

CSCE 606 Project: Final Report

Team Name: AppleSauce

Team Members:

Amulya Agarwal

Yerania Hernandez

Siru Li

Kevin J Nguyen

Yang Yang

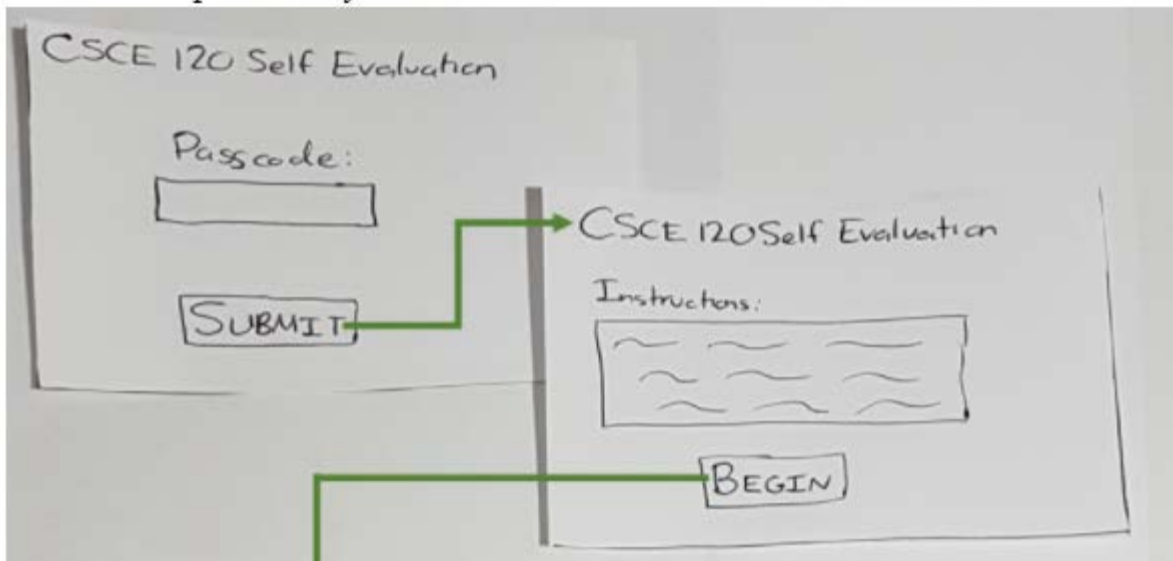
Summary:

The need for the CSCE 120 Self-Evaluation Exam is to evaluate incoming student's knowledge on introductory programming concepts. Dr. Shell, being the main customer for this project, is in need of an application that students will take before the start of the course and will help him and the students understand how much they know and how much more work they need in order to be prepared for the course. As the course requires students to have little programming experience, he is looking for an application that could test various aspects in basic programming. The main stakeholders of this project will be the incoming students as they will use the system that will provide an evaluation on their readiness to take the course. Future professors will have the ability to decide what changes and adjustments are needed to the introductory material and should be covered so that most of the students successfully complete this class.

The application we developed has two main aspects to it: (1) an administrative side for the professor and (2) a client side for the students. The administrative side has the ability to create the questions and evaluations for the students. Each answer choice in the question is scalable and the admin has the ability to select between two to five answer choices as part of the question. Each evaluation also has a unique access code in order for a student to take a specific evaluation. In addition, the admin can export data from each of the students, such as their scores, and export data about the questions such as the number of times a specific answer choice was selected. The client side for the student has the ability to see their score of the latest attempt, change their section in case they switch class sections, and an input for the unique access code to take the evaluation. All the questions are multiple choice, but the admin has the ability to add images, code snippets, tables, and other extensive features in HTML tags. With a variety of features and functionality implemented, we were able to provide Dr. Shell the system he requested.

User Stories:

1. Setup Ruby/Rails project
 - Points: 2 points were given to this story considering we needed an environment to work with and documentation on how each member could setup the same environment.
 - Implementation Status: Completed with no changes
2. Create and setup initial question database
 - Points: 2 points were given to this story considering this provided the initial student relationships for all our members
 - Implementation Status: Completed with no changes
3. Create account with Student UIN
 - Points: 2 points were given to this story considering it required researching what type of login system would be used
 - Implementation Status: Completed with modifications
 - This feature was changed from originally having student's enter the specific passcode in order to log into the evaluation. Using the UIN however, provides an easier interface and would not have to provide encryption for passwords and such at the moment.



