National College of Ireland

BSc in Computing

2017/2018

Aaron Fitzgerald

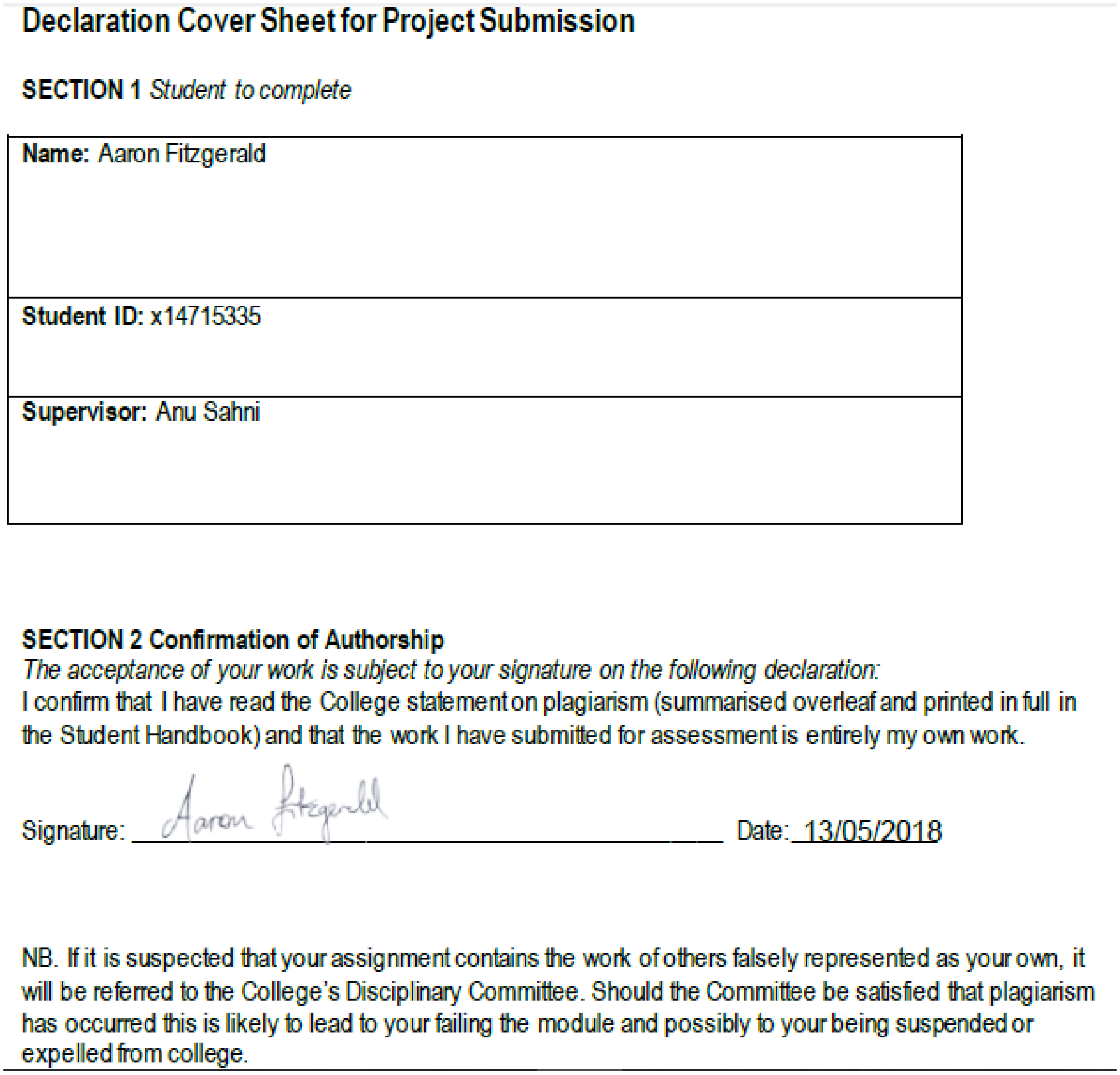
x14715335

x14715335@student.ncirl.ie

CyBly – Online Bullying Education

Technical Report

Table of Contents



Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Aaron Fitzgerald
Student ID: x14715335
Supervisor: Anu Sahni

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: Aaron Fitzgerald Date: 13/05/2018

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Executive Summary

This document outlines the approach taken from the requirements elicitation to the development of a modern web application integrated with resourceful tools focussing on cyber bullying. Utilizing multiple programming languages across two Integrated Development Environments (IDE's), the application seeks to deliver the best solution to a sensitive and complex issue.

Cyber bullying, bullying that happens over digital devices (phones, laptops, games consoles) is a growing problem among users of technology and the internet however primarily with younger users such as teenagers. Discussed within this technical report is CyBly, an interactive web application that provides methods of analysing and visualising cyber bullying.

The application is responsive and accessible, aiming to accommodate mobile and desktop users. The main goal is to improve detection, provide new educational techniques and guide victims to help.

Key features include:

Text Analysation Tool

Developed in C#, the text analysation tool uses several datasets to analyse text and detect uses of inappropriate language targeted at a person or group.

Data Visualisation

Data gathered through user questionnaires is used to populate charts from the Google Charts library and are displayed throughout the application.

Educational Challenges

Presenting facts to the user, a time-based quiz is presented to provoke thought and educate.

1 Introduction

My objective for this project is to create an online educational platform about online bullying, more commonly known as cyber-bullying. This will be achieved through the implementation of several strategies:

- Text Analyzation Tool
    - A downloadable program that can be used to analyse any text (e.g. Tweets/Posts) and returns whether there is content with the potential to cause offence.
- Gamification
    - Quizzes and interactive challenges to test the understanding on the subject while improving user knowledge.
- Visualization
    - Through analyzation of responses to quizzes, results will be aggregated by type to represent an overall picture. As more users use the application the visualization will provide a more accurate picture.
- Online Resources / Where to turn for help
    - References to other resources on the web, made easily accessible throughout the application.

1.1 Background

My project idea is to develop a cyber bullying information portal with elements including: gamification, data visualization, online resources and a text analyzation tool. The rapid growth of users connecting to the internet via Social Media and Gaming among others means many children/teens are being exposed to open platforms which, due to their scale are hard to regulate. This opens the door to cyber bullying, which is more prevalent now than ever and can have serious effects on teens and even younger. My project will aim to provide information on how to get help, how to enhance knowledge on the subject and even detect instances of it.

1.2 Aims

- Educate users on the topic via information, all categorizations of cyber bullying
- Educate the user through gamification. Interactive quizzes (identify cyber bullying, reinforces key topics)
- Provide insightful visualisations
- Provide a tool to detect cyber bullying by analysing contents contained in each line of text for a whole file.
- Enable users to access the portal from a desktop or mobile.

1.3 Technologies

PHP

PHP (Hypertext Pre-processor) is a server-side scripting language used primarily for web applications. This language is commonly used to transfer client-side data to a database.

SQL / ClearDB

SQL (Structured Query Language) is a language used for managing data in relational databases. This language helps with delivering CRUD (Create, Read, Update, Delete) functionality.

Bootstrap / HTML / CSS / JavaScript / jQuery

The combination of these languages and technologies will ensure a well-integrated client-side design is implemented. HTML (Hypertext Mark-up Language) is used to develop elements on the page and is structured so that it can be understood by the browser. CSS (Cascading Style Sheets) are used to bring design to the html elements ranging from colours to fonts etc. Bootstrap is a front-end library made up of HTML and CSS components which help create intuitive applications. JavaScript and jQuery are interpreted languages that add dynamic behaviour to client-side pages. Elements taken into consideration when developing with these technologies:

- Mobile responsiveness and compatibility
- Single theme is present throughout application
- Commented, structured and well-defined code

Visual Studio / C#

C# (C Sharp) is an object-oriented general-purpose programming language developed by Microsoft. It is often compared to Java as its main competitor as both share similar characteristics such as being object-oriented and strongly typed.

Visual Studio is an IDE (Integrated Development Environment) developed by Microsoft. Visual Studio can be used with many languages created by Microsoft and is used to create web services, programs, mobile applications and so on.

Cloud9

Cloud9 is a cloud based IDE (Integrated Development Environment) currently hosted by Amazon Web Services. Cloud9 facilitates a large range of programming languages.

Heroku

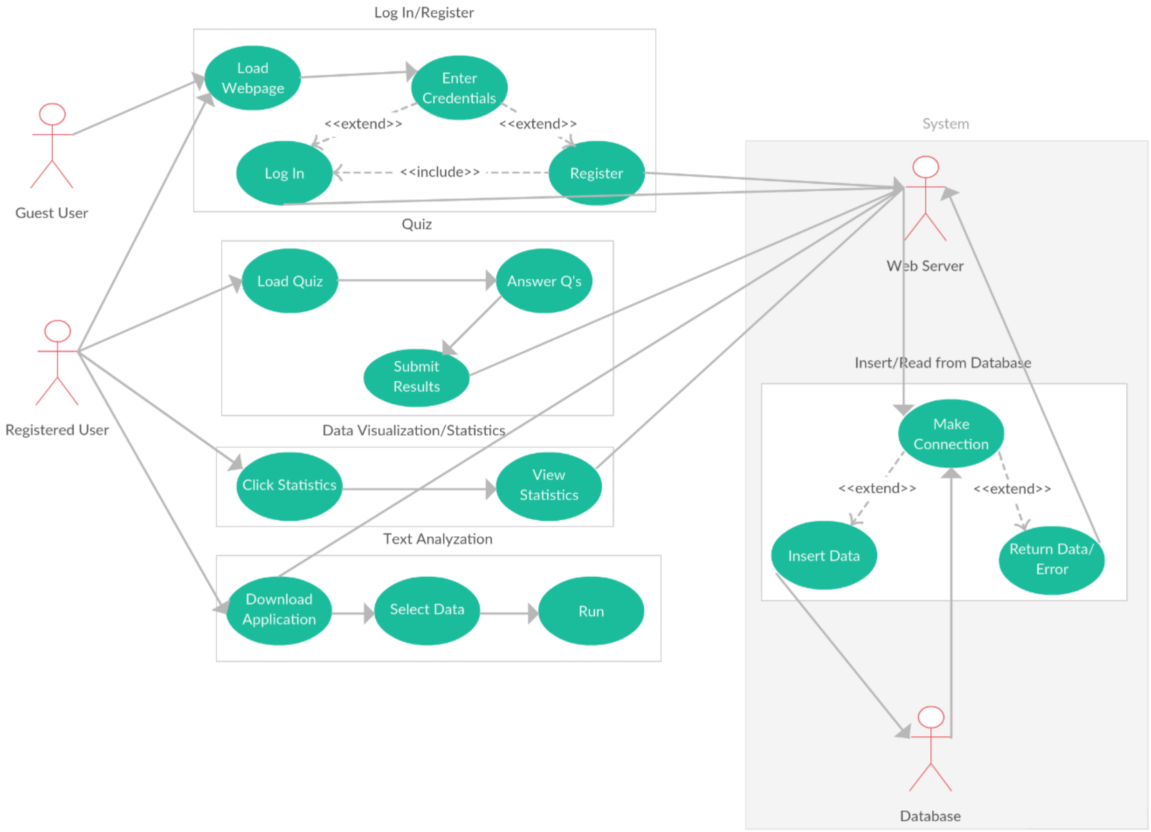
Heroku is a cloud platform used to deploy applications to a Heroku app which is hosted on their servers.

2 System

2.1 Requirements

2.1.1 Functional Requirements

2.1.1.1 Use Case Diagram



2.1.1.2 Requirement 1 <Read/Write From/To Database>

2.1.1.2.1 Description & Priority

The <Web Server> needs to communicate data to/from a <Registered User> to/from the <Database>.

Priority: 1

2.1.1.2.2 Use Case

Read/Write To/From Database

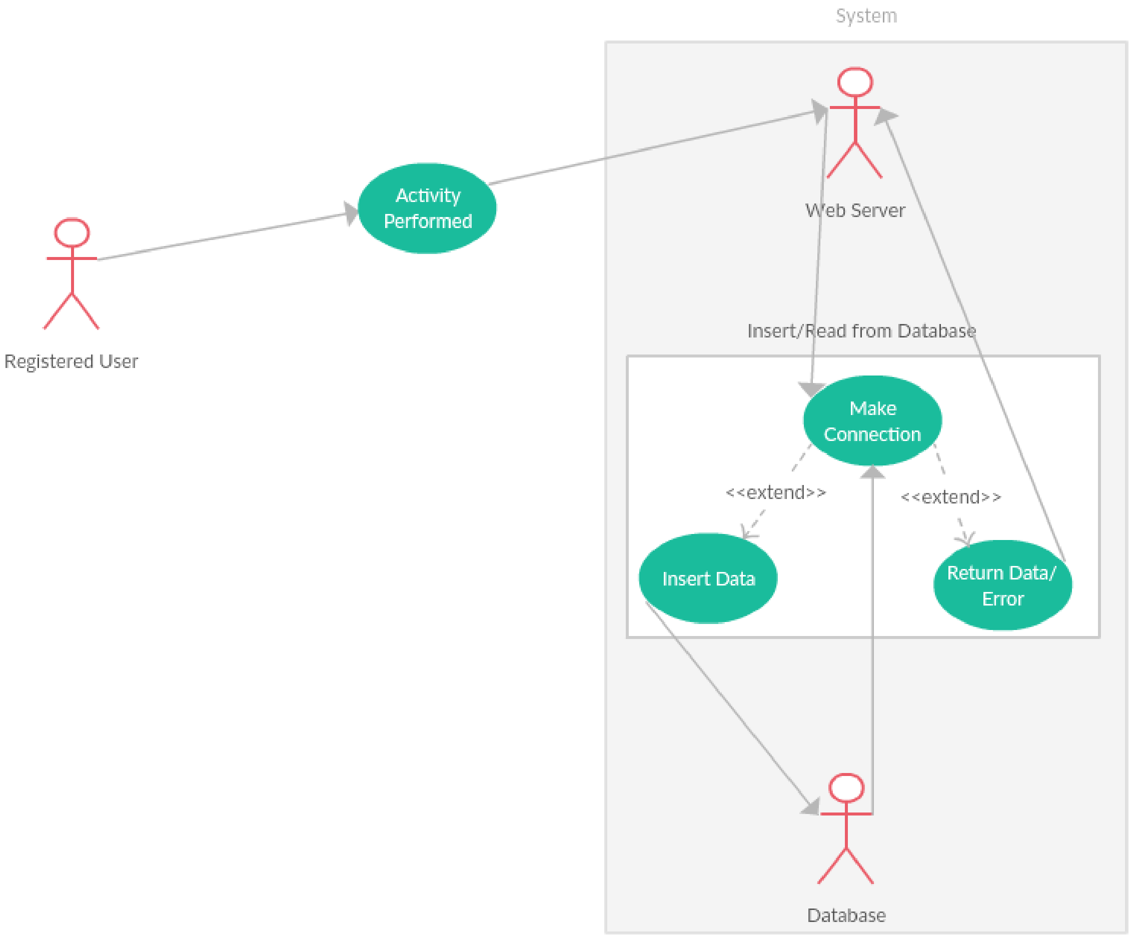
Scope

The scope of this use case is to successfully pass data through the entirety of the system.

