For the initial questionnaire, I hadn’t considered how I would analyse the results or the number of responses I would recieve. Although the 42 responses came steadily, fewer questions would have sufficed. I took this learning into the second “Timeline vs Task vs Calendar” questionnaire, which instead included one question with three multiple choice answers, which was much easier to process. (Appendix 5) What I have learnt from this is that an important factor in designing questionnaires is to think how the information will be processed.

I am using Zotero as a reference manager and I have got better at saving the a reference to Zotero straight away from the browser plug in, and then writing up in the journal my days learning. However, I would have benefitted from adding reference specific notes at the time of saving, becuase I has been hard to remember exactly what conent was used in a reference for a particular task when it has come to writing up TMAs. Learning the PROMPT (ref) method has helped me to criticially evaluate references, something that I need to continue to work on before the EMA. Along side papers and official documentation, the project has extremely benefited from Stack Overflow posts. When I have been stuck on a particular problem, the majority of time Stack Overflow has a post discussing something similar. (Appendix xx)

Before starting implementation, I should have added a story to complete the research tasks such as those resources identified as high risk. I left researching tools, or APIs, such as the weather API or gathering vegetable data until starting the story, at which point it was almost too when, as my schedule has only considered implementation timings. I should have spend more time mitigating the risks of not finding certial resources before development work started. For this final stage, I am going to ensure I have the resources that have been identiied required before starting development.

However, I did instead spend alot of time creating and assessing user stories, adding mock ups, testing criterial, acceptance criteria, texual description etc. This meant when it did come to executing a story, I had a clear idea of what needed to be done, which definitely influenced rate of development; 10 stories in around a month.

Postman was a tool which also helped during development of the Rails API. (ref) It allowed me to test the API before any front-end implementation, easily adapting parameters, URLs, and checking what response and status are returned for different methods. As the Rail API was implementing a JSON API ([ref](https://jsonapi.org/format/)) the body and responses were relatively straight forward. Postman also provides automatic documentation of the requests, and therefore API, which will be used an a way of evidencing work done. The API is also a successful interface, and I like how the front-end and back-end are completely seperated. This gives me more piece of mind when I am working on a piece of code, that changes are in isolation. Additioanlly, becuase I have experience in Rails, I found it quite quick and straight forward to get the API up and working. Allowing me to focus more time on the front-end.

Throughout, I would have liked to do more testing. When estimating story times and my project schedule, I should have factored in an extra hour for each story to implement tests. I did some basic test-driven development when I was struggling on something particular, however I will now go back and add some more.

As for project management, I have learnt how to use Gantt charts and appreciate the value of them as I have referred back to and updated mine lots throughout the project. It was particularly useful when estimating how much development work could be completed when decidign scope. Although I have used my Project Journal (Appendix 36) increasingly more, stating what I have done each session, I have been using GitHub story cards more.

GitHub has been an invalueable tool for this project. Each sub-task has a individual card, where I have been able to add notes, images, tables etc, all in markdown so they can be copy and pasted directly into my word processor, Zettlr. This has and pulling in information for the TMAs more straight forward. The project board with column: Todo, doing and done makes it easy to see the current and upcoming state of work. Card can be moved, and therefore priotisied easily, as well as grouped into "Milestones". In my WaterScrumFall process, these have acted like stages. (appendix 30) Finally, cards can have labels, which has been useful when assigning sizes to cards, as these can be filtered easily. However, the card almost make it too easy, and it can get a little unorganised. For the final few months, I will try an maintain the journal better and organise the information I add to cards better.

I appreciated having friends and family who I could send the questionnaires, mock-ups and MVP too and get quick feedback. I enjoyed that I wasn't required to prioritise stories, as I feel like I would prioritised different features to them, making the application less useful. For example, the UX improvements are not something I usually am concern about improvement, however they are the majority of the users feedback which means I have learn't more about UX that I expected when starting the project.

Finally, more than anything, I have learnt to manage my time during this project. Although my work life balance, isn't very balanced at the moment, I feel like I have been as organised as I can be. However, this next few months, I will also schedule in "nights off" and prioritise "down-time" as well, to try enjoy the final few months of my Open Univeristy degree. \_