5-1-2013

Download Question and Section Results Output for INetTest System

Pradeep Jagannath Rohokale
Utah State University

DOWNLOAD QUESTION AND SECTION RESULTS OUTPUT FOR INetTest SYSTEM

by

Pradeep J. Rohokale

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

_____	_____
Dr. Donald Cooley	Dr. Daniel Watson
Major Professor	Committee Member

_____
Dr. Curtis Dyreson
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah
2013

ABSTRACT

Download Question and Section Results Output for iNetTest System

By

Pradeep J. Rohokale, Master of Science
Utah State University, 2013

Major Professor: Dr. Donald Cooley
Department: Computer Science

iNetTest is a useful tool to help instructors deliver computer-based tests, but when an instructor gives a test, they would often like to know individual question results, e.g.

- What was the average score for a question, how many got it correct and how many missed it,
- What was the most common answer chosen (for multiple choice, T/F, etc.),
- What was the score for a section?
- What was the average score for a section?
- etc.

Whenever individuals consider such data they often prefer different or multiple ways in which to analyze the data. For that reason, it was decided not to try to build a "Do everything for everyone" type module. We also considered simply extending the statistics module in iNetTest but felt we would never be able to add all of the capabilities to meet all users' requests. Thus it was decided to give to an instructor a file that could be loaded into Excel, and they could then generate their own statistics. Generating such a file, so that a user could extract from it, whatever statistics they desired was still a significant challenge. The option added to iNetTest became the "Download Question and Section Results for a Test" option. Ultimately, it will be the instructor's responsibility to load such a file into a spreadsheet program like Excel and then generate the desired statics, plots, etc.

CONTENTS

LIST OF TABLES

Table Page

List of Figures

Figure Page

CHAPTER I
INTRODUCTION

Automated Testing

Automated testing software tools help to reduce the amount of time an instructor must spend in creating and especially in grading class assessments such as tests. The grading process for a test can be monotonous as well as very time consuming. Computers are excellent at performing such tasks, but may pose security issues. For example, with a computer, students have access to the Internet and the wealth of information available through powerful search engines. Another security concern would be protecting the content of the exam so that it is only available to authorized test takers.

The iNetTest system is a web-based computer aided testing system. INetTest uses current J2EE technologies running within a JBoss Application Server 6. It uses struts to manage the business logic between the client and the server. The database interface is implemented using Enterprise Java Beans to access a postgreSQL database. The user interface utilizes the current jQuery tools to provide a dynamic web environment. Instructors using iNetTest can house the creation, editing, administration, and grading of a test over the Internet. The software allows the instructor to specify the time span for a test and the IP range for the test taking computers, restricting when and where an exam may be taken. The iNetTest system also includes software to monitor any web browsing that is done during the test. Because it is a web-based application, iNetTest can be accessed from any web-enabled computer but restricts access to certain test information to authorized users only. INetTest has a complex and flexible security environment allowing for various levels and degrees of access.

INetTest allows for ten different types of questions. The various question types allow the instructor significant flexibility and accuracy in assessing student learning. Nearly all question types are automatically graded with only essay questions to be manually graded by the instructor. Automatic grading of questions can be overridden by the instructor as circumstances require. For example, if a question is inherently flawed or the correct answer is mislabeled, the instructor may choose to re-grade it. These capabilities combined with ease of test creation, delivery, and taking make iNetTest a powerful tool for instructors to use in delivering and administering exams to students. INetTest was designed to make the testing process as straightforward as possible for both the instructor and the student. Considering all of the easy to use features implemented in iNetTest, it is an excellent candidate for a starting point in the development of software to manage a computer aided testing center.

CHAPTER II
PROBLEM ANALYSIS

Currently, under the new administrator interface, there is an option to download scores for the tests taken to date by a specified group. The file downloaded is a .csv file showing scores for all of the selected tests associated with that group. If an individual does not have a score for a particular test, it shows as #NA. In addition to individual test scores, when an instructor gives a test, they would often like to know individual question scores, e.g. what was the average score for a question, how many scored correct and/or incorrect, what was the most common answer chosen (for multiple choice, T/F, etc.), what was the score for a section what was the average score for a section, etc.

In addition to wanting/requiring such data, instructors may envision multiple ways in which they would like to analyze the data. Incorporating all of the possible different views of the data into a single option, would have been infeasible. For this reason, it was decided to develop an option which would generate a rich set of data which could then be manipulated by the instructor to generate the desired question information. Ultimately, it will be the instructor's responsibility to load the .csv file generated by the option into a spreadsheet program like Excel and generate the desired statistics, plots, etc.

CSV file output format

1. CSV file header

   At the beginning of a .csv file is a header. This header contains labeling information to assist the user to understand the question abbreviations used in the display of the question answer data. The header information is as given in Table 1. As an example from Table 1, for a multiple choice type question the label is MC.

Test Title	
Question Labels	
True/False	TF
Fill in the Blank	FIB
Matching	MTCH
Multiple Answer	MA
Multiple Choice	MC
Multiple Question	MQ
Calculated	CALC
Essay	ESY
Sequencing	SQ
Programming	PROG
[(section title)(# questions)(# selected)][(question type)(points)(# answers) (answer(s))]	

Table 1: Header Information in CSV File

2. CSV file data

Following the labeling information at the head of the .csv file is the actual question answer results for each student. The following example is given to explain how data is populated into the .csv file. For this example, in each section there will be 10 questions and each question will be worth 5 points. All appropriate questions can have partial credit given. Partial credit can be given by the instructor when they grade as for essay questions, or can be done automatically by iNetTest as in a matching question. The order of the questions in each section in the example will be the same (see Table 2)

Question #	Question Type	# of correct answers in file
Q1	T/F (TF)	1
Q2	Fill in the Blank (FIB)	0
Q3	Matching (MTCH)	>1
Q4	Multiple Answer (MA)	>1
Q5	Multiple Choice (MC)	1
Q6	Multiple Question (MQ)	0
Q7	Calculated (CALC)	1
Q8	Essay (ESY)	0
Q9	Sequencing (SQ)	>1
Q10	Programming (PROG)	0

Table 2: Question Abbreviations and Number of Answers for iNetTest Questions

Note: As shown in table 1, the .csv spreadsheet header includes a row labeled [(section title)(# questions)(# selected)] [(question type)(points)(# answers) (answer(s))]. This label defines the grouping of columns associated with sections and questions within sections, and the syntax used. The "[(section title)(# questions)(# selected)]" begins each section and within in each section the "[(question type)(points)(# answers) (answer(s))]" label defines each question.

Section display format: [(section title)(# questions)(# selected)]

The section label is displayed at the start of each section. The section label is simply the section title set when the test was created. This label is then followed by two numbers. The first value "# number questions" is the total number of questions in the section, not including disabled questions. The second value "# selected" is the number of questions to be randomly selected from the section. In the sample test, the first section is titled "Programming" and there are 10 questions of which 8 are randomly selected. To maintain consistent column structure for each student, the "section title", "# questions", and "# selected" is given in each row.

For example on section syntax consider [Data structure, 20, 15] where Data structure is the name of the section, 20 is the number of questions in the section and 15 is the number of selected questions.

Question display format: [(question type)(points)(# answers) (answer(s))]

Following the section information columns, there are columns giving information about the individual questions in that section. "Question type" is the type of the question and hence is one of the labels of Table 1, e.g. MA for a multiple answer question. The "points" column displays the number of points available for that question, i.e. for a correct answer or full credit. The "# answers" column is the number of answers for that question and the "answer(s)" column(s) is the actual correct answer(s) for that question. If there exists one or zero correct answers for a question, this column is not included. For example, the number of answers for a T/F question is 1 and thus a count for the number of answers is not supplied since it is fixed for all T/F questions. However, since the number of answers for a multiple answer question is simply >1, a value must be supplied.

For example, CSV file entry [MA, 5, 3, T, T, F, F, T]

MA – Multiple answer question, 5 – points on the question, 3 – number of correct answers and T, T, F, F, T represents the answers(T – for correct answer, F – false answer).

The number of answers for an essay question, even though an essay must be supplied, is viewed as 0. Likewise, a programming question, which can have a function or even several blanks to fill in, is viewed as having 0 answers.

In addition to generating information about individual question answers, there were three iNetTest functions that presented an added challenge for this project.

- Randomization of questions and/or sections

  When a test is delivered to a student, if randomization is used, the order of questions and sections may be different from other students. The database maintains the student test answers in the order in which they were presented to the student. Thus, in order to present the summarized data, we had to link the order of a student's questions/answers with the actual order of questions and sections in the base test. Once this was done, the .csv file output could be set, i.e. what is presented in the .csv file MUST be the same order for all students, and that order is the order of the sections and questions as defined at the time the test was created.

- Randomization of answers

  Why is there no header here, like in the previous section? For full randomization, iNetTest randomizes the answer choice order. This would apply to multiple choice questions, matching questions, sequencing questions, etc. in addition to the randomization of the section and question orders already mentioned. Thus for different students the correct answer may have a different label. For example, on a multiple choice question for one student, the correct answer may be 'A' and on another it may be 'E'. The iNetTest database maintains the student test answer choices in the format that was displayed to him/her. Thus, where ordering can change, a student's answer should be translated back to the original ordering of the question. This means that for a multiple choice question, the correct answer is always "A", etc.

- Selected Questions

  In addition to differences in orderings, iNetTest provides functionality to ask random questions from a section, i.e. not all questions will be asked. Hence there is a possibility that some questions in a section may not be asked to all the students. Therefore, in the .csv file for a student's test result row, the entries for such questions are listed as "NA."

