Solution document

EMA report-style document to support the development of the OU Walking Club project; explaining concepts such as Wireframes, Responsive Design, Authentication and Authorisation, Acceptance Testing and Version Control.

Harriet Craven

F5256798

OUCU: zy740452

TT284

EMA Part 2

Date: 3rd June 2019

1a)

Included in a ZIP archive named zy740452\_Q1.zip

1b)

The role of Wireframes

The role of wireframes in web development is to help plan the design and structure of a website, looking at how content is placed on a page and how multiple sections could work together elegantly. The Open University (2018) says wireframes are meant to be a “rough sketch” so they are easily edited. Wireframes are free from clutter and detail, designing only the website’s framework, without distractions from images, fonts or real text.

Segue Technologies (2016) states wireframes are also an important “communication tool” for designers, developers and clients. They help create a common understanding by clarifying what clients say, and what exactly they want, making unexpected changes further down the line less likely. Clients can not only see how the pages are going to look, but how they could carry out specific tasks as wireframes are often interactive. This allows users to get a ‘feel’ for the website, clicking around as they would on the real website.

Wireframes can also show how content would be displayed on different size devices, helping determine which content is prioritised and what devices should be considered supporting. Not every page of a website is included, but enough so the website’s core message is delivered.

The design of these wireframes has been influenced by Nielsen’s usability definitions (NN/g, n.d.). The “hierarchical structure” of the site provides a logical and intuitive global navigation between sections. (The Open University, 2018) These sections group user requirements, helping the website’s learnability. The prominent breadcrumbs banner at the top of the page shows users where they are at any time, making content accessible.

Pages are consistent to aid memorability and site identification; the logo is at the top left and the log out button top right, as expected. The theme incorporates the Open University’s branding and there are dividers, white-space and image placeholders to break up content, making the design satisfying. Additionally, the ‘Log In’ page ensures only the admin team can access functionally, a necessary security requirement.

The tables are self-explanatory regarding the naming of buttons and columns, helping the efficiency of the website and removing the need to work with MariaDB directly. For example, the ‘Membership’ table includes when each member was last active; helping the admin team mark inactive members, and the ‘News Items’ table includes the item’s visibility on the club website, keeping the admin team in sync with club members.

(400 words)

References:

1. The Open University (2018) ‘Paper wireframes’ [Video], *TT284 Web Technologies*. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1345887&section=2.1.1> (Accessed 2 May 2018).
2. Segue Technologies (2016) *The Importance of Wireframing for a Responsive Website* [Online]. Available at <https://www.seguetech.com/the-importance-of-wireframing-for-a-responsive-website/> (Accessed 5 May 2018).
3. The Open University (2018) ‘8.4 Hierarchical structure, *TT284 Block 1 Part 3: The process of designing for a usable and navigable web* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1364829&section=7.4> (Accessed 2 May 2018).
4. Nielsen Norman Group (n.d.) *Usability 101: Introduction to Usability* [Online]. Available at <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (Accessed 1 June 2018).

2a)

