COM3610: Dissertation
Survey And Analysis Report

Managing The Extenuating Circumstances Process

The University Of Sheffield

This report is submitted in partial fulfilment of the requirement for the degree of Computer Science by **Rushil Shah**.

Date: 03/12/2018

Supervisor: Dr. Siobhán North

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

———————

Rushil Shah

Abstract

Background

Students who suffer from personal problems usually fill out the Extenuating Circumstances Form (ECF) which are then processed by various departments, some with higher clearance than others. This project is based on creating a system which can achieve confidentiality, ease of access and increase efficiency by reducing paperwork. Data has to remain confidential not only with different levels of hierarchy but also from outsiders trying to gain access.

Project Aims

The system should achieve confidentiality, allow the stakeholders to toggle around quickly and efficiently allowing feedback and communication at all times and should reduce the paperwork pile.

Achievements to date

Comparison between potential systems; Web based system or a mobile phone application. Information processing with a lot of data would not be readily accessible on an application and so a web based system is a potential project.

Acknowledgements

I would like to thank my supervisor Dr. Siobhán North for guiding and supporting me throughout the project, without the guidance it would not have been possible to complete this project.

I would also like to thank my family and friends for encouraging me and believing in me to complete this project.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background

Students who suffer from personal problems usually fill out the Extenuating Circumstances Form (ECF) which then allows the examination board (the scrutiny committee) to consider their problems. The process usually begins with the student submitting the form and being reviewed. In situations where there is not enough evidence, the committee asks the student to provide more evidence and then processes the information accordingly, for example the committee may ask the student to present a death certificate. The form at this point has detailed information about the personal problem but from here on, only the general problem (without including details of the specific problem the student is facing) is taken ahead to the other departments and also added to the Students Record to allow leniency in terms of examinations and other assignments. Information regarding the ECFs can be found on the University Of Sheffield's page.[4]

1.2 Problem

Students sometimes fill out the form with less detail and so they need to be re-filled or the students have to provide more feedback/evidence. This leads to multiple documents being created and a lot of paperwork explaining the scenario. The forms are then accessed and duplicated by other departments and also added to the system for the Student Report. Multiple departments with different hieratic allowances creates a lot of confusion which then leads to improper book keeping and a reduction in information confidentiality. These forms may also be up for appeal and with multiple and mixed up forms the whole process tends to be biased.

1.3 Aims And Objectives

The system has to reduce paperwork, be secure enough to allow different levels of confidentiality and access and also be an easy but efficient to use system. A system which can be accessed by students, the examination board and can also be integrated into the student report. The forms contain important information in terms of text and so using a device such as a smart phone would not be efficient and could being up accessibility issues. On the other hand a web based system with the correct levels of security and encryption can be access by anyone as long as they have a device connected to the Internet. All University buildings have access to computers making browsing easy. A big screen would allow accessibility and information can be processed swiftly.
Django[5], a high-level Web Framework based on Python allows the design of the system we are looking for. Its built-in features allows the coder to avoid and prevent

security threats such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking[5]. It also data to be stored and encrypted in a database,

1.4 Report Summary

The report consists of five different chapters including this one. Chapter Two - Literature Review which expands on the use of web systems for solving the problem as well as other possible techniques that could be used but were not because of limiting factors. This chapter contains a broad variety of information and research. Chapter Three - Requirements And Analysis, contains specific information building on the previous chapter and contains evaluation of the project. Chapter Four - Progress will talk about the the testing and results that have taken place with all achievements to date. Lastly Chapter Five - Conclusion And Project Plan will summarise and end the report on the projects plan.

Chapter 2. Literature Review

2.1   Introduction

Digitalising the Extenuating Circumstances Forms would have different platforms to choose from. Technology has advanced to different levels. Web based Systems, Smart Phone Applications, Virtual Reality, Augmented Reality, Artificial Intelligence[6] and recently introduced Blockchain Technology. However, for a system allowing users to fill forms it would be difficult to use Virtual Reality and Augmented Reality as these are used for altering the perception of the world[7]. Same goes for Artificial Intelligence as this would be costly to implement and not very useful when it comes to Extenuating Circumstances. In this section we will look at Web Based Systems, Mobile Phone Applications and the Blockchain Technology. Further ahead we will eliminate Mobile Applications and Blockchain and focus more on the Web Based Systems.