Description

This use case describes the action of an <Registered User> performing an activity where data is sent and received.

Use Case Diagram



Flow Description

Precondition

The main page of the application is loaded on the <Guest User> device.

Activation

This use case starts when a <Registered User> performs an activity.

Main flow

1. The <Registered User> makes a request to the <Web Server> via some action.
2. The <Web Server> initiates a connection with the <Database>.
3. The connection was unsuccessful. (See E1)
4. The <Database> is connected successfully.
5. Data from the <Registered User> is passed to the <Web Server> and down to the <Database>.
6. Data is then sent back from the <Database> to the <Web Server> and then presented to the <Registered User>

Exceptional flow

E1: <Unable to establish database connection>

1. The appropriate error code is presented to the <Web Server>
2. An appropriate error message is returned to the <Registered User>.

Termination

The system closes the connection with the database.

Post condition

The system goes into a wait state.

2.1.1.3 Requirement 2 <Registration>

2.1.1.3.1 Description & Priority

The <Guest User> wishes to register to gain all privileges.

Priority: 2

2.1.1.3.2 Use Case

Registration

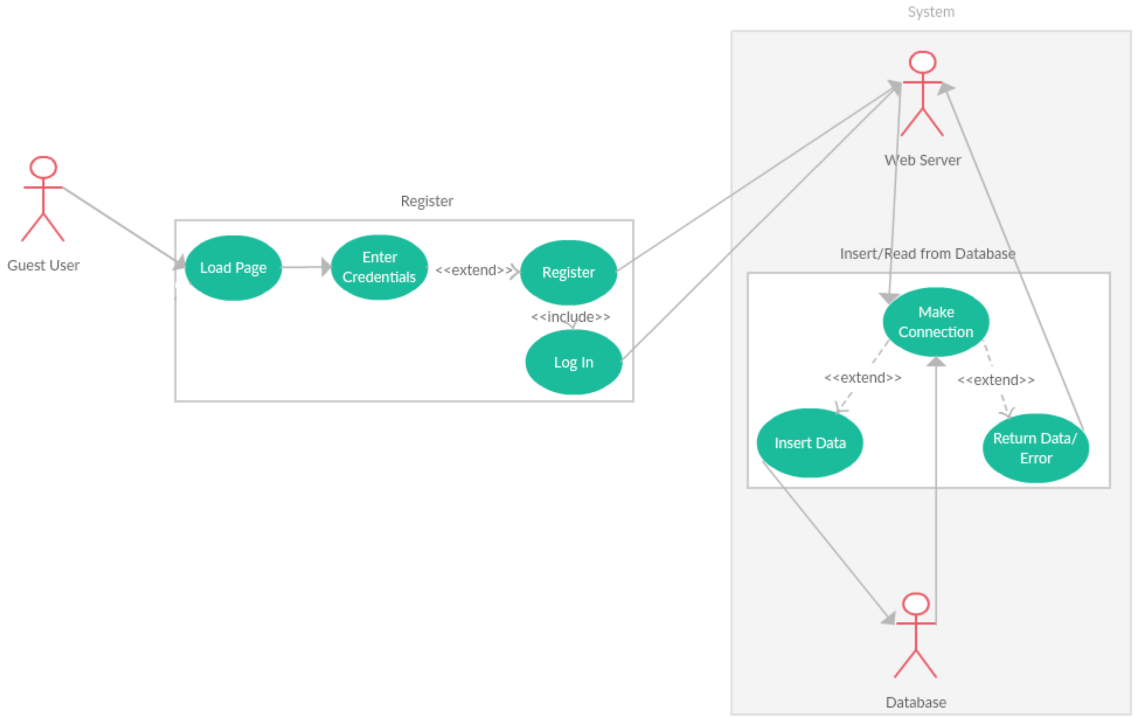
Scope

The scope of this use case is to register a <Guest User> onto the system.

Description

This use case describes the action of an <Guest User> registering onto the system.

Use Case Diagram



Flow Description

Precondition

The main page of the application is loaded on the <Guest User> device.

Activation

This use case starts when a <Guest User> clicks ‘Register’.

Main flow

7. The system presents the Registration form.
8. The <Guest User> may want to cancel the registration process. (See E1)
9. The <Guest User> fills out the on-screen form. (All Required Fields)
10. The <Guest User> finishes the registration process by pressing ‘Register’.

11. If the system concludes that the information entered is not valid, the system displays a message. (See E2)
12. Once the information provided by the <Guest User> is correct the system stores the data to the database and permits registration and sign in.

Exceptional flow

E1: <Cancellation of Registration Process>

3. The system displays the registration form
4. The <Guest User> decides to cancel the registration.
5. The system returns to the main page.

E2: <Incorrect Details>

1. The system displays the form.
2. The <Guest User> fills out the form with required information.
3. The details are invalid and the system notifies the user to try again with errors provided

Termination

The system presents the home page and permits access to all functions.

Post condition

The system goes into a wait state.

2.1.1.4 Requirement 3 <Registered User Sign In>

2.1.1.4.1 Description & Priority

A <Registered User> must sign into the application to access all features.

Priority: 3

2.1.1.4.2 Use Case

Log In

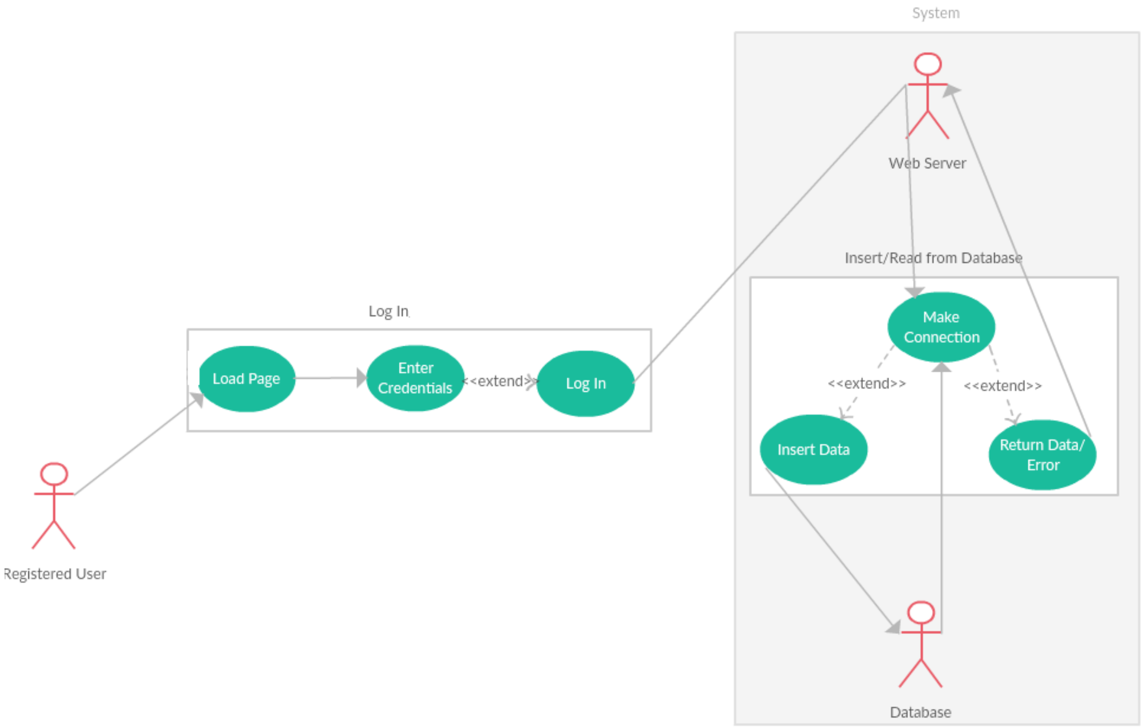
Scope

The scope of this use case is to sign a <Registered User> into the system.

Description

This use case describes the action of an <Registered User> signing into the system.

Use Case Diagram



Flow Description

Precondition

The main page of the application is loaded on the <Registered User> device.

Activation

This use case starts when a <Registered User> clicks 'Sign in'.

Main flow

1. The system presents the sign in form.
2. The <Registered User> may want to cancel the sign in process. (See A1)
3. The <Registered User> fills out the on-screen form. (Login/Password)
4. The <Registered User> finishes the sign in process by pressing 'sign in'.
5. If the system concludes that the information entered is not valid, the system displays a message. (See E1)
6. Once the information provided by the <Registered User> is correct the system retrieves the data from the database and permits sign in.

Alternate flow

A1: <Cancellation of Sign in Process>

1. The system displays the sign in form
2. The <Registered User> decides to cancel the sign in.
3. The system returns to the main page.

Exceptional flow

E1: <Incorrect Details>

1. The system displays the form.
2. The <Registered User> fills out the form with their credentials
3. The details are incorrect and the system notifies the user to either try again or register for an account.

Termination

The system presents the home page and permits access to all functions.

Post condition

The system goes into a wait state.

2.1.1.5 Requirement 4 <Take Questionnaire/Quiz>

2.1.1.5.1 Description & Priority

The <Registered User> wishes to take a quiz.

Priority: 4

2.1.1.5.2 Use Case

Quiz

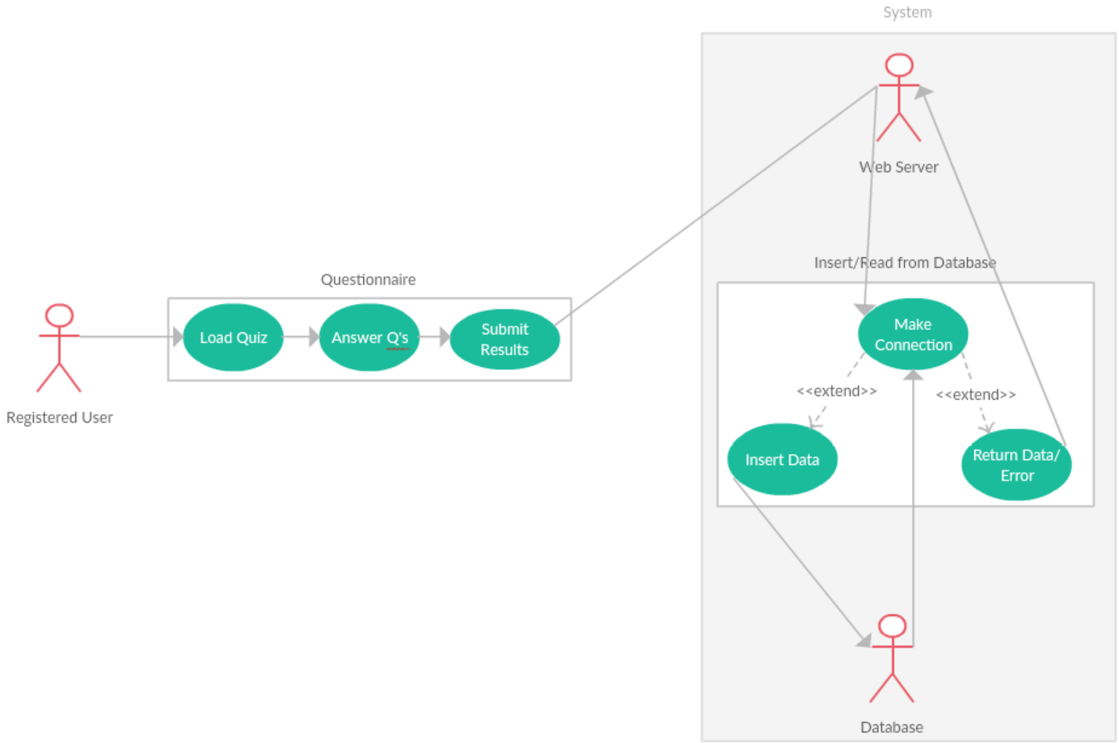
Scope

The scope of this use case is to allow a <Registered User> to complete a Questionnaire.

Description

This use case describes the action of an <Registered User> taking a Questionnaire.

Use Case Diagram



Flow Description

Precondition

The <Registered User> is signed in.

The quiz page of the application is loaded on the <Registered User> device.

Activation

This use case starts when a <Registered User> clicks ‘Take Questionnaire’.

Main flow

1. The system presents the quiz form/interface.
2. The <Registered User> may want to cancel the quiz taking process. (See A1)
3. The <Registered User> answers quiz questions. (All Required Fields)
4. The <Registered User> finishes the quiz process by pressing ‘Submit’.
5. If the system concludes that the information entered is not valid, the system displays a message. (See E1)
6. Once the information provided by the <Registered User> is correct the system stores the data to the database and returns a success message to the <Registered User>.

Alternate flow

A1: <Cancellation of Quiz>

1. The system displays the Quiz.
2. The <Registered User> decides to cancel the quiz.
3. The system returns to the main page.

Exceptional flow

E1: <Invalid Answers>

1. The system displays the form.
2. The <Registered User> completes the quiz with required information.
3. The answers are invalid and the system notifies the user to try again with errors provided.

Termination

The system returns a success message.

Post condition

The system goes into a wait state.

2.1.1.6 Requirement 5 <View Statistics>

2.1.1.6.1 Description & Priority

The <Registered User> wishes to view statistics stored in the system database

Priority: 5

2.1.1.6.2 Use Case

Data Visualization/Statistics

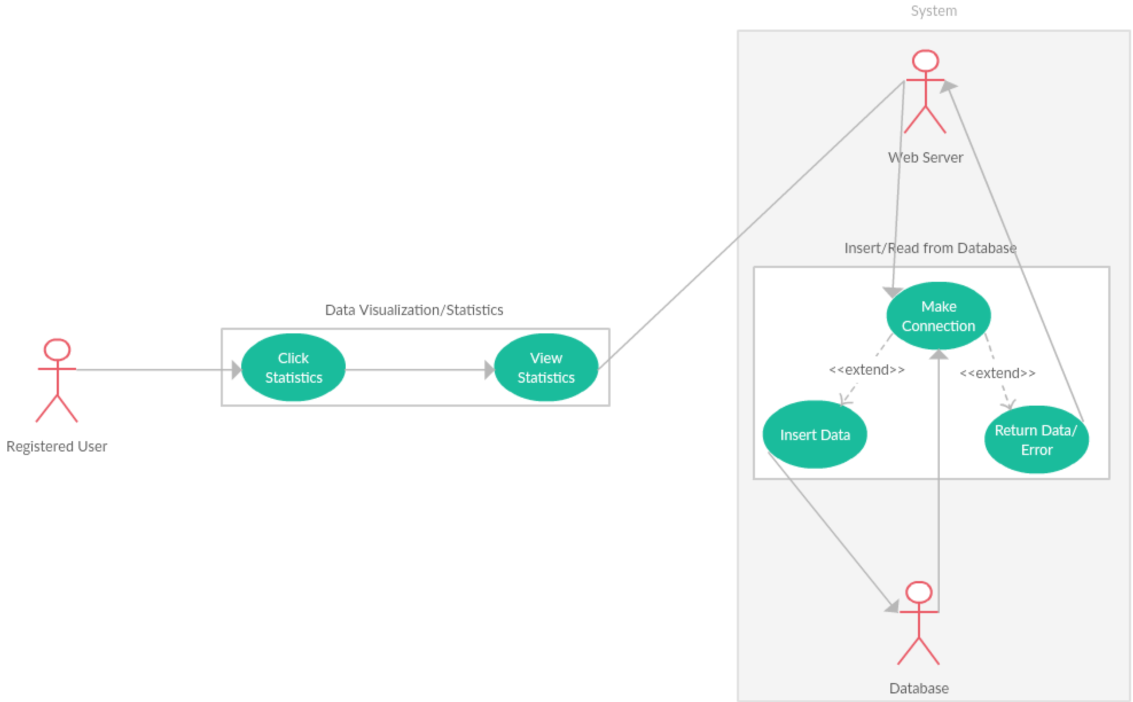
Scope

The scope of this use case is to allow a <Registered User> to view a graphical representation of statistics.

