NEO Learning Systems

design document – Project Raptor

teacher app

july 2017

VERSION 1.2

CONFIDENTIAL

# Table of Contents

[1 Table of Contents 2](#_Toc489267767)

[2 Overview 3](#_Toc489267768)

[2.1 Project Objectives 3](#_Toc489267769)

[3 TeacherRole 4](#_Toc489267770)

[3.1 Calendar 4](#_Toc489267771)

[3.1.1 Tables to Use 5](#_Toc489267772)

[3.2 LibraryTab 5](#_Toc489267773)

[3.2.1 Question Tags 6](#_Toc489267774)

[3.2.2 Text ANd Images 7](#_Toc489267775)

[3.2.3 Answers 8](#_Toc489267776)

[3.2.4 Buttons 9](#_Toc489267777)

[4 Tickets Tab 9](#_Toc489267778)

[4.1 Setup 10](#_Toc489267779)

[4.2 Review 11](#_Toc489267780)

[5 Pending Tab 12](#_Toc489267781)

[5.1.1 Scoring Open Responses Questions 15](#_Toc489267782)

[5.1.2 Viewing The Solution 17](#_Toc489267783)

[6 Feedback 18](#_Toc489267784)

[7 White Board 19](#_Toc489267785)

[7.1.1 Question Selection 19](#_Toc489267786)

[7.1.2 Whiteborad 20](#_Toc489267787)

[8 Color Palette 21](#_Toc489267788)

[9 Figure Guide 23](#_Toc489267789)

[10 Non Disclosure 24](#_Toc489267790)

# 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.

## Project Objectives

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

* Change the UI to meet the guidelines set by the Ontario Curriculum
* Create the appropriate level of application security for the Students.
* Responsiveness of the Project for Students to be able to view in any device in Chrome browser.
* Deliver the changed functioning applications by July 2017.

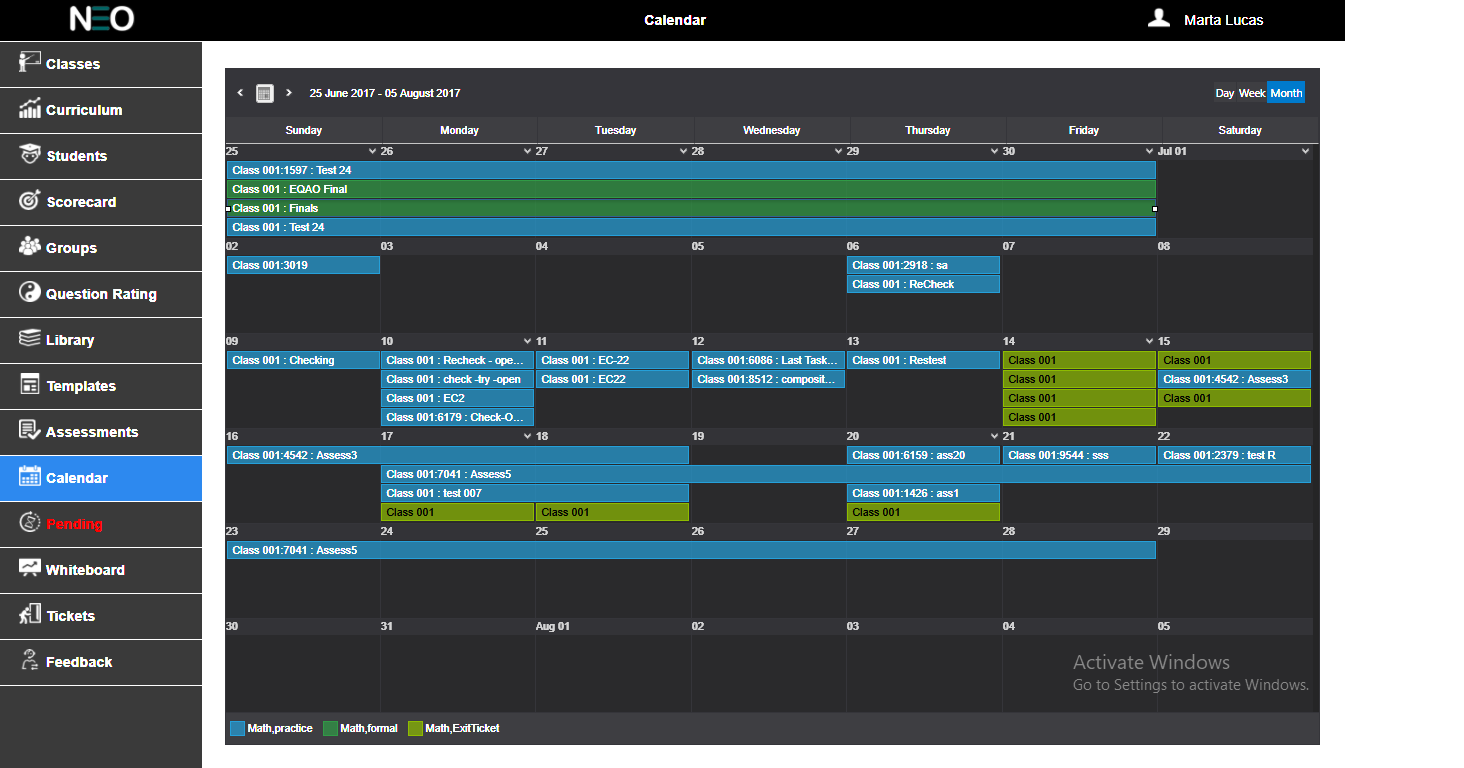
# TeacherRole

The following sub-sections correspond to the various screens or tabs that have changed to the changing requirements of the customer.

## Calendar

The Calendar tab, pictured in Figure 1 below, is used by the teacher role to review, modify or delete their scheduled assessments. Only future assessments can be modified and/or deleted; assessment older than the current date cannot be deleted or changed.

Only the Assessment Details (see Section) 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. Currently user cannot modify the tests – mention anything connected to modify as a future enhancement



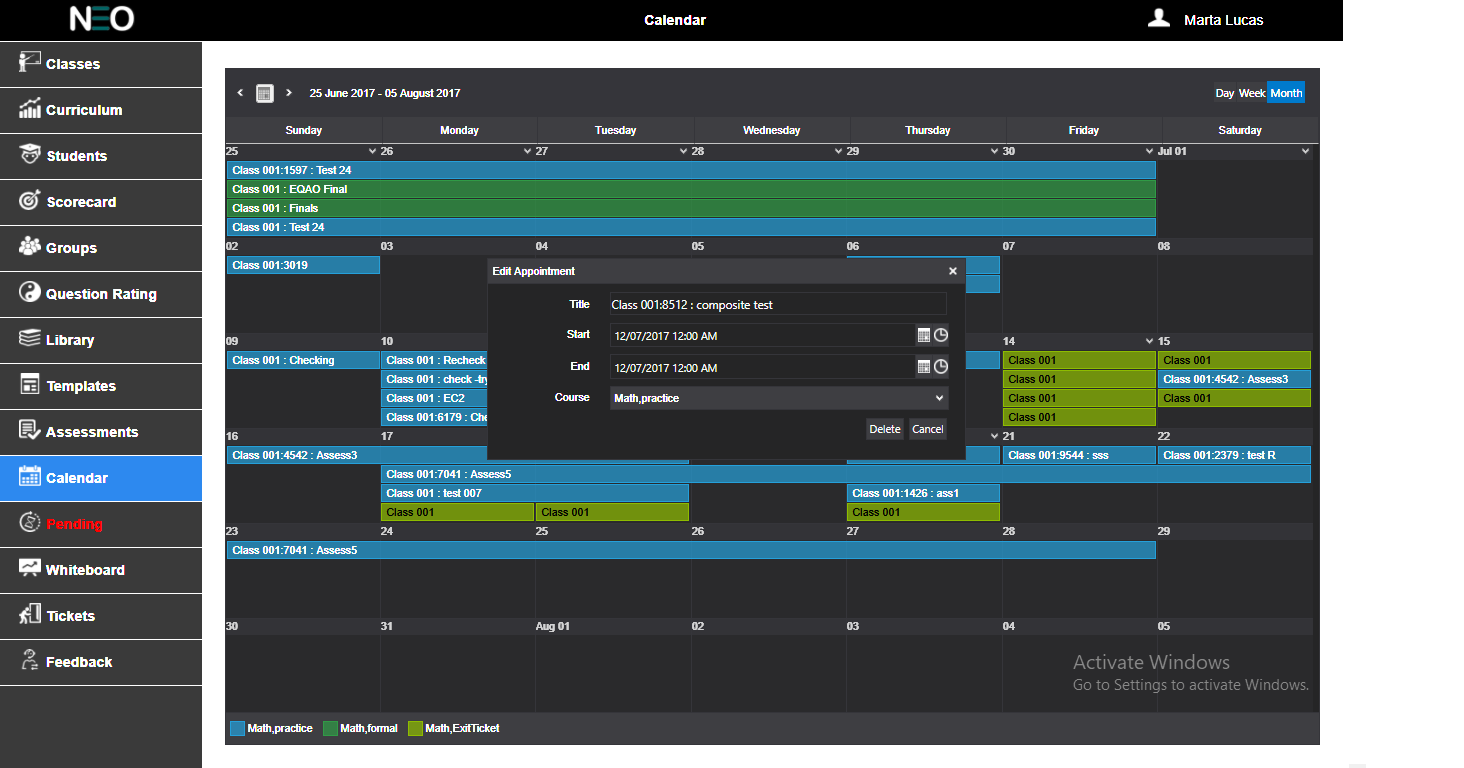
**Figure 1: Calendar Tab**

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. – Not clear enough

Double clicking on the block (on any particular test date) will bring up pop up window with the assessment details, see Figure 2 below. From this point, the user should be able to modify or delete the assessment entirely (applies to future assessments only).

.

**Figure 2: Calendar Tab– Preview**

Within the assigned block of time for the assessment, the user should be able to see the specific target, assessment name and pin number (if selected) for the assessment.

The pop-up window will for now only display Title, Start Date, End Date, Course of the said assessment. The title will display the assessment name, target and pin number. The user can only delete future assessments. The legends in the Calendar should display Course and grade. The color of legend should change for different grades.

### Tables to Use

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

If this is an enhancement document then show information that are missing or enhanced when compared to the previous document. We don’t need the entire topic

## LibraryTab

The Library tab, pictured below in Figure 3, is used by teachers to create test questions for later use on student assessments (tests). This tab will be used to both edit existing questions and create new questions.

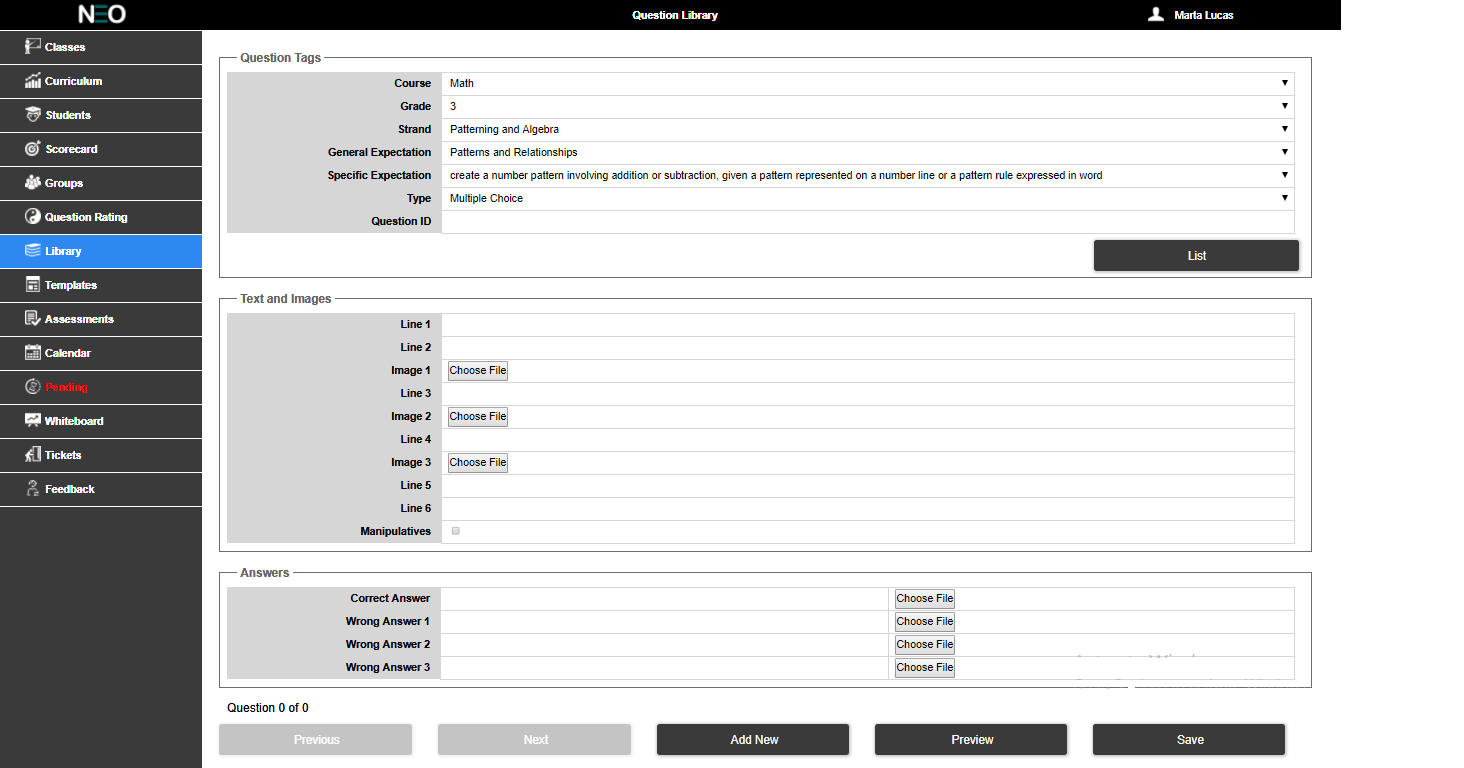


Figure 3: Library Tab

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.

