Requirements Document group 24

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

03/01/2008

1 Preface

Expected readership of this document

We expect the computer administrators of KTH, in addition to the students and teachers of the MVK course to read this document.

Version history

1.0 - This is the first version of our RD

Rational for creation of each new version

The first version was created to describe the requirements of our project

A summary of the changes made in each version

1.0 - The document was created

Contents

2 Introduction

Users

Our project is to write a student result report system. The users will be the students, the teachers and the assistants at KTH. The students will be able to see if they passed and what grade they got on their labs and other course parts. The teachers will be able to use the system to keep track of all the students grades, and as a way to report the grades to the students.

The current system, res, has several problems some of which are:

Assistants can't report grades for students in the same course at the same times.

Backspace doesn't work.

It could be easear for assistants to find a specific students in a course.

Users are required to enter there personal information for each course.

We also want to make it easier for the students to access the system from home through a web-interface.

The system will also be an aid in the administration of the courses such as keeping track of assistants working hours.

Usage

Anders reporting grades

Anders, a labassistant, tired after a long lab session and a bit stressed because his next lecture starts in 15 minutes, logs on to a computer, runs res report intro09 in his terminal, presses enter a few times to accept the defaults of date and course part. He starts writing the first name of his list of students that passed the lab. He continues typing the name until the list of students that matches is short enough that he can scroll to the student. When has selected the student he presses enter to select it and presses "P", for pass, to set the grade, he then continues with writing the next name. When he is done grading all the students on his list he presses ctrl+Q to quit. He then gets a list of all the students he has graded and is prompted to accept the grades which he does by pressing "Y". He then logs out of the system.

Mikael viewing his status

Mikael, a student who follows the "intnet09" course, wants to see how many labs he has finished, to see how he had done so far. He clicks on the link to the res system on the course's homepage, as he doesn't remember it's address. After logging in, he clicks on "intnet09" course from a list of the courses hes

registered in, then clicks on "Assignments", which lists every assignments due for the courses, and the grade and/or extra points the student managed to get so far. Seeing that he had only done the first laboratory without getting any extra points, he decides to start working a bit. What Mikael likes about the web-based user interface is its clarity, its user-friendly interface, and, of course, the fact he can access to it from his home.

Environment

The web-based UI is to be used from everywhere. The goal here is to enable students and teacher to access the database from home, or wherever they work.

The command-line UI is to be used on any secured UNIX post where the res terminal application is installed, and use the login and password used to log in the post.

Both will let the user know which courses they applied to, if they passed an exam, if the assistant did take in account the fact they have done such assignment, or could return the list of students who have applied to a particular course, etc.

The server running this should be a secure machine having PostgreSQL and Java.

Scope

Res2 will handle results from laboratory exercises, and let users search through these results and display them in a easy-to-use manner. The system will also let teachers add new courses, and add or update their students results.

There will be two ways to access this system, both a command line interface for compatibility with the old system usage, and a web interface to be used either from the terminals at the school or from any Internet browser at home or elsewhere.

We will not implement a non-web graphical interface.

Teachers will be able to do complex queries, such as "Give me a list of all students who passed 4 or 5 of the 8 assignments for this course" or "Give me a list of students who was one or two days late with the first assignment in that course".

Topic	In	Out
Command line user interface	X	
Web user interface	X	
Non-software components		X
Encryption	X	
Remote access	X	
Simultaneous reporting	X	
Getting a list of your results	X	
Obtaining results from old res	X	
Complex queries	X	
Time reporting for assistants	X	
Evaluation of courses		X
Booking examination time		X

Main factors

The system needs to be secure. The system will handle information about students grades and most people don't want that information to be readable for anyone. Also only teacher or assistants should be able to change the grade on someone. Its also important that none except the assistants and the teachers can access the time-reporting part of the system. The system should be easy to use so that it can be used on other sections and not only the computer-section. Also it wont scare the new students: they must not constantly fear to do an error while using the system that could screw up their signing up, or be completely lost, not knowing what to click or what to do. We must also keep in mind that, though quite unlikely, two different personns could be trying to grade the same thing at the same time, and that it must not screw up the data. AS the system is quite critical to a university (as students get their grades throught it), we will have to focus on the reliability of the system. We don't want it to be constantly down.