Description

This use case describes how a <Registered User> views a graphical representation of statistics retrieved from the system database.

Use Case Diagram



Flow Description

Precondition

The <Registered User> is signed in.

Activation

This use case starts when a <Registered User> clicks ‘Visualization’.

Main flow

1. The web server queries the database for data.
2. Insufficient data is returned. (See E1)
3. The data retrieved is sorted into relevant charts.
4. The <Registered User> views statistics.
5. The <Registered User> may want additional information. (See A1)

Alternate flow

A1: <Additional Information>

1. The <Registered User> clicks on a chart

2. The system expands chart details to provide further information

Exceptional flow

E1: <Insufficient Data>

1. The system notifies the user that insufficient data was found for the request.
2. The <Registered User> clicks ‘ok’.

Termination

The <Registered User> finishes the activity by navigating away from this page.

Post condition

The system goes into a wait state.

2.1.1.7 Requirement 6 <Text Analyser>

2.1.1.7.1 Description & Priority

The <Registered User> wishes to analyse text from their local machine.

Priority: 6

2.1.1.7.2 Use Case

Text Analyzation

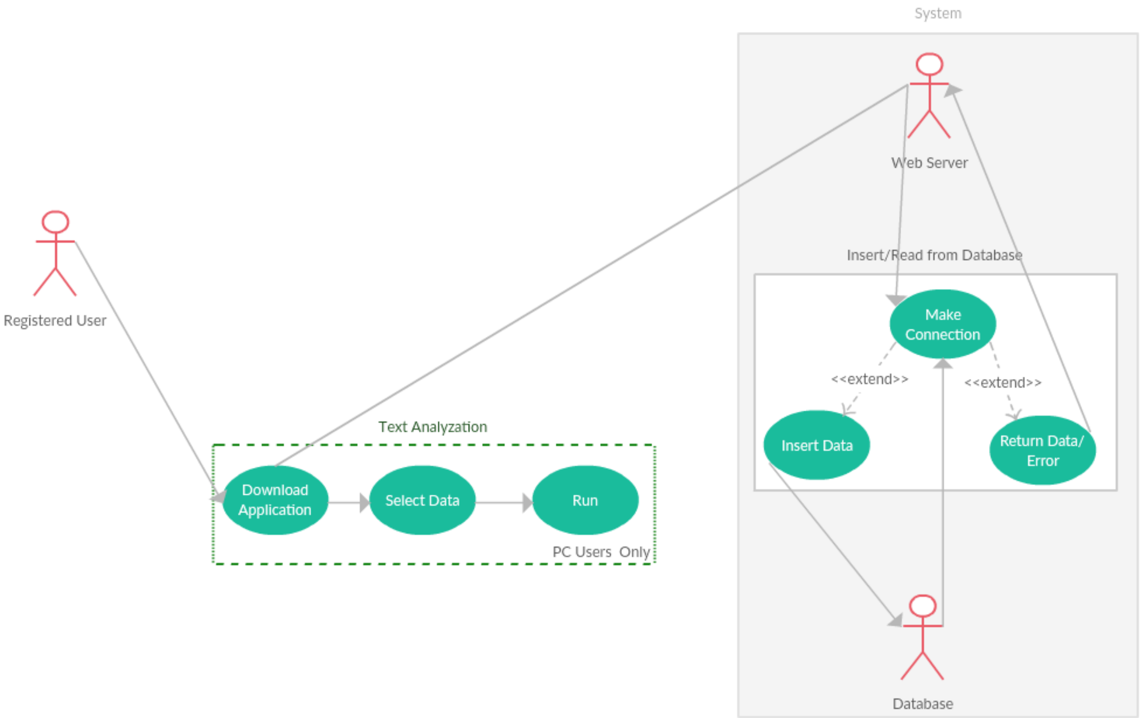
Scope

The scope of this use case is to allow a <Registered User> to analyse text via a downloadable program.

Description

This use case describes how a <Registered User> performs a run on a given piece of text for the presence of potentially offensive material.

Use Case Diagram



Flow Description

Precondition

The <Registered User> has previously signed in.

The text analysation tool has been downloaded.

Activation

This use case starts when a <Registered User> launches the text analysation program.

Main flow

1. The <Registered User> creates a .txt file with content they wish to be searched.
2. The <Registered User> selects this file as the target for the program.
3. The <Registered User> selects ‘Run’.
4. The program analyses all data in the .txt file and returns appropriate output.

Termination

The <Registered User> finishes the activity by navigating away from this page.

Post condition

The system goes into a wait state.

2.1.2 Non-Functional Requirements

2.1.2.1 Performance

The most demanding portion of the application may be generating the graphics produced from the data after passing through the algorithms. Using the lightweight Google Charts API, a response time classified between instantaneous and 3 seconds can be achieved.

Ensuring the system doesn't hang at any point is also essential in achieving the desired response time. Reducing the workload of the server using the methods such as minimal animations and only performing light server actions help us reach this target.

A possible future issue of high traffic on the server will be continually monitored through the web portal.

2.1.2.2 Availability

The requirement of availability is crucial to all areas of the application.

Web

- Heroku publishes uptime for their servers which currently averages 99.999%. 99% uptime will be the minimum acceptable for our application.
- The user should have unrestricted access to all areas of the application if they are registered; database inaccuracies must be nullified.

Mobile

- Mobile users will access CyBly through the web browser on their mobile so the same conditions as for web apply.

Database

- Accuracy is the highest priority requirement of the database. When designed correctly we should achieve maximum availability.

2.1.2.3 Security

Given the sensitivity of data handled, the user would expect their data to be protected. Passwords will be encrypted and hashed when interacting with the database to ensure that if the line of communication is intercepted data can’t be easily understood.

- For first time sign-ups, captchas will be used to ensure the user is genuine.

2.1.3 Data Requirements

- Verification between the client, webserver and database must happen in the event of user sign in.
- The database must valid user credentials and information for use at all times.
- Usernames and emails must be unique.
- A primary key must be present in all tables of the database, with linkages as appropriate.

2.1.4 User Requirements

The user has the following requirements expected through any stage of the application:

- The <End User> expects the system to react to requests in a timely manner where waiting times above 3 seconds is unacceptable for basic operations.
- The <End User> expects to retrieve accurate data on request.
- The <End User> expects an easy to navigate and intuitive user interface.
- The <End User> expects their data to be secure.

2.1.5 Environmental Requirements

Since CyBly is a project that that operated in many environments, the user must consider and adhere to the following requirements for the selected client device to run optimally.

Web Environment

- The CyBly web application will run on modern web browsers such as Chrome, Firefox and Microsoft Edge.

- JavaScript must be allowed to run.
- The hardware shouldn’t play much of a factor once the previous two conditions are met; memory usage will be low, certainly under 50mb which is more than reasonable.
- Desktop and mobile are supported as design is responsive.
- An internet connection is required.

Text Analysation Tool

- .NET framework to be installed on users’ PC to run executable windows form application.

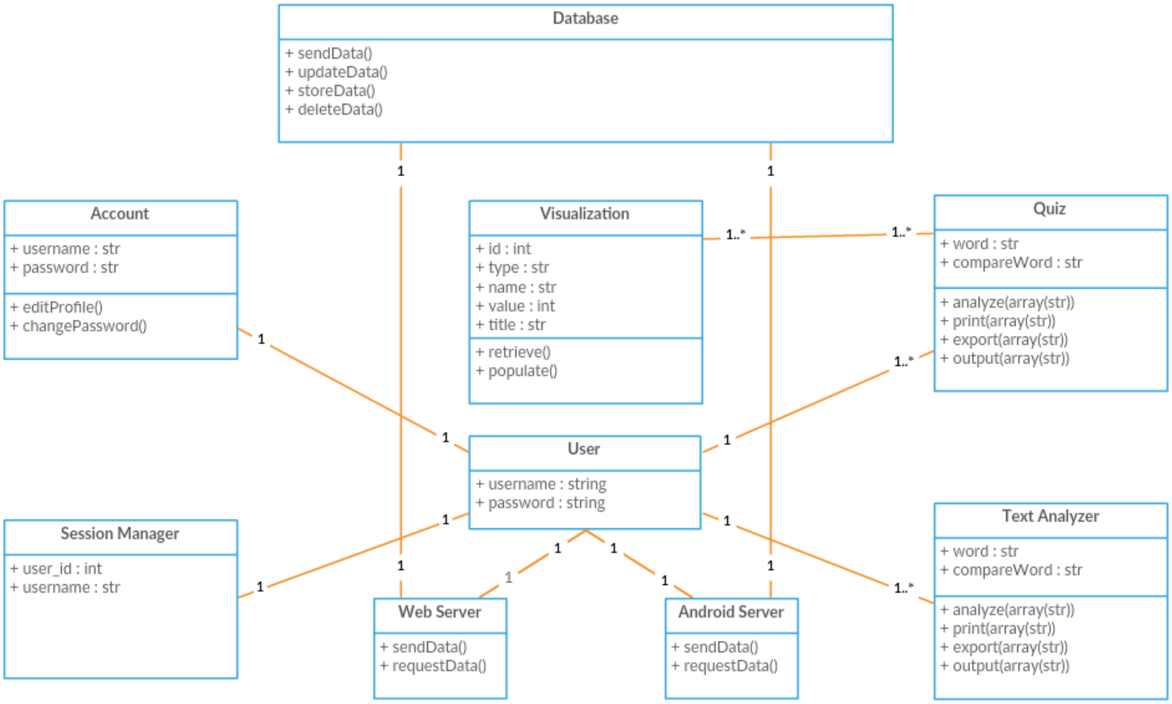
2.1.6 Usability Requirements

All areas of the system should be overly accommodating even for users who are not use to websites/apps. Ensuring calls for action are clearly labelled and leave no ambiguity is key to ensure a usable system.

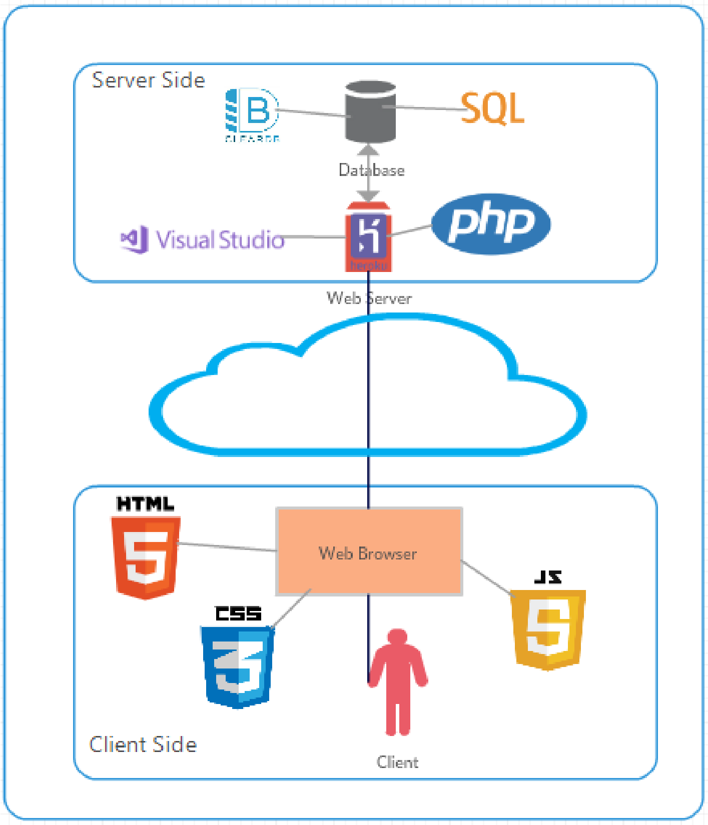
- Speed to goal. The user should reach their goal in the fastest achievable way. Achieved by minimizing unnecessary content.
- Intuitiveness. Smooth animations, large icons and text make the application more user friendly.

2.2 Design and Architecture

Class Diagram



Architecture Diagram



2.3 *Implementation*

2.3.1 Development Environment

The development environment involved multiple IDE's (Integrated Development Environments) in order to accommodate the need for my application.

Cloud9

Cloud9 was where the entirety of the web application portion of my project was completed. This IDE was invaluable as it allows programming with multiple languages, supports local and remote git repositories and supported the Heroku cli for deployment later in the project. Accessibility was also a key factor for choosing Cloud9, the ability to develop from any computer as the IDE stores all data in the cloud was a great convenience.

Visual Studio

Visual Studio was straight-forward to get working with as I remember doing some .NET Framework web applications in third year. Although this wasn’t a web application I wanted to try something new and create an executable program from a windows form application. I believe C# is a great programming language that is more flexible and more user friendly than Java and is a good choice for my needs.

2.3.2 Database

2.3.2.1 Language

The database language chosen for my project was SQL (Structured Query Language). I chose SQL because I had experience with it through second year and like the structured, relational nature of the language.

To use SQL and deploy on Heroku I had to use the Heroku add-on ClearDB. ClearDB is a database as a service (DBaaS) which stores and manages MySQL databases and uses advance fail resistant technology.

