National College of Ireland BSc in Computing 2016/2017



Timothy Yong x13548867 x13548867@student.ncirl.ie

Integrity Checker

**Technical Report** 



#### **Declaration Cover Sheet for Project Submission**

**SECTION 1** Student to complete

Name:	
Timothy Yong	
Student ID:	
X13548867	
Supervisor:	
Sara Kadry	

#### **SECTION 2 Confirmation of Authorship**

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:_	 	 	
Date:			

#### **Table of Contents**

- 1 Executive Summary
- 2 Introduction
  - 2.1 Background
  - 2.2 Aims
  - 2.3 Technologies

Definitions, Acronyms, and Abbreviations

- 3 System
  - 3.1 Requirements
    - 3.1.1 Functional requirements
    - 3.1.2 Data requirements
    - 3.1.3 User requirements
    - 3.1.4 Environmental requirements
    - 3.1.5 Usability requirements
    - 2.1.6 Security requirements
  - 3.2 Design and Architecture
  - 3.3 Implementation
  - 3.4 Graphical User Interface (GUI) Layout
  - 3.5 Testing
- 4 Conclusions
- 5 Further development or research.
- 6 Appendixes

Appendix 1 Project Proposal

Appendix 2 Project Requirements Spec

Appendix 3 Monthly learning reports

7 References/ Research

#### 1 Executive Summary

"Cyber security, also referred to as information technology security, focuses on protecting computers, networks, programs and data from unintended or unauthorized access, change or destruction." [1] (What is Cyber security?, 2016).

Cyber security is all about protecting information/ data kept on computers. Computers have been used to store sensitive information for nearly as long as they've been around. Whenever there is something of value, there will always be people who are trying to access it.

Cyber terrorists, hackers and their tools are only getting more smart. There is more and more attacks where the complexity and the techniques used are becoming more intuitive and also dangerous.

This just shows the need for increased security surrounding anything technological these days as absolutely everything is hackable. Nowadays it's not just about hacking things but also about not even letting the owners of the data know they've been hacked.

If something technological has been hacked how does one know? This is where the idea for my project comes from.

My project will allow the user to check to see if their file has in fact been changed.

My project is an integrity checker; it will allow users to check if their document/ file has been adjusted.

#### **2** Introduction

#### 2.1 Background

Why? The purpose of my project is to provide a sense of security for the safe keeping of data. Allowing user peace of mind ensuring that they have full ability to verify their documents integrity.

#### 2.2 Aims

What? It is a way for users to check the integrity of their documents. Ensuring that they are where they were left. The project shows users their files metadata. It will also allow users to check the metadata of their document against a previous version so that they can verify its integrity. In other words, that their document has not been hacked.

#### 2.3 Technologies

How? - The project was built using Visual Studio 2015. It is coded in C#. It is connected to a DB using MySQL. The application requires users to register before using the primary feature (Integrity Checker). The application will also have a Multi Factor login system. The Multi Factor authentication will be both a QR scanner and SMS confirmation. The application will encrypt user login details using MD5. The application will also store the users uploaded files metadata. This will be the file name, size, extension, last modified and the file's hash value.



#### 2.4 Real world problem

My Project goals are to solve this real world problem. Data and Information can easily be stolen and replaced without users realizing. What my project aims to do is prevent this.

Users will be able to verify their documents integrity with my project. Using the documents md5 hash, which is unique to every file and cannot be changed.

#### Definitions, Acronyms, and Abbreviations

AD	Another Definition
Condf	Confidentiality
Integ	Integrity
Sy. Arch	System Architecture
Auth	Authentication
DB	Database
RWD	Read, Write, Delete
IP Address	Internet Protocol Address - this is an address unique to every
device.	
User	This is the client, the user of the application
System	This is the application, what the user uses.
Doc	Document
MFA	Multi Factor Authentication
APP	Application
GUI	Graphical User Interface
URL	Uniform Resource Locator

#### 3 System

#### 3.1 Requirements

#### 3.1.1 Functional requirements

The program will have lots of functional requirements. The project has respectable up to the industry standard performance, which will have a short response time and a large capacity. It will have good availability and also reliable.

Only users will be able to login.

Only non-registered email ids will be able to register.

Only users will be able to use the application.

The system will accept all the correct information and record the logins.

The system will deny access to users who have entered their information incorrectly.

Users will be able to check their files metadata.

Only registered users will be able to check their files metadata

The system will be functional as long as these requirements are fulfilled.

#### 3.1.2 Data requirements

The program will have certain data requirements. The user will be required to have their own login and or registration data. This will consist of their first name, last name, email id, password, mobile no.

#### 3.1.3 User requirements

The user interface is menu driven, meaning that it user friendly. It dialog boxes, radio buttons, dropdown list boxes and spin buttons for user inputs.

When a user accesses any screen it will appear on the monitor within 3 seconds. Making the system quick and user friendly.

On loss of power on the user's operating system, the database will be backed up online to ensure full maintainability.

