Week 8: Project work 1

This week, I chose one of the remaining tasks from week 8, the issue here is that it is brand new, one I can't just reproduce on the spot. The goal here is to create a subteam with members that are already created. One tough part of the issue is to get in touch with the coworker that worked on it first. It may seem easy but it wasn't at first and I was planning on redoing it all over. After some contacting, we were able to put together the data base and the creation of the team members.

Code showcase

I had to update my coworker's class for the team members as I had to implement a new parametter: If they could be assigned to the team or not.

```
using SQLite;



namespace UNDAC_App.Models
{
    public class TeamMember
    {
        [PrimaryKey, AutoIncrement]
        public int Id { get; set; }
        public string Name { get; set; }
        public string Skill { get; set; }
        public string CurrentAssignment { get; set; }
        public string TeamMemberInfo { get { return "Skill: " + Skill + " \n" +
"Current Assignment: " + CurrentAssignment; } }
        public string Dispo { get; set; }
    }
}
```

The new parametter is "Dispo". It stands for "Disponibility" and in further code we will set it to "Add to team" or to "unavailable" in case they already are in the team.

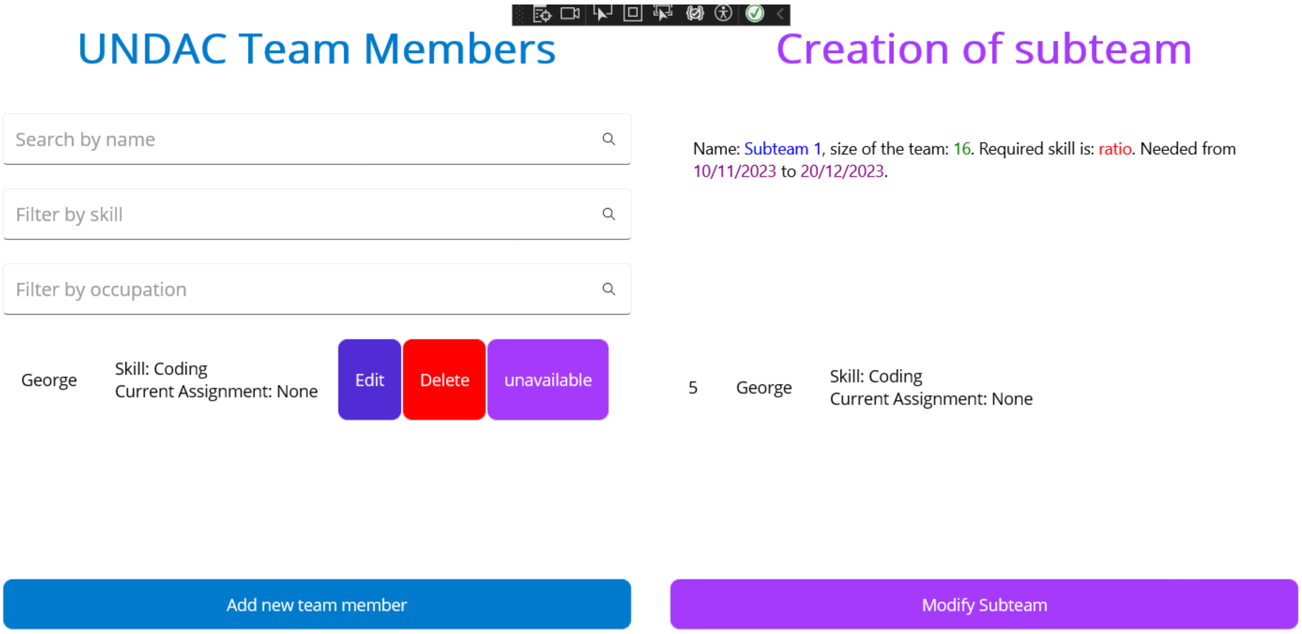
```
public void AddTeamMember(string name, string skill, string currAssignment)
{
    var newTeamMember = new TeamMember { Name = name, Skill = skill,
CurrentAssignment = currAssignment, Dispo="Add to team" };
    dbConnection.Insert(newTeamMember);
}
```

Another functionality that was in the requirements for vaildation of the task was the ability to have different criteras for the Subteam. I created a new class for it and a new manager.

```
public void UpdateSubteam(int id, string name, string size, string skill,
string start, string end)
{
    var subteam = dbConnection.Table<Subteam>().FirstOrDefault(b => b.Id ==
id);
    if (subteam != null)
    {
        subteam.Name = name;
        subteam.Size = size;
        subteam.RequiredSkill = skill;
        subteam.Start = start;
        subteam.End = end;
        dbConnection.Update(subteam);
    }
    else
    {
        var newSubteam = new Subteam { Name = name, Size = size, RequiredSkill
= skill, Start = start, End = end };
        dbConnection.Insert(newSubteam);
    }
}
```

Here is the modification of a Subteam. As there is only one, I made it so that if no subteam was in the list, the button would create one instead.

After talking with my coworker, I had modified their page for creation of a member to both create a member and move them to the subteam, the result looks like this:



UNDAC Team Members

Creation of subteam

Name: Subteam 1, size of the team: 16. Required skill is: ratio. Needed from 10/11/2023 to 20/12/2023.

George — Skill: Coding — Current Assignment: None

5 — George — Skill: Coding — Current Assignment: None

Code review for my code

Link to the code review done for my code

The code review for my code is on the same pull request as last.

My coworker noticed that part of my code for the Subteam was not in a Manager.cs file. The code was working but dirty and did not match with what our team as done so far.

Code review I did

Link to the code review I did

Still on the same PR as last time. Here on my coworker's new code, I noticed something that is familiar to me, you can modify a vehicule by:

- by clicking the modify button.
- by clicking the element.
  Which is the same review they did about my code last time. After telling them they quickly fixed the issue, which is not making the code worse, but creating duplication has the code is the same but with another name "OnVehicleEquipmentTapped" and "OnVehicleEquipmentModify".

Reflective section

Testing is still an issue but doing it by hand was easy this time. I can create members with an already working code and I can manipulate them to test my code manualy.

For next time I wish I could start something from scratches and not have to work after somebody else, as we may note approach the issues the same way. It would make developping the issue easier.

I think I need to be more mature when it comes to team working.