

COM814: Project 2016

Dissertation

School of Computing & Information Engineering

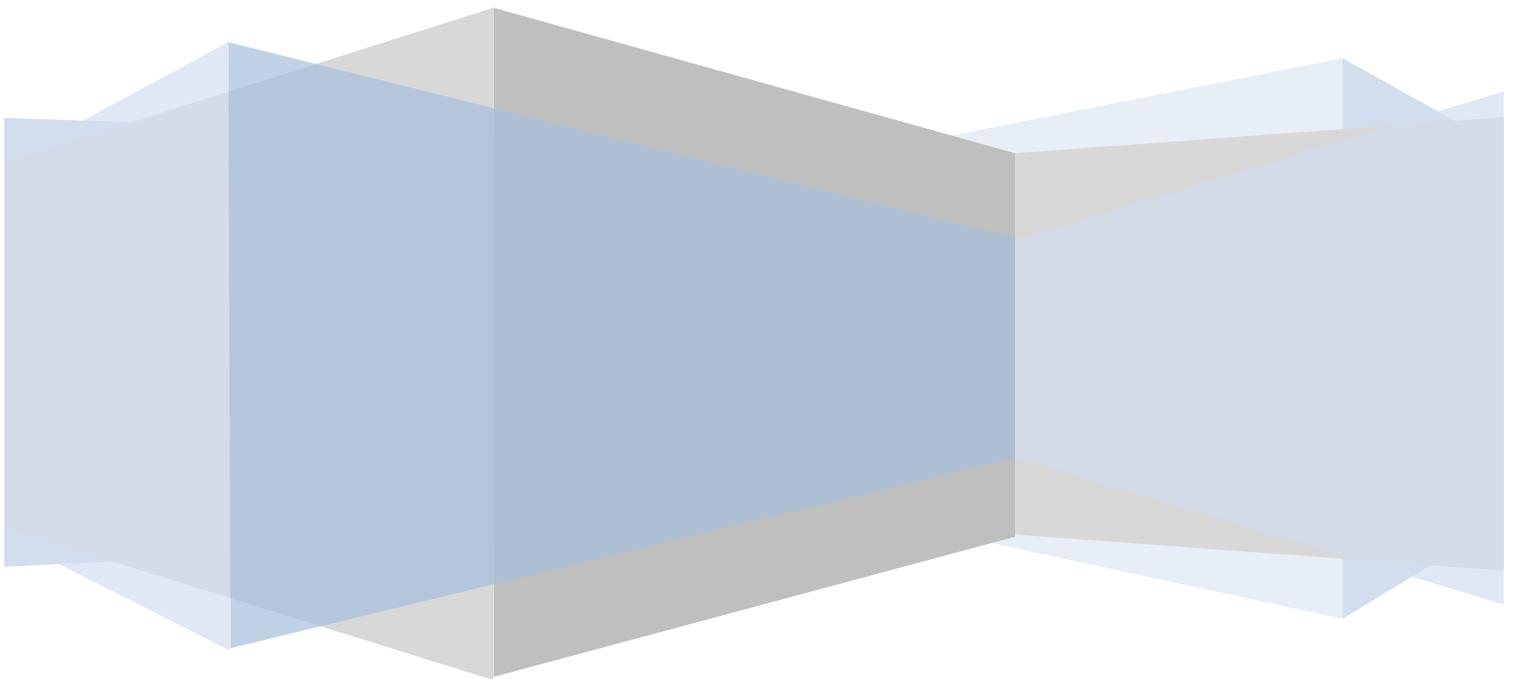
Stacey Crockett B00527540

**Collaborative Knowledge Management: System Development
and Deployment**

Supervisor: Sandra Moffett

Second Marker: Inaki Rano

1st September 2016



Plagiarism Statement

I declare that this is all my own work and does not contain unreferenced material copied from any other source. I have read the University's policy on plagiarism and understand the definition of plagiarism. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

Acknowledgements

I would like to thank my supervisor, Sandra Moffett, for her guidance, assistance and feedback throughout the project. Also thanks to Inaki Rano for his advice and recommendations.

Thanks to all the participants who completed the research questionnaire and the user testing questionnaire. Their feedback was essential for the successful completion of the project.

Thanks to my family and friends who have supported me throughout the completion of the MSc program.

Contents

Plagiarism Statement	2
Acknowledgements.....	2
List of Figures	6
List of Tables.....	9
Abstract	10
1: Introduction	11
1.1: Introduction	11
1.2: Problem Statement.....	12
1.3: Aim.....	12
1.4: Objectives	12
1.5: Requirements for Development	12
1.6: Dissertation Outline.....	13
2. Background Research	14
2.1: Introduction	14
2.2: Online Knowledge Sharing Systems.....	14
2.3: Electronic Word of Mouth (EWOM)	14
2.4: What motivates a user to engage in EWOM?.....	15
2.5: Requirements established from background research.....	16
2.6: Comparative Analysis.....	17
2.6.1: Goodreads	18
2.6.2: LibraryThing.....	19
2.6.3: Requirements established from the comparative analysis	20
2.7: Summary.....	21
3: Requirements Analysis	22
3.1: Introduction	22
3.2: Requirements Research	22
3.2.1: Research Questionnaire	22
3.2.2: User Stories	25
3.3: Requirements	26
3.3.1: Functional Requirements	26
3.3.2: Non-Functional Requirements	26
3.4: Summary.....	27
4: Design.....	28
4.1: Introduction	28

4.2: Design Methodology.....	28
4.3: User Interface Design	28
4.3.1: Interface Design Guidelines	28
4.3.2: Website Mock-ups	30
4.3.3: Navigation	31
4.3.4: Colour Scheme	31
4.3.5: Layout.....	32
4.4: Architectural Design	34
4.4.1: Site Map	34
4.4.2: Programming Languages	34
4.4.3: Database Design.....	35
4.5: Summary.....	37
5: Implementation.....	38
5.1: Introduction	38
5.2: Hardware	38
5.3: Software.....	38
5.3.1: Implementation Platform.....	38
5.3.2: Web Host and FTP Client.....	38
5.3.3: Database Implementation.....	38
5.4: Core Functionalities	39
5.4.1: Provide users with book information.....	39
5.4.2: Search for books.....	41
5.4.3: Read reviews and ratings	42
5.4.4: Register and Login	43
5.4.5: User Profile.....	45
5.4.6: Follow other Users	48
5.4.7: Newsfeed	50
5.4.8: Add ratings and reviews	51
5.4.9: Mark reviews as helpful	53
5.4.10: Discussions	54
5.4.11: Lists.....	57
5.4.12: Recommendations	60
5.4.13: Chat	62
5.5: Implementation of non-function requirements	65
5.5.1: Usability and Accessibility	65
5.5.2: Performance.....	65

5.5.3: Security.....	65
5.6: Summary.....	65
6: Testing.....	66
6.1: Introduction.....	66
6.2: In-House Testing.....	66
6.2.1: Functional Testing.....	66
6.2.2: Cross-Browser Testing.....	67
6.2.3: Cross-Platform Testing.....	68
6.2.4: Improvements after in-house testing.....	70
6.3: External Testing.....	71
6.3.1: Improvements after external testing.....	73
6.4: Testing after improvements.....	77
6.5: Testing non-functional requirements.....	77
6.5.1: Usability and Accessibility.....	77
6.5.2: Performance.....	77
6.5.3: Security.....	77
6.6: Summary.....	78
7: Conclusion.....	79
7.1: Introduction.....	79
7.2: Project Summary.....	79
7.3: Results Assessment.....	80
7.4: Objectives and Achievements.....	80
7.5: Recommendations.....	81
7.6: Summary.....	81
References.....	83
Appendices.....	87
Appendix 1: Research Questionnaire.....	87
Appendix 2: Storyboards.....	92
Appendix 3: User Testing Questionnaire.....	94

List of Figures

Figure 1: Goodreads Homepage.....	18
Figure 2: LibraryThing Homepage	19
Figure 3: Digital Mock-up of the Homepage	30
Figure 4: Flatly Colour Scheme.....	31
Figure 5: Website Layout.....	32
Figure 6: Book Page Layout	32
Figure 7: User profile layout.....	33
Figure 8: List/Discussion page layout	33
Figure 9: Site Map	34
Figure 10: ER diagram	35
Figure 11: Inserting a book into the database.....	39
Figure 12: SQL query to retrieve new releases	39
Figure 13: Code to display books	40
Figure 14: Homepage showing books	40
Figure 15: SQL query to get all book details.....	40
Figure 16: PHP code to display book details	41
Figure 17: Book details being displayed on web page	41
Figure 18: Search bar HTML	41
Figure 19: SQL query to search the database.....	42
Figure 20: Search Results	42
Figure 21: SQL query to get book reviews.....	42
Figure 22: Review for a user who is not logged in.....	43
Figure 23: Review for a user who is logged in	43
Figure 24: Code to show average rating.....	43
Figure 25: Login and Register links	43
Figure 26: Form group with input for user's first name	43
Figure 27: Validation for first name	44
Figure 28: Error message is displayed	44
Figure 29: New user being created	44
Figure 30: Login validation	45
Figure 31: Login error message	45
Figure 32: User profile link in navigation bar	45
Figure 33: User Profile.....	46

Figure 34: Empty div to hold AJAX response	46
Figure 35: Function to create a new XMLHttpRequest	47
Figure 36: showActivity() function	47
Figure 37: SQL query to get user's activity	47
Figure 38: Code to output user activity	48
Figure 39: Functions to follow and unfollow a user	49
Figure 40: Code showing follow div and follow buttons	49
Figure 41: followUser.php	50
Figure 42: SQL query to get newsfeed activity	50
Figure 43: Code for newsfeed activity	51
Figure 44: Rating Widget	51
Figure 45: showRating() function	52
Figure 46: Response is displayed	52
Figure 47: showRatingReviewDiv() function	52
Figure 48: Review form for book already rated	53
Figure 49: Code to add review to database	53
Figure 50: Helpfulness code	54
Figure 51: createNewDiscussion.php	55
Figure 52: Discussion has been created	55
Figure 53: Before joining the discussion	56
Figure 54: joinDiscussion() function	56
Figure 55: After joining the discussion	56
Figure 56: addPost.php	57
Figure 57: After creating a new list	58
Figure 58: followList() function	58
Figure 59: Add to dropdown	59
Figure 60: Add to button code	59
Figure 61: getAddToList() function	59
Figure 62: getAddToList.php	60
Figure 63: recommendations.php	60
Figure 64: SQL query to get recommended books	61
Figure 65: SQL to get recommended lists	61
Figure 66: SQL to get recommended discussions	62
Figure 67: messages.php	62

Figure 68: Messages	63
Figure 69: Messages JavaScript functions	63
Figure 70: Unread Message.....	64
Figure 71: Marking messages as seen	64
Figure 72: Unread messages	64
Figure 73: refreshUnread() function	64
Figure 74: SQL query preventing SQL injection	65
Figure 75: Layout problems on book details page	69
Figure 76: Layout issues on profile page	69
Figure 77: Layout issues on list details page	69
Figure 78: Layout issues on new releases page.....	70
Figure 79: Layouts after changes.....	70
Figure 80: New profile layout.....	71
Figure 81: Notifications Dropdown	74
Figure 82: Replying to a discussion post	74
Figure 83: New search results page	75
Figure 84: Autofill dropdown	75
Figure 85: Recommended For You section on homepage	76
Figure 86: New Recommendations page	76

List of Tables

Table 1: Table showing statistics of competitor websites.....	17
Table 2: Pros and Cons of Goodreads	19
Table 3: Pros and Cons of LibraryThing	20
Table 4: Questionnaire Results.....	24
Table 5: Functionality Testing Results Table	67
Table 6: Cross-Browser Testing Results	68

Abstract

Online knowledge sharing systems are becoming more popular with users who are seeking reviews of items and services. People rely on these sites to help them when making decisions and purchasing products. Users can also help others by sharing their knowledge on these websites.

This dissertation describes the development and deployment of a new online knowledge sharing system, “The Book Network”, that allows users to read and share their book reviews online.

“The Book Network” was created as a cross-browser and cross-platform website that can be accessed on both computers and tablets. To establish the requirements for the website, background research, a comparative analysis and a research questionnaire were used. Usability and accessibility guidelines were followed during the design to deliver a positive user experience. The system was tested successfully and additional features and improvements were implemented, followed by further testing to verify the functionality of the new additions. The website was found to have successfully met all the objectives and requirements that were established. Recommendations were then made for further development.

Keywords: knowledge sharing, website, electronic word of mouth, cross-browser, cross-platform.

1: Introduction

1.1: Introduction

In today's society, the internet has become an integral and often essential part of everyday life. In 1995 less than 1% of the population had an internet connection. Today, the number of internet users is over 3 billion, around 40% of the world population (Internet Live Stats, 2016). These statistics show how internet usage is continuously increasing around the world.

People use the internet for many different purposes, and one of the most popular uses is for knowledge sharing. This can be in the form of a site specifically designed for knowledge sharing, an example of this would be TripAdvisor or IMDB. Knowledge sharing can also happen on social media sites where users can review businesses and products.

TripAdvisor has 320 million reviews on their website, 96 million members and every minute 200 new reviews are posted (TripAdvisor, 2016). IMDB have more than 250 million unique visitors every month (IMDB, 2016) and Goodreads have reported 41.8 million monthly unique visitors (Quantcast, 2016). These statistics show the popularity of knowledge sharing websites.

In recent years, people have become more and more reliant on online reviews when purchasing products. People rely on knowledge sharing websites to make decisions and solve problems, with many users sharing their knowledge and experience to help other users. Users are also able to access a wide variety and large volume of reviews in a shorter period of time than if they were engaging in traditional word of mouth.

A user can find reviews for almost any product online, and a discussion forum for almost any topic, therefore this project will focus on the improvement of existing systems.

Currently there are several knowledge sharing sites for book reviews and discussions. However, each site has drawbacks which can lead to user dissatisfaction. Therefore, the aim of this project is to design and implement a prototype knowledge sharing system centred around book reviews, which will facilitate knowledge sharing between users, allow social interactions and discussion between users, and address the shortcomings of current knowledge sharing sites.

In order to meet these aims, a competitive analysis was undertaken and a questionnaire was used to determine user requirements and other features of the website. A review of the literature related to knowledge sharing via electronic word of mouth was also completed. This research had a specific focus on what encourages users to share their knowledge and how this can be implemented in the new website.

1.2: Problem Statement

The problem can be summarised in the following problem statement:

“Within the area of book review sites, design and implement a prototype knowledge sharing system, which will a) facilitate knowledge sharing between users, b) allow social interactions and discussion between users and c) address the shortcomings of current knowledge sharing sites.”

1.3: Aim

To develop a website which addresses the requirements outlined in the problem statement, that will allow users to share knowledge and interact with each other while eliminating the problems associated with current competitor sites.

1.4: Objectives

In order to achieve this aim, the following objectives were established:

- Conduct background research to determine why users participate in online knowledge sharing and how to encourage user participation.
- Evaluate current book based online knowledge sharing websites, identifying their benefits and drawbacks, and how these can be integrated and eliminated.
- Create and distribute a questionnaire to the target user group via online forums and use the results of this to determine the user requirements and features for the website.
- Design and implement the website according to the requirements previously defined, while ensuring that the website meets established accessibility and User Interface design guidelines.
- Create and administer a testing questionnaire and use this feedback to determine future requirements.

1.5: Requirements for Development

To develop the website, the hardware, software and knowledge requirements were as follows:

- A laptop/desktop computer with Windows operating system.
- IntelliJ development platform with plugin to support PHP development.
- MySQL database to store website data
- Kindle tablet to assess mobile capabilities
- Knowledge of HTML5, CSS, JavaScript and PHP programming languages

1.6: Dissertation Outline

Chapter 2 outlines the definitions, uses and benefits of online knowledge sharing systems and electronic word of mouth. A discussion of the reasons a user engages in electronic word of mouth is then included, exploring the current literature available on this topic. From this background research, several important factors were identified that must be considered during the development of the website in order to encourage users to share knowledge with others. A competitive analysis is then included to examine the pros and cons of similar websites.

Chapter 3 discusses the results of a research questionnaire which was administered to target users via online forums. The findings of this questionnaire are discussed, and user requirements and features are then presented in the form of user stories.

Chapter 4 outlines the design phase of the development. The user interface and design guidelines are discussed. The system architecture and database design are also included.

Chapter 5 discusses the implementation phase of the website. The development process is discussed, including the languages, tools and processes used to create the website.

Chapter 6 includes the outcome of testing by the developer and the testing by target users. The feedback from this testing is then discussed, including any new features and improvements implemented as a result.

Chapter 7 is the conclusion of the project. A summary of the project and the results is included. The project is assessed against the aims and objectives and whether it meets the user requirements defined by the questionnaire results and background research. Finally, short term and long term requirements for future development are then discussed.

2. Background Research

2.1: Introduction

This chapter starts by discussing online knowledge systems and electronic word of mouth, looking at their definitions, uses and benefits. The reasons that users engage in EWOM are then discussed extensively, identifying steps to be taken during development to encourage users to share their knowledge with others. A competitive analysis of similar websites is then included, along with any further requirements.

2.2: Online Knowledge Sharing Systems

Knowledge sharing systems promote the creation, transfer and application of knowledge (Alavi & Leidner, 2001). Knowledge sharing can be defined as “the combination of one or both parties seeking knowledge in response to a request, such that one or both parties are affected by the experience” (Ghosh & Scott, 2007). People rely on knowledge to solve problems and make decisions and, because of the internet, they can now rely on online communities to find this knowledge and help them to solve their problem, for example, choosing a product or service. Online knowledge has become an important factor for consumers when they are making decisions, and online communities facilitate the sharing of this knowledge (Ku et al., 2015).

2.3: Electronic Word of Mouth (EWOM)

Electronic word of mouth (EWOM) is defined as “any positive or negative statement made by potential, actual, or former customers about a product or company, which is made available to a multitude of people and institutions via the internet” (Hennig-Thurau et al., 2004). EWOM communications include blogs, discussion forums, product review sites and social networking sites. Sites for rating products and services are extremely popular, boasting hundreds of millions of users (Gehring & Duong, 2015). The people within these online communities share a mutual interest in the topics or the products, and engage in public discussions online (King et al., 2014). Where people would previously have engaged in traditional word of mouth discussions, EWOM allows people to have access to a high volume of opinions from a large community of users in a shorter period of time, where they can draw greater benefits from the ‘crowd’ (Dellarocas, 2003). EWOM allows customers to interact socially, share product information and experiences, and go on to make informed purchases based on this information (Blazevic et al. 2013; Hoffman & Novak, 1996).

Research has found that people seek EWOM to reduce search and evaluation efforts (Dabholkar, 2006; Goldsmith & Horovitz, 2006), to reduce risk (Kim et al, 2011; Sweeney et al, 2008) and to find social

assurance/reassurance (Bailey, 2005). Cheung & Lee (2012) found that reputation, a sense of belonging and the enjoyment of helping others were the most common reasons for people to engage in EWOM. With this increased reliance on EWOM and online purchasing, along with the continuing growth of social media, EWOM has become extremely important (King et al, 2014).

This helps us to understand why someone would seek EWOM in the first place, but it is also important to consider what makes a user then share knowledge and continue to do so within these online communities.

2.4: What motivates a user to engage in EWOM?

The creation of a knowledge sharing site does not guarantee that any knowledge exchange will actually take place (Cheung & Lee, 2007), so it is essential to understand the motivation for continuous knowledge sharing. Due to the nature of the intended site, it would be easy for users to read reviews but not contribute themselves, therefore it will be critical to facilitate social interaction and encourage all users to share their knowledge (Shen et al, 2010).

To understand this, it would be useful to look at knowledge sharing in a more general sense, not only within knowledge sharing systems.

A public good can be defined as “a shared resource from which every member of a group may benefit, regardless of whether or not they personally contribute to its provision, and whose availability does not diminish with use” (Cabrera & Cabrera, 2002). A social dilemma occurs when a user consumes resources without contributing to the group. However, even though public goods are subjected to social dilemmas, they are usually created and maintained through collective action (Wasko & Teigland 2004). Batson (1994) proposed a framework to explain why people act for the public good. He identified four different motives – egoism, collectivism, altruism and principlism. He theorised that there was an ultimate goal within each of these motives. The ultimate goal for egoism is self-benefit, for collectivism it is to increase group welfare, for altruism it is to increase one or more other individuals’ welfare, and for principlism it is to uphold one or more moral principles.

Cheung and Lee (2007) attempted to apply Batson’s framework to knowledge sharing systems. They found that the key factors that encourage users to engage in continuous knowledge sharing were moral obligation, commitment, knowledge self-efficacy and satisfaction. They make three suggestions for designing virtual communities:

1. In order to promote the norm of knowledge sharing in the virtual community and create a moral obligation, they suggest providing guidelines for users that place an emphasis on knowledge exchange among members.

2. To build a sense of community among members, creating user commitment, they suggest giving users the option to post their personal profiles, including their expertise and experience in a specific area.
3. Within the areas of knowledge self-efficacy and satisfaction, in order to indicate to users that their contribution is significant, they suggest providing a way for contributors to be identified and informed that they have helped others, and allowing the people who use this knowledge to show that their appreciation.

