

Project Report

ExamIT

AUTOMATIC TEST CORRECTION PLATFORM

4th year, Software Development project
Institute Of Technology Carlow

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Roger Marciniak

Student Number: C00169733

Supervisor: Paul Barry

3rd April, 2017

Table Of Contents

Table Of Contents	1
Introduction	2
Project Description	3
Login Screen	3
Menu	4
Categories	5
Display Questions	6
Add Question	7
Display Tests	8
Generate Test	9
Print Test	11
Printed Test Example (Blank Answer Book)	12
Printed Test Example (Filled Answer Book)	13
Printed Test Example (Question Sheet)	14
Correct Test	15
Test Results	16
Annotated Test Result (available for viewing after clicking)	17
Conformance to Specification and Design	19
Learning	20
Technical Achievements	20
Personal Achievements	20
Project Review	21
Advice for Attempting a Similar Project	23
Evaluation of Technology Choices	23
Acknowledgements	24



Introduction

This is the final report for the ExamIT web application project done as part of the Fourth year of Software Development course at the Carlow Institute of Technology.

ExamIT is a web application with the aim of helping lecturers assess their students using simple, paper based, multiple choice tests automatically generated by the ExamIT system. With it the lecturer is able to build a library of questions which they can then use to generate tests. The tests are turned into PDF documents which can then be printed and handed to students for completion. When the students complete the tests the lecturer can scan them in using the institutes scanners and upload them to the ExamIT system for automatic correction.

This document will explain the details of the project including conformance to specification and design, what went well and what didn't, what were the learning outcomes and achievements.

Lastly, the document will offer a piece of advice to someone attempting a similar project and will evaluate the technology choices.

Project Description

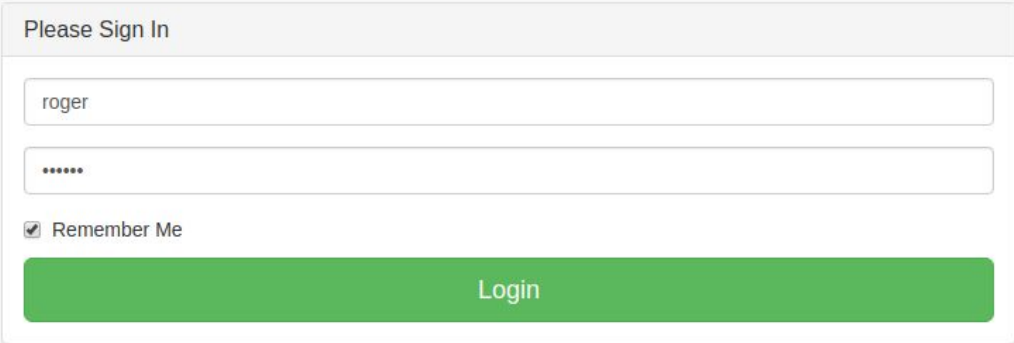
In this section, I will describe the functionality of my project. These will take the form of screenshots of the system with explanation in text form.

The visual aspect of the application was achieved using Bootstrap elements. The app was designed with different devices in mind, therefore it is just as easy to use using a tablet or a different small screen device. The app is responsive and moves the elements vertically for mobile phones.

Login Screen

The login screen was purposely kept simple. The goal was to achieve the basic login functionality and move on to the core functionality of the project.

Here, the lecturer is able to access the admin section of the system by entering their credentials and clicking 'login'. It is important to note that nearly all of the functionality of the system is accessible through the admin section available only through logging in.