Included in a ZIP archive named zy740452\_Q2.zip

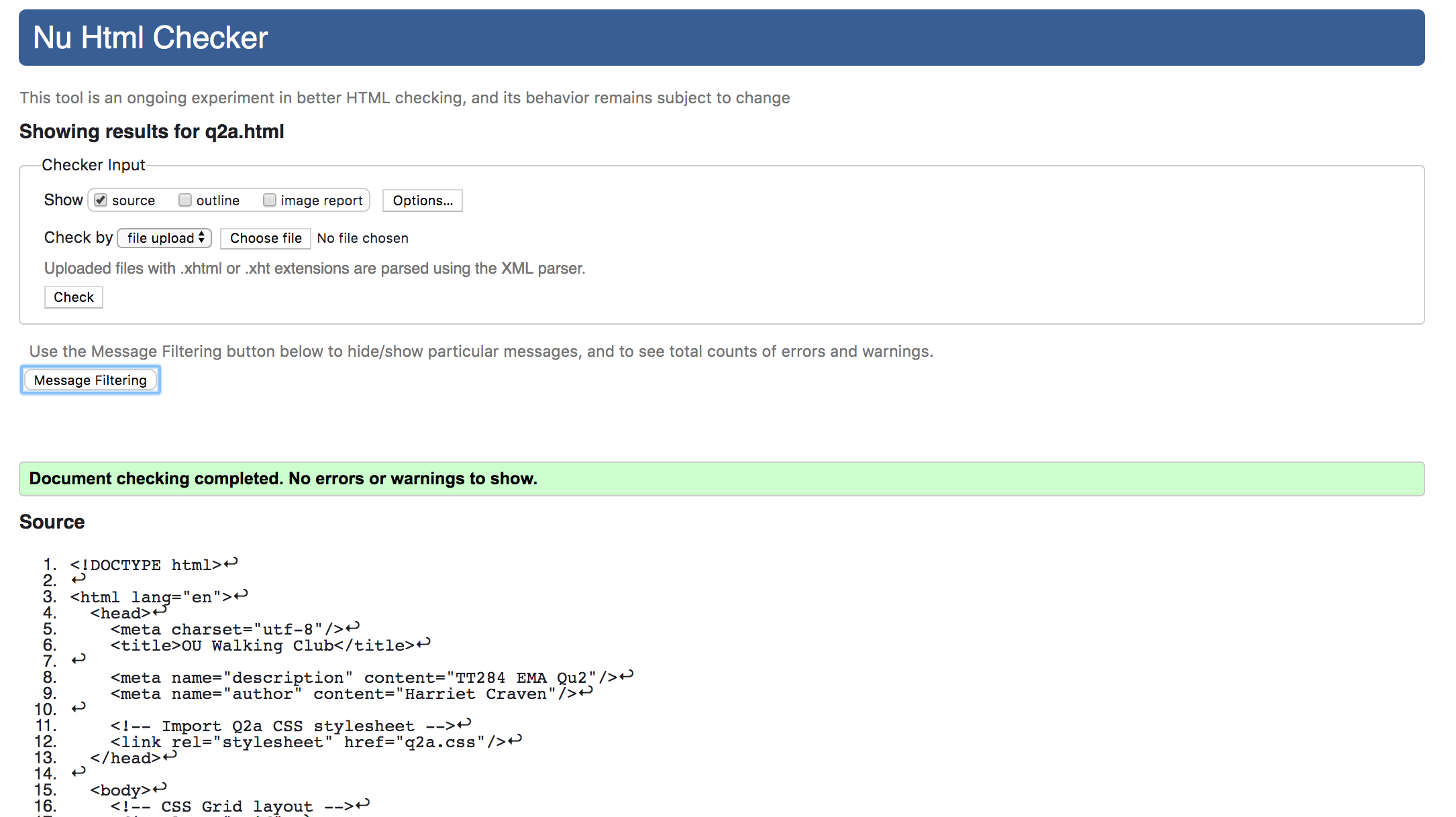


Figure 1. A screenshot of the HTML validator showing ‘Document checking completed’ for my q2a.html file.

2b)

Included in a ZIP archive named zy740452\_Q2.zip

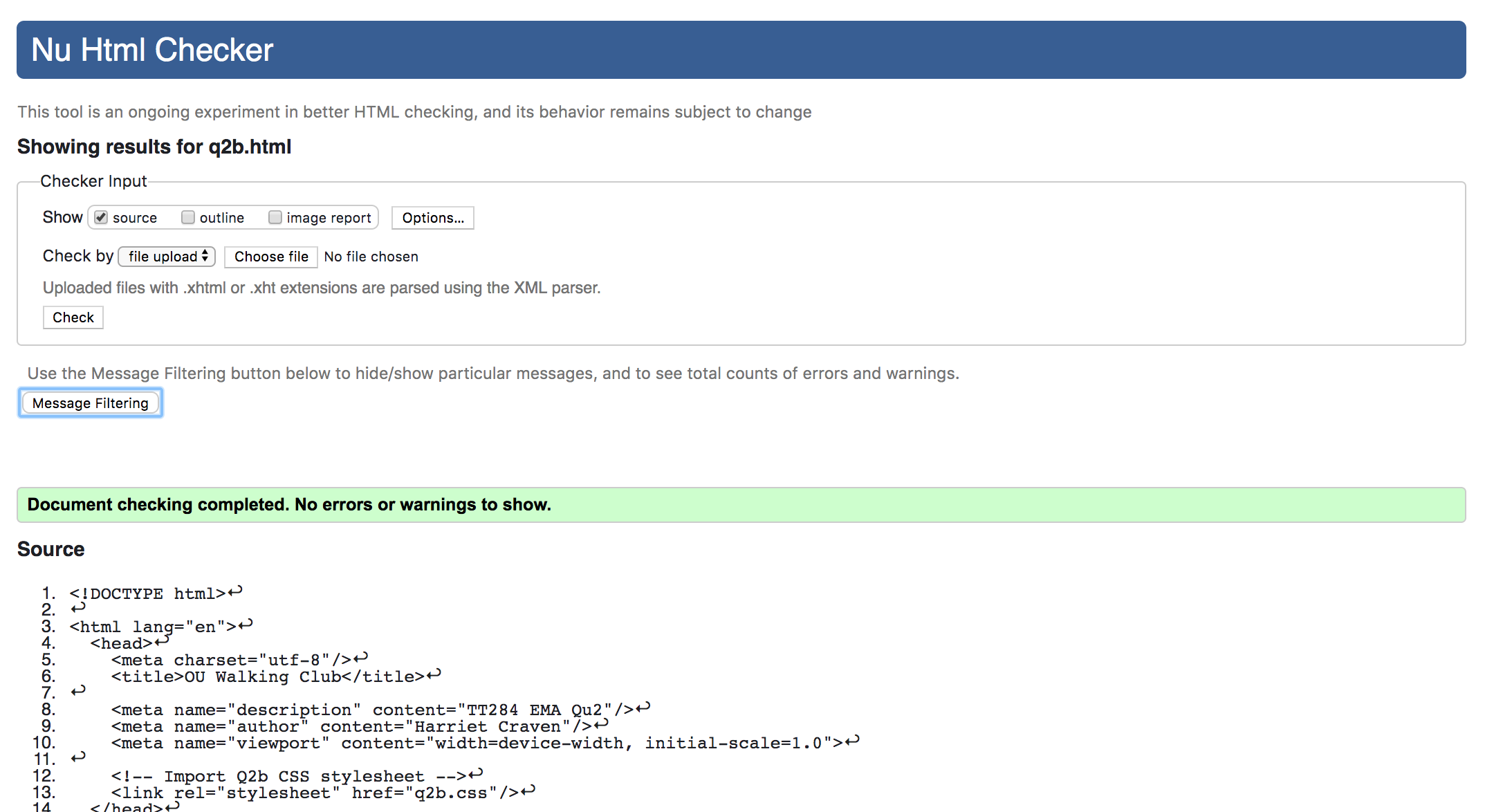


Figure 2. A screenshot of the HTML validator showing ‘Document checking completed’ for my q2b.html file.

2c)

Response design for the Admin Application

To the HTML file I added a meta viewport tag, giving the browser instructions on how to control the “page’s dimensions and scaling". (W3Schools, n.d.) I added a media query to the CSS which overrides the styles for screens up to 375px wide. (The Open University, 2018) This included grid styling so a mobile would have a single column and I moved the form inputs below their respective labels, as these are standard for mobile devices. (Holst, 2013)

I removed user credentials, page description and breadcrumb elements as I assumed another form of menu would be added later. I comprised on white space as the form is the width of the screen, but removed unnecessary elements such as dividers to declutter the page. I made elements smaller throughout, including the text font and buttons, but I kept the logo and navigation panel for consistency.

I found it challenging to prioritise content, deciding what should be presented first. I also struggled to decide the size of elements, for example the create button, so I made it the width of the page because ultimately, that is the users goal.

I recommend the club website is made responsive to support mobile devices as users will likely interact with the website from their phone, during a walk for example. The website could make use of notification features; alerting a user of upcoming events. The inbuilt camera could upload event photos or the geolocation sensor could select a meeting location and track the walk. (The Open University, 2018)

A downside for developers is the time and effort required to make the website responsive. However nowadays, external websites are expected to be usable on a variety of devices. The Open University (2018) states touch screen input may be “cumbersome”, if trying to propose multiple events and the dependency on internet connection unstable. Nevertheless, I think it is beneficial to make the website responsive.

Contrastingly, I don’t think it is necessary for the admin application. I assume admin input will likely be during the work day on a desktop computer, allowing easier “data entry and manipulation, analysis and planning”. (The Open University, 2018) Larger screens with external peripherals are safer to use for a longer period of time, and the desktop would have reliable internet connection. Time could be better spent elsewhere, instead of the admin applications front end.

