
Computer Science 2212b

Team 7

Assignment #3

March 14th, 2014

Nick Delben

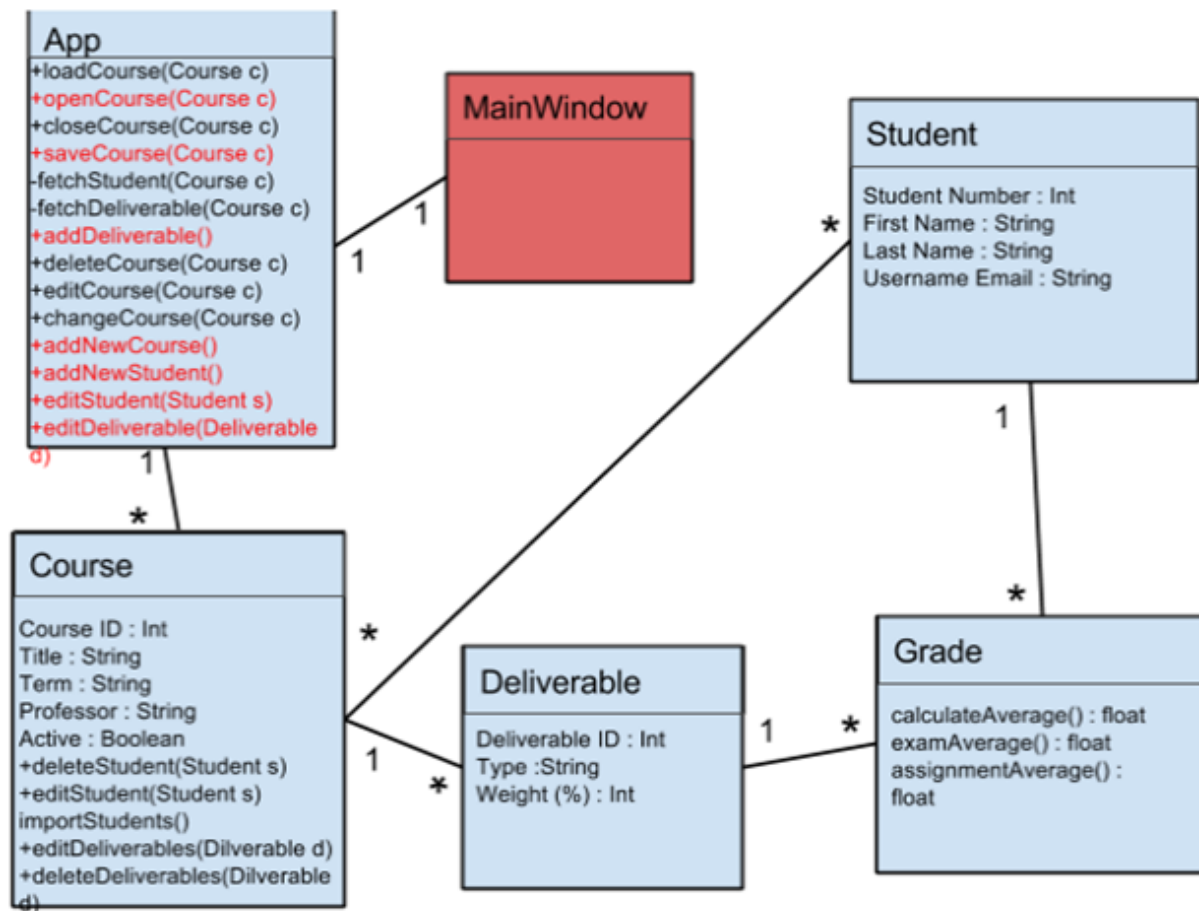
Ben Dumais

Alex Gyori

Holly Voysey

Revised UML

This is our updated UML diagram, any changes or updates are coloured in red.



Revised User Stories

Link to PivotalTracker: <https://www.pivotaltracker.com/s/projects/1004174#>

Completed since last release:

- User can remove student from a course (can be done through “unenroll student” button and having a row selected) **NOTE:** The student is not properly removed from the database at the moment, but is still unenrolled from the course itself and will not show up again
- User can change the active course (can be done by selecting the desired course in the drop-down course selector and clicking “Open Course”)
- User can view data as a spreadsheet (table will load upon opening a new active course)
- User can enroll a student in a course (by clicking “enroll student” button and creating a new student) **NOTE:** The ability to add a student from the database is not implemented yet
- User can add a deliverable to the course (done through the “create new deliverable” button)
- User can remove a deliverable from the course (done through the “remove deliverable” button)
- User can work with grades by clicking on cells (Can be done by clicking and editing the cells when an active course is loaded)
- User can exit the program at anytime (can be done by simply clicking the close window button at top right)
- User’s work should save automatically (this is automatically done whenever changing the active course or closing the program)
- User’s data is persistent (Exiting the program/reloading a course will load up the same data)

Any function not yet implemented will simply print a message that it is not implemented yet. The remove course option is not working properly as of right now due to the implementation of the database: we have moved the “User can delete a course” story from finished back to started. The Course will appear removed at first, but upon loading course lists again it will reappear (same issue with students database).

