Documentation for OSCE/DOPS examination on Windows 8 application.

Step by step guide on how to create exam forms on Moodle and how to download them on your application.

Pantelis Georgiades

OSCE/DOPS Documentation

Contents

[Moodle exam forms 1](#_Toc426925325)

[Create a form template 1](#_Toc426925326)

[Create exam-forms 2](#_Toc426925327)

[Total mark 5](#_Toc426925328)

[Run out of time 5](#_Toc426925329)

[Feedback Comments: 6](#_Toc426925330)

[User Environments 9](#_Toc426925331)

[Normal user: 9](#_Toc426925332)

[Admin: 12](#_Toc426925333)

[Student file (txt OR csv file): 14](#_Toc426925334)

[Download form from Moodle 19](#_Toc426925335)

[Edit form 23](#_Toc426925336)

[Api calls 27](#_Toc426925337)

[Gets all categories from Moodle 27](#_Toc426925338)

[Folders of the exams in Moodle 29](#_Toc426925339)

[Exams in the particular 'id' folder. 30](#_Toc426925340)

[Get the particular form on the tablet – Get the fields for that form 32](#_Toc426925341)

[Gets the form's feedback. 39](#_Toc426925342)

# Moodle exam forms

## Create a form template

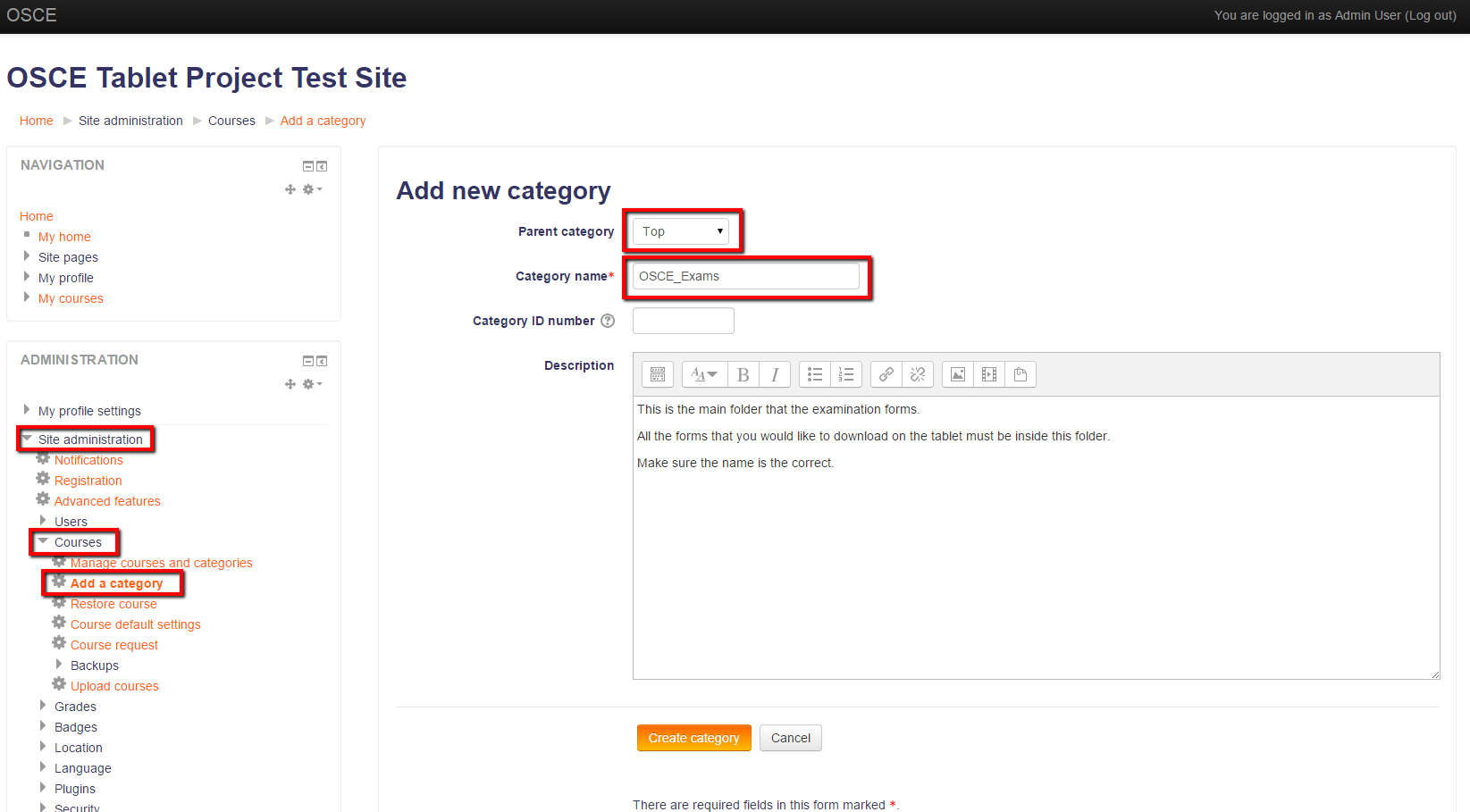
Log in as administrator in Moodle and make sure you‘ve clicked on ‘Turn editing on’. Go to Site administration -> Courses -> Add a category. Parent category should be ‘Top’, Category Name should be ‘OSCE\_Exams’, category id number can be blank and you can give a description and click ‘create course’.

Image 1: Add new category.

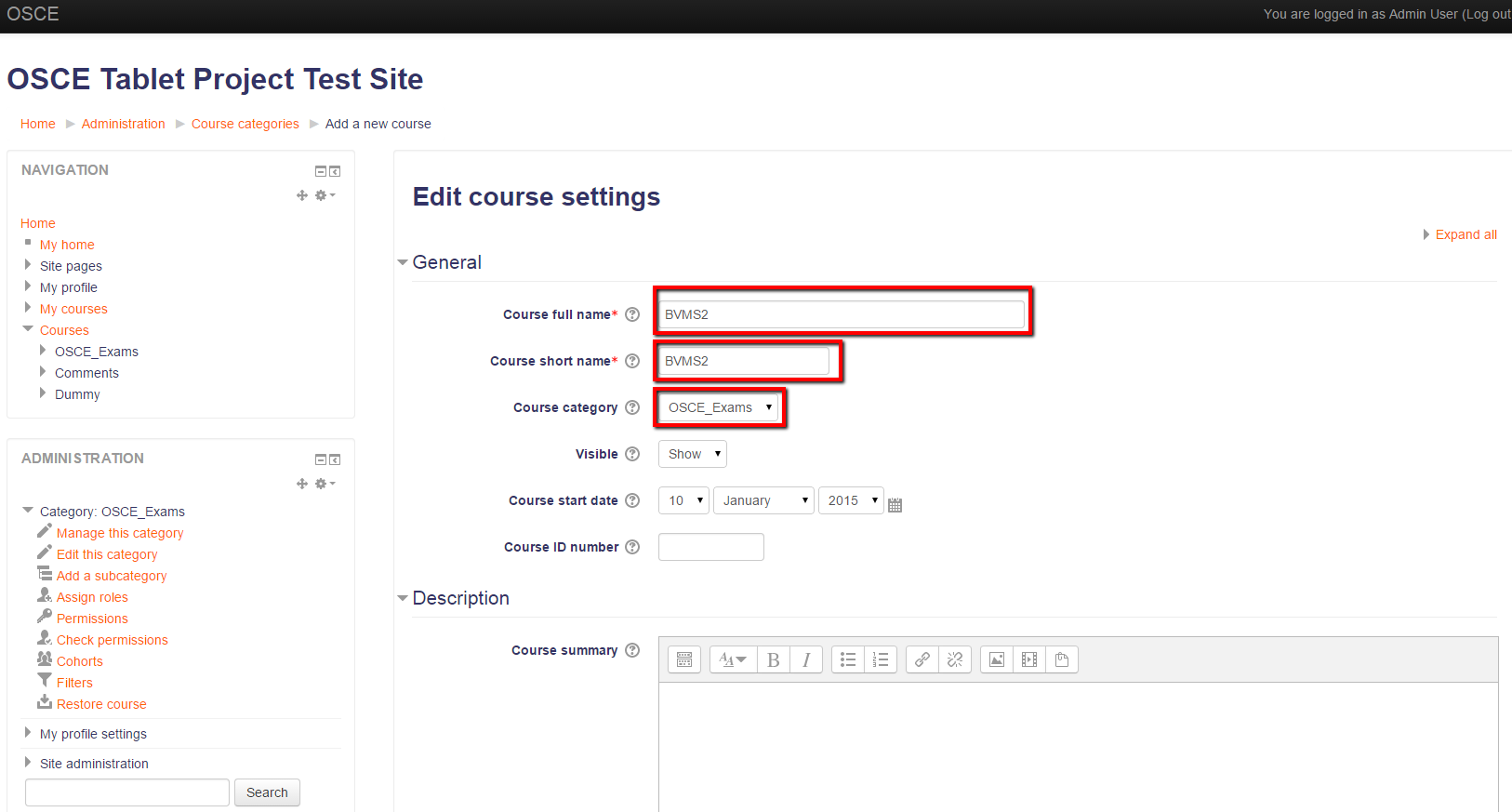
It will navigate you to ‘Course and category management’ and there you can create your new course folders.

Image 2: Add new template - settings

You can enroll teacher users for the specific course. After you are done with enrolling users you can navigate to your course and rename your sections from ‘Topic #’ to whatever makes sense and start creating your exam forms. It is not necessary to enroll students with the current setup (it was designed with the intention that students would not be enrolled on the courses).

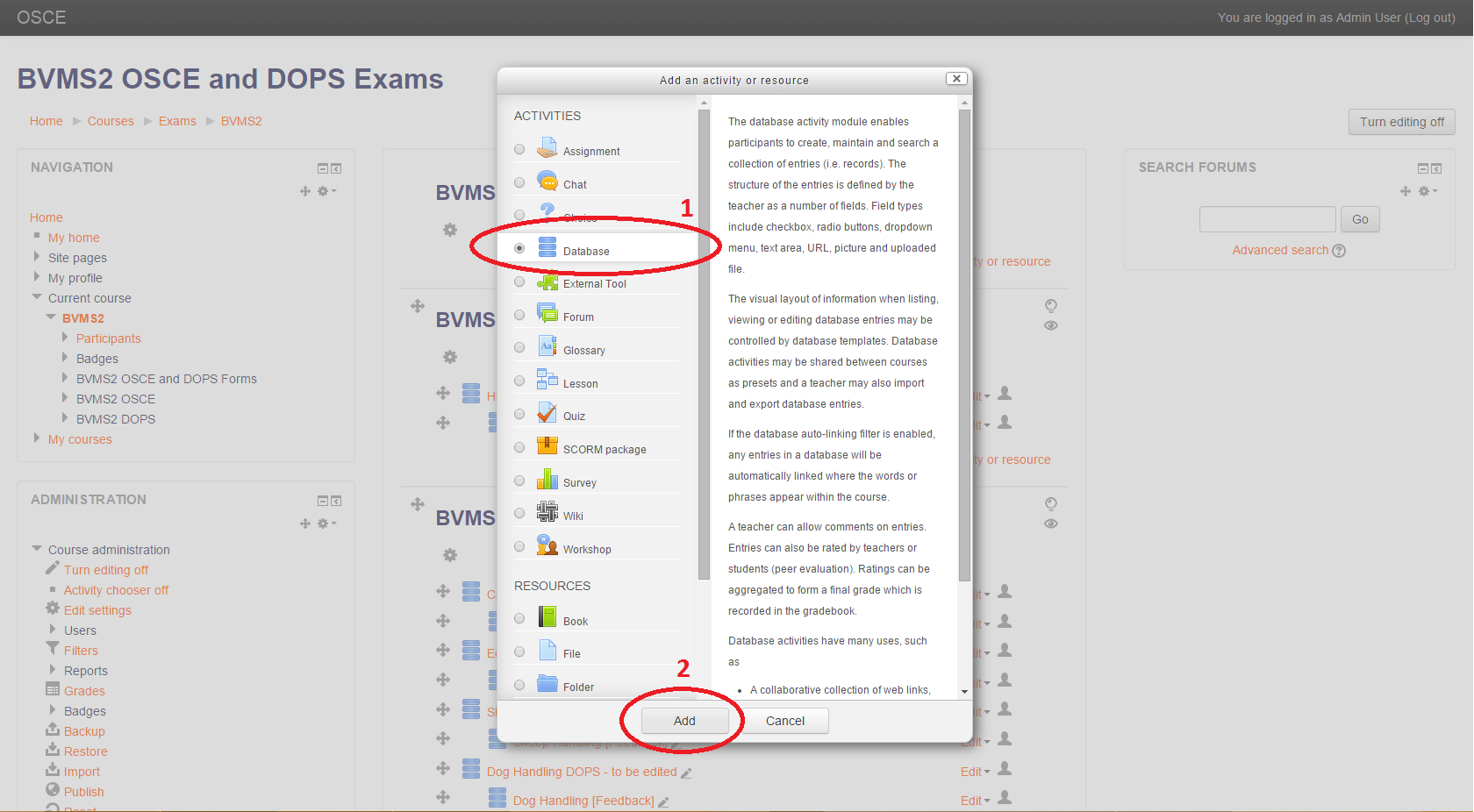
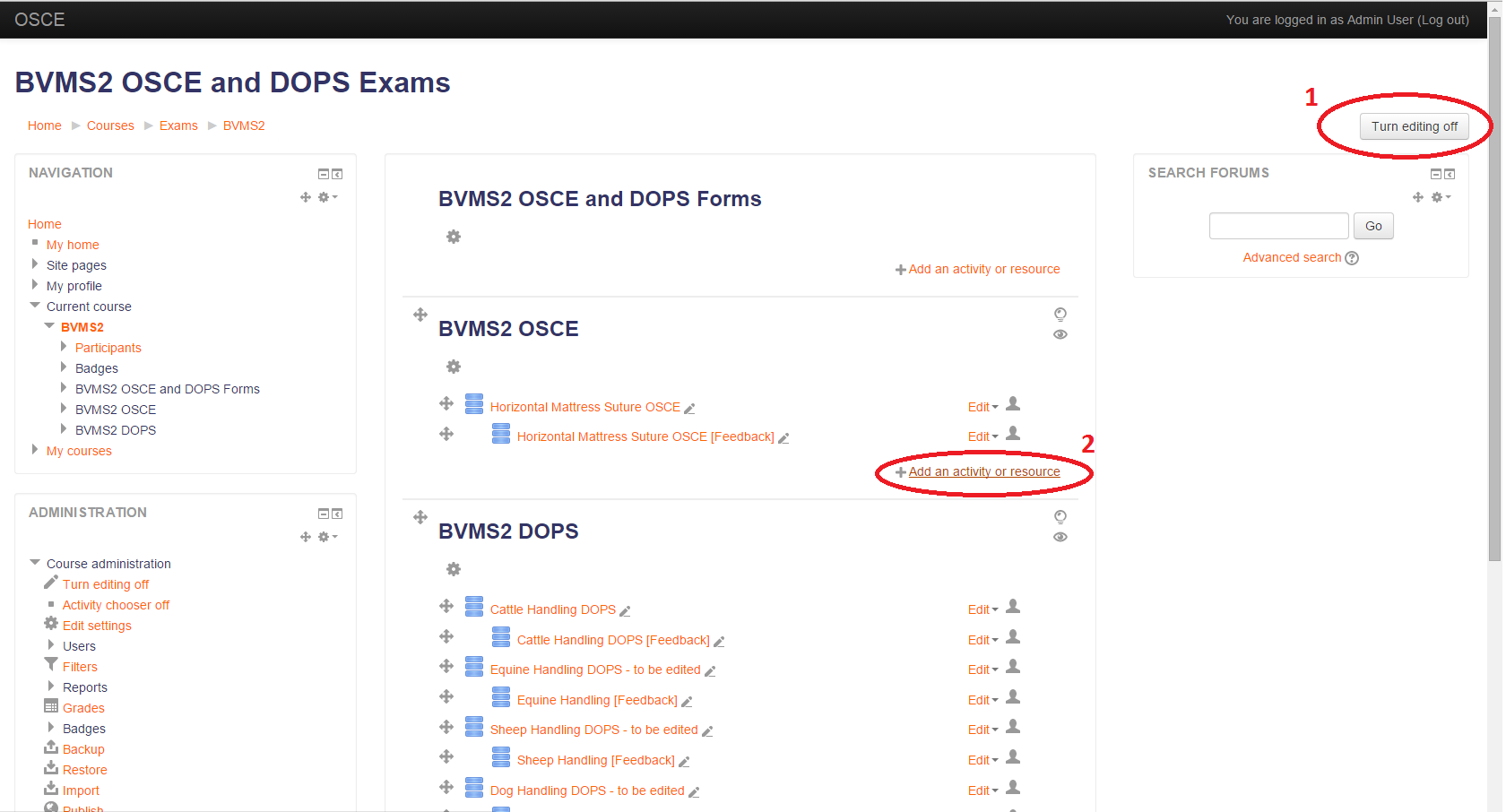
Image 3: Add new template - settings

## Create exam-forms

An example Moodle database activity is included on the GitHub site which can be restored to a Moodle site, providing a template sample activity, however you can create new exam forms (database activities) from scratch. Click on the 'Add an activity or resource', tick on the Database option and click 'Add'.

Complete the new exam's name and description and hit 'save and display'.

Image 4: Create new exam-form (database activity) on Moodle



It will automatically navigate you to the fields' tab of the new exam you have created. You can start adding fields. It is advised to make the form's layout in a different editor (for html) and copy paste it to the 'Template' tab under 'Add Template'. The ‘Add Template’ provides the layout of the form that will be populated on the app.

