Team BoardsReview - CM Capstone Spring 2010

Chicago Board of Review Course Site

Use Cases

Past Team Use Cases

The following use cases were implemented by the previous team and we would like to keep them throughout our work on the project.

Use Case 1: Student logs into site

Primary Actor: Student wants to access lecture material online
Stakeholders and Interests: Professors want to make sure that only authorized users are allowed to view course materials online.
Preconditions: The student has registered, or was manually added to system by an admin.
Post-conditions: Student should be authenticated and shown the main content area of the site.
Main Success Scenario:
1. Student navigates to main page and enters their user id and password.
2. Student notifies system that login is complete
3. System verifies that username and password are correct
4. System displays the site's main page.

Extensions and Alternative Flows:
3. Username or password is incorrect.
   a. Redisplay main page with error message

Open Issues: None

Use Case 2: Site Admin logs into site
(Note: Lecture and Class admin is similar process)

Primary Actor: Admin wants to access admin panel
Stakeholders and Interests: Admins, professors, and students do not want unauthorized admins editing material.
Preconditions: The admin has registered, or was manually added to system by an admin.
Post-conditions: Admin should be authenticated and shown the main content area of the site, and from this page they should be able to access the admin panel.
Main Success Scenario:
1. Admin navigates to main page and enters their user id and password.

2. Admin notifies system that login is complete
3. System verifies that username and password are correct
4. System displays the sites main page, and an admin panel option is available.
Extensions and Alternative Flows:
3. Username or password is incorrect.
    a. Redisplay main page
Open Issues: None

Use Case 3: User logouts

Primary Actor: Student, lecture admin, course admin, or site admin
Stakeholders and Interests: If on a public computer, a user does not want someone else to be able to access his account when he leaves.
Preconditions: User logged in.
Post-conditions: User is logged out. It should not be possible to return to any pages that the user viewed by hitting the back button, typing in URL directly, etc. User is redirected to main page.
Main Success Scenario:
1. User clicks logout button.
2. System destroys session data and redirects user to main page.
Extensions and Alternative Flows:
Open Issues: None

Use Case 4: A user who has not paid for the course wants to register

Primary Actor: User who has not paid for course
Stakeholders and Interests: Professors want to make sure that only authorized users (who have paid) are allowed to view course materials online.
Preconditions: None
Post-conditions: User should have an account that allows for login into site. User should get an email with verification.
Main Success Scenario:
1. User navigates to registration page. User enters their user id (email), password (twice), and name. User chooses a course to subscribe to.
2. User notifies system that registration is complete.
3. System verifies that passwords match and email is valid, and user id is not already in system.
4. System sends user to ecommerce page (PayPal).
5. User goes through the PayPal process, and the System receives a confirmation notice.
6. System verifies confirmation.
7. System creates an account with the chosen subscription.
Extensions and Alternative Flows:
3.1. Passwords did not match
    a. Clear passwords and make user enter both again
3.2. User ID already exists in system

a. Notify user that ID already exists and display page again.

6.1 Payment not verified

a. Notify user of no payment and display register page again.

Open Issues: None

Use Case 5: A site or course admin wants to add a list of users to the system for a given course

Primary Actor: Site or course admin

Stakeholders and Interests: Professors want to make sure that only authorized users (who have paid) are allowed to view course materials online. Site admins want an easy to way to add a lot of users to the system.

Preconditions: Admin is logged in, and admin has a spreadsheet of user ID emails and knows which course to add the list to. The spreadsheet may also contain user names.

Post-conditions: All valid user IDs on the list are created with the given course subscription added, or already existing IDs have the new subscription added.

Main Success Scenario:

1. Site admin navigates to course management page. Site admin chooses the correct course and navigates to the students list.
2. Site admin uploads a spreadsheet of user ID emails. The spreadsheet may also contain user names.
3. System parses the data and verifies that IDs are valid.
4. If the IDs are not in the system, the system creates random passwords for the accounts, and the passwords are emailed to the ID emails. The email recommends changing the password, and a link is provided. Accounts are created with the temporary passwords.
5. All newly created accounts and accounts that were already in the system with the provided ID are given the specified course subscription.