After 3 unsuccessful attempts to logon, a user will be locked out of the system.

#### 3.1.4 Environmental requirements

Computer system reliability is dependent upon a stable environment. The design of the environmental control system for your data center must ensure that each system can operate reliably while remaining within the range of its operating specifications. The Computer system must have java 8 installed.

Currently the project is setup to run only on local machines. There is no need for Internet access.

The user must also have a monitor and a mouse and a keyboard to use the application.

#### 3.1.5 Usability requirements

The intended users will not be expected to have a mass knowledge of computing and any coding knowledge.

Users will just need to be able to move around a GUI and know how to upload a file.

They will also need to know their username and password.

#### 2.1.6 Security requirements

Confidentiality - Protecting the information from disclosure to unauthorized

parties. (The System will restrict access to unregistered/-unauthorized users.)

Access Control- Users will not be able to access the system unless they are registered and have permission or are authorized to access the data.

*Privacy* - The User's information will not be shared with third parties. When a User saves data within the DB it is not made public that the user has data stored. It is private.

Integrity - Integrity of information refers to protecting information from being modified by unauthorized parties. The information saved within the database will not be available to unregistered and unauthorized users.

*Origin Authenticity* - This is insuring that the users who are in fact using the application are in fact the intended user. (E.g. someone using someone else's account.)

Availability - Availability ensures that authorized parties are able to access the information when needed. I.e. it is available to them.

Audit logs shall be verbose enough to support forensics. I.e. All account

modification events shall be logged. The event log shall contain date, time, user, action, object, and prior value, new value.

Audit logs have integrity protection. Does not allow third parties apps to operate within the System

Session Timeout - To prevent a User leaving their application unattended for large amounts of time granting other unauthorized users access to the application, the System will timeout after a certain amount of time wherein the application is not used.

Secure Coding - I will also ensure all the code follows a Secure Coding practice. This ensures my software will be resistant to attack by malicious or mischievous Users/ Programs. See more below.

#### 3.2 Design and Architecture

Algorithm:

- Step 1 Start
- Step 2 Click Register
- Step 3 Enter Details
- Step 4 Click Login
- Step 5 Enter Login details.
- Step 6 Input MFA code
- Step 7 Move to File verification page

Step 8 - Input file

- **Step 9 Receive file Metadata**
- Step 10 Repeat steps 4 8
- Step 10 Compare file data to Metadata

Logging in:



### 3.3 Graphical User Interface (GUI) Layout

#### Register Page

	タ ナ C II: Register - My ASP.NET Appl×	- • × h * 0
Application name	ome About Contact Upload file Register Log in	Â
Register. Create a new account.		
Email Password Confirm password	Register	
© 2017 - My ASP.NET Applic	ation	

#### Login page

	- •
E http://localhost51831/Account/Login	ń ×
Application name Home About Contact Upload file	
Log in.	
Use a local account to log in.	Use another service to log in.
Email	Google
Password	
Remember me?	
Log in	
Register as a new user	
© 2017 - My ASP.NET Application	

#### MFA Page

#### Manage.

Change your account settings



© 2017 - My ASP.NET Application

Carrier 🗢	12:00 PM		-	
≡ Aut	henticator		1	
524	998			
user@example	e.com		6	
		-		

#### Phone Number.

Add a phone number

Phone Number	
	Submit

© 2017 - My ASP.NET Application

#### EnableGoogleAuthenticator

#### 1\. Add MySuperApplication to Google Authenticator

Open Google Authenticator and add MySuperApplication by scanning the QR Code to the right.

#### 2\. Enter the 6 digit code that Google Authenticator generates

Verify that MySuperApplication is added correctly in Google Authenticator by entering the 6 digit code which Google Authenticator generates for MySuperApplication below, and then click Enable.

Code	
	Enable



© 2017 - My ASP.NET Application

#### File Upload page/ Integrity Checker Page

Application name Home About Contact Upload file
Upload file
File: Browse Submit
2017 - My ASP.NET Application

Uploaded file to Verification page



#### Uploaded file

Name:	"\vmware-host\Shared Folders\Desktop\Timmy college project\Payslip.docx"
Size:	30,281 bytes
Extension:	docx
Last Modified Time:	5/8/2017 6:16:43 PM UTC
MD5 hash:	C5-7E-3C-62-9E-26-9B-67-64-85-76-EB-AB-98-F7-96

© 2017 - My ASP.NET Application

## Input validation Register.

#### Create a new account.

- · The Email field is not a valid e-mail address.
- . The password and confirmation password do not match.

Email	test123
Password	
Confirm password	•••
	Register

© 2017 - My ASP.NET Application

#### 3.4 Testing

I will evaluate the system for its integrity, confidentiality and availability but primarily I am going to test for Integrity using SIT.

System integration testing (SIT) is a software testing process that in which testers can verify that all related systems maintain data integrity and can operate in coordination with other systems in the same environment. The testing process ensures that all subcomponents are integrated successfully to provide expected results.