2.3.2.2 Database Structure

Database Name

cybly

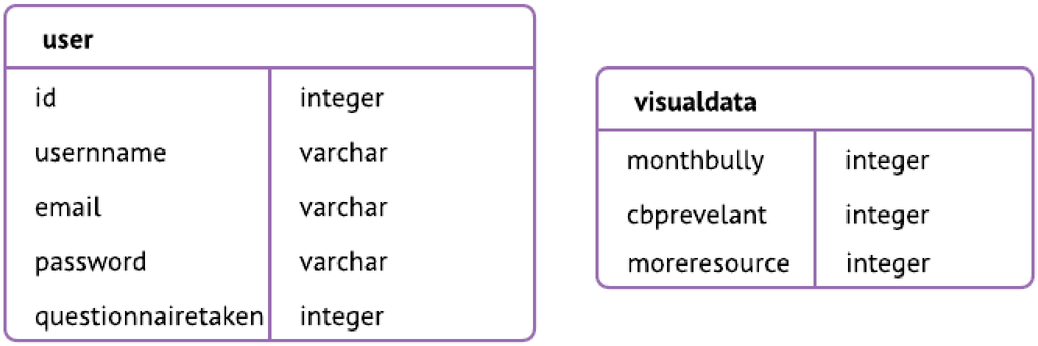
Database Tables

Table Name	users		
Field Name	Data Type (length)	Properties	Description
id	int(50)	AUTO_INCREMENT, PK	Auto assigned user id
username	varchar(100)		Unique user username
email	varchar(200)		Unique user email address

password	varchar(100)		Password stored in hash form
questionnairetaken	int(50)	RANGE (0..1) DEFAULT 0	0 if questionnaire not taken, 1 otherwise

Table Name	visualdata		
Field Name	Data Type (length)	Properties	Description (cb = cyber bullying)
monthbully	int(50)	RANGE (1..3)	Has cb been witnessed in the past month
cbprevelant	int(50)	RANGE (1..4)	Where is cb most prevelant
moreresource	int(50)	RANGE (1..2)	Is more resources needed for cb

Database ER Diagram



user	
id	integer
usernname	varchar
email	varchar
password	varchar
questionnairetaken	integer

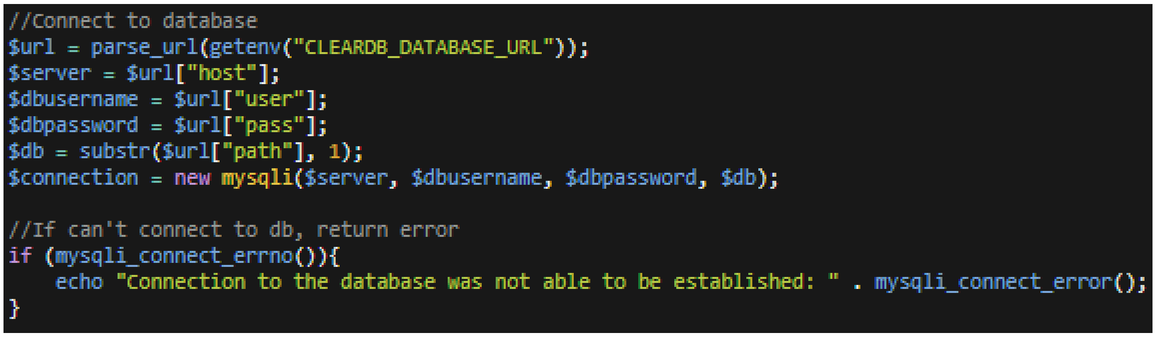
visualdata	
monthbully	integer
cbprevelant	integer
moreresource	integer

Notice how there is no relationship between the tables in my database. The reason for this is to keep data submitted for the questionnaire anonymised as it is a sensitive subject. When a user takes the questionnaire then the field ‘questionnairetaken’ get assigned a value of 1 so they cannot take another one and their answers are not linked to them.

2.3.2.3 *Connecting to the Database*

Connecting to the database is achieved through the PHP language using the MySQLi functions that are integrated with PHP.

Opening a new connection



```
//Connect to database
$url = parse_url(getenv("CLEARDB_DATABASE_URL"));
$server = $url["host"];
$dbusername = $url["user"];
$dbpassword = $url["pass"];
$db = substr($url["path"], 1);
$connection = new mysqli($server, $dbusername, $dbpassword, $db);

//If can't connect to db, return error
if (mysqli_connect_errno()){
    echo "Connection to the database was not able to be established: " . mysqli_connect_error();
}
```

The above mysqli connection function takes four parameters extracted from the CLEARDB_DATABASE_URL URL which are: server, database username, database password and database path. If the connection is not successful, PHP will echo the error message and a user-friendly message.

Connection to the database is done in the index.php to populate data elements, registration.php, login.php and visualdata.php.

2.3.3 Registration, Login and Security

2.3.3.1 Registration

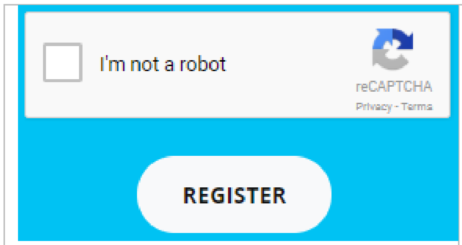
The first step in developing a user registration is determining what data you want to store relating to a user. For my application I only wanted to store what was necessary to register a user and get them on the application quickly. The fields required for registration and their conditions are as follows:

Field Name	**Condition(s)/Validation**
Username	Not null, not empty, at least 3 chars, must not be taken
Email	Must be a valid email, must not be taken
Password	Not null, not empty, at least 7 chars
Confirm Password	Must match password
Google Captcha	Must be successful

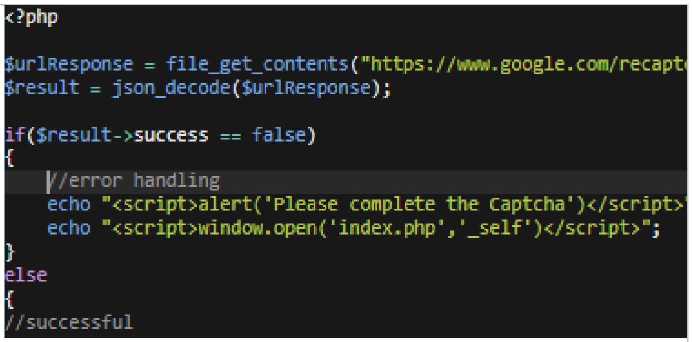
Validation is done through PHP after collecting the ‘POST’ variables via the name tags on the HTML form.

Authenticating User

To authenticate the user as a genuine human user, a Google Captcha must be completed. This is checked first to ensure it returned successfully before attempting to registering the user.



Google Captcha



```
<?php

$urlResponse = file_get_contents("https://www.google.com/recapt
$result = json_decode($urlResponse);

if($result->success == false)
{
    //error handling
    echo "<script>alert('Please complete the Captcha')</script>'
    echo "<script>window.open('index.php','_self')</script>";
}
else
{
//successful
```

Checking captcha response before registration

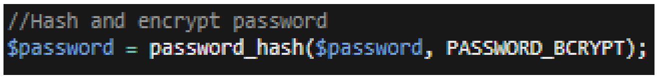
2.3.3.2 Login

The login process requires the user to enter a combination of either their username and password or email address and password. The first check performed is ensuring the username or email entered is registered in the database. This user record is then retrieved for authentication.

Field Name	Condition(s)/Validation
Username/Email	Must be registered to a user record
Password	Must pass password authentication

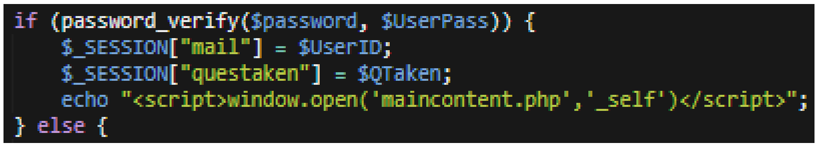
Password Authentication

Upon successfully registering for an account, the users password is hashed and encrypted using the secure method 'password_hash($pass, PASSWORD_BCRYPT)' through PHP and stored in the database. A great feature of this method is that it generates and contains a salt for the hashed password therefore doesn't require the manual storage of one:



```
//Hash and encrypt password
$password = password_hash($password, PASSWORD_BCRYPT);
```

To then login, the password must be decrypted using the PHP method 'password_verify($plaintextPassword, $encryptedPassword)':



```
if (password_verify($password, $UserPass)) {
    $_SESSION["mail"] = $UserID;
    $_SESSION["questaken"] = $QTaken;
    echo "<script>window.open('maincontent.php','_self')</script>";
} else {
```

This is a very secure approach to handling user passwords. As they are stored in encrypted form, if a data breach were to occur, user passwords would still be unidentifiable.

Sessions

The need to know if a user is logged in and the need to access and display a users' username across multiple pages was a necessary function. To acquire this I used PHP sessions. PHP sessions allows the web server to store certain information in session variable for a user, usually until the browser is closed. Users may not know what is being stored in a session which makes it different to a cookie. Session data is user and session unique so can be applied uniquely to all users.

At the top of each page I want to set or retrieve a session variable I declare the following function which starts or resumes a session:

session_start();

A session variable can be set such as:

$_SESSION["username"] = $username;

And retrieved like so:

$_SESSION["username"]

Sessions are used throughout my application server-side to dynamically display data based on session values. Uses include:

- Storing username and welcoming user
- Storing and displaying visualisation data
- Adding/removing page content based on value of session variable
- Restricting access to main content is session variable is null
- Displaying result of quiz
- Parsing query result string to assign session values

2.3.4 Text Analysation Tool

2.3.4.1 Overview

The function of the text analysation tool is to read the contents of a .txt file and determine whether each line contains potentially offensive/inappropriate language. This is done using a combination of data sets that contain bad words, second person pronouns, third person pronouns and good words. If a line is found to be inappropriate/offensive it is communicated to the user.

2.3.4.2 Data Used

The four datasets used in the program are taken from a paper by L. Engman which describes the construction of a piece of software that uses natural language processing and machine learning in the attempt to identify inappropriate comments on social media. (Engman, 2016)

Dataset descriptions

Bad words

This dataset contains words labelled as bad words taken from various websites and compiled to create a single dataset.

Good words

This dataset is compiled by Mind Map Inspiration who identified a list of positive words. (MindMapInspiration, 2016)

Second Person Pronouns

A list of second person pronouns such as you and your

Third Person Pronouns

A list of third person pronouns such as he, she, him and her

2.3.4.3 Operation

The first operation of the program is to create four lists for the above datasets and then add the data to their respective list on initialisation. The list datatype was chosen for the flexibility over ArrayLists as you don’t have to specify a set size.

Next, the Graphical User Interface will load and present to the user. The user must click ‘select’ and a file dialog will appear and filter files in a chosen directory by their type, .txt. in this case. The user will select the .txt file they wish to use and click ‘ok’. The ability to analyse will now be accepted by enabling the ‘analyse’ button. The user clicks this button to begin the analyse operation and internal operations are as follows:

Analyse Operations	
Clear and refresh results grid	
Create and initialise count variable to 1 to number lines	
Create an initialise 4 bools to keep track of words (initialised to false, 1 for each dataset)	
Open the stream-reader to read the file and store contents in string	
Split the resulting string into lines by ‘.’ Or ‘!’ encountered	
For each line of file:	Dataset bools reset to false
	For each second person pronoun, set corresponding bool to true if in line
	For each third person pronoun, set corresponding bool to true if in line
	For each bad word, set corresponding bool to true if in line
	For each good word, set corresponding bool to true if in line

	If line contains bad word and second person pronoun or third person pronoun and doesn't contain a good word then append line content, number and notice to results grid
	Count variable +1
Close stream-reader	

Difficulties and Limitations

This tool is a step in the right direction and has led to successful detection in early tests but there are certainly difficulties and limitations that the program has and it is important to recognise these:

Spelling Mistakes

As this tool uses defined datasets of correctly spelled words, misspelled words that can still lead to offence cannot be detected without adding a substantial amount of misspelled words to the data. Even that poses new challenges.

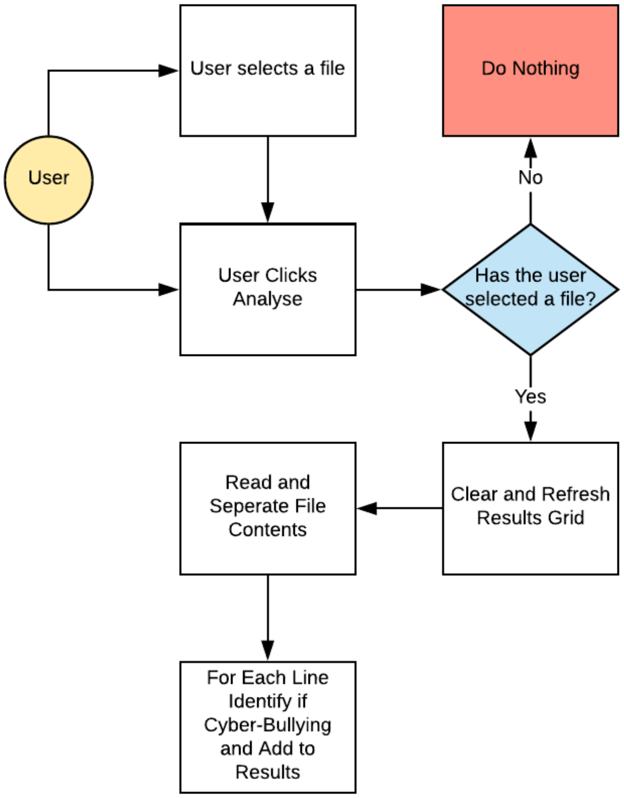
Sarcasm

Sarcasm is a natural part of human conversation and is widely used on the internet. Sarcasm is the use of irony or mocking something or someone which generally isn't offensive but the terms that can be used to portray sarcasm can be detected with our datasets leading to inaccuracies.

Cyber Bullying or Just Bad Words?

This is an interesting question and poses further challenges to the accurate detection of cyber bullying. Bad words are often used by individuals on social media, forums etc. but these words are simply part of a person's vocabulary and would not be deemed offensive.

2.3.4.4 System Flow Diagram


User selects a file
Do Nothing
User
No
User Clicks Analyse
Has the user selected a file?
Yes
Read and Seperate File Contents
Clear and Refresh Results Grid
For Each Line Identify if Cyber-Bullying and Add to Results

2.3.5 Data Visualisation

2.3.5.1 Overview

The data visualised within my application is drawn from the responses to the questionnaire offered to registered users. Users who answer these questions have their answers saved anonymously to the 'visualdata' table in the database. This data is retrieved to view on various charts throughout multiple pages in the application. The aim is to learn more about the users' view and experience with cyber bullying and is completely optional.

