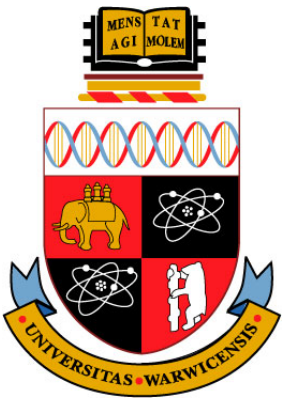

Fidelis: Trust-based social networking for user engagement and protection

Project Report

by

Isheneasu Gambe, Thomas Mcaloone,
Jordan Olney and Naqash Tanzeel

Supervisor: Dr Matthew Leeke

Department of Computer Science

University of Warwick

2016–17

Abstract

Dedicated to...

Acknowledgements

Declarations

Sponsorship and Grants

The research presented in this dissertation was made possible by the support of the following benefactors and sources:

- The Department of Computer Science, University of Warwick, Coventry, United Kingdom - Some Award (2004-2012)

Abbreviations

ACB	Apple Banana Carrot

Notations

r_2	Has some meaning

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Social media has been a rapidly growing industry in the 21st century, with the social network Facebook worth just under $315 billion dollars as of May 2016 despite only being founded in 2004 [1]. However, with the popularity of Facebook and other social networks such as Twitter and Instagram, numerous social issues have arisen which are yet to be fully addressed. Fidelis is an alternative to these networks, which attempts to address these issues. This specification discusses the requirements of the social network Fidelis, as well as the strategies which will be used to design, develop and then test the system.

1.1 Problem Statement

1.2 Project Aims

Abuse detection, provision of user-specific content and reputation scoring are three key features which can be implemented to ensure that a social network is trustworthy. Therefore, the aim of this project is to build a new social network called Fidelis, which will demonstrate the methods of implementing these features and to evaluate how effective these methods are in solving the issues which established social networks currently face in their attempt to earn and retain the trust of their users.

1.2.1 Abuse Detection

As discussed in section 1.1, abuse has played a large role in users losing trust in social networks. Therefore, Fidelis should be able to detect abusive content, which can then be hidden from the user. This will allow Fidelis to provide a platform for open discussion and debate without the threat of abuse and intimidation. However, what is perceived as abusive is dependent on the individual user, therefore how strict the detection algorithm is at identifying abuse must be flexible. In addition, traditional methods of handling abuse such as blocking users and reporting posts must still be implemented so that no abuse goes undetected. On the other hand, the user should also have the option to view content which has been flagged as abuse by the algorithm, so that the network does not enforce censorship on content which is posted. An exception is when the content posted is deemed illegal and therefore should be removed from the platform completely.

1.2.2 Content Recommendation and Filtering

Content which is provided to each user by a social network should be engaging. Therefore, Fidelis should be able to recommend content which is known to interest the user. This does not necessarily mean that the posts/users provided agree with a user's point of view, because it is important the platform is able to stimulate debate so that it is representative of the real world. In order for the system to ascertain which topics a particular user is interested in, it should be able to detect the topics of the user's posts as well as the topics of posts which the user interacts with. In addition to recommending content, the user should be able to easily access new content. Therefore, Fidelis must provide feeds which are filtered based on their topics and provide post tagging, which allows users to explore posts addressing more specific topics.

1.2.3 Reputation Scoring

By determining the reputation of users, Fidelis should be able to determine the reliability of content and then tailor how visible it is on the platform, so that the user is provided with high quality content from trustworthy sources. This requires a score to be calculated for each user, which represents how reputable they are and therefore how worthwhile it would be to follow them. This score can then be used to determine which users should be recommended to others. The features which determine the user reputation are:

- Number of followers. If a user has a large number of followers, they would have a higher reputation score because it suggests that their content is worth following.
- Number of votes. A user which receives a large number of votes posts more engaging content and is therefore more reputable.
- Number of positive votes vs. number of negative votes. A user which posts more positively received content is more likely to be reputable.
- Number of comments on their posts. As with the votes, the number of comments on a post can be used to determine how engaging the content is. Therefore users which receive more responses to their content have a higher reputation.