2.2   Web-Based, Smart Application Or Blockchain?

Blockchain is built on transparency. It will allow the storage of forms and data to be secure and encrypted with no central database holding everything. However, Blockchain is more useful when a digital property is involved as it works efficiently with trading and not storage for single entries. It still is a new technology and there is not enough detail on how exactly it can be helpful[8].

Smart Phone Applications on the other hand are used almost daily. There are officially more Mobile Phones than people in the world[9]. Students all use smart phones and it would be easy for them to use a simple application to fill in all the questions and feedbacks for the form. However, the same brings an issue for data processing and readability for the examination board and student report. The scrutiny committee would want the data to be readily available, not having to download a new application where they can barely read the large amounts of data and attachments uploaded by the student. Using mobile applications would allow instant communication as the system can use notifications for constant feedback and review processes.

Web Based Systems have been in the market for quite a while now and they still are the most used compared to any other digital technology[10]. Being accessible from any internet connected device this makes it usable from Mobile Phones, Desktops/Personal Computers and other devices. Web system would also allow the scrutiny committee to access the system and view it on a big screen.

2.3   Web-Based System
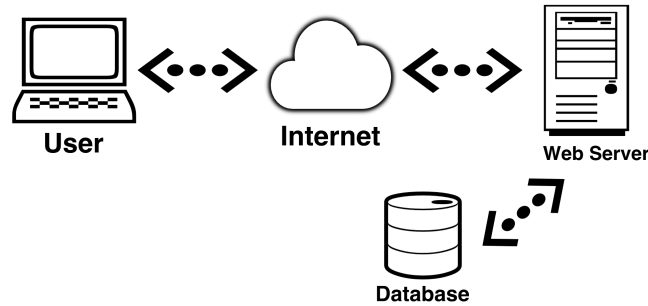
2.3.1   Basic Web

Figure 2.1: Basic Web Server.

Figure 2.1 shows the basic version of how a web server works. The user makes a request via their Internet Service Provider(ISP) onto the Internet and that then connects to the web server hosting the site built and the database connected to it. Once the relevant information has been retrieved the server forwards it back to the internet and it is received by the user. Having is system like this would allow the user to access all data from any device, screen size, and hardware that is able to connect to the Internet.

2.3.2   The System

Web-Based systems are spread into various different languages with both positives and negatives. Java, .NET, PHP, ASP, Python, Ruby, ColdFusion are just a few of the most used language examples for web systems[11].
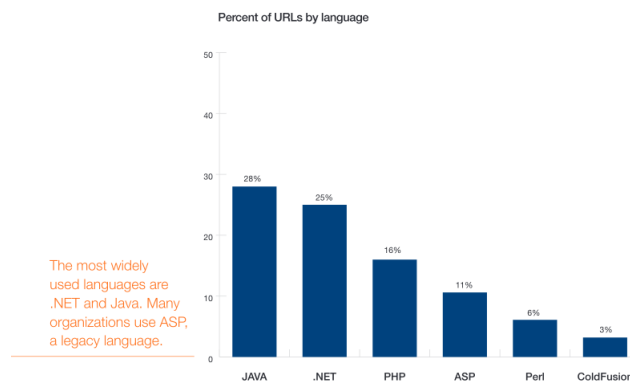

Figure 2.2: Most widely used languages for Web Systems - WhiteHatSec[1]

With all these languages, the highest priority would be to find a language which has Cyber Security as one of its strong points. .NET, Java, ASP, PHP all come with a high number of vulnerabilities as they are highly used[11].

Days open of top 5 vulnerability classes

ASP

ColdFusion

PHP

.NET

Perl

Java

Top five vulnerabilities:
- Cross Site Scripting
- Information Leakage
- Content Spoofing
- HTTP Response Splitting
- Predictable Resource Location

The following vulnerabilities are unremarkable on graph:
- Predictable Resource Location
- SQL Injection
- Cross Site Request Forgery
- Insufficient Transport Layer Protection
- Abuse of Functionality
- Brute Force
- URL Redirector Abuse
- Insufficient Authorization
- Fingerprinting
- Session Fixation
- Directory Indexing