I will also run the following tests.

- I will paste the internal URL straight onto the URL/ browsers address bar without logging in. The upload document page should not open. Application should create an error message.
- Trying false inputs into the input fields like; login username, password, input text boxes etc.
- Testing if SSL is actually being used in project for security. When used properly, the message should get displayed when user switch from non-secure *http://* pages to secure *https://* pages and the same for the other way round.
- I will also test the application on several different computers with different OS versions installed. [2]

In my code, I addressed a type of security breach in a web application that is called Cross Site Request Forgery or CSRF hack. CSRF is the lesser-known cousin of XSS. Cross Site Request forgery is a type of a hack where the hacker exploits the trust of a website on the user. In other words, the site trusts the user (because they have authenticated themselves) and accepts data that turns out to be malicious.

MVC's anti-forgery support writes a unique value to an HTTP-only cookie and then the same value is written to the form. When the page is submitted, an error is raised if the cookie value doesn't match the form value.

It's important to note that the feature prevents cross-site request forgeries. That is, a form from another site that posts to your site in an attempt to submit hidden content using an authenticated user's credentials. The attack involves tricking the logged in user into submitting a form, or by simply programmatically triggering a form when the page loads. The feature doesn't prevent any other type of data forgery or tampering based attacks.

To use it, decorate the action method or controller with the ValidateAntiForgeryToken attribute and place a call to @Html.AntiForgeryToken() in the forms posting to the method.

The basic purpose of ValidateAntiForgeryToken attribute is to prevent cross-site request forgery attacks. Cross-Site request forgery is an attack in which attacker sends harmful script element or malicious command or code from browser of trusted user.

```
Google Authenticator Enabled:
@if (Model.TwoFactor)
{
    using (Html.BeginForm("DisableTwoFactorAuthentication", "Manage", FormMethod.Pc
    {
        @Html.AntiForgeryToken()
        <text>Enabled
            <input type="submit" value="Disable" class="btn btn-link" />
        </text>
    }
}
else
{
```

"Cross-Site Request Forgery (CSRF) Is a sort of an attack which makes and or forces a client/user to execute unwanted actions on a web application in which they themselves are currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, due to the attacker having no way to see the response to the forged request. With a little help of social engineering (such as sending a link via email or chat), an attacker may trick the users of a web application into executing actions of the attacker's choosing. If the victim is a normal user, a successful CSRF attack can force the user to perform state changing requests like transferring funds, changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application. "[3] APPLICATION SELF-PROTECTION IS FINALLY HERE

It is simple to use, you need to decorate method with ValidateAntiForgeryToken attribute as below:

#### **4** Conclusions

The advantages of the product are that users will have full assurance that their files are in fact safe and they're own and unmodified. Their files will have complete integrity.

More advantages are that they know that only registered users can access the application and access the file verification page.

Disadvantages are that if a user's id and password are stolen their account is in jeopardy.

#### 5 Further development or research.

With further research and more time and development, this project could be developed into an integrity checker for wide scale use. Such as a Database for sensitive information for a hospital or for use with Banks.

#### 6 Appendix 1 Project Proposal

**Project Proposal** 

### Data Integrity Checker

Timothy Yong, X13548867, <u>x13548867@student.ncirl.ie</u> / <u>tiimmyyong@gmail.com</u>

BSc (Hones) in Computing

Cyber Security

#### 18/10/2016

### 1. Objectives

1.1 The 1st objective of the Project Is to allow users to view the operations data on their documents stored. This means that it will allow users to Delete, Copy, Move and Modify their data.

1.2 The 2nd objective is that the users will be able to view that recently has done any operations to any data stored.

1.3 The 3rd objective is that whenever the document or data file save has been altered, it will send a report to the account owner showing who altered the file, what time, what they did, from what IP address and how they did.

1.4 The 4th objective is that all of the coding around the software will have safe coding principles implemented everywhere.

### 2. Background

"Cyber security, also referred to as information technology security, focuses on protecting computers, networks, programs and data from unintended or unauthorized access, change or destruction."

(*What is Cyber security*? 2016). Cyber security is all about protecting information on computers. Computers have been used to store sensitive information for nearly

as long as they've been around. Whenever there is something of value, there will be people trying to get it.

Cyber security has been needed as far back as 1988. One of the first recognized worms to affect the world's cyber infrastructure system. The Morris worm caused damage by slowing down computers to the state where they would be unusable. Sine the Morris worm there is only becoming more and more attacks where the attackers are becoming smarter and smarter.

This just shows the need for increased security surrounding anything technological these days, as everything is hack able. This is where my project idea comes from. If a document is as secure as possible but is still hacked knowing who and when hacked it is the most useful information one can have at the time. Data integrity is very important for example. If there is a document stating that someone owes \$100 and it is then altered by an intruder that it is actually \$10,000 or that it is owed to someone else. This could prove very costly to companies.

