

Course Information Management System

Group 2

David Chang
Linda Chowdhury
Oscar Fitinghoff
Patrik Parberg
Tomas Hansson

Functional requirements

Course Description

- Course leaders shall be able to create, update and remove course description.
- Course leaders, course assistants and students shall be able to view course description.

Schedule

- Course leaders shall be able to import schedule from iCalendar format.
- Course leaders shall be able to create, update and remove scheduled activities.
- Course leaders, course assistants and students shall be able to view the schedule.
- Course leaders, course assistants and students shall be able to export the schedule to iCalendar format.

Deadlines

- Course leaders shall be able to create, update and remove deadlines for activities in specific courses that they are responsible for.
- Course leaders, course assistants and students shall be able to view deadlines.
- Students shall be able to get an overview of deadlines for the courses they are registered for.

Course Registration

- Course leaders shall be able to view registered students for specific courses that they are responsible for.
- Course leaders shall verify students' registrations before the students are accepted in the specified course.
- Students shall be able to register to courses to confirm their attendance of the course (after which the course leader has to accept them before they are entered into the course).

Course News

- Course leaders shall be able to create, update and remove course news for courses they are responsible for.
- Course leaders, course assistants and students shall be able to view course news.
- Students shall be able to view RSS-feeds for news from courses the student is registered for.
- Students shall be able to get an overview of course news on the students' own personal page from all the students' registered courses.

Uploaded Files

- Course leaders and course assistants shall be able to upload, delete and update files.
- Course leaders, course assistants and students which are registered in the course shall be able to view and download uploaded files.

Creation Guide for Course Website

- There shall be a creation guide for the course leader that will help them create a course website.

Authentication System

- Course leaders, course assistant, students and system administrators shall be able to identify themselves using a log in system.

Miscellaneous

- System administrators shall be able to create, update and remove course leaders, course assistants and students.
- System administrator shall be able to create, update and remove courses.
- Course leaders shall be able to create and remove course assistants.

Non-functional requirements

Product Requirements

Availability – The system shall be available for 22 hours of a 24 hour period with one request per minute for the personal student pages.

Performance - Page generation (not including response time or the time required to transfer the data to the client) in less than 5 seconds, when there are 10 students trying to access their personal pages simultaneously.

Scalability – If the amount of concurrent requests for the news page of a course website increases from 10 to 100, the page generation time should at most increase by a factor of three.

Organisational Requirements

Implementation – The actual site will be implemented in JavaServer Pages (JSP), with a MySQL database backend. This requirement will be fulfilled if the actual JSP code compiles without errors, and the SQL queries to the MySQL database don't return query errors when the queries are run.

External Requirements

Security – Users who don't have the required privileges shall not be able to create, edit or delete information in the system.

Privacy – Students shall only be able to view their own results, the system shall not allow students to view results that belong to other users. We intend to test this by making sure that a user that isn't authenticated can't access any results by trying to access a URL directly, and that an authenticated user cannot access other students' results by trying to access a URL directly.

Use Cases

Use Case UC1: User Authentication

Primary actor: Course leader, course assistant, student and system administrator

Stakeholders and Interests:

- *Course leaders:* Want to make sure that only course leaders can add, edit and delete content.
- *Students:* Want to be certain that they can view their results.
- *Course assistants:* Want to be able to register results for the courses they are assigned to.
- *University:* Has interests in much the same as the course leaders, making sure that unauthorized persons can't add inappropriate content.
- *System administrators:* Want to make sure that unauthorized users can't make changes to information in the system.

Preconditions: The user is using a computer with a working Internet connection, has started the web browser Mozilla Firefox and has navigated to the system.

Success Guarantee: The user is authenticated and the system can identify the user.

Main Success Scenario

1. The user selects the option to log in.
2. The system displays a form asking the user to input:
 - Username
 - Password
2. The user enters her username and her password.
3. The system validates the user's login details.
4. The system creates a session with the user and stores data necessary for identification, such as cookies.
5. The system redirects the user to her personal page.

Extensions

- *a. At any time, the system fails.
 1. The course leaders authenticates again.
 - 1a. The course leader is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the course leader will have to try again another time or contact the system administrator.
 - 3a. The username entered doesn't exist or the password entered doesn't match the username.
 1. The system displays a message telling the user that the user information didn't validate. The username field contains the username that the user entered and the password field is blank.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times per week and user.

Open Issues: None.

Use Case UC2: Create course website

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want a course website that is informative, simple to create and maintain.
- *Students:* Want a course website that is informative, available, fast to access and simple to navigate.
- *Course assistants:* Want to view course news and schedule on the course website.
- *University:* Wants a course website that satisfies the students' needs.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: A course website is created with course description, schedule and the ability for students to register for the course. The course website shall be maintainable through the system.

