

Summary

The purpose behind this project is to help students to assess themselves before they take graduate level courses (specifically designed for CSCE 625 Artificial Intelligence in mind). This software project can help students to better understand the course prerequisites and evaluate themselves if they are prepared to take the course.

The requirements for the project were provided by Dr Duncan Walker, Professor, Texas A&M University. The stakeholders for the project are prospective students, teaching assistants and the course instructors.

This project was taken up in Fall 2018 semester and major requirements were satisfied so we have to enhance the existing features and add some new features. The major customer requirements were:

1. Generating Statistics report for the instructor
2. Adding support for images for a question
3. Enhance the existing UI interface
4. Fixing the bugs of legacy code
5. Adding student login feature

User stories

User Story 1:

Complete implementation user login/signup.

As a student, I want to login into the portal to start course evaluation. This feature enables the student to create a login into the evaluation portal. The necessary signup/login pages with their functionalities (controller and models) were added over a span of 3 iterations to complete this feature. However, this feature is disabled as per Prof. Walkers request but not removed from code base (could be used for keeping track of student progress – low priority feature).

The image displays two side-by-side screenshots of web forms. The left form is titled 'Student Sign Up' and contains input fields for 'Username', 'Email', 'Password', and 'Password confirmation', followed by a blue 'Create My Account' button. The right form is titled 'Student Login' and contains input fields for 'Email' and 'Password' (with a 'forgot password' link), a 'Remember me on this computer' checkbox, a blue 'Log in' button, and a link for 'New Student? Sign Up!'.

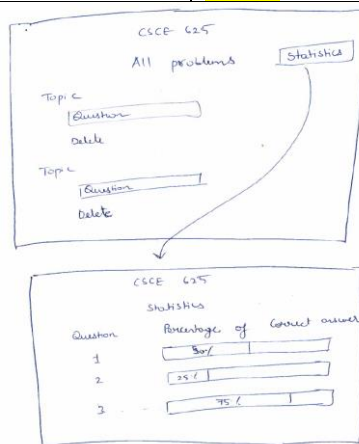
User Story 2:

Generate the statistics report of the student responses.

As the Instructor I want to see the number of responses and number of correct responses for each question. When the student finishes the evaluation, the responses must be recorded. This user story was implemented in all the four iterations. Two columns have been added in the database, one column is for number of responses, and the other is for the number of correct responses. These columns will get updated after the user submits the evaluation.

Database table for problems:

Old table		New table	
Data type	Value	Data type	Value
text	question	text	question
text	answer	text	answer
text	remark	text	remark
datetime	Created_at	datetime	Created_at
datetime	Updated_at	datetime	Updated_at
string	img	string	img
bigint	topic_id	bigint	topic_id
bigint	question_type_id	bigint	question_type_id
		bigint	num_of_attempts
		bigint	correct_attempts



Select Topic to display Statistics

Data Structures and Algorithms

Programming

Basic Mathematics

Tree Based Ensembles

← The instructor can select a topic to view statistics

Display All Statistics

Statistics for Data Structures and Algorithms

Which data structures are used for Breadth-first search (BFS) of a graph?
Total Number of Responses : 13 Correct Attempts : 7

shows percentage of correct answers for this question

53.85% Correct Answers

46.15% Incorrect Answers

Which data structures are used for Depth-first search (DFS) of a graph?
Total Number of Responses : 10 Correct Attempts : 3

30.0% Correct Answers

70.0% Incorrect Answers

Which one of the following uses LIFO (Last In First Out) for accessing elements?
Total Number of Responses : 7 Correct Attempts : 5

shows percentage of negative answers for this question

71.43% Correct Answers

28.57% Incorrect Answers

Which one of the following is non-linear data structure?
Total Number of Responses : 6 Correct Attempts : 1

16.67% Correct Answers

83.33% Incorrect Answers

Which one of the following is False about Array and Linked List?
Total Number of Responses : 8 Correct Attempts : 3

37.5% Correct Answers

62.5% Incorrect Answers

For a binary search algorithm to work, it is necessary that the array (list) must be?
Total Number of Responses : 5 Correct Attempts : 2

40.0% Correct Answers

60.0% Incorrect Answers

sample question

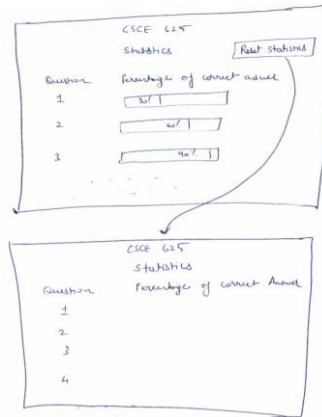
Total Number of Responses : 0 Correct Attempts : 0

No Responses yet

User Story 3:

Reset the previous statistics.

