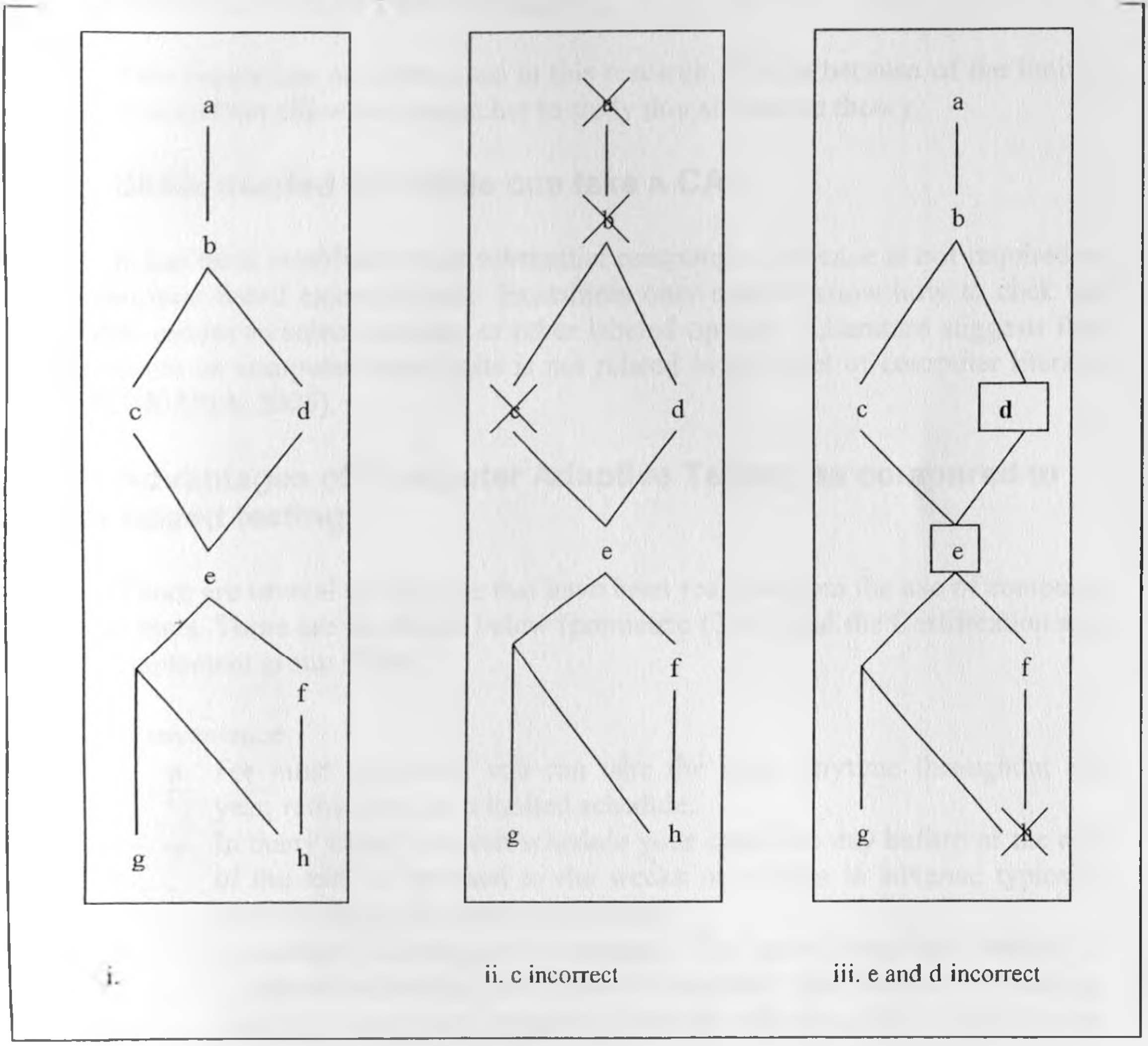
A body of knowledge is characterised by a set of items called the domain, say {a,b,c,d}. This gives rise to 16 possible knowledge states:
{}, {a}, {b}, {c}, {d},
{a,b}, {a,c}, {a,d}, {b,c}, {b,d}, {c,d},
{a,b,c}, {a,b,d}, {a,c,d}, {b,c,d}, {a,b,c,d}
A student's *knowledge state* is defined as the set of items in the domain that the student is capable of solving. For example, if a student has the knowledge state {a,b,d}, this means that he can solve items a, b and d. Not all possible subsets of the domain are *feasible* knowledge states. For example, if the student can solve item d, and that it is inferred that the student can also solve item a, then any knowledge state that contains item d must also contain item a.

This means that knowledge states {d}, {b,d}, {c,d} and {b,c,d} are not feasible. This means that a feasible knowledge state is one which contains not only all the items that the student has demonstrated mastery of, but also the items which can be inferred. In effect, a feasible knowledge state describes the *prerequisite relationships* between items. For example, in the knowledge state {a,d}, item a is a prerequisite of item d. The collection of all feasible knowledge states is called the *knowledge structure*. The knowledge structure must also contain the *null state {}* which corresponds to the student who cannot solve any item, and the *domain* which corresponds to the student who can solve or master all items.

2.6.2.2 The Problem Progression Strategy

The knowledge space will serve as the core of a knowledge assessment system. Once the domain is represented as a knowledge space, the adaptive testing strategy is then to locate as efficiently and as accurately as possible, a student's knowledge state, which is a point in the knowledge space. Problem progression works like this. An item is selected, usually; some predictive probabilistic model is used to determine the sequence of items in a test (Villano, 1992). If a student has answered the item correctly (incorrectly), it can be inferred that he can (cannot) answer a prerequisite (parent) item and will thus not be asked to solve the latter. An item is a problem from a pool of similar problems, for example, problems which 'add two fractions of common denominators'. Inferences progressively prune the search space and at the end of the test, a student's knowledge of the subject domain is represented by a knowledge state. In an example given by Dowling and Kaluscha (1995), a knowledge space is represented as an AND/OR graph, as shown in Figure 10. The nodes represent problems and the arcs state the prerequisite relationship between the nodes.

Figure 10: Illustration of Prerequisite Relationships and the Assessment



Item *b* represents an AND node. This means that if item b is answered correctly (mastered), then it can be inferred that both its prerequisites *c* and *d* are mastered. Item *e* represents an OR node. This means that if item *e* is mastered, it infers that all the test items in at least one of its prerequisite sub graphs must be known. These sub graphs are the one with the nodes *g* and *h*, or the one with the nodes *f* and *h*.

Suppose item *c* is chosen and presented to the student and the student provides an incorrect answer. From this incorrect response, it can be inferred that the student will not be able to solve problems *a* and *b* either, as *c is* a prerequisite of *a* and *b*. Problems *a* and *b* are considered more difficult than *c* and are thus not presented to the student. The next problem to be selected will be one which is not *a* or *b*, nor one which has *a, b* or *c* as its prerequisites.

Suppose the next problem chosen is *e and* it is answered correctly. This infers a correct answer to problem *h*, as *h* is a member of both sub graphs of the OR node, and *h* will be removed from the list of candidate problems to be presented. Suppose *d* is presented next and is answered correctly, and *g* is presented next and is answered incorrectly. This infers a correct answer to *f*. The test stops and the student's knowledge state is inferred as *{d, e, f, h}* which was reached with only four questions being presented out of a maximum of eight.

This theory has not been used in this research. This is because of the limited time that would not allow the researcher to study this alternative theory.

2.7 Skills needed to enable one take a CAT

It has been established that substantial computer experience is not required to take computer-based examinations. Examinees only need to know how to click the computer mouse to select answers or other labeled options. Literature suggests that performance on computer-based tests is not related to the level of computer literacy (COMLEX-USA, 2005).

2.8 Advantages of Computer Adaptive Testing as compared to paper based testing

There are several advantages that have been realized from the use of computer adaptive tests. These are explained below (prometric (2005) and the Certification and Skills Assessment group (2004),

- Convenience
  - For most programs, you can take the exam anytime throughout the year, rather than on a limited schedule.
  - In many cases, you can schedule your exam the day before or the day of the test, as opposed to the weeks or months in advance typically required for paper-and-pencil testing.
  - Immediate Scoring and Feedback. The most important benefit of computerized testing is to know immediately the result of the testing. Getting a score and a pass/fail decision right away for a certification candidate is very important.
  - Convenient Individualized Administration. Computerized tests can be administered at times and locations more convenient to the test taker.

- Efficiency
  - Traditional exam methods such as multiple choices, combined with techniques like computer adaptive testing (CAT), produce a more effective exam. The exam is just as reliable, but often takes less time to complete and provides more accurate results. CATs are NOT more accurate at determining pass/fail certification decisions than fixed-length computerized exams—they are equally accurate. But they are more efficient. That is, the CAT determines the pass/fail decision with fewer test questions and less testing time.
  - Computer-based test results are immediately available, compared to a four- to six-week turnaround typical for traditional tests. The process of generating results for a paper based approach has untimely feedbacks that may take months to generate results. (Xplanations.com)

- Results can be published electronically.

- Security
  - The computer can be programmed to select from numerous types of exams, randomize the order of questions, or devise exams instantaneously by selecting questions from a range of subjects.
  - Computer-based testing eliminates the need for exam papers and answer sheets, providing an extra level of security.

- Comfort
  - Self-paced lessons show you how to take the test, ensuring that even without computer experience you are comfortable with the exam process.

- Accuracy
  - Computer-based assessment permits a diverse range of question types. Exams can include simulations of real-life situations, graphics, voice-activated responses and split screens, which can simultaneously display passages of text and questions
  - Improved Accuracy for Scores for High and Low Ability Test Takers. An adaptive test is as accurate as any other test at determining a pass/fail decision; however, for providing a score for high and low ability individuals, it is actually more accurate. Because it can present many items at any ability level, it can compute an accurate score. Traditional tests are less accurate at these extremes.
- Unbiased Scoring.
  - Computers score everyone the same way and do not consider factors irrelevant to the score, such as examinee gender or culture.
- New Question Types
  - (point-and-click, drag and drop, and simulations). New types of questions improve the ability to measure the important skills.

- Less Expensive.
  - Although it is probably not true today, the increased use of computers in test development and delivery promises to reduce the testing costs in the future for the test developer, test publisher, test user and examinee. There are high costs associated with administrative overhead and use of multiple resources to duplicate, administer, collect, collate, code, score and analyze the data. (xplanations.com 2004)

2.9 What are the disadvantages/drawbacks as compared to paper based testing?

Apart from the advantages that have bee associated with computer adaptive tests, the system has drawbacks that cannot be overlooked. These are as follows (Grist, 1989):

- CATs should not be used for some subjects and skills. Most CATs are based on an item-response theory model, which assumes that all the information needed in selecting items can be summarized in one to three parameters that describe the item's difficulty for students who have different abilities. Many tests cover a number of different skills or topics, however. Specifications for traditional tests seek to ensure an even range across skills or topics. Most common CAT strategies do not accommodate such additional considerations.

- Hardware limitations further restrict the types of items that can be administered by computer. Items involving detailed art work and graphs or extensive reading passages, for example, are hard to present using the types of computers found in most schools.
- Another limitation of CATs stems from the need for careful item calibration. Since each student takes a set of items, comparable scores depend heavily on precise estimates of item characteristics. Therefore, relatively large samples must be used. A minimal number in a sample is 1,000 students; 2,000 is more common. Such sample size requirements are prohibitive for most locally developed tests.
- Finally, for CATs to be manageable a facility must have enough computers for a large number of students and the students must be at least partially computer-literate. While the number of computers in schools continues to grow, many schools simply do not have the resources to use CATs as a standard practice.

2.10 Places where CATs has been used

Today, with the common use of computers in test delivery, adaptive testing has become more popular and is called computerized adaptive testing or CAT. The computer can make the necessary calculations needed to estimate a person's proficiency and to choose the questions to present. Several well-known high-stakes testing programs have adopted adaptive testing as their current and future method to test.

1. The Educational Testing Service, the world's largest testing organization, published the Graduate Record Exam (GRE) as an adaptive test in 1993, and has been slowly reducing its use of the paper version of the test.
2. The Nursing Boards converted completely from paper-based testing to a computerized adaptive test in 1994. Over 400,000 exams for Registered Nurses are given each year.
3. In the information technology industry, Novell successfully introduced CATs into its certification program in 1991. Over 1,000,000 Novell adaptive tests have been given (Certification and Skills Assessment Group, 2004).
4. For the past decade, the U.S. military has pioneered basic and applied research in CATs. One step in this research program is the development of a computerized version of the Armed Services Vocational Aptitude Battery (ASVAB), headed by the Naval Personnel Research and Development Center in San Diego. Administered to roughly a half million applicants each year, the paper-and-pencil version of the ASVAB takes three hours to complete while the experimental CAT version takes about 90 minutes. With the computerized version, an examinee's qualifying scores can be immediately compared with requirements for all available positions (Grist, 1989).

2.11 Summary of Literature Review

The researcher has identified several issues from the research:

- Paper based testing, though widely used in Kenya, has shortcomings and may not accurately determine the best candidates for a job position when the students are subjected to a written aptitude test. When students of various abilities are subjected to the same test, then it can result to careless slips either because the test is too easy or too hard for particular individuals.

- Computer adaptive testing can be used to determine the ability of a student with a commendable level of accuracy.

- The research problem needs to be further studied so that we can perfect grading systems and increase the chances of getting the best candidates for a job position. Currently, I feel that this is not the case.

- This research will contribute to the body of knowledge in that such a system can be customized for Kenyan universities during pre-selection of candidates for university courses. In certain cases, a student may perform poorly in KCSE because of other factors other that lack of ability. Such students can be subjected to further testing to ascertain their ability in order to determine whether or not they are suited for a course.

- This study is the first to explore the potential for applying CAT in the assessment of students/candidates related ability for outcome measurement in examinations. Using a combination approach of Rasch model, together with new developments in CAT software, I have been able to show that items can be calibrated onto a single metric and that they can be used to provide the basis of a CAT application which map on to the candidate's ability. In this way, a simple, precise estimate of the person's ability can be determined and, given the use of the Rasch model, one that is interval scaled.

In this study, I have also provided a way of ensuring test result accuracy as well as reduce issues of exam cheating. In CAT the items are chosen to match the estimated ability level (or aptitude level, etc.) of the current test-taker. If the test-taker succeeds on an item, a slightly more challenging item is presented next, and vice-versa. This technique usually quickly converges into sequence of items bracketing, and converging on, the test-taker's effective ability level. This would greatly reduce the issues of exam cheating as each candidate will be presented with a different set of questions. It also ensures test result accuracy.

3. CHAPTER 3 - RESEARCH METHODOLOGY

3.1 Introduction

This chapter explains the methods that will be used in the research.

3.2 Research method and analysis

The researcher will use constructive research method. This type of approach demands a form of validation that doesn't need to be quite as empirically based as in other types of research (Wikipedia, 2007).