### Question Tags

The question tags looks like below (Figure 4).The Course drop list should query the database for available courses that are taught by the teacher. Like the Course drop list, the Grades drop list should be populated by a database query and 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.

The Type drop list can be hard coded to use “Multiple Choice”, “Text Input” and “Open Response” and should default to multiple choice. The Question ID should be unique and follow a naming convention such as prefixing a 5 digit number with a “QM”.

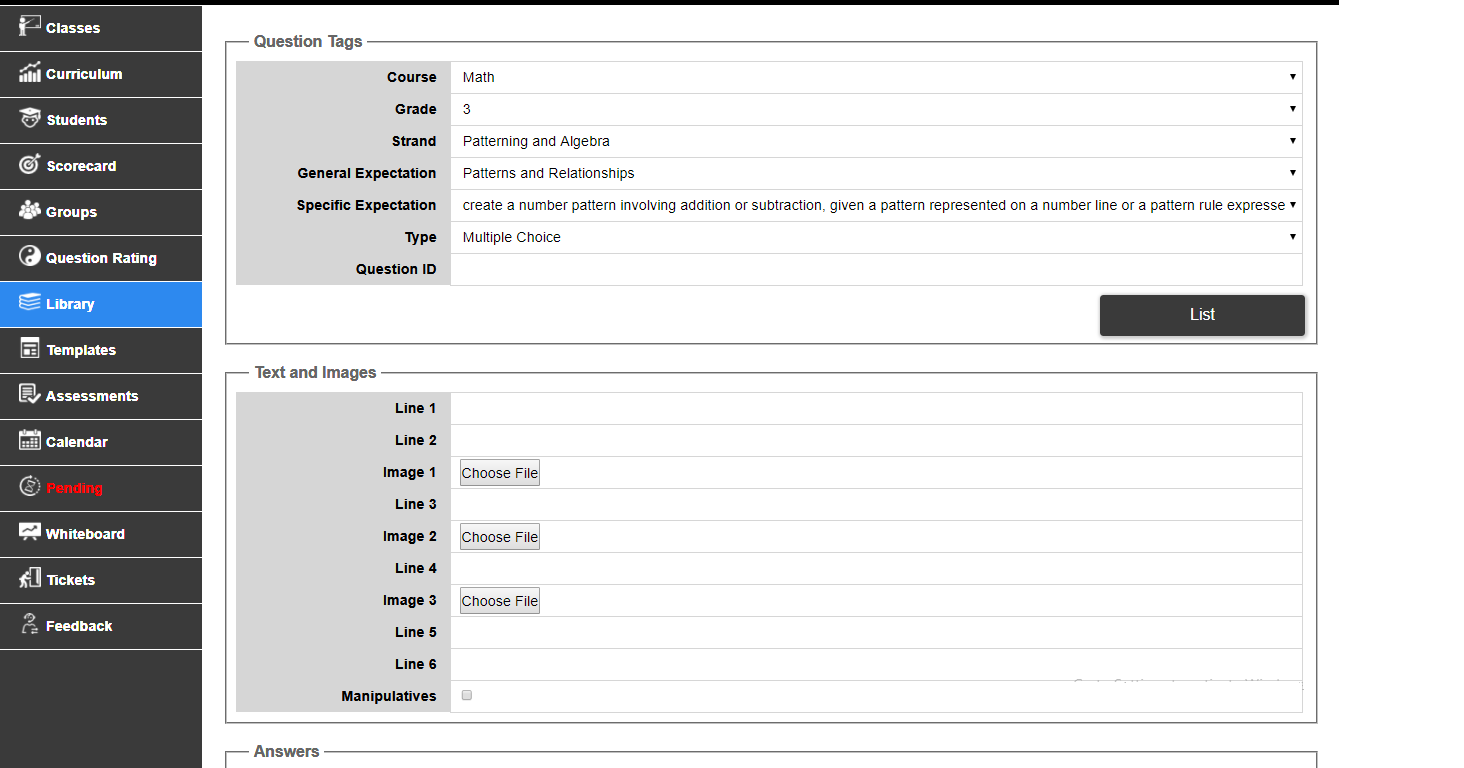


Figure 4: Question Tags

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. There is no restriction on the list of Grades in this module. Depending on the Grade selected Strand and Expectations are listed accordingly.

On click of List button the list of questions are displayed one by one. They 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 Previous and Next button. The Question Id is displayed only on click of the List button or when a new question is saved to the Library. This button should be used to run the query for a list of questions fitting the selected filters above.

The “List” button should be added between the Question Tags grouping and the Text and Images grouping.

### Text ANd Images

This part of the Question Library controls the layout of the question being edited/created (see below Figure 5). 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.A check box is added to this grouping called “Manipulatives”. The check box is defaulted to unchecked and saved as “false” in the field “Manipulatives” in the database.

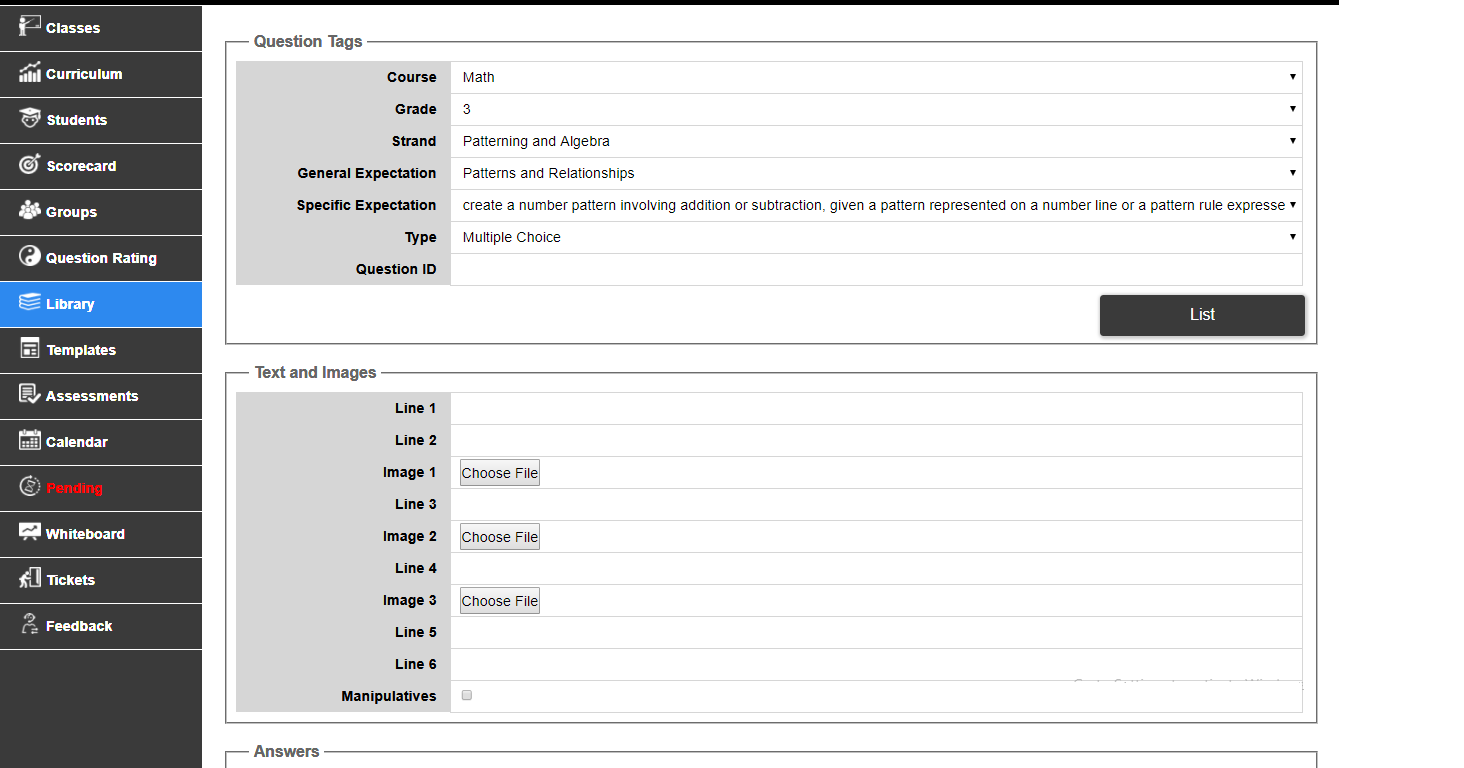


Figure 5: Text and Images

### Answers

This section is for the teacher to enter an appropriate answer for the question and the question type they have chosen in the Questions Tags section.

For multiple choice questions, the user should be able to enter one correct answer with three incorrect answers (see Figure 6)

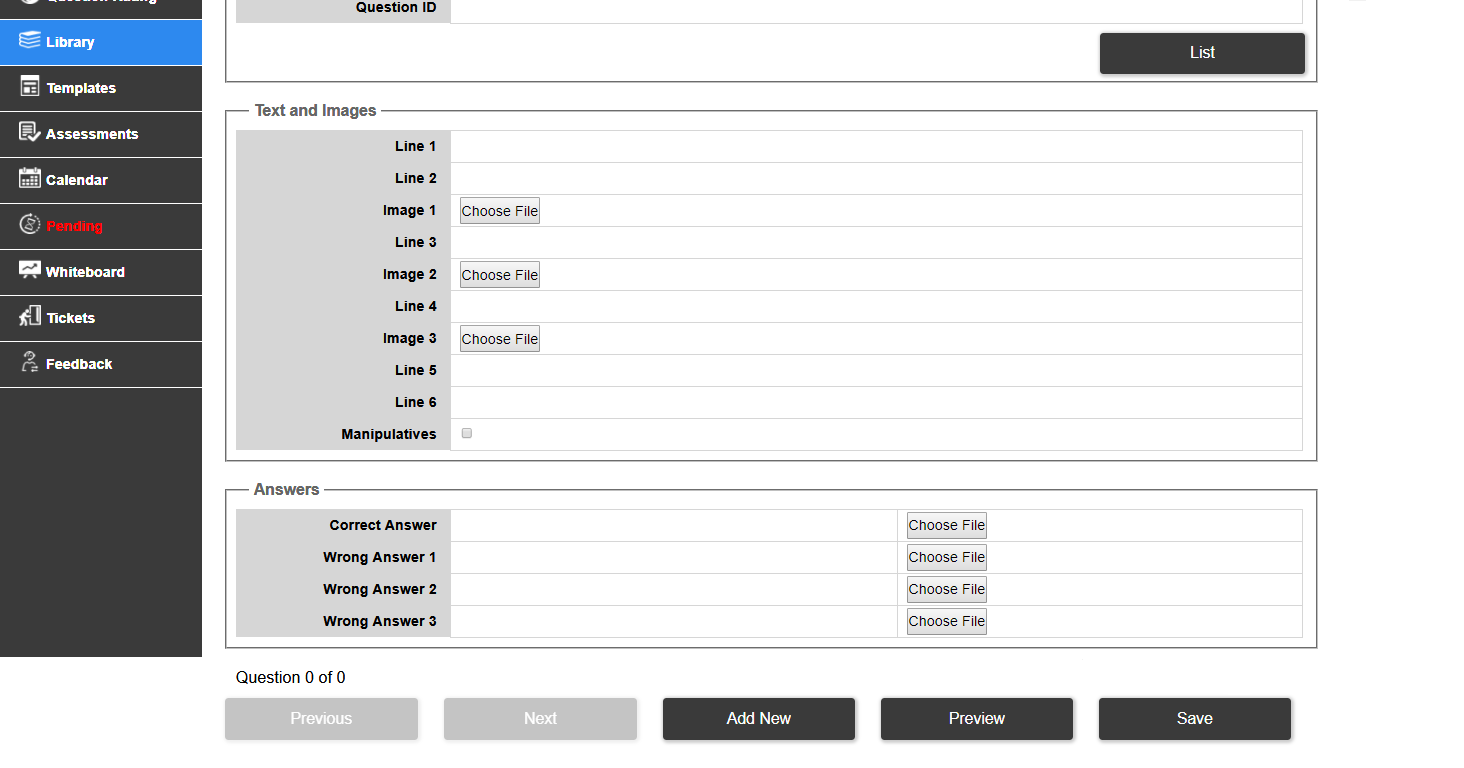


Figure 6: Answer - Multiple Choice

For questions of type “Text Input”, the Answers section gives the user the option to input a single correct answer along with a Leading Text and Trailing Text (see Figure 7). The leading text and trailing text are displayed in the Student App for Input Type questions and the values for this are stored under the field name “part1” and “part2” in the database.



