Introduction

The purpose of this application is to store and manage information about a given habit. The information stored informs of the date, quantity and location that the habit occurred.

This application uses CRUD functions which are an essential part of practically any app. The purpose of this project is as an introduction to basic CRUD options.

How It Works

The application has four main components:

The DatabaseManager Class – This class manages all database operations, it creates a database and a table if they don't already exist. It also includes all the methods for inserting, deleting and updating habit occurrences as required by the user.

The OptionsManager Class – This class show on screen the different actions that the user can take it updates the selected option highlighting it and informs the UserInterface Class which option has been selected by the user.

The LogViewManager Class – This class shows the habits table con screen its job is to update the table selecting one of the records as the active record by highlighting it. If the table is to large to fit inside its grid it permits the scrolling of the table. It also updates the table when a record is added, modified or deleted.

The UserInterface Class – This is the heart of the application. When it is instantiated its first job is to instantiate the _logviewManager used to show on screen the contents of the habits table. To really get it going the MainMenu method has to be called, this method instantiates the OptionsManager used to select the different operations that the user might want to do. The next step is to generate on screen the user interface using the GenerateUserInterface method this draws all the tables and the writes the main, and columns titles. The MainMenu then combines both the _logViewManager and the OptionsManager to control the users actions on both the table view and the menu options. When the user has selected an option the main menu passes control to one of the action methods used to add, modify, delete records from the table, or exit the application.

Requirements

- This is an application where you'll log occurrences of a habit.
- This habit can't be tracked by time (ex. hours of sleep), only by quantity (ex. number of water glasses a day)
- Users need to be able to input the date of the occurrence of the habit
- The application should store and retrieve data from a real database
- When the application starts, it should create a sqlite database, if one isn't present.
- It should also create a table in the database, where the habit will be logged.
- The users should be able to insert, delete, update and view their logged habit.
- You should handle all possible errors so that the application never crashes.

- You can only interact with the database using ADO.NET. You can’t use mappers such as Entity Framework or Dapper.
- Follow the DRY Principle, and avoid code repetition.
- Your project needs to contain a Read Me file where you'll explain how your app works.

Challenges

Bellow is a list of the challenges that I have introduced to the project

- The use of parametrised queries to make the application more secure
- Incorporate a seeding function to populate the table. Specially helpful for testing purposes
- As a personal challenge I’ve incorporated a grid table presentation that permits scrolling through the habits table.

New Knowledge Acquired

The most important understanding for me personally during this project has been to start getting the right mindset when using and creating objects and their instantiation. I begin to see the amazing and empowering possibilities that OOP offers.

I will also mention the use of CRUD functions for the first time. I understand perfectly what they do, but I must say that I’ll have to program many many many more applications before I can type from memory any of these functions.