Lucia Asencio Martin
Juan Riera

Lucia.asecio@estudiante.uam.es
juan.riera@estudiante.uam.es

TEST PLAN AND TEST RESULTS

Application: Moon

DATE: 10/05/2017

1. TEST CASE ADD EXERCISE

1.1 Use Case: Add exercise

This use case represents the moment when a teacher wants to add an exercise to an existing unit in an existing course.
We are going to test the main path, with some mistakes made by the user so that the test shows how the app behaves in those alternative path.

Its primary actors will be teachers, and the goal is to make an exercise using the exercise maker the application provides. The success guarantee is that the exercise will be added to the course tree, and accessible to students when visible and active.

Main success scenario:

i. Teacher has logged in, went in the course and selects to edit the unit where he wants to add the exercise.

ii. Teacher selects to Add an Exercise

iii. Teacher selects the dates where it will be active, a relevance, a penalty for each question, the order of the question, and a name.

iv. Teacher selects to add a new T/F question, and gives it a relevance and a number.

v. Teacher writes the statement and selects a truth value for it. Saves the exercises.

Extension 1:

Teacher selects to add a Single Choice question, and gives it a relevance and a number.

Teacher writes the question, and enters a right answer and some wrong answers. Teacher selects an order for the different options. Saves the exercise.

Extension 2:

Teacher selects to add a Multiple Choice question, and gives it a relevance and a number.

Teacher writes the question, and enters some right answers and some wrong answers. Teacher selects an order for the different options. Saves the exercise.

Extension 3:

Teacher selects an Open answer question, and gives it a relevance and a number.

Teacher writes down the question and adds different correct answers. Saves the exercise.

1.2. Test case design (including expected inputs and outputs)**:**

Preconditions:

i. *The Academy needs to have at least one course (subject) with at least one unit.*
ii. *You need to be a user that has logged in as a teacher (this means: email tea.cher@edu.es, password IsALotOfWork13579) .*
iii. *You need to have entered the course page (clicking in the course of the course list which is in the teacher main page).*
iv. *You need to have opened the course tree in the course page*

Scenario:

1. User makes selects *unit* in the tree.
2. User clicks on edit element and contents.
3. System changes view to EditUnitView.
4. User clicks on *Save & add exe.*

5. System changes view to *AddExeView,* and creates an exercise where to put the information it will collect from the view.
6. User fills the *activity date from, activity date to. T*he user tries to choose from 1/1/2014 to 2/2/2017. As he/she is a fool, he writes “1017” instead of “2017” and “201A” instead of “2014”.
7. User selects checkbox visibility, as he wants a visible exercise, and random order, as he wants the questions to be displayed randomly.
8. User selects penalty (-5) for each question and the relevance (5) of the exercise in the Jspinners. If he enters invalid values (negative relevance, positive penalty) values are automaticlly fixed.
9. User presses *Save and back*
10. System detects no name has been chosen, tells the user.
11. User presses *OK*
12. System detects invalid dates have been chosen, tells the user.
13. User presses *OK*
14. User correctly fill the fields: changes 201A with 2014, 1017 with 2017, fills name field with “MyBeautifulExercise”, presses *Save and back*
15. System detects no added questions , tells the user.
16. User presses *OK*
17. User presses *True/False question*
18. System opens a popup where to fill in the True/False question
19. User selects a relevance 1.9 for the question, a question number (1), then writes the question (“I am fantasticly fantabulous”) and selects the correct answer (True).
20. User presses *save and exit*.
21. System closes pop up, saves the question.
22. User presses *Mutiple Choice Question*.
23. System opens a pop up where to fill the question.
24. User selects relevance 1.6, number of question 2, writes the question(“Who is the best?”), tries to save.
25. System detects no correct/incorrect answers to the question have been added to the question, tells the user. *We are not going to repeat this step of “forgetting to add an answer to the question” for each type, but it is the same in the four types of answers*
26. User presses *OK*
27. User fills correct answers;
    Writes “Lucia”, presses *Add*
    Writes “Both”, presses *Add*
    Writes “Lucia”, presses *Add*
    Writes “Lucia”, presses *Add*
    And an incorrect answer: Writes “Juan Riera”, presses *Add*
28. User presses *save and exit*
29. System saves the question, closes the pop up
30. User presses *Single Choice Question*
31. System opens a pop up where to fill the question.
32. User selects relevance 1.6, number of question 3, writes the question(“Do I want to study SOPER?”), add the correct answer in the text field (“No”), the incorrect answers (Writes “Maybe”, presses *Add*, writes “Of course”, presses *Add*, writes “My imagination for making exercises is null”, presses *Add*). For this question, user decides to display options randomly: selects *Random order of options* checkbox. Tries to save pressing *Save and exit*.
33. System saves the question, closes the pop up
34. User presses *Open answer question*
35. System opens a pop up where to fill the question.
36. User selects relevance 1.3, number of the question 4, writes the question “Who is the Java Master?”, write some correct answers (writes “Juan de Lara”, presses *Add,*

writes “Juan Riera”, presses *Add*, writes “Lucia Asencio”, presses *Add*). User tries to save: presses *Save and exit*.

