

**University of Zakho**  
**Faculty of Science**  
**DEPARTMENT OF COMPUTER SCIENCES**



**Task Management**

**Student's Name: -**

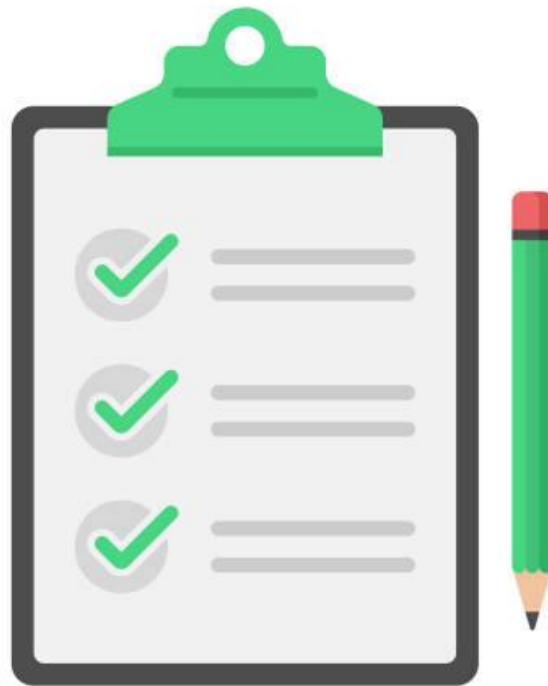
1. Ahmed Omer
2. Abdulrahman Younis
3. Lolav Ihsan
4. Mina Dlovan

Supervisor: Mr. Yousif

Academic Year  
2022-2023

## **Table of Contain**

- 1- What is task management
- 2- Why we use task management
- 3- Goals of project
- 4- Explain we program and what we use
- 5- Conclusion



## **What is task management?**

Task management is **the process of monitoring your project's tasks through their various stages from start to finish**. This involves actively making decisions for your tasks to accommodate changes that can occur real-time, with your end goal being the successful completion of your tasks.

## **Why we use task management?**

Task management systems are used to manage tasks, track time, and easily collaborate with the team. These are efficient for individuals, teams, and organizations to help them complete tasks efficiently without missing any deadlines.

## **Goal of Project?**

Goal of the project is we make this app to help us to write daily goal and to be remind to us after we finish the daily goal we can remove it this help do not forget goal us.

## Explain the program and what we use:

### MainPage.xaml:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TaskManager.MainPage" Title="TaskManagement"
             BackgroundColor="Black">
    <StackLayout>
        <Frame BackgroundColor="Black" Padding="25" CornerRadius="0">
            <Label Text=" Your personal assistant" TextColor="Yellow" FontSize="30"
                VerticalOptions="Center" HorizontalOptions="CenterAndExpand" BackgroundColor="Black" />
        </Frame>
        <Label Text="Make small hobits and big goals achievable"
            VerticalOptions="Center" HorizontalOptions="Center" FontSize="20" TextColor="White"/>
        <Image Source="abc.jpg" />
        <Button x:Name="mybutton" Text="Get startd" BackgroundColor="Red" FontSize="25"
            TextColor="White" Clicked="Button_Clicked" VerticalOptions="CenterAndExpand" />
    </StackLayout>
</ContentPage>
```

### Explain code:

In this code we have inside **<StackLayout>** block have **<Frame>** The job of this Frame is The Xamarin.Forms class is a layout used to wrap a view with a border that can be configured with color, shadow, and other options. Frames are commonly used to create borders around controls but can be used to create more complex UI. we have inside this frame label this label is have text and position of the text is **Center** and we make another label, and we have Image after the we make **button** the job of this button is after we top that go to another page the name of page is **Page1.xaml**

The code of button to go the another page :

### MainPage.xaml.cs:

```
Navigation.PushModalAsync(new NavigationPage(newPage1()));
```

We use navigationPge to go another page.

Result of this code after we made :



## Page1.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TaskManager.Page1"
             BackgroundColor="BlanchedAlmond">

    <StackLayout Orientation="Vertical">

        <Grid Padding="15" HorizontalOptions="Center" VerticalOptions="Center"
              ColumnDefinitions="4*,*" Grid.Row="1">
            <Button Text="Add " FontSize="13" TextColor="Black" BackgroundColor="Red"
                  Grid.Column="1" Command="{Binding AddCommand}" />
            <Entry x:Name="Task" Placeholder="Enter your task" Grid.Column="0"
                  FontSize="18" Text="{Binding text}" />
        </Grid>
        <ListView x:Name="MyList" SeparatorVisibility="None" HasUnevenRows="True"
              ItemsSource="{Binding CollectionsList}" Footer="">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <Grid Margin="10" BackgroundColor="Red" >
                            <Grid.ColumnDefinitions>
                                <ColumnDefinition Width="Auto"/>
                                <ColumnDefinition Width="*" />
                            </Grid.ColumnDefinitions>
                            <Label VerticalOptions="CenterAndExpand" Text="{Binding
Name}" TextColor="Black" FontSize="20"/>
                            <Button Command="{Binding
Path=BindingContext.DeleteCommand,Source={x:Reference MyList}}"
CommandParameter="{Binding .}" Grid.Column="1" Padding="30,10" Text="Delete"
                                HorizontalOptions="EndAndExpand"
                                BackgroundColor="Black" TextColor="White"/>
                        </Grid>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
        </ListView>

    </StackLayout>

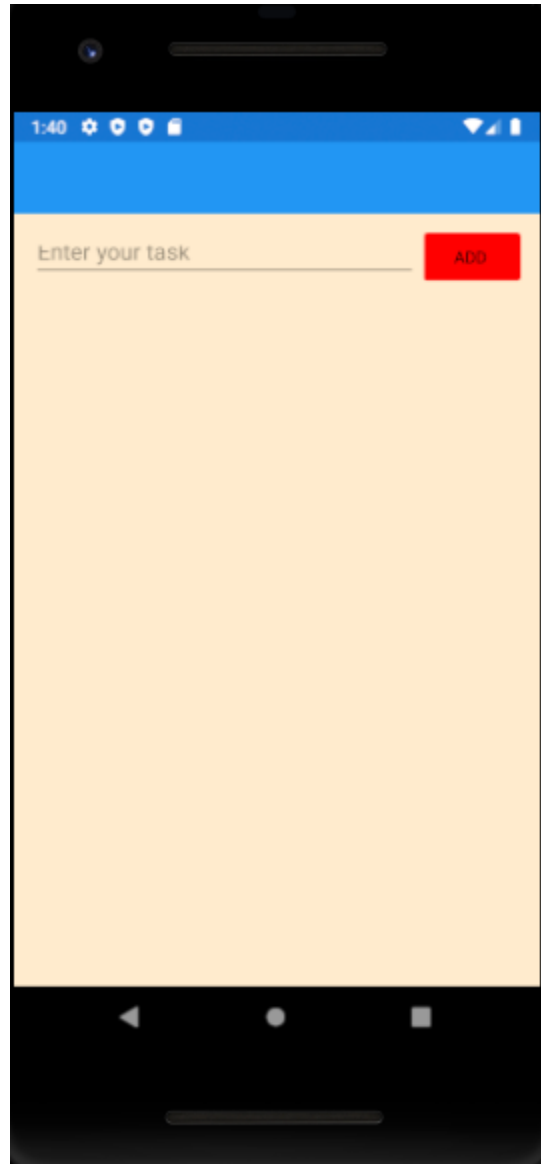
</ContentPage>
```

Explain code:

In this page we use **<StackLayout>** inside the **<StackLayout>** block we use **<Grid>**

Inside we have have 2 thing first we have **Entry** job of entry is to write Your task inside it and Second thing we have **Button** job of button is to Add your task inside the **ListView** and we Use **ListView** to Display our Tasks.

Result of Page1.xaml:



We have 2 class to **Biding** button and **ListView** Of **Page1.xaml** First class we have

```
using System;
using System.Collections.Generic;
using System.Text;

namespace TaskManager
{
    public class OffersModel
    {
        public string Name { get; set; }
    }
}
```

**OffersModel.cs** We made this class to use in another we explain inside the class

**MainPageViewModel.cs** the of this page :

```
public class MainPageViewModel : BindableObject
{
    private ObservableCollection<OffersModel> _CollectionsList;
    public ObservableCollection<OffersModel> CollectionsList
    {
        get { return _CollectionsList; }
        set
        {
            _CollectionsList = value;
            OnPropertyChanged();
        }
    }
    string text1;
    public string text { get => text1;
        set
        {
            text1 = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteCommand { get; }
    public ICommand AddCommand { get; }
    public MainPageViewModel()
    {
        CollectionsList = new ObservableCollection<OffersModel>();
        DeleteCommand = new Command(OnDeleteTapped);
        AddCommand = new Command(AddItmes);
    }
    private void AddItmes(object obj)
    {
        OffersModel offersModel = new OffersModel();
    }
}
```



```

        offersModel.Name = text;
        CollectionsList.Add(offersModel);
    }
    private void OnDeleteTapped(object obj)
    {
        var content = obj as OffersModel;
        CollectionsList.Remove(content);
    }

```

---

```

private ObservableCollection<OffersModel> _CollectionsList;
public ObservableCollection<OffersModel> CollectionsList
{
    get { return _CollectionsList; }
    set
    {
        _CollectionsList = value;
        OnPropertyChanged();
    }
}

```

We made dynamic collection of objects of a given type of OffersModel  
To give source to listView .

---

```

string text1;
public string text { get => text1;
    set
    {
        text1 = value;
        OnPropertyChanged();
    }
}

```

In this code we get the text of entry to add into the listview after press into  
button.

```

public ICommand DeleteCommand { get; }
public ICommand AddCommand { get; }
public MainPageViewModel()
{
    CollectionsList = new ObservableCollection<OffersModel>();
    DeleteCommand = new Command(OnDeleteTapped);
    AddCommand = new Command(AddItmes);
}
private void AddItmes(object obj)
{
    OffersModel offersModel = new OffersModel();
    offersModel.Name = text;
    CollectionsList.Add(offersModel);
}
private void OnDeleteTapped(object obj)
{
    var content = obj as OffersModel;
    CollectionsList.Remove(content);
}

```

This line of code explain we made 2 table **DeleteCommand** and **AddCommand** to Add and delete item into the ListView.

## **Conclusion**

Task management includes planning, analyzing, evaluating, and reporting about a particular task's progress. It is an important aspect of the management of a project because it helps to follow every task thoroughly.