The design of a web application to act as a hub of services for the transgender community in Liverpool

Lucy Alexandria Morris – G4272527

TM470 – EMA – 11th September 2023

A person holding a flag

Description automatically generated with medium confidence

Figure A transgender flag being waved at LGBT gay pride march by ‘ink drop’ used under Standard License from Adobe

Contents

[1. Problem description 3](#_Toc141959916)

[1.1 Title 3](#_Toc141959917)

[1.2 Description 3](#_Toc141959918)

[1.3 Analysis of likely impact 4](#_Toc141959919)

[2 Account of related literature 4](#_Toc141959920)

[2.1 DBaaS 4](#_Toc141959921)

[2.2 Requirements 6](#_Toc141959922)

[2.3 CSS 7](#_Toc141959923)

[3. Account of Project Work and its outcome 11](#_Toc141959924)

[3.1 User Interface 11](#_Toc141959925)

[3.2 Requirements elicitation 14](#_Toc141959926)

[3.3 misc work 18](#_Toc141959927)

[Setting up the database and API’s 18](#_Toc141959928)

[Implementing the map and pins 19](#_Toc141959929)

[1.3.2.3 Implementing tags and info sidebar 22](#_Toc141959930)

[4. Review 27](#_Toc141959931)

[4.1 Current stage of project work 27](#_Toc141959932)

[4.1.1 Legal, social, ethical and professional issues (LSEPIs) 27](#_Toc141959933)

[4.1.2 Equality, Diversity and Inclusion concerns (EDI) 28](#_Toc141959934)

[4.2 Review of project management 29](#_Toc141959935)

[4.3 Personal development 29](#_Toc141959936)

[5. Epilogue 29](#_Toc141959937)

[6. References 29](#_Toc141959938)

[7. Appendices 29](#_Toc141959939)

# 1. Problem description

## 1.1 Title

The design of a web application to act as a hub of services for the transgender community in Liverpool

## 1.2 Description

There are many vital services available to the transgender community in Liverpool, but it can be a challenge to find them. This is particularly the case for new members of the community who may be in a vulnerable situation, as the point of coming out is a huge life change that may be accompanied by mental health issues and the loss of relationships. This is compounded by transgender healthcare facing profound ignorance (and in some cases outright bigotry) from many GPs, who may attempt to block or delay attempts at medical transition, alongside extreme waiting lists for appointments at Gender Identity Clinics (McAuley, 2022).

The trans community attempts to solve these issues in numerous ways, mostly informal (e.g. by word of mouth in group chats), but two more formal ways are the Liverpool Trans Wiki (TransLiverpool Wiki, 2023) which catalogues and comments on many services that are available; and the Spirit Level peer support group which invites in guests from services to explain what they offer to the community. The proposed web app will seek to build on these solutions and incorporate this specialist knowledge into it.

The web app will be usable on web browsers and on mobile devices using HTML, CSS, and JavaScript and will utilize an AWS DynamoDB table as a database, consisting of several pages. Firstly, a page that contains a map which displays services on it, which will be filterable by type (e.g. mental health or peer support) and gender identity (e.g. transfeminine and transmasculine); and when a service is selected more information and contact details will be provided. Another page will contain a search function for a database which will return services based on user queries. There will also be a page that will contain an events calendar. The app will be designed such that information will be one way, to protect the user’s privacy and limit the amount of sensitive data stored by the app. Ethical considerations are vital for the project, since some users may not wish to be open about their trans identity and therefore would need to conceal their use of the app. This must be at the forefront of the final product to reassure users that their identity is safe.

The requirements elicitation will also explore another feature, a map which displays the location of gender-neutral toilets at businesses such as cafes, restaurants, and pubs. However, this will be considered for further work beyond this project and will be considered out of scope to keep the amount of development to an achievable level. The scope of the project is for services in Liverpool & Merseyside due to the ability to engage with services and their users directly and due to already existing knowledge. However, if successful the app could be expanded to cover the whole of the UK, it would be a matter of gathering the data rather than any technical challenges. Online only services that have no physical footprint in Liverpool & Merseyside would also be out of scope, though some of these may be included in an ‘important links’ page but not as part of the core services covered by the app. The scope of the services included is that they must offer something specific to the trans community, rather than broader services (e.g. the trans focused CMAGIC & TSS counselling services would be included, but not the general NHS counselling service offered by Talk Liverpool.)

### 1.3 Legal, social, ethical and professional issues (LSEPIs)

A requirement of the project involves following the BCS Code of Conduct (2023), which includes promoting equal access to IT (discussed below, when considering the impact to equality and accessibility) and not to disclose confidential information. Additionally, there are responsibilities to data protection under GDPR, so ICO (2023) guidance must be followed in relation to personal data. Therefore, in addition to gaining the proper informed consent of participants consideration had to be taken to protect the data and anonymity of participants. This included storage of completed questionnaires, with Microsoft OneDrive to be used as per Open University Research Data Management Policy (2021). The names of participants were to be redacted, as was any other statements or information that could be used to identify them so they could be quoted anonymously. The exception to this was any service providers who participated, since by noting them as a provider of a particular service they may be unavoidably identifiable, but they were informed of this and consent gained so that they could choose to opt-in or opt-out.

When creating a GET request to access the database (a DynamoDB table), valid credentials needed to be supplied. There are multiple ways to do this, including hardcoding them into the code itself, which is not recommended, as it presents a security risk even if they were later removed due to version control systems retaining older versions of code (Amazon, 2023d). Instead, the recommended approach by AWS is to use two other AWS services, Cognito and IAM (Identity and Access Management) (Amazon, 2023e; Amazon, 2023f).

The next step was to use a GET request to access the DynamoDB table to find location data for services and then put a pin at that location. However, in order to access AWS services valid credentials need to be supplied. There are multiple ways to do this, including hardcoding them into the code itself, which is not recommended, as it presents a security risk even if they were later removed due to version control systems retaining older versions of code (Amazon, 2023d). Instead, the recommended approach by AWS is to use two other AWS services, Cognito and IAM (Identity and Access Management) (Amazon, 2023e; Amazon, 2023f). Cognito identity pools can be used to generate temporary credentials for the users of the app, which are applied for the role created in IAM. In this case the intention that users will not require accounts, so the users will be unauthenticated, but if that were to change these services could be used to authenticate users. Once this has been setup then the AWS SDK for JavaScript was included in the HTML file as shown in figure 9, and the Identity Pool ID and region supplied to the AWS config as shown in figure 9. This allows the app access to AWS services, such as DynamoDB.



