

---

# 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

<b>Title Page</b>	<b>i</b>
<b>Declaration</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem . . . . .	1
1.3 Aims And Objectives . . . . .	1
1.4 Report Summary . . . . .	2
<b>2 Literature Review</b>	<b>3</b>
2.1 Introduction . . . . .	3
2.2 Web-Based, Smart Application Or Blockchain? . . . . .	3
2.3 Web-Based System . . . . .	4
2.3.1 Basic Web . . . . .	4
2.3.2 The System . . . . .	4
2.4 Django . . . . .	6
2.4.1 Django vs Laravel . . . . .	6
2.4.2 Django vs Ruby on Rails . . . . .	6
2.4.3 Summary . . . . .	7
<b>3 Requirement &amp; Analysis</b>	<b>8</b>
3.1 Introduction . . . . .	8
3.2 Process . . . . .	8
3.3 Interface . . . . .	9
3.3.1 Student Panel . . . . .	9
3.3.2 Scrutiny Committee Panel . . . . .	9
3.4 Testing . . . . .	11

3.4.1	SQL Queries . . . . .	11
3.4.2	Cross Site Scripting . . . . .	11
3.4.3	Automated Testing . . . . .	11
3.4.4	User Testing . . . . .	11
3.5	Summary . . . . .	11
<b>4</b>	<b>Progress, Conclusion &amp; Project Plan</b>	<b>12</b>
4.1	Introduction . . . . .	12
4.2	Achievements . . . . .	12
4.3	Plan Of Work . . . . .	12
	<b>Bibliography</b>	<b>14</b>

# List of Figures

2.1	Basic Web Server. . . . .	4
2.2	Most widely used languages for Web Systems - WhiteHatSec[1] . . .	5
2.3	Security Vulnerabilites in ASP, PHP, Perl, Java, .NET and ColdFu- sion - WhiteHatSec[1] . . . . .	5
2.4	Django vs Laravel Speeds 2016 - CabotSolutions[2] . . . . .	6
3.1	Portal and users. . . . .	9
3.2	Form with a feedback section . . . . .	10
4.1	Gantt Chart . . . . .	12

# List of Tables

2.1 Django vs Ruby on Rails Pros and Cons [3] . . . . . 7



# 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 ECF's 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 scrutiny committee 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 bring 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 four 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, Conclusion And Project Plan; will talk about the the testing and results that have taken place with all achievements to date with a future plan.

# Chapter 2. Literature Review

## 2.1 Introduction

Digitalising the Extenuating Circumstances Forms would have different platforms to choose from as technology has upgraded significantly over the past decade. 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]. Being a new advancement, it might not be stable enough in the long run and education on a Blockchain based system would be required for further maintenance and bug fixing.

Smart Phone Applications on the other hand are used almost daily. There are officially more Mobile Phones than people in the world[9]. All students 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 scrutiny committee 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 but would only be advantageous for the students as they will be able to fill the forms on their finger tips, on the other end the scrutiny committee and other stakeholders will not like the idea of having to do everything from a small five inch phone.

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 on a larger screen and allow them to make any significant changes and requests with enough detail and also move data from one stakeholder to another easily such as exporting as a PDF document. Students already use MUSE[11] and so our system would only need to be linked to the University of Sheffield logins allowing all users to login easily.

## 2.3 Web-Based System

### 2.3.1 Basic Web

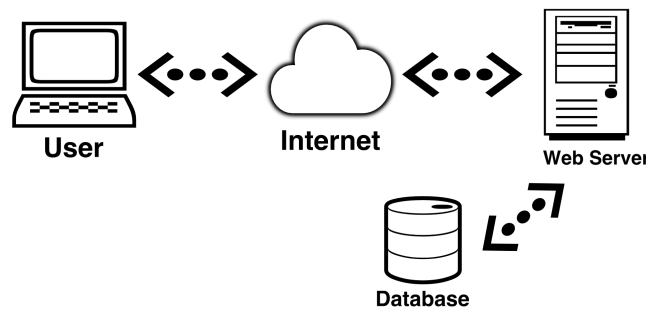


Figure 2.1: Basic Web Server.

