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

Project Specification

by

**Isheneasu Gambe, Thomas Mcaloone,
Jordan Olney and Naqash Tanzeel**

Supervisor: Dr Matthew Leeke

Department of Computer Science

University of Warwick

2016–17

Contents

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 [10] despite only being founded in 2004. 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 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" [11].

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.

1.2 Project Aims

The aim of this project is to create a social network called Fidelis, where the content any user sees is centred around "trust". In order to achieve this, there are three main features which need to be successfully implemented: user reputation, abuse detection and content filtering.

1.2.1 Abuse Detection

To combat the issue of abusive messages on social media, the system must be able to flag and remove or hide any messages which are abusive. This will allow Fidelis to create an environment which encourages open discussion without threat or intimidation. Users who regularly post abusive messages will have a reputation score which reflects this. However, it is not necessarily clear cut what constitutes abuse, so there will still be functions for manually removing posts, blocking users, as well as a means for users to select contacts for whom all their content is displayed, regardless of if it is detected as abusive by the system.

1.2.2 Topic Filtering and Recommendation

One of the main features of Fidelis is that it will aim to only show content to users which may be of interest of them. Therefore, it must be able to detect what the interests of each user are and then only display content on their feed which is relevant. In order to do this, the system must detect the topic of each post, via a combination of automatic topic modelling and also user selection to mitigate any error with the topic model.

1.2.3 Reputation Scoring

By calculating a score for the reputation of a user, it will allow the content users post to have varying levels of visibility. On top of these user scores, further reputation scores must be used for each post a user creates to determine whether the user is trustworthy with respect to that particular topic. The factors which could participate to these user reputation scores are:

- **User-based**
  - Amount of abuse posted by that user.

- Number of contacts that user has. The more contacts a user has suggests that user has a higher reputation.
- Reputation of that user's contacts. If the user's contacts are reputable, then it implies that the user is more likely to be reputable through association.
- Feedback on previous posts (via likes or the sentiment of comments on the post). If the response to previous posts is largely positive, then this suggests that the user has a reputation for posting higher quality content and thus has a higher reputation.

- **Topic-based**

  - Experience of the user of that topic (e.g. occupation). Someone posting about an area of expertise for them is more likely to post reliable content than someone who is not involved in that field directly.
  - Feedback on previous posts within that field. Positive feedback on previous posts suggests that the user has some reputable knowledge within the field.
  - User's contacts within that area. If the user associates with people with an expertise of a topic, then they are more likely to post reputable content.
  - Whether the post has made use of external sources, because the use of external sources suggests that the opinion being made in the post is based on some sort of wider reading, and hence suggests some knowledge of the topic.

An added complication to the issue of reputation would be that each user's reputation would be dependent on the user viewing their content. This is because reputation is not objective and therefore the score would have to be adjusted depending on the user for whom the content is being chosen.

1.3 Stakeholders

The main stakeholders of the project are the development team and the team supervisor, who offers advice and support to the team throughout the course of the project. In addition, there will be input from the wider public, to gauge the popularity of Fidelis during development and during testing of the social network. Current existing social networks will also be involved, with data publicly available being used to simulate content on Fidelis.

1.4 Report Structure

Section 2 of this report will look into related work that has already been carried out and could be employed for the purpose of this project. As part of this, research has been done around existing algorithms and existing solutions. Section 3 covers the legal, ethical, social, and professional issues that may have to be overcome as part of the development of the project. The requirements, both functional and non-functional, along with the constraints and foreseeable challenges are thoroughly detailed in section 4. Section 5 aims to address the initial design decisions that have been made but are subject to change. Section 6 of this report will outline the project management approach which includes a project timeline, the roles and responsibilities of each team member and any tools or techniques that will be employed to make project management easier. Testing and evaluation strategies are listed and explained in section 7. Finally, section 8 provides a conclusion to summarise the contents of the report.

2 Related Work

Before commencing development of the system, it was important to explore research that has been done related to the project and consider existing systems that are similar to Fidelis. This chapter will discuss research that has been conducted in abuse detection, recommendation algorithms, trust scoring and topic filtering.

2.1 Research

Research for Project Fidelis focused on the four core system functionalities: abuse detection, recommendation algorithms, topic filtering and providing users with a trust/reputation score. A number of papers were researched to explore different techniques that will allow the group to implement the required functionality. The research conducted to date is not final, and will be on-going as the project progresses.

2.1.1 Abuse Detection

Most abuse detection systems make use of natural language processing and machine learning techniques to detect abusive user content. These systems learn from a set of abusive posts, and use the resulting model to classify future posts as abusive or not. The way in which these systems vary is the features extracted from posts to determine them being abusive or not. Based on the selected features, the detection models vary in their effectiveness of detecting abusive content. Abuse detection will be used by the Fidelis to identify abusive or hurtful user content. The detection of such content will allow the scores of the users posting such content to reflect this.

Very recently, a team from Yahoo built a supervised classifier for use as an abuse detection system [22]. The classifier made use of natural language processing (NLP) features which measure different aspects of a user's comments. Namely, the features used by the classifier are n-grams, linguistic, syntactic and distributional semantic features. Through the combination of these features, the resulting classifier is able to detect hate speech, derogatory and profane language.

Work by Ravazi et al aimed to build an automatic flame detection system [25]. A message is considered to be flame if its main intention is attack (hate speech, racism, extremism, homophobia) or it contains abusive or hostile words, phrases or language [25]. The system is built upon a three-level classifier. Each level extracts a set of features which are then used as the input for the next level. The first level uses the Complement Naïve Bayes classifier to select the most discriminative features. The second level uses the Multinomial Update Naïve Bayes Classifier. The third and final level uses DTNB, which is a rule-based classifier. The third level also makes use of an Insulting and Abusing Language Dictionary, which is a collection of words, phrases, and expressions, with different degrees of manifestation of flame varieties.

2.1.2 Content Recommendation

The rise in popularity of different media platforms and social networks such as Netflix, Facebook and Twitter has led to these companies rushing to find different ways through which they can engage and grow their user base. One technique that has become prevalent is the use of content-based recommendation for users. These so called recommender systems provide item recommendations that might be of use to a user [27]. We see this through Netflix's movie and TV show recommendations, or Facebook's People you may know feature. Recommendation techniques will be implemented within Fidelis to allow users to engage with content and other network users who not only share similar opinions, but also content and users that hold a view opposing that of the user. Exposing users to both sides of the coin will ensure that they remain f,

Most recommender systems implement either item or collaborative-based filtering algorithms. Under collaborative-based filtering algorithms, users are recommended new content based on the opinion of other like-minded users [28]. Collaborative filtering (CF) algorithms provide new item recommendations based on a user's previous likes or those of similar-minded users. Item recommendations take the form either of predictions, which are numerical values that represent how likely it is for a user to like an

item they have not previously liked, or recommendations, that are a list of N items a user will like the most [28]. Such algorithms can be memory-based, building a network of neighbours around a user who have a history of agreeing with the user [28], or model-based, whereby a machine learning approach is taken and item recommendations are viewed probabilistically.

Alternatively, an item-based filtering algorithm can be used for item recommendations. Item-based algorithms provide recommendations to a user based on item similarities. Items that a user has previously liked are scored against unseen items, and a user is presented with a list of the top N recommendations [16]. The key component for these recommendations is how the item-to-item similarity is calculated. Similarity can be determined using either cosine or conditional probability-based calculations. Cosine-based similarity works by treating items as vectors and calculating the cosine measure between item vectors. With conditional probability-based similarity, two items are similar if the probability that a new item will be consumed given another item already consumed by the user is high [16]. The conditional probability is calculated based on the frequency of an item, where frequency for an item X is the number of users that have consumed X.

2.1.3 Reputation Scoring

Trust is a very important aspect across all social networks. Different networks choose to model user trust in different ways, with some networks choosing to follow an explicit trust model (if User A trusts User B, B also trusts A), and others preferring an implicit model (User A can trust User B, but trust is not reflexive). Trust on Project Fidelis will be based on user reputation scores. Having these scores attached to each user on the network will allow users to make informed decisions on who they choose to engage with.

One interesting way to explore the idea of reputation scoring is through the use of collaboration-based reputation. In their paper, McNally et al explore the idea of modelling user reputations through collaboration events [19]. Collaboration events are events that occur when users provide feedback on the content posted by other users. This feedback, for example in the form of voting on a post, is used in a calculation to determine how much the user giving feedback trusts the original poster. From these collaboration events, a collaboration graph can be created where nodes in the graph correspond to the original poster and users who provided feedback on the post. Trust scores are represented as edges in the graph, and are aggregated to calculate the reputation of individual users using Weighted-Sum and Page Rank scoring metrics [19].

Zacharia et al explore the application of collaborative reputation mechanisms used in electronic markets for wider use. Two mechanisms for scoring user reputations are discussed which can handle both loosely and highly connected online communities. The first, Sporas, handles scoring reputations in a loosely connected community [33]. Under Sporas, users have a single reputation which is updated according to feedback provided by other users. New users start with the minimum reputation value, and as a user's reputation grows they experience smaller changes in their reputation after each update. The second mechanism, Histos, is a pairwise rating system [33]. These pairwise ratings between users are modelled through a directed graph, with nodes in the graph representing users and weighted edges between nodes representing the most recent reputation given by one user to another, with the arrow pointing towards the rated user [33].

2.1.4 Topic 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" [4]. 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" [15].

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 [17]. 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" [2], 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" [5].

K-nearest neighbours "directly uses the data sample to estimate the density" [34]. 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" [18].

2.2 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.

2.2.1 Facebook

"Founded in 2004, Facebook's mission is to give people the power to share and make the world more open and connected" [9]. Since its launch, it is the most popular social network to date and boasts an active user base of approximately 1.7 billion people [29]. According to Facebook, people use the social network "to stay connected with friends and family, to discover what's going on in the world, and to share and express what matters to them". Facebook users must register before using the social network and are free to create a personal profile in order to interact with other users which they can add as friends. Furthermore, Facebook users may join user groups based on workplace, college or school and can also categorise their Facebook contacts into lists. Users can post status updates or other content and message eachother [29].

The Facebook friend system operates on a mutual trust basis. This means if person A and person B are friends on Facebook because person A wishes to see content from Person B then person B implicitly trusts person B and also sees content from person B. However, Facebook later realised that mutual trust is not always implicit of a friendship and added a following and unfollowing feature. As of this update, if two users are friends then they are automatically following each other but they can go and optionally unfollow one another, meaning that they will still have a friendship but will not see content from one another. This adds the concept of a one way trust relationship which Fidelis aims to implement on an explicit level which the user has control over and an implicit level which filters content automatically through the use of reputation.

2.2.2 Twitter

Twitter is an online social networking and microblogging service, which allows registered users to read and post 'tweets' [31]. Twitter posts are limited to a short 140 characters and can either be posted on the users timeline or sent directly to users as messages. If a user chooses to make their account public then anyone can view their timeline and browse their tweets. Twitter is one of the most popular social networks worldwide with 313 million monthly active users [29, 32]. Part of the appeal is the ability of users to follow any other user with a private or verified public profile, enabling users to interact with celebrities or friends who regularly post on the social media site.

Unlike the Facebook friend system, Twitter operates purely on a following system. This means if user A wishes to follow user B, given user B has a public profile, user A will see all of the content shared by user B but the same will not apply vice versa until user B also chooses to follow user A. The success of the system lies in this core functionality which allowed popular figures to create accounts that could be follows by the mass public, enabling a one way stream of communication. Twitter also introduced another popular feature, the concept of adding 'hashtags' to tweets, which allows tweets to be categorised into popular topics. This can then be used to increase activity by recommending trending topics to users. However, one thing twitter does not implement well is abuse detection and spam filtering which results in masses of 'trolling' on public profiles and even bulling. Fidelis will inherit the core functionality of Twitter but extend it to include abuse detection and content filtering so that users are not thrown huge amounts of content at them but instead receive very personalised and appealing content.

2.2.3 Reddit

Reddit is a forum based social network that 'bridges communities and individuals with ideas, the latest digital trends, and breaking news' [26]. It works by allowing users to create boards and sub-boards known as sub-reddits. Once a board has been created, users can start threads which anyone may read but only registered users may post replies to. The system was designed to allow users to ask questions where a large community can leave answers. A user may reply to a reply or a reply of a reply. The social network was launched in 2005 and, according to statistics from August 2016, has approximately 243 millions unique monthly visitors [30].

Like most social networks, Reddit also has a way of rating and categorising posts however, unlike the others, reddit uses a up and down voting system which allows users to explicitly dislike content and change

the quality score. Categorising in Reddit is done through sub-reddits, meaning users must explicitly go and post in the appropriate rather than automatically categorising the content using hashtags. The advantage of this approach is that categories are maintained by users and should always be up to date. Content inside these categories is then ordered by the overall score (number of up votes - number of down votes). A similar approach to this will be employed when scoring content on Fidelis along with some other scores contributing to the order of posts on a feed.

3 Legal, Ethical, Social and Professional Issues

With any system that involves human interaction, there are always a number of legal, ethical, social and professional issues that can arise. Due to the scope of this project and the amount of interaction by humans that may be involved, it is critical to be aware of the possible issues that could occur and find a way to address them.

3.1 Legal Issues

One particular legal pitfall to consider is local vs. international law. Social networks often have users posting from many different countries all over the world. A social media platform may adhere to all pertinent local laws in the country it was created in, however complications may arise due to unforeseen and possibly conflicting laws if the platform is operating in another country. With a vast number of legal jurisdictions, it can be difficult to keep up with international law, however there are some practical things that can be done to maintain a stronger legal position. For example, serving only select countries by limiting the selection of language options, or even by restricting the IP addresses of visitors to only allow users from specific countries [14].

Since the proposed system focuses on users creating and sharing content, it is difficult to control what users may be posting. The platform is intended to promote an environment for free speech and sharing ideas/opinions, however there may be cases where a user publicly shares some information which was intended to be private or confidential. This could be personal information, or information pertaining to a third party such as an employer. Once information has been shared in this way there is little recourse; as soon as it is public, anyone can view it. In such a case one may argue that since the information was posted and shared through the Fidelis platform, the liability is with the platform and not with the user. It is therefore important to make it clear through policies and procedures that users are responsible for any content that they post or share through this platform. In a similar way, the use of unauthorised third party trademarks or copyright-protected works may be used by an individual in a way that infringes on the trademark/copyright. Users should therefore also be made aware that they are liable for any such infringement and that their posts may be removed due to this.

Just as third-party materials posted to a social media site may infringe copyright or trademarks, posting photographs and video without proper releases may violate the privacy or publicity rights of individuals. Additionally, within certain industries, employees must ensure that they do not violate specific privacy regulations in their activities on social media sites. This phenomena should, however, be combatted by the combination of trust/reputation-based content filtering and the ability for users to report content.

A common trend across many social media platforms is targeted 'abuse' towards a specific individual. This can be in the form of 'cyber bullying', 'trolling' or in some extreme cases, defamation. Defamation is defined as "A false statement or fact, not made under privilege, that is communicated to a third person and that causes damage to a persons reputation. For public figures, the plaintiff must also prove actual malice." [20]. Fidelis is focused on filtering the content that is presented to a user, which is intended to reduce the amount of negative content the user will see. While very negative content will be hidden from many users, the posts will still persist on the platform and so any content considered defamatory is still subject to defamation law. It should therefore be possible to delete these posts if needed as well as possibly using the posts to find the IP address or some other information about the original poster

(i.e. to aid authorities whilst following both the law and any policies regarding privacy or anonymity of the user).

3.2   Ethical Issues

As a social media platform, there is a level of ethical responsibility to protect users by ensuring an agreed upon level of privacy. For example it is fairly common for company HR departments to review the social media pages of both job candidates and current employees. While this practice may be of use to the company, the design of the Fidelis platform is such that users should feel comfortable posting and discussing content without feeling the need to censor themselves. The user may not wish for everybody to see their posts and so as a compromise the user will be able to make their account private, preventing non-approved users from seeing the content they share.

This system aims to provide a platform for users to post and share content relating to a wide range of topics. In some cases, for example with politics, there are often issues that see people taking very different viewpoints. To provide a balanced platform for conversation and debate on these topics, it is crucial to take an unbiased approach to how the system decides which content to show to a user and which users to recommend to each other. It would be very easy for a person designing such a system to implement bias based on their own opinions, presenting all users with the same content that only shows one side of an argument. This would defeat the point of the Fidelis platform and be taking advantage of the users trust, so clearly these tactics will not be employed. All content suggestion will be based upon user preference above anything else.

3.3   Social Issues

Despite one of the main features of Fidelis being abuse detection, it is important to consider the implications should certain abusive posts go undetected. Due to the nature of social media and the difficulty to systematically detect the semantics of a post, it is likely that there will be posts which are not detected by the abuse algorithms. As a result, functions such as being able to block a user and delete posts must be included, in the event a user does receive abusive content.

As well as abusive content, further offensive content may be posted on the site which does not fall under the category of abuse. This includes nudity and links to inappropriate or illegal content. As with abusive content, the user will be able to remove content from their feed which they find offensive, but in extreme cases users may have to be reported to the authorities or removed from the Fidelis site altogether.

In addition to abusive content, Fidelis would also be subject to other social issues which have been recognised with preexisting social networks. This includes possible mental health implications of socialising online instead of face-to-face. Research has suggested that "digital communications less able to lower depression risk" in comparison with "people who regularly met in person with family and friends" [23]. Therefore, it is important that Fidelis attempts to mitigate this risk by encouraging users not to spend too much time on the social network in its 'Terms of Use' and also offers support by providing contact details of relevant helplines for those who seek help.

3.4   Professional Issues

For the Fidelis to be able to function, it will be required to store large amounts of user data. It is therefore necessary to ensure that the privacy of the users is protected, by abiding by the 'Data Protection Act' of 1998. This legislates that data is "fairly and lawfully processed; processed for limited purposes and not in any manner incompatible with those purposes; adequate, relevant and not excessive; accurate and where necessary, up to date; not kept for longer than is necessary; processed in line with the data subject's rights; secure and that personal information shall not be transferred to countries outside the EEA without adequate Protection" [1]. So that the Fidelis system meets these requirements it is paramount that measures are taken to secure the database and to minimise corruption to the data, as

well as regularly maintaining the data to remove anything which is surplus to the requirements of the system.

4 System Requirements

It is vital that clear project requirements are specified to set clear goals and track the progress of the project. The following requirements outline the current vision for the project but are subject to change over time. In order for the system to be successful and provide a useful service, it must meet these requirements as closely as possible.

4.1 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.

4.2   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 [13].

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.

4.3 Hardware and Software Constraints

It is essential that optimisation and efficiency is maintained in regards to scalability as the system gathers more data. Throughout the whole duration of this project it is likely that development will be carried out on a local or a cheap virtual server with low resources. As a result of this testing the system under extreme amounts of data can be a huge challenge. It is also important to bear in mind that any components that are added to the system should be designed to be scalable and should not impact the efficiency and scalability of the system negatively.

4.4 Foreseeable Challenges

The project is currently in the planning stage where the requirements and the vision are communicated to all members of the team as well as the project supervisor. Although research has been carried out to determine what is feasible and achievable in the time available, the requirements are based on theory. The codebase is yet to deployed and once the development has been initiated, challenges may arise which prevent some requirements from being fulfilled, either because of the language or frameworks chosen or just purely because of technological and time constraints. As a result of this, it is likely that the requirements will need to be updated.

5   System Design

When designing the system, it is important to first consider the platform upon which it is to be built and any underlying frameworks that will be used. Although no strict decision has been made, this section will outline the rough design plan for the system based on the requirements laid out previously.

5.1   System Architecture

An initial decision has been made to use web development technologies for deploying the first iteration of the system. A vast number of frameworks allow quick development of web applications meaning the system can be quickly prototyped and refined after testing. Furthermore, frameworks such as bootstrap allow responsive web development so that one application can be delivered to all platforms via a web browser.
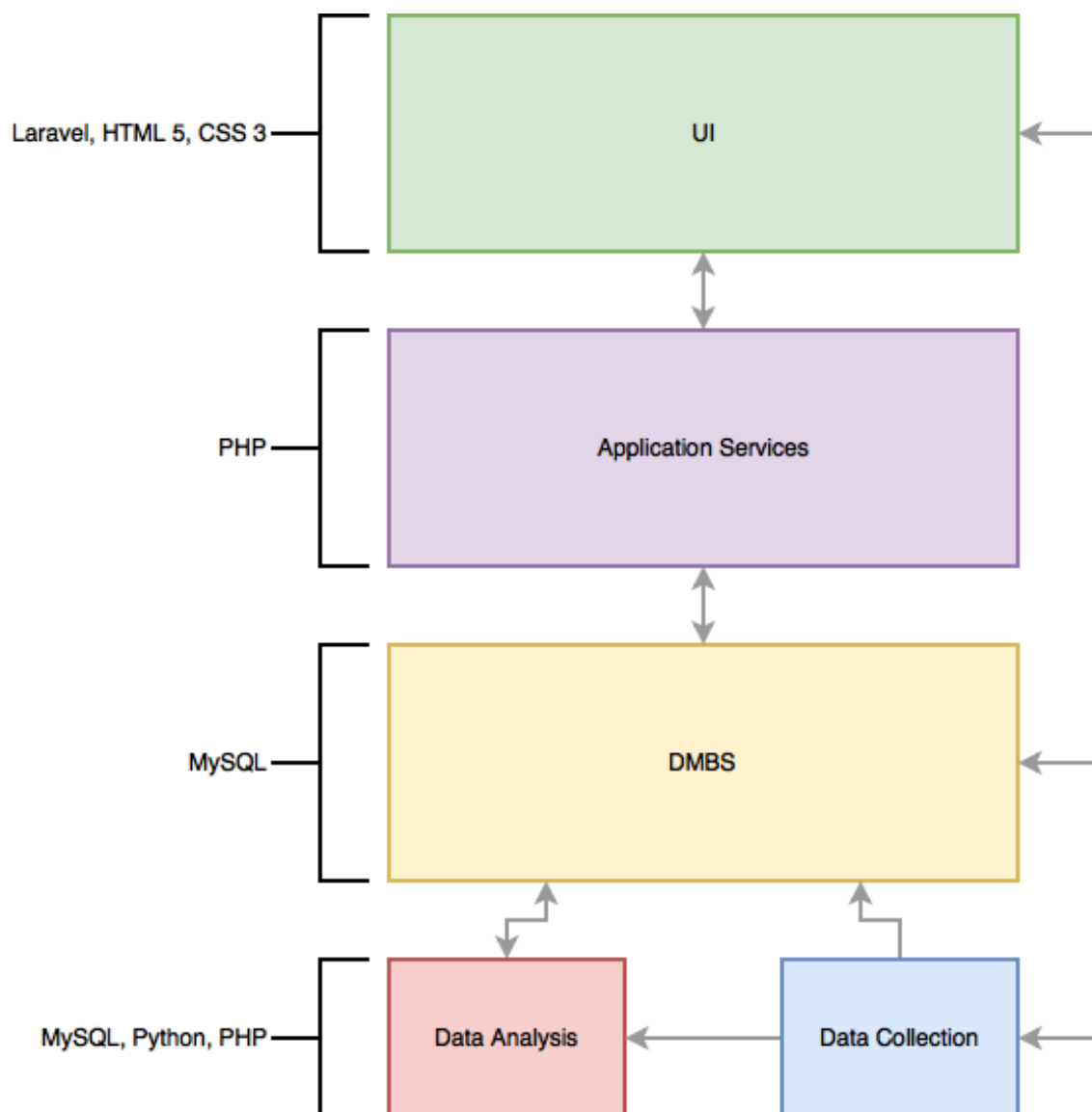
Figure 1: Layered System Architecture Diagram

The diagram in figure **??** above shows the proposed system architecture. As visible in the diagram, the users will communicate with the web server through a browser which will render pages to the user. The pages may contain dynamic content that is loaded from the database and output appropriately through the use of a server sided scripting language such as PHP. This data on its own will not be enough to provide content as required and PHP is not powerful enough to process such large quantities of data. For this reason data analysis algorithms will be run as jobs in another language that will read, analyse and update the data so it can simply be read and output by the application.

5.2 Mockup

The mockup in figure 2 shows a rough initial mockup of what the application homepage would show to a logged in user. The information on the page is split into sections so the structure is clear to the user and each section can be designed as a separate component. Sections such as 'Recommendations' and 'Discover' are always visible on the right as these are some of the key components of the applications. The navigation also shows a 'Discover' dropdown menu which will contain all the predefined categories.

Figure 2: Homepage Mockup

It is important to note that the mockup is only a first iteration and it is very likely to change as the design and development process starts and progresses.

6 Project Management

Due to the extensive length and size of this project, consideration into how the project can be effectively managed is very important to project success. This section will detail various ways in which the group will handle project management.

6.1 Design and Software Development Methodology

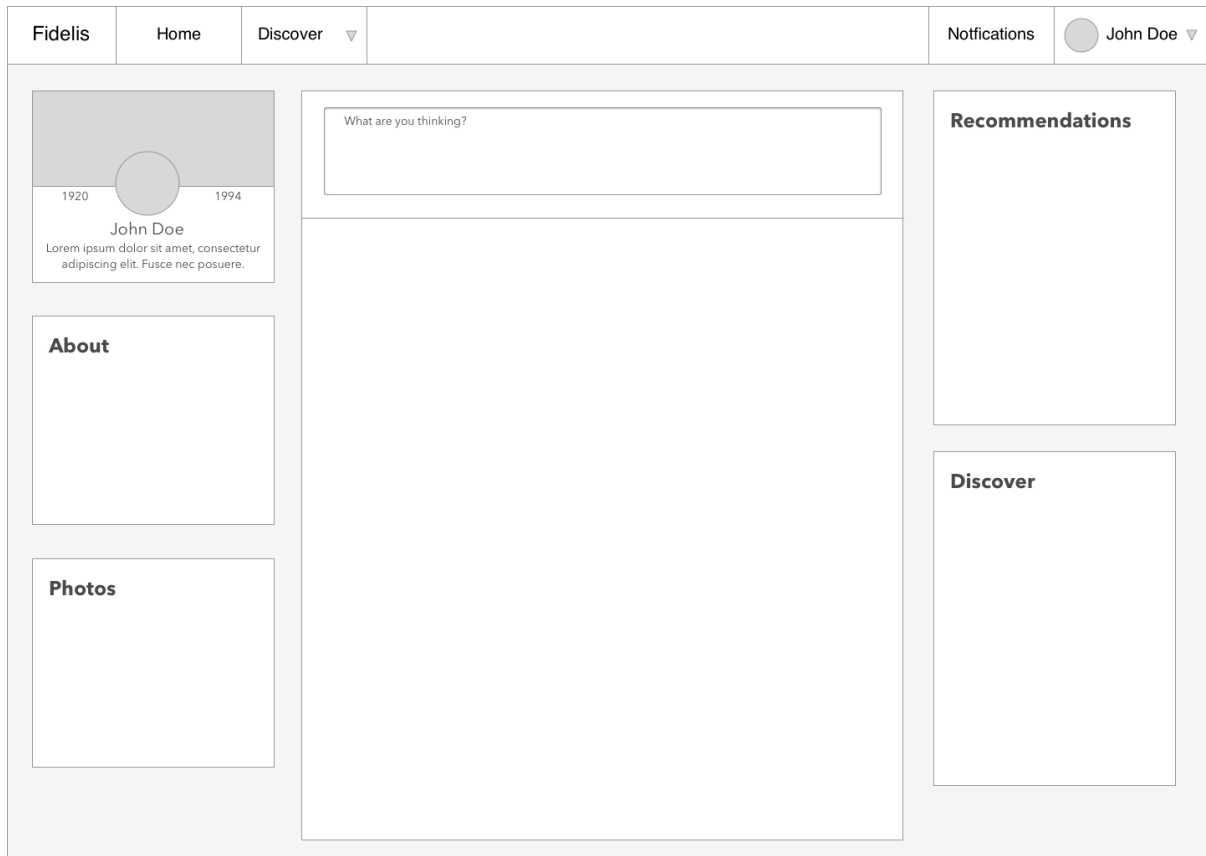Choosing a suitable design and software development methodology for Project Fidelis is imperative to the success of the project. It is therefore important to choose methodologies that will aid the project in its progress.

The design approach that will be adopted will be component-driven design. Component-driven design will allow more focus and thought to be given for each component. Such an approach works extremely well with Agile development as each component can be designed, built and tested. Agile techniques will be used due to their flexibility in handling evolving requirements. The project duration is over 6 months, so it is important to develop using an adaptive methodology. Building the system in this manner will help make components easier to integrate into the system, and will increase the efficacy of unit testing.

6.2 Project Timeline
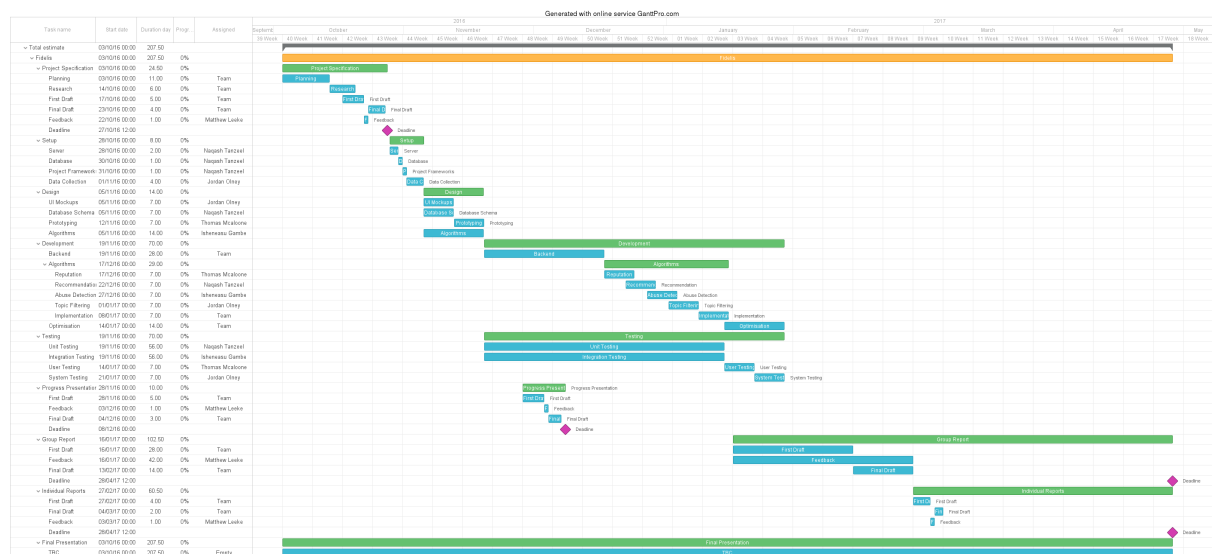
Figure 3: Project Timeline Gantt Chart

A Gantt chart will be used by the team to ensure all deadlines are tracked and met. The timeline sets goals for completion of project components and will provide the team with a metric to measure project progress. Different aspects of the project that need to be completed and who is assigned to complete each task are displayed on the chart. Showing the group members assigned to each project component ensures that everyone is accountable for their own tasks.

6.3 Roles and Responsibilities

The team consists of 4 members along with a project supervisor. Each team member has a unique role and their own responsibilities to ensure that the project is completed easily and smoothly. The

project supervisor will ensure that all assessed deadlines are met for the project. The project manager will ensure that internal deadlines, development related and otherwise, are met. Each team member is also responsible for ensuring that they complete their tasks on time, as assigned by the Gantt chart or project manager along the way. Below is a list of all the team members along with their roles and their responsibilities.

- **Matthew Leeke**, *Project Supervisor* - Ensure the external deadlines are met and the project is headed in the correct direction.

- **Isheneasu Gambe**, *Project Manager* - Ensure that the internal project deadlines are met and assign work to team members as the progress changes.

- **Naqash Tanzeel**, *Lead Developer* - Responsible for the development of the system and monitoring all technical aspects of the project. Responsible for making any decision regarding development.

- **Jordan Olney**, *Researcher* - Carrying out research on possible approaches to tackle any issues we face as well as making decisions on approaches and papers that are employed.

- **Thomas Mcaloone**, *Designer* - The design of the system as well as any graphical content for the project. Responsible for collecting images, icons and creating mockups which the system will be designed around.

The list above shows the unique responsibilities for each user but all members, excluding the project supervisor, also adopt developer as their secondary role.

6.4 Tools and Techniques

Several technologies and third-party tools will be utilised during the development and documentation of this project in order to improve efficiency and quality of work.

6.4.1 Data Collection and Analysis

(a) PHP 7   (b) JavaScript   (c) jQuery   (d) Python

Figure 4: Data Collection and Analysis Tools

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language [12]. PHP will be used for interacting with the database and majority of the data processing. JavaScript is a high level, dynamic, untyped, and interpreted programming language [7]. JavaScript will be used for some data processing but mainly for an improved user experience by updating the page without reloading.

In addition to the data processing at load time, background data processing and collection will be carried out using a scripting language such as Python or 'jobs' written in Python. These scripts will analyse the data in the database at a regular basis, as well as collect data from third-party APIs and store updated records in the database, ready to be displayed to the user.

6.4.2 Data Storage

(a) MySQL   (b) PhpMyAdmin   (c) Sequel Pro
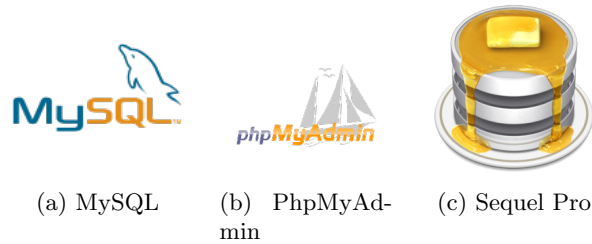
Figure 5: Data Storage Tools and Packages

MySQL is an open source database management system and will be used as the main database server for this project [24]. MySQL provides good read and write times in comparison to NoSQL databases such as MongoDB which provide good read times but slower write times. It is possible in the future when the amount of data increases drastically that a mixture of database solutions will be required, but this is not the case with the current amount of data. The system will be designed with a custom interface to the database so the source can be switched easily.

On the developer side, tools such as Sequel Pro and PhpMyAdmin will be used to visualise the data and browse through it. Tools like Transmit allow queries to be written and tested quickly before they are implemented in the application.

6.4.3 Data Visualisation

(a) HTML 5   (b) CSS 3   (c) JavaScript   (d) Laravel
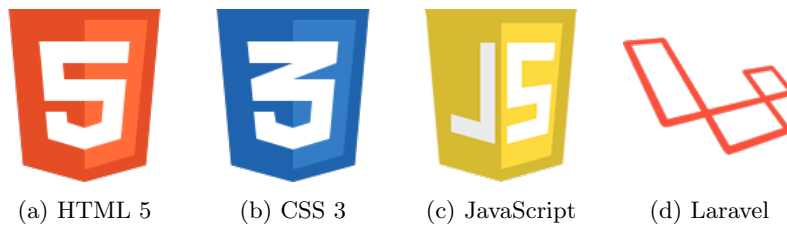
Figure 6: Data Visualisation Tools and Frameworks

HTML5 is a markup language used for structuring and presenting content on the World Wide Web [6]. It is likely a version of HTML will be used for the structure of the application. CSS3 is the latest evolution of the Cascading Style Sheets language [21]. CSS3 will be used to style the user interface so it is user friendly and simple to use. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web [3]. It contains predefined CSS selectors to ease design and development. The main purpose of using Bootstrap is that it will allow for a mobile responsive web app to be developed.

Additionally, frameworks such as Laravel will be used to speed up the process of creating a visual interface. Laravel provides Model-View-Controller (MVC) architecture which allows views to be created in separate files and rendered inside a main layout, thus reducing the amount of duplicate code whilst also speeding up development.

6.4.4 Organisation and Communication

(a) Github (b) Dropbox (c) Trello (d) Wunderlist
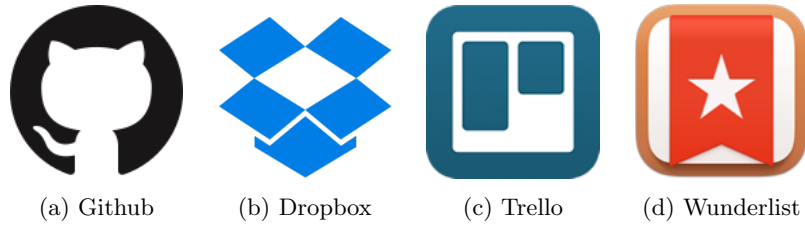
Figure 7: Organisation and Management Tools

Version control and backup will be carried out using Git. A private repository will be created on GitHub which will contain all the revisions. Additionally, rough documentation will be achieved through the use of GitHub by adding comments on each commit.

Dropbox will be used for storing all the documentation and other miscellaneous files regarding the project. This will allow collaboration on all the resources whilst also allowing the supervisor access to up-to-date copies of all project work.

A Trello board along with an excel sheet and a gantt chart is being used for the timeline of the project. This can be referred to in order to keep track of progress and plan the tasks for upcoming weeks.

WunderList is a todo list tool which will be used to keep track of tasks on a weekly basis. It will be synchronised with the gantt chart and the project timeline. Once the tasks are completed they can be marked as completed. If there any incomplete tasks from the previous week, they will be carried on to the next week.

6.4.5 Other Considerations

(a) PhpStorm (b) DigitalO- (c) XAMPP (d) Apache
cean

Figure 8: Other Tools and Resources

Along with the local development server, mentioned previously, DigitalOcean will be used to create a virtual private server with a distribution of Linux installed on it [8]. This VPS will be used as a web server with an XAMPP installation containing Apache and MySQL. This can be accessed through the DigitalOcean control panel using a browser. The server will primarily be used to host a functional commit, this allows user testing to be carried out without disruption whilst development continues.

7 Testing and Evaluation

It is important to ensure that all functional and non-functional requirements are met by the software through systematic requirements analysis and testing. Fidelis is a system, in which large amounts of data will be processed, stored, and presented to the user, through the cooperation of many components. By employing a component-driven design, coupled with the agile development, testing can take place at several levels.

7.1 Unit Testing

Unit testing involves verifying the functionality of a specific section of code such as individual function or classes. Any original or new code should be thoroughly tested, including the identification and handling of issues such as errors and corner cases. The appropriate testing framework for each incorporated technology will be employed in an effort to ensure good testing practices are followed.

7.2 Integration Testing

When individual components have been sufficiently assessed through unit testing, the modules will be integrated together to test the functionality at a larger scale, steadily moving towards real-world conditions for the system. Integration testing aims to expose any issues that may arise directly from the interactions of components, rather than issues with the components themselves.

7.3 System Testing

Once all components of the system have been integrated (and therefore undergone integration testing), system testing can take place. System testing is focused on testing the holistic system that the end users would interact with, thus indicating how well the system performs (with regards to the system requirements) at a high level.

7.4 User Acceptance Testing and Success Measurement

This type of testing involves having a third party test the software to make sure it can handle required tasks in real-world scenarios, according to the specifications. It can prove extremely valuable to have another person test the limits of the system this way, as they are more likely to approach a task in a different way, which may uncover an issue that otherwise wouldn't have been found. Once the testing has been approved, peers of the developers and the project supervisor will be given access to an alpha version of the software. Hopefully this will provide useful feedback on both the functionality and quality of the system.

7.5 Continuous Requirements Analysis

To keep on track with the aims of this project, the functional and non-functional requirements must be regularly analysed to ensure that they are still in line with the project aims, as well as making sure they have been met or are being worked towards. Because of the agile methodology being used, it is possible that the goals of the project may have to be modified somewhat. If this happens the requirements would also have to be modified to ensure that they still encompass what the project is to achieve.

8   Conclusion

This report states the specification for Fidelis. The aim for this project is to create a trust-based social network that will allow users to engage with each other in a manner conducive to intelligent debate, and also protect them from abusive content and users. Creating a system that meets this aim will introduce a new social network through which users can become self-informed on current topics through interactions with Fidelis.

   In this report, the project aims and key stakeholders were introduced in the first chapter. The second chapter details work on research into key system functionalities and existing systems. In the third chapter, various legal, ethical, social and professional issues that might affect the project are explored. The fourth and fifth chapters deal with system requirements and design respectively. Details of how project management will be approached were discussed in Chapter 6, and the group's strategy for testing was mentioned in Chapter 7.

References

[1] Data protection act, 1998.

[2] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] Bootstrap. Bootstrap. `http://getbootstrap.com`, oct 2015.

[4] M. Bramer. *Principles of Data Mining*. Springer, 2013.

[5] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, pages 121–167, 1998.

[6] W. Contributors. Html 5. `https://en.wikipedia.org/wiki/HTML5`, oct 2015.

[7] R. Data. Javascript. `http://www.w3schools.com/js/`, oct 2015.

[8] DigitalOcean. Digitalocean. `https://www.digitalocean.com`, oct 2015.

[9] Facebook. Company info — facebook newsroom. `https://newsroom.fb.com/company-info/`, January 2016.

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

[11] 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.

[12] T. P. Group. Php. `http://php.net`, oct 2015.

[13] S. Jacob. Speed is a killer – why decreasing page load time can drastically increase conversions. `https://blog.kissmetrics.com/speed-is-a-killer/`, May 2011.

[14] S. Jacob. Social media and international law. `http://www.olswang.com/articles/2014/07/social-media-and-international-law/`, July 2014.

[15] D. Jurafsky and J. Martin. *Speech and Language Processing*. Pearson Education, Inc, 2009.

[16] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 247–254. ACM, 2001.

[17] L. Kuncheva. *Combining Pattern Classifiers*. Wiley, 2014.

[18] L. Kuncheva. *Combining Pattern Classifiers*. Wiley, 2014.

[19] K. McNally, M. P. O'Mahony, and B. Smyth. A model of collaboration-based reputation for the social web. In *ICWSM*, 2013.

[20] R. L. Miller. *Business Law Today: The Essentials*. South-Western Cengage Learning, 2011.

[21] M. D. Network. Css3 - css — mdn. `https://developer.mozilla.org/en/docs/Web/CSS/CSS3`, sep 2015.

[22] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[23] OHSU. Research: Face-to-face socializing more powerful than phone calls, emails in guarding against depression in older adults. `https://www.ohsu.edu/xd/about/news_events/news/2015/10-05-research-face-to-face-s.cfm`, 2015.

[24] Oracle. Mysql. `https://www.mysql.com`, oct 2015.

[25] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer, 2010.

[26] Reddit. Reddit - about. `https://about.reddit.com`, January 2016.

[27] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[29] Statista. Facebook users worldwide 2016 — statista. `https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/`, July 2016.

[30] Statista. Reddit users: Unique monthly visits 2016 — statista. `https://www.statista.com/statistics/443332/reddit-monthly-visitors/`, August 2016.

[31] Statista. Twitter: Number of active users 2012-2016 — statista. `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`, July 2016.

[32] Twitter. Company — about. `https://about.twitter.com/company`, January 2016.

[33] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29(4):371–388, 2000.

[34] M. Zaki and W. M. Jr. *Data mining and analysis : fundamental concepts and algorithms*. Cambridge University Press, 2014.

Appendices

A   Gantt Chart

Generated with online service GanttPro.com

Task name	Start date	Duration day	Progr.	Assigned
Total estimate	03/10/16 00:00	207.50	0%	
v Roles				
v Project Specification	03/10/16 00:00	24.50	0%	
Planning	03/10/16 00:00	11.00	0%	Team
Research	14/10/16 00:00	6.00	0%	Team
First Draft	17/10/16 00:00	5.00	0%	Team
Final Draft	23/10/16 00:00	4.00	0%	Team
Feedback	22/10/16 00:00	1.00	0%	Matthew Leske
Deadline	27/10/16 12:00		0%	
v Setup	28/10/16 00:00	8.00	0%	
Server	28/10/16 00:00	2.00	0%	Naqash Tanzeel
Database	30/10/16 00:00	1.00	0%	Naqash Tanzeel
Project Framework	31/10/16 00:00	1.00	0%	Naqash Tanzeel
Data Collection	01/11/16 00:00	4.00	0%	Jordan Olney
v Design	05/11/16 00:00	14.00	0%	
UI Mockups	05/11/16 00:00	7.00	0%	Jordan Olney
Database Schema	05/11/16 00:00	7.00	0%	Naqash Tanzeel
Prototyping	12/11/16 00:00	7.00	0%	Thomas Moatoone
Algorithms	05/11/16 00:00	14.00	0%	Ishenassa Gambe
v Development	19/11/16 00:00	70.00	0%	
Backend	19/11/16 00:00	28.00	0%	Team
v Algorithms	17/12/16 00:00	29.00	0%	
Reputation	17/12/16 00:00	7.00	0%	Thomas Moatoone
Recommendatic	22/12/16 00:00	7.00	0%	Naqash Tanzeel
Abuse Detection	27/12/16 00:00	7.00	0%	Ishenassa Gambe
Topic Filtering	01/01/17 00:00	7.00	0%	Jordan Olney
Implementation	08/01/17 00:00	7.00	0%	Team
Optimisation	14/01/17 00:00	14.00	0%	Team
v Testing	19/11/16 00:00	70.00	0%	
Unit Testing	19/11/16 00:00	56.00	0%	Naqash Tanzeel
Integration Testing	19/11/16 00:00	56.00	0%	Ishenassa Gambe
User Testing	14/01/17 00:00	7.00	0%	Thomas Moatoone
System Testing	21/01/17 00:00	7.00	0%	Jordan Olney
v Progress Presentatic	28/11/16 00:00	10.00	0%	
First Draft	28/11/16 00:00	5.00	0%	Team
Feedback	03/12/16 00:00	1.00	0%	Matthew Leske
Final Draft	04/12/16 00:00	3.00	0%	Team
Deadline	08/12/16 00:00		0%	
v Group Report	16/01/17 00:00	102.50	0%	
First Draft	16/01/17 00:00	28.00	0%	Team
Feedback	16/01/17 00:00	42.00	0%	Matthew Leske
Final Draft	13/02/17 00:00	14.00	0%	Team
Deadline	28/04/17 12:00		0%	
v Individual Reports	27/02/17 00:00	60.50	0%	
First Draft	27/02/17 00:00	4.00	0%	Team
Final Draft	04/03/17 00:00	2.00	0%	Team
Feedback	03/03/17 00:00	1.00	0%	Matthew Leske
Deadline	28/04/17 12:00		0%	
v Final Presentation	03/10/16 00:00	207.50	0%	
TBC	03/10/16 00:00	207.50	0%	Empty

