Part 4 Interim Report

# 4. Prototype Development

## 4.1. Introduction

The description of how the system design choices were implemented as part of the prototyping process combined with some unexpected encounters that were met. The link to the project can be found at the following link: <https://github.com/WilliamCareySemiColon/DietaryAssistanceSystem> .

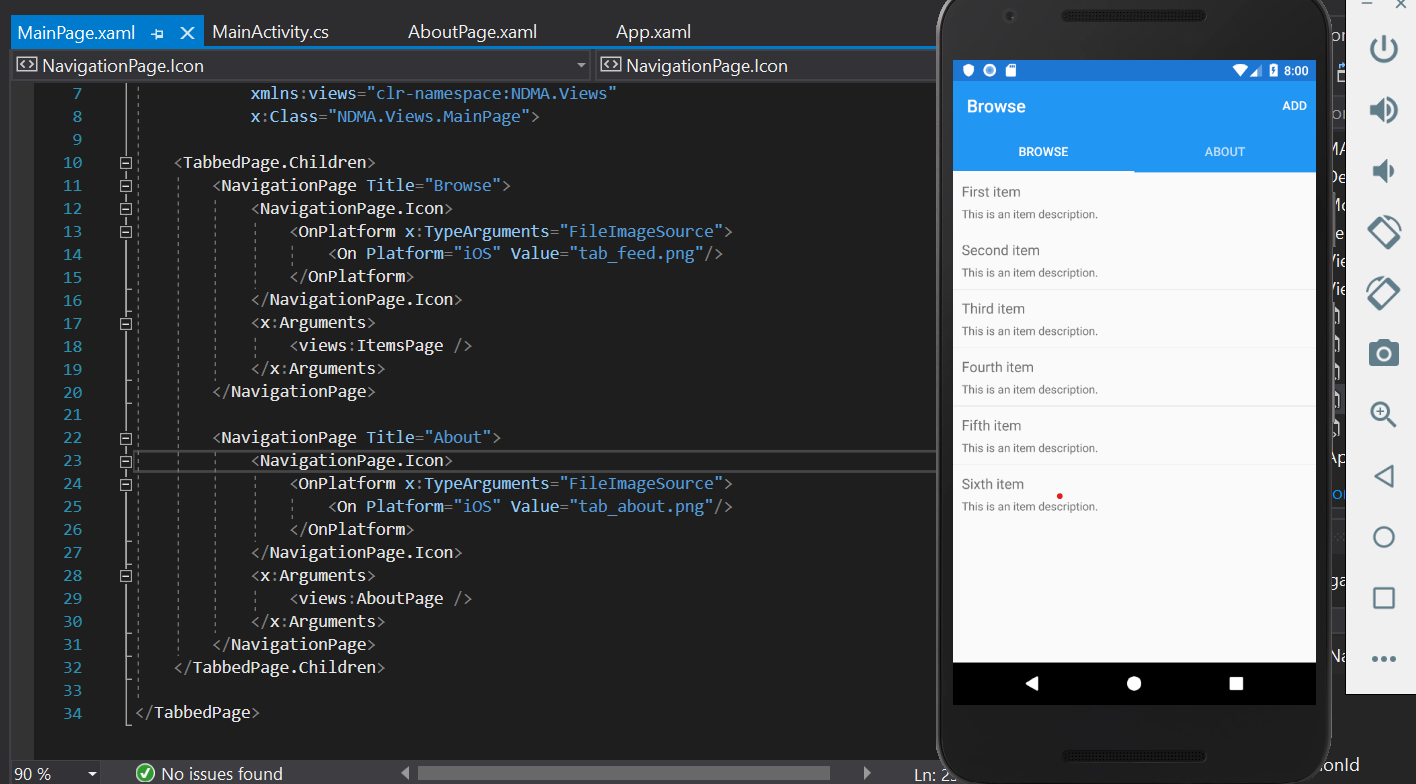
## 4.2. Prototype Development

There were numerous steps taken for the work on the prototype. The first server was to set up the version control so we can ensure we manage the project properly in iterations. GitHub was the choice to work with due to the familiarity of the technology, a reliable web hosting service for managing projects and combined well with its seamless integration into the visual studio IDE with the workload. After setting up the version control properly, the project could then be started on within the local version of the repository.

The next step was to connect everything together. From ensuring the visual studio knew there was a version control tracking the project to connecting to the Azure Cloud Services from within the environment. Finally, creating a way to store the data locally in case the device at the time does not have internet connection. For the prototype itself, a demonstration with the localhost database would be acceptable as a temporary solution.