Main Success Scenario

1. The user starts the creation guide for the course website.
2. The system displays a form asking the user to input:
 - A course name to be used to identify the course.
 - The course's credits.
 - The period and year when the course starts.
 - A textual description of the course.
3. The user fills out the form.
4. The system validates input.
5. The system asks the course leader whether she wants to import a schedule using the iCalendar format.
6. User selects to not import a schedule
7. The system presents a summary of the information that the user has entered and asks for confirmation.
8. User confirms the information entered.
9. The system creates the course website.
10. The system displays a confirmation that the course website has been created.

Extensions

- *a. At any time, the system fails.
 1. The course leader authenticates again.
 - 1a. The course leader is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the course leader will have to try again another time or contact the system administrator.
 - 4a. Value in some field is not correctly entered.
 1. The system displays the form once again with the data entered by the user and points out which fields don't validate.
 2. The user enters new values in the fields and retries sending the form.
 - 5a. The course leader provides the system with an iCalendar file.
 1. The system imports the schedule.
 - 8a. The user doesn't confirm the entered information.
 1. The system displays the form allowing the user to make changes to the entered information.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: One time for each course. More often in between periods when new courses start.

Open Issues: None.

Use Case UC3: Managing Course Schedule

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want a schedule that is simple to manage.
- *Students:* Want an accurate schedule so that they know when activities are held.
- *Course assistants:* Want an accurate schedule so that they know when activities are held.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: The schedule, including the new additions and changes made by the course leader, is available to all students and course assistants. The changes are reflected in the compiled schedules of any students currently registered in the course.

Main Success Scenario

1. The user selects the option to import a schedule.
2. The system displays a form containing fields for:
 - A file containing an iCalendar representation of the schedule to be imported.
3. The user provides the schedule to import.
4. The system imports the schedule.
5. The system confirms that the schedule was imported.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.

1a. The user is authenticated and can start working from scratch.

1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

1a. Edit schedule activity.

1. The user selects the option to edit the schedule.

2. The system displays a list of all activities currently in the schedule, and asks the user to select the one to be edited.

3. The user selects an activity.

4. The system displays a form asking the user to input:

- Title.
- Description.
- Date.
- Time.

5. The user inputs new values.

1a. The new values were invalid (wrong format, too long, unparseable etc.).

1. The system displays the form and tells the user of the problem, and asks the user to correct the problem.
2. User fixes the problem.

6. The system displays the new values and asks the user to confirm them.
7. User confirms new values.
8. System returns to step 2, and displays a confirmation message that the edit was successful.

- 2a. The schedule provided was not in an iCalendar compatible format.
 1. The system displays an error telling the user about the problem.
 2. The user has the option to start over.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Once at the creation of each new course website, then very rarely.

Open Issues: None.

Use Case UC4: Update Course Description

Primary Actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Update the course description.
- *Course assistants:* Want an accurate course description.
- *Students:* Want an accurate course description.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success guarantee: The changes are saved and the course description on the website is updated.

Main Success Scenario

1. The user selects the option to update the course description for a specific course.
2. The system displays a form with the current course description containing fields for:
 - A course name to be used to identify the course.
 - The course's credits.
 - The period and year when the course starts.
 - A textual description of the course.
3. The user performs the desired changes to the course description.
4. The user saves the changes.
5. The system validates the input.
6. The system saves the changes.
7. The system displays a confirmation that the changes have been saved.

Extensions

- *a. At any time, the system fails.
 1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
- 5a. The input doesn't validate (too long, not numeric where it should be etc).
 1. The system displays the form (step 2) and highlights the input that didn't validate
 2. The user makes changes to the data and resubmits.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Once or a few times at the start of each course, then very rarely.

Open Issues: None.

Use Case UC5: Manage Information Pages

Primary actor: Course leader.

Stakeholders and Interests:

- *Course leaders:* Want a course website with informative pages.
- *Course assistants:* Want a course website with up-to-date information.
- *Students:* Want a course website with up-to-date information.
- *University:* Wants a course website with information that satisfies the students' needs.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: The changes are saved and the information pages on the website are updated.

Main Success Scenario

1. The user selects the option to create a new information page.
2. The system displays a form asking the user to input:
 - Title for the information page.
 - Content for the information page.
3. The user inputs the title and content and continues to preview the page.
4. The system validates the input and displays a preview.
5. The user accepts the preview.
6. The system stores the information page.
7. The system displays a confirmation that the information page was saved.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

1a. The user wants to update existing information pages.