Figure 7: Answer – Text Input

For questions of type “Open Response”, the Answer section gives the user the option to input a single correct answer along with an image that the student should in the student app if given (see Figure 8). The “Choose File” button should pop up a window that allows the user to browse for a specific file. Only one image file is allowed for selection per question.



Figure 8: Answer – Text Input

### Buttons

There are 5 buttons on this screen (see Figure 9):

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

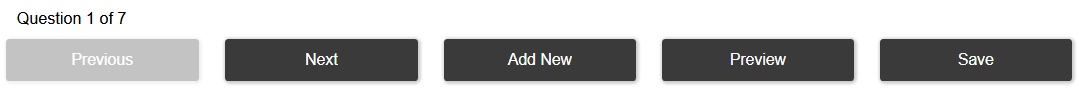


Figure 9: Buttons

Above the buttons, the total number of questions available entered by the logged in teacher is displayed. Clicking on the Next button the list iterates forward. The user is able to see the question number they are in, in compared to the total list.

Why was this tab added? There are no changes in the Tickets tab, just Exit Tickets was renamed to Tickets.

# Tickets Tab

An Exit Ticket (see Figure 10) is a test that only has a single question. It’s meant for the teacher to very quickly assign a single question to gauge the level of the class’ understanding.

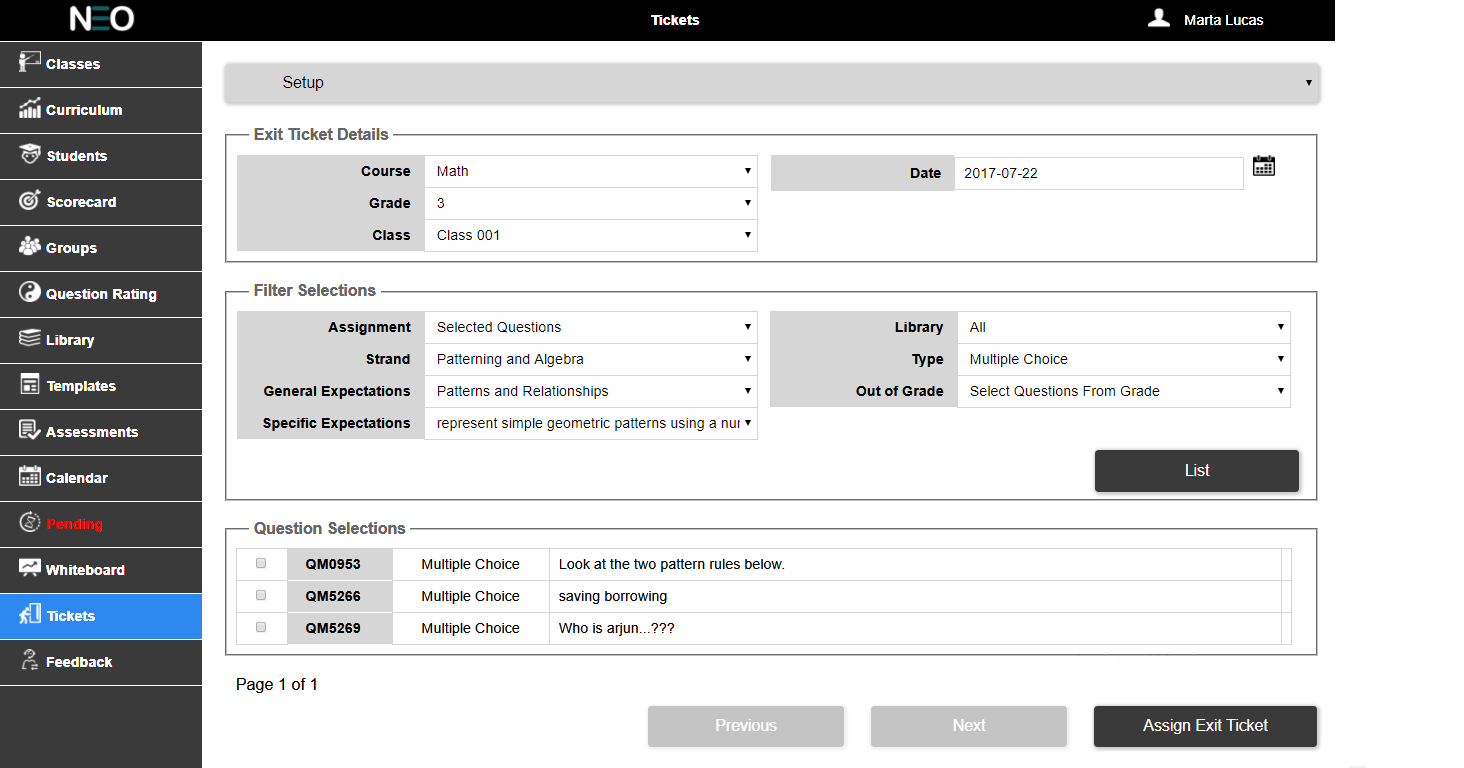
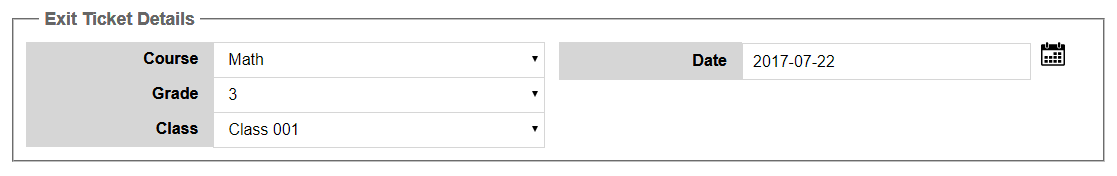


Figure 10: Exit Tickets Tab

At the top of this screen, there is a drop list with two options: Setup and Review. The Setup option looks like a simplified version of the Assessments tab. While the Review option allows the teacher to quickly see the result of the Exit Ticket.

## Setup

The Setup screen has three sections: Exit Ticket Details, Filter Selections, and Question Selection.

****

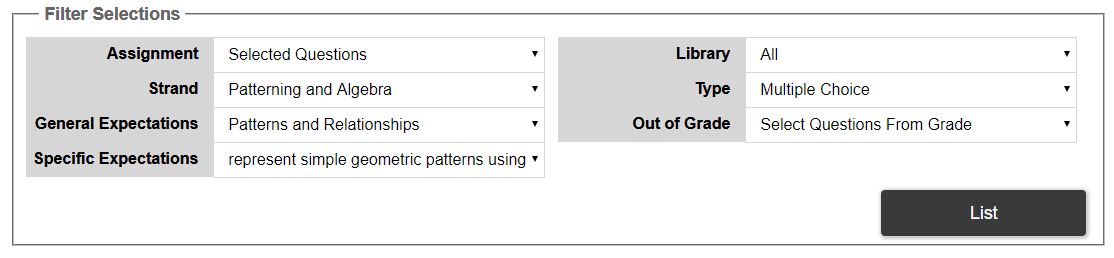
**Figure 11: Details Selections for the Exit Ticket**

The Exit Ticket Details field set contains the common drop lists for Course, Grade and Class; these are allfiltered by the logged in User ID as well (for example, if Teacher1 is logged in, only the courses taught byTeacher1 are listed in the Course drop list).The only Target for an Exit Ticket is always “Class”; therefore there is no need for a drop list to select thetarget type.

The Class drop list should be populated by querying the database for a list of classes using the selectedCourse, the selected Grade, and the currently logged in User ID as filters.

The date field should default to the current date. As a rule each specific class can only have one Exit

Ticket assigned per day.The test type for Exit Ticket is always “ExitTicket” (no space and this is how it should be saved in thedatabase).



**Figure 12: Filter Selections for the Exit Ticket**

The Filter Selection field set only has several filters but the option within those filters is restricted. The

Assignment filter should only have one option: Selected Question. In a future version of the software,there will be an option to randomly select a question, but that is not needed for this phase of theproject.

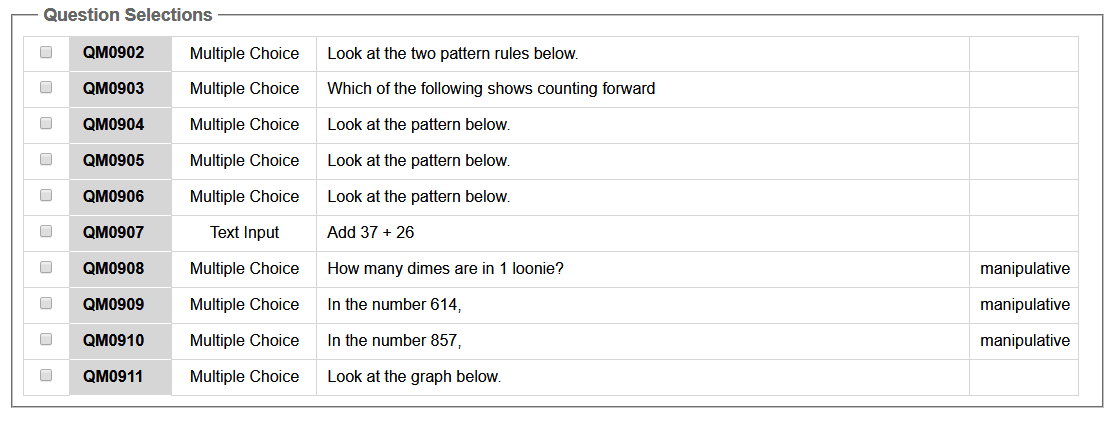
The Strand drop list should be populated with all of the available strands in the selected course.

However, “All” should NOT be an option; therefore the list should default to the first available coursestrand.

The Expectations drop list should be populated based on the user’s selection of Strand. For the generalexpectations, “All” is an acceptable option.

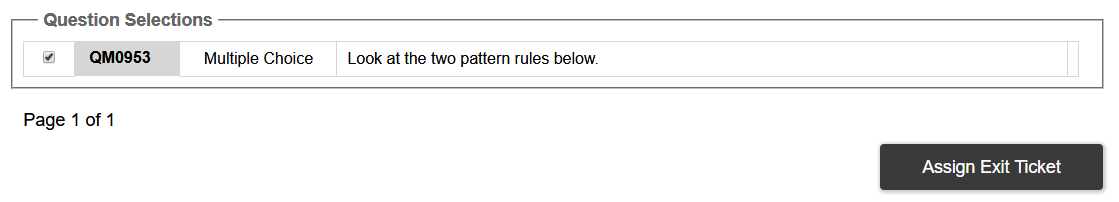
The Library drop list should have the choice of All, Public, or Private. “All” should be the defaultselection.The Type drop list should default to Multiple Choice.

Also, like the Assessments screen, a “List” button should be added within this field set and used to fetchthe appropriate list of questions.



**Figure 13: Question Selection for the Exit Ticket**

The Question Selection field set should be a filtered list of available questions. They should be displayedin blocks of 10 questions, and the fields used are: Question ID, Question Type, and the first line of thequestion. Each line should be preceded by a checkbox (to be used to select the question).Rolling over the Question ID should highlight that field, while clicking on the Question ID should bring apreview of that question. When a question is selected, all other choices should be hidden (see Figure 14 below).