### 3. Technical Approach

#### Research:

Twilio.io, Raspberry PI, Own Cloud.org, Cloud Developer SDK java, Facebook Developer.

I will begin by developing mock-ups for the monitoring reports. This will include all the fields, which will show, file name changed, email account, time & date, IP address. I will create this mock up using balsamiq or another GUI creator.

I will then address what problems I have from a technical standpoint. These include not knowing fully how to use developing tool kits. This will require me to learn about and how to use the developer's tool kits.

### 4. Requirements

The software is expected and required to work in a particular way. These requirements will be listed below.

The program will work as expected with no errors. It will not be slow or buggy.

The program will allow users to check the integrity of their data.

The program automatically sends out a report every given amount of days/weeks and upon request and also when a data file has been accessed/ modified.

The program will send out accurate reports to the account holder and only the account holder.

## **5. Reading Material**

What is Cyber security? (2016) Available at: http://www.umuc.edu/cybersecurity/about/cybersecurity-basics.cfm (Accessed: 18 October 2016).

Chia, T. (2012) Confidentiality, integrity, and availability: The three components of the CIA triad. Available at: http://security.blogoverflow.com/2012/08/confidentiality-integrity-availability-the-three-components-of-the-cia-triad/ (Accessed: 19 October 2016).

Facebook SDK, OwnCloud.

## 6. Project Plan

So	ftware Project	ect 2016																					
Deli	verables	Name		Oct	41	42	43	44	Nov	46	47	48	Dec	50	51	52	53	Jan	2	3	4	Feb	6
Requirements phase							1	2	3	4	5												
	Task I	Mock ups						n	n	n													
	Task II	Do requirements						d	d	d													
	Task III	Learn Dev tools								n	n												
Prototype phase												1	2	3	4	5							
	Task I	Create prototype									n	n											
	Task II	Start Coding										n	n	n	n								
	Task III	Ensure coding is secure											n	n									
	Task IV	review coding												n	с								
	Task V	begin testing of online capabilitites.													n	n	n						
Ann	otations																						
w	Duration (Weeks)															<b>_</b>							
n	normal																						
d	dependent on others																						
с	critical																						
	type the letter and the colour will change automatically																						

## 7. Evaluation

I will evaluate the system for its integrity, confidentiality and availability but primarily I am going to test for Integrity using SIT.

System integration testing (SIT) is a software testing process that in which testers can verify that all related systems maintain data integrity and can operate in coordination with other systems in the same environment. The testing process ensures that all subcomponents are integrated successfully to provide expected results.

I will evaluate my system on an application. To be decided.

Timothy Yong 19/10/2016

Signature of student and date

#### 7 Appendix 2 Project Requirements Spec

# Requirements Spec (RS)

Integrity Checker

Timothy Yong X13548867 20/04/2017

#### **Document Control**

#### **Revision History**

Date	Version	Scope of Activity	Prepared	Reviewed	Approved
8\11\2016	1	Create	TY	Х	Х
9\11\2016	2	Update	TY	Х	Х
18\04\2017	3	Update	TY	Х	Х
26\04\2017	4	Update	TY		

#### **Distribution List**

Name	Title	Version
Eamon Nolan	Lecturer	1
Sara Kadry	Project Supervisor	1,2,3,4

#### **Related Documents**

Title	Comments
Title of Use Case Model	
Title of Use Case Description	

#### Table of Contents

Appendix 2 Project Requirements Spec

- 1 Introduction
  - 1.1 Purpose
  - 1.2 Project Scope
  - 1.3 Definitions, Acronyms, and Abbreviations
- 2 User Requirements Definition
- 3 Requirements Specification
  - 3.1 Functional requirements
    - 3.1.1 Use Case Diagram
    - 3.1.2 Requirement 1: User Registration and Login
      - 3.1.2.1 Description & Priority
      - 3.1.2.2 Use Case
    - 3.1.3 Requirement 2: File Upload & RWD
      - 3.1.3.1 Description & Priority
      - 3.1.3.2 Use Case
    - 3.1.4 Requirement 3: Report Generation
      - 3.1.4.1 Description & Priority
      - 3.1.4.2 Use Case
  - 3.2 Non-Functional Requirements
    - 3.2.1 Performance/Response time requirement
    - 3.2.2 Availability requirement
    - 3.2.3 Robustness requirement
    - 3.2.4 Security requirement

- 3.2.5 Reliability requirement
- 3.2.6 Maintainability requirement
- 3.2.7 Extendibility requirement
- 4 Interface requirements
  - 4.1 GUI
  - 4.2 Application Programming Interfaces (API)
- 5 System Architecture
- 6 System Evolution

#### **1** Introduction

#### 1.1 Purpose

The purpose of this document is to set out the requirements for the development of my software project called *Integrity Checker* 

The intended customers are anyone who creates and shares sensitive information with the need of ensuring its complete integrity and safety. It caters to users who have the need to have complete control over the viewing and editing of their documents.