1. The user selects the option to work with an existing information page.
2. The system displays a list of existing information pages.
3. The user selects the information page to edit.
4. The system displays a form with data from the selected page containing fields for:
 - Title for the information page.
 - Content for the information page.
5. The user edits the data in the form and submits it for preview.
6. The system validates the data and displays a preview.
 - 1a. The data doesn't validate.
 1. The system displays a form with the fields that didn't validate highlighted.
 2. The user edits the data and submits it again, continue from step 5.
7. The user accepts the preview.
 - 1a. The user is not satisfied with the preview.
 1. The user goes back to the form (step 4).

2. The user edits the content previously entered.
 3. The user continues to step 5 again.
 8. The system stores the updated page.
 9. The system displays a confirmation that the information page was updated.
- 1b. The user wants to delete existing information pages.
1. The user selects the option to work with existing information pages.
 2. The system displays a list of existing information pages.
 3. The user selects the information page to delete.
 4. The system displays the page selected by the user and asks for confirmation.
 5. The user confirms the deletion.
 - 1a. The user doesn't want to delete the information page and cancels the deletion.
 1. The system redirects the user to the list of existing pages (step 2).
 6. The system deletes the information page and removes it from the course website.
 7. The system displays a confirmation that the information page was deleted.
- 4a. The title or content is missing.
1. The system displays a form with the fields that didn't validate highlighted.
 2. The user edits the data and submits it again.
- 5a. The user is not satisfied with the preview and wants to make changes to the page.
1. The user goes back to the form (step 3).
 2. The user edits the content she entered previously.
 3. The user resubmits the form.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Once or a few times at the start of each course, then quite rarely during the course.

Open Issues: None.

Use Case UC6: Manage Course News

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want to post course news without having to upload new HTML files.
- *Course assistants:* Want the course website to contain up-to-date information.
- *Students:* Want the course website to be updated frequently and contain up-to-date information.
- *University:* Wants course websites to contain up-to-date information so that misunderstandings are kept at a minimum, and that students are satisfied with the information provided.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: The changes are saved and the course news on the website is updated.

Main Success Scenario

1. The user selects the option to create new course news
2. The system displays a form asking the user to input:
 - Headline for course news.
 - Content for course news.

3. The user inputs the headline and content in a form and continues to preview the news.
4. The system validates the input and displays a preview.
5. The user accepts the preview.
6. The system stores the course news.
7. The system displays a confirmation that the course news has been published.

Extensions

- *a. At any time, the system fails.
 1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

- 1a. The user wants to update existing course news.
 1. The user selects to work with existing course news.
 2. The system displays a list of existing course news.
 3. The user selects the course news to edit.
 4. The system displays a form with data from the selected news containing fields for:
 - Headline for course news.
 - Content for course news.
 5. The user edits the data in the form and submits it for preview.
 6. The system validates the data and displays a preview.
 - 1a. The data doesn't validate.
 1. The system displays the form with the fields that didn't validate highlighted.
 2. The user edits the data and submits it again, continue from step 5.
 7. The user accepts the preview.
 - 1a. The user is not satisfied with the preview.
 1. The user goes back to the form (step 4).
 2. The user edits the content she entered previously.
 3. The user continues to step 6 again.
 8. The system stores the updated course news, and displays the new version on the course website.
 9. The system displays a confirmation that the course news was updated.

- 1b. The user wants to delete existing course news.
 1. The user selects the option to work with existing course news.
 2. The system displays a list of existing course news.
 3. The user selects the course news to delete.
 4. The system displays the course news selected by the user and asks for confirmation.
 5. The user confirms the deletion.
 - 1a. The user doesn't want to delete the course news and cancels the deletion.
 1. The system redirects the user to the list of existing course news (step 2).
 6. The system deletes the course news and removes it from the course website.
 7. The system displays a confirmation that the course news was deleted.

- 4a. The headline or content is missing.
 1. The system displays a form with the fields that didn't validate highlighted.
 2. The user submits the news for preview.

- 5a. The user is not satisfied with the preview and wants to make changes to the news.
 1. The user goes back to the form (step 3).
 2. The user edits the content she entered previously.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times per week for each course when the course is active.

Open Issues: None.

Use Case UC7: Managing deadlines

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want deadlines spaced so that their workload is evenly spread.
- *Course assistants:* Want deadlines spaced so that their workload is evenly spread.
- *Students:* Want to know deadlines for courses, to better spread their workload.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: The list of deadlines for the course in question is updated, reflecting the changes made by the course leader.

