NEO LEARNING SYSTEMS

DESIGN DOCUMENT – PROJECT RAPTOR

OCTOBER 2016

VERSION 1.10

CONFIDENTIAL

1 TABLE OF CONTENTS

2 OVERVIEW

Neo Learning Systems (NLS) is a software provider of classroom testing and analytics for the education market. Our software is directly tied to the Ontario standardized mathematics curriculum. The software is role based and provides user interfaces for students, teachers and administrators of varying levels.

This software is to be used as an in-class pilot project in September 2016 for the purpose of collecting user/institutional feedback, as well as a showcase for future product development prior to a production rollout no later than January 2017.

Web development will be executed by Net Cloud Systems (NC).

2.1 PROJECT OBJECTIVES

The project must meet the following general objectives (specific objectives and project deliverables will be covered in other sections of this document).

- Recreate the features and functions of the NLS demo for the user roles of Student, Teacher, Principal, Board, Ministry, and Admin and as detailed in this document.
- Recreate the visual styling of the NLS demo.
- Create the appropriate level of application security for all user roles.
- At a minimum, deliver functioning applications for both the student and teacher roles for the classroom pilot starting at the end of September 2016 and as described by the Project Order Milestones.
- Deliver functioning applications for all user roles by January 2017.

3 TEACHER ROLE

The following sub-sections correspond to the various screens or tabs showcased in the NLS demo. They are listed in order of development priority.

3.1 TEACHER LOGIN

The teacher, principal, board, ministry and admin roles will share a common login screen; pictured below in Figure 1. A user's associated role will determine which collection of UI screens they are able to access. Whenever a user logs out of the application, they should be brought back to this login screen and any sessional data stored within the browser should be deleted.

Figure 1: Login screen for the teacher, principal, board and ministry user roles

3.1.1 BROWSER SESSION STORAGE

The following values will need to be queried from the database after login and placed in the web browser's session storage for later use on other pages:

User ID	Role
First Name	Last Name
Location ID (for the school)	Location Name
Location ID (for the board)	Board Name
Province	Courses Taught (for the Teacher role)
Grades Taught (for the Teacher role)	

3.1.2 TABLES TO USE

Query the following tables to retrieve the relevant data: User, ClassUser, Class, Location, Board and Ministry.

Additional Notes about this screen:

- Part of milestone #1 due August 5, 2016.
- All logos, graphics and style files can be found within the application repository on GitLab.
- We will use plain text passwords initially to meet the September milestone deadline. However, Proper security, with encrypted passwords, should be implemented as soon as possible.

3.2 LIBRARY TAB

The Library tab, pictured below in Figure 2, is used by teachers to create test questions for later use on student assessments (tests). This is the main content creation screen for the application and will be needed during the month of August in preparation for the classroom pilot.

This tab will be used to both edit existing questions and create new questions. ***The user should only be able to see a filtered list of questions for which they are the creator and are tagged as "Private"***; there are database fields to facilitate this called "Creator" and "Library". Any new questions the user creates should automatically populate the Creator field with their User ID.

Also, by default, all newly created questions should be tagged as "Private" in the database. The initial bank of questions that is to be created in August will be changed to Public sometime in September; this will be done by directly changing the database after all the questions have been created and verified.

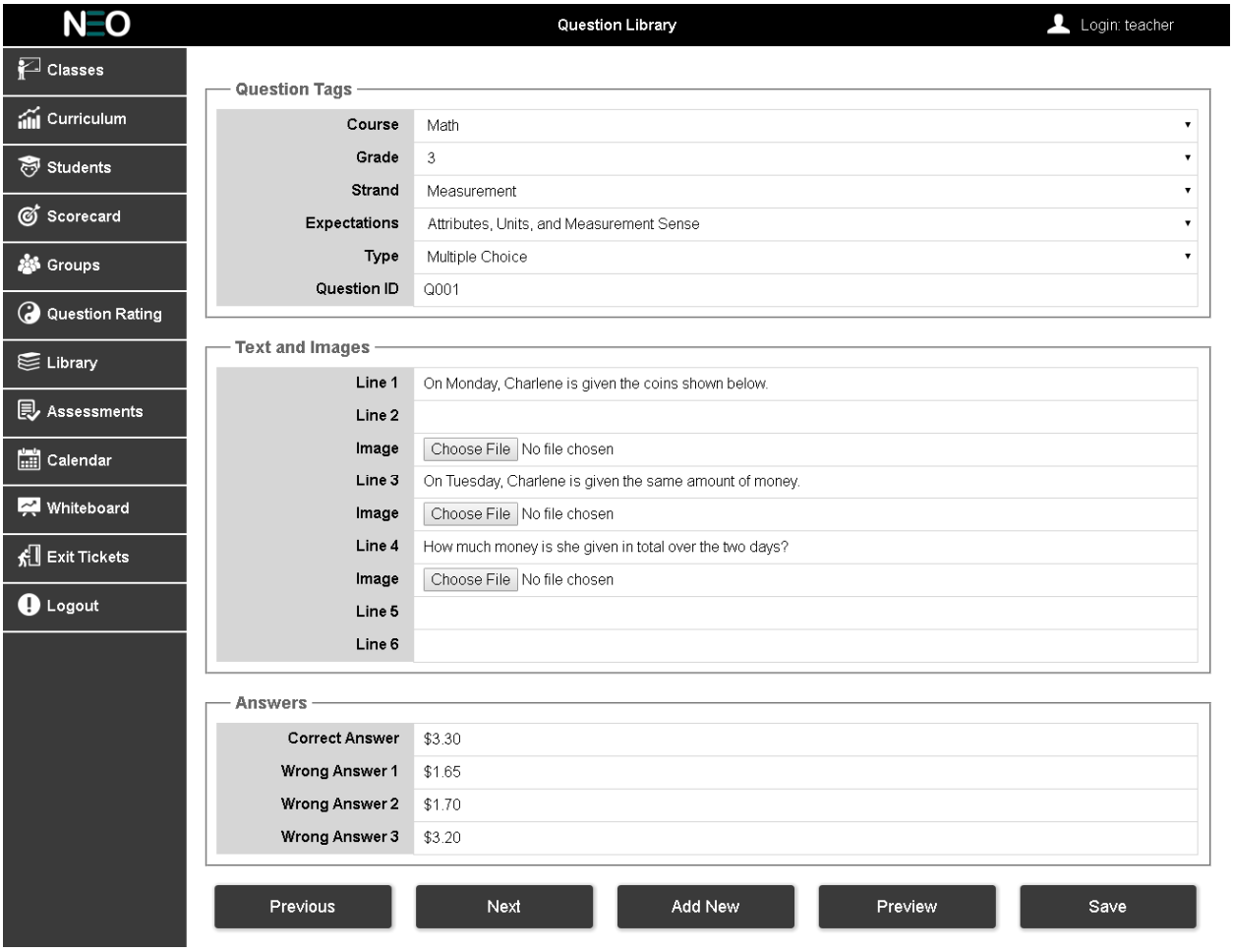


Figure 2: Library Tab

3.2.1 QUESTION TAGS

The Course drop list should query the database for available courses that are taught by the teacher (this info should be stored in web browser session storage upon login).

Like the Course drop list, the Grades drop list should be populated by a database query (this info should be stored in the web browser session storage upon login) to display all grades currently taught by the teacher in descending order.

The Strands drop list must be populated by querying the database for all unique curriculum strands and using the selected Course and Grade as filters.

The Expectations drop list also must be populated by querying the database. This list will use Course, Grade and Strand selections as filters for this query.

A drop list for Specific Expectations will need to be added, and it should be populated by querying the database. This list will use Course, Grade, Strand and General Expectation as filters. ***Please note: this is different than how the demo works.***

The Type drop list can be hard coded to use "Multiple Choice" and "Text Input" and should default to multiple choice.

Another drop list should be added to this grouping called "Manipulatives"; added below the Type selection list. There will be two possible choices: "Do Not Allow" and "Allow". These choices can be hard coded and should default to "Do Not Allow". There will be a field in the database called "Manipulatives" to accommodate this.

The Question ID should be unique and follow a naming convention such as prefixing a 5 digit number with an "M".

All of the drop lists in the Question Tags section should default to the first available choice when this screen is accessed by the user. For example, when the user first opens the Library screen, the course drop list should be populated with the first available course that the user teaches; so the course drop might default to Math. If that user teaches two math classes, then the grade drop list might default to grade 3. And so on down through the drop lists, always defaulting the first available option.

A button should be added between the Question Tags grouping and the Text and Images grouping. This button should be used to run the query for a list of questions fitting the selected filters above. The displayed list of questions should start with the first available question; so if 5 questions fit the query, the first question from the query result is displayed. The user can then parse through the available questions using the button described in section 3.2.4. ***Please note: this is different than how the demo works.***

3.2.2 TEXT AND IMAGES

This part of the Question Library controls the layout of the question being edited/created. The order of the Lines and Images should be exactly mirrored in the display of the final question.

Images should be copied to a directory on the web server that is accessible by all user roles.

The "Choose File" button on each Image line should pop up a window that allows the user to browse to a specific file. Only one image file is allowed for selection and no more than 3 images should be allowed per question.

3.2.3 ANSWERS

The Answers grouping should allow for 1 correct and 3 incorrect answers for questions of type "Multiple Choice".

While not shown in Figure 2, or in the demo, multiple choice answers can contain pictures (for an example, see question #13 of the testing demo or follow this link http://neols.com/eqao_test_8.html) or a combination of inline text with an image.

Combining text answers with associated pictures can be considered a stretch goal for this version of the software given the tight timeline for this milestone.

To accommodate these images, the Answers grouping should setup like the Text and Images grouping where a line of text input is followed by an image selection below it.

For questions of type "Text Input", the Answers section should only give the user the option to input a single correct answer. The input boxes for the incorrect answers should be hidden for this question type and images are not allowed for answers either.

3.2.4 BUTTONS

There are 5 button on this screen:

- Previous (display previous question)
- Next (display next record)
- Add New (create a new question)
- Preview (pop up window showing what the formatted question will look like to the student)
- Save (save changes back to the database).

Advanced seeking and searching functions can be left for the next version of the software. When at the beginning of a displayed list of questions, the Previous button should be inactive (greyed out). When at the end of the list of available questions, the Next button should be inactive.

3.2.5 TABLES TO USE

You must submit complete records when submitting data back to the database. Use the following table for setting up Library content: Question.

Additional Notes about this tab:

- Part of milestone #1 due August 5, 2016.
- All logos and style files can be found within the application repository on GitLab.

3.3 ASSESSMENTS TAB

The Assessments tab, pictured in Figure 3 below, is used by the teacher role to assign tests to classes, groups (a subset of a class) or individual students (within a specific class). Once an assessment of any kind has been assigned to a class, group or student, it should appear on the Calendar tab.

Modifying or deleting an existing Assessment can be done from the Calendar tab; the Assessment tab that is accessed from the left side menu bar should only be used to create new assessments.

A unique Test ID should be assigned for every new assessment that is created.

When assessments are saved back to the database, the "Owner" field must also be updated to reflect the User ID of the logged in user.

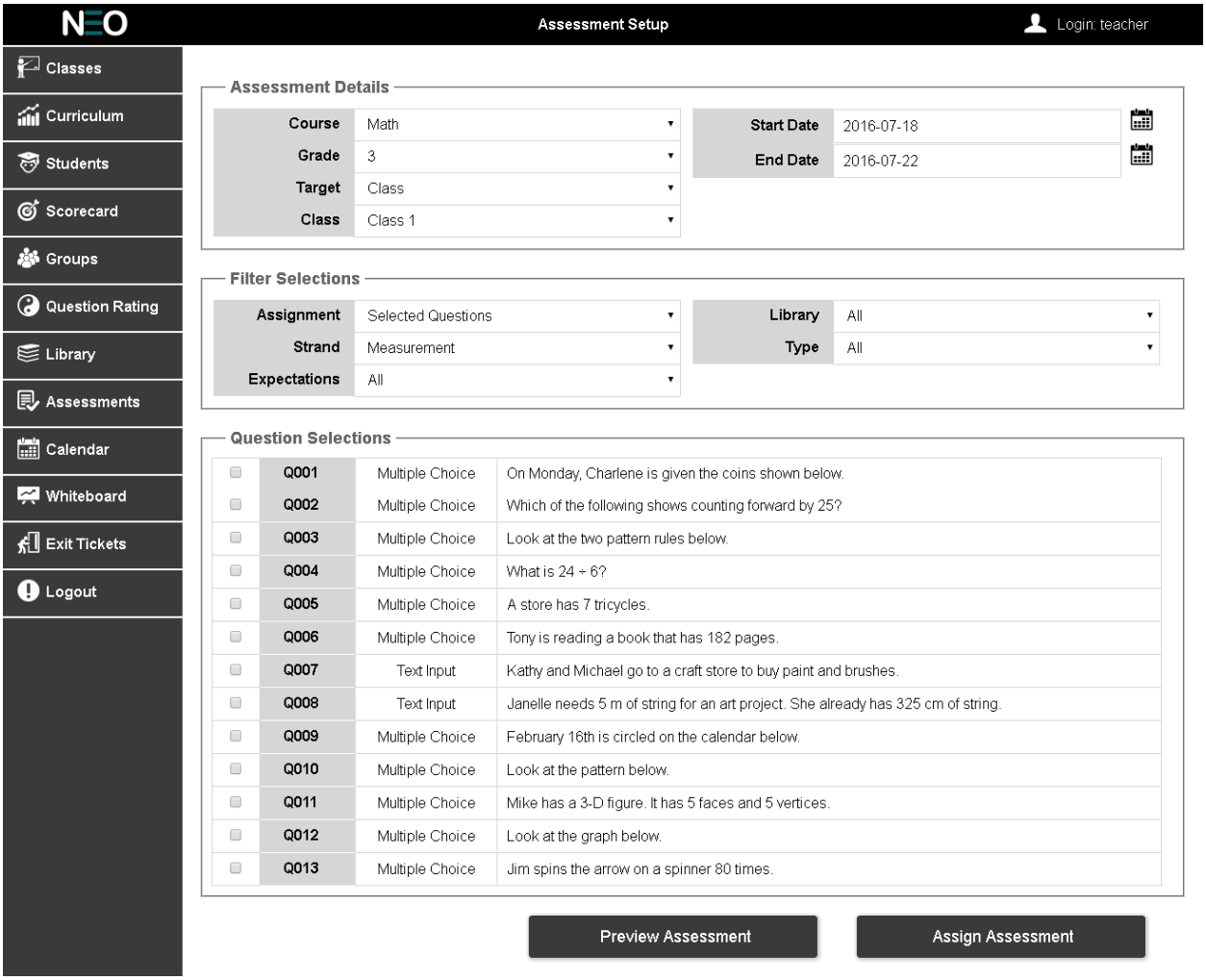


Figure 3: Assessments Tab

3.3.1 ASSESSMENT DETAILS

The Course drop list should be a filtered list of Courses that are being taught by the user currently logged in.

