A Digital Guide to Being a Looked After Child

Final Report CM3203 One Semester Individual Project 40 credits Caitlyn Powell Supervisor: Dr Catherine Teehan Moderator: Dr Martin Caminada

Abstract

Being a child or young person in care can be a daunting experience as lots of different adults and agencies are involved at different stages. Child Friendly Cardiff and Children's Services have discussed the need for a guide which outlines the role of these adults and agencies for children and young people in care in Cardiff. A young person in care may move home and change social workers and personal assistants many times, and a document outlining their rights and what they can expect during their time in care would be extremely beneficial in helping them understand what is happening and ease their transition.

Young people in care need easy access to key information about their placement, their next steps and the support available to them, and this information needs to be independently accessible to them regardless of their current support system. This project aims to work with young people previously in care to create a digital guide for young people currently in care to access on any device, as and when they need it.

Acknowledgements

I would like to thank my supervisor Dr Catherine Teehan and my moderator Dr Martin Caminada for their help and support, as well as Alfie Rowett, whose related work with Children's Services meant that we worked together to attend user research meetings and discuss implementation of our projects.

I would also like to thank Thomas Pughsley from Child Friendly Cardiff and Samantha Anderson from NYAS for their help in facilitating this project and the many meetings we've had, along with the care-experienced young people we've spoken to whose insight was invaluable.

Finally, I'd like to thank Toni White for her unwavering support throughout this project.

Table of Contents

Abstract	2
Acknowledgements	3
Table of Figures	6
Table of Tables	6
1. Introduction	7
1.1. Project Aims	7
1.2. Project Scope	7
1.3. Outcomes	7
2. Background	9
2.1. Existing solutions	9
2.2. User Research 1	0
2.3. Project justification	0
2.4. Constraints 1	1
3. Approach1	2
3.1. Technologies for development1	2
3.1.1. Web application frameworks: Django vs Flask1	2
3.1.2. Web application tools1	2
3.1.3. Data storage1	13
3.1.4. Version Control	10
	12
3.1.5. Hosting	.3
3.1.5. Hosting	L3 L3 .3
 3.1.5. Hosting	.3 .4
 3.1.5. Hosting	.3 .3 .4
 3.1.5. Hosting	13 13 .4 .4
 3.1.5. Hosting	13 13 14 .4 .5
 3.1.5. Hosting	13 13 14 .4 .5 .5
 3.1.5. Hosting	13 13 14 14 14 .4 .5 .5
3.1.5. Hosting.13.1.6. Design & Prototyping.13.2. Technologies for user research and feedback13.2.1. Microsoft Teams.13.2.2. Microsoft Forms14. Specification and Design14.1. Functional Requirements14.1.1. Must14.1.2. Should1	13 13 14 14 14 .4 .5 .5 .5
3.1.5. Hosting.13.1.6. Design & Prototyping.13.2. Technologies for user research and feedback13.2.1. Microsoft Teams13.2.2. Microsoft Forms14. Specification and Design14.1. Functional Requirements14.1.1. Must14.1.2. Should14.1.3. Could1	13 13 14 14 14 15 15 15 .6

4.2. Non-Functional Requirements	17
4.3. Discussion Board UML	
4.4. Use Cases	20
4.4.1. Use Case Diagram	20
4.4.2. Use Cases	20
4.5. User interface prototyping	23
5. Implementation	24
5.1. Main application	
5.2. Discussion Board	
5.2.1. Discussion Board Moderation	
Post Approval	24
Post/Comment/Reply blacklisted word validation	
5.3. Web accessibility	25
5.4. Restriction to age-appropriate content	25
5.5. Code quality	26
6. Results and Evaluation	27
6.1. Test Cases	27
6.2. User Testing	27
6.3. Evaluation	28
7. Future Work	29
8. Conclusions	
9. Reflection on Learning	
References	33
Bibliography	
Appendices	

Table of Figures

Figure 1: UML Class Diagram of main components. Not all variables and methods for the Djan	١go
AbstractUser class have been displayed for simplicity	. 19
Figure 2: Use Cases for the system and the actors involved, based on the functional	
requirements	. 20
Figure 3: Mockups for several devices created in Balsamiq	. 23
Figure 4: Screenshot of main components in PyCharm	.24
Figure 5: Screenshot of 'home.html' showing headings describing their topic as required for	
WCAG Level AA	. 25
Figure 6: Screenshot of implemented CustomUser model	. 25
Figure 7: Screenshot of view used for pages restricted to users over 16. Class inherits from	
UserPassesTextMixin, LoginRequiredMixin, and TemplateView, which are all default classes ir	n
Django	. 26

Table of Tables

Table 1: Table of test case results 27
--

1. Introduction

1.1. Project Aims

This project aims to create a digital guide for Looked After Children (LAC) to access on any device, as and when they need it, through working with care-experienced young people. The guide needs to be accessible and informative, and include topics wanted by children and young people in care, in order to better inform them of their rights and resources available to them. Having a digital resource to refer to would also help the agencies involved to explain the process, as the number of young people in care has grown steadily in the last two decades (Statistics for Wales 2019).

The user should be able to:

- Access resources appropriate to their age and circumstances
- Login and post on a discussion board

The administrator should be able to:

- Create user profiles for LAC
- Approve posts created by LAC on the discussion board

It will be a proof-of-concept for Cardiff Council Children's Services to determine the level of need (and/or want) for such a website, and what further work is necessary in order to make it a useful, complete resource.

1.2. Project Scope

The original project objectives included features to collate key contacts, meetings and documents for LAC. This side of the project has been undertaken by Alfie Rowett, who has created a "My Important Things" application, and therefore will not be covered by this project.

This project will attempt to cover information for LAC of all ages in Cardiff, from 5-16+, and for LAC at different stages of being in care, from those transitioning into being in care to those who are experienced with the processes involved.

Additionally, there would be value in a discussion board for LAC Wales-wide or UK-wide, but this project will focus on LAC in Cardiff.

1.3. Outcomes

The initial expected outcomes stated by Cardiff Council Children's services were that:

- Looked After Children are better informed about their rights, next steps, and support services. Specifically, those aging out of care should have better access to resources, and those transitioning to being in care should have access to information about the process.
- Children Services are better equipped to support Looked After Children through use of [this] digital resource.

Through further discussion with care-experienced young people, we have identified an additional outcome to help better support LAC:

- LAC can speak with and support each other through use of a discussion board feature on the website.