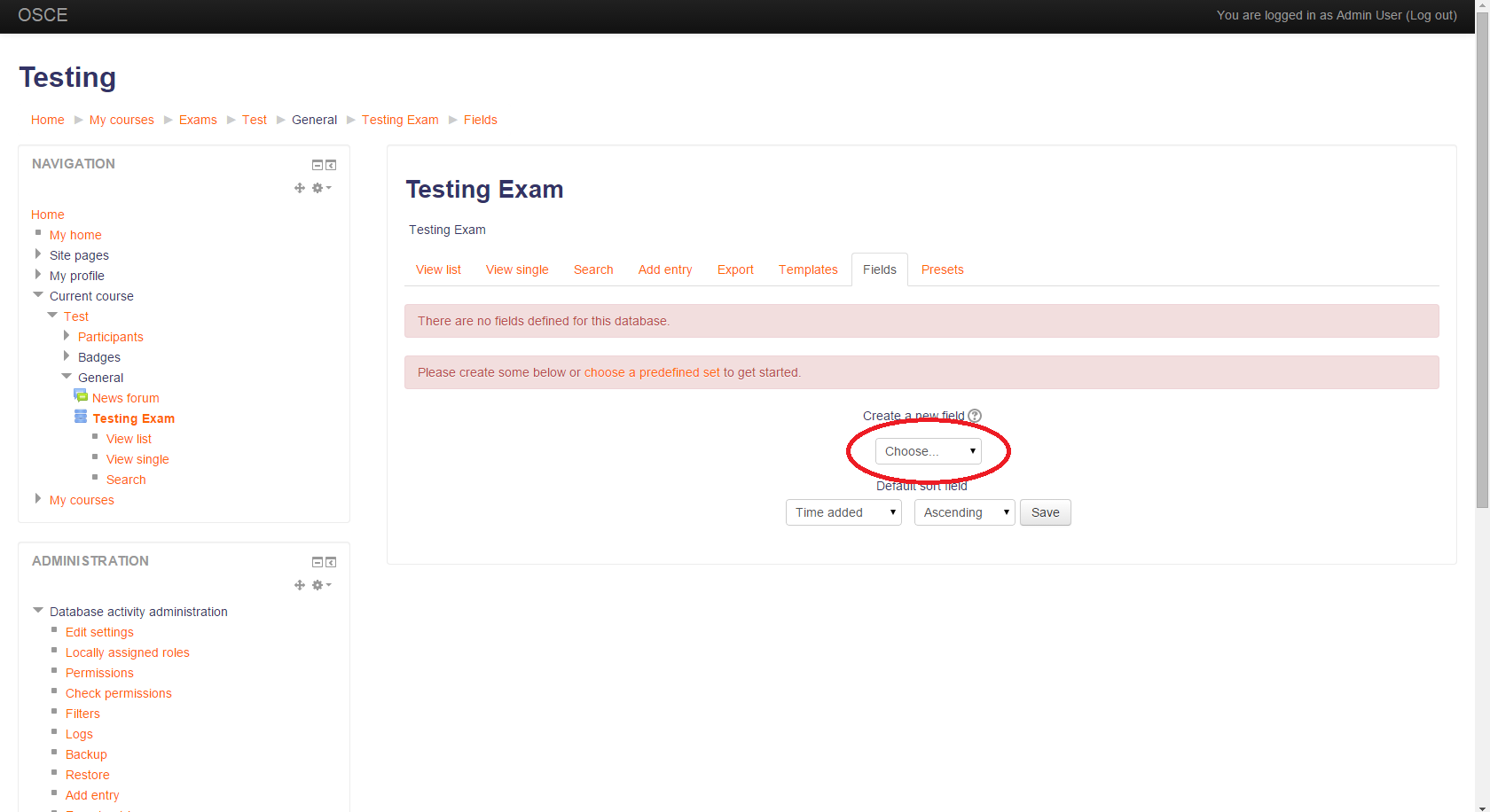
It is recommended that the structure of the form is written in html code.

Image 5: Create new exam-form on Moodle: Add fields

There are some fields that are currently hardcoded in order to behave with the appropriate functionality. For example:

* The 'student name' field must be 'Name' and must be a text field.
* The 'student id' field must be 'Matric' and must be a text field.
* Dropdown menus names can be like the previous format. E.g. ‘Year’, ‘Assessor’.
* The rest of the field’s names can be '#(number)' e.g '#1'. (radio button and textareas). The names of these fields require to have hash at the beginning.

These forms are designed to have multiple radio buttons with some numerical values. Then the examiner will tick the desired radio button (points) while assessing the student.

**Reminder: The form’s templates must be written in HTML. The forms are designed to have multiple radio buttons with some numerical values.**

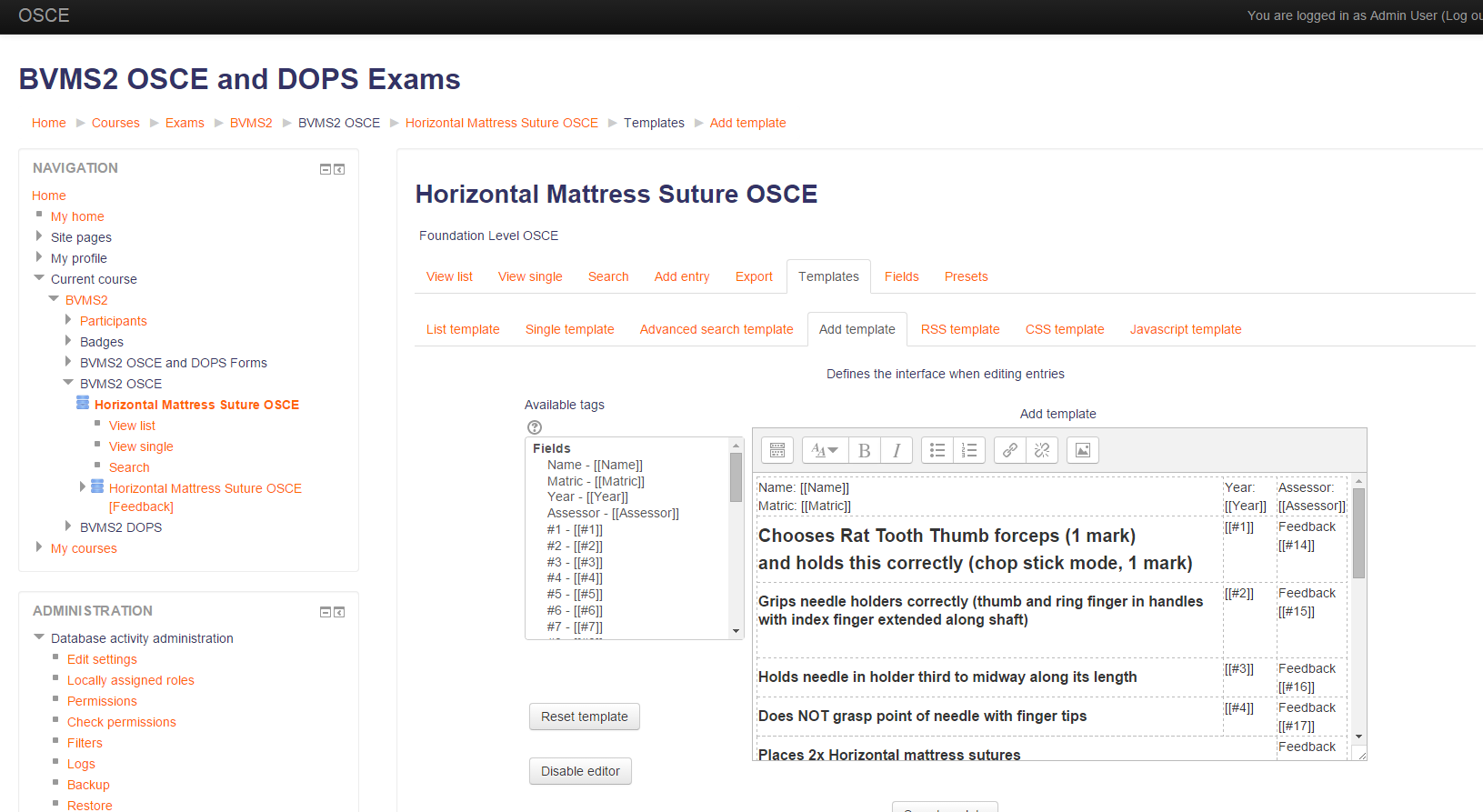


Image 7: Paste your raw HTML code of your form and place your field names where you want.

Image 6: Create new field – radio button

When the fields are created navigate to the ‘Add Template’ to preview the form. It gives you a default template for your form but you can override it with the one you have created. Click on the ‘Disable editor’ button and paste your raw HTML code of your form into the textarea. Click on the button again and you will see your form. You can add your fields inside your form by just adding the name of the field in the appropriate place (see image 7).

### Total Mark

Total mark on the form: If you would like to have an automated calculation of the total mark for your forms then you must create a new text field and name it ‘TotalMark’. Do not add it to your ‘Add template’. When the form is downloaded on your tablet then the total mark of the form will be updated as the examiner ticks the corresponding radio buttons with the numeric values. In order for this functionality to work properly your radio buttons must consist of numeric values.

If your form does not consist of numeric evaluation and it is only yes or no options then do not create a ‘TotalMark’ text field on the Moodle exam.

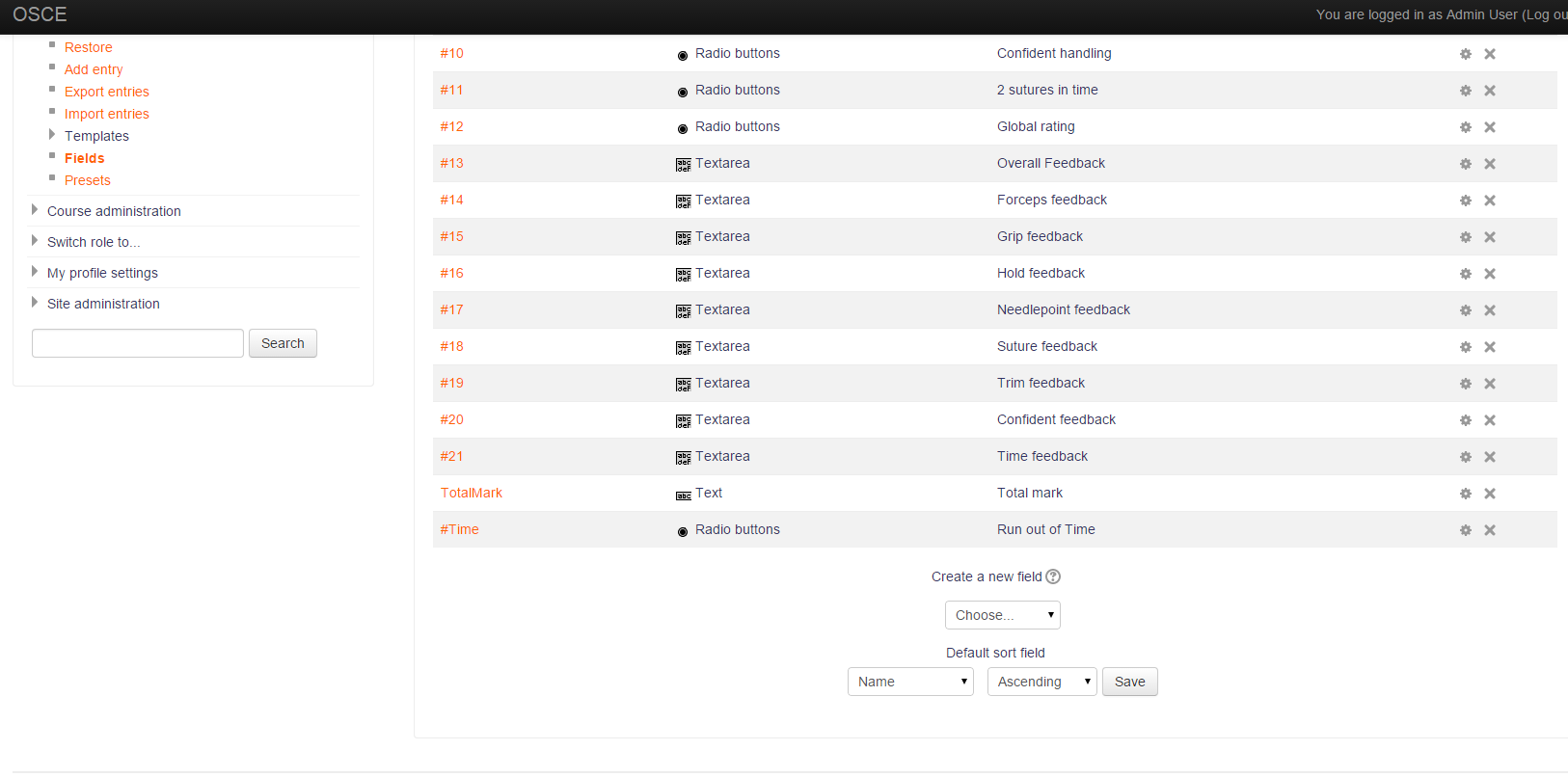


Image 8: Total Mark field

**Reminder: Create a ‘TotalMark’ field in the database but DO NOT add it to ‘Add template’.**

### Run out of time

Run out of time: Radio button, name: ‘#Time’, options: yes/no. This special radio button can be created and it will have as a default value the ‘no’ option.

While creating the templates in Moodle keep in mind to avoid the following characters inside the text (instructions, details and what so ever):

" (double quote)

\_ (underscore)

/ slash

\ slash

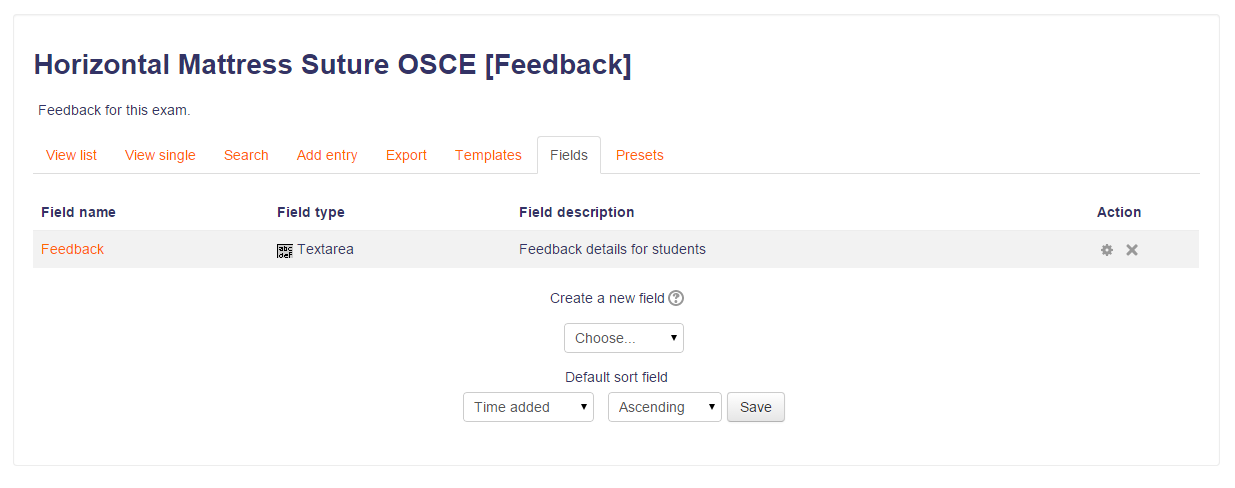
| vertical line

**Reminder: Please make sure that the 'Add template' is correct. Make sure that you are not using any of the 'invalid' characters.**

## Feedback Comments:

After finishing with your form you can create the default feedback comments for that exam. Create a new activity (same as [before](#_Create_exam-forms:)) and the name of the feedback activity should be the same as the exam and add '[Feedback]' at the end of the name. Example:

If the name of the exam (activity/resource) is 'Cow Anatomy 1' then the feedback (activity/resource) should be 'Cow Anatomy 1 [Feedback]'

Feedback templates should be only one line (no new lines) because it will be visible on a dropdown menu inside the forms (on the tablet). (image 9,10,11 and 30)

**Reminder: You must also keep in mind to avoid the above ‘invalid’ characters with the feedback comments.**

Image 9: Feedback main field

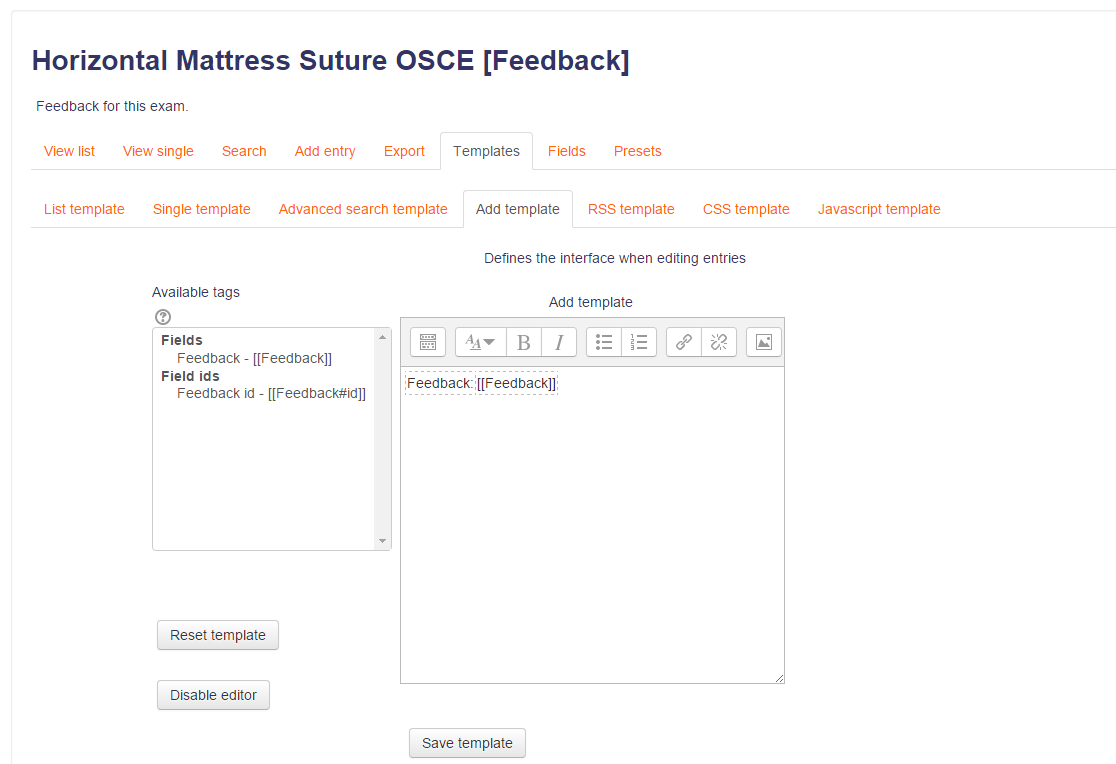
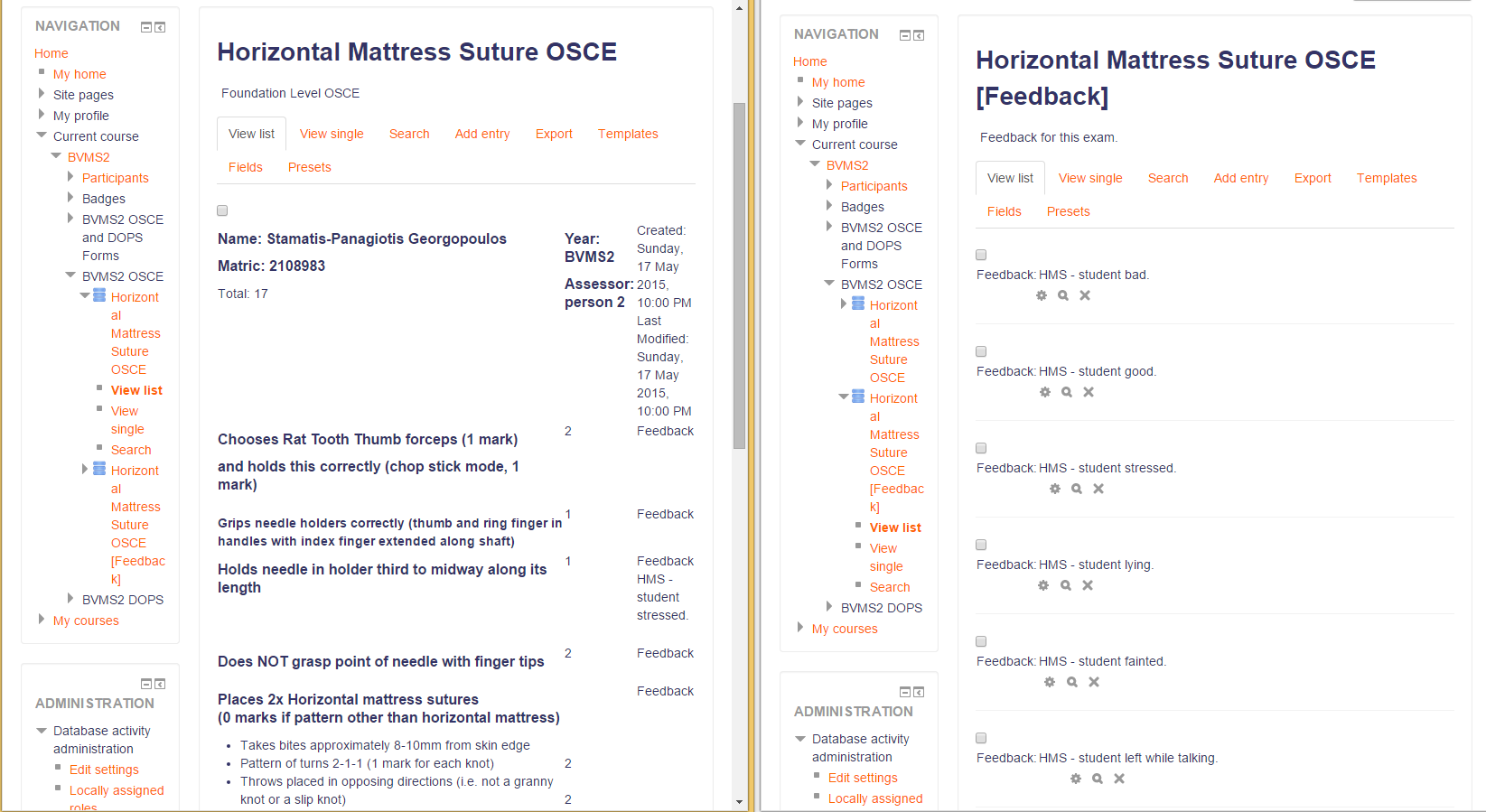