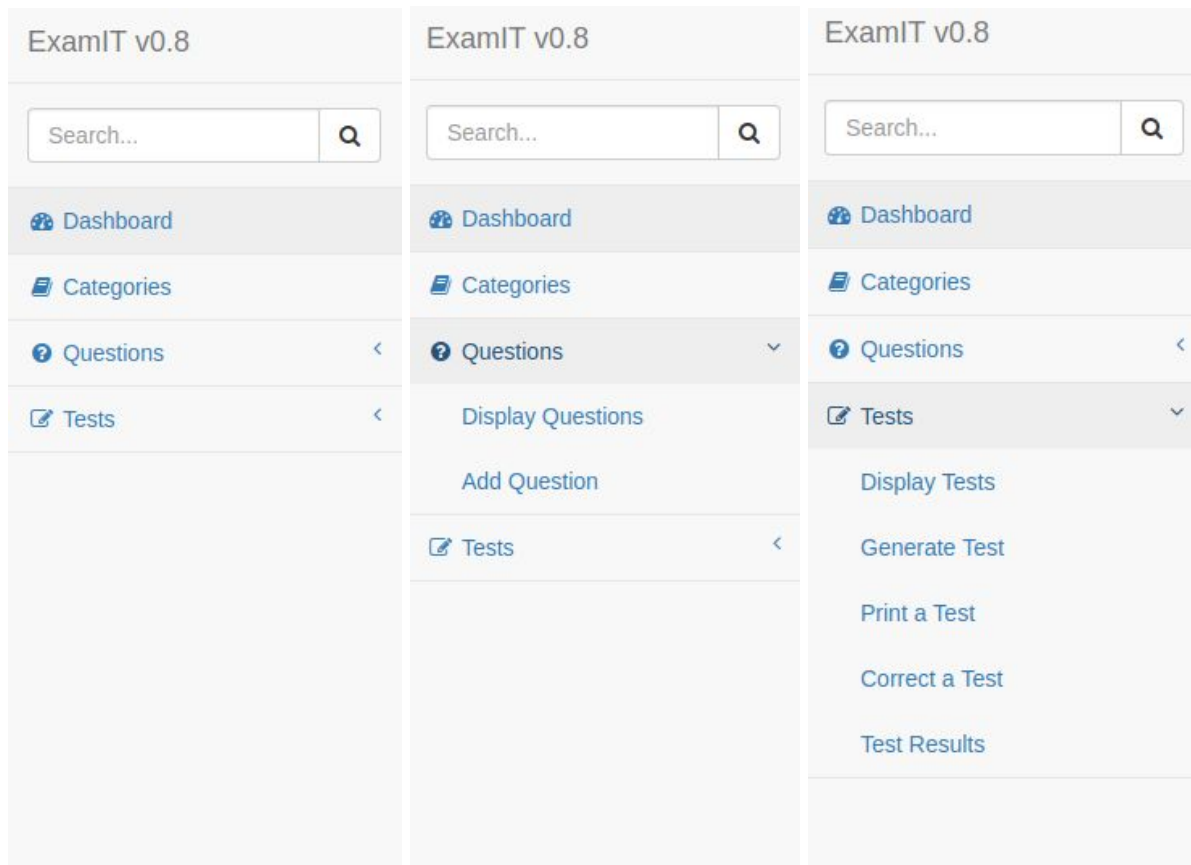
The screenshot shows a login form with the following elements:

- Title: Please Sign In
- Username input field: Contains the text "roger".
- Password input field: Contains six dots for masking.
- Remember Me checkbox: Checked.
- Login button: A green button with the text "Login".

Menu

The menu is always available in the sidebar and allows for the navigation of the web application. All of the operations available in the web application are accessible via the menu, some have multiple steps that are not listed in the menu but are available after completing a step. For example, when attempting to generate a test, a confirmation view appears at the end.

Below are three screenshots of the menu with the basic appearance, the 'questions' opened up and next, the 'tests' section opened up.



Everything nested

Questions exposed

Tests Exposed

Categories

Categories view consists of two separate panels, the top one used for adding a category and the bottom one for displaying the available categories. This was initially split up but since the two are very simple and do not take a lot of screen space, they were put together for convenience.

The categories are used to categorise questions and tests which will be explained in more detail later.

