CITS5552 Reflective Essay

Willem Meyer 21496322

20/09/2021

The Energy Management System (EMS) was designed and developed by a team consisting of Yiwen Li, Jason Chu, Lifsania Robin and I on behalf of our Client, Dr Thomas Braunl. EMS aims to promote UWA's green energy initiatives by gathering, storing and visualising energy production and consumption data. I served as team lead and front-end developer on the project. It was a pleasure to work with this team on an interesting project. While I learned a lot, especially about web development, the experience was coloured by my medical absence during the final stages.

There was no pre-existing structure which we could draw upon to determine seniority and skillsets. Typically, the project manager at your company manages the project, the web developer handles the website and so on. We spent some of our first two meetings getting to know each other, our experience and our skillsets. We discovered quickly that one of our weaknesses was our lack of web development experience, especially with more recent technologies. We also found we had a significant asset in Yiwen. He is a mature age student with real world experience in cloud computing, database management and project management. As a result, Yiwen was our first choice for project leader. Unfortunately, Yiwen said he wouldn't be able to fulfill the role, citing a lack of time due to work and family commitments. Hence, Yiwen suggested me for the position.

Early during the design of EMS it was found that system divided logically into four layers.

> **Data Collection layer:** Periodically access, clean, transform and store energy data.
>
> **Data Storage Layer:** Database of all data collected including timestamps, units and origin,
>
> **Data Access Layer:** Aggregate data for easy access and presentation
>
> **Presentation Layer:** Present the data in a web interface.

Early in the feasibility analysis process each of us was assigned one of these layers. These assignments naturally carried over to design and development. These assignments worked well as each layer harnessed different technologies, allowing its' overseer to specialize. It also encouraged a sense of ownership of the project for each of us in the team as there was a sub-section where we were responsible and in command. This also had the negative effect of making team members reluctant to work on systems that weren't "theirs". This effect was largely overcome by the end by reinforcement of a more holistic mindset during team meetings.

Naturally, most of our communication with external stakeholders occurred by email. It was established early that as team lead, I would send and receive these emails in order to establish an easy port-of-call for stakeholders needing to reach us. It also ensured that we were communicating effectively because it was clear who's responsibility it was. This project had a significant number of stakeholders on account of our data sources being managed by various separate groups. I encountered some issues stemming from this. For example, one data source, the solar farm at UWA future farm in Pingelly, required additional configuration of its plant-monitoring device to allow us access. My inexperience hampered my attempts to organise this, as I wasn't sure which stakeholder would be able to assist. In my quest I had contacted the farm manager, plant installer, device manufacturer, the Client and the professor heading the Future Farm. Asking technical questions of people is sometimes difficult, as is "cold calling" people and asking them for their help! This, like most problems, was

resolved by knowledge and experience. Every time I learned something about the way the related systems were set up, it made it easier to ask the right questions from the right people and to reassure them that I knew what I was doing and deserved their help.

Microsoft Teams was used to fulfill most of our administrative and communication needs. The in-app chat provided a more informal communication channel than email, with urgent matters being denoted by tagging affected team members and requiring acknowledgement within 24 hours as a team rule. Team meetings and Client meetings were also conducted on Teams when on-campus meetings weren't an option. A small wiki was maintained on Teams and was largely used to collate instructions on accessing the various APIs and servers we had been granted access to. Teams was also a convenient place to store and edit various project documents including timesheets and meeting minutes. Despite our heavy use of Teams, we didn't encounter any issues and it streamlined much of the project's administrative work. Git integration wasn't something we needed which was fortuitous as Teams' doesn't seem to include it natively like an alternative like Slack. I'm eager to get some more experience with Slack to see which I think is a better option for software projects going forward.

As noted, the lack of web development experience was a major weakness of our team which was identified even before meeting with the Client. However, during that meeting it was made clear that a web interface would be essential. Moreover, the requirements called for an incredibly robust interface, capable of storing hundreds of thousands of datapoints and mapping that data to different visualizations effortlessly and in real-time. During our feasibility study and design research phase, I was assigned the front-end. My lack of experience made choosing an implementation challenging. I was aware that web development was no longer just HTML, CSS, a dash of JavaScript and a pinch of PHP (as it had been when I dabbled in it during high school). The proliferation of libraries and frameworks over the last decade demonstrates the ways in which web development has expanded and diversified. I chose one of these libraries, D3.js, to drive our visualization. It wasn't the newest or the most popular option for DOM manipulation and I wasn't always sure that I had made the right decision.

In retrospect, D3 worked extremely well. The line chart visulation we required was easy to set up because it is a common visualization with plenty of online examples and in-library support. Rendering the line chart on the client-side (as opposed to rendering the line chart as an image on the server) was a risk, as I didn't know if the performance hit would be acceptable, given how much data required processing. D3's data join proved very efficient and the page load remained fast. D3 also worked well in conjunction with SVG, the medium I chose to render the graphics in. Given the opportunity to go back, I might have opted to use a hybrid React/D3 approach, as the two seem to complement each other's weaknesses. In particular, React uses an update pattern which is easier to read and learn.

The setting up of a comprehensive web development environment was another among the many new things I learned as part of this project. I found it surprisingly difficult to find an out-of-the-box solution that suited our needs. I suspect this was because initially I had no idea where to begin my search, what best practice was or even which words to use in my search queries. Coming across Node Packet Manager early on was a major breakthrough. From there, I was able to quickly set up all the things my research told me I might need; the server, a linter, a bundler. I also set up the generic stuff; version control with Git, Documentation with JSdocs and VS Code as my editor. The development environment worked well and I was able to protype rapidly as a result. I was never sure if the way I was working bore any semblance to the way "real" web developers work. This was a deficiency in my professional skillset which I am glad this project uncovered. Regardless of what kind of software I end up working on, I think web development knowledge, skills and experience will be beneficial and I hope to improve in this area in the future.

The greatest challenge I faced during this project did not come from the project itself, it was personal. I was admitted into hospital in week 7 of Semester 1. I was initially optimistic that I would be able to meet my commitments to the project. Unfortunately, during the month and a half I was in the hospital I was incredibly unproductive. This was in part due to the intensive regime wherein I was under general anaesthetic three times a week, undergoing procedures which heavy impaired my memory. Work on the front-end went slowly because I couldn't recall syntax or what I had been working on even a day earlier. After my release, and the initial deadline for the project, I focused all my attention on finalising the project. This effort was interrupted when a routine out-patient procedure resulted in my spending the night in the emergency department. I was bed-ridden for another month following that incident. After my recovery, I once again returned my attention to the project and briefly experienced the issues one experiences when refamiliarizing themselves with a codebase. These were exacerbated by the quality of the work I had done in the hospital. I had documented my work poorly and implemented some questionable design decisions. In hindsight, I should have redesigned these sections as some of these lead to avoidable and preventable issues. I ultimately moved down to Busselton so that my Parents could help look after me while I worked to meet my academic commitments. Perhaps the most frustrating aspect of this series of events has been that I don't know what lessons to draw from them. I don't think they have made me a better person and they halted the momentum I had built up to finish this project and my degree. Most of all I felt incredible guilt for the effect my absence has had on my team and professors; and gratitude towards them for the patience and understanding they have shown me.

I feel I also made some mistakes as a result of my medical absence. Firstly, when I started work again, I was very reluctant to contact my team, as most of them had graduated and moved onto other things. In retrospect, this was foolish. I struggled to make small changes in systems I was unfamiliar with and when I finally contacted the team, they were very willing to help. A related problem was with my access to the deployment server. Initially, our Client negotiated access to the server (within UWA) with IT. My access elapsed after the end of the semester and I was hesitant to ask the Client to reinstate it. Instead, I chose to try to contact IT myself which proved ineffective. As always when problems arise during project management, the solution requires communication and had I reached out earlier, not only would everything have worked out faster, but the other parties would have been more understanding.

I am proud of the work my team has done on EMS. I am also very excited to continue to work with the Client to continue to add features and improve the quality of this project. As seems to often be the case, a lot of the time I allotted for testing was cannibalised to work on other areas. In particular, because of the constraints my absence placed on the project, system tests and acceptance tests weren't as thorough I would have liked. I intend to revisit these. My health had a major impact on my work during the latter stages of the project and I am pleased to finally be able to return to fulfill my commitments. However, I believe if I had accepted my position earlier and delegated my work it may have been possible to deliver the project, in its entirety, on time. I hope to continue improving in the areas this project highlighted: web development, team management and stakeholder communication. I hope to visit the university and see the interface I designed for this project being displayed on a monitor, engaging the wider community with the amazing things happening at UWA.