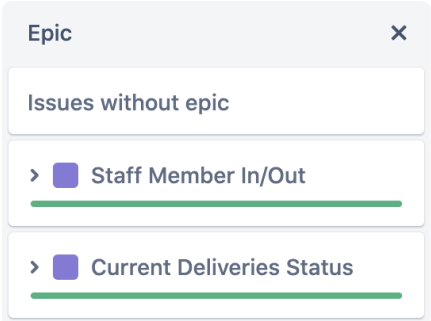
Brittany Bolling-Nilsen
Reflection Report

Semester One
Backend Development

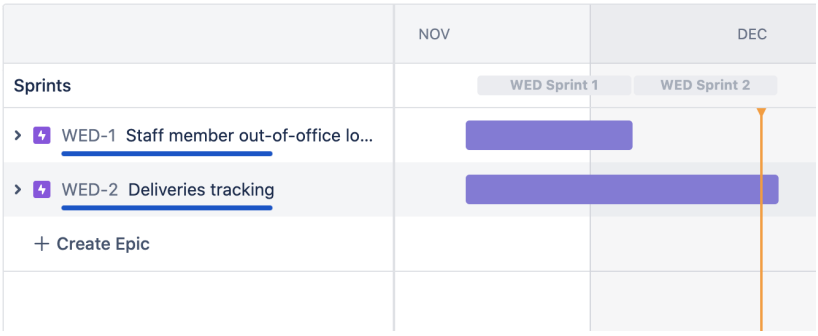
This project was planned and developed using Jira software and Scrum with a time-line of four weeks. I created two sprints which lasted two weeks each, and completed one Epic per sprint.

When I planned this project I started by identifying the main goals of the project, which I then decided were the Epics. I identified the Epics as the ability to keep track of their staff members (in/out of office) and the functionality to keep track of current deliveries.

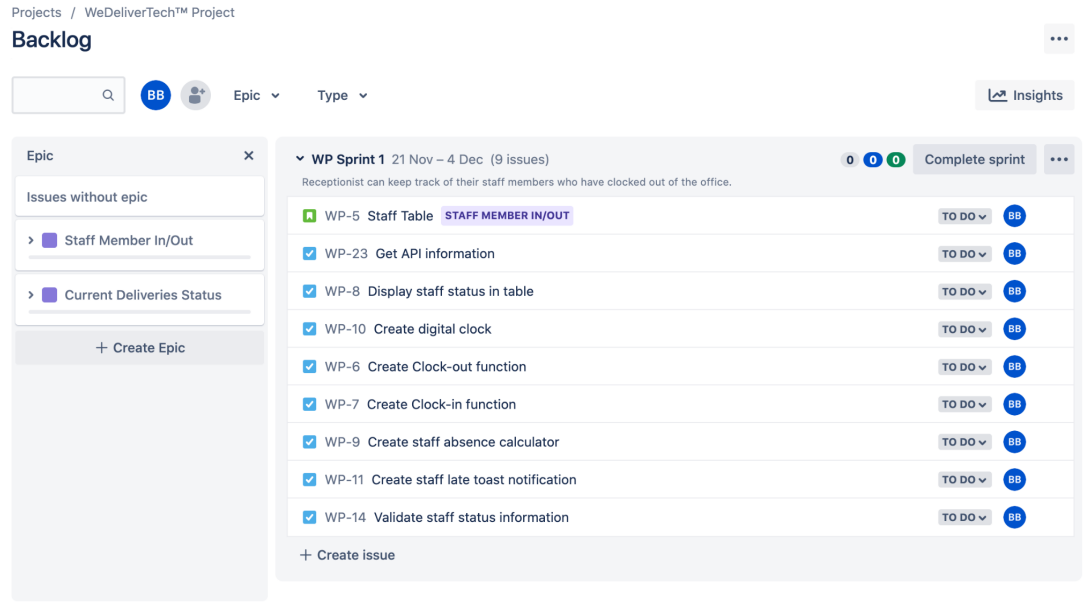


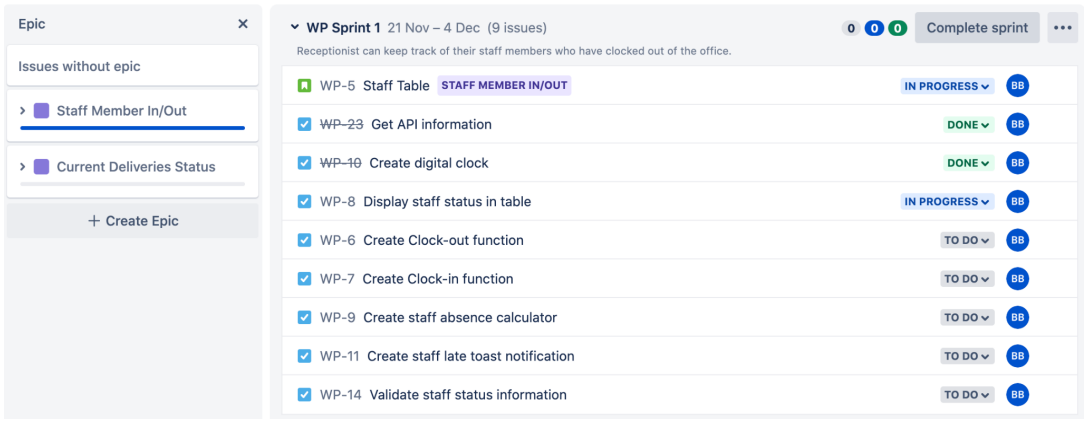
When creating the stories for the project, I read through the customer's requests and identified what needed to be present for the users to accomplish their goals. The stories matched the three tables we were required to create: Staff Table, Schedule Deliveries Table, and a Current Deliveries Table. In order to satisfy the customer's needs I broke down each story into small issues, which told me what work needs to be done to ensure I fulfilled each functionality request. I attempted to create issues that solved a problem which was essential for the program to function correctly. The issues were small enough that I had one goal to achieve the issue, but big enough that I wasn't updating my project on Jira every time I wrote a line of code. I focused on the value that the project would deliver to the end user, and broke that value down into smaller, manageable chunks that I could tackle one at a time.

Roadmap at beginning:

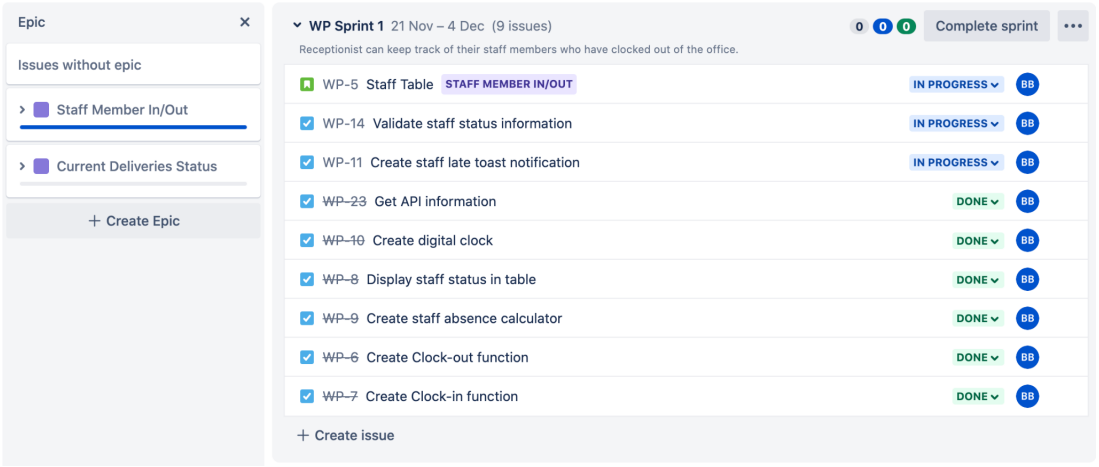


I have attached screenshots of the backlog during sprint 1 below:

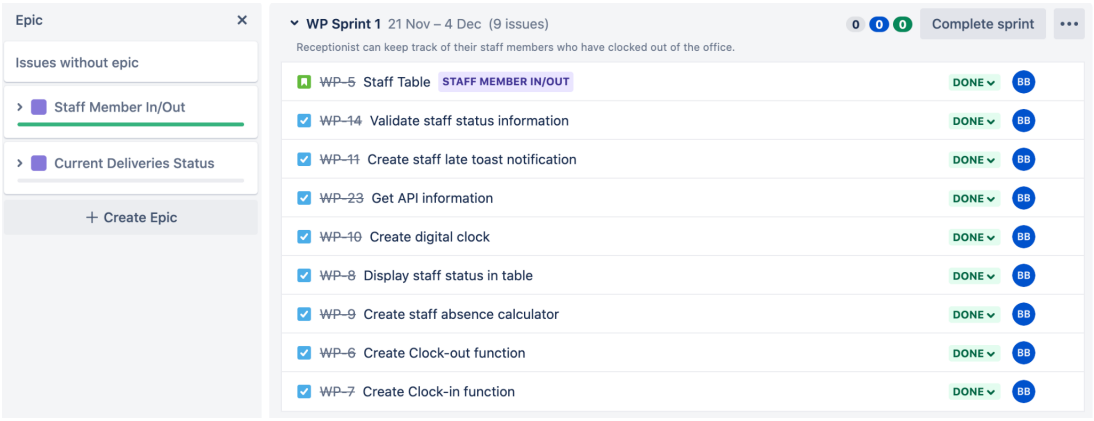




I first made an API request so I could populate the staff table. The digital clock was an easy, independent task. The digital clock was necessary to always show the receptionist the current time, and to update the staff member's clock-out time.

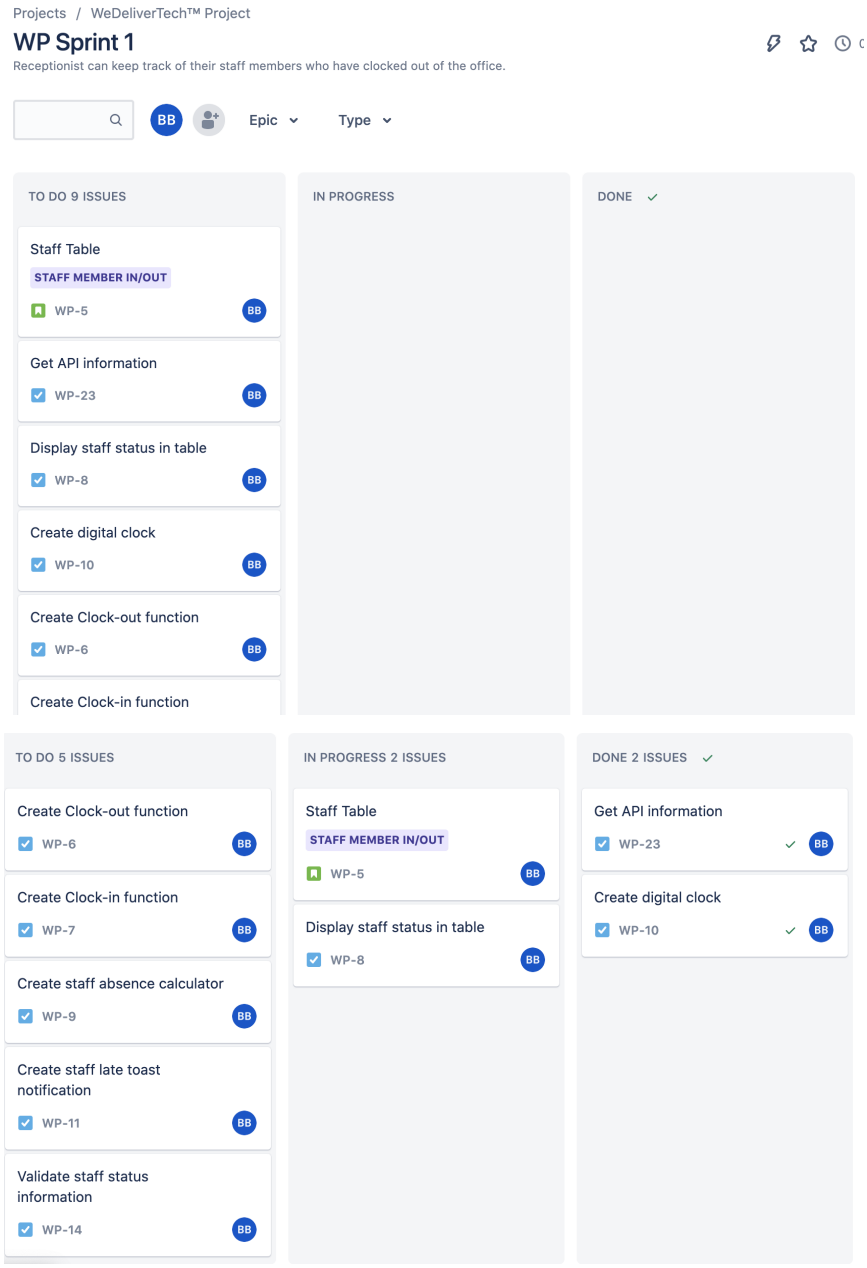


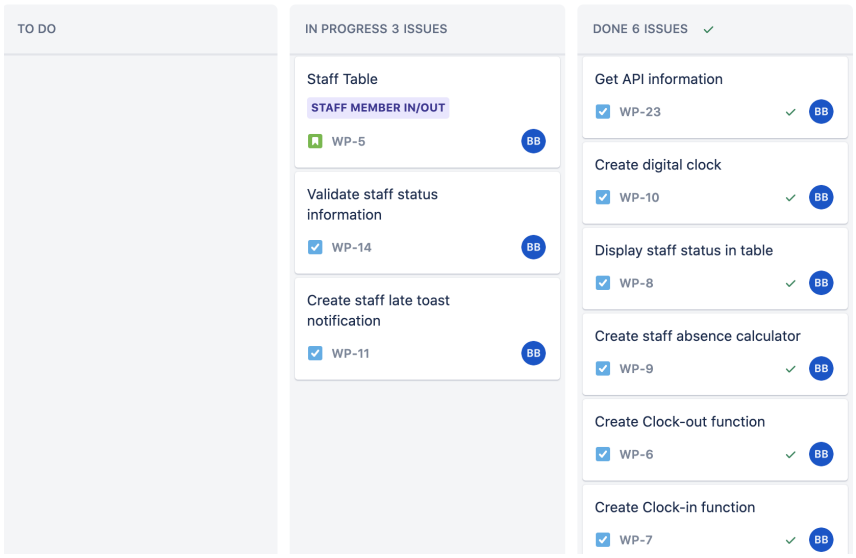
Once the staff members were present in the table it was important to create the clock-out function. As I created this function I made sure to complete validations, so the receptionist couldn't continue to update a staff member's staff if they entered the incorrect information.

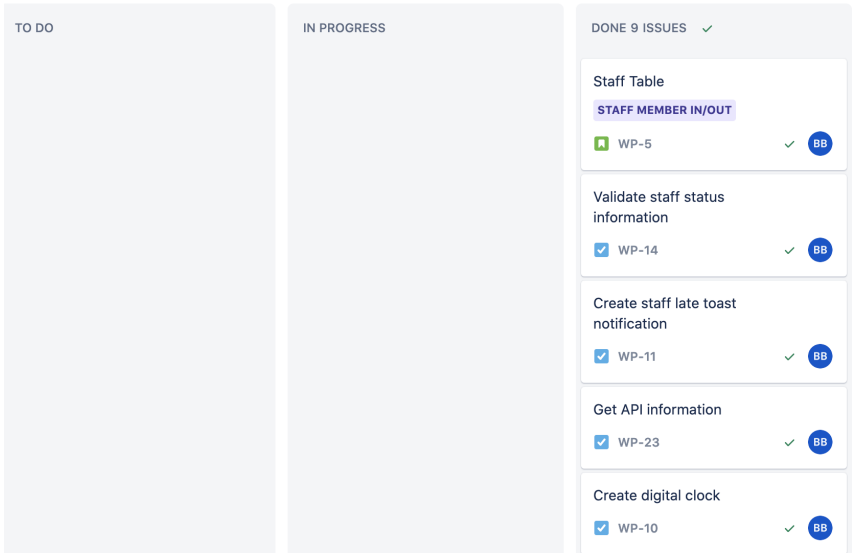


While it did take time to perfect the staff member table it worked out well completing each issue in a certain logical order. I chose to completely finish the staff table before moving onto the delivery tables in order to keep track of the issue I'm working on, and so I wasn't overwhelmed with all of the issues I needed to solve. The last issue I solved during Sprint 1 was activating a toast notification if a staff member was late.

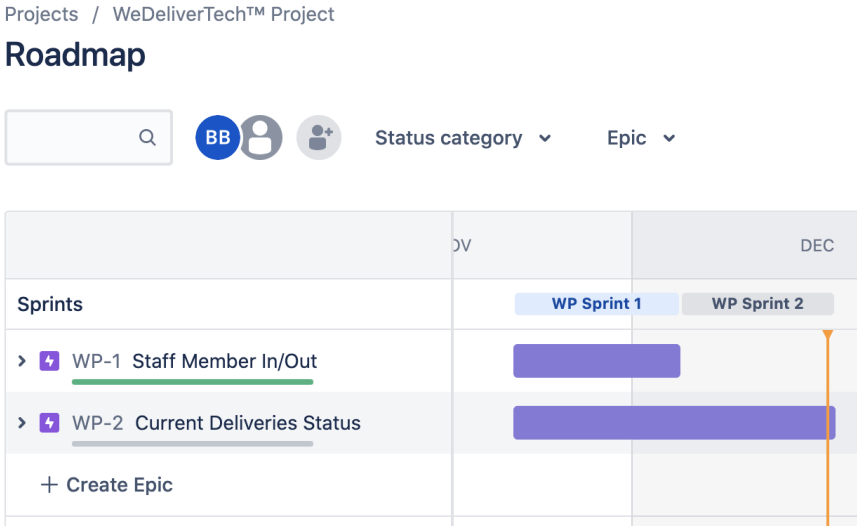
I have attached screenshots of the board during sprint 1 below:



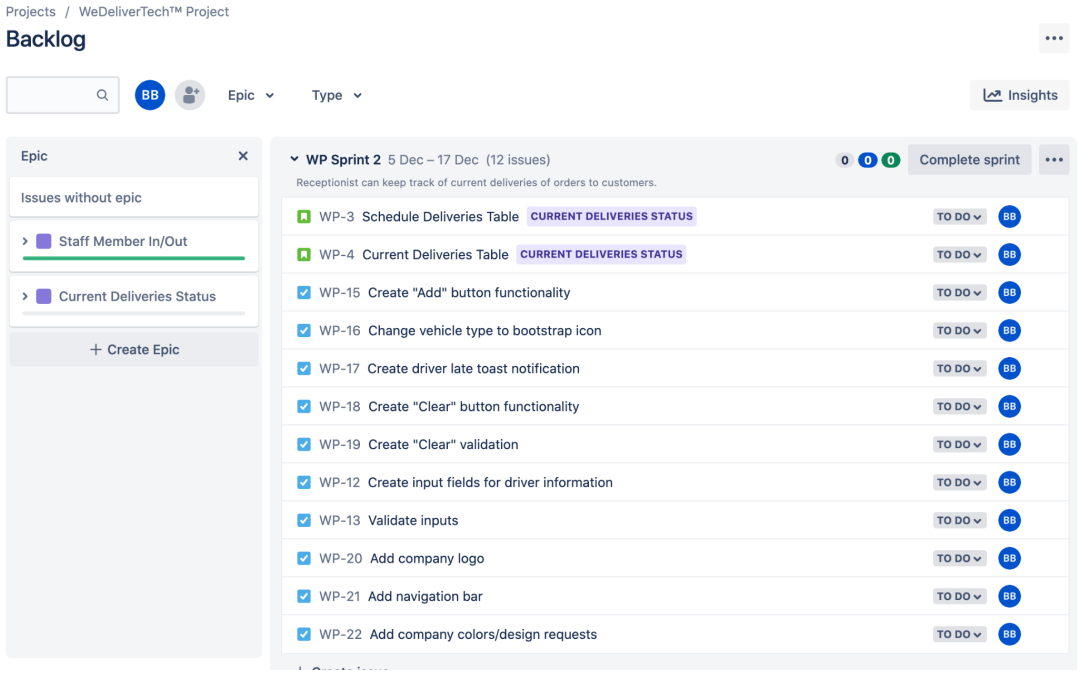

TO DO
IN PROGRESS 3 ISSUES
Staff Table
STAFF MEMBER IN/OUT
WP-5
BB
Validate staff status information
WP-14
BB
Create staff late toast notification
WP-11
BB
DONE 6 ISSUES
Get API information
WP-23
BB
Create digital clock
WP-10
BB
Display staff status in table
WP-8
BB
Create staff absence calculator
WP-9
BB
Create Clock-out function
WP-6
BB
Create Clock-in function
WP-7
BB

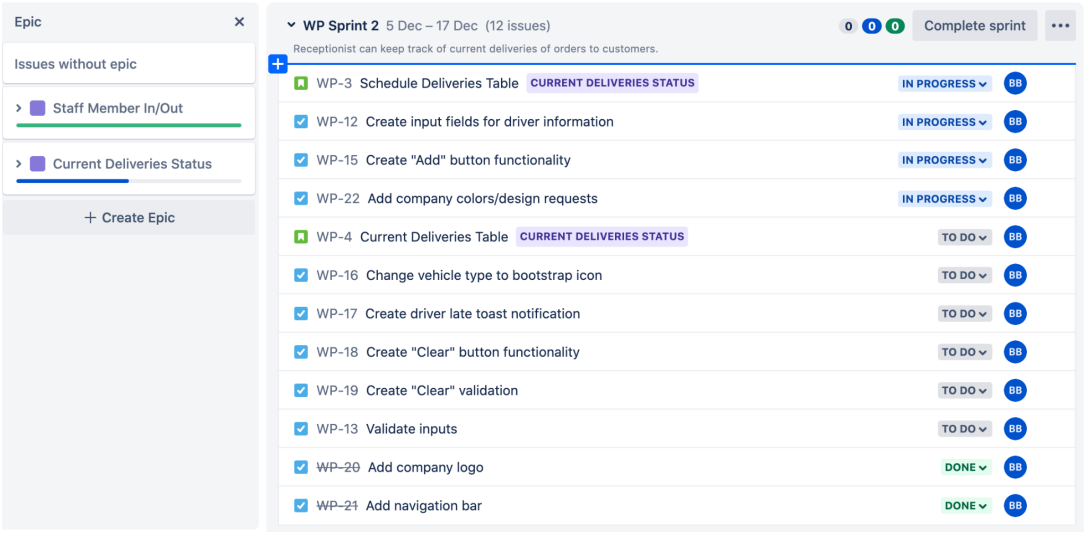

TO DO
IN PROGRESS
DONE 9 ISSUES
Staff Table
STAFF MEMBER IN/OUT
WP-5
BB
Validate staff status information
WP-14
BB
Create staff late toast notification
WP-11
BB
Get API information
WP-23
BB
Create digital clock
WP-10
BB

I have attached the Roadmap after sprint 1 was completed below:

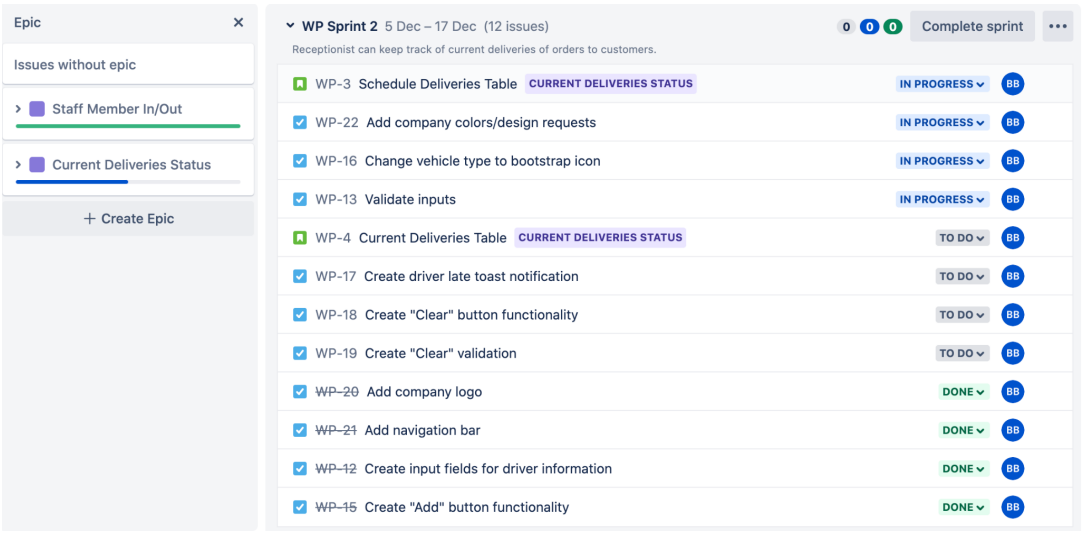


I have attached screenshots of the backlog during sprint 2 below:

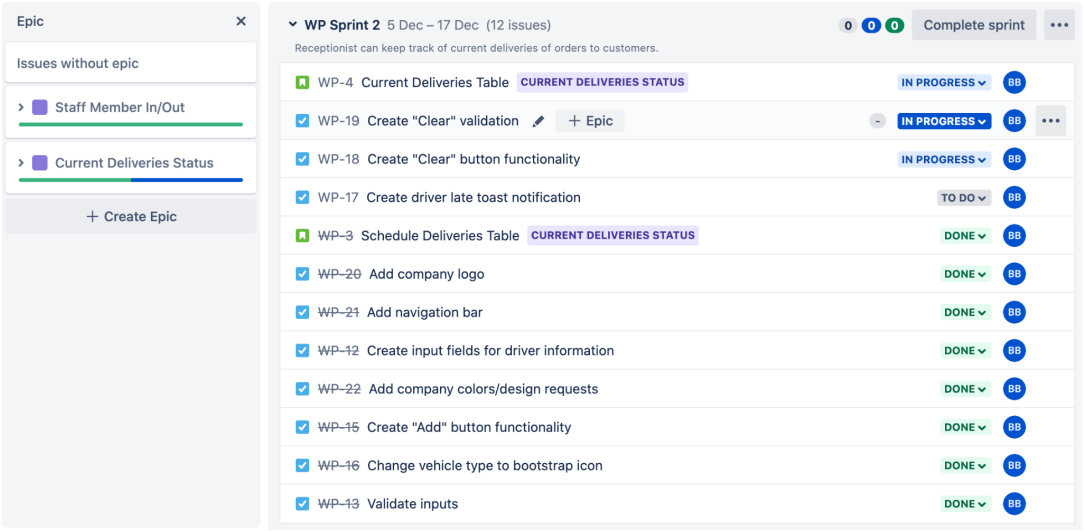




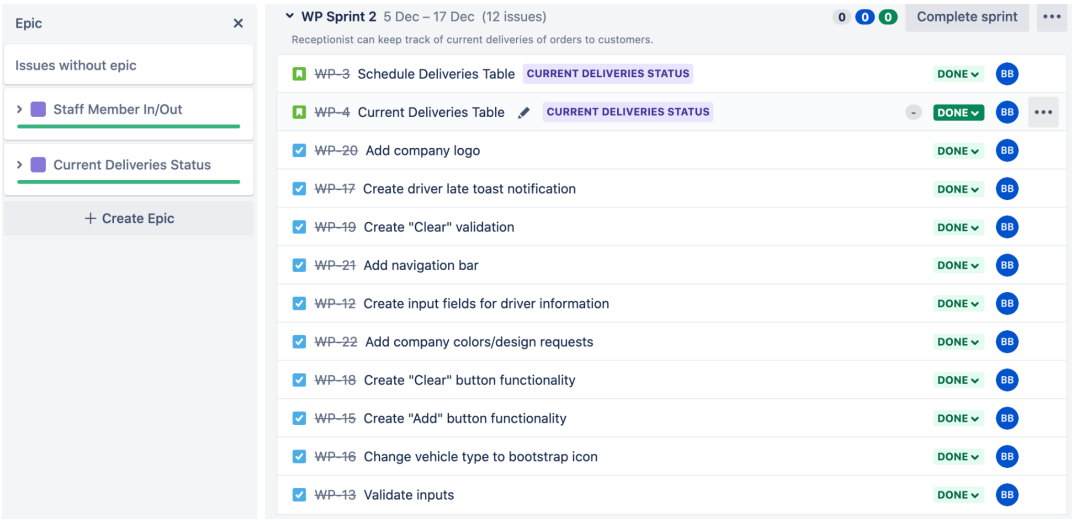
The validated inputs from the "Schedule Delivery" table were essential for the "Delivery Board" to be updated, so the first issues I started with were to enter information into input fields, validate the information, then add it to the "Delivery Board." Adding the company logo and navigation bar were independent tasks which were required to satisfy the customer, so I completed them before starting the tables.



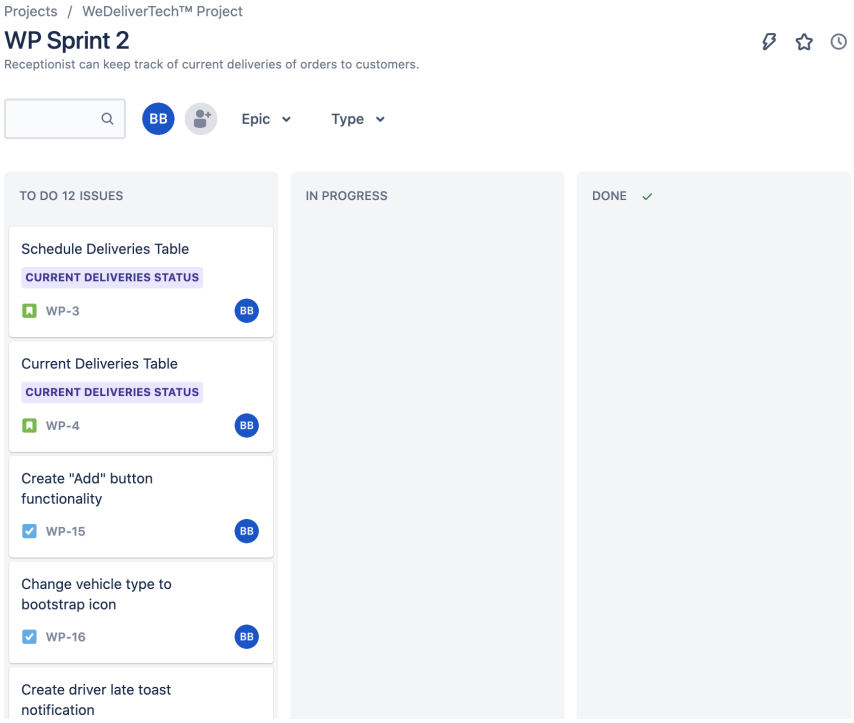
Once I completed the "Add" functionality it was important I perfected the validation so the receptionist could not enter incorrect information into the "Delivery Board." The customer requested the vehicle type entered as a Bootstrap icon, so I started to complete this task. I didn't complete issue WP-16 until I validated the user had correctly entered "motorcycle" or "car."

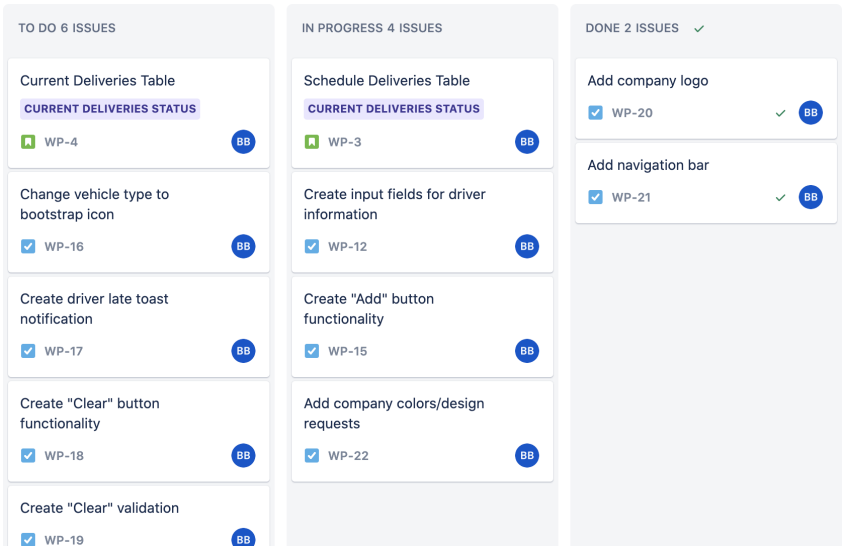


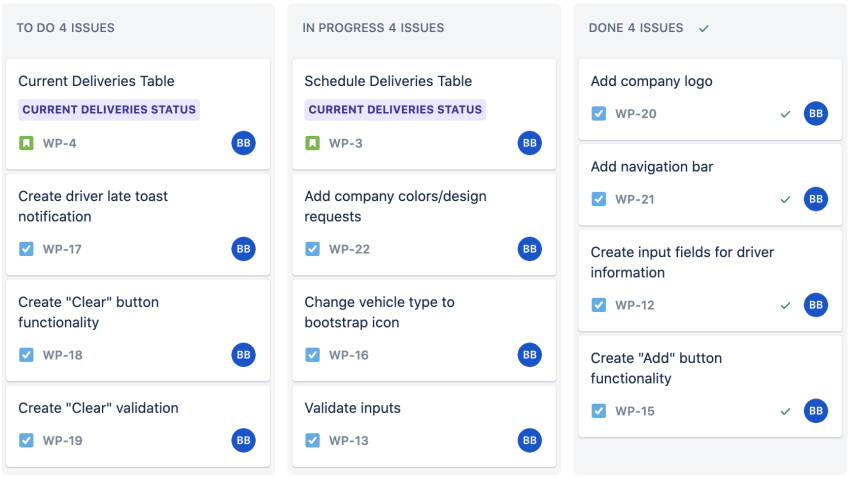
Once the delivery drivers were successfully and correctly added to the "Delivery Board" I was able to begin working on the "Clear" functionality, and lastly the late driver toast notification.

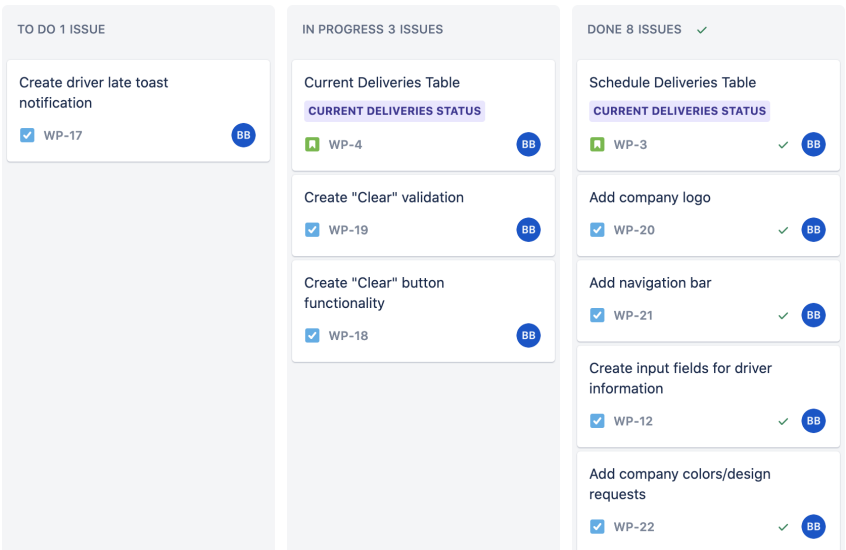


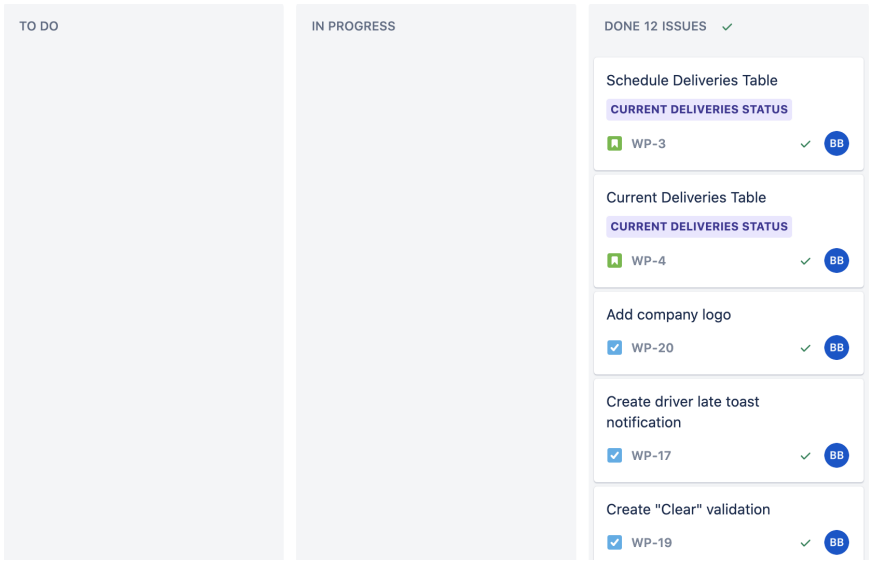
I have attached screenshots of the board during sprint 2 below:



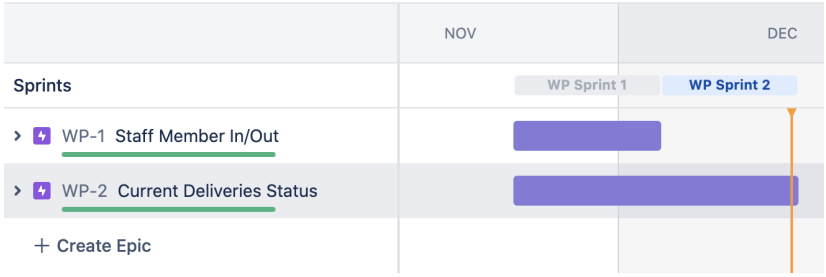

TO DO 6 ISSUES
Current Deliveries Table
CURRENT DELIVERIES STATUS
WP-4
BB
Change vehicle type to bootstrap icon
WP-16
BB
Create driver late toast notification
WP-17
BB
Create "Clear" button functionality
WP-18
BB
Create "Clear" validation
WP-19
BB
IN PROGRESS 4 ISSUES
Schedule Deliveries Table
CURRENT DELIVERIES STATUS
WP-3
BB
Create input fields for driver information
WP-12
BB
Create "Add" button functionality
WP-15
BB
Add company colors/design requests
WP-22
BB
DONE 2 ISSUES
Add company logo
WP-20
BB
Add navigation bar
WP-21
BB


TO DO 4 ISSUES
Current Deliveries Table
CURRENT DELIVERIES STATUS
WP-4
BB
Create driver late toast notification
WP-17
BB
Create "Clear" button functionality
WP-18
BB
Create "Clear" validation
WP-19
BB
IN PROGRESS 4 ISSUES
Schedule Deliveries Table
CURRENT DELIVERIES STATUS
WP-3
BB
Add company colors/design requests
WP-22
BB
Change vehicle type to bootstrap icon
WP-16
BB
Validate inputs
WP-13
BB
DONE 4 ISSUES
Add company logo
WP-20
BB
Add navigation bar
WP-21
BB
Create input fields for driver information
WP-12
BB
Create "Add" button functionality
WP-15
BB


TO DO 1 ISSUE
Create driver late toast notification
WP-17
BB
IN PROGRESS 3 ISSUES
Current Deliveries Table
CURRENT DELIVERIES STATUS
WP-4
BB
Create "Clear" validation
WP-19
BB
Create "Clear" button functionality
WP-18
BB
DONE 8 ISSUES
Schedule Deliveries Table
CURRENT DELIVERIES STATUS
WP-3
BB
Add company logo
WP-20
BB
Add navigation bar
WP-21
BB
Create input fields for driver information
WP-12
BB
Add company colors/design requests
WP-22
BB



Roadmap at completion:



Sprint Reflection:

While deciding which issue to complete first I thought about what would be necessary to solve the following issues. For example, in order to check a staff member out of the office they must be present in the first place, so I should populate the staff table with staff members first. Once the staff members are present (their default status is “Checked-In”) there must be a way to check them out before I can check them in again. Additionally, in order to validate an input, there must be an input to be validated. This is how I decided which order to solve the issues in.

Project Management:

I attempted to use Object Oriented Programming in order to write code that is reusable, easy to maintain, further develop, and test. This is something that challenged me, because I could easily update a staff member in the table, but it was an extra step to update the object as well. I always focused on small issues instead of the entire task in order to efficiently complete the entire project. Overall I felt I progressed smoothly through the project.