Figure 8 AWS SDK in the markup

A screen shot of a computer code

Description automatically generated with low confidence

Figure 9 AWS Config submitted credentials to allow access to services

The Open University guidelines on conducting research with human participants were followed, with the research not being considered high risk as all participants being over the age of 18 with full cognitive capacity, the full knowledge and consent of all participants was gained, and the questions did not broach sensitive topics. The Consent Form and Participant Information Sheet, which explain what data is stored, what it will be used for and to gain consent are in appendices J and K (Open University, 2023a).

Review this section, particularly the BCS.

<https://learn2.open.ac.uk/mod/oucontent/view.php?id=2093267>

<https://learn2.open.ac.uk/mod/oucontent/view.php?id=2093267&section=1.4>

Rewrite the paragraph in green (which is copied from Requirements section) to go into more detail, that the original paragraph can refer to.

### 1.4 Equality, Diversity and Inclusion concerns (EDI)

Although the application is being developed for a minority community, this does not necessarily mean it automatically meets responsibilities for Equality, Diversity, and Inclusion. Even within the transgender community there are many different experiences that are quite different, trans masculine, trans feminine and non-binary for example and these identities may intersect with other identities such as neurodivergence, race and disability. For transgender people, examining the nature of gender and how it exists in society is often part of the experience of transition and trans feminine people frequently experience what is known as transmisogyny (Julia Serano, 2007), the simultaneous experience of sexism and transphobia i.e. being dismissed as lesser or weak for being feminine, but also dangerous or ‘perverted’ at the same time. However, as a white, neurotypical, trans feminine person there are still unconscious biases that come with this that must be examined. It is also important to note that while the legal framework is important, it may be flawed with for example the Equality Act 2010 containing language considered outdated and offensive to the transgender community such as ‘gender reassignment and transsexual’, as noted by a report by the Women and Equalities Committee (Miller et al, 2015).

The questionnaire for requirements elicitation resulted in respondents discussing some of these issues, making suggestions for transmasculine, non-binary people and those who are neurodivergent, such as having service tags to filter content specifically for identities e.g. transmasculine people. This is not just for the convenience of the user, but also their mental wellbeing as some services that people transitioning from male to female might trigger gender incongruence in those transitioning from female to male, and in my own experience it is common in trans spaces to be aware of these issues.

However, beyond content the coding of the app must also consider accessibility issues, such as users that use screen readers, those with colour blindness and users that may be using a variety of devices or browsers. As Gray states in his CSS Tutorial – Full Course for Beginners (2022), if the font size is set to a specific pixel size, then it will stay at that size regardless of what settings the user might have. This may cause issues particularly for partially sighted users, but also for any user who simply prefers to have larger font. Instead using a ‘rem’ unit will mean that the font size is relative to the root element and will scale accordingly. Additionally, setting fallback fonts improves accessibility by ensuring the text will always be displayed with a ‘web safe’ font.

It was also important to consider how it would be read by a screen reader and with particular care taken when using a property such as ‘display:none’, which not only hides an element from view but also from being read by a screen reader. The pop-in side menu is triggered by toggling a checkbox (by pressing the hamburger menu symbol) and initially this was hidden from view by using ‘display:none’, which made it un-selectable by tabbing through elements. To change this, a style was added to the checkbox so that when it is in focus, the hamburger icon changes background colour as it does when hovered over and the checkbox was then hidden by setting the opacity to 0. The code snippet in figure 13 shows this change.

![Text

Description automatically generated]()

Figure 13 Removal of display:none for screen readers

Consider more about accessibility, such as colour contrast & colour blindness.

## 1.5 Analysis of likely impact

The project, if successful would increase the visibility of services available to the trans community and so would mean that transgender individuals would gain the benefit of accessing them. As these services provide very important and worthwhile things, it could mean that a trans person gets access to essential sexual health care much faster to avoid potentially living with an STD; finds a peer support group to help them with their transition; or a trans-friendly GP who is prepared to offer HRT (Hormone Replacement Therapy) through bridging prescriptions.

However, increasing the visibility of these services could also have unintended negative consequences too. In the last few years some trans related events and organisations have been targeted by far right and ‘Gender Critical’ groups, including protests accusing trans and gender non-conforming people of paedophilia (Hansford, 2023); violence directed at trans people and allies; and faeces smeared on the entrance of a supportive church (O’Thomson, 2023). During requirements elicitation one of the respondents raised this as a potential risk for the app: “not sure if this might allow terfs to find and cause problems for those services/users who attend that place?” (n.b. ‘terf’ stands for Trans Exclusionary Radical Feminist and is a disparaging term for people from a feminist tradition who are anti-trans, but it is often used as a synonym for the broader ‘Gender Critical’ movement or all people who are anti-trans). There is a risk therefore that this app would provide a way for those with ill intent to target these services.

However, on balance, the positive aspects of the app strongly outweigh the risks it may present. Those who wish to target the trans community could find out about these services through other means already and although it is important to take reasonable precautions for safety, it is also important to increase the visibility and awareness of the transgender community. Indeed, one of the main organised events for the community is ‘Transgender Day of Visibility’ for this purpose.

# 2 Account of related literature

## 2.1 DBaaS

The focus of the first literature review was on sources relating to databases, DataBase-as-a-Service (DBaaS) products and security and privacy issues relating to databases. The database is key to the functioning of the app and was implemented early in development. This also will involve work beyond existing skills, since in TM352, the module the project builds on, the database was already setup and provided with the focus on API calls. The search considered the date that the sources were written, as for some of them there would be a risk of being obsolete, since cloud technology is a relatively recent innovation with the term being coined in 2006 around the release of Amazon Web Services (Regalado, 2011) and has experienced a rapid expansion since then. The source was also considered, with a strong preference given to papers published in journals or from conferences; textbooks; and official documentation.

Poljak et al (2017) compare three popular relational database management systems, MySQL, PostgreSQL and Oracle database 11g using criteria such as differences in syntax and performance. It concludes that Oracle is the best option where speed and performance of complex operations is important, but that MySQL is a good open-source alternative if the cost of Oracle is prohibitive. This may help form a conclusion around which database technology to use for the web app and understand what the trade-offs are since there is no budget for the project. The paper was from a MIPRO conference and has been cited in other published works.

Patil et al (2017) explain the differences between relational databases (which feature linked tables) and non-relational databases (also referred to as NoSQL, it is a more flexible type, including in a document with JSON style structure) and then compares the performance of them primarily using MySQL (relational) and MongoDB (non-relational), among the most popular of each type respectively. They found significant performance advantages for the non-relational database for performing basic operations, as shown in figure 18 below, which shows the number of records inserted and the time each database takes to load them, with MongoDB being the faster of the two.