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 a 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 making it very flexible and easy to use.

### 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[12]. Figure 2.2 and Figure 2.3 shows us programming languages that are widely used for web systems and a comparison of their vulnerabilities. Selecting a secure language is an important aspect for security and confidentiality of this project.

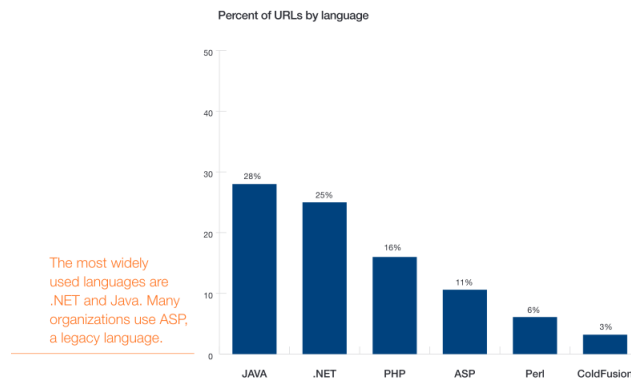


Figure 2.2: Most widely used languages for Web Systems - WhiteHatSec[1]

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[12].

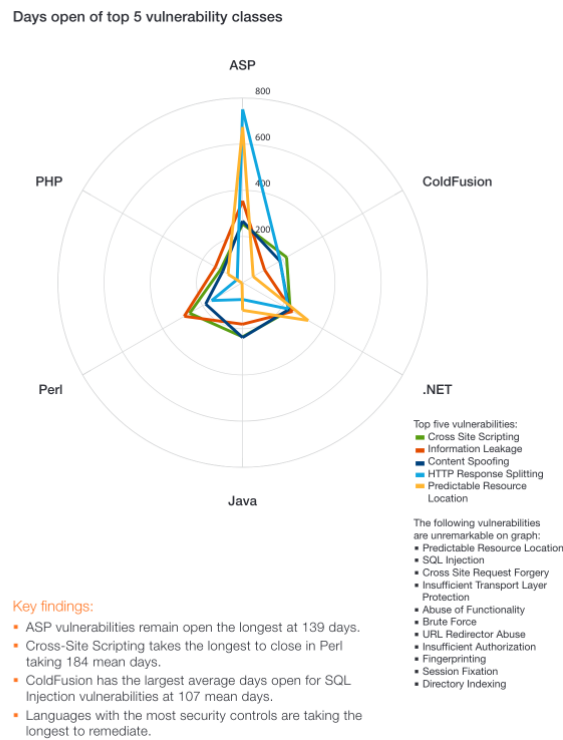


Figure 2.3: Security Vulnerabilites in ASP, PHP, Perl, Java, .NET and ColdFusion - WhiteHatSec[1]

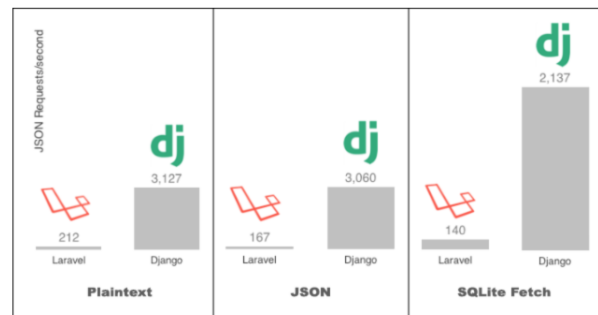
This would then leave us with Python and Ruby which are both easy to code with and used widely. Using a web framework of the languages, Django(Python) or Ruby On Rails(Ruby) would be a smart idea because frameworks allow increased security as they are built for this and increase the efficiency and are 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[13] 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 its contents are not as informative and detailed as Django. Django is naturally very fast as it uses Python which is known for its speed and processing. Django beats Laravels speed in all; Plaintext, JSON and SQLite Fetch[2] as shown in 2.4.