There have been many research studies conducted on knowledge sharing in virtual communities. Other key factors that have been shown to affect knowledge sharing behaviour are trust (Usoro et al, 2007; Wu et al., 2009), knowledge sharing self-efficacy and perceived relative advantage (Lin et al. 2009), community identity and social awareness (Tseng & Kuo, 2010).

A recent study by Lai & Hsieh (2013) found that another important factor was reputation and this had a stronger positive effect on continued knowledge sharing intention when members have a high level of knowledge self-efficacy. They also found that when members have a high level of knowledge sharing experience in a virtual community, reciprocity and moral obligation both have a stronger positive effect on continuous knowledge sharing. Finally, they found that when members have a high level of knowledge self-efficacy, the enjoyment of helping others has a stronger positive effect on continuous knowledge sharing.

In 2015, Ku et al. found five key factors that influence continuous knowledge sharing. These are the number of reviews written, the average helpfulness scores given to others, the average helpfulness score received by others, the ratio of negative reviews to total reviews and the number of trustors.

2.5: Requirements established from background research

By exploring the relevant background research, we can identify several important factors to consider when designing and implementing the knowledge sharing system:

- An emphasis should be placed on knowledge sharing, encouraging users to share their reviews by making reviews and ratings an important part of the web design.
- To address knowledge self-efficacy and satisfaction needs, users will be able to post reviews publicly with their username, and other users will be able to mark the review as helpful if they find it helpful. This will indicate to users that their contribution is valued.
- To address commitment and create a feeling of community, users will be able to create their own personal profile. They will be able to follow other users; this will encourage trust among users and allow users to see that they have trustors.

- Users will also be able to create and join discussion boards allowing users to build a community identity, social awareness and reputation. Also by joining discussions, the need for reciprocity is addressed.
- Users will be able to make personalised lists that other users can follow. This will address reputation and trust between users.
- By having a newsfeed, the user will be able to see the activity of any users they follow as well as their own activity. This will encourage users to interact with others, share reviews and join in discussions.

2.6: Comparative Analysis

In order to review the current websites available for knowledge sharing in the area of book reviews, a comparative analysis was carried out. This analysis investigated the popularity of several different websites based on their global rank, the number of monthly visits, the average time spent on the site by a user, the daily page views per visitor, the bounce rate and the speed of the website. The statistics were taken from similarweb.com and alexa.com.

Provided by	Similarweb.com					Alexa.com
Webpage	Global Rank	Estimated Monthly Visits	Time on site	Daily page views per visitor	Bounce Rate (%)	Speed
Goodreads.com	515	46.6M	4:48	4.62	50.13	Slow
LibraryThing.com	24,484	930K	4:48	5.82	47.96	Average
Shelfari	98,894	170K	7:09	6.70	31.16	Slow
The Reading Room	132,996	190K	1:55	2.27	71.82	Very Slow
aNobii	36,686	610K	5:53	5.81	43.97	Very Slow
Riffle	906,374	15K	2:47	4.00	56.93	Not provided
Book Likes	125,452	210K	2:30	2.33	70.82	Slow
Book Browse	129,103	210K	1:31	2.12	70.10	Slow

Table 1: Table showing statistics of competitor websites

The above table shows that the two most popular book review sites are Goodreads and LibraryThing. Therefore, the next step will be to look at these two sites in greater detail. Also, with the exception of LibraryThing which had an average speed, all of the other sites had a speed of slow or very slow.

Speed is important requirement for the website. A 1 second delay in load time results in fewer page views, decreased customer satisfaction and fewer conversions. Around 47% of people expect a web page to load in 2 seconds or less, and 40% will leave a web page if it takes more than 3 seconds to load (Aragon, 2013). The steps taken to increase speed are discussed in the design phase of the development.

2.6.1: Goodreads

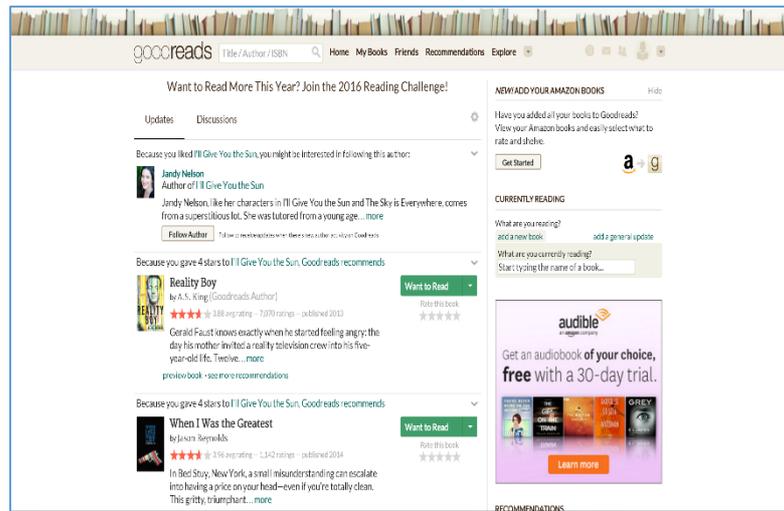


Figure 1: Goodreads Homepage

Goodreads launched in January 2007. On their website they describe themselves as ‘the world’s largest site for readers and book recommendations’, with a mission to ‘help people find and share books they love’. They describe Goodreads as ‘a large library that you can wander through and see everyone's bookshelves, their reviews, and their ratings’.

Goodreads was the most popular of all the websites investigated. The global rank of the site is 515 and they have 46.6 million estimated monthly visits. In order to investigate the pros and cons of the Goodreads, the website was explored at length and the following pros and cons were determined:

Pros	Cons
<p>43 million reviews</p> <p>40 million users</p> <p>1.1 billion books</p> <p>Free to use</p> <p>Can import books from Amazon</p> <p>Can create a personal profile</p> <p>Can search for book and read reviews without registering</p> <p>Can add friends from Facebook, Gmail and Twitter</p> <p>Can follow other users</p> <p>Able to create 'shelves'</p> <p>Discussion boards</p> <p>Personalised recommendations</p>	<p>Not ideal for cataloguing large collections</p> <p>Layout can be confusing</p> <p>Colour scheme is bland</p> <p>Can be complicated to use</p> <p>Cluttered layout</p> <p>A lot of advertising</p> <p>Lost user trust when taken over by Amazon</p> <p>Slow loading</p>

Table 2: Pros and Cons of Goodreads

2.6.2: LibraryThing

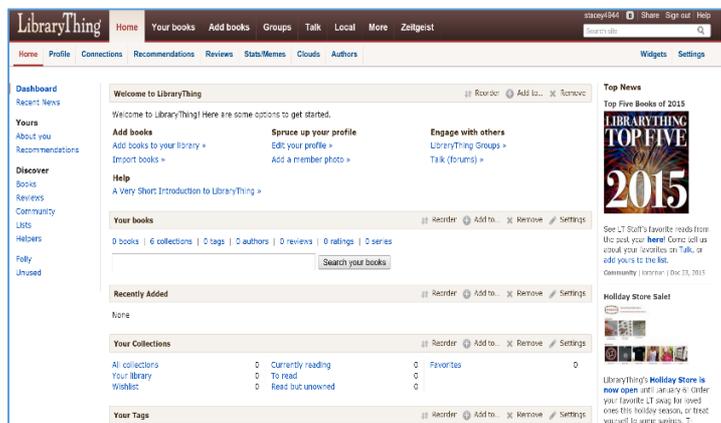


Figure 2: LibraryThing Homepage

LibraryThing was created in 2006 and allows users to catalogue their book collection. On their website, LibraryThing is described as ‘a community brought together by the love of books’.

LibraryThing was the second most popular of all the sites investigated. Their global rank is 24,484 and they have 930K estimated monthly visits. Although LibraryThing is not as popular as Goodreads, their number of daily page views per visitor is higher and their bounce rate is lower. This may be due to the

speed of LibraryThing which is reported as average compared to Goodreads speed which is reported as slow.

In order to investigate the pros and cons of the LibraryThing, the website was explored at length and the following pros and cons were determined:

Pros	Cons
2 million members 101 million books 2.5 million reviews and 15 million ratings 10 thousand groups and 175 thousand discussion topics Can search for nearby events Discussion forums Can find nearby members Ideal for cataloguing large collections Can be accessed from a mobile phone	Cost if user adds over 200 books Can be awkward to navigate Bland colour scheme Processes can be drawn out

Table 3: Pros and Cons of LibraryThing

2.6.3: Requirements established from the comparative analysis

From this analysis, several requirements were established:

- The layout of the website should be uncluttered and clear so the user is not confused and they are able to navigate the site easily.
- A colour scheme should be chosen that is aesthetically pleasing.
- No advertising on the site.
- No cost to register.
- Allow users to search for books and read reviews without creating an account.
- Allow users to create a profile and follow other users.
- Provide personalised recommendations.
- Give users a way of cataloguing books, e.g. lists, with no limits on the number of books they can add to the list.
- Allow users to join discussions
- Ensure that all processes on the site are as streamlined and fast as possible.
- Ensure that the website loads quickly.

2.7: Summary

This chapter has discussed the background research related to knowledge sharing and EWOM, and what encourages users to participate. The comparative analysis investigated the pros and cons of similar websites. Requirements for the website were established from both the background research and from the comparative analysis.

3: Requirements Analysis

3.1: Introduction

This chapter discusses the results of the research questionnaire that was distributed to target users via online forums. The requirements from the questionnaire results are then presented in the form of user stories. The functional and non-functional requirements are then discussed.

3.2: Requirements Research

3.2.1: Research Questionnaire

Following on from the background research and competitive analysis, a questionnaire (Appendix 1) was created in order to research further requirements and to assess the importance of any already established requirements. The questionnaire was created using SurveyMonkey.

The target audience for the website is book lovers and people who already do or could potentially share knowledge in the form of book reviews and ratings. Therefore, to make sure that the questionnaire was completed by members of the target audience, a link to the questionnaire, along with a brief explanation of its purpose and the project, were posted in several forums on Google+, Goodreads and LibraryThing. There was a total of 78 responses to the questionnaire. The results are as follows:

Q1	47.44% of participants read 49 or more fiction books per year, and all participants read at least one book per month.
Q2	Goodreads is the site that the most users would recommend, followed by LibraryThing. Also Goodreads and LibraryThing were the two most used sites.
Q3	The preferred name for the new website is "The Book Network".
Q4	This question asked users to rate certain features in order of importance. The features in order of importance were: <ol style="list-style-type: none">1. Discussion boards2. Personalised lists3. Being able to follow other users4. Receiving recommendations5. Being able to search for books without logging in6. Being able to create a personal profile

	<p>7. Connecting with friends</p> <p>8. Being able to mark other users' reviews as helpful</p> <p>As part of this question, users were also given the option to leave a comment with any other features that are important to them. The comments included:</p> <ul style="list-style-type: none"> • Reading challenges • A tool to find books that the user would rate 4 or 5 stars • For an author, to sell more books • Placing books in categories by tag, keyword or collections • Being able to share via social media • Book information is extremely important • Cataloguing books • Undiscovered authors • Being able to mark books and authors as not interested • Sections for children's books and translations
Q5	<p>This question asked users to rate the importance of several factors when generating book recommendations. The factors in order of importance were:</p> <ol style="list-style-type: none"> 1. Other books from the same author 2. Other books from the same genre 3. Books that friends have enjoyed 4. Highest rated this week <p>Participants were also given the option to leave a comment with any other features that they thought were important when generating book recommendations. The comments included:</p> <ul style="list-style-type: none"> • Filtering out fake reviews • What book users with similar libraries have liked • Interacting with authors • Being able to find something new based on factors like genres, favourite authors, other elements for example, how important humour is in fiction the user reads.

Q6	<p>The most popular colour scheme was Superhero:</p>  <p>The second favourite colour scheme was Flatly.</p>
Q7	<p>This question asked users what encourages them to share their reviews. The factors in order of importance were:</p> <ol style="list-style-type: none"> 1. The enjoyment of helping others 2. Feeling that they are part of an online community 3. Building a reputation within the community 4. Getting helpful ratings from other users 5. Having lots of followers <p>Participants were also given the opportunity to leave a comment with any other factors that encourage them to share their reviews. The comments included:</p> <ul style="list-style-type: none"> • Having their reviews be part of the algorithm to find books they would rate 4 or 5 • Comparing their reviews with others • Preventing other websites from plagiarising their review • Anonymity • Limiting the reviews to the most recent 10 so reviews do not get lost in the shuffle • Easy access to the reviewing page
Q8	<p>Users were given the opportunity to leave any other comments they thought were relevant. The comments included:</p> <ul style="list-style-type: none"> • Book giveaways are popular and ARCs • A simple chat feature as Goodreads is lacking this • Posting short previews of books

Table 4: Questionnaire Results

3.2.2: User Stories

The following user stories illustrate the features and requirements identified by the research questionnaire. Green user stories are those that are an essential requirement for the prototype. Orange requirements would be useful for the initial prototype but are not completely necessary or cannot be completed due to time constraints. Red requirements will not be implemented in the prototype and will be included in future development:

As a user, I want to be able to join discussion boards.	As a user, I want to be able to create personalised lists.	As a user, I want to be able to follow other users.	As a user, I want to be able to receive personalised recommendations.
As a user, I want to be able to search for books without logging in.	As a user, I want to be able to create a personal profile.	As a user, I want to be able to connect with friends.	As a user, I want to be able to give and receive helpfulness ratings
As a user, book information is important to me.	As a user, I want to be able to be able to navigate the website easily.	As a user, I want to be able to talk to other users through a chat feature.	As a user, I want to be able to feel part of an online community.
As a user, I want to be able to build a reputation and enjoy helping others.	As a user, I want to be able to view children's books and translations.	As a user, I want to be able to use tags to categorise books.	As a user, I want to be able to find undiscovered authors.
As a user, I want to be able to interact with authors.	As a user, I want to be able to participate in reading challenges.	As a user, I want to be able to share via social media.	As a user, I want to be able to mark books and authors as not interested.
As a user, I want to be able to see previews of books.	As an author, I want to be able to sell more books.	As a developer, I want to be able to filter out fake reviews.	

3.3: Requirements

Based on the requirements that have been established from the background research, comparative analysis and research questionnaire, the functional and non-functional requirements were defined. The functional requirements describe what the system should do and the non-functional requirements describe how the system works (Eriksson, 2012).

3.3.1: Functional Requirements

The functional requirements identified are as follows:

- The website should provide the user with books from different genres including children's books and translations. It should also provide them with information about the book.
- Users should be able to search for books and read reviews without logging in.
- Users should be able to create a personal profile. They should be able to follow other users and have a newsfeed that shows updates from these users.
- Users should be able to leave reviews and ratings for books. Users should also be able to mark reviews as helpful and receive helpful votes from other users.
- Users should be able to create, join and post in discussions.
- Users should be able to create their own personal lists.
- Users should be given personalised recommendations based on their previous ratings, authors and genres.
- Users should be able to connect with friends using a chat feature.
- Users should be able to build their reputation and feel part of a community.
- There should be no advertising or cost for users.

3.3.2: Non-Functional Requirements

The non-functional requirements identified are as follows:

Usability

From the competitive analysis, it can be seen that there are other sites which offer similar services to users. For this reason, it is essential that the website has excellent usability. If a user visits the site and finds it confusing to use, or ugly to look at they can easily find an alternative website offering a similar experience. Usability is concerned with making the website as user friendly as possible, making sure that everything on the site is easy to find, read and use. Everything on the site must be optimised based on the needs and requirements of the users (Hallam, 2015).

Accessibility

Designing an accessible website will mean that users with disabilities will be able to access the site without any restrictions. It will be important to use contrasting colours to help users who are visually impaired.

Performance

It is important for users to be able to navigate through the website quickly. If a user finds a website to be slow, they are more likely to abandon their session. Increased website performance can increase revenues and reduce costs, and it is also important to consider that Google takes site speed into account when providing search results (Lydon, 2014).

Security

When users are providing their personal details they will expect the website to be secure and that their data will be safe. Building trust will increase user retention. Security failures can damage a websites reputation and result in a decline in user trust.

3.4: Summary

This chapter discussed the results of the research questionnaire and illustrated these results in the form of user stories. The functional and non-functional requirements were then identified, taking into account all requirements identified through the background research, comparative analysis and research questionnaire.

4: Design

4.1: Introduction

This chapter begins by discussing the design methodology used including accessibility and usability guidelines for web design. User Interface design is then discussed and examples are given of how they have been applied during the design process. The website architecture is then defined, including a site map, a discussion of the programming languages used, and an outline of the database design including the ER diagram and shorthand notation of the entities and attributes.

4.2: Design Methodology

W3C (2008) provide Web Content Accessibility Guidelines (WCAG) which provide developers with guidance on how to make web content more accessible for people with disabilities. They provide 4 principles:

1. **Perceivable** – Information and user interface components should be presented to users in a way they can perceive. WebAIM (2016) provide some ways in which information can be made perceivable for people who use screen readers. For example, content on the website was organised with headings which provide an accurate outline of the content.
2. **Operable** – Components and navigation must be operable. An example of this was allowing keyboard navigation instead of mouse navigation as some users are unable to use a mouse.
3. **Understandable** – Users must be able to understand the information and the operation of the user interface. An example of this was to make sure that text was readable and that if there were any errors users are advised how to correct them.
4. **Robust** – the content should be compatible with current and future technologies.

As well as accessibility, it was important to consider usability. ISO (1998) defines usability as the ‘extent to which a product can be used by specified users to achieve specified goals effectively, efficiently and with satisfaction in a specified context of use’ (ISO 9241-11). It was important to take into consideration the goals of a user and how the web design allows them to meet these goals effectively and efficiently, and satisfies the user’s needs.

4.3: User Interface Design

4.3.1: Interface Design Guidelines

During the design of the user interface, it was important to adhere to the 8 Golden Rules of Interface Design (Shneiderman, 2010). These rules are as follows:

1. **Strive for consistency** – This was done by providing users with familiar icons, colours, menus and terminology throughout the website. For example, all buttons appear in the same green colour, except any cancel and delete buttons which are red. All links also appear in the same green colour so that users know they are clickable. Layouts are also consistent throughout.
2. **Enable users to use shortcuts** - In order to reduce the number of user interactions needed, the navigation bar is on all pages of the website so the user can quickly navigate to the home page, to any of the top rated, genre or new releases sections, search for a book, view their newsfeed or view discussions, lists and recommendations. They can also logout and login using the navigation bar.
3. **Offer informative feedback** – Some examples of offering informative feedback are:
 - When a user clicks a button to follow a list, user or joins a discussion, the button will then change to say following or joined so the user knows their request was completed successfully.
 - When a user saves a rating it appears immediately in place of the rating system. The average rating is also automatically updated.
 - When the user adds a book to their list, they are advised that the book has been added.
4. **Design dialog to yield closure** - Any sequence of actions on the website that a user will complete should allow the user to see what stage they are at in the sequence and give them feedback when it is completed, allowing them to then prepare for the next sequence of actions without still considering the previous sequence. Some examples of this are:
 - When a user writes a review and clicks save, the review appears in the list of reviews.
 - When a user posts in a discussion, once they click save, the post appears in the list of discussion posts.

Previous examples of offering informative feedback also apply to this rule.

5. **Offer simple error handling** - The user login and register processes are designed to detect any user errors before they are submitted, and advise the user how to rectify the error. Also when updating their profile, if there are any errors the user is also advised how to fix them.
6. **Permit easy reversal of actions** – Some examples of easy reversal of actions include:
 - Follow/Unfollow users and lists
 - Join/Leave discussions
 - Delete/Edit reviews
 - Delete items from lists

- Delete posts from discussions
7. **Support internal locus of control** – It is important that the user feels that they are in control of the system. The design of the UI will ensure that the user can look at the website and understand how to use it without having to be prompted.
 8. **Reduce short-term memory control** - It is important for the UI to provide the user with cues so as they are not required to remember too much information. This is done with explanatory labels on text fields and making sure the user does not have to remember information to use the system. Also when a user wants to edit their review, the title and text fields are filled with the previous title and text.

4.3.2: Website Mock-ups

Mock-ups were initially created by hand on paper. Once they had been completed they were then recreated in digital form. One of the mock-ups created was the homepage of the website.