Chart, line chart

Description automatically generated

Figure Time taken to load records by two databases

Al-Refai et al (2021) lays out some of the challenges facing Database-as-a-Service (DBaaS) service model; unavailability, interoperability and confidentiality and proposes solutions to tackle them. If a DBaaS solution is used for the project, understanding the implications of that is important. The paper also includes its own literature review, which is a useful jumping off point to further reading on the subject. The paper was from an ACIT conference and has been cited by another published work.

The official documentation (e.g., from Amazon (2023g) and Openstack (2020)) for the DBaaS’s will be important for comparing them and deciding which one to use. Additionally, they will be a crucial resource for setting up and using it. Although the Trove/Openstack DBaaS was used in TM352, this had been setup ahead of time so some learning may still be required if that is chosen.

Security is another important consideration when using databases and is particularly crucial for this project as protecting users’ privacy is essential both from a moral point of view and to maintain user confidence. Mehak et al (2014) outline the challenges of DBaaS security in detail, including confidentiality, integrity, availability, and privacy. They conclude that further research is needed into the topic, and it should be noted that since this book was published in 2014 more recent literature should be considered to supplement this. The book ‘Cloud Computing: challenges, limitations and R&D solutions’ was described as a “comprehensive overview” by Beidler (2015) for Choice Reviews and the book and the relevant chapter has been widely cited in published works.

Some important factors when choosing database technology are whether to use a relational database or a NoSQL database; whether the database is free; familiarity with the database and the performance of the database. There are multiple solutions that could be viewed as equally correct, or with only marginal differences, so in some instances an arbitrary decision must be made because the time to consider the differences in detail could be used to on more important work. As currently conceived the database would not require interlinked tables since each service would only have properties associated with itself, so a better performing (Patil et al, 2017) NoSQL database such as MongoDB could be used. The OpenStack Trove DBaaS is free and open source, whereas an AWS solution may have costs associated depending on length of use and options chosen (Amazon, 2023g). Additionally, having already some familiarity with OpenStack Trove makes it a good choice for the project.

### 2.2 Requirements

The first part of the second literature review was relating to requirements including how to identify them, the differences between functional and non-functional requirements, their elicitation and analysis. This is crucial for the project, so that the goals and boundaries are clear, as are the expected outcomes and the characteristics it should have. The goal was to find sources that cover all these aspects of requirements, and unlike other literature reviews for this project the date when written is less crucial here, as this subject has not changed over time in the same way that a technology might.

Nilsson & Fagerström (2006) discuss the analysis of requirements once stakeholders have been consulted and how to balance their potentially competing interests to aid in the decision-making process. Furthermore, they outline a method of analysis considers different needs stakeholders might have and the extent to which requirements may meet them. They outline a method of collating all this in a ‘stakeholder and requirement matrix’ as shown in figure 19. This piece of literature is a research paper that was published in the Computers in Industry journal and has been widely cited by other academics, so can be seen as a trustworthy source.

A picture containing text, crossword puzzle, receipt

Description automatically generated

Figure Example Stakeholder and Requirements matrix

Robertson & Robertson (2006) provide a detailed overview of requirements, covering topics such as what requirements are, why they are important and the process of writing them. It also covers the Volere Requirements Specification Template that was pioneered by Robertson & Robertson with others, that forms a foundation and structure for requirements specifications. This book covers important foundational concepts and informed the TM354 module on software engineering, so can be considered a trustworthy source.

These sources will provide both different, complimentary ways to consider and analyse requirements. Both the stakeholder and requirements matrix and Volere template are useful tools that can be utilised, but care must also be taken that they are used in a proportionate way for this project. Using them to their full potential would take a very significant amount of time and as previously stated, so care must be taken to use shortened versions which will still be elucidating and provide context for discussion on future. The textbook for TM354: Software Engineering Block 1 Units 1-4 From domain to requirements (The Open University, 2014) does this with the Volere Template and so will be a useful resource as an example of narrowing this down.

### 2.3 CSS

The second part of this literature review was on sources related to skills development for CSS, to assist in the development of the visual elements of the user interface. Previous modules such as TM352 did not cover CSS in much detail, so the goal was to learn enough basic CSS to create a simple, but functional appearance for the app. Since the CSS language is being continually updated, more recent sources were preferred, though some older sources may still be of use since the basics of the language have stayed the same.

Gray with CSS Tutorial – Full Course for Beginners (2022), produced an extensive video tutorial for freeCodeCamp on CSS starting at the very basics including fonts and colour changes. It also covers grid layout and flexbox, as well as media queries which may be useful to ensure that the app maintains the intended appearance on different devices. The tutorial also covers accessibility issues, for example mentioning how code will affect the way screen readers will read the page, so it also will be a useful resource to ensure FreeCodeCamp is a charity founded by a teacher to provide free online courses teaching coding and has other learning materials that may be useful going forward for the project, for example on JavaScript and REST APIs, if required. Gray is a lecturer and PhD student at Fort Hays State University and has produced many teaching resources on web development. This source will be a useful starting place to learn the basics of CSS and could be combined with another source such as a textbook to fill out knowledge gaps as needed.

Meiert (2015) in ‘The Little Book of HTML/CSS Coding Guidelines’ provides guidelines for coding, explaining good practise and the reasons for it. For example, naming classes/IDs so they properly reflect the purpose of an element and are ‘as short as possible but as long as necessary’. This will help ensure that the code is consistent and easy to read, both for other people but also for myself as the project goes on, the amount of code expands and there is a need to return to code written months earlier. Similarly, the W3C markup validation service (2023) will help ensure the code is valid and using proper syntax.

There are numerous books on HTML and CSS which would be suitable for skills development for this project, so there must be a degree of arbitrary choice when considering which one to use. Web Design Playground: HTML and CSS the Interactive Way (McFedries, 2019) is a more recent one, ensuring that newer updates to the language can be covered if necessary. Additionally, the author has written many other books which have sold ‘over four million copies’ on the web development and related topics, so can be considered reliable source when covering an introductory topic. The Book covers the basics of CSS and HTML including topics such as pseudo-elements and how the cascade and inheritance work.

These sources will give a solid grounding in CSS, combining audio-visual and written learning to aid in the skills development required for the project. They will also help consider accessibility issues as development continues, to ensure that the app is still pleasant to use for people who use screen readers, have colour blindness or any other potential barriers to using the app.

The first literature review in section 1.2.1 concluded that a NoSQL database would be a suitable for the project, and that OpenStack Trove would be a good choice due to previous familiarity gained from TM352, as well as it being a free open-source option. Another possible choice considered was AWS (Amazon Web Services) which has a NoSQL database DynamoDB, but concerns were around cost as AWS charges for some of its services and there is no budget for this project. The familiarity with OpenStack Trove was with the use of the database and API calls, not its installation and setup, so some skills development was required first. This literature review will therefore consider sources for skills development for the setup and installation of a database and either consider whether the decision to use OpenStack Trove was sound or whether using the alternative option of AWS DynamoDB would be more suitable.

