

LESSON: Testing of the lessonproject (<http://lessonproject.origo.ethz.ch>)

Group Name on Origo: lesson-softeng-prj-sl

Luca Mazzoni 765813 luca.mazzoni@mail.polimi.it

Sylvain Lemoine 763951 sylvain.lemoine@mail.polimi.it

Adriana Vargas Sanchez 755917 adriana.vargas@mail.polimi.it

A.A. 2010-2011

1 Introduction.....	1
2 Testing strategy	1
2.1 Testing strategy & Test documentation structure.....	1
3 Testing results	2
3.1 Installation testing results	2
3.2 Functional tests results.....	2
(3.3.1) Use cases for actor administrator	2
Name Register Trainer.....	2
Register Trainee	3
Register Administrator	3
(3.3.2) use cases for an actor not logged in the system.....	4
Log on	4
(3.3.3) Use cases for checker.....	4
Mark Test	4
(3.3.4) Use cases for trainee	5
Search for a Course	5
Log Out	5
View Profile Trainee	6
View Course.....	6
Request Join Course	7
Download material	7
Do test.....	7
Change Profile	8
(3.3.5) Use cases for trainer	9
Search for a Course	9

Log Out	9
View Profile Trainer.....	10
Change Profile	10
Create Course.....	11
View Course Profile	11
Manage Course	12
Modify Material.....	12
Create Test	13
Upload Checker	13
View Tests Results	14

1 Introduction

Our testing activity is based on dynamic techniques to obtain information of interest about the tested application by observing some executions. The objective of our testing activity is to verify that the functional specifications are implemented correctly according to the RASD. We have also carried out installation testing, verifying the system upon installation in the target environment.

We have lead a Beta-testing since we are not members of the team which has developed the application thus creating our own environment, selecting our data, and determining what functions, features or tasks to explore.

Our testing activity can be seen as a conformance/functional testing as it is aimed at validating that the observed behavior conforms to the specifications. In particular, it checks whether the implemented functions are as intended, and provides the required services and methods.

2 Testing strategy

2.1 Testing strategy & Test documentation structure

As we carried out a specification based testing, we derived our reference model from the documentation relative to program specifications (RASD), looking at the Input/Output relation of the program seen as a “black box”. Each test is associated to one use case in the RASD. Each paragraph of chapter 3 corresponds to a specific actor of the system and its name begins with the number of the corresponding sub-paragraph of the RASD. For each actor, we have tested all the corresponding use cases.

This document can be seen as a test log since it presents the results of the tests execution and as a test incidents or problem report as it provides a description of the incidents including inputs, expected, obtained results and anomalies.

3 Testing results

3.1 Installation testing results

Following the user manual released with the server-side application and client-side application, we were not able to run the application at the client-side. Nevertheless, we successfully deployed the server-side application. We have solved the problem with the client-side application by importing the source code into the Eclipse IDE and running the application client with Eclipse.

3.2 Functional tests results

(3.3.1) Use cases for actor administrator

Register Trainer

Input data

Name: Laura

Surname : Bissi

Password : password

E-mail : laura.bissi@gmail.com

Expected result

Laura Bissi is registred into system with a user-id and a password and a mail is sent to Laura Bissi.

Obtained result

Registration succeeds. Laura Bissi has been successfully inserted into the database with the informations above. A user-id has been generated for her. No email has been sent to laura.bissi@gmail.com as specified in the user manual (the sent mail part has not been implemented).

Register Trainee

Input data

Name: Sylvain

Surname : Lemoine

Password : password

E-mail : sylvain.lemoine@gmail.com

Expected result

Sylvain Lemoine is registered into the system with a user-id and a password and a mail is sent to his email address.

Obtained result

Registration succeeds. Sylvain Lemoine has been successfully inserted into the database with the information above. A user-id has been generated for him. No mail has been sent to sylvain.lemoine@gmail.com as specified in the user manual (the sent mail part has not been implemented).

Register Administrator

Input data

Name: Luca

Surname: Mazzoni

Password: password

E-mail: luca.mazzoni@gmail.com

Expected result

Luca mazzoni is registered into system with a user-id and a password and a mail is sent to his email address.