Question Answer explanation

1. True/False **(TF)**

   For a True/False question, a letter T or F is given to denote the correct answer. Thus for a TF question worth 5 points with a correct answer of T, the "Definition columns" entry in the row defining that question would be:

   Definition columns TF 5 T

   While the entry for each student would be:

Student answer columns	TF	0/5	F/T	
i.e.	TF	0	F	if they answered the question incorrectly or
	TF	5	T	if they answered the question correctly
	TF	0	NA	if they do not answer the question

No partial credit is given for the wrong answer to a TF question.

2. Fill in the Blank **(FIB)**

No answer is given for a FIB question. This is because a FIB question can have a variable number of blanks with a possibility of more than one correct answer for a blank. Therefore, we won't show either the correct answer(s) or the student's answer(s). We only show the points awarded. In time, as an extension, we may wish to expand the data for this question type to include the answer(s) given by the student and whether it was correct or incorrect.
Consider a FIB question worth 5 points with two blanks giving partial credit. The entry in "Definition columns" row for this question would be:

Definition columns	FIB	5

While, entry for each student would be

Student's columns	FIB	0-5	(i.e. a score between 0 and 5 points would show)

3. Matching **(MTCH)**

On the left column of a matching question (as displayed in iNettest) are the terms or phrases to match with and a box next to each term or phrase into which the student gives a number representing which term or phrase in the right column is the correct match. Thus, for a matching question with 4 items to match the answer is simply an ordered list of the numbers in the boxes. The problem is because of randomization for each student there may be a different order for the matches. At the time of authoring a matching question in iNettest, the matching items are placed next to one another, i.e. item 1 in the left column matches with item 1 in the right column. For example, if there are four matches to give, when the question is defined by the instructor, the term or phrase in row one, column 1 matches with the term or phrase in row 1 column 2, etc. Therefore, the correct sequence for the matches would be 1,2,3,4 in this initial definition. If a student gets the answer correct, then their answer should also show as 1,2,3,4 even though the answers on their test may be different. As an example, assume a matching question is defined as follows:

Match the word on the right with the closest definition on the left:

Four legged animal	*1. Dog*
Found in the ocean	*2. Shark*
Very large mammal	*3. Whale*
Can fly for long distances	*4. Eagle*

Assume a student (Joey) has answered this question displayed as follows: (Note that in this case it is the "best" match and thus, even though a whale is found in the ocean the best choice is that it is a very large mammal thus leaving shark for found in the ocean since it is not a mammal.)

Match the word on the left with the closest definition on the right:

1 Four legged animal	*1. Shark*
2 Found in the ocean	*2. Whale*
4 Very large mammal	*3. Eagle*
3 Can fly for long distances	*4. Dog*

If partial credit for this question was given, since the only correct answer is "Can fly for long distances = eagle", the points awarded would be (1/4)*5 = 1.25 points. This is because only one of the 4 answers is correct. The entry for a correct answer to this question would be:
MTCH,5,4,1,2,3,4 (it will always be pts 1,2,…,n where n is the number of items to match)
5 – points possible, 4 – number of answers, and 1,2,3,4 is the correct answers.

The answer options are randomized and given to Joey for the test. According to the Joey's answers order his answers are 1,2,4,3. Now Joey's answer order is mapped back to the original answer order. According to the original answer order his answers will be 2,3,1,4. Hence the entry for the student (Joey) in the CSV file, rather than being [MTCH,1.25,4,1,2,4,3] would be [MTCH,1.25,4,2,3,1,4]

Also note that if a student does not give a match for one of the items on the left, the student's answer would show as NA which would be considered incorrect. For example:

Match the word on the left with the closest definition on the right: The question is backwards or the answers are backwards

1 Four legged animal	*1. Shark*
_ Found in the ocean	*2. Whale*
4 Very large mammal	*3. Eagle*
3 Can fly for long distances	*4. Dog*

MTCH,1.25,4,2,NA,1,4

If the question is not answered the student entry will be MTCH,0,4,NA,NA,NA,NA

4. Multiple Answer **(MA)**
   A multiple answer question is similar to a multiple choice question except rather than *one* correct answer, there is *at least one* correct answer. Consider the following multiple answer question:

   Which cities are in Utah?

   __Salt Lake City

   __New York

   __Logan

   __Philadelphia

   The entry in the CSV file would be: MA,5,4,T,F,T,F

   If a student Smith answered:

   _*X*_ Logan

   ___ Salt Lake City

   _*X*_ Philadelphia

   __ New York

   The answer entry in the CSV for this multiple answer question for student Smith would be:

   MA,2.5,4,F,T,T,F (Only the first and third entries are correct.)

   (Two of four answers are correct, hence the point awarded are 2/4*5 = 2.5) If a student did not give any correct answers to a question, they would receive 0 points and have a profile of:

   MA,0,4,F,T,F,T (no partial credit)

   MA,2.5,4,F,T,F,T (partial credit)

   It is interesting to note that if they answered as follows:

   _*X*_ Logan

   ___ Salt Lake City

   ___ Philadelphia

   ___ New York

   They would receive a score of MA,3.75,4,F,F,T,F

   If the question is not answered the student entry will be MA,0,4,F,T,F,T

5. Multiple Choice **(MC)**

For this question there is only one correct answer. Because of randomization we face the same issues as for the matching and multiple answer questions. For most multiple choice questions, the correct answer (before randomization) is generally A. Unfortunately, when the correct answer is something like "all of the above", that answer may be the last in the list. It is also the case that the instructor can choose to place a particular answer in a specific location. Under those circumstances, the correct answer can be fixed in any location. However, at the definition of the question/answers, before randomization, everything will be fixed or ordered and it is that order that will be used. The following is an example of a multiple choice question:

The sum of 5 and 12 is:

In authoring the question the author had set up four answers as follows:

(1) Answer: 17 (Fixed by the author to position B)
(2) 22 (fixed by the author to D)
(3) 5
(4) none of the above (fixed by the instructor to E)
(5) 23

Thus the possible versions of this question would be:

Version 1	Version 2
A. 5	A. 23
B. 17	B. 17
C. 23	C. 5
D. 22	D. 22
E. none of the above	E. none of the above

Table 3: Versions of randomized MC question

Even though the specific location of an answer, unless the author chooses a specific position, is random, at the time of question definition the positions are fixed. This fact is used in displaying the results. Thus, even though the answer may be "none of the above" and thus when it is displayed to the user it will be the last answer; we translate answers back to the sequence given in the definition of the question, ignoring an author's position fixes.

As an example, look at the two possible versions for the display of the question, i.e. versions 1 and 2. Let's assume that the student has chosen 23 (which is not correct) as their answer. That means for version 1 their answer would be C or position 3 and for version 2 their answer would be A or position 1. By translating everything back to the original definition, it means that in both cases their answer would be position 5. This means that for every multiple choice question, the correct answer when translated in this fashion is always position 1. Also, since the number of correct answers is one, the # answers field is not included for this question type.

Assume that both Smith and Joey received this question but Smith received version 1 and Joey version 2. Also, assume that Joey's answer was A (23, incorrect) and Smith's was B (17, correct). Their entries would be:

(Joey) MC 0,5
(Smith) MC 5,1

If the question is not answered the student entry will be MC,0,NA

6. Multiple Question **(MQ)**
   The multiple question (MQ) question is actually an essay question with multiple topics to choose to write the essay on. This question must be graded manually and it has a single score. What the student sees is a choice of >1 questions to answer. The student selects which of the N questions to answer and then supplies their answer (essay) in the field provided. If we assume that the essay is worth 5 points (the same points must be awarded for each of the questions), then the entry for the definition would be:

   Definition columns MQ 5 N

   {5 => maximum points for the answer; N => number of options or questions to choose from}

   The entry for each student would be

   Student's columns MQ 0-5 1-N

   {0-5=> points awarded; 1-N=> a value from 1 to N indicating which of the questions was selected or NA if the student did not answer the question}

7. Calculated Question **(CALC)**
   A calculated question is one for which the student performs a calculation and supplies as their answer the result of that calculation. If we assume that the question is worth 10 points, then the entry for the definition would be:

   Definition columns CALC 10 N
   {10 => points awarded for a correct answer; N => the correct answer or value for the calculation, i.e. the answer}

   The entry for each student would be

   Student's columns CALC 0-10 M

   {0-10 => points awarded for the student's answer; M => M is a number, the student's actual answer or NA if the student did not answer the question}

8. Essay Question **(ESY)**
   The essay question is simply a question on which students are required to write an essay. This question must be graded manually and it has a single score. If we assume that the essay is worth 5 points, then the entry for the definition would be:

   Definition columns ESY 5 {5=> maximum points for the answer}

The entry for each student would be

Student's columns ESY 0-5

{0-5 => a value from 0 to 5 indicating the points awarded for the student's answer}

9. Sequencing Question **(SQ)**

The sequencing question is a variation of the matching question. It is automatically graded. For a sequencing question the student is given a question like "In what order should one …" They are then given 2-N steps with a box next to each step. So if N (the number of steps) was 5, there would be 5 steps shown in some order with a box to the left of each step. In each box they would put a value of 1 to 5 (no repeats). As with the multiple choice (MC) question, the steps are randomized. In this case, we will proceed similarly to the MC question. In the sequence of steps supplied by the instructor the first one (#1) is the first step and the second one supplied (#2) is the second step, etc. Let's assume that the three steps in the sequence are:

Check for breathing

Check for bleeding

Check for broken bones

Thus the correct answer for this sequence, if given as shown would be 1, 2, 3.