The initial Project Initiation Document (PID) by Cardiff Council can be found in Appendix A.

2. Background

Being a child or young person in care can be a daunting experience as lots of different adults and agencies are involved at different stages. Other Local Authorities have developed guides to being a young person in care to outline the role of these important adults and agencies in their life as well important meetings during their experience. Child Friendly Cardiff and Children's Services have discussed the need for such a guide to being in care for children and young people in Cardiff. A young person in care may move home and change social workers and personal assistants many times, and an online resource outlining their rights and what they can expect during their time in care would be extremely beneficial in helping them understand what is happening and ease their transition. This information needs to be independently accessible to them regardless of their current support system.

Young people in care often feel powerless due to the uncertainty involved with their care placement which can change seemingly unpredictably (Molin 1990). Giving them easy access to clear information about being in care and resources for the future is important in alleviating this anxiety.

2.1. Existing solutions

There are many existing sources of information for children in care in Wales, and these are usually paper documents. These are easily lost as a child in care may move home several times, and thus they are difficult to manage.

For other authority guides to being in care used to inform this project, see Appendix B.

The most suitable existing website is the Rhondda Cynon Taf "Two Sides" website (Rhondda Cynon Taf County Borough Council [no date]). Two Sides is a website for children and young people who are looked after by Rhondda Cynon Taf County Borough Council, and as such contains information about resources specific to this Rhondda Cynon Taf County Borough Council and Wales, such as funds available only to Rhondda Cynon Taf County Council, or the Wales-wide Education Maintenance Allowance (EMA).

This project aims to create a similar web application for children and young people specifically in Cardiff, with signposting to local and/or Cardiff-specific services and groups. It will therefore use some of the same resources as the current existing solutions for other councils in Wales but will focus on guidance specific to Cardiff.

Generally, no existing solution for online discussion between young people in care specifically – there exist support groups and general discussion board websites like Reddit where there can be spaces for this, but no specific site where young people in care (in Cardiff especially) can talk to each other and discuss their experiences safely.

2.2. User Research

A large part of this project was work with care-experienced young people. We met semiregularly with young people who participated in meetings with NYAS. They had several suggestions for what a digital guide in care should include, such as the topics that should be included.

They also had suggestions for how this content should be presented, in terms of medium and layout. They suggested videos by people who had previously been in care for them to relate to and use of positive quotes. One person had dyslexia and discussed the need for a "dark mode" in order to make the text easier for her to read. Notes taken during these user research meetings can be found in Appendix C.

As well as this research, I attended the webinar "Demystifying Digital Accessibility with GDS" by the Government Digital Service, hosted through Texthelp. This webinar built on my ideas of what web accessibility means and how to implement accessible features on a website. Notes taken when attending this webinar can be found in Appendix D.

2.3. Project justification

Through speaking with care-experienced young people, we have found that there are many experiences throughout being in care that they felt they weren't ready for, and suggested areas where the information given to new children in care could be improved – for example, they said that there were some realities of being in care which aren't talked about, such as having several social workers, and they said that it would make more sense for young people to be aware of the possibility of this. They also said that they had several rights that they didn't know about during their time in care, such as being able to choose what school they go to or choose who goes to personal appointments with them. Information like this can be overlooked, and they suggested this guide can be a definitive place for them to find out exactly what they are entitled to or allowed to do without having to speak to their placement providers or foster family or rely on paper documents which can't be updated – a digital resource is much more easily managed to distribute this important information.

As this digital resource is required to be accessible on multiple devices, a website is a more suitable solution compared to a mobile application as websites are accessible on most smart devices as they generally have browsers, whereas a mobile application would need to be ported into a separate version to be accessible on computers.

The decision was made to create a discussion board feature due to the suggestions given at the initial user research meetings – one person suggested a 'buddy up' system in which two LAC of differing ages could support each other directly, however I was unsure of how I could implement this at just the web application level while ensuring safeguarding. I still wanted to implement a feature where they could speak with each other in a safe environment – a discussion board would be important in supporting LAC as it facilitates bringing them together and giving them a sense of belonging, which is important in coping with stigma (Rogers 2017).

Rogers found that peer support is extremely important in helping them cope with the stigma of being in care, through both formal and informal opportunities – this discussion board allows them to create connections in a safely moderated fashion, with potential for informal friendships (which may not focus on their status as LAC) to develop outside of it.

2.4. Constraints

The timescale for this project is 3 months. The original Project Initiation Document (PID) states that project will finish at the end of April, however in order to leave enough time to finish this report, actual work done on the application may end sooner.

Due to the time available, and financial constraints on hosting due to the nature of this being a student project, the end product will be a prototype and not a finished product.

3. Approach

This project will follow an agile development methodology due to the number of stakeholders involved who may suggest different requirements at different stages – this type of methodology allows for iterative work, meaning feedback can be given continuously throughout the project as the prototype is being built.

As previously mentioned, this project focused on gaining insight from care-experienced young people as to what features they were looking for in a digital guide, and their feedback on the prototype once completed. I attended a UNICEF Children's rights in practice online training module which was organised by Cardiff Council, which introduced the idea of using a child's rights-based approach – making sure the people involved in making this project have an awareness of children's rights in Cardiff, and that children participate in requirements gathering and testing to make sure it caters to what they actually want and need.

3.1. Technologies for development

3.1.1. Web application frameworks: Django vs Flask

Django will be used due to its maturity and vast number of third-party libraries available for integration, as well as to ensure consistency with sibling "My Important Things" project which also uses Django. Flask, another web application framework, was also considered – Flask also has a mature community and is very flexible as it doesn't come with the same features built into Django, but they're instead available as different libraries to be incorporated with Flask. Due to this project being a prototype, Cardiff Council may decide to continue its progress and Django is more suitable for this as it is more popular, with a bigger and more active community.

It has many in-built features and tools which are very helpful in creating web applications in general such as pagination, authentication, and protection against cross site scripting (XSS), cross site request forgery (CSRF), SQL injection and clickjacking. It also has several in-built features which are relevant to this project: it has its own ORM, which will be useful for working with the discussion board data; an inbuilt login system which is easy to override, for the administrators and users; and an administrator site powered by Django's ORM, which will be useful for administrators to immediately start adding users and discussion board topics, as well as making post moderation simple to implement.