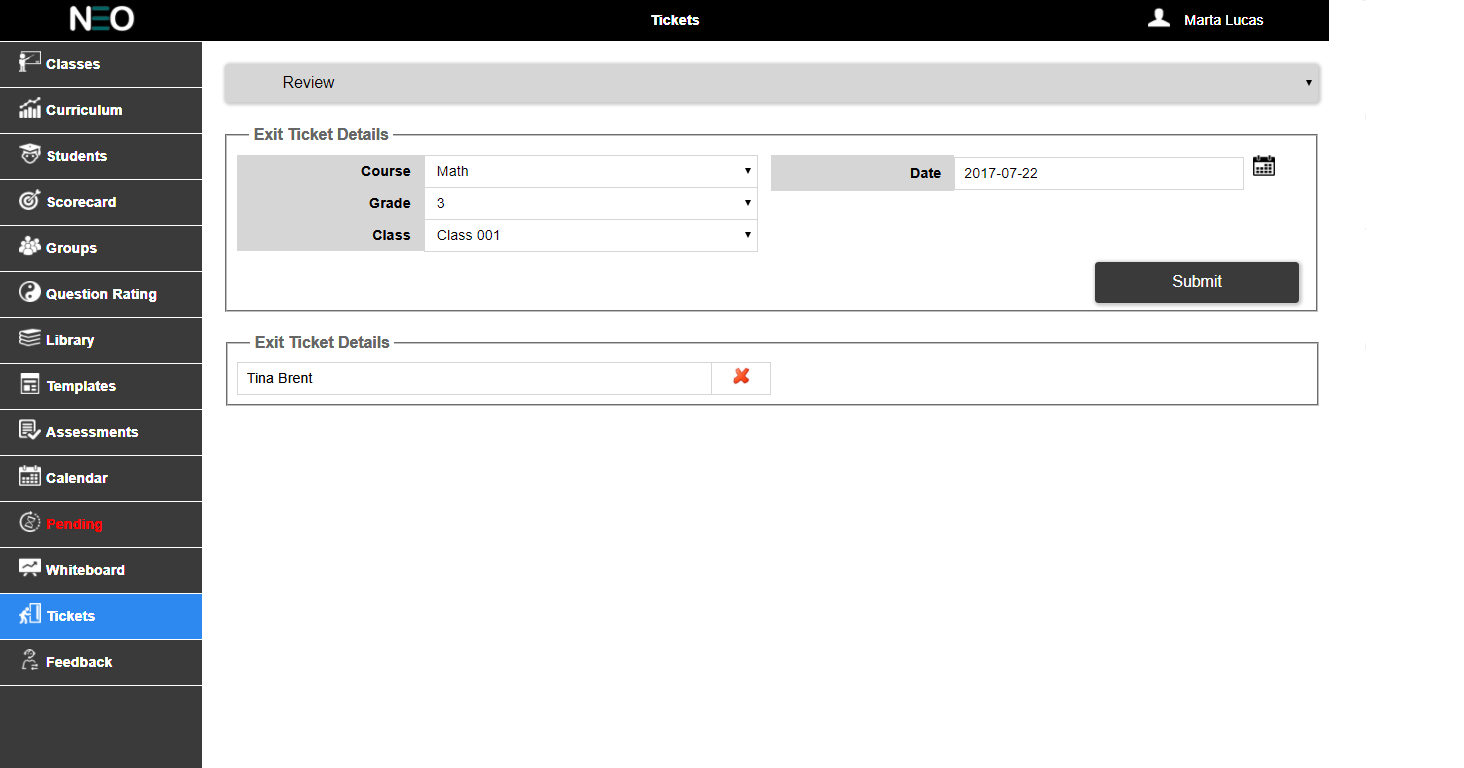
**Figure 14: Selected Question**

Clicking on the “Assign Exit Ticket” button, commits the test to the database. The Exit Ticket shouldappear on the Calendar tab just like Formal and Practice test types. Exit Tickets can only be deletedfrom the Calendar screen.

## Review

Switching to the Review option allows the teacher to see the results of a selected Exit Ticket (Figure 15).

The Exit Ticket Details field set persists to this screen and should also default to the current date.

****

**Figure 15: Exit Ticket Review**

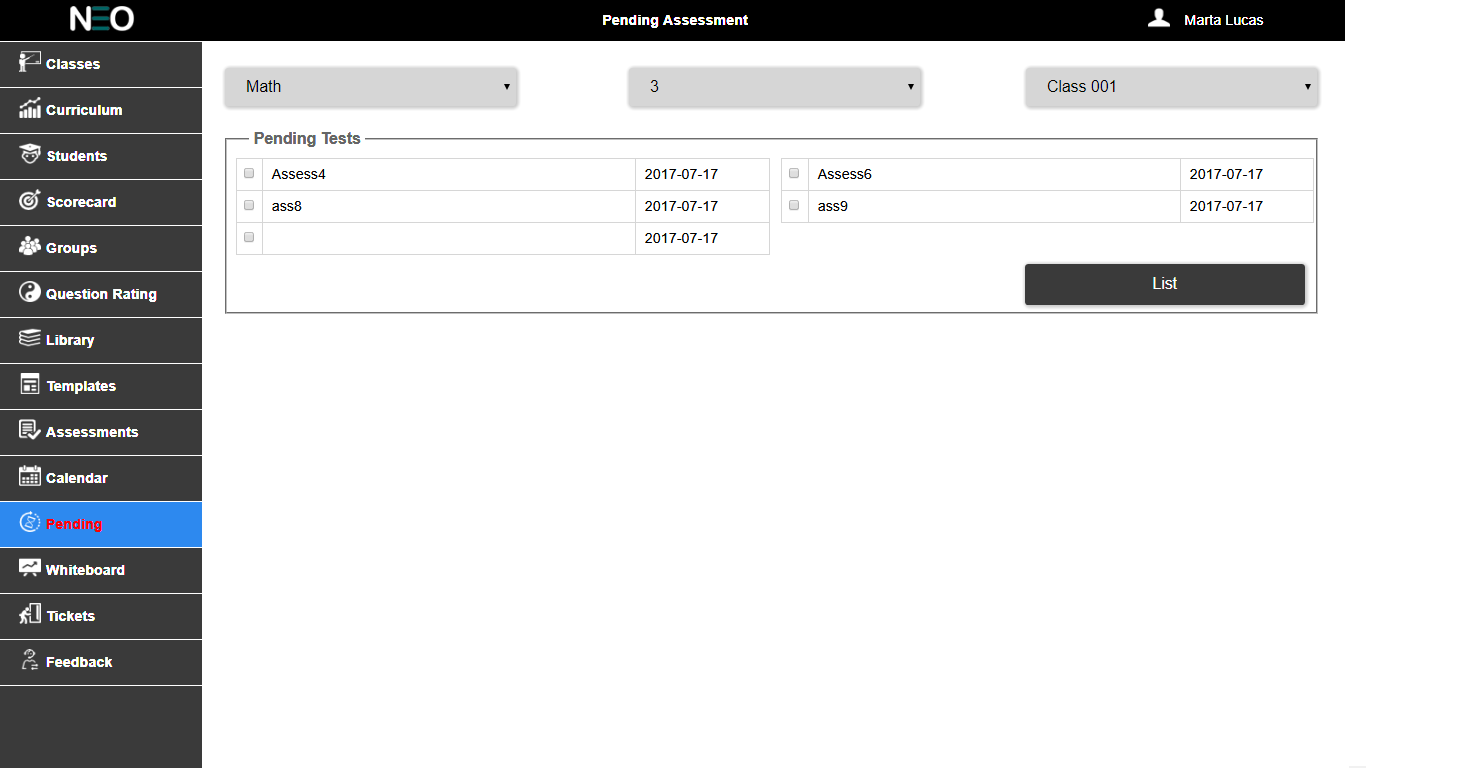
The Exit Ticket Results field set is simply a filtered list of student names (first and last) that answered the assigned Exit Ticket question incorrectly. This list is filtered by the selected Course, Grade and Class. As well as by Valid field of the database Transaction table (the value should be “False”). **Only thestudentswith incorrect responses are ever listed here.**

There should be a visual indication after every students name that they answered incorrectly (red X), as pictured in Figure 15 above.

This tab is also not required as there is a completely different requirement document for this. Any changes should be noted for ‘Pending’ module should be only noted

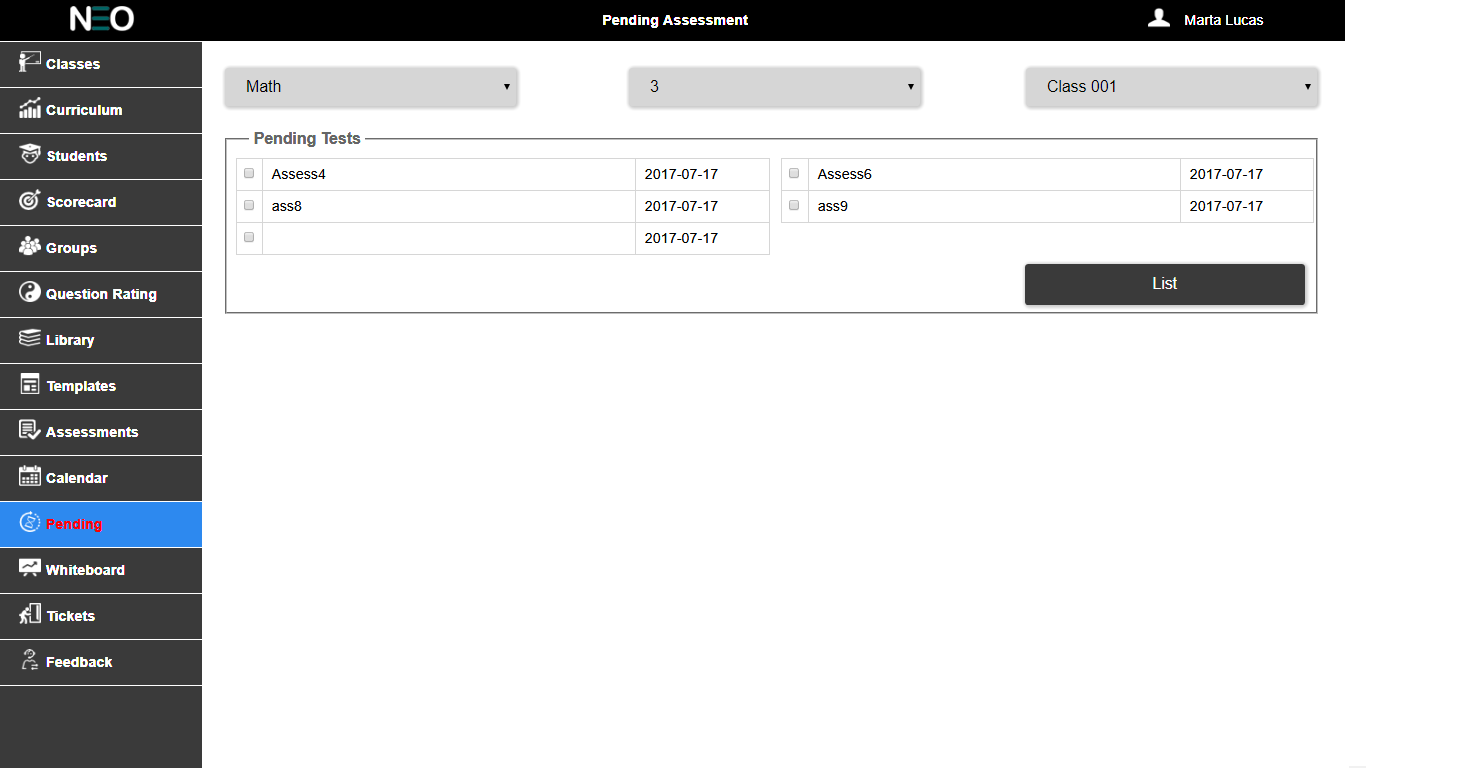
# Pending Tab

The pending tab will look like below (see Figure 16).



