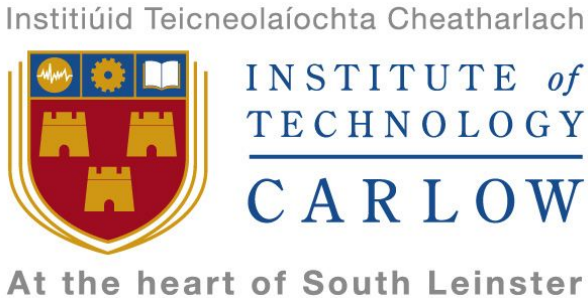
Functional Specification

ExamIT

AUTOMATIC TEST CORRECTION PLATFORM

4th year, Software Development project

Institute Of Technology Carlow



Roger Marciniak

Student Number: C00169733

Supervisor: Paul Barry

31st October, 2016

Table Of Contents

Vision

Problem Description

This project aims to simplify the process of examining students. For the purpose of this project, a web platform will be developed that will automatically correct exams and tests, saving lecturers time and effort of repeating the same task over and over again.

The system needs to allow lecturers to generate exam questions, answers and answer keys and print them off in order to assess their students. The system also needs to allow lecturers to later batch upload completed answer sheets and correct them, taking the load off the lecturer.

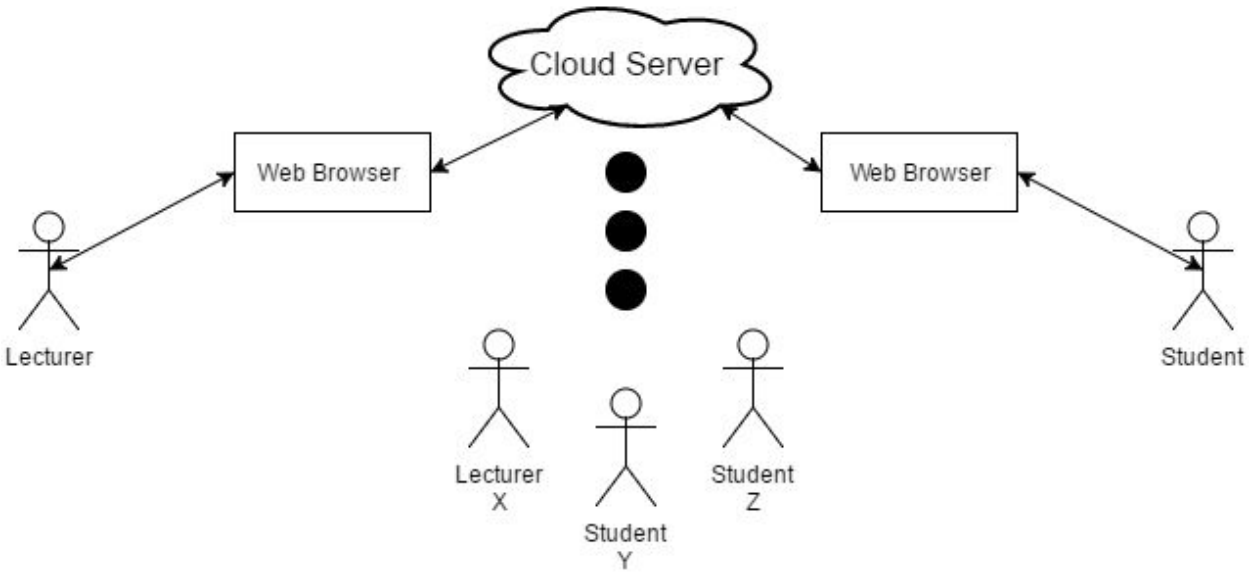
Proposed Solution

A web platform, accessible from any modern browser with internet access. The service will allow the generation and corrections of multiple choice exams by the lecturers, as well as allow each student who took part in the exam to check their result.

The service will also display statistics about the exam to both the lecturer and students. The statistics will take form of best answered questions, worst answered questions, class average etc. These will provide useful data about the student's understanding of the topics, toughest elements of the modules which might need extra lecture time, as well as guide the lecturers and provide them with ways of improving their modules.

The exam results and necessary information for the whole class will be displayed in a table so the lecturer can quickly take note of them.