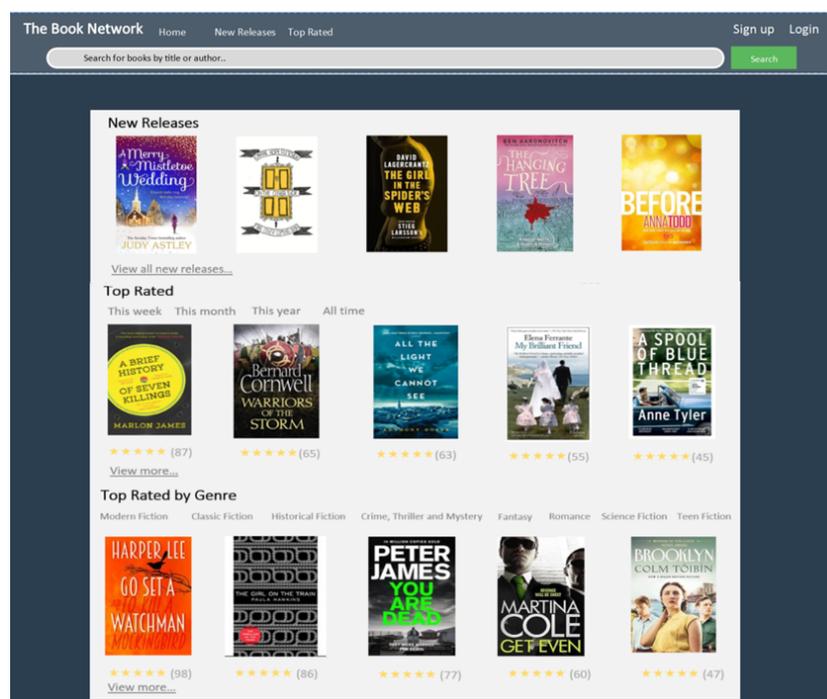


Figure 3: Digital Mock-up of the Homepage

When a user first visits the homepage, without logging in, their goal will be to search for a book, to read reviews of a book, or to browse books. The homepage meets these needs by clearly showing the search bar, the browse dropdown, and showing the user books from different categories. Once a user clicks on a book they are then able to see the reviews for that book.

These mock-ups were essential when implementing the design, as it allowed the developer to have a clear picture of the layout and to assess how the website meets user requirements.

4.3.3: Navigation

It was important that the website was easy to navigate and this was done by providing explanatory links and buttons throughout the website. All links are displayed in the same colour and all buttons are also in this colour except any delete or cancel buttons which are red.

When designing the navigation, the aim was to apply the '3-click rule'. This is the principle that 'access to any feature, or each logical step in a process, should require no more than three clicks' (Usability First, 2016). All of the main features of the website can be accessed directly from the homepage. Any processes on the website also minimise the number of clicks required, for example, to view a list:



Storyboards were created to document how a user would navigate through the website. They also show the buttons and links that the user would click. These storyboards are included in Appendix 2.

4.3.4: Colour Scheme

From the competitive analysis, one of the drawbacks of the both of the websites investigated was the bland colour scheme. Therefore, it is important for the colour scheme to be aesthetically pleasing, without compromising the readability of the website.

During the research phase, the colour scheme 'Superhero' was the most popular colour scheme. However, when the mock-ups were produced and the implementation began it was noted that the colour scheme was quite dark and there were concerns that this darkness would reduce the readability of the website. As a result of this, the second most popular colour scheme was then chosen – Flatly. By changing the colour scheme to Flatly, the contrasting colours will increase the readability of the website.



Figure 4: Flatly Colour Scheme

4.3.5: Layout

The layout of the website is consistent throughout. Each page has the navigation bar at the top, then a container which is where the information is displayed. Additional options appear in the navigation bar when the user logs in.

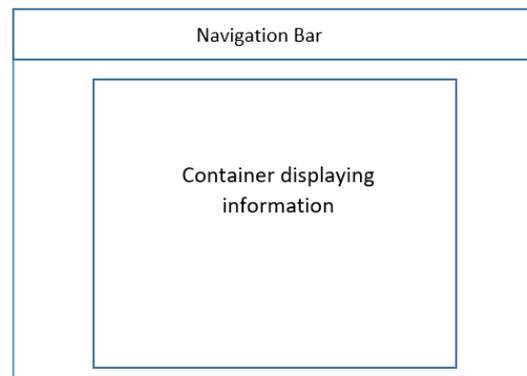


Figure 5: Website Layout

Each book page has the same layout as shown below:

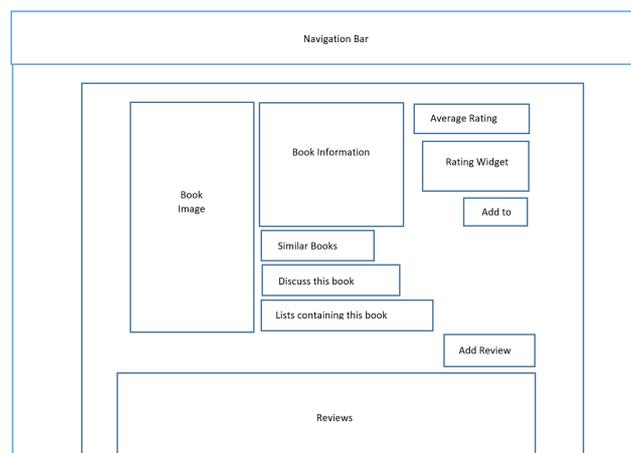


Figure 6: Book Page Layout

If the user is not logged in, they are only shown the book image, book information, similar books button and average rating. The buttons to discuss the book, view lists containing the book, add the book to a list and add a review will not be shown until the user is logged in. The rating widget will also not be shown unless they are logged in.

The layout of the user profile is shown below:

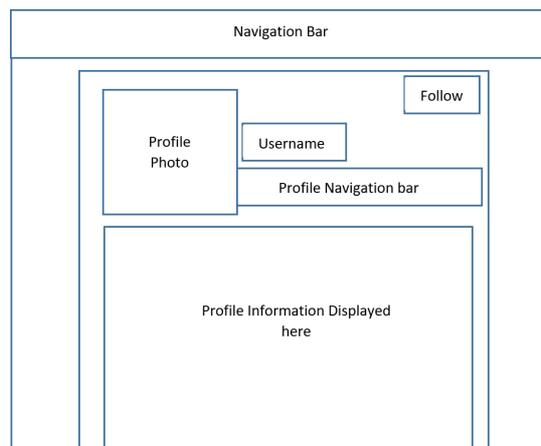


Figure 7: User profile layout

If a user is viewing another user's profile, the follow/unfollow button is shown depending on whether the user already follows the user. If the user is viewing their own profile, the follow button is replaced with two buttons, one to edit their profile and one to delete their profile. Clicking on either of these buttons will open the edit modal or a modal that asks the user if they are sure they want to delete their profile. The profile navigation bar allows the user to move between their activity feed, followers, users they are following, list and following lists. Depending on which option is chosen the information is displayed in the area below the navigation bar.

The layouts for the lists page and the discussions page are identical:

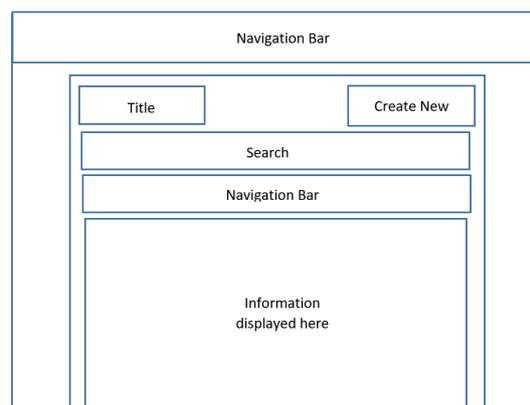


Figure 8: List/Discussion page layout

On the page the title is displayed so this will either show 'Lists' or 'Discussions'. Under this the user can search lists or discussions. The navigation bar then allows the user to navigate between four sections. These are Your Lists/Your Discussions, Lists/Discussions You Are Following, Popular Lists/Discussions and Recommended Lists/Discussions.

4.4: Architectural Design

4.4.1: Site Map

The website needed to be easy to navigate and the site map reflects this. All main features can be accessed from the navigation bar. Within each section of the site, navigating through the different parts of the section is also easy and uncomplicated.

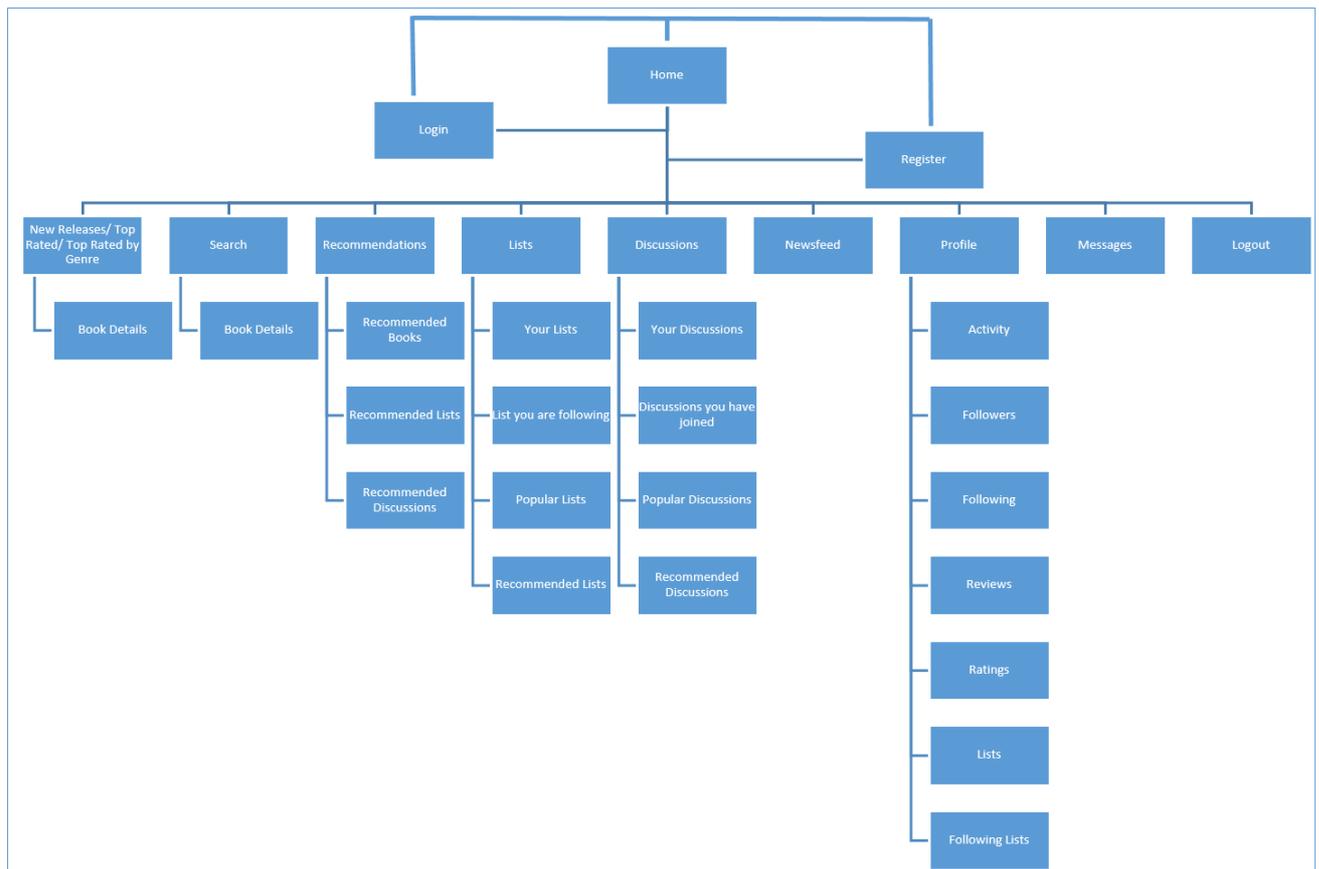


Figure 9: Site Map

4.4.2: Programming Languages

The website has been developed using HTML5, CSS, JavaScript, PHP and SQL programming languages.

On the client side, HTML5 (Hyper Text Markup Language) was used to describe the web documents, using HTML tags which describe the different document content. HTML provides the structure of the web pages and CSS (Cascading Style Sheets) provides the layout of the webpages by describing the style of the HTML documents and how HTML elements should be displayed.

JavaScript was used to program the web page's behaviour and make the web pages more dynamic. AJAX (Asynchronous JavaScript and XML) was also used which allowed parts of the web page to be updated without having to reload the whole page.

On the server side, PHP was used to communicate with the database and generate page content. This execution generates HTML which is then sent to the client (php.net, 2016).

SQL was used to access the database. This included executing queries, retrieving data, and inserting, updating and deleting records.

4.4.3: Database Design

A relational database was used to store all of the data for the website. After determining what data would be stored and the relationships between the data the following ER diagram was constructed, making sure that no many-to-many relationships existed:

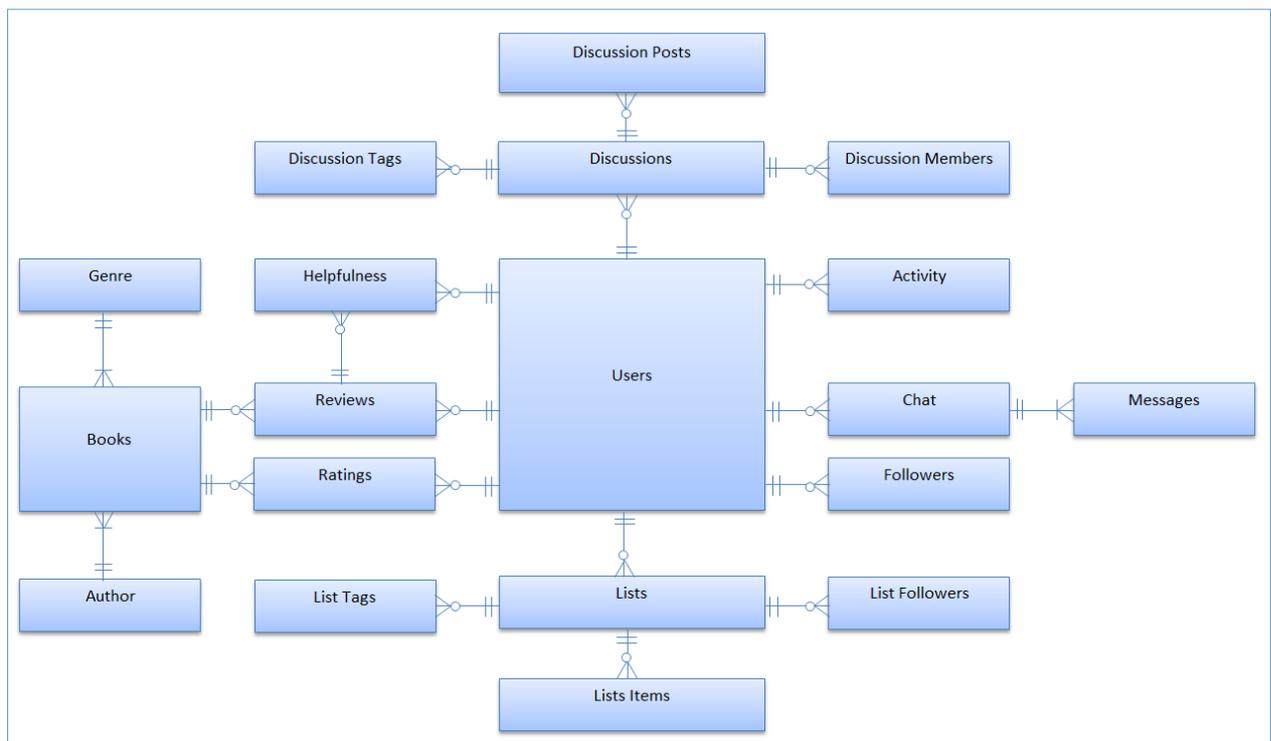


Figure 10: ER diagram

The database consists of 19 entities, each with their own attributes. The shorthand representation of these entities and attributes is as follows:

- Users (user_id, first_name, last_name, email_address, pword, username, profile_photo)

- Books (book_id, author_id*, book_name, book_image, description, avg_rating, genre_id*, date_added)
- Genre (genre_id, genre)
- Author (author_id, author_name)
- Reviews (review_id, user_id*, book_id*, review_title, review_text, helpfulness_score, dt)
- Ratings (rating_id, user_id*, book_id*, rating, dt)
- Helpfulness (help_id, user_id*, review_id*)
- Discussions (discussion_id, creator_id*, discussion_name, no_members, no_posts, dt)
- DiscussionPosts (post_id, discussion_id*, user_id*, post_text, dt)
- DiscussionMembers (join_id, discussion_id*, user_id*, dt)
- DiscussionTags (tag_id, discussion_id*, tag)
- Lists (list_id, creator_id*, list_name, no_followers, noItems, dt)
- ListItems (list_id*, book_id*, dt)
- ListFollowers (listFollowID, user_id*, list_id*, dt)
- ListTags (tagID, list_id*, tag)
- Followers (user1*, user2*, dt)
- Chat (chat_id, user1*, user2*, last_message, dt_created)
- Messages (message_id, chat_id*, message, sender_id*, seen, time_sent)
- Activity (feedID, activityType, follow_user1*, follow_user2*, review_user*, review_id*, rating_user*, rating_id*, list_id*, creator_id*, list_follower*, book_added*, addedBy*, discussion_id*, discussion_creator*, user_joined*, user_posted*, post_id*, dt)

The tables allow all information about users, books, lists, discussions, reviews, ratings and messages to be stored in the database. The activity table is used to create the user's activity feed and newsfeed. The activity feed shows only the users activity and is accessed within their profile and the newsfeed shows their activity and the activity of users, list and discussions they follow. Within the Activity table, apart from the feedID and activityType, all other fields can be null. When an activity is added, based on its activity type, other fields are then filled. For example, activity type 2 is when a user leaves a rating. When this is added to the Activity table, the activityType will be 2, the rating_user field will contain the user_id of the user that left the rating, and the rating_id field will contain the id of the rating. All other fields will be NULL. When this record is then taken from the database, because it is of type 2, the code will then use the values in the rating_user and rating_id field to display the activity.

4.5: Summary

This chapter discussed the design methodology that was followed, including the guidelines that were followed in relation to the accessibility and usability of the website. The user interface design was then discussed including the rules that were followed when designing the interface and how they were applied to the design. The mock-ups of the website, the colour scheme and the layout were also discussed. Within the architectural design, a site map was provided, along with an outline of the programming languages used and what they were used for during the development. Finally, the database design was discussed, including an ER diagram of the database showing the entities and relationships, and a shorthand representation of these entities and their attributes.

5: Implementation

5.1: Introduction

This chapter firstly outlines the hardware used during the implementation stage of the project. The software used is then discussed, including the implementation platform, Webhost and FTP Client, and the database implementation. The implementation of the core functionalities is then discussed, looking at each functionality and detailing how it was implemented, followed by the implementation of the non-functional requirements.

5.2: Hardware

An HP Envy Intel Core i5-6200U CPU 2.3 GHz 8GB Windows 10 64-bit operated Laptop and a Dell Intel Core i7-3770 CPU 3.40GHz 8GB Windows 8 operated desktop computer were used for the implementation of the website.

5.3: Software

5.3.1: Implementation Platform

IntelliJ development platform with an added PHP plugin was used to program the website. This development platform was chosen as it was already available and was a familiar development environment for the developer.

5.3.2: Web Host and FTP Client

The website was hosted on the Dunluce server provided by University of Ulster. This was free to use and easily accessible for the developer.

To FTP the web files to the server, Filezilla was used. This was chosen as the developer had previous experience using it and it was also readily available as it was already installed on the hardware devices to be used for the implementation.

5.3.3: Database Implementation

The Dunluce server provided by University of Ulster also provided SQL databases along with phpMyAdmin to facilitate the management of the database. This was also provided free of charge to the developer and was easily accessed.

5.4: Core Functionalities

During the development there were 13 core functionalities to be implemented. These core functionalities were determined from the functional user requirements that were identified during the research phase.

5.4.1: Provide users with book information

One of the functional requirements was to provide users with books from different genres, and for each book, to provide information about that book. In order to do this, the book information first needed to be stored in the database, then a PHP script needed to be created to access the database and retrieve the book information, then the book information needed to be displayed to the user.

Firstly, within the database, 12 genres were identified. These genres included children's book and translations as specified during the research phase. Once these were added to the 'genre' table, 10 books from each genre were added to the 'books' table. Before each book was added, the author of the book was added to the 'author' table. All book images and descriptions were taken from the WHSmith website.

```
INSERT INTO books (book_id, author_id, book_name, book_image, description, avg_rating, genre_id)
VALUES (1, 1, 'Lord of The Rings: The Return of the King', 'images/001.jpg', 'Concluding the story begun
in The Hobbit, this is the final part of Tolkien\'s epic masterpiece, The Lord of the Rings. Featuring a
striking black cover based on Tolkien\'s own design, the definitive text, and a detailed map of Middle-
earth. The armies of the Dark Lord are massing as his evil shadow spreads ever wider. Men, Dwarves,
Elves and Ents unite forces to do battle against the Dark. Meanwhile, Frodo and Sam struggle further into
Mordor in their heroic quest to destroy the One Ring. Impossible to describe in a few words, JRR
Tolkien\'s great work of imaginative fiction has been labelled both a heroic romance and a classic
fantasy fiction. By turns comic and homely, epic and diabolic, the narrative moves through countless
changes of scene and character in an imaginary world which is totally convincing in its detail. Tolkien
created a vast new mythology in an invented world which has proved timeless in its appeal.', 0, 4)
```

Figure 11: Inserting a book into the database

An 'images' folder was created to store the book images. This allowed the path to the book image to be stored in the database, instead for storing the images as a BLOB type.

On the homepage of the website, books are separated into different categories – new releases, top rated and the top rated books for each genre. Under each category heading 6 books are displayed. Each one of these books on the homepage links to the book details page. A PHP file called newReleasesHome.php was created. This was then included within the main.php file.