As the Instructor, I want to reset the statistics for each question recorded till now. This user story was implemented in iteration 4. The statistics values in the database will reset when we click reset statistics button.



CSCE 625

Click on Topic to Reset Statistics

Data Structures and Algorithms

Programming

Basic Mathematics

Tree Based Ensembles

instructor can select a
topic to reset statistics

Reset All Statistics

CSCE 625

Home Questions

Statistics for Basic Mathematics

Have you taken Linear Algebra course before?
Total Number of Responses : 0 Correct Attempts : 0

Statistics page for a topic after reset



No Responses yet

Have you taken Probability and Statistics course before?
Total Number of Responses : 0 Correct Attempts : 0

No Responses yet

$\sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6}$. What is x ?
Total Number of Responses : 0 Correct Attempts : 0

No Responses yet

When $a \neq 0$, there are two solutions to $ax^2 + bx + c = 0$ and they are
Total Number of Responses : 0 Correct Attempts : 0

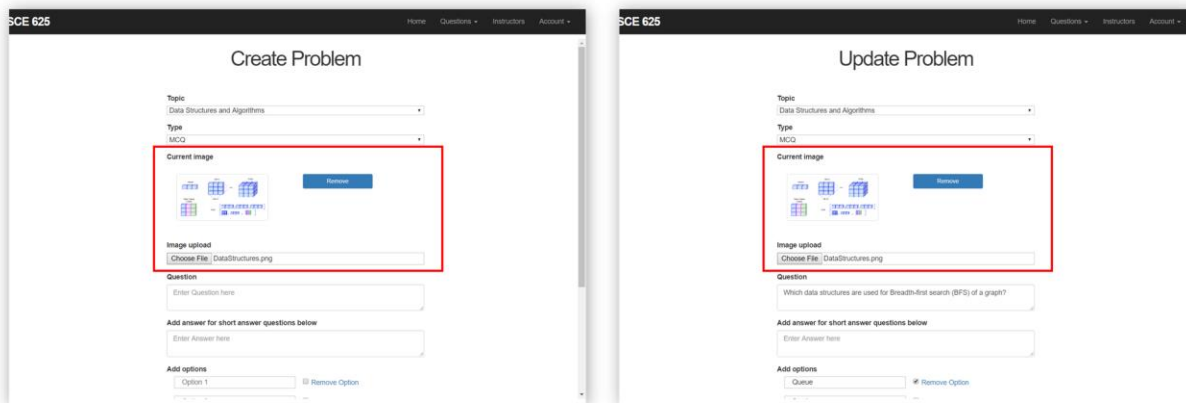
No Responses yet

User Story 4:

Image upload functionality to create/update problems

As the instructor, I want to upload relevant images for each question so that the student can understand the question clearly. The image upload functionality was completed in the final iteration 4. It uses base64 encoded string to store images in the database. Only images less than

64kB in size and format png, jpeg, jpg are allowed (these are tested with necessary tests and condition).



User Story 5:

Necessary bug fixes in legacy code.

Options bug fix

When you create/update a problem, if you remove all options in MCQ type and then click on Add options href link, it adds Add options href link instead of adding a valid option text box. This was a JavaScript error; the code was fixed to by enforcing limit of at least 2 MCQ options because a MCQ logically needs at least 2 options.

The image shows a detailed view of the 'Create Problem' form. It includes an 'Image upload' section with a 'Choose file' button and 'No file chosen' text. Below is the 'Question' field with the text 'For a binary search algorithm to work, it is necessary that the array (list) must be?'. The 'Add answer for short answer questions below' section has an 'Enter Answer here' field. The 'Add options' section contains four 'Add New Option' links. Below these links is a text input field containing the URL 'https://www.khanacademy.org/computing/computer-science/algorithms/binary-se'. At the bottom is a 'Save Changes' button.

Login bug fix

Legacy Product Understanding

Previous group reports and code documentation gave us necessary insights needed to successfully complete our project this semester. The legacy project is almost completed in its

final stages and requires minor bug fixes only. The database schema was designed and works as expected, the UI was also minimal and on point without any clutter. However, there were a few drawbacks such as the instructor will not be able to get a feedback on the test results because there is no tracking mechanism on test scores. As per customer's request, we added the statistics and reset pages to provide the instructor this core functionality. Another missing functionality was image upload by the instructor for problems that might involve graphs, models and trees, etc. To facilitate that image uploads feature was added to the create/update problems page. The legacy project has decent test coverage, we have further improved the test coverage to ----- . Overall the legacy project repository was well maintained, with correct links to project demo and application which overall helped in our understanding of the code base.

Team Roles

John Mathai Reji was the Product Owner for the entirety of the project. Venkata Naidu Marineni was the scrum master for iteration 0 to 2, Raja Mohamed Aarif was the scrum master for iteration 3 and Projna Paromita for iteration 4. Development and Testing tasks were taken up by all six members of the team.

Summary Iterations

This project consists of 4 iterations for this coursework.

Iteration 0:

In this iteration the main priority was to understand the customer expectations and get the necessary requirements to complete the project. The team worked on creating UI mockups and on drafting user stories.

Iteration 1:

The team started working parallelly on the user stories. In this iteration most of the time was spent on Legacy code walk through and identifying the potential files to change and planning of the tasks. The split up of the work is updated in Pivotal tracker. Database Schema was modified to accommodate user logins/signups and for the statistics page, two new columns were added to the problems table with necessary changes to the respective view pages.

Iteration 2:

Added necessary functions to user controller and user models. Created necessary view and links to the nav-menu-bar. Completed Statistics report, the instructor should be able to see percentage of students that have answered a question correctly. For this, added functionality such that when the user clicks submit evaluation, the corresponding two columns will be updated in the database.

Iteration 3:

The user login/signup was completed the views were connected to the backend database and all errors corrected. This feature however disabled in the current build (as per customers request). Completed the view pages of the view statistics feature in the current form.

Iteration 4:

Image upload feature was finished in this iteration, necessary changes made in update/create problems controller and respective view pages. This iteration saw more importance given to improving the test coverage. Reset statistics feature was implemented in this iteration. This feature was implemented by resetting the statistics values in the database, and by creating the necessary changes in the view pages.

Customer Meetings

The Customer meetings include our interactions with the TA (Gang Zhao) and our customer (Prof. Hank Walker)

Iteration date	Description
03/09/2019 (0)	Interviewed the customer to get requirements and know expectations
03/28/2019 (1)	Showed statistics page and got feedback from customer and TA
04/09/2019 (2)	Completed statistics page (with updated views) customer insisted on adding an image upload feature and concentrate on test coverage.
04/19/2019 (3)	Showed the TA full implementation of user login/signup page and its functionality.
04/30/2019 (4)	Completed Image upload feature and discussed details regarding the poster and showed the final project to the customer. Asked us to disable the user login for the final build but asked us not to remove it from the code base.

BDD/TDD process:

For behavior-driven development (BDD) and test-driven development (TDD), we have followed a similar technique introduced in the course. For acceptance testing, we have written `Cucumber` tests while for unit testing, we have written `RSpec` tests. However, initially, we have faced issues regarding executing the existing test files primarily because of failing to load the test database into the program. To overcome this issue, we have added files, e.g., `code/spec/fixtures/instructors.yml`, which prepares the test database (e.g., of instructors and problems) before executing the test files. Moreover, in the `code/features/step_definitions/web_steps.rb` file, we have defined several new steps to create (and verify) database of instructors and problems during the Cucumber test cases. After resolving this issue, we have rewritten many of the pre-existing tests files and included new tests for the latest features. Throughout the entire development process, we have

rigorously tested each of the newly added features to ensure that it is compatible with the legacy features as well as it meets the specification requested by the customer. To note, our current test coverage is ~80% with ~12 hits per line of code.

Configuration Management Approach:

We used the GitHub to manage the source code. The main branch is the 'master' branch. Each team member creates a branch from the master branch, works on it and creates a pull request, the scrum master will review the changes on the branch and merge the changes if everything works as expected. Once the master branch is updated the changes are pushed to heroku branch for production.

Issues with Production releases and Development (AWS, Cloud9 and GitHub)

We didn't face any issues with the Development and Production environments. Instead of image files we have used base64 encode images to store them in database. This is because if need to have many files then an AWS account is required (S3 bucket), because if the heroku dynos stop working the image files will be lost.

Links to

Final Customer Interview: <https://vimeo.com/325113255>

Application Demo: <https://vimeo.com/334559598>

Pivotal tracker: <https://www.pivotaltracker.com/n/projects/2317545>

GitHub: <https://github.com/NaiduMarineni/CSCE625-Self-Evaluation-Portal-v2.0>

Heroku: <https://arcane-reef-17079.herokuapp.com/>