While there are very powerful and updated third party libraries for Flask which also implement these features (such as SQLAlchemy), the built-in administration site in Django is a key feature for this project due to its ease of use.

3.1.2. Web application tools

Bootstrap, a CSS framework, will also be used due to its popularity as this means that it is constantly being updated and issues are fixed regularly. It has very clear documentation and is very easy to start using, which is important in a project where a large proportion of feedback will be about usability and page design, and when this feedback will be given at regular intervals.

Web accessibility needs to be taken into consideration in particular due to the audience of the resulting prototype – it needs to be accessible by children of all ages and abilities. While there are other web UI frameworks available which are more accessible out of the box such as Reakit and Material UI (Kay 2019), they have a greater focus on interoperability with JavaScript libraries like React which is beyond the scope of this project. Additionally, Bootstrap is aware of where it initially fails to meet common accessibility principles such as sufficient contrast (Bootstrap [no date]), whereas other frameworks are not as forthright about their limitations and may take longer to adjust for the project's needs.

Miniconda, a minimal installer for Anaconda, was used to manage the packages used in this project such as Django and django-crispy-forms, and to create a persistent environment. Virtualenv was used to install needed modules in the Pythonanywhere environment as it doesn't support Anaconda environments.

3.1.3. Data storage

SQLite is used to store the data created by this prototype as it is included in Python, and the data is relatively simple and structured. SQLite however is not designed for very large datasets or high-volume websites, so a true client/server SQL database like Postgres would need to be considered for a full product if the discussion board feature were to be heavily used, as posts and their comment threads would be retrieved very often.

3.1.4. Version Control

Git is used to track changes and ensure a constant up-to-date backup of the project. The repository was hosted on Cardiff University's instance of GitLab, and later on GitHub for easy retrieval when hosting on Pythonanywhere.

Git conventions were used such as including a .gitignore file, including a README.md file for other contributors to create a development environment and work on the project. It was chosen as the method of version control for this project due to the ease of creating "issues" on the GitLab website in order to keep track of what features are yet to be implemented, and what bugs still need to be fixed.

3.1.5. Hosting

While this project's aim is to create only a prototype, a hosting service is required for user testing. Pythonanywhere was chosen due to ease of use and specialty in hosting Django web applications.

3.1.6. Design & Prototyping

Creately was used to create use case diagrams and UML diagrams due to its simplicity and range of templates available to use for free.

Balsamiq was used to create initial mock-ups of the web application, as I was familiar with using it from the Human Computer Interaction module at university in second year. However, it was

used before confirming the discussion board as a feature to be included, and thus the mock-ups created do not reflect the design of the application in its final state.

3.2. Technologies for user research and feedback

3.2.1. Microsoft Teams

Microsoft Teams was used to meet with the project leader to define requirements and also to meet with care-experienced young people and care professionals for their insight and suggestions for web application features.

3.2.2. Microsoft Forms

Microsoft Forms was used to create the user testing feedback form due to its simplicity and integration with my university email account.

4. Specification and Design

4.1. Functional Requirements

Due to on-going input from care-experienced young people and care professionals throughout the process, requirements have changed as different stakeholders have had different ideas and suggestions. The requirements set out in this section are the final confirmed requirements.

4.1.1. Must

Requirement 1 (FR1)

Administrators must be able to create a young person's account.

Acceptance Criteria:

- Administrator can login as an administrator with permissions not available to regular users.
- Administrator can input young person's username, email address, date of birth and initial password in administration site.
- Young person can then log in on main application with said username and initial password.

Functional Requirement 2 (FR2)

Administrators must be able to approve a young person's post before it is available on the web application.

Acceptance Criteria:

- Administrator can login as an administrator with permissions not available to regular users.
- Administrator can view created posts on administration site and approve posts.
- All users can then see post published on web application.

Functional Requirement 3 (FR3)

Administrators must be able to create topics for the discussion board.

Acceptance Criteria:

- Administrator can login as an administrator with permissions not available to regular users.
- Administrator can input new discussion board topic and description in administration site.
- Young person can then login and create posts related to this topic.

Functional Requirement 4 (FR4)

Young people must be able to create posts in the discussion board.

Acceptance Criteria:

- Young person can log in, navigate to a topic page within the discussion board and create a new post.

Functional Requirement 5 (FR5)

Young people must be able to comment on approved posts in the discussion board.

Acceptance Criteria:

- Young person can navigate to a post detail page and leave a comment underneath the post.

Functional Requirement 6 (FR6)

Young people must be able to reply to comments on an approved post in the discussion board.

Acceptance Criteria:

- Young person can navigate to a post detail page, view the comments on it and leave a reply to any of those comments.

Functional Requirement 7 (FR7)

Young people must be able to view information regarding their time in care.

Acceptance Criteria:

- Young people must be able to access pages about their wellbeing and education while in care as well as a glossary for terms used in the care system, and a page with information about what to do if they're unhappy with the care they receive.
- Young people over 16 must be able to access information regarding leaving care, and skills they will need once they're living independently.

4.1.2. Should

Functional Requirement 8 (FR8)

Unregistered users should be able to view the information regarding their time in care.

Acceptance Criteria:

- On navigating to website, guides should be accessible through navbar without needing to login.

Functional Requirement 9 (FR9)

Young people under 16 should be blocked from viewing pages related to leaving care or sensitive topics.

Acceptance Criteria:

 Users who are not logged in, or young people who are logged in and under 16 cannot see 16+ services link on the navigation bar and are redirected to a 403 Forbidden page.

Functional Requirement 10 (FR10)

Young people under 9 should be able to access a separate section of the website which focuses on the initial transition into care in an age-appropriate way (e.g. storytelling)

Acceptance Criteria:

- On navigating to website, young person has the choice between choosing the version of the site for 5–8-year-olds, or the version of the site for young people older than 8.

4.1.3. Could

Functional Requirement 11 (FR11)

Allow users to change text size and the colour scheme of website.

Acceptance Criteria:

- Young person can change the size of the text on the site or the colour scheme by selecting an Accessibility button which brings an overlay window with toggle buttons for these two options.

4.1.4. Wont

Requirement

Users will not be able to access a Welsh version of the web application features or content, nor any in-built translation feature.

4.2. Non-Functional Requirements

Requirement

System must be usable on desktop and mobile devices.

Acceptance Criteria:

- All navigation bar links and pages are accessible when using Firefox Responsive Design mode.