If, when the question is generated, the sequence looks like:

____ Check for bleeding

____ Check for broken bones

____ Check for breathing

A correct answer for this sequence would be 2, 3, 1, i.e. "check for breathing" would have the value 1 placed beside it (actually in a box beside it rather than a "___". Like the multiple choice question we must translate the student's answer so that it corresponds with the sequence in the question definition. There is also the issue of partial credit being awarded. As an example, using the correct sequence definition as given above (Check for breathing, check for bleeding, check for broken bones); a student's answer of 2, 1, 3, i.e. 2 for "Check for bleeding", 1 for "Check for broken bones", and 3 for "Check for breathing" should award partial credit of 1/3 of the points possible.

Definition columns SEQ 5 {5=> maximum points for the answer}

The entry for each student would be

Student's columns SEQ 0-5

{0-5 => a value from 0 to 5 indicating the points awarded for the student's answer}

Using the example sequence question above for both Joey and Smith, if their answers were:

Joey 2,3,1 translates to 1,2,3 all correct 5 points

Joey's actual entry: SEQ,5,3,1,2,3

SEQ – Sequencing question type, 5 – points, 3 – number of answers, 1,2,3 – Joey's answers

Smith 1,3,2 translates to 3,2,1 one correct 1/3*5 = 1.67 points

Smith's actual entry: SEQ,1.67,3,1,3,2

SEQ – Sequencing question type, 1.67 – points, 3 – number of answers, 1,3,2 – Smith's answers

If the question is not answered the student entry will be MA,0,3,NA,NA,NA

10. Programming Question **(PROG)**

While there is considerable code to facilitate the grading of a programming question and there are multiple versions of the programming question, in the end, this question must be manually assigned a score. If we assume that the programming question is worth 10 points, then the entry for the definition of a PROG question would be:

Definition columns PROG 5 {5=> maximum points for the answer}

The entry for each student would be

Student's columns PROG 0-5

{0-5 => a value from 0 to 5 indicating the points awarded for the student's answer}

Chapter III

Design

The proposed csv file download software will be an addition to the already functioning iNetTest system. This will allow for the csv file download software to use the already existing features of iNetTest like tests, classes, and various roles that the download software must consider. This chapter discusses the integration of the csv file download software with the current iNetTest system, the specific designs of the three challenges in the system, and an overview of the algorithm used to overcome these challenges in the system.

Integration With iNetTest

The iNetTest system already has several of the components that will be needed for csv download. First, there are already various roles that can be formed in the system. Instructors will require administrator privileges in the system to access the csv download page. Instructor permissions are already set in the system and will only require checking of appropriate permissions on the csv download page which can be done with the already existing permissions model available in the iNetTest software. Features for creating permissions for different roles are already a part of the current iNetTest system.

Classes in the iNetTest system are referred to as groups. The groups and tests that already exist in the iNetTest system can be used for the csv file download software. Knowing the group and test will help the system find the last attempt of all the students. It will be necessary for the instructor to use a test created in iNetTest and attempted by students to be able to download test results in csv file. There are two types of tests that can be created in the iNetTest software, Concept and Performance test. For now we will be only considering the concept tests for csv file download software.

The current iNetTest software has all that is needed for delivering tests except the ability for instructors to download detailed test results in a csv file.

Randomization of sections

When a test is given to a student the order of sections may be different from another student. The database maintains the student test answers in the student's test order. What we want is the actual order of sections to be displayed in the CSV file output (i.e. what is displayed in the CSV file MUST be the same order for all students). The Figure 1 shows how we convert back a student's section order to the actual test section order.

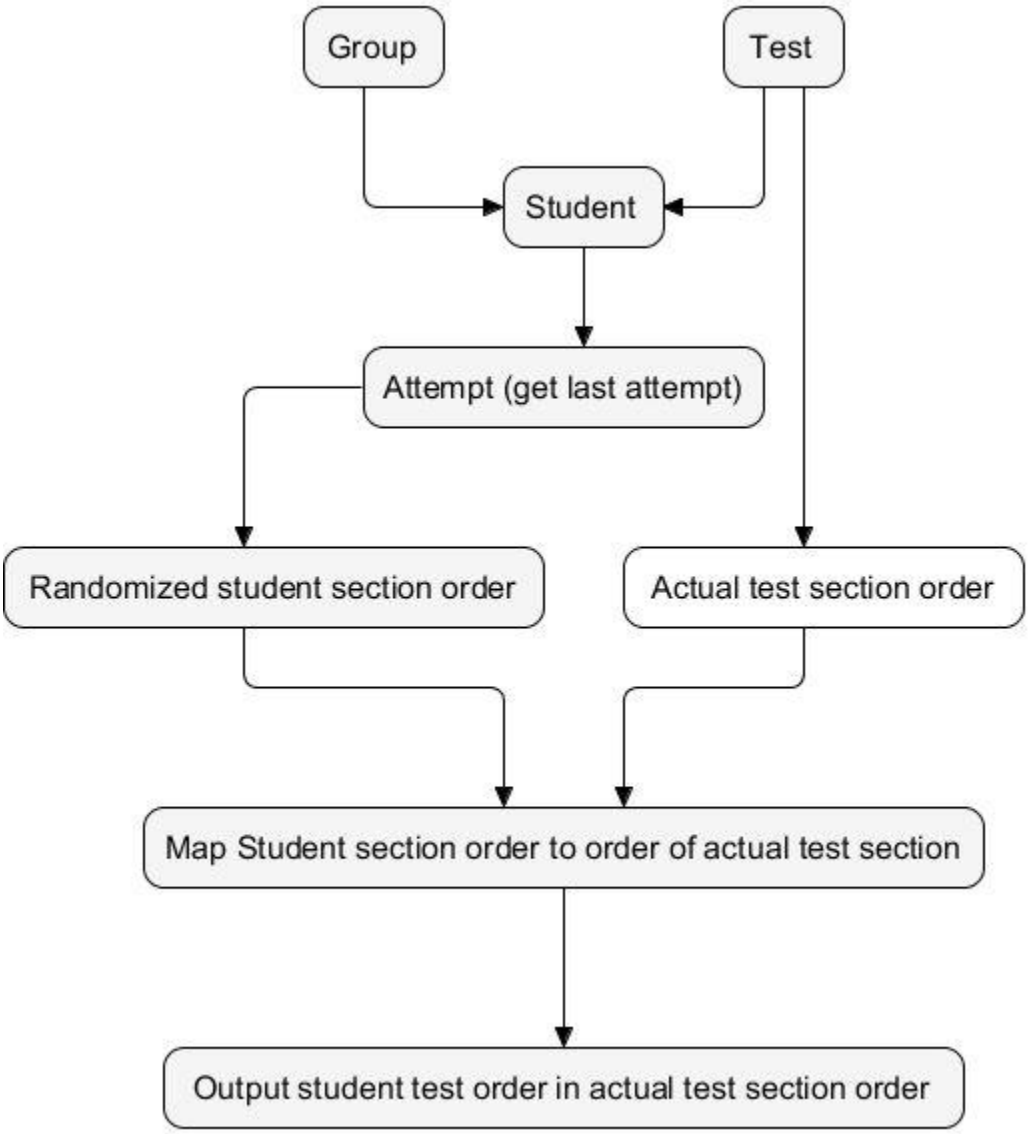

Figure 1: Revert back student test section order to actual test section order

Figure 1 interprets the conversion of the student's section orders to the actual test section order. Initially the system determines the student section order by performing the following steps:

1. Supplying the details of group and the test as the initial input.
2. With the selected group and test, the system determines the number of students taking that particular test.
3. Among the listed students, determine the student section order for each individual student.
4. For a selected student use only their last or most recent attempt.
5. The last attempt is used as a reference ID to extract the student's section order from the database.

The actual test section order is derived from the test database. Once the student section order and test section order are obtained they are mapped together. Once mapped, the result is the student's sections in the same order as the actual test sections.

Randomization of questions

When a test is given to a student the order of questions is randomized and hence different from other students. The database maintains the student test answers in the student's test order. What we want is the actual order of questions to be displayed in the CSV file output (i.e. what is displayed in the CSV file MUST be the same order for all students). Figure 2 shows how we convert the student's question order to the actual test question order.