Key findings:
- ASP vulnerabilities remain open the longest at 139 days.
- Cross-Site Scripting takes the longest to close in Perl taking 184 mean days.
- ColdFusion has the largest average days open for SQL Injection vulnerabilities at 107 mean days.
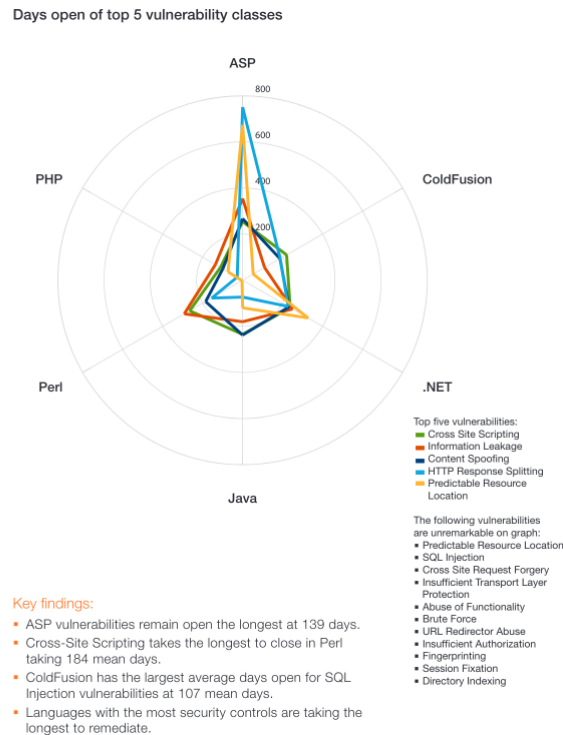- Languages with the most security controls are taking the longest to remediate.

Figure 2.3: Security Vulnerabilites in ASP, PHP, Perl, Java, .NET and ColdFusion - WhiteHatSec[1]

This would then leave us with Python and Ruby which are both easy to code with and used widely. Using a web framework of the languages, Django(Python) or Ruby On Rails(Ruby) would be a smart idea because frameworks allow increased security as they are built for this and increase the efficiency and are cost effective[2].

2.4 Django

Django is a Python based web framework which allows clean and pragmatic design as well as rapid development. The framework is designed to enable the coders to code without worrying about the security of the code as it guides the coders through securing the vulnerabilities such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Django also has a user authentication system which can be customised to allow a secure way for data to be transferred[5].

2.4.1 Django vs Laravel

Django is a Python based framework as stated above but Laravel[12] on the other hand is a PHP based web framework. Django follows the Model View Template similar to Ruby on Rails but Laravel follows Object Oriented Programming as

well as the Model View Template. In terms of website security, Django takes it extremely seriously and helps developers avoid the common mistakes which lead to the website having vulnerabilities while Laravel also has a guide to avoid making such mistakes but it does not beat Django.

Django is naturally very fast as it uses Python which is known for its speed and processing. Djano beats Laravels speed in all; Plaintext, JSON and SQLite Fetch[2].

Django: Plaintext test - 3127 requests/seconds, JSON test - 3060 requests/ seconds, Random SQLite Fetch test - 2137 requests/ seconds.

Laravel: Plaintext test - 212 requests/seconds, JSON test - 167 requests/ seconds, Random SQLite Fetch test - 140 requests/ seconds.

Figure 2.4: Django vs Laravel Speeds 2016 - CabotSolutions[2]

2.4.2 Django vs Ruby on Rails

Both are popular web frameworks, Django more than Ruby on Rails for professional developers. They are open-source which allows all code to be customised in any way. Ruby on Rails allows the use of gems which are designed by other users, these come in handy when programming personal or individual projects however with professional projects licensing is a good idea to allow the code to be usable in the future[3].

Based on JetBrains research[13][14] we know that Python is a really popular language compared to Ruby and it is highly used for Data Analysis and Web Development among various other things while developers use a mixture Ruby Versions which makes future updates to the code harder.

The table 2.1 shows the differences between Django and Ruby on Rails

	Django	Ruby on Rails
Pros	- Python is versatile - Fast - Caching System - Data Analysis - Great Security and Autentication	- Flexible - Large Community - Gems Available - Easy Migration
Cons	- Hard to debug - Monolithic architecture	- Bloated - No Data Analysis - Very explicit and inelegant to read
Started By	- Django-Started in 2003 by Adrian Holovaty and Simon Willison	- Ruby on Rails-Started in 2005 by David Heinemeier Hansson

Table 2.1: Django vs Ruby on Rails Pros and Cons [3]

2.4.3 Summary

Django, compared to various other web frameworks, is a good environment to use for the Extenuating Circumstance Forms because it provides security we are looking for, avoids vulnerabilities and is really fast when accessing and using with SQLite. Even with small or large numbers of data, it will be able to process it really quickly. Upgrading the system at a later stage will also be simple as Python is a highly developed language.