Image 11: Feedback entries by the user in Moodle

Image 10: Feedback template structure.

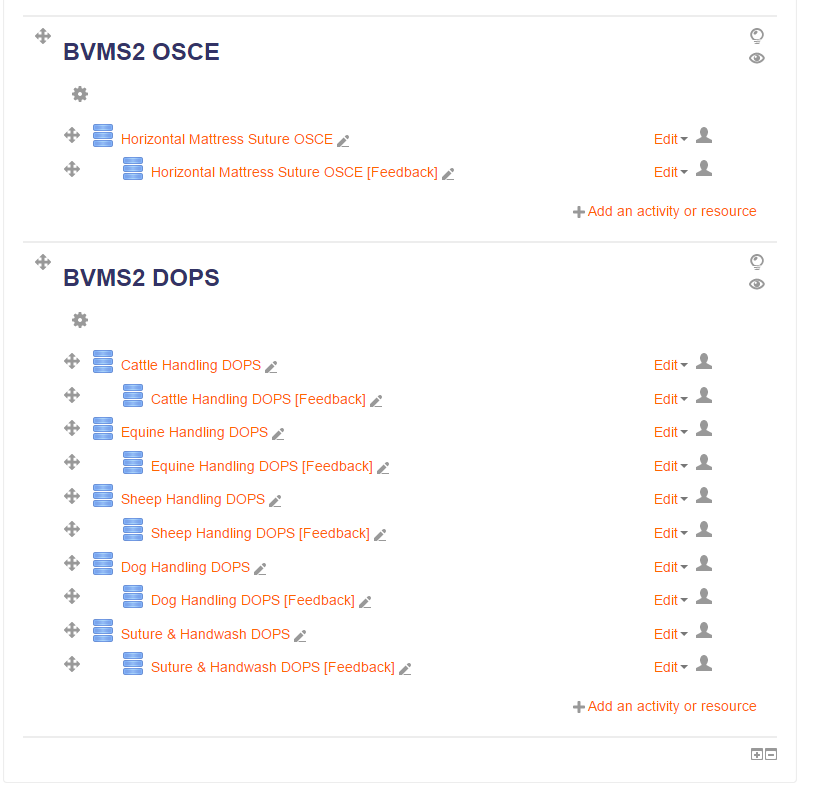
When you are done adding your courses and the corresponding feedback comments your folder should look similar to this:

Image 12: Your Moodle course after adding all your exam database activities and their corresponding feedback.

# User Environments

There are 3 user environments for the application.

## Normal user:

On the main page the user can see 3 options.

* Download forms from Moodle. The user can provide his/hers Moodle password and can choose and download an exam form on the tablet. The user can type the number of times that the form will be populated on the tablet. The username of the user has to be the same as the Moodle username. Some configurations must set up before anyone can download forms on the tablet. Details will be on the admin environment.
* Exam Repository: Can browse through all of the already downloaded forms on the tablet and download them to his/hers account. The user can preview the exam forms before downloading. When the user is sure of what forms he/she needs, he/she taps on the exam and a new list will appear on the right side (or bottom half of the screen if the tablet is held vertically) and can then type the number of the forms to be downloaded on the tablet.
* My Forms (Downloaded): The user can browse through the exam forms that are already downloaded on his/hers account.

Image 13: Exams Repository. When you click on an exam then another table is shown with the users that are registered on the tablet. You can populate the selected exam on the users.

* + Delete the whole folder with all the exam forms that are currently inside.
  + Upload to Moodle. This action requires the user’s Moodle password. Only the completed forms will be uploaded on Moodle.
  + Browse the forms of the particular exam folder:
    - The user can add more forms either by clicking the ‘New Forms(s)’ button multiple times or type a number and then click the button.
    - Filter the forms by his/hers preferences.
    - Sort the forms by the student’s ID or Name.
    - Delete the form.
    - ‘Flag’ the form for future reference. The user can “tick/untick” the form.
    - Edit form (See ‘Edit Form’ to see images):

If the format of the form (on Moodle) consists of a total mark then the total sum will be calculated and be visible at the top of the page for the examiner. The total is updated while the examiner is assessing the student – while the radio buttons of the corresponding markings are being ‘ticked’ (OSCE). If the format of the form does not have a total mark then the sum will not be visible (DOPS).

* Search for a student. The user can search for a student by either start typing the name or the id of the student. A dropdown list will appear with a typeahead search functionality throughout all of the students that are currently uploaded on the applications database. When you choose the student you want then the name, ID, year and assessor of that student will be populated automatically. That is if the appropriate data exist in the student text file that was uploaded to the application prior the examination process. (The administrator will be responsible for doing that)
* See the Name ID and total marking of the student.
* Can scan the student’s barcode ID and the name will be populated automatically (among will the rest of the information).
* Can full screen the form.
* Increase/Decrease the size of the text.
* When the user saves the form the incomplete parts will get a red background.
* Can add prepopulated feedback on the form instead of typing while the examination is happening.
* Can make a recording of the form and it will be saved on the tablet.
* The back button also saves the form.

When the examiner saves a form the incomplete parts display a red background. That way the user will know which parts of the form are incomplete and focus on them. (form\_edit4)

If the user does not full complete a form and returns to the form list then the corresponding form will have a red background. If a form is fully completed then it will have a green background. Otherwise if it is empty it will have a grey background color. (forms2)

In the ‘My Forms’ folder the user can also observe that if there are any completed forms then it will have a green background as well. Otherwise it has a white background. (myFolder2)

There are some more options that are common for all users. When the user swipes from top to middle (just a few centimeters not all the way, same when you have a mouse connected to the tablet and right click anywhere on the screen) then the navigation app menu will show up. There you can see the following options:

* Log out.
* Log/History: The user can browse through his/hers logs and also filter.

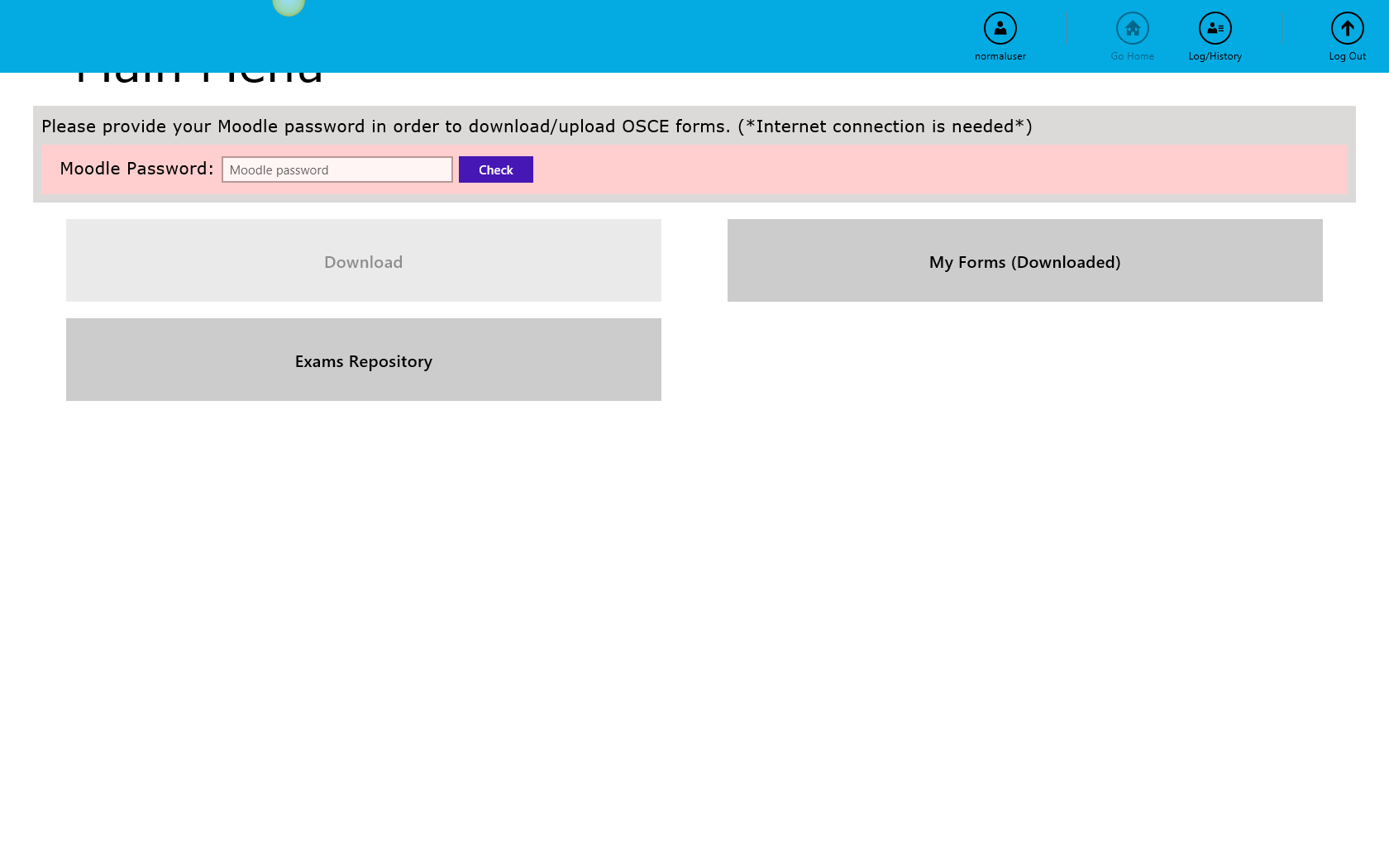


Image 14: Navigation app menu. Swipe from top to middle (just a few cm – same when a connected mouse is right clicked)

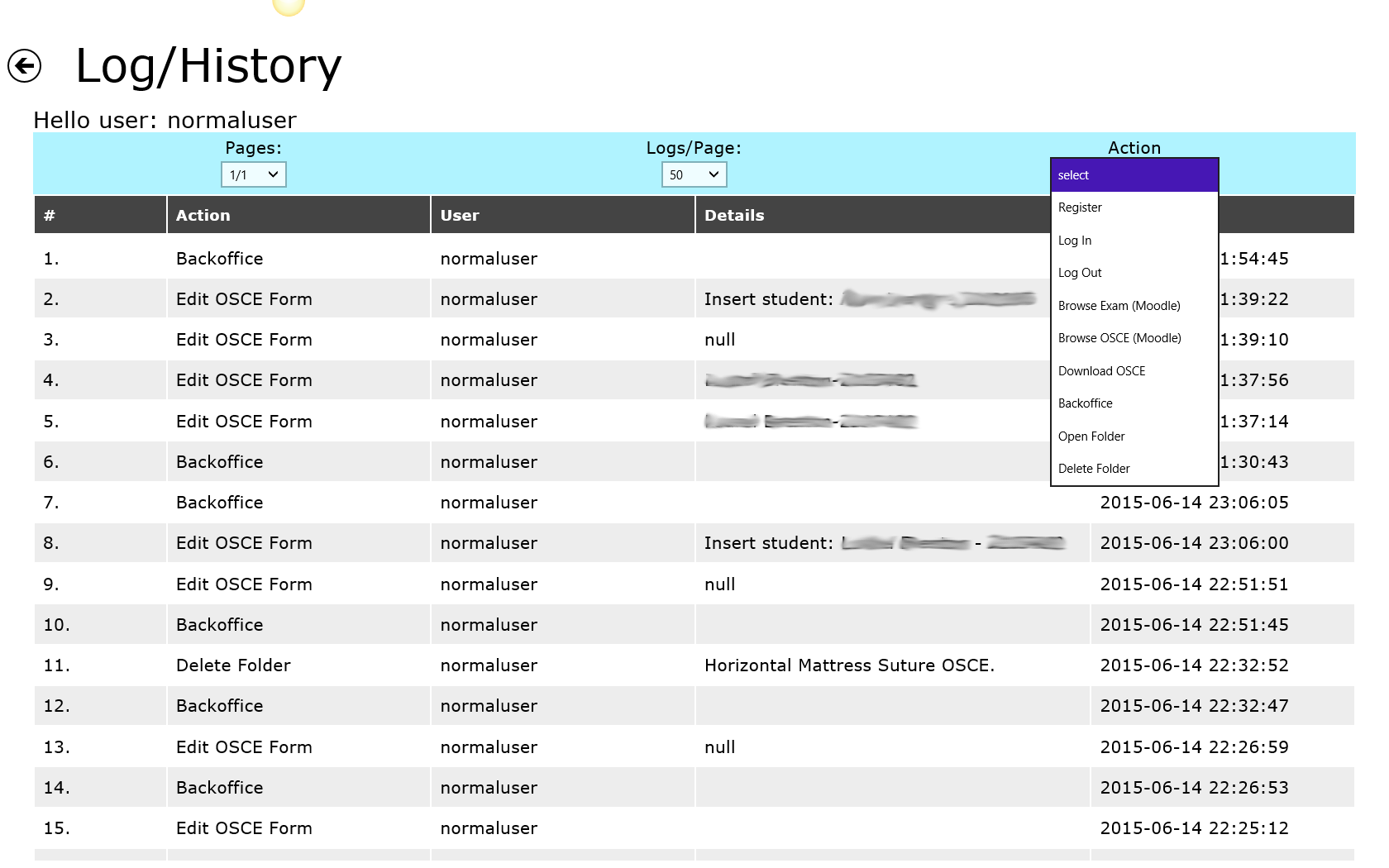
* Home: Navigate to home screen.

Image 15: You can browse the logs and filter them by the actions.

* Profile: Navigates to the user’s profile page where the user can change password, secret word and also set a default green/red color (error/success) that will be appearing throughout the application. (need images for profile)

## Admin:

The admin user has same functionalities as the normal user and more.