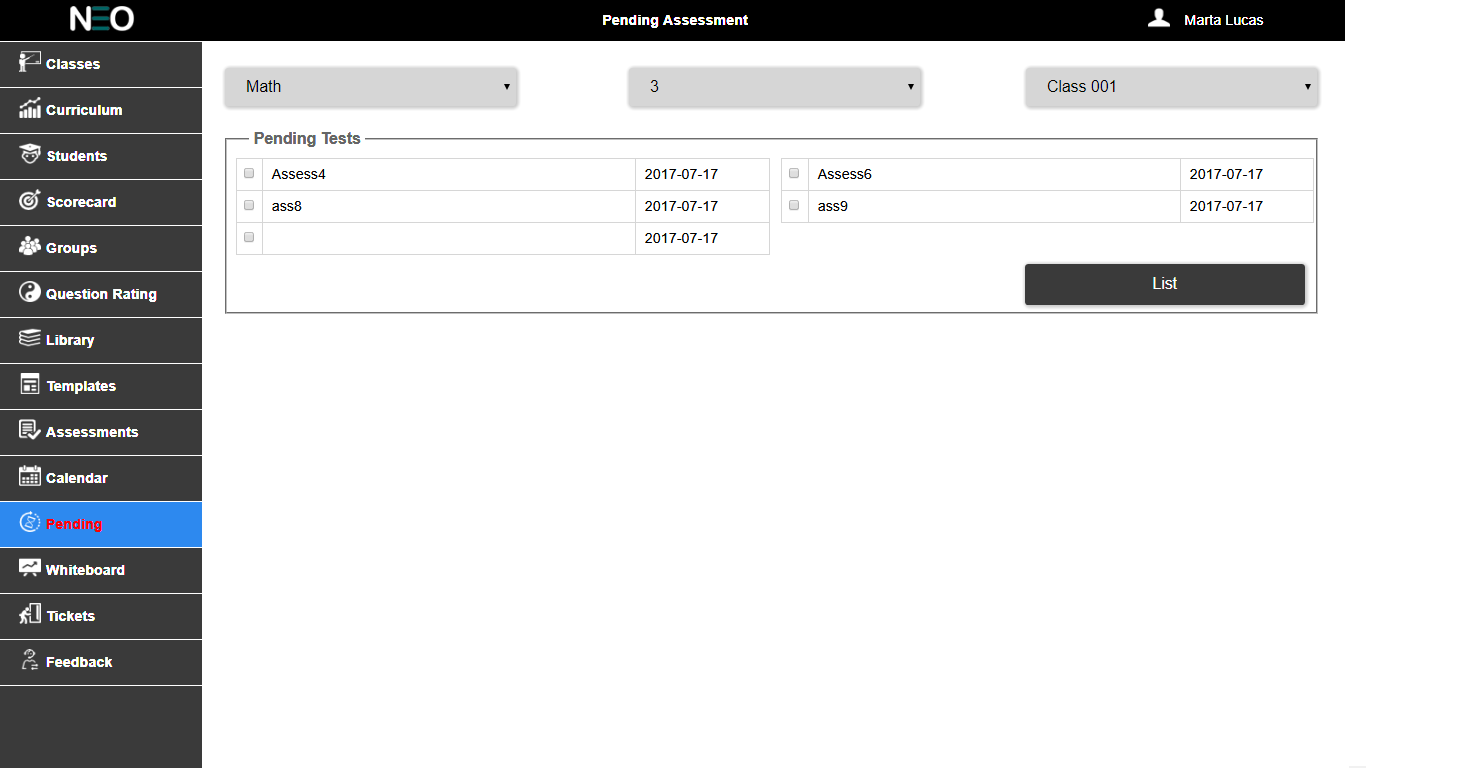
**Figure 16: Pending Tab**

If the user has to score an assessment then the font color of the pending should be in red color (HEX #ED1C24). If there is no assessment to score then the font color can be in white at all other times (see Figure17).



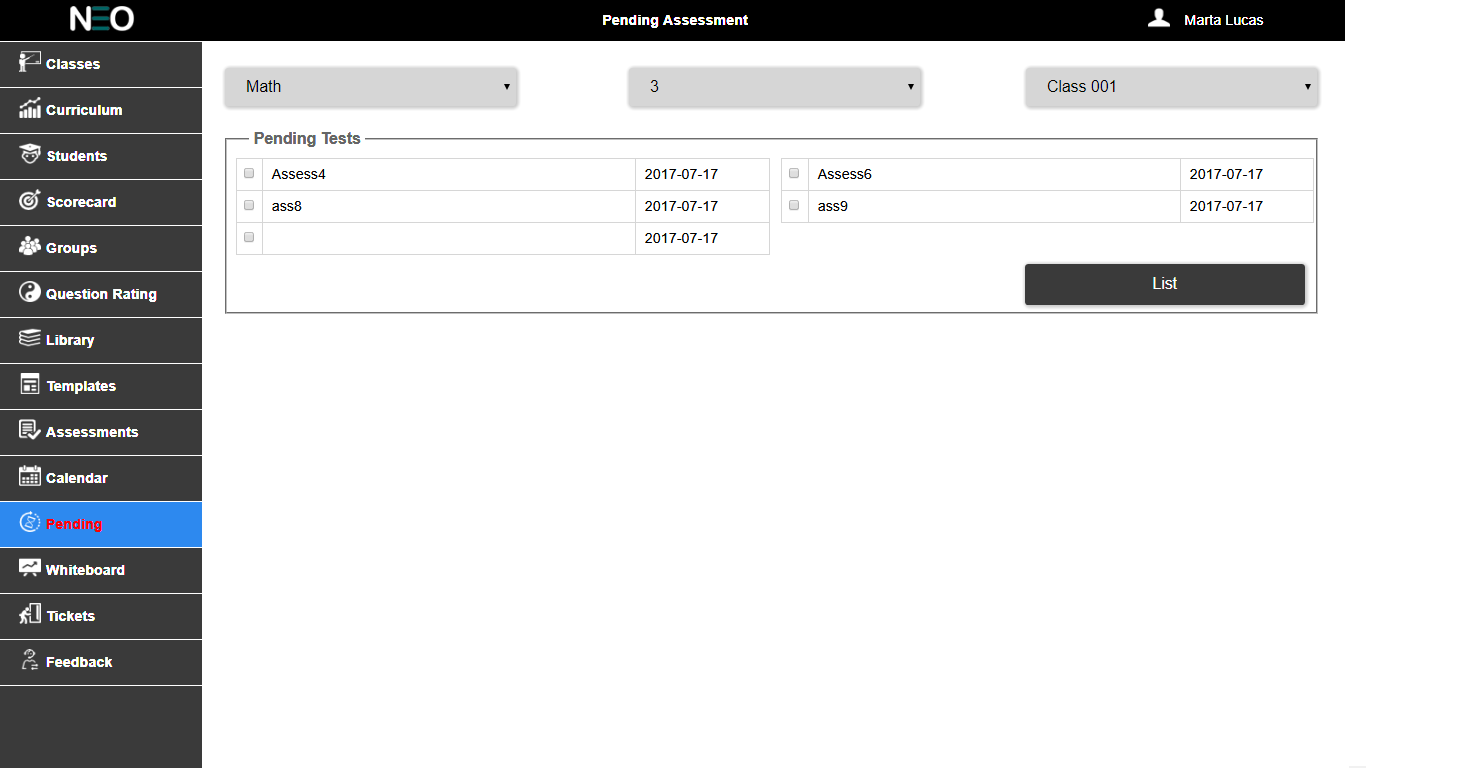
**Figure 17: Pending Tab Menu**

At the top of this screen there should be the standard UI filter elements for Course, Grade and Class. As shown in Figure 18 below.



**Figure 18: Pending Tab – Top Filters**

When the user first selects the Pending tab, the app should execute a query against the database Pending collection. The query should use the user selected values for Course, Grade and Class. This query should return a unique list of test names and associated dates. In turn these will be used to populate the top field set called “**Pending Assessments**” (See Figure 19 below).



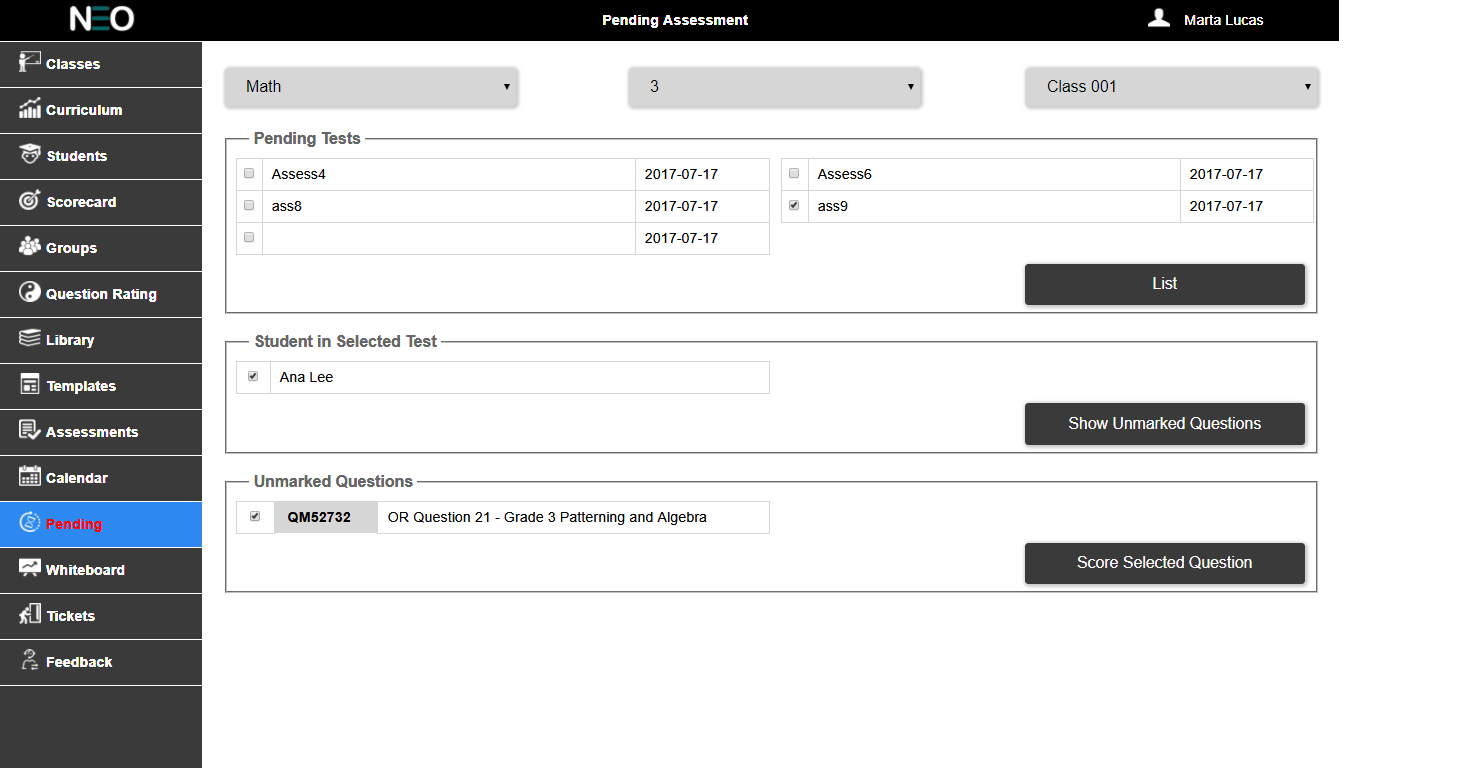
**Figure 19: Pending Test**

Each assessment will be preceded by a check box for selection purposes. Only one box can be selected at any given time. Selecting a box should automatically deselect all other check boxes. The results list should be sorted by date in ascending order. And the list should follow a Z-pattern back and forth between the two columns within the field set.

There should also be a list button within this field set to execute the second query to retrieve a list of student names associated with the selected test. This second query should also use the Pending collection to retrieve the needed data; filtering on the selected Test ID. The query should return a list of

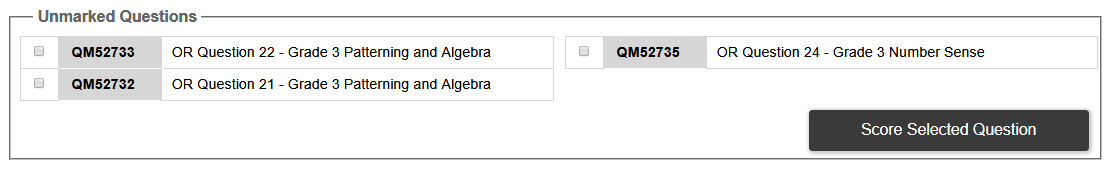
Student Names (User Name) and their OEN numbers (User ID).

Clicking on the List button should reveal the second field set called “**Students in Selected Assessment**”. The list of names should follow a Z-pattern back and forth between the two columns within the field set. Each name should be preceded by a check box for selection purposes (see Figure 20).



**Figure 20: Students in Selected Assessment**

The second field set also contains a button labeled: **Show Unscored Questions**. Clicking on this button will reveal a third field set (see Figure 21 below) that lists all unscored Open Response questions that relate to the selected student and the previously selected Test ID; this query should be against the Holding collection.



**Figure 21: Unmarked Questions**

The third field set should be called: **Unscored Questions**. Like the other field sets, it should consist of two columns going back and forth in a Z-pattern. Each result should be preceded by a check box for selection purposes. Followed by the Question ID number, which in turn is followed by Line 1 of the question. The user cannot select more than one question at a time. Within this field set is a button called: **Score Selected Question**. Clicking on this button, after selecting a question, will take the user to another screen where the teacher can finalize the question score.