Technologies and Risks

Technologies

We plan to use Java with a PostgreSQL backend. The web interface will be writen using Java servlets. The web-based interface should work with the common browsers (IE, Firefox, Netscape, Opera...) on any machine. We will use XHTML and CSS to display the pages, so nothing else than an XHTML-compliant browser should be needed (No java machine). The command line interface will be coded in Java, so any machine having Java 1.5 or newer installed on it will be able to run it. During the development, we will use SVN to keep tracks of the changes done by the different members of our team. so when one will open a file he has previously used, he will be

able to know what has changed since then. There is no particular hardware requirement neither client-side (they just have to be able to run a browser or a Java machine), nor server-side (will have to run a pgSQL databse, though).

Risks

Risk	Probability	Impact
Short on time	high	serious
Low of interest from teachers	high	low
The competition is to harsh	moderate	serious
Difficulties to use pgSQL	low	moderate
Difficulties to Java Servlets	moderate	moderate
Comunication failure among the team	moderate	moderate

Some in the team have extensively used mySQL, and feel confident in using pgSQL, as both are quite similar. We have decided using pgSQL instead of mySQL because we think it is safier. If we have problems using pgSQL, we will always be able to replace it by mySQL if really needed. This should not be a real matter, depending on how advanced the project is. We will use SVN to help us limit the risks of lack of commuincation. As everyone will be able to know who has done what, we should be able to avoid unwilling loss of work due to unfortunate changes in a file. We lack of experience in Java servlets. We don't think it should be too hard and we already have started documenting about it, but just in case, we have thought about several possible different options, so the problem should not be too difficult to overcome.

3 Glossary

API	Application Programming Interface. It is a source code interface that an operating system or library provides to support requests for services to be made of it by computer programs.
Database	A structured collection of records or data that is stored in a computer system.
Encryption	The process of transforming information using an algorithm to make it unreadable to anyone except those possessing special knowledge.
End-user	A person who uses a product.
HTML	Hypertext Markup Language. It is the predominant markup language for web pages.
Java	A programming language.
KTH	Royal Institute of Technology. A swedish university based in Stockholm.
Markup language	A language which combines a text and extra information about it.
MVK	Mjukvarukonstruktion, ie Software Engineering. A course taught at KTH.
pgSQL or postgreSQL	A particular type of database.
RD	Requirements Document. This document.
Servlet	API which allows to generate dynamically some web pages.
SVN	Subversion. A version control system which is used to maintain current and historical versions of files such as source code, web pages, and documentation.
UI	User Interface. It is the aggregate of means by which people interact with a particular system.
UNIX	A computer Operating system, as Windows is.
XHTLM	Extensible HyperText Markup Language. It is a markup language that has the same depth of expression as HTML, but also conforms to XML syntax.
XML	A general-purpose markup language.

4 User requirements definition

Functional requirements

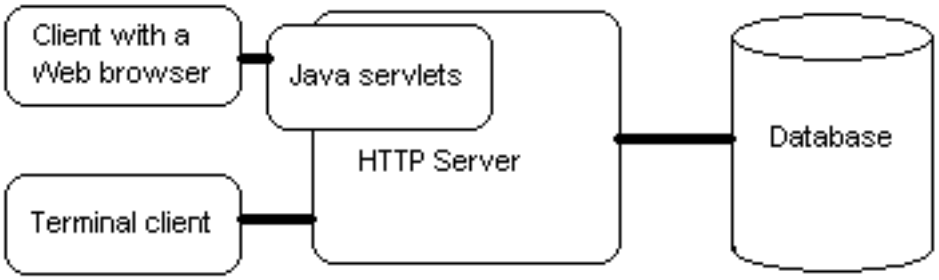
Requirement	Description
Grant access	The system administrator should be able to grant a user teachers-access
Withdraw access	The system administrator should be able to withdraw teachers permissions form a user account
Create assignments	Teachers should be able to create an assignment for their course
Update assignments	Teacher should be able to change/delete an assigment
Access assignments	Teachers should be able to mark any assignment from their course and access these
List students	Teachers should be able to list students in a course
Add a course	A teacher should be able to add a new course
Assign assistants	A teacher should be able to assign assistants to a course
Remove assistants	A teacher should be able to remove assistants from a course
Uptade grades	A teacher or an assistant should be able to update a students' grade
Join a course	A student should be able to join a course
List courses	A student should be able to list his courses
Show grades	A student should be able to see his grades in a course
Leave course	A student should be able to withdraw a course
Create course	A System Administrator should be able to create a course
Validate course	A System Administrator should be able to validate a course
Course creation	A Teacher can submit a request for a course creation