The Grade drop list should also be populated by a query using the selected course above and the currently logged in User ID as filters.

The Target is a hard coded list of choices: “Class”, “Group” or “Student”.

The drop list(s) that appear below Target will become visible or invisible depending on the choice of Target.

When the selected target is “Class”, then the Class drop list is visible below it. The Class drop list should be populated by a query using User ID, Course, Grade, Start Date and End Date (the current date should be between these two dates) as filters. The Group and Student drop lists should be invisible.

When the selected target is “Group”, then the Class drop list should be populated by a query using User ID, Course, Grade and End Date (the current date should be between these two dates) as filters. The Group drop list will be visible below the Class drop list. The Group drop list should be populated by a query using User ID, Class and End Date (the group end date should equal the class end date) as filters. The Student drop list should be invisible. ***Please note: this is different than how the demo currently works.***

When the selected target is “Student”, then the Class drop list should be populated by a query using User ID, Course, Grade, Start Date and End Date (the current date should be between these two dates) as filters. The Student drop list will be visible below the Class drop list. The Student drop list should be populated by a query using Class ID and Role (equal to “Student”) and displaying the Student’s first and last name. The Group drop list should be invisible.

Both the Start Date and End Date should default to the current date.

Below the End Date there should be a selection list for “Type”. This will be a hard coded drop list with the following two choices: “Formal” and “Practice”. A third type will exist called “ExitTicket” but that will be exclusively used on the Exit Ticket tab. The database field “TestType” has been created to accommodate this selection. This selection should default to “Practice”. ***Please note: this is different than how the demo currently works.***

If practice is the selected type, then there should be a visible drop list below the Type called “Retest”. There should only be two possible hard coded selections: “Allow” and “Do Not Allow” (true or false respectively). A “Retest” field has been setup in the database to accommodate this.

If formal is the selected type, then the Retest drop list should be invisible and the value of Retest in the database should be set to “Do Not Allow” or False by default.

There should also be a final drop list in this grouping called “Show Results”. This will be a hard coded drop list with the choices: “Yes” and “No” (true or false respectively). This should default to No or false. The database field “ShowResults” has been setup to accommodate this. ***Please note: this is different than how the demo currently works.***

3.3.2 FILTER SELECTIONS

The Assignment drop list should have two hard coded choices: “Selected Questions” and “Random Questions”; it should default to selected questions.

When the Assignment is set to Selected Questions, the following drop lists should only be visible: Strand, Expectation, Library, Type and Manipulatives.

When the Assignment is set to Random Questions, the following drop list should be visible: Strand, Expectations, Library, Type, Manipulatives and Number of Questions.

The Strand drop list should be populated by a query using the following filters: Course and Grade. There should be an additional selection called “All”, meaning all Strands and this should be the default.

If “All” is selected for Strand, then the Expectations drop list below it should be invisible. If any other option for Strand is selected, then the Expectations drop list should be visible and populated by a query using the following filters: Course, Grade and Strand. There should be an additional selection called “All”, meaning all Expectations; “All” should be the default value.

The Library drop list should be hard coded to use the following three choices: “All”, “Public” and “Private”; where “All” is the default value.

The Type drop list should be hard coded to use the following three choices: “All”, “Multiple Choice” and “Text Input”; where “All is the default value.

The Manipulatives drop list should be hard coded to use the following three choices: “All”, “Allow” and “Do Not Allow”; where “All” is the default value. ***Please note: this is different than how the demo works.***

When the Assignment drop list is set to “Random Questions”, another input called “Number of Questions” should become visible. This will be a spinner input element that defaults to the number 3. To accommodate this, use the “MaxNumber” field in the database. ***Please note: this is different than how the demo works.***

Here are two examples of how the MaxNumber selection should be applied:

Case 1 – The user has selected Random, Specific Strand selected (e.g. Measurement), Specific Expectation selected, max questions = 3 (default). Result is 3 random questions constrained to the

subset of questions that deal with their filter selection. So max questions applies to the single Strand they selected.

Case 2 – The user has selected Random, All Strands, All Expectations, max questions = 3 (default). Result is 3 randomly selected questions for each Strand (all five strands are used) for a total of 15 questions. So max questions applies to every Strand and the result is guaranteed to be an evenly distributed number of questions.

3.3.3 QUESTION SELECTIONS

The questions listed in this section should be displayed in blocks of 10 questions. The list of questions should be the result of a database query using the following filters: Course, Grade, Strand, Expectation, Library and Manipulatives.

As in the demo, the list of questions should display the Question ID, Type (multiple choice or text input), and Line 1 of the question.

Every question displayed should be preceded by a check box for selection purposes. And the application should remember which questions were selected as it moves from block to block, and when the filters are changed as well.

Clicking on the Question ID should pop up a window showing the entire question, see Figure 4 below.

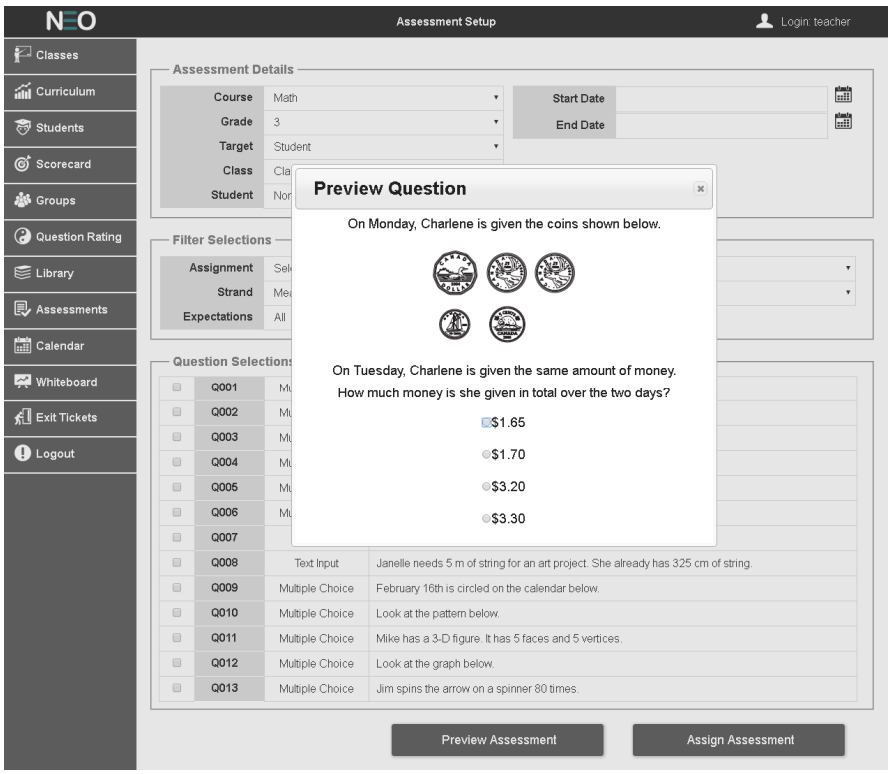


Figure 4: Assessments Tab showing Question preview

3.3.4 BUTTONS

There are two buttons on this screen:

- Preview Assessment (which open a new browser tab and displays a fully formatted view of what the student will see given the users question selections)
- Assign Assessment (which advances the user to the confirmation stage of the assessment setup process)

3.3.5 ASSESSMENT SETUP – ASSIGNMENT CONFIRMATION

This page will allow the user to confirm their selections from the previous step in the assessment setup process, see Figure 5 below.

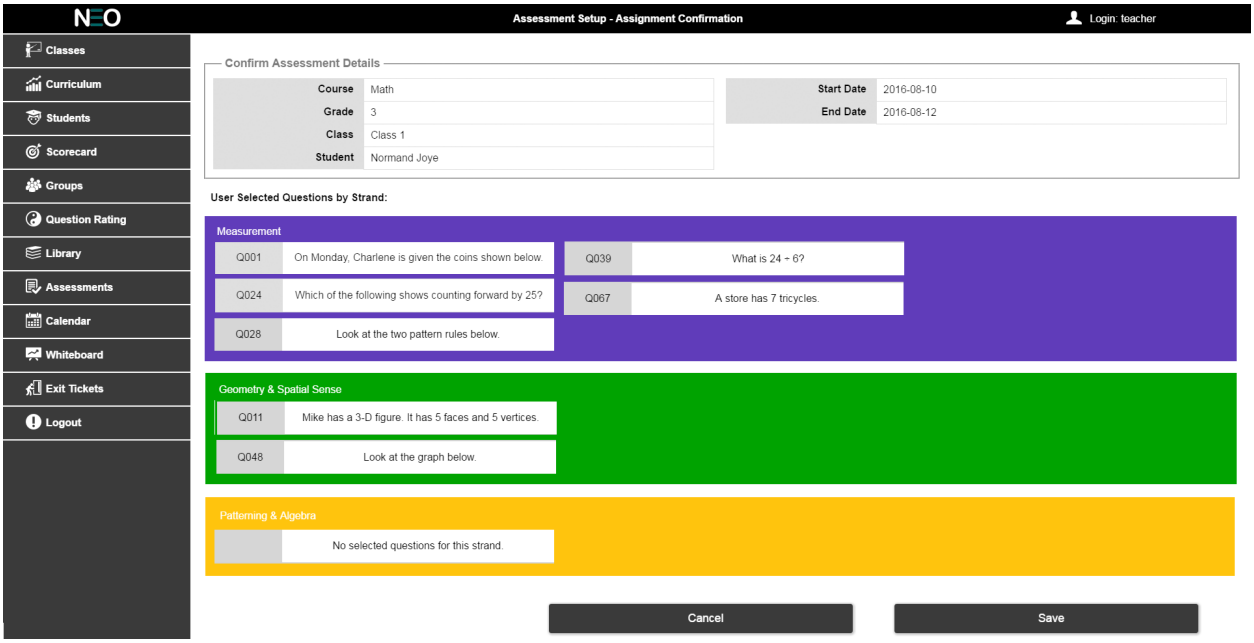


Figure 5: Assessments Tab showing Question preview

The “Confirm Assessment Details” grouping is read-only and lists the following user selections: Course, Grade, Target details (Class, Group or Student and specifics of each), Start Date, End Date and PIN number.

The PIN should be a random, 4 digit number generated by the application. This PIN number will become part of the security process when students try to write their tests (explained in the section: Student Role later in this document). This PIN number should be stored in a database field called “PIN”. ***Please note: this is different than how the demo works.***

The lower portion of the page will list the selected questions with the title depending on which option was selected during setup: “Randomly Selected Questions by Strand” or “User Selected Questions by Strand”.

The list of questions is organized in Strands by colour coded column; for example purple for Measurement, green for Geometry & Spatial Sense, etc. The Strand colour coding should be consistent across the entire application (reference the demo). In the demo the questions are grouped in columns, however that should be changed so that they are grouped in rows (see Figure 5 above). ***Please note: this is different than how the demo works.***

Unlike the demo, the user should be able to deselect questions from this screen, so a check box should be added to each question to allow for this.

Clicking on the Question ID will pop up a window for the user to preview that specific question (just like the previous setup step).

3.3.6 CONFIRMATION BUTTONS

There are two buttons on this screen:

- Cancel (which takes the user back to the previous step in the process)
- Save (which commits the new assessment details to the database and takes the users to the Calendar screen)

3.3.7 TABLES TO USE

Use the following tables for setting up Assessments: Test, TestQuestion, Question, Course and User.