Main Success Scenario

1. The user selects the course for which to manage deadlines.
2. The user selects the option to add new deadline.
3. The system displays a form asking the user to input:
 1. The date for the deadline.
 2. A description for the deadline.
4. The user inputs a date and a description.
5. The system validates the input.
6. The system displays a summary of the new deadline, and asks the user to confirm.
7. The user confirms the deadline.
8. The system saves the deadline.
9. The system confirms that the new deadline was successfully saved.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

2a. Edit existing deadline.

1. User selects the option to edit an existing deadline.
2. The system displays a list of existing deadlines.
3. The user selects the deadline to be edited.
4. The system displays a form asking the user to input:
 - Date for the deadline.
 - A description for the deadline.
5. The user inputs changes.

6. The system displays a preview of the deadline and asks for confirmation.
 7. The user confirms the changes.
 - 1a. The user is not satisfied with the preview.
 1. The user goes back to the form (step 4).
 2. The user edits the content she entered previously.
 3. The user continues to step 6 again.
 8. The system saves the changes.
 9. The system displays a confirmation that the changes have been saved.
- 2b. Delete existing deadline.
1. The user selects the option to delete an existing deadline.
 2. The system displays a list of existing deadlines.
 3. The user selects the deadline to be deleted.
 4. The system displays details for the deadline and asks for confirmation.
 5. The user confirms deletion.
 - 1a. The user doesn't want to delete the deadline and cancels the deletion.
 1. The system redirects the user to the list of existing deadlines (step 2).
 6. The system deletes the deadline.
 7. The system displays a confirmation that the deadline has been deleted.
- 5a. The date is in an unrecognized format.
1. The system displays the form again, highlighting the fields that failed validation and asks the user to correct the error.
 2. The user inputs the date in the correct format and resubmits the form.
- 5b. The description is invalid.
1. The system displays the form again, highlighting the fields that failed validation and asks the user to correct the error.
 2. The user inputs the description in the correct format and resubmits the form.
- 7a. The user doesn't confirm the deadline
1. The system displays the form again, allowing the user to make changes.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times at the beginning of each course, but might also be used at rare occasions during the course.

Open Issues: None.

Use case UC8: Register Results

Primary Actor: Course leader and course assistant

Stakeholders and Interests:

- *Course leaders:* Want an efficient way to register results for students registered in the course they are responsible for.
- *Course assistants:* Want an efficient way to register results for students registered in the course that they have been assigned to.
- *Students:* Want results to be saved and updated as soon as possible after the grading.

Preconditions: The course leader or course assistant is authenticated and have the necessary privileges to register results for the course.

Success Guarantee: One or more students' results are registered to the system and saved.

Main Success Scenario

1. The user selects the course for which to register results.
2. The user selects the option to register results.
3. The system displays a list of the registered students and the gradable course assignments.
4. The user assigns the grades for students and assignments.
5. The user saves the changes.
6. The system validates the input.
7. The system saves the changes.
8. The system displays a confirmation that the results have been saved.

Extensions

- *a. At any time, the system fails.
1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
 - 2a. The input doesn't validate.
 1. The system displays the form once again and asks the user to correct the errors.
 2. The user makes changes to the input and resubmits the form.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Frequently during each course and student depending very much on the course layout.

Open Issues: None.

Use Case UC9: Manage Files

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want to easily publish files on the course website.
- *Course assistants:* Want the course website to be up-to-date with files related to the course.
- *Students:* Want the course website to be up-to-date with files related to the course.
- *University:* Wants the course website to have files that satisfies the student's needs but don't want lengthy documents to be made available that the students will later print using the university's resources.

Preconditions: The course leader or course assistant is authenticated and have the necessary privileges to register results for the course.

Success Guarantee: The files for the course in question are up-to-date and are accessible for the students via the course website.

Main Success Scenario

1. The user selects the course for which to manage files.
2. The user selects the option to upload files.
3. The system displays a form asking the user to input:
 - A title for the file.

- A description for the file.
 - The file itself.
4. The user enters the required information and selects the file for upload.
 5. The system validates the input.
 6. The system saves the file, title and description.
 7. The system publishes the file on the course website.
 8. The system displays a confirmation that the file has been uploaded and published.