(398 words)

References:

1. W3Schools (n.d.) *HTML <meta> Tag* [Online]. Available at <https://www.w3schools.com/tags/tag_meta.asp> (Accessed 10 May 2018).
2. The Open University (2018) ‘5.2 CSS media queries’, *TT284 Block 3 Part 2: Responsive Design* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1403137&section=5.2> (Accessed 15 May 2018).
3. Holst, C. (2013) *Mobile Form Usability: Place Labels About the Field* [Online]. Available at <https://baymard.com/blog/mobile-form-usability-label-position> (Accessed 17 May 2018).
4. The Open University (2018) ‘6.3 Sensors’, *TT284 Block 3 Part 1: The Mobile Device* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1403058&section=6.3> (Accessed 16 May 2018).
5. The Open University (2018) ‘5 Limitations of mobile devices’, *TT284 Block 3 Part 1: The Mobile Device* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1403058&section=5.1> (Accessed 20 May 2018).
6. The Open University (2018) ‘7 How mobile devices are used today’, *TT284 Block 3 Part 1: The Mobile Device* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1403058&section=7> (Accessed 15 May 2018).

3)

Authentication and Authorisation for both the Admin Application and Club Website

Authentication is proving the identify of a person to control access for websites or applications. Assuming the admin application is hosted on a company intranet with admin user accounts, basic authentication could be implemented. This HTTP method encodes user’s passwords, before validating on the server to allow access. (The Open University, 2018)

However, for more protection I recommend digest authentication which uses the SHA-3 hash to encrypt passwords, along with a ‘nonce’ provided by the servers ‘unauthorised’ response. On return, the stored password is encrypted with the same hash and local copy of the ‘nonce’, which is then compared against the user’s credentials. (The Open University, 2018) If the values are equal, authenticity is confirmed and the client is granted access. This is more secure as the password is never transmitted.

Digital certificates could be used authenticate club members. This binds a public key to the user’s email address or network credentials, creating a certificate which can be used to authenticate the user. However, I would recommend using single sign-on, which could authenticate users via a centralised log-in system such as The Open University. This would confirm they are an OU student and removes the strain of password management.

The TLS protocol should be adopted for basic and digest authentication and URLs redirected to https://. This forces the browser to request a secure connection, protecting data in transit as the service’s private key is needed to decrypt data. TLS could also “detect any data change, loss, or an attempt to repeat requests”, exposing attacks. (The Open University, 2018) However for the browser to encrypt the payload, the server has to provide its certificate, preventing data being sent to a unknown source as the server’s authentication has been validated by a third party.

Securing authentication data at rest is equally important; “passwords should be combined with a ‘salt’ of random text generated by the server.” (The Open University, 2018) This helps prevent theft of sensitive data as passwords are not stored as plain text. Separate databases could be used to store sensitive data with stricter permissions and enhanced security, reducing the impact of a breach if the main database was targeted as the attacked wouldn’t be able to gain access to more secure databases. Additionally, back-up databases should be equally protected and system patches kept up to date, protecting the software from weaknesses that attackers could exploit.

Authorisation is giving users certain permissions and access, based on who they are and what they are allowed to do. Authorisation is important to control access, encourage trust of the application and “protect individuals’ rights to privacy.” (The Open University, 2018)

The club website could have two levels of authorisation: general users and club members. All users could view planned walks, news and other events, but only club members could propose walk events and view event details. This would reassure members that events have been approved by the admin team, otherwise any user could create fake events. Importantly, authorisation levels means members can only view and update their own personal credentials, preventing any user viewing other people’s private information, which is a breach of privacy.

The admin application could group users, authorising some groups with greater privileges. A generic group may allow all members of the admin team to approve and create walk events, however only management groups would be authorised to update members’ details. These levels ensures specific actions are only permitted by authorised people, preventing a rogue admin user making every club member inactive or editing their consent to receive emails.

(600 words)

References:

1. The Open University (2018) ‘7.1 Basic authentication’, *TT284 Block 4 Part 6: System security* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419173&section=6.1> (Accessed 5 May 2018).
2. The Open University (2018) ‘7.2 Digest authentication and Client certificates’, *TT284 Block 4 Part 6: System security* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419173&section=6.2> (Accessed 8 May 2018).
3. The Open University (2018) ‘8.1 Secure data transmission’, *TT284 Block 4 Part 6: System security* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419173&section=7.1> (Accessed 7 May 2018).
4. The Open University (2018) ‘6.2 Password storage’, *TT284 Block 4 Part 6: System security* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1405483&section=6.2> (Accessed 7 May 2018).
5. The Open University (2018) ‘7 User authentication and authorisation’, *TT284 Block 4 Part 6: System security* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419173&section=6> (Accessed 6 May 2018).