Chapter 3. Requirement & Analysis

3.1 Introduction

This chapter will take Django as the main language used for developing the web based system and present ways in which the project can be brought together. From ways to submit the form to ways in which the scrutiny committee can access the same, this can be with extra feedback required or accepted the way it is.

3.2 Process

To begin with, the extenuating circumstances page[4] will have a link to take the student to the new website portal which will contain a login page allowing the student to login with his or her university email. Once the student has logged in, an option to start a new extenuating circumstance form will be displayed and clicking on that the student will be presented with a form to fill online. Once the online form is filled with all the details it will be stored on their profile and displayed on the "Dashboard" of the portal. The same form can then be accessed over time with new feedback, approval and other statuses. Any updates by the scrutiny committee will be notified to the student via email.

The portal will also allow the scrutiny committee to login and control the movement of the forms as well as provide feedback and comments to the student regarding their circumstance. They will also be able to request more proof and evidence if need be. The forms can then be processed further and exported as documents such as PDF's and TXT's allowing the relevant stakeholder to print or share. Figure 3.1 describes the authentication and users process.
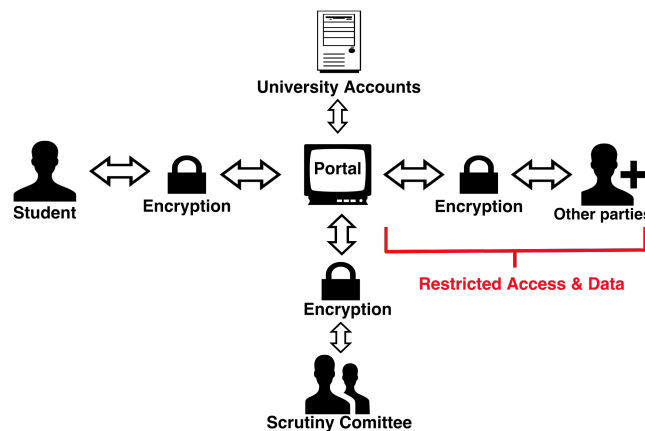

Figure 3.1: Portal and users.

The portal will be entirely encrypted and data will be stored in a secure database as presented earlier in Figure 2.1. This allows total security of the information stored and only users with access will be allowed to get that information. Some users will be restricted such as the other parties who get the students condition without knowing exactly what it is they are having trouble with. This allows confidentiality while allowing access at the same time.

3.3 Testing

Testing is an important aspect of knowing the functionality is as required. In this system we can carry out testing in various different elements to be sure that the system is how we expect it.

3.3.1 SQL Queries

Since data will be stored into the database, we can use RAW SQL Queries to test if a certain field in the software is giving access to the database without checking the entry first. For example, if a student fills a RAW SQL Query instead of his details then the resulting should be an error and should not display the data being retrieved from the database.

3.3.2 Cross Site Scripting

Cross Site Scripting (XSS) is a malicious method of injecting scripts into the system which then target other users. These scripts can be used with JavaScript which will be used when designing the portal. Testing this will make sure that the portal is secure from outside attacks.

3.3.3 Automated Testing

Django allows writing of test code (unittest)[15] which then runs as a normal user(in our case) and checks if the response is exactly how we need it to be. For example if a student clicks on "Submit Form" then it should store the form in the database. The unittest will be able to test if the data is actually being stored in the database.

3.3.4 User Testing

In order to have covered all parts of the system it would be a good idea to have a user from the scrutiny committee as well as a student to test the system out, if they feel there is anything left out or any other relevant issues that need to be solved. This would be the last bit of testing once the system has passed the development stage.

Chapter 4. Progress, Conclusion & Project Plan

4.1 Introduction

This chapter briefly talks about the progress on the project, the work that has been completed till this date and the next steps from this point forward. It will also present a Gantt Chart to display the progress.

4.2 Achievements

Till date, i have been able to research between different types of systems and advancements, different web based applications and Django itself as described in Chapter 2. Being able to select the best of these systems was one of the most important aims and Django seems to be fulfilling it.