As well as scoring the reputation of each user, Fidelis should score the reputation of each post. This is because a reputable user may not always post content of the same quality. Therefore, the following features of a post are used to determine its reputation score:

- Number of votes.
- Number of positive votes vs. number of negative votes.
- Number of comments.

1.3 Project Motivation

As mentioned previously, the popular social networks of today have given rise to numerous social issues. One of these such issues is 'trolling', with 'The Guardian' reporting that "one in four teenagers suffered hate incidents online last year" [2].

The social networks' current failure to find an effective solution to these pitfalls of social media therefore offers the question of whether a network could be created, which tailors the content users see based on what that particular user would prefer to see. This does not only include filtering any abusive posts, but also not offering them content which is not of interest to them, either based on the theme of the content or who posted it.

1.4 Project Stakeholders

The internal stakeholders for this project are the development team - Isheanesu Gambe, Naqash Tanzeel, Thomas Mcaloone and Jordan Olney - and Dr Matthew Leeke, who is the project supervisor. To ensure the project is successful, the wider public will also be involved to offer feedback on the progress of Fidelis throughout the duration of the project. In order to test the system, data from other social networks will be required to populate the database with realistic posts. Therefore, the Twitter accounts of multiple stakeholders, both external and internal, will be used, so that large quantities of data can be collected in a short period of time, without exceeding the rate limits imposed by the Twitter API.

1.5 Report Structure

The purpose of this report is to provide a comprehensive account of the process undertaken whilst developing the system associated with the project. The report has been broke down into 3 main sections which identify the high level stages this project went through.

Research and Analysis An introduction to the problem being faced and combatted, motivations behind the undertaking of this project and an analysis of any stakeholders is presented in chapter 1. Chapter 2 discusses and analyses any existing solutions, along with any technologies that may be used throughout the project, listing their advantages. Chapter 3 briefly discusses any issues that may arise and must be considered, whilst chapter 4 outlines the original and final requirements for the system.

Development and Testing Chapter 5 discusses the thought process behind the designing of the system, whilst chapter 6 details the implementation of the system. Chapter 7 outlines the testing procedures that were carried out, prior to, throughout, and post development. Collectively chapters 5-7 cover the development process from start to finish.

Evaluation and Reflection The final 3 chapters reflect on the entire process of developing the system. Chapter 8 discusses the project management strategies employed to tackle the project whereas chapter 9 provides an analysis of the work carried out and how well it satisfies the initial requirements of the project. Finally, chapter 10 concludes with a summary and any suggestions for extending the system further.

CHAPTER 2

Research

2.1 Related Work

2.1.1 Abuse Detection

2.1.2 Content Filtering

In order to be able to show users content which is of interest to them, topic modelling will be required to both determine what a user's interests are and what posts correspond to those interests. However, due to colloquialism and limited length of social media posts, there can be a large amount of ambiguity in posts. Therefore, it is a challenge to be able to effectively categorise all posts, but there are measures which can be taken to improve the performance of the topic models.

Before considering the type of model to use, feature extraction must be used to represent posts in a way which allows a model to detect patterns. The most common method for textual content is Bag-of-Words (BoW), where "a document is considered to be simply a collection of the words which occur in it at least once. The order of the words, the combinations in which they occur, paragraph structuring, punctuation and of course the meanings of the words are all ignored" [**?**]. The loss of ordering in the content can result in some of the meaning of the post being lost. Despite this, "BoW features are effective at capturing the general topic of the discourse in which the topic had occurred" [**?**].

In order to combat this loss of information, n-grams may be used instead of or alongside BoW, which uses a consecutive sequence of n words as a feature.

The model will then need to be designed. Due to the nature of the problem, the model would be a classifier. This classifier would then categorise each post to its relevant topic. The table in Appendix B shows the categories used for Twitter posts, which could be adapted to be used in the Fidelis model. For example, Twitter has subdivided the 'Conversational' category into subcategories such as 'Query' and 'Referral', which may not be beneficial to the users of Fidelis. On the other hand, it would be important to break down the 'News' category into subcategories such as 'Sport' and 'Events' because this covers a broad spectrum of user interests.