In this approach, it will involve the development of an adaptive system prototype. A prototype is a small scale, incomplete but working model of a desired system. It uses the Rapid Application Development (RAD) technique to accelerate the system development process. The system will then be evaluated analytically by performing some benchmark tests with the prototype. The technique was adopted for the following reasons:

i. To save on development time – The system is not complete but only contains the critical components

ii. Prototypes are an active, rather than a passive model that can be seen and experienced

iii. Testing and training is a natural by-product of the prototype approach

iv. Prototyping can increase creativity because it allows for quicker user feedback which can lead to better solutions

3.3 Conceptual and Theoretical Framework

How can institutions that use aptitude tests in interviews, ensure that they do not lock out potential candidates who 'fail' these tests? Aptitude tests in Kenya are paper based and consist of 30 to 100 questions that are to be tackled within a limited time. The more questions answered correctly, the more the likelihood of passing the interview. However, many good candidates may lose out because of anxiety and careless slips rather than their lack of ability. This is largely caused by the limited time and the large number of questions that may not be appropriate for the candidate.

A testing system should and must be able to get the best candidates. The researcher has used Item Response Theory (IRT) as the theoretical framework in demonstrating how an adaptive testing system can be created. One of the advantages of the adaptive testing system over the conventional paper based testing is that the student will be presented questions based on his estimated ability. The ability estimate will be calculated after every question is answered and appropriate questions will be selected based on the ability estimate. The test will stop when the ability estimate is consistent. The test length will therefore be reduced and the result will demonstrate what kind of questions the student answered together with the final ability estimate. He will not have to answer all the questions as in the case of a paper based test. This will reduce anxiety and careless slips caused by questions either being too hard or too easy.

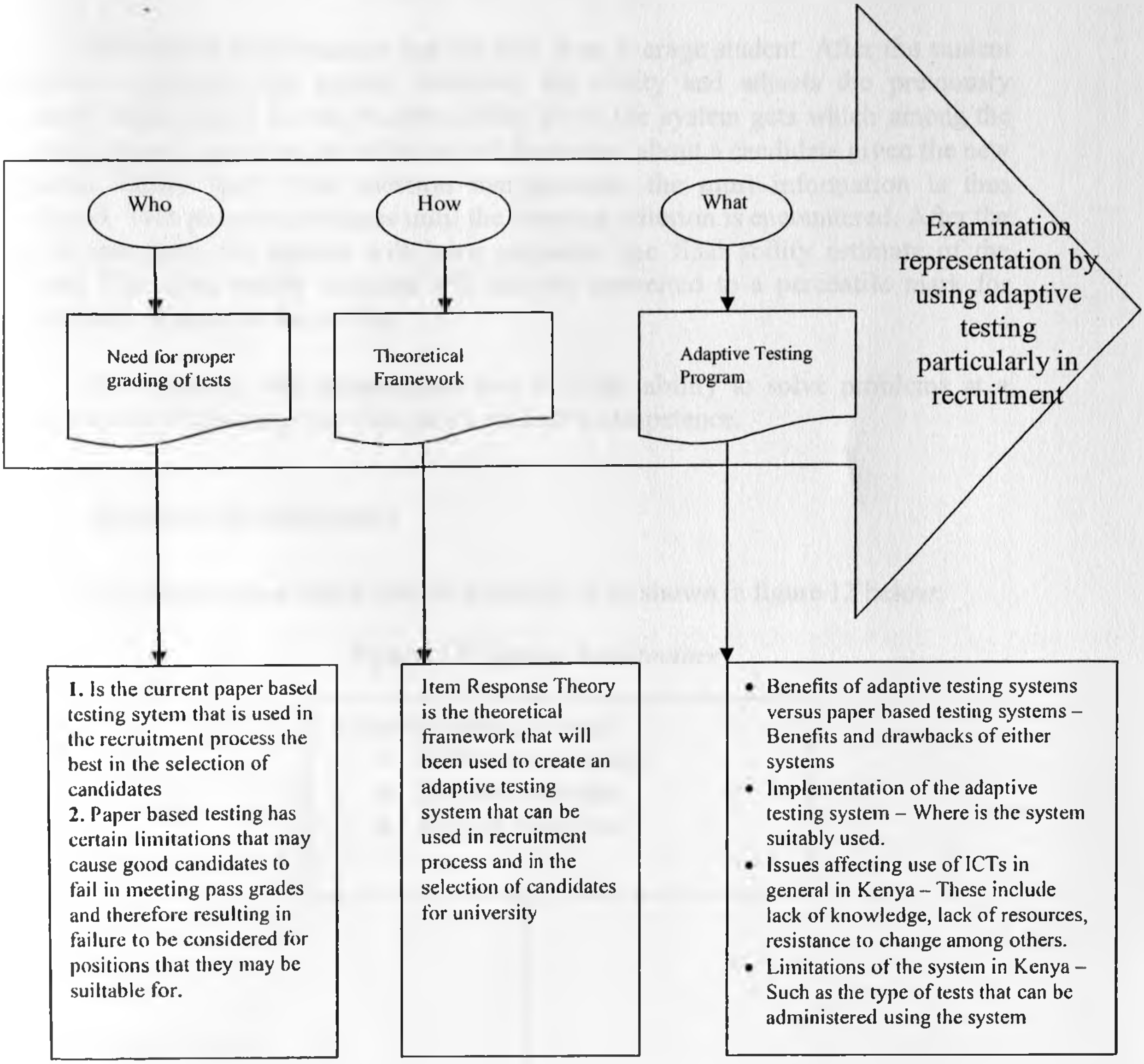
The researcher will develop a model of an adaptive system to demonstrate the workings. However, it is understood that no one test is perfect and without errors. This research proposes that this is one of the improvements of a component in the whole recruitment and selection process.

The benefits of an adaptive testing system such as convenience, efficiency, security, comfort and accuracy in computing results justify its inclusion in testing systems. In addition, the benefits derived from proper selection of candidates far outweigh the drawbacks of the system.

However, only after examining the issues of the current system of testing (paper based) versus the adaptive testing system can the potential of the adaptive testing system be drawn.

The conceptual framework is as shown in the figure 11 below.

Figure 11: Conceptual Framework for the Examination representation by using adaptive testing a case study of placement interviews


Who
How
What
Examination representation by using adaptive testing particularly in recruitment
Need for proper grading of tests
Theoretical Framework
Adaptive Testing Program

1. Is the current paper based testing sytem that is used in the recruitment process the best in the selection of candidates
2. Paper based testing has certain limitations that may cause good candidates to fail in meeting pass grades and therefore resulting in failure to be considered for positions that they may be suiltable for.

Item Response Theory is the theoretical framework that will been used to create an adaptive testing system that can be used in recruitment process and in the selection of candidates for university

- Benefits of adaptive testing systems versus paper based testing systems – Benefits and drawbacks of either systems
- Implementation of the adaptive testing system – Where is the system suitably used.
- Issues affecting use of ICTs in general in Kenya – These include lack of knowledge, lack of resources, resistance to change among others.
- Limitations of the system in Kenya – Such as the type of tests that can be administered using the system

4. CHAPTER 4. – DESIGN AND IMPLEMENTATION

4.1 Operation of the System

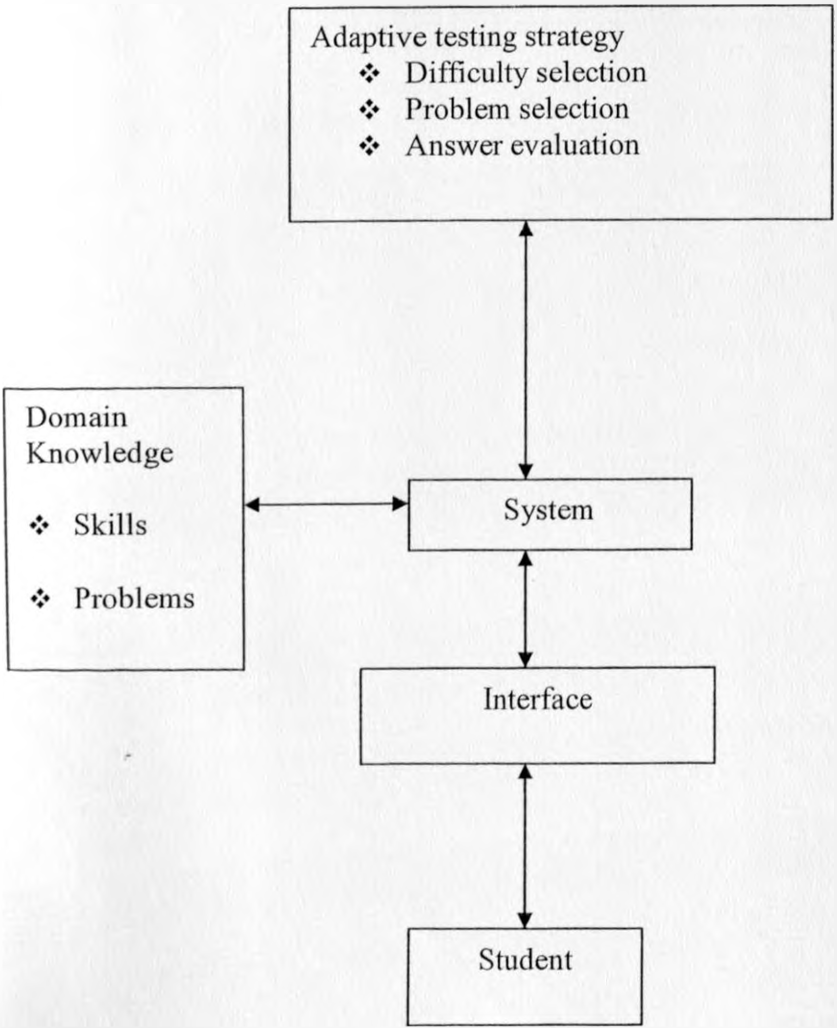
The system first assumes that the user is an average student. After the student answers a question, the system estimates his ability and adjusts the previously assumed ability level. Given the new ability level, the system gets which among the un-administered questions provides most information about a candidate given the new adjusted ability level. The question that provides the most information is thus displayed. This process continues until the stopping criterion is encountered. After the test is complete, the system will have computed the final ability estimate of the student. The final ability estimate will also be converted to a percentile mark for easier interpretation of the ability.

The research will demonstrate that it is the ability to solve problems at a certain level of difficulty that indicates a student's competence.

4.2 System Architecture

The Architecture which will be used will be as shown in figure 12 below:

Figure 12: System Architecture



The architecture components are defined below:

1. The **System** coordinates the CAT process
2. The **Difficulty selection** module determines the level of difficulty of the first problem to be selected
3. The **Problem selection** module selects a problem from the database of questions
4. **Interface** is for presenting the problem to the student.
5. The **answer evaluation** module compares the student's answer to that in the system.

The research will be centered on the components above only. It will not be involved in the construction of the tests which are assumed to be done by an expert and is beyond the scope of this research.

4.3 System Design

The working of the Architecture is shown in the level 0 and Level 1 Data flow diagrams as follows:

Figure 13: Computer Adaptive Testing System – Context Diagram (Level 0 DFD)

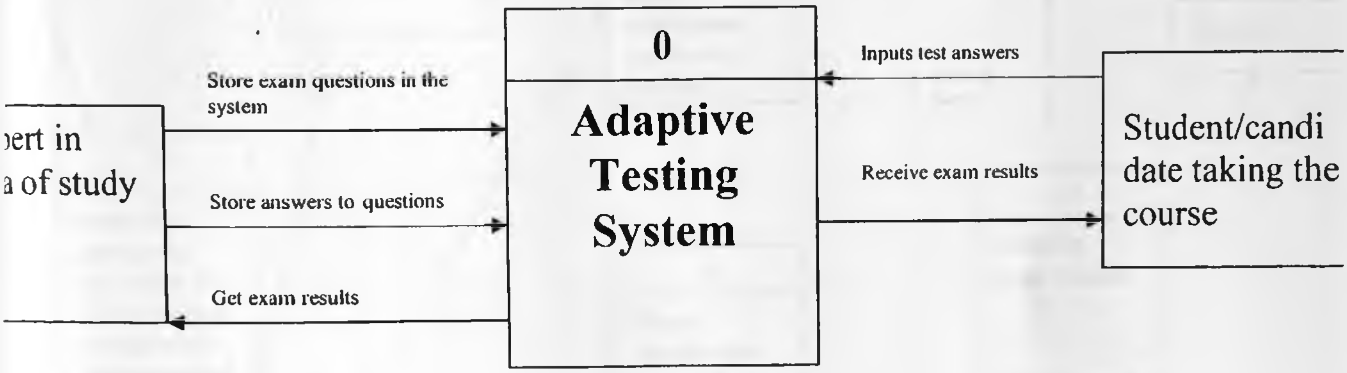
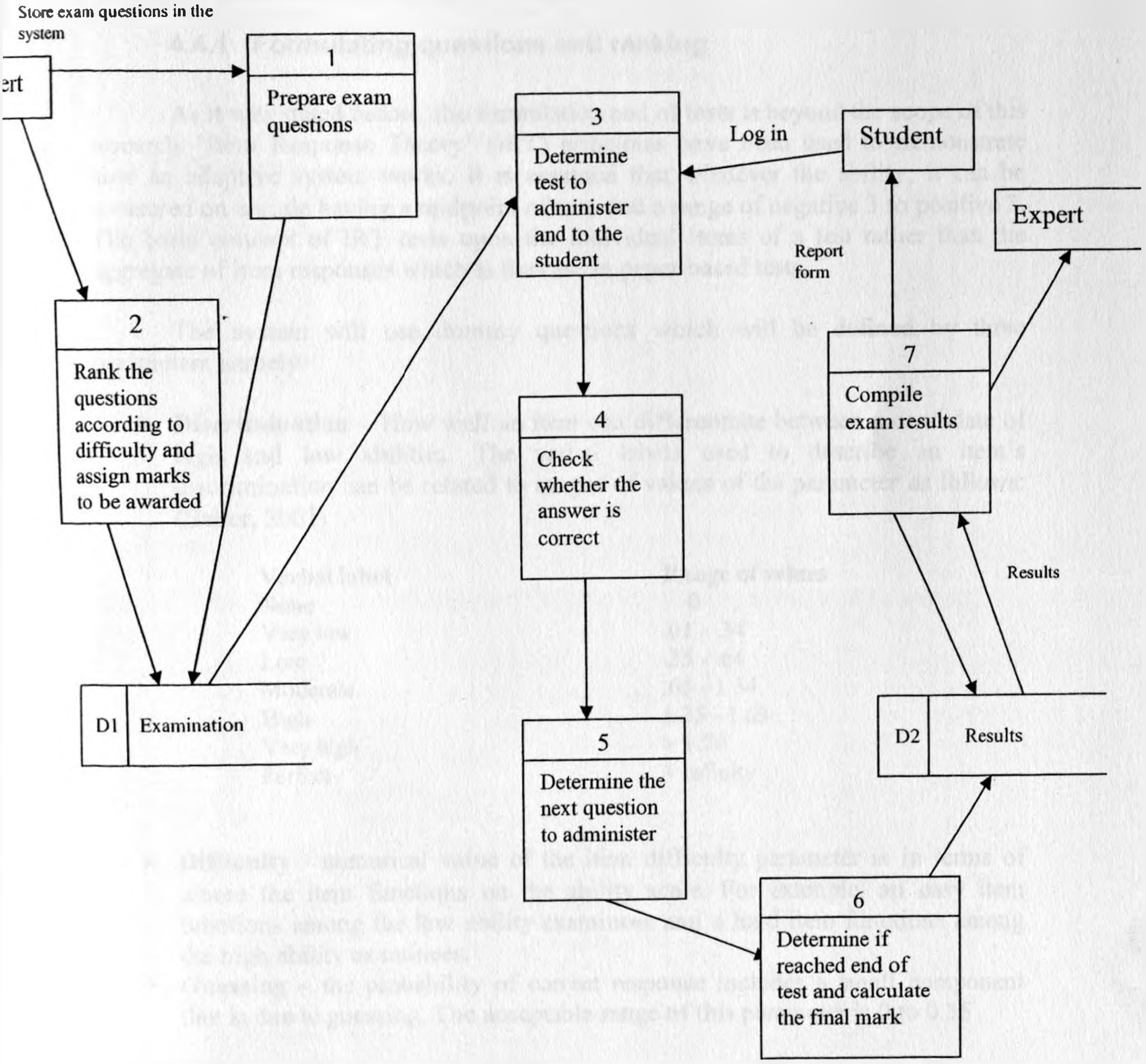


Figure 14: Computer Adaptive Testing System - Level 1 DFD


Store exam questions in the system
ert
1
Prepare exam questions
3
Determine test to administer and to the student
Log in
Student
Expert
Report form
2
Rank the questions according to difficulty and assign marks to be awarded
7
Compile exam results
4
Check whether the answer is correct
Results
D1
Examination
5
Determine the next question to administer
D2
Results
6
Determine if reached end of test and calculate the final mark

4.4 System Operation

The system has the following main functions:

4.4.1 Formulating questions and ranking

As it was stated before, the formulation and of tests is beyond the scope of this research. 'Item Response Theory' (IRT) principles have been used to demonstrate how an adaptive system works. It is assumed that whatever the ability, it can be measured on a scale having a midpoint of zero and a range of negative 3 to positive 3. The basic concept of IRT rests upon the individual items of a test rather than the aggregate of item responses which is the case in paper based tests.