Non-functional requirements

Requirement	Description
Private data	Private data must be securely handled
High uptime	The System must be operational most of the time
Secure	The Database must be protected against Hostiles Actors
Easy to use	Security should not hinder System's global performances
Data safety	The Data must be safe
Stable system	The System must withstand high levels of usage
Quick response	The System must answer quickly

5 System architecture

We will have a program that use a database, and connect to this program with two different clients, one servlet/browser solution, and one terminal program.



6 System requirements description

The system administrator should be able to grant a user teachers-access

The system administrator should be able to tag a user account as teacher, and by that grant him rights to create new courses and assign assistants.

Test

Tag a user account as teacher, and try to do things that only a teacher can, like creating a new course.

Expected errors

The user does not exist, and the system will say so.

The user is already a teacher, the system will say so and not change his permissions.

The system administrator should be able to withdraw teachers permissions form a user account

The system administrator should be able to withdraw teachers-permissions from a user account.

Test

Withdraw the permissions, and after that attempt to create a new course. This should fail.

Expected errors

The user does not exist, and the system will say so.

The user isn't a teacher, the system will say so and not change the account permissions.

Teachers should be able to create an assignment for their course

Prior marking an assignment in an existing course, it must be created. It is up to the teacher to decide wich are the assignements for his course, so he should be able to create them via res, give it a unique name, a description and several other details.

Test

Create a course and name a teacher for this course. Try to create an assignment for this course.

Expected errors

The assignment's name is already used for this course. If so, the system won't do anything the action and notify it to the user.

The course does not exist (the user has mistyped its name or the course has just been deleted). If so, the system won't do anything the action and notify it to the user.

Teacher should be able to change/delete an assigment

Once an assignment has been created, a teacher should be able to change all the details about the assignment (maybe he decided to change it) or decid to delete it.

Test

Once an assignement has been created, try to change it and access the assignment, then check if the changes has been made.

Expected errors

The assignment has been mistyped or does no longer exist. Then, do nothing and notify it to the user.

Teachers should be able to mark any assignment from their course and access these

Once a course has been created as well as an assignment in it, the teacher should be able to give a mark to each student registered in the course. The marks will then be stored, and visible to the teacher.

Test

Once a course has been created with an assignment and at least one student is registered, try marking an assignement for a student then try accessing the mark.

Expected errors

The assignment or the course does not exist (mistyped or deleted), or the student isn't registered to the course. If so, the system won't do anything the action and notify it to the user.

The assignement has already been marked. If so, return the current mark and ask confirmation.

Teachers should be able to list students in a course

In an existing course with registered students, a teacher should be able to get a list of the students of the course matching simple criteria, such as who hasn't been marked for a particular assignment.

Test

We will have to create a course with several students registered. Then test listing. If we add possible criteria for the research, we will have to populate the course with students relevantly matching or not these criteria, and check if we have the expected result.

Expected errors

The criteria match with no students, or the course has been mistyped/deleted. Simply notify to the user.

The criteria has been mistyped. Notify the user and underline what the program has not understand in the research.

A teacher should be able to add a new course

A user with teachers permissions should be able to create a new course. However, the system administrator will have to approve the course before it is visible to the students.