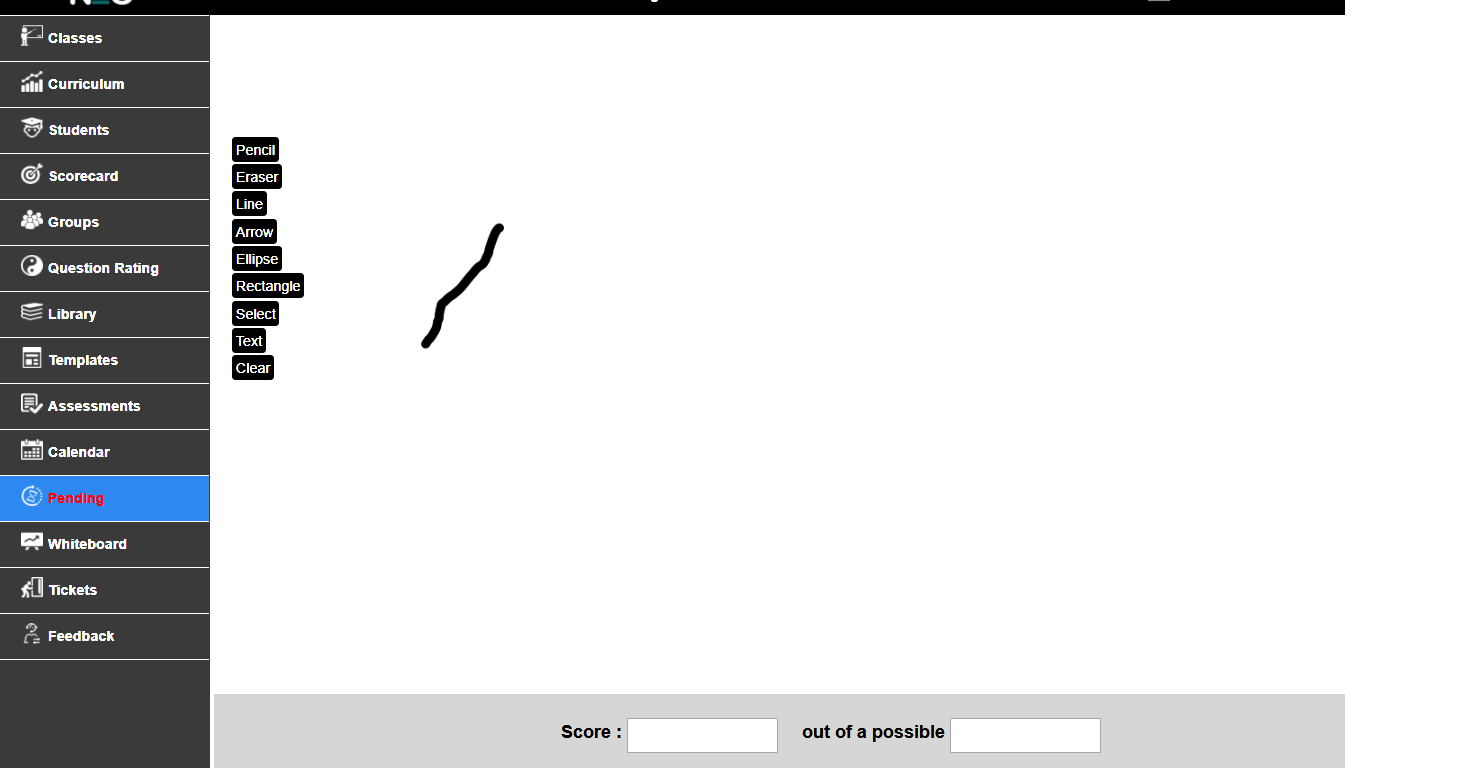
The selected course, grade, class, test ID and student ID will be required **after**the teacher has finished scoring the selected question. As such, this information should be temporarily stored somewhere for later use.

### Scoring Open Responses Questions

The scoring screen should consist of four elements:

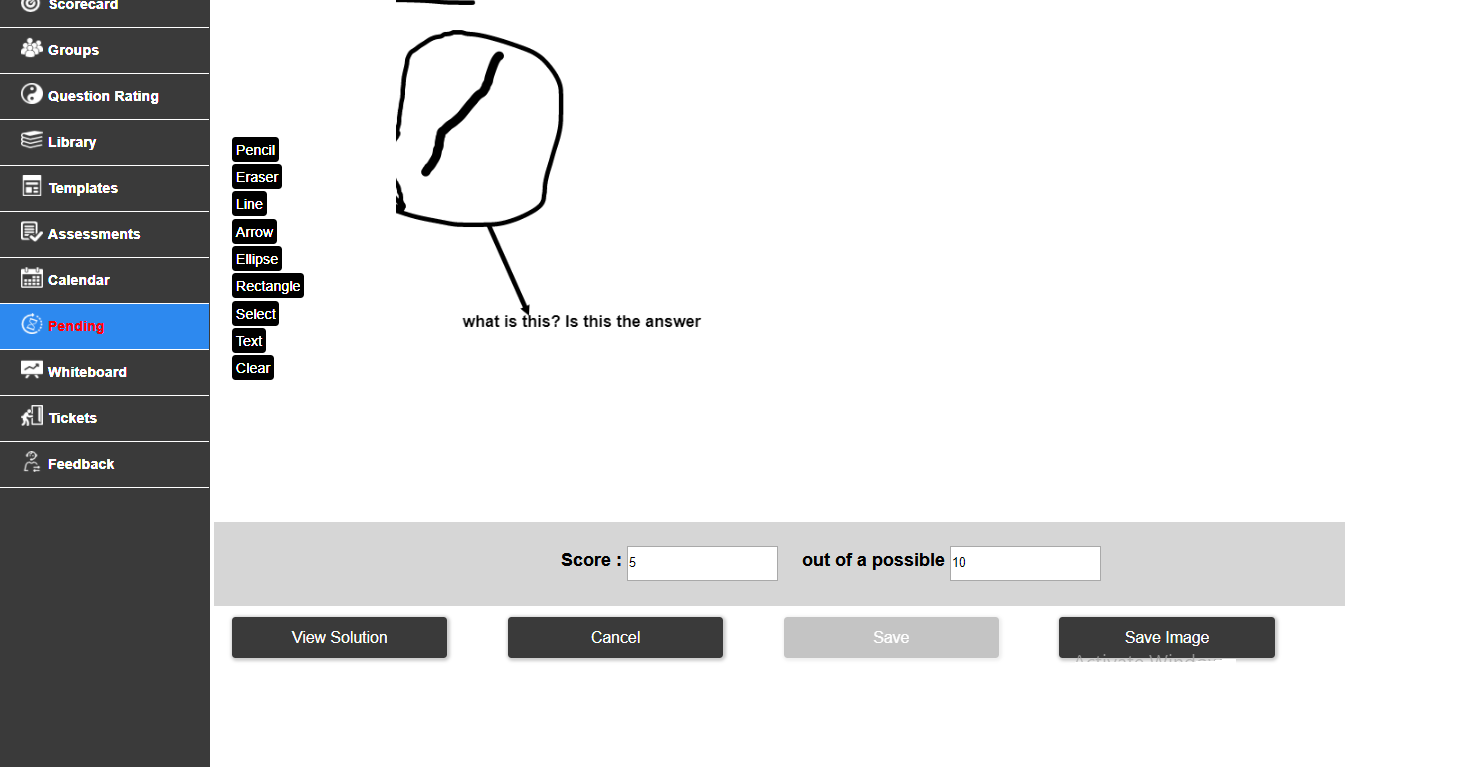
1. A background image, which is the image of the selected student’s response that was previously stored in AWS S3 (and reference in the database).
2. Literally canvas overlay. Including the standard drawing tools used previously and located along the left side of the screen.
3. Two text input boxes. The teacher will use these to assign a score to this question and assign a total weight to the question (total possible score) as well.
4. Save Cancel and View Solution buttons at the very bottom of the screen.

Refer to Figures 22 and 23 below for examples of this screen.



**Figure 22: Scoring Screen**

The background image should be horizontally centered within the canvas drawing area. However, the background image should be vertically placed very close to the top of the canvas drawing area. This placement will allow for commenting and markup around the entire image. The teacher should have access to the standard drawing tools for marking up the image. The database Holding and Transaction collections will likely need to be altered to store these two fields: question score and possible score.The two text boxes should be within a larger grey box and surrounded by the text indicated at the bottom.



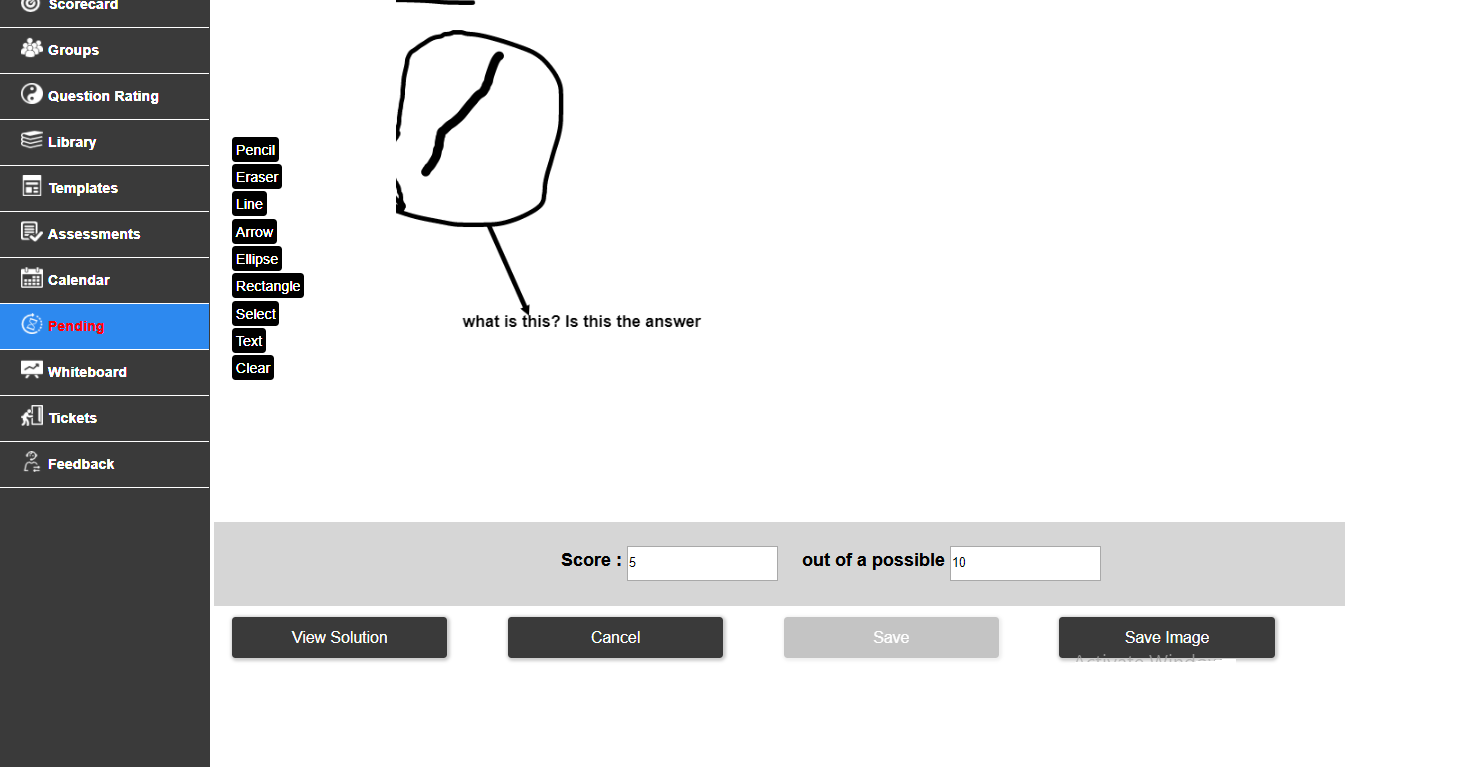
**Figure 23: Scored and Marked Up by the Teacher**

Clicking on the save image button at the bottom of this screen will do several things:

1. Save the combined image. Once the teacher has finished scoring and marking up the student’s response (example above in Figure 23), the new canvas markup and the background image should be saved back to S3. This new combined image should replace the original background image. For example, if the original background image was stored as 1234.png, then the new combined image should overwrite the old image and still be stored as 1234.png.
2. Verify that the question score and possible score have been entered into the provided fields. If it has not, display a short warning that this information is missing and then continue to step 3 below. If it has, then store the scoring information (that is the question score and the total possible score for this question) in the Holding Collection, and continue to step 3.
3. Use the previously stored course, grade, class, test ID and student ID to query the Holding collection for missing question scores AND possible scores for the Open Response question type. If any of this scoring data is missing, then skip to step 7 below in this process. If all of the scoring data exists (for Open Response question types), then the selected student’s test is ready for final calculation.
4. Query the Holding collection for all documents that relate to the selected course, grade, class, student ID and test ID. Use this to calculate a final result document and store it the Result collection.
5. Move all the documents from step 4 into the Transaction collection. These documents should no longer exist in the Holding collection after this.
6. Delete the related record from the Pending collection.
7. Return the user to the initial Pending screen making use of their last selected Course, Grade and Class (as in Figure 16).