37. System saves the question, closes pop up.
38. User presses *Save and back* to save the exercise
39. The add exercise view closes, the course main page is opened.
40. User selects *curso*
41. User opens the tree to see the exercise

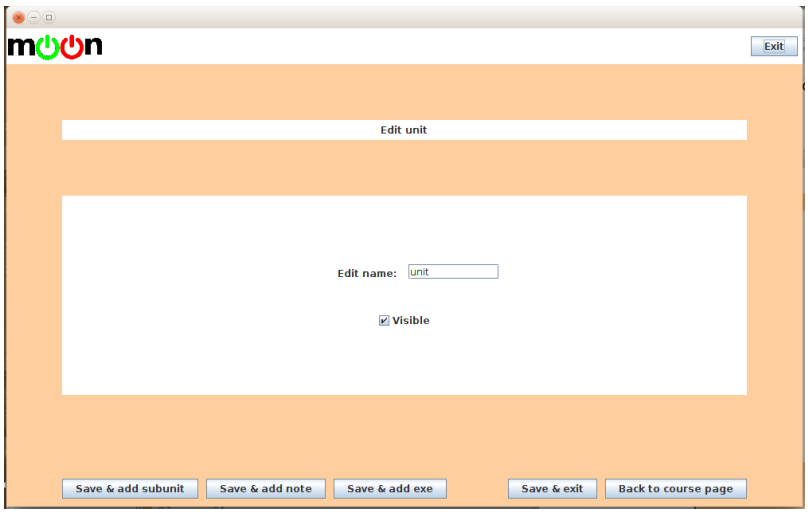
Some of the steps in the usecase are a bit different from the ones in the testcase, but this is caused because of the user entering invalid data to the application, such as empty fields, wrong dates, etc. that make the system react and prevent the user from saving invalid data.

1.3. Test execution result

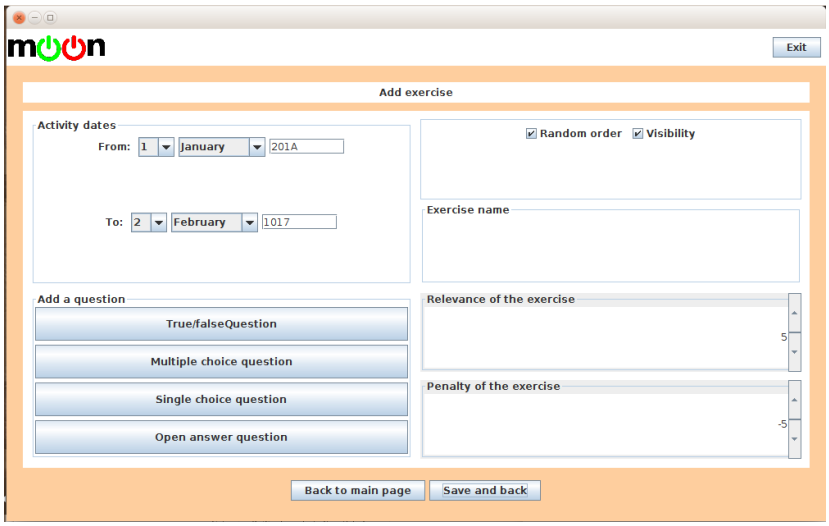
- Screenshot, associated with the steps enumerated in **Scenario**

  Steps 1, 2: entering the course page and selecting the unit

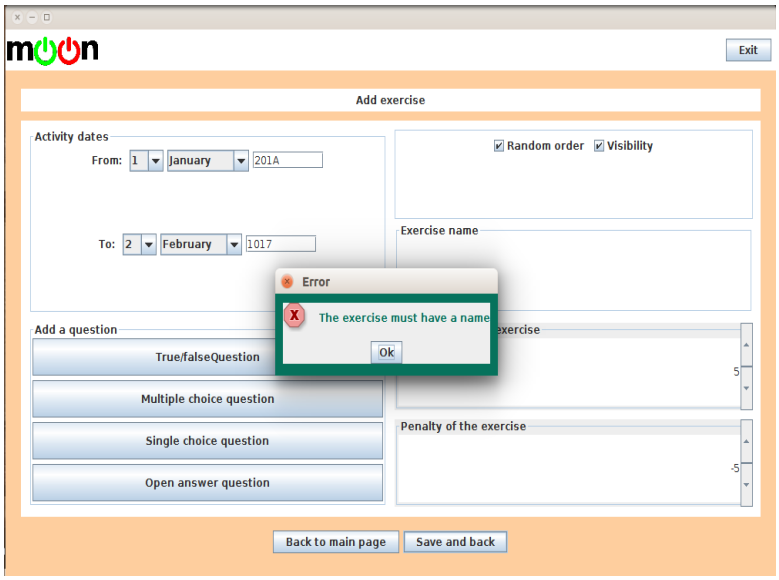
Steps 2, 3: editing the unit by selecting add exercise



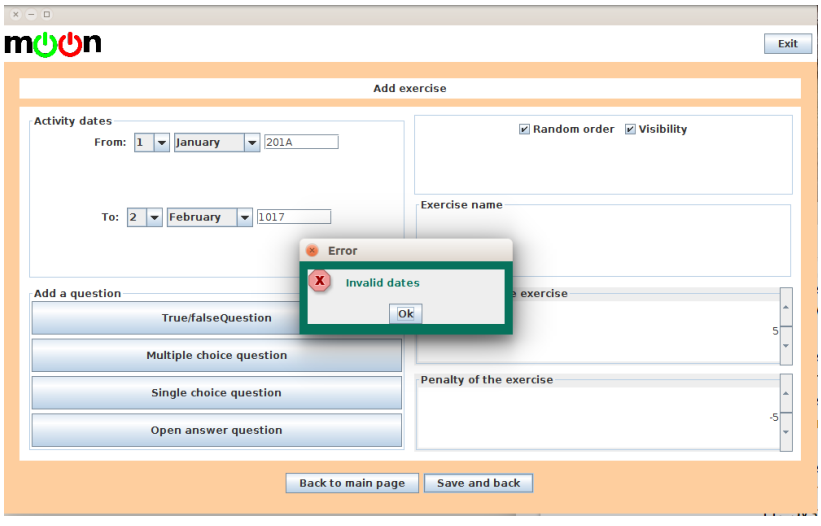
Steps 5,6,7,8,9: filling some fields of the add exercise view, trying to save