#### 1.2 Project Scope

The scope of the project is to develop a system, which will return reports allowing the users to view the editorial timeline and viewing privileges of their documents. The system shall have a report system, which will generate a report based on the specification of the customer. I.e. if a customer selects to choose to receive a report every time a document is viewed or every time a document is edited, the employee will receive a report detailing what document was viewed/ edited and the user, the IP address of the user viewing/editing, the time/ date it was viewed/edited and a previous version of the document before it was edited.

The application will check the integrity of the file by checking its file size and file last updated/-modified field against a previous version to verify its integrity.

The project as it currently is planned out won't cost anything to build. I am using open source programs and free applications such as netbeans to build my project.

As for the software environment I am planning on building the program using Netbeans 8.0.2. I also intend on possibly using owncloud, SDK java, Facebook dev. tools, rubyonrails.

AD	Another Definition
Condf	Confidentiality
Integ	Integrity
Sy. Arch	System Architecture
Auth	Authentication
DB	Database
RWD	Read, Write, Delete
IP Address	Internet Protocol Address - this is an address unique to every
device.	
User	This is the client, The user of the application
System	This is the application, what the user uses.
Doc	Document
MFA	Multi Factor Authentication
APP	Application
GUI	Graphical User Interface

#### 1.3 Definitions, Acronyms, and Abbreviations

#### 2 User Requirements Definition

This section describes the set of objectives and requirements for the system from the customer's perspective. What are the clients saying they want?

Objectives:

The customers want their data to be safe. They want it to be secure and free from external interferences. They want it to be only viewed and edited by those who should be able to view and edit it. They want to be able to see who has edited their documents. They want peace of mind knowing if it has infact been edited that they will then know about it. They want complete control and to be able to monitor the documents integrity.

They want to be able to share sensitive data with the availability of viewing who has accessed said data and from where(ip address).

#### **Requirements:**

The user will require a computer, access username, password, the applications itself stored on the computer. The user will also require a basic understanding of data storage services. eg. File explorer or Finder (mac os x).

#### **3** Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

#### 3.1 Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish.

These requirements are as follows;

User creates an account. User logs in using MFA. Multi Factor Authentication will be used to ensure the most secure connections from the user to the DB.

User is able to upload their data. This can vary from Documents to Images.

System measures file against previous existing files to check for integrity, Ensuring the file has been safe from outside parties.



#### 3.1.1 Use Case Diagram

Actor	Description
User	This is the client, the user of the application.
System Admin	This is the developer, can modify the code, fix bugs.
DBA	DataBase Administrator

• The User uses the system application/ software to create login credentials, this is stored on the DB.

- The User can then login. From here the User can upload files.
- User can use the application to then check the file's integrity. Ensuring it is still confidential.
- The User can also request a report here.

#### 3.1.2 Requirement 1: User Registration and Login

#### 3.1.2.1 Description & Priority

This requirement is essential, as a user cannot use the software without registering and logging in. The login system has extra security features such as a password, captcha and a security question.

#### 3.1.2.2 Use Case

#### Scope

The scope of this use case is to define the registration and login system.

#### Description

This use case describes the registration and login phase.

#### Use Case Diagram



#### **Flow Description**

#### Precondition

The system is in an idle mode until the user begins to register/ login.

#### Activation

This use case starts when an User attempts to register/ login

#### Main flow

- 1. The system identifies the User
- 2. The User enters their information
- 3. The System checks this information.
- 4. The User is granted access to the System. (if information is correct)

#### Alternate flow

A1 : The System accepts this information.

- 1. The system checks the information is correct and allows sign in/ registration.
- 2. The User has access to the system.

#### Exceptional flow

E1 : The System check this

- 3. The system confirms that the information given is incorrect.
- 4. The User is denied access if the information given is incorrect.
- 5. The User is returned to the login/registration page.

#### Termination

The system presents the next level of access for the user. The user now has access to the System.

#### Post condition

The system goes into a wait state

#### 3.1.3 Requirement 2: File Upload & RWD

#### 3.1.3.1 Description & Priority

The file upload and RWD is a very important requirement as it is one of the main functional requirements that the user is able to successfully upload and edit their files.

#### 3.1.3.2 Use Case

#### Scope

The scope of this use case is to show that users will use the application to upload, create, read, write and delete files.

#### Description

This use case describes the user accessing the files. A User may be allowed to access files and read but may not have access to write or delete files.

#### **Use Case Diagram**



File Upload/ RWD

**Flow Description** 

#### Precondition

The system is in initialization mode

#### Activation

This use case starts when a User clicks to access the files.

#### Main flow

- 5. The system identifies the User
- 6. The User accesses the system to upload files.
- 7. The System allows the user to upload files.
- 8. The User has accessed the files.

#### Alternate flow

A1 : The User accesses the system to RWD files.

- 6. The System allows the User to access files.
- 7. The User has RWD access to files.

#### **Exceptional flow**

E1 : The system does not allow access to the user to upload/ rwd files.