CSCE 120 Online Assessment

Student Login

UIN:

Login

Register

4. View registered students
 - Points: 1 point was given to this story considering it was only a view to properly display the students database table to the administrative side
 - Implementation Status: Completed with no changes
5. View the course evaluation results
 - Points: 1 point was given to this story considering it was only a view to properly display the evaluation results from the database
 - Implementation Status: Completed with no changes
6. Create evaluations
 - Points: 3 points were given to this story considering this would allow the administrative side to create an evaluation by selecting questions or allowing the system to randomly select non-repetitive questions. In addition, it would save these selected questions as separate evaluations and a separate database table was created to keep track of this information.
 - Implementation Status: Completed with modifications
 - Bug: New Evaluations Title
Fixed the ability to keep the title even though clicking on random button

■ Bug: New Evaluations Save Button

Fixed the ability of admin saving an evaluation despite missing a parameter and displaying a flash statement if a parameter is missing.

Create new question

Number of Choices

Add Question

Enter new evaluation

Title

Random Q's: Random

☐ 1001

☐

☒

☒

☒

☐ 1002

☐

☒

☒

Save

CSCE120 Assessment Evaluation Manager

ID	Title	Q IDs	Access Code	Change AC	Delete
----	-------	-------	-------------	-----------	--------

Add new evaluation

Instructions for Evaluations:

Update

Enter the new evaluation

Check next to each question to enable that question for the new evaluation. Select from the scale option for each answer to adjust the number of points each answer is worth.

Title:

How many random questions do you want?

☐ Question: 1

Which of the following correctly declares an array?

Scale:

```
int array[10];
```

Scale:

7. Add questions to database
 - Points: 3 points were given to this story considering we needed the ability to display code, images, tables, and any additional formatting for the administrator. This would also have the ability to select the different answer choices and the correct answer. This implementation would play a strong part in editing and removing a question.
 - Implementation Status: Completed with no changes
8. Generate access link and password for evaluation
 - Points: 2 points were given to this story considering we needed the ability for certain students to be able to begin an evaluation, but should only be accessed through an access code.
 - Implementation Status: Completed with modifications
 - This feature has been changed from originally having passcodes and specific URL links to providing a personalized access link once the student has actually registered and logged into their account. This removes the need of scripting emails for each student and provide the possibility of repeating an evaluation test. This feature was further developed later through the use of access codes to allow the administrative side to disable and enable evaluations.

Evaluation Manager

Current Evaluations:

ID	Title	QuestionIDs	Access Code	Change Access Code	Delete
1	Evaluation 1	[1, 3, 8, 9, 10]	test1	<input type="text"/> <input type="button" value="Update"/>	Remove

9. View initial homepage for student

- Points: 1 point was given to this story considering it was only the landing page for the student after they log in, providing their latest score and the ability to begin an evaluation.
- Implementation Status: Completed with no changes

10. Create admin login

- Points: 2 points were given to this story considering researched needed to be done to figure out how to allow the admin access to the administrative side and not allow students the ability to login along with the implementation of this login system.
- Implementation Status: Completed with modifications
 - The feature has been changed from originally logging in with a username and password to being able to use a key access code for the administrators.

CSCE 120 Online Assessment

Admin Login

Key:

11. Create and remove sections

- Points: 2 points were given to this story considering various views and controllers needed to be adjusted for the student and admin to change the section of a specific student based on an existing section from the database along with allowing the admin to add sections or delete sections.
- Implementation Status: Completed with no changes

12. Edit a question from the database

- Points: 2 points were given to this story considering the basic implementation was made by adding a question, but it needs to keep track of these changes and allow only that specific question to change
 - Implementation Status: Completed with no changes
13. Remove a question from the database
- Points: 2 points were given to this story considering the basic implementation was made by adding a question, but it needs to keep track of the removal of this question id and only remove this specific question.
 - Implementation Status: Completed with changes
 - This feature was later updated in order to adjust for the fact that if a question is removed, then the question should also be removed from any existing evaluation that was made that contains this question. This would be the proper removal of a question since it could also exist in an evaluation.