Within the newReleasesHome.php file an SQL query was created to fetch the 6 most recently added books from the database.

```
//get most recently added books
if($result = $db->query("SELECT * from books
INNER JOIN author ON books.author_id=author.author_id
ORDER BY date_added DESC LIMIT 6")){
```

Figure 12: SQL query to retrieve new releases

Each book was then displayed to the user:

```
//display books
while($row = $result->fetch_array()){
    $book_id = $row['book_id'];

    echo "<div class='col-sm-2' id='homepageBookDiv'>";
    echo "<table>";
    echo "<tr>";
    echo "<td align='center'>";
    echo "<a href='\"bookDetails.php?book_id=$book_id\"'>";
    ?>

    " height="150" width="100" class="homepageBookImage">

    <?php
    echo "</a></td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td align='center'>";
    echo "<a href='\"bookDetails.php?book_id=$book_id\"'>";
    echo $row['book_name'] ;
    echo "</a>";
    echo "</td>";
    echo "</tr>";
    echo "<tr><td align='center'>";
    echo $row['author_name'];
    echo "</td></tr>";
    echo "</table>";
    echo "</div>";
}
```

Figure 13: Code to display books

This results in the following being displayed to the user:

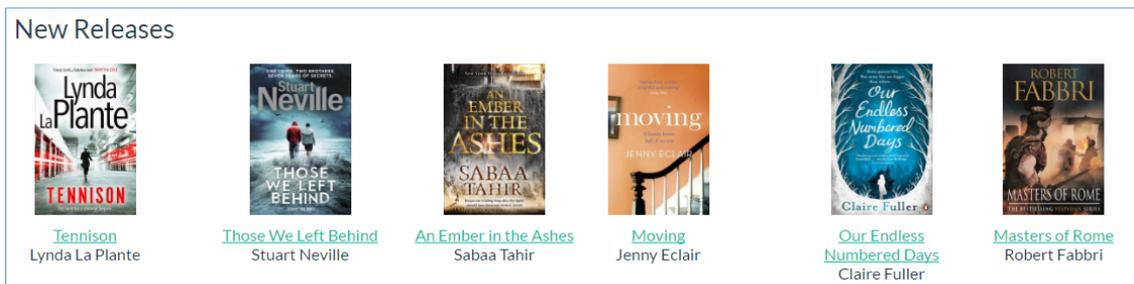


Figure 14: Homepage showing books

When the user clicks on the name of the book or the book image, they are then taken to the book details page – bookDetails.php. The book details page gets the book ID that has been sent to it. This book ID is then used to select all the book details from the database, including the author and genre of the book, using the following SQL query:

```
//get all book details
$result = $db->query("SELECT * from books
                    INNER JOIN author ON books.author_id=author.author_id
                    INNER JOIN genre ON books.genre_id=genre.genre_id
                    WHERE book_id = $book_id");
```

Figure 15: SQL query to get all book details

These book details are then displayed to the user:

```
<div class="col-sm-3">
  " height="400" width="250" id="bookDetailsImage">
</div>

<div class="col-sm-7">
  <h2 id="bookDetailsBookName">
    <?php
    echo $row["book_name"];
    ?>
  </h2>
  <h3 id="bookDetailsAuthor">
    <?php
    echo "<a href=\"authorOther.php?author_id=$authorID&author_name=$authorName\">";
    echo $row["author_name"];
    echo "</a>";
    ?>
  </h3>
  <h4>
    Description
  </h4>
  <?php
  echo $row["description"];
  ?>
  <h4>
    Genre
  </h4>
  <p>
    <?php
    echo $row["genre"];
    ?>
  </p>
</div>
```

Figure 16: PHP code to display book details

This results in the following being displayed to the user:

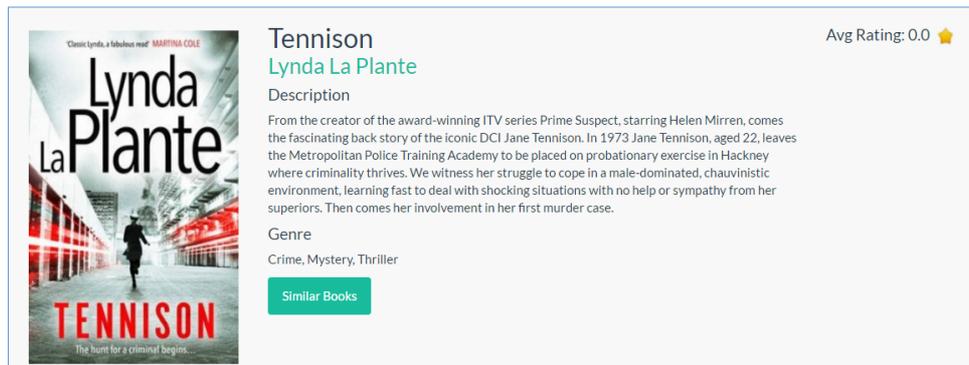


Figure 17: Book details being displayed on web page

5.4.2: Search for books

The second core functionality was that the user should be able to search for books. On the website the user can search for books by title or by author. Within the header.php file, on the navigation bar, a search bar is included:

```
<!-- Search Bar -->
<form class="navbar-form navbar-left" action="search.php" method="get" role="search">
  <div class="form-group">
    <input type="text" name="search" class="form-control" placeholder="Search">
  </div>
  <button type="submit" class="btn btn-default">Search</button>
</form>
```

Figure 18: Search bar HTML

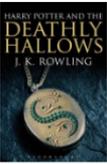
Once a user types in their search term and presses the Search button, the search term entered is then sent to search.php. The search.php file uses the search term within an SQL query to find books that have a title or author like the search term.

```
//search for books where title or author is like search term
$result = $db->query("SELECT * FROM books
                    INNER JOIN author ON books.author_id=author.author_id
                    WHERE book_name LIKE '%".$search."%' OR author_name LIKE '%".$search."%'");
```

Figure 19: SQL query to search the database

If there are no results the user is advised of this. Otherwise, each book returned is displayed to the user:

Search Results: "harry potter"



[Harry Potter and the Deathly Hallows](#)

J. K. Rowling

Avg Rating: 4.5 

[View Details](#)

As he climbs into the sidecar of Hagrid's motorbike and takes to the skies, leaving Privet Drive for the last time, Harry Potter knows that Lord Voldemort and the Death Eaters are not far behind. The protective charm that has kept Harry safe until now is now broken, but he cannot keep hiding. The Dark Lord is breathing fear into everything Harry loves, and to stop him Harry will have to find and destroy the remaining Horcruxes. The final battle must begin - Harry must stand and face his enemy. T...

Figure 20: Search Results

From here the user can click on the name of the book or on the View Details button to go to the book details page.

5.4.3: Read reviews and ratings

The book details page includes a PHP file called review.php. When a book is viewed the book ID is stored in a session variable. This variable is then used in an SQL query to select all the reviews of the book:

```
$reviewsResult = mysqli_query($db, "SELECT * FROM reviews
INNER JOIN users ON reviews.user_id = users.user_id
INNER JOIN ratings ON (reviews.user_id = ratings.user_id AND reviews.book_id = ratings.book_id)
WHERE reviews.book_id='".$mysql_real_escape_string($db, $_SESSION['bookID'])."'
ORDER BY reviews.dt DESC");
```

Figure 21: SQL query to get book reviews

If there are no reviews the user is advised of this, otherwise the reviews are displayed to the user. If the user is not logged in the helpfulness rating is shown but the user cannot rate a review as helpful. Also they are able to see the username of the person who left the review but they are not able to click on this to view the user's profile.

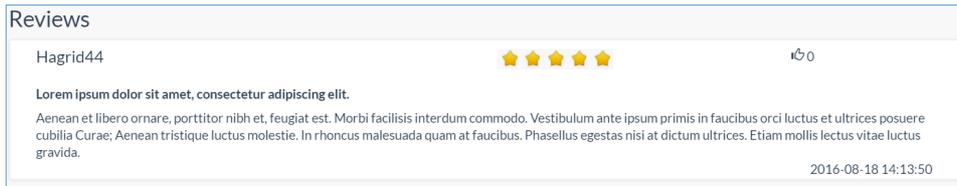


Figure 22: Review for a user who is not logged in

When a user is logged in they can view the users profile and rate the review as helpful using the button.



Figure 23: Review for a user who is logged in

On the book details page the user is also able to view the average rating of each book.

```
<h4 id="averageRating">
  Avg Rating: <?php echo $row["avg_rating"]?>
  
</h4>
```

Figure 24: Code to show average rating

5.4.4: Register and Login

The register and login links are accessed from the navigation bar in the header.

```
<!-- Login-->
<li id="loginLink"><a href="login.php">Login</a></li>

<!-- Register-->
<li id="registerLink"><a href="register.php" id="registerLinkHref">Register</a></li>
```

Figure 25: Login and Register links

When a user clicks the register link they are taken to a page called register.php. This displays a form to the user to fill in their details. The register.php page includes a PHP script called validate.php which checks that the details the user enters are valid. For example, when the user enters their first name:

```
<div class="form-group">
  <label for="registerFormFirstName">First Name</label>
  <input type="text" value="<?php echo $firstName?>" class="form-control"
    id="registerFormFirstName" name="registerFormFirstName" placeholder="First Name">
  <span class="error"><?php echo $firstNameErr?></span>
</div>
```

Figure 26: Form group with input for user's first name

When the form is submitted the validate.php script checks that a first name has been entered and it only contains letters:

```
//check the characters of first name
if(empty($_POST["registerFormFirstName"])){
    $error = true;
    $firstnameErr = "First name is required";
}else{
    $firstName = test_input($_POST["registerFormFirstName"]);
    if(!preg_match("/^[a-zA-Z ]*$/",$firstName)){
        $error=true;
        $firstnameErr = "Only letters allowed";
    }
}
}
```

Figure 27: Validation for first name

If the user has not entered a first name, or the name contains numbers an error message is displayed to the user. The form cannot be submitted without all details entered being correct.

First Name
<input type="text" value="1234"/>
Only letters allowed

Figure 28: Error message is displayed

Once the user has entered all their details correctly, the new user will be created and added to the database:

```
if(!$error && $password == $passwordRe){
    $hashPassword = password_hash($password, PASSWORD_DEFAULT);
    $add = $db->prepare("INSERT INTO users (first_name, last_name, email_address, pword, username)
        VALUES(?, ?, ?, ?, ?)");
    $add->bind_param('sssss', $firstName, $surname, $emailAddress, $hashPassword, $username);
    $add->execute();

    $_SESSION["username"] = $username;

    $getID = mysqli_query($db, "SELECT user_id FROM users
        WHERE username='$username'");
    while($newRow = mysqli_fetch_assoc($getID)){
        $_SESSION["userID"] = $newRow["user_id"];
    };
    ?>
    <script>>window.location="index.php";</script>
    <?php
}
```

Figure 29: New user being created

After the user is added to the database, another query is performed to get the user's ID and store it in a session variable. The user is then redirected to the homepage.

When a user clicks the link to login, they are taken to a page called login.php. This page also includes validate.php. When a user enters their username and password and clicks the Login button, the

validation script will first of all check that both of these fields have been completed. If they have been, it will then check if the username and password are correct.

```

if(!$loginError){
    $query3 = mysqli_query($db, "SELECT * FROM users
                                WHERE username='$loginUsername'");
    $row3 = mysqli_num_rows($query3);

    if($row3 == 1){
        while($row4 = $query3->fetch_array()){
            $userID = $row4["user_id"];
            $hashedPassword = $row4["password"];
            $dbUsername = $row4["username"];
        }
        if(password_verify($loginPassword, $hashedPassword)){
            $_SESSION["userID"] = $userID;
            $_SESSION["username"] = $loginUsername;

            ?>
            <script>>window.location="index.php";</script>
            <?php
        }else{
            $loginError = true;
            $loginPasswordErr = "Incorrect details. Please try again.";
        }
    }
    else{
        $loginError = true;
        $loginPasswordErr = "Incorrect details. Please try again.";
    }
}
}

```

Figure 30: Login validation

If the user has entered the correct details their user ID and username are then stored in session variables and they are redirected to the homepage. If they have not successfully logged in an error message is displayed.

Figure 31: Login error message

5.4.5: User Profile

When a user registers on the site, or has logged in successfully, their username will appear on the navigation bar as a link to their profile.

```

<!-- User profile -->
<li id="name"><a href="viewProfile.php?id=<?php echo $_SESSION["userID"]?>"><b><?php echo $_SESSION["username"];?></b></a></li>

```

Figure 32: User profile link in navigation bar

When a user clicks on this they are taken to their profile. This is created using viewProfile.php.

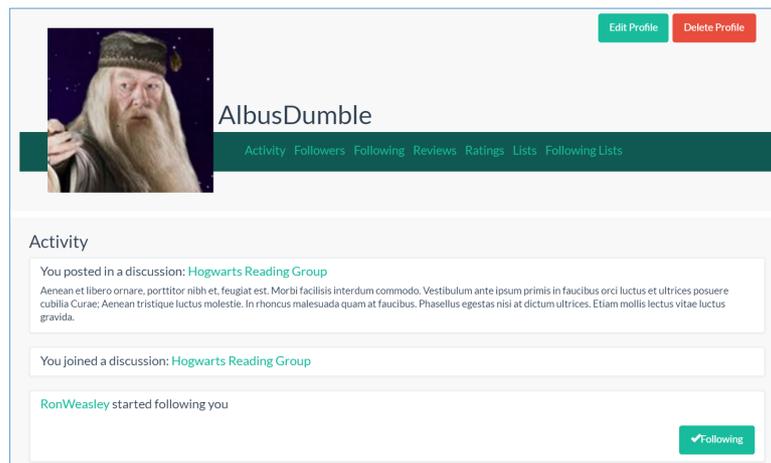


Figure 33: User Profile

The user can view their own profile, as well as other users' profiles, therefore the user ID of the profile to be viewed is sent to the viewProfile.php script. This ID is then used in an SQL query to get the users details from the database. These details are then displayed on the profile.

If the user is viewing their own profile, there will be two buttons at the top of the screen, one to edit their profile and one to delete the profile. If the user clicks on the Edit button a modal will appear where the user can enter new values for their username, password and email address or they can upload a new profile picture. If the user clicks on the Delete Profile button a modal will appear to ask the user if they are sure they want to delete their profile.

If the profile the user is viewing is not their own, instead of the Edit and Delete buttons they are shown a Follow/Unfollow button.

On the profile there is a navigation bar that lets the user move between the activity feed, followers, following, reviews, ratings, lists and following lists. For each of these options, AJAX is used to get the requested data from the database and display it in an empty div with the id "profilePanel".

```
<div class="container" style="...">
  <div class="col-sm-12" id="profilePanel">
  </div>
</div>
```

Figure 34: Empty div to hold AJAX response

When the body of the profile loads the activity feed is automatically displayed. When the showActivity() function is called, it will first of all call another function called createXML() which creates a new XMLHttpRequest:

```

function createXml(){
    var xmlhttp;

    if (window.XMLHttpRequest) {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();
    } else {
        // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    }
    return xmlhttp;
}

```

Figure 35: Function to create a new XMLHttpRequest

This is then returned and stored in the variable xmlhttp within the showActivity() function:

```

//function to show users activity
function showActivity(id, thisUser){
    var xmlhttp = createXml();

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById("profilePanel").innerHTML = xmlhttp.responseText;
        }
    };

    xmlhttp.open("GET", "getProfileActivity.php?user1="+id+"&thisUser="+thisUser, true);
    xmlhttp.send();
}

```

Figure 36: showActivity() function

The response text that is received will be placed inside the empty div on the profile page. A script called getProfileActivity.php is then called, with the profile ID and the ID of the current user being sent to it. In the getProfileActivity.php script, an SQL query is constructed using the ID of the profile being viewed. This query will select any record from the activity table in the database where user has followed someone, been followed, added a review or rating, created, followed or added a book to a list or created, joined or posted in a discussion:

```

//get users activity from db
$getActivity = mysqli_query($db, "SELECT * FROM activity
    WHERE follow_user1 = $user1
    OR follow_user2 = $user1
    OR review_user = $user1
    OR rating_user = $user1
    OR creator_id = $user1
    OR list_follower = $user1
    OR addedBy = $user1
    OR discussion_creator = $user1
    OR user_joined = $user1
    OR user_posted = $user1
    ORDER BY dt DESC");

```

Figure 37: SQL query to get user's activity

If there is no user activity the user will be advised of this, otherwise the activity will be displayed to the user. For each record retrieved, the activityType is examined using a switch statement. Depending on what the activity type is, the code to output the activity will be different.

The reason for sending both the ID of the profile and the ID of the current user, is to make sure that the output is correct depending on if the user is viewing their own profile or someone else's profile.

For example, if the record returned shows that the user created a discussion, first of all another query must be performed to get all of the discussion details. If the user is looking at their own profile activity, the output should be "You created a new discussion" and if they are looking at another users profile it should say "[username] created a new discussion".

```
case 7:
//user created a discussion
$discussionID = $row["discussion_id"];

//get discussion details
$getDiscussionDetails = mysqli_query($db, "SELECT * FROM discussions
INNER JOIN users ON discussions.creator_id = users.user_id
WHERE discussion_id = $discussionID");

while($row0 = $getDiscussionDetails->fetch_array()){
    $disName = $row0["discussion_name"];
    $noPost = $row0["no_posts"];
    $noMembers = $row0["no_members"];
    $username = $row0["username"];
}

//display activity
echo "<div class='col-sm-12 panel panel-default'>";

if($row["discussion_creator"] == $thisUser){
    echo "<h4>You created a new discussion: <a href='discussionPage.php?discussionID=".$discussionID."'>".$disName."</a></h4>";
}
else{
    echo "<h4>".$username." created a new discussion: <a href='discussionPage.php?discussionID=".$discussionID."'>".$disName."</a></h4>";
}

echo "</div>";

break;
```

Figure 38: Code to output user activity

The process is the same for each of the other profile options.

5.4.6: Follow other Users

The next core functionality is that the user should be able to follow other users. As previously discussed a user can follow another user by clicking the follow button on their profile. They can also follow another user from list followers, discussion members and from the followers section of their profile. A JavaScript file called follow.js was created to store the followUser() and unfollowUser() functions.

```

function unfollowUser(userid, thisUser) {
    var xmlhttp = createXml();

    var divID = "followDiv"+userid;

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById(divID).innerHTML = xmlhttp.responseText;
        }
    };

    xmlhttp.open("GET", "unfollowUser.php?user1="+thisUser+"&user2="+userid, true);
    xmlhttp.send();
}

function followUser(userid, thisUser) {
    var xmlhttp = createXml();

    var divID = "followDiv"+userid;

    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {

            document.getElementById(divID).innerHTML = xmlhttp.responseText;
        }
    };
    xmlhttp.open("GET", "followUser.php?user1="+thisUser+"&user2="+userid, true);
    xmlhttp.send();
}

```

Figure 39: Functions to follow and unfollow a user

When a follow or unfollow button is displayed on the website, it will either be on a profile or within a div which is part of a list of users. Each time a follow/unfollow button is displayed, it is within a div with the ID "followDiv" with the ID of the user it is related to also included in the div ID. For example, on a user's profile:

```

<div id="followDiv<?php echo $id?>">

    <?php
    if (mysqli_num_rows($isFollowing) == 1) {
        ?>

        <button class="btn btn-success" id='followButton'
            onclick="unfollowUser(<?php echo $row["user_id"] ?>, <?php echo $_SESSION["userID"]?>)">
            <span class="glyphicon glyphicon-ok"></span>
            Following
        </button>

        <?php
    } else {
        ?>

        <button class="btn btn-success" id="followButton"
            onclick="followUser(<?php echo $row["user_id"] ?>, <?php echo $_SESSION["userID"]?>)">
            Follow
        </button>

        <?php
    }
    ?>
</div>

```

Figure 40: Code showing follow div and follow buttons

If the current user is following this user, a button will be show that says Following, if the user clicks on this it will then change to say Follow. If they are not following a user, the button will say Follow, and once it is clicked it will then say Following.

If, for example, a user visits a profile and they are not following that user, they then click the Follow button. Once this is clicked the followUser() function is called. The ID of the current user and the ID of the user they want to follow are then sent to a file called followUser.php. It is important that the current user is passed as user1 and the user they are following is passed as user2, because in the database user1 follows user2. If they are stored in the wrong order, this would mean that instead of following the user, it would be stored as the user following them.