Test

Attempt to create a new course, and see if the system administrator can list this for approvement.

Expected errors

The user does not have teachers permissions. The system will not let the user create a new course.

A course with this name does already exist, the system will prompt the user for a new name.

A teacher should be able to assign assistants to a course

A teacher should be able to add users to the list of assistants to any course that the teacher has created.

Test

Attempt to assign a user as assistant in the course.

Expected errors

The teacher is not the creator of this course, only the creater or a course or the system administrator can assign assistants.

The user that we attempt to add as assistant does not exist, and the system will say so.

A teacher should be able to remove assistants from a course

A teacher should be able to remove assistants from the courses that he created.

Test

Attempt to remove a user from the list of assistants, and then let the fomer assistant attempt to update some students grade. He should not be able to do this anymore.

Expected errors

The user isn't an assistant in this course, the system will say so.

The teacher is not the creater of the course, and hence can not alter the list of assistants.

A teacher or an assistant should be able to update a students' grade

The teacher or one of the assistants in the given course should be able to update the grade on an assigment handed in by a student.

Test

Attempt to update the grade of a student and see if this succeeds.

Expected errors

The user trying to do the update is not a teacher or assistant of the given course. The system will refuse to update the grade.

The student that handed in the assigment is not registered at the course. The system will complain about this and not update any grade.

A student should be able to join a course

A student should be able to join an existing course

Test

Attempt to join an exsting course, and see if this succeeds.

Expected errors

The course does not exist, the system will tell the student.

The user is already registered in this course, the system will say so.

A student should be able to list his courses

A student should be able to get a list of the courses he is currently registered in.

Test

Attempt to retrieve such a list, and see if it succeeds.

Expected errors

The student isn't registered to any courses. The system will let him know this.

A student should be able to see his grades in a course

The student should be able to see his grades on a specific course.

Test

Attempt do show the grades of the assignments in a course of his choise, and see if this succeeds.

Expected errors

The user isn't registered to any courses. So he can't select the course he want to show the grades in.

The user has no grades in the course, the system till let him know that he hasn't passed any assignment.

A student should be able to withdraw a course

If, for a reason or another, a Student wishes to give up with a course, he should be able to do so.

Test

Once a Student is logged in and has signed in to a course, try to withdraw it.

Expected errors

The Student tries to withdraw a course which does not exist or to which he has not signed in. Then the System does nothing and notifies it to the Student.

A System Administrator should be able to create a course

The System Administrator is the only one who can create directly a course. This will ensure that the a better data maintenance (The less creater there are, the less chaos there is.)

Test

A System Administrator who is logged tries to create a course. Then we check if the course is created.

Expected errors

A course with the same name is already created. Then the Sytem does nothing and notifies it to the user.

A System Administrator should be able to validate a course

A Teacher should be able to request for a course creation, to help the Administrator, who ultimately is the only one who can create them.

Test

A System Administrator who is logged tries to validate a course request previously made by a Teacher. Then we check if the course is created.

Expected errors

A course with the same name is already created. Then the Sytem does nothing and notifies it to the user.

A Teacher can submit a request for a course creation

To help the Administrators to create all the courses which are required, a Teacher will be able to create a temporary course which will be available to no one but the teacher and the Administrators, that an Administrator will just have to validate to create it (after having checked it is OK), or to withdraw (which will destroy the request).

Test

A Teacher who is logged tries to create a course creation request. Then we check with an Adminstrator if he can have access to it.

Expected errors

A course with the same name is already created. Then the Sytem does nothing and notifies it to the user.

Private data must be securely handled

Private data such as login and password must be safely stored and sent by the System from and to the database, so these datas won't be easily stolen.

The System must be operational most of the time

A small amount of maintenance shall be required. We will have to ensure that the up time can be at least 23h a day.

The Database must be protected against Hostiles Actors

Hostiles Actors may want to hack into the system to change marks, add/remove users or courses, change permissions, etc. The System must be resilient to the most standard attacks, and the Administrator should be able to easily undo any harm.