The system will use dummy questions which will be defined by three parameters namely:

- **Discrimination** – How well an item can differentiate between a candidate of high and low abilities. The verbal labels used to describe an item's discrimination can be related to ranges of values of the parameter as follows: (Baker, 2001)

Verbal label	**Range of values**
None	0
Very low	.01 - .34
Low	.35 - .64
Moderate	.65 - 1.34
High	1.35 - 1.69
Very high	> 1.70
Perfect	+ infinity

- **Difficulty** - numerical value of the item difficulty parameter is in terms of where the item functions on the ability scale. For example, an easy item functions among the low ability examinees and a hard item functions among the high ability examinees.
- **Guessing** – the probability of correct response includes a small component that is due to guessing. The acceptable range of this parameter is 0 to 0.35

The system allows login by the Administrator/expert. They are given the rights to upload questions and also to input the parameters of discrimination, difficulty and guessing that are used in computing the ability of the student. In this system dummy questions have been used for illustration purposes. They are stored in the database for reference.

An important point to note is that the type of questions used is multiple choice questions. From a practical point of view, open ended questions are difficult to use as they are difficult to score in a reliable manner.

4.4.2 Displaying questions

The first time a student logs in to do a test, the system assumes that he is of average ability. A range of -3 to +3 has been used to represent ability. The midpoint of this scale is the value '0' and represents an average student. The algorithm to display questions is as shown below:

```
If first time examining candidate

Student has average ability = 0
Previous ability estimate = 0
Previous standard error = 0
Current standard error = 0

Get next question to administer

Calculate new ability estimate

Update summary information of items administered

Repeat process till stopping mechanism is arrived at
```

4.4.3 Adapting to the ability of the student

The system presents questions according to how the candidate answers the previous questions. The simplified code for this is as follows:

```
Get the probability of correct response with parameters of difficulty, discrimination
and guessing where
        Probability = 0
        logit = discrimination x (ability level – difficulty)
        exponent = math.Exp (-1 x logit)
        denominator = 1 + exponent
        result = 1/denominator
        Probability = guessing + (1 – guessing) x result

Get probabilities at different ability levels for the question by drawing a graph where
        Graph point = new graph point
        Probability = 0
        For (int I = -3; i<= 3; i++)

Generate information function for the question
        Info = Get information (I, question)
        point = new Graph point
        points.Add(point)
        return points

Get information for a question given the ability level and question parameters where:
        Info = 0
        p = Get probability (ability level, difficulty, discrimination, guessing)
        q = 1 – p
        q divide p = q / p
        disc squared = discrimination x discrimination
        1 – guessing$^2$ = math.Pow ((1 – guessing) 2) ;
        p – guessing$^2$ = math.Pow ((p – guessing) 2) ;
```

```
    info = disc squared x q divide p x (p – $guessing^2$) / (1 – $guessing^2$);
    return info
```

Get list of un administered questions

Get question that gives the best information about a candidate with a specified ability level (Adaptation is done here)

```
    Get list of un administered questions
    Calculate information given the ability level
    Return question
```

Get estimated ability level

```
    Get the answered question
    For each question,
        Get probability p
        q = 1 – p
        numerator = question.discrimination x (question. Answer – p) ;
        denominator = $question.discrimination^2$ x p x q ;
        new ability level = previous estimated ability – numerator /
        denominator)
        return new ability level
```

Summary

```
    List the questions answered
    Calculate probability of a correct response
    Calculate new ability level
    Return the ability level
```

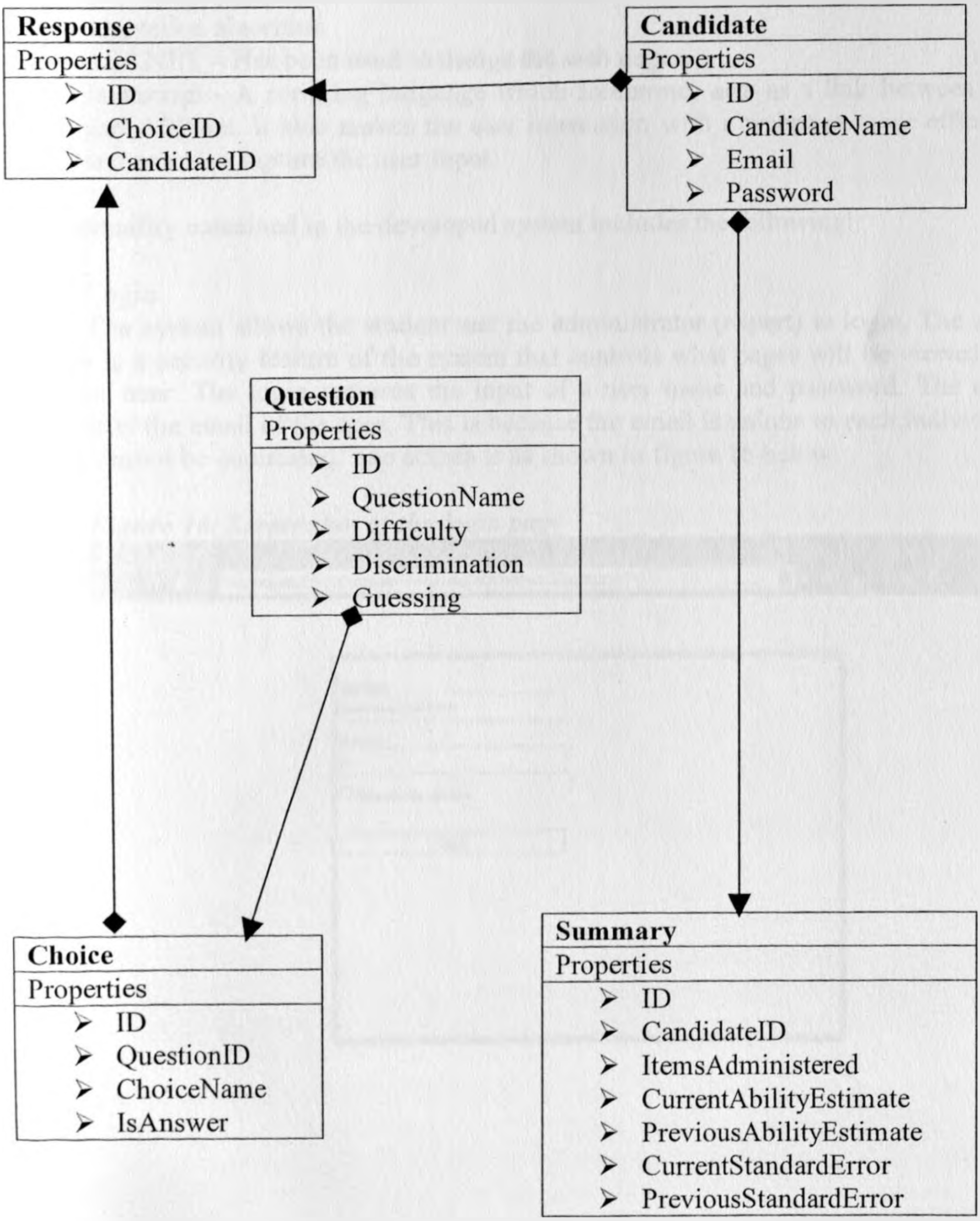
4.4.4 Reporting

After completion of the test, the candidate receives a report indicating his final mark. The expert also receives a report that shows the name of the student, which questions he answered, the responses, whether they are correct or not and the final ability estimate of the student

4.5 Database Structure

The database structure is as shown in Figure 15 below:

Figure 15: Database Structure



5. CHAPTER 5. – RESULTS

The Microsoft Windows platform, specifically Windows XP was used in the development of the program. The system has been developed using three languages namely:

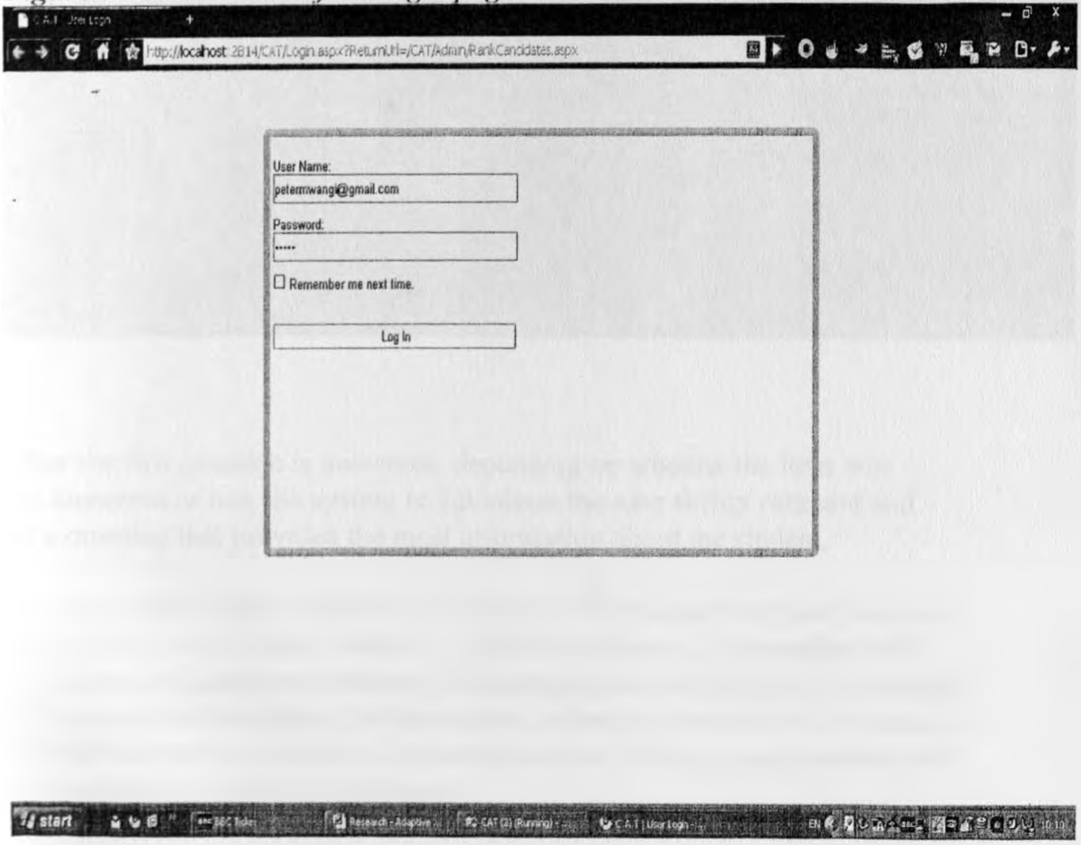
- C# - Will provide the means for interaction with the database as well as the adaptation algorithm
- ASP.NET – Has been used to design the web pages.
- Javascript - A scripting language which sometimes acts as a link between C# and ASP.net. It also makes the user interaction with a webpage more efficient and can also capture the user input.

Functionality contained in the developed system includes the following:

i) Login

The system allows the student and the administrator (expert) to login. The user type is a security feature of the system that controls what pages will be viewed by either user. The login requires the input of a user name and password. The user name is the email of the user. This is because the email is unique to each individual and cannot be duplicated. The screen is as shown in figure 16 below:

Figure 16: Screenshot of the login page



http://localhost:2814/CAT/Login.aspx?ReturnUrl=/CAT/Admin/RankCandidates.aspx

User Name: petermwangi@gmail.com

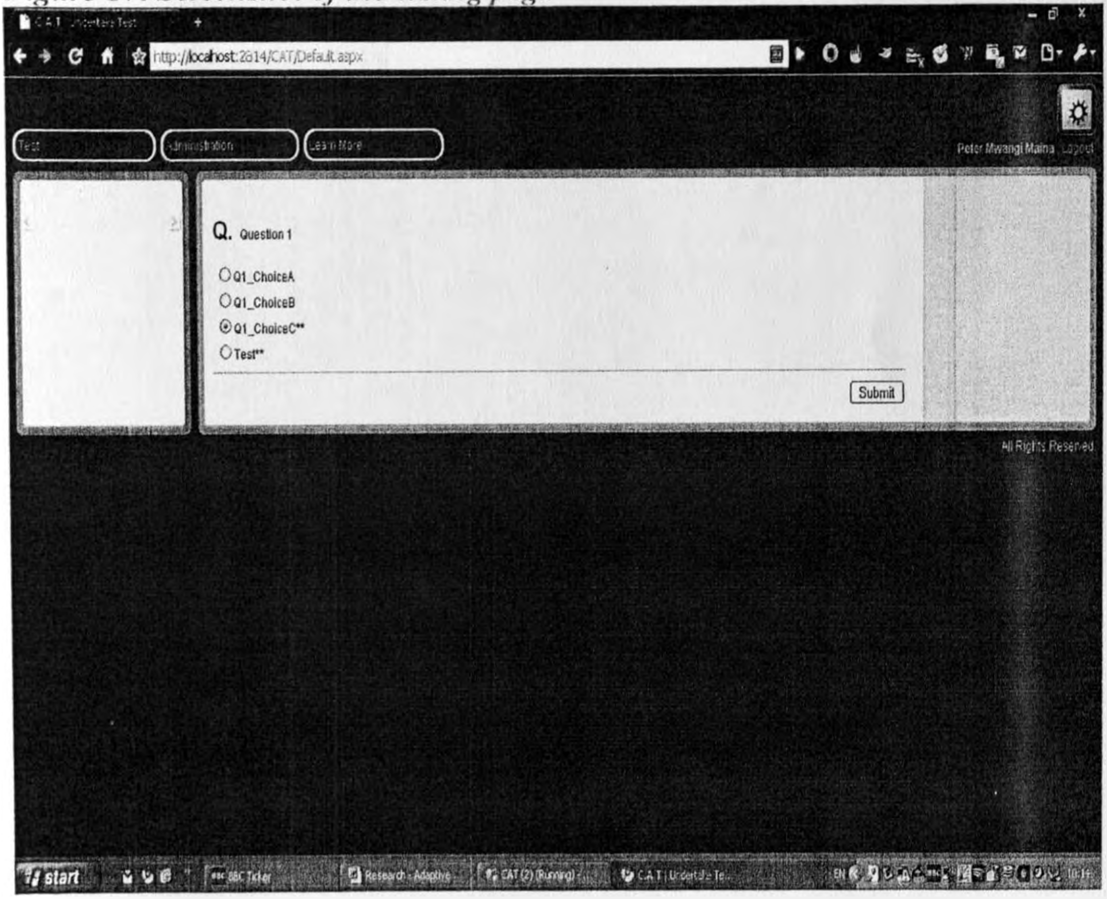
Password: •••••

☐ Remember me next time.

