

Use Cases group 24

Eric Druid (edruid@kth.se)
Erika Ekberg (eekeberg@kth.se)
Mattias Johansson (mattias4@kth.se)
Enault Adrien-Marie (enault@kth.se)

27/11/2007

1 Use cases

1.1 Adding a course through the web interface

1.1.1 Primary Actor

Teacher

1.1.2 Stakeholders and Interests

Teacher: A fast and easy way to add a course to the res system.

1.1.3 Preconditions

The user has got a teachers-account in the res system, and have a computer with internet available.

1.1.4 Success Guarantee (postconditions)

A new course is available for students to join, and for assistants and teachers to set grades in.

1.1.5 Main Success Scenario (or Basic Flow)

- 1 The user type the URL to the res systems web interface.
- 2 The system ask for his account name and password, over an encrypted connection so that noone else can read the log in information.
- 3 The user verifies the log in information
- 4 The system display the main menu
- 5 The user chooses the function he wants to use, for instance 'Add course'
- 6 The system present a form that the user should fill in. Where the user can supply name of the course, the number of elements and their respective name/available grades.
- 7 The user finish the creation of the course by clicking the create button in the form.
- 8 The system display a receipt for the new course
- 9 The system display the main menu to the user
- 10 The user chooses 'Exit', and the system exits

1.1.6 Extensions (or Alternative Flows)

3a. The user supply wrong log in information, and have to retype the accountname and password. If this happens tree times the IP will be temporary banned, to prevent brute force attacks.

7a. Some vital fields were not filled in, the site will ask the teacher to either fill these in or to cancel the creation.

1.1.7 Special Requirements

The user must have a teachers account in the res system to be able to add courses

1.1.8 Technology and Data Variations List

- There is no special technology needed.

1.1.9 Frequency of Occurrence

Once at the beginning of a new course

1.1.10 Open Issues

None

1.2 Reporting a students grade

1.2.1 Primary actor

Teacher or Lab assistant

1.2.2 Stakeholders and Interests

Lab assistant/Teacher: wants fast entery of grades.

Student: wants his/her grades reported for when the assignment was turned in.

Course Administrator: wants the assignments graded.

1.2.3 Preconditions

The teacher/assistant is logged on to a school computer.

1.2.4 Success Guarantee (postcondissions)

The grades are saved.

1.2.5 Main success scenario (or Basic Flow)

- 1 assistant or teacher (user) starts the program.
- 2 user selects "grade"
- 3 system displays a list of courses the user can grade in.
- 4 user selects course
- 5 user enters the date he recieved the assignments
- 6 user locates the first student to be graded
- 7 user sets the grade
- 8 system updates the database, and logs the grading.
- 9 system sends an update to all other clients logged on to the same course
- 10 user repeats steps 6-9 until the end of his list of students.
- 11 user requests logout
- 12 system presents user with a list of users he has graded and asks if he wants to change anything
- 13 accepts the list and logs out

1.2.6 Extensions (or Alternative Flows)

- 2a The user may specify that he wants to grade on the commandline
- 3a The user may specify course on commandline
- 7a The student has recived a grade from another teacher/assistant since the student was last updated in the client.
 - 1 system prompts user about the conflict and asks what the grade should be.
 - 2 user chooes whether to keep the other grade or enter an own grade.

1.2.7 Special requirements

None

1.2.8 Technology and Data Variations List

None

1.2.9 Frequency of Occurrence

Comes in batches where all assistants on a course report the same assignment all at once.

1.2.10 Open issues

None

1.3 Register in course

1.3.1 Primary actor

Student

1.3.2 Stakeholders and Interests

Student: wants to register in the course so he/she can get and see his/her grades.

Course Administrator: wants students to register so he/she knows how many are taking the course.

1.3.3 Preconditions

The student is a student at KTH.

1.3.4 Success Guarantee (postconditions)

The grades are saved.

1.3.5 Main success scenario (or Basic Flow)

- 1 student arrives at the res home page
- 2 student logs in
- 3 system prompts the user to fill in his/her personal information
- 4 student fills in the form and submits
- 5 system adds the data to the database
- 6 system shows a view of all courses he/she is registered in and some extra menus
- 7 user chooses "new course"
- 8 system prompts for a course id and offers a search tool

- 9 user finds and submits the course id
- 10 system stores the data
- 11 system shows the lists of courses the student is registered in
- 12 student logs out

1.3.6 Extensions (or Alternative Flows)

3a If the user already has entered his/her personal information earlier the system skips to 6.

9a Wrong course code

1 system rejects the entry and asks the student to try again

1.3.7 Special requirements

None

1.3.8 Technology and Data Variations List

2a Users might be using different login systems in different setups.

1.3.9 Frequency of Occurrence

A thousand a day at most.

1.3.10 Open issues

None

1.4 Time reporting

1.4.1 Primary Actor

Teacher assistants

1.4.2 Stakeholders and Interests

Teacher Assistants: A fast and easy way to report their working-hours.

1.4.3 Preconditions

The user has got a assistant-account in the res system, and have a computer with Internet available.

1.4.4 Success Guarantee (postconditions)

The assistants working time is successfully reported to the system and the teacher can access the information to sort out the payments.

