

# WEEKLY PLANNER APPLICATION

TEAM: TRISTAN KELLS, DAN WOOD, JUNHA YU  
MENTOR: KENNETH JOHNSON

PROJECT  
NO: 3.43

**BULLETPROOF**  
MISSION CRITICAL CLOUD

AUT

## PURPOSE & OBJECTIVES

Bulletproof is a cloud consulting company based in Auckland that specialises in Amazon Web Services (AWS). They needed a digital whiteboard system they could use to remotely track their consultant's weekly schedules, replacing a physical whiteboard. The drive for this project came from a lack of results when attempting to find a solution online. Popular project management tools like Google Calendar, Jira and Monday.com, were too feature dense and lacked the desired ease of use.

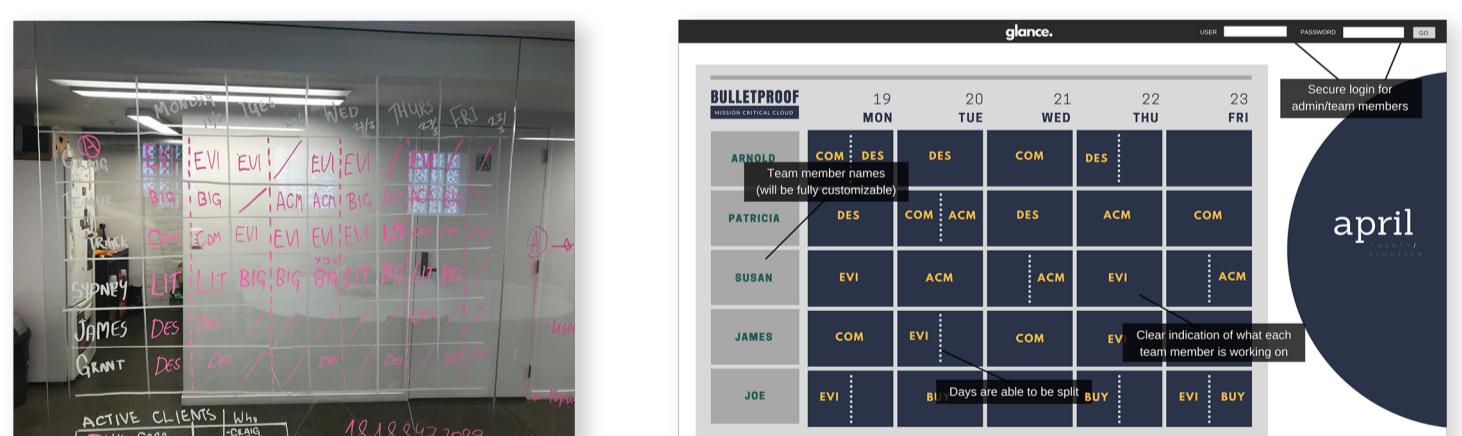
The planned minimum viable product (MVP) needed to be viewable through a web browser, to support a secure login system and to receive remote updates. An early stretch goal included integration with existing project management tools, such as Jira, Harvest, and Asana.

## CONCEPT

The idea was to emulate a physical whiteboard that was in use at the Bulletproof Auckland offices. The opportunity that was presented in this project was to optimise and digitise a core team management tool to Bulletproof and streamline the way they managed the relationship between consultants and respective clients. In addition, we learned that this tool could be beneficial to several other organisations that were using a similar physical workflow model - a sentiment that was reflected in one of our early meetings with the team.

## PROTOTYPING

From a photo of the original calendar and an initial requirements gathering session we developed a working prototype for analysis and feedback.



## ACKNOWLEDGEMENTS

We would like to thank the team at Bulletproof for being so supportive and knowledgeable throughout this project - James Burton, Craig Hurley, Grant Sweeney and Geethika Guruge. Without your input and positive attitude we would not have been able to complete the project.

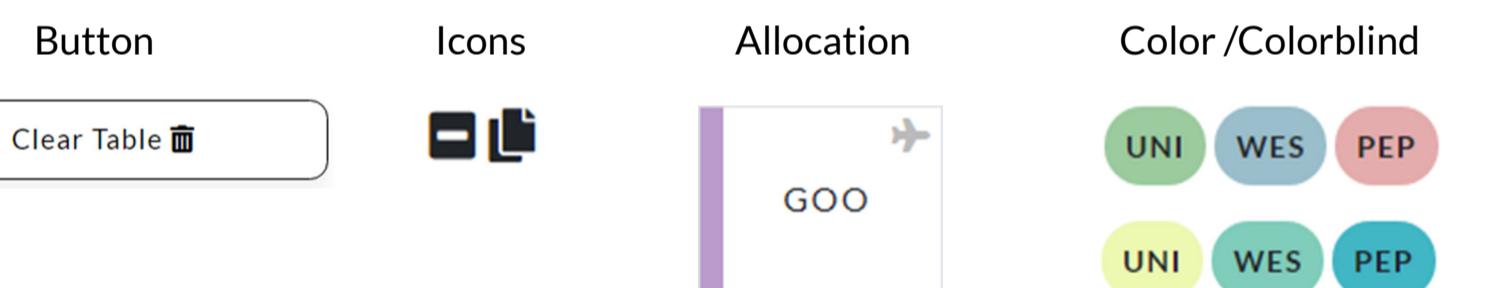
We would also like to thank our mentor Kenneth Johnson for all of the advice and recommendations. His depth of knowledge assisted us in making important decisions and lead us to a product we are proud of.

- Dan, Tristan and Junha

## METHOD

Scrum Process - The project was completed using the Agile Scrum methodology over 11 two week sprints. We maintained a backlog of user stories representing product features. At the beginning of every sprint, our team would prune and prioritise the backlog based on client feedback. The outcome of this would be a product vision for the sprint and a list of user stories assigned to members of the team. We identified the importance of working regularly in shared spaces to foster a productive, agile development environment [1]. To support this realisation we made weekly collaborative coding a key part of our process.

Usability Studies - Replacing a tactile and easy to use physical whiteboard meant that our application needed to be designed with a heavy emphasis on usability. To ensure our product succeeded, it was critical to gather user feedback. We held usability testing sessions at each of our client meetings, using our stakeholders and later external users as testers to ensure that each feature we implemented followed a human-centered design. Taking advantage of the iterative nature of Agile development, we used the feedback given at each meeting to update our product backlog. User Interface/Experience features were carefully chosen/designed to be as intuitive and meaningful as possible.



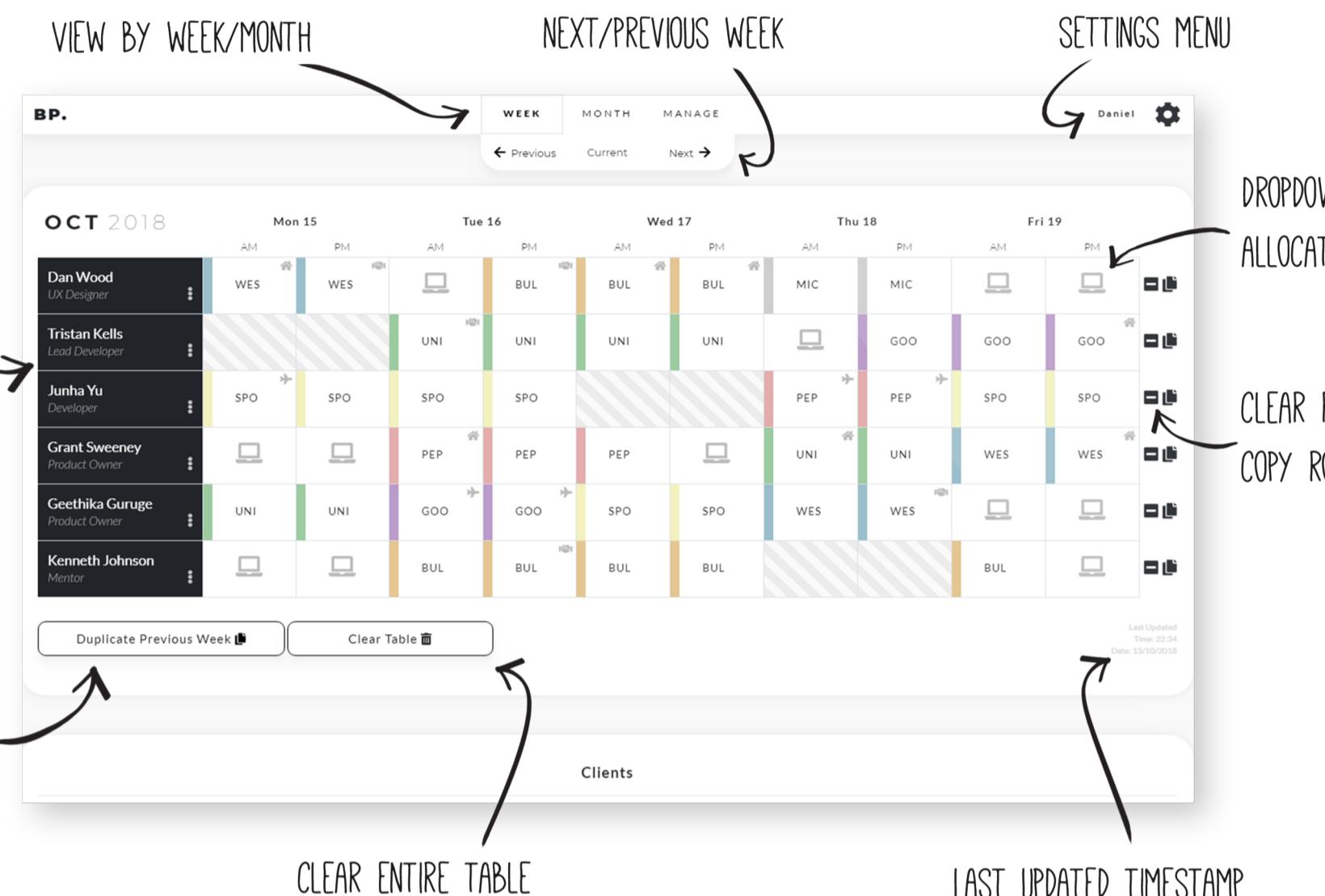
Stack - As a web development project with a strong front-end focus, all members of the team had to upskill. Because of our inexperience with front-end development, we chose not to use a modern framework such as React or Angular, instead deciding to use a more straightforward approach using Javascript, HTML, and CSS. This would allow us to gain a solid understanding of front-end development from the ground up. Using Javascript and the jQuery library, we adopted a revealing module design pattern [2]. PHP and MySQL were used in the backend implementation of our application as all members of the team were familiar with those languages.

AWS - With assistance from Bulletproof, our application is hosted on an Amazon Elastic Compute Cloud (EC2) instance. The EC2 instance is a T2 nano instance with one virtual CPU and 0.5 gigs of RAM. The database is Amazon Relational Database Service (RDS) t2.db.micro instance. Both of these cloud services are hosted in Oregon and paid for by Bulletproof. This Amazon cloud service was a robust server solution and allowed us to fulfil the project requirements around the accessibility and availability of the application.

## REFERENCES

- [1] Ozierańska, A., Kuchta, D., Skomra, A., Rola, P. (2016) The critical factors of Scrum implementation in IT project – the case study. Journal of Economics & Management, 25(3) 79-96
- [2] Osmani, A. (2017) Learning Javascript design patterns. Retrieved from <https://addyosmani.com/resources/essentialjs-designpatterns/book/>
- [3] Ma, W., Chen, L., Zhou, Y., Xu, B. (2016) Do We Have a Chance to Fix Bugs When Refactoring Code Smells? Paper presented at the 2016 International Conference on Software Analysis, Testing and Evolution (SATE). doi:10.1109/SATE.2016.11

## RESULTS



### The Product (Cloud Web Application)

Entity Diagram - This changed throughout the project as new features and relationships were added.

Product Backlog - Due to the iterative nature of our development our Scrum backlog saw many adjustments as features were deemed high or low priority and in some cases taken out completely.

Product Manual - Documentation including setup, operation and FAQs.

## AREAS OF DIFFICULTY

Jira Setback - Up until our mid-project review, we were firm on creating the application as a Jira plugin. This would allow us to join an existing project management ecosystem and use their data to populate our application. However, after additional client consultation, we decided to build a standalone web application because the Jira plugin experience failed to meet their accessibility requirements. Research spent learning about the Jira platform did not support our final product and considerably slowed down early development. Clear and frequent client communication helped us pivot from this misstep and deliver the final solution.

Domain Knowledge - Before our project started, only one member of our team had experience with a web development stack, and none of us had any experience with hosting an application using the cloud. Upskilling was a sizeable initial project investment. We utilised the cloud expertise of our clients and high-quality online learning tools such as Linux Academy and W3Schools to assist us in this process. As a team, we were successfully able to leverage each other's learning to build a collective set of skills to deliver our product.

Code Refactor - Code refactor was not initially accounted for in the project plans. As the code base grew, time was needed to refactor old code to support productive team collaboration [3]. This was especially pronounced as we were still actively learning the best practices of our chosen technologies as we developed, meaning old code would often not meet our current standards. Once we recognised the importance of code refactoring, we accounted for it in our sprint plans and left time at the end of the project to tidy up the code base.

Quality Assurance - Quality Assurance was a missed opportunity for us. We did no automated testing and did not take advantage of web application testing tools such as Jest or Selenium. This was in part due to a lack of understanding and the rush to develop a feature-rich application in the time frame available. Due to this, updates often broke key application features. Our experience with this project has emphasised the importance of robust testing frameworks.