Additional Notes about this tab:

- Part of milestone #2 due August 22, 2016.
- All logos and style files can be found within the application repository on GitLab.

3.4 CALENDAR TAB

The Calendar tab, pictured in Figure 6 below, is used by the teacher role to review, modify or delete their scheduled assessments. Only future assessments can be modified and/or deleted; assessments older than the current date cannot be deleted or changed.

Only the Assessment Details (see Section 3.3.1) can be modified. The questions selected for a given assessment cannot be changed. To change questions associated with a specific assessment requires that the scheduled test be entirely deleted and recreated. PIN numbers also cannot be modified.

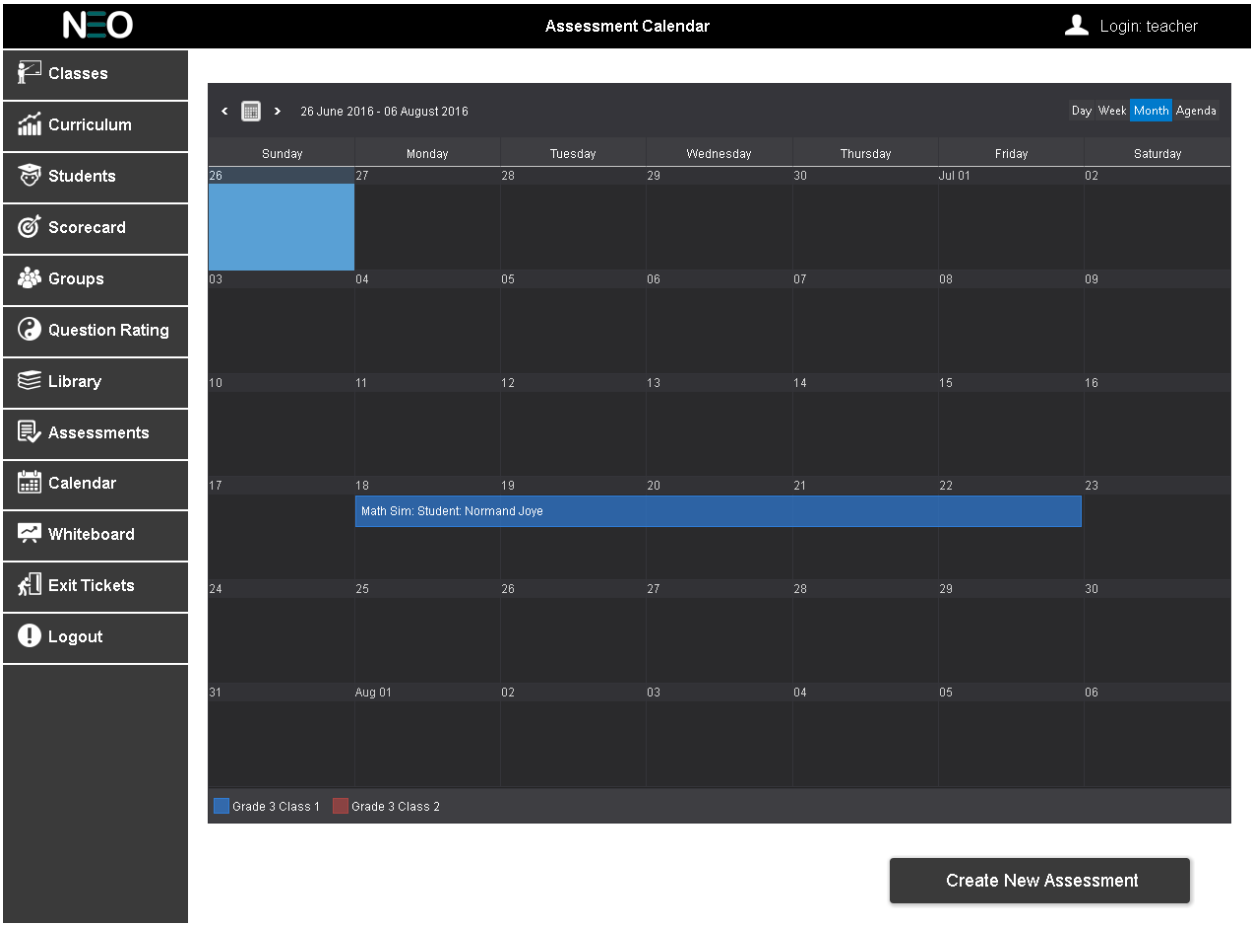


Figure 6: Calendar Tab

The code for this calendar is supplied by jQWidgets (a license has been obtained and the code can be found on the GitLab repo). This particular widget is called the jqxScheduler and documentation for it can be found using this link: http://www.jqwidgets.com/jquery-widgets-documentation/.

When opened from the left menu bar, the calendar should automatically focus on the current date in the month type view.

When opened as part of the Assessment setup process, the focus should be the new assessment start date.

Within the assigned block of time, the user should be able to see the Course and specific target for the assessment.

Double clicking on the block will bring up pop up window with the assessment details, see Figure 7 below. From this point, the user should be able to modify or delete the assessment entirely (applies to future assessments only). ***Please note: this will require more discussion regarding the capabilities of this calendar widget.***

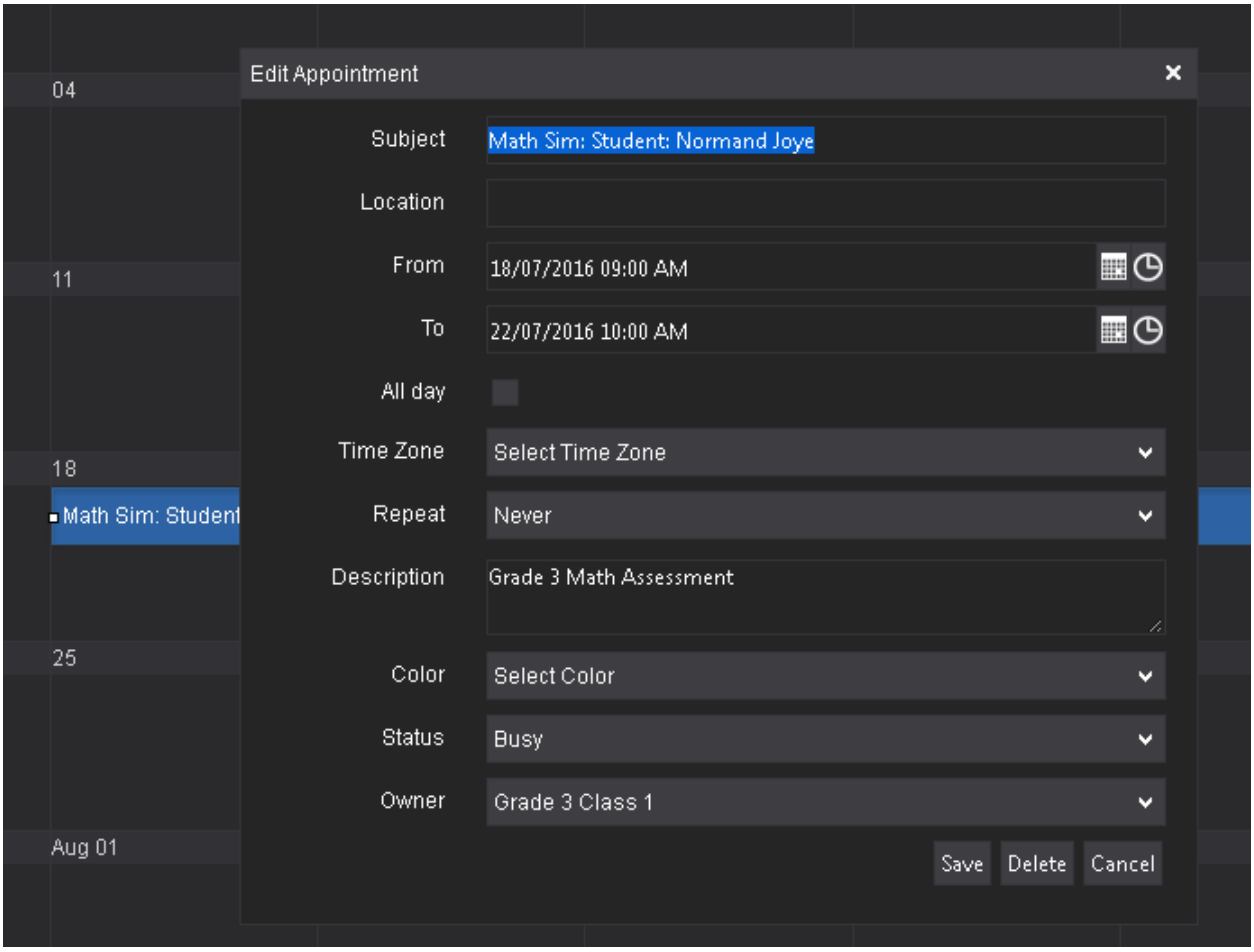


Figure 7: Assessment Details

All the events displayed in the calendar should be the result of a database query filtered by "Owner" (which is the same as the User ID of the current user). Event details will include: Course, Grade, Target, Start Date, End Date, Retest, Show Results, Test Type and PIN.

3.4.1 TABLES TO USE

Use the following tables to retrieve relevant data: Test and User.

4 STUDENT ROLE

The following sub-sections are related to the Student web application that is showcased in the NLS demo.

4.1 STUDENT LOGIN

The student role will have its own login screen; pictured below in Figure 8.

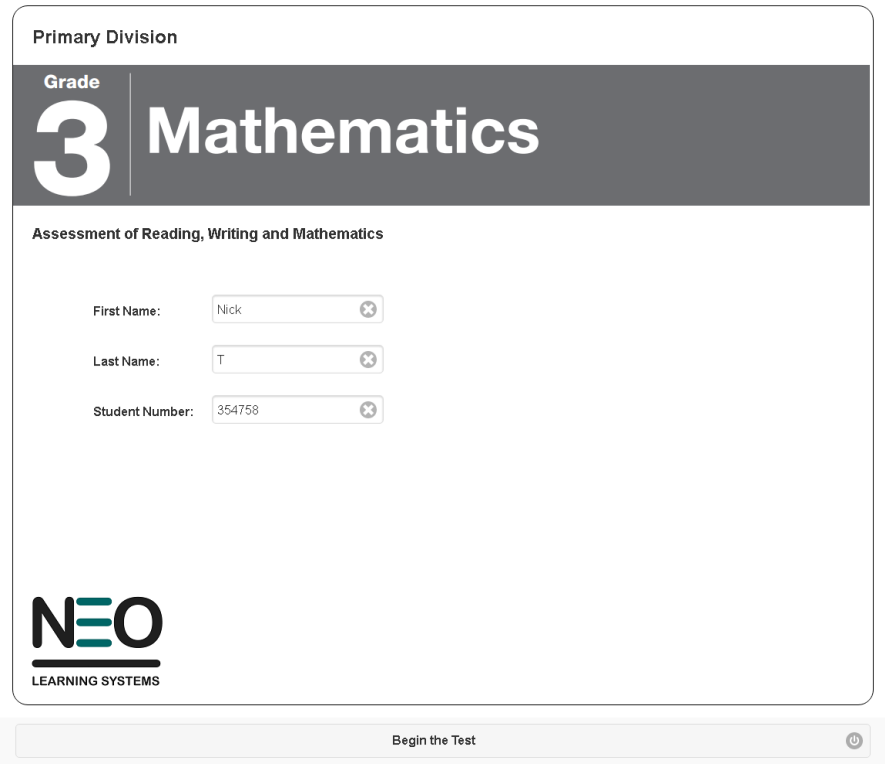


Figure 8: Student Login Page

The functionality needed for the student login page differs significantly from the demo. The general order of operations for this login screen are:

1. Initially, the larger grey bar should have no information. The student can only enter their OEN number (User ID).
2. After they enter their OEN, their first and last name should appear along with two buttons on the bottom of the screen: “Cancel” and “Continue”.

3. If they select Cancel, they are returned to the login screen to re-enter their OEN. If they select Continue, they are shown a lists of tests waiting for them. The test list should simply appear on this screen.

4. When a test is selected, the list disappears.

5. The large grey bar is then populated with the details of the selected test (Grade and Course).

6. The buttons at the bottom of the screen are now “Cancel” and “Begin Test”.

7. If Cancel is selected, the student is returned to the list of waiting tests. If Begin Test is selected, they are prompted to enter the PIN number for that test.

8. Entering a correct PIN number advances the student to the first page of the selected test. Students are allowed to enter an incorrect PIN number a maximum of 3 times; after the third failure, they are returned to the list of tests.

4.1.1 BROWSER SESSION STORAGE

The following values will need to be queried from the database after login and placed in the web browser’s session storage for later use on other pages:

User ID	Role
First Name	Last Name
Location ID	School Name
Board ID	Test ID (for selected test)
Test Type (for selected test)	Course (for selected test)
Grade (for selected test)	Retest (for selected test)
Show Results (for selected test)	Class ID (for selected test)

4.2 ASSESSMENT

After successfully entering the correct test PIN, the student will see a screen similar to that shown in Figure 9 below. The layout, page headings, colouring, images and font used for the test are crucial as they are meant to exactly match that of the Ontario standardized test. All the images used in the test can be found on the project repo.