In the followUser.php file, the details are stored in the followers table of the database. The details are then added to the activity table. Finally the button is echoed back which says Following. If this button is clicked, the user will be unfollowed using a similar process.

```

$user1 = $_GET["user1"];
$user2 = $_GET["user2"];

if(isset($_GET["user1"])) {

    //insert into followers table
    $follow = mysqli_query($db, "INSERT INTO followers(user1, user2)
                                VALUES('$user1', '$user2')");

    //add to activity table
    $addActivity = mysqli_query($db, "INSERT INTO activity(activityType, follow_user1, follow_user2)
                                    VALUES(1, $user1, $user2)");
}

echo "<button class='btn btn-success profilebtn' id='followButton' onclick='unfollowUser(\".$user2.\")'>
      <span class='glyphicon glyphicon-ok'></span>Following</button>";

```

Figure 41: followUser.php

5.4.7: Newsfeed

When a user clicks on the Newsfeed link in the navigation bar they are taken to a page called newsfeed.php. The newsfeed is similar to the profile activity, but instead of only being the activity of one user, it shows the user their activity as well as the activity of any users they follow. It will also show a user if a book is added to a list they follow, or a comment is posted in a discussion they have joined.

```

//get users activity, users they follow activity, lists they follow
$getNewsfeed = mysqli_query($db, "SELECT * FROM activity
                                  WHERE follow_user1 = $currentUser
                                  OR follow_user2 = $currentUser
                                  OR review_user = $currentUser
                                  OR rating_user = $currentUser
                                  OR creator_id = $currentUser
                                  OR list_follower = $currentUser
                                  OR addedBy = $currentUser
                                  OR discussion_creator = $currentUser
                                  OR user_joined = $currentUser
                                  OR user_posted = $currentUser
                                  OR follow_user1 IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR review_user IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR rating_user IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR creator_id IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR list_follower IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR addedBy IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR discussion_creator IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR user_joined IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR user_posted IN (SELECT user2 FROM followers WHERE user1 = $currentUser)
                                  OR (list_id IN (SELECT list_id FROM listfollowers WHERE user_id = $currentUser) AND activityType = 6)
                                  OR (discussion_id IN (SELECT discussion_id FROM discussionmembers WHERE user_id = $currentUser) AND activityType = 9)
                                  ORDER BY activity.dt DESC");

```

Figure 42: SQL query to get newsfeed activity

When the record is returned, the activity type is investigated using a switch statement. Different code will be executed depending on the type of activity. As with the profile activity, the record must be examined to determine the correct output. For example, if a user posted in a discussion, someone they follow posted in a discussion or a post is added to a discussion the user has joined the following code is executed:

```

case 9:
    $disID = $row["discussion_id"];
    $postedID = $row["user_posted"];
    $postID = $row["post_id"];

    $postedOutput = "";

    if($postedID == $currentUser){
        $postedOutput = "You";
    }else{
        $getUsername = mysql_query($db, "SELECT * FROM users WHERE user_id = $postedID");
        $username = $getUsername->fetch_object()->username;
        $postedOutput = "<a href='viewProfile.php?id=".$postedID."'>".$username."</a>";
    }

    $getDiscussionName = mysql_query($db, "SELECT * FROM discussions WHERE discussion_id = $disID");
    $disName = $getDiscussionName->fetch_object()->discussion_name;

    $getPostText = mysql_query($db, "SELECT * FROM discussionposts WHERE post_id = $postID");
    $postText = $getPostText->fetch_object()->post_text;

    echo "<div class='col-sm-12 panel panel-default'><h4>";
    echo $postedOutput." posted in a discussion: <a href='discussionPage.php?discussionID=".$disID."'>".$disName."</a></h4>";
    echo "<p>".$postText."</p>";

    echo "</div>";

    break;

```

Figure 43: Code for newsfeed activity

5.4.8: Add ratings and reviews

When a user visits the book details page, they can either leave a rating or leave a review with a rating. The book details page includes a script called ratingWidget.php. This script will check if the user is logged in. If they are logged in, the script will query the database to see if the user has already rated the book. If they have, their rating is displayed. If they have not already rated the book, the rating widget is shown.

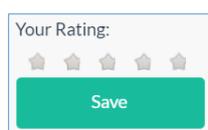


Figure 44: Rating Widget

The user can then choose a rating and once they click save, a JavaScript function called showRating() is called.

```

//function to check for rating and show it or rating system
function showRating(book, user) {
    var xmlhttp = createXml();

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {

            document.getElementById("ratingResult").innerHTML = xmlhttp.responseText;
            $("#ratingForm").empty();
            getAverage(book);
        }
    };

    var rating = getRadioVal(document.getElementById('ratingForm'), "rating-input-1");

    xmlhttp.open("GET", "checkRating.php?q="+rating+"&user="+user+"&book="+book, true);
    xmlhttp.send();
}

```

Figure 45: showRating() function

This function creates an XMLHttpRequest, gets the Radio Value from the rating widget, then sends this along with the user ID to a script called checkRating.php. The response that is received is displayed in place of the rating widget. The average rating of the book is then updated by calling the getAverage() function. This then updates the average rating on the book details page.

The checkRating.php script adds the rating to the database, adds to the activity table and then echoes back a number of stars to represent the users rating.

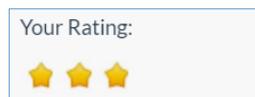


Figure 46: Response is displayed

On the book details page, if the user is logged in, a script called writeReview.php is included. Within this script, if the user is logged in, a query is performed to check if the user has already reviewed the book. If they have not already left a review, the Add a Review button is shown.

When the user clicks this button, a function called showRatingReviewDiv() is called.

```

function showRatingReviewDiv(book, user) {
    var xmlhttp = createXml();

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {

            document.getElementById("ratingDivReview").innerHTML = xmlhttp.responseText;
            $("#reviewDiv").show();
        }
    };

    xmlhttp.open("GET", "checkForRating.php?user="+user+"&book="+book, true);
    xmlhttp.send();
}

```

Figure 47: showRatingReviewDiv() function

This function sends the users ID and the book ID to a script called checkForRating.php. The response that is returned is displayed in the review div. This review div, is hidden until the user clicks the Add a Review button.

The checkForRating.php script firstly checks if the user has rated the book before. If they have the review form echoed back will display this rating along with the inputs for the title and text of the review. If they have not previously rated the book, the form echoed back contains a rating system as well as the title and text inputs.

The buttons on each form have been given different names depending on whether a rating will be given or not. This means that when the Save button is sent, depending on the button name, different code will be executed. For example, if a user has already rated a book and wants to leave a review they will see the following:

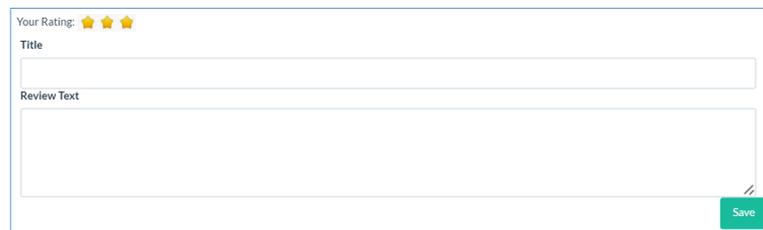


Figure 48: Review form for book already rated

When the user enters their review title and text and presses save the following code is executed:

```
if(isset($_POST["saveReviewNoRating"])) {
    $reviewTitle = $_POST["reviewTitle2"];
    $reviewTitle = mysqli_real_escape_string($db, $reviewTitle);
    $reviewText = $_POST["reviewText2"];
    $reviewText = mysqli_real_escape_string($db, $reviewText);
    $user = $_SESSION["userID"];
    $book = $_SESSION["bookID"];

    //add review to db
    $addReview = $db->prepare("INSERT INTO reviews (user_id, book_id, review_title, review_text, helpfulness_score)
        VALUES (?, ?, ?, ?, 0)");
    $addReview->bind_param('iiss', $user, $book, $reviewTitle, $reviewText);
    $addReview->execute();
}
```

Figure 49: Code to add review to database

The new review is then displayed in the Reviews section of the webpage.

5.4.9: Mark reviews as helpful

One of the functional requirements was that users should be able to rate other reviews as helpful. There were certain conditions to take into account for this functionality:

1. Users who are not logged in cannot rate a review as helpful
2. Users cannot mark their own review as helpful

3. Users can only mark a review as helpful once

This is achieved using the following code:

```
<?php
if(isset($_SESSION["userID"])){
?>

<div class="col-sm-2" id="helpButton"<?php echo $rowReview2["review_id"] ?>">

  <?php
  if ($userid != $_SESSION["userID"]) {

    $thisUser = $_SESSION["userID"];
    $reviewID = $rowReview2["review_id"];

    $checkRated = mysqli_query($db, "SELECT * FROM helpfulness
                                     WHERE user_id = $thisUser
                                     AND review_id = $reviewID");

    if (mysqli_num_rows($checkRated) == 0) {
      ?>

      <button class="btn btn-success" id="helpfulButton" onclick="addHelpfulness(<?php echo $reviewID ?>, <?php echo $thisUser ?>)">
        <span class="glyphicon glyphicon-thumbs-up">
          <?php echo " " . $rowReview2["helpfulness_score"] ?>
        </span>
      </button>

    <?php
    } else {

      echo "<span class='glyphicon glyphicon-thumbs-up' id='helpSpan'></span> " . $rowReview2["helpfulness_score"];
    }
  } else {

    echo "<span class='glyphicon glyphicon-thumbs-up' id='helpSpan'></span> " . $rowReview2["helpfulness_score"];
  }
  ?>

```

Figure 50: Helpfulness code

If the user is not logged in, or they have already rated the review as helpful they are shown a glyphicon and the helpfulness rating. If they have not marked the review as helpful they are shown the helpfulness button.

If the user clicks the helpfulness button the `addHelpfulness()` function is called. This function will send the users ID and the review ID to a script called `addHelpfulness.php`. This script will add the record to the helpfulness table in the database, increase the helpfulness score of the review and echo back the new helpfulness score to the user in place of the button.

5.4.10: Discussions

Within this core functionality, there were three requirements:

1. A user can create a discussion
2. A user can join a discussion
3. A user can post in a discussion

Creating a discussion

When a user clicks on the Discussions link on the navigation bar, they are taken to the main Discussions page – viewDiscussions.php. This page includes a PHP script called createNewDiscussion.php.

When a user clicks on the Create New button a modal appears for the user to fill in the name of the new discussion and up to five tags to describe the discussion. When the user clicks Create Discussion in the modal, the createNewDiscussion.php script will then perform database queries to add the new discussion to the database, then any tags that have been added are added to the tags table in the database. The discussion name is initialised as “Unnamed Discussion” in case the user does not enter a name. They can then edit this in the discussion details page.

```
if(isset($_POST["discussionName"])){
    $discussionName = $_POST["discussionName"];
    $discussionName = mysqli_real_escape_string($db, $discussionName);
}

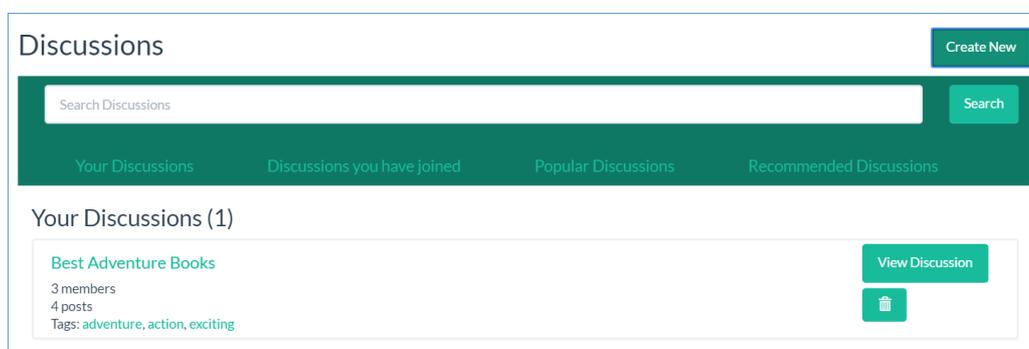
//add discussion to discussions table
$stmt = $db->prepare("INSERT INTO discussions(creator_id, discussion_name)
                    VALUES (?, ?)");
$stmt->bind_param('is', $userID, $discussionName);
$stmt->execute();

//get discussion ID
$stmt = $db->prepare("SELECT * FROM discussions
                    WHERE creator_id = $userID
                    AND discussion_name = '$discussionName'");
$disID = $stmt->fetch_object()->discussion_id;

//if tags have been added, add to discussion tags table
if(!empty($_POST["tag1Dis"])){
    $tag1 = $_POST["tag1Dis"];
    $tag1 = mysqli_real_escape_string($db, $tag1);
    $stmt = $db->prepare("INSERT INTO discussiontags(discussion_id, tag)
                        VALUES (?, ?)");
    $stmt->bind_param('is', $disID, $tag1);
    $stmt->execute();
}
```

Figure 51: createNewDiscussion.php

The discussion has been created and will now show under the “Your Discussions” tab.



The screenshot shows a web interface for a 'Discussions' page. At the top right is a 'Create New' button. Below it is a search bar with the text 'Search Discussions' and a 'Search' button. There are four tabs: 'Your Discussions', 'Discussions you have joined', 'Popular Discussions', and 'Recommended Discussions'. The 'Your Discussions' tab is active, showing a list of one discussion: 'Best Adventure Books'. The discussion card includes the title, '3 members', '4 posts', and tags 'adventure, action, exciting'. To the right of the card are two buttons: 'View Discussion' and a trash icon.

Figure 52: Discussion has been created

Join a Discussion

A user can also view popular discussions and recommended discussions. The user can then click the View Discussion button to view the discussion details. On the discussions page there will be a “Join” button at the top. The user must join a discussion before posting.



Figure 53: Before joining the discussion

Once the user clicks the Join button, a function called `joinDiscussion()` is called. This function sends the user ID and discussion ID to a script called `joinDiscussion.php`.

```
function joinDiscussion(disID, userID) {
    var divID = "joinDiscussionDiv"+disID;
    var xmlhttp = createXml();

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById(divID).innerHTML = xmlhttp.responseText;
        }
    };

    xmlhttp.open("GET", "joinDiscussion.php?id="+userID+"&disID="+disID, true);
    xmlhttp.send();
}
```

Figure 54: `joinDiscussion()` function

Clicking on the join button also calls a function called `showCommentBox()` which will then allow the user to post in the discussion.

The `joinDiscussion.php` script adds a record to the discussion members table in the database. It then updates the number of members and adds to the activity table. This will then echo back the Joined button. When this is clicked, the `leaveDiscussion()` and `hideCommentBox()` functions are called.

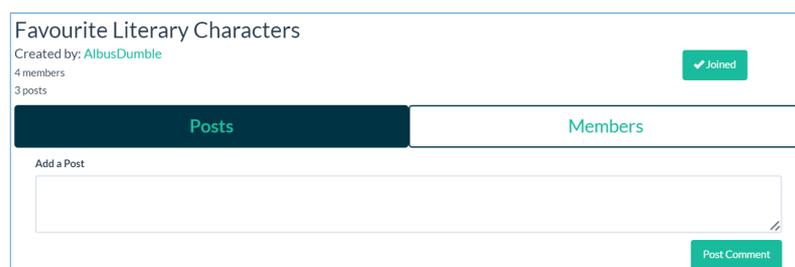


Figure 55: After joining the discussion

Post in a discussion

To post in the discussion the user writes their post in the text box and clicks the Post Comment button. The discussion page includes the script addPost.php. When the Post Comment button is clicked, the addPost.php script adds a new record to the discussion posts table in the database, increments the number of posts in the discussion and adds to the activity table.

```
if(isset($_POST["discussionPost"])){  
  
    $disPost = $_POST["discussionPost"];  
    $disPost = mysqli_real_escape_string($db, $disPost);  
  
    //add post to discussion posts table  
    $insertPost = $db->prepare("INSERT INTO discussionposts (discussion_id, user_id, post_text)  
                                VALUES (?, ?, ?)");  
    $insertPost->bind_param('iis', $disID, $userID, $disPost);  
    $insertPost->execute();  
  
    //increment number of posts  
    $addPost = mysqli_query($db, "UPDATE discussions  
                                SET no_posts = no_posts +1  
                                WHERE discussion_id = $disID");  
  
    //get new post ID  
    $getPostID = mysqli_query($db, "SELECT * FROM discussionposts  
                                WHERE discussion_id = $disID  
                                AND user_id = $userID  
                                AND post_text = '$disPost'");  
    $postID = $getPostID->fetch_object()->post_id;  
  
    //add to activity  
    $addActivity = mysqli_query($db, "INSERT INTO activity(activityType, discussion_id, user_posted, post_id)  
                                VALUES (9, $disID, $userID, $postID)");  
  
}
```

Figure 56: addPost.php

5.4.11: Lists

Within this core functionality, there were three requirements:

1. A user can create a list
2. A user can follow a list
3. A user can add a book to their list

Create a List

This process is similar to the process used to create a new discussion. When the user visits the Lists homepage – viewLists.php, they click on the Create New button at the top of the page. This then opens a modal which allows the user to name the list and add up to 5 tags.

The viewLists.php page includes a script called createNewList.php. When the form in the Create New modal is submitted, the createNewList.php script adds the list to the database, adds any tags that

have been added and then adds to the activity table. After creating the list, it will then appear under the heading “Your Lists”.

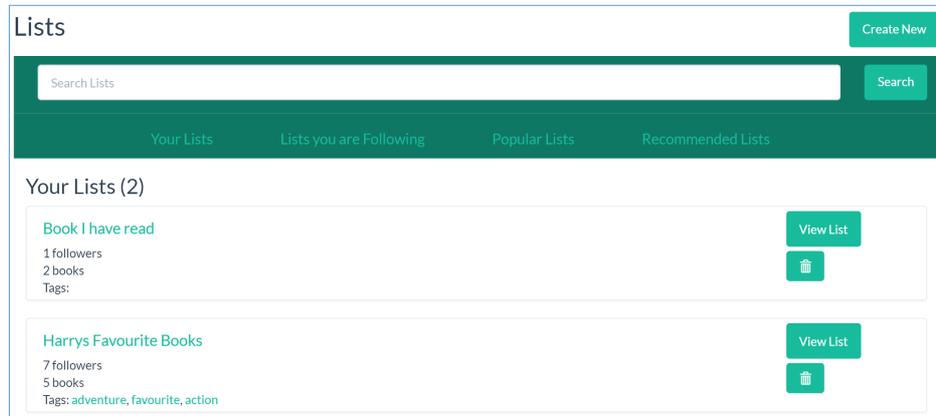


Figure 57: After creating a new list

On the book details page, when a user clicks the Add to button to add a book to a list, they also have the option to create a new list which appears at the bottom of the dropdown. When this is clicked, a modal will appear to fill in the list details. The createNewList.php script is also included in the bookDetails.php page.

Follow a List

Similar to the Discussions page, a user can click on the heading “Popular Lists” or “Recommended Lists”. These will show users either a list of other users lists in order of number of members or a list of recommended lists based on the lists they have created and followed.

When a user clicks on the list name or the View List button, they are taken to the list details page – listDetails.php. On the list details page there is a button at the top that will say Follow. When this button is clicked the method followList() is called.

```
function followList(listID, userID) {  
  
    var divID = "followListDiv"+listID;  
    var xmlhttp = createXml();  
  
    xmlhttp.onreadystatechange = function() {  
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {  
            document.getElementById(divID).innerHTML = xmlhttp.responseText;  
        }  
    };  
  
    xmlhttp.open("GET", "followList.php?id="+userID+"&listID="+listID, true);  
    xmlhttp.send();  
}
```

Figure 58: followList() function

This function sends the ID of the user and the list ID to a script called followList.php. This script then inserts a record into the list followers table in the database, increments the number of followers and adds to the activity table. The new button which shows “Following” is then echoed back and displayed in the correct div. When this button is clicked a function called unfollowList() is called. This will delete the record from the list followers table, decrement the number of followers and remove the related activity. It then echoes back a button that says “Follow”.

Add a Book to a List

On the book details page there is a dropdown button that says “Add to”. When it is clicked a dropdown containing all the user’s lists is shown.

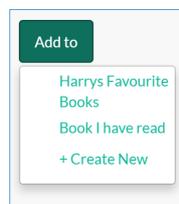


Figure 59: Add to dropdown

On the book details page there is the following code:

```

if(isset($_SESSION["userID"])) {
    ?>
    <button type="button" id="addToButton"
    onclick='getAddToList(<?php echo $_SESSION["userID"] ?>, <?php echo $book_id ?>)'
    class="btn btn-success dropdown-toggle" data-toggle="dropdown"
    aria-haspopup="true" aria-expanded="false">
        Add to
    </button>
    <div class="dropdown-menu" id="dropdownDiv">
    </div>
    <?php
}

```

Figure 60: Add to button code

When the Add to button is clicked, the function getAddToList() is called. This function sends the user ID and book ID to a script called getAddToList.php.

```

function getAddToList(userID, bookID) {
    var xmlhttp = createXml();

    xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            document.getElementById("dropdownDiv").innerHTML = xmlhttp.responseText;
        }
    };

    xmlhttp.open("GET", "getAddToList.php?userID="+userID+"&bookID="+bookID, true);
    xmlhttp.send();
}

```

Figure 61: getAddToList() function

This script will perform a database query to get all of the users list. Each list is then displayed in the dropdown. After the lists have been displayed, another option is added to create a new list. If this option is clicked, the Create New List modal is opened.

```

if(mysqli_num_rows($getLists)==0){
    //show 0 lists
}else{
    echo "<ul style='list-style-type: none;'>";

    //show each list on dropdown menu
    while($row=$getLists->fetch_array()){

        $listID = $row["list_id"];
        $listName = $row["list_name"];

        ?>

        <li onclick="addBook(<?php echo $listID ?>, <?php echo $bookID ?>) " style="..."><a href="#"><?php echo $row["list_name"]?></a></li>

    <?php
    }

    //show create new list in dropdown
    echo "<li data-toggle='modal' data-target='#newListModal' style='margin-top:10px;'><a href='#'>+ Create New</a></li> ";
    echo "</ul>";
}

```

Figure 62: getAddToList.php

If a user clicks on a list name to add a book to that list, a function called addBook() is called. This function sends the list ID and bookID to a script called addBook.php This script adds a record to the list items table in the database, increments the number of items and adds to the activity table. The feedback “Added to List” is then echoed back to the user and appears just under the Add to button.