Obtained result

Registration succeeds. Luca Mazzoni has been successfully inserted into the database with the information above. A user-id has been generated for him. No mail has been sent luca.mazzoni@gmail.com as specified in the user manual (the sent mail part has not been implemented).

(3.3.2) use cases for an actor not logged in the system

Log on

Preconditions

The trainee Sylvain Lemoine (user-id = 8) is already registered into the system

Input data

user-id: 8

password: password

Expected result

Sylvain Lemoine is logged in and sees his main personal interface.

Obtained result

Sylvain Lemoine is correctly logged in and sees his main personal interface.

(3.3.3) Use cases for checker

Mark Test

Preconditions

The checker with AuC “7507bfq20001” has been created and associated to the test with ID “4” of the “Software Engineering” course.

Input data

It is not possible to insert any input data. It should at least been allowed to insert a grade for each test.

Expected result

The corrected tests with their grades are updated into the database.

Obtained result

No test mark can be assigned so the correction is not possible.

(3.3.4) Use cases for trainee

Search for a Course

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged in by using the trainee client application and is viewing his main window.

Input data

The option “search on course name” is selected
Keyword: Software Engineering

Expected result

The course named “Software Engineering” is returned by the searching engine, and it is shown in the results table.

Obtained result

The course named “Software Engineering” is shown in the results table.

Log Out

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged into the system, and he’s viewing his main window.

Input data

none

Expected result

The application is closed and “Raheel Maqsood Hashmi” is logged out of the system.

Obtained result

The application is closed and “Raheel Maqsood Hashmi” is logged out of the system.

View Profile Trainee

Preconditions

The trainer “Raheel Maqsood Hashmi” is logged into the system, and he’s on the main window of the trainee client.

Input data

none

Expected result

The profile is shown in the main window.

“Raheel Maqsood Hashmi” can see the following information on his profile: e-mail address, name, surname and the list of the taken courses.

Obtained result

The profile is shown in the main window as expected.

View Course

Notes

The use case “View Course” is associated with “View Course Details” function in the main window.

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged into the system and is viewing his profile. “Raheel Maqsood Hashmi” selects one of his taken course in the courses table.

Input data

none

Expected result

The course details are shown including course description and the list of the tests.

Obtained result

The obtained result is the expected one.

Request Join Course

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged into the system and is viewing his profile. “Raheel Maqsood Hashmi” selects one course from the courses he has not already taken.

Input data

none

Expected result

The join request is saved.

Obtained result

The join request has been saved properly.

Download material

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged into the system, and he clicks on the “show materials” button in the course tab.

Input data

He selects a lecture and then a material for the lecture. Then he clicks on the “download” button.

Expected result

The course material has been downloaded on the “Raheel Maqsood Hashmi” device.

Obtained result

The trainee can only view all the materials, but he can’t download it (as specified in the user manual, this functionality has not been implemented).

Do test

Notes

Questions are not displayed in the “Questions” text field area.

Preconditions

The trainee “Raheel Maqsood Hashmi” is logged into the system, accesses the course profile page and clicks on the “take test” button after having selected a test from the tests table.

Input data

“Raheel” inserts the answers for the questions in the “Answer” text field area and then submits the test.

Expected result

The test is submitted.

Obtained result

The test is submitted, but the input data are not saved in the database.

Change Profile

Preconditions

The trainer “Raheel Maqsood Hashmi” is logged into the system, and he has clicked on the “modify profile” function.

Input data

old password: traineePSW

new password: pass

e-mail: test@polimi.it

Expected result

The profile is updated with the input data above.

“Raheel Maqsood Hashmi” sees his updated information in his main window.

Obtained result

The profile is updated with the input data above.

“Raheel Maqsood Hashmi” sees his updated information in his main window.

(3.3.5) Use cases for trainer

Search for a Course

Preconditions

The trainer Carlo Ghezzi is logged in by using the trainer client and he had previously created the course of “Software Engineering”.

Input data

the option “search on course name” is selected
keyword : Software Engineering

Expected result

The course named “Software Engineering” is returned by the searching engine, and it is shown in the results table.

Obtained result

The course named “Software Engineering” is shown in the results table.

Log Out

Preconditions

The trainer Carlo Ghezzi is logged into the system, and he’s on the main windows of the trainer client.