When coding for this, please use the following guidelines:

- Group non-manipulative questions at the beginning of the test.
- Only display a maximum of two multiple choice questions per page.
- Only display a maximum of one text input question per page.

- Assessments with 20 questions or more should be split into two sections. Where each section will contain half of the total number of questions. For example, a test with 25 questions would have 13 questions in Section 1 and the remaining 12 questions in Section 2.
- Students should be able to make use of “Previous” and “Next” buttons at the bottom of every test page to go back and forth through the test.
- Once they submit their test, they cannot go back to previous test pages.

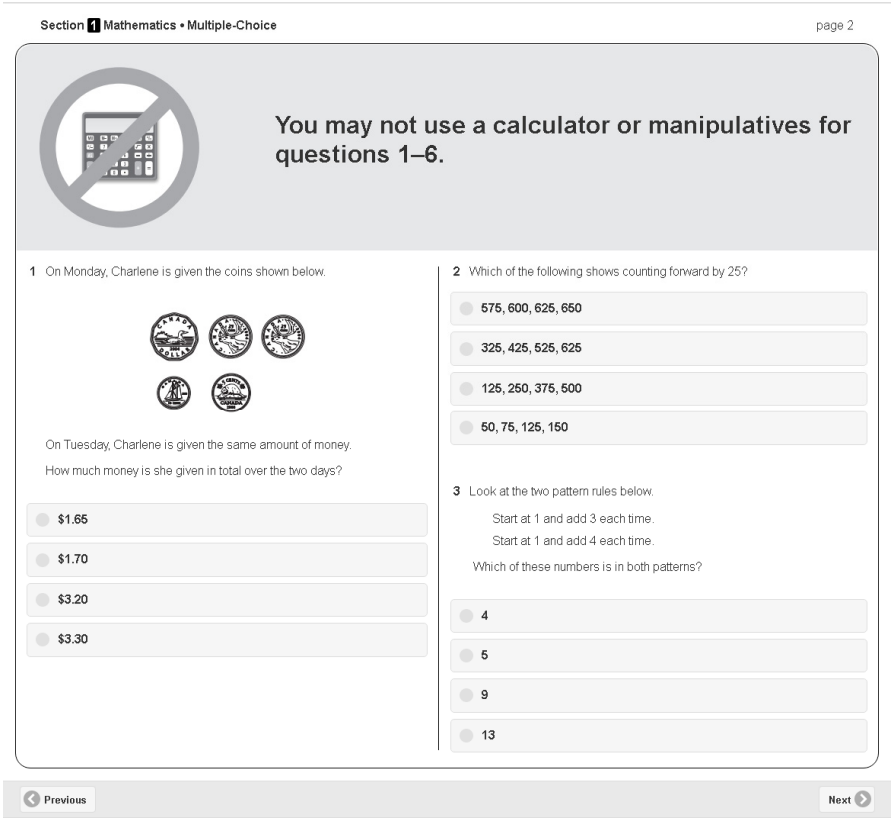


Figure 9: Multiple Choice Questions

4.2.1 OPEN-RESPONSE VS TEXT INPUT QUESTION TYPES

The Ontario School Boards only make reference to Open-Response questions on their standardized tests. However, for our development purposes, we have identified two questions types within this category; internally, we will refer to these two types as “Text Input” and “Open-Response” types.

In general, Text Input type questions have a final answer that student responses can be measured against and that they must type in. There is also an area where they can write in to show their work. See the Figure 10 below.

Open-Response type questions have no final answer, rather they simply have an open area where the student can draw their response. These hand drawn responses are then manually scored by the teacher. See the Open-Response section below for more details.

In either case, displayed in the page header for the student assessment will be “Open-Response” for both of these question types.

4.2.2 TEXT INPUT QUESTIONS

The text input questions on the test must be formatted as pictured below in Figure 10. There are two major components for these types of questions: the drawing (canvas) portion of the page and the text input box at the bottom of the page for the student’s final answer.

The drawing or canvas element is open source code called “Literally Canvas”. The code for this can be found on the repo. ***This differs from what was used in the demo.*** More information about Literally Canvas can be found on their webpage: http://literallycanvas.com.

Please note that Literally Canvas has a dependency on React.js.

It would also be helpful if the canvas UI could be simplified to show fewer buttons (limit the functionality to just what is needed). Further discussion on this will be required.

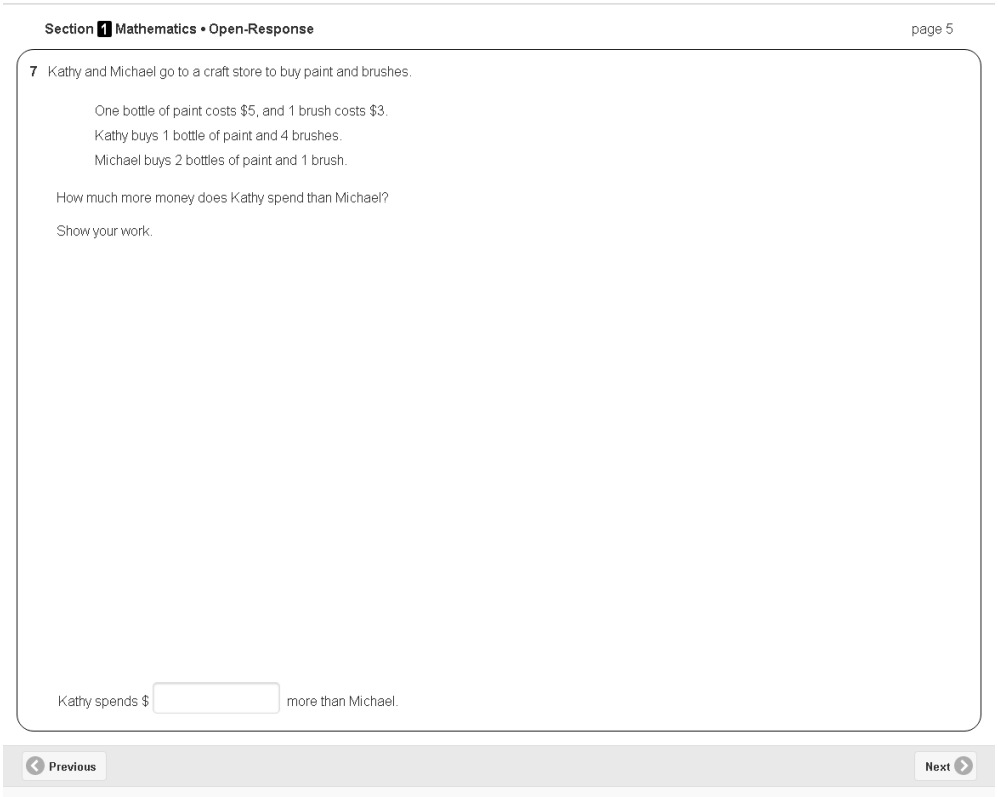


Figure 10: Text Input Question

4.2.3 OPEN-RESPONSE QUESTIONS

Open-Response type questions are similar to Text Input questions in that they feature a large open area for the student to hand draw their answers. Unlike Text Input questions, there is **NO** text box for the student to enter a final answer.

In most cases, the canvas area where they are drawing will have a background. As an example, see Figure 11 below which has a lined chart as the background image for the students to draw their answer on top of.

Open-Response type questions will have an impact on how assessments are marked, when they are marked, how results are displayed, and how assessments are set up; because of this, development around them should be left until after the October 1 pilot rollout, assuming that there is insufficient time before that to include them in the software.

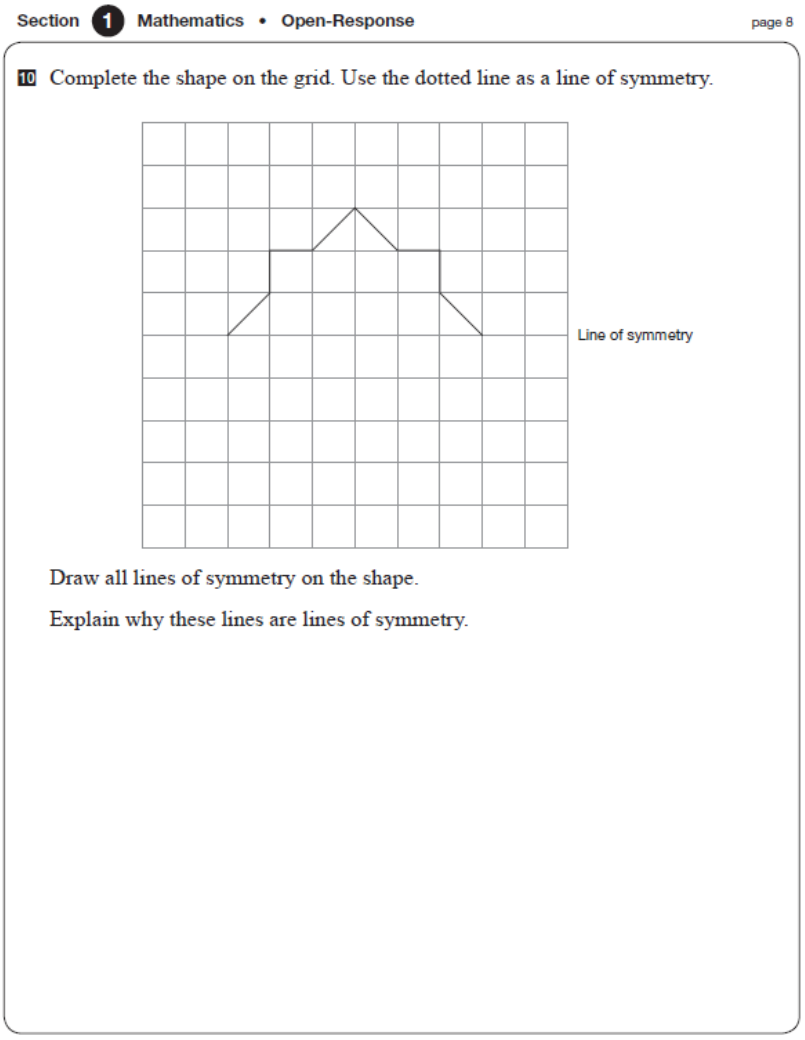


Figure 11: Open-Response Question Type

4.2.4 TEST SUBMISSION

When the student reaches the final page of the test, they should see a visible graphic that indicates that there are no more questions. Also, the Next button at the bottom of the page should be replaced with a button that reads “Submit Your Test”. See Figure 12 below.

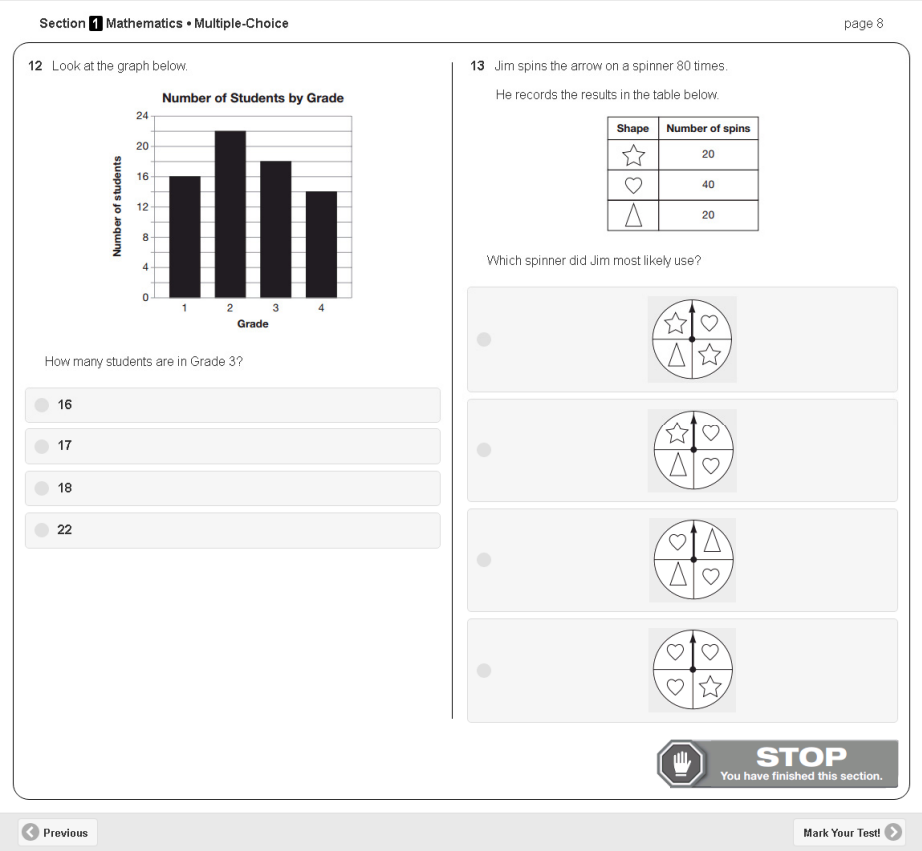


Figure 12: Last Page of Test

If the student chooses to submit their test, they cannot go back to the previous pages anymore. The application should also verify that the student has answered all questions before submitting the test for marking. If the student has skipped questions, they should be offered the chance to go back before the test is graded.