Context Diagram



Risk & Difficulties

Risks

- Lack of lecturer interest in the ExamIT platform
- The multiple choice approach might not suit every lecturer
- Lecturers might not want to change their method of correcting papers

Difficulties

- Creating an efficient correction algorithm
- Achieving close to a 100% good correction rate (no wrong corrections)
- No prior experience with vision/mark recognition frameworks
- Limited time assigned for the project

Main Functionality

Lecturers

Lecturers can perform the following tasks:

- Generate exams
- Automatically correct exams
- Display exam statistics

Students

Students can perform the following tasks:

- Display their result
- Display exam statistics

Target Audience & Unique Value Proposition

While similar software is already available on the market, it costs a lot of money and is usually made for general purpose use. This project aims to cater to the needs of IT Carlow lecturers specifically, and will incorporate their needs and opinions into the development process.

ExamIT will focus specifically on examining and gathering statistics instead of being an all purpose platform with questionnaires and such.

Everything will be stored in the cloud and will be accessible from the browser. This means that there is no process of installing new software involved, the computers do not need to be set up in any specific way before using the service and that the service can be accessed from anywhere with internet access.

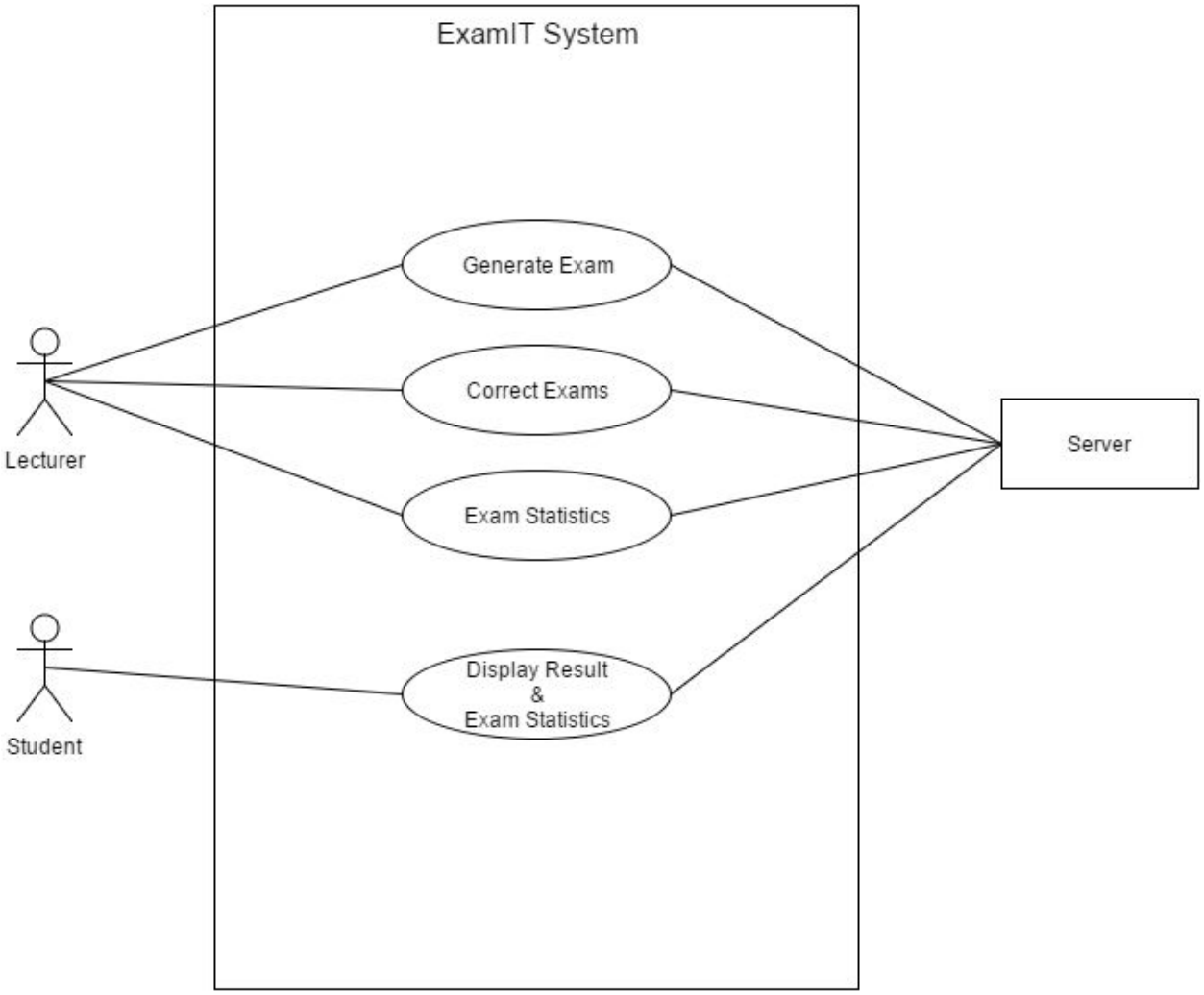
Main Goals

The goal of this project is to take unnecessary work off of lecturers hands and to provide them with useful information that they can use to improve their students learning process and their modules.