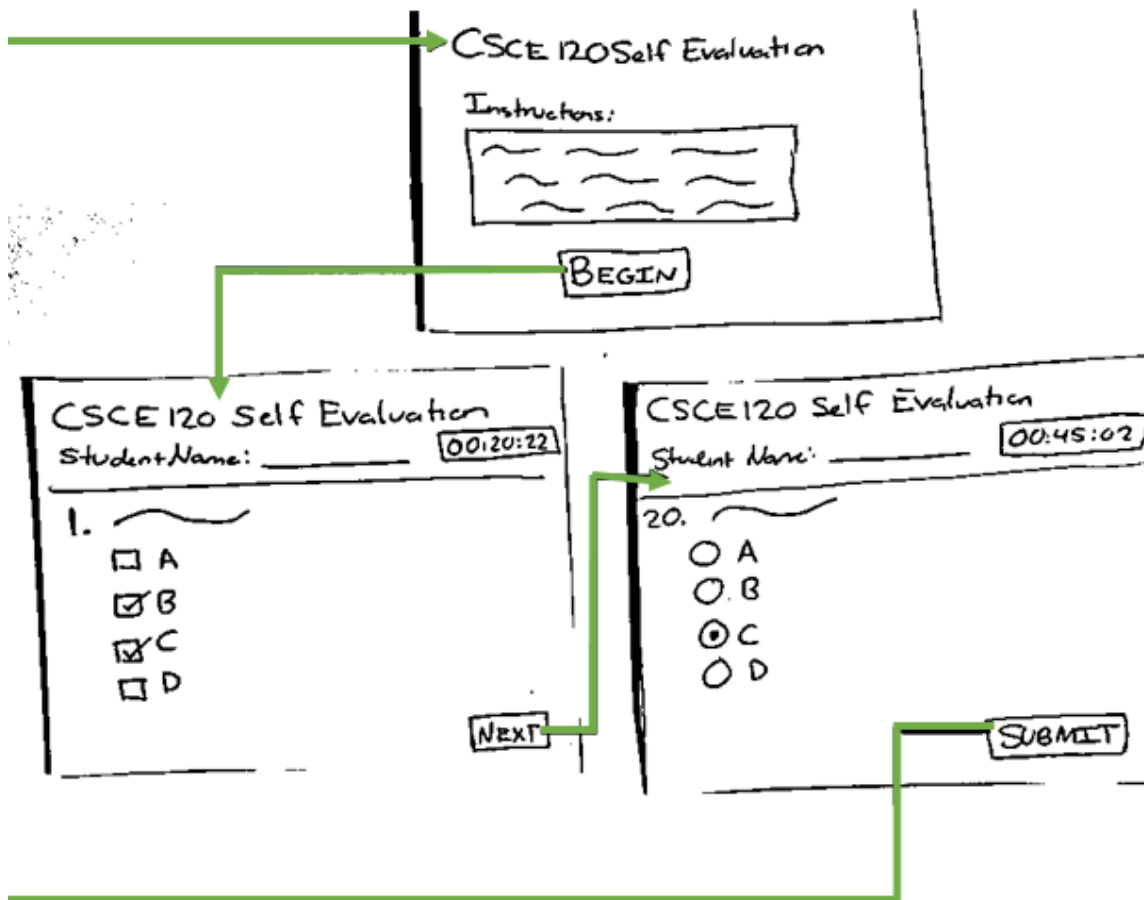
Question Manager

Create, update, and remove questions.

ID	Content	Answer	Choice 1	Choice 2	Choice 3	Choice 4	Choice 5	Delete	Copy
1	<p><p>Which of the following correctly declares an array?</p></p>	<p><p></p>	<p><p></p>	<p><p></p>	<p><p></p>	<p><p></p>		Remove	Copy

14. View for students test
- Points: 2 points were given to this story considering this actually was the evaluation the student will take and what they will see to take the test, along with the score they will receive after submitting. They had to be able to click next between questions, not be able to return, see all answer choices even when two, three, four, or five options, and any instructions the admin has decided.
 - Implementation Status: Completed with modifications
 - Bug: Student evaluation notice
Fixed the flash notice that shows up for the student evaluation throughout the entire application
 - Bug: Student evaluation choice selection
Fixed the ability to save each selected answer choice of the student into the question database and doing the proper score calculation for a student's evaluation.
 - Bug: Student evaluation back button

Fixed the ability of forcing the student to select an answer to the specific question, not return back to a previous question, and show the next question despite pressing the back button.



