

# Work Placement - POS App Development

## Introduction

My PRJ701 work placement was with e-commerce platform Airsquare, a Nelson based content management system for hosting e-commerce websites. The focus of the placement was to participate in the development of a Point of Sale (POS) application for tablets designed to integrate into Airsquare's system with the goals of gaining experience, contacts and references while working on a commercial project.

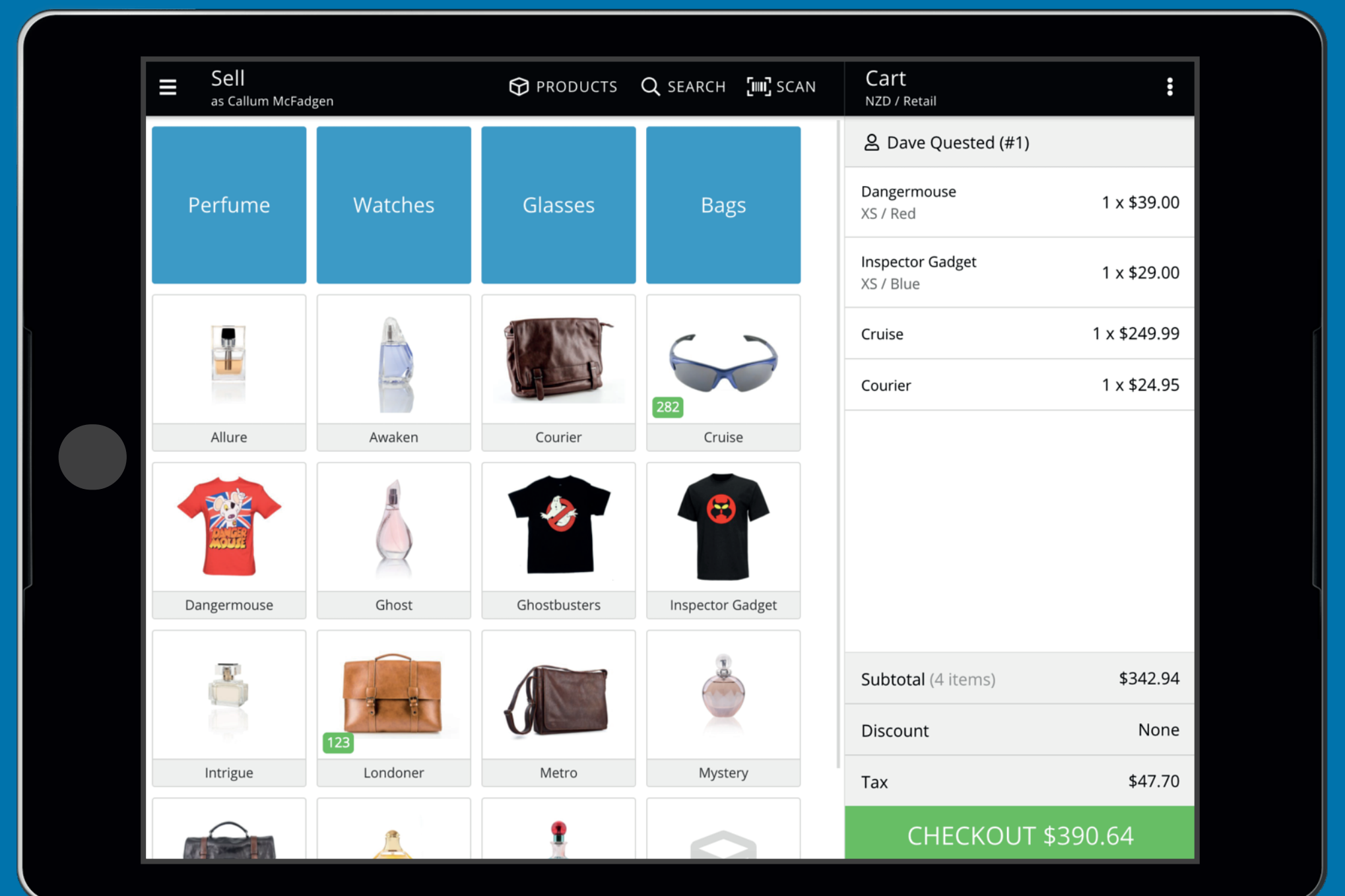
## Application Overview

The primary purpose of the application was to allow users to create and process orders in a physical environment with real-time data synchronisation between their e-commerce websites and the point of sale.