DONE

124 Feb

Pgs: 9

★

user can manage active course

User can edit a course (BD)

★

user can manage active course

User can edit a deliverable (HV)

★

users can manage deliverables

User can delete a Deliverable (BD)

★

user can manage courses

User can create a course (BD)

★

users can manage students.

User can delete a student (ND)

★

users can manage students.

User can create a new student (ND)

★

users can manage students.

User can edit a student's information (ND)

★

users can manage deliverables

User can create a new deliverable (BD)

2

3 Mar

Pgs: 0

CURRENT

310 Mar - Current

Pgs: 21 of 23

Hide accepted stories

...

★

user can manage courses

User can change the active course (BD)

★

user can manage active course

User can view course data as a spreadsheet (BD)

★

user can manage active course

User can remove student from a course (BD)

★

user can manage active course

User can add a deliverable to a course (BD)

★

user can manage active course

User can remove a deliverable from a course (BD)

★

user can manage active course

User can work with grades by clicking on cells in spreadsheet (BD)

★

user can manage courses

User can exit the program at anytime (BD)

★

user can manage courses

User's work should save automatically (BD)

★

user can manage courses

User's data should be persistent (ND)

★

user can manage active course,

users can manage students.

User can enroll a student in a course (BD)

★

user can manage courses

User can delete a course (BD)

Finish

2

3 Mar

Pgs: 0

BACKLOG

417 Mar

Pgs: 3

★

users can manage students.

User can import a list of students from csv (BD)

Start

ICEBOX

★

user can manage courses

[OPTIONAL] User can import from csv file (ND)

★

user can manage courses

[OPTIONAL] User can export csv file (ND)

★

user can manage active courses

[OPTIONAL] User can email students in the active courses

★

user can manage active courses

[OPTIONAL] User can generate for a student in the active course

★

user can manage active courses

[OPTIONAL] User can import active course from a csv file

★

user can manage active courses

[OPTIONAL] User can export active course to a CSV file

Acceptance Tests

Here are our team's acceptance tests, that completely test the following:

A user creating a new course:

ID	1
Testing	Adding a new course; Invalid Course Title
References	Requirement 4.2.1
Test Case	1. Select "Create New...", then press the "Open course" button 2. Enter "Hi" for the course title
Expected Output	Error Message: Invalid course title – Course title is too short

ID	2
Testing	Adding a new course; Invalid Course Title
References	Requirement 4.2.1
Test Case	1. Enter "This is a really really long name for a course title for the course title, which wouldn't be allowed" for the course title
Expected Output	Error Message: Invalid course title – Course title is too long

ID	3
Testing	Adding a new course; Invalid Course Title
References	Requirement 4.2.1
Test Case	1. Enter "123" for the course title
Expected Output	Error Message: Invalid course title – Course title must not contain numbers

ID	4
Testing	Adding a new course; Valid Course Title
References	Requirement 4.2.1
Test Case	1. Enter "Software Engineering" for the course title
Expected Output	Accepted The user is asked to enter a course code

ID	5
Testing	Adding a new course; Invalid Course code
References	Requirement 4.2.1
Test Case	1. Enter "Software Engineering" for the course title 2. Enter "H" for the course title
Expected Output	Error Message: Invalid course code – Course code is too short

ID	6
Testing	Adding a new course; Invalid Course Code
References	Requirement 4.2.1
Test Case	1. Enter "CS23653252456" for the course title
Expected Output	Error Message: Invalid course code – Course code is too long

ID	7
Testing	Adding a new course; Invalid Course Code
References	Requirement 4.2.1
Test Case	1. Enter "*-+" for the course code
Expected Output	Error Message: Invalid course code – Course Code must not contain symbols

ID	8
Testing	Adding a new course; Invalid Course Code
References	Requirement 4.2.1
Test Case	1. Enter "123" for the course code
Expected Output	Error Message: Invalid course code – Course Code must contain both numbers and letters

ID	9
Testing	Adding a new course; Invalid Course Code
References	Requirement 4.2.1
Test Case	1. Enter "ABC" for the course code
Expected Output	Error Message: Invalid course code – Course Code must contain both numbers and letters

ID	10
Testing	Adding a new course; Valid course Term
References	Requirement 4.2.1
Test Case	1. Enter "CS2212" for the course code
Expected Output	Accepted The user is asked to enter the course's term

ID	11
Testing	Adding a new course; Invalid course Term
References	Requirement 4.2.1
Test Case	1. Enter "T" for the course code
Expected Output	Error Message: Invalid course term – Course Term is too short