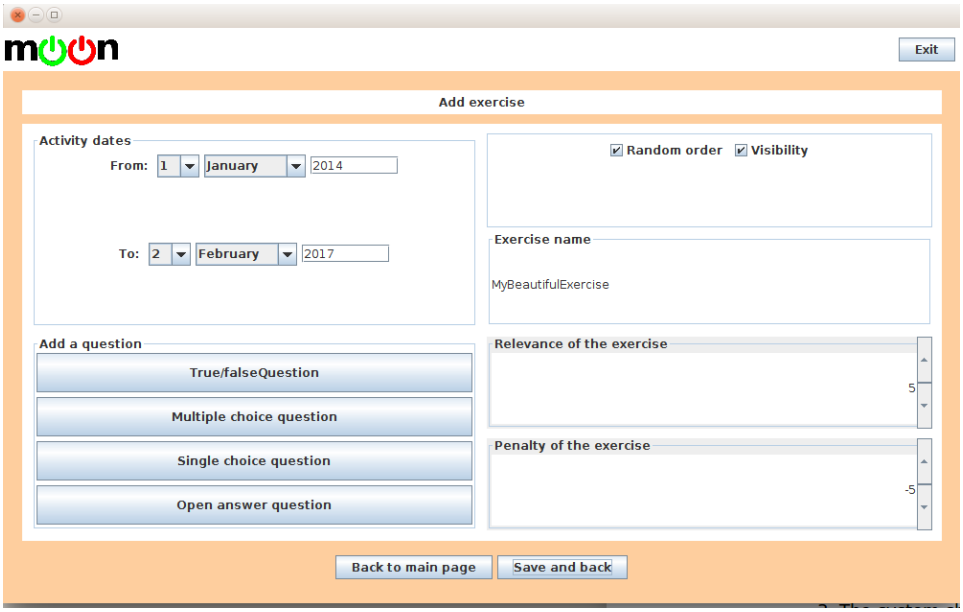
Step 10, 11: invalid name



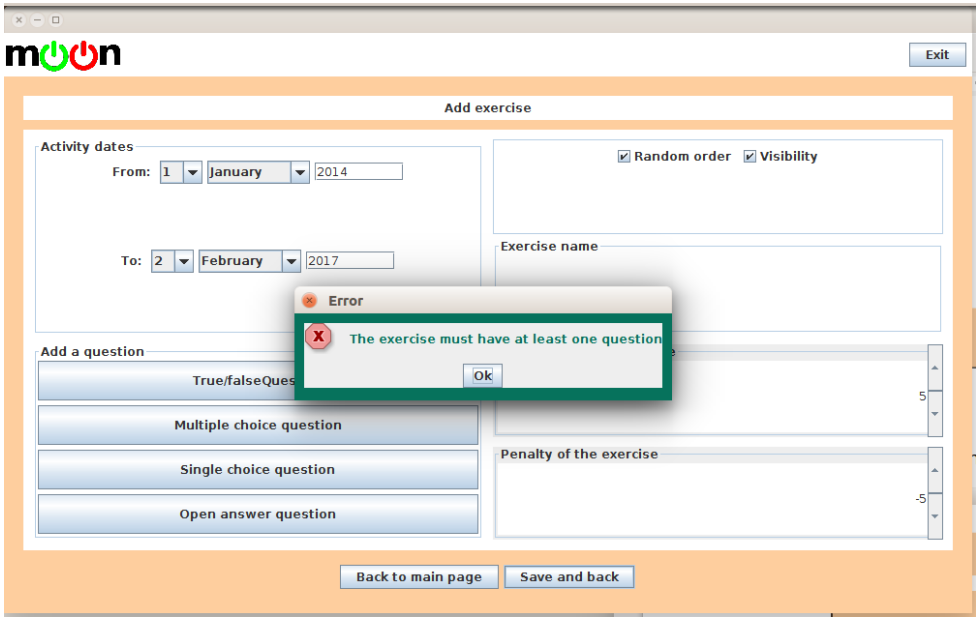
Step 12, 13: invalid date



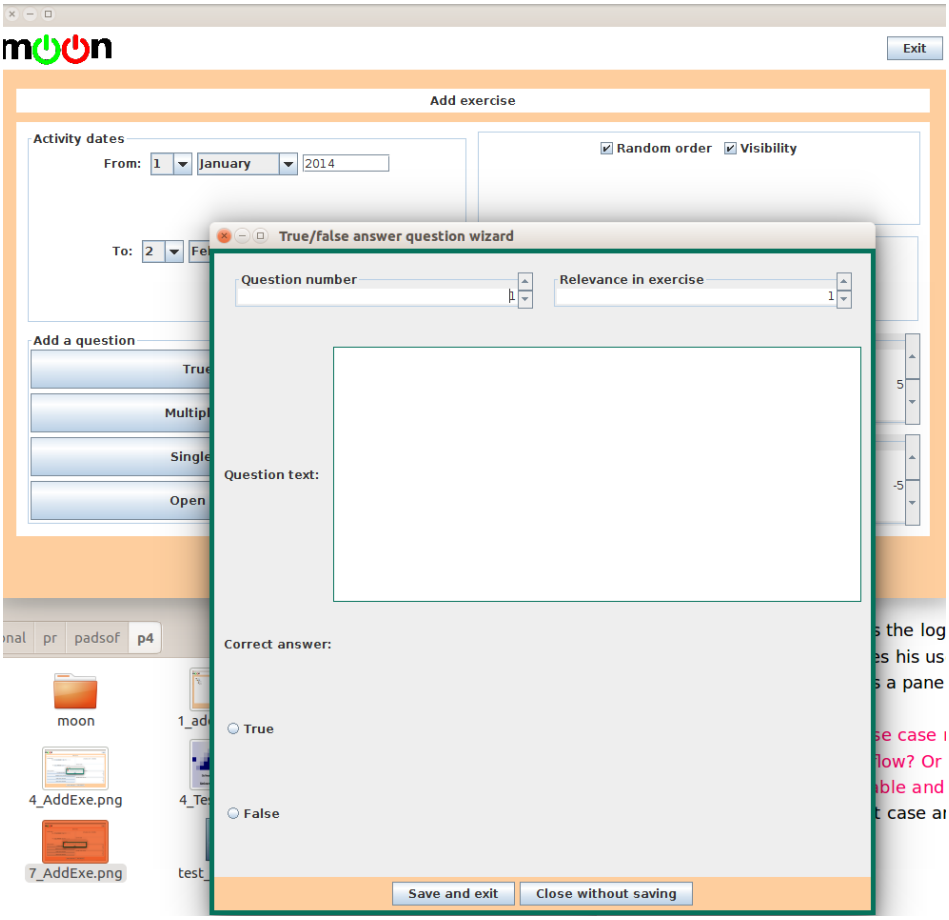