If the assessment has been flagged NOT to show results, then they are simply returned to the login page. If the assessment has been flagged to show their test result, then they will be shown a screen that breaks down their test results by question (showing right and wrong answers) and the level they have achieved. See Figure 13 below.

Additionally, if the test has been flagged to allow for a retest then the student is show a button at the bottom of the page that will allow them to restart the test from the beginning again. Only one retest is allowed. The last completed test score is the only one that is submitted to the database as a final grade.

The above functionality for showing results and retesting is different than in the demo.

Figure 13: Test Results Page

The achievement level is calculated as follows:

- Level 1 – Scores less than 50%
- Level 2 – Scores greater than and equal to 50% and less than 65%
- Level 3 – Scores greater than and equal to 65% and less than 75%
- Level 4 – Scores greater than and equal to 75%

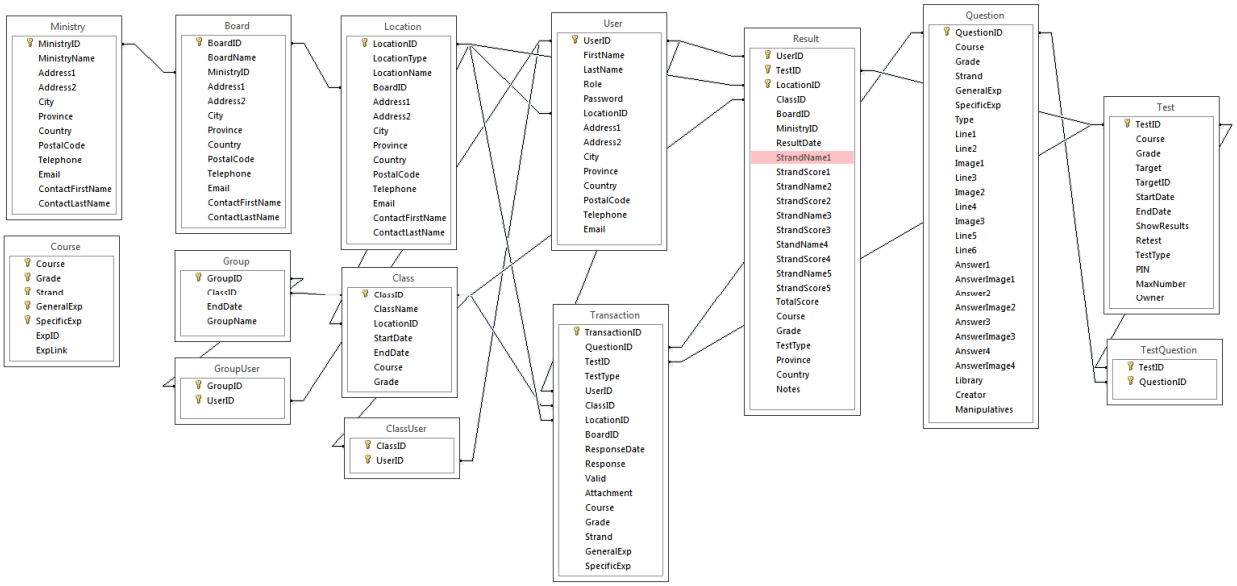
4.2.5 TABLES TO USE

Tables to use for querying data for the Student application include: User, Location, Board, Course, Class, ClassUser, Group, GroupUser, Test, TestQuestion and Question.

When submitting test results back to the database, append to the following tables: Transaction and Result.

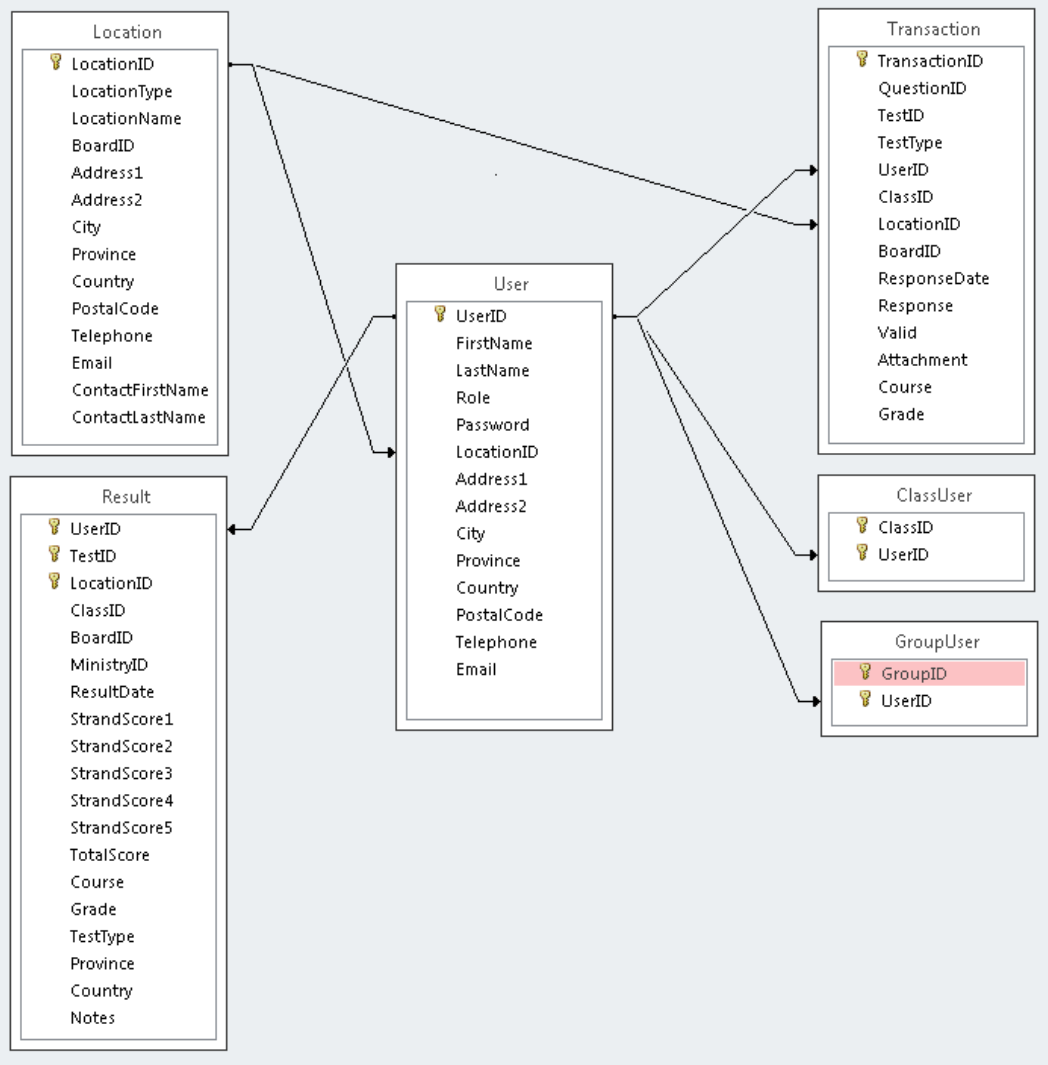
5 DATABASE

This section should serve as a quick reference to the tables in the database. The overall database, and the relationships between tables is pictured below.



5.1 TABLE: USER

	Field Name	
🔑	UserID	OEN Number
	FirstName	
	LastName	
	Role	Valid values: Student, Teacher, Principal, Board, Ministry, Admin
	Password	
	LocationID	Prefix "LOC" followed by unique 5 digit number
	Address1	
	Address2	
	City	
	Province	
	Country	
	PostalCode	
	Telephone	
	Email	

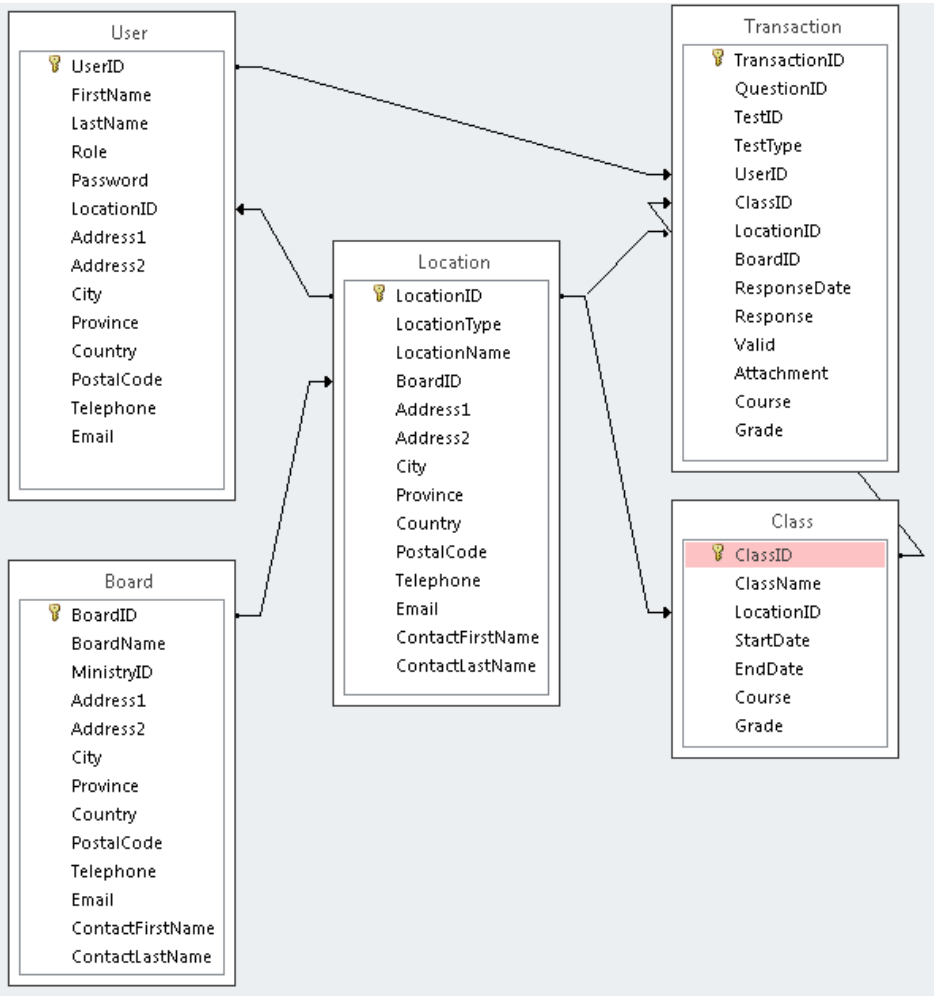


Sample Data:

123456789,Tom,Smith,Student,password,LOC12345,address line 1, address line 2,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,tom@email.com
111222333,Alex,Jackson,Teacher,password,LOC12345,address line 3,address line 4,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,alex@email.com
444555666,Drew,Stevens,Principal,password,LOC12345,address line 5,address line 6,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,drew@email.com
777888999,Joe,Blow,Board,password,LOC11111,address line 7,address line 8,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,joe@email.com
000999888,Jack,Black,LOC54321,Ministry,password,address line 9,address line 10,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,jack@email.com
987654321,John,Strong,Student,password,LOC12345,address line 11,address line 12,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,john@email.com

5.2 TABLE: LOCATION

Field Name	
LocationID	Prefix "LOC" followed by unique 5 digit number
LocationType	Valid values: School (other types will be added in future)
LocationName	
BoardID	
Address1	
Address2	
City	
Province	
Country	
PostalCode	
Telephone	
Email	
ContactFirstName	
ContactLastName	

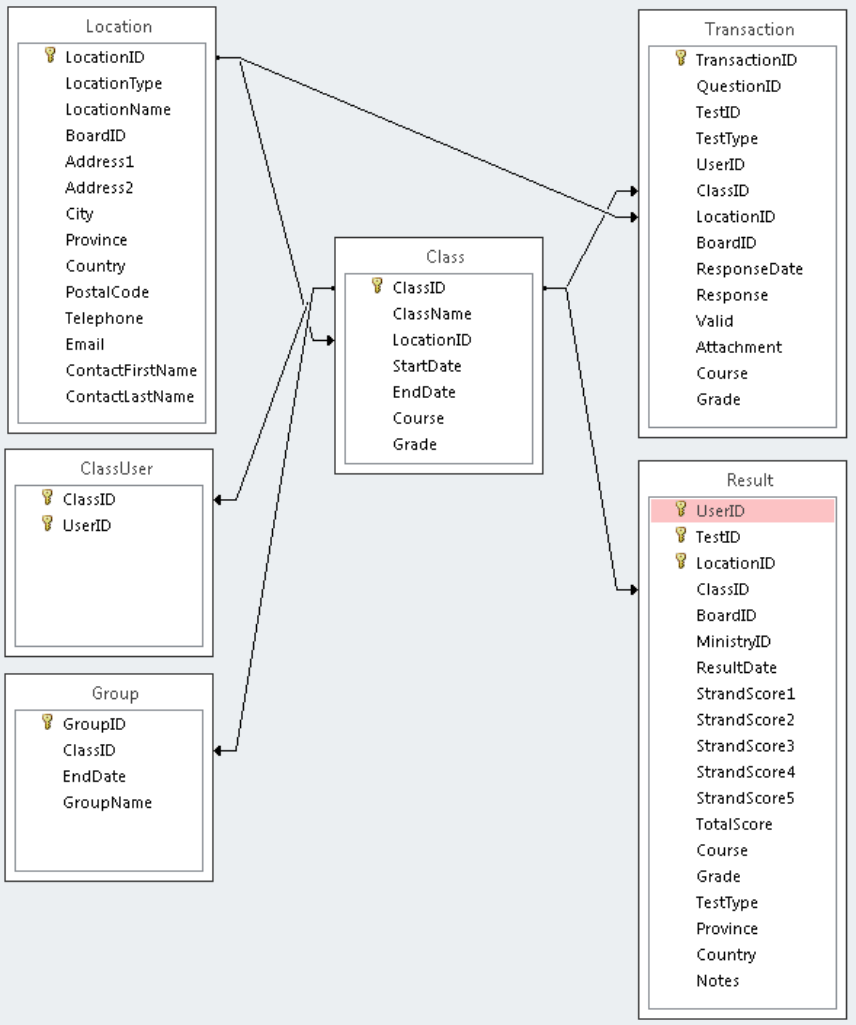


Sample Data:

LOC12345,School,School Name,BRD12345,Address Line 1,Address Line 2,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,drew@email.com,Drew,Stevens
LOC11111,Board,BRD12345,Address Line 1,Address Line 2,Toronto,Ontario,Canada,M5V 3J6,416-555-5555,joe@email.com,Joe,Blow

5.3 TABLE: CLASS

Field Name	
ClassID	Prefix "CL" followed by unique 5 digit number
ClassName	
LocationID	Prefix "LOC" followed by unique 5 digit number
StartDate	Generally courses will start Sept 1, Jan 1 or July 1
EndDate	Generally courses will end Dec 31, May 30 or Aug 31
Course	Valid values: Math; other course will follow in time
Grade	Valid values: K to 12



Sample Data:

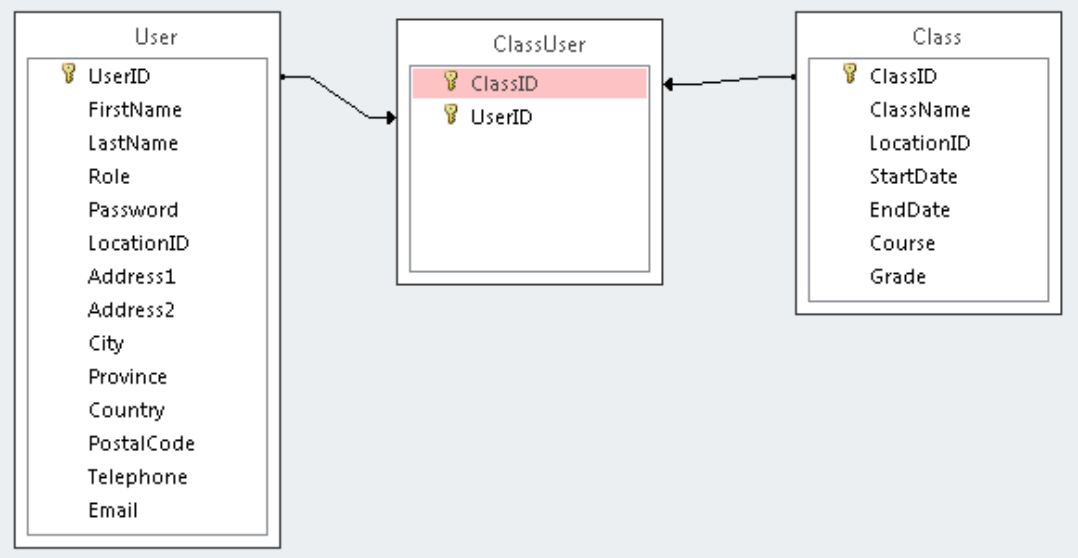
CL12345,Math Period 1,LOC12345,01/09/2016,31/12/2016,Math,3

CL12345, Math Period 1,LOC12345,01/09/2016,31/12/2016,Math,3

CL12345, Math Special Needs,LOC12345,01/09/2016,31/12/2016,Math,6

5.4 TABLE: CLASSUSER

Field Name	
ClassID	Prefix "CL" followed by unique 5 digit number
UserID	OEN Number



Sample Data:

CL12345, 111222333

5.5 TABLE: COURSE

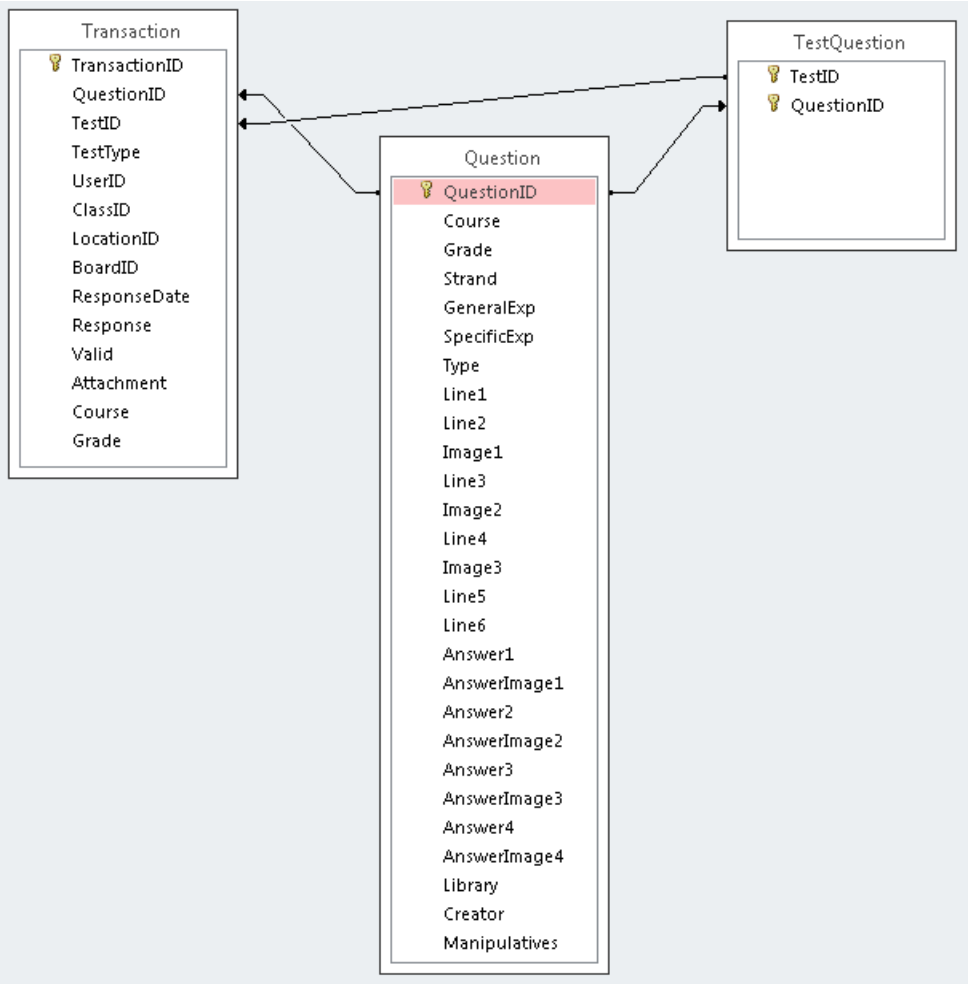
Field Name	
Course	Valid values: Math; other course will follow in time
Grade	Valid values: K to 12
Strand	
GeneralExp	
SpecificExp	
ExpID	Prefix "EXP" followed by a unique 5 digit number
ExpLink	URL to destination in curriculum PDF file

Sample Data:

"Math","3","Measurement","Measurement Relationships","compare standard units of length and select and justify the most appropriate standard unit to measure length",EXP00001, http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Measurement Relationships
"Math","3","Measurement","Measurement Relationships","describe the relationship between the size of a unit of area and the number of units needed to cover a surface",EXP00002, http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Measurement Relationships
"Math","3","Measurement","Attributes, Units & Measurement Sense","estimate, measure, and record length, height and distance",EXP00003, http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Attributes, Units, and Measurement Sense
"Math","3","Measurement","Attributes, Units & Measurement Sense","draw items using a ruler, given specific lengths in centimetres",EXP00004, http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Attributes, Units, and Measurement Sense
"Math","3","Geometry & Spatial Sense","Geometric Properties","identify right angles and describe angles as greater than, equal to, or less than a right angle",EXP00005, http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Geometric Properties
"Math","3","Geometry & Spatial Sense","Geometric Properties","compare various angles and describe angles as bigger than, smaller than, or about the same as other angles",EXP00006,
http://www.neols.com/assets/Curriculum.pdf#nameddest=Grade 1 - Geometric Properties

5.6 TABLE: QUESTION

Field Name	
QuestionID	For Math, prefix with "QM" then unique 4 digit number
Course	Valid values: Math; other course will follow in time
Grade	Valid values: K to 12
Strand	
GeneralExp	General Expectation
SpecificExp	Specific Expectation
Type	Valid values: Multiple Choice, Text Input
Line1	
Line2	
Image1	Path to image file
Line3	
Image2	Path to image file
Line4	
Image3	Path to image file
Line5	
Line6	
Answer1	
AnswerImage1	Path to image file
Answer2	
AnswerImage2	Path to image file
Answer3	
AnswerImage3	Path to image file
Answer4	
AnswerImage4	Path to image file
Library	Public or Private Library
Creator	UserID of the user who created the question
Manipulatives	Allow or Disallow the use of calculators and manipulatives

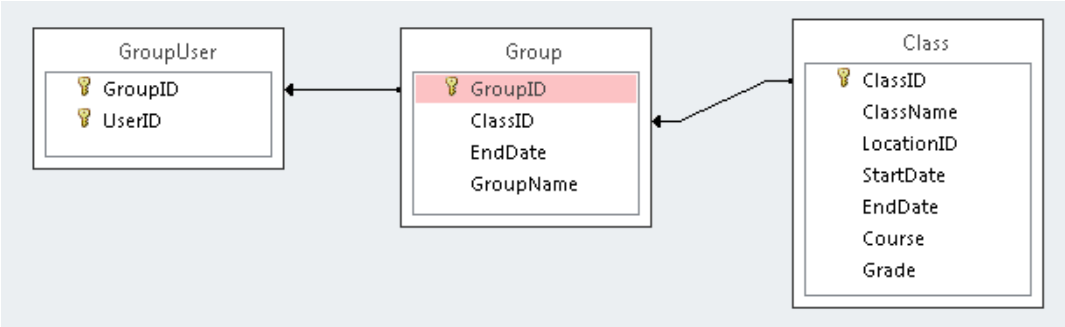


Sample Data:

QM12345,Math,3,"Measurement","Measurement Relationships","compare standard units of length and select and justify the most appropriate standard unit to measure length",Multiple Choice,"On Monday, Charlene is given the coins shown below.",,"C:\Pictures\image1.png","On Tuesday, Charlene is given the same amount of money.",,"How much money is she given in total over the two days?",,,,"$3.30",,"$1.65",,"$1.70",,"$3.20",,Private,111222333,"Do Not Allow"

5.7 TABLE: GROUP

Field Name	Data Type	
GroupID	AutoNumber	Prefix "GRP" followed by unique 6 digit number
ClassID	Number	Prefix "CL" followed by unique 5 digit number
EndDate	Date/Time	
GroupName	Short Text	



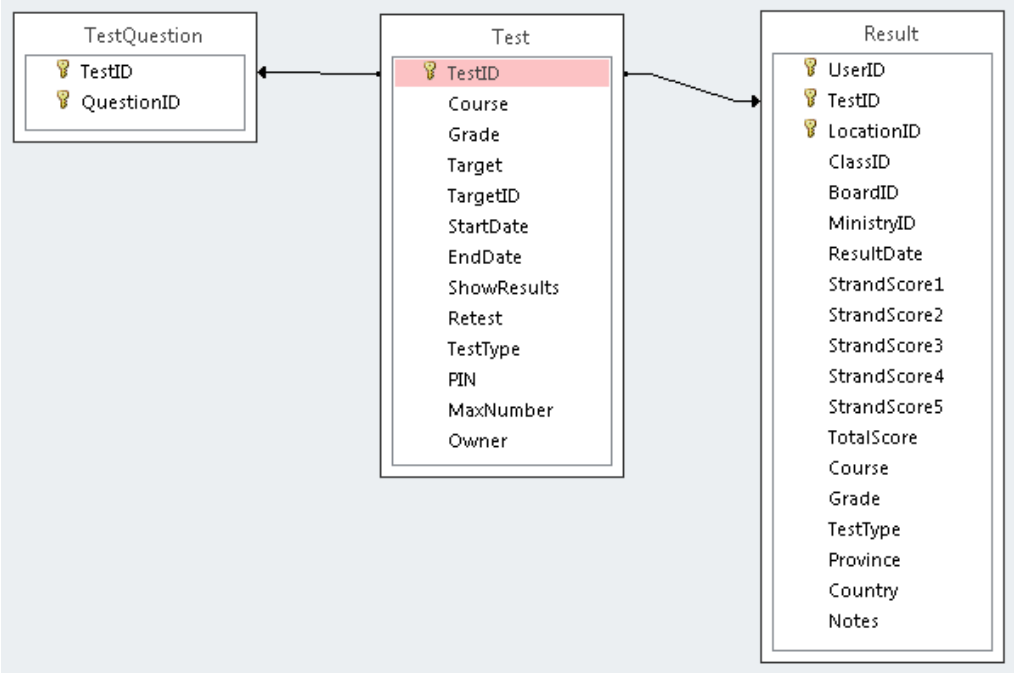
Sample Data:

GRP123456,CL12345,2016/12/31,Fast Learners

GRP123499,CL12345,2016/12/31,Measures Study Group

5.8 TABLE: TEST

Field Name	Desc
TestID	Prefix with "TF" for Formal test type. Prefix with "TP" for Practice test type. Followed by unique 6 digit number.
Course	Valid values: Math; other course will follow in time
Grade	Valid values: K to 12
Target	Valid values: Class, Group or Student
TargetID	Equal to UserID, GroupID or ClassID
StartDate	
EndDate	
ShowResults	Valid values: Yes or No
Retest	Valid values: Yes or No
TestType	can be "formal", "practice" or "exitticket"
PIN	User for security; random 4 digit number
MaxNumber	user selected number of random test questions to assign
Owner	UserID of the teacher who created the test

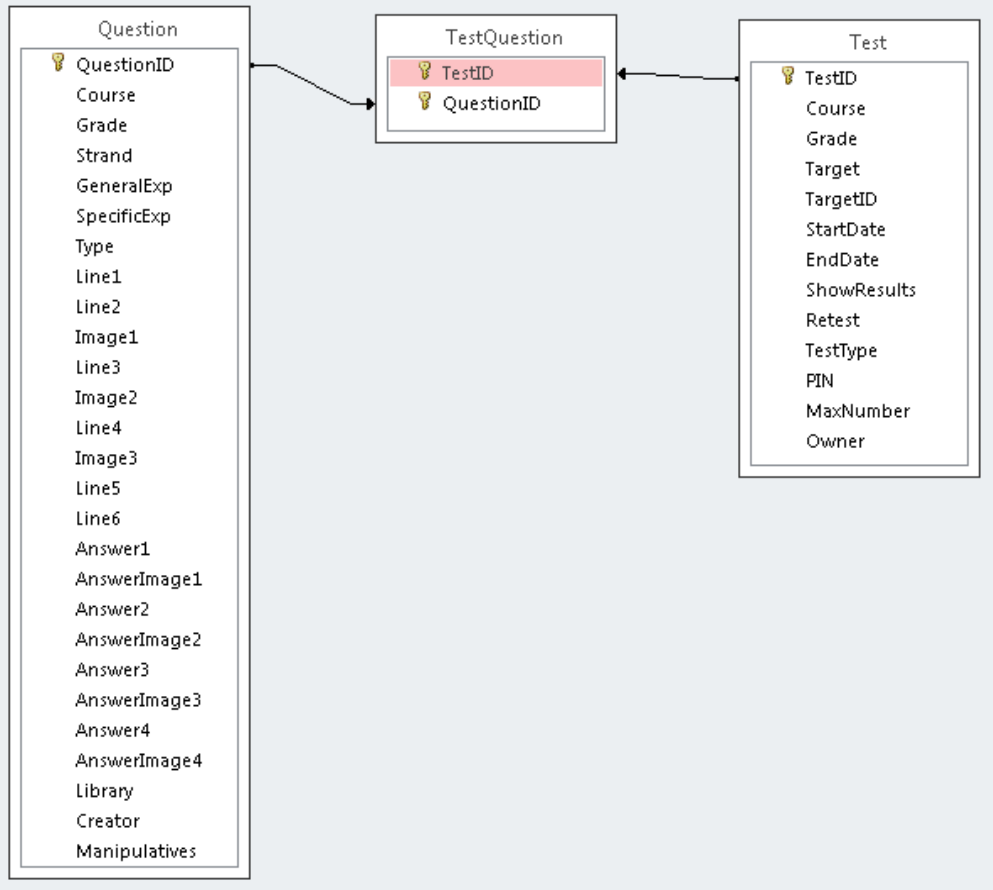


Sample Data:

TF123456,Math,3,Class,CL12345,2016/09/20,2016/09/20,Yes,No,Formal,1234,10,111222333
TP123456,Math,3,Group,GRP123456,2016/10/01,2016/10/06,Yes,No,Practice,4321,7,111222333
TP654321,Math,3,Student,123456789,2016/09/14,2016/09/16,No,No,Practice,2135,30,111222333
TF112233,Math,6,Class,CL12399,2016/10/10,2016/10/10,No,No,Formal,9999,4,111222333

5.9 TABLE: TESTQUESTION

Field Name	Descr
TestID	Prefix with "TF" for Formal test type. Prefix with "TP" for Practice test type. Followed by unique 6 digit number.
QuestionID	For Math, prefix with "QM" then unique 4 digit number

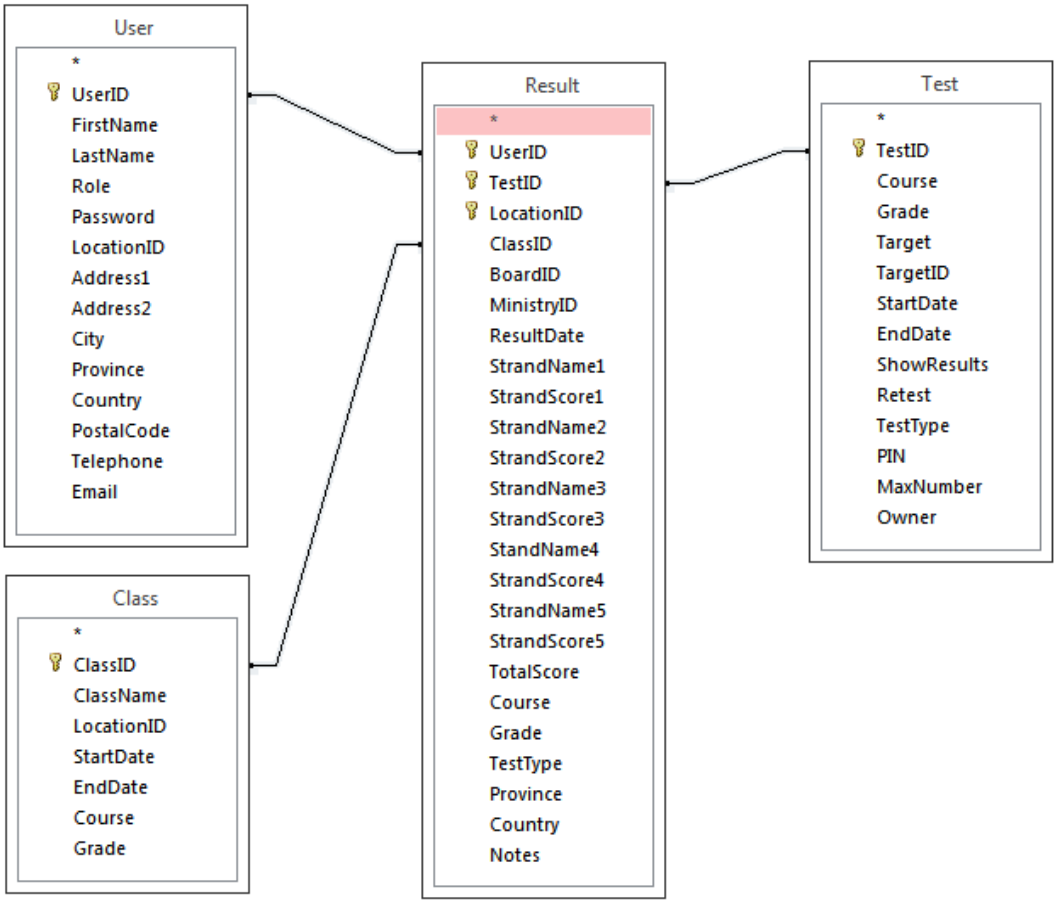


Sample Data:

TF123456,QM12345
TF123456,QM12564
TF123456,QM22987

5.10 TABLE: RESULT

Field Name	Data Type	Description (Optional)
UserID	Short Text	OEN Number
TestID	Short Text	Prefix with “TF” for Formal test type. Prefix with “TP” for Practice test type. Followed by unique 6 digit number.
LocationID	Short Text	Prefix “LOC” followed by unique 5 digit number
ClassID	Short Text	Prefix “CL” followed by unique 5 digit number
BoardID	Short Text	Prefix "BRD" followed by a unique 5 digit number
MinistryID	Short Text	Prefix "MNS" followed by a unique 5 digit number
ResultDate	Date/Time	
StrandName1	Short Text	
StrandScore1	Number	
StrandName2	Short Text	
StrandScore2	Number	
StrandName3	Short Text	
StrandScore3	Number	
StandName4	Short Text	
StrandScore4	Number	
StrandName5	Short Text	
StrandScore5	Number	
TotalScore	Number	
Course	Short Text	Valid values: Math; other course will follow in time
Grade	Short Text	Valid values: K to 12
TestType	Short Text	Valid values: "Formal", "Practice" or "ExitTicket"
Province	Short Text	
Country	Short Text	
Notes	Long Text	

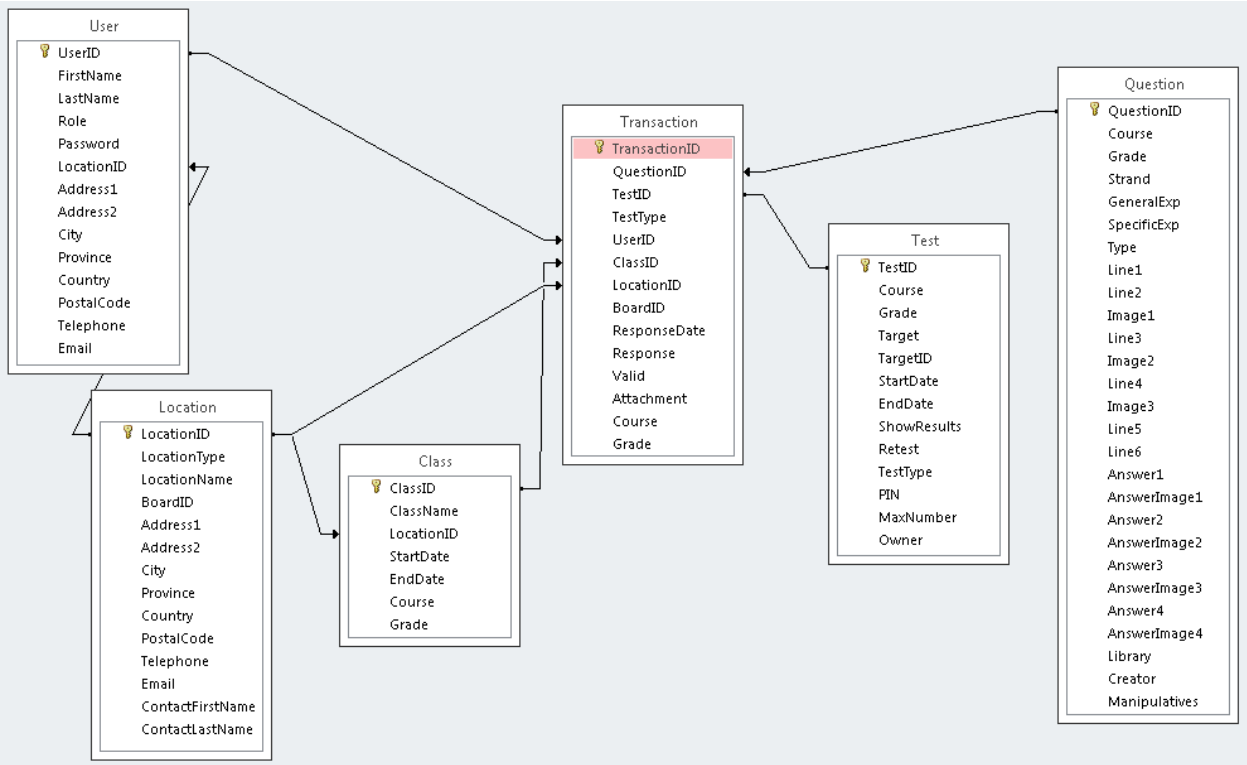


Sample Data:

123456789,TF123456,LOC12345,CL12345,BRD12345,MNS12345,2016/10/29,Number Sense and Numeration,11.1,Measurement,22.2, Geometry and Spatial Sense,33.3, Patterning and Algebra,44.4, Data Management and Probability,55.5,99.9,Math,3,Formal,ON,Canada,Some notes go here

5.11 TABLE: TRANSACTION

Field Name	Desc
TransactionID	
QuestionID	For Math, prefix with "QM" then unique 4 digit number
TestID	Prefix with "TF" for Formal test type. Prefix with "TP" for Practice test type. Followed by unique 6 digit number.
TestType	Valid values: "Formal", "Practice" or "ExitTicket"
UserID	OEN Number
ClassID	Prefix "CL" followed by unique 5 digit number
LocationID	Prefix "LOC" followed by unique 5 digit number
BoardID	Prefix "BRD" followed by a unique 5 digit number
ResponseDate	
Response	user selected response
Valid	was the response correct; valid values: True or False
Attachment	path to "show your work" image file
Course	Valid values: Math; other course will follow in time
Grade	Valid values: K to 12
Strand	
GeneralExp	
SpecificExp	