- 8. The System denies access as the User does not have access.
- 9. The User does not have access to files it should not be able to access.

#### Termination

The system presents home screen

#### **Post condition**

The system goes into a wait state

#### 3.1.4 Requirement 3: Report Generation

#### 3.1.4.1 Description & Priority

This process in the software is one of the most important and unique to this project's requirements. It outlines the process how a user asks the system for a report. This report will include the file name, the action involved and the user involved along with the time/ date of application and the ip address/ location of the user.

#### 3.1.4.2 Use Case

#### Scope

The scope of this use case is to ask the system for a report on the integrity of said files.

#### Description

This use case describes the User accessing the System to request a report.

#### Use Case Diagram

#### Report Generation



#### **Flow Description**

#### Precondition

The system is in initialisation mode.

#### Activation

This use case starts when an User requests a report on a file. This makes the application compare the 2 documents and ensures that the File is secure.

#### Main flow

- 9. The system identifies the User and allows access to the report generator.
- 10. The User requests a report from the System for a File.

- 11. The System acknowledges the request and checks to ensure the User is allowed to access reports on this file.
- 12. The System generates a report and returns it to the User.

#### Alternate flow

- A1 :The User requests a report from the System for a File.
  - 10. The System accepts the request and fills out the report for the user 11. The User receives the report.

#### **Exceptional flow**

E1 : The System checks to ensure the User is allowed to access reports on

this file.

- 12. The System checks if the User has access to this file.
- 13. The System sees the User should not have access to this files reports.
- 14. The System lets the User know that access to the report has been denied.

#### Termination

The system presents the next page after sending the report.

#### Post condition

The system goes into a wait state

#### 3.2 Non-Functional Requirements

Specifies any other particular non-functional attributes required by the system. Examples are provided below.

#### 3.2.1 Performance/Response time requirement

The system is required to perform as expected. When a User requests to login, The user should be brought to the login page and not the register page.

The User is also expected to be logged in a timely fashion and no loading time is expected.

The system is expected to perform in a timely fashion and load times kept to a minimum.

Upon request, the report generated by the Software is expected to be delivered almost instantly.

#### 3.2.2 Availability requirement

The Application is required to be available at all times once a machine is turned on. The application is running off the computer and not online so Internet availability is not an issue here.

#### 3.2.3 Robustness requirement

In case that the application process cannot connect to the database (eg. DB exceeded the session limit or the account has been locked) the application process shall not freeze / mutate to a zombie process but will close with an error message.

If the application cannot register a new user, the application will notify the user with a message indicating the reason for the user not being able to register (eg. User may not register due to preexisting user with some uname.)

#### 3.2.4 Security requirement

Confidentiality - Protecting the information from disclosure to unauthorized parties. (The System will restrict access to unregistered/ unauthorized users.)

Access Control- Users will not be able to access the system unless they are registered and have permission or are authorized to access the data.

*Privacy* - The User's information will not be shared with third parties. When a User saves data within the DB it is not made public that the user has data stored. it is private.

Integrity - Integrity of information refers to protecting information from being modified by unauthorized parties. The information saved within the database will not be available to unregistered and unauthorized users.

*Origin Authenticity* - This is insuring that the users who are in fact using the the application are in fact the intended user. (eg. someone using someone else's account.)

Availability - Availability ensures that authorized parties are able to access the information when needed. ie it is available to them.

Accountability - Accountability will be found within the software as the System will ensure that the actions of any users can be identified and traced to the author of the operation on the data. (eg. whenever a user modifies something on the DB, the System will store this action and it's time/ date and the user's account and also the IP address.)

Audit logs shall be verbose enough to support forensics. ie. All account modification events shall be logged. The event log shall contain date, time, user, action, object, prior value, new value.

Audit logs shall have integrity protection. Does not allow third parties apps to operate within the System

Session Timeout - To prevent a User leaving their application unattended for large amounts of time granting other unauthorised users access to the application, the System will timeout after a certain amount of time wherein the application is not used. Secure Coding - I will also ensure all the code follows a Secure Coding practice. This ensures my software will be resistant to attack by malicious or mischievous Users/ Programs.

#### 3.2.5 Reliability requirement

This is that the System is reliable. ie it has minimal crashing of the software to none. When a User requests something the system meets this request and does not give an error message or just freeze. I will ensure this with vigorous system testing.

The application will also allow up to 50 users to request a report on a file at once. This will follow the industry standard.

#### 3.2.6 Maintainability requirement

Maintainability is the requirement concerned with how the System can be
restored after a failure, while also considering concepts like preventive
maintenance and Built-In-Test (BIT), required maintainer skill level, and
support equipment. When dealing with the availability requirement, the
maintainability requirement must also be invoked as some level of repair
and restoration to a mission-capable state must be included. One can see
how logistics and logistic support strategies would also be closely related
and be dependent variables at play in the availability requirement.