Step 14: Refill, tries to save



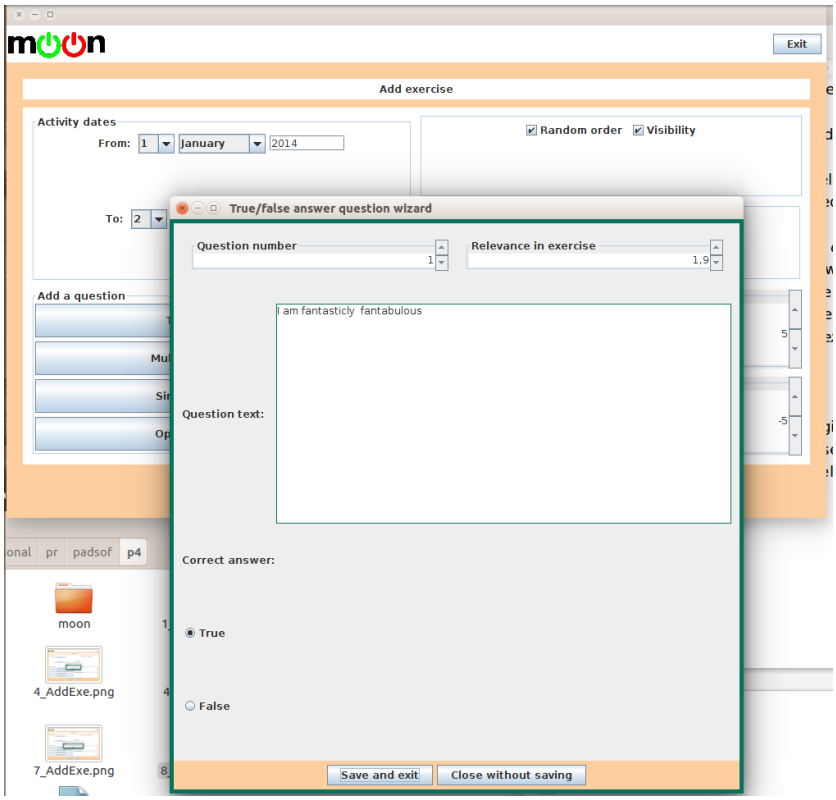
Steps 15, 16: no added questions



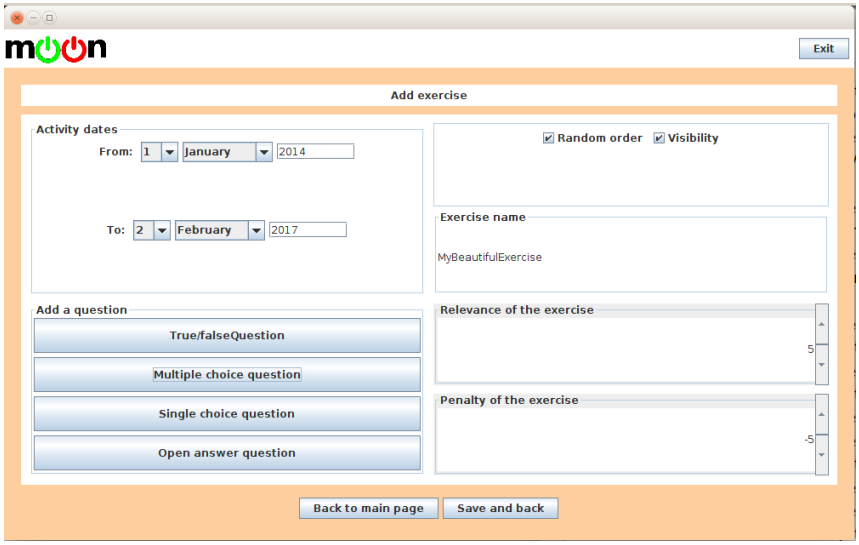
Step 17, 18: add T/F question