Extensions

- *a. At any time, the system fails.
1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
- 2a. The user wants to update an existing file.
1. The user selects the option to work with existing files.
 2. The system displays a list of existing files.
 3. The user selects the file to edit.
 4. The system displays a form asking the user to input:
 - A title for the file.
 - A description for the file.
 - The file itself.
 5. The user edits the data in the form, selects a new file and submits it.
 6. The system validates the data.
 - 1a. The data doesn't validate.
 1. The system displays a form with the fields that didn't validate highlighted.
 2. The user edits the data and submits it again.
 7. The system deletes the old file, stores the new file, updates the title and description for the file, and displays the new version on the course website.
 8. The system displays a confirmation that the file has been updated.
- 2b. The user wants to delete an existing file.
1. The user selects the option to work with existing files.
 2. The system displays a list of existing files.
 3. The user selects the file to delete.
 4. The system displays the file selected by the user and asks for confirmation.
 5. The user confirms the deletion.
 - 1a. The user doesn't want to delete the file and cancels the deletion.
 1. The system redirects the user to the list of existing files (step 2).
 6. The system deletes the file and removes it from the course website.
 7. The system displays a confirmation that the file has been deleted.
- 5a. The title or description doesn't validate.
1. The system displays a form with the fields that didn't validate highlighted.
 2. The user submits the form again.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times per week for each active course, but varying a lot depending on the course layout.

Open Issues: None.

Use Case UC10: Manage Course Registration

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want to maintain an accurate list of students taking the course.
- *Course assistants:* Want to have a complete list of registered students in the course when they register results for students.
- *Students:* Want to get accepted for a course.
- *University:* Wants an efficient administration of registration for courses.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: All users approved for the course by the course leader has been accepted into the course, and the course leader can view registered users.

Main Success Scenario

1. The user selects the course for which to manage course registration.
2. The user selects the option to confirm course registrations.
3. The system displays a list of students who have tried to register for the course.
4. The user selects all students that are to be accepted into the course and accepts these.
5. The system updates the status for the selected students.
6. The system displays a confirmation of which students have been accepted to the course.

Extensions

- *a. At any time, the system fails.
 1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
 - 2a. The user wants to unregister a student from the course.
 1. The user selects the option to work with registered students.
 2. The system displays a list of registered students.
 3. The user selects a student to unregister from the course.
 4. The system displays the student selected by the user and asks for confirmation.
 5. The user confirms.
 6. The system unregisters the student from the course, updating the student's status.
 7. The system displays a confirmation that the student was unregistered from the course.
 - 2b. The user wants to print a list of students registered for the course.
 1. The user selects the option to work with registered students.
 2. The system displays a list of registered students.
 3. The user prints the list, using the built-in print functionality in the web browser.
 4. The list is printed.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times during the start of each course but very rarely otherwise. Most courses starts in between periods.

Open Issues: None.

Use Case UC11: Manage Course Assistants

Primary actor: Course leader

Stakeholders and Interests:

- *Course leaders:* Want course assistants who can register results for graded assignments on their own, to lessen the workload of the course leader.
- *Course assistants:* Don't want to be dependent on course leaders to register results.
- *Students:* Want to see their results as soon as possible.
- *University:* Wants the course to be run smoothly without major delays for results to get registered.

Preconditions: The course leader is authenticated and assigned as the course leader for the course in question.

Success Guarantee: The privileges for the assistants are updated.

Main Success Scenario

1. The user selects the course for which to manage course assistants.
2. The user selects the option to add a course assistant.
3. The system displays a form asking the user to input:
 - The username for the course assistant.
4. The user enters the username of the assistant.
5. The system asks the user to confirm.
6. The user confirms that the course assistant should be added.
7. The system confirms that the course assistant has been added to the course.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start working from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
- 2a. The user wants to remove an assistant.
 1. The user selects the option to remove a course assistant.
 2. The system displays a list of existing course assistants.
 3. The user selects the course assistant to be removed.
 4. The system displays the course assistant that the user selected and asks for confirmation.
 5. The user confirms removal.
 - 1a. The user doesn't want to remove the course assistants and cancels the removal.
 1. The system redirects the user to the list of existing course assistants (step 2).
 6. The system confirms that the course assistant was removed.
- 3a. The username entered is invalid.
 1. The system displays the form containing the data that the user submitted and highlights the invalid field.

2. The user corrects the error and continues.
- 4a. The user has made a mistake, and cancels the operation.
 1. The system displays the form again (step 2).

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Once or a few times at the start of the course for each course, then very rarely.

Open Issues: None.

Use case UC12: Register for Course

Primary Actor: Students

Stakeholders and Interests:

- *Course leaders:* Want the students to register for the course to plan resource allocation and be able to register results for students.
- *Course assistants:* Want the students to register for the course to be able to register results for students.
- *Students:* Want to register for the course to be able to get results and credits.
- *University:* Wants to know how many students are registered, and wants an efficient administration.