Categories

Add a category

Category:

Categories

Category	Creation Date	Unique ID
electronics	Thu, 16 Feb 2017 13:27:11 GMT	58a5a8af35f23f071e50d042
mathematics	Thu, 16 Feb 2017 13:27:17 GMT	58a5a8b535f23f071e50d046
graphics	Thu, 16 Feb 2017 13:29:18 GMT	58a5a92e35f23f071e50d067
web development	Thu, 16 Feb 2017 13:29:42 GMT	58a5a93f35f23f071e50d06f
logic	Thu, 16 Feb 2017 13:33:35 GMT	58a5aa2f35f23f071e50d0aa
datastructures	Thu, 16 Feb 2017 13:38:11 GMT	58a5ab3135f23f071e50d0e2
networking	Thu, 16 Feb 2017 13:38:51 GMT	58a5ab6b35f23f071e50d0f5
programming	Thu, 16 Feb 2017 13:39:43 GMT	58a5ab9735f23f071e50d101

Display Questions

Displays a table of all the questions previously generated in the system. Basically, when a lecturer creates a question, it appears here.

Questions are used to fill up a test. Each question object contains the question itself, the possible answers and the correct answer key.

Questions

Question Collection				
Question	Category	Key	Creation Date	Unique ID
question1?	electronics	A	Thu, 16 Feb 2017 14:06:41 GMT	58a5b1f135f23f071e50d27c
The schmitt trigger may be used to?	electronics	C	Thu, 16 Feb 2017 14:49:33 GMT	58a5bbfd35f23f071e50d4bd
Which of the following is minimum error code?	electronics	B	Thu, 16 Feb 2017 14:53:34 GMT	58a5bcee35f23f071e50d4f8
A simple flip-flop	electronics	B	Thu, 16 Feb 2017 14:57:16 GMT	58a5bdcc35f23f071e50d534
Radix of binary number system is ____?	electronics	C	Thu, 16 Feb 2017 14:58:11 GMT	58a5be0335f23f071e50d542
Which of the following expressions is in the sum-of-products (SOP) form?	electronics	A	Thu, 16 Feb 2017 14:59:25 GMT	58a5be4d35f23f071e50d554
The systematic reduction of logic circuits is accomplished by:	electronics	B	Thu, 16 Feb 2017 15:00:48 GMT	58a5bea035f23f071e50d567

Add Question

This view allows the lecturer to create a new question and add it to the collection of questions. Later on, the lecturer can use the questions to put them on a test.

The questions are created separately and added to the question collection instead of letting the lecturer create the questions on test creation, because this way the questions can be reused and do not have to be rewritten every time. Also, other lecturers can use another lecturers questions. Sometimes college modules are so similar that the same questions might fit multiple modules.

Add Question

Fill in the form to add a question

Category: Category missing?
datastructures ▼

Question:
What is a current? (up to 200 chars)

Answers:

Answer A

Answer B

Answer C

Answer D

Answer E

Correct answer key:
A ▼

Add question Reset

Display Tests

Displays all the available tests previously generated in the system. When a lecturer creates a test, it appears here.

Tests contain information like the lecturer, time allowed, module and so on, as well as a number of questions. Currently, tests support multiples of five questions.

It is important to note that the questions here, are not referred to by id alone, they are copied directly in their complete form. This causes the same information to be saved more than once but is essential. This is because, if a lecturer deletes a question from the question collection, if the question would be used in a test, that test would be broken.

Tests

Generated Tests							
Test	Lecturer	Time Allowed	Module	Questions	Category	Creation Date	Unique ID
Web Dev Quiz - Feb 19	Roger Tester	10 minutes	Web Development	5	web development	Sun, 19 Feb 2017 18:28:02 GMT	58a9e3c5480879efef699939
Spring Electronics Assessment	Frank Shocke	25 minutes	Advanced Electronics	5	electronics	Sun, 19 Feb 2017 19:01:48 GMT	58a9eb39480879efef699ac3
Electronics Basics Assessment	John Marks	25 minutes	Advanced Electronics	5	electronics	Thu, 23 Feb 2017 10:27:49 GMT	58aeb93df2afa3ae1495df4b
democat test	Demo Lecturer	5 mins	Demo Module	5	demo category	Sat, 25 Mar 2017 17:18:42 GMT	58d6a689939af12074f6c434
Electronics Christmas Assessment	Roger Spark	15 minutes	Applied Electronics	15	electronics	Sat, 01 Apr 2017 16:11:14 GMT	58dfc37dfdb041e8360dda61

Generate Test

This view, analogically to the 'Add Question' view, is used for the creation of tests.