ii) Testing

When the student logs in, s/he clicks on the menu item Test/ Undertake test. A screen shot of the start page of the test, where the first question is presented is as shown in figure 17 below:

Figure 17: Screenshot of the testing page



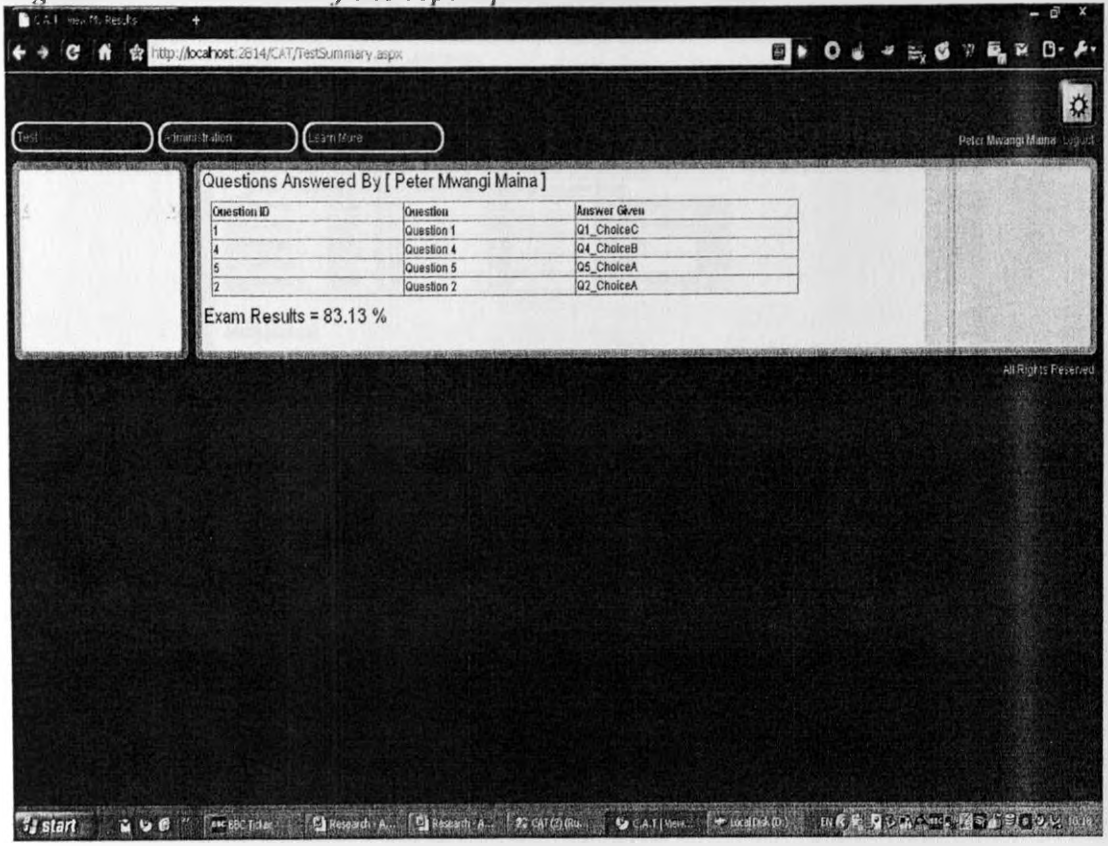
Q. Question 1

○ Q1_ChoiceA
○ Q1_ChoiceB
● Q1_ChoiceC**
○ Test**

After the first question is answered, depending on whether the item was correctly answered or not, the system re-calculates the new ability estimate and presents a question that provides the most information about the student.

During the test or upon completion of the test, the student can view their test results by clicking on the menu item Test – View my answers. The student will receive a display of questions answered, the answers given and the final percentage score. If the test is not complete, the student can go back to complete it. However, he cannot revisit the previous questions answered and can therefore only proceed with the test. The resulting screen is as follows:

Figure 18: Screen shot of the report presented to the student



http://localhost:2814/CAT/TestSummary.aspx
Test
Administration
Learn More
Peter Mwangi Maina Logout
All Rights Reserved
start

Questions Answered By [Peter Mwangi Maina]

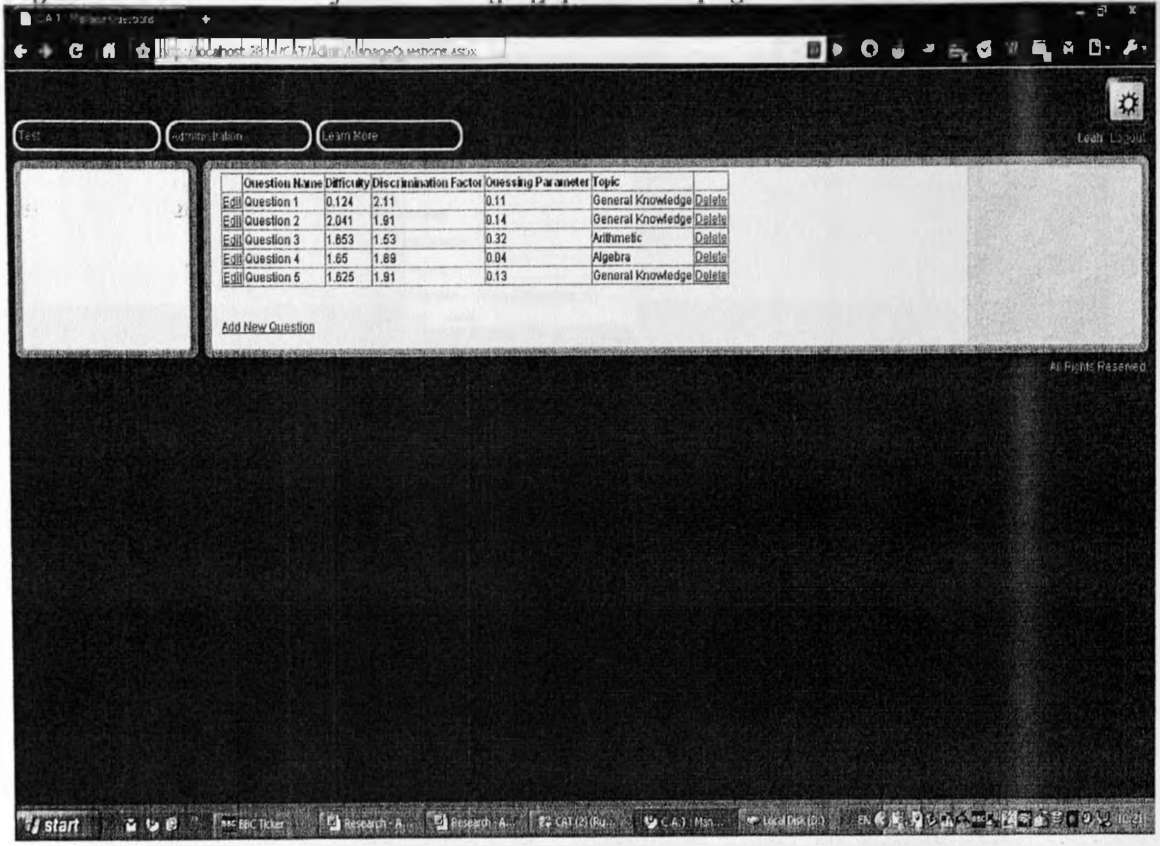
Question ID	Question	Answer Given
1	Question 1	Q1_ChoiceC
4	Question 4	Q4_ChoiceB
5	Question 5	Q5_ChoiceA
2	Question 2	Q2_ChoiceA

Exam Results = 83.13 %

iii) Managing Questions

The administrator can view the questions contained in a particular test by clicking the menu item Administration – Managing questions. He will be shown the Question name, the parameters of the question (difficulty, discrimination and the guessing), and the topic that the question is derived from. The screen shot is as shown in figure 19:

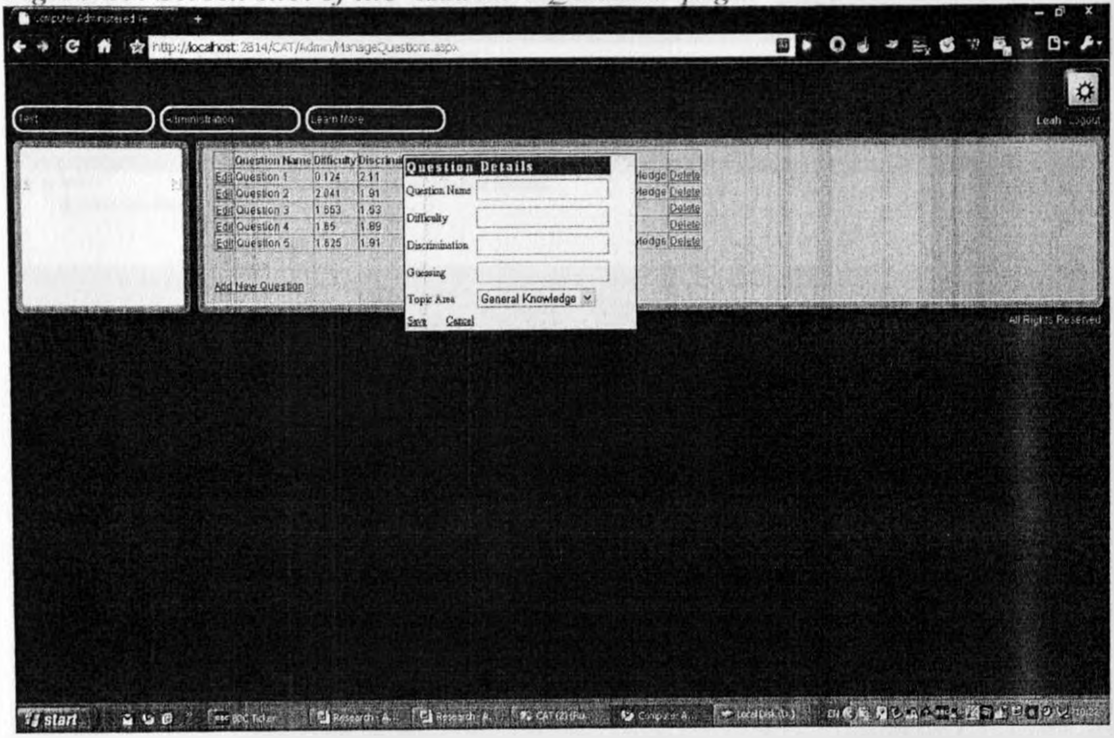
Figure 19: Screen shot of the 'managing questions' page



	Question Name	Difficulty	Discrimination Factor	Guessing Parameter	Topic	
Edit	Question 1	0.124	2.11	0.11	General Knowledge	Delete
Edit	Question 2	2.041	1.91	0.14	General Knowledge	Delete
Edit	Question 3	1.853	1.53	0.32	Arithmetic	Delete
Edit	Question 4	1.65	1.89	0.04	Algebra	Delete
Edit	Question 5	1.625	1.91	0.13	General Knowledge	Delete

Adding of new questions is also performed here by clicking the 'Add New Question' link. Upon clicking it, the system will allow the administrator to input the question, with its corresponding parameters of difficulty, discrimination and guessing. This is as shown in figure 20:

Figure 20: Screen shot of the 'Add New Question' page



	Question Name	Difficulty	Discrimi		
Edit	Question 1	0.124	2.11	ledge	Delete
Edit	Question 2	2.041	1.91	ledge	Delete
Edit	Question 3	1.653	1.63		Delete
Edit	Question 4	1.85	1.89		Delete
Edit	Question 5	1.625	1.91	ledge	Delete

Add New Question

Question Details

Question Name

Difficulty

Discrimination

Guessing

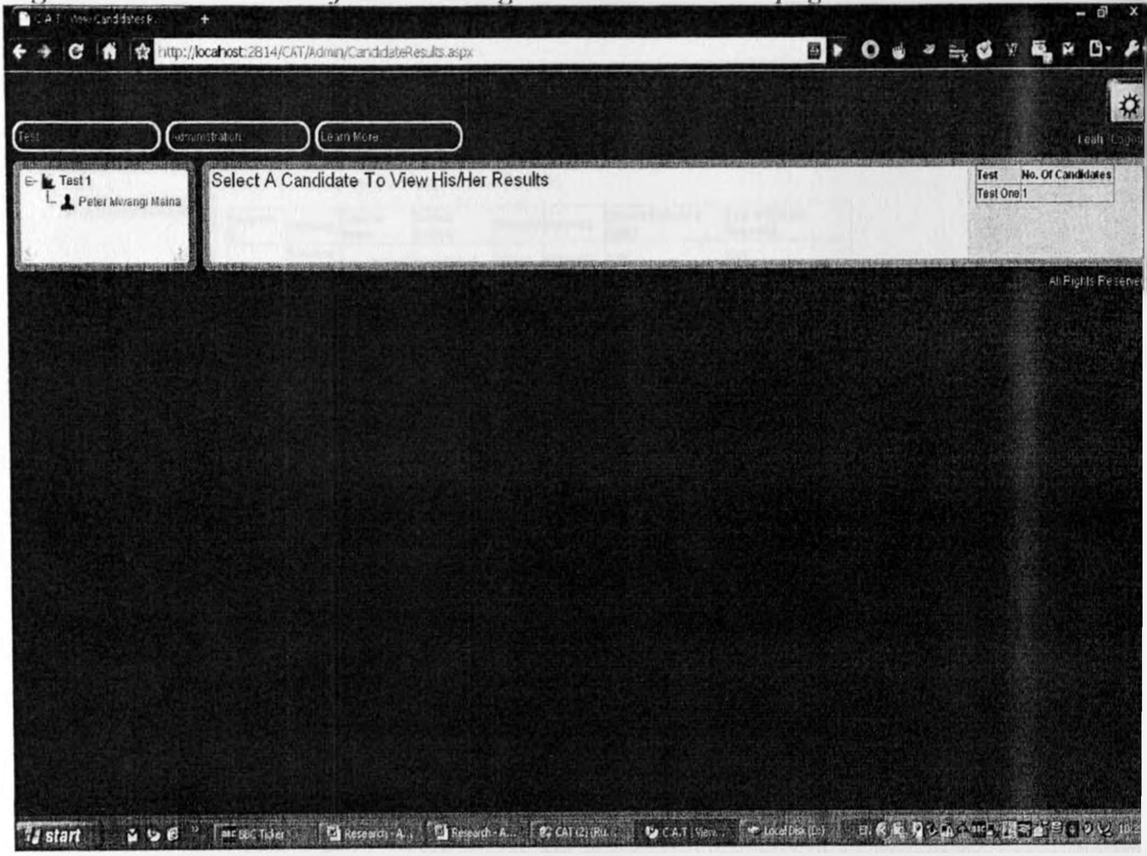
Topic Area: General Knowledge

Save Cancel

iv) Viewing candidate results

The Administrator is also able to view the candidate's results. He does this by clicking on the menu item Administrator – View Candidate's Results. This produces a screen as follows:

Figure 21: Screen shot of the 'viewing candidates results' page



Select A Candidate To View His/Her Results

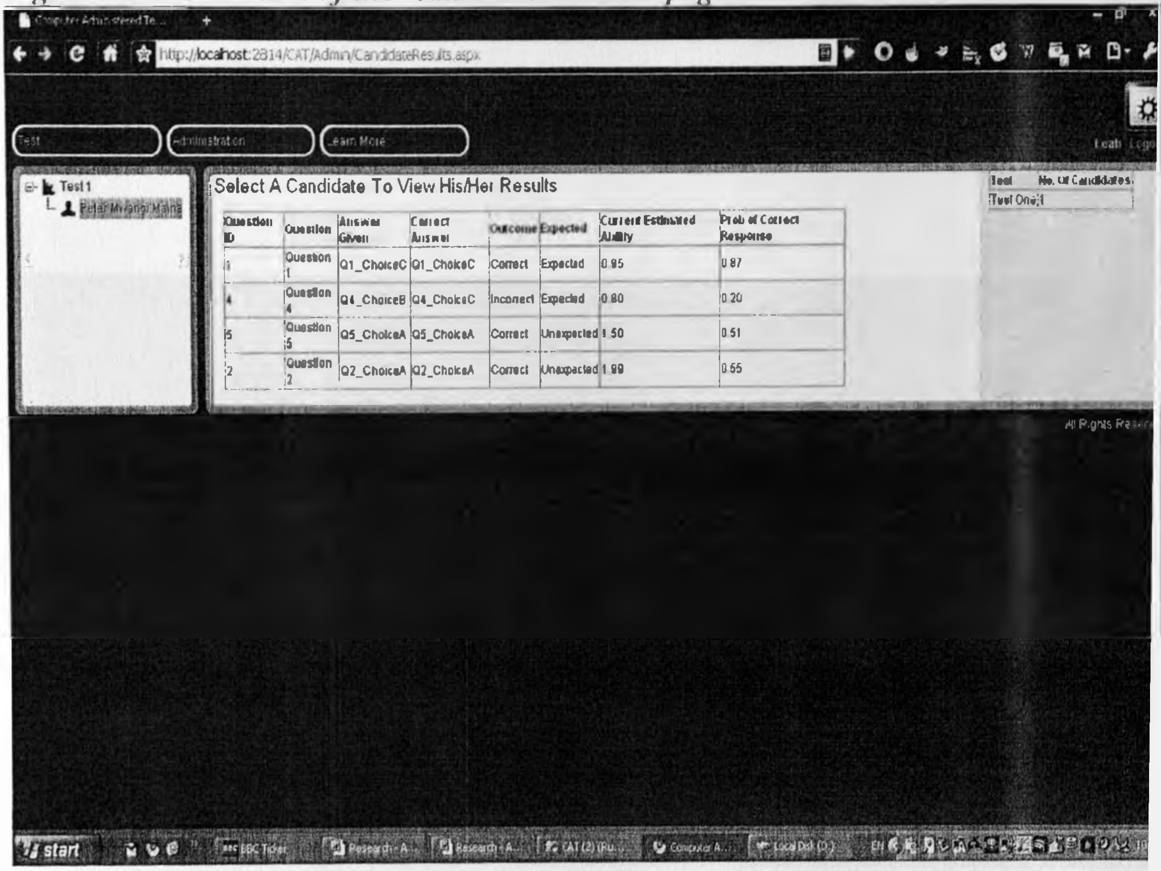
Test	No. Of Candidates
Test One	1

The Administrator then selects the name of a particular student whose results will be viewed. The test results contain the following aspects:

- Question ID
- Question – Name of the question
- Answer given – The choice that the student selected
- Correct answer – The correct answer
- Outcome – Whether the student got the question correct or not
- Expected – The system compares the probability of getting the item correct and the outcome. If the probability < (1 + c)/2, then it is expected that the student will get the question wrong.
- Current estimated ability
- Probability of correct response

The Administrator will interpret these results. The test results are as displayed:

Figure 22: Screen shot of the 'candidates results' page



http://localhost:2814/CAT/Admin/CandidateResults.aspx

Select A Candidate To View His/Her Results

Question ID	Question	Answer Given	Correct Answer	Outcome	Expected	Current Estimated Ability	Prob of Correct Response
1	Question 1	Q1_ChoiceC	Q1_ChoiceC	Correct	Expected	0.95	0.87
4	Question 4	Q4_ChoiceB	Q4_ChoiceC	Incorrect	Expected	0.80	0.20
5	Question 5	Q5_ChoiceA	Q5_ChoiceA	Correct	Unexpected	1.50	0.51
2	Question 2	Q2_ChoiceA	Q2_ChoiceA	Correct	Unexpected	1.99	0.55

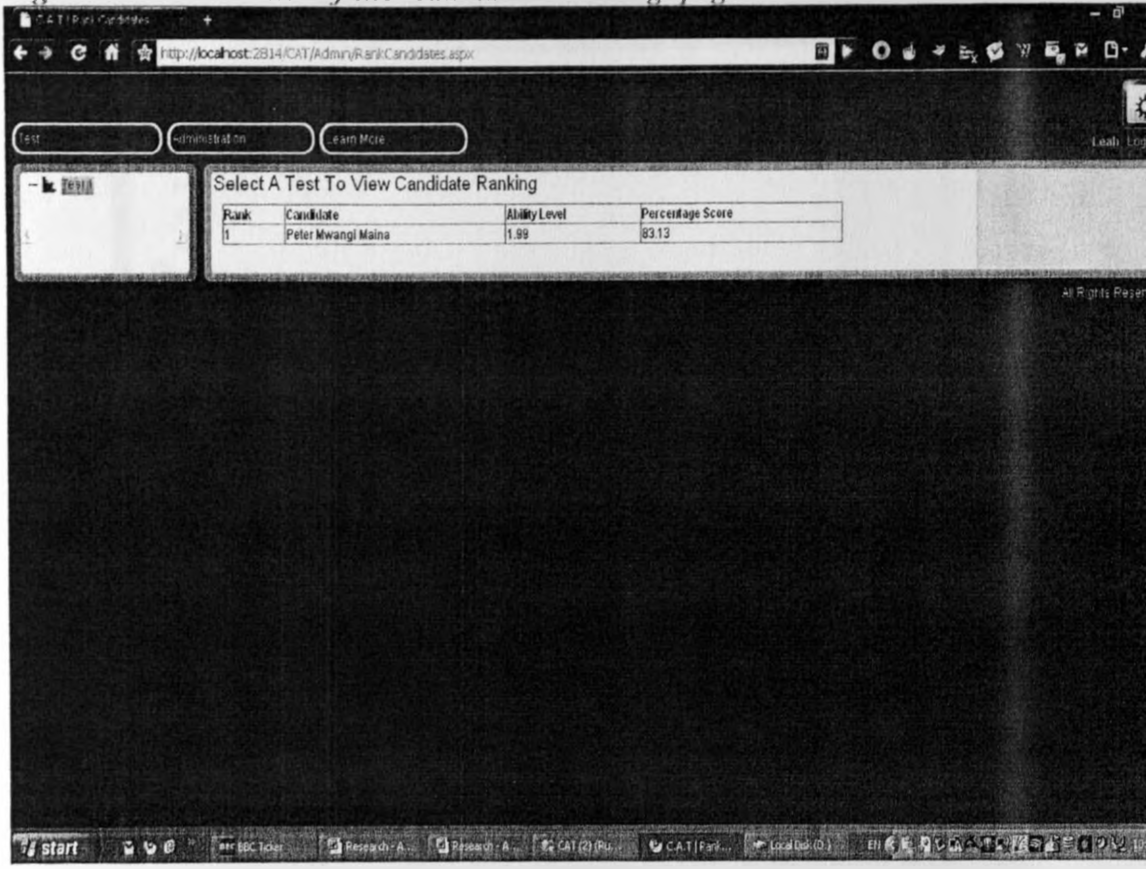
Test	No. Of Candidates
Test One	1

The administrator can also view candidate ranking which comprises:

- Name of the candidate
- Final ability level
- Percentage score of the student

This is as shown on figure 23 below:

Figure'23: Screen shot of the 'candidates ranking' page



Select A Test To View Candidate Ranking

Rank	Candidate	Ability Level	Percentage Score
1	Peter Mwangi Maina	1.99	83.13

v) Adaptivity of the system

The test will present questions depending on how the student is performing on each question. When the questions are input into the system, the system uses the parameters of difficulty, discrimination and guessing, together with the ability estimates to plot their item characteristic curves. The curve for all the questions 1, 2, 3 and 4 (when the ability estimate is '0') are as shown in the figures 24 (a) and (b) below:

Figure 24 (a): Item response curves of the questions before the test begins

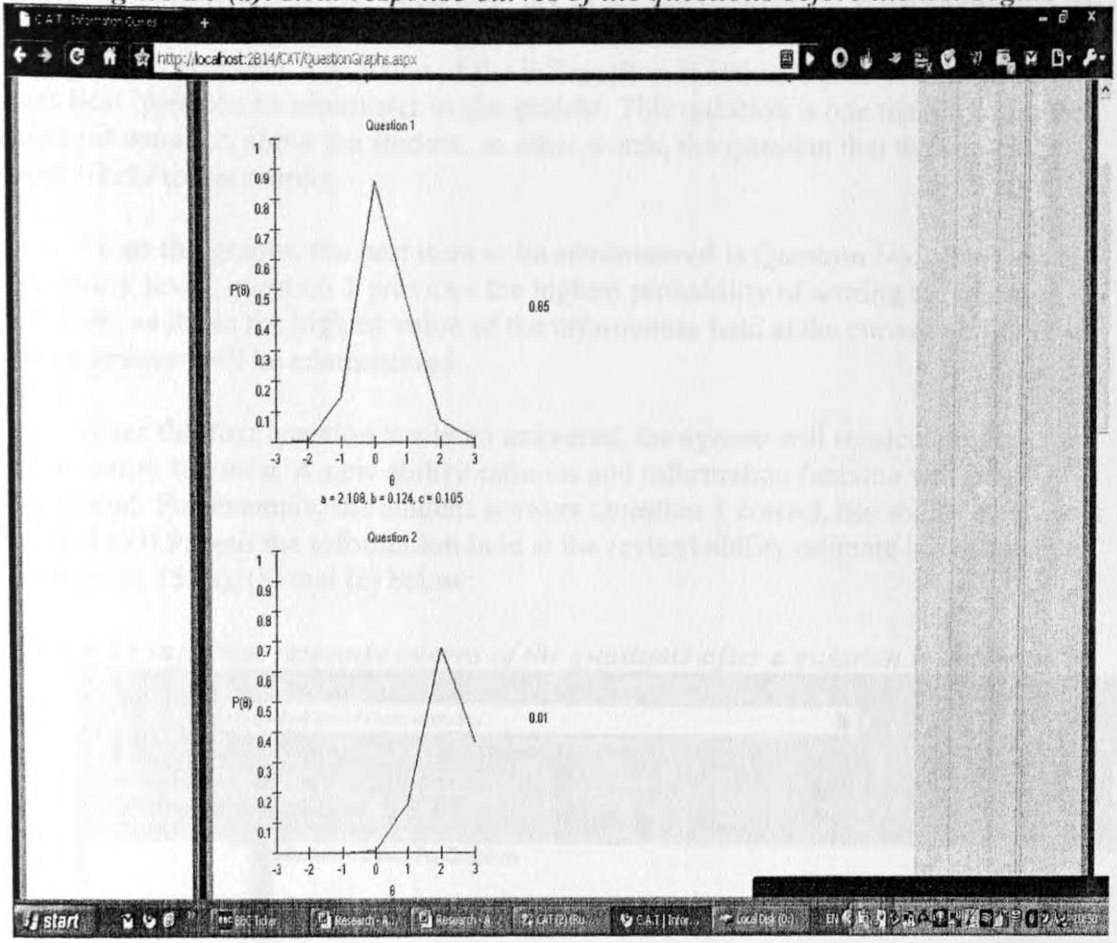
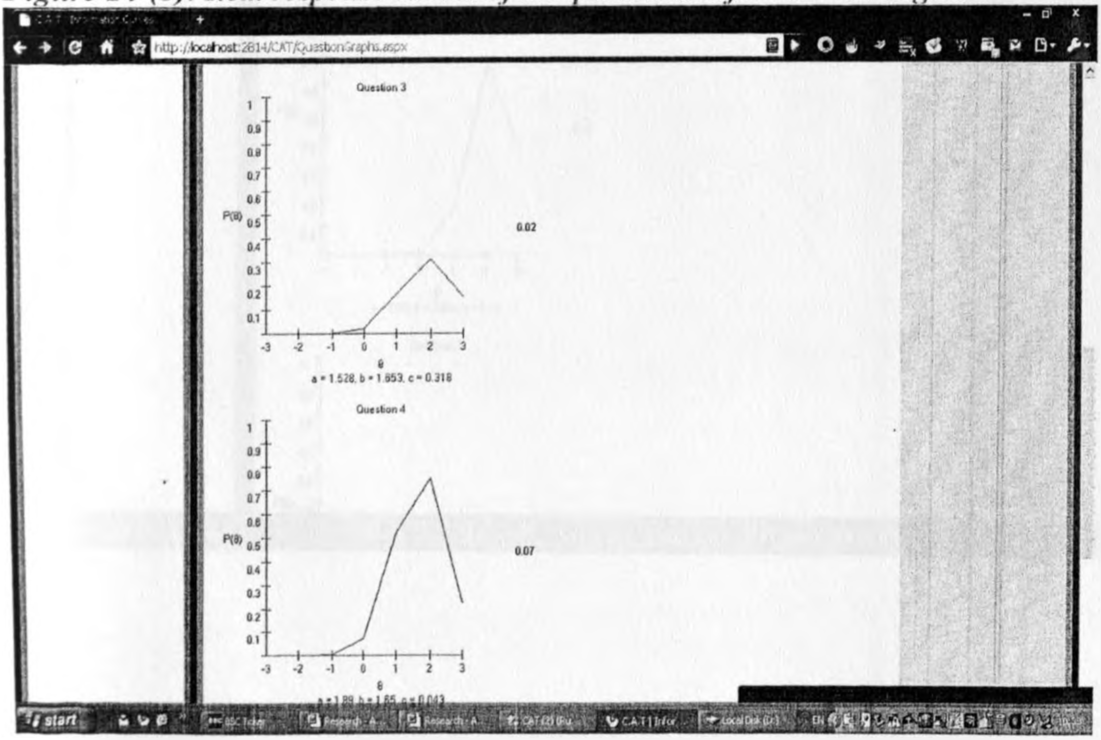