5.4.12: Recommendations

One of the functional requirements was for the user to receive personalised recommendations for books, lists and discussions. The user can access their recommendations by clicking on the “Recommendations” link. Their recommended lists can also be accessed on the Lists homepage by clicking on the “Recommended Lists” heading, and their recommended discussions can also be accessed via the Discussions homepage, under the heading “Recommended Discussions”. On the recommendations page – recommendations.php – there are three headings, one for each type of recommendation.

```

<div class="container">
    <h2>Recommendations</h2>

    <div class="col-sm-4" style="...">
        <h4><a href="#" onclick="getRecommendedBooks(<?php echo $_SESSION["userID"] ?>)">Books</a></h4>
    </div>
    <div class="col-sm-4" style="...">
        <h4><a href="#" onclick="getRecommendedLists(<?php echo $_SESSION["userID"]?>, 1)">Lists</a></h4>
    </div>
    <div class="col-sm-4" style="...">
        <h4><a href="#" onclick="getRecommendedDiscussions(<?php echo $_SESSION["userID"]?>, 1)">Discussions</a></h4>
    </div>

</div>
<div id="recommendationsDiv"></div>
</div>

```

Figure 63: recommendations.php

When the headings are clicked, different JavaScript functions are called, each function will then call a PHP script which gets the recommendations. The recommendations are then displayed in the empty div with the ID “recommendationsDiv”.

Recommended Books

The users book recommendations are based on books that the user has rated 4 or 5 stars. The SQL query will look at the books the user has rated 4 or 5 and return any book with the same author or in the same genre.

```
$getRecommendedBooks = mysqli_query($db, "SELECT DISTINCT book_id FROM books
WHERE author_id IN (SELECT author_id FROM books INNER JOIN ratings ON books.book_id = ratings.book_id
WHERE ratings.user_id = $userID AND rating >= 4)
OR genre_id IN (SELECT genre_id FROM books INNER JOIN ratings ON books.book_id = ratings.book_id
WHERE ratings.user_id = $userID AND rating >= 4)");
```

Figure 64: SQL query to get recommended books

When displaying the books, the script will check if the user has already rated it, and if they have the book is not shown.

Recommended Lists

The users recommended lists are based on their own lists and the lists they already follow. The SQL statements will retrieve any lists that has a tag that the user has used on their own list, any list which has tag where the tag has been used for a list that the user follows, any list where the list name is the same as any of the users lists or where the lists contains a book that the user has rated 4 or 5 stars.

```
$getRecUserTags = mysqli_query($db, "SELECT DISTINCT lists.list_id FROM lists
INNER JOIN listtags ON lists.list_id = listtags.list_id
WHERE (tag IN (SELECT tag FROM listtags
INNER JOIN lists ON listtags.list_id = lists.list_id
WHERE creator_id = $userID)
OR tag IN (SELECT tag FROM listtags
INNER JOIN lists ON listtags.list_id = lists.list_id
INNER JOIN listfollowers ON lists.list_id = listfollowers.list_id
WHERE listfollowers.user_id = $userID)
OR list_name IN (SELECT lists.list_name FROM lists WHERE creator_id = $userID)
OR lists.list_id IN (SELECT list_id FROM listitems WHERE book_id IN (SELECT book_id from ratings WHERE user_id = $userID AND rating >= 4)))
AND (creator_id != $userID)");
```

Figure 65: SQL to get recommended lists

The last condition is that the user did not create the list. This is to ensure that the recommendations do not contain any lists the user created.

Recommended Discussions

The users recommended discussions are based on their own discussions and the discussions they have already joined. The SQL query returns any discussions with the same tag as any of the user’s discussion tags, any discussion that has a tag which has been added to any discussions the user has joined and any discussions with the same name as the user’s discussions.

```

$getRecDis = mysqli_query($db, "SELECT DISTINCT discussions.discussion_id FROM discussions
INNER JOIN discussiontags ON discussions.discussion_id = discussiontags.discussion_id
WHERE (tag IN (SELECT tag FROM discussiontags
INNER JOIN discussions ON discussiontags.discussion_id = discussions.discussion_id
WHERE creator_id = $userID)
OR tag IN (SELECT tag FROM discussiontags
INNER JOIN discussions ON discussiontags.discussion_id = discussions.discussion_id
INNER JOIN discussionmembers ON discussions.discussion_id = discussionmembers.discussion_id
WHERE discussionmembers.user_id = $userID)
OR discussion_name IN (SELECT discussions.discussion_name FROM discussions WHERE creator_id = $userID))
AND (creator_id != $userID)");

```

Figure 66: SQL to get recommended discussions

The last condition of the SQL statement is that the user is not the person who created the discussion. This ensures the recommendations do not contain any of the user's discussions.

5.4.13: Chat

Another function requirement was a chat feature for the website. When the user clicks on the envelope icon at the top of the screen they are taken to messages.php. When the user goes to this page, if they have any messages these will be shown in a list on the left hand side, and the most recent chat will be displayed. If they have never sent any messages before they will automatically be show the new message window. There are empty divs on the messages page to hold each component.

```

<div class="col-sm-3" id="messagesList">
  <!-- list of messages -->
</div>

<div class="col-sm-9" id="messagesMainWindow">
  <!-- main chat window -->
</div>

<div class="col-sm-9" id="sendForm">
  <!-- message form-->
</div>

```

Figure 67: messages.php

Also on this page, there is a button for the user to create a new message. When the button is clicked a function called `getNewMessageWindow()` is called. The new message window is then displayed to the user. On the new message screen, the user chooses who they want to send a message to by using a dropdown which shows all the users they follow. They can then type their message in the text box and click the Send button.

Once a user has sent the message it is then shown on the messages list on the left hand side of the page.

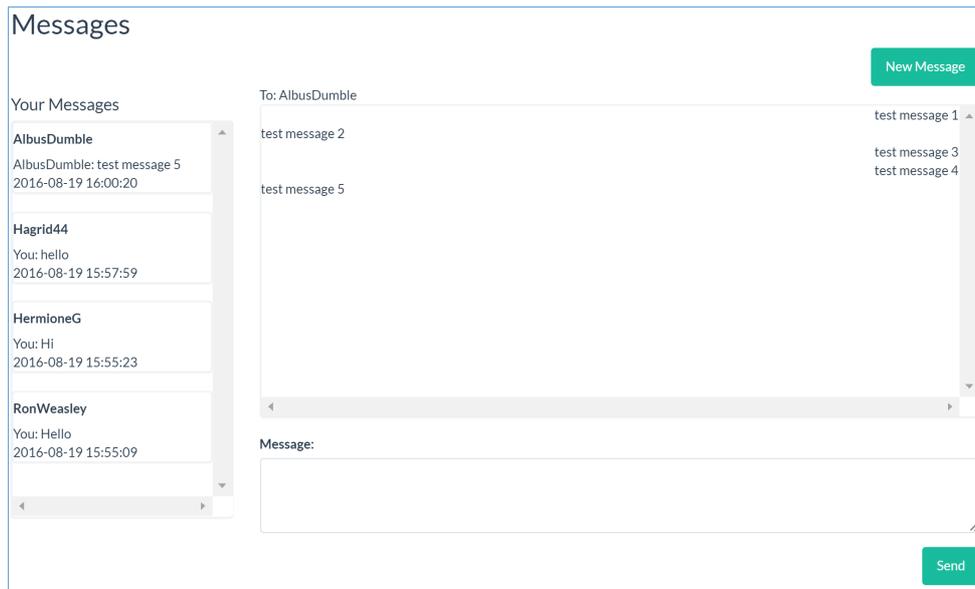


Figure 68: Messages

JavaScript functions are used to check for messages and update the chat window.

```

var updateChat;
var updateList;

//function to refresh chat
function refreshChat(id){
    updateChat = setInterval(function(){
        getChat(id);
    }, 1000);
}

//function to stop refreshing chat
function stopRefresh(){
    clearInterval(updateChat);
}

//function to refresh chat list
function refreshList(){
    updateList = setInterval(function(){
        getChatList();
    }, 1000);
}

```

Figure 69: Messages JavaScript functions

When the messages are loaded, if it exists, the most recent message is show by calling the getChat() function. The refreshChat() function is also called, this will called the getChat() function every second. This chat is then displayed in the main window on the page. By doing this, when a message is sent to the user this will appear straight away in the chat window.

The list of messages is also refreshed every second. The refreshList() function is called when the page loads. This means that if the user receives a message it will then show in the chat list at the top of the list and appear in bold.

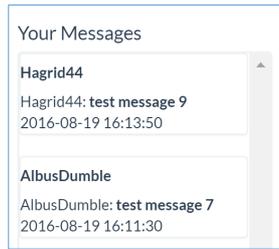


Figure 70: Unread Message

Each time a user clicks on a message from the message list, the stopRefresh() function is called to clear the previous interval and the refreshChat() function is called to start the interval for the chosen chat.

When the getChat() function is called, it then calls a script called getChat.php. If the last message in the chat is from the other user, then all messages in the chat can be marked as seen.

```

if($messageCount == $numMessages){
    //if the last message in the chat is from the other
    //user, mark all messages as seen
    if($row["sender_id"] != $thisUser){
        $markAsSeen = mysqli_query($db, "UPDATE messages SET seen = 1 WHERE chat_id = $chatID");
    }
}

```

Figure 71: Marking messages as seen

If the user has unread messages these will be shown beside the envelope icon in the navigation bar.



Figure 72: Unread messages

Within the header script, a JavaScript function called getUnreadMessages() will call a PHP script to check how many unread messages the user has, then display this number in the navigation bar. Another function called refreshUnread() will create an interval to run the getUnreadMessages() function every second.

```

function refreshUnread(){
    unreadMessages = setInterval(function(){
        getUnreadMessages();
    }, 1000);
}

```

Figure 73: refreshUnread() function

This function is called when the header loads and the user is logged in. This means that if someone sends the user a message, they will be notified immediately.

5.5: Implementation of non-function requirements

5.5.1: Usability and Accessibility

The navigation of the website was implemented whilst keeping in mind that ease of use was extremely important for the user. The accessibility of the website was improved by adding a language tag to the header.php file, providing alt tags for image links and adding a page description to the header page. This improves the accessibility of the website for users who have a screen reader.

5.5.2: Performance

To try and increase the speed of the website, all image files were compressed and all scripts were moved to the bottom of the page to ensure they do not slow down the page rendering. Also, all CSS was added to one stylesheet.

5.5.3: Security

To make the website more secure, steps were taken to avoid SQL injection. To do this, any SQL statements that included input from the user were prepared as follows:

```
$addReview = $db->prepare("INSERT INTO reviews (user_id, book_id, review_title, review_text, helpfulness_score)
                           VALUES (?, ?, ?, ?, 0)");
$addReview->bind_param('iiss', $user, $book, $reviewTitle, $reviewText);
$addReview->execute();
```

Figure 74: SQL query preventing SQL injection

Another feature to improve security was to hash the user's password before storing it in the database. To do this the PHP password_hash function was used. The PHP function password_verify was then used when the user enters their password to login.

5.6: Summary

This chapter discussed the hardware and software that was used during the implementation of the design. It discussed the implementation of the core functionalities, detailing how each of these functionalities was implemented, as well as the steps taken to implement the non-function requirements.

6: Testing

6.1: Introduction

This chapter begins by discussing the in-house testing that was performed by the developer. The website was tested for functionality, cross browsers capabilities and cross platform capabilities. The improvements made as a result of this testing are then discussed. The external testing results are then discussed, along with the improvements that were made as a result.

6.2: In-House Testing

6.2.1: Functional Testing

To test the functionality of the website, a list of functionalities was created. The in-house functionality testing was performed on an HP Envy Intel Core i5-6200U CPU 2.3 GHz 8GB Windows 10 64-bit operated Laptop. This testing determined if the functionalities had been successfully implemented. The results of this testing are detailed in the following table:

Functionality	Result
Navigation Bar links	✓
Register	✓
Login	✓
Search for a book	✓
View book details	✓
Leave a rating	✓
Delete a rating	✓
Leave a review	✓
Delete a review	✓
Edit a review	✓
Mark a review as helpful	✓
Search for a list	✓
Create a list	✓
Delete a list	✓
Follow a list	✓
Unfollow a list	✓
Add a book to a list	✓

Remove a book from a list	✓
Search for a discussion	✓
Create a discussion	✓
Delete a discussion	✓
Join a discussion	✓
Leave a discussion	✓
Post in a discussion	✓
Delete discussion post	✓
Recommendations	✓
Chat	✓
View Newsfeed	✓
View Profile	✓
Edit Profile	✓
Delete Profile	✓

Table 5: Functionality Testing Results Table

The results show that all of the functionalities passed the testing successfully.

6.2.2: Cross-Browser Testing

In order to test the cross-browser capability of the website, an HP Envy Intel Core i5-6200U CPU 2.3 GHz 8GB Windows 10 64-bit operated Laptop was used to test the functionalities on Chrome and Internet Explorer. This was done to make sure that the website functioned correctly across different browsers.

Functionality	Google Chrome	IE
Navigation Bar links	✓	✓
Register	✓	✓
Login	✓	✓
Search for a book	✓	✓
View book details	✓	✓
Leave a rating	✓	✓
Delete a rating	✓	✓
Leave a review	✓	✓
Delete a review	✓	✓
Edit a review	✓	✓

Mark a review as helpful	✓	✓
Search for a list	✓	✓
Create a list	✓	✓
Delete a list	✓	✓
Follow a list	✓	✓
Unfollow a list	✓	✓
Add a book to a list	✓	✓
Remove a book from a list	✓	✓
Search for a discussion	✓	✓
Create a discussion	✓	✓
Delete a discussion	✓	✓
Join a discussion	✓	✓
Leave a discussion	✓	✓
Post in a discussion	✓	✓
Delete discussion post	✓	✓
Recommendations	✓	✓
Chat	✓	✓
View Newsfeed	✓	✓
View Profile	✓	✓
Edit Profile	✓	✓
Delete Profile	✓	✓

Table 6: Cross-Browser Testing Results

The results show that all of the functionalities passed the testing successfully on both Google Chrome and Internet Explorer. This shows that the website has cross-browser compatibility.

6.2.3: Cross-Platform Testing

The next stage was to test the website across different platforms. Although when the website was implemented, the implementation did not take different platforms into consideration, however, in order to test the website rigorously, it was decided that the website should still be tested across different platforms.

The testing was performed in an Amazon Kindle Fire tablet and an iPad Air. As well as functionality being tested, it was also important to test the layout of the website and ensure that it was not affected.

During the testing, all of the functionalities passed the testing successfully on both platforms. However, the layout of some of the pages were affected. The same problems occurred on both of the platforms.

On the book details page, the book details overlapped the cover image.



Figure 75: Layout problems on book details page

On the user profile page, the username and some of the options were obscured by the profile photo. Also all of the options were not on the same line.



Figure 76: Layout issues on profile page

On the list details page, there were several issues. The Edit and Delete buttons were not on the same line, the headings for List Items and Followers were not aligned properly and the View Details button was outside of the panel.

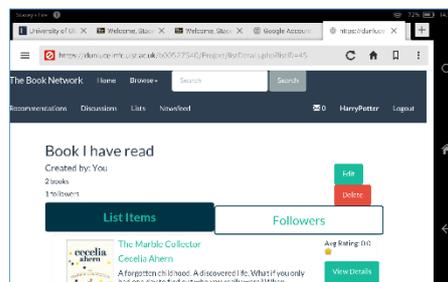


Figure 77: Layout issues on list details page

On the New Releases page, the book description was continuing outside of the panel. Also the view details buttons were outside of the panel.

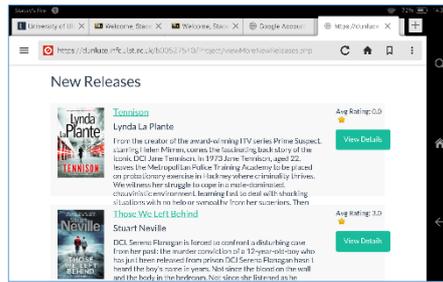


Figure 78: Layout issues on new releases page

The results show that although the website did function across different platforms, the layout was not compatible with these platforms.

6.2.4: Improvements after in-house testing

After identifying that the layout was not compatible across different platforms, the developer adapted the existing code to improve the compatibility.

By changing div classes from col-sm to col-md, this rectified most of the layout issues. The following screenshots show the layout after these changes.

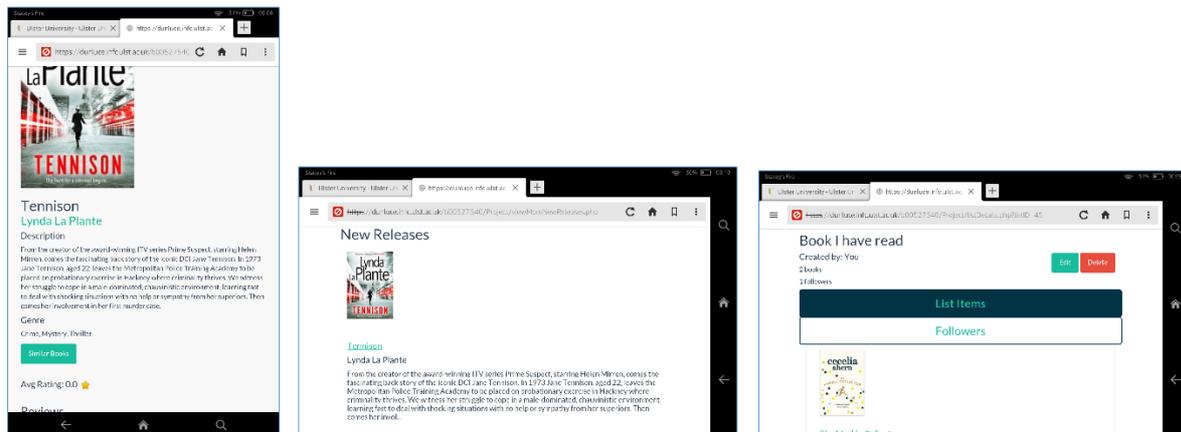


Figure 79: Layouts after changes

Changing the div class did not rectify the layout issues on the user profile page. To fix this issue, the style attribute of the username was set to float right. This moved the username to the right of the profile and allowed it to be seen on the tablet.

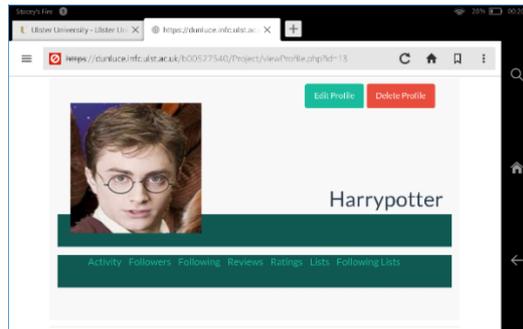


Figure 80: New profile layout

After these improvements, the original cross-platform test was performed again. This time, the test showed that both the functionalities and the layout of the website were compatible across different platforms.

6.3: External Testing

As well as performing in-house testing, a User Testing Questionnaire (Appendix 3) was also created and distributed to eight people. All of the participants were book lovers and all use the internet daily. The questionnaire asked the user to provide details of their hardware, operating system and browser. It then presented the user with thirteen tasks to complete which tested each of the core functionalities of the website. The fourteenth task asked the user to rate the overall presentation and navigation of the website.

Of the eight participants six used a Laptop to complete the testing, and two people used a desktop computer. Six of the participants were using Windows 10 operating system and two were using Windows 7. Six of the participants used Google Chrome and two used Internet Explorer.

After each task, the user was asked to rate the process on a scale of 1 to 5, 1 being “Very Difficult” and 5 being “Very Easy”. When the responses were analysed, the scoring system reflected this scale with 1 point given if the process was rated as “Very Difficult” and 5 points given if the process was rated as “Very Easy”.

The following table shows a description of each task, along with the average score for that task.

Task	Description	Average Score
1	Register as a user	5
2	Search for a book	5
3	Rate a book	5
4	Review a book	4.75
5	Mark a review as helpful	5
6	Follow another user	5
7	Create a new list	4.87
8	Follow a list	4.75
9	Add a book to your list	4.75
10	Create a new discussion	5
11	Join a discussion	5
12	Post in a discussion	5
13	Send a message	4.87
	Overall Average Score:	4.923

Table 7: Table showing user testing results

The final task asked the user what they thought of the overall presentation of the website and how easy they found the navigation. The first question asked the user to rate the presentation from 1 to 5 with 1 being “Very Bad” and 5 being “Excellent”. The overall average score for the question was 4.75.

The second question asked the user to rate the ease of navigation from 1 being “Very difficult” to 5 being “Very Easy”. The overall average score for this question was 5.