Requirement

System colour contrasts must be accessible – they must have a contrast ratio of at least 7:1 to meet WCAG Level AAA requirements.

Acceptance Criteria:

- Colour contrasts in navbar and elsewhere will be checked using WebAIM's Contrast Checker.

Requirement

System must allow for future developments.

Acceptance Criteria:

- Development of the project and technologies used will be thoroughly documented, including resources used for content. This documentation will be clear and comprehensive enough for a new developer to continue work on the project.

4.3. Discussion Board UML



Figure 1: UML Class Diagram of main components. Not all variables and methods for the Django AbstractUser class have been displayed for simplicity.

The CustomUser class inherits from Django's AbstractUser class in order to make use of its attributes and methods while adding a date of birth attribute for future methods which check for age for authentication purposes.

4.4. Use Cases

4.4.1. Use Case Diagram



Figure 2: Use Cases for the system and the actors involved, based on the functional requirements.

4.4.2. Use Cases

Where validation occurs in forms, this will be described in the test cases later in this document, rather than as an alternate flow or exception in these use cases.

Use Case Id: 1	Type: Must have		
Purpose:	Administrators must be able to create a young person	's account.	
Main Actor:Secondary Actors: Young PersonAdministrator			
Pre-conditions	Young person does not have a profile. Administrator is as superuser.	Young person does not have a profile. Administrator is logged in as superuser.	
Basic Flow:	 Admin navigates to administration site Admin selects "+ Add" button next to "Users" underneath the "ACCOUNTS" header Admin enters user's personal details: username, email address, date of birth and initial password Admin selects 'Save' button 		
Post-condition	:		

Use Case Id: 2	I: 2 Type: Must have		
Purpose:		Administrators must be able to approve a young person's post	
		before it is available on the web application.	
Main Actor:		Secondary Actors: Young Person	
Administrator	Administrator		
Pre-condition	Pre-conditions: Young person has created a post within a topic. Administrato		
		logged in as superuser.	
Basic Flow:	1. Admin navigates to administration site		
	2. Admi	n selects "+ Add" button next to "Topics" underneath the	
"DISCUSSION_BOARD" header		CUSSION_BOARD" header	
	3. Admi	n enters title and description into form	
	4. Admi	Admin selects 'Save' button	
Post-condition	ns:	Topic is now published on main site and available to add posts for.	

Use Case Id: 3		Type: Must have	
Purpose:		Administrators must be able to create topics for the discussion	
		board	
Main Actor:		Secondary Actors:	
Administrator			
Pre-conditions:		Administrator is logged in as superuser.	
Basic Flow:	1. Admin navigates to administration site		
	2. Adr	nin selects "Posts" link underneath the "DISCUSSION_BOARD"	
	hea	der	
	3. Adr	nin selects unapproved post in list	
Admin ticks 'approved' checkbox in post detail view		nin ticks 'approved' checkbox in post detail view	
	5. Adr	5. Admin selects 'Save' button	
Post-conditio	ns:	Post is now published on main site and visible in its associated	
		topic.	

Use Case Id: 4	ļ	Type: Must have
Purpose:		Young people over 12 must be able to create posts in the
		discussion board
Main Actor: Y	oung Person	Secondary Actors:
Pre-condition	s:	Young person is logged in, and on the main discussion board page.
Basic Flow:	1. Youn	g person selects a topic link
	2. Youn	g person selects 'Create new post' button
	3. Youn	g person enters post title in 'Title' field and content in 'Content'
	field	and selects the topic in 'Category' drop down
	4. Youn	g person selects 'submit' button
	5. Youn	g person is redirected to the post success page

Post-conditions:	New post is created in database, awaiting moderator approval.
	Post is not published on the main site.

Use Case Id: 5	5	Type: Must have
Purpose:		Young people over 12 must be able to comment on approved posts
		in the discussion board.
Main Actor: Y	oung	Secondary Actors:
Person		
Pre-condition	s:	Young person is logged in, and on the main discussion board page.
Basic Flow:	1. Young person selects a topic link	
	2. Yo	ung person selects post detail link
	3. Yo	ung person selects 'Comment' button underneath the main post
	4. Yo	ung person enters valid comment in 'Content' field of comment form
5. Young person selects 'submit' button		ung person selects 'submit' button
	6. Yo	ung person is redirected to the post detail page
Post-conditio	ns:	New comment is created in database and appears underneath post
		on post detail page.

Use Case Id: 6		Type: Must have
Purpose:		Young people over 12 must be able to reply to comments on an
		approved post in the discussion board.
Main Actor: Y	Main Actor: Young Person Secondary Actors:	
Pre-condition	s:	Young person is logged in, and on the main discussion board page.
Basic Flow:	1. Youn	g person selects a topic link
	2. Youn	g person selects post detail link
	3. Youn	g person selects 'Reply' button underneath a comment
	4. Youn	g person enters reply in 'Content' field of reply form
	5. Youn	g person selects 'submit' button
	6. Youn	g person is redirected to the post detail page
Post-conditio	ns:	New reply is created in database and appears underneath post on
		post detail page.

Use Case Id: 7	7	Type: Must have
Purpose:		Young people must be able to view information regarding their
		time in care.
Main Actor: Y	oung Person	Secondary Actors:
Pre-condition	s:	Young person is logged in and on the main home page.
Basic Flow:	 1. Young person selects a guide from the navigation bar: [Your education, your wellbeing, your voice, 16+ services, what does that mean?] 2. Guide for selected topic is displayed 	

Alternate	2. If young person is under 16, 16+ services pages return a 403 Forbidden page	
Flow:	instead of displaying the content.	
Post-conditions:		New reply is created in database and appears underneath post on
		post detail page.

4.5. User interface prototyping

I used Balsamiq to create initial mockups of the home page and the age-check page (which was not implemented in the final product), to get an idea of how the web application might look on several different devices and consider the different types of input fields available on computers compared to mobile devices – for example, the "How old are you?" page design incorporates an Apple-style number picker.



Figure 3: Mockups for several devices created in Balsamiq.

5. Implementation

The different aspects of the application (authentication, discussion board and main guide content) have been separated by app within the Django project.

Due to how Django's administration site and ORM is implemented, any model instance that can be added in the administration site can also be edited and deleted in the administration site, unless the user explicitly decides not to show edit and delete forms.