I will ensure this by following these steps:

- Design for maintainability from the outset.
- Iterative development and regular reviews to ensure a high quality.
- Readable code is easy to understand. I'll comment on every bit of code stating what it does.
- Refactor code to improves its understandability.
- Relevant documentation helps developers understand the software
- Automated builds make the code easy to compile
- Automated tests make it easy to validate changes
- Continuous integration makes the code easier to build and test

Version control helps keep code, tests and documentation up to date and synchronized

#### 3.2.7 Extendibility requirement

The extendibility of this application goes as follows.

It may be used within other applications like cloud based programs (eg. Google drive, Dropbox & OneCloud.

Perhaps there could be a facebook login. So that the registration is instant with a connection to the Users Facebook profile.

Could also be used with photos.

#### 4 Interface requirements



#### 4.1 GUI

File Upload & RWD





#### 4.2 Application Programming Interfaces (API)

Software

The User accesses the software via the GUI.

Here the User is able to access the DB and view their files.

The User also is able to use the Application to check their files integrity.

#### 5 System Architecture

Class Diagram:



#### 6 System Evolution

Perhaps the System may evolve to be able to be used with cloud based DBs. Allowing Users full integrity checking and monitoring over cloud based data storage. Perhaps it could be used with Google Drive or DropBox in the future.

Perhaps it could develop into its own cloud based storage application.

It is possible it could evolve to be used within Facebook as a cloud storage service which would allow users to view reports on data stored on a fb cloud service.

#### References- Reading material.

[1] Chia, T. (2012) Confidentiality, integrity, availability: The three components of the CIA triad. Available at:

http://security.blogoverflow.com/2012/08/confidentiality-integrity-availability-the-three-components-of-the-cia-triad/ (Accessed: 15 November 2016).

Software Security Requirements Available at: http://sqgne.org/presentations/2007-08/Hope-Sep-2007.pdf (Accessed: 15 November 2016).

2016 (2010) *Developing maintainable software*. Available at: https://www.software.ac.uk/resources/guides/developing-maintainable-software (Accessed: 15 November 2016).

#### 8 Appendix 3 Monthly learning reports

**Reflective Journal September** 

## **Reflective Journal**

Student name: Timothy Yong X13548867 Programme: BS Computing - Cyber Security Month: September

#### **1 My Achievements**

This month, I was able to decide on a software idea for my software project class. I had to begin with my thought process which started with trying to figure out a project I could do which I felt was within my ability while still aiming to get as good a grade as possible.

I thought of several different projects but decided with one which related most to my stream of Cyber Security. I planned to create a software which revolved around an alarm system with several different parts for functionality while having a very secure log in system which would have several security functions.

I also prepared slides to present to the other lecturers so that they could have an understanding of all the projects that they didn't get to view pitches off.

I also prepared myself for the pitch and ensured I was on time and well prepared.

#### 2 My Reflection

I felt, i worked well to prepare myself and pitch myself to the lecturers. I was on time and curtious and had I had a better pitch idea I feel it could notve gone better.

However, I feel I was not successful in thinking of an idea which could be used as my softeware project due to lack of scope in the project I submitted.

#### **3 Intended Changes**

Next month, I will try to think of a better software project idea.

I realised that I need to work harder and spend more time in college researching for my software project.

### **4 Supervisor Meetings**

Date of Meeting:

Items discussed:

## **Reflective Journal**

Student name:Timothy Yong X13548867 Programme: BS Computing - Cyber Security Month: October

#### **1 My Achievements**

This month I completed my project proposal and sent it to my Supervisor Sara for review before further reviewing it myself and then submitting it. I submitted it on time and I believe it was to the best of my ability.

I also met with my supervisor Sara once to discuss the project I have choosen for my final year and discussed the complete scope and my assigments regarding the project. She also gave me valuable information to research regarding my project.

I researched the topics Sara gave me among other things I found to further my knowledge on the field and to continute to understand and develop what my project is about.

I have also begun my requirements with basic mockups of how I would like my software to work.

#### **2 My Reflection**

I feel my proposal was done well. I spent a lot of time doing the Gantt chart and I feel this will serve me well in the future as it stands as a timeline for myself for which I must get tasks completed.

I also feel I have done well in researching my project and I have begun my requirements.

#### **3 Intended Changes**

I intend to devote more time to my software project and dedicate a certain amount of time outside college to just doing project work.

#### **4 Supervisor Meetings**

Date of Meeting: 17/10/2016

Items discussed:

Project ideas and scope, research materials, requirements, proposal.

## **Reflective Journal**

Student name: Timothy Yong Programme (e.g., BSc in Computing): Month: November

#### **1 My Achievements**

This month, I was able to begin on implementation on my software project. I also began my technical report.

Building a basic web page with login and registration capabilities.

Connecting a form to allow users to submit data to a database.

Allow users to search the database for their data.

Complete the technical report and send in on time.

I also arranged meetings with my supervisor throughout the month atleast once a week.

#### **2 My Reflection**

I felt I worked well to begin implementation, fill out my documentation and assign meetings with my supervisor.