So far, two lecturers were consulted about the potential of this project, both were interested in using the platform if it was properly made. One of them even mentioned what he would like implemented in order for him to use it regularly.

The project author believes that there is potential for a service of this type at the institute because, as of now, the lecturers are correcting all their assessments by hand.

Use Case Diagram



Brief Use Cases

Generate Exam

Name: Generate Exam
Authors: Lecturer, Server

Description: This use case begins when the lecturer wishes to generate an exam. The lecturer chooses the ‘Generate Exam’ option and the system retrieves a form for generating an exam. The lecturer is asked to fill out the form by entering exam details (description, instructions, questions, answers & answer keys). When the form is being submitted, the system validates the data entered. If no errors have been found the system displays a confirmation screen, if the lecturer confirms, the exam is saved to the server and can be printed. This use case ends when the lecturer is presented with the confirmation screen.

Correct Exam

Name: Correct Exam
Authors: Lecturer, Server

Description: This use case begins when the lecturer wishes to correct an exam. The lecturer chooses the ‘Correct Exam’ option and the system retrieves a form for correcting an exam. The lecturer chooses an exam from the list and uploads the exam papers (in a batch), using the appropriate button. When the form is being submitted, the system checks that the correct type of files have been uploaded. If no errors have been found the system displays a confirmation screen and corrects the exams in the background. Once finished, the system displays a table with exam results. This use case ends when the lecturer is presented with the results.

Exam Statistics

Name: Exam Statistics
Authors: Lecturer, Server

Description: This use case begins when the lecturer wishes to view exam statistics. The lecturer chooses the ‘Exams Statistics’ option and the system retrieves a page for exam statistics. The lecturer chooses an exam from the list and clicks the appropriate button. When the page is submitted, the system presents the selected exam statistics to the lecturer. This use case ends when the lecturer is presented with the exam statistics.

Display Result

Name: Display Result
Authors: Student, Server

Description: This use case begins when the student wishes to view their exam result. The student chooses the ‘Display Result’ option and the system retrieves a form for displaying the result. The student enters their student number and unique code into the form and clicks the appropriate button. When the form is submitted, the system presents the result & class statistics to the student. This use case ends when the student is presented with the result.

Detailed Use Cases

Generate Exam

Name: Generate Exam
Authors: Lecturer, Server

Description: This use case begins when the lecturer wishes to generate an exam. The lecturer chooses the ‘Generate Exam’ option and the system retrieves a form for generating an exam. The lecturer is asked to fill out the form by entering exam details (description, instructions, questions, answers & answer keys). When the form is being submitted, the system validates the data entered. If no errors have been found the system displays a confirmation screen, if the lecturer confirms, the exam is saved to the server and can be printed. This use case ends when the lecturer is presented with the confirmation screen.

Main Success Scenario:

1. The lecturer chooses the ‘Generate Exam’ option from the menu.
2. The system displays a form on the screen.
3. The lecturer enters exam details (description, instructions, questions, answers & answer keys).
4. The lecturer submits the form.
5. The system receives and validates the form.
6. The system displays a confirmation screen to the lecturer.
7. The lecturer confirms by clicking ‘Confirm’ button.
8. The system saves the exam and displays it.
9. The lecturer prints copies of the displayed exam.

Alternatives:

5a. The form contains errors

I. The form is displayed again, with the error field highlighted.
II. The lecturer corrects the error and submits the form.

Display Result

Name: Display Result
Authors: Student, Server

Description: This use case begins when the student wishes to view their exam result. The student chooses the ‘Display Result’ option and the system retrieves a form for displaying the result. The student enters their student number and unique exam code into the form and clicks the appropriate button. When the form is submitted, the system presents the result & class statistics to the student. This use case ends when the student is presented with the result.

Main Success Scenario:

1. The student chooses the ‘Display Result’ option from the menu.
2. The system displays a form on the screen.
3. The student enters their student number and unique exam code.
4. The student submits the form.
5. The system receives and validates the form.
6. The system displays the student’s result and class statistics for the exam.