Extensions and Alternative Flows:

2. A non-applicable file type is uploaded.

a. System prints a warning

3. IDs are not valid email address

a. Ignore ID, and print a warning

5. The user already had an active subscription in the course

a. If the new subscription would expire after the old subscription, replace it.

b. If not, do nothing.

Open Issues: None

Use Case 6: User forgets password

Primary Actor: User who forgets password

Stakeholders and Interests: Professors want to make sure that only authorized users (who have paid) are allowed to view course materials online. Student wants to be able to access account even though they forgot password.

Preconditions: User is registered

Post-conditions: User gets a new password

Main Success Scenario:

1. User navigates to forgotten password page and enters user ID
2. User notifies system that ID has been typed in.
3. System verifies that ID is valid.
4. System emails user a link to a page that will confirm the user's email. The system prints a message concerning this.
5. User opens email and clicks on link.
6. System generates a random password, sets the user's new password, and sends an email with the new password. System prints a message concerning this.

Extensions and Alternative Flows:

3. a. ID is not valid email address

a. Print message stating userID not in system

Open Issues: None

Use Case 7: User wants to change password

Primary Actor: User who wants to change password

Stakeholders and Interests: Users wants to be able to change password (especially from random one).

Preconditions: User has logged in

Post-conditions: User gets a new password

Main Success Scenario:

1. User navigates to profile page and enters new password twice
2. User notifies system that fields have been typed in.
3. System verifies that passwords match.
4. System sets the user's new password and prints a message on completion.

Extensions and Alternative Flows:

3. New passwords do not match

a. Print message that passwords do not match.

Open Issues: None

Use Case 8: A site or course admin wants to give a user (Tim) lecture admin privileges for a certain course's lecture (Neurosurgery 2010: Brain Tumors)

Primary Actor: Site or course admin

Stakeholders and Interests: Someone who has registered as a user does not want to re-register just to become a lecture/course admin. Admins do not want to give someone rights that they should not have them.

Preconditions: Site or course admin is logged in. The user in this example already has an account (Tim)

Post-conditions: A user (Tim) is a now a lecture admin for a certain lecture in a certain course (Neurosurgery 2010: Brain Tumors).

Main Success Scenario:

1. Site admin navigates to course management page. Site admin chooses a course (Neurosurgery 2010). Site admin navigates to lecture management page and chooses a lecture (Brain Tumors). Site admin navigates to Lecture Admin Page.
2. Site admin notifies system he wants to add a lecture admin.
3. System brings up list of users.
4. Site admin may filter/sort list (by typing in “Tim”)
5. Site admin finds user, selects user, and notifies system to add him.
5. System gives Tim lecture admin privilege for the designated lecture in the designated course. (Neurosurgery 2010: Brain Tumors) System redirects to Lecture Admin Page.

Extensions and Alternative Flows:

Open Issues: None

Use Case 9: A site or course admin wants to extend a user’s (Tim’s) subscription

Primary Actor: Site or course admin
Stakeholders and Interests: Site admins may want to extend, shorten, or end specific users’ subscription for various reasons.
Preconditions: Site admin is logged in.
Post-conditions: A user (Tim) has 90 more days on his subscription’s end date, or 90 days from current date if the subscription expired.
Main Success Scenario:

1. Site admin navigates to course management page, chooses a course (Neurosurgery 2009), and navigates to Student list.
2. Site admin enters user ID or name (Tim) into filter, selects the user, and notifies System he wants to edit the subscription.
3. Site admin chooses to extend subscription by the standard amount (pulled from database for that course. In this case, 90 days), and notifies the System.
4. System extends Tim’s subscription’s end date by 90 days

Extensions and Alternative Flows:

3. a. Site admin chooses to extend subscription by a number of days he enters.
   b. Site admin chooses to edit the end date of the subscription directly.
4. a. If the subscription is expired, the new subscription end date is current date + X days.
   b. If a specific end date is entered, the System just updates the end date.

Open Issues: None

Use Case 10: A site admin wants to create a course

Primary Actor: Site admin
Stakeholders and Interests: Site admins will want to create new courses to reflect the new courses in real life. Students want to be able to see these courses.

Preconditions: Site admin is logged in.
Post-conditions: A new course is created.
Main Success Scenario:

1. Site admin navigates to course management page.
2. Site admin tells System that he wants to create a new course.
3. System asks for a course name, date, and subscription pricing/length.
4. Site admin enters data and notifies system of completion.
5. System verifies that data has been entered and conforms to expected input (no blank fields and the date field is a date, etc)
6. System creates a new course with data specified. System redirects to course management page.

Extensions and Alternative Flows:

4. Site admin cancels
5. Incorrect input
    a. System notifies of missing fields or errors, and asks admin to enter again

Open Issues: None

Use Case 11: A site or course admin wants to create a lecture

Primary Actor: Site or course admin
Stakeholders and Interests: Site and course admins will want to create new lectures to reflect the new lectures in real life. Students want to be able to see these lectures.
Preconditions: Site admin is logged in.
Post-conditions: A new lecture is created
Main Success Scenario:

1. Site admin navigates to lecture management page of 2009 neuro course.
2. Site admin tells System that he wants to create a new lecture.
3. System asks for a lecture date/time, topic, and instructor.
4. Site admin enters data and notifies system of completion.
5. System verifies that data has been entered and conforms to expected input (no blank fields and the date field is a date, etc)
6. System creates a new lecture with data specified. System redirects to lecture management page.

Extensions and Alternative Flows:

4. Site admin cancels
5. Incorrect input
    a. System notifies of missing fields or errors, and asks admin to enter again

Open Issues: None

Use Case 12: A Student wants to view his subscriptions

Primary Actor: Student
Stakeholders and Interests: Students want to view subscriptions.
Preconditions: Student is logged in
Post-conditions: Student views subscriptions
Main Success Scenario:

1. Student navigates to subscription page
2. Student can see his subscription for each course, along with start and expiration date, and days left

Extensions and Alternative Flows:

None

Open Issues: None

Use Case 13: A Site Admin wants to add a user

Primary Actor: Site Admin
Stakeholders and Interests: Site Admins want to be able to manually add users in case of difficultly or perhaps to increase security in creating course admins. Users don't want to have to pay a fee again if something happens to their account and a new one needs to be created.
Preconditions: Site admin is logged in.
Post-conditions: A new user account is created.
Main Success Scenario:

1. Site admins navigates to user management page. Site admin notifies system he wants to add a user.
2. System displays data fields for the Site admin to enter (name, email ID, password).
3. Site admin enters in the data and notifies the system.
4. System verifies data.
5. System creates new account.
6. If the site admin indicated it, the system will email the user to notify of account creation.

Extensions and Alternative Flows:

4. Data was not entered correctly (blank fields, not an email, etc)
    a. System notifies site admin of error

Open Issues: None

Use Case 14: A Site, Course, or Lecture Admin wants to edit a lecture's title

Primary Actor: Site, Course, or Lecture Admin

Stakeholders and Interests: Admins want to correct typos and errors in a lecture's basic information, and viewers do not want to see errors.
Preconditions: Site admin is logged in. The lecture already exists
Post-conditions: A lecture's title is updated
Main Success Scenario:

1. Admin navigates to course management page. Admin chooses a course, and then chooses a lecture.
2. System displays basic information about lecture.
3. Admin change's the lecture's title and notifies system.
4. System validates input.
5. System commits fields.

Extensions and Alternative Flows:

4. Incorrect input (blank fields, etc)

   a. System notifies admin

Open Issues: None

Use Case 15: A Student wants to buy a new subscription

Primary Actor: Student
Stakeholders and Interests: Students want to add new content to their account to view, and the course managers want to make money.
Preconditions: None, or Student is logged in. Student does not need to be registered.
Post-conditions: Student has a new subscription. He may have a new account if he does not have an account.
Main Success Scenario:

1. Student navigates to main page and navigates to courses section.
2. System displays all the courses offered. It may only show courses within the last year and upcoming courses, with an option to display older courses.
3. Student chooses a course and notifies system.
4. System displays all relevant information about the course (name, description, time it was/is offered, subscription price(s), subscription length(es)).
5. Student chooses a subscription option and notifies system.
6. System displays a login option and register option.
7. Student enters login information and notifies system.
8. System verifies credentials
9. System displays a confirmation of payment screen displaying the user's account name, the subscription requested, the price, and the end date.
10. Student notifies the system it wants to buy the subscription.
11. System takes the Student to the PayPal screen, and the Student goes through the PayPal process. The system is sent a confirmation of payment.
12. The System verifies the confirmation of payment
13. The System adds the subscription to the user's account.

Extensions and Alternative Flows:

Alternate #1:

1. Student is logged in already (see login Use Case)
2. Student navigates to Available Courses page.
3. Student chooses a course and notifies system.
4. System displays all relevant information about the course (name, description, time it was/is offered, subscription price(s), subscription length(es)).
5. Student chooses a subscription option and notifies system.
6. System displays a confirmation of payment screen displaying the user's account name, the subscription requested, the price, and the end date.
7. Student notifies the system it wants to buy the subscription.
8. System takes the Student to the PayPal screen, and the Student goes through the PayPal process. The system is sent a confirmation of payment.
9. The System verifies the confirmation of payment
10. The System adds the subscription to the user's account.

Alternate #2:

6. Student notifies system he wants to register.
7. System displays the Register screen with the indicated course selection chosen.
8. Student goes through the Registration process indicated in Use Case 4

Alternate #3

8. Login fails
    a. System displays page again and prints error.

Alternate #4

12. The verification of payment fails
    a. System prints error and displays confirmation payment screen again.

Open Issues: None

Current Team Use Cases

The following use cases are those that we want to achieve by the end of the semester.

Sprint 1:
Due to the fact that we were familiarizing ourselves with the work that had been completed and the work that needed to be done, we did not complete any use cases for Sprint 1.

Sprint 2:
Use Case 1: A lecture, site, or course admin wants to upload a video (or notes or slideshow)

Primary Actor: Lecture admin (or site/course admin)
Stakeholders and Interests: Admins want to add lecture content. Students want to see this content.
Preconditions: Lecture admin is logged in.
Post-conditions: A lecture resource is uploaded
Main Success Scenario:

1. Lecture admin navigates to lecture page of the 2009 neuro course.
2. Lecture admin chooses Lecture 1 on brain tumors.
3. Lecture admin specifies a video and tells system to upload.
4. Lecture admin specifies whether the video will be private or downloadable.
5. System uploads video and assigns it to the lecture.

Extensions and Alternative Flows:
Alternative #1:

3. Video is the wrong format
    a. System notifies admin and does not download.

Extension #1:

Resources will be converted automatically to an flv or swf file which will be embedded into the resources page.

Open Issues: The server conversion has to have a dedicated server for it to completely work. The customer is waiting to obtain a dedicated server and the automatic embedding has been transferred to a later time.

Use Case 2: A Student wants to view a video (or notes or slideshow)

Primary Actor: Student
Stakeholders and Interests: Students want to view course content online. Instructors do not want material to be easily downloaded.
Preconditions: Student is logged in
Post-conditions: Student views video
Main Success Scenario:

1. Student navigates to Neuro 2009 course, and navigates to Lecture page.
2. Student chooses brain tumor lecture from the list and navigates to associated page.
3. Student notifies system to view a certain resource.
4. System displays the video, slides, or notes.

Extensions and Alternative Flows:
Alternative #1

3. System displays all resources at once.
4. Student plays each individually or together.

Alternative #2

3. System displays links to all resources.

4. Student chooses a resource.
4. A new window opens with the resource viewer.

Open Issues: None

Sprint 3:

Use Case 3: A Site Admin wants to change the look of the site

Primary Actor: Site Admin
Stakeholders and Interests: Site admins want to update banners and change colors when needed to attract different customers or to express a new look/feel.
Preconditions: Site admin is logged in.
Post-conditions: The course is given a new banner and the main color is changed.
Main Success Scenario:

1. Site admins navigates to theme management page. Site admin specifies an image and tells system to download.
2. System downloads the image and sets it as the new banner.
3. Site admin inputs a new hex color in the main color input box and tells system to commit.
4. System sets the new site main color

Extensions and Alternative Flows:

2. Incorrect file type
   a. System notifies admin of error

Open Issues: Not all of the values are connected to the CSS files that are being used.