4)

|  |  |
| --- | --- |
| Test ID | T001 |
| Description | Checking a new Walk Event can be created. As long as all the required fields have valid data provided |
| Setup | Admin application is set-up and running. Admin user successfully logs in to the admin application |
| Instructions | 1. Navigate to the ‘Walk Events’ page 2. Click ‘Create’ button 3. Input valid data:   Name: ‘Saturday walk and lunch’  Date: 11th July 2019  Start Time: ‘9am’  Meet: ‘OU Cafe’  Distance: ‘3’   1. Click ‘Create’ button |
| Expected Results | User is redirected to the Walk Events main table. The table will contain the newly created walk event and an alert will appear at the top of the screen saying ‘Walk Event created.’ |

Table 1: Acceptance test script for successfully creating a walk event

|  |  |
| --- | --- |
| Test ID | T002 |
| Description | Checking the correct validation error messages appear and a new Walk Event is not created. As long as some of the required data is missing or invalid. |
| Setup | Admin application is set-up and running. Admin user successfully logs in to the admin application |
| Instructions | 1. Navigate to the ‘Walk Events’ page 2. Click ‘Create’ button 3. Input invalid data:   Name: ‘Saturday walk and lunch’  Date: 11th July 2019  Start Time: ‘10am’  Meet: “”  Distance: ‘3’   1. Click ‘Create’ button |
| Expected Results | A validation error message should appear, saying ‘Please fill in this field’ above the ‘Meet’ input field. No Walk Event should have been created, and user should not be redirected to the Walk Events main table. |

Table 2: Acceptance test script for failing to create a walk event

|  |  |
| --- | --- |
| Test ID | T003 |
| Description | Checking an existing Walk Event can be updated. As long as the updated fields have valid data provided |
| Setup | Admin application is set-up and running. Admin user successfully logs in to the admin application. A successful Walk Event exists. See T001. |
| Instructions | 1. Navigate to the ‘Walk Events’ page 2. Select the checkbox of the Walk Event to be updated 3. Click ‘Edit’ button 4. Input valid data:   Name: ‘Saturday walk and dinner’  Meet: ‘1pm’   1. Click ‘Update’ button |
| Expected Results | User is redirected to the Walk Events main table. The table will contain the updated walk event, and an alert will appear at the top of the screen saying ‘Walk Event updated.’ |

Table 3: Acceptance test script for successfully updating an existing walk event

|  |  |
| --- | --- |
| Test ID | T004 |
| Description | Checking the correct validation error messages appear and an existing Walk Event is not updated. As long updated data is missing or invalid. |
| Setup | Admin application is set-up and running. Admin user successfully logs in to the admin application. A successful Walk Event exists. See T001. |
| Instructions | 1. Navigate to the ‘Walk Events’ page 2. Select the checkbox of the Walk Event to be updated 3. Click on ‘Edit’ button 4. Input invalid data:   Name: ‘Sunday walk and dinner’  Meet: “”   1. Click ‘Update’ button |
| Expected Results | A validation error message should appear, saying ‘Please fill in this field’ above the ‘Meet’ input field. No Walk Event should have been updated, and user should not be redirected to the Walk Events main table. |

Table 4: Acceptance test script for failing to update an existing walk event

5a)

Version Control advice for the Admin Application

The purpose of version control is to manage a project’s assets though storing, updating and tracking files within a complete working system, ensuring consistency and integrity throughout a project’s development.

With multi-faceted projects, version control allows developers to have independent versions of the same project at any one time, without effecting the original. Version control’s “single storage system” (The Open University, 2018) allows users to always develop the latest files, reducing team-wide errors as it keeps users from “working on older versions that may have bugs in them that other team members may have addressed and eliminated”. (George, 2016)

Version control also allows developers to revert back to a working version at any time if bugs are introduced, tracking what changes were made when, by who, helping error diagnostics and transparency.

To use version control, developers ‘check out’ the projects repository, giving them a personal copy. When a single change is made over a short period of time, it is committed with a description of the changes. This is then pushed to the remote shared copy.