Notice the 'Amount of questions' select field at the bottom of the screenshot. If the lecturer selects more questions than a category contains, then they will be notified that they cannot generate the test.

In that case, the lecturer has to either add more questions to the category before generating the test or choose a different category.

Generate Test

Fill in the form to generate a test

Test title:

Time allowed:

Lecturer:

Module:

Category: [Category missing?](#)

Amount of questions:

This is a confirmation view displayed when the system confirms that there are enough questions and the form was completed correctly.

Confirm Test Generation

Title: Quick Web Recap
Time allocated: 10 minutes
Module: Web Development
Lecturer: Paul Barry
No. of questions: 5
Questions: <ol style="list-style-type: none">1. What is the correct HTML tag for inserting a line break?2. Which of these is not a Python web framework?3. Which tag inserts a line horizontally on your web page?4. A webpage displays a picture. What tag was used to display that picture?5. How can you make an e-mail link?

If the lecturer confirms their choice, they will be brought to a view where they can select what to do next and will be notified of the success by the green alert at the top.

Test was successfully generated! ×

Test Generation

Test generation complete!
If you would like to continue working with tests, you can:

Display all tests
Generate another test

Print Test

Here, the lecturer can use the select field to choose a test from the collection of previously generated tests. This means that once a test is created it can be reused multiple times by them or even by a different lecturer.

Print Test

Choose a test for printing

Pick a test:

Electronics Christmas Assessment ▼

✓ Choose this test

When the lecturer clicks 'Choose this test', the test will be translated from the system onto a newly generated PDF document which will display in the browser PDF viewer in which usually PDF documents display. On the next page, an example of what a printed test looks like is shown.

Printed Test Example (Blank Answer Book)

ExamIT - Copyright 2017

C001

STUDENT NUMBER

ANSWERBOOK

Title: Electronics Basics
Time: 20 minutes
Lecturer: John Marks
Module: Advanced Electronics
Questions: 15

1 A B C D E

2 A B C D E

3 A B C D E

4 A B C D E

5 A B C D E

6 A B C D E

7 A B C D E

8 A B C D E

9 A B C D E

10 A B C D E

11 A B C D E

12 A B C D E

13 A B C D E

14 A B C D E

15 A B C D E

Printed Test Example (Filled Answer Book)

ExamIT - Copyright 2017

C001
STUDENT NUMBER

ANSWERBOOK

Title: Electronics Basics
Time: 20 minutes
Lecturer: John Marks
Module: Advanced Electronics
Questions: 15

- 1 A B C D E
- 2 A B C D E
- 3 A B C D E
- 4 A B C D E
- 5 A B C D E
- 6 A B C D E
- 7 A B C D E
- 8 A B C D E
- 9 A B C D E
- 10 A B C D E
- 11 A B C D E
- 12 A B C D E
- 13 A B C D E
- 14 A B C D E
- 15 A B C D E



QUESTION SHEET

1. Which tag inserts a line horizontally on your web page?
 - a. `<hr>`
 - b. `<line>`
 - c. `<line direction="horizontal">`
 - d. `<tr>`
 - e. `<th>`
2. `` tag makes the enclosed text bold. What is other tag to make text bold?
 - a. ``
 - b. `<dar>`
 - c. `<black>`
 - d. `<emp>`
 - e. `<color>`
3. How can you make an e-mail link?
 - a. ``
 - b. `<mail href="xxx@yyy">`
 - c. `<mail>xxx@yyy</mail>`
 - d. ``
 - e. `<email:xxx@yyy>`
4. How can you make a numbered list?
 - a. `<dl>`
 - b. ``
 - c. `<list>`
 - d. ``
 - e. `
`
5. A webpage displays a picture. What tag was used to display that picture?
 - a. `picture`
 - b. `image`
 - c. `img`
 - d. `src`
 - e. `br`

Test completion guidelines

1. Only use pen inside the answer circles and for student no.
2. Try not to go outside the circles when marking them
3. Try to mark only one circle in each row
4. The circle filled in more, will be considered your answer
5. Only return the answerbook

Correct Test


This view is used for correcting scanned tests. To scan a test, the lecturer needs to use the select field to specify which test is being scanned and then browse their computer for the files.