In regards to the model itself, there are numerous models which perform classification, including support Naïve Bayes, vector machines (SVMs), k-nearest neighbours and latent Dirichlet allocation (LDA), with each model having their own advantages and disadvantages.

Naïve Bayes assumes that each feature in the data is independent in order to calculate the maximum posterior given the training data [**?**]. This is done maximising the Bayes formula, where c is the number classes and i is each possible model:

$$P(\omega_i|\mathbf{x}) = \frac{P(\omega_i)p(\mathbf{x}|\omega_i)}{\sum_{j=1}^{c} P(\omega_j)p(\mathbf{x}|\omega_j)} \quad i = 1, ..., c.$$

As we know features are independent, this substitution can be made to simplify the above equation, where n is the number of inputs:

$$p(\mathbf{x}|\omega_i) = \prod_{j=1}^{n} p(x_j|\omega_i) \quad i = 1, ..., c.$$

This simplification makes computation inexpensive, but the assumption may not necessarily hold with the posts on Fidelis. For example, the presence of one word in the BoW could increase the likelihood of another word being present, thus assuming independence may make the model less effective.

Conversely, SVMs uses a decision boundary to categorise the data. This "is chosen to be the one for which the margin is maximized" [**?**], where the margin is the distance between the boundary and the nearest data points from each class. One possible difficulty which arises with the use of SVMs is the choice of kernel, which is used to measure the similarity of two data points. This is because "once the kernel is fixed, SVM classifiers have only one user-chosen parameter (the error penalty), but the kernel is a very big rug under which to sweep parameters" [**?**].

K-nearest neighbours "directly uses the data sample to estimate the density" [**?**]. The performance of this model decreases as the dimensionality of the data increases, so due to the likely sparsity of the data which the model will be using in Fidelis, this could have a major implication on the effectiveness of the topic categorisation. Another drawback is that the use of k-nearest neighbours is "computationally expensive" [**?**].

2.1.3 Existing Systems

There are a range of existing system which provide a similar service or a subset of the services described in the project aims. However these solutions have all been designed to tackle different problem and all have their respective drawbacks. In order to avoid facing problems that have been solved by these services, an analysis is carried out to discuss the advantages and shortcomings of each service. Using this, a superior social network which implements the advantages of these services whilst tackling the disadvantages can be be designed.

2.1.4 Content Recommendation

2.1.5 Reputation Scoring

2.2 Existing System

2.2.1 Facebook

2.2.2 Twitter

2.2.3 Reddit

2.3 Technologies

When developing a web application, it is important to explore the range of technologies already available in order to speed up the development process. For this reason, several third party technologies were used to provide some of the core functionality of the system. These technologies include but are not limited to jQuery, Laravel and Google Maps. In addition to the third party technologies, the latest web development technologies were also employed to provide the essential and basic functionality of a web application.

2.3.1 Web Technologies

HTML

Hyper Text Markup Language (HTML) is a markup language used for structuring and presenting content on the world wide web [11]. HTML 5 is the recommended standard by W3 as of October 28, 2014 and as such will be used to structure the web application being developed. HTML in itself cannot be used to change the style of page as it was only designed for defining the structure of a page but fortunately HTML tags can be styled.

CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colours, spacing) to Web documents [12]. CSS3 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1. It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like multi-columns, flexible box or grid layouts [6]. CSS3 is being used in this project as it provides a much larger range of styles which are required by Bootstrap.

JavaScript

JavaScript (JS) is a high-level, lightweight, interpreted, programming language with first-class functions [7]. Although JavaScript is not a necessary requirement, it offers many advantages such as allowing manipulation of the document structure after the page has been loaded. As a result of this, one can add, remove or animate content on the page. Inarguably, the main advantage of JavaScript is that it is interpreted on the client side which means less processing is done on the server. This allows for faster loading times as parts of the document can be loaded later, on demand, when required. Collectively, all these and more features improve the user experience and provide a good reason to incorporate JavaScript into a web app.