2.3.5.2 Implementation

Gathering Data

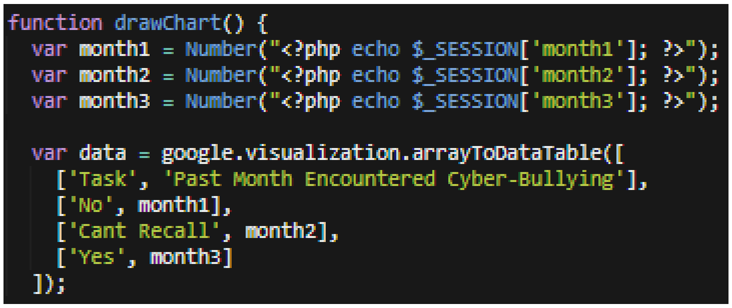
Data is gathered through the questionnaire offered to the user once they have logged in. The questions and accepted responses are as follows:

Question	Responses
Have you seen what you believe was Cyber-Bullying in the past month?	No, Can't Recall, Yes
Where do you believe Cyber-Bullying is most prevalent?	Social Media, Gaming, SMS/Email, Other
Should there be more done to combat Cyber-Bullying and penalise those at fault?	No, Yes

Populating Charts

Once data from the questionnaire is submitted from the client-side HTML form, it is passed to the database via the PHP file 'visualdata.php'. This file performs validation to ensure all questions have been answered and then performs two queries on the database. The first query performed updates the 'users' table and sets the 'questionnnairetaken' variable to '1', meaning the user cannot take the questionnaire again. The second query inserts the data received into the 'visualdata' table to record their answers.

The charts used to display the data are implemented using Google Charts which supplies interactive charts to web applications on desktop and mobile. The charts are added via JavaScript in the 'index' and 'maincontent' pages. Data is added to the charts by echoing PHP session variables containing data from the 'visualdata' table into the JavaScript Google Chart method:



```
function drawChart() {
  var month1 = Number("<?php echo $_SESSION['month1']; ?>");
  var month2 = Number("<?php echo $_SESSION['month2']; ?>");
  var month3 = Number("<?php echo $_SESSION['month3']; ?>");

  var data = google.visualization.arrayToDataTable([
    ['Task', 'Past Month Encountered Cyber-Bullying'],
    ['No', month1],
    ['Cant Recall', month2],
    ['Yes', month3]
  ]);
```

2.3.6 Educational Challenge

2.3.6.1 Overview

The educational challenge is designed to be an engaging method of learning about various facts surrounding cyber-bullying. The concept of the challenge is a user gets 30 seconds to view these facts until they disappear and they must recall the correct answers in a quiz like, multiple choice question format. The challenge can be repeated at any time and isn't saved to the database.

2.3.6.2 Implementation

Data Used

The data was retrieved from the website 'Do Something' who compiled a list of interesting facts:

Fact Questions (Do Something, 2018)
What % of students witnessed bullying online?
What is the primary medium for cyber bullying?
How many victims report cyber bullying to a parent or trusted adult?
Who is more likely to be bullied online?
How many adults say they have been bullied online?
What platform sees the most cyber bullying?
What is the most common action taken by victims of cyber bullying?

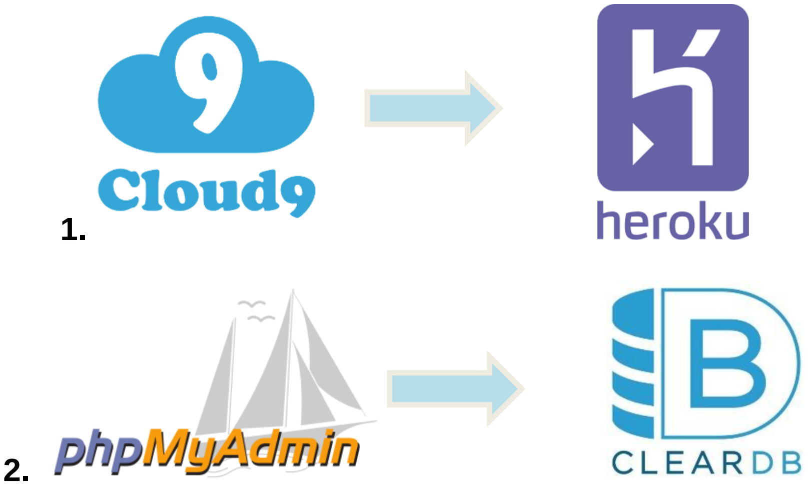
Operations Performed

The data is presented and submitted via a HTML form on the 'maincontent' page. The action method of the form directs answers to the questions to the 'quiz.php' file where they are extracted and saved in a local PHP variable. This file also contains the answers to the questions. The method of marking is simply performed by checking if the answer received equals the expected answer and if so, increasing a counter variable which represents the mark attained. This mark is saved in a PHP session variable and is echoed into the 'maincontent' page for the duration of the session or until the quiz is taken again.

The time constraint is imposed via jQuery's 'setInterval()' method which updated the clock every second. This is contained within the 'onClick()' method of the challenge start button which also controls hiding/showing content when appropriate.

2.3.7 Deployment

2.3.7.1 Strategy



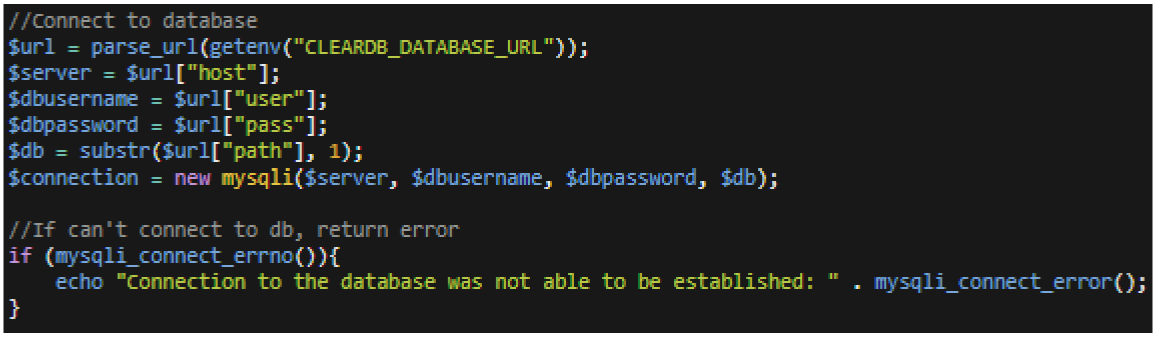
The strategy for development was to move from Cloud9 and deploy on Heroku. Heroku was chosen due to previous experience and integration with git repositories and Cloud9 was well supported. The database management system used in production was 'phpMyAdmin' using SQL. Heroku natively uses Postgres so I had to find an alternative to be able to use an SQL database, which was ClearDB.

2.3.7.2 Production to Development

Steps Taken

1. *Database*

First, I had to change connection elements as they were structured to connect to a phpMyAdmin database. These changes were made in all PHP files which made a connection to the database:



```php
//Connect to database
$url = parse_url(getenv("CLEARDB_DATABASE_URL"));
$server = $url["host"];
$dbusername = $url["user"];
$dbpassword = $url["pass"];
$db = substr($url["path"], 1);
$connection = new mysqli($server, $dbusername, $dbpassword, $db);

//If can't connect to db, return error
if (mysqli_connect_errno()){
    echo "Connection to the database was not able to be established: " . mysqli_connect_error();
}
```

2. *Additional Files*

Additional files were needed to enable the deployment of my web application on Heroku:

Procfile

The Procfile is needed to declare what type of application you are going to be deploying. The contents needed in my Procfile were:

```
web: vendor/bin/heroku-php-apache2
```

This tells Heroku I'm deploying a web application and to use an apache server.

Composer.json

The composer tells Heroku what version a certain language should use and locks it to that version.

The contents of my composer were:

```
"require": {
    "php": "^5.5.9"
}
```

This instructs Heroku to use PHP version 5.5.9 or greater.

3. *Renaming vendor*

Due to vendor being a reserved word in Heroku I had to change the name of this folder and all references to it to ‘v’.

4. *Committing final changes*

When all changes were made the next step was to commit all changes to my git repository using the following commands:

```
Git add .

Git commit -m “final commit”

Git push origin master
```

5. *Creating a Heroku App*

Creating a Heroku app was done from the command line in Cloud9. ‘Heroku create’ creates an empty application ready for use.

6. *Deploying Code*

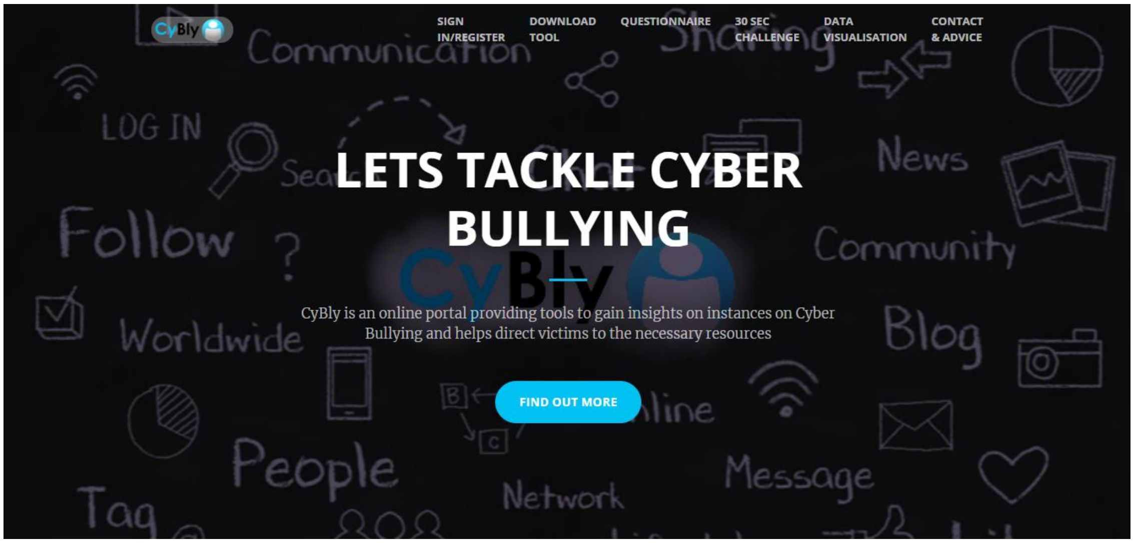
To deploy my code from my git repository to the new Heroku app, the command ‘git push Heroku master’ was run which pushes all code to the Heroku app and is then live.

To make changes to my deployment I must modify my code in the development environment and repeat steps 4 and 6.

2.4 Graphical User Interface (GUI) Layout

Web Interfaces

Landing page

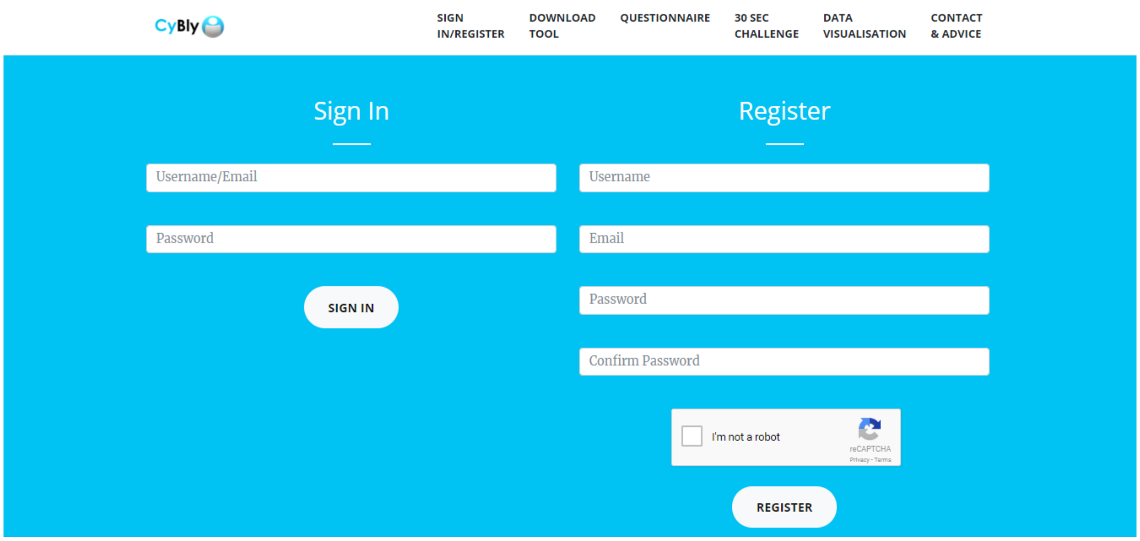


LETS TACKLE CYBER BULLYING

CyBly is an online portal providing tools to gain insights on instances on Cyber Bullying and helps direct victims to the necessary resources

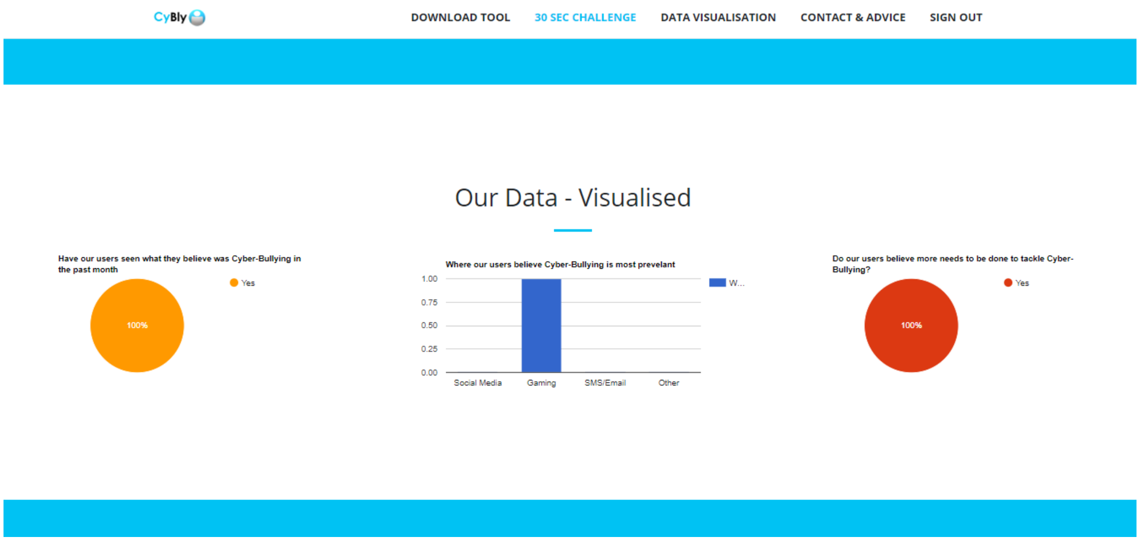
Above is the first page that will greet a user upon loading. The logo is in the top left corner as expected of most applications. Navigation is handled by the navigation bar that is stuck to the top of the page. On click of a navigation element, the page will smoothly scroll to the relevant section on the webpage. On smaller devices, this navigation compresses into a hamburger menu that expands on click.

Log In and Register



The sign in and register forms are presented when the user presses the 'sign in/register' button. The div handling this forms is hidden by default until this button is pressed. The jQuery method 'fadeIn('slow')' is used to fade this div in slowly when requested. The form fields are given decoration through the bootstrap ‘form-control’ class.