The last part of the final task was a question that asked the user if there were any features or improvements they feel should be added to the website. The following responses were collected:

- Notifications
- One search bar to search for books, lists, discussions and people
- Using an apostrophe in a list name caused an error
- Autofill on search bar
- Being able to reply to a post on a discussion
- Having a “Recommended for You” section on the homepage
- Bigger and bolder book icons and titles
- White background is very bright

- Books are not evenly spaced on home page
- The purpose of the Add to button was not clear

As well as these improvements, some positive comments were also left. These included:

- “Very accessible and easy to navigate”
- “Nice design and layout”
- “Nice features including the newsfeed and lists sections”
- “I especially like the background picture”
- “I think it is great”

6.3.1: Improvements after external testing

The comments collecting from the User Testing Questionnaire showed that there were four main improvements that were required. These were then implemented by the developer:

1. Notifications

In order to implement this function, it was decided that the notification function would only notify the user that someone followed them. This functionality could then be extended in future development.

To create a notifications system, a new column was added to the followers table in the database called `seen_By2`. When a record is entered into this table, user1 follows user2. The column `seen_By2` would have the value 0 if user2 has not yet seen the notification that user1 is following them, and will change to 1 when they see the notification.

Similar to the function used to set an interval to check for new messages, a function was created to set an interval to check for new notifications. If there are any new notifications, the icon turns red and displays how many notifications there are. When the user clicks the icon a dropdown appears. This dropdown is populated via AJAX and shows the user who has followed them. The dropdown will show the user their 10 most recent notifications, and any that are new will be in bold, any that have already been seen are shown under a heading “Older”. The user can click on the notifications to view the user’s profile. Once the user has viewed their notifications they are all marked as seen.

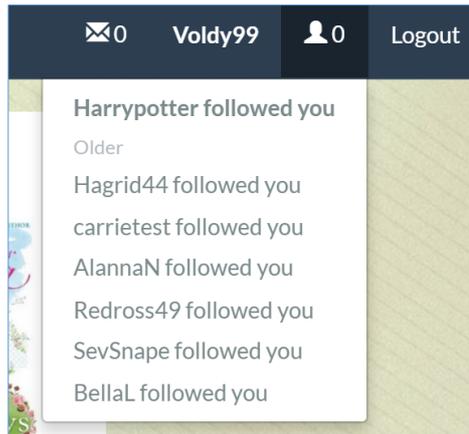


Figure 81: Notifications Dropdown

2. Being able to reply to discussion posts

To implement this function, a new table was created in the database called postreplies:

PostReplies (reply_id, post_id*, user_id*, reply_text, dt)

On the discussion details page, after each discussion post is displayed, the post replies were then displayed and under the replies, a text input was displayed to add a reply to the post. A function was also created to delete a reply. This can be done by the user who wrote the reply or the person who created the discussion. The user can only post replies if they created the discussion or have joined the discussion.

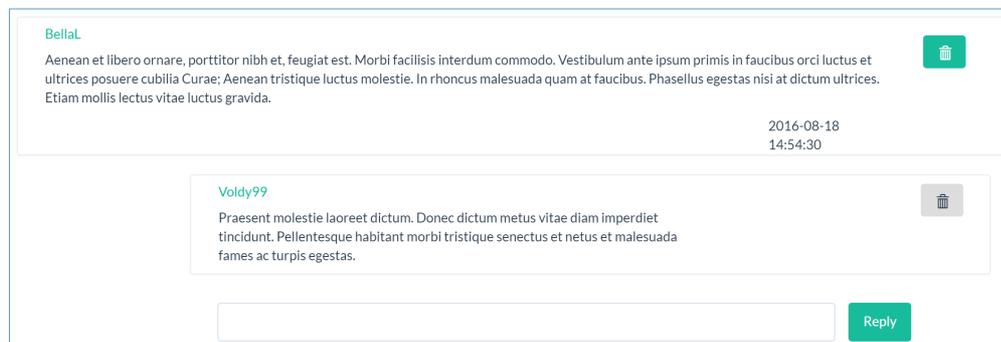


Figure 82: Replying to a discussion post

3. Improved search function

The search function was improved by allowing the user to search for books, users, lists and discussions using the main search bar in the header. The results were then displayed in different sections. If there are no results in a section, it is not shown at all.

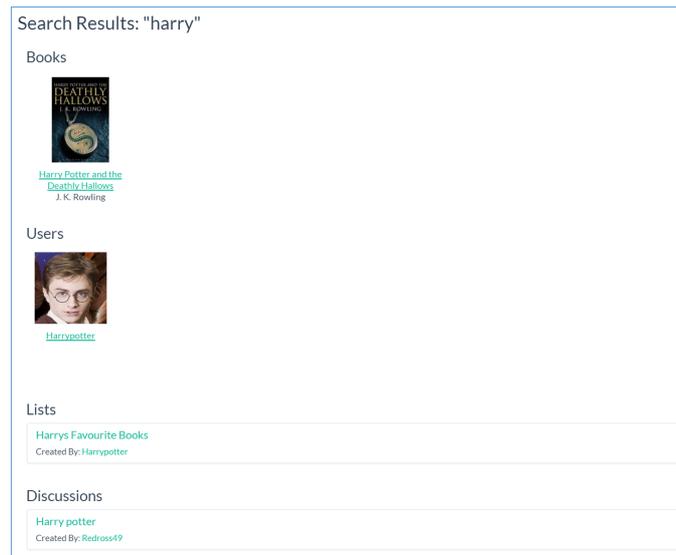


Figure 83: New search results page

Another comment in the feedback mentioned that an autofill option would be useful. This was implemented using AJAX. The onkeyup function of the search input searched the database and returned results into a dropdown under the search bar. The user can click on any of these results to go straight to a book, user profile, list details page or discussion page. The user can also use the tab button to go down the list of results. If there are no suggestions the user is shown "No Suggestions".

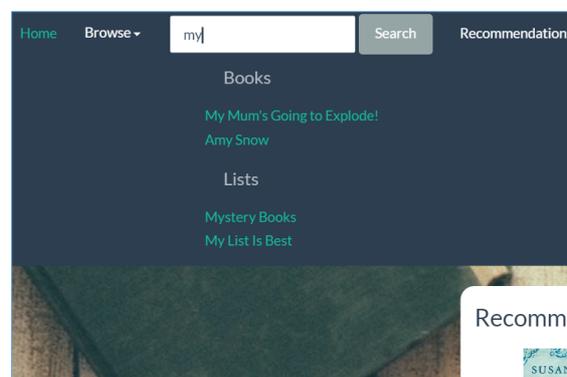


Figure 84: Autofill dropdown

4. Improved Recommendations

The first improvement that was made to the recommendations was to update the SQL queries to get list and discussions recommendations. The SQL statements were updated to that the users recommended lists did not contain any list they had already followed and the recommended discussions did not contain any discussions they had already joined.

Next, as suggested, a Recommended For You section was added to the home page. This only appears once the user is logged in. If they have not rated any books yet, it will not appear. This uses the same SQL statement to get recommended books that is used in the recommended section but only shows 6 books. The user can then click view more to go to the getRecommendedBooks.php page.

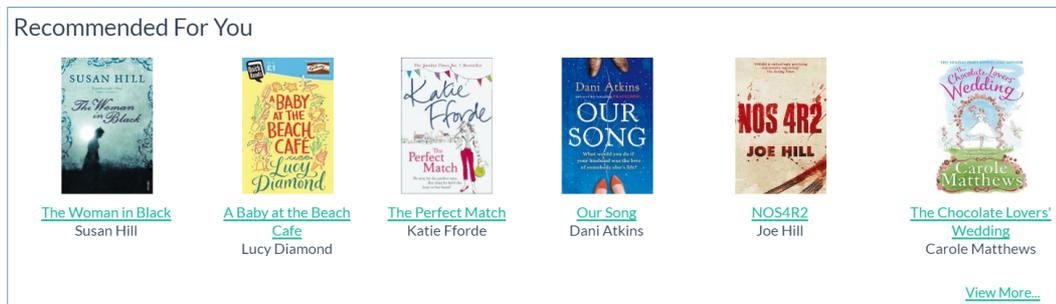


Figure 85: Recommended For You section on homepage

The final improvement for recommendations was to update the layout of the main recommendations page – recommendations.php. Instead of allowing the user to click on headers to move through the different kinds of recommendations, recommendations for books, lists and discussions all appear on the same page. If the user then wishes to view more, they can click on the view more button for the section. This will then take them to a separate recommendation pages for books, lists or discussions.

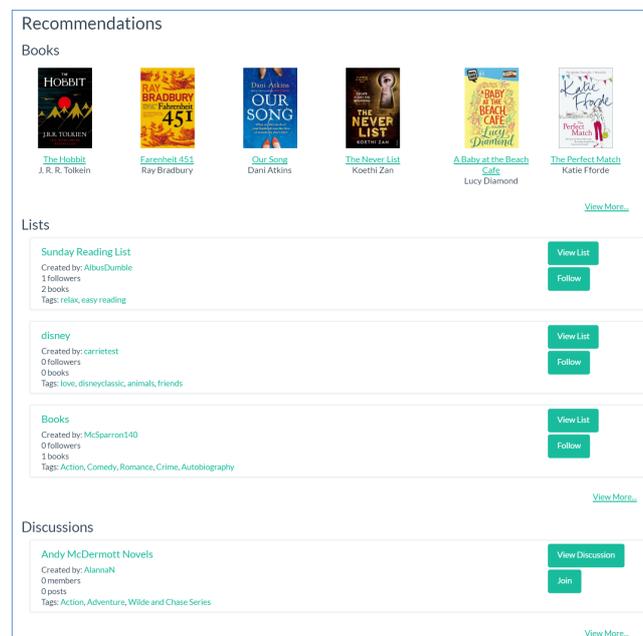


Figure 86: New Recommendations page

6.4: Testing after improvements

After the improvements from both the in-house testing and the external testing were implemented, it was important to test the functionality of the new features, and to test across browsers and platforms. To do this, the functionalities were tested on both Google Chrome and Internet Explorer on a laptop computer, and also on a Kindle Fire tablet.

Additional Requirement	Google Chrome	IE	Kindle Fire
Notifications	✓	✓	✓
Post replies to discussion posts	✓	✓	✓
Improved search function	✓	✓	✓
Improved recommendations	✓	✓	✓

Table 8: Testing results of additional features

These results show that the additional features have passed the tests for functionality, cross browser compatibility and cross platform compatibility.

6.5: Testing non-functional requirements

6.5.1: Usability and Accessibility

Usability has already been addressed during external testing. To test the accessibility of the website an online web accessibility testing tool called WAVE (webAim, 2016) was used. The results of this testing showed that only one accessibility error was present on the home page. This error was due to the button on the navigation bar that is used when the website is displayed on a mobile screen to show the navigation bar as a dropdown.

6.5.2: Performance

To test the speed of the website, an online testing tool was used (pingdom.com). The results of this testing showed that the performance grade of the website was B with a score of 83. The load time for the homepage was shown to be considerably longer than other pages on the website. This may be due to the large number of images that are shown on the homepage, as well as the large background image.

6.5.3: Security

To test the security of the website, an online tool (QUALYS, 2016) was used. This showed that the server's certificate was not trusted. However, ignoring these issues, the website was graded as B for overall security.

6.6: Summary

This chapter discussed the results of in-house testing including cross-browser and cross-platform testing. The improvements from this testing were then implemented. The external testing was then discussed, followed by improvements and how they were implemented. All the new improvements were then tested. Finally, the testing of the non-functional requirements was discussed.

7: Conclusion

7.1: Introduction

The final chapter of this report provides a summary of the project, as well as an assessment of the results and how these outcomes have met the requirements previously established. It then discusses the objectives and whether these have been achieved. Finally, recommendations are made for further development, including the improvement of existing features and new features that could be added in the future.

7.2: Project Summary

Initial research was carried out to investigate the popularity and benefits of online knowledge sharing systems. The research established that, as these websites are so popular and similar websites already exist, the focus would be on improving on current systems.

From the initial research, a problem statement was then created. The problem statement set out the requirement for a prototype online knowledge sharing system to be implemented that allows users to share their book reviews and connect with other users. The website was also required to address any shortcomings in current competitor websites. The aims and objectives of the project were then defined.

The next stage was in depth background research. Online knowledge sharing systems were investigated further, as well as Electronic Word of Mouth and its benefits. The reasons for engaging in EWOM were then investigated to examine the motivation for users to share their knowledge. From this research, several requirements were established.

A comparative analysis was then performed to examine the pros and cons of the two most popular existing competitor websites. From this analysis several requirements were then established.

A research questionnaire was then distributed to members of the target audience via forums on book review sites. The results of this questionnaire were then analysed and presented in the form of user stories. The user stories were assessed in term of importance and time constraints to decide which features and requirements would be implemented in the prototype.

The requirements of the website were then established. The functional requirements were made up of requirements that were established from the background research, comparative analysis and research questionnaire. The non-functional requirements were also established.

The design phase of the website was then completed. When designing the website, accessibility and usability were extremely important. To ensure accessibility, guidelines from W3C were followed. To enhance the usability, Shneiderman's interface design guidelines were also followed. Mock-ups were created and the navigation, layout and colour scheme of the website were also addressed.

The architectural design of the website was then discussed. A site map was used to demonstrate how the user would move between the pages of the website. The programming languages to be used during implementation were then considered. An ER diagram was then constructed to outline the entities and attributes required in the database.

The implementation of the website was then completed. There were 13 core functionalities that were implemented in order to meet the requirements.

During the testing phase, in-house testing was carried out to test the functionality, cross-browser capability and cross-platform capabilities of the website. During the cross-platform testing it was noted that the layout of the website was affected when it was displayed on a tablet. The developer then rectified these issues. External testing was then completed by distributing a user testing questionnaire. The questionnaire asked the user to complete a series of tasks and rate the difficulty of each one. They were then asked to rate the overall presentation and navigation, and to note any features or improvements they thought should be added to the website. From this testing, several additional features were identified. The developer then implemented these additional features. After the implementation of the layout changes and additional features, the website was then tested again. The functionality, cross-browser and cross-platform capabilities were tested, including the new features. These tests were passed successfully.

7.3: Results Assessment

The background research, comparative analysis and research questionnaire were used to establish the requirements of the website. The requirements have been implemented successfully. Testing showed that the website has met all of the functional and non-functional requirements that were defined, and that the website is usable across different browsers and platforms.

7.4: Objectives and Achievements

The objectives that were defined at the beginning of the project have been successfully achieved:

- Background research was conducted to determine why users participate in online knowledge sharing and how to encourage user participation.

- Current book based online knowledge sharing websites were evaluated, identifying their benefits and drawbacks, and how these can be integrated and eliminated respectively.
- A questionnaire was created and distributed to the target user group via online forums and the results were used to determine the user requirements and features for the website.
- The website was designed and implemented according to the requirements that were established, while ensuring that the website met established accessibility and User Interface design guidelines.
- A user testing questionnaire was created and administered and the feedback was used to determine additional features and user requirements.

7.5: Recommendations

The recommendations for improvements for additional features are as follows:

- Add more books to the website. Users could also add books themselves.
- Allow authors to create an Author Profile on the website. This would allow them to promote their books and would allow users to find undiscovered authors. Users would also be able to interact with authors.
- To improve the performance of the website, browser caching should be used to specify a caching policy for page content.
- Allow users to:
 - Share on social media
 - Take part in reading challenges
 - Mark recommendations as not interested
 - Share status updates
 - Reply to reviews
 - Block other users and report content
 - Read book previews
- Add to the notifications system by giving the user notifications for other activity types, for example, if someone has rated their review as helpful, or someone replies to their post.
- Develop a way to filter out fake reviews.

7.6: Summary

The project aimed to design and implement an online knowledge sharing system that allowed users to share their book reviews with others, connect with other users, and that addressed any issues with current similar websites. Background research, a comparative analysis and a research questionnaire

were used to define requirements and design the website, taking into account established design guidelines. The website was then successfully implemented, ensuring that all requirements were met. In-house and external testing were successfully performed, with improvements and additions being implemented. Overall the project has been successful and has met all objectives. Recommendations were given for additional features and improvements which would enhance the website further and increase its chance of success within the current competitive market.

References

- Alavi, M & Leidner, DE 2001, 'Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues', *MIS Quarterly*, vol. 25, no. 1, pp. 107-136.
- Alexa Web Analytics 2015*. Retrieved on January 10, 2016, from <https://www.alexa.com/>.
- Aragon, K 2013, *10 Ways to Speed Up Your Website – and Improve Conversion by 7%*, Retrieved August 10, 2016, from <https://blog.crazyegg.com/2013/12/11/speed-up-your-website/>.
- Bailey, AA 2005, 'Consumer Awareness and Use of Product Review Websites', *Journal of Interactive Advertising*, vol. 6, no. 1, pp. 90–108.
- Batson, CD 1994, 'Why act for the public good? Four answers', *Personality and Social Psychology*, vol. 20, no. 5, pp. 603-610.
- Blazevic, V, Hammedi, W, Garnefeld, I, Rust, R, Keiningham, T, Andreassen, T, Donthu, N & Carl, W 2013, 'Beyond Traditional Word-of-Mouth: An Expanded Model of Customer Influence', *Journal of Service Management*, vol. 24, no. 3, pp. 294–313.
- Cabrera, A & Cabrera, EF 2002, 'Knowledge-sharing dilemmas', *Organisation Studies*, vol. 23, pp. 687-710.
- Cheung, C & Lee, M 2007, 'Understanding User Intention to Continue Sharing Knowledge in Virtual Communities', *ECIS 2007 Proceedings*, Paper 6.
- Cheung, C & Lee, M 2012, 'What drives consumers to spread electronic word of mouth in online consumer-opinion platforms', *Decision Support Systems*, vol. 53, no. 1, pp. 218-225.
- Dabholkar, PA 2006, 'Factors Influencing Consumer Choice of a "Rating Web Site": An Experimental Investigation of an Online Interactive Decision Aid', *Journal of Marketing Theory and Practice*, vol. 14 no. 4, pp. 259–273.
- Dellarocas, C 2003, 'The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms', *Management Science*, vol. 49, no. 10, pp. 1407–1424.
- Eriksson, U 2012, *Functional vs Non Functional Requirements*, Retrieved August 10, 2016, from <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>
- Gehring, EF, Ma, K & Duong, VT 2015, 'What Peer-Review Systems Can Learn from Online Rating Sites', *Lecture Notes in Educational Technology*, pp. 341-350.

Ghosh, B & Scott, JE 2007, 'Social capital in knowledge based business process outsourcing', *Proceedings of the 13th Americas Conference on Information Systems*, pp. 1-12.

Goldsmith, RE & Horowitz, D 2006, 'Measuring Motivations for Online Opinion Seeking', *Journal of Interactive Advertising*, vol. 6, no. 2, pp. 1–16.

Goodreads 2015, Retrieved January 10, 2016, from <https://www.goodreads.com/>.

Hallam, 2015, *Website Usability is Important. Here's Why 2015*. Retrieved January 10, 2016, from <https://www.hallaminternet.com/2015/website-usability-important/>.

Hennig-Thurau, T, Gwinner KP, Walsh G & Gremler, DD 2004, 'Electronic word-of-mouth via consumer-opinion platforms: What motivates consumers to articulate themselves on the Internet?', *Journal of Interactive Marketing*, vol. 18, no. 1, pp. 38-52.

Hoffman, DL & Novak, TP 1996, 'Marketing in Hypermedia Computer-mediated Environments: Conceptual Foundations', *Journal of Marketing*, vol. 60, no. 3, pp. 50–69.

IMDB, 2016, *Press Room: About IMDB*, Retrieved on July 24, 2016, from <http://www.imdb.com/pressroom/about/>.

Internet Live Stats, 2016, *Internet Users*, Retrieved on July 24, 2016, from <http://www.internetlivestats.com/internet-users/>.

ISO, 1998, *ISO 9241-11:1998: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability*, Retrieved on August 10, 2016, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883.

Johnson, J 2013, *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*, Elsevier.

Kim, EEK, Mattila, AS & Baloglu, S 2011, 'Effects of Gender and Expertise on Consumers' Motivation to Read Online Hotel Reviews', *Cornell Hospitality Quarterly*, vol. 52, no. 4, pp. 399–406.

King, RA, Racherla, P & Bush, VD 2014, 'What We Know and Don't Know About Online Word-of-Mouth: A Review and Synthesis of the Literature', *Journal of Interactive Marketing*, vol. 28, no. 3, pp. 167-183.

Ku, Y, Peng, C, Wei, C & Chen, Y 2015, 'Contribute Knowledge Continuously or Occasionally? Determining the Continuous Contributors in Opinion-Sharing Communities', *Lecture Notes in Computer Science*, vol. 9191, pp. 130-141.

Lai, M & Hsieh, P 2013, 'The Decision to Continue Sharing Knowledge in Virtual Communities: The Moderating Role of Knowledge-Sharing Experience and Knowledge Self-Efficacy', *PACIS 2013 Proceedings*, Paper 15.

LibraryThing 2015, Retrieved January 10, 2016, from <https://www.librarything.com/>.

Lin, M, Hung, S & Chen, C 2009, 'Fostering the determinants of knowledge sharing in professional virtual communities', *Computers in Human Behaviour*, vol. 25, no. 4, pp. 929-939.

Lydon, T 2014, *Top 5 Reasons Why Website Performance Matters... and How to Optimize!* Retrieved January 10, 2016, from <https://www.ironistic.com/why-website-performance-matters/>.