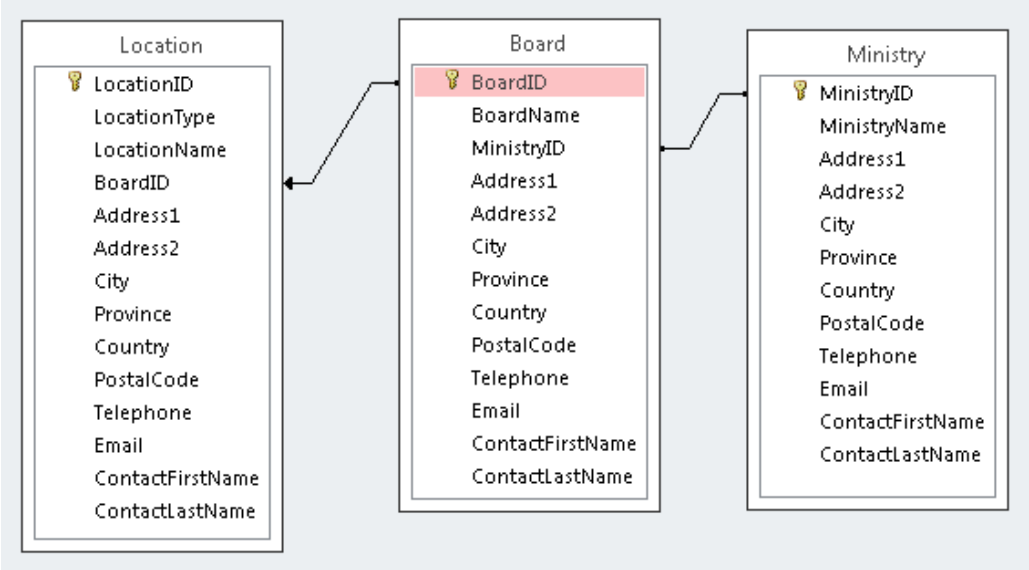


Sample Data:

000000001,QM1234,TF123456,Formal,123456789,CL12345,LOC12345,BRD12345,2016/10/29,$3.30,Yes,C:\Files\image.png,Math,3, Number Sense and Numeration, Quantity Relationships, “represent, compare, and order whole numbers to 50”

5.12 TABLE: BOARD

Field Name	
BoardID	Prefix "BRD" followed by a unique 5 digit number
BoardName	
MinistryID	Prefix "MNS" followed by a unique 5 digit number
Address1	
Address2	
City	
Province	
Country	
PostalCode	
Telephone	
Email	
ContactFirstName	
ContactLastName	

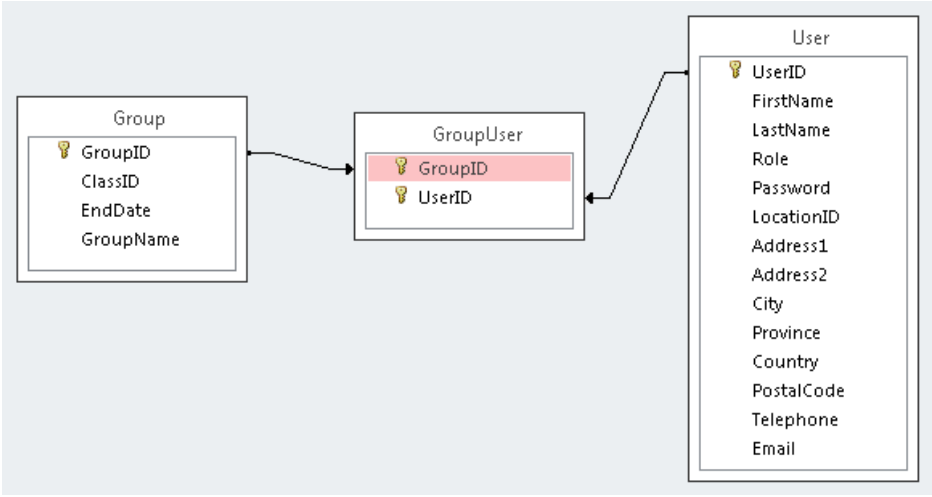


Sample Data:

BRD12345,Board Name,MNS12345,Address Line 1,Address Line 2,Toronto,ON,Canada,M5V 3J6,555-555-5555,board@email.com,Jack,Black

5.13 TABLE: GROUPUSER

Field Name	
GroupID	Prefix "GRP" followed by unique 6 digit number
UserID	OEN Number

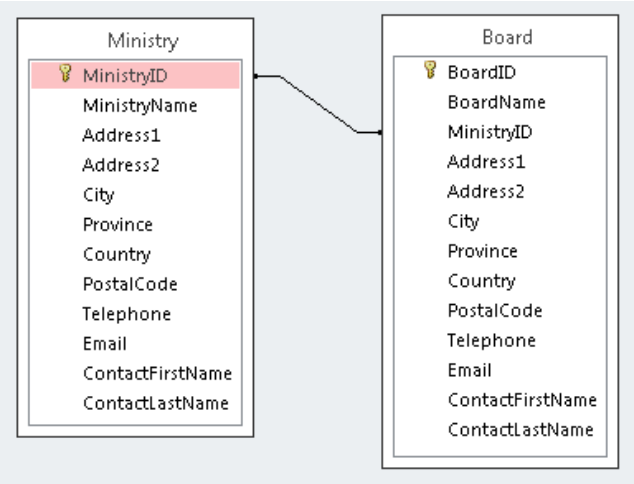


Sample Data:

GRP12345,123456789
GRP12345,986498675
GRP45378,123456789
GRP45378,345127556

5.14 TABLE: MINISTRY

Field Name	
MinistryID	Prefix "MNS" followed by a unique 5 digit number
MinistryName	
Address1	
Address2	
City	
Province	
Country	
PostalCode	
Telephone	
Email	
ContactFirstName	
ContactLastName	



Sample Data:

MNS12345,Ministry Name,Address Line 1,Address Line 2,Toronto,ON,Canada,M5V 3J6,555-555-5555,minister@email.com,Frank,Sinatra

6 COLOUR PALETTE

Use the following chart as a guide to UI colouring. Curriculum strand colours should follow the chart colours in the order displayed below.

HEX	R	G	B	Sample
603CBA	96	60	186	
00A300	0	163	0	
FFC40D	255	196	13	
1E7145	30	113	69	
9F00A7	159	0	167	
E3A21A	227	162	26	
2D89EF	45	137	239	

Basic UI colours:

HEX	R	G	B	Sample
E9E9E9	233	233	233	
CCCCCC	204	204	204	
D6D6D6	214	214	214	
3A3A3A	58	58	58	
666666	102	102	102	
F9F9F9	249	249	249	
000000	0	0	0	

7 PROJECT DELIVERABLES

The project deliverables section of this document does NOT represent the whole of the work required for this project, rather it is meant to facilitate project discussions, meetings and updates. This document should be considered in its entirety to successfully deliver this project.

This list of deliverables has been broken down into sections that follow those outlined in the preceding portion of this document. Therefore, detailed information on specific list items can be found by simply referring to prior document sections of the same name (and their associated sub-sections).

7.1.1 TEACHER LOGIN

- ☐ build login screen for teacher role
- ☐ security: use plain text passwords for user login until September milestone
- ☐ security: implement encrypted passwords for user login after September milestone

7.1.2 LIBRARY TAB

- ☐ build UI screen for Library tab
- ☐ initial display list of questions should, by default, be filtered by Creator (User ID) and Library type equal to "Private"
- ☐ New questions populate the Creator field with the currently logged in User ID
- ☐ New questions are automatically tagged as "Private" for the Library type
- ☐ The Course drop list should query the database for available courses
- ☐ Grade drop list hard coded to display grades 12 to Kindergarten
- ☐ Strands drop list populated by DB query
- ☐ Expectations drop list populated by DB query
- ☐ Type drop list hard coded to use "Multiple Choice" and "Text Input"
- ☐ Add drop list for Manipulatives
- ☐ Unique, application generated Question IDs using naming convention
- ☐ Question setup should allow for 6 lines of text and 3 images

- ☐ Answer setup for multiple choice questions should allow for 1 correct and 3 incorrect answers (can also be images instead of text)
- ☐ Answer setup for text input questions should only allow for 1 correct answer of text only
- ☐ Question preview pop up window

7.1.3 ASSESSMENTS TAB

- ☐ a unique Test ID should be assigned to all new assessments
- ☐ when new assessments are saved back to the DB, the Owner field must be set to the logged in User ID
- ☐ Course drop list is a filtered list by User ID
- ☐ Grade drop list is populated by DB query
- ☐ Target is hard coded to Class, Group or Student
- ☐ Other drop lists are visible or invisible depending on the selected Target
- ☐ Class drop list is populated by DB query
- ☐ Group drop list is populated by DB query
- ☐ Student drop list is populated by DB query
- ☐ Start Date and End Date should default to the current date
- ☐ Type drop list will be hard coded to use Formal and Practice
- ☐ If Type is "Practice" then Retest is visible
- ☐ Retest should be hard coded to Allow and Do Not Allow
- ☐ If Type is "Formal" then Retest is invisible
- ☐ Show Results drop list should be hard coded to use Yes or No
- ☐ Assignment drop list should be hard coded to use Selected Questions and Random Questions
- ☐ Other drop lists are visible or invisible depending on the selected Assignment
- ☐ Strand drop list should be populated by DB query
- ☐ Expectations drop list should be populated by DB query

- ☐ Library drop list should be hard coded to use All, Public or Private
- ☐ Type drop list should be hard coded to use All, Multiple Choice or Text Input
- ☐ Manipulatives drop list should be hard coded to use All, Allow or Do Not Allow
- ☐ Number of Questions input spinner should default to 3
- ☐ List of questions should be filtered and displayed in blocks of 10
- ☐ Every question should be preceded by a selection check box
- ☐ the application should remember all selections
- ☐ clicking on the Question ID should pop up a preview window of that question
- ☐ Preview Assessment button should open a new browser tab
- ☐ Assign Assessment button should advance to the Confirmation screen
- ☐ PIN number should be a randomly generated four digit number
- ☐ Selected questions should be organized by Strand and in colour coded columns
- ☐ User should be able to deselect questions from the Confirmation screen
- ☐ User should be able to preview specific questions from the Confirmation screen by clicking on the Question ID (pop up window)
- ☐ Cancel button takes the user back to the Assessment setup screen
- ☐ Save button commits the new assessment to the DB and moves the user to the Calendar tab

7.1.4 CALENDAR TAB

- ☐ Calendar tab leverages jqxScheduler code
- ☐ the calendar defaults to the Month view
- ☐ when opened from the left side menu bar, the calendar should focus on the current date
- ☐ when opened as part of the assessment setup process, the focus should be on the new assessment start date
- ☐ each block of scheduled time should show the Course and specific Target for the assessment
- ☐ double clicking on a time block should open a pop up window with assessment details

- ☐ from the detail pop up window, the user should be able to modify or delete an assessment (future assessments only)
- ☐ all events displayed in the calendar should be the result of a DB query
- ☐ detail pop up window should include Course, Grade, Target, Start Date, End Date, Retest, Show Result, Test Type and PIN
- ☐ Only the Target, Start Date, End Date, Retest, Show Result and Test Type fields can be modified by the user. Changing anything else requires that the assessment be deleted and recreated.

7.1.5 STUDENT ROLE

- ☐ Build student login logic as described at the start of Section 4.1 earlier in this document. This is significantly different than in the demo.
- ☐ Code for browser session storage to hold data as described in Section 4.1.1 earlier in this document
- ☐ Student assessment style should exactly match that of the demo as displayed in Google Chrome browser
- ☐ Follow Question display guidelines as outlined in Section 4.2 of this document
- ☐ Implement Text Input type questions to use Literally Canvas as the drawing layer
- ☐ “Next”, “Previous” and “Submit Your Test” buttons should follow the logic described in Section 4.2.2 of this document
- ☐ Check that all questions are answered before allowing a test to be submitted for marking
- ☐ “Show Results” and “Retest” functionality should follow the logic outlined in Section 4.2.2
- ☐ When “Show Results” are displayed, it should follow the calculation logic described in Section 4.2.2
- ☐ Append test data generated by students to the Transaction and Result tables

8 SOFTWARE

After consulting with NC, all third party software required for this project will be provided by NLS. However, every consideration should be made for using open source software for this project; for example, Cassandra database, charts from Highcharts.com, etc.

9 CONTACT INFORMATION

If you have any questions, please contact either:

Nick Tomljenovic
Email: nt@neols.com
Telephone: 647-808-1160

Dave Berry
Email: dave@neols.com

Ryan Solomon
Email: ryan@neols.com

10 NON-DISCLOSURE

All of the information within this document is considered to be confidential and the property of Neo Learning Systems. The software developer will treat this document, and any other information or materials disclosed or provided, as private and confidential and will not disclose, reproduce or transmit said information and materials to any other party without the written consent of Neo Learning Systems. It is the responsibility of the software developer to ensure that all materials provided are returned to Neo Learning Systems.