Data Visualisation



Our Data - Visualised

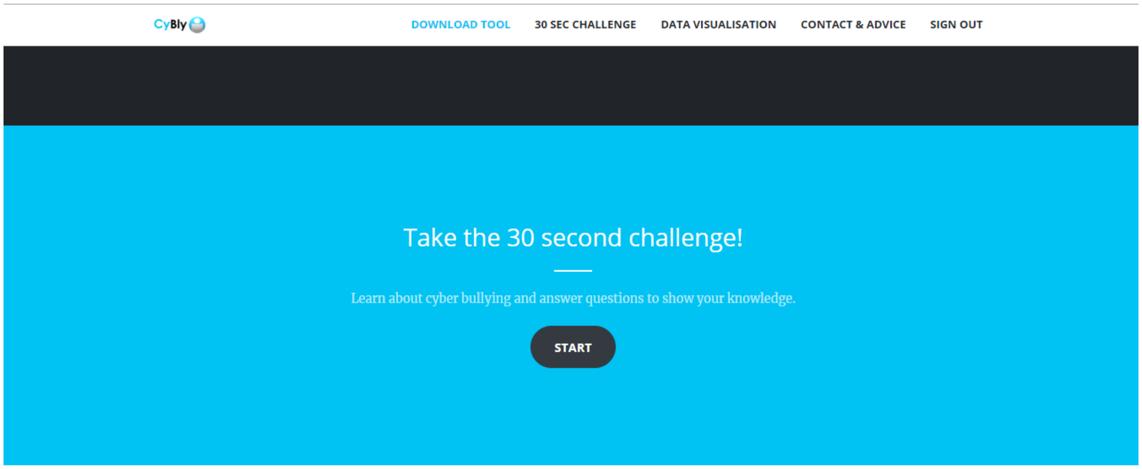
Have our users seen what they believe was Cyber-Bullying in the past month

Where our users believe Cyber-Bullying is most prevelant

Do our users believe more needs to be done to tackle Cyber-Bullying?

This data is present at the bottom of the ‘index’ and ‘maincontent’ pages and is a visual representation of data stored in the ‘visualdata’ table. This data is visualised through the use of Google Charts.

30 Second Educational Challenge - Start

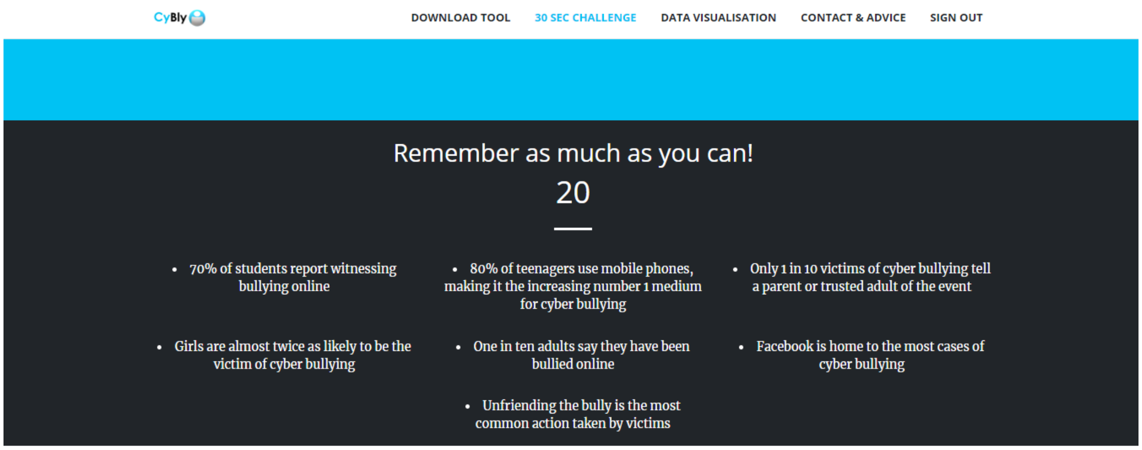


Take the 30 second challenge!

Learn about cyber bullying and answer questions to show your knowledge.

On click of the start button, the following cyber bullying facts are presented.

Challenge - Facts



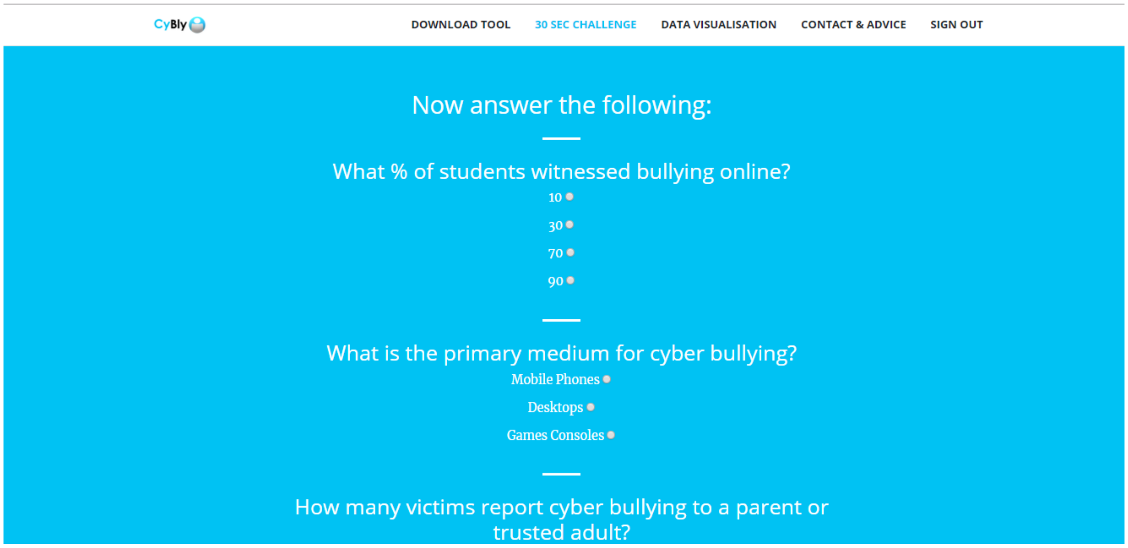
Remember as much as you can!

20

- 70% of students report witnessing bullying online
- 80% of teenagers use mobile phones, making it the increasing number 1 medium for cyber bullying
- Only 1 in 10 victims of cyber bullying tell a parent or trusted adult of the event
- Girls are almost twice as likely to be the victim of cyber bullying
- One in ten adults say they have been bullied online
- Facebook is home to the most cases of cyber bullying
- Unfriending the bully is the most common action taken by victims

This div which is originally hidden is presented with the jQuery 'fadeIn('slow')' method. The timer for viewing these facts is 30 seconds, controlled by the jQuery 'setInterval' method which updates the time left every second and presents the div.

Challenge - Questions



Now answer the following:

What % of students witnessed bullying online?

10 ○
30 ○
70 ○
90 ○

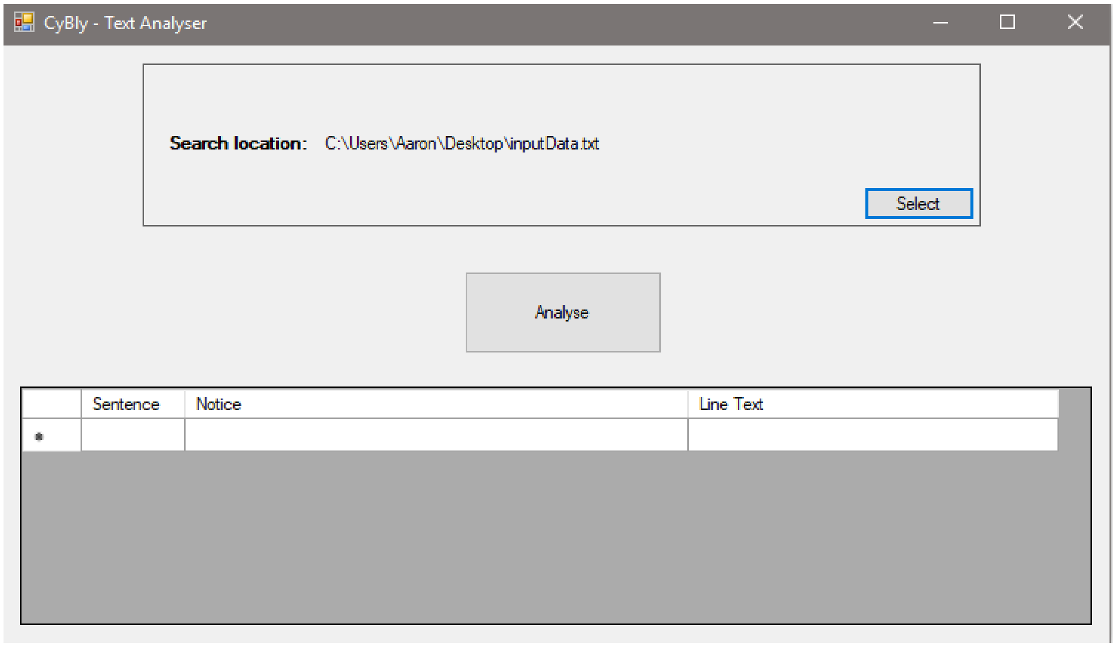
What is the primary medium for cyber bullying?

Mobile Phones ○
Desktops ○
Games Consoles ○

How many victims report cyber bullying to a parent or trusted adult?

Text Analyser

Program UI



Search location: C:\Users\Aaron\Desktop\inputData.txt

The panel with the label ‘Search location:’ contains the file and its location which it will analyse. The ‘select’ button enables the file dialog to choose a new file. The analyse button will initiate analysation of the text and update the results grid below as appropriate.

2.5 Testing and Evaluation

As my application makes use of multiple programming environments and languages, I had to choose different testing techniques to apply to each. Testing and evaluation methods were also chosen depending on what would deliver the most valuable impact.

Project Component	Test Applied
Text Analyser (Visual Studio)	Unit Testing
Web Application (Cloud9)	Usability Testing, Trunk Test, Five Second Test

2.5.1 Unit Tests

The purpose of unit testing is to test individual methods/functions of an application to ensure it meets certain criteria outlined in the test. For this testing you need to understand each element of the method to implement the test correctly.

Unit testing was carried out in Visual Studio for the text analysation tool. The core analysation method was tested to ensure that the actual result matched the expected result of a run. A description of the test along with the result and code snippet can be seen below:

Description	Result
Running the analyse method on a test sentence should ensure that the sentence is identified and result Boolean is equal to the expected Boolean	Pass

Code Snippet

```
[TestMethod()]
public void analyseBtn_ClickTest()
{

    bool containsBWord;
    bool containsGWord;
    bool containsSPP;
    bool containsTPP;

    //Testing variables
    bool actual = false;
    bool expected = true;

    //Add to badWord dataset
    //Add to secondPersonPronoun dataset
    //Add to thirdPersonPronoun dataset
    //Add to goodWords dataset

    sentenceArray.Add("i hate you");

    //initialise booleans to false
    containsBWord = false;
    containsGWord = false;
    containsSPP = false;
    containsTPP = false;
    actual = false;
    expected = true;

    //Read each line for analysis
    foreach (var line in sentenceArray)
    {
        //Comparisons
```

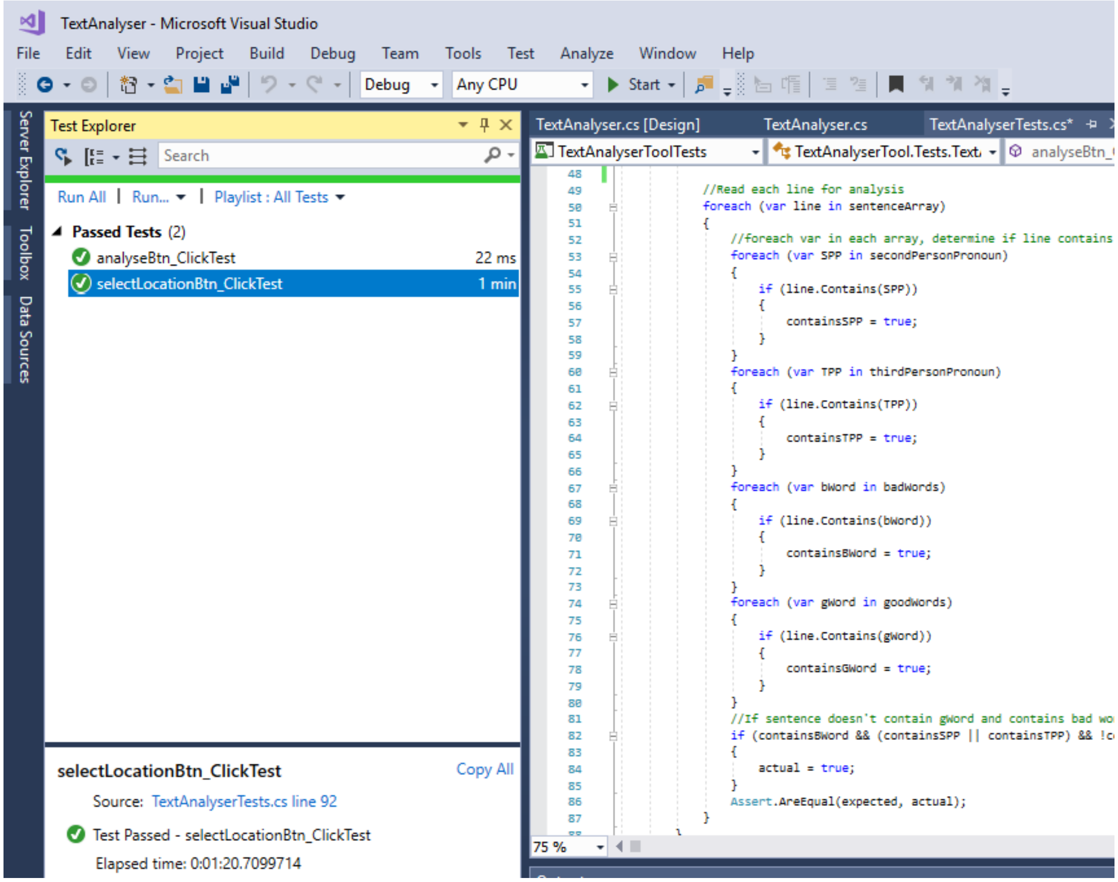
```
                //If sentence doesn't contain gWord and contains bad word and SPP or TPP then append to results gridview
                if (containsBWord && (containsSPP || containsTPP) && !containsGWord)
                {
                    actual = true;
                }
                Assert.AreEqual(expected, actual);
            }
        }
```

Description	Result
Selection of a file should assign the string value to the label on the form i.e. filename and label string should be equal	Pass