4.3 Plan Of Work

Being able to have all research in hand i now plan to start working on the Django web portal prototype. This will require learing of Django and building a website from scratch. Running tests is an important task and will be completed along with the building of the prototype. This way any issues with the testing can be handled right away as the coding is still in progress. Figure 4.1 shows the a Gantt Chart with the progress based on the development of this project.
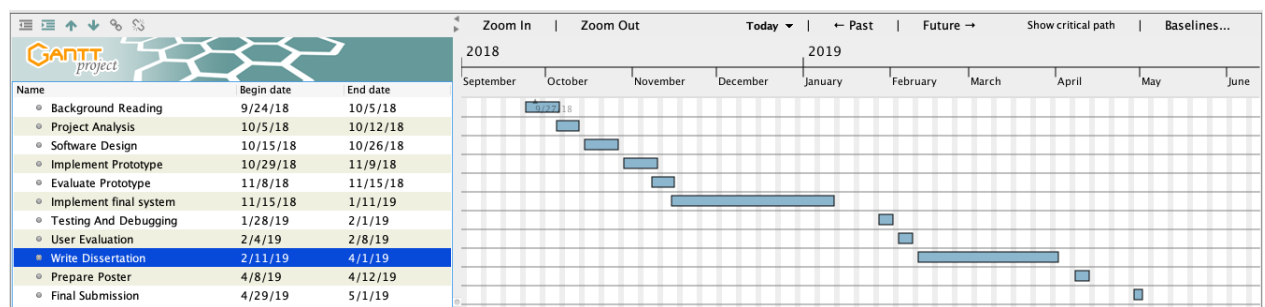

Figure 4.1: Gantt Chart

Bibliography

[1] W. Security, "2014 website security statistics report." `http://info.whitehatsec.com/rs/whitehatsecurity/images/statsreport2014-20140410.pdf`, 2014. Accessed: 06.11.2018.

[2] Cabotsolutions, "Django vs laravel: A comparison of web application frameworks." `https://www.cabotsolutions.com/django-vs-laravel-a-comparison-of-web-application-frameworks`, 2018. Accessed: 01.10.2018.

[3] J. Hargan, "Django vs rails: Picking the right web framework." `https://www.tivix.com/blog/django-vs-rails-picking-right-web-framework`, 2018. Accessed: 11.11.2018.

[4] U. Sheffield, "Extenuating circumstances form: Explanatory notes - forms - ssid - the university of sheffield." `https://www.sheffield.ac.uk/ssid/forms/circsnotes`, 2018. Accessed: 01.10.2018.

[5] Django, "The web framework for perfectionists." `https://www.djangoproject.com/`, 2018. Accessed: 01.11.2018.

[6] F. J. DeMers, "7 technology trends that will dominate 2018." `https://www.forbes.com/sites/jaysondemers/2017/12/30/7-technology-trends-that-will-dominate-2018/#7e4e30157d76`, 2017. Accessed: 01.11.2018.

[7] J. Ronzio, "7 technology trends that will dominate 2018." `https://cramer.com/story/the-difference-between-ar-and-vr/`, 2018. Accessed: 03.11.2018.

[8] A. Castro, "Blockchain' is meaningless." `https://www.theverge.com/2018/3/7/17091766/blockchain-bitcoin-ethereum-cryptocurrency-meaning`, 2018. Accessed: 06.11.2018.

[9] Z. D. Boren, "There are officially more mobile devices than people in the world." `https://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html`, 2014. Accessed: 06.11.2018.

[10] A. Larson, "How many websites are there?." `https://www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html`, 2018. Accessed: 08.11.2018.

[11] Upguard, "Which web programming language is the most secure?." `https://www.upguard.com/blog/which-web-programming-language-is-the-most-secure`, 2018. Accessed: 01.10.2018.

[12] Laravel, "Laravel - the php framework for web artisans." `https://laravel.com/`, 2018. Accessed: 11.11.2018.

[13] JetBrains, "Jetbrains - python." `https://www.jetbrains.com/research/devecosystem-2018/python/`, 2018. Accessed: 11.11.2018.

[14] JetBrains, "Jetbrains - ruby." `https://www.jetbrains.com/research/devecosystem-2018/ruby/`, 2018. Accessed: 11.11.2018.

[15] Django, "Documentation - writing and running tests." `https://docs.djangoproject.com/en/2.1/topics/testing/overview/`, 2018. Accessed: 20.11.2018.