Student Information

UIN	Name	Section	Attempts	Score	Change Section
123000123	Ruth Morris	501	0	-1/0(-1.0)	<div>500 ▾</div> <div>Update</div>

Self-Evaluation Exam

Warning: Don't press the back arrow, your choices will be saved after you click "Next" and "Submit"!

Which of the following correctly declares an array?

☐ `int array[10];`

☐ `int array;`

☐ `array{10};`

☐ `array array[10];`

Next

15. Display questions in evaluation manager

- Points: 2 points were given to this story considering displaying the evaluation was necessary for the admin to see what existed, what choices they had selected, and the scaling of the questions.
- Implementation Status: Completed with no changes

16. Copy question

- Points: 2 points were given to this story in order to research and implement the ability to copy a specific question with a new question id that did not have a relationship with the original question, allowing the copied question to be updated and changed.
- Implementation Status: Completed with no changes

17. Selection of answer options

- Points: 2 points were given to this story considered it would take additional changes to the database, views, and research on how to implement the ability of having different answer choices (2, 3, 4, or 5) and selecting one of these as the answer.
- Implementation Status: Completed with no changes

18. Question scale

- Points: 2 points were given to this task considering it required various changes in the student and administrative sides. The scoring for a student evaluation would be based on the question scale and the admin should have the ability of providing a different scale for each answer choice in each of the questions selected for the evaluation.
- Implementation Status: Completed with no changes

19. Access code for evaluation

- Points: 2 points were given to this task considering the original access code for all evaluations was changed to have an access code for each evaluation. This required changing the various views from the admin side, the ability to update specific code for the evaluation, and the code should correlate to a specific evaluation.
- Implementation Status: Completed with modifications
 - Bug: Back button breaks sessions
Fixed the error messages after logging out for a student or a professor and pressing the back button, it will properly redirect to the admin or student login view.
 - Bug: Flash warnings
Fixed the flash warnings that keep showing up in different views by standardizing all the flash statements to warnings and success messages instead of notices.

Evaluation Manager

Current Evaluations:

ID	Title	QuestionIDs	Access Code	Change Access Code	Delete
1	Evaluation 1	[1, 3, 8, 9, 10]	test1	<input type="text"/> <button>Update</button>	Remove

[Add new evaluation](#)

Begin Test

In order to take the evaluation, the instructor will provide you an access code. Once you obtain the access code, enter it below and press begin.

Access Code:

[Begin](#)

20. Database updates evaluation instructions

- Points: 1 point was given to this task considering instead of having a view for personal instructions, the student will see the instructions before the evaluation begins and in addition the admin would be able to change the instructions that can be updated.

- Implementation Status: Completed with no changes
- 21. Remove evaluations
 - Points: 1 point was given to this story considering it was a link added to remove an individual evaluation from the database
 - Implementation Status: Completed with no changes
- 22. Export data results
 - Points: 3 points were given to this story considering different data about the questions and the students should be exported. Research and implementation needed to be done in order to complete this task on how to export the tables for each student and for the questions. This story provides the ability to deselect the student while exporting the data. In addition, specific student information should be exported and excluded from the export.
 - Implementation Status: Completed with no changes
- 23. Styling application
 - Points: 2 points was given to this story considering our application was not concise in coloring, buttons, headings, or any type of styling except minimal tables and options. As a result, we left styling for the final iteration and agreed on a specific style to modify our entire application and every view we have.
 - Implementation Status: Completed with no changes

Team Roles:

For all the iterations, the team roles were the same as described below.

- Scrum Master: Yerania Hernandez
- Product Owner: Kevin J Nguyen

Scrum Iteration Summaries:

- *Iteration 1:* In our first iteration, we were able to accomplish setting up the Ruby/Rails environment along with a frontend template of the site we wanted to create. After being able to setup our environment on AWS Cloud9, our team also created the initial student and question database in order to establish the different parameters and relationships needed. In addition, we set up a login page for the students (client side) and provided the initial view for the professor (administrator side).
 - *Points Completed:* 10
- *Iteration 2:* In our second iteration, we focused on the administrative side of our system by providing the functionality of adding evaluations, adding questions, and generating an access code to access the evaluation. For the evaluations, we had to create a new database table for the different parameters and correlate between the existing questions in the database along with having the ability of randomly selecting questions to create an evaluation. We also focused on creating an access code for the administrator to select and change to enable and disable the evaluation, and for a student to be able to use this access code to take that specific evaluation it correlated to. As a result, from the student side we were able to finish displaying the initial page after logging in and being able to login to a specific evaluation with the access code.
 - *Points Completed:* 9
- *Iteration 3:* In our third iteration, we began to divide our focus between the administrative side and the client side considering both were being integrated with one another. On the administrative side we began developing a login page, which ended up consisting of a key access code. We also implemented the ability of being able to edit and remove questions considering we had the basic functionality of adding a question. From the administrative and student side, we

also implemented creating and removing sections in order for students to register within a specific section and from the student side, they are able to update their section in case they change schedules in the future. Lastly, we focused on the implementation of the view for the student's evaluation. This component needs to show a specific evaluation, only allow to go next and not back, not allow for changes in questions despite going back, and after submitting, scoring the evaluation and displaying that new score on the initial login page for the student, all of which were accomplished.

- *Points Completed:* 10
- *Iteration 4:* In our last iteration, there was a variety of stories that came our direction after discussing with our client, implementations that could be accomplished but just required time to work through. As a result, from the administrative side, we were able to have the ability of selecting multiple answer choice options, where it could range from 2, 3, 4, or 5 answer options, but only one selection for the correct answer. We also included a question scale that was implemented within the creation of the evaluation which had a variation of points. The access code was redesigned to be specific for each evaluation, allowing the admin to change the code per evaluation and the student being able to login into that specific evaluation. Our last main feature was the ability of exporting data results, meaning that we saved the student's answer selection per question and the scores for all the students for further data analysis as the administrator desires. In addition, we implemented copying a question, removing evaluations, updating evaluation instructions from the administrative side, and displaying the questions for each evaluation after their initial creation along with styling and fixing any additional bugs that we found on the way in order to finalize our system.
 - *Points Completed:* 17

Customer Summaries:

- Saturday, March 3rd, 2018
 - For Iteration 0, we discussed the purpose of the project, the expectations Dr. Shell, our customer, had for our team, and what he wanted to have at the end as a final product. We did not have a demo considering it was a meeting to learn more details about the project.
- Monday, March 19th, 2018
 - After the completion of iteration 1, we met with Dr. Shell and discussed our current progress. At this point we had doubts about how to proceed with administrative login and what he found acceptable for this aspect before we began. Afterwards, we demoed the tasks described in the *Iteration 1 Summary* section. Dr. Shell gave us feedback on other details he wanted, such as being able to have questions with HTML, images, and code style.
- Friday, April 6th, 2018
 - After the completion of iteration 2, we met with Dr. Shell and demoed the tasks described in *Iteration 2 Summary* section. In this meeting, we addressed our implementation of the access code compared to using a URL link in order for the student to have a personalized evaluation. Dr. Shell agreed with this implementation and requested the ability to have some data information about these questions students take and their scores. In addition, he requested the ability to export such data for selected students.
- Saturday, April 21st, 2018
 - After the completion of iteration 3, we met with Dr. Shell and demoed the tasks described in *Iteration 3 Summary* section. In this meeting, we discussed our implementation of administrative login and we discussed the future possibilities. He also requested for their to be an access code per evaluation and for each question to have a scale for each answer option. Lastly, we discussed what we would deliver to him at the end of our iterations and

that we agreed to provide him a service in which he could actually use this system, not simply just the code.

- Tuesday, May 1st, 2018
 - After the completion of iteration 4, we met with Dr. Shell and demoed the tasks described in *Iteration 4 Summary* section. In this meeting, we discussed our implementation of the scaling of questions, the ability to select the number of choices for a specific question, access code for a single evaluation, and the information he could export from the student and question database. In addition, we explained how he will deploy the site on to Heroku in order to have a working product. Lastly, we thanked him for working with us and requested him to fill out the final survey.

Discussion:

- *Explain your BDD/TDD process, and any benefits/problems from it.*

Each member was in charge of their test cases based on the user story they implemented. Behavioral test cases were written through cucumber and functional test cases were written through Rspec, in addition to further testing being implemented in the code. The features were each individually tested by another member who did not program this specific feature and attempted to find bugs, break the code, or find something wrong with the logic of the functionality. At certain times, these test cases were beneficial because it actually caught if indeed the functionality was properly implemented and narrowed the amount of time necessary to debug where the error was coming from considering the test case would demonstrate where the issue was coming from. A main issue was the limitation these test cases caused. Our team various times wanted to use JavaScript to accomplish a certain task, but after extensive work and research we had to scrap the work we done and restart the task in some other way than JavaScript because we were not able to find a way to test the functionality. With JavaScript, we were unable to provide 100% test coverage, yet the functionality was indeed there. Despite the importance of test coverage, there are definitely some advantages and disadvantages to this process.
- *Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?*

For our configuration, we all setup the same environment, as shown in our Github documentation. In addition, for each user story, there was separate branch created which was branched off from the develop branch. Each member worked on their indicated branch and when they were ready to merge they would push into develop. Each branch was conventionally named “feature/user-story-name” from the pivotal tracker. After merging and after each release, all these branches were removed considering the task had been completed. In addition, any bugs were named “bug/bug-name” from the pivotal tracker. We did not set aside any story as only a spike. Any question or gathering of information that was needed was included within each of our user stories and tasks considering the information was needed in order to accomplish the task. We had 40 different branches created throughout our four iterations and four releases that were tagged.
- *Discuss any issues you had in the production release process to Heroku.*

In the production release process, we were able to successfully deploy to Heroku every iteration considering we did the initial setup with AWS Cloud9 and verified that it worked on that server. As soon as it did, we would simply merge at the end of each iteration our code into the master’s branch and finally deploy to Heroku. There were no other issues that we found.
- *Describe any issues you had using AWS Cloud9 and GitHub.*

Our initial setup on AWS Cloud9 was definitely difficult and required extensive research and documentation in order for all our team members to setup the same environment. Our documentation is included in the ReadMe file on Github, but the most difficult aspect was setting

up Cloud 9's Postgresql considering it required changes in the configuration files, in the port settings, and creating a development database in order to finally have the ability to run the server locally through Cloud 9. After this initial setup, we only had the usual merge issues from Github considering we used different branches and finally branched into the master for deployment.

- *Describe the other tools/GEMs you used, such as CodeClimate, or SimpleCov, and their benefits.*
We used a number of tools to help us implement various aspects of our project and to help analyze the test coverage of our controllers and models as well. For testing purposes, we used SimpleCov, Cucumber, Rspec, and Capybara in order to verify through each different feature and function had 100% coverage and that when we integrated our different parts at the end of each iteration, we still maintained 100% coverage through each of our test cases. In order to display questions in more than just an HTML format, we used TinyMCE to provide a rich tech editor with different editor features such as adding images, code snippets, tables, formatting, lists and much more. We used this tool to store the questions, edit questions, and view the questions from the admin and client side. To stylize our site, we used Bootstrap as a template for our site and further modified it at the last iteration to represent Texas A&M. Another tool we used was Seed Dump, which allowed us to create seed data files from the existing data in our database and use it as our final seed for our demo. Lastly, we used Export-To-CSV in order to provide the ability to the admin side of exporting different tables we have in our database to a CSV for further data analysis.

Project Links:

- Link to Customer Video: <https://vimeo.com/267489119>
- Link to Application Demo: <https://vimeo.com/267494611>
- Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2153120>
- GitHub Repo: <https://github.com/hyerania/Course-Evaluation-System>
- Heroku Deployment: <https://enigmatic-cliffs-53495.herokuapp.com/>