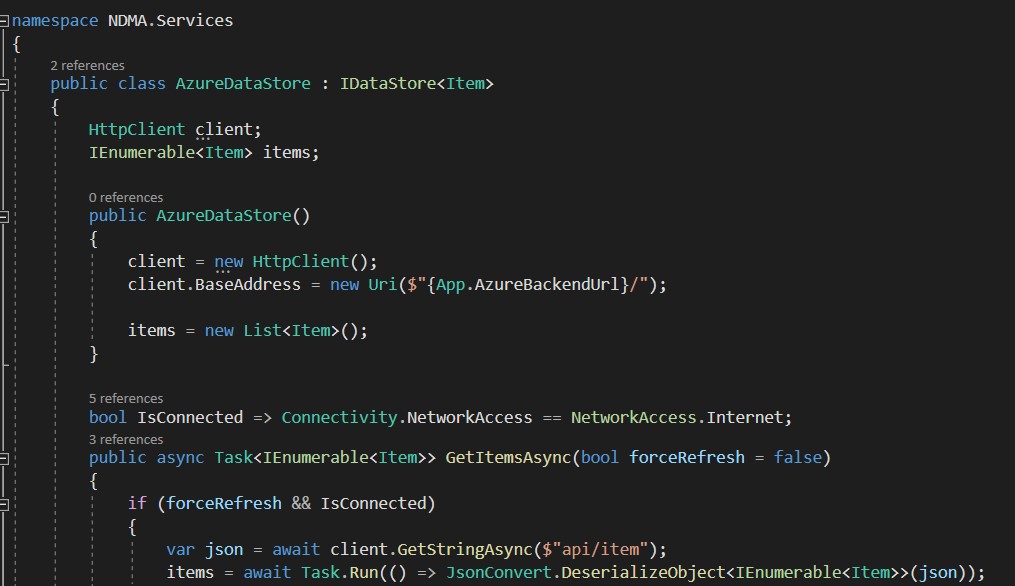
## 4.3. Front-End

For the front-end, eXtensible Markup Language (XML) was used. XML is a universally language that can be used across different environments platforms to display information in the desired way. It is like HTML in this way and is recommended by the World Wide Web Consortium (W3C). The prototype front-view was still in progress at the time of write-up.



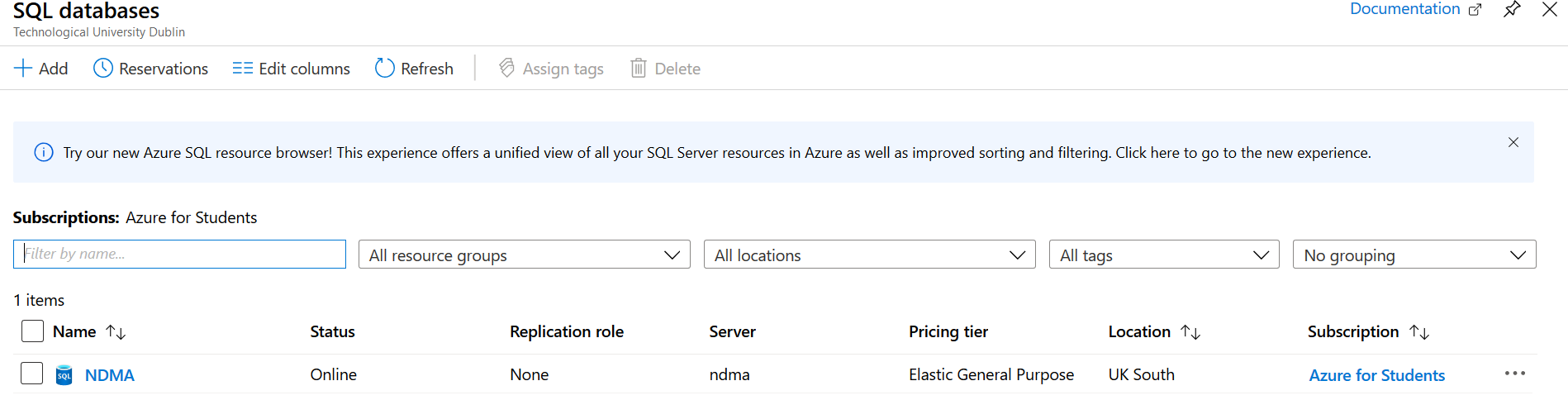
## 4.4. Middle-Tier

For the middleware, the Xamarin, .Net framework and C# language is the driver of the application. The Xamarin framework is the developer for mobile applications, which allows for the cross-platform development. The .Net framework separates the front-end from the backend, which is the primary reason of its existence. The C# programming language is the implementer of the frameworks. It provides the functionality to the front-end xml fields, implements the business logic and captures data provided to and from the backend. The main logic for the proof of concept is currently in progress.



## 4.5. Back-End

The database used for storage is both Azure Cloud SQL Database and SQLite database as the localhost. This is to provide both remote and local storage. Both provided issues as they had connectivity issues with the visual studio IDE, which took some time to figure this out.



## 4.6. Conclusions

This chapter delved into the implementation of the prototypes. From the time of recording drafts to the connectivity issues and further plans to update the design of the prototype. The plan of action was also decided.