The official documentation for Openstack was part of the first literature review in 1.2.1, which provides information on how to install OpenStack Trove and how to setup a database. In addition to this, FreeCodeCamp previous cited in the second literature review in 1.2.2 also have a video guide for setting up and working with OpenStack Trove (OpenStack Tutorial – Operate Your Own Private Cloud (Full Course), 2022). If OpenStack Trove is used as a database, then this will be a good starting point for skills development in using the database.

The OpenInfra Foundation was formed to govern the OpenStack project and its mission is to help people “build and operate open infrastructure” (OpenInfra Foundation, n.d.). It hosts yearly summits which features industry professionals working with open infrastructure to give talks and presentations, which pertain to open infrastructure, and are hosted on their YouTube channel. Configure, Debug and Install OpenStack Trove (2016) is one of these presentations by Sadasiva Pillalamari & Rama Krishna Bhupathi, software engineers for Hewlitt Packard Enterprise, each with decades of experience in the industry. The presentation includes a demonstration of a non-relational database, MongoDB, which it was decided in the literature review in section 1.2.1 would be most appropriate for this project. This presentation and others from OpenIntra Foundation would complement the previous sources for skills development for using OpenStack Trove.

Having done the above research, one thing that became immediately apparent is that OpenStack is not available for Windows and must be installed on Linux, so would require either an installation of a dual boot or virtual machine (VM) for Ubuntu 16.04 and the installation would require the following steps:

1. Create a VM (or dual boot install) for Ubuntu 16.04
2. Install OpenStack using the Linux CLI (command line interface)
3. Install relevant packages including Horizon, the OpenStack GUI
4. Install Trove

While this is certainly achievable, it would require further skills development and refreshing knowledge for installing an Ubuntu VM and using the Linux CLI. Additionally, the process may take a significant amount of time and could stall the project if there was an unforeseen issue with VM installation. Given the time constraints of the project, it was considered prudent to investigate the alternative of AWS DynamoDB. To begin using AWS DynamoDB is very straightforward in comparison with OpenStack Trove, as it is a matter of creating an AWS account and confirming credentials, then DynamoDB can be interacted with through the AWS GUI on a browser. The concern was noted previously about cost; however, AWS provides a ‘free tier’ of 25GB of storage, 25 Write Capacity Units and Read Capacity Units, enough to allow 200 million requests per month (Amazon, 2023a), which would be more than sufficient for this project given the relatively small amount of data involved. Were the project to have a full release, this must be kept in mind so that unanticipated charges were not incurred if the amount of data stored or transferred were to grow significantly.

Skills development for the use of AWS DynamoDB therefore also needed to be undertaken, with the AWS documentation developer guide being a crucial resource. As it is provided by Amazon, it can be considered a definitive source and it features guides for setting up and using DynamoDB tables (the terminology used to refer to a database) and how the unique identifiers, the partition key and sort key work. One of the challenges of using DynamoDB is that the API’s need to be set up manually, as although AWS provides low level APIs these can be cumbersome to use, and it is recommended that using the AWS SDK (Software Development Kit) is a better approach. The documentation includes a step-by-step guide on how to set this up and how to use the SDK. The AWS SDK constructs the requests and converts the responses on your behalf, and the structure of the SDK between the application and DynamoDB is shown in figure 2 (Amazon, 2023b).

A diagram of a software application

Description automatically generated with low confidence

Figure High-level overview of Amazon DynamoDB application programming using the AWS SDKs

Two textbooks have also been identified to supplement the AWS documentation for skills development. Tankariya & Parmar (2019) provide a guide to passing the AWS Developer’s Certification, which has a chapter on DynamoDB featuring an explanation of the Query and Scan operations that may be useful for applying the service tags and allowing users to search for the services they need. Deshpande (2015) in the DynamoDB Cookbook explains from the basics of how to get started with AWS to designing applications and provides many ‘coding recipes’ which may provide useful jumping off points when implementing the code for the project. All these authors are experienced, active industry professionals and given the subject matter is relatively basic, it is reasonable to consider them reliable experts.

While conducting the literature review, the decision was made to switch from using OpenStack Trove to AWS DynamoDB as a DBaaS solution for the project. This was due to the time investment required to implement the OpenStack Trove Database, and the uncertainty around issues that may have arisen during the installation process including an Ubuntu VM. The sources outlined will assist in the skills development required to implement and use the DynamoDB database to move forward with the project.

Review credibility of sources in trove and dynamodb

Add source about CRUD?

Cut down trove/dynamo section, make it less narrative perhaps move some of that to project work

Review how decisions were made, go back to original DB decision to make it link up in retrospect

# 3. Account of Project Work and its outcome

## 3.1 User Interface

Three initial ideas for UI layout were roughly hand sketched in Figures 20, 21 and 22 with Figure 20 using Google Maps for inspiration, Figure 21 having menus that swipe in from the sides and Figure 22 being a much simpler and more basic. The advantage of the layout in Figure 20 is that it will be familiar to users, since Google Maps is such a ubiquitous app any app using a similar layout should feel natural and easy to use. The layout in Figure 21 would have a clean and pleasant interface, keeping it uncluttered, particularly for smaller devices. This style is quite common, including in apps such as Twitter and Discord. However, it runs the risk of users not realising the menu is there without some visual cue, so care must be taken to carefully gather feedback.

As previously stated, current experience with CSS and visual styling is limited and both the designs in Figures 4 and 5 may require some time learning how to implement them. Time was allocated in the schedule for skills development on CSS and the literature review in section 1.2.2.2 identified appropriate sources for this (sentence removed, as the skills development occurred, and some CSS was attempted). Figure 22 shows a more simplified layout that could have been used as an alternative if the skills development were unsuccessful or time constrains meant that attempts styling were limited.

Diagram, engineering drawing

Description automatically generated

Figure Rough sketch of UI using Google Maps as inspiration

A piece of paper with writing

Description automatically generated with medium confidence

Figure Rough sketch of UI with menus that swipe in

Diagram

Description automatically generated

Figure Rough sketch of UI in a simplified style

## 3.2 Requirements elicitation

Initially the preliminary project description was reviewed to identify some potential requirements, these would represent the outline of what the system as initially proposed would deliver. The next step would be to consult with stakeholders so that their feedback could potentially confirm they were correct, remove them entirely, or adjust them. These proposed requirements are outlined below, with functional requirements and non-functional requirements having the abbreviations FR and NFR.

The system shall:

* FR1: display services for the transgender community in Liverpool on a map.
* FR2: provide information and contact details for each service.
* FR3: have tags to show or hide the services displayed on the map.
* FR4: show directions to the location of a selected service.
* FR5: have a searchable database of services.
* FR6: display events related to the services or for the community in a calendar.
* NFR1: give the user control over privacy.
* FR7: display the location of businesses with gender neutral toilets on a map.

Using the classes of non-functional requirement identified by Robertson & Robertson, NF1 could possibly be considered a legal requirement (due to legal requirements of handling sensitive data) or a security requirement (due to maintaining confidentiality). However, it could also be described a cultural requirement since there are unique aspects to the trans community that go beyond what might normally be considered ‘privacy’, such as the name of the app and when notifications may occur.

Nilsson & Fagerstrom (2005) suggest constructing a ‘stakeholder and requirements matrix’ which can be used to show “a rich picture of all the stakeholders” and the relative importance of their needs. As previously discussed, the amount of feedback sought from stakeholders has been scaled down to be manageable for the scope of this project, so only a partial matrix will be constructed, but a full release that elicited more responses could flesh it out further. A questionnaire was therefore designed to elicit responses from stakeholders that would assist in both the creation of the matrix and to get feedback on the proposed requirements. Participants were asked how useful a feature (that that delivered a proposed requirement) would be to them; to rate the importance of it; and for further feedback, as shown in figure 23. A full copy of the questionnaire is included in Appendix I.

The Open University guidelines on conducting research with human participants were followed, with the research not being considered high risk as all participants being over the age of 18 with full cognitive capacity, the full knowledge and consent of all participants was gained, and the questions did not broach sensitive topics. The Consent Form and Participant Information Sheet, which explain what data is stored, what it will be used for and to gain consent are in appendices J and K (Open University, 2023a).

![Table

Description automatically generated]()

Figure Example question for stakeholder elicitation

When analysing the completed questionnaires, the goal was to see if any changes to the requirements were required; to identify the needs of a stakeholder group; to match how well a requirement meets that need and to be aware of any unexpected feedback that may elicit entirely separate requirements. There were three respondents to the questionnaire (Note: removed reference to section not in this TMA) who were potential users of the application, rather than service providers, so the analysis will focus on this single stakeholder group.

Responses indicated that FR1, FR2 and FR5 all were valid for this stakeholder group and will remain unchanged, some key quotes are:

* *“it would allow me to find what services are nearest rather than just what services are beat advertised”* (FR1).
* “Yes it would help to find what is accessible nearby especially if someone can’t access support during the day due to commitments or work” (FR2)
* “Yes, searching by the name/description of the service would be good” (FR5)

A stakeholder need can also be identified from this feedback:

* *N1.1: Finding a service that I need to access*

This is the core need that the app is attempting to fulfil, and all respondents rated this as of high importance to them. All respondents felt that there was some difficulty for them in finding and accessing the services that are available to them.

FR3 elicited a lot of suggestions about how it might be delivered, perhaps suggesting that it was vaguely worded. The feedback also indicated that users would value personalising their use of the app to tailor to their individual situation. As discussed in section 3.3, there are different experiences under the ‘trans umbrella’ that the app should try to cater for, such as transfeminine, transmasculine and non-binary. A key quote is:

* *“I wondered about a tag could be selected to specify services intended for specific people ie just trans women/men like make up support”* (FR3)

Therefore, FR3 has been amended to:

* *FR3: have tags to show or hide the services displayed on the map, based on type of service and the gender identities they provide for.*

Additionally, a stakeholder need can be identified from this feedback:

* *N1.2: Personalise the experience of finding services to my identity*

For FR6 there was a lot of feedback that users find it difficult to find out about events because they are spread across multiple places such as Facebook groups and discord servers. It highlights a problem that with this functional requirement that in delivering it there would be a risk that this would be simply ‘another place where events can be created’ and would potentially exacerbate the problem that users have rather than improving it. A key quote is:

* *“It’s hard to keep track of events, some are on Facebook, some by discord, so a central location would be lovely”* (FR6)

Therefore, FR6 has been amended to:

* *FR6: collate existing events for the community from other sources, in a calendar.*

A need can also be identified:

* *N1.3: Find out about community events*

Previously it has been stressed that maintaining privacy and safety of users is crucial to the project, both from a moral and reputational point of view. This community anxiety is highlighted by an unpromoted in response to the question about the filter tags:

* "not sure if this might allow terfs to find and cause problems for those services/users who attend that place?”

This is an important consideration for the app fundamentally, that while increasing the visibility of services is a positive thing for the trans community, that visibility may also increase for those who would do the community harm. However, this does not require change to NFR1 and other feedback indicated that this requirement was valid. A key quote is:

* “it would help with feeling safer particularly before coming out” (NFR1)

A need can also be identified:

*N4: Maintain my safety and privacy when using it*

Feedback also indicated that FR4 was not important to users, as they have other apps they use to give them directions. Additionally, many services require some interaction or planning before using them, so a user would not likely go to the service directly and instead make an appointment. A key quote is:

* “Not particularly useful to me as I use google maps for navigation. A simple link to a maps provider would do for me” (FR4)

Therefore, FR4 has been removed entirely.

Table 6 below is the resulting stakeholder and requirements matrix. The extent to which a requirement meets a need is expressed as either none, low, medium, or high. This was parsed from the priority that users gave in their responses but also by interpreting their written responses and an understanding of how the requirements and needs will interact.

Table Stakeholder and requirements matrix work in progress

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | FR1 | FR2 | FR3 | FR5 | FR6 | NFR1 |
| Trans community app users | N1.1 | H | H |  | H |  |  |
| N1.2 |  |  | H | M |  |  |
| N1.3 | L | L |  |  | H |  |
| N1.4 |  |  |  |  |  | H |
| Service providers | N2.1 |  |  |  |  |  |  |
| N2.2 |  |  |  |  |  |  |

Further analysis will be conducted to produce user stories from the feedback that has been gathered, and then all of this will be collated in a condensed version of a Volaire template. So far only one non-functional requirement has been considered, so more could be identified and as development continues further requirements may emerge.

## 3.3 misc work

#### Setting up the database and API’s

To begin setting up the database, the first task was to create a DynamoDB table (the terminology used for the database) and the APIs using the AWS SDKs, as explained in section 1.2.3. Amazon provides step by step instructions on how to build a CRUD API (i.e. with operations to Create, Read, Update and Delete), which also utilizes other Amazon services; Lambda and API Gateway (Amazon, 2023c). Firstly, a table is created in DynamoDB, then create a function in AWS Lambda which serves as the back-end of the API. Lambda runs code without provisioning or managing servers, and the code for the function is provided in the tutorial but then edited to reflect the table name and the ‘put’ operation to include the columns of the DynamoDB table. Figure 3 shows the edited put operation using placeholder attributes.

A screen shot of a computer code

Description automatically generated with low confidence

Figure PUT operation for the CRUD API edited with placeholder attributes

Then the HTTP API’s are created in AWS API Gateway, providing an end point for the function that has been created in Lambda. Routes are also created in API Gateway with integrations that connect with the Lambda function. This can then be tested using the CURL command line tool, and figure 4 shows the use of the PUT function adding 3 items to the DynamoDB table with the result shown in the AWS Dashboard GUI in figure 5 and with two GET requests in the command line, one for all the table and one for a specific ID. The item with ID = 124 is deleted using the DELETE command and the result is shown again using a GET request and in the GUI in figure 6.

![A picture containing text, screenshot, font, black and white

Description automatically generated]()

Figure Testing of the API's using the CURL command line tool

A picture containing text, receipt, screenshot, font

Description automatically generated

Figure Table in the AWS GUI after 3 PUT requests

A picture containing text, receipt, screenshot, font

Description automatically generated

Figure Table in the AWS GUI after a DELETE request for item with ID=124

#### Implementing the map and pins

With the Table created and API’s functioning, the map and map pins can be implemented and the APIs utilised to retrieve information from the database to place the pins in the correct location. Google Maps and HERE Maps are two reasonable choices for the project, however HERE Maps was chosen due to previous experience using it in TMA352. HERE provide numerous code examples, including some which illustrate how to initialise the map, how to centre it on a specified location, how to restrict the movement of the map within specified boundaries and place a marker at a specified location (HERE, 2023). These code examples were modified with map centred on Liverpool and a marker at a hard coded location. Accessing the API for the map requires creating an account with HERE, registering an app with them and then generating an API key which can be inserted in the code as shown in figure 7.

A screen shot of a computer

Description automatically generated with low confidence

Figure API Key for the HERE Map

The next step was to use a GET request to access the DynamoDB table to find location data for services and then put a pin at that location. However, in order to access AWS services valid credentials need to be supplied. There are multiple ways to do this, including hardcoding them into the code itself, which is not recommended, as it presents a security risk even if they were later removed due to version control systems retaining older versions of code (Amazon, 2023d). Instead, the recommended approach by AWS is to use two other AWS services, Cognito and IAM (Identity and Access Management) (Amazon, 2023e; Amazon, 2023f). Cognito identity pools can be used to generate temporary credentials for the users of the app, which are applied for the role created in IAM. In this case the intention that users will not require accounts, so the users will be unauthenticated, but if that were to change these services could be used to authenticate users. Once this has been setup then the AWS SDK for JavaScript was included in the HTML file as shown in figure 9, and the Identity Pool ID and region supplied to the AWS config as shown in figure 9. This allows the app access to AWS services, such as DynamoDB.



Figure AWS SDK in the markup

A screen shot of a computer code

Description automatically generated with low confidence

Figure AWS Config submitted credentials to allow access to services

A function was then written called getServicesFromDatabase (figure 10) that utilised the API with a scan request to retrieve all the data from the DynamoDB Table, with the data fed into the addMarkersToMap (figure 11) function which has been modified from the example provided by HERE to accept latitude and longitude in the argument. This adds the markers to the map at the given locations and figure 12 shows the result shown on the app with two markers placed after being retrieved from the database.

A screen shot of a computer program

Description automatically generated with low confidence

Figure getServicesFromDatabase function scanning a DynamoDB table to retrieve all data

A picture containing text, screenshot, font, line

Description automatically generated

Figure addMarkersToMap function accepting latitude & longitude in the argument

A map of a city

Description automatically generated

Figure Two markers added to the map after their data has been retrieved from the database

#### 1.3.2.3 Implementing tags and info sidebar

In order to provide information to the user, when the map markers are tapped they should display information about the service. Arbitrary data can be associated with the marker, so the getServicesFromDatabase function is modified to pass the partition key (the unique identifier) for that service to the addMarkerstoMap function, which is further modified to instead add the markers to a container (and renamed to addMarkersToContainer, shown in figure 13). This is so that an Event Listener can be created for all the objects in the container rather than for each individual marker. When a marker is tapped the Event Listener retrieves the partition key and uses it to do a getItem request to retrieve the data and inserts that data into the info bar. This is wrapped in an ‘if’ condition to check if the selected service is already displayed so that unnecessary requests to the database are not made. The Event Listener is shown in figure 14 and the result in figure 15 shows the name of the service InTrust correctly retrieved from the database and inserted into the info sidebar replacing the placeholder ‘Support Group’. Currently only the name has been added to the services for the database, but future work will involve fleshing this out.

![A picture containing text, screenshot, font

Description automatically generated]()

Figure addMarkersToContainer function adds marker objects to a container with arbitrary data

![A screen shot of a computer code

Description automatically generated with low confidence]()

Figure Event listener retrieving the single item from the database associated with the tapped marker

A picture containing text, screenshot, font, rectangle

Description automatically generated

Figure Name in the info sidebar changed from the placeholder to the one retrieved from the database

To implement the tags that would filter the services by type, firstly the html markup was changed so that the placeholder buttons were replaced with checkboxes so that the user can mix and match which services they would like to be displayed at once. Additionally, the scan operation which was previously used to return all items in the DynamoDB table can also have filter conditions applied so that only specific results are returned (Tankariya & Parmar, 2015). Each time a checkbox is checked or unchecked, it triggers a function checkService (shown in figure 16) which checks which tags are selected. It then constructs a JSON object with the types of service which the user wishes to be displayed and creates a Filter Expression which uses an IN operator to check for the checked services in the ‘type’ attribute in the DynamoDB table. All the current markers are removed, and the Filter Expression and JSON object are sent with scan request, with the results being added to the map in the same way as described previously. The result is shown in figure 17 with the tags for peer support and hair removal selected and those services being displayed on the map.

![A screen shot of a computer code

Description automatically generated with low confidence]()

Figure checkService function constructing a filter expression to submit to DynamoDB based on user selected tags

![A screenshot of a map

Description automatically generated]()

Figure 17 The result of tags being selected by the user with peer support and hair removal markers correctly showing on the map

Write up work done after tma03

Link up all UI work

Make sure the ‘why’ of what is happening is included, rather than just ‘what’ and ‘how’ – why is the database being done this way, use of css grid. Fundamental concepts

Citation for CRUD

Citation for using HERE map in TM352

Volere template for requirements

Move some things to appendices

# 4. Review

## 4.1 Current stage of project work

## 4.2 Review of project management

### 4.2.1 project lifecycle

The project lifecycle was initially conceived of as being a user-centre lifecycle with regular feedback from users in the trans community and an extensive use of prototyping. The first part of this that was planned was the requirements elicitation that was undertaken and documented in section 1.3.1.2. The plans for this were scaled back due to tutor feedback, as it was felt that collating and analysing all the data would be very challenging and time consuming. Additionally, the process of gathering feedback proved more challenging than anticipated than when the lifecycle was chosen, with 8 people agreeing to provide feedback, but only 3 ultimately completing the questionnaire. The exercise was ultimately successful and provided much useful information for the project, but it clearly indicated that a rethink was required in terms of the project lifecycle.