**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[14][15] 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. Table 2.1 shows the differences between Django and Ruby on Rails

	Django	Ruby on Rails
Pros	<ul style="list-style-type: none"> <li>- Python is versatile</li> <li>- Fast</li> <li>- Caching System</li> <li>- Data Analysis</li> <li>- Great Security and Autentication</li> </ul>	<ul style="list-style-type: none"> <li>- Flexible</li> <li>- Large Community</li> <li>- Gems Available</li> <li>- Easy Migration</li> </ul>
Cons	<ul style="list-style-type: none"> <li>- Hard to debug</li> <li>- Monolithic architecture</li> </ul>	<ul style="list-style-type: none"> <li>- Bloated</li> <li>- No Data Analysis</li> <li>- Very explicit and inelegant to read</li> </ul>

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. It will also provide detailed information on the interface of the website as well as provide test options.

## 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.



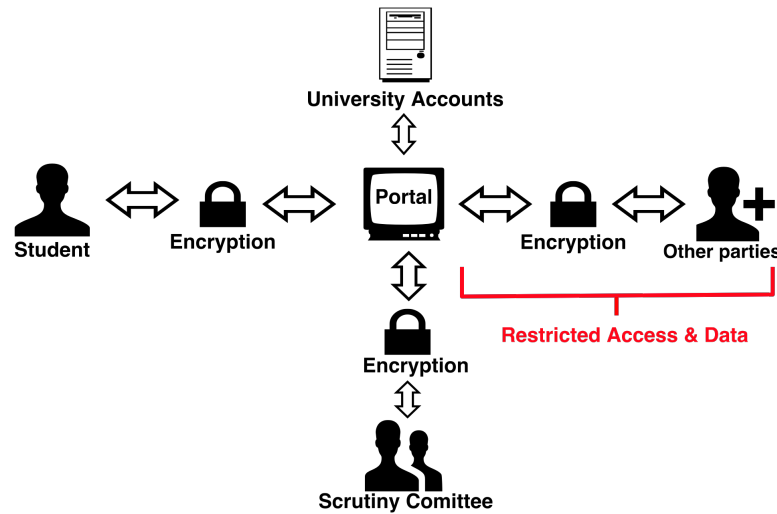


Figure 3.1: Portal and users.

The web system 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 Interface

The interface will consist of two different types of users; The Students and the Scrutiny Committee. There will be other third party users who will get the data from the student records such as the professors and personal tutors.

#### 3.3.1 Student Panel

The student panel will consist of three main components; They should be able to submit a new circumstance form, access any previously submitted form and lastly check the status(described under Scrutiny Committee Panel) for any new feedbacks or requests from the scrutiny committee and upload a response or evidence as required. As stated in the previous section, the students should be able to login with their own university accounts making is easy to access from within the portal.

#### 3.3.2 Scrutiny Committee Panel

This panel will be a little more detailed than the student panel. The panel would notify the scrutiny committee by email when a new form has been received The committee would be able to view the form and request any new feedback or

evidence and change the status of the form. The status could be "Approved" or "Requires changes" or "Pending" which would allow different members of the scrutiny committee to know the status of a form. In this case;

- Approved would mean the student does not need to provide more information or evidence and the details of the form have been passed on to the third parties.
- Requires changes would mean the student has to provide more information such as evidence or detailed writing. The requirements would be accessible by the student under the "Feedback" section of the form.
- Pending would mean the form has not yet been checked by the members of the scrutiny committee and so there is nothing to be done.

They can also export the same form as a PDF and download it for printing purposes or for documentation.