ID	12
Testing	Adding a new course; Invalid course Term
References	Requirement 4.2.1
Test Case	1. Enter "This is too long for the course's term" for the course code
Expected Output	Error Message: Invalid course term – Course Term is too long

ID	13
Testing	Adding a new course; Invalid course Term
References	Requirement 4.2.1
Test Case	1. Enter "*"-/ " for the course code
Expected Output	Error Message: Invalid course term – Course Term must not contain symbols

ID	14
Testing	Adding a new course; Valid course Term
References	Requirement 4.2.1
Test Case	1. Enter "Winter 2014" for the course code
Expected Output	Accepted The user has successfully created a new course

A user editing an existing student:

ID	1
Testing	Editing an existing student; Invalid First Name
References	Requirement 4.3.2
Test Case	1. Select an existing student, and then press 'Edit Selected Student' 2. Enter "A" for first name
Expected Output	Error Message: Invalid First Name – Name is too short

ID	2
Testing	Editing an existing student; Invalid First Name
References	Requirement 4.3.2
Test Case	1. Enter "+*/E " for first name
Expected Output	Error Message: Invalid First Name – Name must not include symbols

ID	3
Testing	Editing an existing student; Invalid First Name
References	Requirement 4.3.2
Test Case	1. Enter "1234 " for first name
Expected Output	Error Message: Invalid First Name – Name must not include numbers

ID	4
Testing	Editing an existing student; Invalid Last Name
References	Requirement 4.3.2
Test Case	1. Enter "B" for last name
Expected Output	Error Message: Invalid Last Name – Name is too short

ID	5
Testing	Editing an existing student; Invalid Last Name
References	Requirement 4.3.2
Test Case	1. Enter "+*/E " for last name
Expected Output	Error Message: Invalid Last Name – Name must not include symbols

ID	6
Testing	Editing an existing student; Invalid Last Name
References	Requirement 4.3.2
Test Case	1. Enter "1234 " for last name
Expected Output	Error Message: Invalid Last Name – Name must not include numbers

ID	7
Testing	Editing an existing student; Invalid Last Name
References	Requirement 4.3.2
Test Case	1. Enter "A really really long last name is not allowed " for last name
Expected Output	Error Message: Invalid Last Name – Name is too long

ID	8
Testing	Editing an existing student; Invalid Email Address
References	Requirement 4.3.2
Test Case	1. Enter "A " for email address
Expected Output	Error Message: Invalid Email Address – Email address is too short

ID	9
Testing	Editing an existing student; Invalid Email Address
References	Requirement 4.3.2
Test Case	1. Enter "astudentuwo.ca" for email address
Expected Output	Error Message: Invalid Email Address – Email address must contain an @ symbol

ID	10
Testing	Editing an existing student; Invalid Student ID
References	Requirement 4.3.2
Test Case	1. Enter "12" for student ID
Expected Output	Error Message: Invalid Student ID – Student ID is too short

ID	11
Testing	Editing an existing student; Invalid Student ID
References	Requirement 4.3.2
Test Case	1. Enter "1234567891011121314151617181920" for student ID
Expected Output	Error Message: Invalid Student ID – Student ID is too long

ID	12
Testing	Editing an existing student; Invalid Student ID
References	Requirement 4.3.2
Test Case	1. Enter "-*/6^^" for student ID
Expected Output	Error Message: Invalid Student ID – Student ID must not contain symbols

ID	13
Testing	Editing an existing student; Invalid Student ID
References	Requirement 4.3.2
Test Case	1. Enter "studentA" for student ID
Expected Output	Error Message: Invalid Student ID – Student ID must contain numbers

ID	14
Testing	Editing an existing student; Valid First Name
References	Requirement 4.3.2
Test Case	1. Enter "John" for first name
Expected Output	Accepted The user is asked to enter a last name

ID	15
Testing	Editing an existing student; Valid Last Name
References	Requirement 4.3.2
Test Case	1. Enter "Smith" for last name
Expected Output	Accepted The user is asked to enter an email address

ID	16
Testing	Editing an existing student; Valid Email Address
References	Requirement 4.3.2
Test Case	1. Enter "jsmith@uwo.ca" for email address
Expected Output	Accepted The user is asked to enter a student ID

ID	17
Testing	Editing an existing student; Valid Student ID
References	Requirement 4.3.2
Test Case	1. Enter "25069875" for email address
Expected Output	Accepted The user has successfully edited a student

Jenkins CI

Our team's CI server is up and running. It currently pulls the repository after each new commit, builds the code, and runs our tests. It is also set up with JaCoCo, proving line coverage for our tests.

Below is the password for the user 'ta', and the url of the server

Cs2212TA

<http://ec2-184-73-103-210.compute-1.amazonaws.com/>

Project Plan

Here is the group's plan; detailing what we had to carry out in order to complete assignment 3:

