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

Lucy Alexandria Morris – G4272527

TM470 – TMA02 – 25th April 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. Preparation and planning 3](#_Toc130589460)

[1.1 Title and scope 3](#_Toc130589461)

[1.1.1 Title 3](#_Toc130589462)

[1.1.2 Revised description 3](#_Toc130589463)

[1.2 Resources 3](#_Toc130589464)

[1.3 Future plan 3](#_Toc130589465)

[2. Project work completed 4](#_Toc130589466)

[2.1 Literature review 4](#_Toc130589467)

[2.1.1 Requirements 4](#_Toc130589468)

[2.1.2 CSS 4](#_Toc130589469)

[2.2 Project Work 4](#_Toc130589470)

[2.2.1 Requirements elicitation 4](#_Toc130589471)

[2.2.2 User Interface design 4](#_Toc130589472)

[3. Review and reflection 4](#_Toc130589473)

[3.1 Ways of working 4](#_Toc130589474)

[3.2 Evaluating project management 4](#_Toc130589475)

[3.3 LSEP and EDI 5](#_Toc130589476)

[4. References 5](#_Toc130589477)

[5. Appendix 5](#_Toc130589478)

# 1. Preparation and planning

## 1.1 Title and scope

### 1.1.1 Title

Same as before

### 1.1.2 Revised description

Specificity about what the finished product will look like and how it can be accessed.

Scope – what is in and out of scope

Make definitive decision on user registration – (maybe based on feedback?)

## 1.2 Resources

Paring back of feedback part – overambitious for this project, but would be appropriate for a full roll out of the app. Should be covered by a future work section.

Development of skills for JS, CSS, SQL

Devices – mobile, PC, laptop, tablet

Visual Studio Code

Cordova & Github

Hosting services

Online backup

Risks for cloud, risk of data loss

## 1.3 Future plan

Restate tasks/subtasks, updated and with feedback from test users

More detail on subtasks

Revisions to plan for TMA01->TMA02

Revisions for TMA02->

# 2. Project work completed

## 2.1 Literature review

### 2.1.1 Requirements

The first part of the literature review was on sources relating to requirements gathering and analysis.

3-4 sources, (include TM354 text book & volaire template??)

### 2.1.2 CSS

The second part of the literature review was on sources related to CSS, so that some learning could take place and the development of the user interface could begin. Prior to this project my experience using CSS was quite limited since the TM352 module did not include anything on it. 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 (2022) produced an extensive video tutorial for freeCodeCamp on CSS starting at the very basics including fonts and colour changes. It also covers the grid model and flexbox, as well as …

This will be a good starting point to learn some CSS in an accessible way, and could be combined with textbooks to fill out any particular points as necessary. FreeCodeCamp is a

3-4 sources

Youtube series?

Textbook?

Look up key dates to account for recent developments

## 2.2 Project Work

### 2.2.1 Requirements elicitation

Some requirements were initially identified by reviewing the project description:

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.
* FR7: give the user control over privacy.
* FR8: display the location of businesses with gender neutral toilets on a map.

The next stage was to consult with stakeholders to see if these requirements match with their needs, and how different stakeholders might prioritise different requirements to meet them.

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.

Appendix X contains a full copy of the questionnaire that was sent to stakeholders. The questions were designed to line up with each of the requirements previously identified with a feature described that would meet that requirement. The participants were asked how useful the feature would or would not be to them; to rate the importance of it; and if anything could be added to it, as shown in figure x. Some features had additional questions specific to it, for example with the events calendar participants were asked about how they would feel about ways they might interact with it (e.g. if they had to add the events themselves). The goal was to then parse needs that stakeholders might express in their answers and then line up their importance

![Table

Description automatically generated]()

Explanation of questionnaire and choices for elicitation

Refer to lit review

Analysis of results of elicitation

Volaire template

Needs/requirements matrix

Limits of elicitation and what would be needed for a full release

### 2.2.2 User Interface design

During the initial stages of development, Microsoft OneDrive was used as a simple form of version control, to make it easier to keep everything updated when working on the project on different devices. This was done by saving separate folders for the project code noting the date, as shown in figure x. The alternative of using GitHub was not felt necessary at that time, since the project has only a single developer and shouldn’t require much, if any, branching. However, after some several days, it started becoming difficult to keep track of which version was the most up to date as there were three places where the code was stored (PC, Laptop and OneDrive) that didn’t automatically sync up. Therefore, at this point GitHub was used instead, as it is much simpler to keep everything up to date on all devices and branching could be used to preserve specific development states, which might need to be referred to for reflection and analysis. Figure x shows the use of GitHub and some early commits.

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

On reflection, since CSS did not feature in TM352 and would therefore involve some learning, the schedule was changed so that some of the CSS would be tackled first to give more time to deal with any issues that might arise. Initially, the YouTube video by FreeCodeCamp (reference) provided a basis for learning some basic CSS and then referencing textbooks as needed such as xxx (reference).

The goal was to gain enough knowledge to produce a simple, but effective layout that would maximise screen space to ensure that information is legible to the user. Previously three types of designs were considered, a design similar to google maps, a design with menus that slide in from the sides and a very simplified design and the conclusion was that a balance would have to be struck between investing time on learning how to produce these designs and spending time on other critical aspects of the project.

The first task was to create a simple menu bar and decided to start by using an unordered list to do this, as shown with the markup in figure x. Initially a horizontal bar was attempted, but on reflection this may have taken up unnecessarily screen real estate. So instead, this adapted into a drop down ‘hamburger’ menu utilizing a hidden checkbox as shown in figure x. While this would be adequate, it seemed relatively simple to change this to have it instead pop in from the side rather than drop down to emulate the ‘sliding menus’ design. This opens and closes by tapping/clicking the ‘hamburger’ icon, though the original concept conceived of it also opening using swipe gestures, this is something that could be added later in the project.

The same technique was used to add an info box that pops out from the other side which would contain the information and contact details about a service selected from the map. Since the scripts to implement the map and pins have not yet been implemented, a visible checkbox was placed in the header to use as a placeholder trigger for it to pop out. As with the navigation menu, the use of gestures to interact with the info box could be added later in the project.

Issues – having menus not overlap the viewport; sizing of info box and flexing of the elements.

HTML template

CSS learning and attempts at different styles

Lots of images of development

Code snippets

# 3. Review and reflection

## 3.1 Ways of working

Things that have gone well/badly

Effective ways of working and factors preventing progress – day planner?

Ask craig – should I discuss personal issues here? To what extent?

## 3.2 Evaluating project management

Project lifecycle

Ask craig – should I address the feedback from TMA01 on lifecycle here?

## 3.3 LSEP and EDI

LSEP in context of questionnaire, anonymity and data handling

EDI considering how the app will be safe for LGBT (also gender/racial bias)

Also consider accessibility issues (ideally look at this while developing UI)

While developing the user interface, accessibility was considered throughout in several ways. As Gray states in his tutorial (reference), 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 ‘rem’ unit, so that the font size is relative to the font size of 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 lowest in order. In a situation when a browser cannot display the desired font or a character from the font, it will attempt to display the next listed font and so on. By having a ‘web safe’ font listed last i.e. a font that is ubiquitous across browsers and devices (reference?), the text should always display even if it is less aesthetically pleasing.

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 below shows this change.

.side-menu {

    /\* display: none; \*/

    opacity: 0;

}

.side-menu:focus +.hamburger{

    background: #85888c54;

}

# 4. References

# 5. Appendix