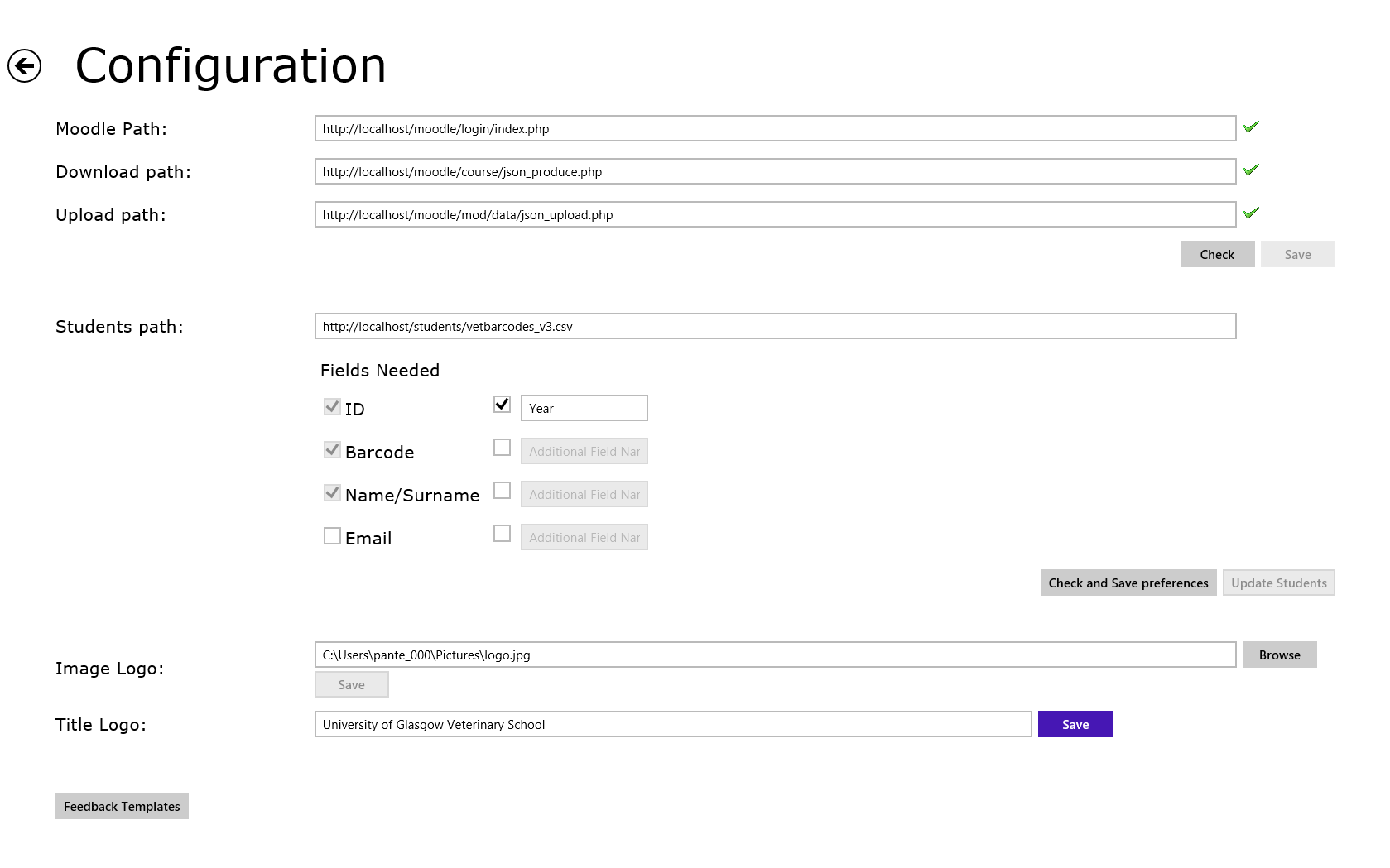
* Users: The admin can create, edit and delete users.
  + New User: The admin can create a new ‘User’, ‘Administrative Staff’ or ‘Administrator’.
  + Edit user: The administrator can change the user’s password, secret word and assign a different role to him/her. The administrator can also see the user’s exams. Can edit, delete and create new forms.
  + Delete user: The admin can delete a user.
* Configuration: The administrator must set these settings in order for the application to work properly.
  + Moodle url path: this is your moodle’s url login page.

Image 16: Configuration page.

* + Download url path: this is where the application will obtain the forms. The application will request the data for all the available folders, exams and forms that exists in Moodle. (show image with the link and data)
  + Upload url path: this is where the application will send the forms in order to be saved on moodle. The forms are send via ajax request to the api using a json object.
  + Student’s text file path: this is the path for the student’s text file (or csv) that can be located somewhere on your server or on the application. If it is located on your server make sure that it is on a location that it doesn’t require for the user to authenticate. There are some images in the next chapter.

Optional settings:

* + Image logo: you set an image for your login page. It cannot be an online image because the application will not have access to the internet most of the times. Download the image and place it in the following directory: “C:\Users\’user’\Pictures”.
  + Title logo: Set the title for the login page.
  + Feedback templates: The idea of feedback templates is that there would be a number of default feedback stored in the Moodle DB against each exam. The examiner will save time during the assessment by adding some of the default feedback. When an exam is downloaded on the application the corresponding feedback list will be downloaded as well. The administrator will be responsible of populating the feedback list prior the examination process, on Moodle.

Moodle:

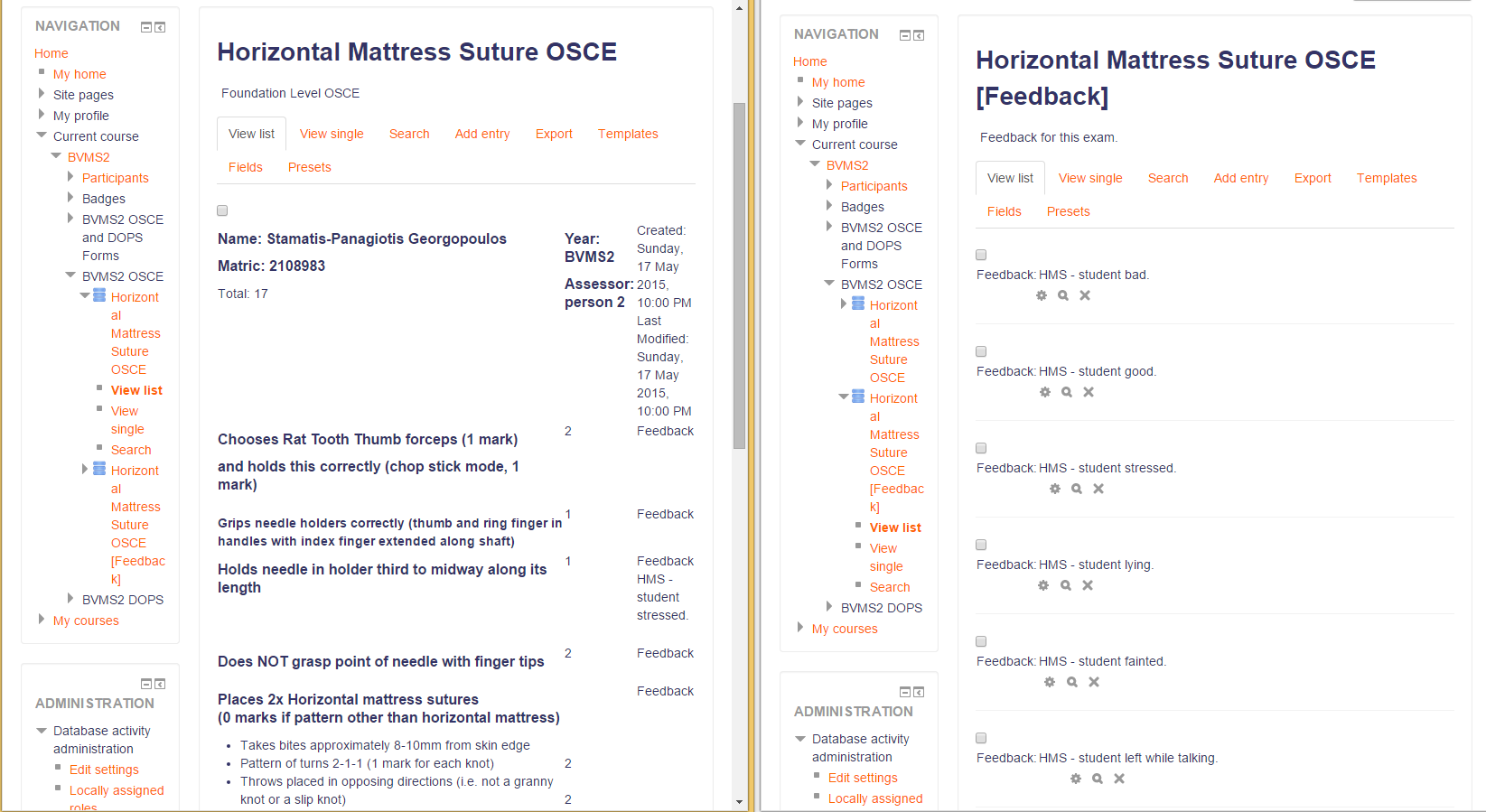
* + - The list must be populated before the exam is downloaded on the tablet.
    - The feedback folder must have the same name with the exam and ‘[Feedback]’ at the end of the name.
* Logs: The administrator can see the logs of all the registered users on the application.

Image 17: Feedback entries by the user in Moodle

# Student file (txt OR csv file):

This file must be placed somewhere on your server. It is recommended to be somewhere that it will not need authentication. Inside the file you can see that the first line is the column headers and the values are comma separated. Each row is a student. This is easy to produce from the database (if you have access to the student list or it is recommended to ask someone from IT to create this file for you). The app includes a feature to read barcodes, however after piloting it was found that typing the first few letter of the student name (or student Id number) was faster.

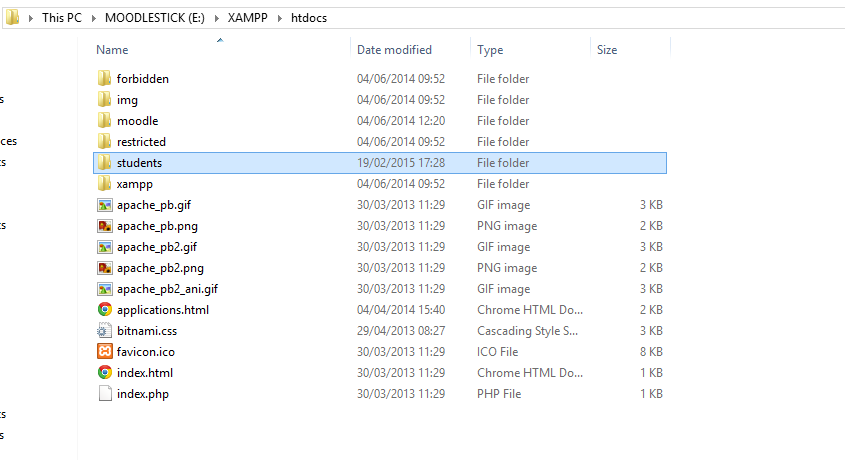


Image 18: Student file on your server

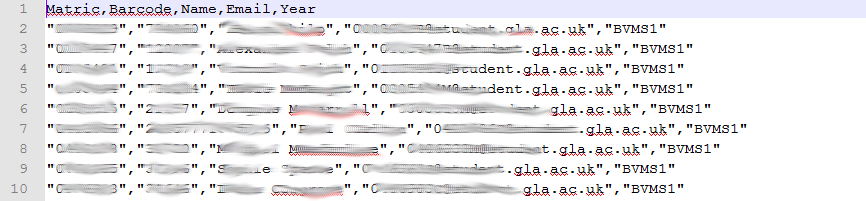
Inside the application’s configuration you need to specify the location of the file and tick the columns that you would like to have for your students. Id, barcode and name/surname are mandatory. The rest you can configure and set some other custom columns that you have on the file like year or email. Click “check and save preferences” and the according prompt will pop up at the top right corner of the screen.

Image 19: Student file. It can be either TXT or CSV

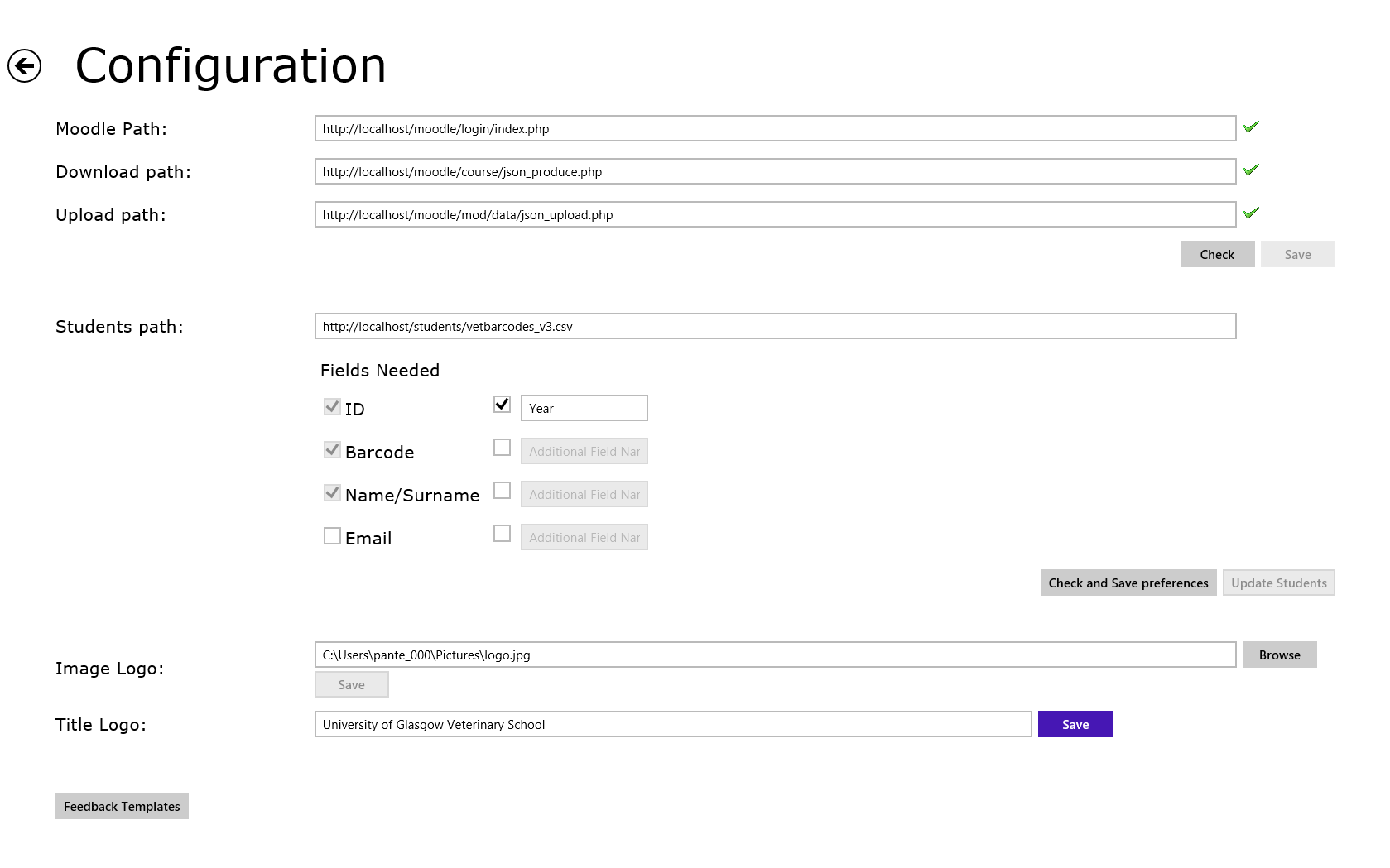
In our example the email of each student is going to be ignored and the year of study will be saved in the local student DB.

Image 20: Configure the student file. Add a custom column – year

**Reminder: Be careful with the student file. Some student names have double or single quotes inside them. This might cause errors while the file is being stored. Make sure to remove any of the two and then try to populate to the application.**

This data will be used inside the forms. Your forms need to have the according fields as well in order to function properly.

**See ‘how to create forms in Moodle’ first in order to understand the following**

The idea of the student list on the tablet is to automatically fill some of the form’s fields, either by the type-ahead functionality (search for student) or by taking a photo (scan) of the student’s id card (barcode). The name and matriculation number will be automatically be set on the form as soon as the student is chosen from the dropdown list or the id card of the student is photographed. If you would like to have some other information, linked with the student, to be automatically filled/completed then you must follow these steps.

First you must be sure that inside the student text file you have the extra information you would like to set for the student on the form e.g. year of study or email. Set the link of the student text file and click ‘Check and save preferences’ and after that click ‘update students’.

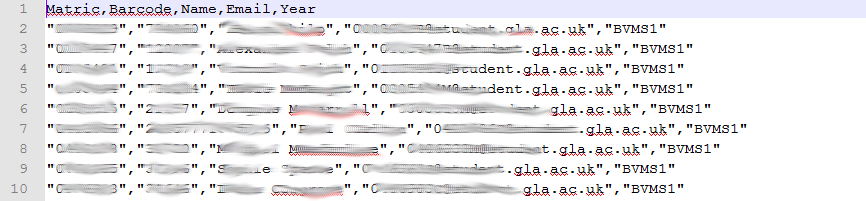
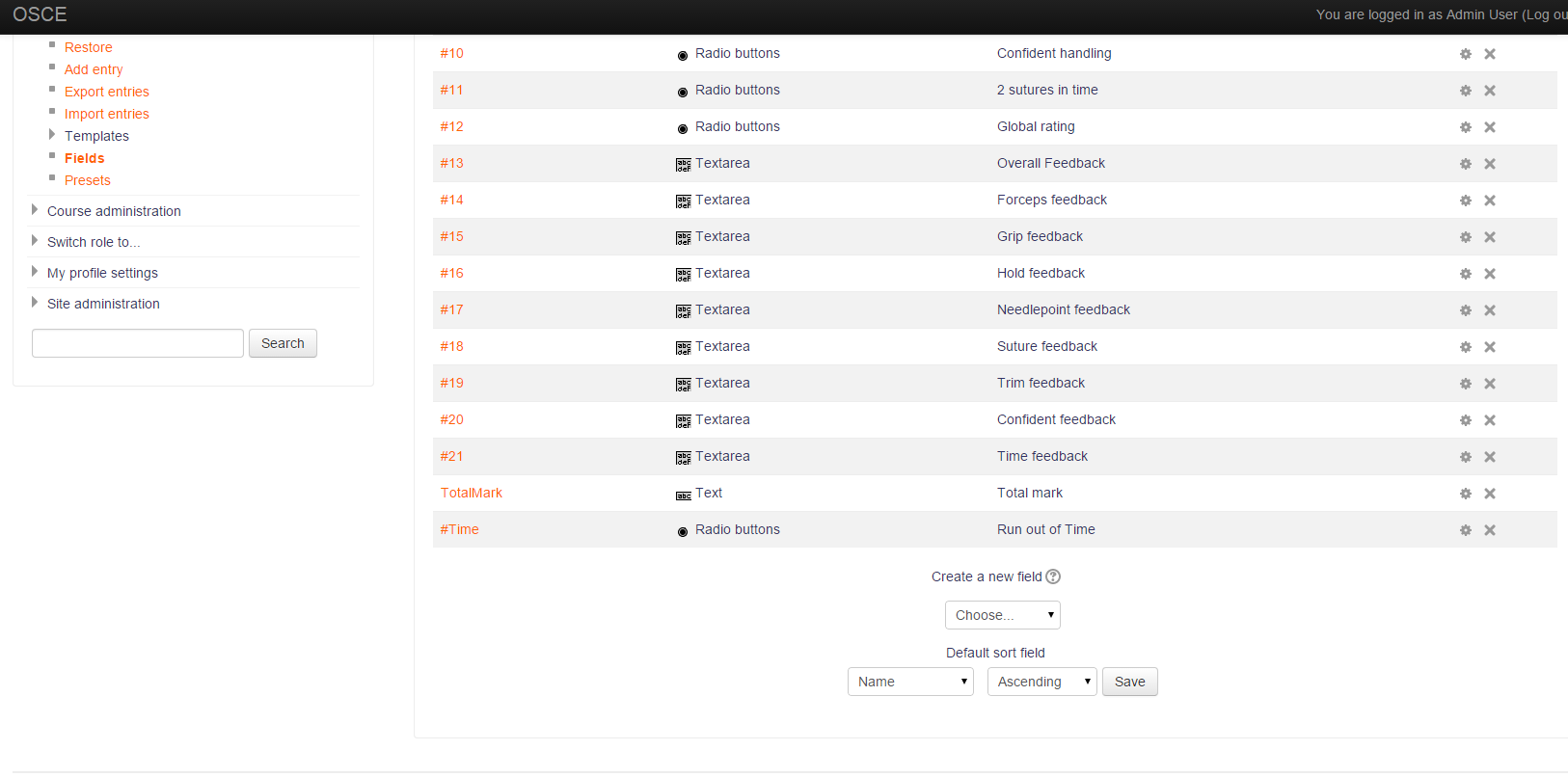


Image 21: Student file. Make sure you have the extra information that you are going to use

**Let’s try to add the ‘Year’ field in the form and populate it when a student is selected.**

Create a new field on the form (on Moodle) and set the name of that field the same as in the text file header. For our example the name will be ‘Year’.