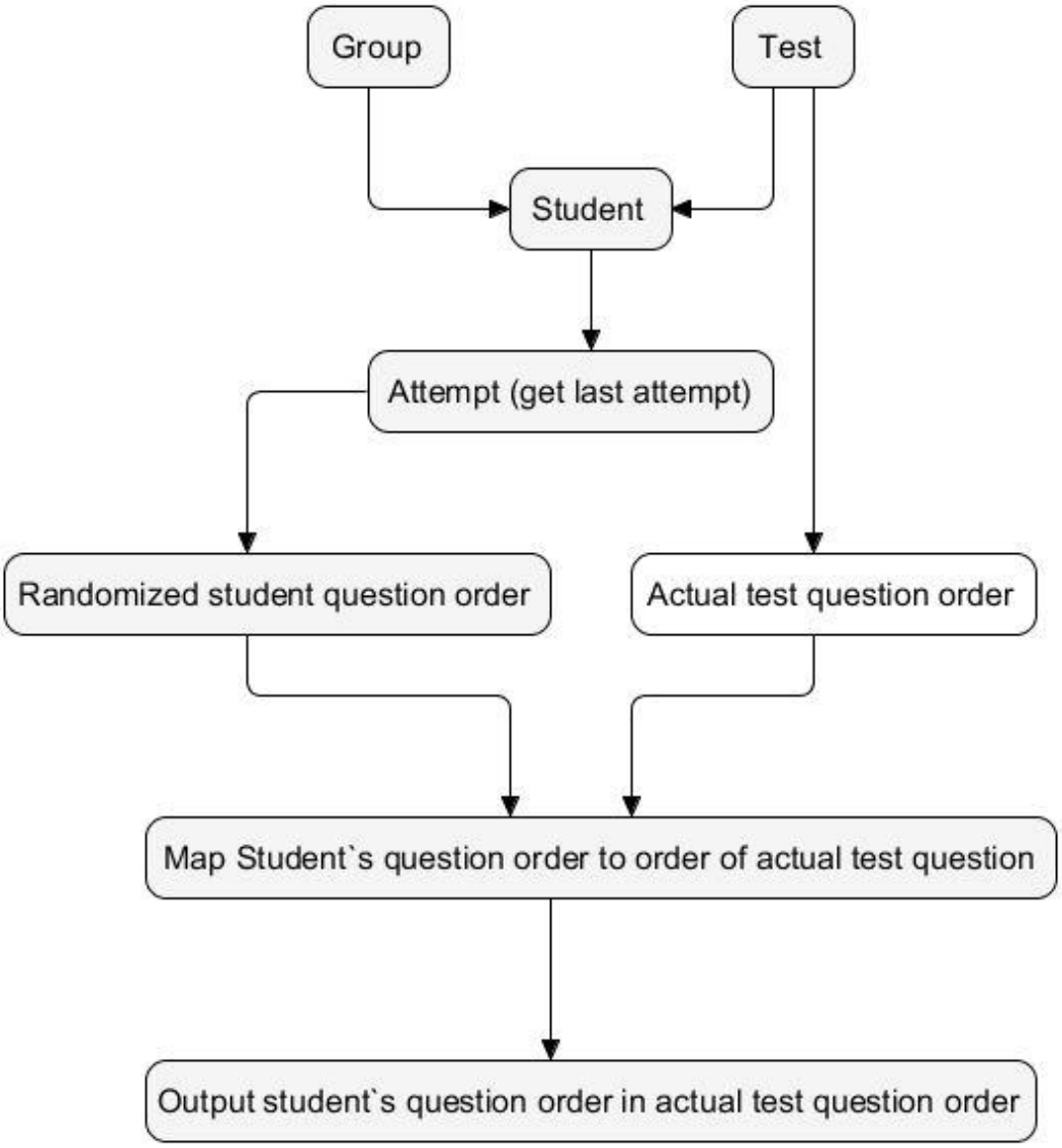

Figure 2: Revert back student's test question order to actual test question order

Figure 2 interprets the conversion of the student's question orders to the actual test question order. Initially the system determines the student question order by performing the following steps:

1. Supplying the details of group and the test as the initial input.
2. With the selected group and test, the system determines the number of students taking that particular test.
3. Among the listed students, we determine the student section order for each individual student as discussed above. For each section of a student we determine the student question order.

4. For a selected student we use only their last attempt or most recent attempt.
5. The last attempt is used as a reference ID to extract the student section order from the database.

The actual test question order for a section is derived from the test database. Once the student question for a section and test question for the same section are obtained, they are mapped together. Once mapped, the result is the student's questions in the same order as the actual test questions.

Randomization of answers

Full randomization randomizes the question answer choice order (e.g. multiple choice questions, matching question, sequencing question, etc.) along with the question order and the section order. Thus, for different students the correct answer may have a different label. For example, on a multiple choice question for one student, the correct answer may be 'A' and on another it may be 'E'. Again, the database maintains the student's test answers choices in the format that were displayed to him/her. Thus, where ordering can change, a student's answer should be translated back to the original ordering of the questions. This means that for a multiple choice question, the correct answer is always "A". Figure 3 shows how we convert the student's answer order to the actual test question order.

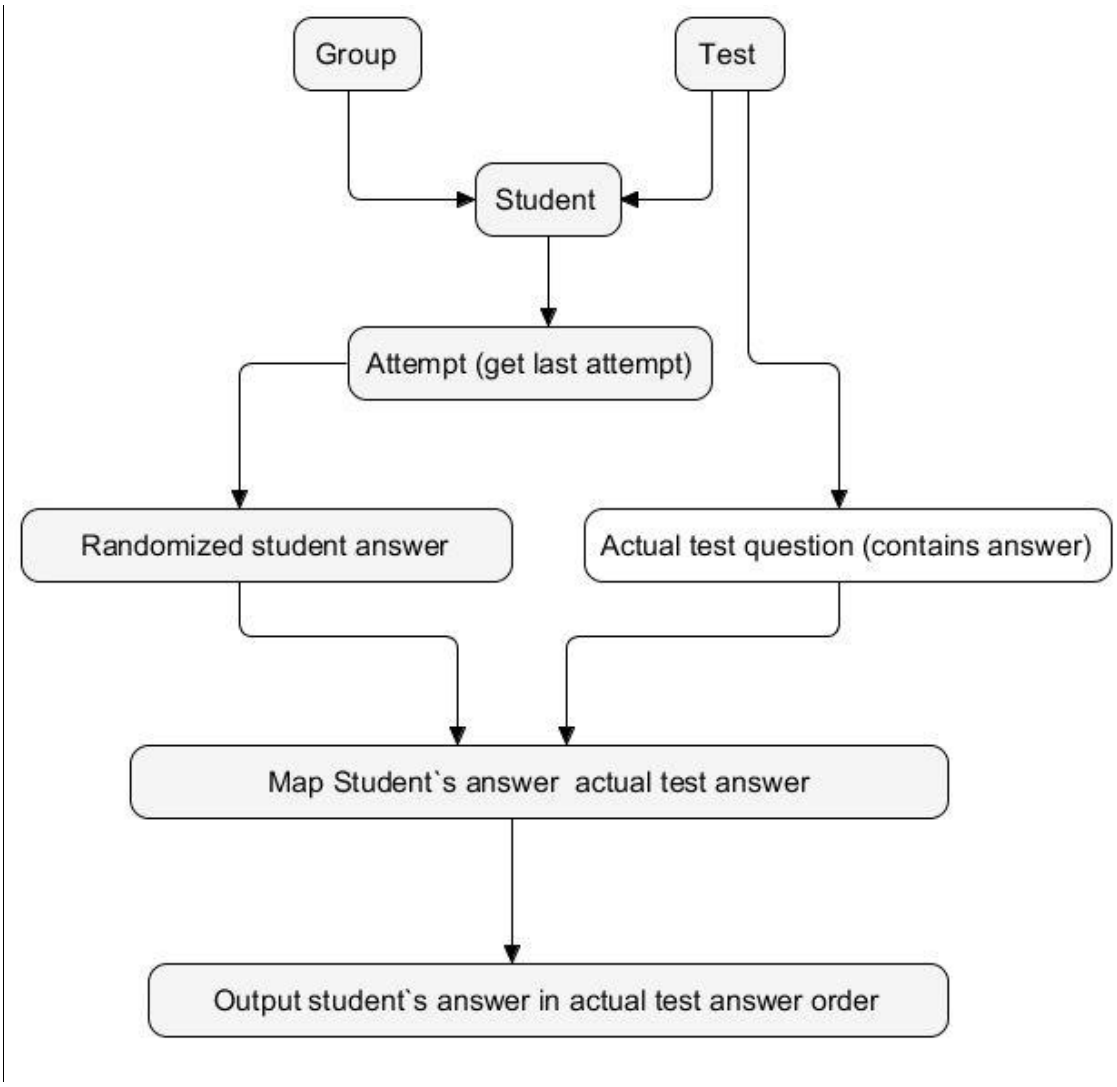

Figure 3: Revert back student's answer order to actual answer order

Figure 3 interprets the conversion of student's question orders to actual test question order. Initially the system determines the student answer order by performing the following steps:

1. Supplying the details of group and the test as the initial input.
2. With the selected group and test, the system determines the number of taking that particular test.
3. Among the listed students, determine the student section order and question order for each individual student as discussed above. For each question of a student we determine the student answer order.
4. For a selected student use only their last or more recent attempt.

5. The last attempt is used as a reference ID to extract the student answer order from the database.

The actual test answer order for a question is derived from the test database. Once the student answer for a question and test answer for the same question are obtained, they are mapped together. Once mapped, the result is the student's answers in the same order as the actual test answers.

Chapter IV

Implementation

This chapter focuses on the most complex of the tasks of integration of the csv file download system into iNetTest software. The discussion of the implementation will be split into four sections. The first section is a comprehensive description of how the iNetTest database is used to store the information needed for the download scores. The second section discusses the implementation of the interface. The third section discusses the business logic and complexity implemented in each of the questions. Lastly, there is a section that includes a detailed report on the business logic implemented to link the front and back ends of the application.

Database

For this project no new tables were created. All the existing tables were used. The diagram below shows the tables used by the download question software.

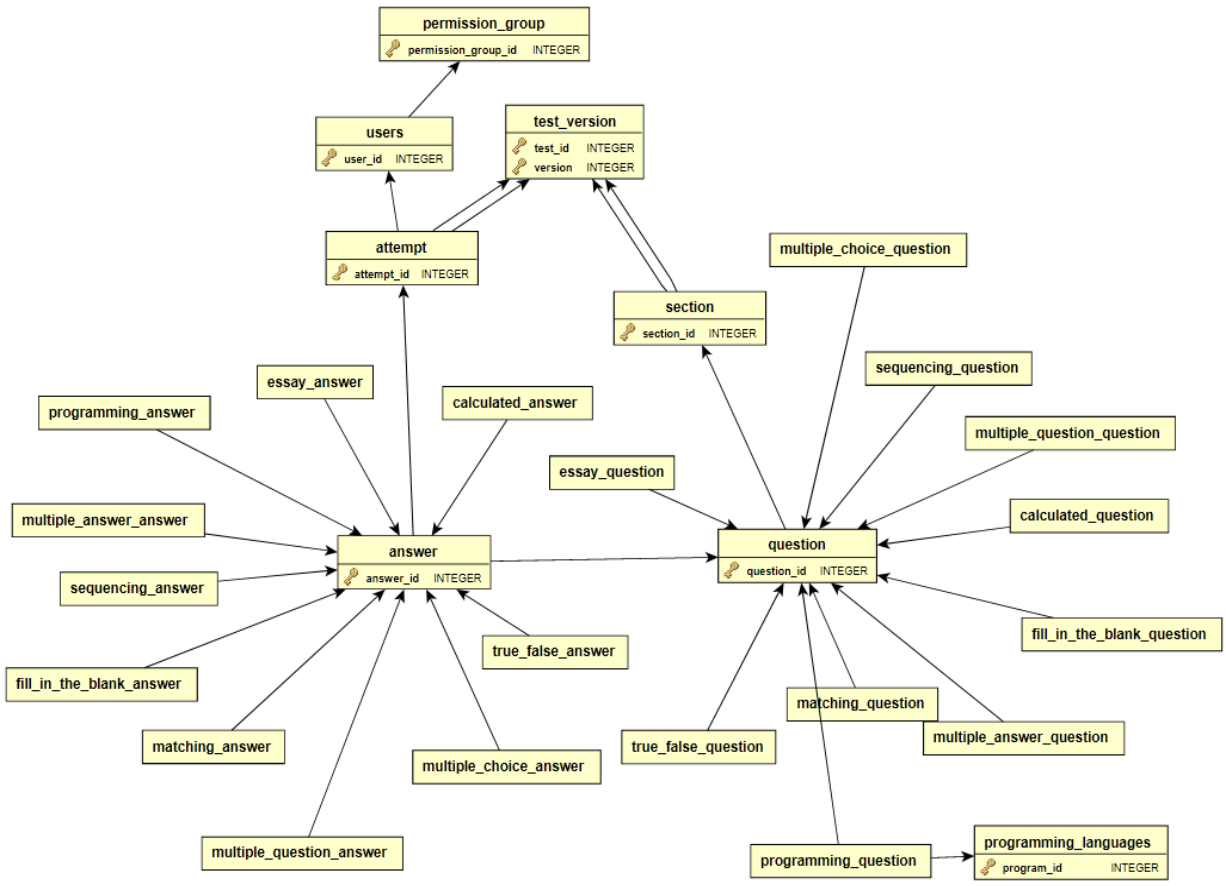

Figure 4: Diagram of the iNetTest Database Tables used by the Download Questions and Section Results software

Interface

The interface needed to provide for the download scores capability is simple and straightforward. On the attempt page a download scores button was added which will pop-up a form to select a group and a respective test for which scores are to be downloaded (see Figure 5).

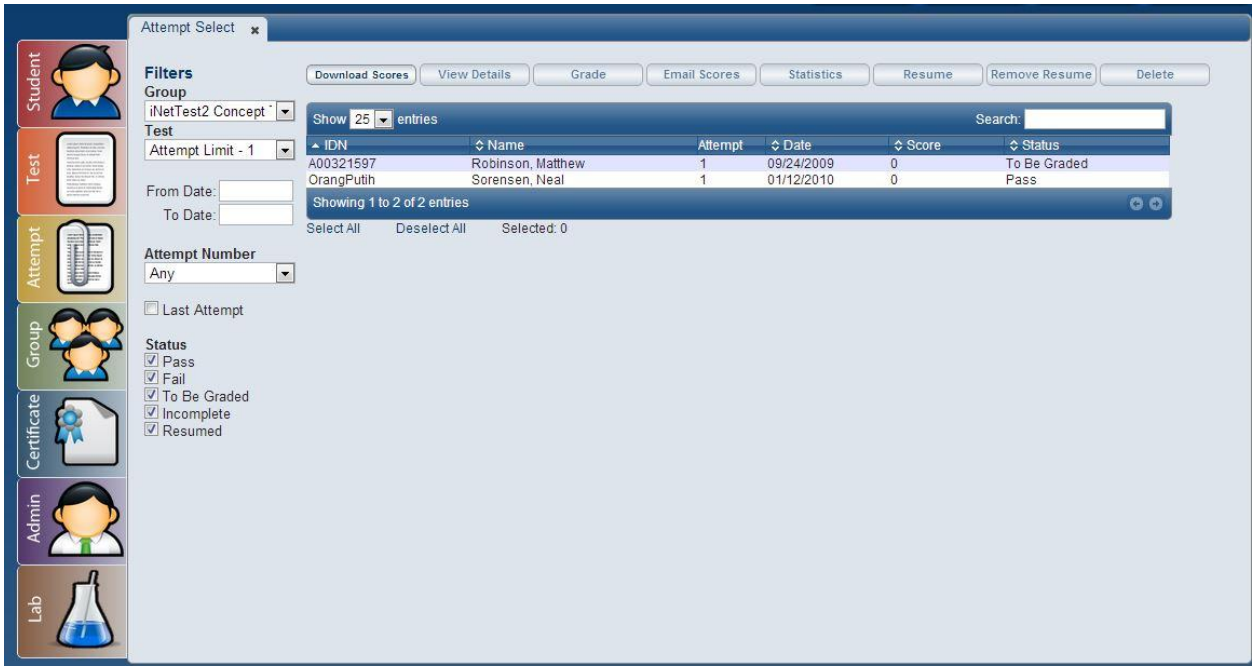

Figure 5: Download Scores button on attempt page

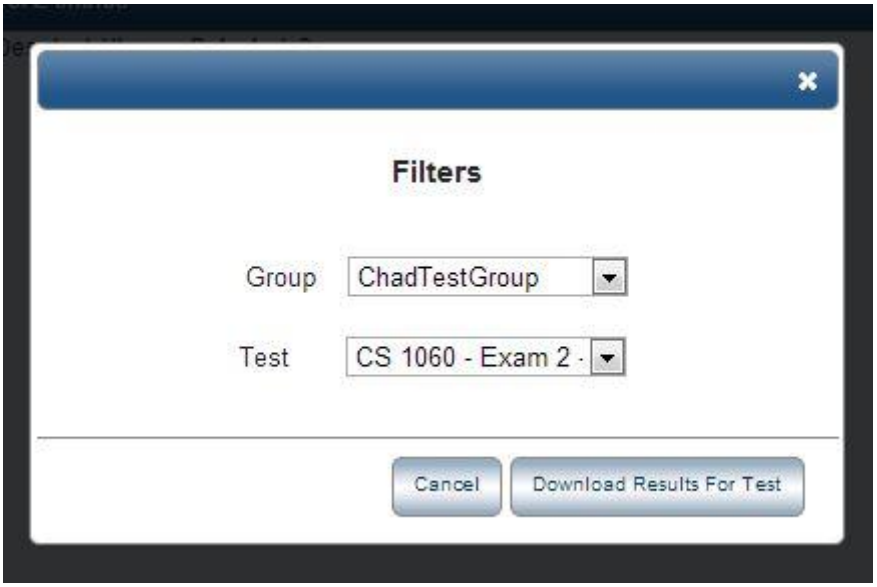

Figure 6: Widget to select group and respective test to download scores

Checking Permissions

When the user clicks on the download scores button, the permission model will check to determine if the user has the permissions to download the scores. If the user does not have such permission a window will pop-up saying no permission to access this page.

List of students in the test

In iNetTest nomenclature, a group is generally a class. Thus, for the selected group and test, the system collects all scores from the student test database. As noted earlier, only the score for the last attempt of each student in the group is selected. Once these scores are selected, the system performs the following operations.

Section & Question Ordering

As discussed earlier the order of sections/questions for a student can be different for each student. The database stores the test sections/questions in the order in which the student has answered the questions on their test. As we generate a detailed test result report all student's sections/questions must be in the same order. In order to accomplish that goal, we translate every student's section/question order to the section/question order of the test as it was created. We determine the actual test sections/questions order from the database.. Based on the sections/questions of the original test, we map each student's sections/questions to actual test's sections/questions.

Answer Ordering

1. Multiple Answer Question

Actual Question	Student's question along with answer
Which cities are in Utah? Salt Lake City New York Los Angeles Logan Philadelphia	Which city is in Utah? ☒ Los Angeles ☒ Salt Lake City ☐ Philadelphia ☐ New York ☐ Logan First column: student's answer Second column: randomized options

Table 4: "Multiple Answer question" Actual question and student's question format

Database Actual Question Format (multiple_answer_question)		Database Student's Answer format (multiple_answer_answer)	
Choicetext1	Salt Lake City	position1	2
Choicetext2	New York	position2	4
Choicetext3	Los Angeles	position3	1
Choicetext4	Chicago	position4	5
Choicetext5	Philadelphia	position5	3
isCorrect1	true	Answer1	true
isCorrect2	false	Answer2	true
isCorrect3	false	Answer3	false
isCorrect4	true	Answer4	false
isCorrect5	false	Answer5	false
First Column: Database Table Column Label Second Column: Database Table Column Value		First Column: Database Table Column Label Second Column: Database Table Column Value	

Table 5: "Multiple Answer question" Actual question and student's answer database format

Each position (in multiple_answer_answer table) will store the new respective position of ChoiceText (in multiple_answer_qestion table). For example, after randomization position1 will store the new position of ChoiceText1 "Salt Lake City" i.e 2.

Each Answer column will store the student's answer with respect to the randomized order. For example, Answer1 will store the student's answer at position 1 i.e. true.

Now to print the student's answer into the csv file we will perform the below steps

1. First we revert back the student's choices (along with respective answers) back to actual question order.
2. Get position1 and answer1. We will put the answer to the position of position1 value. For example, in our example position1 is equal to 2 and answer1 equal to true. Hence we will put the answer1 value at position 2 as shown in the table below.

Position	1	2	3	4	5
Answer		true			

Table 6: "Multiple Answer question" csv answer format

Similarly, we will find the answers for the remaining positions and add the total result to the csv output.

3. Get position1 value and compare it with Answer value. If they are equal student's answer is correct and is 1.

4. If the student's answer is not correct, we have to find his/her answer position. For this we will find what position holds his/her answer. For example if the student's answer is 3, position5 holds the value 3, hence the student answer is 5.
5. After finding the student's answer we will add it to csv output.
1. Now to calculate the scores, we will compare answer 1 in the above table to isCorrect1 from multiple_answer_question. Similarly, we will check for all the answers and find the number of correct answers. Like in our example the number of choices is equal to 5 and number of correct answers is equal to 3 and the number of points on this question is equal to 5, the student will get 5/5*3 ((points on the total question / number of choices)* number of correct answers) points on this question for his/her answer.

For explanation on the remaining answer ordering see Appendix I.

CSV file download option in a Widget

To make the jquery widget gives the option to download and save the csv file, we will create a small jquery plug-in. The jquery plugin will have the below three objects

1. HTML

   The html page contains the form to generate csv file. Before the closing body tag, we are adding the jQuery library which contains the *generateFile* jQuery code.

2. Servlet download page

   What this servlet does is simply add some headers on top of return statement. The plugin we are building must pass two parameters along with the POST request: *filename* and *content*. The servlet will print the content of the file, while setting three headers that will force the file download box to appear (instead of your browser simply opening it).

   The code in the servlet's download page to set the HTML page headers i.e. the response content type and filename :

```
String fileName = vars.request.getParameter("filename");
ServletOutputStream out = vars.response.getOutputStream();
vars.response.setContentType("text/csv");
vars.response.setHeader("Content-Disposition", "attachment; filename=\""+fileName+"\"");
StringBuffer sb = new StringBuffer(vars.request.getParameter("content"));
out.print(sb.toString());
```

3. jQuery

To make the html page give the option to download and save the csv file, we need to set the following parameters in the html DOM object.

- Content Type (“text/csv”)
- Header ("Content-Disposition", "attachment; filename="xyztest.csv")

The jquery widget does not have a DOM object hence we cannot set these parameters. Now we want the jquery page to give the option to download and save the csv file. Now, to overcome this shortcoming of jquery widget, we will be dynamically creating a hidden iframe and write a form to it, which we will later submit via POST. The action attribute of the form points to *download page servlet*, so the file download dialog will pop up, exactly as we need it to.

jQuery Ccode to create HTML iframe object which calls the above created servlet to download the csv file content:

```
1.  $.generateFile = function(options){
2.          options = options || {};
3.          if(!options.script || !options.filename || !options.content){
4.                  throw new Error("Please enter all the required config options!");
5.          }
6.          // Creating a 1 by 1 px invisible iframe:
7.          var iframe = $('<iframe>',{
8.                  width:1,
9.                  height:1,
10.                 frameborder:0,
11.                 css:{
12.                         display:'none'
13.                 }
14.         }).appendTo('body');
15.         var formHTML = '<form action="" method="post">'+
16.                 '<input type="hidden" name="filename" />'+
17.                 '<input type="hidden" name="content" />'+
18.                 '</form>';
19.         // Giving IE a chance to build the DOM in the iframe with a short timeout:
20.         setTimeout(function(){
21.                 // The body element of the iframe document:
22.                 var body = (iframe.prop('contentDocument') !== undefined) ?
23.                                                         iframe.prop('contentDocument').body :
24.                                                         iframe.prop('document').body;   // IE
25.                 body = $(body);
26.                 // Adding the form to the body:
27.                 body.html(formHTML);
```

```
28.                         var form = body.find('form');
29.                         form.attr('action',options.script);
30.                         form.find('input[name=filename]').val(options.filename);
31.                         form.find('input[name=content]').val(options.content);
32.                         // Submitting the form to download.php. This will
33.                         // cause the file download dialog box to appear.
34.                         form.submit();
35.                 },50);
36.         }; [2]
```

Chapter V

Technologies

Struts

Apache Struts is an open-source web application framework for developing Java EE web applications. It is based on model–view–controller (MVC) architecture which enables developers to adopt MVC architecture.

In a typical Java EE web application, the client (browser) calls the server using a web form. The information is then passed to a servlet or a jsp page which in turn returns a HTML response. Both the approaches are found to be inadequate for large size projects as they mix presentation and business logic because of which maintenance becomes difficult.

Struts separates the business logic from the presentation by using the components model (application logic that interacts with a database), view (presentation) and controller (passing the information between view and model). The controller is a servlet known as ActionServlet which is used to design templates for presentation. A central configuration file is created while using struts that binds together model, view and controller.

Whenever a client request is made, the request is sent to the controller based on the definition in the central configuration file. The controller in turn interacts with the model code to generate the user required response. The model also returns an ActionForward string which tells the controller to which page the client needs to be directed. Model and view share information with the help of JavaBeans. Tag library is used to read and write the content of these beans from the presentation layer without the need for any embedded Java code.

EJB (Enterprise Java Beans)

EBJ is designed to free the developer from handling low-level system details like managing transactions, thread, load balancing etc. With the help of EJB developer can focus on development of business logic and EJB can take care of managing data processing**.**

Types of Enterprise JavaBeans

There are mainly two types of Enterprise JavaBeans

1. A session bean encapsulates business logic that can be invoked by a client over a local, remote or web service client views. Stateful session beans maintain conversational state between the client and its bean. Stateless session beans are not persistent i.e. a stateless session bean does not maintain a conversational state with the

client. Whenever a client invokes a stateless bean, the instance variables of the bean may contain a state specific to the client but is maintained only for the duration of the invocation. The implementation of business logic in iNetTest is developed using mainly stateless session beans.

2. An *entity bean* represents a business object persistently stored in a database. Entity beans are associated with database transactions, and may provide data access to multiple users. Entity bean represents persistent data and hence entity bean survive server crashes i.e. if the server is restored back it can reconstruct the bean from the persistent data. Every EJB has a unique identifier. For entity beans, this unique identifier forms the identity of the information. For example, an EmployeeIDNumber might uniquely identify an Employee object. This is analogous to the concept of a *primary key* in a relational database system. Every table in the iNetTest application is represented using Entity bean. [3]

PostgreSQL

PostgreSQL is an object-relational database management system (ORDBMS) available for many platforms including Linux, FreeBSD, Solaris, Microsoft Windows and Mac OS X.

Features

1. Blocks of code can be executed by the database. It can also use other programming languages other than SQL and C.
2. It has a built in support for B+ tree, hash, generalized search trees (GiST) and generalized inverted indexes (GIN).
3. Multiversion concurrency control (MVCC) system is used to manage concurrency.
4. Query tee can rewrite rules for the incoming query. For example, it is used to implement views.
5. Security within the database is managed on the basis per role. A role is a user or a group. It can grant permissions and also revoke them on any object down to the column level and can also prevent users or groups from creating objects at the database, schema or table level. PostgreSQL natively supports a broad number of authentication mechanisms including trust (no enforcement), password (either MD5 or plain-text), GSSAPI, SSPI, Kerberos, ident (maps O/S user name as provided by an ident server to database user name), peer (maps local user name to database user name), LDAP, RADIUS, certificate, PAM.

jQuery

jQuery is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. It is currently developed by a team of developers led

by Dave Methvin. Used by over 55% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today. jQuery is free, open source software, licensed under the MIT License.

jQuery's syntax makes it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This makes it easy for developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets.

jQuery was mainly used for the creation of the widget that pops-up after clicking on the download scores button. jQuery was used to make AJAX calls to the server and handling of the JSON object to create the CSV file.

JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on JavaScript. JSON is a text format which is completely language independent. It uses programming conventions from languages C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON mainly has two components:

- A collection of name/value pairs similar to a HashMap in java.
- An ordered list of values similar to Array/ArrayList in java.

These are universal data structures. Almost all modern programming languages support them in one or the other form.

JSON was mainly used to pass the CSV file data to client (browser) which in turn generates CSV file from the received JSON object.

Chapter VI

Testing

Software testing is defined as the process of determining if the software runs correctly under all conditions or tests, thus ensuring the quality and efficiency of the software implementation. Software testing mainly comprises of two testing types,

1. Automatic Testing

Automatic testing is a testing methodology with the use of special software to control the execution of tests and the comparison of actual results with expected results. Test automation can automate previous repetitive but necessary testing with the use of a testing process and the testing which would be difficult to perform manually. Automated testing was not used to test the application as iNetTest was always tested using manual testing. Also it does not make sense to implement automated testing at the stage when the project is matured. [4]

2. Manual Testing

Manual testing is a testing methodology using manual methods to determine the defects in software. In manual testing, the tester acts as an end user and runs all the possible applications of the software under test.
Manual testing consists of two important functions, one is the test plan and the other is test cases. A manual tester usually executes the following functions while performing manual testing,

- Develop an understanding of the function and implementation of the software to be tested.
- Based on the functionality of the software, prepare a test environment that includes test plans and test cases.
- Run the test cases on the software.
- Record the results of manual testing and compare them with the expected output of the software and generate a report describing any discrepancies.

