Application Name: CourierService

Description: A console application that computes the total delivery cost for a package.

Tools/Tech used:

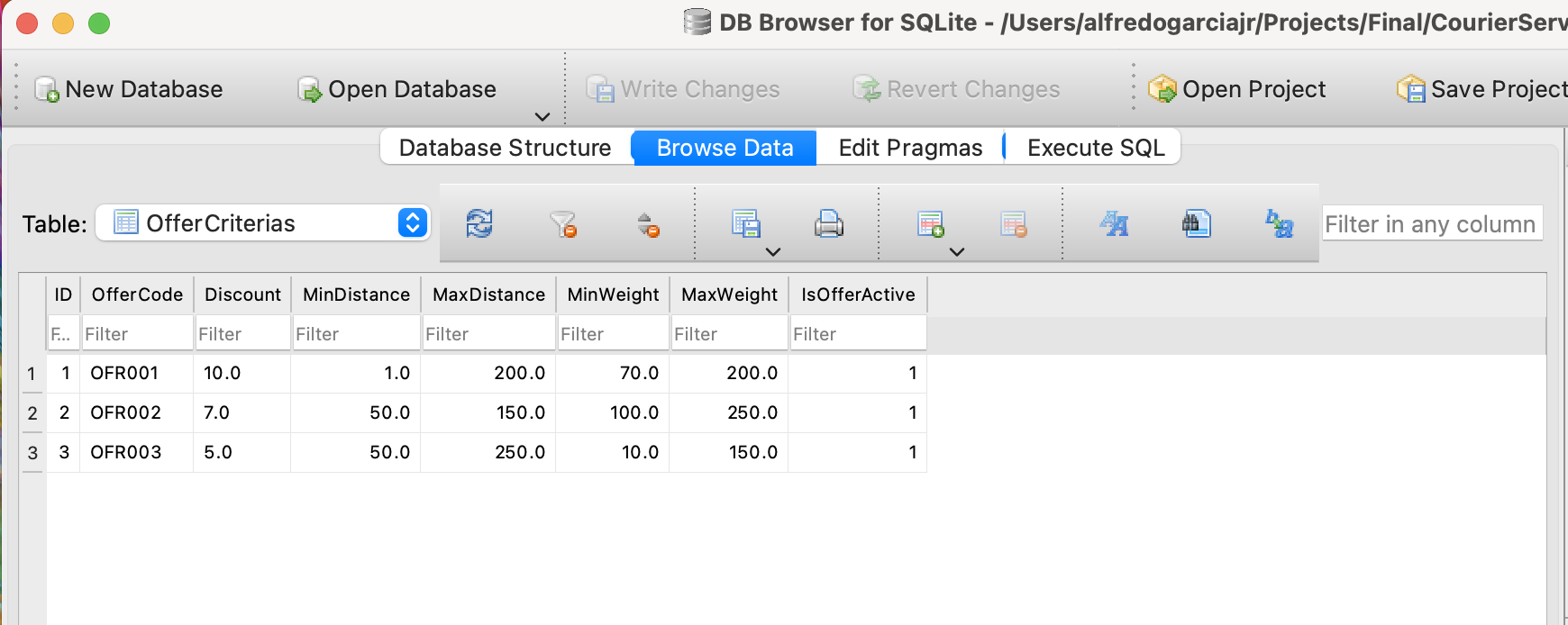
* Visual Studio for Mac 2019
* C# 8.0
* .Net Core 3.1
* Console Application Project
* EntityFramework
* SQL lite
* Git
* MS Test

Limitations:

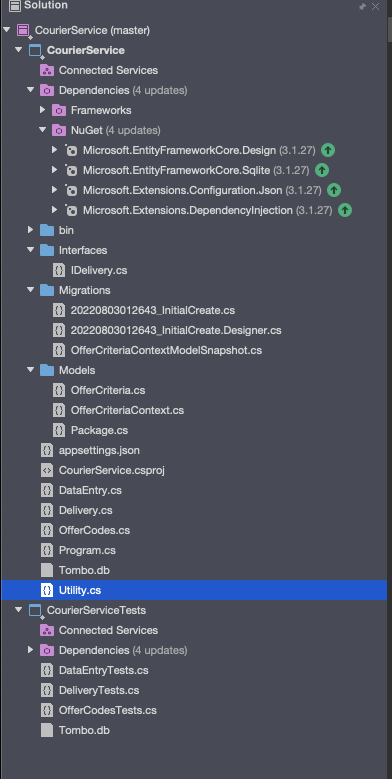
For this project demonstration, SQL Lite was used as a local DB attached or included in the application to hold or store the offer criteria for easy configuration and extendability. One of its limitation is that it’s case sensitive (for strings). For this application, we store the offer code, and so when a user inputs an offer code in the console, it has to match what’s stored in the database, otherwise the offer code will not be recognise or will be invalid.

For the distance and weight of the package, I set a limitation whereby the minimum value that can be entered is 1. So if a package’s weight is really less than 1 kg, or it’s distance is less than 1 km, I thought of suggesting to input 1 kg/km for simplicity.

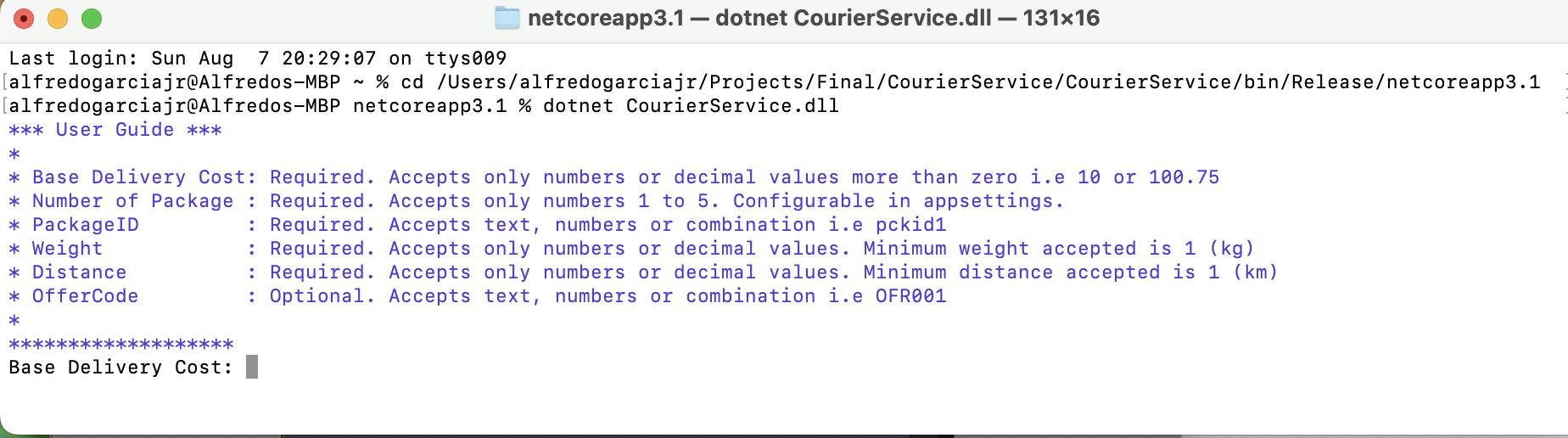
DB Browser for SQLite



Project

To run the app, we open a terminal or command window and go to the location of the dll of the app CourierService.dll which we need to execute using dotnet command:

Example:



Sample Input/Output:



Logic Flow

Program.cs

DataEntry.cs

Delivery.cs

Run()

EvaluateInput()

DisplayOutput()

CalculateDeliveryCost()

CalculateDiscount()

OfferCodes.cs

dbo.OfferCriteria