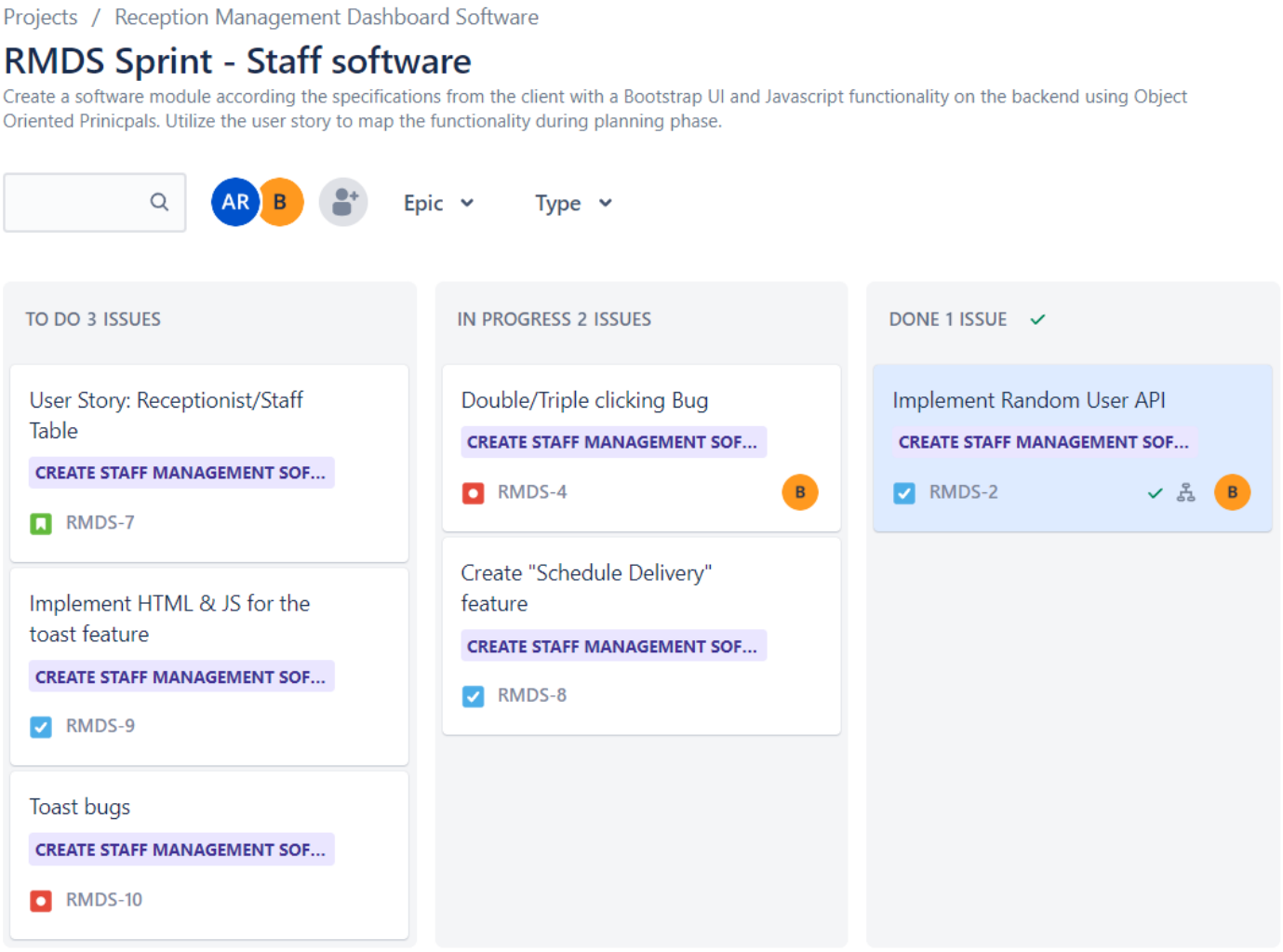
Reflection Report

- Project: Reception Management Dashboard
- Client: WeDeliverTECH

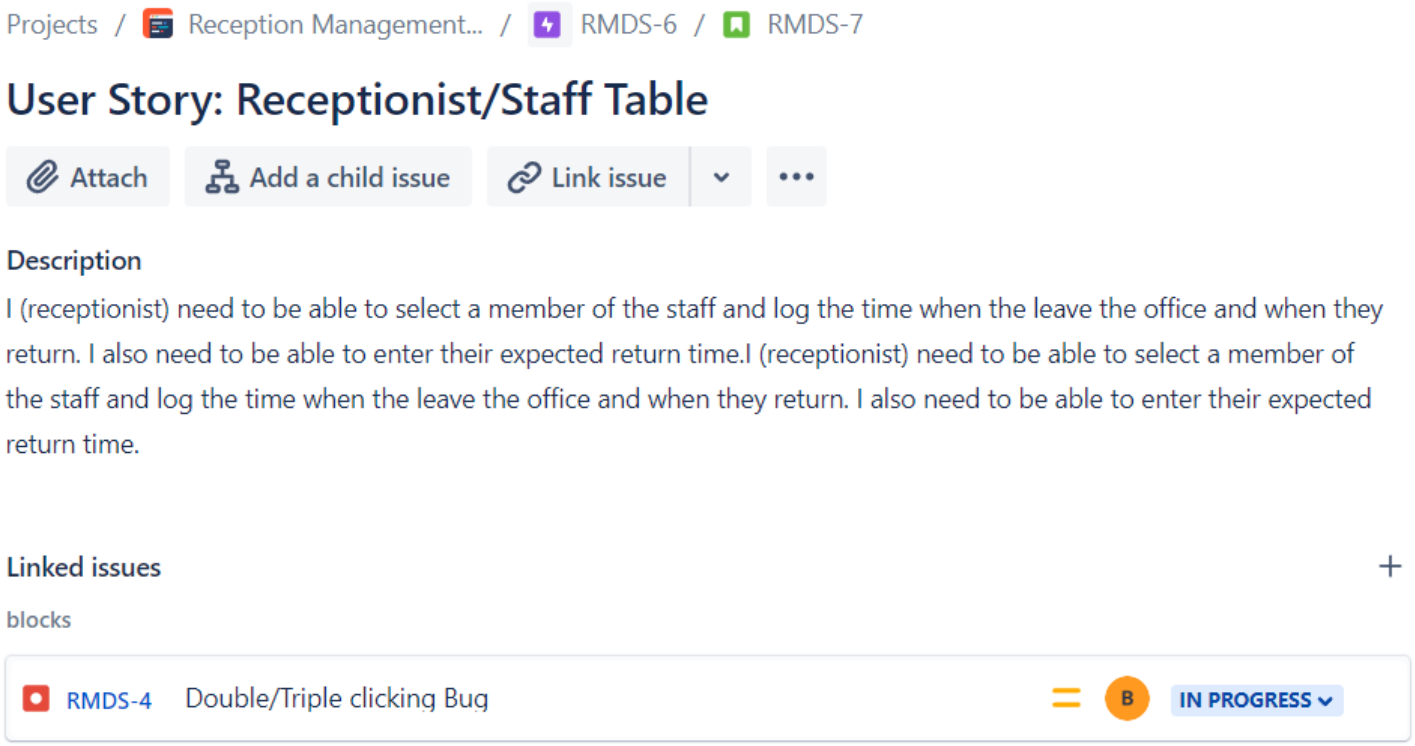
Our first 2 Week Sprint

During our first sprint we chose to focus our attention on the creation of the first module of the dashboard, namely the “staff dashboard” that logs the current staff members time in and out of the office. We created a user story, and proceeded to list the key tasks involved for meeting the requirements in the user story. They included a Bootstrap toast feature, and implementing the random users from an API. We had several challenges fixing crucial bugs and the actual implementation of the API into the HTML was quite challenging as you can read on in the section of the reflection report titled “challenges”. All in all we are pleased with how handled the issues and bugs that arose, and we focused on clear and frequent communication between the project lead, the back-end developer and the front-end developer.



User Story 1

We truly appreciated that WeDeliverTech employees were so forthcoming with their needs and wishes for their software solution. After listening closely, we were able to create a solid user story that helped guide us in the planning and development stages of this first sprint.

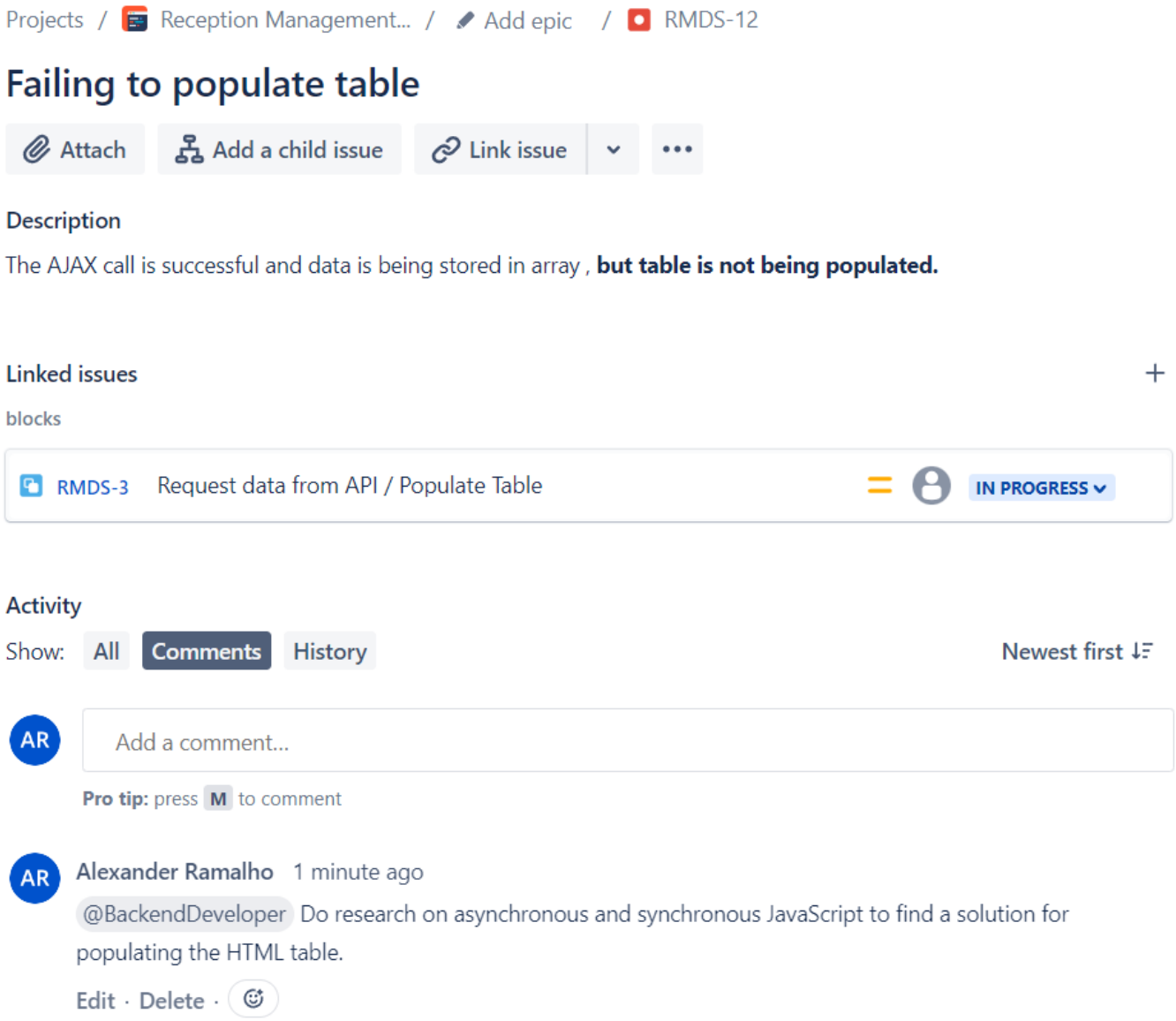


Challenge 1: User Interface Issues

One of our more challenging moments was implementing the UI for the staff and delivery board. We wanted to highlight the selected row and remove the highlight after deselecting it. During this period, we talked back-and-forth between the back-end developers and front-end developers to ensure that everyone was on the "same page". After some trial and error, we found an elegant solution that fulfilled the functionality and user friendliness we sought. This bug taught us about the importance of code readability and utilizing cleaner code.

Challenge 2: API's and Asynchronous Promises

Asynchronous Data Fetching: One of the challenges encountered was fetching random user data asynchronously using AJAX. This required understanding how promises work and handling the success and error cases appropriately. Through this process, we gained a better understanding of asynchronous programming and how to handle data retrieval from external sources.

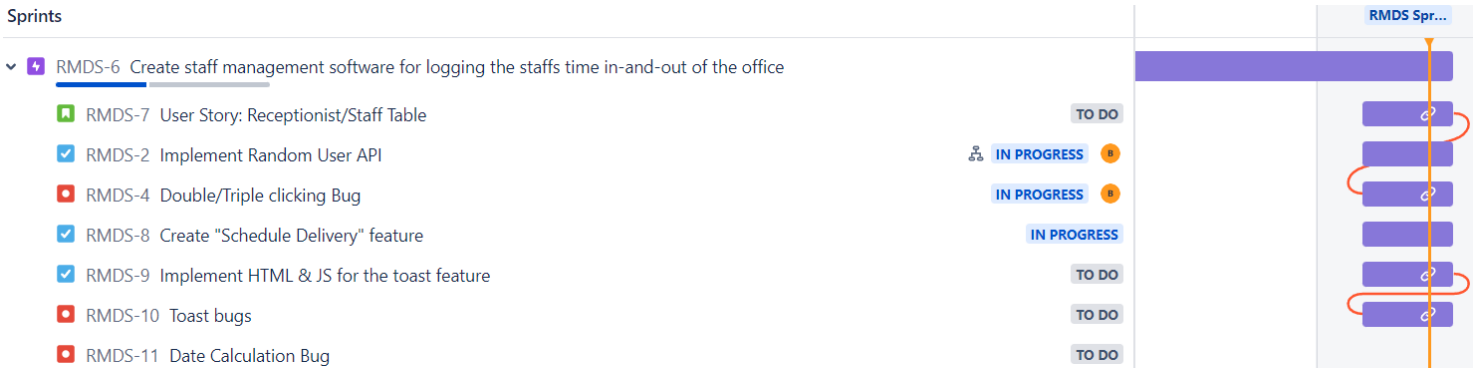


Challenge 2 : Object-Oriented Design

Object-Oriented Programming: Implementing object-oriented programming (OOP) principles was crucial to organizing and managing the staff and delivery driver data. The code demonstrates the use of classes and inheritance to create Employee, StaffMember, and DeliveryDriver objects. This experience reinforced the importance of OOP in building scalable and maintainable applications. We learned about the incredible flexibility that thoughtful OOP can lend to ever changing code base with new features being implemented on request of the client.

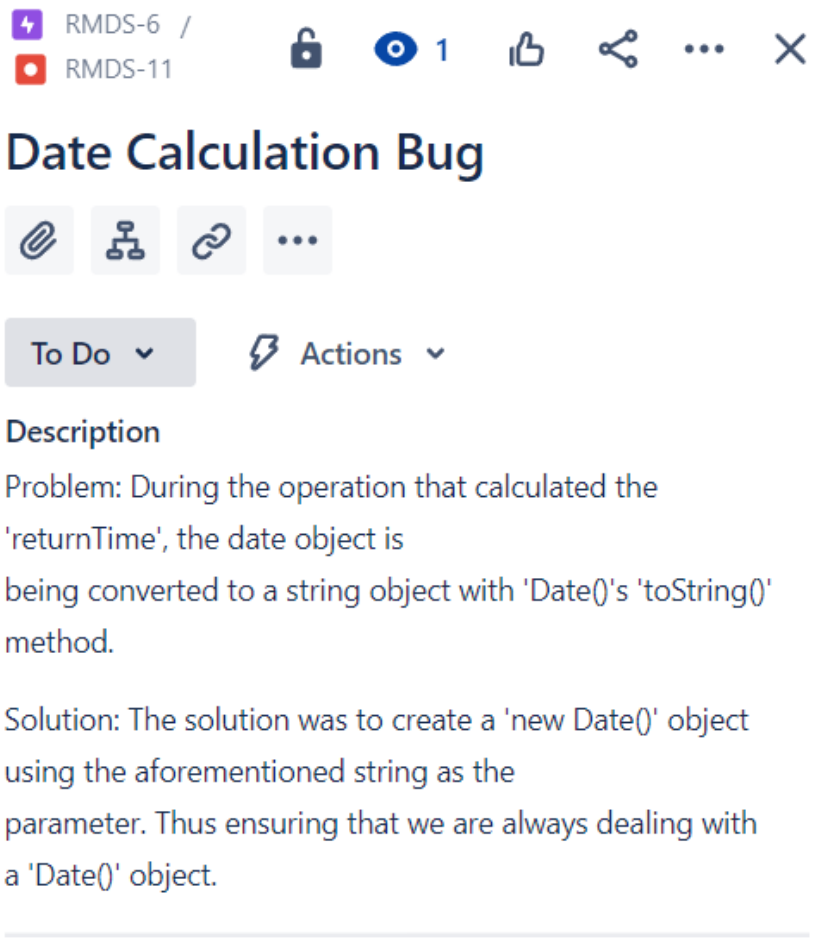
Roadmap to Success

After spending many hours solving bugs and creating better functionality, we were able to solve all our issues on time and move on to the next sprint with a good feeling about our efforts.



Last Epic Adventure & Backlogs

After solving some really tough problems during our first epic, we were able to efficiently utilize what we had learned and apply it faster and better to the issues in our second and last epic. The team really enjoyed the process during this epic. We paced ourselves well, and were not afraid to keep certain non-critical (at the time) issues sitting in the “backlog section” of JIRA. Once we felt confident in our main progress, we moved the issues into our epic. Here’s an example from the backlog.



Communication is Key

Thanks to our project lead who encouraged us daily to communicate between the developers, especially front-end and back-end, to solve problems together when needed.