1.4.5 Main Success Scenario (or Basic Flow)

- 1 The user type the URL to the res systems web interface.
- 2 The system ask for his account name and password, over an encrypted connection so that none else can read the log in information.
- 3 The user verifies the log in information
- 4 The system display the main menu
- 5 The user chooses the function 'Report Time'
- 6 The system present a form that the user should fill in. Where the user can supply name of the course and the number of hours he worked.
- 7 The user finish the reporting by clicking the report button in the form.
- 8 The system display a receipt for the new hours
- 9 The system display the main menu to the user
- 10 The user chooses 'Exit', and the system exits

1.4.6 Extensions (or Alternative Flows)

- 1a. The user may want to access using the command line interface if.
- 2a. If the user is already logged on or logged to a school computer he wont be asked his account name and password.
- 3a. The user supply wrong log in information, and have to retype the account name and password. If this happens tree times the IP will be temporary banned, to prevent brute force attacks.
- 7a. Some vital fields were not filled in, the site will ask the assistant to either fill these in or to cancel the reporting.

1.4.7 Special Requirements

The user must have a assistant account in the res system to be able to time report

The assistant must be assigned by a teacher to a course to be able to time report on that course.

1.4.8 Technology and Data Variations List

- There is no special technology needed.

1.4.9 Frequency of Occurrence

Probably the assistants will want to time report at least once a week.

1.4.10 Open Issues

None

1.5 View course results

1.5.1 Primary Actor

Teacher

1.5.2 Stakeholders and Interests

Teacher: The teacher wants to see who completed what laboration in the course

1.5.3 Preconditions

The is the creator of the course that he wants to see a summary of.

1.5.4 Success Guarantee (postconditions)

The user gets his report, and the results are not changed.

1.5.5 Main Success Scenario (or Basic Flow)

- 1 The user starts the program
- 2 If the user is not logged in to a school computer, the system ask for his account name and password.
- 3 The user verifies the log in information
- 4 The system display the main menu
- 5 The user chooses the function 'Course summary'
- 6 The system display a list of courses that the user has created.
- 7 The user choose the course he wants to view.
- 8 The system ask for what filters the user wants to apply, like only students that passed lab. 1 for instance.

9 The system display a list of the students that fit the filter criteria of the selected course.

10 The system display the main menu to the user

11 The user chooses 'Exit', and the system exits

1.5.6 Extensions (or Alternative Flows)

3a. The user supply wrong log in information, and have to retype the account name and password.

4a. The main menu will contain different items depending on the users account. Teachers can see items that a student can not.

1.5.7 Special Requirements

The user must be the creator of the course that he wants a summary of.

1.5.8 Technology and Data Variations List

There is no special technology needed.

1.5.9 Frequency of Occurrence

Afew times near the end of the course.

1.5.10 Open Issues

Other persons than the creator might want to view the summary

1.6 Command line user interface

1.6.1 Primary Actor

Student, Lab. assistant or Teacher

1.6.2 Stakeholders and Interests

Student: A fast way to display grades on laboratory exercises.

Teacher: An easy way to find a student and update his grade. And to add new courses / laborations.

- Laboration assistant: A fast way to upgrade a students grade on a laboratory exercise.

1.6.3 Preconditions

The user is logged in to a school computer, either locally or from home via a ssh client.

1.6.4 Success Guarantee (postconditions)

The requested information is displayed on the screen. The database is updated.

1.6.5 Main Success Scenario (or Basic Flow)

- 1 The user starts the Res2 system by typing 'res2' in the console window.
- 2 The system display the main menu
- 3 The user chooses the function he wants to use, for instance 'Show grades'
- 4 The system asks the student to enter the name of the course he wants to see, and presents a list of courses that he is registered in
- 5 The user select the course he is interested in, such as 'mdk07'.
- 6 The system displays the users grades on the assignments in mdk07.
- 7 The system display the main menu to the user
- 8 The user chooses 'Exit', and the system exits

1.6.6 Extensions (or Alternative Flows)

1a. There are several shortcuts to the most common functions, such as displaying the grades of a particular course, accessible from the command line. For instance "res2 show mvk07" will show the points gained on the assignments in mjukvarukonstruktion 07.

3a. The choises you can do depends on your userlevel. If you are a student you can show course, join a course, display a list of your current courses and leave a course.

1. If you choose show, you will get a list of your current courses, enter the name of the course that you want information about.

1a. If you enter a name of a course that you have not joined, the system will print an error, and you will get back to the main menu.

2. If you choose join, you will have to enter the name of the course that you want to join. If the name you enter doesn't exist, the system will print an error.

3. If you choose to display a list of your courses, the system will print the courses that you have joined on the screen.

4. If you choose to leave a course, the system will display a list of your current courses, and you will have to enter the name of the course that you want to leave. If you enter the name of a course that you are not registered in, the system will print an error.

3b. If you are an assistant or a teacher, you will also be able to report grades on student assignments.

3c. If you are a teacher you also will be able to add new courses, or edit old courses.

1.6.7 Special Requirements

Several users must be able to update grades at the same time. An assistant shouldn't have to lock the database during his update.

1.6.8 Technology and Data Variations List

There is no special technology needed.

1.6.9 Frequency of Occurrence

Could be nearly continuous.

1.6.10 Open Issues

None