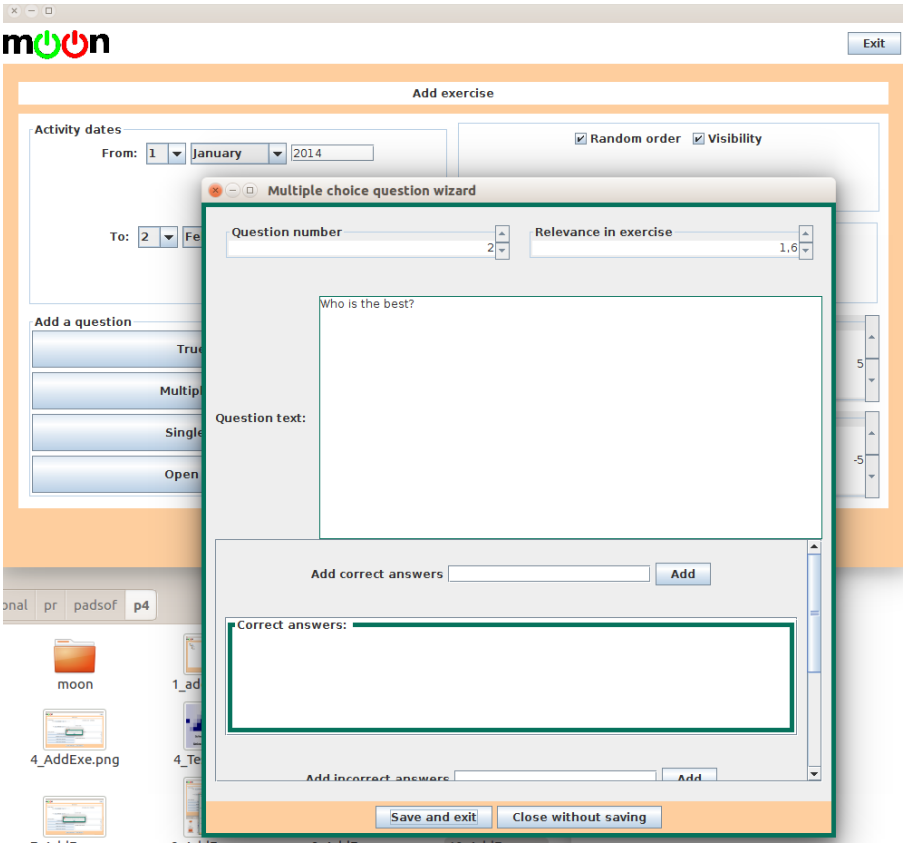
Step 19, 20: user fills fields, saves



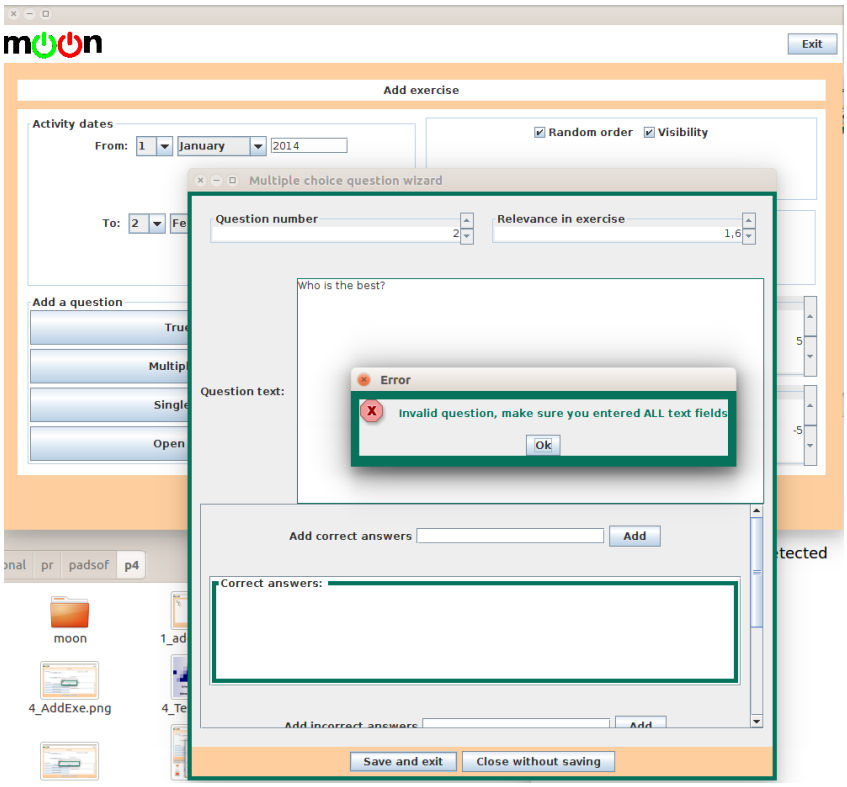
Steps 21, 22: pop up closed, presses multiple choice question.