To allow developers committing to the same shared repository, an independent copy of the assets, called a branch, is created. Many branches create strands off the original trunk, which then get merged back in. This “copy, modify, merge” approach allows multiple developers to work in isolation simultaneously. (The Open University, 2018)

When the projects assets are in a significant state, a tag can be created to lock a snapshot of the assets as configuration. This can create a release with an unique identifier, allowing developers to refer back to this particular version for user testing or deployment. Importantly, developers should not include sensitive files or files that are generated from “compilation process[es]”. (The Open University, 2018)

(299 words)

References:

1. The Open University (2018) ‘3.2 Storing assets’, *TT284 Block 4 Part 3: Managing assets* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419382&section=2> (Accessed 13 May 2018).
2. George, J. (2016) *The Value of Version Control* [Online]. Available at <https://www.webdesignerdepot.com/2016/04/the-value-of-version-control/> (Accessed 12 May 2018).
3. The Open University (2018) ‘4.3 Merging’, *TT284 Block 4 Part 3: Managing assets* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419382&section=3.3> (Accessed 13 May 2018).
4. The Open University (2018) ‘5.4 Repository assets’, *TT284 Block 4 Part 3: Managing assets* [Online]. Available at <https://learn2.open.ac.uk/mod/oucontent/view.php?id=1419382&section=4.4> (Accessed 13 May 2018).

5b)

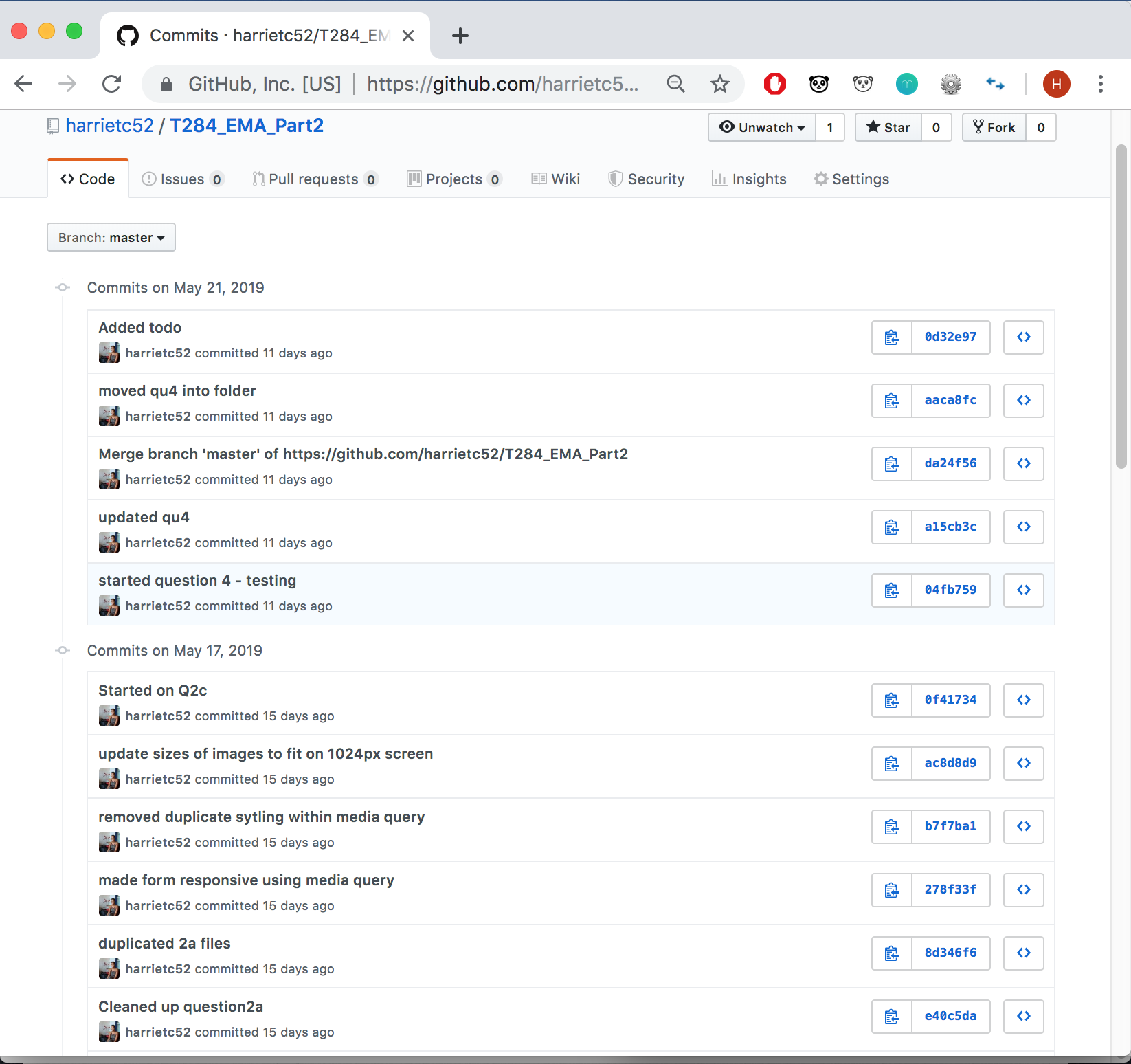


Figure 3. GitHub Log Messages window showing commits to my EMA repository about half way through my EMA development.

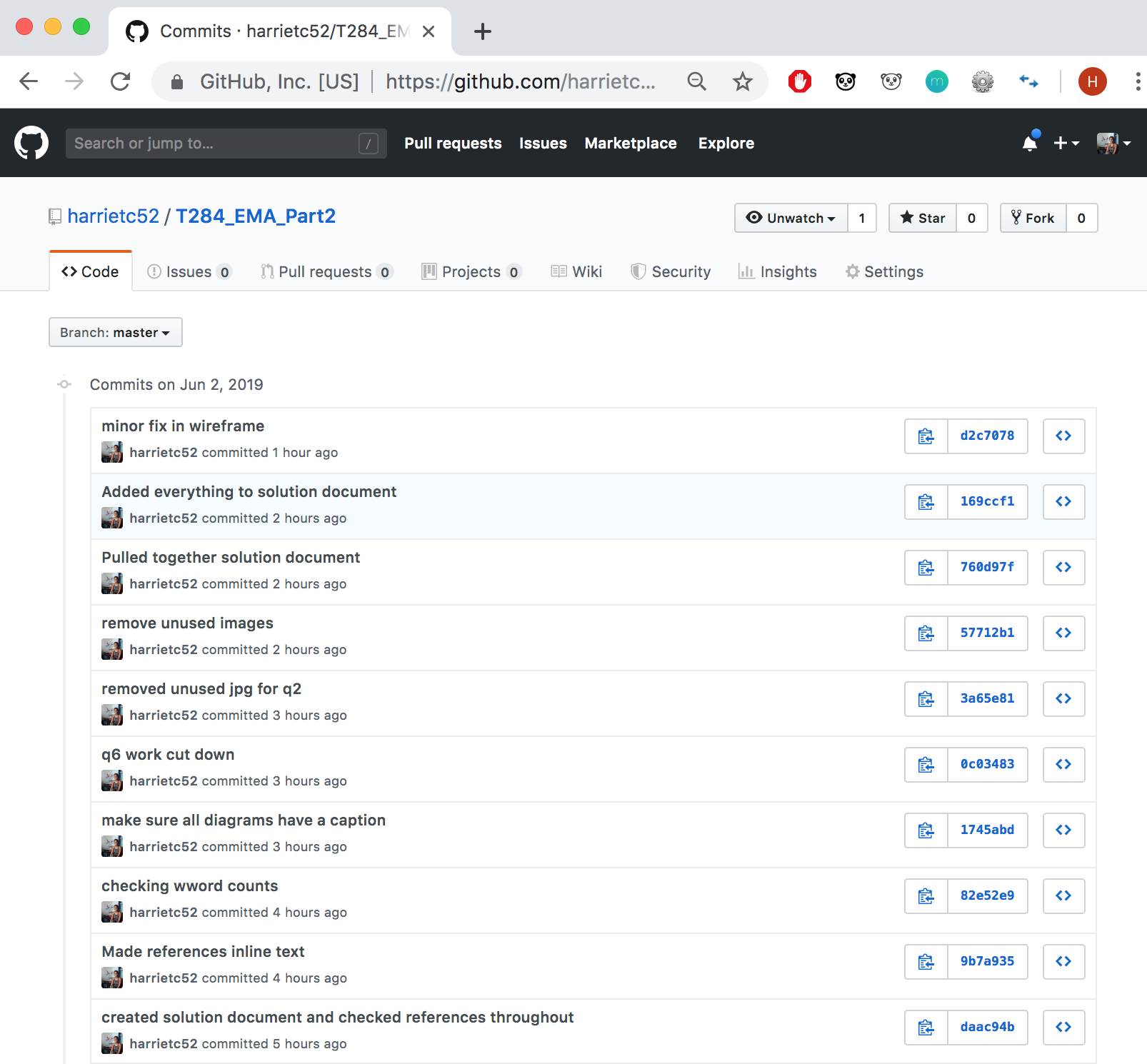


Figure 4. GitHub Log Messages window showing commits my EMA repository towards the end of my EMA development.

6)

Reflection on the effectiveness of my EMA planning

The level of detail in my Activity Schedule was most useful as it included links to resources, question dependencies and sub-divided tasks into smaller activities; all which saved time later. Grouping and writing in detail user requirements was especially helpful for the wireframes. Using version control for all questions let me work from both my personal and work laptop, making development more flexible. However, putting a timeline on my work plan in terms of weeks wasn’t useful as I ended up attempting the questions in a different order. I also made a dairy but I didn’t find it beneficial; if there were longer periods of time between development it might remind me what I last did, but the project was short enough not to need it.

I think I identified an appropriate set of tasks as I didn’t come across many tasks during development that I couldn’t tick off my Activity Schedule. Tutorials reassured me that my break down of questions were accurate. In general most tasks took a little longer than expected, specifically the Wireframes and therefore Question 2, as I had to update the HTML and CSS every time. Version control and testing however took less time than I expected. I didn’t look back at the Risk Assessment after having done it for Part 1, but at the start it helped me feel in control of the project because I knew if any problems arose, I had already considered solutions.

I kept my Activity Schedule up to date with my progress, checking off each question when completed, adding relevant comments from forums or tutorials and amending timelines when I fell behind. With three weeks to go I made another plan (Figure 1), as it got busy with M250 revision and the EMA deadline. I planned out every day which was extremely helpful and reassuring, as I knew if I stuck to it, everything would be finished on time.

The planning really helped me complete the EMA. I had been honesty regarding busy weekends and holidays, which meant I didn’t fall behind. It also helped me not feel guilty when I did have a day off, as I knew it had been factored in.

If another EMA had a similar structure in the sense a question, such as Question 2, is dependent on the Wireframes, I would have definitely spent more time on the wireframes, saving time in the long run. My lesson is learnt to seek out coupled questions early. I would again create an Activity Schedule to the same level of detail but I wouldn’t put a timeline against it, instead a percentage of completion. For example, “By June I should be 80% complete”, using the marking percentages for each question as a guideline, therefore it wouldn’t matter the order in which I completed the questions.

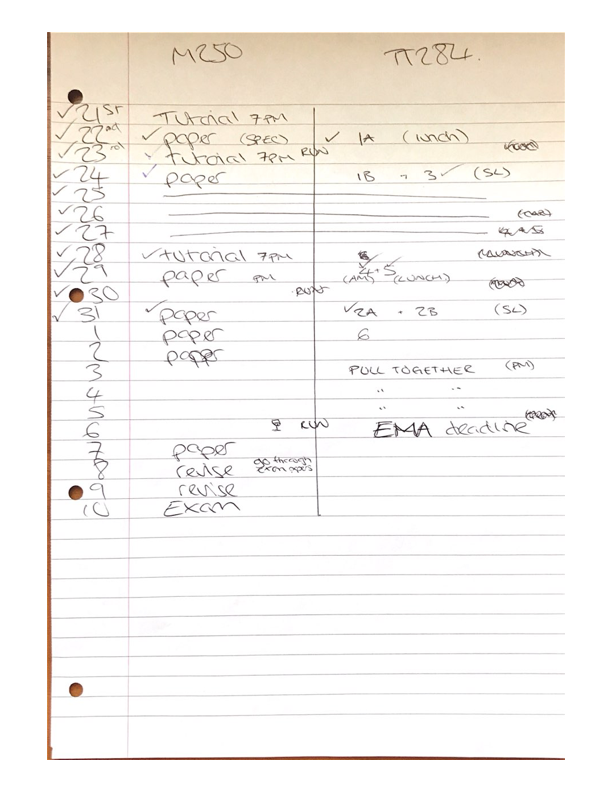


Figure 5. My plan for the remaining three weeks, showing EMA questions amongst M250 revision.

(492 words)