Figure 24 (b): Item response curves of the questions before the test begins

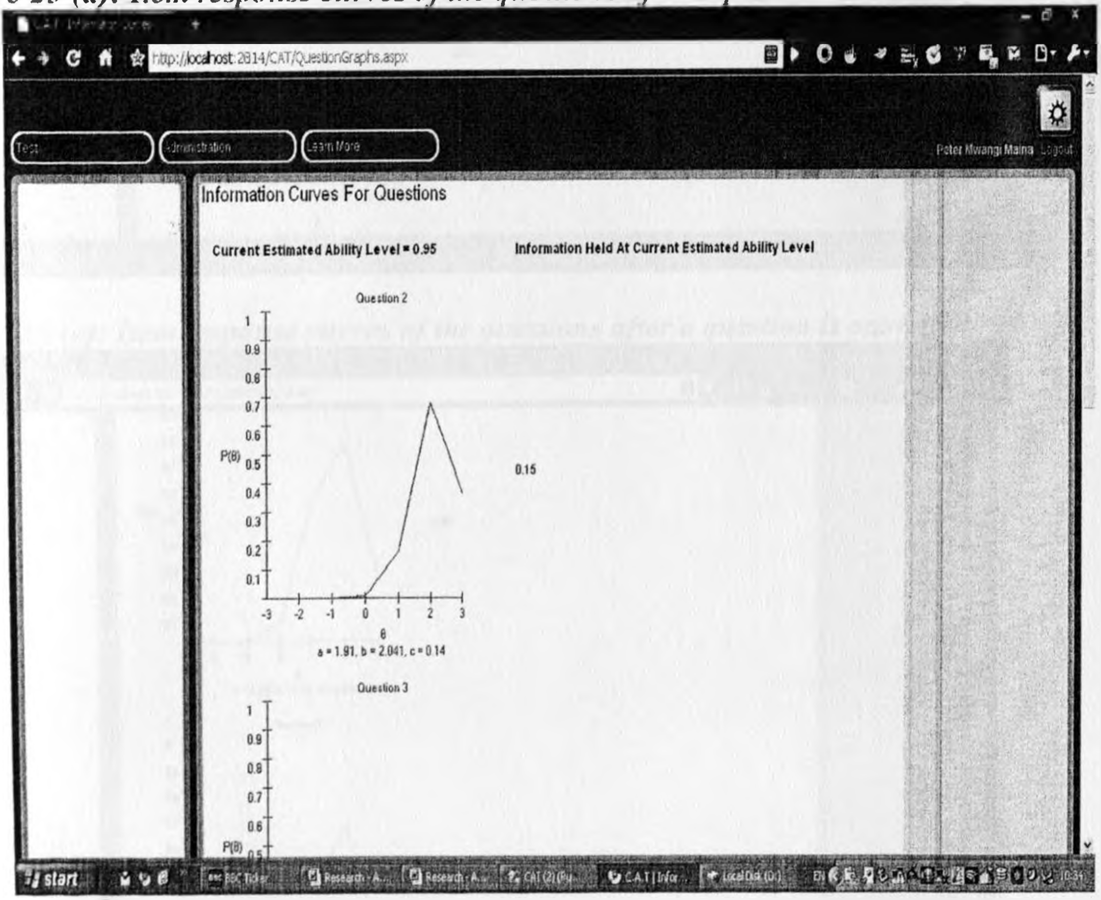


The characteristic curves do not vary as the test progresses. What changes is the current ability estimate and consequently the information function as the student tackles each question. The value of the information function is used to determine the next best question to administer to the student. This question is one that provides the most information about the student. In other words, the question that the student is most likely to get correct.

From the graphs, the best item to be administered is Question No. 1 because at '0' ability level, question 1 provides the highest probability of scoring the question correctly as it has the highest value of the information held at the current ability level. This question will be administered.

After the first question has been answered, the system will recalculate the information function. A new ability estimate and information function will be calculated. For example, the student answers Question 1 correct, the ability estimate is revised to 0.95 and the information held at the revised ability estimate is as shown in the figures 25 (a), (b) and (c) below:

Figure 25 (a): Item response curves of the questions after a question is answered



http://localhost:2814/CAT/QuestionGraphs.aspx

Information Curves For Questions

Current Estimated Ability Level = 0.95

Information Held At Current Estimated Ability Level

Question 2

0.15

a = 1.91, b = 2.041, c = 0.14

Question 3

Figure 25 (b): Item response curves of the questions after a question is answered

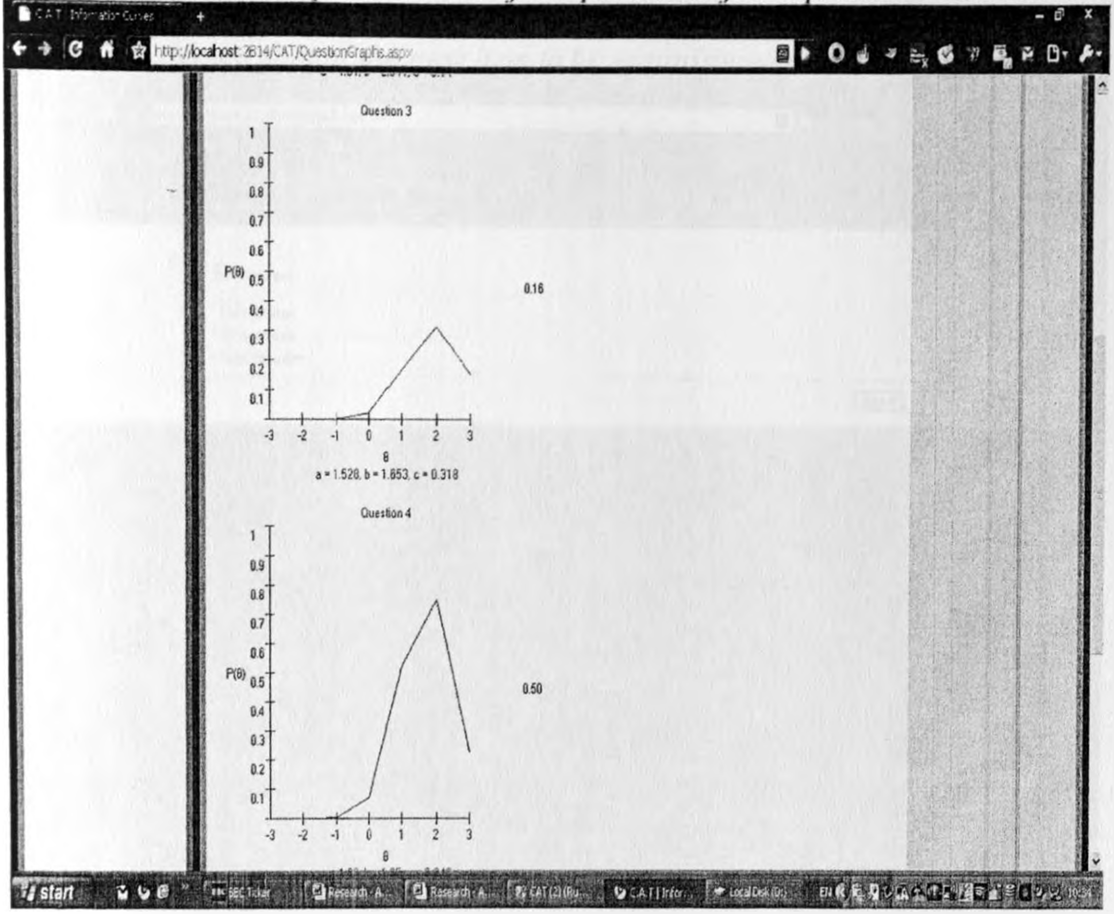
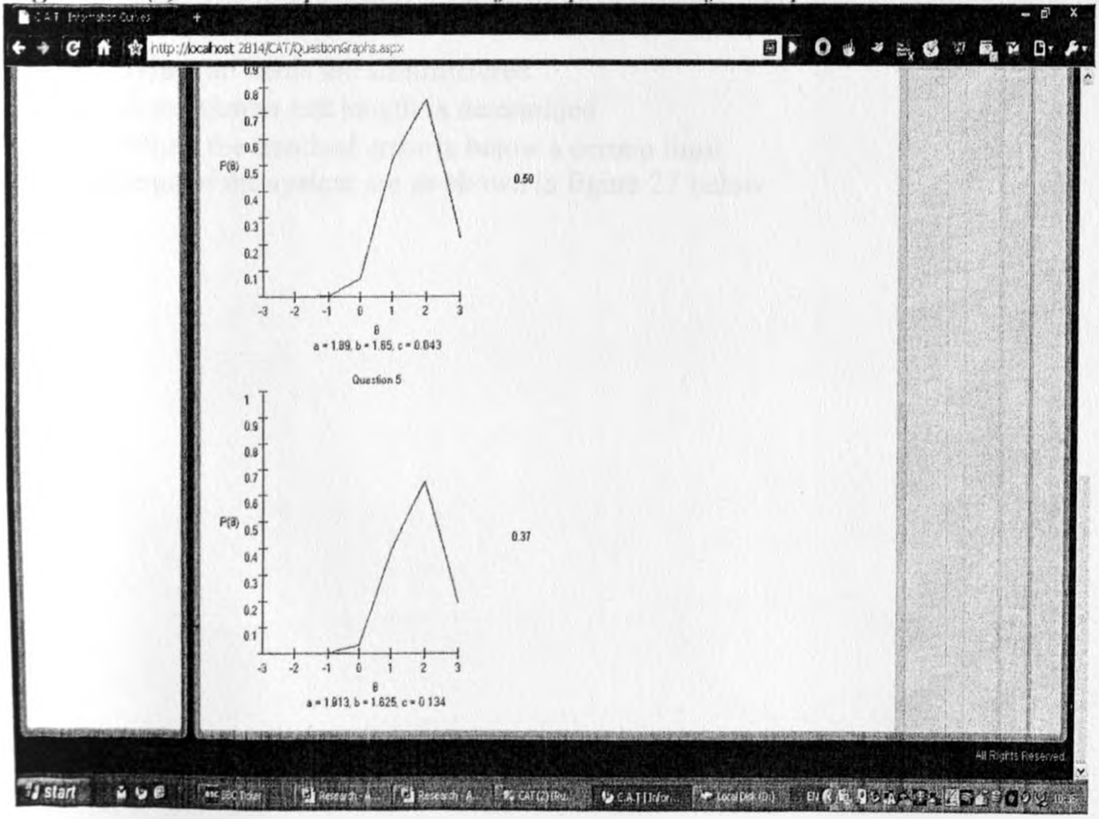
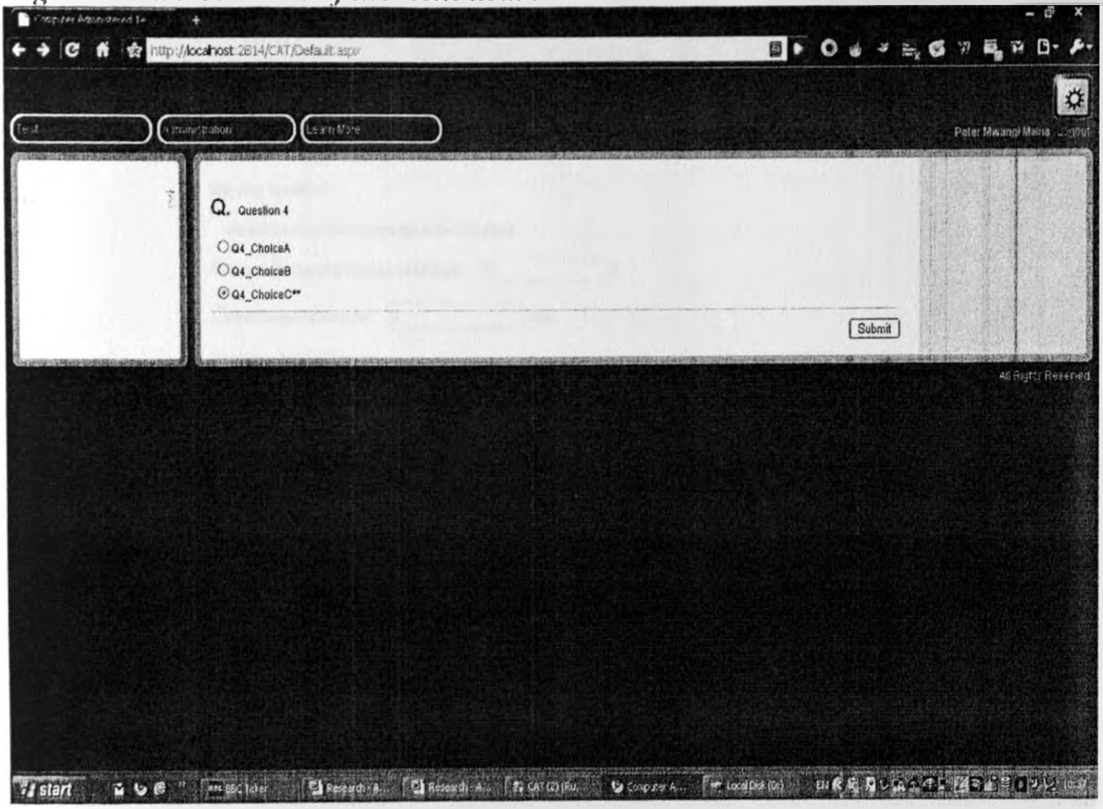


Figure 25 (c): Item response curves of the questions after a question is answered



The next best question to be administered is Question 4 as shown in figure 26:

Figure 26: Screen shot of the next item to be administered on the test



Q. Question 4

○ Q4_ChoiceA

○ Q4_ChoiceB

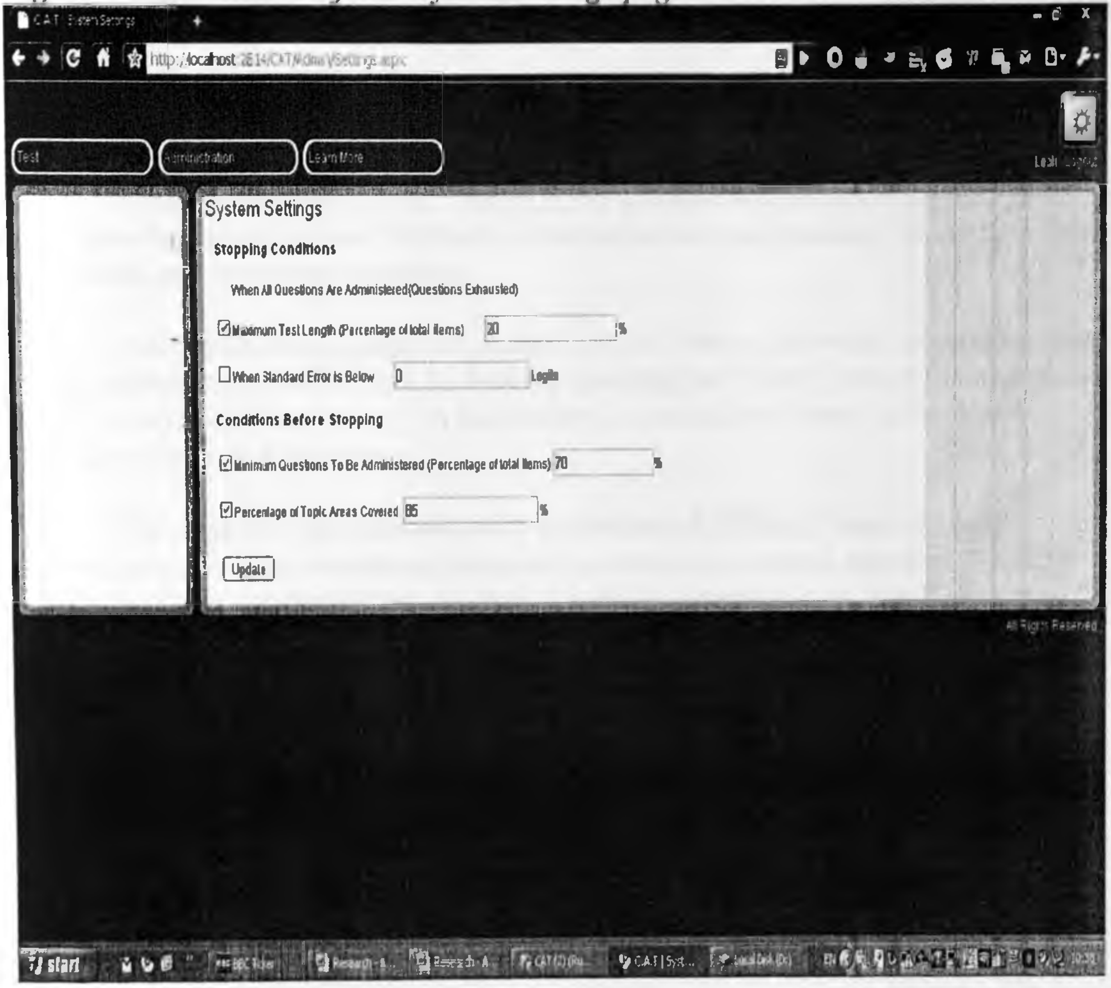
◉ Q4_ChoiceC**

The iteration of adapting to the student will continue until the stopping criterion is encountered. The test can stop because of the following criteria:

- When all items are administered
- A maximum test length is determined
- When the standard error is below a certain limit

The settings of the system are as shown in figure 27 below:

Figure 27: Screen shot of the 'System settings'page



System Settings

Stopping Conditions

When All Questions Are Administered(Questions Exhausted)

☑ Maximum Test Length (Percentage of total items) 20 %

☐ When Standard Error Is Below 0 Logits

Conditions Before Stopping

☑ Minimum Questions To Be Administered (Percentage of total items) 70 %

☑ Percentage of Topic Areas Covered 85 %

Update

5.1 Summary

Up to this point, the research focused on the design and implementation of the computer adaptive test prototype.

A computer-adaptive test system has a database where the questions with their corresponding parameters of difficulty, discrimination and guessing. These have been calculated and set by the examiner.

In addition to the question database, a CAT testing algorithm is required. The testing algorithm of a CAT can be broadly described as "start", "select the item to be administered next" and "stop". In this system, an examinee cannot go back and change his previous responses.

The adaptive test can starts with a question of difficulty based on prior information about the test-taker, a random question or a random question of middle difficulty. After answering the question, the ability estimate is revised. The revised ability estimate is used to determine the best question to present next. The best question is the one that provides the most information about the candidate as it is neither too hard nor too easy.

The stop condition of the adaptive test can be based on a single stopping rule, or a combination of stopping rules. The stopping rules in this system included: a percentage of total number questions has been administered, all the questions have been administered, percentage coverage of topic has been presented and a standard error for the proficiency level estimate has been attained.

6. CHAPTER 6 – CONCLUSION

6.1 Overview

The aim of this project was to develop a computer adaptive test system that can aid in determining the sequence in which test questions should be presented to students depending on the trend of answering of previous questions.

This chapter presents a summary of this work. It also discusses the scope for future work.

6.2 Summary

Chapter 1 discusses the problem statement, objectives of the study, justification of the study, hypothesis and also defines some terms that will be used in the write up.

Chapter 2 is the literature review. Here, previous research concerning the area of adaptive testing is highlighted. Paper based testing and adaptive testing is compared.

Chapter 3 discusses the methodology that the researcher will use in doing the research. Prototyping will be used in demonstrating the working of an adaptive test.

Chapter 4 looks at the design and implementation of the system. It explains thc system Architecture, the system design and the database structure.

Chapter 5 shows the results of the system testing. This chapter expounds on the operation of the system and its functionality.

6.3 Main Contribution

This research has been concerned with a method of developing and delivering tests by using adaptive testing.

The main contribution of this research is in the improvement of a stage of recruitment/placement system in institutions that uses aptitude tests. In addition, this system can also be used in university during selection of students for training courses. This is because some bright students perform less than they could have performed in KCSE because of certain factors that were beyond anyone's control such as death of a parent. Such a student is able, but his grade does not permit him to join a course. Universities can use this system to test and grade such students to determine their ability and determine whether they can undertake the course or not.

In addition, the system can be used in identifying weaknesses and strengths of individuals thereby resulting to a fairly accurate assessment of potential candidates for a job position.

6.4 Assessment of the research

The first research was 'the potential applications of adaptive testing in the interview process'. The answer proposed in this research is that the CAT approach is both valid and reliable in assessment during interviews. The research demonstrated that the CAT approach was effective in tailoring the level of difficulty of the test to individual students. The CAT will ensure that the student is presented with questions that will determine his ability estimate because they are suited to his ability.

From the test results, the assessor will see the ability of the candidates. If the job description requires that the person be knowledgeable in a certain area, the test questions will be tailored and calibrated appropriately. The final report will clearly show how the student answered and will give an indication of his ability. The ability estimate will be fairly accurate because the candidate only answers questions that are suitable for his ability and are neither too hard nor too easy.

In summary, it was found that the prototype in this research supports the provision of feedback that is timely and effective at matching questions to the proficiency level of each individual candidate.

6.5 Research Objectives

The objectives were as follows:

i. Identify issues in designing and implementing a CAT application.
ii. Examine the existing system of testing in recruitment institutions noting its strengths and drawbacks. In addition, the research will examine computer adaptive testing, also noting its positives and negatives.
iii. The second objective of this study will be to design and implement a CAT application. The corresponding results of the test will be presented in a report format showing the question answered, the answer given, whether the answer is correct or not, the expected outcome and the final ability estimate as a percentage.

The research first sought to understand the fundamental issues and concerns in the use of the CAT approach. Issues facing the current system of testing and those facing the implementation of CATs were identified. Shortfalls of the current system were also identified.

The prototype was developed and needed to demonstrate that it can adapt the questions to the ability of the student and that there will be immediate and useful feedback to him and to the examiner as well. The system was developed using the item response theory principles. The system demonstrated that the test content will be varied depending on the ability of the individual candidate. Secondly, after the test completion, the system gives a report to the candidate showing the final percentage score, and the examiner also receives a report showing how the candidate answered and the revised ability estimate at every point of undertaking the test.

6.6 Limitations of The Study

- As stated before, the type of questions are limited to multiple choice types. This is because open ended questions are difficult to score in a reliable manner.
- The researcher is not trained in psychometric analysis of the questions. An 'expert' who is trained in testing would be able to correctly formulate a real test.
- A small sample of questions has been used. A large sample of 1,000 to 2,000 items would yield more accurate results but due to the constraint of time and resources, only a small sample has been used.

6.7 Further Work

These include the following:

- The research has only concentrated on Item Response Theory in representing the parameters of the test items. However, there are other ways and methods that can be used such as 'Knowledge Space Theory' and 'Constraint Logic Programming'. A study of these theories can be done so as to compare their results of testing
- Currently, the tests utilizing adaptive testing are "western" such as GMAT, GRE and SATs. These tests have been formulated from western syllabuses. However in Kenya particularly, no study has been conducted to represent any of our tests in a computer adaptive test format. Educational facilities will benefit immensely from this technology. Experts who are trained in test development should embark on this study.

APPENDIX A

References and Bibliography

Ben Onyasimi, 8-4-4: Focus lost amid calls for change of course (2003) http://www.kentimes.com/04jun05/nwsstory/opinion.html

Center for Advanced Research on Language Acquisition [CARLA] (2009), Frequently Asked Questions about CAT from http://www.carla.umn.edu/assessment/CATfaq.html

Comprehensive Osteopathic Medical Licensing Examinations (COMLEX-USA) Frequently Asked Questions (2005) from http://www.nbome.org/copyright.htm

Computerized Adaptive Testing: From Inquiry to Operation Edited By: William A. Sands, Brian K. Waters and James R. McBride Constructive Research (2007) from http://en.wikipedia.org/wiki/Constructive_research

Doignon,J.P., Falmagne,J.C. (1985). Spaces for the Assessment of Knowledge. International Journal of Man- Machine Studies 23 (2), pp.175-196.

Dowling,C.E., Kaluscha,R. (1995). Prerequisite Relationships for the Adaptive Assessment of Knowledge. In: J.Greer, (Ed.), Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, 16-19 August1995, AACE pp. 43-50.

Dr. Randall E. Schumacker, Computer Adaptive Testing (2004) from http://www.coe.unt.edu/schumacker/Acrobat%20Files/cat.pdf

Falmagne,J.C., Doignon,J.P., Koppen,M., Villano,M., Johannesen,L. (1990). Introduction to Knowledge Spaces - How to Build, Test, and Search Them. Psychological Review 97 (2), pp. 201 – 224.

Falmagne,J.C., Doignon,J.P., Koppen,M., Villano,M., Johannesen,L. (1990). Introduction to Knowledge Spaces- How to Build, Test, and Search Them. Psychological Review 97 (2), pp 201-224.

Frank B. Baker, The Basics of Item Response Theory (2001) pp. 28 – 35 from http://echo.edres.org:8080/irt/baker/

Grist, Susan, Computerized Adaptive Tests (1989) from http://www.ericdigests.org/pre-9213/tests.htm - Computerized Adaptive Tests. ERIC Digest No. 107.

Jeff Weekley Phd, Computer Adaptive Testing (2004) - Kenexa HR Newsletter from http://www.kenexa.com/newsletter/11040.asp?uid=1&tbl=0704

Kingsbury G., Zara A. (1991). A Comparison of Procedures for Content-Sensitive Item Selection in Computerized Adaptive Tests. *Applied Measurement in Education*, 4 (3) 241-61

Lawrence M. Rudner, An On-line, Interactive, Computer Adaptive Testing Tutorial from: http://edres .org/scripts/cat/catdemo.htm

Lord, F. M. Applications of Item Response Theory to Practical Testing Problems (1980) New Jersey: Lawrence Erlbaum Associates, Publishers.

Lord,F.M. (1980). Applications of Item Response Theory to Practical Testing Problems. Lawrence Erlbaum Associates, Publishers, New Jersey.

Mary E. Lunz Phd., Administrative Issues for Computerized Tests and Surveys (2005) from http://www.measurementresearch.com/media/admin_issues_cmptest.pdf -

Microsoft Inc., Advantages and Disadvantages of adaptive testing (2006) from http://www.windowsgalore.com/cert/adaptive_testing/advant.html

Microsoft inc., History of Adaptive Testing in high Stakes Testing (2004) from http://www.windowsgalore.com/cert/adaptive_testing/ - Adaptive Testing

Otieno J.K, We must enable more Kenyans to access quality education (2003) from http://www.kentimes.com/12jan05/schtimes/sch2.html

Patricia A. Dunkel, Georgia state university (1999), Considerations in Developing and Using Computer-Adaptive Tests to Assess Second Language Proficiency from http://www.cal.org/resources/digest/cat.html

Patricia A. Dunkel, Georgia state university, Guidelines for computer based testing (1999), from http://www.isoc.org/oti/articles/0500/olsen.html and http://jalt-publications.org/tlt/files/97/oct/dunkel.html

Psychometric Success, Aptitude Tests – Introduction (2009) from http://www.psychometric-success.com/aptitude-tests/aptitude-tests-introduction.htm

Thomson Prometric, Testing Questions (2004), from http://www.prometric.com/Candidates/FAQs/TestingQuestions.htm#2

Villano,M. (1992). Probabilistic Student Models- Bayesian Belief Networks and Knowledge Space Theory. In: Frasson,C., Gauthier,G., McCalla,G.I., (Eds.), Intelligent Tutoring Systems. Second International Conference, ITS'92, Montréal, Canada, June 1992 proceedings. Lecture Notes in Computer Science608. Springer-Verlag, Berlin, Heidelberg, New York, pp. 491-498.

Wainer,H., Mislevy,R.J. (1990). Item Response Theory, Item Calibration and Proficiency Estimation. In: Wainer,H., (Ed.), Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates, Publishers, New Jersey, pp. 65-102

Weiss,D.J., Kingsbury,G.G. (1984). Application of Computerized Adaptive Testing to Educational Problems . Journal of Educational Measurement 21 (4), pp.361-375.

Welch,R.E., Frick,T.W. (1993). Computerized Adaptive Testing in Instructional Settings. Educational Technology Research & Development 41 (3), pp.47-62.

APPENDIX B

Sample Programs

1. Engine.cs – This has the adaptivity of the system

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Drawing;
using DataAccess;
using System.Collections;
using System.Linq.Expressions;

/// Summary description for Engine

public class Engine
{
      public Engine()
      {
      }

    /// <summary>
    /// Get probability of correct response
    /// </summary>
    /// <param name="ability_level"></param>
    /// <param name="question"></param>
    /// <returns></returns>
    public static double GetProbability(double ability_level,
 Question question)
    {
        return GetProbability(ability_level,
 question.Difficulty.Value, question.Discrimination.Value,
 question.Guessing.Value);
    }

    /// <summary>
    /// Get probability of correct response
    /// </summary>
    /// <param name="difficulty">-3 <= b <= +3.</param>
    /// <param name="discrimination"> -2.8 <= a <= +2.8</param>
    /// <param name="guessing">0<=c<=1.0 Acceptable range is 0 -
 .35</param>
    /// <returns>probability of correct answer</returns>
    public static double GetProbability(double ability_level, double
 difficulty, double discrimination, double guessing)
    {
        //Logit = discrimination (abilitylevel - difficulty)
        double probability = 0;
        double logit = discrimination * (ability_level - difficulty);
        double exp = Math.Exp(-1 * logit);
        double denominator = 1 + exp;
        double result = 1 / denominator;
        probability = guessing + (1 - guessing) * result;
        return probability;
    }

     /// <summary>
     /// Get all probabilities at different ability levels for tha
  question
```

```
    /// </summary>
    /// <param name="question"></param>
    /// <returns></returns>
    public static GraphPoints GetAbilityLevelProbabilities(Question
question)
    {
        GraphPoints points = new GraphPoints();
        GraphPoint point = new GraphPoint();
        double dProb = 0;
        for (int i = -3; i <= 3; i++)
        {
            dProb = GetProbability(i, question);
            point = new GraphPoint(Convert.ToSingle(i),
Convert.ToSingle(dProb));
            points.Add(point);
        }
        return points;
    }

    /// <summary>
    /// Generate information function for the question
    /// </summary>
    /// <param name="question"></param>
    /// <returns></returns>
    public static GraphPoints GetItemInformationFunction(Question
question)
    {
        GraphPoints points = new GraphPoints();
        GraphPoint point = new GraphPoint();
        double info = 0;
        for (int i = -3; i <= 3; i++)
        {
            info = GetInformation(i, question);
            point = new GraphPoint(Convert.ToSingle(i),
Convert.ToSingle(info));
            points.Add(point);
        }
        return points;
    }