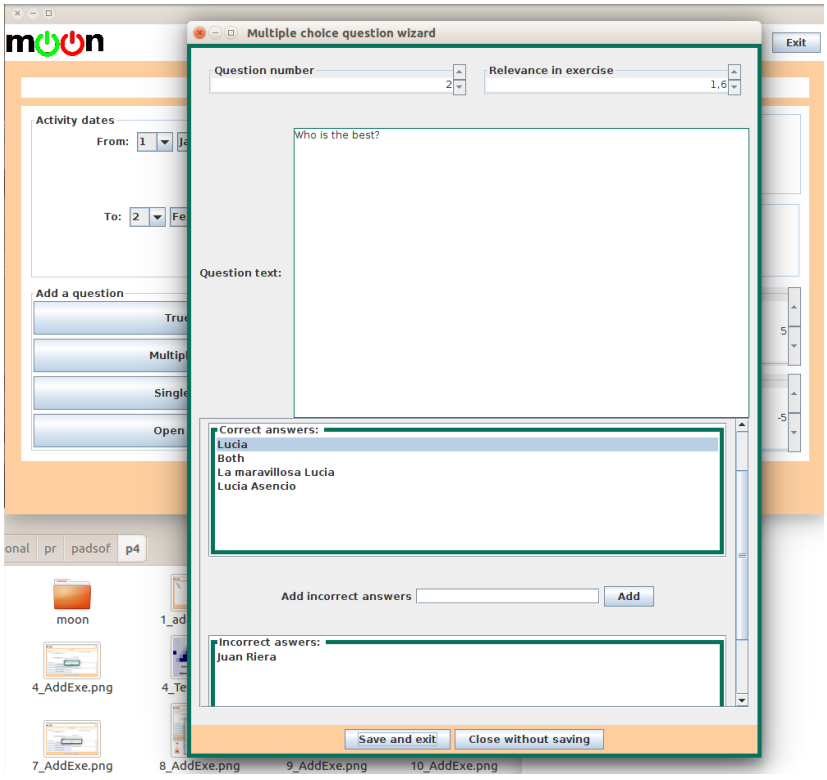
Steps 23, 24: fill some fields in pop up, tries to save



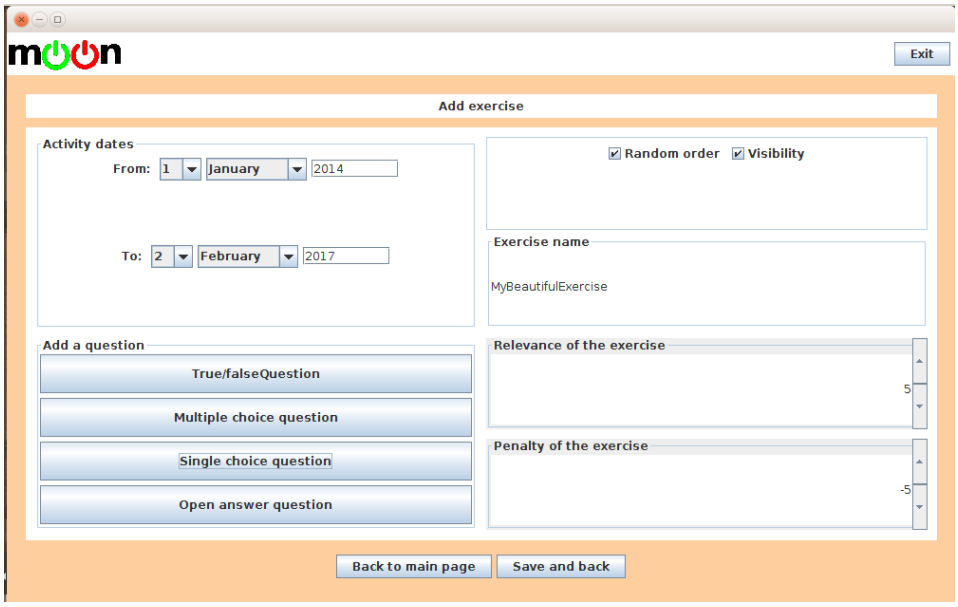
25, 26: no added answers detected



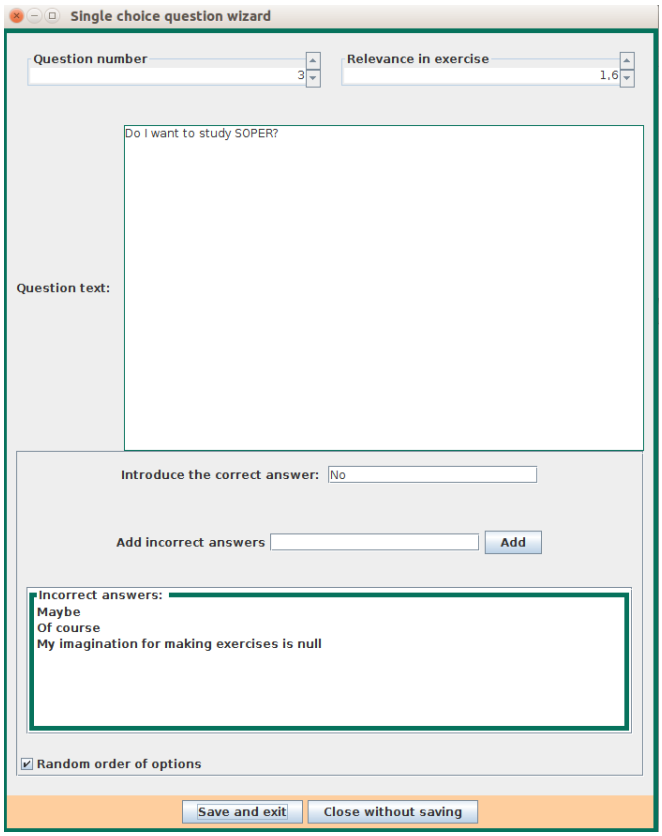
Steps 27, 28, 29: fills the other fields, saves