Input data

none

Expected result

The application is closed and Carlo Ghezzi is logged out of the system.

Obtained result

The application is closed and Carlo Ghezzi is logged out of the system.

View Profile Trainer

Preconditions

The trainer Carlo Ghezzi is logged into the system, and he's on the main windows of the trainer client.

Input data

none

Expected result

The profile is shown in the main window.

Carlo Ghezzi can see the following information on his profile: user-id, e-mail address, list of the courses he is in charge of with the relative status of each course.

Obtained result

The profile is shown in the main window with e-mail address, list of the courses he is in charge of with the relative status of each course but the user-id is missing.

Change Profile

Preconditions

The trainer Carlo Ghezzi is logged into the system, and he has clicked on the "modify profile" function.

Input data

old password : trainerpsw

new password : password

e-mail : carlo.ghezzi@polimi.it

Expected result

The profile is updated with the input data above.

Carlo Ghezzi sees his updated information in his main window.

Obtained result

The profile is updated with the input data above.

Carlo Ghezzi sees his updated information in his main window.

Create Course

Preconditions

The trainer Carlo Ghezzi is logged into the system, and he has clicked on the “create course” function.

Input data

Name: AAC

Open date: 2010/03/07

Description: Corso di architetture

Expected result

The course is created and saved in the database.

Obtained result

The course is created and saved in the database.

View Course Profile

Notes

The use case “View Course Profile” is associated with “View Course Details” function in the main window.

Preconditions

The trainer Carlo Ghezzi is logged into the system and is viewing his profile. Carlo Ghezzi selects one of his course in the courses table.

Input data

none

Expected result

The course profile is shown including course name, course description and the list of the tests.

Obtained result

The interaction with the application to view the course profile is not as described in the RASD: what has been called course profile in the RASD is called course detail in the trainer client application. But even if the interaction is different, the obtained result is the expected one.

Manage Course

Notes

The use case “View Course Profile” is associated with “View Course Details” function in the main window.

Preconditions

The trainer Carlo Ghezzi is logged into the system and is viewing the course profile (course details) of the “Software Engineering” course.

Input data

none

Expected result

Carlo Ghezzi has rejected one or more trainee’s requests, and/or has enabled/disabled the course.

Obtained result

All the functionalities specified in expected results above work correctly.

Modify Material

Preconditions

The trainer Carlo Ghezzi is logged into the system and is viewing the course profile (course details) of the “Software Engineering” course.

Input data

none

Expected result

Carlo Ghezzi has created/deleted one or more lectures and added/deleted one or more materials to a lecture.

Obtained result

All the functionalities specified in expected results above work correctly.

Create Test

Preconditions

The trainer Carlo Ghezzi is logged into the system and is viewing the course profile (course details) of the “Software Engineering” course.

Input data

template file: test001.txt

due date: 2010/01/02 13:43:50

questions:

“1- What is SE?”

2- Who is Carlo Ghezzi?”

Expected result

The test has been created with the specified input data and has been added to the “Software Engineering” course tests.

Obtained result

As expected the course has been created and added to the “Software Engineering” course tests.

Upload Checker

Preconditions

The trainer Carlo Ghezzi is logged into the system and is viewing the course profile (course details) of the “Software Engineering” course. A test with ID “2” has been added to the course tests.

Input data

Carlo Ghezzi clicks on “Add checker” button. He chooses the “Software” radio button and types “http://myServer/Tests/Checker001.jar” in the “Software Checker Location”.

He selects the test with ID “2” from the “Tests” table and clicks on “Assign Checker” button.

Then he selects the previously created software checker.

Expected result

The software checker is created, and associated to the test with ID “2”.

Obtained result

The software checker is created, and associated to the test with ID “2”.

View Tests Results

Preconditions

The trainer Luca Breveglieri is logged into the system and is viewing the test tab. A course named “Formal languages and compilers” has been created with ID “3”. A test with ID “4” and a test with ID “5” have been added to the course “Formal languages and compilers” tests. A trainee has taken both of them.

Input data

By course radio button is selected and in the corresponding text field, Luca writes 3.

Expected result

The test with ID “4” and the test with ID “5” are displayed in the results table.

Obtained result

Only the test with ID “4” is displayed in the results table.