Feature parity needed to be maintained with a pre-existing POS application and the code base would be used to produce applications for Android and IOS by using the cross-platform Ionic Framework.

The Client-Server model was used to integrate with the Airsquare system to access user's data, but the application also needed to maintain a local state in case of network failure.

The application's functionality was generated by the interaction between Pusher.js, Angular and Dexie.js.



## Pusher.js

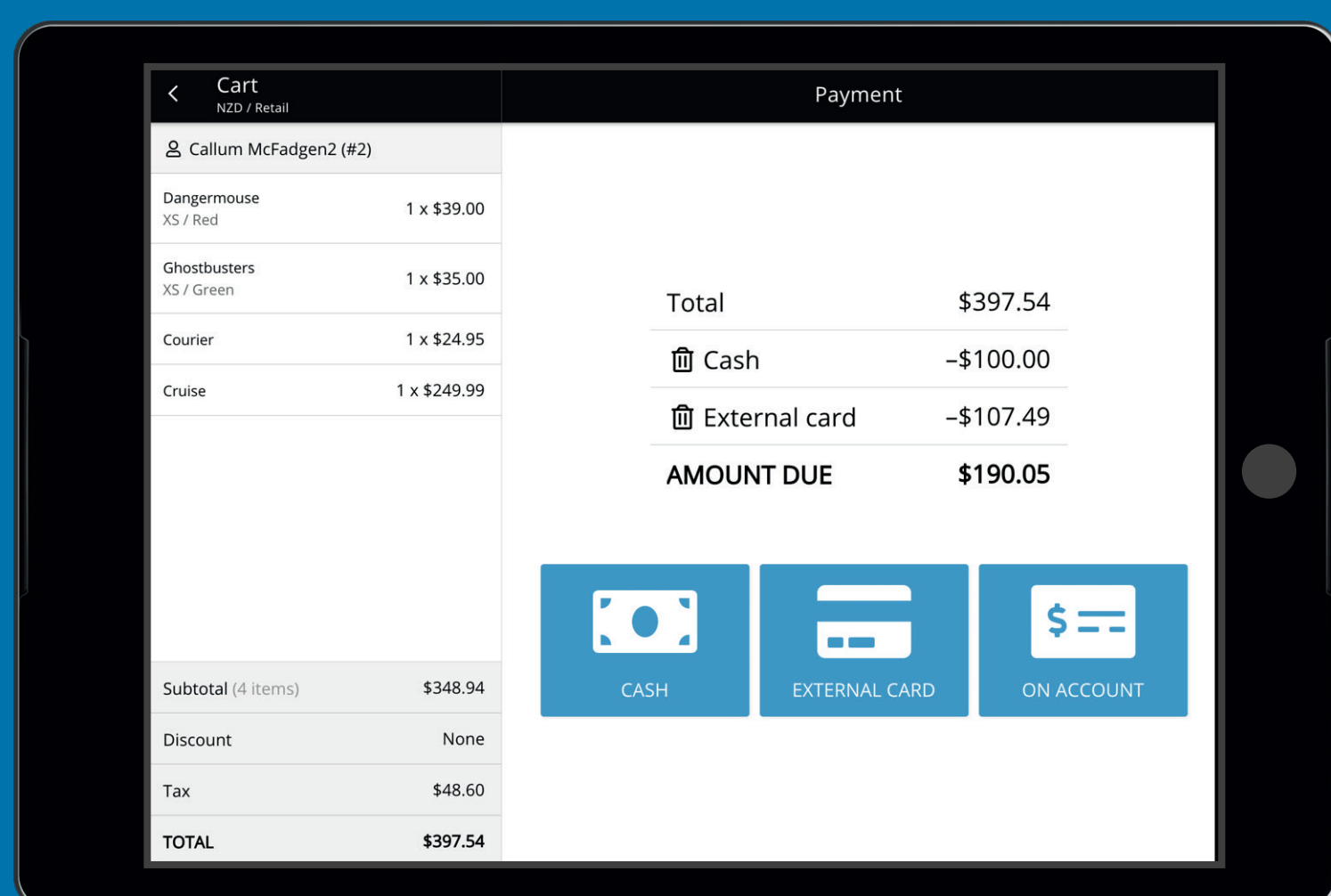
Provides a persistent real-time WebSocket connection between client applications and servers that uses a Publish-subscribe pattern to distribute data updates.

## Dexie.js

Lightweight IndexedDB wrapper supporting SQL queries and transactional integrity with highly performant bulk operations.