Correct Test

Which test do you wish to correct?

Select test:

Web Dev Quiz - Feb 19

 Browse

 Upload for Correction


After a test is corrected the lecturer is notified by a green alert about the success of the correction. They can then use the menu to access the test results.


File was corrected. Visit 'Test Results' to see scores

Which test do you wish to correct?

Select test:

Web Dev Quiz - Feb 19

 Browse

 Upload for Correction

Test Results

In this view the lecturer has access to a list of tests that were corrected. In this case the screenshot only displays one but in the usual scenario there would be multiple paginated tests to choose from.

Corrected Assessments

Corrected Assessments

Select one of the corrected assessments:

Electronics Christmas Assessment

When a lecturer selects one of the tests they can see a list with the results of the test. In this case, three copies were corrected, each of the students scored 15/15 and received 100%. Usually there would be more results with different scores. A result can be selected to view a copy of the answer book with annotations, so the lecturer can double check that everything is fine. In the case where the algorithm can't correct the test, there would be 'VOID' displayed beside the test result, this would be the perfect situation for the lecturer to manually correct it.

Results: Electronics Christmas Assessment

Results

Select one to view the annotated test:

15/15 [Sun, 02 Apr 2017 21:35:01 GMT]	100.0%
15/15 [Sun, 02 Apr 2017 22:23:17 GMT]	100.0%
15/15 [Tue, 04 Apr 2017 16:39:41 GMT]	100.0%

Annotated Test Result (available for viewing after clicking)


15/15 100.00% ExamIT - Copyright 2017

C001
STUDENT NUMBER

ANSWERBOOK

Title: Electronics Basics
Time: 20 minutes
Lecturer: John Marks
Module: Advanced Electronics
Questions: 15

- 1 A B C D E
- 2 A B C D E
- 3 A B C D E
- 4 A B C D E
- 5 A B C D E
- 6 A B C D E
- 7 A B C D E
- 8 A B C D E
- 9 A B C D E
- 10 A B C D E
- 11 A B C D E
- 12 A B C D E
- 13 A B C D E
- 14 A B C D E
- 15 A B C D E



Note that where the student marked two answers on the same line, the algorithm did not fail. Instead, it selected the one marked in more as the correct answer.

The algorithm is also successful when the user goes over the circle borders.



Conformance to Specification and Design

For the most part, the end product conforms to the specification and design documents. The main functionalities of the system were achieved and as it stands, the system, although still not perfect, could be used with success.

The lecturers can create questions and add them to the question pool as stated in the specification.

The lecturers can correct exams using the correction algorithm written with the help of a vision library, as stated in the specification.

However, the lecturers were also supposed to be able to display advanced statistics about passability, questions causing the most trouble ie. As it stands, it is only possible to display exam results and basic statistics. Therefore, this part was completed only partially.

The students were supposed to have a method in the web app of viewing their own results, as it stands, I did not have enough time to develop this.

The system also does not support batch scanning of the tests. Although this would be very easy to implement, there was no time left and I had to focus on the most important functionality.

However, the modular way in which the system was designed allows for an easy extension of the functionality in the future.

Most of the elements of the supplementary specification were achieved.

Learning

Technical Achievements

First and most important achievement is the correction algorithm. Before starting this project vision libraries seemed like black magic to me. After learning OpenCV in small steps, bit by bit every second or third day, after a while I felt comfortable using it. Writing the correction algorithm was the biggest programming challenge for me to date.

Currently, I could write an even better algorithm using a different method for correcting that would practically never fail, whatever the user did to the page. Unfortunately I did not have enough time to tinker with it more. Overall, I am still happy with its current state.

I also spent some time looking at different uses for OpenCV while going through the documentation, this has given me many ideas for future projects.

Also, through OpenCV, I learned some of the NumPy library and I know how to use it when future projects will require a scientific package like this.