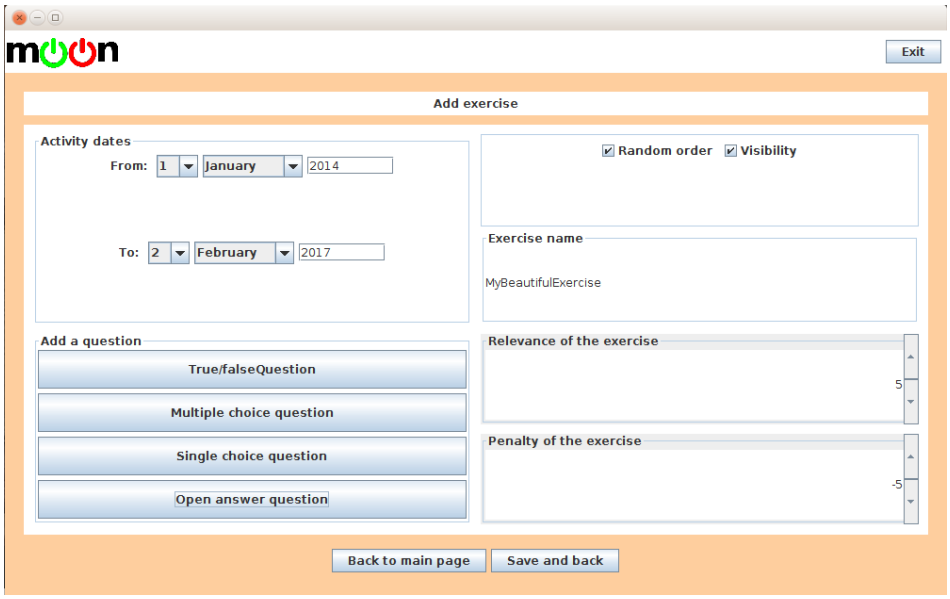
Step 30: user selects single choice questions



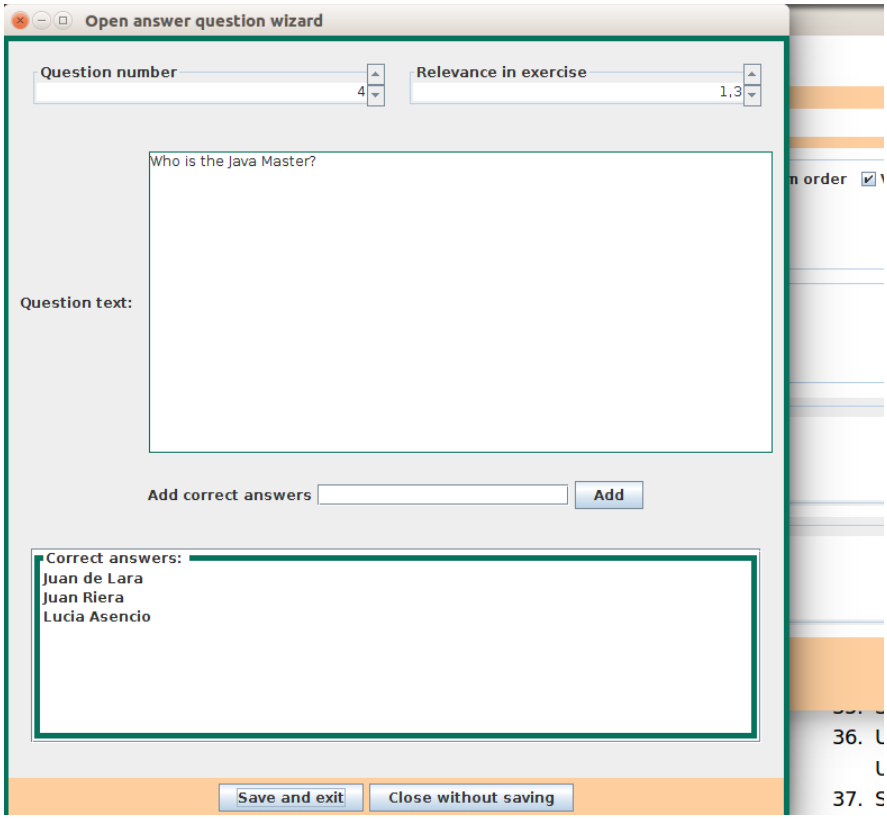
Step 31, 32, 33: Fill all the fields and save