## Angular

Open-source TypeScript based web application framework that uses observables and promises to manage asynchronous functionality.



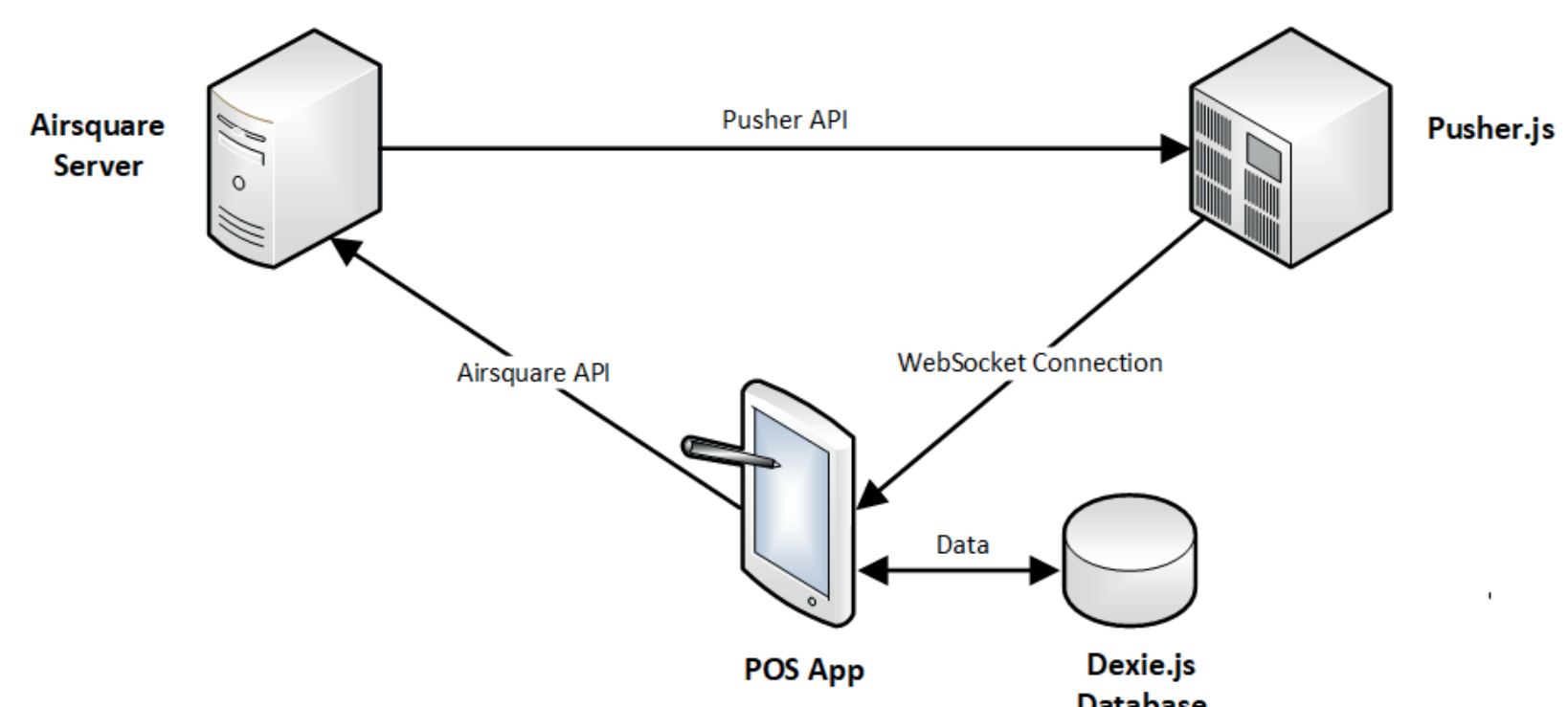
## Core Functionality

Combining the functionality of the three technologies produced a very stable application that performed well during initial testing.

Pusher.js kept the POS app in sync with the data stored in the Airsquare system by maintaining the connection between application and server and publishing updates whenever the data was changed.

When an update arrived, Angular provided promises that were used to manage the API and the data would be processed and stored into the applications Dexie.js database where it could be accessed very efficiently by other parts of the application.

Angular Observables were then used to publish the updated data throughout the application to keep all of the application views up to date.



## Conclusion

At the conclusion of work placement, the application was a functional prototype, able to complete the sale of products. It was significantly more performant than the previous POS application and was ready for user testing. The work placement provided valuable experience and led to the offer of a contract with Airsquare working on the continued development of the POS application.