Further to this, conducting prototyping would take up significant amounts of time in designing it, conducting the research, and collating the feedback. The original plans were in retrospect a little ambitious in terms of the amount of time that some things would take and once the project was well underway it became apparent that the amount of time investment required to undertake the prototyping would mean that there would be significant sacrifices that would have to be made elsewhere in the project. Also, given that some user feedback had already been undertaken with the requirements elicitation, it was ultimately felt that this was not the highest priority.

For a full release of the app, feedback gathering and prototyping would be an essential part of the development. As a result, the project lifecycle has changed considerably over the course of the project and came to closer resemble an iterative waterfall with each task being evaluated and being iterated on as they were ongoing due to the nature of most of them being unfamiliar. Additionally, it resembled some agile methods with a serious of sprints producing a TMA as a deliverable, with a retrospective conducted via a 1-to-1 meeting with the tutor discussing the feedback for that TMA.

### 4.2.2 Resources, skills, and activities

The resources focused on previously involved gathering feedback from services and members of the community, and while this is very important for a full release, for the purposes of this project it has been scaled back. While feedback is important, too much time focused on it would not leave enough time for other aspects of development. This aspect of the resources list has been condensed and other essential resources added. The full list of resources is in Appendix G.

* Representatives of services e.g. manager of a laser hair removal clinic
  + Can offer feedback about the events system and the ways in which users can contact their service.
  + May offer vital perspectives of how the app effects the community
* Members of the Liverpool trans community
  + Different members of the community may have different needs regarding accessing services.
  + Could offer feedback on prototypes of the app.
  + May be a source of information about services that should be included.
  + The Liverpool Trans Wiki may an essential source of community information about the services, as it documents many of them.
* Programming languages
  + JavaScript – proficient enough to tackle much of the proposed features, but some learning may be required as it will likely extend beyond current knowledge.
  + HTML – reasonably proficient, significant problems are not anticipated.
  + CSS – only minimal experience, may need some time to learn and to use trial and error to achieve goals.
* Cloud storage and version control
  + Microsoft OneDrive and GitHub can be used to backup files in the cloud and continue work between different devices. GitHub also provides version control with branching and reverting.
* DBaaS
  + AWS DynamoDB – A serverless database solution from Amazon that has a limited free access that is accessed through an online GUI via a browser.
  + Other AWS services to combine with DynamoDB such as Lambda, API Gateway, Cognito & IAM.
* Map
  + HERE Maps – Map solution accessed using an API after creating an account with HERE.
* Hardware
  + PC – used for coding, testing, writing up reports and conducting research.
  + Laptop – as above, but with portability. More focus on writing up reports, since other tasks are easier with a larger screen.
  + Phone – used for support with research, but also to test the app on a mobile device.
* Software
  + Chrome – Browser used for testing with developer tools, for investigating issues with code and to see how the app responds to different resolutions.
  + Visual Studio Code - A commonly used code editor that will be used throughout the project for all coding purposes.
  + CURL (Client URL) – for testing and using API’s using the command line.
  + Discord – a bot can be used to gather information for the events calendar.

Review list in light of final project work done

Reword so it makes sense in this context

### 4.2.3 Risk management

The goal of risk management is to identify, mitigate and minimise risks prior to them becoming a threat to the successful completion of the project (Hughes, 2012). The first part of managing risk is to identify and assess potential risks, considering how likely they are to occur and the severity of their impact. Table 4 (Open University, 2023b) considers risks for the project and labels them so they may more easily be referred to for the risk management process.

Table Project-specific risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Label** | **Project Activity** | **Risk Description** | **Likelihood (low, medium, high)** | **Impact (low, medium, high)** |
| R1 | Feedback from service providers for requirements elicitation and prototyping | Providers may be busy and not wish to engage with giving feedback or may only engage in a limited way. | Medium | Medium |
| R2 | Feedback from service users for requirements elicitation and prototyping. | The feedback given by users may be not useful or relevant if questions don’t properly direct users or may miss vital feedback if questions are too closed. | High | Medium |
| R3 | Feedback from service users for requirements elicitation and prototyping. | Users may not wish to engage with giving feedback or may only engage in a limited way. | Low | High |
| R4 | Coding the application. | Attempting to code non-routine tasks may cause significant delay. | High | Medium |
| R5 | Using Visual Studio Code | Since this is the first time using this software, there may be some adjustment time to using it causing delays. | Medium | Low |
| R6 | Storing data in the cloud | Data loss in the cloud, which could be due to several reasons including accidental deletion and server failure. | Low | High |
| R7 | Installing and setting up a database | The task may take longer than anticipated, since previous work with databases was after it had been set up. | Medium | Medium |
| R8 | Unexpected Illness | Time could be lost to illness, including potentially at a critical time such as in the run up to a TMA deadline. | Low | Medium |
| R9 | Hardware failure | A sudden mechanical failure requiring the repair or replacement of a piece of hardware | Low | Medium |
| R10 | Undertaking project tasks in the run up to the EMA | Tasks such as the Events Calendar prove more time consuming than anticipated and could impact the amount of time spent on the final EMA write up. | Medium | High |

The next stage of managing risk is to identify which risks are most important to address, so the likelihood and impact of each risk have been combined in a probability impact grid, as shown by Hughes, but instead of a ‘Line of Tolerance’, the grid (shown in table 5) has been colour-coded to indicate the most crucial risks to address with red and orange being most critical and to be addressed with urgency and in detail; yellow important to be addressed as much as feasible; green being least important and the risk may be accepted or only convenient mitigations undertaken. Risks 2 and 4 have been identified as the most crucial and mitigating them will form notable parts of the project; whereas risks 5, 8 and 9 have been identified as the least crucial and will be accepted without mitigation.

Table Probability Impact Grid

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact** | *High* | 3, 6 | 10 |  |
| *Medium* | 8, 9 | 1, 7 | 2, 4 |
| *Low* |  | 5 |  |
|  | *Low* | *Medium* | *High* |
| **Probability** | | | |

Appendix H contains the table showing the mitigating actions that were taken for each risk and explains what happened. In most cases the mitigating actions were effective, but risk R1 still occurred as far fewer people completed the feedback questionnaire for requirements than anticipated and the mitigating action of identifying likely respondents was unsuccessful. This resulted in adjusting the project and analysis to compensate and is discussed in sections 1.3.1.2 and 2.2.

