Week 8: Project work 1

This week we were asigned new tasks for our MAUI App. These tasks were harder than week 3's, For exemple, for week 3 I had to create a page that allowed me to create an element with an Id and a name, modify it and delete it. The task I chose for week 8 is the same, but now the element as a type, which can also be modified, and we can sort items by type and look for them by names. The task I chose was: As an UNDAC Team Leader I want to view team alerts so that I can take appropriate action.

Code showcase

It's the third year since I've started coding and I have been teached abour the reusability of code. For week 8's code, I have copied and pasted every page I have done for week 3's work and started from there.

```
using SQLite;



namespace UNDAC_App.Models
{
    public class Alert
    {
        [PrimaryKey, AutoIncrement]
        public int Id { get; set; }

        public string Name { get; set; }

        public string Type { get; set; }

        public string Color { get; set; }
    }
}
```

This is my Alert model. There is a color variable for I wanted to have different displays depending on what type of alert was on screen, thus making better readability for user.

```
private void OnFilterByTypeClicked(object sender, EventArgs e)
{
    // Create a list of available alert types
    string[] alertTypes = { "new", "ongoing", "reviewed", "cleared", "blocked",
"Cancel" };

    // Show an action sheet for selecting the alert type
    DisplayActionSheet("Filter by Alert Type", null, null,
alertTypes).ContinueWith(task =>
    {
        if (task.Result == "Cancel")
        {
            var filteredAlerts = AlertTypes;
            AlertTypesListView.ItemsSource = filteredAlerts;
        }
        else if (alertTypes.Contains(task.Result))
        {
            string selectedType = task.Result;

            // Filter alerts by the selected type and update the ListView
            var filteredAlerts = AlertTypes.Where(alert => alert.Type ==
selectedType).ToList();
            AlertTypesListView.ItemsSource = filteredAlerts;
        }
    }, TaskScheduler.FromCurrentSynchronizationContext());
}
```

This code is the code that makes the filter button work. We first create the list of the types of alerts, plus the cancel button. Xarmin.Forms has a `csharp DisplayActionSheet` function that creates an onscreen list with clickable elements. Then with the `csharp ContinueWith(task =>`function, we can wright the code that will play directly when the choice is made.

```
<StackLayout Orientation="Horizontal" Spacing="8">
    <Label Text="{Binding Name}" VerticalOptions="CenterAndExpand"
TextColor="Black" />
    <Label Text="{Binding Color}" VerticalOptions="CenterAndExpand"
TextColor="Black" />
    <Label Text="{Binding Type}" VerticalOptions="CenterAndExpand" TextColor="
{Binding Color}"/>
    <Button Text="Modify" Clicked="OnAlertTypeModify" CommandParameter="
{Binding .}" BackgroundColor="BurlyWood" TextColor="White" WidthRequest="80"
HeightRequest="40" HorizontalOptions="EndAndExpand"/>
    <Button Text="Delete" Clicked="OnAlertTypeDelete" CommandParameter="
{Binding .}" BackgroundColor="Coral" TextColor="White" WidthRequest="80"
HeightRequest="40" HorizontalOptions="End"/>
</StackLayout>
```

Here is the xaml for showing on screen an element. Here I am showing it's name, it's colour and it's type. I am having an issue with the type not showing in the selected colour but being shown black. Here, the code from week 3's barely changed, I added 2 lines of code and this part was changed for good.

For me, the reusability of code is very importent. It makes work faster and easier. It requires good naming of functions and variables and good code ethique. If these rules are respected, new code can be created from already existing one. The issue is that the code has to be modified, so I can't use the same functions and can't apply DRY. If the functions were the same, I would've found a way to apply DRY.

Code review for my code

Link to the code review done for my code

Here is the code review that has been done for my code. Here, a coworker pointed out that two functions were doing the same thing. Originally my code contained 2 ways to modify an element:

- by clicking the modify button.
- by clicking the element.
  At first I thought it would be nice to have an evenement occure when you click the element but I didn't know what to put. I didn't delete the clicked element but I did put it as comment. If I have to update the code with, for example, a description of the Alert, I can make it so that it shows when you click the element. By keeping in mind this way of interacting with the app, I will be able to show info to the user without showing yet another button in front of the element.

Code review I did

Link to the code review I did

I reviewed a coworker's work and noticed that one acceptence criteria was not respected. As their task was ressemblant to mine, I proposed to him that we have a call so that I show him how I did mine. The missing criteria is the search bar. I use it to search for issues by name and they use it to search for Partners.

A coworker forgetting a criteria would be alarming in a real company situation. A good practice I will onwards adopt is to look at what was the acceptence criteria before judging the code. If I did not find their code to be a bit shorter then what I had expected, I wouldn't have looked at the criteria and would have authorised the pull request.

Reflective section

This week I have realised that I'm not the only one having issues with code testing. A lot of my coworkers also have errors in their code on bits that work for others etc.

In a team situation, I think that overviewing a codereview, being to lite on the checking or doing it very fast can be a common problem. I myself am part of the problem since I almost authorised the PR.

Conclusion

This week again I omitted the code testing. I appear to not comprehend how it works and it keeps on generating errors in my code. For next week I would like to understand it more and be more efficient in code reviewing and not let any mistake slide. I don't think I have issues with coding for now as the code produced was similar to what I had already done and we were multiple coworkers on the same type of tasks, so we could help each other in practicals.