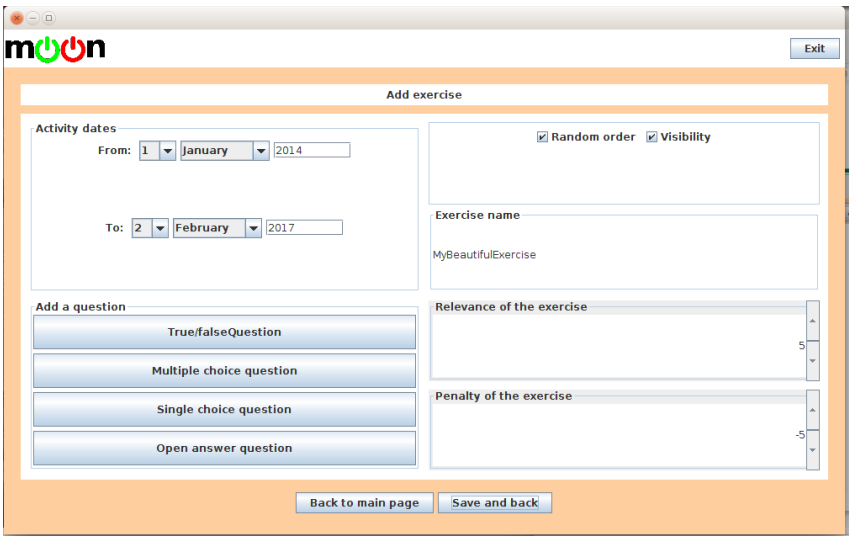
34: Selects open answer questions



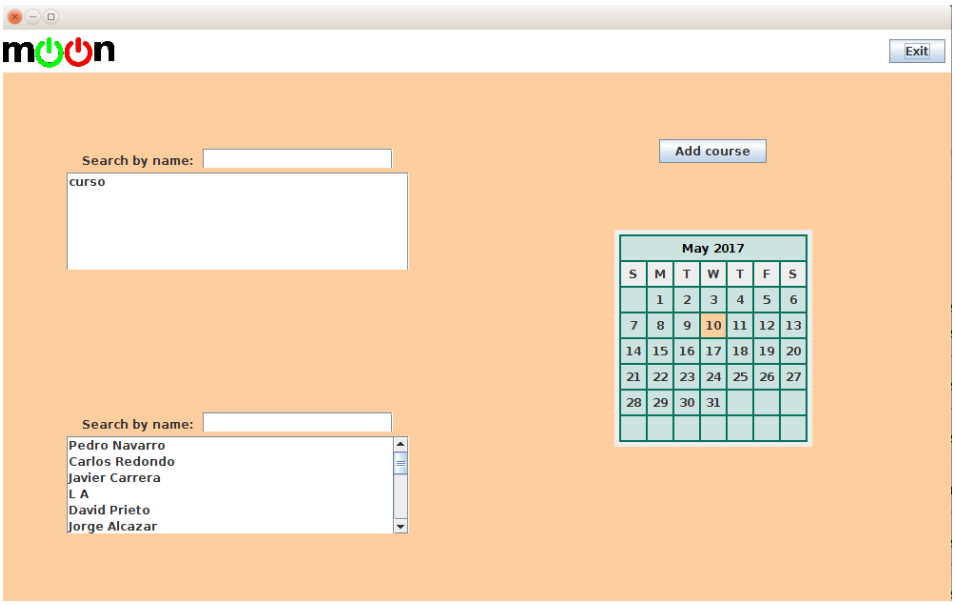
35, 36, 37: fill fields and save



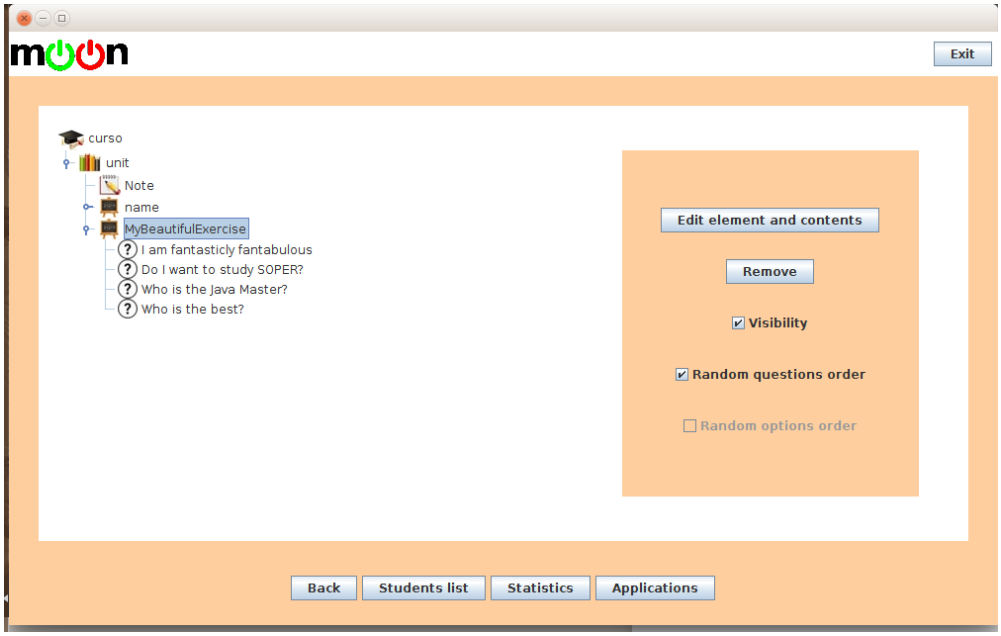
Step 38. saves exercise



Step 39, 40: main page, selects curso



Step 41: opens tree and checks exercise was correctly added.



The results in the test case are exactly the same as the ones in the result of the test, as the screenshots show.