Figure 9: Project Timeline Gantt Chart

B   Twitter Content Categories

Category	Definition	Reference	Examplar
Conversational	*Post addresses another user*		
1. Query	Questions, question marks or polls	Pear Analytics, 2009; Java, *et al.*, 2007; Jansen, *et al.*, 2009; Naaman,*et al.*, 2010; Honeycutt and Herring, 2009	Invading Germany from France. Who's with me?
2. Referral	A response which contains URLs or recommendation of other users.	Pear Analytics, 2009; Naaman, *et al.*, 2010; Honeycutt and Herring, 2009	@USERNAME Items under $1000 are exempt. http://is.gd/AV7K
3. Action	Activities involving other users	Jansen, *et al.*, 2009; Hohl, 2009; Honeycutt and Herring, 2009	*waves at @USERNAME*
4. Response	Catch–all classification for conversation	Jansen, *et al.*, 2009; Java, *et al.*, 2007; Steiner, 2009; Honeycutt and Herring, 2009	@USERNAME Beware the polar bears.
Status	*An answer to "What are you doing now?"*		
1. Personal	Positive or negative sentiment in the form of personal opinion or emotional status	Jansen, *et al.*, 2009; Naaman, *et al.*, 2010; Honeycutt and Herring, 2009	I liked Modest Mouse after they became famous.
2. Temporal	Content referencing specific dates, times, statements of temporal nature (waiting) and temporal action ("Time to")	Naaman, *et al.*, 2010	Waiting for my 2pm performance review to start.
3. Location	Geographic references and location statements, including statements of traveling, location change	Makice, 2009; Naaman, *et al.*, 2010	Standing in a lecture theatre talking about Marketing Management.
4. Mechanical	Statements relating to any form of technology or mechanical systems (cars, phones, printers and photocopiers)		Well … I'm in trouble. Used 3829.060MB (62.322%) of your 6GB. You have 22 days remaining.
5. Physical	Sensory experiences of a physical nature	Naaman, *et al.*, 2010	It's freezing out there this morning.
6. Work	Reference to work related activity	Heany and McClurg, 2009; DiMicco,*et al.*, 2008; Zhao and Rosen, 2009	Firing off e–mail after e–mail to clear my to do list (knowing that's a great way to regenerate to do list items doesn't stop me or help me).
7. Automated	Status announcements triggered by third party applications such as media players, games or software	Honeycutt and Herring, 2009	
8. Activity	Activity statements answering "What are you doing now?"	Naaman, *et al.*, 2010; Honeycutt and Herring, 2009	Playing with the Internet in the name of science.
Pass along	*References other content*		
1. RT	Any statement reproducing another status	Java, *et al.*, 2007; Pear Analytics, 2009; Naaman, *et al.*, 2010; boyd, *et al.*, 2010	L4D Survivors in Rockband2 singing L7 Pretend We're Dead. http://is.gd/BsVE (HT to

			@USERNAME). It's seriously amazing.
2. UGC	Links to content created by the user (blog/video/picture)	Pear Analytics, 2009; Naaman, *et al.*, 2010; Honeycutt and Herring, 2009	http://twitpic.com/2o1c1 — Bus Slogan Generator Time — http://is.gd/hU2Q.
3. Endorsement	Links to Web content not created by the sender	Zhao and Rosson, 2009; Naaman, *et al.*, 2010; Honeycutt and Herring, 2009; boyd, *et al.*, 2010	I'm looking myself up on Publish or Perish (http://rurl.org/iw4) to find a reference to a paper that cited me because I want to cite them.
News	*Identifiable news content which is not UGC*		
1. Headlines	Coverage of breaking news and personal eyewitness accounts of news events	Fahmi, 2009; Lariscy, *et al.*, 2009; Gay, *et al.*, 2009; Pear Analytics, 2009; Java, *et al.*, 2007	
2. Sport	Identifiable results of sporting events	Java, *et al.*, 2007; Pear Analytics, 2009	
3. Event	Any tweet which represents the live discussion of an identified or identifiable event	Lariscy, *et al.*, 2009; Gay, *et al.*, 2009	Between NASA's satellite and autoanalysis of imagery, and Google Map data, scientific proof where there's smoke, there are fires #bcc2
4. Weather	Report of weather conditions without commentary	Honeycutt and Herring, 2009	
Phatic	*Content independent connected presence*		
1. Greeting	Statements of greetings to the broader Twitter community	Naaman, *et al.*, 2010, Hohl, 2009; Honeycutt and Herring, 2009; Miller, 2008	Good morning Twitterverse. How's the world outside?
2. Fourth wall	Textual equivalent of comments made directly to camera in television or cinema	Miller, 2008; Honeycutt and Herring, 2009	Note to self: Just because you're carrying tiny vials of hypercaffeine is no reason to start calculating remote delivery systems for them.
3. Broadcast	Textual soliloquy, monologue and undirected statements of opinion	Zhao and Rosson, 2009; Crawford, 2009; Honeycutt and Herring, 2009; Naaman, *et al.*, 2010	Diplomacy is the art of saying "Nice doggy" until you find a big enough rock. Captaincy is the timely provision of large enough rocks.
4. Unclassifiable	'saf''12 ^H^H. Errors, cat–on–keyboard input and unclassifiable strings of text	Honeycutt and Herring, 2009	AAAAAAAAAAAAAARGH
Spam	Junk traffic, unsolicited automated posts, and other tweets generated without user consent due to malware	Pear Analytics, 2009	

Figure 10: Twitter Content Categories Table