5.1. Main application

The content on the guide pages (Your Wellbeing, Your Education, Your Voice, What does that mean? pages) has been informed through discussions with care-experienced young people and care professionals as well as documents for being in care in other counties. This content should be considered a placeholder, as the actual content should be written and checked directly by actual care professionals and care experienced young people as they know best what resources are available.

5.2. Discussion Board

CustomUser, Author, Post, Comment and Reply classes were implemented using Django models. Django signals were used to automatically create a new author instance when adding a new user – this ensured the two apps (accounts and discussion_board) remained decoupled.

5.2.1. Discussion Board Moderation

Due to the vulnerability of the target audience for the

discussion board, ways of moderating the content posted has been implemented. All posts must be approved by an administrator, and the forms used to submit posts, comments and replies have validators to ensure profane language isn't posted.

Post Approval

I made use of the built-in Django administration feature to enable administrators to approve posts, using a Boolean field on the Post model which can be ticked on the administration site. This value is then checked on the Post detail page, and if it is true then the page is displayed.

Post/Comment/Reply blacklisted word validation

The post/comment/reply blacklist validator uses the django-profanity-filter library. This library is capable of detecting distorted profanity using deep analysis however this has not been implemented in this project due to lack of time available. I believe in its current state this is

Figure 4: Screenshot of main components in PyCharm.



FYP ~/FYP

acceptable due to the presumed overall moderation of the discussion board by administrators and acts more as a reminder for young people that profane language is not acceptable.

5.3. Web accessibility

The main difficulty of this project was integrating Django, Bootstrap and a third party Djangosass library in order to overwrite Bootstrap classes with SASS to conform to better standards of web accessibility. In order to conform to WCAG at Level AAA, the text-dark class in Bootstrap needed to be overwritten to a darker shade for greater contrast.

Making sure that the website was navigable was also important – adding page titles to every page was made easier by using the Django block content tags in templates, and I made sure that appropriate HTML tags were used to identify content.



Figure 5: Screenshot of 'home.html' showing headings describing their topic as required for WCAG Level AA.

5.4. Restriction to age-appropriate content

The created custom user model included a date of birth attribute as defined by the UML Class Diagram, as well as an "is_over_16" property using this date of birth attribute.



Figure 6: Screenshot of implemented CustomUser model.

This property is then accessed by templates to test whether this property is true for a user and if it is, the templates show the content. This is used by the base template to test whether the

user has access to the pages through the navigation bar. This property is also used for the test in the SixteenPlusPageView, which uses the UserPassesTestMixin to incorporate the test.



Figure 7: Screenshot of view used for pages restricted to users over 16. Class inherits from UserPassesTextMixin, LoginRequiredMixin, and TemplateView, which are all default classes in Django.

5.5. Code quality

I used several aspects of Django to ensure the high quality of the code I wrote.

The Django template system is a way to express the presentation of pages in a Django web application, with some capability for logic. Template inheritance allows you to build a base template which contains all the elements needed for several pages, such as the navigation bar and footer. Using template inheritance with a base template reduced code repetition (and allowed me to follow the DRY principle), as I would otherwise need to have the navigation bar in every template for every page. The Django template system also allowed me to use model properties to only show navigation links which didn't have an age requirement to view, if the user is under 16.

I also made use of class-based views in Django. Django has several view types for common tasks, such as FormView and DetailView, which can be subclassed and used in user defined views. Using class-based views also provides the capability for using mixins, and making use of both means associated functions can be overwritten only when necessary to reduce redundant code.

6. Results and Evaluation

6.1. Test Cases

These test cases have been carried out in order to verify whether the system meets the outlined requirements. In these tests, multiple types of test data including valid data, invalid data and missing data were used to test the systems capability with handling unexpected data.

Test ID	Related Requirement	Pass/Fail
1	FR1	PARTIAL FAIL
2	FR3	PASS
3	FR4	PASS
4	FR2	PASS
5	FR5	PASS
6	FR6	PASS
7	FR7	PASS
8	FR8	PASS
9	FR9	PASS
10	FR10	FAIL
11	FR11	FAIL

For full test cases, see Appendix E.

Table 1: Table of test case results

Most test cases passed, with test case 1 failing in the case of invalid data, and others passing in meeting the requirement but with unexpected side effects. Test cases 10 and 11 failed due to the related features not being implemented in time.

6.2. User Testing

Due to limited opportunities for meetings with care-experienced young people, I decided that user testing would be done asynchronously by hosting the prototype online and providing testers with test user account credentials and a feedback form to fill out after testing. The feedback form was created with Microsoft Forms, and kept simple to encourage giving feedback as it was easy to do and didn't take up a lot of their time.

For the feedback form used, see Appendix F. For the responses see Appendix G.

One responder said that they had the issue that "On the discussion Board when creating a post, it did not go onto the Board although it said that it had". This is due to the post approval system in place, and while the new post success page says "An admin will check [the post] to approve it, and it will be up soon", this feedback indicates that more clarity is needed in communicating the fact that a post must be approved before it will appear on the main site.

Several respondents discussed adding more variety in the content – on average respondents gave the look of the site a rating of 4/5, however two respondents suggested more colours while others suggested pictures would help. Two out of 4 responses included suggestions to

make the site more dyslexia-friendly, which was discussed in prior meetings and was included in FR11, which due to time was not implemented.

6.3. Evaluation

Django was very effective for creating this project. Many of its features were used in creating this application and its "batteries-included" approach meant that focus could remain on implementing the discussion board and designing the website. In particular, the administration site (and its use of the Django ORM) was extremely useful in managing objects created.

The initial plan specified the creation of a chat-bot to share information with young people using the site. On further reflection after the initial plan was written and after extensive meetings with care-experienced young people, it was decided that a chat-bot was not the appropriate way to share information, so the planned deliverables were re-evaluated and changed. The initial plan outlined several prototypes during the process, however careexperienced young people were often unavailable for meetings to gain feedback, and therefore only one version of the prototype was tested by users towards the end of the project.

The discussion board in its current state does have limitations. As posts currently require approval from an administrator before they can be published, and user accounts must be created by administrators (rather than allowing young people to sign up themselves) the discussion board creates more work for an already burdened care system as these administrators would need to continuously monitor and approve posts by young people. This works for a small number of users, and this may be sufficient for the small userbase, however it would be difficult to determine how popular it would be for young people to use this discussion board. Currently its intended use is as an initial way for young people to meet others in their situation and befriend them before, before using other avenues to continue contact, however if Cardiff Council Children's Services were to decide to promote this discussion board and young people did want to use it, other ways of moderation may need to be considered – for example, a reputation system could be implemented that would allow users to gain points through admissible posts, and upon gaining a certain number of points, their posts would no longer need approval before being published.