Table Mitigating action and the results of identified risks

|  |  |  |
| --- | --- | --- |
| **Label** | **Mitigating Actions** | **Result** |
| R1 | Identify people most likely to provide feedback and be respectful of their time to get as much out of any engagements as possible | The most likely service provider to provide feedback was identified and engaged with, but personal reasons meant they had to drop out of the requirements elicitation study. Since mitigation failed adjustment to the analysis was necessary, so the requirements elicitation still went ahead without this stakeholder group and future work suggested to engage with other stakeholders. |
| R2 | Questionnaire to be carefully designed to facilitate getting high quality feedback. | The feedback was good quality and useful information was gained, so mitigation of carefully considering the questionnaire was successful. |
| R3 | Undertake feedback gathering as soon as possible, attempt to identify most likely candidates to participate and consider changes to project lifecycle if necessary. | This partially occurred, with less users participating than anticipated; however, this only had a minimal effect on the analysis and changes to the project lifecycle were made to reduce reliance on gathering feedback. |
| R4 | A skills development plan will be implemented, particularly for CSS & databases. | Skills development was undertaken for CSS and using the AWS services such as DynamoDB. |
| R5 | No mitigation to be undertaken. The primary purpose of the software is very similar to other code editors, and any shortcuts learned will only speed up tasks. | Over time familiarity of VS Code has resulted in increasing the ease of use. |
| R6 | Data will be stored locally on two machines (PC & laptop) as well as in the cloud, so there is no single point of failure. | As of writing, no data loss has occurred. |
| R7 | Begin working on this early in the project to ensure there is time to work through any issues. | This occurred due to the time scale involved of setting up Trove, so there was a switch to AWS. This was mitigated by starting the databases relatively early in the project, so disruption was minimised. |
| R8 | No mitigation to be undertaken but contact with tutor to be maintained so that an extension can be requested if necessary. | Unexpected illness caused a week’s delay around the time of a TMA deadline, but an extension was requested and granted, so disruption was minimal. |
| R9 | No mitigation to be undertaken, if either the PC or laptop were to suddenly fail then the other could be used as backup. | As of writing, no hardware failure has occurred. |
| R10 | Discuss the remaining tasks with tutor to prioritise and set a hard cut off after which no further work on the app will be done. | N/A |

As the final deadline for the EMA approaches, the risks R6 and R9 become increasingly important as data loss or losing the ability to write up the report could be catastrophic. However, the mitigation of using GitHub makes the likely impact of sudden data loss to be only a single day or half a day’s work as the version control is used to submit data after each session to ensure that all devices are up to date. Risk R10 is also crucial for the next phase of the project to ensure that there is enough time to write up the final EMA report, so this risk will have to be closely monitored in the coming weeks.

Reword to make sense in context, particularly risks for TMA03->EMA

## 4.3 Personal development

During the project two skills development activities were undertaken, firstly for using CSS to style the app, as detailed in sections in 1.2.2.2 and 1.3.1.3; and for using and setting up AWS DynamoDB as detailed in sections 1.2.3 and 1.3.2.1. These activities were crucial in the progress of the project, particularly using DynamoDB as without a database the application would not function. The skills development is still at an introductory level for both these topics but does provide a springboard for working on them in more depth in future.

Ongoing communication with the tutor has been a crucial aspect of maintaining progress with the project and has been maintained well throughout the project with updates provided at on at least a 2-weekly basis throughout most of the project. Personal issues made this more challenging during April, and ultimately resulted in an extension being required for TMA02. After each TMA a 1-to-1 meeting with the tutor has been organised, as well as attending or watching the recording of the tutorials. Additionally regular questions have been asked of the tutor when assistance has been required. A record of these interactions has been recorded in Appendix F.

Table Table of communications

|  |  |  |
| --- | --- | --- |
| **Date** | **Format** | **Details** |
| 26/1/23 | Email | Introduction and link to forum post with initial project idea |
| 27/1/23 | Zoom chat | Discussed further thoughts on the initial idea |
| 30/1/23 | Zoom chat | Agreed Zoom meeting |
| 2/2/23 | Zoom meeting | Discussed initial ideas and how to move forward to TMA01 |
| 9/2/23 | Tutorial | TMA01 |
| 11/2/23 | Zoom chat | Updated tutor with progress |
| 17/2/23 | Zoom chat | Asked for help with finding information about a database used in a previous module |
| 28/2/23 | Zoom chat | Updated tutor with progress |
| 10/3/23 | Zoom chat | Question about legal/ethical implications of requirements elicitation |
| 16/3/23 | Zoom chat | Arranged zoom meeting to discuss TMA01 feedback and asked a question about anonymity with requirements elicitation |
| 16/3/23 | Zoom meeting | Discussed feedback from TMA01 and how to move forward to TMA02 |
| 22/3/23 | Zoom chat | Sent consent form and participant information sheet to get feedback |
| 24/3/23 | Zoom chat | Sent consent form and participant information sheet again after acting on feedback to get the go ahead to proceed |
| 4/4/23 | Tutorial | Watched the recording of the TMA02 tutorial as I was unable to be present at the live event |
| 8/4/23 | Zoom chat | Update on progress |
| 23/4/23 | Email | Extension request for TMA02 due to personal circumstances |
| 9/5/23 | Zoom chat | Update on progress and arranged zoom meeting |
| 12/5/23 | Zoom meeting | Discussed feedback from TMA02 and how to move forward to TMA03 |
| 16/5/23 | Zoom chat | Update on progress and asked a question regarding setting up API’s in AWS |
| 31/5/23 | Zoom chat | Update on progress |
| 5/6/23 | Zoom chat | Update on progress |
| 8/6/23 | Tutorial | TMA03 |
| 16/6/23 | Zoom chat | Update on progress |
| 22/6/23 | Zoom chat | Asked a question on referencing for the AWS documentation |
| 28/6/23 | Zoom chat | Update on progress |

Does this section need more detail on skills development?

Is the tutor interaction table needed, if so update it

# 5. Epilogue

What goes here?

# 6. References

McAuley, P. (2022) ‘Trans lives 'at risk' in Liverpool, say support group claims’, *Liverpool Echo*, 24th November [Online]. Available at <https://www.liverpoolecho.co.uk/news/liverpool-news/trans-lives-at-risk-liverpool-25594158> (Accessed 4th July 2023).

TransLiverpool Wiki (2023) *Trans Liverpool Wiki* [Online], 7th April 2022. Available at [www.transliverpool.com](http://www.transliverpool.com) (Accessed 29th April 2023).

# 7. Appendices