Image 22: Create ‘Year’ field

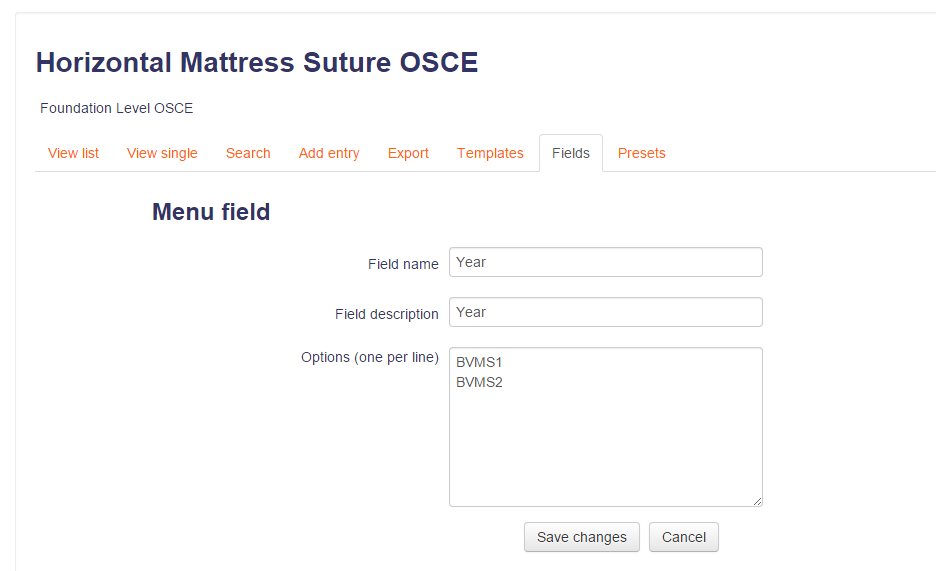


Choose menu

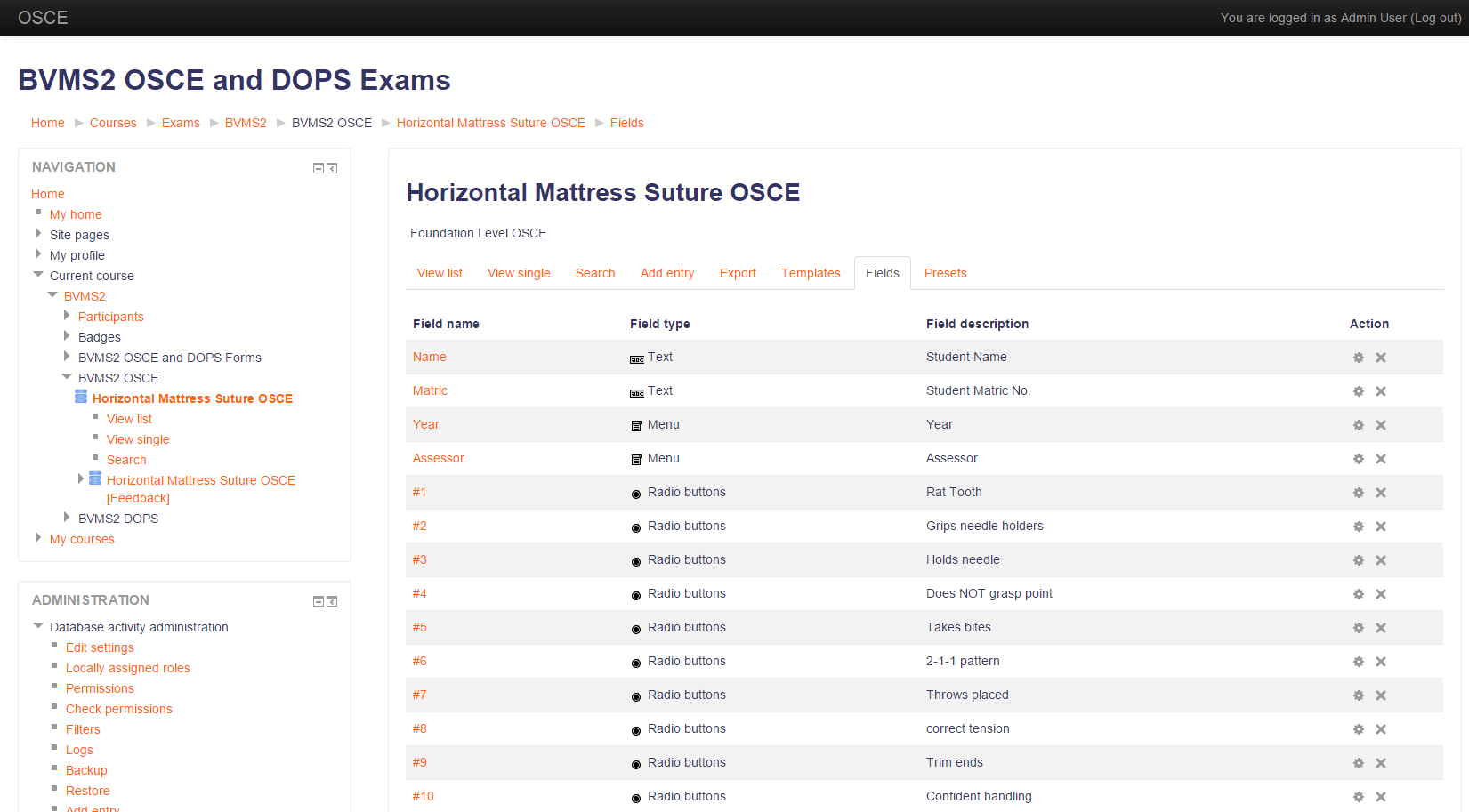
**Reminder: The text fields do not require to have hash ‘#’ in front of the name. Only radio buttons and textareas need hash. Text fields and dropdown menus are okay without.**

Image 24: Create ‘Year’ field

Image 23: Create ‘Year’ field



Must be the same as textfile column header



When you are done with setting up the form you can download it on the application. Provide your Moodle password and you can browse the forms. Choose the form you want and download. You must remember to add the new field to your template before downloading. (image 7)

# Download form from Moodle

In order for a user to be able to download exams on the application he/she must have a Moodle account. The administrator is responsible for that and also to place the PHP script files on the server. Because Moodle doesn’t have a direct way of creating a custom web service, 2 PHP script files were created in order to retrieve/send data from/to Moodle. Some research and further development is to be done in order to resolve this issue.

PHP files:

* json\_produce.php: This file should be placed on the following directory on your server: XAMPP\htdocs\moodle\course

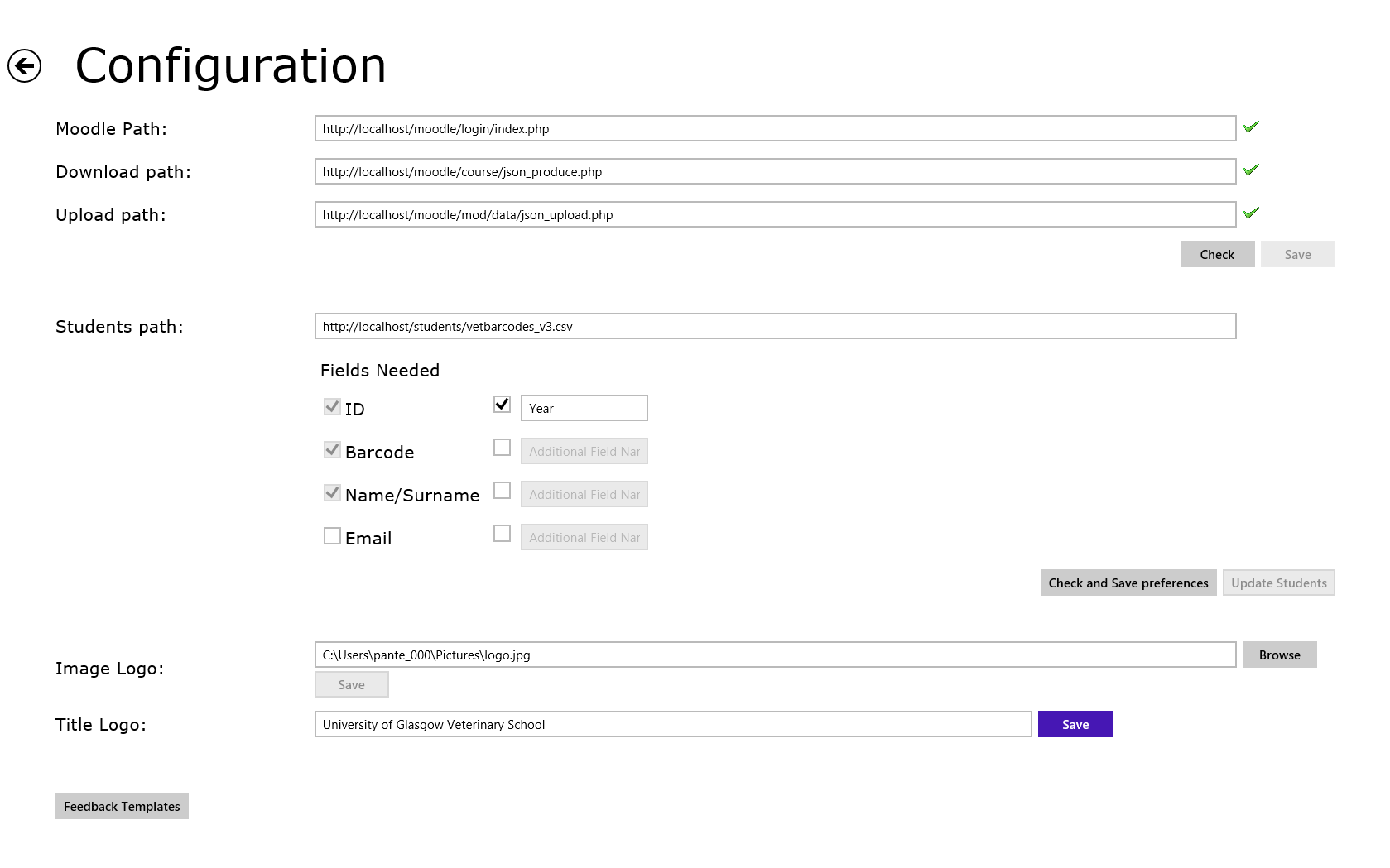
The purpose of this file is to receive requests from the application and construct the json object with the corresponding data that was asked for. You can see more on these requesta in the [API calls](#_Api_calls) chapter.

* json\_upload.php: This file should be placed on the following directory on your server:

XAMPP\htdocs\moodle\mod\data

The purpose of this file is to send the forms to Moolde and store them to the DB. It is basically submitting the forms to Moodle.

After placing the files to the corresponding directories the administrator must set these url paths on the application’s in order for the application to know where to make the requests.



When these configurations are set the user can download exam forms on the application.

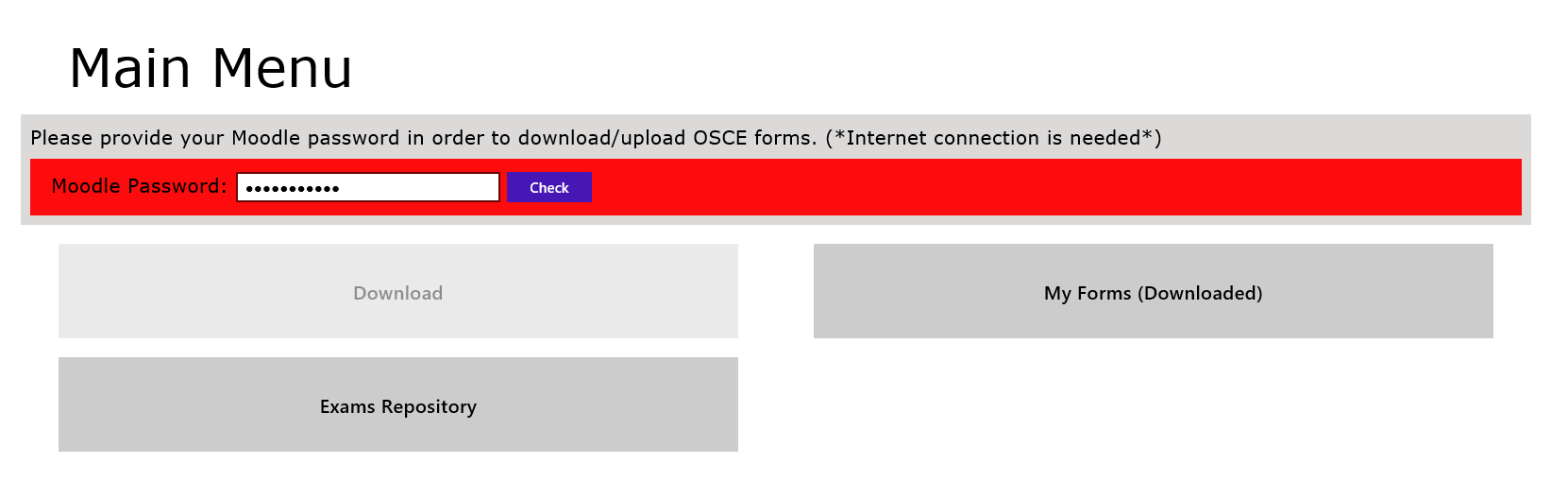
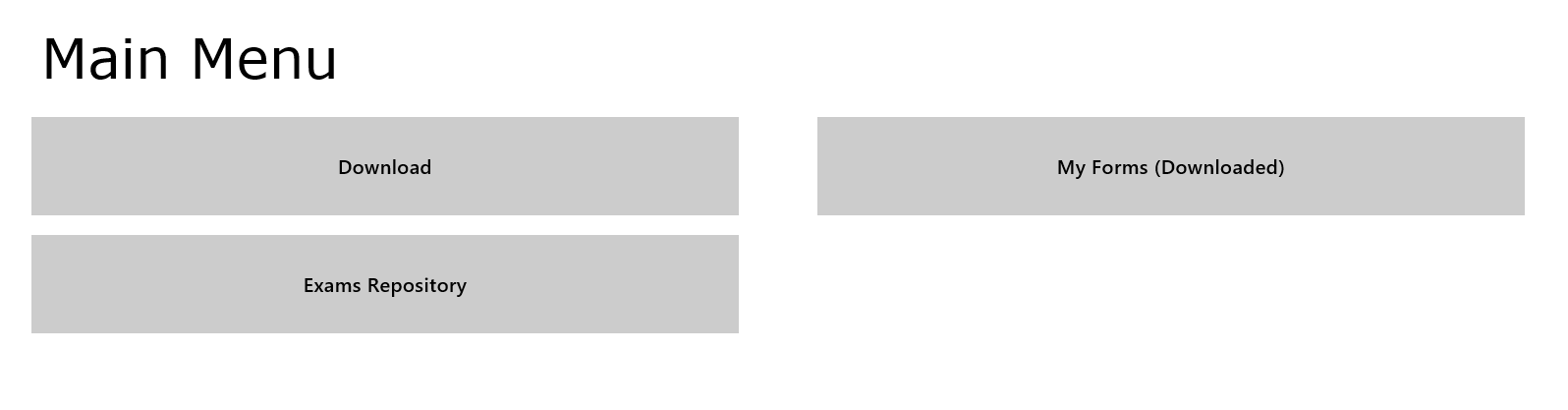
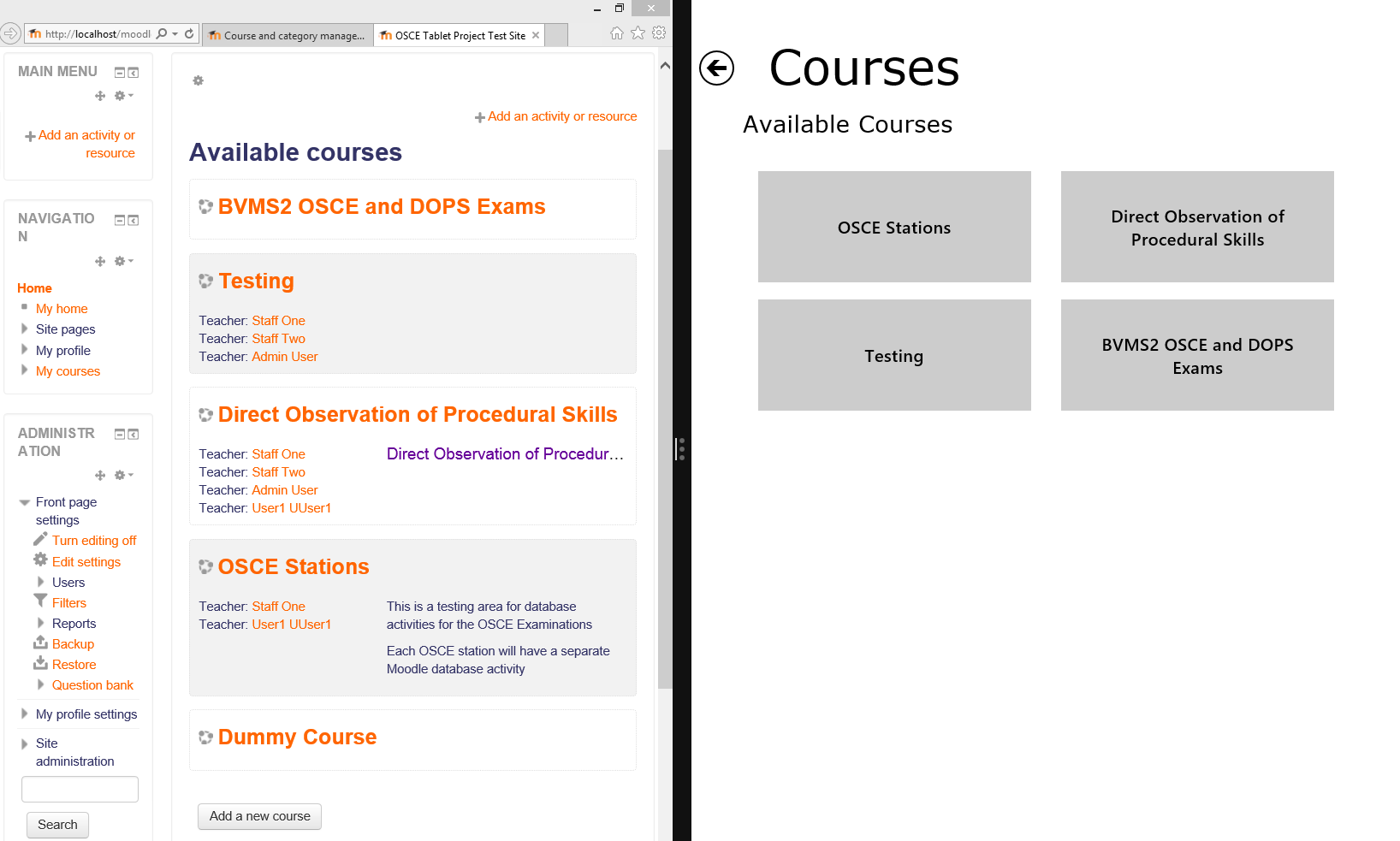
Enter your moodle password. The box will disappear if the user has successfully logged in. The password will automatically erased from the application in 10 minutes.