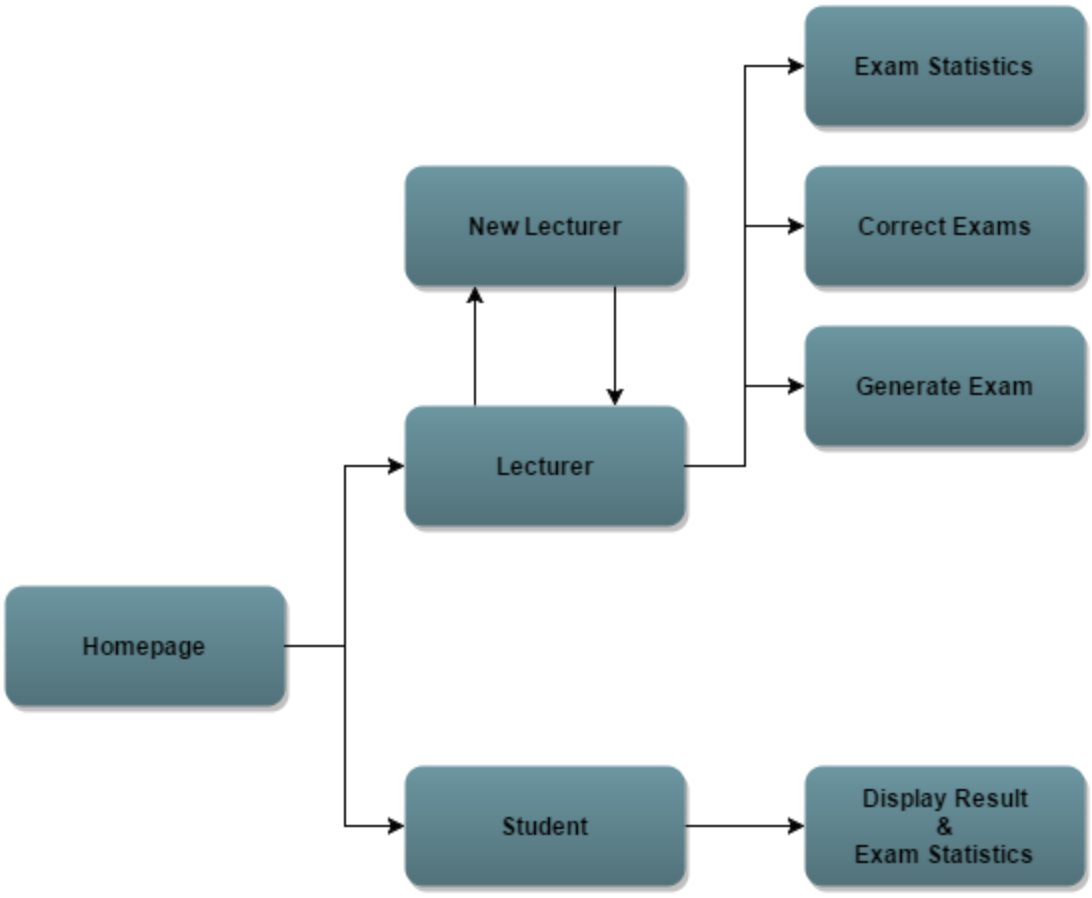
Alternatives:

5a. The details entered do not match database values.

I. The form is displayed again, with the error field highlighted.
II. The student corrects the error and submits the form.

Provisional Site Structure

As of (8 Nov 2016)



Supplementary Specification

Functionality

- All usage requires user authentication.
- Errors must be logged.

Usability

- The system must be fully functional using any modern browser.
- The system should not stand out from popular web services and applications & should not require andy additional computer knowledge.
- The UI should be minimalistic and clean.
- A manual/help page will be available.

Performance

- The system must allow a lag-free experience for a 100 users at the same time.
- Checking a student result should take no longer than 5 seconds.
- Exam generation should take no longer than 10 seconds.
- Batch exam correction should take no longer than 30 seconds.

Reliability

- The algorithm must at least make less than 2% correction errors.
- Mean Time To Repair should be less than 24 hours.
- Users should be able to use the system at whatever hour they wish (24/7).
- Lecturers should have full access to the system from anywhere, not only college grounds.
- Regular database backups must be kept in a separate, protected location.

Supportability

- The system must support 10 simultaneous lecturers and a 100 student users.
- The system must be designed, so it is possible to add new modules or increase the amount of users at a later stage.
- The working system will be reviewed on a weekly basis. Necessary changes/adaptations will be considered.

Security

- The system must use HTTPS protocol to protect personal user data.
- The system must keep user data private and the author has no right to distribute it to third parties.
- Other students should not be able to see their peers exam results.