    /// <summary>
    /// Get information for a question given the ability level
    /// </summary>
    /// <param name="ability_level"></param>
    /// <param name="question"></param>
    /// <returns></returns>
    public static double GetInformation(double ability_level,
Question question)
    {
        return GetInformation(ability_level,
question.Difficulty.Value, question.Discrimination.Value,
question.Guessing.Value);
    }

    /// <summary>
    /// Get information for a question given the ability level and
the question parameters
    /// </summary>
    /// <param name="ability_level"></param>
    /// <param name="difficulty"></param>
    /// <param name="discrimination"></param>
    /// <param name="guessing"></param>
    /// <returns></returns>
    public static double GetInformation(double ability_level, double
difficulty, double discrimination, double guessing)
    {
        double info = 0;
        double p = GetProbability(ability_level, difficulty,
discrimination, guessing);
        double q = 1 - p; ;
        double q_divide_p = q / p;
        double disc_squared = discrimination * discrimination;
        double one_minus_guessing_Squared = Math.Pow((1 - guessing),
2);
        double p_minus_guessing_squared = Math.Pow((p - guessing),
2);
        info = disc_squared * q_divide_p * (p_minus_guessing_squared)
/ (one_minus_guessing_Squared);
        return info;
    }

    /*public static Question2 GetQuestionWithBestInformation(double
ability_level)
    {
        //Get List of unadministered questions
        Questions2 questions = new
Question2().GetUnadministeredQuestions();
        foreach (Question2 q in questions)
        {
            q.Information = GetInformation(ability_level,
q.Difficulty, q.Discrimination, q.Guessing);
        }
        questions.Sort("Information", true);
        return questions.Last();
    }*/

    /// <summary>
    /// Get Question that gives the best information about a
candidate who has the specified ability level
    /// The Question returned is the question to be administered next
to the candidate.
    /// Adaptation is done by this function
    /// </summary>
    /// <param name="ability_level"></param>
    /// <returns></returns>
    public static Question GetQuestionWithBestInformation(double
ability_level)
    {
        //Question_Extended
        //Get List of unadministered questions
        Questions questions_Ext = new Questions();
        List<Question> questions =
Manager.GetUnAdministeredQuestions(CATEnvironment.DefaultCandidate.ID
);
        List<Question> qs = new List<Question>();

        Question_Extended q_ext = null;
        foreach (Question q in questions)
        {
            double curr_info = GetInformation(ability_level, q);
            q_ext = new Question_Extended(q.ID, q.QuestionName,
q.Difficulty.Value, q.Discrimination.Value, q.Guessing.Value,
q.TopicID.Value, curr_info);
```

```
            questions_Ext.Add(q_ext);
        }
        questions_Ext.Sort("Information", false);
        Question q_first = questions_Ext.First();
        return Manager.GetQuestion(q_first.ID);


    /// <summary>
    /// Estimate the ability level based on the responses given by
the candidate.
    /// This is an iterative process performed every time submits a
response
    /// </summary>
    /// <param name="previous_estimated_ability"></param>
    /// <returns></returns>
    public static double GetEstimatedAbilityLevel(double
previous_estimated_ability, int candidate_id)
    {
        double dTotalX = 0, dTotalY = 0;
        double p = 0, q = 0, numerator = 0, denominator = 0;
        //Get answered questions
        List<Question> questions =
Manager.GetAdministeredQuestions(candidate_id);
        foreach (Question question in questions)
        {
            p = GetProbability(previous_estimated_ability, question);
            q = 1 - p;
            int answeredCorrectly = new
Result(question).Choice.IsAnswer.Value == true ? 1 : 0;
            numerator = question.Discrimination.Value *
(answeredCorrectly - p);
            denominator = Math.Pow(question.Discrimination.Value, 2)
* p * q;

            dTotalX += numerator;
            dTotalY += denominator;
        }
        //double new_ability_Level = previous_estimated_ability +
(numerator / denominator);
        double new_ability_Level = previous_estimated_ability +
(dTotalX / dTotalY);
        //ability level ranges from -3 to +3
        if (new_ability_Level > 3)
            return 3;
        if (new_ability_Level < -3)
            return -3;
        //else return value if between -3<=value<=3
        return new_ability_Level;
    }

    /// <summary>
    /// Computes the Std Error:
    /// The standard error is a measure of the variability of the
    /// values of θ^(estimated ability) around the examinee’s unknown
Parameter value θ(actual examinee ability)
    /// </summary>
    /// <param name="previous_estimated_ability"></param>
    /// <param name="candidate_id"></param>
    /// <returns></returns>
    public static double GetStandardError(double
Previous_estimated_ability, int candidate_id)
    {
        double dTotalX = 0, dTotalY = 0;
        double p = 0, q = 0, numerator = 0, denominator = 0;
        //Get answered questions
        List<Question> questions =
Manager.GetAdministeredQuestions(candidate_id);
        foreach (Question question in questions)
        {
            p = GetProbability(previous_estimated_ability, question);
            q = 1 - p;
            int answeredCorrectly = new
Result(question).Choice.IsAnswer.Value == true ? 1 : 0;
            numerator = question.Discrimination.Value *
(answeredCorrectly - p);
            denominator = Math.Pow(question.Discrimination.Value, 2)
* p * q;

            dTotalX += numerator;
            dTotalY += denominator;
        }
        double stderror = 1 / Math.Sqrt(dTotalY);
        return stderror;
    }
    /// <summary>
    /// Convert a list of questions into a list of Extended Question
objects
    /// Causes an initialization of the information parameter in the
destination object
    /// </summary>
    /// <param name="questions"></param>
    /// <returns></returns>
    public static Questions TypeCast(List<Question> questions)
    {
        Questions qs = new Questions();
        Question_Extended q_ext;
        if (questions != null)
        {
            foreach (Question q in questions)
            {
                q_ext = TypeCast(q);
                if(q_ext != null)
                    qs.Add(q_ext);
            }
        }
        return qs;
    }

    /// <summary>
    /// perform conversion from Question to Question_Extended object
    /// </summary>
    /// <param name="question"></param>
    /// <returns></returns>
    public static Question_Extended TypeCast(Question question)
    {
        if (question == null)
            return null;
        return new Question_Extended(question.ID,
question.QuestionName, question.Difficulty.Value,
question.Discrimination.Value, question.Guessing.Value,
question.TopicID.Value, 0);
    }

    #region Evaluate Stopping Conditions
    public static bool IsStopConditionSatisfied(int candidate_id)
    {
        return IsStoppingConditionSatisfied(candidate_id);
    }

    /// <summary>
    /// Called when examinee wants to terminate the test or when the
system wants to determine whether it
    /// should stop automatically.of course if the system 'feels'
that it has estimated the examinee's ability
    /// to precision
    /// </summary>
    /// <param name="candidate_id"></param>
    /// <returns></returns>
    private static bool IsStoppingConditionSatisfied(int
candidate_id)
    {
        bool bSatisfied = false;
        //checking that min questions r adminstered
       List<Question> questions =
Manager.GetAdministeredQuestions(candidate_id);
       List<Question> all_questions = Manager.GetQuestions();

        try
        {
            //check if all questions administered
            if (questions.Count >= all_questions.Count)
                return true;

            //check if max test length reached
            Setting test_length_setting = Manager.GetSetting(2);//2
is pre-defined in the db
            if (test_length_setting != null &&
test_length_setting.Active.Value)
            {
                if((questions.Count * 100)/ all_questions.Count >=
Convert.ToDouble(test_length_setting.Value))
                    bSatisfied = true;
            }

            //check the status of std_error_setting
            Setting std_error_setting = Manager.GetSetting(3);//3 is
pre-defined in the db
            if (std_error_setting != null &&
std_error_setting.Active.Value)
            {
                //get stderror as at the last question administered
entry information
                Summary summary = Manager.GetSummary(candidate_id,
questions.Count);
                if (summary.CurrentStdError.Value >=
Convert.ToDouble(std_error_setting.Value))
                    bSatisfied = true;
            }
        }
        catch { }
        if (bSatisfied)
            //if satisfied then check that conditions b4 stoppn have
been met
            return AreConditionsB4StoppingSatisfied(candidate_id,
questions, all_questions);
```

```
    else
      return false;
  }

  /// <summary>
  /// check that conditions that have to be met b4 terminating are
met
  /// </summary>
  /// <param name="candidate_id"></param>
  /// <param name="questions_admind"></param>
  /// <param name="all_questions"></param>
  /// <returns></returns>
  private static bool AreConditionsB4StoppingSatisfied(int
candidate_id, List<Question> questions_admind, List<Question>
all_questions)
  {
    bool satisfied = false;
    //check that every topic area has been covered if the setting
is active
    Setting topic_area_setting = Manager.GetSetting(5);//5 is
pre-defined in the db
    if (topic_area_setting != null &&
topic_area_setting.Active.Value)
    {
      List<Topic> topics = Manager.GetTopics();
      if (topics != null && topics.Count > 0)
      {
        //get the handled topics
        var handled_topics = (from s in questions_admind
              select s.TopicID).Distinct();

        //topics administerd more or equal to the percentage
value specified
        if ((handled_topics.Count() * 100) / topics.Count >=
Convert.ToInt32(topic_area_setting.Value))
          satisfied = true;
      }
    }
    //check that minimum number of questions have been
administered
    Setting min_percentage_setting = Manager.GetSetting(4);//4 is
pre-defined in the db
    if (min_percentage_setting != null)
    {
      if (questions_admind != null && all_questions != null)
      {
        double percent = (questions_admind.Count * 100) /
all_questions.Count;
        if (percent >=
Convert.ToDouble(min_percentage_setting.Value))
          satisfied = true;
      }
    }
    return satisfied;
  }
  #endregion

  #region results
  /// <summary>
  /// Compute Percentage given the ability level
  /// </summary>
  /// <param name="ability_level"></param>
```

```
    /// <returns></returns>
    public static double ComputePercentage(double ability_level)
    {
        return ((ability_level - (-3)) * 100) / (+3 - (-3));
    }

    #endregion}
```

2. Candidateresults.aspx.cs - This is for displaying results to the administrator

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DataAccess;


public partial class Admin_CandidateResults : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if
(!GeneralUtils.DoesUserHaveRight(CATEnum.GetPageID(PageEnum.Candidate
_Result_Page)))
            Response.Redirect("~/AccessError.aspx");//redirect the
administrator if he tries accessing this page

        if (!IsPostBack)
        {
            ColdBind();
            Page.Title = GeneralUtils.AppName + "View Candidates
Results";
        }
    }

    /// <summary>
    /// Show questions and answers by candidate
    /// </summary>
    protected void ColdBind()
    {
        PopulateTree();
    }

    protected void PopulateTree()
    {
        List<TestCandidate> testCands = new
List<TestCandidate>();//track no of candidates per test
        TestCandidate testCand = new TestCandidate();
        testCand.TestName = "Test One";
        testCand.CandidateCount = 0;

        TreeNode parentNode = GeneralUtils.CreateTreeNode("Test 1",
"Test 1", "Expand to see the results for each candidate",
"~/Icons/Test.png");

        parentNode.SelectAction = TreeNodeSelectAction.None;//disable
selection of the root node

        List<Candidate> cands = Manager.GetCandidates();
        if (cands != null)
        {
            foreach (Candidate cand in cands)
            {
                testCand.CandidateCount++;//increment no of
candidates for the test
                //load all candidates who undertook the test

parentNode.ChildNodes.Add(GeneralUtils.CreateTreeNode(cand.CandidateN
ame, cand.ID.ToString(), "Click Here to See the candidate's result",
"~/Icons/User.png"));
            }
        }
        TreeView1.Nodes.Add(parentNode);

        testCands.Add(testCand);
        GridView2.DataSource = testCands;
        GridView2.DataBind();
    }

    protected void TreeView1_SelectedNodeChanged(object sender,
EventArgs e)
    {
        switch (TreeView1.SelectedNode.Depth)
        {
            case 0://root node
                //do nothing as its not selectable
                break;

            case 1://candidate selected

HotBind(Convert.ToInt32(TreeView1.SelectedNode.Value));//show results
for selected candidate
                break;

            default:
                break;
        }
    }

    protected void HotBind(int candidate_id)
    {
        List<Question> questions =
Manager.GetAdministeredQuestions(candidate_id);
        if (questions == null)
            questions = new List<Question>();

        List<ResultExtended> results = new List<ResultExtended>();
        foreach (Question question in questions)
        {
            results.Add(new ResultExtended(new Result(question),
candidate_id));//evaluate questions
        }
        GridView3.DataSource = results;//show results
        GridView3.DataBind();

    }
}
```

3. Rankcandidates.aspx.cs - For ranking the students

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using DataAccess;

public partial class Admin_RankCandidates : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if
(!GeneralUtils.DoesUserHaveRight(CATEnum.GetPageID(PageEnum.Candidate
_Result_Page)))
            Response.Redirect("~/AccessError.aspx");//redirect the
administrator if he tries accessing this page

        if (!IsPostBack)
        {
            ColdBind();
            Page.Title = GeneralUtils.AppName + "Rank Candidates";
        }
    }

    /// <summary>
    /// Show test details
    /// </summary>
    protected void ColdBind()
    {
        PopulateTree();
        HotBind();
    }

    protected void PopulateTree()
    {
        TreeNode parentNode = GeneralUtils.CreateTreeNode("Test 1",
"Test 1", "Click to View the Ranking of Candidates For The Selected
Test", "~/Icons/Test.png");
        TreeView1.Nodes.Add(parentNode);
        parentNode.Select();
    }

    protected void TreeView1_SelectedNodeChanged(object sender,
EventArgs e)
    {
        switch (TreeView1.SelectedNode.Depth)
        {
            case 0://root node
                HotBind();
                break;

            default:
                break;
        }
    }

    protected void HotBind()
    {
        try
        {
            RankClassCollection rankColl = new RankClassCollection();
            List<Candidate> cands = Manager.GetCandidates();
            foreach (Candidate cand in cands)
            {
                List<Question> questions =
Manager.GetAdministeredQuestions(cand.ID);
                if (questions != null)
                {
                    List<ResultExtended> results = new
List<ResultExtended>();
                    foreach (Question question in questions)
                    {
                        results.Add(new ResultExtended(new
Result(question), cand.ID));//evaluate questions
                    }
                    SummaryList summs = new SummaryList();

                    foreach (ResultExtended r_ext in results)
                    {
                        if(r_ext.SummaryInfo != null)
                            summs.Add(new
Summary_Stripped(r_ext.SummaryInfo.ID,
r_ext.SummaryInfo.ItemsAdministered.Value,
r_ext.SummaryInfo.CurrentAbilityEstimate.Value));
                    }
                    if (summs != null && summs.Count > 0)
                    {
                        summs.Sort("ItemsAdministered", false);
                        RankClass rank = new RankClass();
                        rank.Ability =
summs.First().CurrentAbilityEstimate.Value;
                        rank.candidate = cand;

                        rankColl.Add(rank);
                    }
                }
            }
            rankColl.Sort("Ability", false);

            for (int i = 0; i < rankColl.Count; i++)
            {
                rankColl[i].Rank = i + 1;
            }
            GridView1.DataSource = rankColl;
            GridView1.DataBind();
        }
        catch { }
    }
}
```