Preconditions: The student is registered in the system but not registered in the course in question. The user is authenticated.

Success Guarantee: The student is registered to the course, awaiting confirmation from the course leader.

Main Success Scenario

1. The user navigates to the course website.
2. The system displays the course website.
3. The user selects the option to register for the course.
4. The system asks the user to confirm the registration.
5. The user confirms.
6. The system saves the course registration.
7. The system displays a confirmation that the user has registered for the course, and is awaiting confirmation from the course leader.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start over from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

5a. The user doesn't want to register for the course and cancels the registration.

1. The system redirects the user to the course website (step 2).

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Once per student and course when a new course starts. Most courses start in between periods.

Open Issues: None.

Use Case UC13: View Course Description

Primary actor: Student

Stakeholders and Interests:

- *Course leaders:* Want the students to get a clear picture of the course.
- *Course assistants:* Want the students to get a clear picture of the course.
- *Students:* Want to get a clear picture of the course.
- *University:* Wants students to be able to get a clear picture of the course.

Preconditions: The user is using a computer with a working Internet connection, has started the web browser Mozilla Firefox and has navigated to the system.

Success Guarantee: The student views the course description.

Main Success Scenario

1. The system displays a form asking the user to input:
 - Search string.
2. The user inputs the search string.
3. The system displays a list of courses matching the search string.
4. The user selects the course.
5. The system displays the course website for the selected course.
6. The user selects to display course description.
7. The system displays the course description.

Extensions

*a. At any time, the system fails.

1. The user tries to reload the page.

- 1a. The system is back online, and the user gets the requested information.
- 1b. The system is still unavailable, so the user contacts the administrator.

1a. The user is on the personal page.

1. The user selects the course from the list of courses the user is registered for.
2. The system displays the course website for the selected course.
3. The user selects to display course description.
4. The system displays the course description.

2a. The user cannot find the intended course.

- 1a. The user searches for the course again.
- 1b. The user gives up, and tries to find out the course code or course name elsewhere.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: About one time per student at the beginning of a course.

Open Issues: None.

Use Case UC14: View Results

Primary actor: Student

Stakeholders and Interests:

- *Course leaders:* Want students to be able to view their results.
- *Course assistants:* Want students to be able to view their results.

- *Students*: Want to view their own results, and don't want other students to be able to view their results.
- *University*: Wants students to be able to view their results, and only their own results, respecting the user's privacy.

Preconditions: The student is authenticated and registered for the course she is trying to look up her results for and has navigated to the personal page.

Success Guarantee: The student has seen her results in the specified course.

Main Success Scenario

1. The user selects the course for which to view results.
2. The user selects the option to view her results.
3. The system displays a list of courses the user is registered for.
4. The user selects the course.
5. The system displays all the student's results for the selected course.
6. The user views the results.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start over from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Frequently during the course, and probably more frequently towards the end of the course when more results are registered. The frequency depends very much on how many graded assignments are included in the course. The student is expected to view her results a couple of times per graded assignment, depending on how quickly the result is registered.

Open Issues: None.

Use Case UC15: View Overview of Deadlines

Primary actor: Student

Stakeholders and Interests:

- *Course leaders*: Want assignments handed in on time.
- *Course assistants*: Want assignments handed in on time.
- *Students*: Want an overview of all deadlines to better manage their workload.

Preconditions: The user is using a computer with a working Internet connection, has started the web browser Mozilla Firefox and has navigated to the personal page.

Success Guarantee: A list of all deadlines for all the courses the user is registered in is displayed.

Main Success Scenario

1. User selects the option to view deadlines.
2. The system displays a list of all deadlines for the courses the user is registered in.

Extensions

*a. At any time, the system fails.

1. The user tries to reload the page.
 - 1a. The system is back online, and the user gets the requested information.
 - 1b. The system is still unavailable, so the user contacts the administrator.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times per week for each individual student, but depends very much on the courses and the course layouts that the student is attending.

Open Issues: None.

Use case UC16: View the schedule

Primary Actor: Student

Stakeholders and Interests:

- *Course leaders:* Want students to be able to view the schedule so that they can attend the teaching activities as they wish, so that they can achieve good results and are satisfied with the course.
- *Course assistants:* Want students to be able to view the schedule so that they can attend the teaching activities that the course assistant is involved in.
- *Students:* Want to be able to get an overview of the courses' activities.

Preconditions: The user is using a computer with a working Internet connection, has started the web browser Mozilla Firefox and has navigated to the personal page.

Success Guarantee: The user's schedule is displayed in the web browser.