Image 25: Set the corresponding url links to the paths

When the configuration for downloading forms is set you can test it by clicking on the ‘Download’ button. You can see all the courses that are currently on Moodle under the ‘OSCE\_Exams’ category. (‘Dummy Course’ is under a different category)



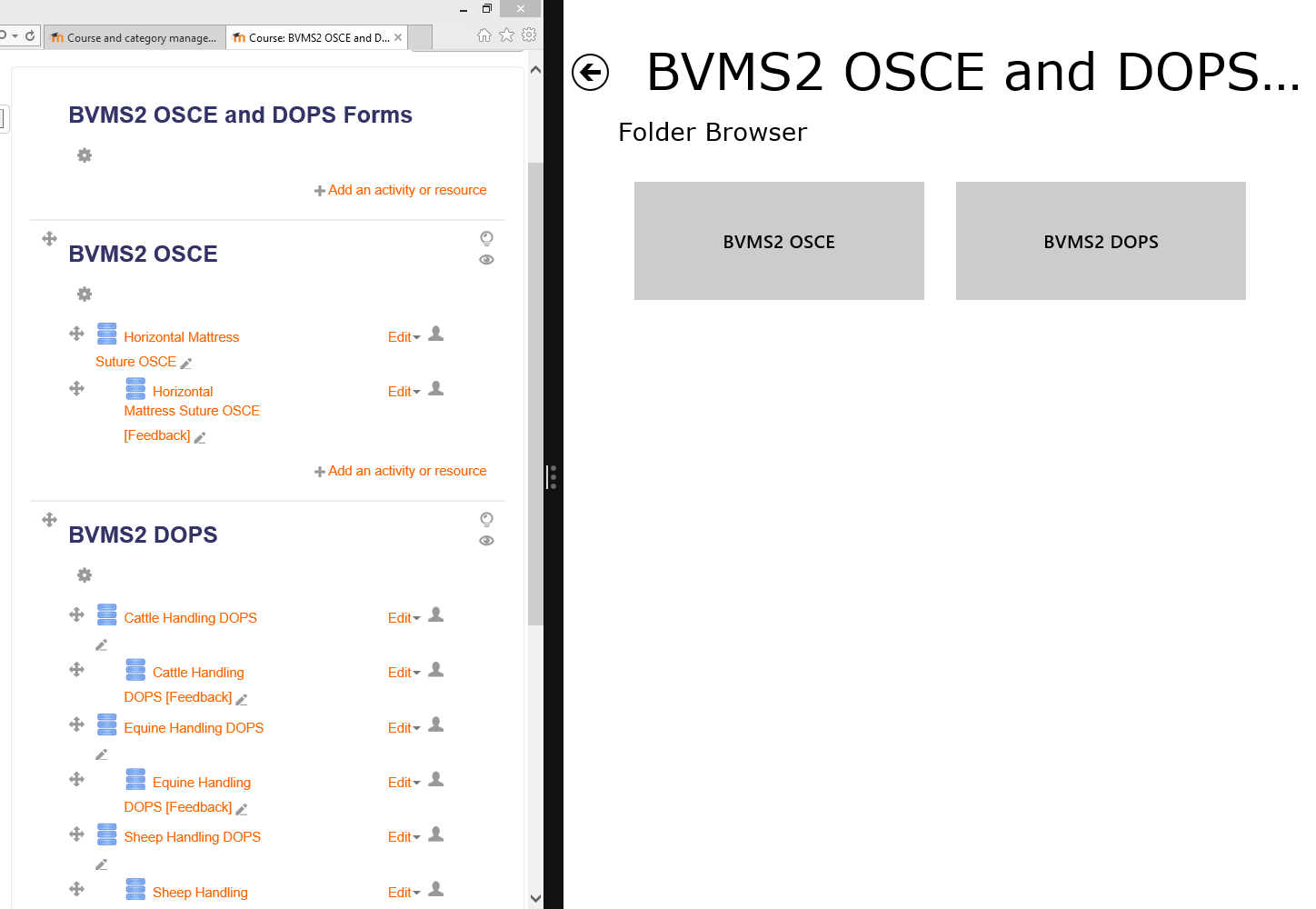


Image 26: Courses on Moodle and on the application

Image 27: Exams on Moodle and on the application

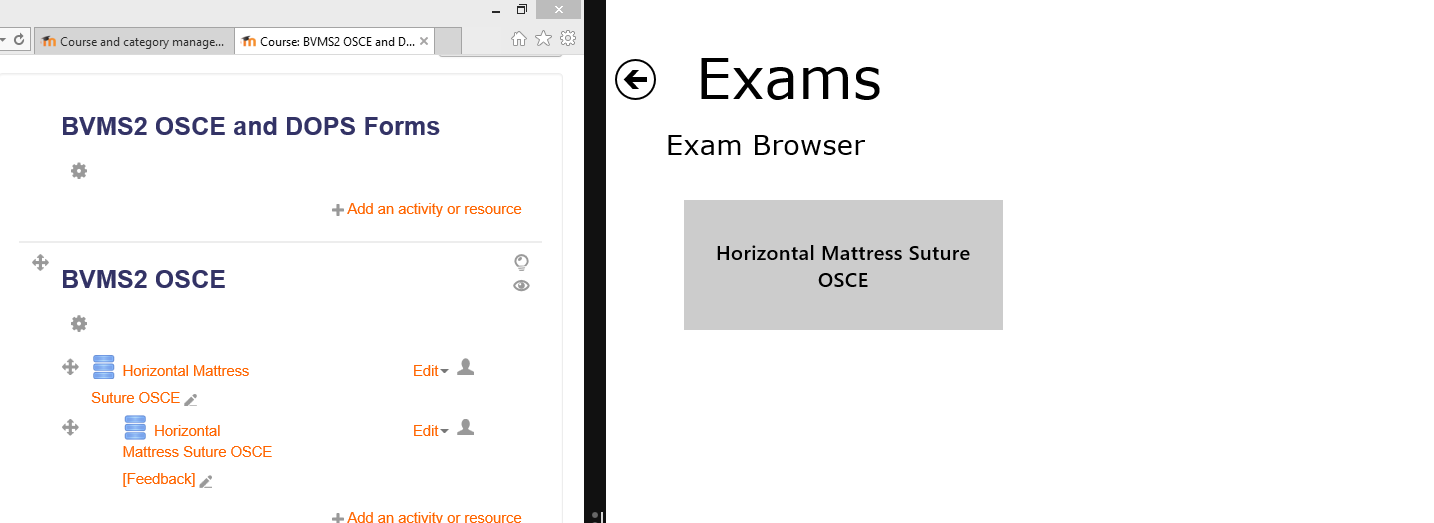
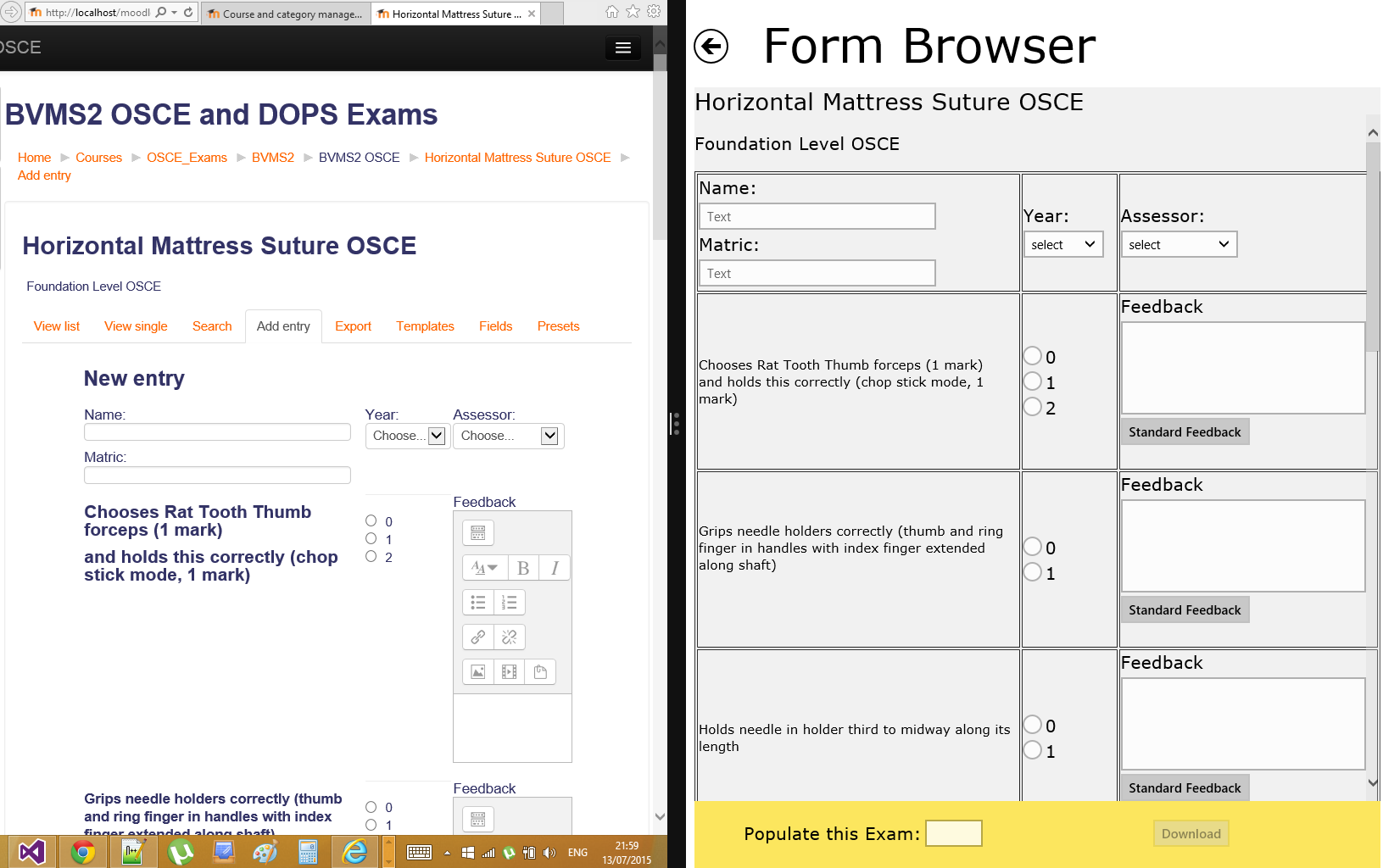
You notice that the feedback folders are not visible. Be sure that the name is the same and you have the ‘[Feedback]’ at the end of the name.

Image 28: Exam on Moodle and on the application

When you click on the form you can preview it in order to be sure that is the one to download. Just add the amount to download and click the button.

Image 29: Preview form on Moodle and on application



When the form has been downloaded on your application you can browse it from the ‘My forms’ folder. When you open a form you can check if the feedback comments were download by clicking on the ‘Standard Feedback’ button on any feedback textarea and click on the dropdown menu. The comments that were added in the [Feedback folder on Moodle](#_Feedback_Comments:) are downloaded with the exam and you can add as many as you want against each feedback textarea. Just select it and click add.