As stated, the potential userbase for this application is quite small, as it is limited to young people in care in Cardiff specifically. It may make more sense to create a discussion board for young people in care in Wales, or even the UK, as the number of people who can use this may be too small for them to really wish to engage.

In its current state, all young people can post on the discussion board. Before this application is used in a professional capacity, a care professional is needed to comment on whether this is suitable, as there may be safeguarding concerns with young people in care who are over 18 interacting with younger children.

7. Future Work

There are several aspects of the web application which were planned in the requirements but unfortunately were not implemented due to time:

- The extent to which accessibility has been achieved while accessibility was a big focus
 of this project, the features of the web application could be made more accessible. One
 of the 'could' functional requirements was to 'allow users to change fonts and the
 colour scheme of website', as this was one of the suggestions made during the initial
 user research stage. Adding this feature would help towards making sure the site is
 more inclusive. Beyond this, taking additional steps like making proper use of aria tags
 and using a code parser to ensure HTML elements have complete start and end tags for
 use with assistive technologies would greatly improve the application. Screen reader
 utilities which hide elements on all devices except screen readers should also be
 implemented.
- The functional requirements which haven't been met still need to be implemented. Regarding FR10, discussion needs to be had with younger children in care to hear what they would like from their side of the site, as only older care-experienced young people were available to speak with during this project.
- As part of Django's login functionality, an SMTP server needs to be created in order to implement password reset emails if the user forgets their password.

These features were beyond the scope of this project, however adding them would result in a high quality, completed application:

- As the web application is specifically for children in Wales, it should offer access to the same content in Welsh.
- There needs to be consideration of the location and expenses for full hosting.

There are also some features which could be implemented which would improve both the experience of administrators and young people:

- Currently, the interface for administrators is not very user friendly. Changes could be made to make adding content easier for non-developers – for example, using a static generator to upload markdown documents to add additional guides rather than editing the HTML directly. While the Django administration site has been effective for managing topics and users, in terms of approving posts a clear improvement would be to implement an "approved" column on the 'all posts' view so that administrators can see if a post is approved without needing to view the post directly. It would also be helpful to implement a way to "bulk-approve" posts.
- This project was created at the same time as its sibling 'My Important Things App' project. The ideal result would be for these two applications to have the same database or be able to access the same database for user authentication to allow users to sign

into both applications with the same credentials. This would also require the two projects to have the same look, as consistency is important to avoid confusion.

8. Conclusions

The aim of this project was to create a prototype of a digital guide to being in care in order for Cardiff Council Children's Services to evaluate whether there is value in such a guide. While not all requirements were fulfilled and there is a lot of work that could be done in the future, I consider this project to meet this aim – there was positive feedback from the project leader as well as from the care-experienced young people who tested it, and there is a clear need for a guide of this kind as the care-experienced young people had many suggestions for what it could include.

Many guides to care already created – this prototype used for Cardiff council to explore whether it is even needed. IRO discussed important point that children in care want to be like other children, and a new application specifically for them and being in care may not appeal.

Steps were taken to ensure another developer could pick up this project and continue work on it as described in 5.5., and by using web development best standards such as a README.md with contribution instructions and a requirements.txt file which specifies the extra libraries used in the project.

9. Reflection on Learning

As this was my first time really collaborating with several parties to create a product, I realised that I sometimes allowed other parties to organise the meetings rather than take initiative myself. I found it very difficult to confirm requirements – stakeholders had ideas for what I could include on the website, but no "must have" features other than the guide content, so this was something I needed to decide myself. In future, I'll try harder to confirm requirements from the stakeholders by writing down suggested requirements and asking about them again, as well as ensure time is made to show prototypes throughout the project lifecycle.

While Git was used throughout the process to always ensure an up-to-date backup, Git flow was not used consistently – I did use feature branches separate to the master branch, but I didn't merge these in with a development branch, choosing instead to merge them directly into the master branch. If this project had been released and in use prior to submission, I would have made sure to push changes to a development branch instead.

I should try to learn more about skills I don't have a lot of experience with first before trying to apply them – I found it difficult to implement an appropriate overall design of the web application as I don't have a lot of front-end experience, and the user testing results reflect the lower quality design. I didn't spend enough time thinking about and creating design mock-ups before I started coding the application, which was detrimental to the end result.

In this project I have only considered smartphones and tablets with browser capability, when on reflection I should have researched more about what "multiple devices" entails and whether I can truly assume that everyone has a responsive device capable of accessing a website through a browser. Cardiff Council Children's services and the associated care professionals in Cardiff already use paper documents to share information with young people in care so this isn't an issue in this instance as those would be shared with those without internet and browser access instead, however in the future I should try to define all the terms used when discussing potential requirements of a software application to understand whether my proposed solution is really the most suitable.

My work was unfortunately impacted by the Covid-19 pandemic. I found it very difficult working from home and managing my time in lockdown throughout the project and have learned that I need to make use of other ways of working, such as going to coworking spaces when possible. Because of this difficulty, several planned sections of the report have not been included, such as a heuristic evaluation of the prototype, and an initial state transition network diagram of how the system would work. If I had been able to manage my time more effectively I would also have been able to properly consider the differences in design for apple, android and PC applications, and design my application to be consistent with these differences per device.