Main Success Scenario

1. The user selects the option to view the schedule.
2. The system displays the personal schedule.

Extensions

- *a. At any time, the system fails.
 1. The user tries to reload the page.
 - 1a. The system is back online, and the user gets the requested information.
 - 1b. The system is still unavailable, so the user contacts the administrator.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Almost daily for students all days preceding a weekday.

Open Issues: None.

Use case UC17: Export the Schedule in iCalendar Format

Primary Actor: Student

Stakeholders and Interests:

- *Course leaders:* Want students to be able to export the schedule so that they can follow the course activities.
- *Course assistants:* Want students to be able to export the schedule so that they can follow the course activities.
- *Students:* Want to be able to export the schedule so that it can be used in external systems, such as mobile phones or calendar applications.

Preconditions: The user is viewing his or her schedule.

Success Guarantee: The user has acquired a file in the iCalendar format representing his or her personal schedule.

Main Success Scenario

1. The user selects the option to export the schedule.
2. The system provides the user with a file containing the schedule in the iCalendar format.
3. The user downloads the iCalendar file to his or her computer.

Extensions

- *a. At any time, the system fails.
 - 1a. The system is available, the user navigates to the system, selects to view the schedule again and starts over.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: More often during the beginning of periods (but reasonably no more than once per student).

Open Issues: None.

Use Case UC18: View News

Primary actor: Student

Stakeholders and Interests:

- *Course leaders:* Want the students to be able stay up-to-date with news in the course.
- *Course assistants:* Want the students to be able stay up-to-date with news in the course.
- *Students:* Want to stay up-to-date with news in the course.
- *University:* Wants students to be able to stay up-to-date with activities in the course, minimizing the number of misunderstandings.

Preconditions: The user is using a computer with a working Internet connection and has started the web browser Mozilla Firefox.

Success Guarantee: The user views the news.

Main Success Scenario

1. The user navigates to his or her personal page.
2. The system displays course news for the courses that the user is registered to.

Extensions

- *a. At any time, the system fails.
 - 1a. The system is available, the user navigates to the system and views the news again and starts over.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.
- 1a. The user uses RSS feeds to read headlines.
 1. The user selects to view course news headlines from the RSS viewer integrated in the Mozilla Firefox web browser.
 2. The user selects the headlines to view.
 3. The system displays the news items selected by the user.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: Daily or almost daily for most students.

Open Issues: None.

Use Case UC19: View Uploaded Files

Primary actor: Students, course assistants and course leaders

Stakeholders and Interests:

- *Course leaders:* Want students and course assistants to be able to view the uploaded files.
- *Students:* Want to be able to get material provided by the course leader.
- *Course assistants:* Want to be able to get material provided by the course leader.

Preconditions: The user is authenticated and registered for the course. They have navigated to the course website.

Success Guarantee: The file has been downloaded to the user's computer.

Main Success Scenario

1. The user selects the option to view uploaded files.
2. The system displays a list of all uploaded files for the course.
3. The user selects a file to download.
4. The system sends the file to the user.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.
 - 1a. The user is authenticated and can start over from scratch.
 - 1b. The system is not accessible, the user will have to try again another time or contact the system administrator.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: A couple of times per week, much depending on how frequently course material is made available.

Open Issues: None.

Use Case UC20: Manage user accounts

Primary actor: System administrator

Stakeholders and Interests:

- *System administrators:* Want to create and manage the accounts.
- *Course leaders:* Want an account so they can create course websites.
- *Students:* Want an account so they can register for courses and get easy access to news and information about the courses they register for.
- *Course assistants:* Want an account to stay updated about the assigned courses and register results for them.
- *University:* Unauthorized people shouldn't be able to change the content of course websites, such as adding inappropriate content.

Preconditions: The system administrator is authenticated.

Success Guarantee: User accounts are created or deleted.

Main Success Scenario

1. The user selects the option to create an account.
2. The system displays a form asking the user to input:
 - Username.
 - Password.
3. The user inputs a username and password.
4. The system validates the input.
5. The system creates the account.
6. The system displays a confirmation that the user account has been created.

Extensions

- *a. At any time, the system fails.
 1. The user authenticates again.
 - 1a. The system is accessible, the user navigates to the system again, authenticates and tries again.
 - 1b. The system is not accessible, the system administrator will reboot the system and try again.