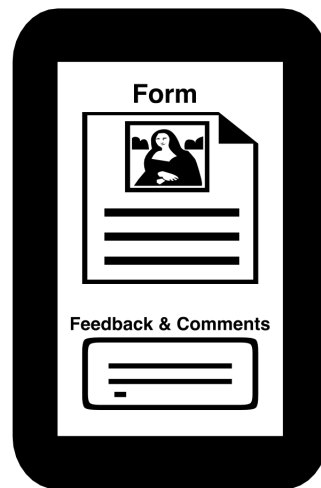


Figure 3.2: Form with a feedback section

Figure 3.2 shows the portal with a form being displayed with a feedback and comments section. This section will allow communication between the student and the scrutiny committee as well as allow uploads of new evidence.

The panel will also have a list of all the students who have submitted a form allowing the scrutiny committee to sort and search through the records to find any specific details they like which would be useful in cases of a review.

It will also enable the exporting of data to third parties, only approved details by the scrutiny committee will be accessible by the third parties that need access and knowledge of an existing extenuating circumstance form.

## 3.4 Testing

Testing is an important aspect of functionality and knowing if the system follows the requirements. In this system we can carry out testing in various different elements from automated to user tests. Below are a few of the tests that we can run in order to make sure the project is reachable.

### 3.4.1 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.

### 3.4.2 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.

### 3.4.3 Automated Testing

Django allows writing of test code (unittest)[16] 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.

### 3.4.4 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.

## 3.5 Summary

This chapter has talked about how the development of a web based system would work, how the stakeholders involved in this project would access their data and process it based on the situation and other parties involved. Being able to make sure everything is following the requirements of the project is extremely important and so different types of testing is really helpful.

# 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

Research between different systems such as Web Based, Smart Phone application, Blockchain and a few others has been completed allowing me to select one from all of them and move it ahead to the development stage in the coming few weeks. Planning for the interface and its functions has also been achieved as seen in chapter 3.3 and having ideas on which possible tests can be done to check the progress of the project in relation to the requirements.

## 4.3 Plan Of Work

From this point forward i plan on learning how to use Django as a programming language and build a simple website for starters which i can then build on to make it a better and secure web system for this project. Following the interface plan is a key and making sure the project does not go off requirements with the help of testing with users as well as unit tests. Testing will be performed side by side with the development as well as at the end to make sure the project does not go off path. Figure 4.1 shows the plan of work from this date onwards.

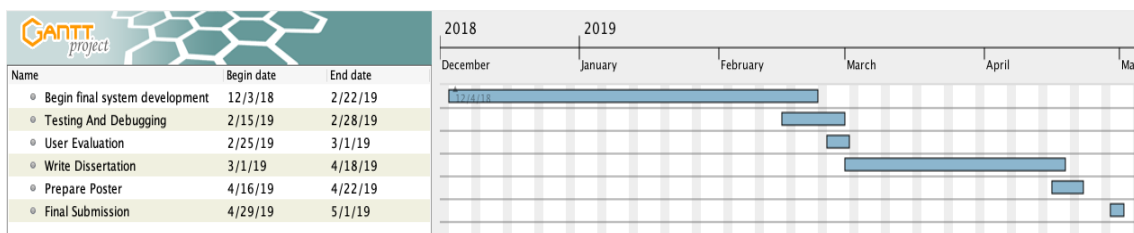


Figure 4.1: Gantt Chart

# 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] U. O. Sheffield, “Making the most of muse.” <https://www.sheffield.ac.uk/bms/undergrad/muse>, 2016. Accessed: 01.12.2018.

- 
- [12] 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.
  - [13] Laravel, “Laravel - the php framework for web artisans.” <https://laravel.com/>, 2018. Accessed: 11.11.2018.
  - [14] JetBrains, “Jetbrains - python.” <https://www.jetbrains.com/research/devecosystem-2018/python/>, 2018. Accessed: 11.11.2018.
  - [15] JetBrains, “Jetbrains - ruby.” <https://www.jetbrains.com/research/devecosystem-2018/ruby/>, 2018. Accessed: 11.11.2018.
  - [16] Django, “Documentation - writing and running tests.” <https://docs.djangoproject.com/en/2.1/topics/testing/overview/>, 2018. Accessed: 20.11.2018.