Security should not hinder System's global performances

Even if security is important, we must keep in mind that we are not dealing with the most sensitive kind of data. Ease to use, speed and functionality should not be too heavily penalized by security.

The Data must be safe

In most cases of hardware failure, no data should be lost.

The System must withstand high levels of usage

The System should be able to handle 50 requests by minute, which should ensure no crash in a school such as KTH.

The System must answer quickly

A request should be handled in less than 5 secs by the system even if it is under heavy use.

7 System evolution

Fundamental assumptions on which the system is based

The system require a SQL server to be available, and that the computer on which the system execute has got a Java interpreter. It also require that the end-user has got access to a terminal or a webbrowser with Internet access.

Specify anticipated changes due to

Hardware evolution

Hardware evolution should not affect the system, since we will not code it with hardware specific code.

Changing user needs

New user needs might arise in the future, such as changed system for grades. BUt since the grades-system was just updated we don't expect this to happen in the near future.

8 Appendices

Use cases

Add a course through the web interface

Primary Actor

Teacher

Stakeholders and Interests

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

Student: Want to be able to see the course in the res system.

Lab assistant: Want to be able to report the result on a course in the res system.

Preconditions

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

Success Guarantee (post conditions)

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

Minimal Guarantee (post conditions)

Attempting to add a new course will not affect existing courses.

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.

3 The system 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 user imputs course information.

7 The user finish the creation of the course by selecting create course.

8 The system display a receipt for the new course

Extensions (or Alternative Flows)

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 teacher to either fill these in or to cancel the creation.

Special Requirements

The user must have a teachers account in the res system to be able to add courses. The connection need some kind of encryption so that noone can read the log in information.

Technology and Data Variations List

- There is no special technology needed.

Frequency of Occurrence

Once at the beginning of a new course

Open Issues

None

Report grade

Primary actor

Teacher or Lab assistant

Stakeholders and Interests

Lab assistant/Teacher: wants fast entry of grades.

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

Course Administrator: wants the assignments graded.

Preconditions

The teacher/assistant is logged on to a school computer. The course need to be in the system.

Success Guarantee (postconditions)

The grades are saved.

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 received 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

Extensions (or Alternative Flows)

2a The user specifies that he wants to grade on the commandline

3a The user specifies course on commandline

7a The student has received 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 chooses whether to keep the other grade or enter an own grade.

Special requirements

None

Technology and Data Variations List

None

Frequency of Occurrence

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

Open issues

None

Register in course

Primary actor

Student

Stakeholders and Interests

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

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

Preconditions

The student is a student at KTH.

Success Guarantee (postconditions)

The student is registered in the course.

Main success scenario (or Basic Flow)

1 student starts the system.

2 student logs in

3 system prompts the student 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

7 the student chooses "new course"

8 system prompts for a course id and offers a search tool

9 the student 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

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

Special requirements

None

Technology and Data Variations List

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

Frequency of Occurrence

A thousand a day at most.

Open issues

None

Report working hours

Primary Actor

Teacher assistants

Stakeholders and Interests

Teacher Assistants: A fast and easy way to report their working-hours. Teacher: Wants to see how many hours an assistant have spent working.

Preconditions

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

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.

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 user inputs course name and hours spent working.

7 The user finish the reporting by clicking the report button in the form.

8 The system display a receipt for the new hours

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.

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.

Technology and Data Variations List

- There is no special technology needed.

Frequency of Occurrence

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

Open Issues

None

View course results

Primary Actor

Teacher

Stakeholders and Interests

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

Preconditions

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

Success Guarantee (postconditions)

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

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

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.

Special Requirements

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

Technology and Data Variations List

There is no special technology needed.

Frequency of Occurrence

A few times near the end of the course.

Open Issues

Other persons than the creator might want to view the summary

Command line user interface

Primary Actor

Student, Lab. assistant or Teacher

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 / assignments.

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

Preconditions

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

Success Guarantee (postconditions)

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

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

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 choices 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 no 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.

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.

Technology and Data Variations List

There is no special technology needed.

Frequency of Occurrence

Could be nearly continuous.

Open Issues

None