- 1a. The user wants to delete a user account.
 1. The user selects the option to delete an account.
 2. The system displays a form asking the user to input:
 - Search string.
 3. The user enters the search string for a specific user account to be searched for.
 4. The system displays all accounts with username containing the user's search string.
 - 1a. No matching user exists.
 1. The system returns to step 3, displaying a message detailing the problem.
 5. The user selects the account to be deletion.
 6. The system asks the user to confirm the deletion.
 7. The user confirms the deletion.
 - 1a. The user doesn't want to delete the selected user account and cancels the deletion.
 1. The system displays the form for searching for user accounts again (step 2).
 8. The system deletes the account.
 9. The system displays a confirmation that the user account was deleted.

- 4a. The username already exists.
 1. The system displays the form again with the data entered by the user (step 2).

- 4b. The username or password doesn't validate.
 1. The system displays the form once again with the data entered by the user, highlighting the incorrect fields.
 2. The user enters new values in the fields and submits the form again (step 4).

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: When a new teacher joins the university. The frequency depends much on the size of the university and the employee turnover. Occurs more frequently in between periods when new courses start and the need for user accounts arises.

Open Issues: None.

Use Case UC21: Assign Privileges to a User Account

Primary actor: System administrator

Stakeholders and Interests:

- *System administrators:* Assigning the necessary privileges to the user accounts.
- *Course leaders:* Want to have the privileges required to manage course website.
- *Course assistants:* Want to have the privileges required to report students' results.
- *University:* Unauthorized people shouldn't be able to change the content of course websites, such as adding inappropriate content.

Preconditions: The system administrator is authenticated.

Success Guarantee: The user is updated with new privileges. Set privileges for course leaders or course assistants to specified courses.

Main Success Scenario

1. The user selects the option to set privileges.
2. The system displays a form asking the user to input:
 - Search string.
3. The user specifies the search string for a specific user account to be searched for.
4. The system displays all user accounts with username containing the search string.
5. The user selects an account.
6. The system displays a form asking the user to input:
 - Search string.
7. The user specifies the search string for a specific course code to be searched for.
8. The system displays all courses that contain the course code.
9. The user selects a course.
10. The system provides a form where the user can select privileges containing fields for:
 - Course leader.
 - Course assistant.
11. The user selects either course leader or course assistant privileges.
12. The system asks the user for confirmation.
13. The user confirms the input.
14. The system saves the changes.
15. The system displays a confirmation that the privileges have been assigned.

Extensions

- *a. At any time, the system fails.
 1. The user authenticates again.
 - 1a. The system is accessible, the user navigates to the system again, authenticates and tries again.
 - 1b. The system is not accessible, the system administrator will reboot the system and try again.
 - 4a. No matching user accounts were found.
 1. The system displays the form again containing the data the user entered.
 2. The user changes the form data and resubmits the form (step 2).
 - 7a. No matching courses were found.
 1. The system displays the form again containing the data the user entered.
 2. The user changes the form data and resubmits the form (step 6).
 - 13a. The user doesn't want to assign privileges to the user account and cancels.

1. The system displays the form in step 10 again.

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: When a new course leader is assigned to a course. The frequency depends a lot on the number of courses and how long the course leader stays with his or her course. Occurs more frequently in between periods when new courses start. Most of the course assistants are expected to be assigned by their course leaders.

Open Issues: None.

Use Case UC22: Create Course

Primary actor: System administrator

Stakeholders and Interests:

- *System administrators:* Want to setup courses to help administration of course websites.
- *Course leaders:* Want the ability to create websites for courses they are responsible for.
- *Students:* Want courses to be created so that they can register for them.
- *Course assistants:* Want courses to be created to be able to register results for students.
- *University:* Wants courses to be created so students can register and administration to be efficient.

Preconditions: The user is authenticated.

Success Guarantee: The course has been added to the system, and a course leader can now be added for the course.

Main Success Scenario

1. The user selects the option to add courses.
2. The system displays a form asking the user to input:
 - Course code.
3. The user inputs the course code.
4. The system validates the course code.
5. The system saves the new course.
6. The system displays a confirmation that the course was added.

Extensions

*a. At any time, the system fails.

1. The user authenticates again.

- 1a. The system is accessible, the user navigates to the system again, authenticates and tries again.

- 1b. The system is not accessible, the system administrator will reboot the system and try again.

- 3a. The course code already exists.

1. The system displays the form again, notifying the user that the course code already exists in the system.
 2. The user inputs a new course code and resubmits the form (step 4).

- 3b. The course code was invalid (too long, contained non-alphanumeric for example)

1. The system displays the form again, notifying the user that the course code was invalid.
2. The user inputs a new course code and resubmits the form (step 4).

Special Requirements: None.

Technology and Data Variations List: None.

Frequency of Occurrence: When a new course is created. Occurs mostly in between periods when new courses start.

Open Issues: None.