Code Snippet

```
[TestMethod()]
        public void selectLocationBtn_ClickTest()
        {
            OpenFileDialog openFileDialog1 = new OpenFileDialog();

            //Options for file dialog incl: file types aloud, title of window
            openFileDialog1.Filter = "Text Files|*.txt";
            openFileDialog1.Title = "Select a file to analyse";

            string locationToSearch = "";
            string locationNameLbl = "";

            //Opening file dialog and allowing selection of text file for analysis
            //Change text of form label to text file path name
            if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
            {
                locationToSearch = openFileDialog1.FileName;
                locationNameLbl = locationToSearch;
            }

            Assert.AreEqual(locationNameLbl, locationToSearch);
        }
```

Unit Test Results in the IDE


TextAnalyser - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug
Any CPU
Start
Server Explorer
Toolbox
Data Sources
Test Explorer
Search
Run All | Run... | Playlist : All Tests
Copy All
TextAnalyser.cs [Design]
TextAnalyser.cs
TextAnalyserTests.cs*
TextAnalyserToolTests
TextAnalyserTool.Tests.Text
analyseBtn_
75 %

Passed Tests (2)

- ✓ analyseBtn_ClickTest 22 ms
- ✓ selectLocationBtn_ClickTest 1 min

selectLocationBtn_ClickTest

Source: TextAnalyserTests.cs line 92

✓ Test Passed - selectLocationBtn_ClickTest

Elapsed time: 0:01:20.7099714

```
            //Read each line for analysis
            foreach (var line in sentenceArray)
            {
                //foreach var in each array, determine if line contains
                foreach (var SPP in secondPersonPronoun)
                {
                    if (line.Contains(SPP))
                    {
                        containsSPP = true;
                    }
                }
                foreach (var TPP in thirdPersonPronoun)
                {
                    if (line.Contains(TPP))
                    {
                        containsTPP = true;
                    }
                }
                foreach (var bWord in badWords)
                {
                    if (line.Contains(bWord))
                    {
                        containsBWord = true;
                    }
                }
                foreach (var gWord in goodWords)
                {
                    if (line.Contains(gWord))
                    {
                        containsGWord = true;
                    }
                }
                //If sentence doesn't contain gWord and contains bad wo
                if (containsBWord && (containsSPP || containsTPP) && !c
                {
                    actual = true;
                }
                Assert.AreEqual(expected, actual);
            }
        }
```

2.5.2 Customer testing

2.5.2.1 Five Second Test

This test has a very simple concept yet can be very effective in terms of design and layout. The test involves letting a participant view a given page in the application for 5 seconds. Straight after this they are asked to write down their immediate thoughts. This can help to determine eye grabbing features and how we're being perceived by a user. Five participants from my workplace volunteered to participate in this test.

Page Tested: Home page (index.php)

Response 1:

The big text in the middle shows me that it's about cyber-bullying. I see a button where I can find out more. Overall it looks well designed and a nice first page.

Response 2:

The key message is displayed in the middle of the page so I couldn't miss it. I then saw some things to click in the navigation bar, I think I saw the option to sign in so you must be able to make an account.

Response 3:

First, I noticed that lots of spacing was around the elements which makes things easier to read. This website deals with cyber-bullying so it may have a goal to help people.

Response 4:

I saw an image with a lot of social media related icons and text on it and a big message about cyber-bullying. The logo was in the top left as expected by most websites. CyBly is an online portal to learn more about cyber-bullying I think it said.

Response 5:

First thing I like was the fonts and the large text. I like the colour scheme implemented here and the call to action buttons that are clearly labelled. I saw the website offers a few different services up at the top but I can't recall what they were exactly.

Takeaway

I am pleased with the overall responses received to the five second test of the index page. When designing the home page there were elements which I wanted to draw the users' attention to including the large topical text, action buttons, navigational elements. These were all mentioned in the above responses which is positive to see and allows me to deem this test as passed.

2.5.2.2 Usability Testing

Usability testing is the process of getting real end users to perform actions on a piece of software. The purpose for this test is to gauge how users are receiving the application, how easy is it to use, is navigation fluid and easy to understand etc. The same five participants who did the five second test from my workplace volunteered to test my application and answer some short questions on their experience with various tasks and navigation etc. Below are the responses received.

Task A: Download the text analysation tool and perform a run on a piece of text.

Task B: Answer questions on the questionnaire and submit

Task C: Take the 30 second fact challenge

Response 1

How was the registration process? (Choose One)
1. Very Easy
How was the overall layout to navigate? (Choose One)
1. Very Easy
Was task 'A' easy to perform? (Choose One)
2. Easy
Did you find task 'A' effective? (Choose One)
3. Yes
Was task 'B' easy to perform? (Choose One)
1. Very Easy
Did you find task 'B' effective? (Choose One)
3. Yes
Was task 'C' easy to perform? (Choose One)
1. Very Easy
Did you find task 'C' effective? (Choose One)
3. Yes

How was navigating to different pages throughout the website? (Choose One)
1. Very Easy
Would you like to list any elements/features the you feel were missing? (Optional)
Did you find this site to be useful? (Choose One)
3. Yes
If no, why? (Optional)
Were there any frustrations you encountered? (Choose One)
1. No
If yes, what were they? (Optional)
Anything else that you would like us to know? (Optional)
Love the concept and the website is insightful

Response 2

How was the registration process? (Choose One)
1. Very Easy
How was the overall layout to navigate? (Choose One)
1. Very Easy
Was task ‘A’ easy to perform? (Choose One)
1. Very Easy

Did you find task ‘A’ effective? (Choose One)
3. Yes
Was task ‘B’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘B’ effective? (Choose One)
3. Yes
Was task ‘C’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘C’ effective? (Choose One)
3. Yes
How was navigating to different pages throughout the website? (Choose One)
1. Very Easy
Would you like to list any elements/features the you feel were missing? (Optional)
Did you find this site to be useful? (Choose One)
3. Yes
If no, why? (Optional)
Were there any frustrations you encountered? (Choose One)
1. No
If yes, what were they? (Optional)

Anything else that you would like us to know? (Optional)

Response 3

How was the registration process? (Choose One)
2. Easy
How was the overall layout to navigate? (Choose One)
1. Very Easy
Was task 'A' easy to perform? (Choose One)
2. Easy
Did you find task 'A' effective? (Choose One)
3. Yes
Was task 'B' easy to perform? (Choose One)
1. Very Easy
Did you find task 'B' effective? (Choose One)
3. Yes
Was task 'C' easy to perform? (Choose One)
2. Easy
Did you find task 'C' effective? (Choose One)
3. Yes
How was navigating to different pages throughout the website? (Choose One)
1. Very Easy

Would you like to list any elements/features the you feel were missing? (Optional)
Did you find this site to be useful? (Choose One)
3. Yes
If no, why? (Optional)
Were there any frustrations you encountered? (Choose One)
1. No
If yes, what were they? (Optional)
Anything else that you would like us to know? (Optional)
Found the text tool interesting

Response 4

How was the registration process? (Choose One)
1. Very Easy
How was the overall layout to navigate? (Choose One)
1. Very Easy
Was task ‘A’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘A’ effective? (Choose One)
2. Moderately
Was task ‘B’ easy to perform? (Choose One)
1. Very Easy

Did you find task ‘B’ effective? (Choose One)
3. Yes
Was task ‘C’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘C’ effective? (Choose One)
3. Yes
How was navigating to different pages throughout the website? (Choose One)
1. Very Easy
Would you like to list any elements/features the you feel were missing? (Optional)
Did you find this site to be useful? (Choose One)
3. Yes
If no, why? (Optional)
Were there any frustrations you encountered? (Choose One)
1. No
If yes, what were they? (Optional)
Anything else that you would like us to know? (Optional)

Response 5

How was the registration process? (Choose One)
2. Easy
How was the overall layout to navigate? (Choose One)
2. Easy
Was task ‘A’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘A’ effective? (Choose One)
3. Yes
Was task ‘B’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘B’ effective? (Choose One)
3. Yes
Was task ‘C’ easy to perform? (Choose One)
1. Very Easy
Did you find task ‘C’ effective? (Choose One)
3. Yes
How was navigating to different pages throughout the website? (Choose One)
1. Very Easy
Would you like to list any elements/features the you feel were missing? (Optional)
Did you find this site to be useful? (Choose One)
Yes

If no, why? (Optional)
Were there any frustrations you encountered? (Choose One)
1. No
If yes, what were they? (Optional)
Anything else that you would like us to know? (Optional)

I am very happy with the results of the usability test. All participants exhibited ease when navigating around the application and performing the outlined tasks. This test assured me that the application is very usable and incorporates good design practices and guided tasks.

2.5.3 User Acceptance Testing (UAT)

User acceptance testing test that the performance of the application is up to par with the users' expectations. For my user acceptance testing I got a small group of friends and family to use my application for the first time. The tests below were performed multiple times to verify the results. I am happy to have completed this method of testing as feedback was positive and helped me make small changes even though all tests passed.

Test Case Description	Testing functionality of login/register and functions of application		
Test Case Result	Pass		
Step Details	**Expected Results**	**Actual Results**	**Pass/Fail**
Navigate to the website URL	Landing page should load and display to user	As expected	Pass
Click sign in/register	Present sign in/register form	As expected	Pass

Register	If username not taken and valid, email not taken and valid, password meets criteria, confirm password correct and captcha complete then pass, otherwise fail	As expected	Pass
Main page loads	When registered, signed in, main page should load, welcoming user	As expected	Pass
Sign out	When sign out clicked, should sign out user	As expected	Pass
Sign in	Valid email/username and password that are registered entered, main page loads	As expected	Pass
Download text tool	Download should begin	As expected	Pass
Perform a run on text tool	Can select a .txt file, click analyse, results populated	As expected	Pass
30 second challenge	Facts appear for 30 seconds then disappear. Questions appear and able to submit and view results	As expected	Pass
Take questionnaire	Questions appear, answers can be chosen. Submitted answers appear on charts	As expected	Pass
Click advice link	Loads external URL for advice	As expected	Pass

3 Conclusions

This project uses various technologies and methods to provide a web application that focusses on cyber bullying and feels complete. Navigation is simple and intuitive and the design is modern and engaging. The implementation of tools such as the text analyser and ability to visualise user submitted data gives the interested user extra resources for their needs.

The development of this application which used multiple development environments was an exciting task to undertake. The creation of a very complete feeling web application is very satisfying and certainly a goal I am proud of. The implementation of a tool created in Visual Studio added to the complexity and allowed me to showcase my skills in another framework and language.

I feel like this is a fitting way to wrap up my studies and now showcase my skills in the workplace which I hope to add value and new ideas to.

4 Further Development or Research

This application has multiple avenues open to further development. For one, the text analysation tool could be extended to include machine learning and natural language processing to improve accuracy of results by greater understanding the contents of the text.

Other developments could include offering the user a profile page to manage their data or perhaps enable the user to make a public profile and interact with other members of the service through a topical forum etc.

As the number of users grows, data on the application will become more accurate. Another development to be made could be the internationalisation of the project, allowing the application to detect the users' country and translate the page automatically.

5 References

Do Something. (2018, May 01). *11 Facts About Cyber Bullying.* Retrieved from Do Something: https://www.dosomething.org/us/facts/11-facts-about-cyber-bullying

Engman, L. (2016, June 8). *Automatic Detection of Cyberbullying on Social Media.* Retrieved from Umea University: http://www8.cs.umu.se/education/examina/Rapporter/LoveEngmanReport.pdf

MindMapInspiration. (2016, January 25). *A to Z Positive Words.* Retrieved from MindMapInspiration: http://www.mindmapinspiration.co.uk/a-to-z-positive-words/4530136070

6 Appendix

6.1 Appendix A: Project Proposal

6.1.1 Objectives

My objective for this project is to create an online educational platform on the subject of online bullying, more commonly known as cyber-bullying. This will be achieved through the implementation of several strategies:

- Gamification
    - Quizzes and interactive games to test the understanding on the subject while improving user knowledge.
- Visualization
    - Through analyzation of responses to quizzes, results will be aggregated by type to represent an overall picture. As more users use the application the visualization will provide a more accurate picture.
- Online Resources / Where to turn for help
    - References to other resources on the web, made easily accessible in a central repository.
- Text Analyzation Tool
    - A downloadable program that can be used to analyse any text (e.g. Tweets/Posts) and returns whether there is content with the potential to cause offence.

6.1.2 Background

Idea

My project idea is to develop a cyber bullying information portal with elements including: gamification, data visualization, help guides, resource centre and a text analyzation tool. The rapid growth of users connecting to the internet via Social Media and Gaming among others means many children/teens are being exposed to open platforms which, due to their scale are hard to regulate. This opens the door to cyber bullying, which is more prevalent now than ever and can have serious effects on teens and even younger. My project will aim to provide information on how to get help, how to enhance knowledge on the subject and even detect instances of it.

Goals of Project

- Educate users on the topic via information, all categorizations of cyber bullying
- Provide instructions on what to do in certain instances of cyber bullying
- Educate the user through gamification. Interactive quizzes (identify cyber bullying, reinforces key topics)
- Provide insightful visualisations
- Provide a tool to detect cyber bullying

Technological Objectives

- To be developed as a web application and Android application.
- Responsive Design
- Sign up/Log in

Technical Approach

1. Capture requirements
    a. Functional
    b. Non-functional
2. Mock-up of UI
3. Implement Web framework core HTML, CSS
4. Design and Implement database(Firebase)
5. (Android)Design and create app.
6. (Android)Link Firebase database.
7. Design quizzes and integrate
8. Dynamic content and Client Side Scripts with JavaScript
9. Visualize data with integrated Google Charts

10. Mock-up + Map out functionality of text analyser
11. Develop and integrate
12. Testing

6.1.3 Technical Details

Web Environment set-up

Initial step is to create a responsive web environment for my project using the following technologies:

- HTML
- CSS
- Bootstrap

Dynamic elements such as animations and transitions will be completed using JavaScript.

Database

- Will be hosted by firebase as this can support cross platform applications
- Tables to be designed to hold data on users, quizzes, feedback

Google Charts

- Used for the visualization of findings through quizzes.
- Google Charts library will be used.

Text analyzation tool

- Developed on the Visual Studio IDE with the C# language.
- Windows Form Application.
- Accepts input of text file.
- ArrayList Data Storage for comparisons of data.
- Comparisons use list of common words from cyber bullying.
- Output is the lines of text where occurrence was found and possibly more insights.

Android Studio – Java

- Implement the same functionality as the web application.
- Connect to the same database via Firebase.

6.1.4 Evaluation

Unit testing

Unit tests will be carried out on an ongoing basis throughout the development of the project to ensure each component of software performs optimally.

Unit tests will follow this testing pattern:

- Ensure all prerequisites are in place
- Trigger the unit being tested
- Use a method to verify results.

Integration tests

After verifying all units of a functionality are performing correctly, an integration test will be carried out on that functionality and verified.