PHP

PHP is...[10]

2.3.2 Frameworks

Model-View-Controller (MVC)

Initially, as stated in the project specification, the system was to be designed as a web based application implemented using the four basic web technologies mentioned previously in section 2.3.1. After completing the initial setup and some implementation, this approach was challenged and a decision was made to find an alternative approach. One of the main reasons for this change in approaches was the overwhelming amount of code that was duplicated across pages making it difficult to make changes whilst maintaining consistency across the pages. As a result, it was decided that a Model-View-Controller approach should be used and implemented using an existing framework.

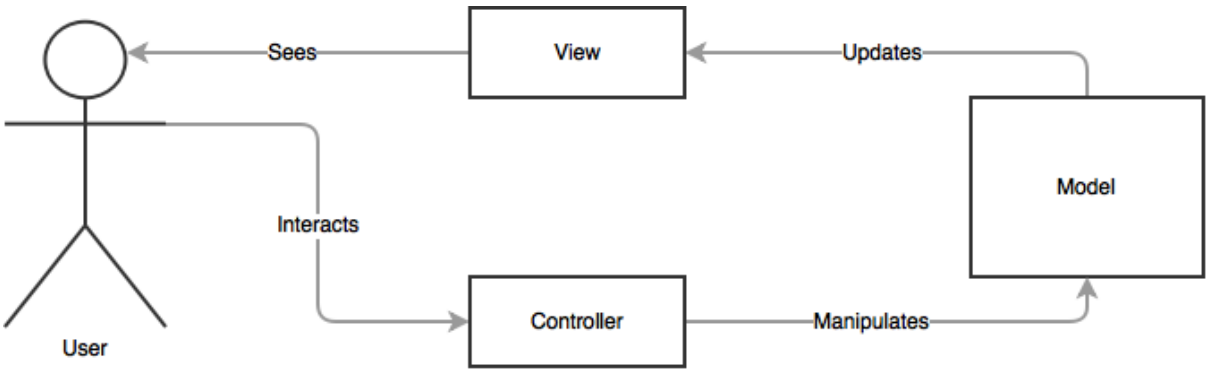

Figure 2.1: Information Exchange in an MVC Application.

The Model-View-Controller (MVC) pattern separates the modelling of the domain, the presentation, and the actions based on user input into three separate classes [5]. A description of each section as well a graphical representation of communication, figure 2.1, is given above.

- **Model.** The model manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller) [5]. In essence the model is responsible for all interaction with the database and contains any code that either queries or updates records in the database along with some output formatting.
- **View.** The view manages the display of information [5]. The advantage of this approach is that repeated content can be split into separate views with placeholders and then included by passing in parameters to fill the placeholders. This means that the developer only needs to change the code in one place, particularly useful for things such as navigation and foorter which are consistent across pages.
- **Controller.** The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate [5]. This is where majority of the logic and PHP code for the application is written. Any algorithms and data pre-processing or post-processing methods are written inside the controller.

Laravel

Laravel, an open-source PHP web application framework intended for the development of web applications following the MVC architectural pattern, developed by Taylor Otwell, was chosen after researching various frameworks [4]. Laravel was chosen over other frameworks for a number of reasons. Not only does it provide an implementation of the MVC pattern, but it also allows allows the user to define custom routes and decide where each route leads rather than having the URL be determined by the file path. Additionally, Laravel comes with solutions to a lot of common tasks and problems straight out of the box making it easier to develop the system straight away without having to “reinvent the wheel”.

jQuery

jQuery is a fast, small, and feature-rich JavaScript library [9]. It will be used over standard JavaScript as it makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers [9]. Although jQuery doesn’t necessarily provide any additional functionality over JavaScript as it is powered by JavaScript, it provides shorter notations for common JavaScript functions and provides implementation of features that are lacking in JavaScript, or take long to implement. It can be setup and used by simply including a single script in the document.

Bootstrap

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web [3]. The main advantage of using Bootstrap is that it easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries. Additionally Bootstrap comes with a whole range of HTML, CSS and jQuery components, that have been pre-styled, further speeding up the development process.

2.3.3 Storage

One of the most fundamental parts of the system is retaining the data that the users enter into the system. There are two different storage solutions for storing and querying the data provided by users. These are both discussed below.

MySQL

MySQL is an open source database management system used for managing data held in a relational database management system (RDBMS) [8]. An SQL database will be used to store all textual information entered by the users. This includes user information, location data, item details and any other necessary information. SQL databases are not encrypted and hence any user credentials such as password will be stored in an encrypted format to prevent access to user accounts in case of a breach.

Resources

Along with textual input, it is also necessary to store the images associated as photos with posts, uploaded by the users. One approach to this could be to convert the file to binary and store the binary data in the SQL database. However this approach would lead to the database rapidly growing in size and in turn affecting the performance. Not only would queries take longer to execute but also increase page load times due to the large record sizes. Another approach would be to store any uploaded files either on a storage server or on the web server so they are available on demand. This would result in faster loading time for pages containing these resources as they do not have to be transferred over the network. Due to this, the latter approach will be used during development.

CHAPTER 3

Legal, Social, Ethical and Professional Issues

3.1 Legal Issues

3.1.1 Sensitive Data

3.1.2 Licensing

3.1.3 Resources

3.2 Social Issues

3.3 Ethical Issues

3.4 Professional Issues

CHAPTER 4

System Requirements

4.1 Requirement Refinement

4.2 Functional Requirements

The functional requirements will guide the development of all aspects of the system. These requirements will ensure that the system provides the necessary functionality, the implementation details of which are entirely up to the developer and may change during the development process.

F1 The system must be able to communicate with a number of third party APIs in order to retrieve data.

- (a) The system must be able to convert all data received from third parties into a consistent format.

F2 Users must be able to register and log into the system.

- (a) Registration may be done through manually entering all the required details or by using a third party which provide oAuth 2.0 services, such as but not limited to Facebook and Twitter.
- (b) These credentials should be stored, in an encrypted format, so that they can later be used for authentication.
- (c) Users should be able to recover their account in the case that they have forgotten their password or their account has been compromised.
- (d) A user maintains the right to be able to delete their account publicly, however this data may still be retained privately.

F3 A user may make their account private, preventing other users from being able to see the content they share.

- (a) If a user attempts to follow a private account, a request to follow is sent to the private user and must be approved before the relationship is created.

F4 A user may block another user.

- (a) If the blocked user is already following the blocking user, then they automatically unfollow the user.

(b) Once a user has been blocked, the blocked user should not be able to find the blocking users profile through search or otherwise.

(c) If a user has blocked a user then they may also unblock the user. This will not restore any previous following relationships.

F5 In order to see content, users should be able to add other users to their trust circle.

(a) This is identified as a one way relationship in which user A follows user B implies that user A trust user B.

(b) Similarly, if users no longer wish to see content from specific users then they may unfollow a user they are already following.

F6 The system will overcome the cold-start problem, under which a user will not see anything on their timeline upon registration, by providing a set of predefined categories which can be used to explore.

(a) Users are able to access a customised 'Discover' feed by subscribing to categories, sub categories or topics.

(b) The 'Discover' feed will then be generated by pulling content from the various subscriptions allowing the user to explore topics based on their interest, discover new content as well as new users who they may follow.

F7 The system will provide a personal feed where the user is able to view content from the people they trust, prioritised by the reputation of the people they follow as well as the reputation of the content itself.

(a) The user may hide content from their personal feed which they find offensive, or just uninteresting. This should be used to learn and adapt the recommendation system by modifying the posters reputation.

F8 A user must be able to post new content onto the system.

(a) A post may include text, images or videos. Additionally an image or video post may also contain text.

(b) When posting content, users may optionally tag the post or leave it untagged.

(c) When the user tags a post with a popular topic, e.g. #Brexit, the post should automatically be assigned to the correct category, e.g. Politics in the aforementioned case, using a bucket of keywords per category.

(d) If a tag is mentioned and the post cannot be categorised automatically then the user may be prompted to assign one of the predefined categories so that the system can learn and adapt for the future. Additional processes may be used to verify that tags are correctly classified.