References

Bootstrap. [No date]. Accessibility. Available at: <u>https://getbootstrap.com/docs/4.0/getting-started/accessibility/</u> [Accessed: 10 April 2021]

Cynon Taf County Borough Council. [no date]. *Welcome to Two Sides!* Available at: <u>https://www.rctcbc.gov.uk/EN/Resident/ChildrenandFamilies/LookedAfterChildren/Welcometo</u> <u>TwoSides.aspx</u> [Accessed: 6 April 2021]

Kay, D. 2019. The state of accessible web UI frameworks. Darek Kay 25 March. Available at: darekkay.com/blog/accessible-ui-frameworks [Accessed: 13 April 2021]

Molin, R. 1990. Future anxiety: Clinical issues of children in the latter phases of foster care. *Child and Adolescent Social Work Journal* 7, pp. 501–512. doi: 10.1007/BF00756080

Rogers, K. 2017. 'Different' and 'Devalued': Managing the Stigma of Foster-Care with the Benefit of Peer Support. *British Journal of Social Work* 47, pp.1078-1093. doi: 10.1093/bjsw/bcw063

Statistics for Wales, 2019. Experimental statistics: Children looked after by local authorities, 2018-19. Cardiff: Welsh Revenue Authority

Bibliography

Abderrahim, S. 2021. *Django-Community-Forum-Website*. [Source code]. Available at: <u>https://github.com/SelmiAbderrahim/Django-Community-Forum-Website</u>

Children's Commissioner for Wales. 2021. *My Planner*. Available at: <u>https://myplanner.childrenscommissioner.wales/</u>

National Youth Advocacy Service. [No date]. NYAS Children & Young People. Available at: https://youngpeople.nyas.net/

Schneiderman, B. 2016. The Eight Golden Rules of Interface Design. Available at: <u>https://www.cs.umd.edu/~ben/goldenrules.html</u>

Web Accessibility Initiative. 2021. How to Meet WCAG (Quick Reference). Available at: https://www.w3.org/WAI/WCAG21/quickref/

WebAIM. 2020. CSS in Action – Invisible Content Just for Screen Reader Users. Available at: https://webaim.org/techniques/css/invisiblecontent/ [Accessed: 21 March 2021]

World Wide Web Consortium (W3C). 2021. *Introduction to Web Accessibility*. Web Accessibility Initiative (WAI). Available at: <u>https://www.w3.org/WAI/fundamentals/accessibility-intro/</u>

Appendices

Appendix A: PID located in support file "AppendixA_PID"

Appendix B: County foster care documents located in support file "AppendixB_FosterCareGuides"

Appendix C: User research meeting notes located in support file "AppendixC_Meetings"

Appendix D: Demystifying Digital Accessibility notes located in support file "AppendixD_DemystifyingDigitalAccessibility"

Appendix E: Test Cases

Test Case ID: 1	Related Requirement: FR1
Test Description:	Verify that an administrator can create a young person's account.
Prerequisites:	Administrator is logged in as superuser.
Test Procedure:	1. Navigate to /admin/
	Select '+ Add' button next to 'Users' under the 'ACCOUNTS'
	heading. Add user page is displayed
	3. Input test data for [Username, Email address, Date of birth,
	Password, Password confirmation]
	4. Select 'SAVE' button.
Test Data:	1. Valid user details:
	Username: atestuser
	Email address: atestuser@gmail.com
	Date of birth: 2002-10-10
	Password: atestpassword222
	2. User details with invalid email address:
	Username: atestuser
	Email address: atestuser
	Date of birth: 2002-10-10
	Password: atestpassword222
	3. User details with invalid date of birth:
	Username: atestuser
	Email address: atestuserm@gmail.com
	Date of birth: 2022-10-10
	Password: atestpassword222
	4. Missing user details:
	[Leave blank]

Expected Result:	 Change user page for newly created user account is displayed. On selecting 'SAVE' button, 'Enter a valid email address' validation error appears next to Email address field and the user account is not created. On selecting 'SAVE' button, 'Enter a valid data of birth' validation
	 On selecting SAVE button, Enter a valid date of birth validation error appears next to Date of birth field and the user account is not created.
	 On selecting 'SAVE' button, 'This field is required.' validation error appears next to username, date of birth, password and password confirmation fields and the user account is not created.
Actual Result:	1. As expected
	2. As expected
	3. Change user page for newly created user account is displayed.
	4. As expected
Pass/Fail:	FAIL
Comments:	Validation for date of birth field not implemented – field acts as a general
	datetime field, however this does not impede overall test that
	administrator is able to create a young person's account.

Test Case ID: 2	Related Requirement: FR3
Test Description:	Verify that an administrator can add a topic for the discussion board.
Prerequisites:	Administrator is logged in as superuser.
Test Procedure:	5. Navigate to /admin/
	Select '+ Add' button next to 'Topics' under the
	'DISCUSSION_BOARD' heading. Add topic page is displayed
	7. Input test data for Title
	8. Select 'SAVE' button.
Test Data:	1. Title: A test topic
	Description: General description for a test topic
Expected Result:	Topic list in administration site now shows new topic 'A test topic', and
	users can now navigate to /discussion-board/posts/a-test-topic/.
Actual Result:	As expected.
Pass/Fail:	PASS
Comments:	N/A

Test Case ID: 3		Related Requirement: FR4
Test Description:	Verify that a young person c	an create a post in the discussion board.
Prerequisites:	Young person is logged in.	

Test Procedure:	1.	Navigate to /discussion-board/
	2.	Select 'General' topic board link
	3.	Select 'Create new post' button
	4.	Input test data for title, content and category fields
	5.	Select 'submit' button.
Test Data:	1.	Valid data:
		Title: test post
		Content: test content 123
		Category: General
	2.	Invalid data – profane language in content field:
		Title: test post
		Content: [profane language]
		Category: General
	3.	Missing data:
		[Leave blank]
Expected Result:	1.	Post success page displaying 'Your post has been created! An
		admin will check to approve it, and it will be up soon.' message is
		shown.
	2.	On selecting 'submit' button, 'Please remove any profanity/swear
		words.' validation error appears next to Content field and the
		post is not created.
	3.	On selecting submit button, 'Please fill in this field' pop-up
		appears by first field left blank.
Actual Result:	1.	As expected.
	2.	As expected.
	3.	As expected.
Pass/Fail:	PASS	
Comments:	N/A	

Test Case ID: 4	Related Requirement: FR2	
Test Description:	Verify that an administrator must approve a young person's post before	
	it is available on the main site.	
Prerequisites:	Administrator is logged in as superuser. A post titled 'test post' has been	
	created under the topic 'General'.	
Test Procedure:	1. Navigate to /discussion-board/posts/general/ and confirm lack of	
	'test post' in list	
	2. Navigate to /admin/	
	3. Select 'Change' link next to Posts under the 'DISCUSSION_BOARD'	
	header. Posts page is displayed	
	4. Select 'test post' post in list	
	5. Tick 'Approved' checkbox	
	6. Select 'SAVE' button	

	7. Navigate back to 'discussion-board/posts/general/
Test Data:	N/A
Expected Result:	'test post' Post link is now visible in General discussion board on main
	site.
Actual Result:	As expected.
Pass/Fail:	PASS