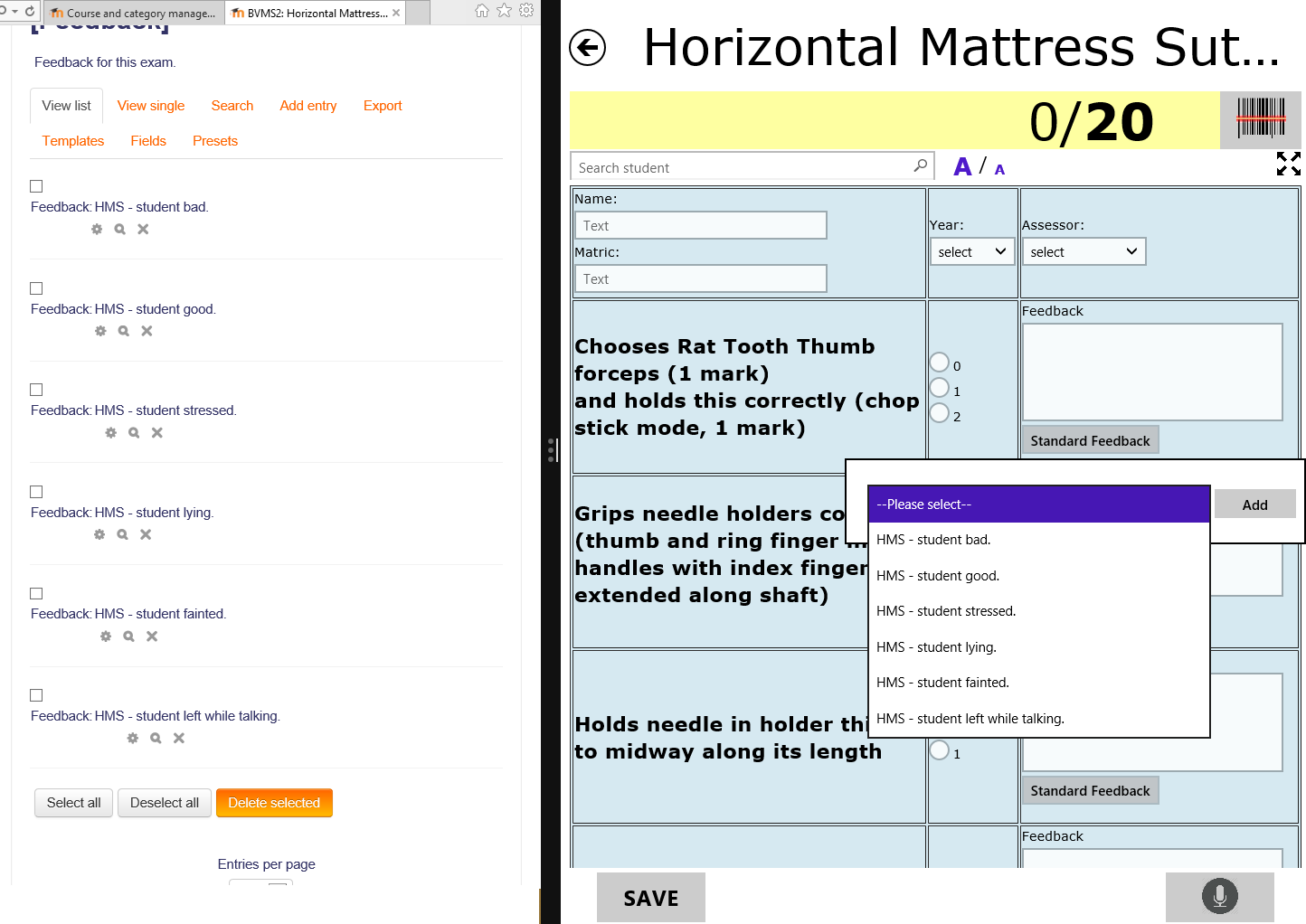


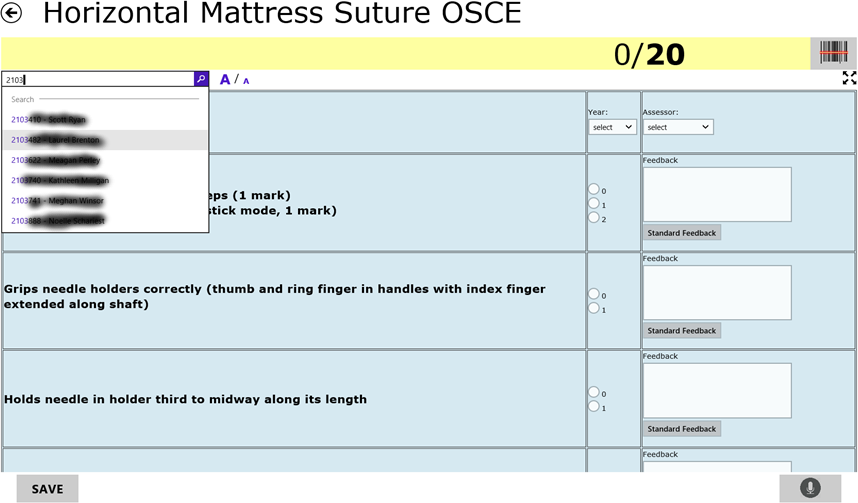
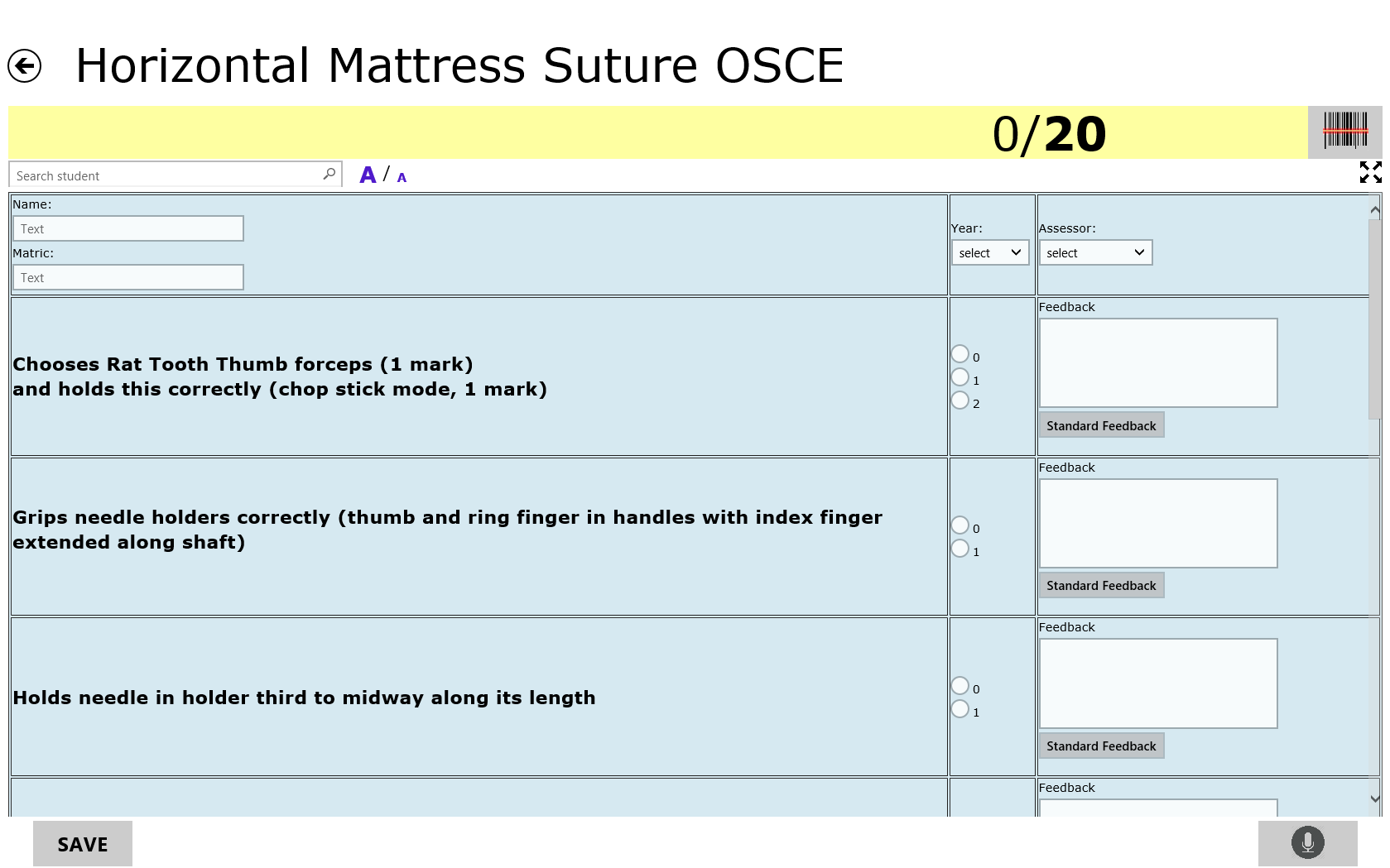
Image 30: The feedback dropdown menu.

# Edit form

Go to My Forms (Downloaded) and choose to open the form folder you would like. Click on one of the forms to start assessing the student.

When you are inside the form (for examination) you can start typing on the ‘search student’ text field and a dropdown list will show up. Choose a student and you can see that the student fields were automatically filled with the student’s information.

Image 31: Student typeahead dropdown list



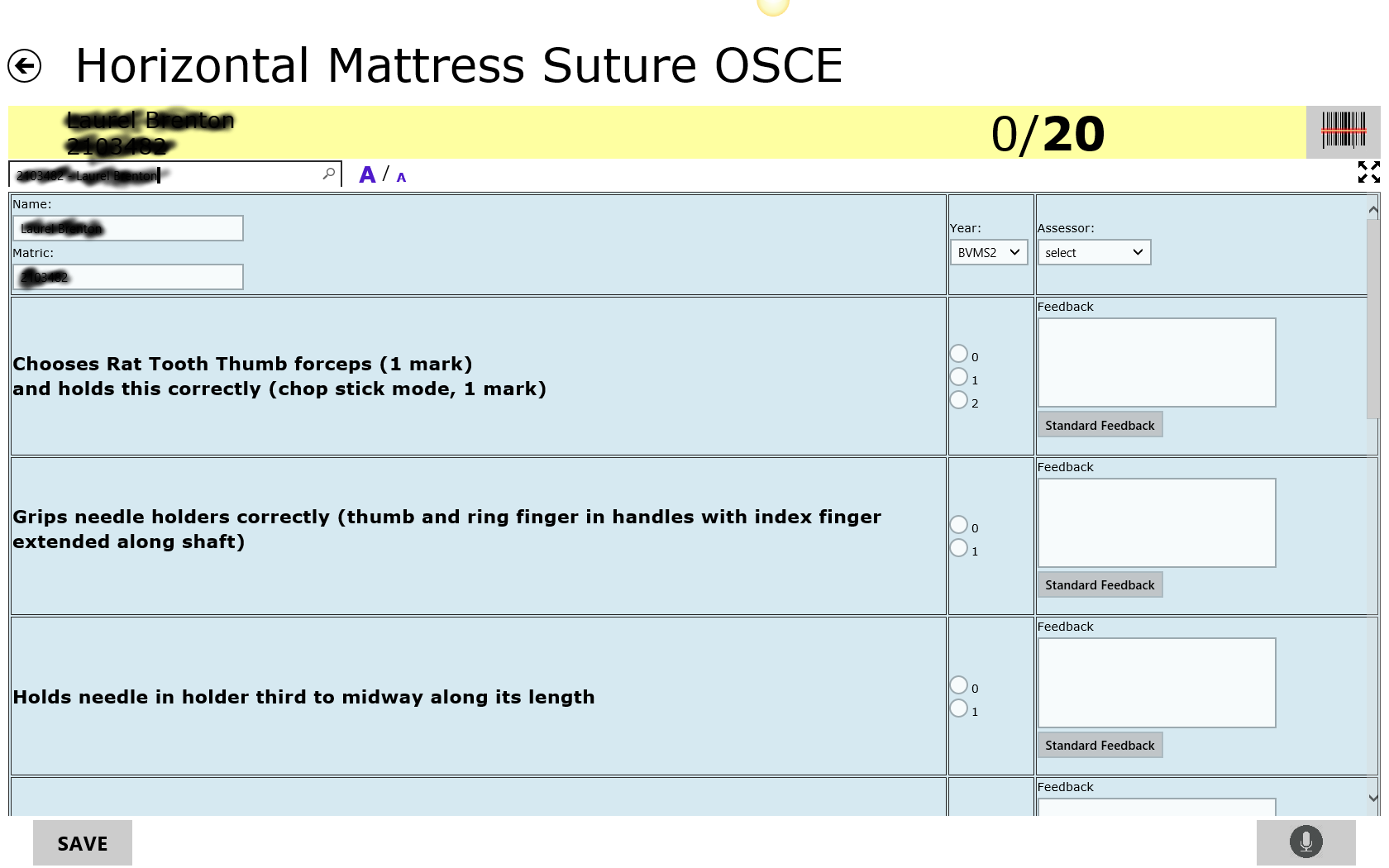
If the format of the form (on Moodle) includes a total mark then the total sum will be calculated and be visible at the top of the page as the examiner goes through the assessment. The total is updated while the examiner is assessing the student – while the radio buttons of the corresponding markings are being ‘ticked’ (usually on OSCE). If the format of the form does not include a total mark then the sum will not be visible (usually on DOPS).

Image 32: The student’s information is populated on the corresponding fields when selected

* Search for a student. The user can search for a student by either start typing the name or the id of the student. A dropdown list will appear with a typeahead search functionality throughout all of the students that are currently uploaded on the applications database. When you choose the student you want then the name, ID, year and assessor of that student will be populated automatically. That is if the appropriate data exist in the student text file that was uploaded to the application prior the examination process. (The administrator will be responsible for doing that)
* See the Name ID and total marking of the student.
* Can scan the student’s barcode ID and the name will be populated automatically (among will the rest of the information).
* Can full screen the form.
* Increase/Decrease the size of the text.
* When the user saves the form the incomplete parts will get a red background.
* Can add prepopulated feedback on the form instead of typing while the examination is happening.
* Can make a recording of the form and it will be saved on the tablet.
* The back button also saves the form.

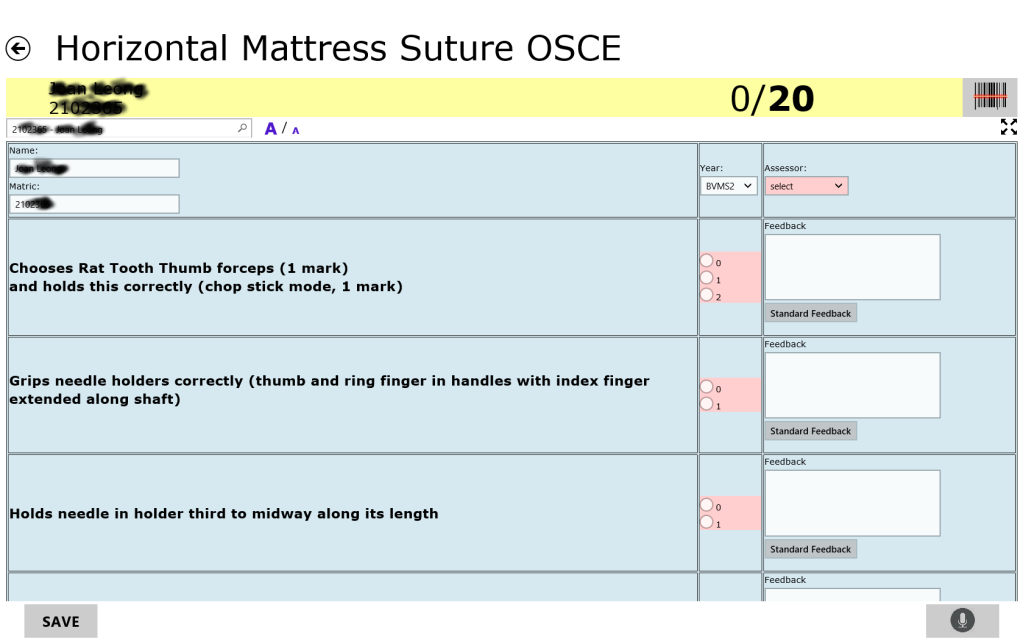
When the examiner saves a form the incomplete parts show a red background. That way the user will know which parts of the form are incomplete and focus on them. (form\_edit4)

Image 33: When the form is saved the incomplete fields get a red-ish background in order to be visible

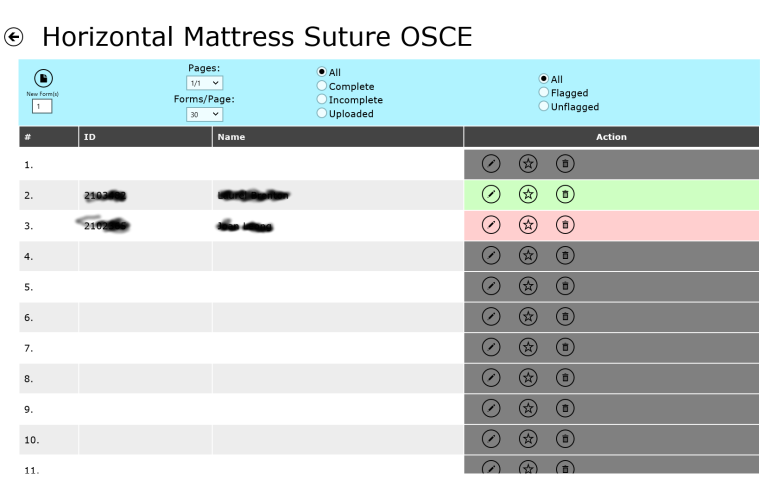
If the user does not fully complete a form and returns to the form list then the corresponding incomplete form will have a red background. If a form is fully completed then it will have a green background. Otherwise if it is empty it will have a grey background color. (forms2).

Image 34: Incomplete and complete form background color

In the ‘My Forms’ folder the user can also observe that if there are any completed forms then it will have a green background as well. Otherwise it has a white background.

Image 35: My forms folder when completed forms are saved

# Api calls

All requests and responses.

## Gets all categories from Moodle

Action: 1

**Request:**

http://localhost/moodle/course/json\_produce.php?action=1

You will get a list of all categories that are in the DB.

**Response:**

{

"Courses":

[

{

"id": "2",

"name": "OSCE Stations"

},

{

"id": "4",

"name": "Direct Observation of Procedural Skills"

},

{

"id": "9",

"name": "Testing"

},

{

"id": "10",

"name": "BVMS2 OSCE and DOPS Exams"

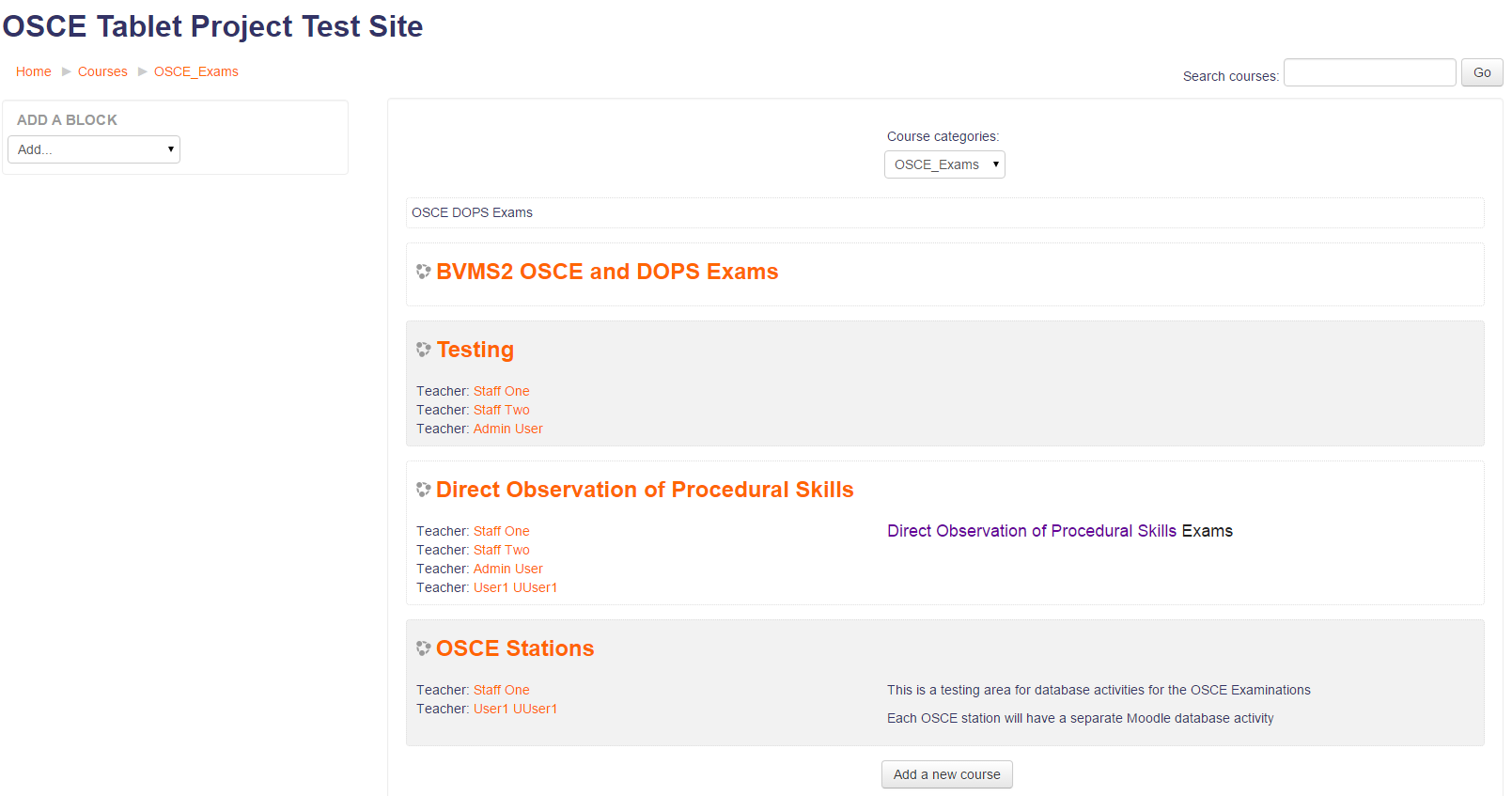
}

]

}

The response json object consists of id and name. Choose an id and pass it to the next url.

Image 36: Exams on Moodle



## Folders of the exams in Moodle

Action: 5 and the use the appropriate course id.

**Request:**

http://localhost/moodle/course/json\_produce.php?action=5&id=10

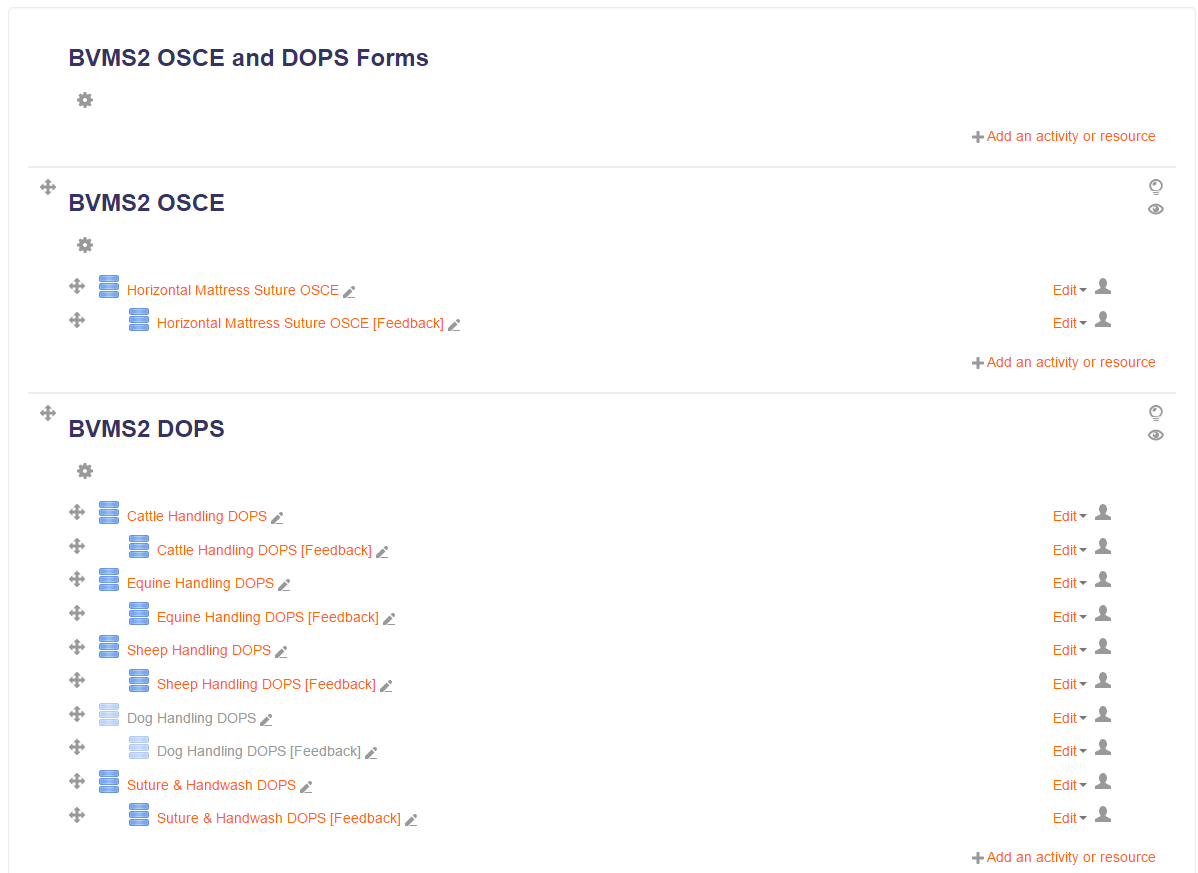
**Response:**

Image 37: Folders in Moodle

{

"Folders":

[

{

"id": "44",

"name": "BVMS2 OSCE"

},

{

"id": "45",

"name": "BVMS2 DOPS"

}

]

}

## Exams in the particular 'id' folder.

Action: 2 with the appropriate folder id.

**Request:**

http://localhost/moodle/course/json\_produce.php?action=2&id=45

**Response**:

{

"Exams":

[

{

"id": "26",

"name": "Cattle Handling DOPS",

"course": "45"

},

{

"id": "28",

"name": "Equine Handling DOPS",

"course": "45"

},

{

"id": "30",

"name": "Sheep Handling DOPS",

"course": "45"

},

{

"id": "34",

"name": "Suture & Handwash DOPS",

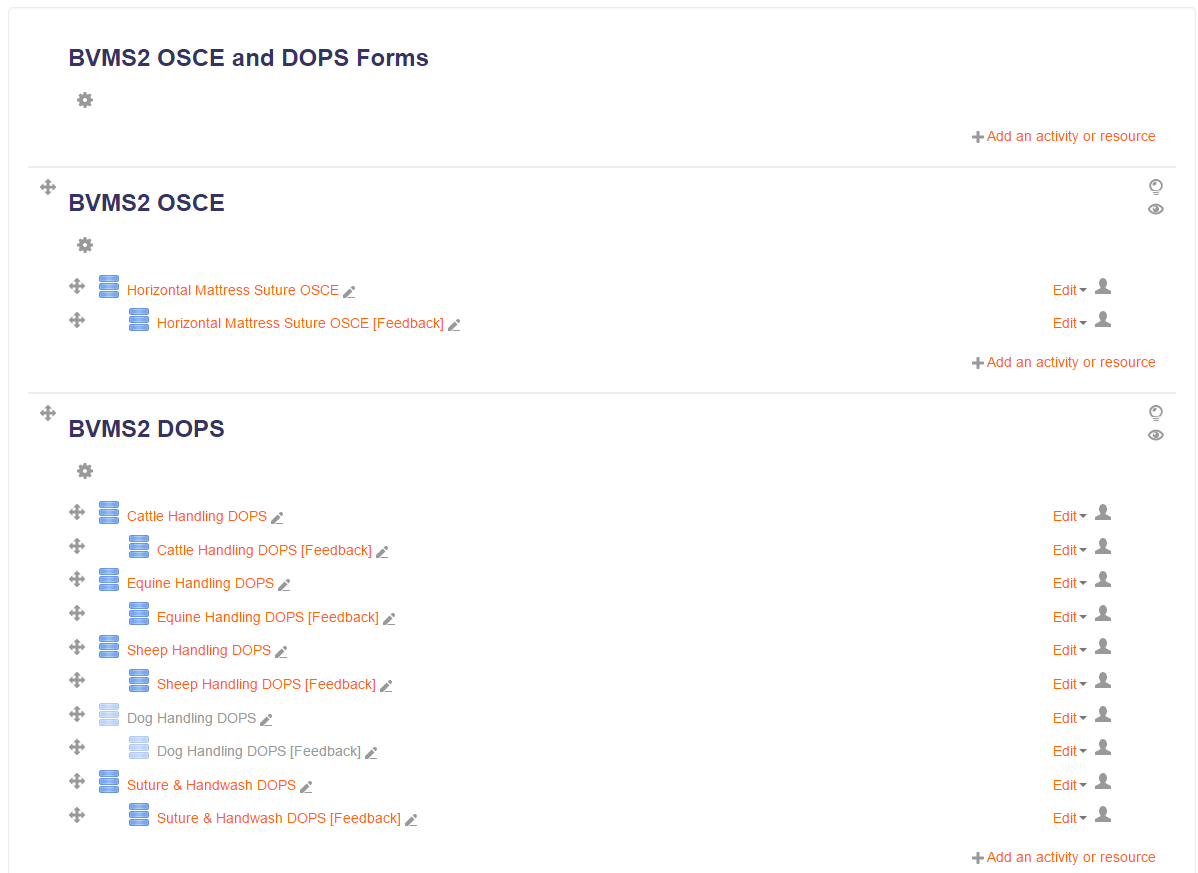
"course": "45"

}

]

}

Image 38: Exam forms in Moodle



**Gets the exam div template from Moodle.**

<http://localhost/moodle/course/json_produce.php?action=3&id=>**id**

SELECT D.id, D.name, D.intro, D.addtemplate FROM mdl\_data D WHERE D.id=?

json consists of id, name, intro, addtemplate

**Pass the form's fields in a way to save it on the app.**

<http://localhost/moodle/course/json_produce.php?action=4&id=>**id**

json consists of di, dataid, type, name, description, param1, param2, param3.

## Get the particular form on the tablet – Get the fields for that form

Action: 3 along with the form’s id

Action: 4 along with the form’s id

**Request:**

http://localhost/moodle/course/json\_produce.php?action=3&id=26

Gets the form’s template.

**Response:**

{

"OSCE":

[

{

"id": "26",

"name": "Cattle Handling DOPS",

"intro": "<p>Foundation level DOPS<br></p>",

"addtemplate": "<div style='text-align: left;' class='defaulttemplate'><table style='text-align: left; margin-left: 0px; margin-right: auto;' class='mod-data-default-template'> <tbody><tr class='r0'> <td class='template-field cell c0'><div style='text-align: left;'>Name: [[Name]]<br></div><div style='text-align: left;'>Matric: [[Matric]] </div></td> <td class='template-token cell c1 lastcol'><div style='text-align: left;'>Year: [[Year]]<br><br></div></td><td style='text-align: left;' class='template-token cell c1 lastcol'>Assessor: [[Assessor]]</td> </tr> <tr class='r1'> <td style='text-align: left;' class='template-field cell c0'><h5> Appropriate clothing</h5>Eg. clean overall, wellington boots, waterproof over trousers, jacket<br></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#1]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#9]]<br></td> </tr> <tr class='r0'> <td style='text-align: left;' class='template-field cell c0'><h5> Safely approach the cow</h5>Approaches animal in a safe and appropriate manner. Does not cause undue alarm to the animal. <br></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#2]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#10]]<br></td> </tr> <tr class='r1'> <td style='text-align: left;' class='template-field cell c0'> <h5>Move/shed the 'candidate' cow away from the group towards the race or adjacent pen</h5><p>Induces the selected animal to move in a manner appropriate to its level of compliance without causing it undue alarm. Stands in an appropriate position and uses appropriate body language. </p> </td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#3]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#11]]</td> </tr> <tr class='r0'> <td style='text-align: left;' class='template-field cell c0'><h5>Direct the cow through the race or into an adjacent pen</h5>As above without compromising their safety or that of the animal.<br></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#4]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#12]]</td> </tr> <tr class='r1'> <td style='text-align: left;' class='template-field cell c0'><h5>Apply a rope halter </h5>Where a cow is restrained in a crush or in locking headgates, applies halter without causing harm or distress to the animal. Finishes with it correctly positioned, at an appropriate tension and tied to a secure point using a quick release hitch. <br></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#5]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#13]]</td> </tr> <tr class='r0'> <td style='text-align: left;' class='template-field cell c0'><h5>Remove the halter</h5>Safely removes the halter without causing harm or distress to the animal<br></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#6]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#14]]</td> </tr> <tr class='r1'> <td style='text-align: left;' class='template-field cell c0'><h5>Demonstrates compassion, care and due respect to animal and owner </h5></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#7]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#16]]</td> </tr> <tr class='r0'> <td style='text-align: left;' class='template-field cell c0'><h5>Observe and adhere to relevant regulations regarding health and safety </h5></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#8]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'>Feedback<br>[[#17]]</td> </tr> <tr class='r1'> <td style='text-align: left;' class='template-field cell c0'><h5>Overall Outcome</h5></td> <td style='text-align: left;' class='template-token cell c1 lastcol'>[[#15]]</td><td style='text-align: left;' class='template-token cell c1 lastcol'><br></td> </tr> <tr class='r0 lastrow'> <td style='text-align: left;' class='template-field cell c0' colspan='2'><h5>Overall Feedback</h5>[[#18]]<br><br></td> </tr> </tbody> </table></div>"

}

]

}

**Request:**

http://localhost/moodle/course/json\_produce.php?action=4&id=26

Gets the fields of the corresponding form.

**Response:**

{

"Fields":

[

{

"id": "164",

"dataid": "26",

"type": "text",

"name": "Name",

"description": "Student Name",

"param1": "",

"param2": "",

"param3": ""

},

{

"id": "165",

"dataid": "26",

"type": "text",

"name": "Matric",

"description": "Student Matric No.",

"param1": "",

"param2": "",

"param3": ""

},

{

"id": "166",

"dataid": "26",

"type": "menu",

"name": "Year",

"description": "Year",

"param1": "BVMS1|BVMS2",

"param2": "",

"param3": ""

},

{

"id": "167",

"dataid": "26",

"type": "menu",

"name": "Assessor",

"description": "Assessor"

"param1": "person 1|person 2",

"param2": "",

"param3": ""

},

{

"id": "168",

"dataid": "26",

"type": "radiobutton",

"name": "#1",

"description": "Appropriate clothing",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "169",

"dataid": "26",

"type": "radiobutton",

"name": "#2",

"description": "Safely approach the cow",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "170",

"dataid": "26",

"type": "radiobutton",

"name": "#3",

"description": "Move/shed â€˜candidateâ€™ cow",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "171",

"dataid": "26",

"type": "radiobutton",

"name": "#4",

"description": "Direct the cow",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "172",

"dataid": "26",

"type": "radiobutton",

"name": "#5",

"description": "Apply halter",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "173",

"dataid": "26",

"type": "radiobutton",

"name": "#6",

"description": "Remove halter",

"param1": "S|NS|N/A",

"param2": "",

"param3": ""

},

{

"id": "174",

"dataid": "26",

"type": "radiobutton",

"name": "#7",

"description": "compassion, care and due respect",

"param1": "S|NS",

"param2": "",

"param3": ""

},

{

"id": "175",

"dataid": "26",

"type": "radiobutton",

"name": "#8",

"description": "health and safety regulations",

"param1": "S|NS",

"param2": "",

"param3": ""

},

{

"id": "176",

"dataid": "26",

"type": "textarea",

"name": "#9",

"description": "clothing comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "177",

"dataid": "26",

"type": "textarea",

"name": "#10",

"description": "approach comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "178",

"dataid": "26",

"type": "textarea",

"name": "#11",

"description": "move comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "179",

"dataid": "26",

"type": "textarea",

"name": "#12",

"description": "direct comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "180",

"dataid": "26",

"type": "textarea",

"name": "#13",

"description": "apply halter comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "181",

"dataid": "26",

"type": "textarea",

"name": "#14",

"description": "remove halter comment",

"param1": "",

"param2": "80",

"param3": "3"

},

{

"id": "182",

"dataid": "26",

"type": "radiobutton",

"name": "#15",

"description": "overall outcome",

"param1": "Needs further development|Competent|Excellent",

"param2": "",

"param3": ""

},

{

"id": "183",

"dataid": "26",

"type": "textarea",

"name": "#16",

"description": "compassion comment",

"param1": "",

"param2": "30",

"param3": "3"

},

{

"id": "184",

"dataid": "26",

"type": "textarea",

"name": "#17",

"description": "safety feedback",

"param1": "",

"param2": "30",

"param3": "3"

},

{

"id": "185",

"dataid": "26",

"type": "textarea",

"name": "#18",

"description": "overall feedback",

"param1": "",

"param2": "60",

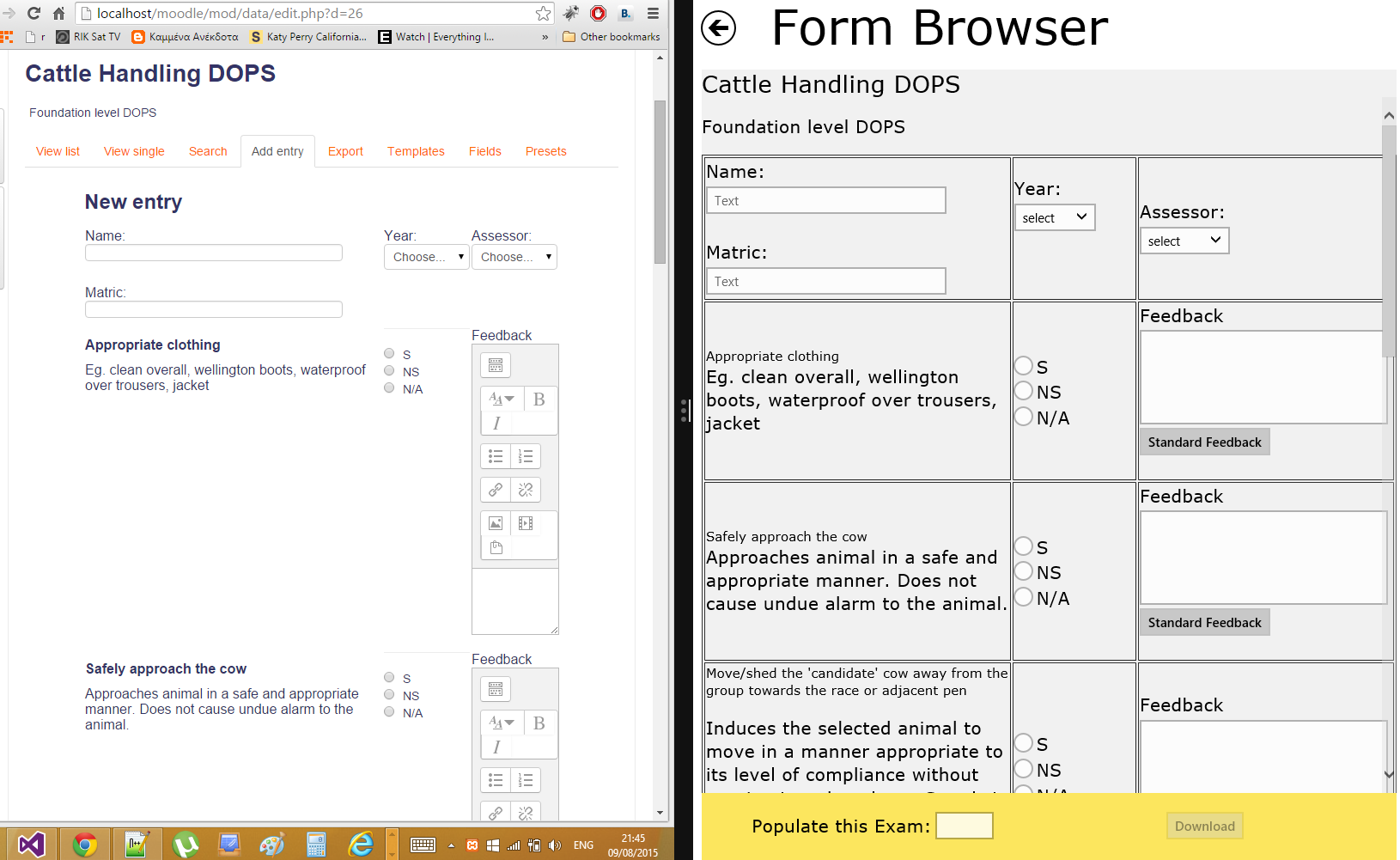
"param3": "5"

}

]

}

Image 40: Form preview in Moodle and on the application



## Gets the form's feedback.

Action: 6 with the corresponding form id.

**Request:**

http://localhost/moodle/course/json\_produce.php?action=6&id=26

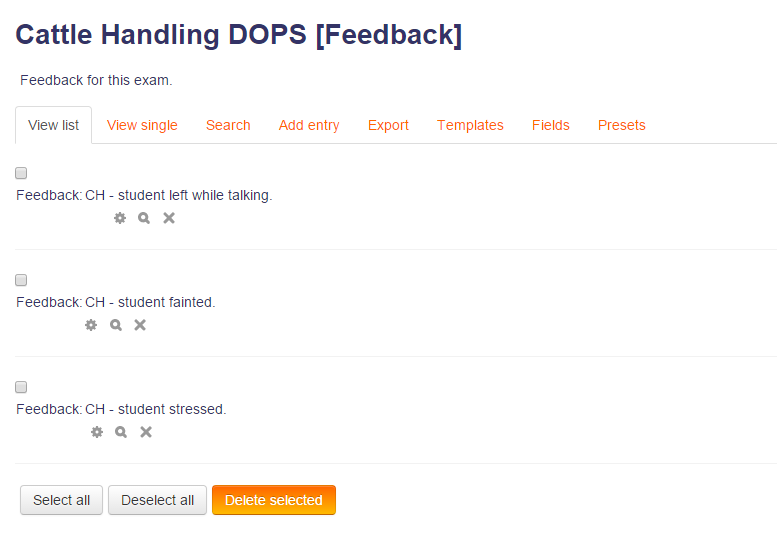
**Response:**

Image 41: Form’s feedback on Moodle

{

"Feedback":

[

{

"id": "903",

"name": "Cattle Handling DOPS [Feedback]",

"content": "CH - student left while talking."

},

{

"id": "904",

"name": "Cattle Handling DOPS [Feedback]",

"content": "CH - student fainted."

},

{

"id": "905",

"name": "Cattle Handling DOPS [Feedback]",

"content": "CH - student stressed."

}

]

}