F9 Users can view any text, photo or video posts made by public accounts or by private accounts which they follow.

(a) Viewing media content should open up in a popup or similar so users can view the post and comment on it.

F10 Users must be able to reply to a post made by another user.

(a) If the user has made their account private then only users following the author of the post may comment on or see the post.

F11 Users may interact with content they come across by liking, disliking or sharing it. Each of these actions will impact the reputation of the content.

(a) The user should also be able to reverse their interaction, e.g. un-liking a liked post.

F12 The system will automatically calculate a reputation score for users which symbolises the trustworthiness of the user and the content they post.

(a) This score will be impacted by a range of factors. For example, a user's trustworthiness increases as their number of followers (people who trust the user) goes up. Similarly, this reputation can also decrease if the users content is disliked.

(b) This will be used, along with other factors such as mutual connections, to recommend other users the user may be interested in following.

F13 The system will also calculate a reputation for each user posted content to represent the quality of content.

(a) As other users react to the content by liking, disliking or sharing it, the reputation of the post will change which in turn impacts the author's reputation.

(b) Highly scored posts will be more likely to be recommended to other users on the 'Discover' or 'Categories' pages.

F14 Each user will be able to choose how the reputation is calculated and how the notion of trust is implemented.

(a) This feature will be available through the settings menu. Each algorithm will use different properties of a user and posts to compute the score.

(b) Additionally, the user will be able to change the sensitivity of the scoring system from mild to strong.

F15 The system will incorporate abuse detection to prevent things such as profanity, swearing and nudity amongst other things.

(a) Each user can adjust the sensitivity of their abuse detection system depending on which content they would like to hide.

(b) Due to the processing requirements, this may be implemented as a job which executes at regular interval to remove content.

F16 Users can report content which they find is inappropriate or offensive, such as profanity or nudity.

(a) The report along with the reported content is sent to a moderation system. This report must be picked up by a moderator who either dismisses the reported if the content adheres to the guidelines or the content is removed and the user's reputation is changed to reflect this.

4.3 Non-Functional Requirements

In order for the system to be successful and used, it is essential that it is available on demand, regardless of time and place. This means that the system must support a range of devices varying from desktop to small hand-held smartphones. The non-functional requirements for this project will not only help to ensure that the systems operates smoothly and provides a responsive user experience, but also ensure that the development of the system is up to a high standard.

NF1 *Compatibility*: The system must be cross browser compatible and support all devices with a modern browser.

(a) A responsive design and structure must be used to ensure the system is compatible with all devices.

(b) The following categories of devices must be well supported.
   i. Mobile
   ii. Tablet
   iii. Desktop

(c) Functionality should not be limited or restricted on any of the given devices but may be implemented differently

NF2 *Usability*: The system must be intuitive and user friendly.

(a) The system must be intuitive and easy to navigate. Users must be able to access pages without any guidance.

(b) The user experience across all pages must be consistent in terms of design and functionality.

(c) The user must never encounter errors, but if the system encounters an error then an appropriate output should be produced.

(d) Appropriate user feedback must be given when interacting with the system.

(e) The system must have an appropriate load time across all pages.
   i. Studies have shown that nearly half of the users consider abandoning a site if it takes longer than 3 seconds to load [**?**].

NF3 *Security*: The system and the data held must be secure.

(a) The users details and credentials, such as email and password must be appropriately secured.

(b) Any data added by the user or associated to the user must only be available to the user unless explicitly made available to the public.

(c) Users may not have access to personal information about other users, such as email address and location data.

(d) Any resources uploaded by the user must not be browsable by other users, unless made public by the user.

NF4 *Scalability*: The system must be scalable and respond well to growth.

(a) Data storage and processing decisions must be made whilst taking growth and expansion into consideration.

(b) Growth should not limit functionality and availability - the system must be able to cope with this.

(c) As storage and processing can be costly, these must be considered when developing the system so efficient use is made of both of these resources.

NF5 *Extensibility*: The system must be extensible and support further development.

(a) The system must be designed modularly, regardless of whether it may need changing or not.