Test Cases

Test cases for each use case defined in the requirements document and core functionality

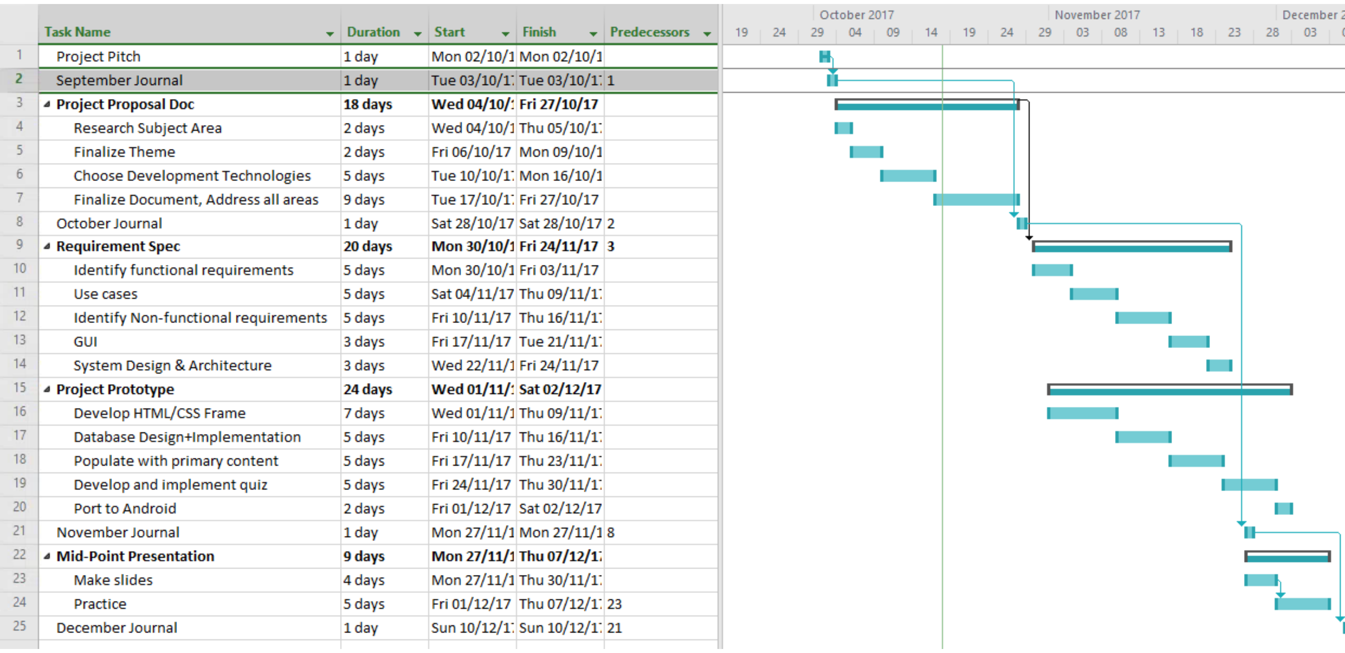
User Acceptance

A user acceptance review will be implemented as a feedback system on the website and will ask user to rate all functionalities. From this performance can be assessed and issues addressed.

6.2 Appendix B: Project Plan

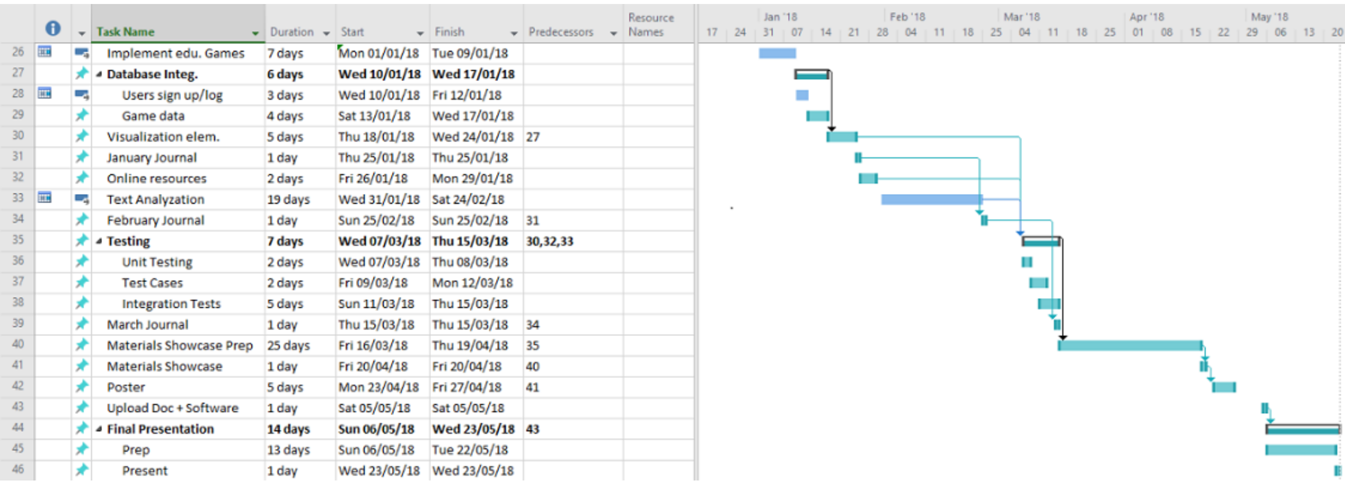
The following are Gantt charts with project goals and deadlines until the end of the college year:

Semester 1



	Task Name	Duration	Start	Finish	Predecessors
1	Project Pitch	1 day	Mon 02/10/1	Mon 02/10/1	
2	September Journal	1 day	Tue 03/10/1	Tue 03/10/1	1
3	**Project Proposal Doc**	**18 days**	**Wed 04/10/**	**Fri 27/10/17**	
4	Research Subject Area	2 days	Wed 04/10/1	Thu 05/10/1	
5	Finalize Theme	2 days	Fri 06/10/17	Mon 09/10/1	
6	Choose Development Technologies	5 days	Tue 10/10/1	Mon 16/10/1	
7	Finalize Document, Address all areas	9 days	Tue 17/10/1	Fri 27/10/17	
8	October Journal	1 day	Sat 28/10/17	Sat 28/10/17	2
9	**Requirement Spec**	**20 days**	**Mon 30/10/1**	**Fri 24/11/17**	**3**
10	Identify functional requirements	5 days	Mon 30/10/1	Fri 03/11/17	
11	Use cases	5 days	Sat 04/11/17	Thu 09/11/1	
12	Identify Non-functional requirements	5 days	Fri 10/11/17	Thu 16/11/1	
13	GUI	3 days	Fri 17/11/17	Tue 21/11/1	
14	System Design & Architecture	3 days	Wed 22/11/1	Fri 24/11/17	
15	**Project Prototype**	**24 days**	**Wed 01/11/**	**Sat 02/12/17**	
16	Develop HTML/CSS Frame	7 days	Wed 01/11/1	Thu 09/11/1	
17	Database Design+Implementation	5 days	Fri 10/11/17	Thu 16/11/1	
18	Populate with primary content	5 days	Fri 17/11/17	Thu 23/11/1	
19	Develop and implement quiz	5 days	Fri 24/11/17	Thu 30/11/1	
20	Port to Android	2 days	Fri 01/12/17	Sat 02/12/17	
21	November Journal	1 day	Mon 27/11/1	Mon 27/11/1	8
22	**Mid-Point Presentation**	**9 days**	**Mon 27/11/1**	**Thu 07/12/1**	
23	Make slides	4 days	Mon 27/11/1	Thu 30/11/1	
24	Practice	5 days	Fri 01/12/17	Thu 07/12/1	23
25	December Journal	1 day	Sun 10/12/1	Sun 10/12/1	21

Semester 2



	Task Name	Duration	Start	Finish	Predecessors	Resource Names
26	Implement edu. Games	7 days	Mon 01/01/18	Tue 09/01/18		
27	**Database Integ.**	**6 days**	**Wed 10/01/18**	**Wed 17/01/18**		
28	Users sign up/log	3 days	Wed 10/01/18	Fri 12/01/18		
29	Game data	4 days	Sat 13/01/18	Wed 17/01/18		
30	Visualization elem.	5 days	Thu 18/01/18	Wed 24/01/18	27	
31	January Journal	1 day	Thu 25/01/18	Thu 25/01/18		
32	Online resources	2 days	Fri 26/01/18	Mon 29/01/18		
33	Text Analyzation	19 days	Wed 31/01/18	Sat 24/02/18		
34	February Journal	1 day	Sun 25/02/18	Sun 25/02/18	31	
35	**Testing**	**7 days**	**Wed 07/03/18**	**Thu 15/03/18**	**30,32,33**	
36	Unit Testing	2 days	Wed 07/03/18	Thu 08/03/18		
37	Test Cases	2 days	Fri 09/03/18	Mon 12/03/18		
38	Integration Tests	5 days	Sun 11/03/18	Thu 15/03/18		
39	March Journal	1 day	Thu 15/03/18	Thu 15/03/18	34	
40	Materials Showcase Prep	25 days	Fri 16/03/18	Thu 19/04/18	35	
41	Materials Showcase	1 day	Fri 20/04/18	Fri 20/04/18	40	
42	Poster	5 days	Mon 23/04/18	Fri 27/04/18	41	
43	Upload Doc + Software	1 day	Sat 05/05/18	Sat 05/05/18		
44	**Final Presentation**	**14 days**	**Sun 06/05/18**	**Wed 23/05/18**	**43**	
45	Prep	13 days	Sun 06/05/18	Tue 22/05/18		
46	Present	1 day	Wed 23/05/18	Wed 23/05/18		

6.3 Appendix C: Monthly Journals

Reflective Journal

Student name: Aaron Fitzgerald
Programme: BSc in Computing software development
Month: October

My Achievements

This month, I was able to …

- Complete the first draft of my project proposal

- Plan for future goals with a Gantt chart

My contributions to the projects included …

- Researching technologies and related materials to help structure the proposal draft.

My Reflection

I felt, it worked well to ….

- Understand more about cyber bullying to help develop strategies for my project

However, I was not successful in …

- Fully completing the proposal as there are certain details which still must be decided on
- Adding a greater level of complexity to the project

Intended Changes

Next month, I will try to …

- Identify a more complex element to include into my project e.g. data analysis or some AI functionality

I realised that I need to …

- Complete all sections of the proposal draft
- Prepare for the requirements document

Supervisor Meetings

Date of Meeting: 26/10/17

Items discussed:

- Approving project idea
- Need to add level of complexity

Action Items:

- Research word recognition API to recognize bullying

Reflective Journal

Student name: Aaron Fitzgerald

Programme: BSc in Computing software development

Month: November

My Achievements

This month, I was able to …

Add to my proposal, making it close to the final version as I could think of more features I want to implement. I also started to have a look at the requirements document which is due next so full steam ahead with that.

My contributions to the projects included …

Finishing the project proposal. Adding a text analyzation tool to identify cyber bullying to my project plan. Added data visualization to my project plan for quiz results etc. Also decided on a name – CyBly.

My Reflection

I felt, it worked well to ….

- Find time to complete the proposal
- Come up with new ideas to add
- Come up with a project name

However, I was not successful in …

- Getting a start on the requirements

Intended Changes

Next month, I will try to …

- Have a prototype ready along with the requirements

I realised that I need to …

- Get going on the requirements

Supervisor Meetings

Date of Meeting: 26/10/17

Items discussed:

- Ideas to complete proposal

Action Items:

- Research text analyzation tool and google charts

Reflective Journal

Student name: Aaron Fitzgerald

Programme: BSc in Computing software development

Month: December

My Achievements

This month, I was able to …

Complete the requirements document and is now ready for upload. Prepare for the midpoint presentation.

My contributions to the projects included …

Put together a basic html mock-up that is operational client side.

My Reflection

I felt, it worked well to ….

- Find time to complete the requirements document
- Get a mock-up developed

However, I was not successful in …

- Looking deeper into possible server-side configurations

Intended Changes

Next month, I will try to …

- Follow requirements to guide me in the right path

I realised that I need to …

- Look up tutorials and acquire knowledge

Supervisor Meetings

Date of Meeting: No meeting; supervisor on leave

Items discussed:

- N/A

Action Items:

- N/A

Reflective Journal

Student name: Aaron Fitzgerald

Programme: BSc in Computing software development

Month: January

My Achievements

This month, I was able to …

Start to get a look back on things. A few weeks off to refresh after a tough semester.

My contributions to the projects included …

Looking at PHP for server-side interaction

My Reflection

I felt, it worked well to ….

- Refresh but also realise there's not long left

However, I was not successful in …

- Getting major advancements in development

Intended Changes

Next month, I will try to …

- Get more done in terms of development

I realised that I need to …

- Get moving to achieve my goals for the project

Supervisor Meetings

Date of Meeting: No meeting; supervisor on leave

Items discussed:

- N/A

Action Items:

- N/A

Reflective Journal

Student name: Aaron Fitzgerald

Programme: BSc in Computing software development

Month: February

My Achievements

This month, I was able to …

Make progress on the development

My contributions to the projects included …

Implementing DB architecture. Getting a login and registration to interact with it also.

My Reflection

I felt, it worked well to ….

- Break up my work load so I can manage the tasks

However, I was not successful in …

- Implementing more functionality yet

Intended Changes

Next month, I will try to …

- Get the text analyser tool underway

I realised that I need to …

- Talk to supervisor

Supervisor Meetings

Date of Meeting: Feb 2018

Items discussed:

- Project progress and direction

Action Items:

- Keep moving on development.
- Discussed ideas on where development can go and tools to use
- Breaking down structure of report

Reflective Journal

Student name: Aaron Fitzgerald
Programme: BSc in Computing software development
Month: March

My Achievements

This month, I was able to …

Get some logic down for analysing text.
Add showcase profile to college system.

My contributions to the projects included …

Getting the core idea for text analyser down and securing 3% from the profile.

My Reflection

I felt, it worked well to ….

- Break up my work load so I can manage the tasks, focus on the core idea

However, I was not successful in …

- Getting a working prototype of the tool

Intended Changes

Next month, I will try to …

- Have the text tool finished

I realised that I need to …

- Wrap up development in C# soon to focus on web

Supervisor Meetings

Date of Meeting: Mar 2018
Items discussed:

- How things are shaping up
- Key steps for production

Action Items:

- Advised to work as much as possible

Reflective Journal

Student name: Aaron Fitzgerald
Programme: BSc in Computing software development
Month: April

My Achievements

This month, I was able to …

Complete text tool and other web components

My contributions to the projects included …

Adding security to sign up. Creating a quiz for users to take. Was able to complete text tool.

My Reflection

I felt, it worked well to ….

- Revise requirements to drive results

However, I was not successful in …

- Getting all components ready

Intended Changes

Next month, I will try to …

- Have all code completed and get the report up to a good standard

I realised that I need to …

- Deploy soon

Supervisor Meetings

Date of Meeting:
Items discussed:
Action Items: