Professional Software Engineering

Assignment 4

Deadline: 15th of January

The aim of this assignment is to test your understanding of LINQ and C#'s Entity Framework. The assignment is split into two parts, which revolve around analyzing an existing web application (Task 1), and programming a console application that can serve as a simplified *backend* [1] for this web application (Task 2). Keep in mind that these tasks should be done in the order presented, i.e. you may have difficulties completing Task 2 before Task 1.

Once you are done with both tasks, make sure you zip your folder before uploading. If you are using windows as your OS, you can do so by following the steps below.

1. Right click on your folder

2. Hover over "Send to"

3. Select "Compressed (zipped) folder"

Task 1 Analysis

The web application is KittySplit (`https://www.kittysplit.com/en/`), an expense (or bill) tracking and sharing app. Before starting, create a new Kitty and test out the options and functionality of the web app. The aim here is to develop an understanding of how all the components interact with one another.

Note: The app is free and registration is not required.

Your tasks are as follows:

- Identify the **five most important** use-cases and describe them (be brief).

- Choose **one** of the use-cases you have identified and describe how the user would interact with the app. Here, you are expected to write a user-story, i.e. how a user would interact with the app (an activity diagram can be helpful but the focus should be on the user's interaction with the app)

Note: Make sure your solution file is a .pdf file (not a word or a .txt file).

Task 2 Programming

Your second task is to write a simple program that is able to support your use-cases.

The program should be separated into the three layers described below.

1. Console Application, containing the following:

   - Main class and method

   - Five functions that follow the use-cases you identified in Task 1. These functions should be called in your main method.

2. Domain/Business logic (classes, interfaces etc.)

---

[1] An application's backend is a term encompassing all the functionality that a user cannot see, e.g. databases, server-side logic, APIs etc.

3. Database Abstraction (context, config etc.)

Some useful tips:

- The program should be able to modify the database, i.e. adding or removing entities, based on your use-cases.

- The use of the Entity Framework is **mandatory**.

- You can ignore the money balance functionality. Focus only on the data storage.