(b) Design decisions must be made with extensibility and growth in mind and as a result, any component or part of the system must be easily replaceable with an upgrade.

(c) Each of these components must be independent and operate independently.

NF6 *Maintainability*: The system must be maintainable.

(a) Standard and good code practices should be adhered to whilst developing the system.

(b) A version control system should be used to make regular checkpoints.

(c) The system and any external resources or technologies should always be kept up-to-date to avoid any issues.

(d) Any hardware decision made must take maintainability into consideration.

NF7 *Readability*: The system must be well documented

(a) The codebase must be thoroughly documented using any standard commenting conventions and guidelines for the specific language.

(b) Any progress logs must provide a clear outline of the progress and changes made.

(c) Version control commits must detail the changes made - any addition or removal of features must be clearly stated.

(d) A clear outline of the system and its functionality must be provided.

4.4 Limitations and Constraints

CHAPTER 5

Design

5.1 System Architecture

5.2 Data Collection

5.3 Data Processing

5.3.1 Abuse Detection

5.3.2 Content Filtering

5.3.3 Content Recommendation

5.3.4 Reputation Scoring

5.4 Database

5.5 User Interface

5.6 Responsive Design

CHAPTER 6

Implementation

6.1 Technologies

6.1.1 Storage

6.1.2 Processing

6.1.3 Visualisation

6.2 Data Collection

6.3 Data Processing

6.3.1 Abuse Detection

6.3.2 Content Filtering

6.3.3 Content Recommendation

6.3.4 Reputation Scoring

6.4 Database

6.5 Authentication

6.6 User Interface

CHAPTER 7

Testing

7.1 Unit Testing

7.2 Integration Testing

7.3 System Testing

7.3.1 Validation Testing

7.3.2 Permission Testing

7.4 User Acceptance Testing

7.4.1 User Feedback

7.4.2 Testimonials

7.5 System Tuning and Assessment

CHAPTER 8

Project Management

8.1 Design Approach

8.2 Software Development Methodology

8.3 Project Timeline

8.4 Tools and Techniques

8.4.1 Development

8.4.2 Management

8.5 Risk Management

8.5.1 Developer

8.5.2 Hardware

8.5.3 Data

8.5.4 Third Party Services

CHAPTER 9

Evaluation

9.1 Requirements

9.1.1 Functional Requirements

9.1.2 Non-Functional Requirements

9.2 Legal, Social, Ethical and Professional Issues

9.2.1 Legal Issues

9.2.2 Social Issues

9.2.3 Ethical Issues

9.2.4 Professional Issues

9.3 Project Management

9.4 Author and Project

CHAPTER 10

Conclusion

10.1 Summary

10.2 Future Work

Bibliography

[1] Forbes. Facebook. http://www.forbes.com/companies/facebook/.

[2] A. Gani. Internet trolling: quarter of teenagers suffered online abuse last year. https://www.theguardian.com/uk-news/2016/feb/09/internet-trolling-teenagers-online-abuse-hate-cyberbullying, 2016.

[3] M. Otto. Bootstrap. http://getbootstrap.com, Nov 2105.

[4] T. Otwell. Laravel - the php framework for web artisans. https://laravel.com, Mar 2016.

[5] Microsoft Developer Network. Model-view-controller. https://msdn.microsoft.com/en-us/library/ff649643.aspx, Apr 2016.

[6] Mozilla Developer Network and individual contributors. Css3 - css — mdn. https://developer.mozilla.org/en/docs/Web/CSS/CSS3, Jan 2016.

[7] Mozilla Developer Network and individual contributors. Javascript — mdn. https://developer.mozilla.org/en-US/docs/Web/JavaScript, Apr 2016.

[8] Oracle Corporation. Mysql. http://www.mysql.com, Mar 2016.

[9] The jQuery Foundation. jquery - write less, do more. https://jquery.com, Jan 2013.

[10] The PHP Group. Php: Hypertext preprocessor. http://php.net, Dec 2015.

[11] W3C. Html5. https://www.w3.org/TR/html5/, Oct 2014.

[12] W3C. Cascading style sheets. https://www.w3.org/Style/CSS/, Mar 2016.