Use Case 4: A Site Admin wants to change the text in the about page

Primary Actor: Site Admin
Stakeholders and Interests: Site admins want to update pages when new content becomes available, and people will want to see this new content.
Preconditions: Site admin is logged in.
Post-conditions: The about page text is updated.
Main Success Scenario:

1. Site admins navigates to the Site Settings page. Site admin navigates to About page settings.
2. System displays the current text of the about page.
3. Site admin adds a few sentences and notifies the System of completion
4. System commits the about page text.

Extensions and Alternative Flows:

Open Issues: None

Future Use Cases

The following use cases are advanced features that are wanted in the future but are out of our scope.

Use Case 1: A Student wants to extend a subscription

Primary Actor: Student
Stakeholders and Interests: Students want to extend a subscription, and the course managers want to make a profit.
Preconditions: Student is logged in. Student has a PayPal account.
Post-conditions: A subscription is extended the correct amount.
Main Success Scenario:

1. Student navigates to subscription page
2. Student selects a subscription to view and notifies system he wants to extend the subscription.
3. System displays the pricing and subscription extension terms (number of days added, final end date)
4. User confirms he wants to extend the subscription and clicks the PayPal button.

Extensions and Alternative Flows:

3.1. An admin has specified that this subscription cannot be extended.
a. System displays notification that the selected subscription cannot be extended.
3.2. An admin has specified multiple options for an extension.
a. The student chooses one of the options.
4.1. Student cancels
a. System returns to subscriptions screen
4.2. The price of extension is $0.
a. No PayPal button is displayed. A confirm button is displayed instead.

Use Case 2: A Site Admin wants to edit a user's email and active status

Primary Actor: Site Admin
Stakeholders and Interests: A site admin may want to edit any mistakes in creating a user account, or to change a password, email, etc. A user may need a site admin to do it for them.

Preconditions: Site admin is logged in. The user's account exists and it happens to be marked as inactive.

Post-conditions: A user's email is changed and his status is set to active.

Main Success Scenario:

1. Site admins navigates to user management page. Site admin searches for a name because this user happens to have forgotten his email.
2. Site admin chooses the account and notifies system.
3. System displays the account information.
4. Site Admin sets active status to active, and he changes the user's email to a different email. He notifies the system.
5. System verifies the fields
6. System commits changes.

Extensions and Alternative Flows:

5. Data was not entered correctly (blank fields, not an email, etc)
    a. System notifies site admin of error

Use Case 3: A Site Admin wants to delete a course

Primary Actor: Site Admin

Stakeholders and Interests: Site admins could accidentally create a course they don't need. Students don't want to see trash courses that should be deleted.

Preconditions: Site admin is logged in. The course already exists

Post-conditions: A course is deleted

Main Success Scenario:

1. Site admins navigates to course management page. Site admin chooses a course and notifies system he wants to delete course.
2. System asks for confirmation on deleting course and all associated content.
3. Site admin confirms.
4. System deletes course and all associates content.

Extensions and Alternative Flows:

2. Site admin cancels
    a. System returns to course management page.

Use Case 4: A Site or Course Admin wants to upload a spreadsheet of lectures for a course

Primary Actor: Site or course Admin

Stakeholders and Interests: Site and course admins want to simplify creating lectures.

Preconditions: Site or course admin is logged in. The course exists.

Post-conditions: All valid lectures on the spreadsheet are created. If a lecturer is specified, they are assigned to the lecture.

Main Success Scenario:

1. Site admins navigates to course management page. Site admin selects a course and selects the lectures page. Site admin designates a spreadsheet and notifies system that he wants to upload a spreadsheet of lectures.
2. System downloads the spreadsheet.
3. System parses the spreadsheet and creates lectures based on the provided data. The system checks for a title, start and end time/dates, description, a lecturer email ID, and lecturer name. No fields are required, but a lecture must have at least one field.

Extensions and Alternative Flows:

3.1. An email ID is provided, and it is already in system.

a. System assigns the user as a lecture admin to the course.

3.2. An email ID is provided, but it is not in the system.

a. System creates a new account for the specified email ID, sets their name if it is provided, and emails them their random password. System assigns the user as a lecture admin to the course.

3.3. No email is provided

a. No one is assigned to the lecture for the moment.