Test Case ID: 5	Related Requirement: FR5
Test Description:	Verify that a young person can comment on an approved post in the
	discussion board.
Prerequisites:	Young person is logged in. An approved post titled 'test post' has been
	created under the topic 'General'.
Test Procedure:	 Navigate to /discussion-board/posts/general/
	Select 'test post' link in list underneath 'Posts'
	3. Select 'Comment' button underneath post
	4. Enter test data in content field
	5. Select 'Submit' button
Test Data:	1. Valid data:
	Content: test comment
	2. Invalid data – profane language:
	Content: [profane language]
	3. Missing data:
	[Leave blank]
Expected Result:	1. Comment is created and user is redirected to the post page, with
	the new comment underneath the post
	2. On selecting 'Submit' button, "Please remove any
	profanity/swear words." validation error appears next to content
	field and comment is not created.
	3. On selecting 'Submit' button, "Please fill in this field." validation
	error appears next to content field and comment is not created.
Actual Result:	1. As expected.
	2. As expected.
	3. As expected.
Pass/Fail:	PASS

Comments:	On 3., validation error does appear next to field, however on page refresh after selecting 'Submit' button the form is hidden until the user selects the 'Comment' button again, at which point the error is displayed. Comment is not created.

Test Case ID: 6	Related Requirement: FR6
Test Description:	Verify that a young person can reply to a comment on an approved post
	in the discussion board.
Prerequisites:	Young person is logged in An approved post titled 'test post' has been
Therequisites.	created under the tonic (General' and a comment with the content (test
	comment' has been created
Test Dressdure	1 Novigato to (discussion board (datail/test pact/
rest Procedure:	1. Navigate to /uiscussion-board/detail/test-post/
	2. Select Reply button underneath test comment comment
	3. Enter test data in content neid
	4. Select Submit button
Test Data:	1. Valid data:
	Content: test reply
	2. Invalid data – profane language:
	Content: [profane language]
	3. Missing data:
	[Leave blank]
Expected Result:	1. Reply is created and user is redirected to the post page, with the
	new reply underneath the comment
	On selecting 'Submit' button, "Please remove any
	profanity/swear words." validation error appears next to content
	field and reply is not created.
	3. On selecting 'Submit' button, "Please fill in this field." validation
	error appears next to content field and reply is not created.
Actual Result:	1. As expected.
	2. As expected.
	3. As expected.
Pass/Fail:	PASS
Comments:	On 3., validation error does appear next to field, however on page
	refresh after selecting 'Submit' button the form is hidden until the user
	selects the 'Reply' button again, at which point the error is displayed.
	Reply is not created.

Test Case ID: 7	Related Requirement: FR7
-----------------	--------------------------

Test Description:	Verify that a young person can view information regarding their time in
	care.
Prerequisites:	Young person (over 16) is logged in.
Test Procedure:	1. Navigate to /
	Navigate to /your-education
	3. Navigate to /your-wellbeing
	4. Navigate to /your-voice
	5. Navigate to /what-does-that-mean
	6. Navigate to /16+
	Navigate to /your-skills
Test Data:	N/A
Expected Result:	All pages return HTTP response status codes 200 and content is viewable.
Actual Result:	As expected.
Pass/Fail:	PASS
Comments:	N/A

Test Case ID: 8		Related Requirement: FR8
Test Description:	Verify that an unauthenticated user is able to access pages with	
	information relating to your	g people's time in care.
Prerequisites:	None.	
Test Procedure:	8. Navigate to /	
	9. Navigate to /your-ed	lucation
	10. Navigate to /your-we	ellbeing
	11. Navigate to /your-vo	ice
	12. Navigate to /what-do	pes-that-mean
Test Data:	N/A	
Expected Result:	All pages return HTTP respo	nse status codes 200 and content is viewable.
Actual Result:	As expected.	
Pass/Fail:	PASS	
Comments:	N/A	

Test Case ID: 9		Related Requirement: FR9
Test Description:	Verify that a young person under 16, or an unauthenticated user, is	
	blocked from viewing pages r	elated to leaving care.
Prerequisites:	None.	
Test Procedure:	13. Navigate to /	
	14. Navigate to /16+	
	15. Navigate to /your-skill	ls

Test Data:	N/A
Expected Result:	 "16+ Services" dropdown navigation item is not visible.
	2. 403 Forbidden Page is displayed.
	3. 403 Forbidden Page is displayed.
Actual Result:	1. As expected.
	2. Page displaying 'Server Error (500)'
	3. Page displaying 'Server Error (500)'
Pass/Fail:	PASS
Comments:	While pages displayed for 2 and 3 are not appropriate HTTP response
	status codes, the unauthenticated user is blocked from viewing these
	pages.

Test Case ID: 10	Related Requirement: FR10	
Test Description:	Verify that a young person under 9 accesses a separate version of the site	
	with more age-appropriate content.	
Prerequisites:	N/A	
Test Procedure:		
Test Data:	N/A	
Expected Result:	Young people under 9 are directed to a separate version of the site.	
Actual Result:	Young people under 9 can only access the main site.	
Pass/Fail:	Fail	
Comments:	This side of the web application was not implemented in time.	

Test Case ID: 11	Related Requirement: FR11	
Test Description:	Verify that a user can change the text size and the colour scheme of	
	website.	
Prerequisites:	User is logged in.	
Test Procedure:	1. User selects 'accessibility' button	
	2. User selects 'Toggle font size' on accessibility overlay	
	3. User selects 'Toggle dark mode' on accessibility overlay	
Test Data:	N/A	
Expected Result:	1. Text size on website is increased	
	2. Website background is changed to a darker colour	
Actual Result:	Test size/colour theme unchanged.	
Pass/Fail:	Fail	
Comments:	Feature was not implemented in time.	

Appendix F: Guide to Being in Care Prototype Feedback Form

Guide to Being in Care Prototype Feedback

....

* Required

1. How well did things work? *

2. How good did things look? *

3. Did you have trouble reading anything or getting things to work? *

- O Yes
- O No

4. If yes, what went wrong? What did you do and how did it respond?

Enter your answer

5. Do you have any ideas for what needs to be added? For example, guides on different topics, or things to make it easier to read or use.

Enter your answer
Submit

Appendix G: Guide to Being in Care Prototype Feedback responses located in support file "AppendixG_PrototypeFeedbackResponses"