I felt I could've organised myself better with time management when it came to completing my technical report.

#### **3 Intended Changes**

Next month I will try to prepare myself for my midpoint meeting.

Create a powerpoint and a somewhat working prototype.

I realise I need to get things done earlier in relation to deadlines and not leave it to the last minute.

#### **4 Supervisor Meetings**

Date of Meeting: 10/11/2016

Items discussed: requirements

Action Items:

Date of Meeting: 17/11/2016

Items discussed: complete requirements and implementation

Action Items:

Date of Meeting: 29/11/2016 Items discussed: How to Implement and also Technical report.

Action Items:

## **Reflective Journal**

Student name: Timothy Yong

Programme (e.g., BSc in Computing): BSHC4 Cyber Security

Month: December

#### **1 My Achievements**

This month, I was able to create a presentation and have some form of a prototype ready for my presentation. I also presented during this month to 2 lecturers. I showed them my prototype and my future plans for my project also. The presentation was spaced over 20 minutes.

My contributions to the projects included adding a db to the project. Adding admin capabilities.

#### 2 My Reflection

I felt, it worked well to create a prototype.

However, I was not successful in finishing the prototype. There were certain aspects of the prototype which I did not have completed.

#### **3 Intended Changes**

Next month, I will try to continue working on the prototype.

#### **4 Supervisor Meetings**

Date of Meeting: 2/12/2016 Items discussed: requirements Action Items:

Date of Meeting: 9/12/2016 Items discussed: Complete powerpoint Action Items:

Date of Meeting:16/12/2016

Items discussed: Report

Action Items:

## **Reflective Journal**

Student name: Timothy Yong

Programme (e.g., BSc in Computing): BSHC4 Cyber Security

Month: January

#### **1 My Achievements**

This month I continued to learn about security features which I intend on implementing. I have continued to try organise myself to set myself weekly deadlines so I can work towards them. I have begun redesigning the user login and register system to create user log sessions to secure usage across the application. I have also looked into implementing a ip look up app to go with the application.

#### **2 My Reflection**

I felt, it worked well to create a prototype.

However, I was not successful in finishing the prototype. There were certain aspects of the prototype, which I did not have completed.

#### **3 Intended Changes**

Next month, I will try to continue working on the prototype and create deadlines and continue to work towards them myself.

## **Reflective Journal**

Student name: Timothy Yong

Programme (e.g., BSc in Computing): BSHC4 Cyber Security

Month: March

#### **1 My Achievements**

This month I had to redesign my project. I decided to use Visual Studio to recode the website as it has a lot more features I could use. I have implemented multi factor authentication into my project. I have begun creating the database and also begun on the report system.

#### **2 My Reflection**

I feel there is still a lot of work to do for the project but as I have more time now due to the labs and lectures finishing up I will get the project finished on time. I also have noticed I do not have as much time as I would like as my exams are fast encroaching also.

#### **3 Intended Changes**

I have changed from Netbeans, HTML and CSS to Visual Studio and C#.

#### **4 Supervisor Meetings**

Date of Meeting: 23/3/2016

Items discussed: Security Features, Project showcase, Project document

#### **5** References

[1]What is Cyber security? (2016) Available at: http://www.umuc.edu/cybersecurity/about/cybersecurity-basics.cfm (Accessed: 18 October 2016).

Chia, T. (2012) *Confidentiality, integrity, availability: The three components of the CIA triad.* Available at: http://security.blogoverflow.com/2012/08/confidentiality-integrity-availability-the-three-components-of-the-cia-triad/ (Accessed: 19 October 2016).

Web testing: A complete guide about testing web applications (2007) Available at: http://www.softwaretestinghelp.com/web-application-testing/ (Accessed: 01 December 2016).

Cyber insurance, security and data integrity (2014) Available at: http://www.ey.com/Publication/vwLUAssets/EY\_-\_Insights\_into\_cyber\_security\_and\_risk/\$FILE/ey-cyber-insurance-thoughtleadership.pdf (Accessed: 21 March 2017).

IBM Knowledge Center - Document Integrity Available at: https://www.ibm.com/support/knowledgecenter/SSYQ89\_10.1.0/com.ibm.help.tip sguide.doc/R\_ContentIntegrity.html (Accessed: 20 April 2017)

[2]Complete Guide to web-testing. [ONLINE] Available at: http://www.softwaretestinghelp.com/web-application-testing/. [Accessed 2 May 2017].

[3] APPLICATION SELF-PROTECTION IS FINALLY HERE [ONLINE] Available at: https://hdivsecurity.com/. [Accessed 5 May 2017].

#### Research:

http://www.jerriepelser.com/blog/using-google-authenticator-asp-net-identity/

https://www.toptal.com/security/10-most-common-web-security-vulnerabilities

http://www.jerriepelser.com/blog/using-google-authenticator-asp-net-identity/

https://www.twilio.com/docs/tutorials/walkthrough/server-notifications/csharp/mvc