

AB-Bilby Team Report

Introduction

Our client PWC required us to build an application that would help detail the current ownership and Native Title status of land in Australia. Land is currently categorised through court hearings and the data regarding each plot of land can only be found within lengthy legal texts. Our job was to create a publicly accessible web application that contains a large interactive map of Australia. This map could be used to search for land plots and extract ownership information. We were also required to include a variety of other features, for example: allowing users to log in and store bookmarks and maintain a history of plots that they have looked up. Additionally we were required to build a system that government users would be able to use to insert land and ownership information that would be included on the map.

Challenges

Unclear requirements

Our team had a somewhat difficult time interpreting the requirements. Despite having two versions of what were supposed to be the same requirements, inconsistent terminologies were used throughout each document to add to our confusion. The background pages of each were also unclear in describing the goal of the project. Our team handled this with their best attempt to interpret the documents, as well as updating certain documents to provide consistent and meaningful information for future reference.

Limited client availability:

From the beginning of our project, we have had very limited communication with the client, and were told that we would be unable to meet the original client. We were then provided with a replacement client, but even then we could only meet during the middle of sprint 2. This was a great challenge for our team as we had to develop most of our product without any client input. Combined with unclear requirements documentation, we were left to make educational guesses and assumptions for what our client expects from the system.

With limited contact, our best course of action was to design our application as similar as possible to the digital prototypes. We also made our code easily modifiable, and did not make any major changes until we met the client, as that may render previous work fruitless.

Lack of supplied data

A data source of plot and Aboriginal country information is essential to our application; it would not be an exaggeration to say our application is completely useless without one. With the late meeting with the client, as well as a lack of an established data source after meeting, we were left under the impression that a bulk mechanism for adding data was not required. This resulted in us implementing a mechanism that only allowed one piece of data to be added at a time, and we were only notified that it may not be sufficient recently near the end of sprint 3. Whilst this may be true, we have designed our adding mechanism to be easily modified to accept bulk-data.

A lack of data also means we cannot evaluate our system performance efficiently and accurately. We cannot be certain that our current system will be scalable when a very large amount of data is added, and could only give our best at making the application run efficiently.

Limited experience with technologies

We built our web application using the MERN stack. While some team members had prior experience, the majority did not. This was an issue that needed to be addressed before the beginning of the first sprint. Thankfully there is extensive documentation and learning materials online for all components of the MERN stack, which is one of the main reasons why we chose it. We split the team into frontend and backend teams and each subteam had a member who had previous experience with the technologies.

For the frontend team, two out of three members were completely inexperienced in using the React and Redux JavaScript library. Fortunately the team was able to catch up rapidly to meet the needs of the development process, albeit at the cost of code quality and efficiency. Similarly, the one member of the backend team had no experience in these technologies. The other member of the backend team had a wealth of knowledge and one on one meetings as well as online resources were invaluable in the learning process.

Challenges building the Map Interface

A critical piece of information we lacked from the client was the format of data in which the plot data were to be stored; as such, we resorted to a standard geoJSON format used commonly in storing and rendering coordinates.

Another issue present was the presence of a bug in the search engine built within the map library we used. It would repeat searches when not instructed to, which severely interfered with our search history functionality in keeping a consistent order of history. As a result, we had to abandon maintaining order of history as it would be too time consuming to fix in our sprint.

Lessons learned

Product

If we were to repeat this project, there are a handful of things that we would change to improve the product. One thing we would change would be the process of uploading plot data to the database. The current process involves filling in a form with all of the required fields for the plot. The final entry in the form requires a JSON object in a specific format. To obtain this JSON, the user has to go to the geoJSON website, create the plot and then copy a specific part of the JSON output, return to our webpage and then paste the JSON object. Given another chance to implement this we would make this feature such that the user would not leave the webpage. Instead, the user could interact with a map on the page to make the plot and the data would automatically be generated and input. This would heavily reduce the chance of human error as now the JSON object is created entirely by the application.

The improvement on the input of plot data above would greatly improve the user experience for inputting a single plot, however there are likely thousands of plots across Australia that would need to be put into the system. Going through this process for every plot one by one would be a tiresome task for the user. If we were to repeat this project, we would implement a system for adding the plots in bulk. Ideally the user would be able to upload a file in a specified format that contained data for multiple plots. Then, our web application would parse the file for the relevant plot data and iterate through until all of the plots were uploaded to the system.

Process

Assignment of responsibilities within JIRA

Another of the key things that we would have changed if we were to repeat this project would be our process of assigning responsibilities of tasks. Initially, we established what tasks everyone would be completing for the entire sprint within the sprint planning session, and assigning them immediately. What we quickly found was that, by doing so, we were unnecessarily creating some bottlenecks and blocking issues, as the understanding of how much time team members had available and when they could work on the project was not fully understood at the time of assignment.

It was also later on recommended to us by our supervisor that we should be assigning tasks when moving tasks from “to be done” to “in progress”. When we took on this suggestion, we found that it both simplified the sprint planning sessions, as well as streamlined development. Because individual team members have a better understanding of how much time they can dedicate to a task and whether they can complete it soon, they were able to estimate whether

they should take on critical blocking tasks. As such, if we were to repeat this project, we would be following this process from the beginning.

Design Documents

Another lesson that we learnt was the importance of design documents - in particular, a visual representation of what the UI should look like. What we discovered was that, without the presence of these documents, each team members had a different idea of how the UI should look like in their mind. As a result, when it came to the review process, the team had to establish what design approach they were looking for as a whole, and refactor it accordingly. If a design document was created initially before development, it would reduce development efforts required to achieve a design that everyone was satisfied with.