There are several stages included in manual testing,

1. Unit testing
2. Integration testing
3. Software testing
4. System testing
5. User acceptance testing
6. Release or deployment testing

In Unit testing manual testing is done by the coder who has generated the code. During this stage, white box testing is performed. Integration testing is carried out using black box testing and white box testing.

At times a combination of black and white box testing is used. In the software testing stage, a qualified test engineer performs manual testing that is roughly divided into two testing types,

a. Functional software testing

b. Non-functional software testing

System testing is a stage in which the software is tested for all system related configurations and their compatibility with respect to various platforms. This testing usually prefers black box testing. User acceptance testing is a manual testing stage where the actual end user implements the application and reports on any further enhancements needed in the existing system. Release or deployment testing is performed by the tester onsite/client base. This testing checks for various system requirements like the amount space utilized by the software, time required for the installation of the software, successful run of the setup etc. [4]

Test Cases

No.	Category	Test Case	Expected Results	Actual Results	Status
1	Randomized sections	1. Create test. 2. Allow randomization of sections. 3. Create multiple sections for the test. 4. Let 2-3 students take the test. 5. Generate csv test results.	All the sections for every student are arranged in order of test section order.	All the sections for every student are arranged in order of test section order.	Pass
2.	nonrandomized section	1. Create test. 2. Uncheck randomization of sections. 3. Create multiple sections for the test. 4. Let 2-3 students take the test. 5. Generate csv test results.	All the sections for every student are arranged in order of test section order.	All the sections for every student are arranged in order of test section order.	Pass
3.	Randomized Questions	1. Create test. 2. Allow randomization of questions. 3. Create multiple sections for the test. 4. Create multiple questions for each section. 5. Let 2-3 students take the test. 6. Generate csv test results.	All the questions of each section for every student are arranged in order of test question order for each section.	All the questions of each section for every student are arranged in order of test question order for each section.	Pass
4.	nonrandomized Questions	1. Create test. 2. Uncheck randomization of questions. 3. Create multiple sections for the test. 4. Create multiple questions for each section. 5. Let 2-3 students take the test. 6. Generate csv test results.	All the questions of each section for every student are arranged in order of test question order for each section.	All the questions of each section for every student are arranged in order of test question order for each section.	Pass

5.	Questions count for a section	1. Create test. 2. Uncheck randomization of questions. 3. Create multiple sections for the test. 4. Create multiple questions for each section. 5. Let 2-3 students take the test. 6. Generate csv test results.	The total number of questions for each section are displayed correctly	The total number of questions for each section are displayed correctly	Pass
6.	Selected Question count for a section	1. Create test. 2. Uncheck randomization of questions. 3. Create multiple sections for the test. 4. Create multiple questions for each section. 5. Let 2-3 students take the test. 6. Generate csv test results.	The total number of selected questions for each section are displayed correctly	The total number of selected questions for each section are displayed correctly	Pass
7.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 7 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different randomized order of answers. 6. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
8.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 6 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different randomized order of answers. 6. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
9.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 5 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass

		randomized order of answers. 6. Generate csv test results.			
10.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 4 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different randomized order of answers. 6. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
11.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 3 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different randomized order of answers. 6. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
12.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 2 answers to that question. 3. Let 2-3 students take the test (all answering the question correctly). 4. The answers are randomized and given to the students. 5. All the students will get different randomized order of answers. 6. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
13.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 7 answers to that question. 3. Let a students take the test (answering the question incorrectly). 4. The answers are randomized and given to the students. 5. Generate csv test results.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	All the answers (true/ false) for each answer text are arranged in order of actual answer order for every student.	Pass
14.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 7 answers to that question. 3. Let a students take the test (does not answer the question incorrectly). 4. The answers are randomized and given to the students.	If the correct answer for answer text 1 is true, the student's answer is false. Similarly for all the remaining	If the correct answer for answer text 1 is true, the student's answer is false. Similarly for all the remaining	Pass

		5. Generate csv test results.	answer texts.	answer texts.	
15.	Randomized Answers - Multiple Answer Question	1. Create MA question in a test. 2. Add 7 answers to that question. 3. Let a students take the test (partially answer the question correctly). 4. The answers are randomized and given to the students. 5. Generate csv test results.	The score (stored in the database) on the question for the student's answer is displayed correctly in the csv file.	The score (stored in the database) on the question for the student's answer is displayed correctly in the csv file.	Pass
16.	Interface – list of groups in groups dropdown box	1. Click on download scores button. 2. Check if groups are populated in the groups dropdown box.	All the groups for the user are populated in the groups dropdown box.	All the groups for the user are populated in the groups dropdown box.	Pass
17.	Interface – list of tests in tests dropdown box	1. Click on download scores button. 2. Select a group from the groups dropdown box. 3. The test dropdown box must automatically populate all the tests for the selected group.	The test dropdown box automatically populates all the tests for the selected group	The test dropdown box automatically populates all the tests for the selected group	Pass
18.	Interface – CSV file format	1. Click download scores button. 2. Select group. 3. Select test. 4. Click Download result for test button. 5. The browser will ask for the location of the file to save. 6. Download the file. 7. The downloaded file must be a CSV file with test results.	The downloaded file is a CSV file with test results.	The downloaded file is a CSV file with test results.	Pass

Similarly, the testing for all the question types is performed.

Chapter VII

Deployment

Software deployment is defined as a cluster of interrelated activities that make software available for use. Software deployment can be performed at the developer's end or the client's end. The activities defined within a software deployment are different for each software system. Therefore it's not possible to specifically define the deployment. Instead it can be generally categorized into the following deployment activities, release, Install and activate, deactivate, adapt, update, built-in, version tracking, uninstall and retire. Software deployment is thus a process initialized after the complete development of a software application has occurred.

To deploy the newly developed "Download test results" tool was first deployed onto the iNetTest test server. On successful testing of the application it was deployed on the iNetTest production system. To carry out the deployment process seamlessly a version control (TortoiseSVN) was used.

TortoiseSVN is a version control system utilized to store and manage all possible versions of the files and directories. As a result, TortoiseSVN maintains a history of the older versions of files and their directories. This helps to determine the time, date and person involved in the modification of the files. TortoiseSVN is a free open source version control system independent of the types of files and directories utilized. TortoiseSVN features provide shell integration, Icon overlays, easy access to subversion commands, directory versioning, atomic commits, versioned metadata, choice of network layers, consistent data handling, efficient branching and tagging, hack ability. TortoiseSVN can be installed and ran on Windows 2000 SP2, Windows XP and higher. Thus TortoiseSVN is anefficient version control system which prevents chaos when simultaneous and concurrent changes are made on a single file among a group of developers.[1]

Chapter VIII

Future Work

1. Add answers for Fill in the blank question type.

   Currently theCSV file does not contain answers for fill in the blank questions. This is because a FIB question can have a variable number of blanks and there can be more than one correct answer for an individual blank. One could extend the fill in the blank question type to accommodate the variable number of blanks and the student's answer for these blanks.

2. Previous Attempts

   The generated CSV file contains only the results for the last or most recent attempt of each student. In future version the module could be extended to generate a report for any or all attempts as well. PDF reports

   If the instructor wishes to share the test results with someone or if he/she wishes to maintain a record of test results in a non-editable file format, PDF file format is the best way to do this. Some advantages of PDF reports are:

   a. PDF document is a "read only" document that cannot be altered without leaving an electronic footprint
   b. PDF Files Can Be Used Across Platforms
   c. PDF Files Offer Document Level Security
   d. PDF Files Will Pass The Test Of Time and Change

Chapter IX

Conclusion

The iNetTest system with its various features makes it feasible to integrate a new feature such as downloading test results. The download test results tool can help instructors to know individual question results, e.g. what was the average score for a question, how many got it correct and how many missed it, what was the most common answer chosen (for multiple choice, T/F, etc.), What was the score for a section? What was the average score for a section? Etc.

Using the power of Excel for spreadsheets along with the availability of data, instructors can analyze student test data in multiple ways and thus accommodate their specific needs/requirements. The download test results tool also provides for a seamless integration with the iNetTest system and a useful, easy to use tool for instructors and administrators.

References

[1] http://en.wikipedia.org/wiki/TortoiseSVN

[2] http://tutorialzine.com/2011/05/generating-files-javascript-php/

[3] http://docs.oracle.com/javaee/6/api/javax/ejb/EJB.html

[4] http://wikipedia.org

Appendix I

Answering ordering for each type of question

1. "True or False" Question

Actual Question	Student's question along with answer
When the final value of an expression is assigned to a variable, it will be converted to the data type of the expression	When the final value of an expression is assigned to a variable, it will be converted to the data type of the expression ◌ True ◌ False

Table 7: "True or False question" Actual question and student's question format

Get score and answer value (Boolean) for the answer. Add the score and answer value to the csv file output.

2. "Fill in the Blank" Question

Actual Question	Student's question along with answer
When the final value of an expression is assigned to a variable, it will be converted to the_____ of the expression	When the final value of an expression is assigned to a variable, it will be converted to the __data type___ of the expression

Table 8: "Fill in the Blank question" Actual question and student's question format

Get score on the answer. Add the score to the csv file output.