Personal Achievements

Through this project, I also learned how large sites/web applications are built using Python and its web frameworks. I have confidence in my skills for the future projects and know where to look to find answers to things that I must still learn in this area. However, I feel like there aren't many things which I couldn't do in this field.

I have been to a work interview for a Python developer position and have presented my code for the app to the interviewer, he was pleased with what he saw enough to offer me a job.

Because this project required a document based database solution, I have become proficient at using MongoDB and can use it for my future projects and in the workplace. Also, because I had to access heavily nested data and transform it, put it back together differently and so on, I have become better at manipulating data in general and in Python specifically.

I have also become more proficient at using jQuery and Javascript in general, these are always necessary for any web application project so this will be extremely helpful in the future.

Project Review

To briefly review the project, I will list information in bullet form under the following headings. Some of this information was already mentioned before.

Things that went right:

- An effective correction algorithm
- Category, question, test creation
- PDF generation and speed
- Manipulating answer sheets PDFs into images to use with OpenCV
- Correction speed
- Smart, effective database calls instead of regular ones
- The whole system is not resource needy, could be hosted on an old server
- A simple login system
- Appealing look of the web app

Things that went wrong:


- No way of differentiating students, the lecturer has to manually take down student numbers from the results page (ran out of time to implement)

Things missing:

- The student panel
- Advanced statistics
- No batch upload functionality
- Login system could be developed more, although not planned it could implement different user roles for user accounts

Things that went great:

- The correction algorithm worked better than initially expected, I know how to make it even better if there was more time
- I learned very practical knowledge that will pay off in the workplace



If I was to start again, I would create the web application layout earlier instead of tunnelling on the correction algorithm for most of the first iteration. I would also research the database technologies much earlier and make a decision on using one.

Learning more advanced jQuery and having a look at Ajax would make the app more dynamic and appealing so I would definitely attempt that.

I would definitely try to be more systematic in spending time on the project. I have spent a lot of time on it, but I could have been more regular. Although it was very tough to manage time for it myself, having all the other college responsibilities and deadlines looming at the same time.



Advice for Attempting a Similar Project

My advice for someone following a similar project would be to design a work schedule and stick to it. There is always more work involved in a task than you think. Especially working with more novel technologies like vision software.

If this is your first large project with a web framework, do read as much about it as you can. Often you can get around problems with simple workarounds using the basic knowledge from the quickstart, but the frameworks have very nice specialised tools for those kind of things and they are always better and work better.

Also, don't always go for the newest technology as it might be lacking in documentation, for me, OpenCV was a good example. I chose it over SimpleCV because of the vastly better documentation although OpenCV is tougher to use. Near the end of the project I looked in SimpleCV for the tools that I used in OpenCV. I realised that it would be so much harder to find the same solutions because of the lacking documentation.

Evaluation of Technology Choices

Choosing Python as the development language was a great choice. This is because it allowed me to write the same functionality in much less lines of code. Also, using Python did not slow down the efficiency of my correction algorithm as it was using C++ under the hood for the OpenCV library. Python only provided the bindings. Python was also a good choice as using Python allowed me to utilise the simple but powerful Flask web framework.

Choosing MongoDB was also a good decision, the data in this project fits document databases perfectly and using any other solution would only slow down the development process.

The pyPDF library was the only library that allowed me to generate the tests with the layout I wanted but it was unpolished and each line of the PDF document had to be explicitly created.

The rest of the choices were minor enough and did not cause any problems.

Overall I am very happy with the choices I made, this is mostly thanks to the extensive research that I performed over iteration 1.



Acknowledgements

This project has been an amazing experience. Through it, I have become a better developer and my understanding of software development and everything that is involved in the process has increased.

Specifically, I have found my programming language of choice for the first time since I have began programming. I would like to thank Paul Barry for that. His web development classes were never a bore and I always learned something new there. I also realised that I would like to work with Python after graduating.

I would also like to thank Christophe Meudec, who never gave us an easy time in class but thanks to him I was able to create those dreadful diagrams for the project.

Lastly, I would like to thank Joseph Kehoe for always answering my neverending questions and making me paranoid about skynet. :)