Pingdom, 2016, Pingdom Website Speed Test, Retrieved on August 20, 2016, from <https://tools.pingdom.com/#!/cgka4F/https://dunluce.infoc.ulst.ac.uk/b00527540/Project>.

Php.net, 2016, What is PHP?, Retrieved on August 15, 2016, from <http://php.net/manual/en/intro-what-is.php>

QUALYS SSL LABS, 2016, *SSL Report: dunluce.infoc.ulst.ac.uk*, Retrieved on August 20, 2016, from <https://www.ssllabs.com/ssltest/analyze.html?d=dunluce.infoc.ulst.ac.uk&latest>

Quantcast, 2016, *Goodreads.com Traffic and Demographic Statistics*, Retrieved July 24, 2016, from <https://www.quantcast.com/goodreads.com#trafficCard>.

Shen, KN, Yu, AY & Khalifa, M 2010, 'Knowledge contribution in virtual communities: accounting for multiple dimensions of social presence through social identity', *Behavioural Information Technology*, vol. 29, pp. 337–348.

Shneiderman, B and Plaisant, C 2010, *Designing the User Interface: Strategies for Effective Human-Computer Interaction: Fifth Edition*, Addison-Wesley Publ. Co., Reading, MA.

SimilarWeb Analytics 2015. Retrived on January 10, 2016, from <https://www.similarweb.com/>.

Stellman, A & Greene, J 2005, *Applied Software Project Management*, O'Reilly Media, Inc.

Sweeney, JC, Soutar, GN & Mazzarol, T 2008, 'Factors Influencing Word of Mouth Effectiveness: Receiver Perspectives', *European Journal of Marketing*, vol. 42, no. 3/4, pp. 344–364.

TripAdvisor, 2016, *Fact Sheet*, Retrieved on July 24, 2016, from https://www.tripadvisor.co.uk/PressCenter-c4-Fact_Sheet.html.

Tseng, F & Kuo, F 2010, 'The way we share and learn: An exploratory study of the self-regulatory mechanisms in the professional online learning community', *Computers in Human Behaviour*, vol. 26, no. 5, pp. 1043-1053.

Usability First, 2016, Glossary: 3-click rule, Retrieved on August 5, 2016, from <http://www.usabilityfirst.com/glossary/3-click-rule/>.

Usability.gov, 2016, *User Interface Design Basics 2015*. Retrieved January 15, 2016, from <http://www.usability.gov/what-and-why/user-interface-design.html>.

Usoro, A, Sharratt, MW, Tsui, E & Shekhar, S 2007, 'Trust as an antecedent to knowledge sharing in virtual communities of practice', *Knowledge Management Research & Practice*, vol. 5, pp. 199–212.

W3C, 2008, *Web Content Accessibility Guidelines (WCAG) 2.0*, Retrieved on July 24, 2016, from <http://www.w3.org/TR/2008/REC-WCAG20-20081211/>.

Waid, A 2014, *Checklist: 13 Things to Test Before Your Website Launches*. Retrieved January 10, 2016, from <http://www.mediacurrent.com/blog/checklist-13-things-test-your-website-launches>.

Wasko, MM & Tieglund, R, 2004, 'Public goods or virtual commons? Applying theories of public goods, social dilemmas, and collective action to electronic networks of practice', *Journal of Information Technology Theory and Application*, vol. 6, no. 1, pp. 25-41.

WebAim, 2016, *WAVE: Web Accessibility Evaluation Tool*, Retrieved on August 20, 2016, from <http://wave.webaim.org/report#/https://dunluce.infoc.ulst.ac.uk/b00527540/Project/>.

WebAim, 2016, *Designing for Screen Reader Compatibility*, Retrieved July 24, 2016, from <http://webaim.org/techniques/screenreader/>.

Wu, W, Lin, C, H, B & Yeh, R 2009, 'Interpersonal trust and knowledge sharing: Moderating effects of individual altruism and a social interaction environment', *Social Behaviour and Personality: an international journal*, vol. 37, no. 1, pp. 83-93.

Appendices

Appendix 1: Research Questionnaire

A new online book review system

Welcome to My Survey

Thank you for participating in my survey!

I am currently studying a MSc in Professional Software Development at University of Ulster Coleraine and as part of my dissertation I am designing and implementing an online book review site. This site will allow users to write reviews and rate books. Users can also create their own personal profile and personalised lists, for example, favourite books or books to read. They will also be able to interact with other users on discussion forums. Users can add friends and follow other users lists that they like. It will also provide personalised recommendations for users. The aim of the site is to combine a book review sharing website with a social network.

I am currently in the design stage of development and I would like to get some feedback from book lovers and readers on the design and features of the site.

The data collected will be used solely for the purposes of my research project. It will not be shared with any third parties and will be erased after analysis has been done.

I will use the responses provided to decide the name and colour scheme of the website. The data will also be used to understand what are the most important features for users, what encourages users to share reviews online and what are the most important factors when generating personalised recommendations.

Thank you!

Next

A new online book review system

1. How frequently do you read fiction books?

1-12 per year (up to 1 per month)

13-24 per year (between 1 and 2 per month)

25-36 per year (between 2 and 3 per month)

37-48 per year (between 3 and 4 per month)

49+ per year (more than 4 per month)

Prev Next

A new online book review system

2. Please indicate if you are aware of, have used and would recommend the following websites:

	Aware of	Have used	Would recommend
Goodreads	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LibraryThing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Shelfari	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The Reading Room	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
aNobii	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Riffle	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Book Likes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Revisish	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Book Browse	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Book Trib	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Prev

Next

A new online book review system

3. For the new website please rank these websites names in order of preference (1 being best, 5 being worst):

⋮	Online Book Database
⋮	Online Book Network
⋮	Book Community
⋮	BookLoversOnline
⋮	The Book Network

Prev

Next

A new online book review system

4. Please rate the importance of the following features:

	Extremely Important		Moderately Important		Not important at all
Being able to connect with friends	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Discussion Boards	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Being able to search the book database without logging in or signing up	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Being able to rate other users reviews/ mark their review as helpful	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Create a personal profile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Create personalised lists of books e.g. Favourites, Currently Reading	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Being able to follow other users lists and review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Receive recommendations based on interests and past reviews	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Any other important features?

Prev

Next

A new online book review system

5. How important would each of the following be when receiving personalised recommendations:

	Extremely Important		Moderately Important		Not important at all
Other books from the same author with good reviews	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Books that friends have enjoyed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Books from the same genre with good reviews	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Highest rated this week	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Anything else that would be important when generating recommendations?

Prev

Next

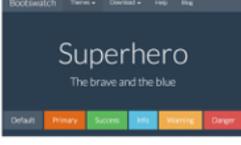
A new online book review system

6. Please choose the colour scheme you prefer:

 Slate
Shades of gunmetal gray

 Darkly
Flatly in night mode

 Flatly
Flat and modern

 Superhero
The brave and the blue

Prev

Next

A new online book review system

7. What would encourage you to share your own book reviews? Check all that apply

- Getting helpful ratings from other users
- Having lots of followers
- Feeling that you are part of an online community
- Building a reputation within the community
- The enjoyment of helping others

Anything else that would encourage you to post reviews?

Prev

Next

A new online book review system

8. Please feel free to share any other comments:

9. If you would like to receive updates about the progress of the website or you would like to test the site when it is ready please include your email address below. This will not be shared with any other parties.

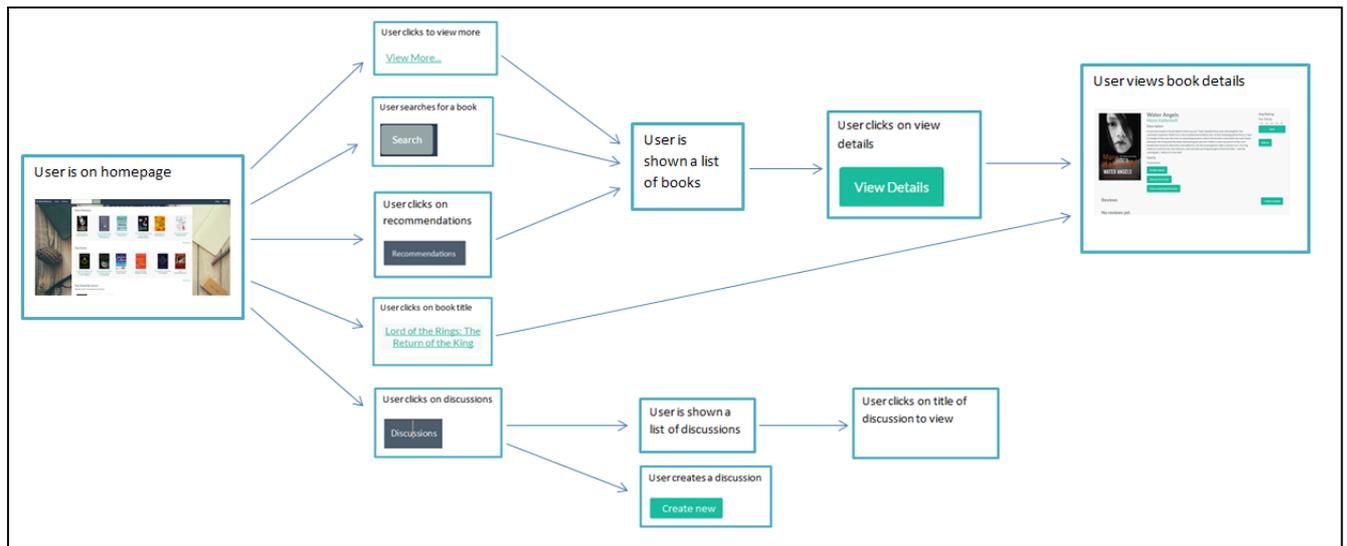
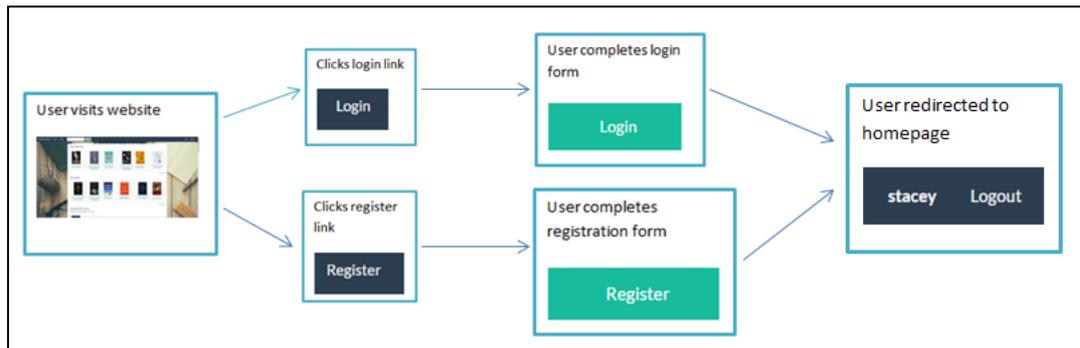
Email Address

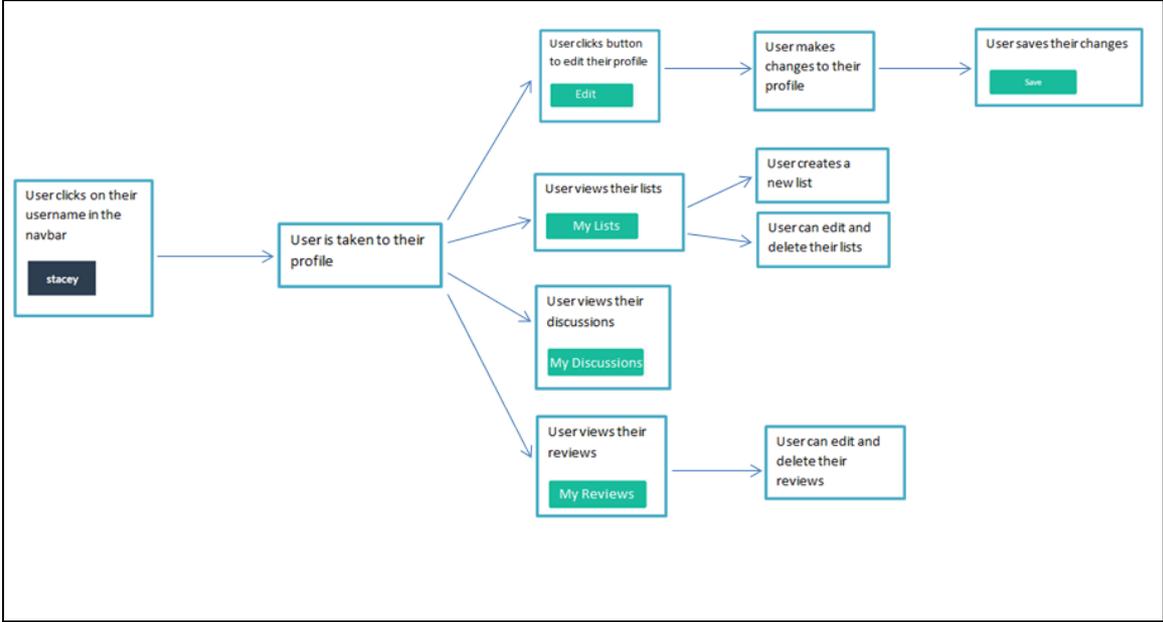
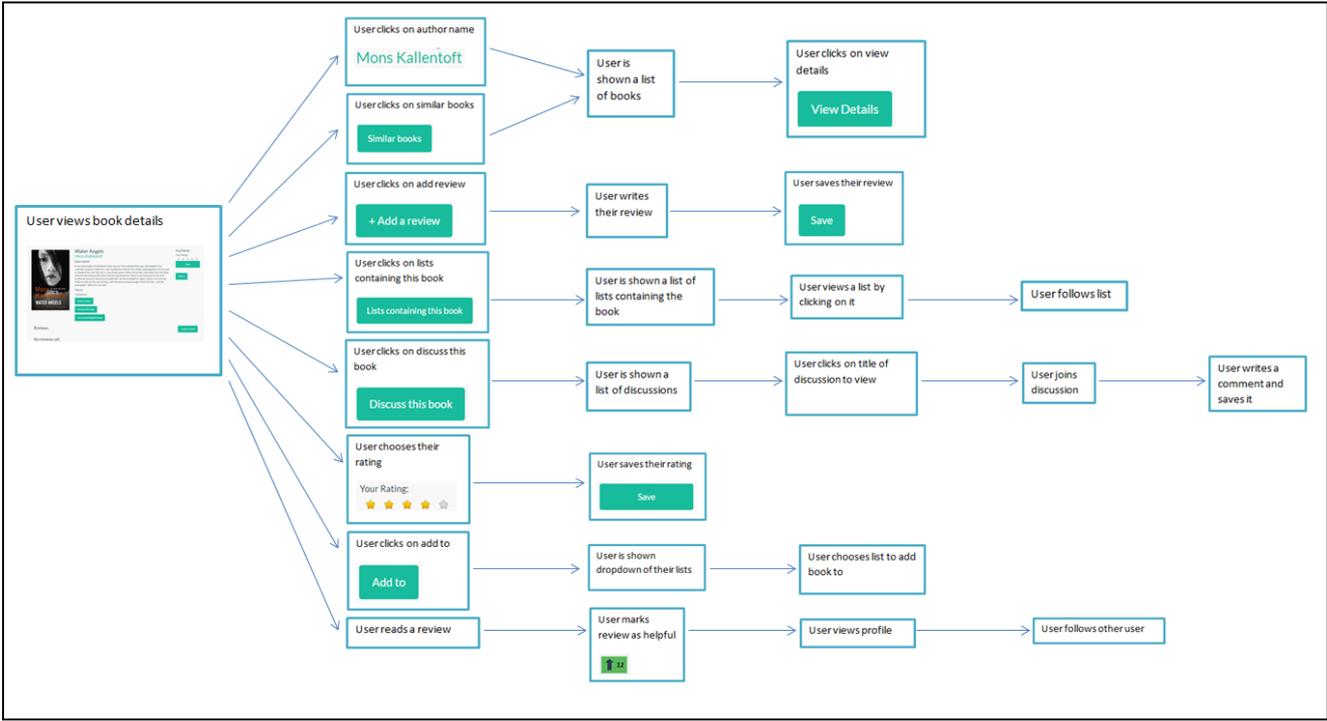
This is the end of the questionnaire. Thank you for your participation!

Prev

Done

Appendix 2: Storyboards





Appendix 3: User Testing Questionnaire



User Testing Questionnaire

Name: Click or tap here to enter text.

Hardware: Click or tap here to enter text.

Operating System: Click or tap here to enter text.

Browser: Click or tap here to enter text.

The following tasks are designed to test the core functionalities of the website. In each task, you will be asked to perform a series of actions on the website. You will then be asked to answer some questions about these processes.

Please visit the following web address:

<https://dunluce.infoc.ulst.ac.uk/b00527540/Project/main.php>

Task 1: Register as a user

Please complete the following steps.

1. Click on the "Register" link on the navigation bar at the top of the page.
2. Complete the registration form and click the Register button.
3. You should be redirected to the home page and your username should be displayed in the navigation bar at the top of the screen.

Please rate the registration process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 2: Search for a book

Please complete the following steps:

1. On the home page, enter the following term in the search bar "harry potter". Click the search button.
2. This will take you to the results page which should display one result.
3. Click on the view details button, or on the book name.
4. This will take you to the book details page.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 3: Rate a book

Please complete the following steps:

1. On the book details page, choose a rating by clicking on the stars. Click save.
2. This should update to show your rating.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 4: Review a book

Please complete the following steps:

1. Staying on the book details page, click the "Add a review" button. This will open a form. It should display the rating you have just left.
2. Fill in the review form and click Save.
3. Your review will then appear at the top of the list of reviews.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

<input type="checkbox"/>				
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Task 5: Mark a review as helpful

Please complete the following steps:

1. Staying on the same book details page, in the reviews section, find a review other than your own review.
2. To rate the review as helpful, click the green button with a thumb.
3. This should update to show that you have rated this review as helpful. The button will disappear and the new helpfulness rating will be shown.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 6: Follow another user

Please complete the following steps:

1. Staying on the same book details page, find a review that is not your own.
2. Click on the username of the user who wrote the review.
3. This will take you to their profile.
4. At the top of their profile, click the Follow button.
5. This should then change to show you are following the user.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 7: Create a new list

Please complete the following steps:

1. Click on the “Lists” link on the navigation bar at the top of the page. This will take you to the Lists homepage.
2. Click on the “Create New” button at the top of the page. This will open a pop up containing a form.

3. Fill in a list name and add up to 5 tags for your list.
4. Click Create List.
5. The new list will now show under the heading “Your Lists”.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 8: Follow a list

Please complete the following steps:

1. Staying on the Lists page, click the heading “Popular Lists”. This will show a list of lists based on the number of followers.
2. Choose one of the lists and click the “View List” button, or click on the list name. This will take you to the list details page.
3. At the top of the list details, click on the Follow button. This will change to show that you are now following this list.
4. Click on the List link in the navigation bar at the top.
5. Click on the heading “Lists you are Following”.
6. The list you have just followed should now be shown.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 9: Add a book to your list

Please complete the following steps:

1. Click on the “Home” button on the navigation bar at the top of the page.
2. One the home button, choose any book and click on the cover image or the book name.
3. This will take you to the book details.
4. Click the “Add to” button. This is under the star rating system.
5. A dropdown menu will appear. This should show the list you have just created.
6. Click this list name to add the book to your list.

7. “Added to List” should now appear under the Add to button.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 10: Create a new discussion

Please complete the following steps:

1. Click on the Discussions link in the navigation bar at the top of the page. This will take you to the Discussions homepage.
2. The layout of this page is the same as the Lists homepage.
3. Click on the Create New button at the top of the page. Enter a name for the discussion and add up to 5 tags. Click Create Discussion.
4. This discussion should now show under the Your Discussions heading.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 11: Join a discussion

Please complete the following steps:

1. Staying on the Discussions page, click on the heading “Popular Discussions”.
2. Choose one of these discussions and click the “View Discussion” button.
3. This will take you to the discussion page. Above the posts, it will say “You must join the discussion to post comments”.
4. Click on the Join button at the top of the page.
5. The button will then update to show you have joined the discussion. Also a text box will appear at the top of the discussions to allow you to post comments.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 12: Post in a discussion

Please complete the following steps:

1. Staying on the same page, write a post in the text box.
2. Click the Post Comment button.
3. The post should now show as part of the discussion.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 13: Send a message

Please complete the following steps:

1. Click on the envelope in the navigation bar at the top, beside your username.
2. This will take you to the Message page.
3. At the top of the message pane, you can choose who you want to send a message to. As you have only followed one user, this is the only user you can message.
4. Type a message in the text box under the message pane.
5. Click Send.
6. Your message will now be displayed in the left hand column under your messages.
7. Click on this message to view it.

Please rate this process:

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Task 14: General Questions

How would you rate the presentation of the website as a whole?

Very Bad	Bad	Average	Good	Excellent
<input type="checkbox"/>				

How would you rate the ease of navigation?

Very Difficult	Difficult	Average	Easy	Very Easy
<input type="checkbox"/>				

Are there any features or improvements you feel should be added to the website?

Click or tap here to enter text.

Please click the Logout button in the navigation bar at the top of the page to end the testing session.

Thank you for your participation.