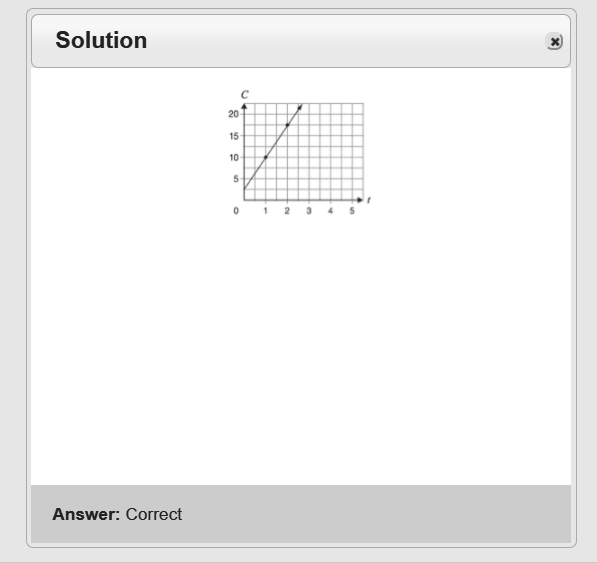
### Viewing The Solution

The “View Solution” button pictured in Figure 24 below is only visible if the selected question has stored answer information (either a text answer and/or an answer image).



**Figure 24: View Solution Button Displayed**

The Question collection will need to be queried to determine the visible state of the View Solutions button. Assuming that this information exists, and that the button is visible, clicking on this button will open a pop up window (very much like the question preview window) showing the answer information; see Figure 25 below.



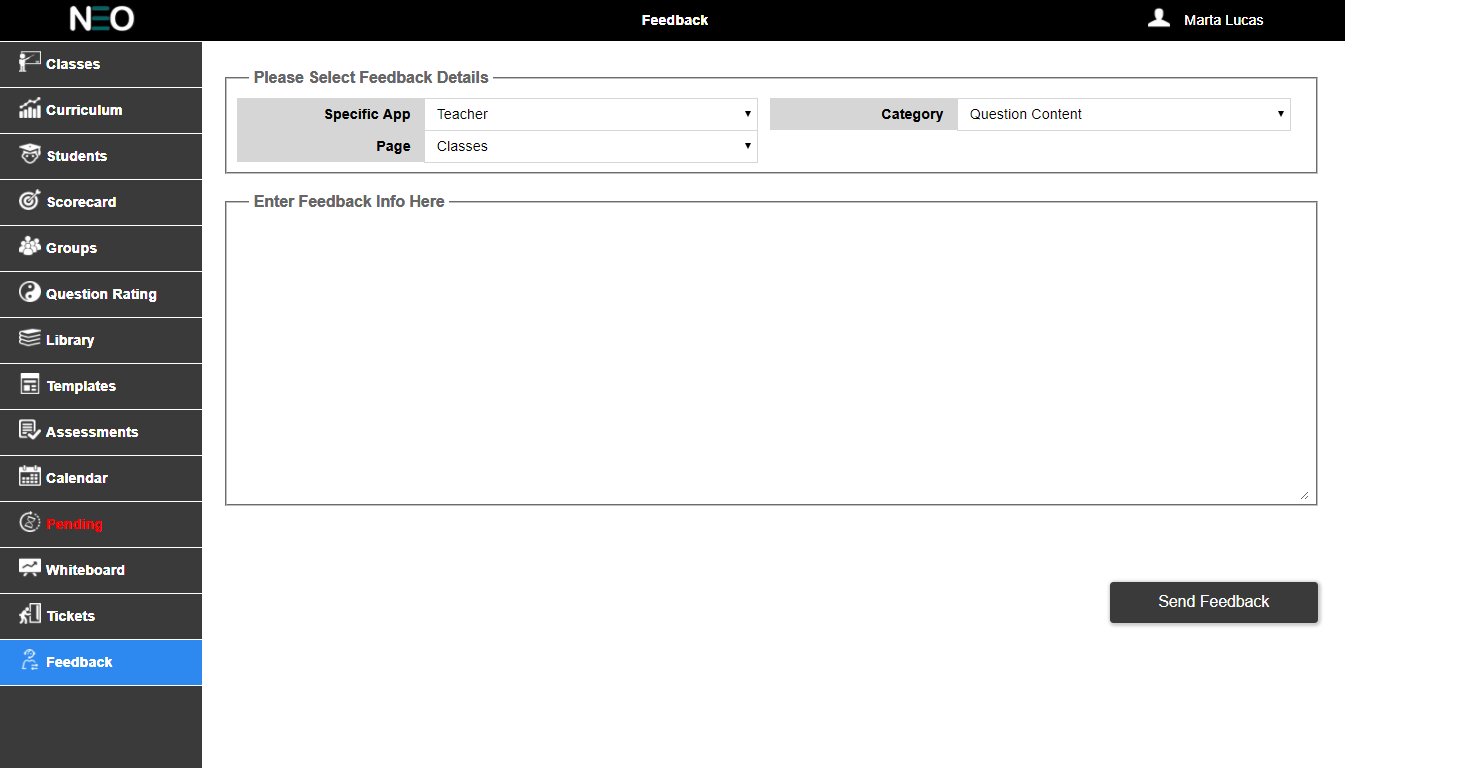
**Figure 25: Solutions Pop Up Window**

The top section of the Solution window should be used for the title of this pop up window. It should be formatted the same as the Question Preview popup windows and should simply read “Solution”.

The middle section of the Solution window (and majority of the window space) should be dedicated to displaying the stored answer image (assuming that one was used). The image should be centered horizontally. The bottom section of this window should show a grey bar (the width of the window) that displays the text of the correct answer (assuming that one was given). Refer to Figure 25 above.

# Feedback

The feedback is used to get the user comments and enhancements for the teacher and student app (see Figure 26).

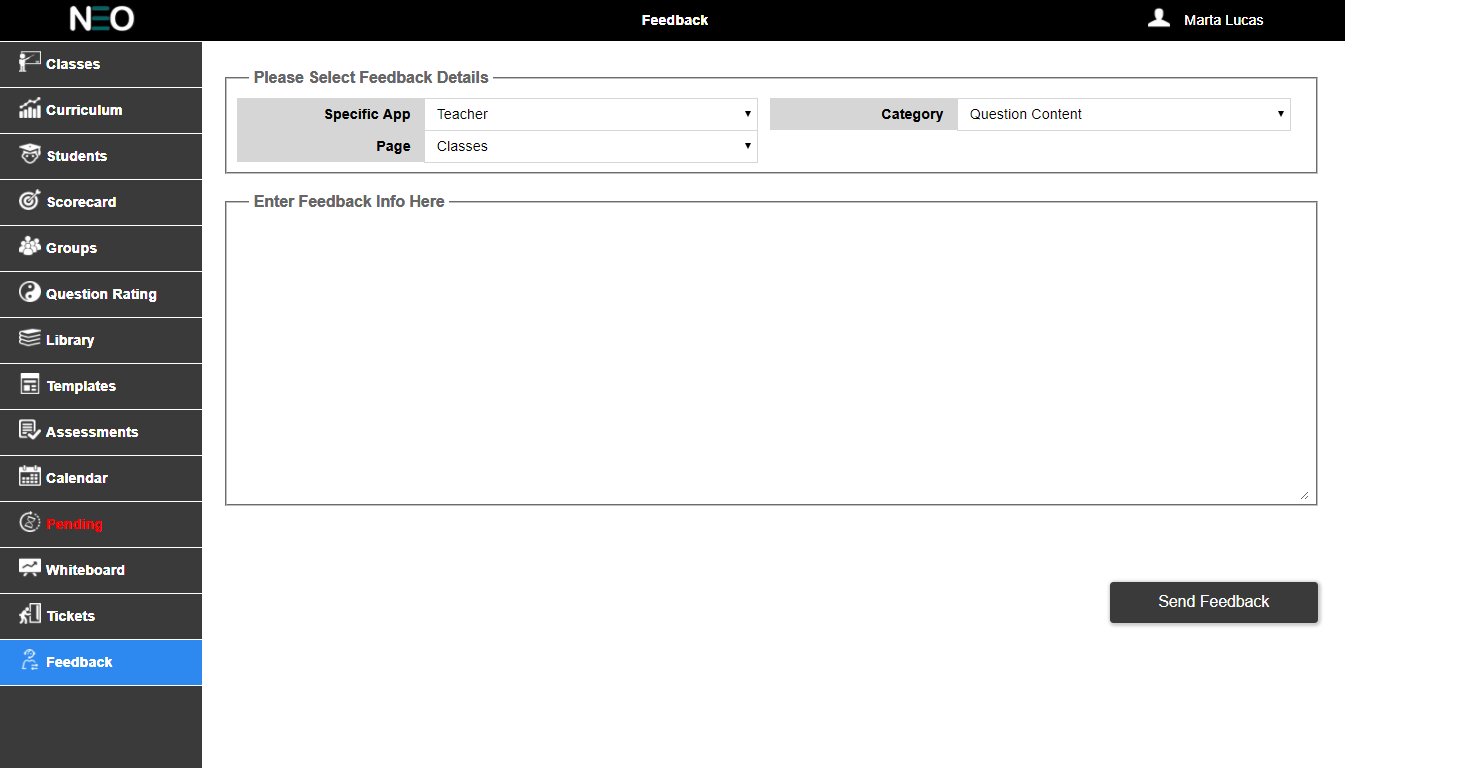


**Figure 26: Feedback Tab**

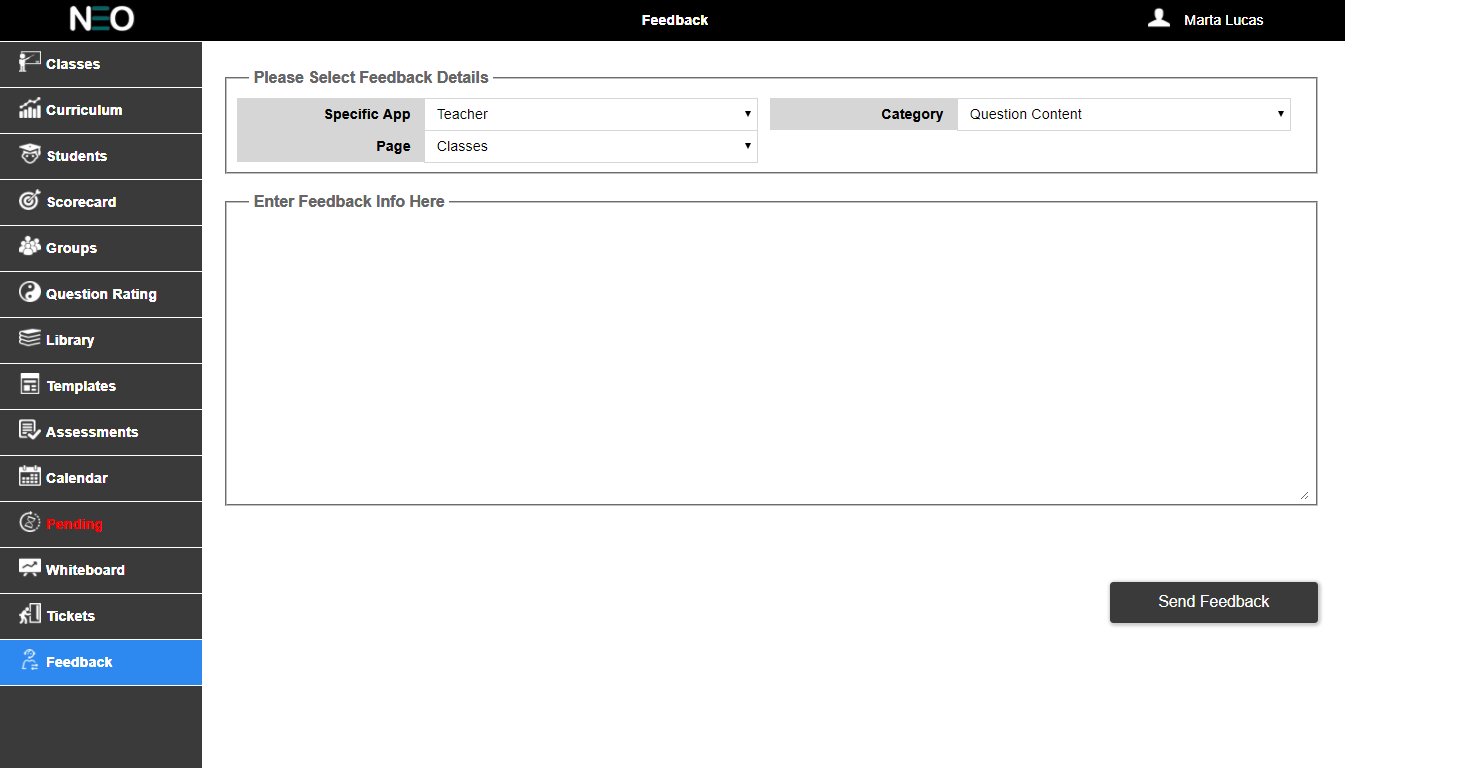
When the user first selects the Feedback tab, the dropdowns will show Feedback details that a user is interested to give (See Figure 27 below). The user has to select for which app the user wants to give feedback, the category of the feedback like general comment or question content. Based on the app the

Page dropdown should be populated. If the user chooses teacher in the Specific App, the Page dropdown should also populate with teacher app pages like classes, students, curriculum etc.

Use better resolution images.



**Figure 27: Selection of Feedback Details Fieldset**

Below Selection of Feedback Details fieldset, there is another fieldset with text area where the user can enter their feedback (see Figure 28). Clicking on the Send Feedback button should send an email message to [nt@neols.com](mailto:nt@neols.com). The Message Title should be: Pilot Feedback. And the text message should automatically contain the following information:User ID, School Name, Board Name, (from the drop lists) Specific App, Page, Category, User's Name, User’s Email Addressand the User's feedback.

**Figure 28: Feedback Text Area**

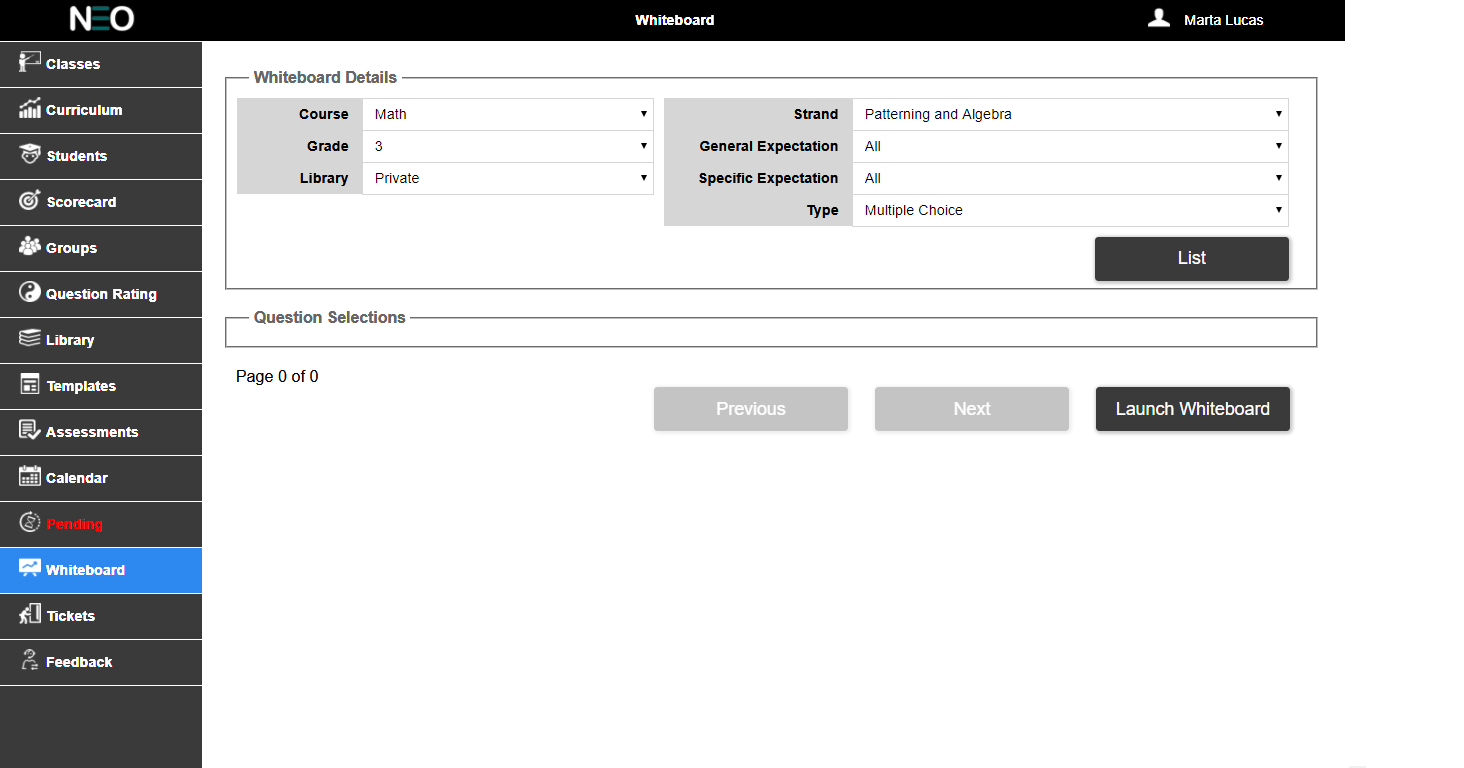
# White Board

This tab details should contain only the changes. And the image added is wrong. The canvas tools have to be aligned correctly

The Whiteboard tab should allow the teacher to select a question and then project that question onto an external screen. In this way they will be able to directly interact with their class when teaching lessons. There are two screens required to make this happen: the question selection screen and the whiteboard screen.

### Question Selection

The question selection screen (picture below in Figure 29) functions as a simplified assessments setup screen (few selections and nothing are ever written to the database).



**Figure 29: Whiteboard Selection Screen**

At the top of this screen, the Whiteboard Details field set should allow the user to select the Course, Grade (both filtered by the User ID) and Library (Public, Private or All); as usual, the course and grade are filtered by logged in User ID. They should also be able to choose the Strand, General Expectation, Specific Expectation, and the question type; the default selection for all of these should be “All”.

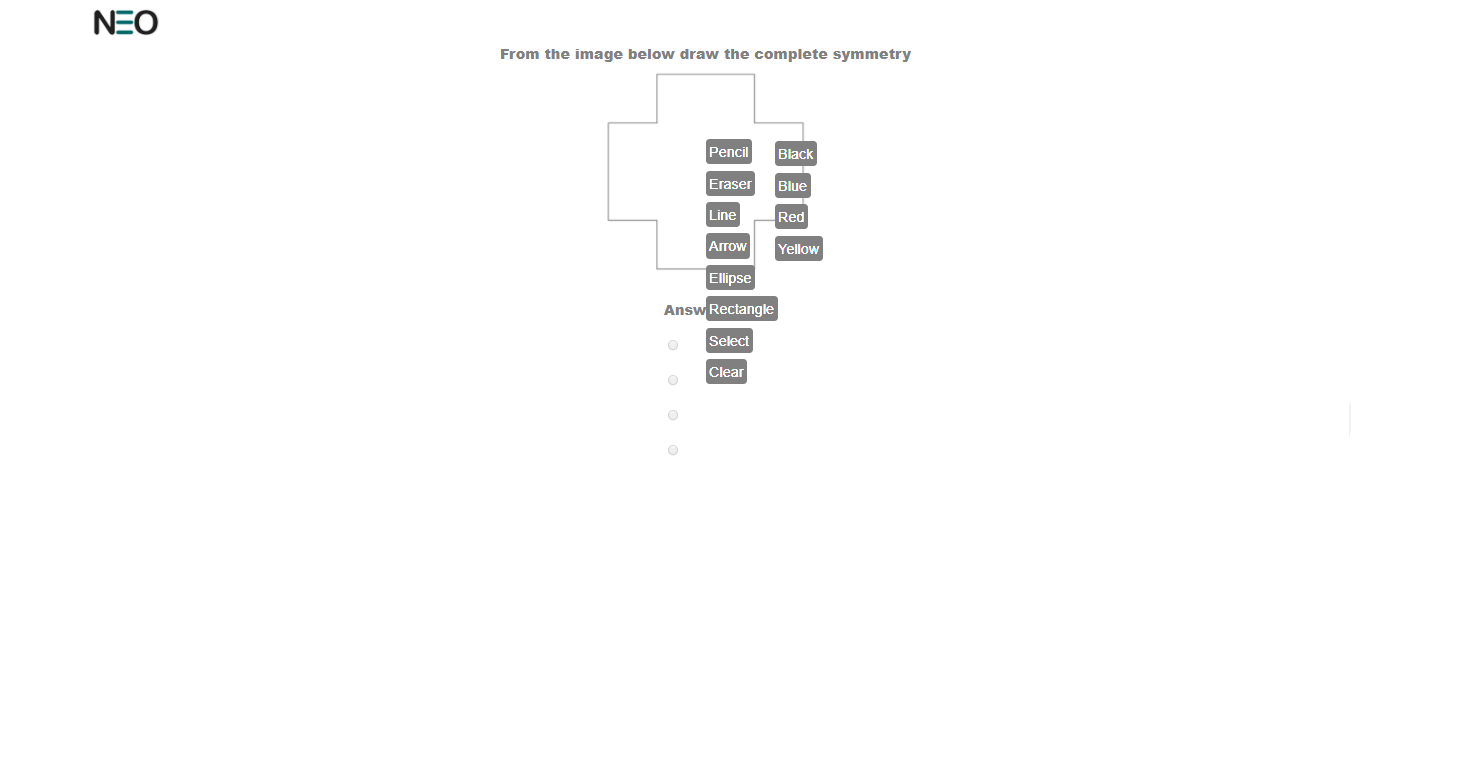
The Question Selections field set should display a filtered list of questions (in blocks of 10) using the above selections from the Whiteboard Details. This displayed list should show Line 1 of the question, the question ID, the question type and a check box.

User can only select a single question for display on the whiteboard; please review how the demo functions. Rolling the mouse pointer over a question ID should highlight this field. Clicking on the question ID should show a preview of this question.

At the bottom of this screen is a button to “Launch Whiteboard” which will take the user to another screen which displays the selected question.

### Whiteborad

The Whiteboard is very much like the question preview except that it is full screen without any of the surrounding menus (see Figure 30).



**Figure 30: Whiteboard Screen**

The whiteboard will also make use of the Literally Canvas software so that the teacher can draw all over the question while teaching the lesson. To accomplish this, the selected question should either be the background image for the Literally Canvas element, or HTML layers should be used so that the question is the bottom layer while the Literally Canvas element is the top layer. The Literally Canvas tool bar should be set along the right side of the window. Clicking on the Neo logo in the top left corner of the screen should return the user to the prior screen.

# Color Palette

Use the following chart as a guide to the color of UI elements. Individual curriculum strand color should follow the colors listed in the chart below and 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 |  |

UI colors:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |  |
| 80AAFF | 128 | 170 | 255 |  |
| B3DBFF | 179 | 219 | 255 |  |
| 0055FF | 0 | 85 | 255 |  |
| E60000 | 230 | 0 | 0 |  |

# Figure Guide

|  |  |
| --- | --- |
| **Figure Number** | **Image Description** |
|  | Calendar |
|  | Calendar preview |
|  | Library tab |
|  | Library - question tags |
|  | Library - text and images |
|  | Library - answer - multiple choice |
|  | Library - answer - text input |
|  | Library - answer - open response |
|  | Library - Buttons |
|  | Exit ticket tab |
|  | Details selections |
|  | Filter selections |
|  | Question selections |
|  | Selected questions |
|  | Exit ticket - Review |
|  | Pending tab |
|  | Pending tab menu |
|  | Pending tab- top filters |
|  | Pending Test |
|  | Students in selected assessment |
|  | Unmarked questions |
|  | Scoring Screen |
|  | Scored and marked up by the teacher |
|  | View solution button |
|  | Solution pop up window |
|  | Feedback |
|  | Selection of feedback details |
|  | Feedback text area |
|  | Whiteboard Selection screen |
|  | Whiteboard Screen |

# Non Disclosure

All of the information within this document is considered to be confidential and the property of NeoLearning 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.