3. Matching question

Actual Question	Student's question along with answer
Match the following cities with their states: Salt Lake City Utah San Francisco California Las Vegas Nevada Philadelphia Pennsylvania Chicago Illinois First column: questions Second column: matches	Match the following cities with their states 2 Philadelphia 1. Illinois 5 Las Vegas 2. Nevada 1 Chicago 3. Utah 4 San Francisco 4. California 3 Salt Lake City 5. Pennsylvania First column: student's answer Second column: randomized questions Third column: randomized matches

Table 9: "Matching" Actual question and student's question format

Database Actual Question Format (matching_question)		Database Student's Answer format (matching_answer)	
Matchtext1	Salt Lake City	Matchpos1	5
Matchtext2	Blacksburg	Matchpos2	4
Matchtext3	San Francisco	Matchpos3	2
Matchtext4	New York City	Matchpos4	1
Matchtext5	Las Vegas	Matchpos5	3
….		…..	
….		Answerpo1	3
AnswerText1	Utah	Answerpo2	4
AnswerText2	California	Answerpo3	2
AnswerText3	Nevada	Answerpo4	5
AnswerText4	Pennsylvania	Answerpo5	1
AnswerText5	Illinois	…..	
….		Answer1	2
….		Answer2	5
First Column: Database Table Column Label		Answer3	1
Second Column: Database Table Column Value		Answer4	4
		Answer5	3
		……	
		First Column: Database Table Column Label	
		Second Column: Database Table Column Value	

Table 10: "Matching" Actual question and student's answer database format

Each MatchPos will store the new respective position of each match. For example, after randomization MatchPos1 will store the new position of MatchText1 “Salt Lake City” i.e. 5.

Similarly, Each AnswerPos will store the new respective position of each answer. For example, after randomization AnswerPos1 will store the new position of AnswerText1 “Utah” i.e. 3.

Each Answer column will store the student’s answer with respect to the randomized order. For example, AnswerPos1 will store the student’s answer at position 1 i.e. 2.

Now to print the student’s answer into the csv file we will perform the below steps

1. Get AnswerPos1
2. Get Answer (AnswerPos1). For example if AnswerPos1 equal to 5 then get Answer5.
3. Add answer5 value to csv output.
4. Similarly, you will get all the student’s answers and add them to the csv output.
5. Now to calculate the scores we get MatchPos1 and compare it with Answer (AnswerPos1). If they are same, the students, answer is correct otherwise not. Similarly, we will check for all the matches and find the number of correct answers. Like in our example the number of matches is equal to 5 and number of correct answers is equal to 3 and the number of points on this question is equal to 5, the student will get 5/5*3 ((points on the total question / number of matches)* number of correct answers)points on this question for his/her answer.

4. Multiple Choice Question

Actual Question	Student’s question along with answer
Which city is in Utah? Salt Lake City New York Los Angeles Chicago Philadelphia	Which city is in Utah? Los Angeles Salt Lake City Philadelphia New York Chicago First column: Student’s answer Second column: randomized choices

Table 11: “Multiple Choice question” Actual question and Student’s question format

Database Actual Question Format (multiple_choice_question)		Database Student's Answer format (multiple_choice_answer)	
Choicetext1	Salt Lake City	position1	2
Choicetext2	New York	position2	4
Choicetext3	Los Angeles	position3	1
Choicetext4	Chicago	position4	5
Choicetext5	Philadelphia	position5	3
Answer	1	Answer	2
First Column: Database Table Column Label Second Column: Database Table Column Value		First Column: Database Table Column Label Second Column: Database Table Column Value	

Table 12: "Multiple Choice question" Actual question and Student's answer database format

Each position (in multiple_choice_answer table) will store the new respective position of ChoiceText (in multiple_choice_qestion table). For example, after randomization position1 will store the new position of ChoiceText1 "Salt Lake City" i.e. 2.

Answer column will store the student's answer with respect to the randomized order. For example, "Answer" column will store the student's answer choice i.e. 2.

Now to print the student's answer into the csv file we will perform the below steps

1. Get Answer
2. Get position1 value and compare it with Answer value. If they are equal student's answer is correct and is 1.
3. If the student's answer is not correct, we have to find his/her answer position. For this we will find what position holds his/her answer. For example if the student's answer is 3, position5 holds the value 3, hence the student answer is 5.
4. After finding the student's answer we will add it to csv output.
5. Now to calculate the scores, as there is no partial credit on this question, if the student's answer is correct he/she will get full credits else zero.

5. “Multiple Question” Question

Actual Question	Student’s question along with answer
What is object oriented programming? What is RDBMS?	What is RDBMS? What is object oriented programming? Object-oriented programming (OOP) is a programming paradigm that represents concepts as "objects" that have data fields and associated procedures known as methods

Table 13: “Multiple Question question” Actual question and student’s question format

Database Actual Question Format (multiple_question_question)		Database Student’s Answer format (multiple_question_answer)	
Question1	What is RDBMS?	answer1	-
Question2	What is object oriented programming?	answer2	OOP is a programming paradigm that represents concepts as "objects" that have data fields and associated procedures known as methods.
Question3		answer3	-
Question4		answer4	-
		ManualScore	5
First Column: Database Table Column Label Second Column: Database Table Column Value		First Column: Database Table Column Label Second Column: Database Table Column Value	

Table 14: “Multiple Question question” Actual question and student’s answer database format

The multiple_question_question table will store the questions text. For example Question1 will store the first question’s text; Question2 will store second question’s text and so on. There are four possible questions in “multiple questions” question type.

The multiple_question_answer table will store the answer of the selected question. For example, like in our example the student has selected question 2, the student's answer for question 2 will be stored in answer2 column of the answer table.

Now to print the student's answer into the csv file we will perform the below steps

1. Get Answer
2. We will check which answer column (answer1, answer2, answer3, and answer4) has some value in it. For example, in our example answer2 has value in it, other answer fields are blank. So the student's selected question is 2. We will add 2 to the csv file output.
3. As the "multiple questions" question is a manual grading question type i.e. the instructor will have to manually grade this question. After manually grading the question by the instructor, the student's score on this question will be stored in the database table answer (with column name as ManualScrore). We will get the value of ManualScore and add it to the csv file output.

6. "Essay" Question

Actual Question	Student's question along with answer
What is object oriented programming?	What is object oriented programming? Object-oriented programming (OOP) is a programming paradigm that represents concepts as "objects" that have data fields and associated procedures known as methods

Table 15: "Essay question" Actual question and student's question format

As the "Essay" question is a manual grading question type i.e. the instructor will have to manually grade this question. After manually grading the question by the instructor, the student's score on this question will be stored in the database table answer (with column name as ManualScrore). We will get the value of ManualScore and add it to the csv file output.

7. Sequencing Question

Actual Question	Student's question along with answer
Order the below cities by population size – largest population at the top: New York Los Angeles Chicago Houston Philadelphia	Order the below cities by population size – largest population at the top: 2 Los Angeles 4 Houston 5 Philadelphia 3 New York 1 Chicago First column: student's answer Second column: randomized questions

Table 16: "Sequencing question" Actual question and student's question format

Database Actual Question Format (sequencing_question)		Database Student's Answer format (sequencing_answer)	
Answertext1	New York	Answerpos1	4
Answertext2	Los Angeles	Answerpos2	1
Answertext3	Chicago	Answerpos3	5
Answertext4	Houston	Answerpos4	2
Answertext5	Philadelphia	Answerpos5	3
		…..	
First Column: Database Table Column Label		…..	
Second Column: Database Table Column Value		Answer1	2
		Answer2	4
		Answer3	5
		Answer4	3
		Answer5	1
		……	
		……	
		First Column: Database Table Column Label	
		Second Column: Database Table Column Value	

Table 17: "Sequencing question" Actual question and student's answer database format

Each AnswerPos (in sequencing_answer table) will store the new respective position of AnswerText (in sequencing_qestion table). For example, after randomization AnswerPos1 will store the new position of AnswerText1 "New York" i.e. 4.

Each Answer column will store the student's answer with respect to the randomized order. For example, AnswerPos1 will store the student's answer at position 1 i.e. 2.

Now to print the student's answer into the csv file we will perform the below steps

2. Get AnswerPos1
3. Get Answer(AnswerPos1) . For example if AnswerPos1 equal to 4 then get Answer4.
4. Add answer4 value to csv output.
5. Similarly, you will get all the student's answers and add them to the csv output.
6. Now to calculate the scores we get Answer (AnswerPos1) and compare it with 1, similarly, we will get Answer (AnswerPos2) and compare it with 2 and so on. If they are same, the student's answer is correct otherwise not. Similarly, we will check for all the answers and find the number of correct answers. Like in our example the number of answers is equal to 5 and number of correct answers is equal to 3 and the number of points on this question is equal to 5, the student will get 5/5*3 ((points on the total question / number of answers)* number of correct answers)points on this question for his/her answer.

8. "Calculated" Question

Actual Question	Student's question along with answer
Calculate the hypotenuse of a 90 degree triangle with sides length {a} and {b} **Variable Ranges:** a: min 3.0 max 3.0 decimal places 0 b: min 4.0 max 10.0 decimal places 0	Calculate the hypotenuse of a 90 degree triangle with sides length 3 and 4? 5.0

Table 18: "Calculated question" Actual question and student's question format

Get score and answer value (5.0) for the answer. Add the score and answer value to the csv file output.

9. “Programming” Question

   As the “Programming” question is a manual grading question type i.e. the instructor will have to manually grade this question. After manually grading the question by the instructor, the student’s score on this question will be stored in the database table answer (with column name as ManualScrore). We will get the value of ManualScore and add it to the csv file output.